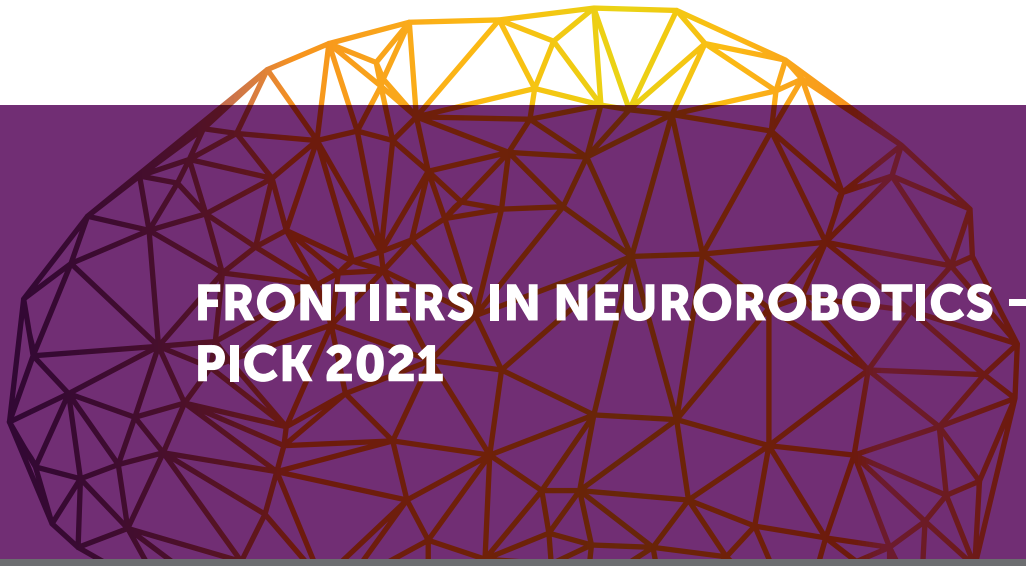
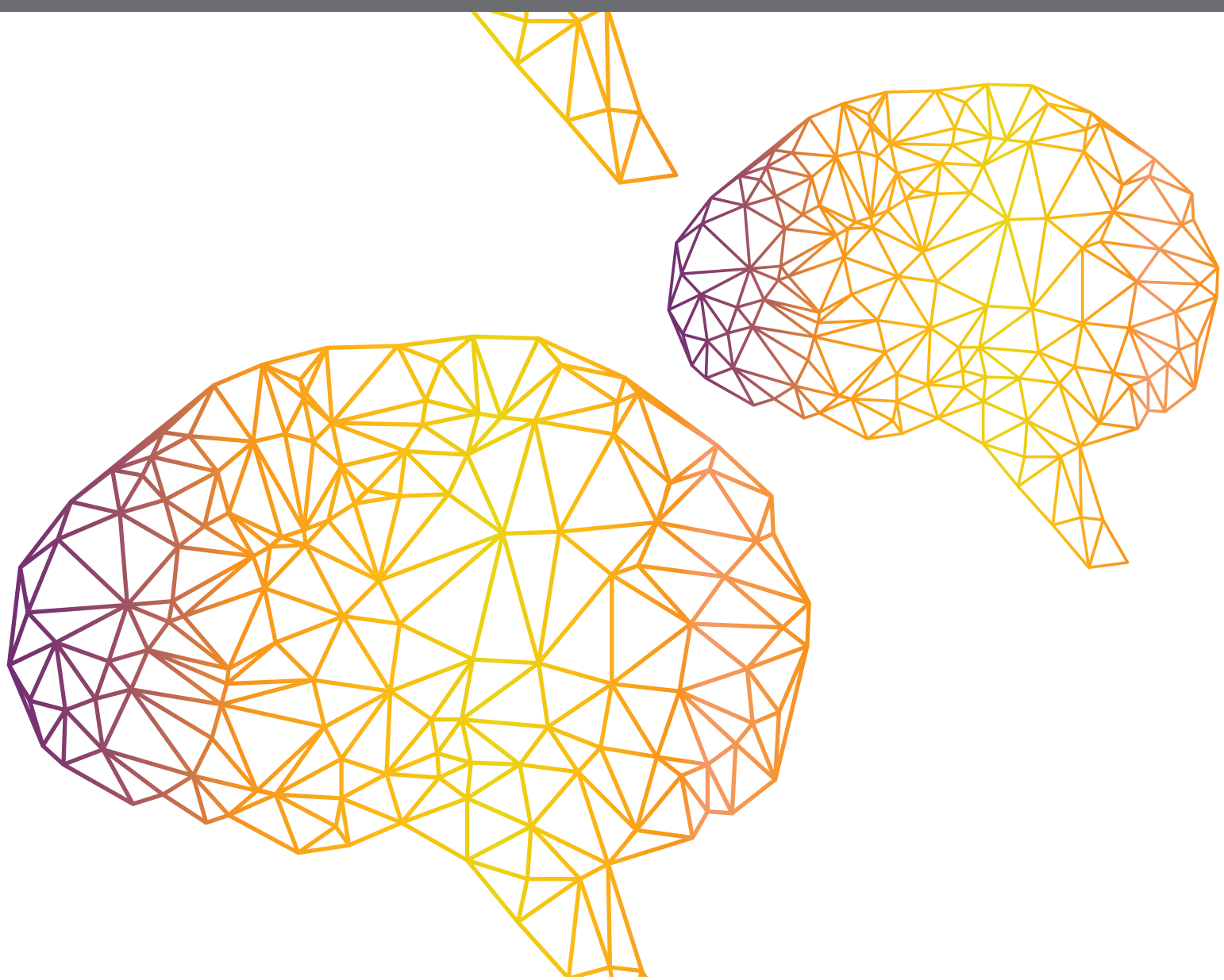


FRONTIERS IN NEUROROBOTICS – EDITOR'S PICK 2021

EDITED BY: Florian Röhrbein
PUBLISHED IN: Frontiers in Neurorobotics



frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-88966-898-4
DOI 10.3389/978-2-88966-898-4

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: frontiersin.org/about/contact

FRONTIERS IN NEUROROBOTICS – EDITOR'S PICK 2021

Topic Editor:
Florian Röhrbein, Independent Researcher, Germany

Citation: Röhrbein, F., ed. (2021). Frontiers in Neurorobotics – Editor's Pick 2021. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88966-898-4

Table of Contents

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 08 January 2019
doi: 10.3389/fnbot.2018.00085

Tracking People in a Mobile Robot From 2D LIDAR Scans Using Full Convolutional Neural Networks for Security in Cluttered Environments

Ángel Manuel Guerrero-Higueras[1]*, *Claudia Álvarez-Aparicio*[1], *María Carmen Calvo Olivera*[1,2], *Francisco J. Rodríguez-Lera*[3], *Camino Fernández-Llamas*[1], *Francisco Martín Rico*[4] *and Vicente Matellán*[1,2]

[1] Grupo de Robótica, Universidad de León, León, Spain, [2] Supercomputación Castilla y León (SCAYLE), León, Spain, [3] Computer Science and Communications Research Unit, University of Luxembourg, Luxembourg City, Luxembourg, [4] Robotics Lab, Universidad Rey Juan Carlos, Madrid, Spain

Tracking people has many applications, such as security or safe use of robots. Many onboard systems are based on Laser Imaging Detection and Ranging (LIDAR) sensors. Tracking peoples' legs using only information from a 2D LIDAR scanner in a mobile robot is a challenging problem because many legs can be present in an indoor environment, there are frequent occlusions and self-occlusions, many items in the environment such as table legs or columns could resemble legs as a result of the limited information provided by two-dimensional LIDAR usually mounted at knee height in mobile robots, etc. On the other hand, LIDAR sensors are affordable in terms of the acquisition price and processing requirements. In this article, we describe a tool named PeTra based on an off-line trained full Convolutional Neural Network capable of tracking pairs of legs in a cluttered environment. We describe the characteristics of the system proposed and evaluate its accuracy using a dataset from a public repository. Results show that PeTra provides better accuracy than Leg Detector (LD), the standard solution for Robot Operating System (ROS)-based robots.

Keywords: convolutional networks, LIDAR, people tracking, robotics, cluttered environment

OPEN ACCESS

Edited by:
Hong Qiao,
University of Chinese Academy of Sciences (UCAS), China

Reviewed by:
Valeri Makarov,
Complutense University of Madrid, Spain
Thiago Teixeira Santos,
Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA), Brazil

***Correspondence:**
Ángel Manuel Guerrero-Higueras
am.guerrero@unileon.es

Received: *16 March 2018*
Accepted: *29 November 2018*
Published: *08 January 2019*

Citation:
Guerrero-Higueras ÁM, Álvarez-Aparicio C, Calvo Olivera MC, Rodríguez-Lera FJ, Fernández-Llamas C, Rico FM and Matellán V (2019) Tracking People in a Mobile Robot From 2D LIDAR Scans Using Full Convolutional Neural Networks for Security in Cluttered Environments. Front. Neurorobot. 12:85. doi: 10.3389/fnbot.2018.00085

1. INTRODUCTION

Detecting and tracking people are very useful capabilities for different systems, in particular for improving navigation in mobile robots and also to facilitate more socially acceptable robots, but also in security applications, for instance using biometric data (Ngo et al., 2015; Gavrilova et al., 2017) or safely using robotics platforms (Morante et al., 2015). There are many solutions in the literature that try to solve this problem using a multi-modal approach, typically with vision and range sensors (Arras et al., 2012), but these kinds of approaches are very expensive both from the point of view of the cost of the sensor and the computing capabilities needed for processing and integrating, and are more likely to generate contradictory information. For this reason, systems based only on range sensors are more desirable. Regarding the classifiers to process sensor data, Convolutional Neural Networks (CNNs) has emerged as a very popular solution (Long et al., 2015).

Laser Imaging Detection and Ranging (LIDAR) sensors are reliable and currently affordable range sensors that provide information about a dynamic environment at good rates ($\sim 20 - 30$ Hz) that can be processed in real-time, as each scan consists of an array of just a few 100 integers.

Usually, mobile robots mount LIDAR scanners in a low position ($\sim 30 - 50$ cm from the ground) to detect obstacles, which are used to build occupancy maps and to navigate. The information provided is an accurate estimate of the distance in precise angles (resolution of 0.5°). This means that objects such as tables or chair legs, trunks of plants, etc., may be easily confused with peoples' legs. It is also difficult to keep track of a particular person (pair of legs) in a crowded environment because many occlusions can happen.

Different solutions have been proposed previously in the literature to deal with the problem of tracking people using a 2D LIDAR scanner mounted on a mobile robot. Navigation in peopled, mapped, indoor environments has been recently reviewed (Rios-Martinez et al., 2015). The use of the geometric characteristics of human legs and the frequency and phase of walking motion have been tested (Lee et al., 2006), but cannot deal with partial occlusions, changes in peoples' speed, etc. In this article we are concerned with the specific problem of detecting pairs of legs (from a person) and being able to track them.

Early research (Schulz et al., 2003) applied Bayesian filtering to track different objects in the perceptual range of the robot to estimate the number of people in the current scan based on the number of moving local minima in the scan. Unfortunately, this supposes that people are continuously moving, giving poor results in cluttered environments (where the number of local minima is misleading).

Other research (Arras et al., 2012) proposed a solution based on AdaBoost to learn to detect individual legs. In a second level they proposed a multi-target tracking framework that uses leg observations to infer peoples' state of motion.

Another report (Leigh and Zhang, 2015) described a tracking method considering both legs, rather than individual ones. They proposed using a combination of Kalman filters to predict behaviors over consecutive scans and a Global Nearest Neighbor filter to solve the scan-to-scan data association problem. This solution has the drawback of using two different steps (prediction and association) that we are trying to solve with a one-step approach. Also, the authors acknowledge the problem of adapting their *ad-hoc* proposal. We propose that a more general system based on machine learning can be built.

Other researchers (Aguirre et al., 2014) have used Support Vector Machines to learn the different patterns of legs that can appear in robot surroundings corresponding to moving or still people. However, their approach is limited, tracking only a single person in a controlled scenario.

Other methods using several steps have been proposed. For instance, some have (Ondruska et al., 2016) deployed a Recurrent Neural Network (RNN) to filter an input stream of raw laser measurements in order to directly infer object locations, along with their identity in both visible and occluded areas. Others (Premebida et al., 2009) described a sensor fusion architecture. The fusion process occurs at the feature level, combining a LIDAR sensor and a camera for improving the detection system's reliability and accuracy. A different approach (Szarvas et al., 2006) presented a real-time pedestrian detection system utilizing a LIDAR-based object detector and CNN-based image classifier. The proposed method achieves a processing speed of over 10 frames/s speed by constraining the search space using the range information from the LIDAR.

Some solutions have also been developed for the Robot Operating System (ROS) framework (Quigley et al., 2009), the most popular framework for developing robotic applications. For instance, the *cob_people_perception* package allows to find leg-like patterns of laser scanner readings. This software is based on the LD approach and implementation. LD is the most popular package for tracking people by using a LIDAR sensor in ROS-based robots. It obtains incoming messages from the LIDAR scanner and uses trained data to classify the groups of laser records as possible legs. Detection is performed by a classifier using Random Trees implemented with the OpenCV API. However, LD has an important drawback; the project has not received continuous development and there is not a version of the software for the latest ROS versions.

Other ideas have been the use of heuristic knowledge, such as some research (Mashad Nemati et al., 2016) that proposes a technique that relies on the estimation of the reappearance event both in time and location. In the same way, a utility function to approximate and predict the trajectory of a walking partner has also been proposed (Morales et al., 2014).

In summary, we consider that learning techniques are a more general approach than heuristic or *ad-hoc* techniques for this problem. We also propose that a single shot approach can solve the problem in a more compact and consistent way than approaches based on several steps. Under this assumption, we propose a system based on CNNs developed by the Robotics Group at the University of León, named *PeTra*, for developing tracking systems based on LIDAR measurements. PeTra has been developed and tested on a mobile robot based on the ROS framework.

CNNs have been typically used on classification tasks, where the output to an image is a single class label (Lawrence et al., 1997; Krizhevsky et al., 2012; Simonyan and Zisserman, 2014). In our case, we want to label the pixels of an occupancy map (image), not the whole image. We have explored different alternatives and we have found that the ideas proposed in the U-Net architecture (Ronneberger et al., 2015) match with our requirements. The next section describes in depth the system built.

Evaluating neural networks requires a good validation dataset. Collecting and organizing a training set needs time and domain-specific knowledge. There is a large collection of robotic datasets available from various mobile robots, vehicles, or just handheld sensors, for instance the *Repository of robotics and computer vision datasets*[1] for Mobile Robot Programming Toolkit (MRPT). However, most may not be suitable for training neural networks. For PeTra's validation, a dataset known as *Range-based*

Abbreviations: CNN, Convolutional Neural Network; ERL, European Robotics league; INCIBE, Instituto Nacional de Ciberseguridad de España; LD, Leg Detector; LIDAR, Laser Imaging Detection and Ranging; MRPT, Mobile Robot Programming Toolkit; ReLU, Rectified Linear Unit; RNN, Recurrent Neural Network; ROS, Robot Operating System; RTLS, Real Time Location System; MA, Moving Average.

[1] https://www.mrpt.org/robotics_datasets

people tracker classifiers Benchmark[2] (RRID:SCR_01574) was compiled.

In order to evaluate PeTra accuracy, we have compared its results with the ones obtained by LD, the most popular package to track people by using a LIDAR sensor in ROS-based robots. Evaluation was conducted in an indoor mock-up apartment located at the Robotics Lab at the University of León (Spain). A mobile robot called Orbi-One, with an on-board LIDAR sensor, was used to track peoples' locations using either PeTra or LD.

The rest of the paper is organized as follows. The next section describes the system that was built, including a detailed explanation of the architecture of the CNN used. Materials and methods used to evaluate the accuracy of PeTra are described in section 2. Section 3 shows the results of PeTra performance compared to LD. Section 4 discusses the above results. Finally, our contribution and the next steps foreseen are presented at section 5.

2. MATERIALS AND METHODS

A set of experiments was carried out to evaluate PeTra accuracy. They were conducted in the indoor mock-up apartment at Leon@Home Testbed[3], a Certified Testbed[4] of the European Robotics league (ERL) located in the Robotics Lab at the University of León. Its main purpose is to benchmark service robots in a realistic home environment. **Figure 1**, left shows the apartment plan.

Orbi-One robot, shown in **Figure 1**, middle, with an on-board LIDAR sensor, was used to track people either using PeTra or LD under different scenarios that will be described in section 2.5. Peoples' location was ascertained using a commercial Real Time Location System (RTLS) based on radio beacons shown in **Figure 1**, right. The location calculated by the RTLS was used as ground-truth data to calculate the error of PeTra and LD systems.

In the Youtube channel of the Robotics Group of the University of León, readers can find a video demonstration[5] of PeTra and LD working as displayed on Rviz, a well-known tool for displaying robot information for the ROS framework.

Below, the main components of the experiment are described in depth as well as the methods used to evaluate accuracy.

2.1. Orbi-One Robot

Orbi-One is a service robot manufactured by Robotnik[6]. It accommodates sensors, such as a RGBD camera, a LIDAR sensor, and an inertial unit. It can operate a manipulator arm attached to its torso and it has a wheeled base for moving around the room. An Intel Core i7 CPU with 8 GB of RAM allows it to run the software to control the robot hardware. The software to control the robot hardware is based on a ROS framework.

[2]http://robotica.unileon.es/index.php/Benchmark_dataset_for_evaluation_of_range-based_people_tracker_classifiers_in_mobile_robots
[3]http://robotica.unileon.es/index.php/Testbed
[4]https://www.eu-robotics.net/robotics_league/erl-service/certified-test-beds/index.html
[5]https://youtu.be/Qa6eEJzUlRg
[6]http://www.robotnik.es/manipuladores-roboticos-moviles/rb-one/

2.2. KIO RTLS

A commercial RTLS (KIO) has been used to provide ground-truth data. KIO calculates the position of a mobile transceiver, called a *tag*, in a two- or three-dimensional space. In order to do so, KIO uses radio beacons, called *anchors*, that have been previously located in known positions in the surroundings. Red markers in **Figure 1**, left show the position of the 6 anchors used in the experiments described in the paper. They are attached at the ceiling of the mock-up apartment. The distribution of the anchors has been chosen following a previously established method (Guerrero-Higueras et al., 2017). **Figure 1**, right shows two KIO anchors.

The KIO tag was carried by the person to be tracked in our experiments. Location estimates provided by this system have an average error of ± 30 cm according to the manufacturer's specifications. Calibrations done by the authors of this paper show that the error is higher in some areas and lower in others, but on average the claims of the manufacturer are correct (Guerrero-Higueras et al., 2017).

2.3. PeTra

PeTra is a tool for detecting and tracking people developed by the Robotics Group at the University of León. The system is based on a CNN which uses an occupancy map built from LIDAR measurements as input. We explored different alternatives and we have found that the configuration proposed in the U-Net architecture by Ronneberger et al. (2015) is the one that best matches with our requirements. U-Net architecture was developed for Biomedical Image Segmentation. It consists of a contracting path to capture context and a symmetric expanding path that enables precise localization that we need to look for an specific pattern (pair of legs) in the occupancy map.

2.3.1. Neural Network Configuration

U-Net architecture was originally proposed to segment biomedical images. We have adapted it to LIDAR map "images." Basically, the architecture is an evolution of the full CNN proposed in Long et al. (2015). It consists of supplementing the usual contracting network by successive layers, where pooling operators are replaced by upsampling operators. Hence, these layers increase the resolution of the output. In this way, the contracting path captures context and a symmetric expanding path enables precise localization of targets.

Figure 2 illustrates the architecture of the CNN embedded in PeTra. It is inspired by the design in Ronneberger et al. (2015), adapting sampling sizes. Basically, the PeTra network consists of a contracting path on the left side and an expansive path on the right side of the picture. The contracting path consists of the repeated application of two 3×3 convolutions, followed by a Rectified Linear Unit (ReLU) and a 4×4 max pooling operation with stride 2 for downsampling. At each downsampling step we doubled the number of feature channels. Every step in the expansive path consists of a 4×4 up-convolution that reduces the number of feature channels, a concatenation with the corresponding feature map from the contracting path, and two 3×3 convolutions, each followed by a ReLU. At the final layer a 1×1 convolution is used to

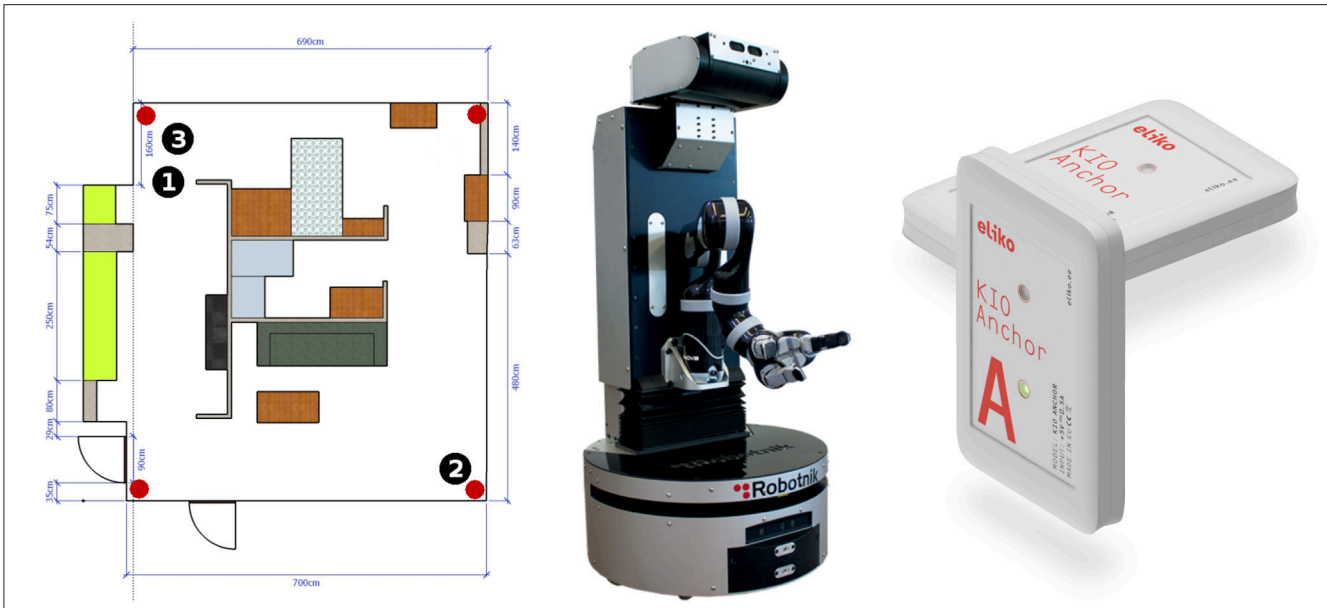


FIGURE 1 | Robotics mobile lab plan **(left)**, Orbi-One robot **(center)**, and KIO RTLS anchors **(right)**.

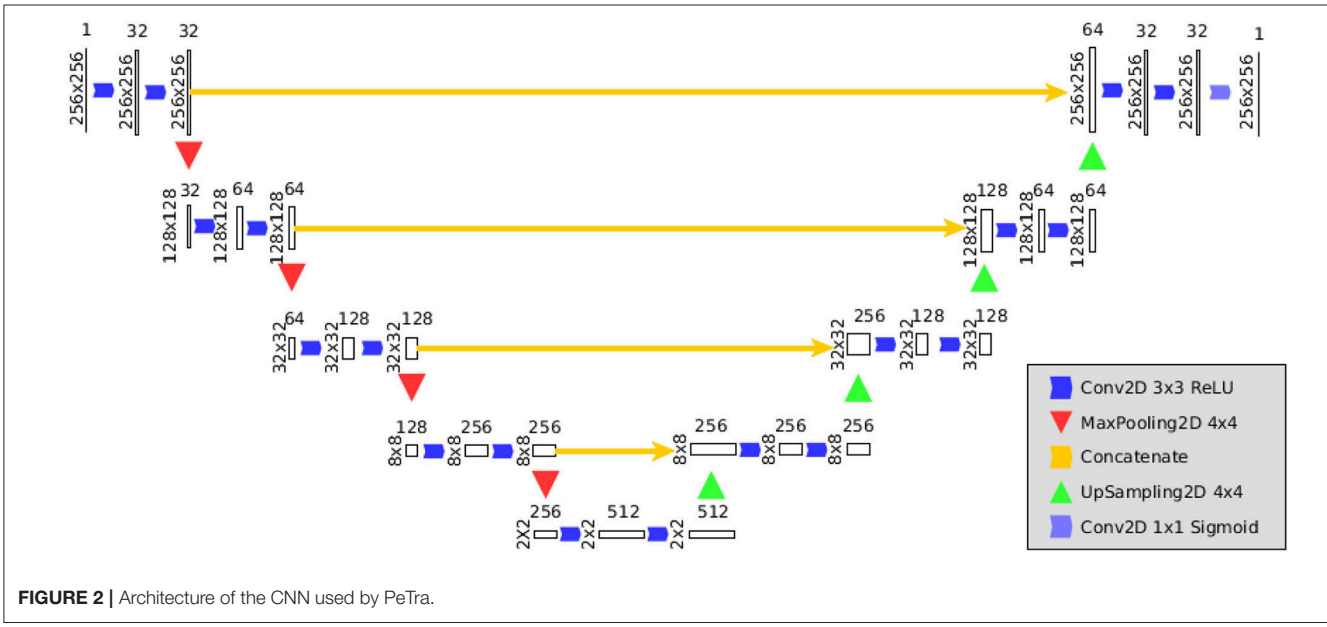


FIGURE 2 | Architecture of the CNN used by PeTra.

map each 64-component feature vector to the desired number of classes.

In order to define the input of the network, we have projected the range measurements into a two dimensional map centered around the robot. This occupancy map is defined as a 256 × 256 matrix, with a resolution of about 2 cm. The occupancy map presented in this work is not a standard occupancy map containing probability values as known in the literature. The values of each cell could be:

0: meaning either the LIDAR scan went through it without detecting any obstacle, or it did not go through that position during the reading due to occlusions or being out of range (LIDAR range is 240 degrees in the front of the robot),

1: , meaning an obstacle was found in that position.

2.3.2. Operation

PeTra has been developed and tested on a mobile robot based on the ROS framework. It has been designed as a ROS node with the CNN described at section 2.3.1 embedded on it. Once the network is trained, the system is capable of performing the following steps in real time:

1. First, the data provided by the LIDAR sensor is processed to build a two-dimensional occupancy map in front of the robot. **Figure 3**, top left shows Orbi-One and two people at the study area. **Figure 3**, top right shows the same situation over the apartment plan. **Figure 3**, bottom left illustrates the occupancy map obtained from the LIDAR's readings on the above situation. The occupancy map is presented as a 256 × 256 picture, where white pixels denote positions where the LIDAR scan found an obstacle and black pixels denote positions where either the LIDAR scan went through without detecting any obstacle or did not go through that position.
2. Then, the occupancy map of the previous step is given to the network as input data. The network produces a second occupancy map representing the zones where legs have been detected. **Figure 3**, bottom right shows the network output after processing the occupancy map shown in **Figure 3**, bottom left.
3. Finally, a mass center calculation returns peoples' locations, which is published as a location message in a specific topic.

Location messages include, among other data, the location coordinates and a timestamp that is precise to the nanosecond, as shown in **Figure 4**.

```
header:
  seq: 207
  stamp:
    secs: 1500547777
    nsecs: 369451999
  frame_id: /hokuyo_laser_link
point:
  x: 1.02
  y: 0.24
  z: 0.0
```

FIGURE 4 | Location message (PointStamped message, in ROS terminology) published at the /person topic. It includes a header with a nanosecond-precise timestamp and some identification data as well as a body with the Cartesian coordinates of the location.

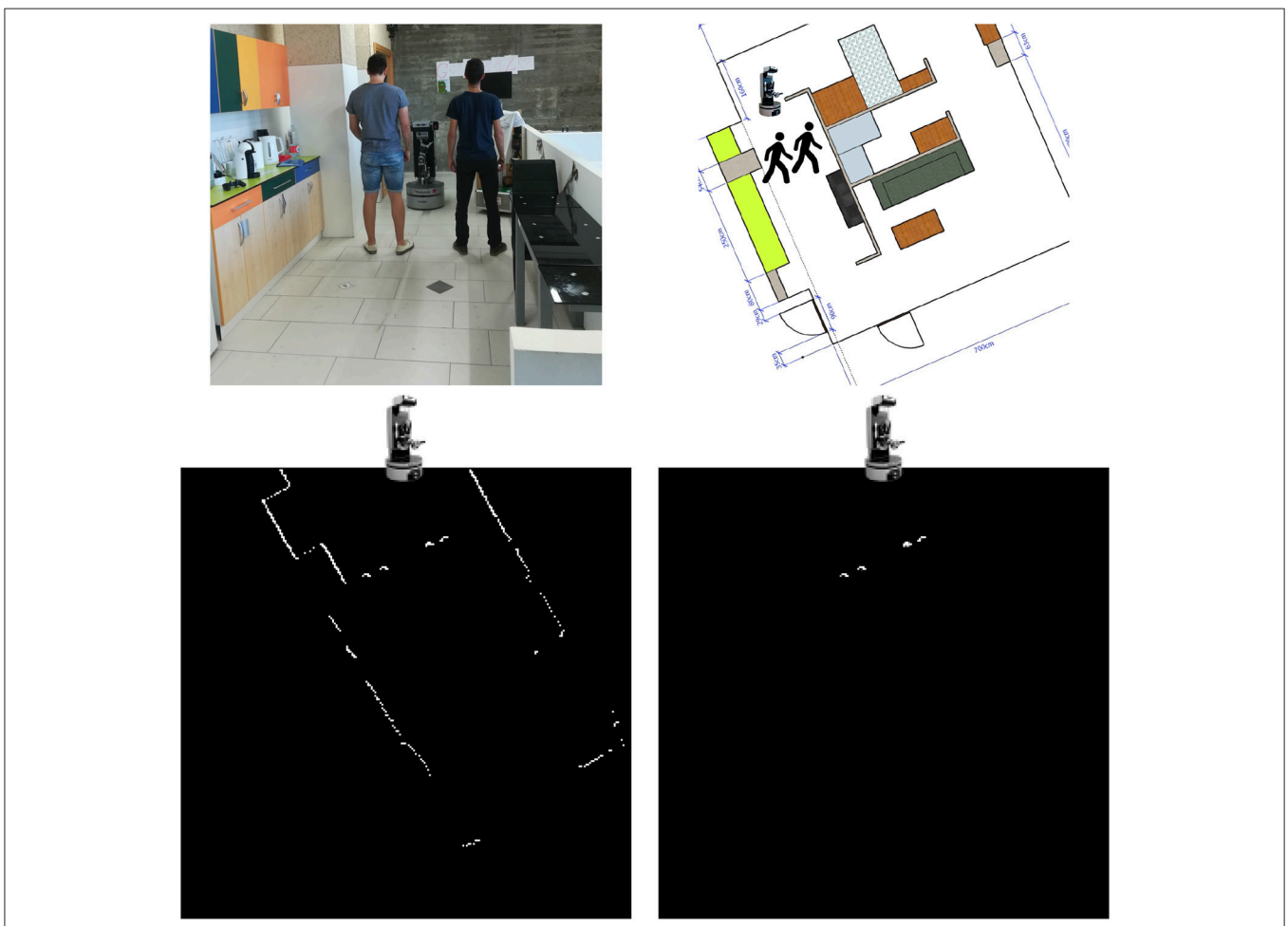


FIGURE 3 | Top left: Orbi-One and two people at the kitchen of the mock-up apartment. **Top right:** same situation over the apartment plan. **Bottom left:** occupancy map built from LIDAR's readings on the above situation. **Bottom right:** PeTra's output after processing the above occupancy map.

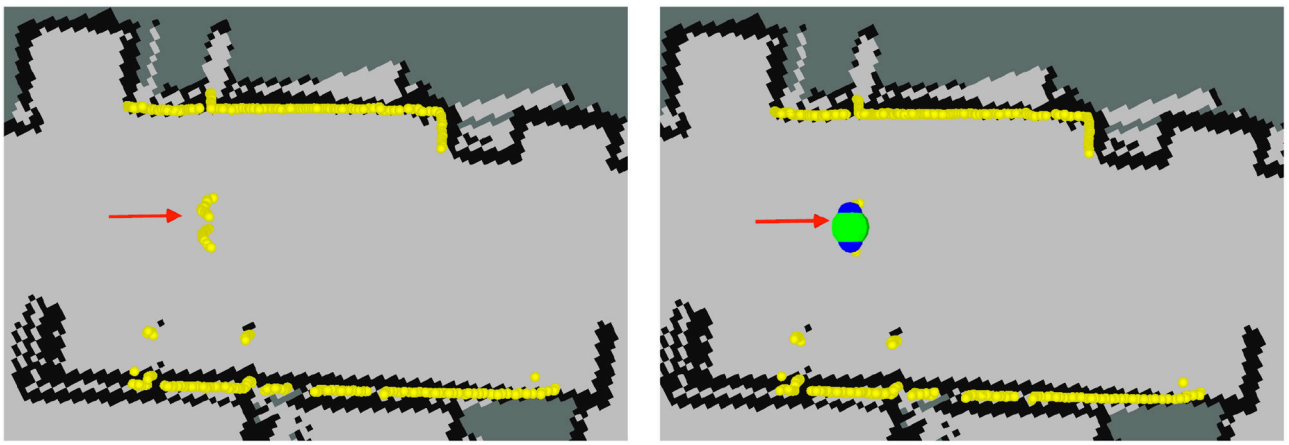


FIGURE 5 | Information of the robot as displayed on Rviz (running PeTra in the right and without running it to the left): Yellow markers show LIDAR readings; the red arrow shows Orbi-One's location and orientation, the beginning of the arrow matches the robot location; blue markers show leg location estimates; and the green marker shows the person center.

Location messages published by PeTra will be used to evaluate its performance by comparing it with ground-truth data. PeTra also publishes visualization Marker messages to indicate where detection happened. These markers can be displayed on Rviz, as shown on **Figure 5**.

2.3.3. Neural Network Training

A CNN embedded in PeTra was trained using data gathered at the Robotics Lab of the University of León (see **Figure 1**, left). To get training data a single person walked in a straight line toward the robot and then turned around to move away while the robot remained still. This situation was repeated in two locations at the mock-up apartment, the kitchen and the living room.

To label training data, KIO location estimates were used. In order to do so, for each LIDAR scan, an occupancy map was created. Then, discarding all LIDAR readings close to the KIO location estimate, leg position were labeled. To illustrate this process **Figure 6** shows the labeling from a specific LIDAR scan. **Figure 6**, left shows some information of the robot as displayed on Rivz. LIDAR's readings are shown as white points. The red arrow shows Orbi-One robot's pose (location and orientation). The robot location matches the beginning of the arrow. The pink marker shows the KIO location estimate of the person in the scene. Blue markers show the LIDAR readings close to the PeTra location estimate. **Figure 6**, middle shows the occupancy map created from the above data. **Figure 6**, right shows the labeled data by gathering the LIDAR readings close to the KIO location estimate.

The training dataset includes 2,790 pairs of images (occupancy- and labeled-map). Two thousand two hundred and twenty four were used to train the network and 550 to test it. PeTra training data is available at the University of León Robotics group website[7] As mentioned in the introduction, for PeTra evaluation, in order to ensure an optimal generalization, a different dataset was used including situations at different environments with several people in the scene, as will be explained later.

2.4. Leg Detector (LD)

LD is a standard ROS package which takes messages published by a LIDAR sensor as input and uses a machine learning-trained classifier to detect groups of LIDAR readings as possible legs. The code is available in a public repository[8], but is unsupported at this time.

Figure 7 shows LD location estimates as displayed on Rviz. It publishes the location for the individual legs. Markers colored in a black-to-blue gradient show leg location estimates. A close-to-black marker represents a small chance that it is a real leg while a close-to-blue marker represents a high probability. Some false positives appear in **Figure 7**, right, see the blue marker close to the wall. LD can also attempt to pair the legs together (displayed as red markers on Rviz) and publish their average as an estimate of where the center of the person is displayed on Rviz as a green marker.

2.5. Evaluation

A complete dataset, different than the training dataset, has been used for PeTra evaluation. This section describes how the evaluation dataset has been built. Later, the evaluation method is proposed.

2.5.1. Data Collection

Evaluation data, different than training data to ensure generalization, is needed to evaluate PeTra. Therefore, *Range-based people tracker classifiers Benchmark* dataset[2] (RRID:SCR_01574) was used. The dataset has different versions and data from version 2 were used in the evaluation. Data were

[7]http://robotica.unileon.es/~datasets/LegTracking/PeTra_training_dataset/npy_train_test_globales.tar.gz

[8]https://github.com/wg-perception/people

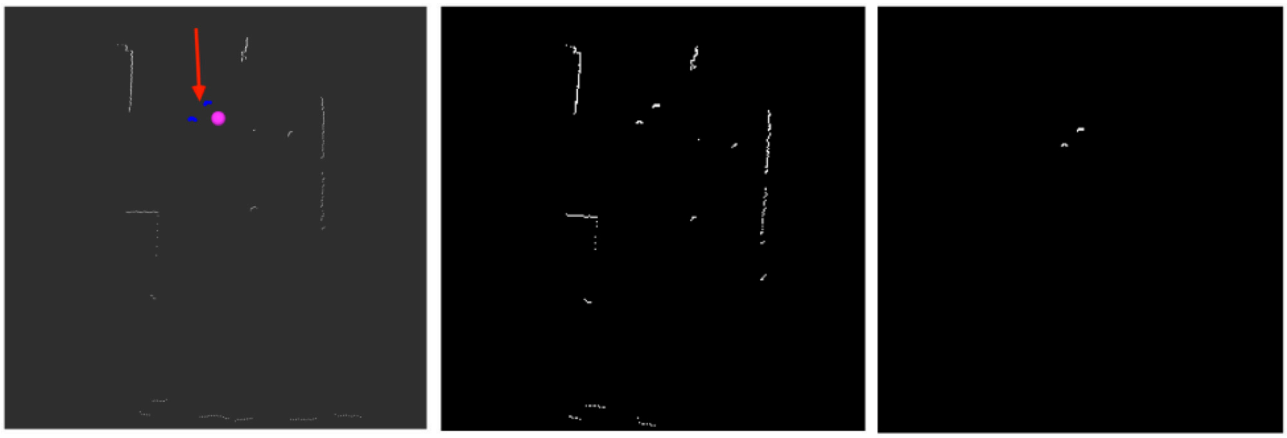


FIGURE 6 | Information of the robot as displayed on Rivz **(left)**: white points show LIDAR readings; the red arrow shows Orbi-One's location and orientation, the beginning of the arrow matches the robot location; the pink marker shows the KIO location estimate; and blue markers show the LIDAR readings close to the PeTra location estimate. Occupancy map built from LIDAR readings **(middle)**. Labeled data **(right)**.

FIGURE 7 | Information of the robot as displayed on Rviz (running LD in the right and without running it to the left): Yellow markers show LIDAR readings; the red arrow shows Orbi-One's location and orientation, the beginning of the arrow matches the robot location; black-to-blue markers show leg location estimate, close-to-black markers represent a small chance that it is a real leg, close-to-blue markers represent a high probability; red markers show paired legs; and the green marker shows the person center.

gathered in 14 different situations, where the robot stood still as one person, carrying a KIO tag, moved around it. Situations represent different human-robot interactions that may occur in robotics competitions such as ERL[9] or RoboCup[10], in particular in the "Following and Guiding" test where a robot has to follow an operator in a cluttered environment. **Figure 8** illustrates each of these situations. Three different locations were defined (black numbered markers in **Figure 1**, left show Orbi-One locations during the data gathering) resulting in 42 scenarios (14 situations $\times$ 3 locations).

Recently, a new version of the dataset was released (version 3). These data were also used in the evaluation. New data were gathered according to situations 5 and 10 (see **Figure 8**) on three different locations (kitchen, living room and bedroom), but in this case Orbi-One was not still but was instead moving around the room.

Rosbag files were used to record the state of the robot by gathering data from the robot's sensors and actuators such as LIDAR scans, odometry data, etc. A rosbag file is equivalent to a recording of the state of the robot in a period. A rosbag file was created for each scenario. In all of them, the person to be tracked was carrying a KIO tag in order to get his real location. Each rosbag file contains the following data:

– LIDAR sensor data which include, among other information, the following: acquisition time of the first ray for each scan, start/end angle, angular distance between measurements, and range data.

[9]https://www.eu-robotics.net/robotics_league/
[10]http://www.robocup.org/

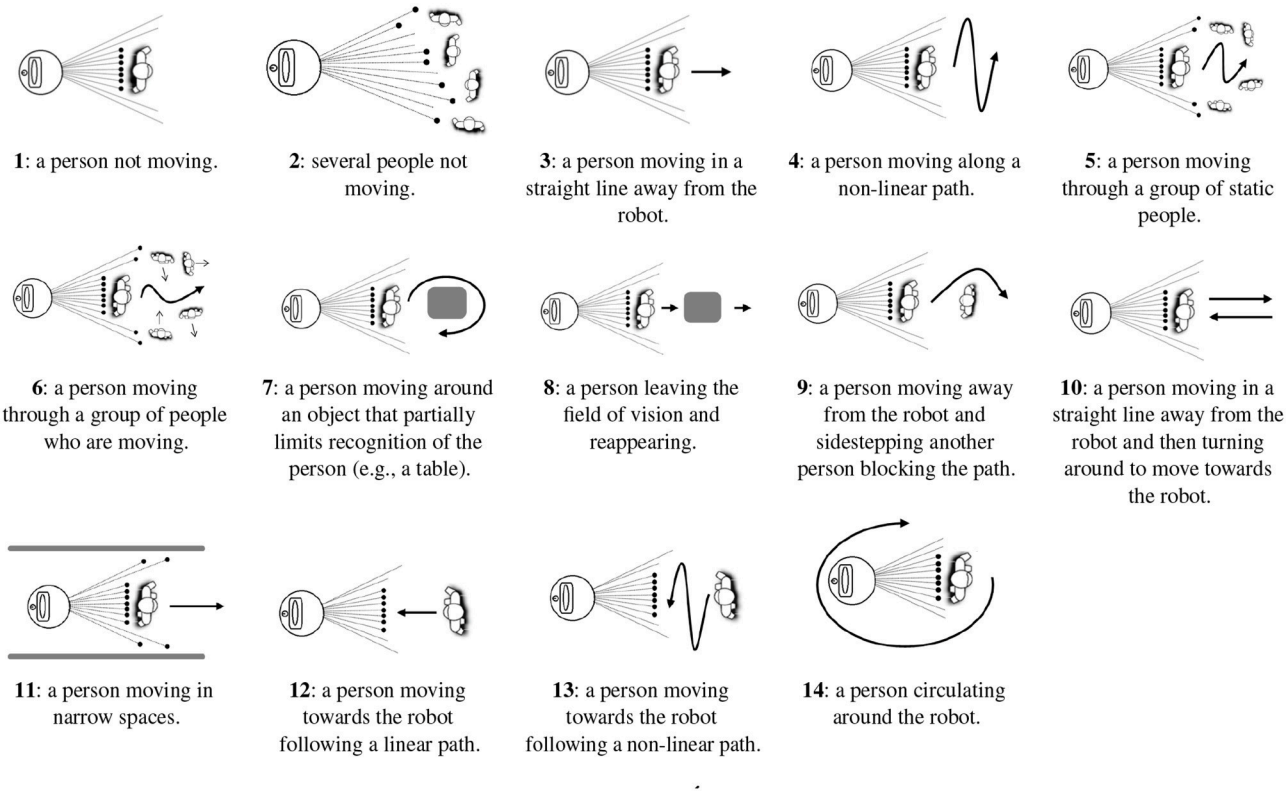


FIGURE 8 | Recognition situations recorded as shown at Álvarez-Aparicio et al. (2018).

- PeTra location estimates which include a position [x, y, z] and a timestamp.
- Location estimates calculated by LD including data for individual legs and paired legs which will be compared to ground-truth data.
- Locations provided by KIO RTLS which also include a position and a timestamp.
- Useful information such as map information, odometry of the robot base, and transformation information, respectively.

Further information about the dataset may be found at Álvarez-Aparicio et al. (2018).

2.5.2. Evaluation Method

In order to evaluate PeTra accuracy, a comparison with LD was done. In order to empirically decide which one offers the best results, location estimates of people from both systems can be compared to the ground-truth data provided by KIO RTLS. The accuracy error of both PeTra (e_{PeTra}) and LD (e_{LD}) in a specific instant of time has been calculated as the euclidean distance between their location estimates (l_{PeTra} and l_{LD}) and ground-truth data provided by KIO-RTLS (l_{KIO}). A comparison may be done just in case there is a valid location estimate from PeTra or LD, otherwise e_{PeTra} and e_{LD} will respectively get a maximum value. Equations 1 and 2 show e_{PeTra} and e_{LD} calculation,

$$e_{PeTra} = \begin{cases} \text{Max. range value} & \text{if } \nexists\, l_{PeTra} \\ d(l_{PeTra}, l_{KIO}) = \sqrt{\sum_{i=1}^{n} (l_{PeTra_i} - l_{KIO_i})^2} & \text{if } \exists\, l_{PeTra} \end{cases} \tag{1}$$

$$e_{LD} = \begin{cases} \text{Max. range value} & \text{if } \nexists\, l_{LD} \\ d(l_{LD}, l_{KIO}) = \sqrt{\sum_{i=1}^{n} (l_{LD_i} - l_{KIO_i})^2} & \text{if } \exists\, l_{LD} \end{cases} \tag{2}$$

where n is the number of dimensions considered. In our experiments, only X and Y coordinates are considered. The Z coordinate is constant since a mobile robot moves on the ground. Ground-truth data location estimates (l_{KIO}) are provided as Cartesian coordinates by KIO RTLS. PeTra and LD location estimates (l_{PeTra} and l_{LD}) are also provided as Cartesian coordinates but using the LIDAR location as the coordinates' origin.

The evolution of e_{PeTra} and e_{LD} over time was used to decide the system which works better. The accuracy error was calculated for the period of time covered for each Rosbag file included in the dataset.

Regarding the above, there are two important issues to deal with as detailed in Álvarez-Aparicio et al. (2018). First, KIO, PeTra and LD use their own coordinate origins to represent locations. In order to compare these locations they ought to be represented using the same coordinate origins. The *tf* package,

see Foote (2013), included in the ROS framework core, is in charge of transforming KIO, PeTra and LD location estimates, allowing for easy comparison. *tf* operates with a central server that contains all transform information. On the other hand, each message published for the recorded topics has its own timestamp with precision to the nanosecond, so comparing locations in a concrete instant of time may not be an easy task. A synchronization method is needed to compare measurements from different topics. To synchronize data in our experiments, we have used the timestamp of the PeTra location estimates and then we have selected the closest measure in time from both LD and KIO.

3. RESULTS

The accuracy of PeTra has been evaluated vs. LD accuracy applying the described method. **Table 1** shows the mean error and standard deviation for the 14 situations of the second version of the dataset. As can be seen in the table, the average error of e_{PeTra} at locations 1, 2, and 3, are 0.17, 0.43, and 0.20 m respectively; it is lower than e_{LD} at all locations (0.30, 0.75, and 0.49 m respectively).

The standard deviations of e_{PeTra} at locations 1, 2, and 3, are 0.13, 0.22, and 0.13 m respectively; which are also slightly lower than e_{LD} (0.15, 0.28, and 0.33 m respectively).

Figure 9 shows the evolution of the accuracy error for both PeTra and LD over the time horizon given by Rosbag files recorded at locations 1, 2, and 3, on situations 5, and 10, of the second version of the dataset. Green markers represent e_{LD}. Red markers represent e_{PeTra}. Green and red lines illustrate e_{LD} and e_{PeTra} Moving Average (MA). MAs were used to smooth out short-term fluctuations and highlight longer-term trends or cycles. The same analysis has been done for all scenarios at all locations. Complete results are shown in **Figures S1–S3**. According to **Table 1**, e_{PeTra} is slightly lower on average than e_{LD} for scenarios 5 and 10. This is shown in the figure, e_{PeTra} is lower than e_{LD} on scenario 5 at location 3, and on scenario 6 at locations 1, 2, and 3. However, there are specific moments where e_{LD} is lower, especially at scenario 5 on locations 1 and 2.

Table 2 shows the mean error and standard deviation for the two situations of the third version of the dataset. As can be seen in the table, the average error of e_{PeTra} at the kitchen, the living room, and the bedroom, are 0.18m, 0.13m, and 0.53m, respectively; it is lower than e_{LD} at the three rooms (0.38, 0.57, and 0.64 m).

The standard deviations of e_{PeTra} at the three rooms are 0.25, 0.15, and 0.23 m respectively are slightly higher than standard deviations of e_{LD} at the kitchen and at the bedroom (0.22 and 0.22 m) and lower at the living room (0.30 m).

Figure 10 shows the evolution of the accuracy error for both PeTra and LD over the time horizon given by rosbag files recorded at locations 1, 2, and 3, on situations 5, and 10, of the third version of the dataset. Green markers represent e_{LD}. Red markers represent e_{PeTra}. Green and red lines illustrate e_{LD} and e_{PeTra} MA. According to the figure, e_{PeTra} is lower than e_{LD} most of the time for both scenarios.

Regarding performance, PeTra spends $\approx$ 0.3 s on calculating a location estimate from LIDAR sensor data when running on Orbi-One hardware configuration as described in 2.1. With the same hardware platform LD spends $\approx$ 0.1 s on calculating a location estimate.

4. DISCUSSION

Table 1 shows that PeTra has a lower mean error than LD at the three locations. These differences represent an accuracy increase of 43%, 42%, and 59% at locations 1, 2, and 3, respectively. Considering the standard deviation, differences between PeTra

TABLE 1 | Mean error and standard deviation (m) using data from version 2 of the dataset.

Situation	Location 1		Location 2		Location 3	
	$\bar{X}_{e_{PeTra}}$	$\bar{X}_{e_{LD}}$	$\bar{X}_{e_{PeTra}}$	$\bar{X}_{e_{LD}}$	$\bar{X}_{e_{PeTra}}$	$\bar{X}_{e_{LD}}$
1	0.02 (±0.02)	0.21 (±0.04)	1.40 (±0.03)	1.16 (±0.03)	0.03 (±0.01)	0.52 (±0.09)
2	0.48 (±0.47)	0.17 (±0.11)	1.25 (±0.29)	1.51 (±0.19)	0.48 (±0.35)	0.33 (±0.08)
3	0.08 (±0.10)	0.32 (±0.15)	0.06 (±0.03)	0.62 (±0.40)	0.06 (±0.02)	0.20 (±0.08)
4	0.09 (±0.06)	0.30 (±0.15)	0.36 (±0.50)	0.85 (±0.39)	0.15 (±0.12)	0.32 (±0.11)
5	0.43 (±0.12)	0.35 (±0.13)	0.95 (±0.46)	0.84 (±0.30)	0.38 (±0.07)	0.54 (±0.10)
6	0.16 (±0.22)	0.21 (±0.10)	0.55 (±0.44)	0.63 (±0.44)	0.36 (±0.29)	0.61 (±0.27)
7	0.18 (±0.22)	0.39 (±0.25)	0.20 (±0.14)	0.95 (±0.52)	0.21 (±0.13)	0.24 (±0.12)
8	0.11 (±0.14)	0.26 (±0.11)	0.08 (±0.07)	0.73 (±0.28)	0.08 (±0.06)	0.39 (±0.16)
9	0.29 (±0.18)	0.25 (±0.15)	0.69 (±0.60)	0.56 (±0.42)	0.41 (±0.20)	1.78 (±2.89)
10	0.10 (±0.06)	0.36 (±0.16)	0.10 (±0.07)	0.71 (±0.24)	0.19 (±0.13)	0.50 (±0.16)
11	0.05 (±0.03)	0.26 (±0.08)	0.08 (±0.18)	0.18 (±0.06)	0.08 (±0.09)	0.28 (±0.05)
12	0.04 (±0.02)	0.27 (±0.16)	0.06 (±0.05)	0.53 (±0.27)	0.04 (±0.04)	0.55 (±0.13)
13	0.12 (±0.11)	0.49 (±0.28)	0.13 (±0.10)	1.03 (±0.34)	0.23 (±0.22)	0.35 (±0.30)
14	0.19 (±0.11)	0.35 (±0.16)	0.12 (±0.12)	0.18 (±0.13)	0.10 (±0.11)	0.18 (±0.09)
Average	0.17 (±0.13)	0.30 (±0.15)	0.43 (±0.22)	0.75 (±0.28)	0.20 (±0.13)	0.49 (±0.33)

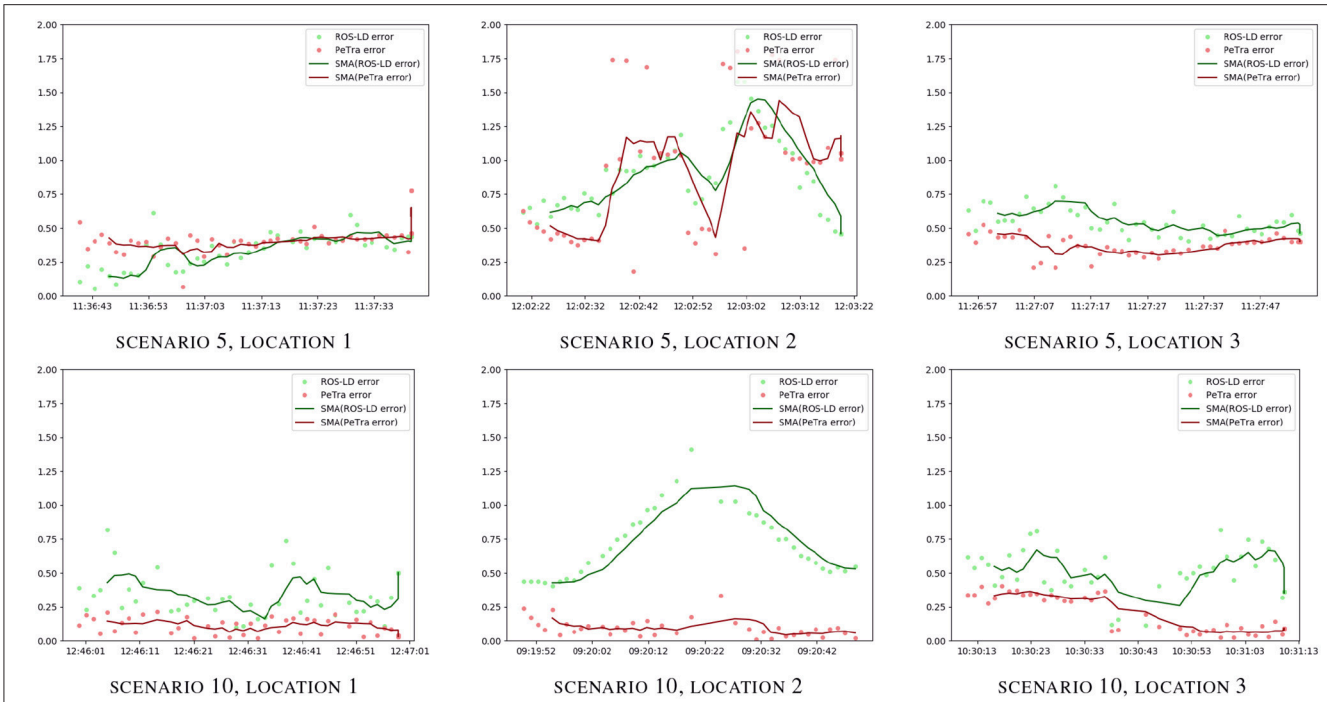


FIGURE 9 | e_{PeTra} (m) and e_{LD} (m) evolution over time (hh:mm:ss) on scenarios 5 (1st row), and 10 (2th row), at locations 1 **(left)**, 2 **(center)**, and 3 **(right)**. Red markers show e_{PeTra}, green markers show e_{LD}, red line show MA(e_{PeTra}), and green line shows MA(e_{LD}).

TABLE 2 | Mean error and standard deviation (m) using data from version 3 of the dataset.

Situation	Kitchen		Living room		Bedroom	
	$\bar{X}_{e_{PeTra}}$	$\bar{X}_{e_{LD}}$	$\bar{X}_{e_{PeTra}}$	$\bar{X}_{e_{LD}}$	$\bar{X}_{e_{PeTra}}$	$\bar{X}_{e_{LD}}$
5	0.20 (±0.33)	0.35 (±0.25)	0.14 (±0.17)	0.56 (±0.29)	0.92 (±0.34)	0.98 (±0.33)
10	0.16 (±0.16)	0.40 (±0.19)	0.12 (±0.12)	0.57 (±0.31)	0.13 (±0.12)	0.29 (±0.11)
Average	0.18 (±0.25)	0.38 (±0.22)	0.13 (±0.15)	0.57 (±0.30)	0.53 (±0.23)	0.64 (±0.22)

and LD represent a reduction of 13, 21, and 60% at locations 1, 2, and 3. Thus, PeTra is more consistent over time at all locations when Orbi-One remains still. Considering each situation, e_{PeTra} is lower on average than e_{LD}, however, there are not too many differences on multi-person scenarios (scenarios 2, 5, 6 and 9). It is expected because the training data employs a single person.

Figures S1–S3 visually illustrate the evolution over time of e_{PeTra} and e_{LD}. They show that e_{PeTra} is lower than e_{LD} most of the time for most of the scenarios at the three locations analyzed. However, we must analyze carefully the differences between different situations. In order to do so, scenarios 5 and 10 have been selected for in-depth analysis. Graphics of **Figure 9** visually illustrate the evolution of e_{PeTra} and e_{LD} specifically on these scenarios. These scenarios have been chosen because they include a large variety of situations with one or several people standing or moving around in the environment. They also include situations where PeTra gets a better performance and a worse performance comparing to LD.

The first row of **Figure 11** shows information of the robot on situation 5 at location 2 as displayed at Rviz. On this situation, recorded at the living room of the mock-up apartment, four people stand still in front of the robot while another one moves away. The red arrow shows the robot's pose. The beginning of the arrow matches the robot's location. Yellow markers show LIDAR's readings. These readings allow the identification of the people on the scene. Two half-circles can be guessed in front of the robot at several places in the image. These two half-circles are the shape that the sensor gathers when a person is in the LIDAR range. In this case, PeTra is able to detect two of the people in the scene as shown in **Figure 11**, top middle. Another person close to the wall is not detected because just one of her legs can be seen. The fourth person is hidden behind another one and is not detected either. Regarding LD, it identifies 3 people in the scene, as shown in **Figure 11**, top right, but just one of them corresponds to a real person, the others have been located by pairing legs from different people or even from furniture.

The second row of **Figure 11** shows information of the robot on situation 10 at location 3. It was recorded at the bedroom of the mock-up apartment. Here, a person first moves away and

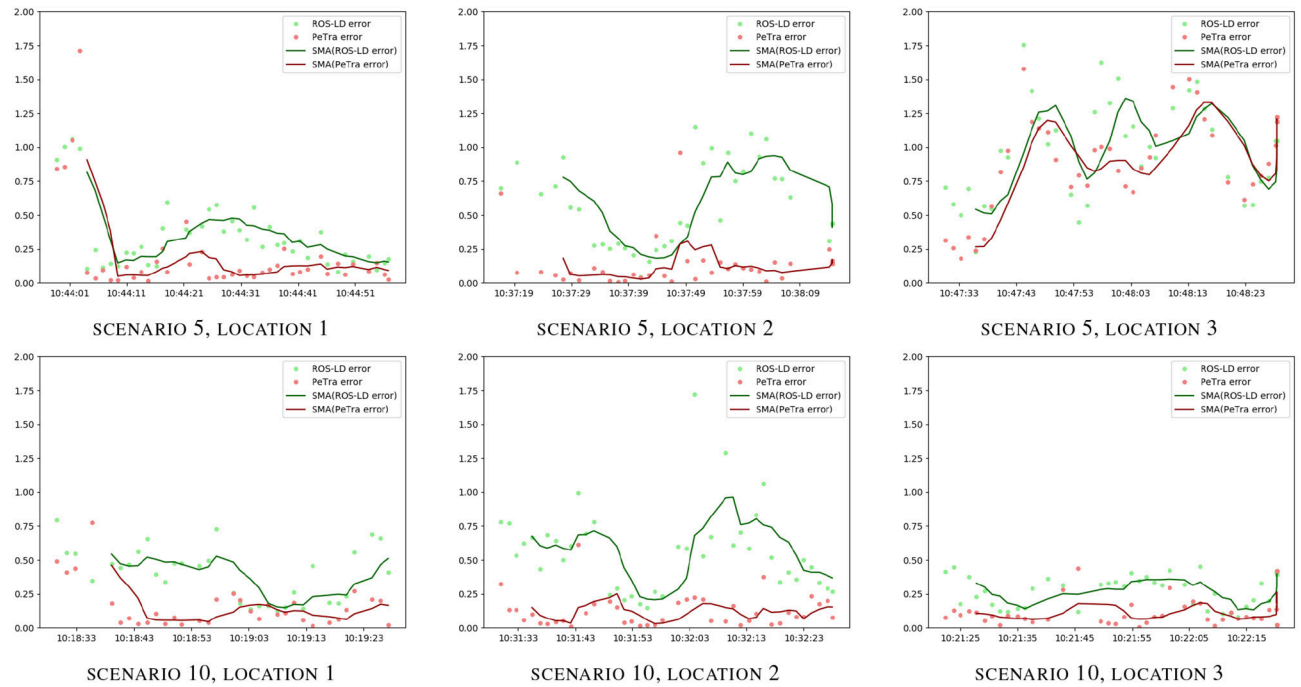


FIGURE 10 | e_{PeTra} (m) and e_{LD} (m) evolution over time (hh:mm:ss) on scenarios 5 (1st row), and 10 (2nd row), at locations 1 **(left)**, 2 **(center)**, and 3 **(right)**. Red markers show e_{PeTra}, green markers show e_{LD}, red line shows MA(e_{PeTra}), and green line shows MA(e_{LD}).

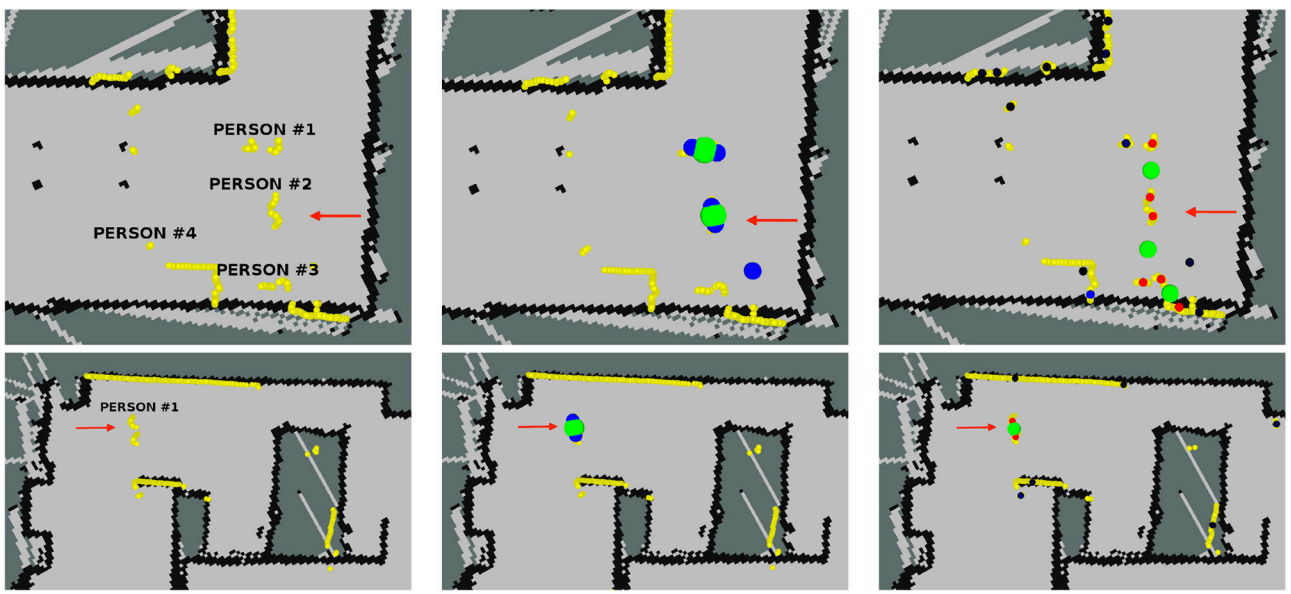


FIGURE 11 | Information of the robot as displayed on Rviz for scenarios 5 (1st row), and 10 (2nd row). Yellow markers show LIDAR readings. The red arrow shows Orbi-One's location and orientation. The beginning of the arrow matches the robot location. PeTra's location estimates are shown on the center: blue markers show leg location estimates and the green marker shows the person's center. LD location estimates are shown on the right: black-to-blue markers show legs location estimates, close-to-black markers represent a low chance it is a real leg, close-to-blue markers represent a high probability; red markers show paired legs; and the green marker shows the person center.

then moves toward the robot. The LIDAR's readings allow the identification of the person on the scene. In this case, both PeTra and LD identify the person correctly, but LD also identifies some points as people's legs that in fact match furniture legs. These matchings are displayed as black-to-blue markers in **Figure 11**, bottom right.

Regarding results of **Table 2**, again e_{PeTra} has a lower mean error than e_{LD}, representing an accuracy increase of 52, 77, and 18% at locations 1, 2, and 3 ,respectively. The standard deviation is slightly higher at locations 1 and 3 (12 and 4%) and lower at location 2 (50%). Graphics of **Figure 10** visually illustrate the evolution of e_{PeTra} and e_{LD} over time, showing that PeTra offers better accuracy than LD most of the time in both situations. These results demonstrate that PeTra is also more consistent over time at all locations when the Robot moves.

5. CONCLUSIONS AND FURTHER WORK

This paper presented a system named PeTra to track pair of legs (people) by processing LIDAR sensor data using CNNs. This could be used in several applications, such as improving navigation, facilitating human-robot interaction, or in security and safety. To demonstrate that PeTra allows the tracking of people, it has been compared with LD, a well-known solution for tracking people from LIDAR sensor data, at several locations and situations. Evaluations have been done using data different than the ones used to train and test both models. As a result, experiments proved that PeTra offers a better accuracy in most scenarios.

The main contribution of this work is the PeTra system itself. But in addition, we also want to point out the technical contributions of this work, which include:

1. A people tracking system ready to be used by any mobile robot using the ROS framework.
2. A method to evaluate the performance of range-based people trackers in mobile robots by comparing their results with the data contained in public dataset (RRID:SCR_015743). The dataset is described in section 2.5.1. Data are available at the University of Leon Robotics group website[2].
3. A system to transform LIDAR sensor data to a two-dimensional occupancy map as described in section 2.3.1, which enables processing them as a picture. This makes LIDAR sensor data treatable by classifiers based on neural networks which used pictures as input.

Regarding further work, there are several aspects that have to be analyzed. On the one hand, a network optimization is needed to improve performance. A mobile robot such as Orbi-One needs to get peoples' locations in real time as soon as LIDAR sensor data is received. A 0.3 s delay can be too muchinon certain situations.

LIDAR sensor data pre-processing could improve performance. Currently, a Boolean matrix is built considering just two situations: in the first one, pointed out as the value 1, LIDAR scan found an obstacle in a given position; in the second one, pointed out as the value 0, LIDAR scan went through it without detecting an obstacle in that position or did not go through it due to an occlusion. Differentiating the cases when the LIDAR scan found an obstacle and when it did not go through that position, could be useful to generate new fit and evaluation data for the CNN used by PeTra.

AUTHOR CONTRIBUTIONS

PeTra is based on CÁ-A's work for her End-of-Degree Assignment that was proposed and advised by VM. MC carried out most of the data gathering advised by ÁG-H and CF-L. ÁG-H carried the accuracy evaluation of PeTra. FM and FR-L independently replicated the experiments in their labs. Finally, ÁG-H and VM did the KIO calibration and drafted this manuscript.

FUNDING

The research described in this article has been partially funded by the Instituto Nacional de Ciberseguridad de España (INCIBE), under the Grant Cybersecurity in Systems with Autonomous Capacities (Addendum 21), Addendum to the framework agreement INCIBE–Universidad de León, 2016–2018; and by Junta de Castilla y León, under grant LE028P17.

ACKNOWLEDGMENTS

We would like to thank A. Yugueros and R. M. Redondo for their support during the data gathering.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2018.00085/full#supplementary-material

REFERENCES

Aguirre, E., Garcia-Silvente, M., and Plata, J. (2014). "Leg detection and tracking for a mobile robot and based on a laser device, supervised learning and particle filtering," in *ROBOT2013: First Iberian Robotics Conference*, Vol 252, eds M. Armada, A. Sanfeliu and M. Ferre (Cham: Springer), 433–440. doi: 10.1007/978-3-319-03413-3_31

Álvarez-Aparicio, C., Guerrero-Higueras, Á. M., Olivera, M. C. C., Rodríguez-Lera, F. J., Martín, F., and Matellán, V. (2018). Benchmark dataset for evaluation of range-based people tracker classifiers in mobile robots. *Front. Neurorob.* 11:72. doi: 10.3389/fnbot.2017.00072

Arras, K. O., Lau, B., Grzonka, S., Luber, M., Mozos, O. M., Meyer, D., et al. (2012). "Range-based people detection and tracking for socially enabled service robots," in *Towards Service Robots for Everyday Environments. STAR*, Vol. 76, eds E. Prassler, R. Bischoff, W. Burgard, R. Haschke, M. Hägele, G. Lawitzky, B. Nebel, P. Plöger, U. Reiser, and M. Zöllner (Berlin; Heidelberg: Springer), 76, 235–280. doi: 10.1007/978-3-642-25116-0_18

Foote, T. (2013). "tf: The transform library," in *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference*. Woburn, MA: Open-Source Software workshop, 1–6.

Gavrilova, M., Ahmed, F., Azam, S., Paul, P., Rahman, W., Sultana, M., et al. (2017). "Emerging trends in security system design using the concept of social behavioural biometrics," in *Information Fusion for Cyber-Security Analytics*, Vol 691, eds I. Alsmadi, G. Karabatis and A. Aleroud (Cham: Springer), 229–251. doi: 10.1007/978-3-319-44257-0_10

Guerrero-Higueras, Á. M., DeCastro-García, N., Rodríguez-Lera, F. J., and Matellán, V. (2017). Empirical analysis of cyber-attacks to an indoor real time localization system for autonomous robots. *Comput. Secur.* 70, 422–435. doi: 10.1016/j.cose.2017.06.013

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (Lake Tahoe: Curran Associates Inc.), 1097–1105. Available online at: http://dl.acm.org/citation.cfm?id=2999134.2999257

Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D. (1997). Face recognition: a convolutional neural-network approach. *IEEE Trans. Neural Netw.* 8, 98–113.

Lee, J. H., Tsubouchi, T., Yamamoto, K., and Egawa, S. (2006). "People tracking using a robot in motion with laser range finder," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference* (Beijing: iEEE), 2936–2942.

Leigh, A., Pineau, N. O. J., and Zhang, H. (2015). "Person tracking and following with 2d laser scanners," in *International Conference on Robotics and Automation (ICRA)*, Seattle (Washington, DC: IEEE).

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Boston, MA), 3431–3440.

Mashad Nemati, H., Gholami Shahbandi, S., and Åstrand, B. (2016). "Human tracking in occlusion based on reappearance event estimation," in *13th International Conference on Informatics in Control, Automation and Robotics, Lisbon, Portugal, 29-31 July, 2016*, Vol. 2 (Lisbon), 505–511.

Morales, Y., Kanda, T., and Hagita, N. (2014). Walking together: side-by-side walking model for an interacting robot. *J. Hum. Rob. Interact.* 3, 50–73. doi: 10.5898/JHRI.3.2.Morales

Morante, S., Victores, J. G., and Balaguer, C. (2015). Cryptobotics: why robots need cyber safety. *Front. Rob. AI* 2:23. doi: 10.3389/frobt.2015.00023

Ngo, D. C. L., Teoh, A. B. J., and Hu, J. (2015). *Biometric security*. Newcastle upon Tyne, UK: Cambridge Scholars Publishing. Available online at: https://books.google.es/books?hl=es&lr=&id=nfmmBgAAQBAJ&oi=fnd&pg=PR5&dq=Ngo+Biometric+security&ots=7ywIvqi1Bs&sig=mkJ_VCPQPZUL4XUmC17m2pLTIFY

Ondruska, P., Dequaire, J., Wang, D. Z., and Posner, I. (2016). End-to-end tracking and semantic segmentation using recurrent neural networks. *arXiv [Preprint]. arXiv:1604.05091*.

Premebida, C., Ludwig, O., and Nunes, U. (2009). Lidar and vision-based pedestrian detection system. *J. Field Rob.* 26, 696–711.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., et al. (2009). "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, Vol. 3 (Kobe), 5–10.

Rios-Martinez, J., Spalanzani, A., and Laugier, C. (2015). From proxemics theory to socially-aware navigation: a survey. *Int. J. Soc. Rob.*, 7, 137–153. doi: 10.1007/s12369-014-0251-1

Ronneberger, O., P.Fischer, and Brox, T. (2015). "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Vol. 9351, *LNCS* (Munich: Springer), 234–241.

Schulz, D., Burgard, W., Fox, D., and Cremers, A. B. (2003). People tracking with mobile robots using sample-based joint probabilistic data association filters. *Int. J. Rob. Res.* 22, 99–116. doi: 10.1177/0278364903022002002

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv [Preprint] arXiv:1409.1556*.

Szarvas, M., Sakai, U., and Ogata, J. (2006). "Real-time pedestrian detection using lidar and convolutional neural networks," in *Intelligent Vehicles Symposium, 2006 IEEE* (Tokyo: IEEE), 213–218.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 07 March 2019
doi: 10.3389/fnbot.2019.00006

A Differentiable Physics Engine for Deep Learning in Robotics

Jonas Degrave[1*†], *Michiel Hermans*[2†], *Joni Dambre*[1] *and Francis wyffels*[1]

[1] IDLab-AIRO, Department of Electronics and Information Systems, Ghent University - imec, Ghent, Belgium, [2] Independent Researcher, Ghent, Belgium

An important field in robotics is the optimization of controllers. Currently, robots are often treated as a black box in this optimization process, which is the reason why derivative-free optimization methods such as evolutionary algorithms or reinforcement learning are omnipresent. When gradient-based methods are used, models are kept small or rely on finite difference approximations for the Jacobian. This method quickly grows expensive with increasing numbers of parameters, such as found in deep learning. We propose the implementation of a modern physics engine, which can differentiate control parameters. This engine is implemented for both CPU and GPU. Firstly, this paper shows how such an engine speeds up the optimization process, even for small problems. Furthermore, it explains why this is an alternative approach to deep Q-learning, for using deep learning in robotics. Finally, we argue that this is a big step for deep learning in robotics, as it opens up new possibilities to optimize robots, both in hardware and software.

Keywords: differentiable physics engine, deep learning, gradient descent, neural network controller, robotics

OPEN ACCESS

Edited by:
Florian Röhrbein, Technische Universität München, Germany

Reviewed by:
Eiji Uchibe, Advanced Telecommunications Research Institute International (ATR), Japan
Keyan Ghazi-Zahedi, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Germany
Jose De Jesus Rubio, Instituto Politécnico Nacional, Mexico

****Correspondence:***
Jonas Degrave
jonas.degrave@ugent.be

†Present Address:
Jonas Degrave, Deepmind, London, United Kingdom
Michiel Hermans, ScriptBook NV, Antwerp, Belgium

Received: *07 June 2018*
Accepted: *11 February 2019*
Published: *07 March 2019*

Citation:
Degrave J, Hermans M, Dambre J and wyffels F (2019) A Differentiable Physics Engine for Deep Learning in Robotics. Front. Neurorobot. 13:6. doi: 10.3389/fnbot.2019.00006

1. INTRODUCTION

To solve tasks efficiently, robots require an optimization of their control system. This optimization process can be done in automated testbeds (Degrave et al., 2015), but typically these controllers are optimized in simulation. Standard methods (Aguilar-Ibañez, 2017; Meda-Campana, 2018) to optimize these controllers include particle swarms, reinforcement learning, genetic algorithms, and evolutionary strategies. These are all derivative-free methods.

A recently popular alternative approach is to use deep Q-learning, a reinforcement learning algorithm. This method requires a lot of evaluations in order to train the many parameters (Levine et al., 2018). However, deep learning experience has taught us that optimizing with a gradient is often faster and more efficient. This fact is especially true when there are a lot of parameters, as is common in deep learning. However, in the optimization processes for control systems, the robot is almost exclusively treated as a non-differentiable black box. The reason for this is that the robot in hardware is not differentiable, nor are current physics engines able to provide the gradient of the robot models. The resulting need for derivative-free optimization approaches limits both the optimization speed and the number of parameters in the controllers. One could tackle this issue by fitting a neural network model and using its gradient (Grzeszczuk et al., 1998), but those gradients tend to be poor a approximations for the gradient of the original system.

Recent physics engines, such as mujoco (Todorov et al., 2012), can derive gradients through the model of a robot. However, they can at most evaluate gradients between actions and states in the transitions of the model, and cannot find the derivatives with respect to model parameters.

In this paper, we suggest an alternative approach, by introducing a differentiable physics engine with analytical gradients. This idea is not novel. It has been done before with spring-damper models in 2D and 3D (Hermans et al., 2014). This technique is also similar to adjoint optimization, a method widely used in various applications such as thermodynamics (Jarny et al., 1991) and fluid dynamics (Iollo et al., 2001). However, modern engines to model robotics are not based on spring-damper systems. The most commonly used ones are 3D rigid body engines, which rely on impulse-based velocity stepping methods (Erez et al., 2015). In this paper, we test whether these engines are also differentiable and whether this gradient is computationally tractable. We will show how this method does speed up the optimization process tremendously, and give some examples where we optimize deep learned neural network controllers with millions of parameters.

2. MATERIALS AND METHODS

2.1. A 3D Rigid Body Engine

The goal is to implement a modern 3D rigid body engine, in which parameters can be differentiated with respect to the fitness a robot achieves in a simulation, such that these parameters can be optimized with methods based on gradient descent.

The most frequently used simulation tools for model-based robotics, such as PhysX, Bullet, Havok, and ODE, go back to MathEngine (Erez et al., 2015). These tools are all 3D rigid body engines, where bodies have 6 degrees of freedom, and the relations between them are defined as constraints. These bodies exert impulses on each other, but their positions are constrained, e.g., to prevent the bodies from penetrating each other. The velocities, positions and constraints of the rigid bodies define a linear complementarity problem (LCP) (Chappuis, 2013), which is then solved using a Gauss-Seidel projection (GSP) method (Jourdan et al., 1998). The solution of this problem are the new velocities of the bodies, which are then integrated by semi-implicit Euler integration to get the new positions (Stewart and Trinkle, 2000). This system is not always numerically stable. Therefore, the constraints are usually softened (Catto, 2009).

The recent growth of automatic differentiation libraries, such as Theano (Al-Rfou et al., 2016), Caffe (Jia et al., 2014), and Tensorflow (Abadi et al., 2016), has allowed for efficient differentiation of remarkably complex functions before (Degrave et al., 2016a). Therefore, we implemented such a physics engine from scratch as a mathematical expression in Theano, a software library which does automatic evaluation and differentiation of expressions with a focus on deep learning. The resulting computational graph to evaluate this expression is then compiled for both CPU and GPU. To be able to compile for GPU however, we had to limit our implementation to a restricted set of elementary operations. The range of implementable functions is therefore severely capped. However, since the analytic gradient is determined automatically, the complexity of correctly implementing the differentiation is removed entirely.

One of these limitations with this restricted set of operations, is the limited support for conditionals. Therefore, we needed to implement our physics engine without branching, as this is not yet available in Theano for GPU. Note that newer systems for automatic differentiation such as PyTorch (Paszke et al., 2017) do allow branching. Therefore, we made sacrificed some abilities of our system. For instance, our system only allows for contact constraints between different spheres or between spheres and the ground plane. Collision detection algorithms for cubes typically have a lot of branching (Mirtich, 1998). However, this sphere based approach can in principle be extended to any other shape (Hubbard, 1996). On the other hand, we did implement a rather accurate model of servo motors, with gain, maximal torque, and maximal velocity parameters.

Another design choice was to use rotation matrices rather than the more common quaternions for representing rotations. Consequently, the states of the bodies are larger, but the operations required are matrix multiplications. This design reduced the complexity of the graph. However, cumulative operations on a rotation matrix might move the rotation matrix away from orthogonality. To correct for this, we renormalize our matrix with the update equation (Premerlani and Bizard, 2009):

$$A' = \frac{3A - A \circ (A \cdot A)}{2} \tag{1}$$

where A' is the renormalized version of the rotation matrix A. "$\circ$" denotes the elementwise multiplication, and "$\cdot$" the matrix multiplication.

These design decisions are the most important aspects of difference with the frequently used simulation tools. In the following section, we will evaluate our physics simulator on some different problems. We take a look at the speed of computation and the number of evaluations required before the parameters of are optimized.

2.1.1. Throwing a Ball

To test our engine, we implemented the model of a giant soccer ball in the physics engine, as shown in **Figure 3A**. The ball has a 1 m diameter, a friction of $\mu = 1.0$ and restitution $e = 0.5$. The ball starts off at position $(0, 0)$. After 5 s it should be at position $(10, 0)$ with zero velocity v and zero angular velocity ω. We optimized the initial velocity v_0 and angular velocity ω_0 at time $t = 0$ s until the errors at $t = 5$ s are <0.01 m and 0.01 m/s respectively.

Since the quantity we optimize is only know at the end of the simulation, but we need to optimize the parameters at the beginning of the simulation, we need to backpropagate our error through time (BPTT) (Sutskever, 2013). This approach is similar to the backpropagation through time method used for optimizing recurrent neural networks (RNN). In our case, every time step in the simulation can be seen as one pass through a neural network, which transforms the inputs from this timestep to inputs for the next time step. For finding the gradient, this RNN is unfolded completely, and the gradient can be obtained by differentiating this unfolded structure. This analytic differentiation is done automatically by the Theano library.

Optimizing the six parameters in v_0 and ω_0 took only 88 iterations with gradient descent and backpropagation through time. Optimizing this problem with CMA-ES (Hansen, 2006),

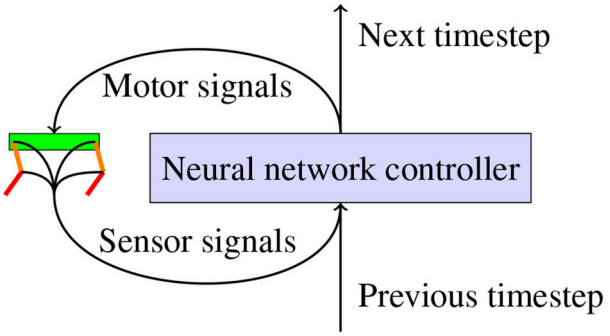


FIGURE 1 | Illustration of how a closed loop neural network controller would be used to actuate a robot. The neural network receives sensor signals from the sensors on the robot and uses these to generate motor signals which are sent to the servo motors. The neural network can also generate a signal which it can use at the next timestep to control the robot.

a state of the art derivative-free optimization method, took 2,422 iterations. Even when taking the time to compute the gradient into account, the optimization with gradient descent takes 16.3 s, compared to 59.9 s with CMA-ES. This result shows that gradient-based optimization of kinematic systems can in some cases already outperform gradient-free optimization algorithms from as little as six parameters.

2.2. Policy Search

To evaluate the relevance of our differentiable physics engine, we use a neural network as a general controller for a robot, as shown in **Figure 1**. We consider a general robot model in a discrete-time dynamical system $\mathbf{x}^{t+1} = f_{\text{ph}}(\mathbf{x}^t, \mathbf{u}^t)$ with a task cost function of $l(\mathbf{x}^t, \mathbf{p})$, where $\mathbf{x}^t$ is the state of the system at time t and $\mathbf{u}^t$ is the input of the system at time t. $\mathbf{p}$ provides some freedom in parameterizing the loss. If X^t is the trajectory of the state up to time $t-1$, the goal is to find a policy $u^t = \pi(X^t)$ such that we minimize the loss $\mathcal{L}_\pi$.

$$\begin{aligned} \mathcal{L}_\pi &= \sum_{t=0}^{T} l(\mathbf{x}^t, \mathbf{p}) \\ \text{s.t.} \quad \mathbf{x}^{t+1} &= f_{\text{ph}}(\mathbf{x}^t, \pi(X^t)) \quad \text{and} \quad \mathbf{x}^0 = x^{\text{init}} \end{aligned} \tag{2}$$

In previous research, finding a gradient for this objective has been described as presenting challenges (Mordatch and Todorov, 2014). An approximation to tackle these issues has been discussed in Levine and Koltun (2013).

We implement this equation into an automatic differentiation library, ignoring these challenges in finding the analytic gradient altogether. The automatic differentiation library, Theano in our case, analytically derives this equation and compiles code to evaluate both the equation and its gradient.

Unlike in previous approaches such as iLQR (Todorov and Li, 2005) and DDP (Bertsekas et al., 2005), we propose not to use this gradient to optimize a trajectory, but to use the gradient obtained to optimize a general controller parameterized by a neural network. This limits the amount of computation at execution time, but requires the optimization of a harder problem with more parameters.

We define our controller as a deep neural network g_{deep} with weights $\mathbf{W}$. We do not pass all information X^t to this neural network, but only a vector of values $\mathbf{s}^t$ observed by the modeled sensors $s(\mathbf{x}^t)$. We also provide our network with (some of the) task-specific parameters $\mathbf{p}'$. Finally, we add a recurrent connection to the controller in the previous timestep $\mathbf{h}^t$. Therefore, our policy is the following:

$$\begin{aligned} \pi(X^t) &= g_{\text{deep}}(s(\mathbf{x}^t), \mathbf{h}^t, \mathbf{p}' \mid \mathbf{W}) \\ \text{s.t.} \quad \mathbf{h}^t &= h_{\text{deep}}(s(\mathbf{x}^{t-1}), \mathbf{h}^{t-1}, \mathbf{p}' \mid \mathbf{W}) \quad \text{and} \quad \mathbf{h}^0 = 0 \end{aligned} \tag{3}$$

Notice the similarity between Equations (2)and (3). Indeed, the equations for recurrent neural networks (RNN) in Equation (3) are very similar to the ones of the loss of a physical model in Equation (2). Therefore, we optimize this entire system as an RNN unfolded over time, as illustrated in **Figure 2**. The weights $\mathbf{W}$ are optimized with stochastic gradient descent. The gradient required for that is the Jacobian $d\mathcal{L}/d\mathbf{W}$, which is found with automatic differentiation software.

We have now reduced the problem to a standard deep learning problem. We need to train our network g_{deep} on a sufficient amount of samples x^{init} and for a sufficient amount of sampled tasks $\mathbf{p}$ in order to get adequate generalization. Standard RNN regularization approaches could also improve this generalization. We reckon that generalization of g_{deep} to more models f_{ph}, in order to ease the transfer of the controller from the model to the real system, is also possible (Hermans et al., 2014), but it is outside the scope of this paper.

3. RESULTS

3.1. Quadrupedal Robot: Computing Speed

To verify the speed of our engine, we also implemented a small quadrupedal robot model, as illustrated in **Figure 3B**. This model has a total of 81 sensors, e.g., encoders and an inertial measurement unit (IMU). The servo motors are controlled in a closed loop by a small neural network g_{deep} with a number of parameters, as shown in **Figure 2**. The gradient is the Jacobian of $\mathcal{L}$, the total traveled distance of the robot in 10 s , differentiated with respect to all the parameters of the controller $\mathbf{W}$. This Jacobian is found by using BPTT and propagating all 10 s back. The time it takes to compute this traveled distance and the accompanying Jacobian is shown in **Table 1**. We include both the computation time with and without the gradient, i.e., both the forward and backward pass and the forward pass alone. This way, the numbers can be compared to other physics engines, as those only calculate without gradient. Our implementation and our model can probably be made more efficient, and evaluating the gradient can probably be made faster a similar factor.

When only a single controller is optimized, our engine runs more slowly on GPU than on CPU. To tackle this issue, we implemented batch gradient descent, which is commonly used in complex optimization problems. In this case, by batching our robot models, we achieve significant acceleration on GPU. Although backpropagating the gradient through physics slows

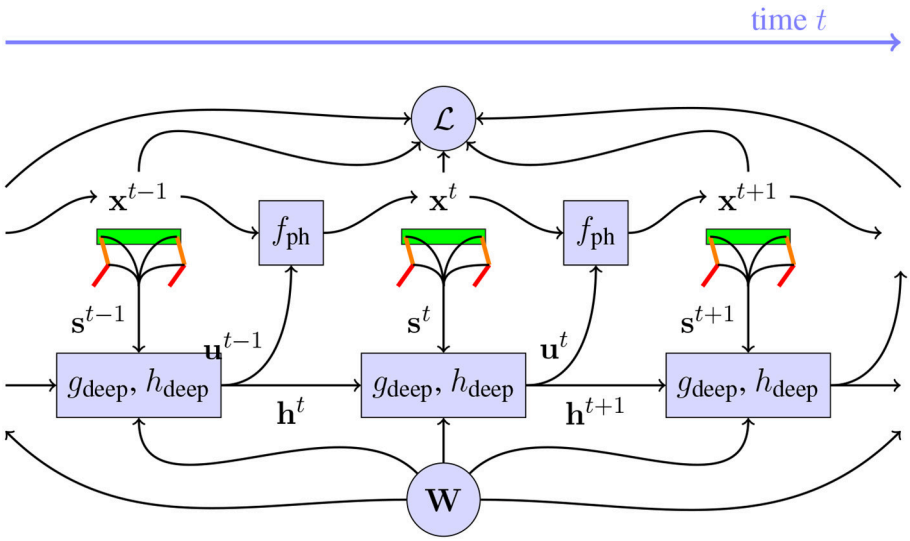


FIGURE 2 | Illustration of the dynamic system with the robot and controller, after unrolling over time. The neural networks g_{deep} and h_{deep} with weights W receive sensor signals s^t from the sensors on the robot and use these to generate motor signals u^t which are used by the physics engine f_{ph} to find the next state of the robot in the physical system. These neural networks also have a memory, implemented with recurrent connections h^t. From the state x^t of these robots, the loss $\mathcal{L}$ can be found. In order to find $d\mathcal{L}/dW$, every block in this chart needs to be differentiable. The contribution of this paper, is to implement a differentiable f_{ph}, which allows us to optimize W to minimize $\mathcal{L}$ more efficiently than was possible before.

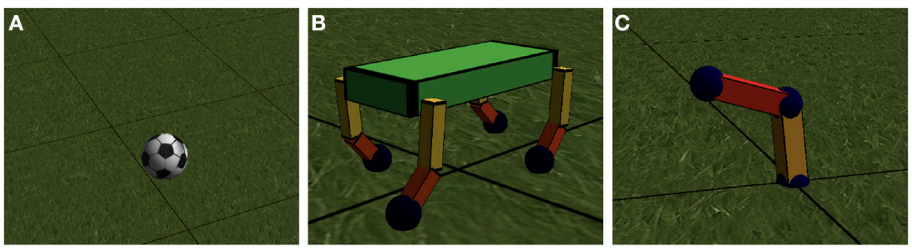


FIGURE 3 | (A) Illustration of the ball model used in the first task. **(B)** Illustration of the quadruped robot model with 8 actuated degrees of freedom, 1 in each shoulder, 1 in each elbow. The spine of the robot can collide with the ground, through 4 spheres in the inside of the cuboid. **(C)** Illustration of the robot arm model with 4 actuated degrees of freedom.

down the computations by roughly a factor 10, this factor only barely increases with the number of parameters in our controller.

Combining this with our previous observation that fewer iterations are needed when using gradient descent, our approach can enable the use of gradient descent through physics for highly complex deep neural network controllers with millions of parameters. Also note that by using a batch method, a single GPU can simulate about 864,000 model seconds per day, or 86,400,000 model states. This should be plenty for deep learning. It also means that a single simulation step of a single robot, which includes collision detection, solving the LCP problem, integrating the velocities and backpropagating the gradient through it all, takes about 1 ms on average. Without the backpropagation, this process is only about seven times faster.

3.2. 4 Degree of Freedom Robot Arm

As a first test of optimizing robot controllers, we implemented a four degree of freedom robotic arm, as depicted in **Figure 3C**. The bottom of the robot has a 2 degrees of freedom actuated universal joint; the elbow has a 2 degree of freedom actuated joint as well. The arm is 1 m long, and has a total mass of 32 kg. The servos have a gain of $30\,\text{s}^{-1}$, a torque of 30 Nm and a velocity of $45^\circ\,\text{s}^{-1}$.

For this robot arm, we train controllers for a task with a gradually increasing amount of difficulty. To be able to train our parameters, we have to use a couple of tricks often used in the training of recurrent neural networks.

- We choose an objective which is evaluated at every time step and then averaged, rather than at specific points of the simulation. This approach vastly increases the number of samples over which the gradient is averaged, which in turn makes the gradient direction more reliable (Sjöberg et al., 1995).
- The value of the gradient is decreased by a factor $\alpha < 1$ at every time step. This trick has the effect of a prior. Namely, events further in the past are less important for influencing

current events, because intermediate events might diminish their influence altogether. It also improves robustness against exploding gradients (Hermans et al., 2014).

- We initialize the controller intelligently. We do not want the controller to shake the actuators violently and explore outside the accurate domain of our simulation model. Therefore, our controllers are initialized with zeros such that they only output zeros at the start of the simulation. The initial policy is the zero policy.
- We constraint the size of the gradient to an L2-norm of 1. This makes sure that gradients close to discontinuities in the fitness landscape do not push the parameter values too far away, such that everything which was learned is forgotten (Sutskever, 2013).

3.2.1. Reaching a Fixed Point

A first simple task, is to have a small neural net controller learn to move the controller to a certain fixed point in space, at coordinates (0.5 m; 0.5 m; 0.5 m). The objective we minimize for this task, is the distance between the end effector and the target point, averaged over the 8 s we simulate our model.

We provide the controller with a single sensor input, namely the current distance between the end effector and the target point. Input is not required for this task, as there are solutions for which the motor signals are constant in time. However, this would not necessarily be the optimal approach for minimizing the average distance over time, it only solves the distance at the end of the simulation, but does not minimize the distance during the trajectory to get at the final position.

As a controller, we use a dense neural network with 1 input, 2 hidden layers of 128 units with a rectifier activation function, and 4 outputs with an identity activation function. Each unit in the neural network also has a bias parameter. This controller has 17,284 parameters in total. We disabled the recurrent connections $\mathbf{h}^t$.

We use gradient descent with a batch size of 1 robot for optimization, as the problem is not stochastic in nature. The parameters are optimized with Adam's rule (Kingma and Ba, 2014) with a learning rate of 0.001. Every update step with this method takes about 5 s on CPU. We find that the controller comes within 4 cm of the target in 100 model evaluations, and within 1 cm in 150 model evaluations, which is small compared to the 1 m arm of the robot. Moreover, the controller does find a more optimal trajectory which takes into account the sensor information.

Solving problems like these in fewer iteration steps than the number of parameters, is unfeasible with derivative free methods (Sjöberg et al., 1995). Despite that, we did try to optimize the same problem with CMA-ES. After a week of computing and 60,000 model evaluations, CMA-ES did not show any sign of improvement nor convergence, as it cannot handle the sheer amount of parameters. In performance, the policy went from a starting performance of 0.995 ± 0.330 m to a not significantly different 0.933 ± 0.369 m after the optimization. For this reason, we did not continue using CMA-ES as a benchmark in the further experiments.

3.2.2. Reaching a Random Point

As a second task, we sample a random target point in the reachable space of the end effector. We give this point as input v' to the controller, and the task is to again minimize the average distance between the end effector and the target point v. Our objective $\mathcal{L}$ is this distance averaged over all timesteps.

As a controller, we use a dense neural network comparable to the previous section, but this time with 3 inputs. Note that this is an open loop controller, which needs to control the system to a set point given as input. We used 3 hidden layers with 1,024 units each, so the controller has 2,107,396 parameters in total. This is not necessary for this task, but we do it like this to demonstrate the power of this approach. In order to train for this task, we use a batch size of 128 robots, such that every update step takes 58 s on GPU. Each simulation takes 8 s with a simulation step of 0.01 s. Therefore, the gradient on the parameters of the controllers has been averaged over 51,200 timesteps at every update step. We update the parameters with Adam's rule, where we scale the learning rate with the average error achieved in the previous step.

We find that it takes 576 update steps before the millions of parameters are optimized, such that the end effector of the robot is on average <10 cm of target, 2,563 update steps before the error is <5 cm.

3.3. A Quadrupedal Robot: Revisited

Optimizing a gait for a quadrupedal robot is a problem of a different order, something the authors have extensive experience with Degrave et al. (2013, 2015) and Sproewitz et al. (2013). The problem is way more challenging and allows for a broad range of possible solutions. In nature, we find a wide variety of gaits, from hopping over trotting, walking and galloping. With hand tuning on the robot model shown in **Figure 3B**, we were able to obtain a trotting motion with an average forward speed of 0.7 m/s. We found it tricky to find a gait where the robot did not end up like an upside down turtle, as 75% of the mass of the robot is located in its torso.

As a controller for our quadrupedal robot, we use a neural network with 2 input signals $\mathbf{s}^t$, namely a sine and a cosine signal with a frequency of 1.5 Hz. On top of this, we added 2 hidden layers of 128 units and a rectifier activation function. As output layer, we have a dense layer with 8 units and a linear activation function, which has as input both the input layer and the top layer of the hidden layers. In total, this controller has 17,952 parameters. Since the problem is not stochastic in nature, we use a batch size of 1 robot. We initialize the output layer with zero weights, so the robot starts the optimization in a stand still position.

We optimize these parameters to maximize the average velocity of the spine over the course of 10 s of time in simulation. This way, the gradient used in the update step is effectively an average of the 1,000 time steps after unrolling the recurrent connections. This objective does not take into account energy use, or other metrics typically employed in robotic problems.

In only 500 model evaluations or about 1 h of optimizing on CPU, the optimization with BPTT comes up with a solution with a speed of 1.17 m/s. This solution is a hopping gait, with

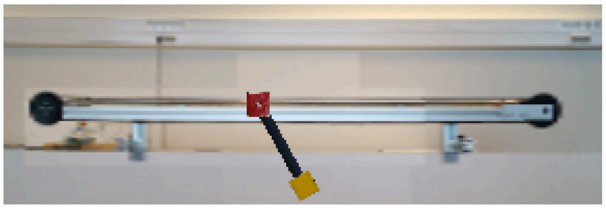


FIGURE 4 | A frame captured by the differentiable camera looking at the model of the pendulum-cart system. The resolution used is 288 by 96 pixels. All the textures are made from pictures of the actual system.

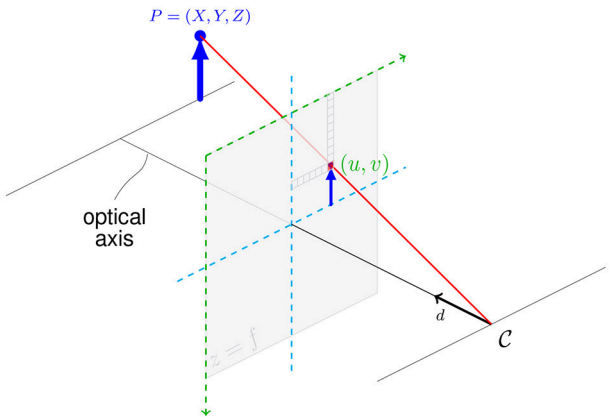


FIGURE 5 | The camera model used to convert the three dimensional point P into a two dimensional pixel on the projection plane (u, v).

a summersault every 3 steps[1], despite limiting the torque of the servos to 4 Nm on this 28.7 kg robot. For more life-like gaits, energy efficiency could be use as a regularization method. Evaluating these improvements are however outside the scope of this paper.

3.4. The Inverted Pendulum With a Camera as Sensor

As a fourth example, we implemented a model of the pendulum-cart system we have in our laboratorium. This pendulum-cart system is used for the classic control task of the underactuated inverted pendulum (Vaccaro, 1995). In this example however, a camera which is set up in front of the system is the only available information for the controller. It therefore has to observe the system it controls using vision, i.e., learning from pixels. A frame captured by this camera is shown in **Figure 4**.

In order to build this model, we implemented a renderer in our physics engine which converts the three dimensional scene into a two dimensional color image, as illustrated in **Figure 5**. In order to perform this operation in a differentiable way, we use a ray tracing approach rather than the more conventional rasterization pipeline. First we cast a set of lines from the point of our camera $\mathcal{C}$ in the direction $\vec{d}$ of the optical axis of the camera. These vectors are then converted with the pinhole camera model into a line going through the center of the pixel with the image coordinates (u, v) on the projection plane. Each of these rays is then intersected with every object in the scene to find the texture and corresponding sample location to sample from in the scene's texture array. From all intersections a single ray makes, all but the one closest in front of the projection plane is kept.

Each of the intersections is then converted to a color by bilinearly interpolating the scene's texture array, in a way similar to the approach used for the spatial transform layer (Jaderberg et al., 2015; Degrave et al., 2016a). This bilinear interpolation is necessary to make the frame captured by the camera differentiable to the state of the robot with non-zero derivatives. If the textures would have been a zero-order, pixelated approximation, then all the gradients would be zero analytically.

Using the above ray-tracing approach, we minimize the distance from the end of the pendulum to the desired point and regularize the speed of the pendulum. The memoryless deep controller receives the current image of the camera, in addition to two images from the past such that it can estimate velocity and acceleration. We observe that a controller with 1,065,888 parameters is able to learn to swing up and keep the pendulum stable after only 2,420 episodes of 3 model seconds. The complete optimization process took 15 h on 1 GPU. The resulting controller keeps the pendulum stable for more than 1 min[2]. In order to do this, the controller has learned to interpret the frames it receives from the camera and found a suitable control strategy.

Note that this would not have been possible using a physics engine such as mujoco, as these engines only allow differentiation through the action and the state, but does not allow to differentiate through the renderer. We want to stress that in this setup we solved the problem by backpropagating through both the computer vision in the form of the convolutional neural network, and the renderer in the form of the differentiable camera.

4. DISCUSSION

We implemented a modern engine which can run a 3D rigid body model, using the same algorithm as other engines commonly used to simulate robots, but we can additionally differentiate control parameters with BPTT. Our implementation also runs on GPU, and we show that using GPUs to simulate the physics can speed up the process for large batches of robots. We show that even complex sensors such as cameras, can be implemented and differentiated through, allowing for computer vision to be learned together with a control policy.

When initially addressing the problem, we did not know whether finding the gradient would be computationally tractable, let alone whether evaluating it would be fast enough to be beneficial for optimization. In this paper, we have demonstrated that evaluating the gradient is tractable enough to speed up optimization on problems with as little as six parameters. The

[1] A video is available at https://goo.gl/5ykZZe

[2] https://twitter.com/317070/status/821062814798331905

speed of this evaluation mainly depends on the complexity of the physics model and only slightly on the number of parameters to optimize. Therefore, our results suggest that this cost is dominated by the gain achieved from the combination of using batch gradient descent and GPU acceleration. Consequently, by using gradient descent with BPTT one can speed up the optimization processes often found in robotics, even for rather small problems, due to the reduced number of model evaluations required. Furthermore, this improvement in speed scales to problems with a lot of parameters. By using the proposed engine, finding policies for robot models can be done faster and in a more straightforward way. This method should allow for a new approach to apply deep learning techniques in robotics.

Optimizing the controller of a robot model with gradient-based optimization is equivalent to optimizing an RNN. After all, the gradient passes through each parameter at every time step. The parameter space is therefore very noisy. Consequently, training the parameters of this controller is a highly non-trivial problem, as it corresponds to training the parameters of an RNN. On top of that, exploding and vanishing signals and gradients cause far more challenging problems compared to feed forward networks.

In section 3.2, we already discussed some of the tricks used for optimizing RNNs. Earlier research shows that these methods can be extended to more complicated tasks than the ones discussed here (Sutskever, 2013; Hermans et al., 2014). Hence, we believe that this approach toward learning controllers for robotics applies to more complex problems than the illustrative examples in this paper.

TABLE 1 | Evaluation of the computing speed of our engine on a robot model controlled by a closed loop controller with a variable number of parameters.

		With gradient		Without gradient	
		CPU	GPU	CPU	GPU
SECONDS OF COMPUTING TIME REQUIRED TO SIMULATE A BATCH OF ROBOTS FOR 10 s					
1 robot	1,296 parameters	8.17	69.6	1.06	9.69
	1,147,904 parameters	13.2	75.0	2.04	9.69
128 robots	1,296 parameters	263	128	47.7	17.8
	1,147,904 parameters	311	129	50.4	18.3
MILLISECONDS OF COMPUTING TIME REQUIRED TO PERFORM ONE TIME STEP OF ONE ROBOT.					
1 robot	1,296 parameters	8.17	69.6	1.06	9.69
	1,147,904 parameters	13.2	75.0	2.04	9.69
128 robots	1,296 parameters	2.05	1.00	0.372	0.139
	1,147,904 parameters	2.43	1.01	0.394	0.143

We evaluated both on CPU (i7 5930K) and GPU (GTX 1080), both for a single robot optimization and for batches of multiple robots in parallel. The numbers are the time required in seconds for simulating the quadruped robot(s) for 10 s, with and without updating the controller parameters through gradient descent. Shorter times are colored in green, longer in red. The gradient calculated here is the Jacobian of the total traveled distance of the robot in 10 s, differentiated with respect to all the parameters of the controller. For comparison, the model has 102 states. It is built from 17 rigid bodies, each having 6 degrees of freedom. These states are constrained by exactly 100 constraints.

All of the results in this paper will largely depend on showing how these controllers will work on the physical counterparts of our models. Nonetheless, we would like to conjecture that to a certain extent, this gradient of a model is close to the gradient of the physical system. The gradient of the model is more susceptible to high-frequency noise introduced by modeling the system, than the imaginary gradient of the system itself. Nonetheless, it contains information which might be indicative, even if it is not perfect. We would theorize that using this noisy gradient is still better than optimizing in the blind and that the transferability to real robots can be improved by evaluating the gradients on batches of (slightly) different robots in (slightly) different situations and averaging the results. This technique has already been applied in Hermans et al. (2014) as a regularization method to avoid bifurcations during online learning. If the previous proves to be correct, our approach can offer an addition or possibly even an alternative to deep Q-learning for deep neural network controllers in robotics.

We can see the use of this extended approach for a broad range of applications in robotics. Not only do we think there are multiple ways where recent advances in deep learning could be applied to robotics more efficiently with a differentiable physics engine, we also see various ways in which this engine could improve existing angles at which robotics are currently approached:

- In this paper, we added memory by introducing recurrent connections in the neural network controller. We reckon that advanced, recurrent connections such as ones with a memory made out of LSTM cells (Hochreiter and Schmidhuber, 1997) can allow for more powerful controllers than the controllers described in this paper.
- Having general differentiable models should allow for an efficient system identification process (Bongard et al., 2006; Ha and Schmidhuber, 2018). The physics engine can find analytic derivatives to all model parameters. This includes masses and lengths, but also parameters which are not typically touched in system identification, such as the textures of the rigid body. As the approach could efficiently optimize many parameters simultaneously, it would be conceivable to find state dependent model parameters using a neural network to map the current state onto e.g., the friction coefficient in that state.
- Using a differentiable physics engine, we reckon that knowledge of a model can be distilled more efficiently into a forward or backward model in the form of a neural network, similar to methods such as used in Johnson et al. (2016) and Dumoulin et al. (2017). By differentiating through an exact model and defining a relevant error on this model, it should be possible to transfer knowledge from a forward or backward model in the differentiable physics engine to a forward or backward neural network model. Neural network models trained this way might be more robust than the ones learned from generated trajectories (Christiano et al., 2016). In turn, this neural model could then be used for faster but approximate evaluation of the model.

- Although we did not address this in this paper, there is no reason why only control parameters could be optimized in the process. Hardware parameters of the robot have been optimized the same way before (Jarny et al., 1991; Iollo et al., 2001; Hermans et al., 2014). The authors reckon that the reverse process is also true. A physics engine can provide a strong prior, which can be used for robots to learn (or adjust) their robot models based on their hardware measurements faster than today. You could optimize the model parameters with gradient descent through physics, to have the model better mimic the actual observations.
- Where adversarial networks are already showing their use in generating image models, we believe adversarial robotics training (ART) will create some inventive ways to design and control robots. Like in generative adversarial nets (GAN) (Goodfellow et al., 2014), where the gradient is pulled through two competing neural networks, the gradient could be pulled through multiple competing robots as well. It would form an interesting approach for swarm robotics, similar to previous results in evolutionary robotics (Sims, 1994; Pfeifer and Bongard, 2006; Cheney et al., 2014), but possibly faster.

AUTHOR CONTRIBUTIONS

The experiments were conceived by JDe, MH, JDa, and FW. Experiments were designed by JDe and MH. The data were analyzed by JDe with help of FW and JDa. The manuscript was mostly written by JDe, with comments and corrections from FW and JDa.

FUNDING

The research leading to these results has received funding from the Agency for Innovation by Science and Technology in Flanders (IWT). The NVIDIA Corporation donated the GTX 1080 used for this research.

ACKNOWLEDGMENTS

Special thanks to David Pfau for pointing out relevant prior art we were previously unaware of, and Iryna Korshunova for proofreading the paper. The original version of this article was previously published in preprint on arXiv (Degrave et al., 2016b).

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2016). TensorFlow: large-scale machine learning on heterogeneous systems. *arXiv [Preprint]. arXiv:1603.04467*. Available online at: https://arxiv.org/abs/1603.04467

Aguilar-Ibañez, C. (2017). Stabilization of the pvtol aircraft based on a sliding mode and a saturation function. *Int. J. Robust Nonlinear Control* 27, 843–859. doi: 10.1002/rnc.3601

Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., et al. (2016). Theano: a Python framework for fast computation of mathematical expressions. *arXiv [Preprint]. arXiv:1605.02688*. Available online at: https://arxiv.org/abs/1605.02688

Bertsekas, D. P., Bertsekas, D. P., Bertsekas, D. P., and Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control*, Vol. 1. Belmont, MA: Athena scientific.

Bongard, J., Zykov, V., and Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science* 314, 1118–1121. doi: 10.1126/science.1133687

Catto, E. (2009). "Modeling and solving constraints," in *Game Developers Conference* (Cologne).

Chappuis, D. (2013). Constraints derivation for rigid body simulation in 3D. Available online at: https://danielchappuis.ch/download/ConstraintsDerivationRigidBody3D.pdf

Cheney, N., Clune, J., and Lipson, H. (2014). Evolved electrophysiological soft robots. *ALIFE* 14, 222–229. doi: 10.7551/978-0-262-32621-6-ch037

Christiano, P., Shah, Z., Mordatch, I., Schneider, J., Blackwell, T., Tobin, J., et al. (2016). Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv [Preprint]. arXiv:1610.03518*.

Degrave, J., Burm, M., Kindermans, P. J., Dambre, J., and wyffels, F. (2015). Transfer learning of gaits on a quadrupedal robot. *Adapt. Behav.* 23, 69-82. doi: 10.1177/1059712314563620

Degrave, J., Burm, M., Waegeman, T., wyffels, F., and Schrauwen, B. (2013). "Comparing trotting and turning strategies on the quadrupedal oncilla robot," in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (Shenzhen: IEEE), 228–233.

Degrave, J., Dieleman, S., Dambre, J., and wyffels, F. (2016a). Spatial chirp-Z transformer networks. in *European Symposium on Artificial Neural Networks (ESANN)* (Bruges).

Degrave, J., Hermans, M., Dambre, J., and Wyffels, F. (2016b). A differentiable physics engine for deep learning in robotics. *arXiv [Preprint]. arXiv:1611.01652*. Available online at: https://arxiv.org/abs/1611.01652

Dumoulin, V., Shlens, J., and Kudlur, M. (2017). "A learned representation for artistic style," in *International Conference on Learning Representations (ICLR)*.

Erez, T., Tassa, Y., and Todorov, E. (2015). "Simulation tools for model-based robotics: comparison of bullet, havok, mujoco, ode, and physx," in *International Conference on Robotics and Automation (ICRA)* (Seattle, WA: IEEE), 4397–4404.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (Montreal, QC), 2672–2680.

Grzeszczuk, R., Terzopoulos, D., and Hinton, G. (1998). "Neuroanimator: fast neural network emulation and control of physics-based models," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (Orlando, FL: ACM), 9–20.

Ha, D., and Schmidhuber, J. (2018). World models Version 1.1. *arXiv [Preprint]. arXiv:1803.10122*. doi: 10.5281/zenodo.1207631

Hansen, N. (2006). "The cma evolution strategy: a comparing review," in *Towards a New Evolutionary Computation* (Berlin; Heidelberg: Springer), 75–102. .

Hermans, M., Schrauwen, B., Bienstman, P., and Dambre, J. (2014). Automated design of complex dynamic systems. *PLoS ONE* 9:e86696. doi: 10.1371/journal.pone.0086696

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780.

Hubbard, P. M. (1996). Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graph.* 15, 179–210.

Iollo, A., Ferlauto, M., and Zannetti, L. (2001). An aerodynamic optimization method based on the inverse problem adjoint equations. *J. Comput. Phys.* 173, 87–115. doi: 10.1006/jcph.2001.6845

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). "Spatial transformer networks," in *Advances in Neural Information Processing Systems* (Montreal, QC), 2017–2025.

Jarny, Y., Ozisik, M., and Bardon, J. (1991). A general optimization method using adjoint equation for solving multidimensional inverse heat conduction. *Int. J. Heat Mass Trans.* 34, 2911–2919.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). Caffe: convolutional architecture for fast feature embedding. *arXiv [Preprint]. arXiv:1408.5093*. Available online at: https://arxiv.org/abs/1408.5093

Johnson, J., Alahi, A., and Fei-Fei, L. (2016, October). "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision* (Cham: Springer), 694–711.

Jourdan, F., Alart, P., and Jean, M. (1998). A gauss-seidel like algorithm to solve frictional contact problems. *Comp. Methods Appl. Mech. Engin.* 155, 31–47.

Kingma, D. P., and Ba, J. (2014). "Adam: a method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)* (Banff).

Levine, S., and Koltun, V. (2013). Variational policy search via trajectory optimization. in *Advances in Neural Information Processing Systems* (Lake Tahoe, NV), 207–215.

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Robot. Res.* 37, 421–436. doi: 10.1177/0278364917710318

Meda-Campana, J. A. (2018). Estimation of complex systems with parametric uncertainties using a jssf heuristically adjusted. *IEEE Latin Am. Trans.* 16, 350–357.

Mirtich, B. (1998). V-clip: Fast and robust polyhedral collision detection. *ACM Trans. Graph.* 17, 177–208.

Mordatch, I., and Todorov, E. (2014). "Combining the benefits of function approximation and trajectory optimization," in *Robotics: Science and Systems (RSS)* (Rome).

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., et al. (2017). "Automatic differentiation in pytorch," in *Autodiff Workshop*.

Pfeifer, R., and Bongard, J. (2006). *How the Body Shapes the Way we Think: A New View of Intelligence*. Cambridge: MIT press.

Premerlani, W., and Bizard, P. (2009). "Direction cosine matrix IMU: theory," in *DIY Drone: USA* (Evendale), 13–15.

Sims, K. (1994). Evolving 3d morphology and behavior by competition. *Artif. Life* 1, 353–372.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.-Y., et al. (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica* 31, 1691–1724.

Sproewitz, A., Tuleu, A., D'Haene, M., Möckel, R., Degrave, J., Vespignani, M., et al. (2013). "Towards dynamically running quadruped robots: performance, scaling, and comparison," in *Adaptive Motion of Animals and Machines* (Darmstadt), 133–135.

Stewart, D., and Trinkle, J. C. (2000). An implicit time-stepping scheme for rigid body dynamics with coulomb friction. in *International Conference on Robotics and Automation (ICRA)*, Vol. 1, (IEEE), 162–169.

Sutskever, I. (2013). *Training Recurrent Neural Networks*. PhD thesis, University of Toronto.

Todorov, E., Erez, T., and Tassa, Y. (2012). "Mujoco: a physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE), 5026–5033.

Todorov, E., and Li, W. (2005). "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference, 2005. Proceedings of the 2005* (IEEE), 300–306.

Vaccaro, R. J. (1995). *Digital Control: A State-Space Approach*, Vol. 196. New York, NY: McGraw-Hill.

Conflict of Interest Statement: JD is currently employed at Deepmind.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 28 March 2019
doi: 10.3389/fnbot.2019.00009

Body Randomization Reduces the Sim-to-Real Gap for Compliant Quadruped Locomotion

Alexander Vandesompele[1]*, *Gabriel Urbain*[1], *Hossain Mahmud*[2], *Francis wyffels*[1] and *Joni Dambre*[1]

[1] AIRO, Electronics and Information Systems Department, Ghent University-Imec, Ghent, Belgium, [2] fortiss GmbH, Munich, Germany

Designing controllers for compliant, underactuated robots is challenging and usually requires a learning procedure. Learning robotic control in simulated environments can speed up the process whilst lowering risk of physical damage. Since perfect simulations are unfeasible, several techniques are used to improve transfer to the real world. Here, we investigate the impact of randomizing body parameters during learning of CPG controllers in simulation. The controllers are evaluated on our physical quadruped robot. We find that body randomization in simulation increases chances of finding gaits that function well on the real robot.

Keywords: compliant robotics, quadruped control, knowledge transfer, simulation-reality gap, dynamics randomization

OPEN ACCESS

Edited by:
Hong Qiao,
University of Chinese Academy of Sciences (UCAS), China

Reviewed by:
Xue Wen Rong,
Shandong University, China
Takeshi Kano,
Tohoku University, Japan

***Correspondence:**
Alexander Vandesompele
alexander.vandesompele@ugent.be

Received: *26 November 2018*
Accepted: *05 May 2019*
Published: *28 March 2019*

Citation:
Vandesompele A, Urbain G, Mahmud H, wyffels F and Dambre J (2019) Body Randomization Reduces the Sim-to-Real Gap for Compliant Quadruped Locomotion. Front. Neurorobot. 13:9. doi: 10.3389/fnbot.2019.00009

1. INTRODUCTION

Compliant robots can provide many benefits over rigid robots (Pfeifer and Iida, 2007). They are more versatile and posses an inherently greater capacity to deal with different environments or with changing body properties due to wear and tear. Additionally, they can be more energy-efficient, safer for humans and less costly. The drawback is that they are generally more difficult to control than rigid robots.

Currently, state-of-the-art robots are usually made of rigid components (e.g., Raibert et al., 2008; Barasuol, 2013; Park et al., 2017). The rigid and well characterized body parts allow for controllers to be explicitly designed, based on accurate knowledge of the robot's physical properties. There are, however, some severe limitations to this approach. It is prohibitively difficult to design controllers that can adapt to a wide variety of environments and to the changing body properties due to wear and tear over the robot's lifetime. Well characterized and reliable components also come at a high cost.

The same approach cannot be applied to compliant robots, as their body parts can interact highly non-linearly with each other and the robots environment. This makes it difficult to accurately model their physical properties. Machine learning approaches are promising to the development of adaptive controllers for compliant robots. The combination of machine learning and compliant robotics may lead to robots moving out of highly standardized environments and into daily life at a cost that is affordable for consumers.

In the field of robot locomotion, machine learning techniques have been increasingly successful in developing adaptive robot controllers in simulation. Especially in the field of deep reinforcement learning, there have been some significant improvements recently (Heess et al., 2017; Peng et al., 2017). These controllers are usually learned in simulation and not on the physical robot. Learning only on the robot is challenging for multiple reasons, it is usually time-costly and unoptimized

controllers may damage the robot. While it is impossible to simulate the real world, it is desirable to optimize controllers as far as possible before training on the physical robot. Particularly, in the case of a locomotion controller, it is desirable to start on the physical robot with a stable gait to prevent damage.

1.1. Related Work

The transfer of knowledge obtained in one domain to a new domain is important to speed up learning. Knowledge transfer can be applied across tasks, where knowledge from a learned task is utilized to speed up learning a new task by the same model (Hamer et al., 2013; Um et al., 2014). For instance, transfer of a quadruped gait learned in a specific environment, speeds up learning in other environments (Degrave et al., 2015). Knowledge transfer can also be applied across models, for instance if knowledge obtained by a first robot is utilized by a second robot (Gupta et al., 2017) or if a model is trained in simulation and then applied to a physical robot (Peng et al., 2018). However, the transfer of knowledge from simulation to reality has proven challenging for locomotion controllers due to discrepancies between simulation and reality, the so-called *simulation-reality gap* (Lipson and Pollack, 2000). This gap can easily cause a controller that is optimized in simulation to fail in the real world. Different methods have been developed to decrease the gap, they can generally be divided into two categories: (i) improving simulation accuracy and (ii) improving controller robustness.

System identification improves simulation accuracy by tuning the simulation parameters to match the behavior of the physical system. In the *embodiment theory* framework (Füchslin et al., 2013), the relation between environment, body and controller is described from a dynamical view point, where each entity can be modeled as a non-linear filter. Improving the simulator accuracy is then reduced to matching the transfer function of these filters. Urbain et al. (2018) provides an automated and parametrized calibration method that improves simulation accuracy by treating both the physical robot and its parametrized model as black box dynamical systems. It optimizes the similarity between the transfer functions by matching their sensor response to a given actuation input.

Similarly, simulation accuracy can be improved with machine learning techniques. For instance, in computer vision tasks (e.g., Taigman et al., 2016; Bousmalis et al., 2017) and visually guided robotic grasping tasks (Bousmalis et al., 2018), synthetic data has been augmented with generative adversarial networks (GANs). The augmentation improves the realism of the synthetic data and hence results in better models.

Another approach for minimizing the simulation-reality gap is by increasing robustness of the learned controllers. This can be achieved by perturbing the simulated robot during learning or by adding noise to the simulated environment (*domain randomization*, Jakobi, 1998; Tobin et al., 2017). The assumption is that if the model is trained on a sufficiently broad range of simulated environments, the real world will seem like just another variation to the model. Similarly, *dynamics randomization* is achieved by randomizing physical properties. Tan et al. (2018) found that dynamics randomization decreased performance but increased stability of a non-compliant quadruped robot. In Mordatch et al. (2015), optimization on ensembles of models instead of only the nominal model enables functional gaits on a small humanoid. In Peng et al. (2018), dynamics randomization was necessary for sim-to-real transfer of a robotic arm controller.

1.2. Our Approach

Whereas, Tan et al. (2018) observed the benefit of dynamics randomization for quadrupedal gait stability, the platform used is a stable, commercial robot used in a non-compliant manner. Passive compliance and underactuation are considered important for robots to cope with a broad range of real-world environments (Pfeifer et al., 2012; Laschi and Cianchetti, 2014). However, the difficulty of modeling the robot accurately increases with compliance and underactuation as well as with the use of low-cost components, exacerbating the simulation-reality gap. In this work we investigate the impact of dynamics randomization on controller robustness for compliant quadruped locomotion.

Measuring the robots physical properties does not necessarily translate into a good model. Especially with compliant robots, the dynamics of the model may be different from the physical robot. Therefor, we use a calibration method that focuses on replicating the dynamics, as described in a previous paper Urbain et al. (2018).

Using the calibrated model, we investigate if and how body randomization reduces the simulation-reality gap. For this purpose, we restrain ourselves to a straightforward controller optimization: a parametrized *central pattern generator* (CPG) optimized with an evolutionary strategy (the *CMA-ES* algorithm). The optimization is repeated for varying degrees of body randomization and subsequently tested on the physical robot. The randomization is applied to body parameters critical for the robot dynamics: mass distribution, spring stiffness and foot friction.

We observed that randomization of body parameters on average improves the stability of gaits when applied to the physical robot. Additionally, the used method is relatively straightforward to implement.

2. MATERIALS AND METHODS

2.1. Robot

The robot used for this paper is an update of the Tigrillo robot (Willems et al., 2017) as described by Urbain et al. (2018) (**Figure 1A**). Tigrillo is a low-cost platform built with off-the-shelf components and a structure laser cut out of ABS. It is developed for researching compliance in quadrupeds and has underactuated legs. Each hip joint is actuated with a Dynamixel RX-24F servomotor. The knee joints are passive compliant due to mounted springs (**Figure 1B**), which can be replaced to tune the passive compliance properties. The angle of the passive joints is measured with Hall sensors and rare-earth magnets placed on respectively the upper and lower leg parts. The Hall sensor will output a voltage between 0 and 5 V proportionally to the magnetic field. As the sensed magnetic field varies non-linearly with the distance to the magnet, the sensor provides us with non-linear body feedback. The total weight is 950 g and the

FIGURE 1 | (A) The Tigrillo robot used in this paper (left) and its parametrized model in Gazebo (right). **(B)** Zoom on a leg with a spring loaded on the knee joint. M denotes the magnet, H denotes the Hall sensor.

robot fits in a box of 30 × 18 cm. The front legs are 15 cm apart and the hind legs 11 cm. A mounted *Raspberry Pi 3* allows wireless control of the robot from a remote computer. Actuator and sensor communication runs on the *Robotic Operating System (ROS*[1]*)*.

2.2. Calibration

The goal of the calibration process is to tune a simulated model to increase similarity in dynamics of the model and robot. The Tigrillo platform has a parametric model (**Figure 1**) that is simulated in the *Neurorobotics platform* (NRP) (Falotico et al., 2017), using Gazebo configured with ODE (Drumwright et al., 2010) physics engine. The model is calibrated using the calibration method detailed by Urbain et al. (2018). This method is an automated procedure in which both the model and real robot are considered sensor-to-actuator transfer functions. As the model is parametrized, its transfer function can be adapted by tuning the parameters.

We start with learning the sensor-to-actuator transfer function from the physical robot by recording the Hall sensor activity in response to an actuation pattern $a(t)$. The actuation pattern is chosen to be a succession of sine waves at three different frequencies (0.4, 0.8, and 1.6 Hz). In order to calibrate the model such that it behaves similarly to the real robot during actual gaits, the sine waves are also used in anti-phase between the front and hind legs, creating bounding-like movement. Hence, in total six actuation patterns are used in the calibration procedure. To reduce sensor noise, an average (N=5) of multiple recordings is used as the target signal y. **Figure 3** shows the actuation and corresponding sensor signals for the legs of the physical robot. The high frequency event in the actuation signal for the front legs at the transition from high to low frequency (15th s) is an artifact caused by the signal generator. It does not significantly impact the calibration procedure as it is an event of short duration.

Next, we want to tune the body parameters of the model to achieve a similar sensor-to-actuator transfer function. We start with an uncalibrated model based on the measured physical properties (see diagram in **Figure 2**). Then, *covariance matrix adaptation evolutionary strategy (CMA-ES)* is applied for the parameter search. The included parameters θ are those observed

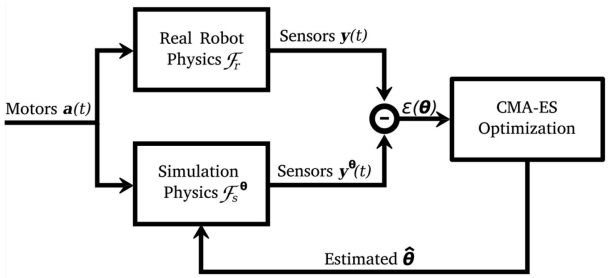


FIGURE 2 | Diagram of the calibration procedure. CMA-ES optimization minimizes the difference between the sensor recording from the robot and the model ($y(t)$ and $y^{\theta}(t)$, respectively), by tuning the model parameters ($\hat{\theta}$). Figure adapted from Urbain et al. (2018).

critical for the dynamic behavior and are listed in **Table 1**. The indices f and h refer to the front and hind part of the body, respectively. Parameter θ_m is the mass of the main body part on the front and hind side, θ_μ is the friction coefficient of the feet, and θ_k the spring constant indicating spring stiffness. The contact depth θ_d is the minimum depth before a contact correction impulse is applied. Parameter θ_c is the compression tolerance, which allows for the minimum angle of the passive joint to be smaller than the spring length, simulating spring compression.

A more detailed description of the *CMA-ES* algorithm can be found in Hansen (2006). It is an evolutionary algorithm that samples solutions from a multi-variate normal distribution. Every iteration, the mean and the covariance matrix of the distribution are updated. The mean is updated to increase the likelihood of previously successful solutions. The covariance matrix is updated to increase the likelihood of a previously successful search step. *CMA-ES* is well suited for a search space with multiple local minima. It requires few initial parameters and doesn't require derivation of the search space.

CMA-ES will minimize the error $\varepsilon(\theta)$, here chosen to be the root mean square error (*RMSE*) with y being the target sensor signal as recorded on the robot and $\hat{y}$ the sensor signal recorded in simulation:

$$\hat{\theta} = \arg\min_{\theta} \varepsilon(\theta) \tag{1}$$

[1]http://wiki.ros.org/

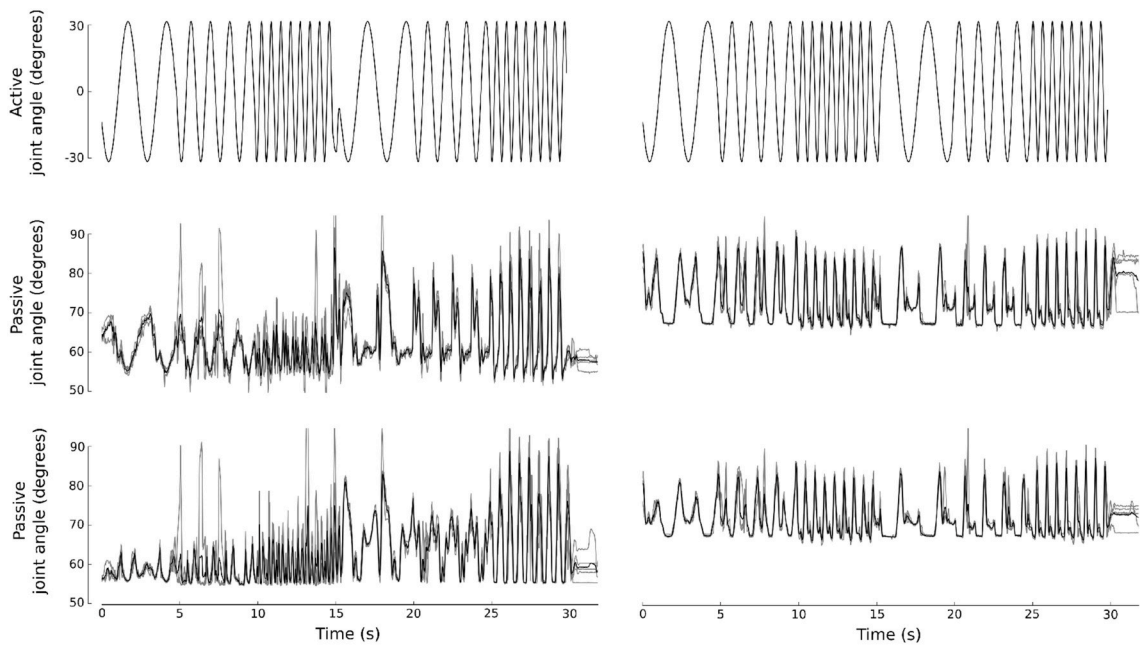


FIGURE 3 | Characterization of the robot dynamics. The robot was actuated with a pattern of sine waves **(top row)**. The front legs **(left)** and hind legs **(right)** were actuated in phase firstly and in antiphase subsequently. The bottom rows show the Hall sensor readout in response to the actuation pattern for the front and hind legs (**left and right**, respectively). An average of 5 trials was used as target signal during model calibration.

TABLE 1 | Parameters included in the calibration procedure with *CMA-ES*.

Parameter	Description	Range	Unit
θ_{mf}	Front mass	[0.1, 0.5]	kg
θ_{mh}	Hind mass	[0.1, 0.5]	kg
$\theta_{\mu f}$	Front feet friction coefficient	$[10^{-3}, 2.]$	*NA*
$\theta_{\mu h}$	Hind feet friction coefficient	$[10^{-3}, 2.]$	*NA*
θ_{df}	Front feet contact depth	$[10^{-4}, 10^{-2}]$	m
θ_{dh}	Hind feet contact depth	$[10^{-4}, 10^{-2}]$	m
θ_{kf}	Front legs spring constant	$[50, 10^3]$	N/m
θ_{kh}	Hind legs spring constant	$[50, 10^3]$	N/m
θ_{cf}	Front compression tolerance	[0.92, 0.98]	mm
θ_{ch}	Hind compression tolerance	[0.7, 1.2]	mm

$$\varepsilon(\theta) = \sqrt{\frac{\Sigma_{i=1}^{n}\left(\hat{y} - y\right)^2}{n}} \tag{2}$$

2.3. Gait Search

2.3.1. Central Pattern Generator

With the calibrated model, a controller is optimized in the same simulation environment. The controller is modeled by a parametrized CPG, based on the open-loop CPG introduced by Gay et al. (2013). The CPG is described by three equations:

$$\begin{aligned} \dot{r} &= \gamma(\mu - r^2)r \\ \dot{\phi} &= \omega \\ \lambda &= r\cos(\phi_L) + o, \end{aligned} \tag{3}$$

Where r describes the radius of the oscillator and ϕ the current phase. Both are used to calculate the actual control value λ in degrees. μ is the target amplitude of the oscillator and γ is a positive gain that defines the convergence speed of the radius to the target amplitude. ω is the radial frequency of the oscillator and o the offset. ϕ_L is a filter applied on the phase of the oscillator, the formula of which is different for the swing and stance phase of the control as determined by the duty factor (d):

$$\phi_L = \begin{cases} \dfrac{\phi_{2\pi}}{2d} & \text{if} \quad \phi_{2\pi} < 2\pi d \\ \dfrac{\phi_{2\pi} + 2\pi(1-2d)}{2(1-d)} & \text{otherwise} \end{cases} \tag{4}$$
$$\text{and } \phi_{2\pi} = \phi \pmod{2\pi}$$

The Tigrillo platform has four actuated joints that are controlled by four phase-coupled CPGs. One leg, the front left, is chosen as reference leg and three phase offset (po) parameters describe the phase difference of the remaining 3 legs to the reference leg. This is implemented by adding a term to the formula for the phase (ϕ) in Equation (3). For instance, for the coupling between the front left and front right oscillators:

$$\dot{\phi_{fr}} = \omega + w_{fr} sin(\phi_{fl} - \phi_{fr} - po_{fr}) \tag{5}$$

where w_{fr} is the coupling strength.

2.3.2. Gait Search With CMA-ES

The CMA-ES algorithm is used again to optimize the CPG controller. The search space consist of a subset of the CPG parameters. To enforce a walking gait, the search space is constrained to the set of parameters as detailed in **Table 2**. The walking gait is characterized by a phase offset among the legs that results in asymmetry along the transverse axis. Additionally, the Tigrillo robot has no feet retraction mechanism in its underactuated legs. Consequently, maintaining balance during a walking-like gait presents a challenge for this platform. The frequency ω is fixed at 2π radian/s (1 Hz).

CMA-ES as described by Hansen (2006) was used to perform the optimization, but with a larger population size ($N = 20$) to increase chance of avoiding local optima. Each solution is evaluated for 10 s of simulation time. As score function the distance of the model from origin after 10 s is used. After convergence of the *CMA-ES* algorithm, the best performing individual of the final generation is chosen as the final solution. Hence each optimization resulted in one set of CPG parameters that corresponds to a gait.

To investigate the effect of randomizing body morphology on transferability, *CMA-ES* optimizations were performed with varying levels of randomization of body properties deemed critical for the gait dynamics: θ_k, θ_μ, and θ_m. The parameters are sampled from a Gaussian distribution with the mean value μ taken from the calibrated model and the randomization parameter ψ affecting the standard deviation σ of the Gaussian distribution in a parameter dependent fashion (see **Figure 4** for an example). Given ψ, the standard deviation is obtained by the following equations, for the parameters θ_k and θ_m:

$$\sigma = \psi\mu \tag{6}$$

And, for the parameters θ_μ:

$$\sigma = 2\psi\mu \tag{7}$$

For θ_m, the mass of the main front body part is sampled from the Gaussian distribution and the mass of the main hind body part is adapted such that the total mass remains constant, varying only the mass distribution. θ_k and θ_μ are sampled independently per leg, hence each individual has 9 variable parameters. Because the noisy body parameters are sampled from a distribution, it is desirable to evaluate a given controller on multiple independent trials. It was observed that the average score over 5 trials gave a reliable estimate.

TABLE 2 | Parameters and their ranges included in the *CMA-ES* optimization for a walking gait.

Parameter	Symbol	Range	Unit
Front amplitude	μ_f	[45, 140]	degrees
Hind amplitude	μ_h	[45, 140]	degrees
Front duty cycle	d_f	[0.15, 0.85]	*NA*
Hind duty cycle	d_h	[0.15, 0.85]	*NA*
Front offset	o_f	[-45, 10]	degrees
Hind offset	o_h	[-45, -10]	degrees
Front right phase offset	po_{fr}	[165, 195]	degrees
Hind left phase offset	po_{hl}	[255, 285]	degrees
Hind right phase offset	po_{hr}	[75, 105]	degrees

2.4. Evaluation Methods

In all experiments performance and stability are measured. Stability is measured as the fraction of all trials in which the model or robot has fallen. In simulation, performance is measured as distance between the original and final position of the model after a short time period (10 s unless mentioned otherwise). For the physical robot, the robot is tracked with a Kinect camera and performance is measured as distance traveled by the robot after a short time period (10 s unless mentioned otherwise).

3. RESULTS

3.1. Calibration

The aim of the calibration is to tune the robot model to achieve a sensor response to an actuation signal that is similar to that of the physical robot. **Figure 5** shows the response of the model

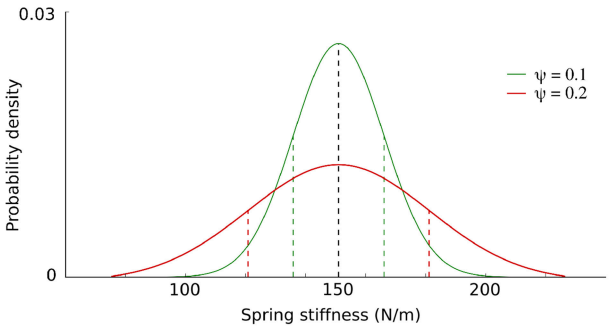


FIGURE 4 | Randomization level ψ affects the sampling distribution of the body parameters. ψ determines the standard deviation (σ, colored dashed lines) of the Gaussian distribution with mean μ (black dashed line). In this example for parameter θ_k, $\sigma = \psi * \mu$, with $\mu = 151N/m$ being the spring stiffness of the calibrated model front legs.

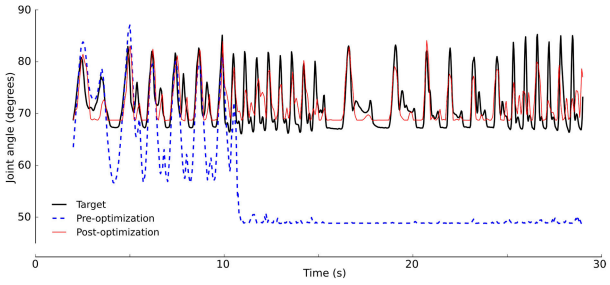


FIGURE 5 | Model calibration. The model was optimized to match the robot sensor response ("target", black). Sensor values after calibration (red, RMSE= 0.245) match better than before calibration (blue, RMSE= 0.741). Signals shown are for the hind right leg.

pre- and post-calibration. The calibration resulted in a model that approximates the dynamics of the physical robot. In line with the hypothesis of body randomization, we do not deem it beneficial to fine tune calibration to the greatest extent possible. Even with an optimally calibrated model, the simulation-reality gap may remain. Rather, we try to bridge the gap by searching for a gait that works on a variety of body morphologies. The calibrated model serves as a default morphology, on which variations are applied.

3.2. Gait Optimization in Simulation

To evaluate the effect of body randomization on the simulation-reality gap, gait optimizations with different levels of randomization were performed (parameter ψ ranging from 0 to 0.4). A higher level of randomization means that the body parameters were sampled from a broader distribution. Since the *CMA-ES* optimization does not guarantee an optimal convergence, experiments were repeated 5 times.

For each optimization, the solution was chosen as best performing individual of the final generation. Subsequently, the performance of each solution was tested in simulation. The controller, trained with a specific level of ψ, was tested on varying degrees of randomization (ψ_{test}). For each level of ψ, the procedure was repeated 5 times, **Figure 6**, Left presents the average performance. Performance of solutions trained on the nominal body (without body randomization, $\psi = 0$) is

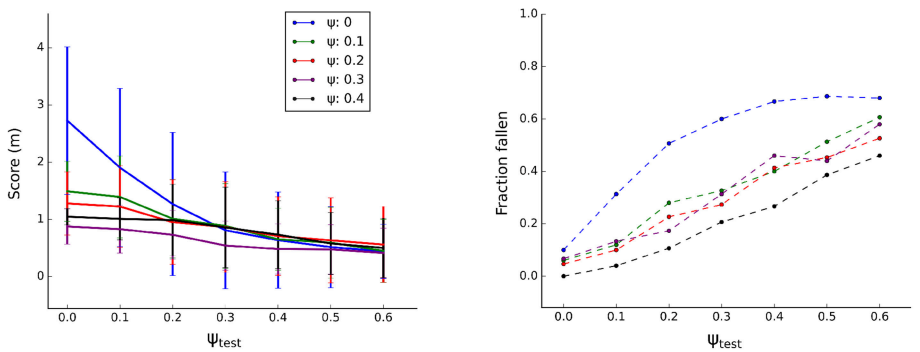


FIGURE 6 | Gait evaluation in simulation. For each level of ψ, 5 optimizations were performed resulting in 5 controllers. Each controller was tested in 30 trials. **Left:** average distance to origin (in 20 s, $N = 150$). **Right:** observed frequency of falling over, normalized.

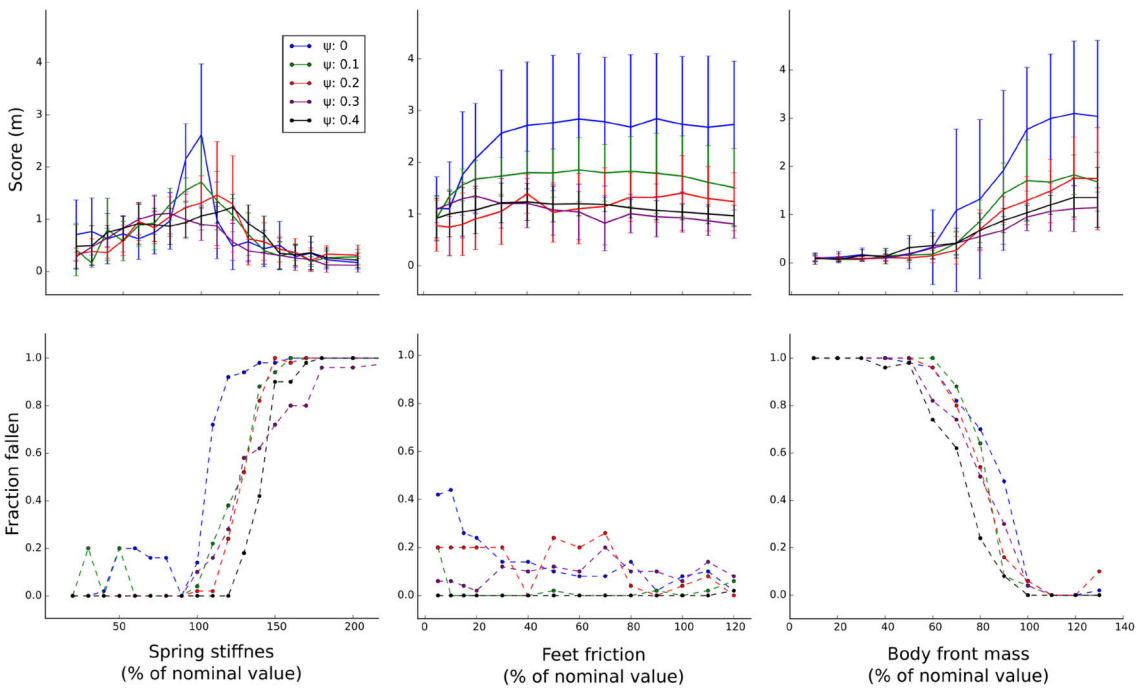


FIGURE 7 | Evaluation of parameters. Gaits are evaluated while incrementally varying a single parameter at a time, other parameters are kept at the default value. **Top:** average performance measured as distance from origin after 20 s ($N = 50$). **Bottom:** observed frequency of falling over, normalized.

higher if tested on bodies with no or limited randomization ($\psi_{test} < 0.3$) and converges with other solutions in the higher randomization regimes ($\psi_{test} \geq 0.3$). The variance of these solutions however is higher, reflecting the performance variation both between solutions and between trials of the same solution. Solutions trained with randomization ($\psi \geq 0.1$) have a lower score when tested without randomization ($\psi_{test} = 0$), because they have developed more prudent locomotion during training as the randomization prevents overfitting of the controller to the dynamics of the simulation environment and model.

Figure 6, Right plots the robustness metric: frequency of falling over. As expected, the fraction of trials resulting in a robot fallen over increases with increasing body randomization (ψ_{test}). More importantly, the amount of randomization during training improves stability of the resulting solution. The gaits trained without randomization ($\psi = 0$) are particularly susceptible to losing balance when tested on body configurations that it is not trained on. Overall, it seems there is a trade-off between speed and stability of a given solution. Randomization impacts this trade-off and favors more prudent gaits that are slower but more stable.

To evaluate the impact of variation of the different body parameters, the gaits were also tested while incrementally varying a single body parameter at the time and keeping other parameters at their default value (**Figure 7**). Similar to the previous test, training with body randomization lowers average performance but also the variance when changing the feet friction and mass distribution parameters. Varying the spring stiffness parameter has a more dramatic effect on the performance and here body randomization seems to improve performance in certain parameter ranges. Generally, the negative impact of varying body parameters on stability is reduced by increasing the amount of training randomization (**Figure 7**, Bottom).

3.3. Transfer to Real World

The final solution of each optimization was tested on the physical robot. Performance is measured as distance traveled in 10 s (**Figure 8**, Top). Generally, adding body randomization ($\psi > 0$) improves average performance and decreases the variability in performance. Forty percent (2/5) of optimizations without randomization ($\psi = 0$) resulted in a functional gait compared to 80% (16/20) of optimizations with randomization ($\psi > 0$). Non-functional gaits result in the robot shuffling in place or consistently falling within 10 s. While a randomization level $\psi > 0$ seems beneficial, the precise level of randomization doesn't seem critical. This could be a consequence of sampling the parameters from a Gaussian distribution around a common mean. The optimization procedure was repeated with a very high randomization level ($\psi = 2$, not shown), which resulted in nonfunctional gaits. Presumably, the gaits learned without randomization ($\psi = 0$) are overfit to the training environment and hence perform well on the nominal body in the simulation, but suffer a performance drop when tested in another setting such as on the physical robot.

Additionally, frequency of falling was recorded (**Figure 8**, Bottom). Lack of body randomization resulted in a higher probability of the robot falling to its side or back. Optimizations with body randomization generally resulted in reduced frequency of falling, using $\psi = 0.3$ resulted in functional gaits that maintained balance in all trials.

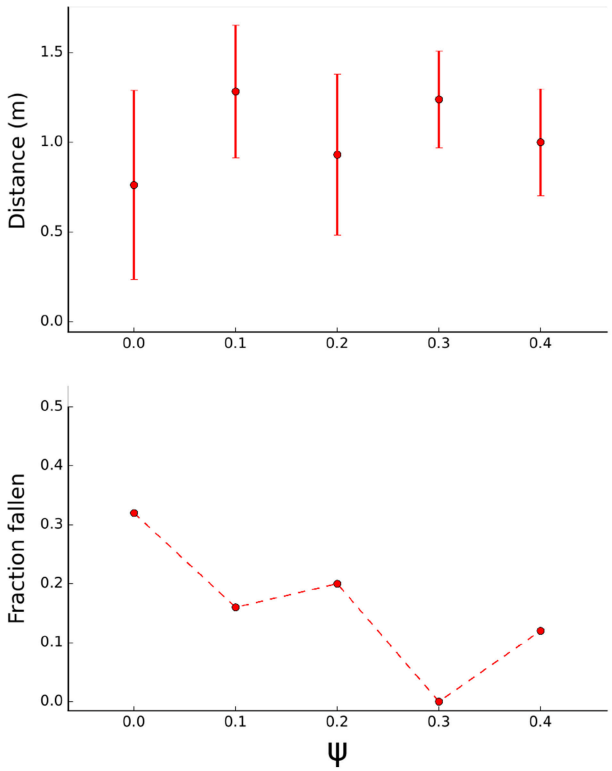


FIGURE 8 | Transfer test on physical robot. **Top:** Average traveled distance of the robot in 10 s trials. Each point represents an average of $N = 25$ trials (each result of the optimizations was tested 5 times on the robot), error bars indicate standard deviation. **Bottom:** fraction of trials where the robot flipped to its side or back.

4. CONCLUSION

In this work, we investigated bridging the simulation-reality gap for a compliant, underactuated robot, by treating a robot and its model as variations of the same dynamical system. Consequently, both the calibration and control optimization procedure focus on body parameters critical for the behavior of the dynamical system.

For the optimization procedure, we showed that body randomization results in improved transferability of the controllers. Lack of randomization results in better performance in simulation but worse performance on the real robot, compared to the optimization with randomization. Addition of randomization also improved stability of controllers, both in simulation and on the physical robot. Body randomization can be interpreted as a regularization method, preventing the optimization procedure from overfitting to the particular simulation environment. While body randomization improves sim-to-real transfer, the precise amount of randomization did not seem critical. For our platform, the use of body randomization

enhances the quality of controllers learned in simulation. The resulting controller has an improved stability, reducing risk of physical harm and providing a safe starting point to continue learning on the physical platform. This method is relatively straightforward to implement and could be used in combination with other tools that reduce the simulation-reality gap, such as domain randomization and data augmentation.

From the evaluations of gaits in simulation, it is clear that the quality of a given gait can be very sensitive to even small changes in physical properties such as the stiffness of springs in the leg. It would therefor be interesting to use a platform with adaptive spring stiffness in future work. This would allow to tune the compliance in function of gait optimization.

DATA AVAILABILITY

The datasets generated for this study are available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

The experiments were conceived by AV, GU, Fw, and JD and designed by AV. The physical platform was co-developed by GU, the virtual platform by HM. The data were analyzed by AV with help of Fw and JD. The manuscript was written by AV, with comments and corrections from GU, HM, Fw, and JD.

ACKNOWLEDGMENTS

This research has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2). This research was supported by the HBP Neurorobotics Platform funded from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2).

REFERENCES

Barasuol, V., B. J. S. C. F. M. d. P. E. C. D. (2013). "A reactive controller framework for quadrupedal locomotion on challenging terrain," In *IEEE International Conference on Robotics and Automation* (Karlsruhe), 2554–2561.

Bousmalis, K., Irpan, A., Wohlhart, P., Bai, Y., Kelcey, M., Kalakrishnan, M., et al. (2018). "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE), 4243–4250.

Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., and Krishnan, D. (2017). "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*(Honolulu), Vol. 1, 7.

Degrave, J., Burm, M., Kindermans, P., Dambre, J., and Wyffels, F. (2015). Transfer learning of gaits on a quadrupedal robot. *Adaptive Behav.* 23, 69–82. doi: 10.1177/1059712314563620

Drumwright, E., Hsu, J., Koenig, N., and Shell, D. (2010). "Extending open dynamics engine for robotics simulation," in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots* (Darmstadt: Springer), 38–50.

Falotico, E., Vannucci, L., Ambrosano, A., Albanese, U., Ulbrich, S., Vasquez Tieck, J. C., et al. (2017). Connecting artificial brains to robots in a comprehensive simulation framework: the neurorobotics platform. *Front. Neurorobt.* 11:2. doi: 10.3389/fnbot.2017.00002

Füchslin, R. M., Dzyakanchuk, A., Flumini, D., Hauser, H., Hunt, K. J., Luchsinger, R., et al. (2013). Morphological computation and morphological control: steps toward a formal theory and applications. *Artif. Life* 19, 9–34. doi: 10.1162/ARTL_a_00079

Gay, S., Santos-Victor, J., and Ijspeert, A. (2013). "Learning robot gait stability using neural networks as sensory feedback function for central pattern generators," in *EEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Tokyo), EPFL-CONF-187784.

Gupta, A., Devin, C., Liu, Y., Abbeel, P., and Levine, S. (2017). Learning invariant feature spaces to transfer skills with reinforcement learning. *arXiv* [preprint] arXiv:1703.02949.

Hamer, M., Waibel, M., and D'Andrea, R. (2013). "Knowledge transfer for high-performance quadrocopter maneuvers," in *Intelligent Robots and Systems (IROS)* (Tokyo), *2013 IEEE/RSJ International Conference on* (IEEE), 1714–1719.

Hansen, N. (2006). "The cma evolution strategy: a comparing review," in *Towards a New Evolutionary Computation* (Berlin: Springer), 75–102.

Heess, N., Dhruva T. B., Sriram, S., Lemmon, J., Merel, J., Wayne, G., et al. (2017). Emergence of locomotion behaviours in rich environments. *CoRR*, abs/1707.02286.

Jakobi, N. (1998). *Minimal Simulations for Evolutionary Robotics*. Ph.D. thesis, University of Sussex.

Laschi, C., and Cianchetti, M. (2014). Soft robotics: new perspectives for robot bodyware and control. *Front. Bioeng. Biotechnol.* 2:3. doi: 10.3389/fbioe.2014.00003

Lipson, H., and Pollack, J. (2000). Automatic design and manufacture of robotic lifeforms. *Nature* 406, 974.

Mordatch, I., Lowrey, K., and Todorov, E. (2015). "Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on* (IEEE), 5307–5314.

Park, H., Wensing, P. M., and Kim, S. (2017). High-speed bounding with the MIT cheetah 2: Control design and experiments. *Int. J. Robot. Res.* 36, 167–192. doi: 10.1177/0278364917694244

Peng, X., Berseth, G., Yin, K., and van de Panne, M. (2017). Deeploco: dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans. Graph.* 36, 1–41. doi: 10.1145/3072959.3073602

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. (2018). "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE) (Brisbane, QLD), 1–8.

Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science* 318, 1088–1093. doi: 10.1126/science.1145803

Pfeifer, R., Lungarella, M., and Iida, F. (2012). The challenges ahead for bio-inspired'soft'robotics. *Commun. ACM* 55, 76–87. doi: 10.1145/2366316.2366335

Raibert, M., Blankespoor, K., Nelson, G., and Playter, R. (2008). Bigdog, the rough-terrain quadruped robot. *IFAC Proc. Vol.* 41, 10822–10825. doi: 10.3182/20080706-5-KR-1001.01833

Taigman, Y., Polyak, A., and Wolf, L. (2016). Unsupervised cross-domain image generation. *arXiv* [preprint] arXiv:1611.02200.

Tan, J., Zhang, T., Coumans, E., Iscen, A., Bai, Y., Hafner, D., et al. (2018). Sim-to-real: Learning agile locomotion for quadruped robots. *CoRR*, abs/1804.10332.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). "Domain randomization for transferring deep neural networks from simulation to the real world," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on* (IEEE) (Vancouver, BC), 23–30.

Um, T. T., Park, M. S., and Park, J.-M. (2014). "Independent joint learning: a novel task-to-task transfer learning scheme for robot models," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE), 5679–5684.

Urbain, G., Vandesompele, A., Wyffels, F., and Dambre, J. (2018). "Calibration method to improve transfer from simulation to quadruped robots," in *International Conference on Simulation of Adaptive Behavior* (Springer), 102–113.

Willems, B., Degrave, J., Dambre, J., and wyffels, F. (2017). "Quadruped robots benefit from compliant leg designs," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017)* (IEEE) (Vancouver, BC).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 31 May 2019
doi: 10.3389/fnbot.2019.00022

Fast and Flexible Multi-Step Cloth Manipulation Planning Using an Encode-Manipulate-Decode Network (EM∗D Net)

Solvi Arnold and Kimitoshi Yamazaki*

Department of Mechanical Systems Engineering, Shinshu University, Nagano, Japan

We propose a deep neural network architecture, the Encode-Manipulate-Decode (EM*D) net, for rapid manipulation planning on deformable objects. We demonstrate its effectiveness on simulated cloth. The net consists of 3D convolutional encoder and decoder modules that map cloth states to and from latent space, with a "manipulation module" in between that learns a forward model of the cloth's dynamics w.r.t. the manipulation repertoire, in latent space. The manipulation module's architecture is specialized for its role as a forward model, iteratively modifying a state representation by means of residual connections and repeated input at every layer. We train the network to predict the post-manipulation cloth state from a pre-manipulation cloth state and a manipulation input. By training the network end-to-end, we force the encoder and decoder modules to learn a latent state representation that facilitates modification by the manipulation module. We show that the network can achieve good generalization from a training dataset of 6,000 manipulation examples. Comparative experiments without the architectural specializations of the manipulation module show reduced performance, confirming the benefits of our architecture. Manipulation plans are generated by performing error back-propagation w.r.t. the manipulation inputs. Recurrent use of the manipulation network during planning allows for generation of multi-step plans. We show results for plans of up to three manipulations, demonstrating generally good approximation of the goal state. Plan generation takes <2.5 s for a three-step plan and is found to be robust to cloth self-occlusion, supporting the approach' viability for practical application.

Keywords: planning, neural networks, forward models, deformable objects, manipulation, machine learning

OPEN ACCESS

Edited by:
Tomohiro Shibata,
Kyushu Institute of Technology, Japan

Reviewed by:
Nishanth Koganti,
Nara Institute of Science and Technology (NAIST), Japan
Jun Takamatsu,
Nara Institute of Science and Technology (NAIST), Japan

****Correspondence:***
Solvi Arnold
s_arnold@shinshu-u.ac.jp

Received: *05 October 2018*
Accepted: *23 April 2019*
Published: *31 May 2019*

Citation:
Arnold S and Yamazaki K (2019) Fast and Flexible Multi-Step Cloth Manipulation Planning Using an Encode-Manipulate-Decode Network (EM∗D Net). Front. Neurorobot. 13:22. doi: 10.3389/fnbot.2019.00022

INTRODUCTION

Within the area of robotic manipulation planning, deformable objects pose a particularly tough challenge. Manipulation changes the shape of such objects, so the common strategy of acquiring a 3D model of the object and planning w.r.t. this model is of little use. Even just predicting the object shape that will result from a given manipulation is far from trivial. Maybe the most common type of deformable object manipulation is cloth manipulation. Cloth exemplifies the difficulties stated above, yet humans manipulate cloth routinely and with ease, without much thought. We seem to acquire an intuitive sense of how cloth reacts to our manipulations. Replicating this ability in AI is a challenge of both theoretical and practical interest.

One point of particular theoretical interest is that our affinity with cloth goes beyond fixed goal-based routines, yet does not fit rule-based reasoning patterns. Despite being notoriously hard to formalize (or even verbalize), our affinity with cloth generalizes well to novel situations. This puts cloth manipulation into somewhat of a gray zone, that present day AI does not yet have a clear solution for.

Practical interest in cloth manipulation derives from the fact that cloth is ubiquitous in our everyday lives. Many everyday chores involve cloth manipulation in one form or another, so efficient cloth manipulation capabilities would be an important feature in household support robots.

Related Work in Cloth Manipulation

Much work in multi-step cloth manipulation avoids the need to plan by assuming fixed, hand-designed manipulation procedures. Such procedures can be quite effective for specific tasks (Maitin-Shepard et al., 2010; Koishihara et al., 2017; Yuba et al., 2017). Assuming a circumscribed starting situation and fixed outcome allows comparatively quick operation with limited computational cost, making this type of approach feasible for real-world applications where the same tasks have to be performed over and over, such as in industrial settings. However, there is no flexibility to accommodate new goals; every new goal state requires a new, human-provided, plan.

To flexibly realize variable goals requires an ability to plan ahead, to foresee the outcomes of individual actions and string actions together accordingly. This naturally leads to simulation-based approaches. While simulation provides high flexibility in terms of the manipulations that can be considered (Kita et al., 2014; Li et al., 2015), application in planning faces at least two major hurdles. The first is computational cost. Simulating a single manipulation is computationally expensive, and planning a sequence of manipulations generally requires consideration of a substantial number of possibilities. This in turn makes explicit search for manipulation sequences slow and impractical. The second hurdle is that obtaining an accurate deformation model of a given object is a difficult problem in itself, an issue that gets more pressing as more complex manipulations and longer sequences of manipulations are considered.

A promising intermediary approach works by retrieving and modifying manipulations from a database (Lee et al., 2015). This offers more operational flexibility than fixed procedures at a smaller computational cost than the simulation-based approach. However, present demonstrations of this approach still assume a fixed goal, and whereas the cost of database retrieval and deformation operations is less than full-fledged simulation, it is not clear whether this approach can be made efficient enough to perform free-form planning in real-time.

Recent years have seen increasing interest in the use of neural networks for manipulation problems. Impressive results have been demonstrated in grasp point detection for rigid objects (Lenz et al., 2015), and visuomotor policy learning (Levine et al., 2016). Given neural networks' natural affinity for fuzzy subject matter, they may have the potential to bring major progress to deformable object manipulation. Neural network-driven grasp point detection has been applied in a bed-making task (Seita et al., 2018). Interesting results have been reported on a dual neural network approach to cloth folding, combining a convolutional autoencoder and a time-delay neural network to achieve fine control over manipulation motions (Yang et al., 2017). Whereas the goal is fixed, motion is guided by network-generated predictions of the very near future, thus realizing some degree of foresight. Neural network-driven prediction has also been employed for prediction of forces exerted on human subjects in a dressing task (Erickson et al., 2018).

Despite these advances, open-goal, multi-step manipulation planning for deformable objects remains largely unexplored territory.

Related Work in Model-Based Learning

There is increasing evidence from neuroscience that humans learn, in part, by acquiring forward models (Gläscher et al., 2010; Liljeholm et al., 2013; Lee et al., 2014). Human ability to generalize implicit knowledge of cloth dynamics to novel circumstances suggests that we acquire forward models of these dynamics. Forward models are commonly used in model-based control and planning, but in the case of cloth manipulation planning, the use of explicit forward models (i.e., physical simulation) is problematic due to computational cost and the difficulty of obtaining an accurate model, as discussed above. However, it has been demonstrated that neural networks can be trained as forward models. Of particular relevance here is (Wahlström et al., 2015) for the use of a neural network trained as a forward model in latent space. The proposed model takes high-dimensional observations (images) of a low-dimensional control task as inputs, maps these observations into low-dimensional latent representations (by means of PCA followed by an encoder network), feeds these through a network functioning as a forward model, and then maps the outputs of this network to high-dimensional predictions of future states. This model is then used to search for control signals that bring about a fixed goal.

Also related is (Watter et al., 2015). Here too, an encoder network is used to map high-dimensional observations to low-dimensional latent representations. The forward model takes the form of linear transformations in latent space (although a non-linear variant is considered as well). We return to these and other related neural network studies in the discussion section. In the context of cloth manipulation, use of a neural network as forward model allows us to side-step the computational cost of explicit simulation (replacing it with forward propagation through the network), as well as the burden of acquiring an accurate model of a given cloth item (instead, the forward model is learned from data).

Contributions and Limitations

This paper presents a fully connectionist approach for efficient deformable object manipulation planning, based on forward modeling of the object's deformation dynamics with respect to a given manipulation repertoire. We avoid explicit simulation and database matching/retrieval, yet realize a substantial degree of flexibility along with fast operation time. Core of the system is a modular neural network architecture, composed of 3D convolutional encoder and decoder modules and a

fully connected manipulation module. Given a start and goal state, the network is used to search for the manipulation sequence that produces the latter from the former, by means of error back-propagation w.r.t. the manipulation input. We can search for manipulation sequences of various lengths by varying the number of recurrent propagation loops through the manipulation module. In the present paper, we apply this manipulation planning approach to free-form manipulation on a (simulated) square cloth.

The main contributions of this work are as follows.

- We propose a neural network architecture for associating manipulations with changes in cloth states, trainable on individual manipulation examples from a comparatively small dataset.
- We show that this network can predict cloth manipulation outcomes.
- We show that this network can be used to generate single and multi-step manipulation plans in seconds, by means of back-propagation w.r.t. the manipulation inputs.

An important distinction between our system and most existing work in cloth manipulation is that our system is, to a large degree, task-agnostic. The task domain and manipulation repertoire are determined by the dataset the system is trained on. We believe this should provide a high degree of flexibility for application to various task domains and manipulation repertoires.

This has consequences for system evaluation as well. The free-form manipulation task we adopt here for evaluation is not intended to represent or resemble any particular practical cloth manipulation task, nor was the manipulation repertoire designed with any specific robotic platform in mind. Instead, our experimental setup is designed to assess the system's capabilities on a broad domain with a basic manipulation repertoire that could be realized on a wide variety of robotic platforms. The motivation for this choice is two-fold.

The first motivation is that success on a broad task would suggest that the approach is viable for a broad variety of more specific tasks. For use in a practical, constrained application, one would want to use a dataset that covers a domain suited to the application, with a manipulation repertoire suited to the specific robotic platform under consideration.

The second motivation is the long-term goal of pursuing a generalized affinity with cloth objects. Human affinity with cloth goes far beyond folding towels and clothes into neat rectangles. We quickly drape a dish towel over the back of a chair when a goal more urgent than drying the dishes presents itself. We extract sizable bed sheets from a washing machine without them sweeping over the floor with little effort. We intuit what will or will not fit into a coat pocket. Much of human cloth manipulation seems better characterized as the flexible application of a general understanding of cloth dynamics than as mastery of a large collection of individual micro-tasks. Progress toward broad generalized cloth manipulation abilities for robots requires that we try and push toward methods that offer increasingly high degrees of generality. In the context of this goal, the value of our results lies not in their practical applicability, but in the fact that they represent progress toward higher generality.

As will be clear from the above, our purpose in this work is not to excel at any one specific example of cloth manipulation, and practical applicability of the system as trained on our dataset is limited at best. Also, whereas we believe that our approach should be viable for a range of task settings more specific than ours, we expect its applicability to highly constrained tasks to be limited: planning ability is only meaningful on tasks that present significant variability and require some level of system autonomy.

TASK DESIGN

The problem of manipulation planning can be formulated as follows: Given a state domain consisting of possible state set S, a manipulation (action) domain consisting of possible manipulation set M, and states s_a and $s_b \in S$, find a manipulation sequence (action sequence) $p_{ab} = <m_0, \ldots, m_{n-1}>$ with $m_i \in M$ such that applying manipulation sequence p_{ab} starting in state s_a will produce state s_b.

In this paper we consider the task of manipulating a square piece of cloth from one configuration into another. We let states represent the cloth in some stable configuration. Manipulations are defined as triplets of real-valued 2D vectors. The first two vectors indicate the x and y coordinates where the cloth is picked up (grasp points below), and the third vector (displacement vector below) indicates the horizontal movement of the grasp points (both points are moved in parallel and by the same distance, so one vector suffices). **Figure 1** illustrates how such manipulations are translated into actuator trajectories. Plans are sequences of manipulations. The height to which the grasp points are lifted is fixed.

The manipulation format is intentionally somewhat minimalistic. Every additional dimension also adds to the complexity of the planning process, so any aspects that can be resolved locally are best excluded from the planning process. In simulation, we perform manipulations by fixing the relevant vertices of the cloth mesh to a non-colliding actuator. The complexities of performing cloth manipulations with any given physical actuator are not considered in the present paper. In ongoing work, we are integrating the planning system described here with a physical dual-armed robot platform (Tanaka et al., 2018).

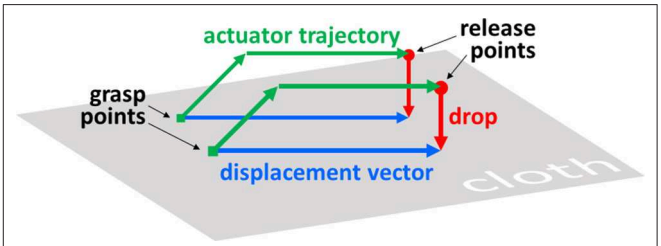


FIGURE 1 | Schematic of cloth manipulation.

SYSTEM ARCHITECTURE

The system is composed of three modules: encoder module E, manipulation module M, and decoder module D. E maps (encodes) state s_i to its latent representation c_i:

$$E(s_i) = c_i. \tag{1}$$

M maps latent representation c_i and manipulation m_i to a prediction $\hat{c}_{i+1}$ of c_{i+1}, the latent representation of the state s_{i+1} that results from applying manipulation m_i to s_i:

$$M(c_i, m_i) = \hat{c}_{i+1}. \tag{2}$$

D maps (decodes) latent representation c_i (or $\hat{c}_i$) to an approximation $\hat{s}_i$ of state s_i:

$$D(c_i) = \hat{s}_i. \tag{3}$$

Given a state s_i and a manipulation m_i, $D(M(E(s_i),m_i))$ computes a prediction of the outcome s_{i+1}. Mapping to latent space before applying M, and mapping back to regular space afterwards, serves two purposes. The first is dimensionality reduction. States in our task are 16384D. Applying manipulations directly on states of this dimensionality is computationally costly and hard (if not impossible) to train. Modules E and D map states to more manageable 512D latent representations. The second reason is manipulability. Depending on how a state is represented, it may be easier or harder to apply specific manipulations to it. By training E, M, and D in compound fashion, the E and D modules are forced to learn a latent representation format that makes M's life easy, i.e., lends itself well to application of the manipulation repertoire.

Movement of one point of a cloth often affects the cloth's shape over a broad region in non-trivial but highly structured ways. For predicting these effects, the substrate of a voxel representation is likely far from ideal. We let E map voxel representations to latent representations with no imposed spatial structure, so the learning process is free to find a way of representing the cloth's spatial contingencies that facilitates prediction of manipulation outcomes.

Each module is instantiated as one neural network. Encoder E and decoder D are structured like the bottom and top halves of a 3D convolutional autoencoder. Manipulation module M is a modified multi-layer perceptron. A concept image of the network is shown in **Figure 2**. The network is implemented in TensorFlow (Abadi et al., 2015). Network specifications are given in **Table 1**.

We avoid pooling, because it discards important spatial information. The use of pooling between convolution layers is usually motivated by the partial translation invariance and dimensionality-reduction it affords, but in the present system the former is detrimental and the second can as well be obtained with strided convolution, which does not destroy spatial information. As can be inferred from the strides and map counts given above, E maps $32 \times 32 \times 16 \times 1$ inputs to $1 \times 1 \times 1 \times 512$ outputs (here the first three dimensions are spatial, the fourth is the channel dimension), and D does the inverse. The latent representations only have extension in the channel dimension, meaning they have no imposed spatial structure.

Cloth state input is given in the form of a binary voxel rasterisation of the (simulation-generated) cloth mesh, at the $32 \times 32 \times 16(\times 1)$ resolution taken by E. Each voxel takes a value of 1 if one or more vertices of the cloth mesh fall in that voxel,

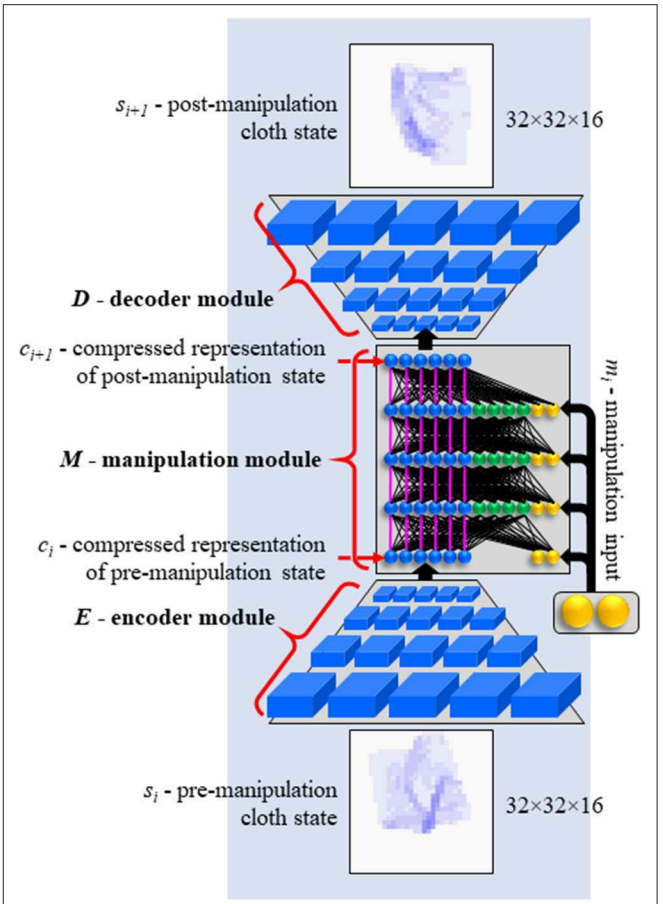


FIGURE 2 | Concept image of EMD network. Functionally denoted as $D(M(E(s_i),m_i))$. See **Table 1** for actual sizes.

TABLE 1 | Network architecture specifications.

3D CONVOLUTIONAL ENCODER (E)/UP-CONVOLUTIONAL DECODER (D)	
Layers	6
Feature maps / layer	1 (input), 32, 32, 64, 128, 256, 512 (output). Order reversed in decoder.
Kernel size	3×3×3 (all layers)
Strides	2×2×1 on the first layer (E) / last layer (D) 2×2×2 on all other layers
Activation function	tanh
MANIPULATION MODULE (M)	
Layers	10 (5 in configuration C2)
Input layer size	512+6
Hidden layer size	512+512+6 (512+512 in configuration C0)
Output layer size	512
Activation function	tanh

and a value of 0 otherwise. Before rasterisation, we multiply the vertices' z coordinates by a factor 4 to emphasize height variations (effectively increasing resolution on the z-axis by a factor 4). This is important to ensure that creases in the cloth (which do not have much height but do provide important shape information) do not get lost in rasterisation.

We scale the view-port of the voxel space so that the cloth, shape-wise, fully fits inside it in any plausible stable shape configuration. However, repeated manipulations can move the cloth by a substantial distance, which would quickly take it out of the range of the viewport. To keep the cloth always fully in view, we introduce periodic boundary conditions on the x and y axes. That is, we bring vertices' x and y coordinates into the [-1, 1] range using $x' = (x + 1) \ mod \ 2 - 1$ (and same for y).

Both the autoencoder (encoder and decoder module) and the manipulation network have some uncommon features. The convolution operations use periodic padding on the x and y dimension to account for periodic boundary conditions on the voxel space. A kernel size of $3 \times 3 \times 3$ implies that for full-size convolution we should pad each map with a border of width 1 before applying the kernel. Instead of the usual zero-padding, we fill the border with the content of the opposite edges and corners of the map. **Figure 3** shows an example of a cloth extending over the edges of the voxel space, with periodic padding applied. Periodic padding is applied at every convolution and up-convolution throughout the encoder and decoder modules. All connections in the encoder and decoder are initialized with random values from the $[-0.05, 0.05]$ range.

The manipulation network is a multi-layer perceptron with a number of modifications. One modification is the introduction of an aspect of Residual Learning (He et al., 2015). As can be seen in **Figure 2**, the network is comprised of three vertical sections, colored in blue, green and yellow in the figure. The blue section of the input layer receives the encoded state representation. In each subsequent layer, the blue section receives a copy of the activation vector on the blue section of the preceding layer (i.e., the pink connections in **Figure 2** have fixed weights of 1.0). Activations computed in the layer are added to the copied values. The blue section essentially serves to hold the state representation as it is incrementally modified through the layers. Note that with all other weights set to 0, this architecture simply passes on the encoded state unchanged.

FIGURE 3 | Top-down view (voxel value means over z-axis) of a voxel representation of a state with the cloth wrapped around the periodic boundaries of the voxel space, with periodic padding.

This style of propagation, where setting all mutable weights to 0 results not in blank output but in pass-through behavior, was originally proposed to facilitate the training of very deep networks (He et al., 2015). The result is a learning style where subsequent layers learn to make incremental improvements to the representation as it propagates through the net. Our network is not particularly deep (the manipulation module has just 10 layers), and our implementation differs (the residual connections do not skip layers), but the concept of incremental modification is applicable to our problem setting. Pre- and post-manipulation cloth states often show some degree of resemblance. A short displacement distance often leaves part of the cloth undisturbed. Many movements displace the cloth in space but leave parts of its shape intact. Hence a computation style of incremental modification seems appropriate. In the experimental results below, we include a variant without these residual connections (Configuration C1) to assess the effect of their inclusion.

Neurons in the green section behave as in a regular neural network. They serve to compute the appropriate modifications and apply them to the state representation. This section has no residual connectivity. Finally, the yellow section receives the manipulation input. The manipulation input is small (6 values) compared to the state representation in the blue section and the activation vector in the green section (512 values each). We offset this imbalance in two ways. Firstly, we initialize the weights on connections from manipulation inputs to larger values (random values from the $[-0.05, 0.05]$ range) than the rest of the weights in the manipulation network (random values from the $[-0.001, 0.001]$ range). Secondly, manipulation inputs are provided (identically) at every layer. This avoids the need for the network to retain the manipulation signal through numerous layers before it can affect computation in the upper layers. We included a variant that feeds the manipulation input only into the first layer of the manipulation network (configuration C0), to assess this feature's effect on performance.

At the borders between any two modules (i.e., on the output of the E and M modules), we introduce a simple discretization layer.

$$activation_{out} = \frac{round(res \cdot activation_{in})}{res} \tag{4}$$

Here *res* is a system parameter controlling the grain of the discretization, which we set to 16. This layer is not differentiable, so during backpropagation we let the gradient pass through unmodified, cf. (Van den Oord et al., 2017). The use of discrete latent representations in generative models has recently been reported on in Jang et al. (2017), Maddison et al. (2017), where it is typically motivated by the latent variables' correspondence to categories. Our motivation to discretize representations is rather different (there is no concept of categories in our task). We will be using the manipulation module recurrently to compute multi-step plans, which can quickly incur a build-up of noise and diffusion in the latent representations. If we let the latent representation be discrete, it can be denoised

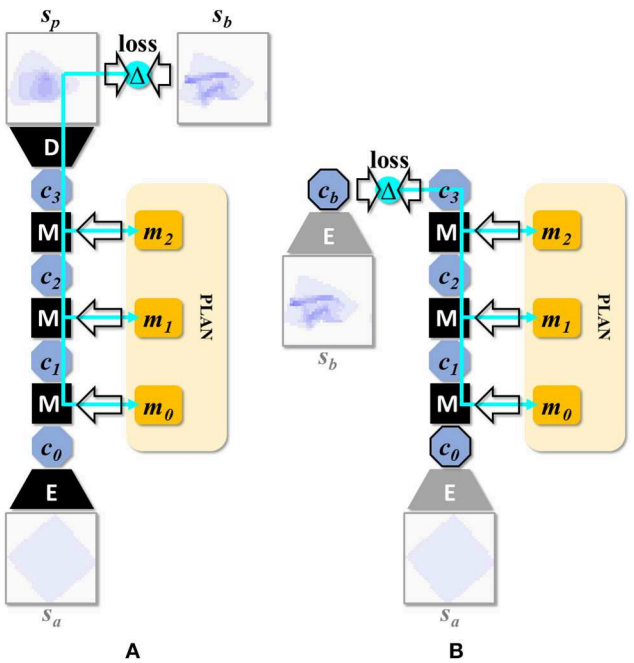


FIGURE 4 | (A) Network setup for generating a 3-step plan. This composition can be denoted as $D(M(M(M(E(s_0),m_0),m_1),m_2))$, or EM^3D for short. Multi-step manipulation planning is done by back-propagation through such recurrent applications of the manipulation network. The cyan arrows represent the back-propagating error signal. **(B)** Network setup after speed optimization. By precomputing latent encodings of s_a and s_b, plan search can be run entirely in latent space, avoiding repeated (and comparatively expensive) propagations through the encoder and decoder modules.

by means of rediscretisation. Specifically, if the magnitude of the noise on a given representation falls below 0.5/*res*, then equation (4) will return the representation perfectly denoised. Theoretically, as long as the error incurred in a single pass through the manipulation network falls below this threshold, predictions would not lose accuracy as the number of passes through the network increases. Conversely, without discretization, any error larger than zero would carry through to the next pass, leading to degradation of prediction quality as the number of passes increases. Hence *in theory*, the discretization layer may improve multi-step prediction and planning ability. Of course, setting *res* too low will harm the expressiveness of the latent representation, as it reduces the number of possible latent representations. We included a non-discretizing variant in our experiments (configuration C3), to assess the effect on performance.

PLANNING ALGORITHM

The network as discussed so far computes predictions of manipulation outcomes, but its actual purpose here is plan generation. Here we discuss how the net is used to generate multi-step manipulation plans, and how planning and manipulation execution are interleaved in operation.

Algorithm 1 Plan generation.

```
n = plan length
size_c = size of latent representation
n_parallel_instances = 10
max_iterations = 100
s_a = current cloth state
s_b = target cloth state
m_ab = n×6 array initialized with
    a random n-step plan
c_0 = E(s_a)
c_b = E(s_b)
r = n×6 array of update rates,
    initialized to 0.1
instance_best=[1.0]*n_parallel_instances[1]
steps_stagnant = [0] * n_parallel_instances
for i_iteration ∈ 0, ..., max_iterations:
  for i_instance ∈0, ..., n_parallel_instances:
      for step ∈ 1, ..., n:
        c_step = M(c_step-1, m_ab[step-1])
      loss = (C_b − C_n)^2 / size_c
      if loss < instance_best[i_instance]:
        instance_best[i_instance] = loss
        steps_stagnant[i_instance] = 0
      else:
        steps_stagnant[i_instance] += 1
      g = n×6 array of gradients for
        manipulation inputs w.r.t. loss
      update r according to iRprop-
      m_ab += r · g
    if minimum(steps_stagnant) ==
      early_stopping_criterion:
      break
return m_ab
```

[1]The (Python) syntax [v] * n initialises a list of length n with value v.

Plan Generation

By applying the manipulation network recurrently for n times, we can predict the outcome of an n-step plan. We refer to a net with n recurrent passes through the manipulation module as EM^nD, and to these nets in general as EM*D nets. By means of iterated backpropagation w.r.t. the manipulation inputs, these nets can be used to generate multi-step manipulation plans. **Figure 4A** illustrates the concept for $n = 3$ (the maximum considered in the present work). This generation process can be further optimized for speed by precomputing latent representations of the start and end state and running the iterative plan generation process in latent space entirely. **Figure 4B** illustrates this optimization. Note that the encoder is used twice while the decoder is not used. **Algorithm 1** specifies the procedure for generating a plan m_{ab} for transforming state s_a into state s_b.

Although the optimized variant has a substantial speed advantage, its viability was found to depend on the loss functions used for training, so for some experiments below we report scores for both variants. An earlier report on this work (Arnold and Yamazaki, 2017) also employed the non-optimized variant.

Manipulation input values are adjusted by means of the iRprop- variant (Igel and Hüsken, 2000) of the Rprop update scheme (Riedmiller and Braun, 1992). Rprop was proposed as an update rule for neural network training. The distinguishing feature of Rprop and its variants is that a separate learning rate η_i is kept for each variable v_i to be optimized. Typically, the variables are neural network connection weights, but in the present case the object of optimization is the manipulation input, so we keep one learning rate for each variable in m_{ab}. The learning rate is updated every iteration of the optimization process, on basis of the sign of the error gradient at its variable. When the sign of the gradient is unchanged w.r.t. the previous iteration, the learning rate is multiplied by η^+, and the variable is updated by -η times the sign of the gradient. When the sign of the gradient has flipped, different variants of the Rprop algorithm operate in subtly different ways. The iRprop- variant multiplies the learning rate by η^-, leaves the variable's value unchanged, and blocks change of the learning rate at the subsequent iteration. Learning rates are clipped to the range $[\Delta_{min}, \Delta_{max}]$. Rprop and its variants are robust against a broad range of initializations of the learning rates, and can quickly zoom in on solutions, even on error functions with small gradients, as only the sign of the gradient is used. A drawback is the need for individual learning rates for each variable, but in our use case the number of variables to be optimized is small ($6n$). We found values of 1.5 for $\eta+$ and 0.33 for η- to perform well in our setup. Learning rates are initialized to 0.1 and the learning rate bounds Δ_{min}, Δ_{max} were set to 10^{-10} and 0.1.

We set the number of search instances (*n_parallel_instances* in **Algorithm 1**) to 10, and let the instances run in parallel on GPU (i.e., the for loop at line 16 is parallelized). Each instance is started from a different random initialization. Search is cut short if all search instances are stagnant for a set number of iterations (*early_stopping_criterion*, set to 25 here). We observed that in practice, most runs run for the full number of iterations (*max_iterations*, set to 100 here), although improvement during the latter half of the search tends to be marginal. We adopt the plan with the lowest remaining loss value as the final result, and obtain its expected outcome state by forward propagation through EM^nD.

Closed-Loop Planning

Here we describe the procedure for assessing the system's planning performance, used to generate the results in the next section. We adopted a "closed-loop" procedure that interleaves planning and execution steps (**Algorithm 2**). The execution step here refers to performance of the first step of the generated manipulation plan. In our test setup, this means that we send the manipulation instruction to the simulator, which then executes the manipulation and returns the resulting cloth state. Interleaving planning and execution ensures that small errors do not build up over multiple manipulations, and affords some degree of correction when outcomes are not as expected. Alternatively, faster but less accurate performance can be achieved by "open-loop" operation: planning just once and performing the obtained sequence "blindly" (i.e., without observing and re-planning w.r.t. the intermediate results).

Algorithm 2 Closed-loop planning.

```
1 n = sequence length
2 s_a = initial state of the sequence
3 s_b = final state of the sequence
4 while n > θ do
5   <m_0, ..., m_n> = generate plan of length n
6   s_a = result of performing m0 on sa
7   n −= 1
```

DATA GENERATION

We generate states using the cloth simulation functionality of the Blender 3D editor (Blender, 2017) (version 2.77a). The cloth is represented by an 80 × 80 mesh with the cloth modifier enabled. The mesh measures 1.4 × 1.4 in Blender's spatial units, spanning from [−0.7, −0.7, 0.03] to [0.7, 0.7, 0.03] in its initial configuration. In the conversion between Blender data and neural network input, the neural network's input space corresponds to a viewport of size 2 × 2 × 0.25 in Blender units. The cloth mesh has no explicit thickness, but we let collision detection maintain some minimal distance between vertices, as well as between vertices and the virtual desk surface (a plane at $z = 0$), so that the cloth behaves as if it has some thickness.

Each sequence starts with the cloth laid out as a square on a flat surface (representing e.g., a table), with the axes of the cloth aligned with the x and y axes of the coordinate system. Then randomly generated manipulations are performed one by one, while we store the resulting cloth state to the dataset after each individual manipulation. Two-handed and one-handed manipulations are generated, with equal probability. Zero-handed examples (i.e., failures to manipulate) need not be generated at this stage; such examples can be generated on the fly during training from successful examples (we return to this point below). One issue that requires consideration when generating examples is the range from which to pick the manipulation coordinates. Covering the entire space the cloth can reach over 3 manipulations is inefficient, and will grow increasingly inefficient as we consider longer sequences. We constrain the range for manipulation coordinates by shifting the coordinate system along the displacement vector of each manipulation. This way the cloth always remains near the origin.

Random grasp points are found by randomly selecting cloth vertices, and values for the displacement vector are randomly picked from the [−1.4, 1.4] range. To manipulate the cloth, we pin the vertex or vertices selected as grasp point(s) to an invisible actuator object (an "empty" in Blender terminology), and assign the relevant movement trajectory to this actuator.

During manipulation, movement speed of the actuators is fixed to 0.02 Blender units per frame in horizontal and vertical direction (independently). The lifting height is set to 0.15 Blender units. Horizontal and vertical actuator movement starts simultaneously. Vertical movement is stopped once the lifting height is reached. A snapshot of a manipulation in progress is shown in **Figure 5**. Once the movement is completed, the actuator releases the cloth, and the simulation is left to run for 16 additional frames to allow the cloth to fall down and settle.

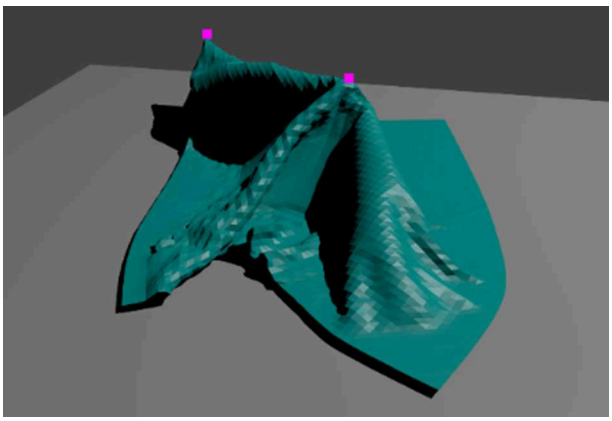


FIGURE 5 | Snapshot from a cloth manipulation in progress in simulation. The pink squares show the points where the cloth is pinned to the (invisible) actuators. Cloth mesh resolution is 80 × 80.

TABLE 2 | Cloth simulation parameter settings.

Material settings	
Mass	1.0
Structural stiffness	10.0
Bending stiffness	50.0
Damping	
Spring damping	50.0
Air damping	0.0
Velocity damping	1.0
Collision settings	
Collision distance	0.015
Self-collision distance	1.0

For interpretation of these values we refer to the Blender documentation (www.blender.org).

We generate 2,900 sequences of length 3, for a total of 8,700 manipulation examples. Sequences are stored as 7-tuples of the form $<s_0, m_0, s_1, m_1, s_2, m_2, s_3>$. We designate 2,000 sequences as training data, 600 as test data, and 300 as validation data. The simulation parameters defining the cloth behavior are given in **Table 2**. Settings not pertaining to cloth specifically were left at their default values.

Whereas all manipulation sequences in the dataset start from the initial, fully spread state, the prediction and planning capabilities of the system are not constrained to starting from this state. In our evaluation experiments below, we assess prediction and planning abilities starting from any non-final state within the sequences.

It should be noted that this data generation procedure limits the scope of the dataset to cloth states that are accessible from the initial spread out state within a few manipulations from the manipulation repertoire under consideration. As such the dataset does not represent a uniform sampling of the space of possible cloth configurations. Uniform sampling of this space is by itself far from trivial, and we have not pursued it here. Whereas by no means exhaustive, the dataset does present a broad variety of starting states. State variation is further enriched by means of data augmentation, as explained below.

Data Augmentation

Data augmentation is performed during training by applying random rotation (rotating all cloth mesh vertices around the origin by a random angle between 0 and 360), mirroring, and grasp point swapping. A grasp point swap changes which grasp point's coordinates go into which pair of coordinate input neurons. The order of the grasp points is immaterial, so the outcome state remains the same. The data is further augmented with failure-to-grasp examples. Training on failure-to-grasp examples is necessary, since many valid manipulation inputs have neither grasp point lying on the cloth. However, there is no need to explicitly generate such examples. In case of a failure to grasp, the cloth remains in the same state, so we can generate failure-examples simply by picking existing examples, replacing the grasp points with random points falling outside the cloth, and replacing the result states with copies of the initial states. We let every batch (16 examples) contain two such failure cases. In these examples we do not shift the coordinate system along with the displacement vector.

Rotational data augmentation in particular proved essential to make training work on the relatively small dataset used here. Without it, training quickly overfit on the training data and never achieved adequate performance on the test set.

The Role of Simulation

Although in this paper we use explicit simulation to generate data, this simulation is not an integral part of the system as it is in simulation-based planning. Simulation data is used here because it is easy to generate, but given a similar data set of real-world data the system could be trained and used without any explicit simulation. This fact that an accurate simulation model of the object is no requirement for this system is an important feature, as it is often difficult in practice to obtain accurate simulation models of cloth items and discrepancy between model and reality can substantially degrade performance of simulation-based systems. Note that our simulated cloth is not a stand-in for the real planning subject; the simulated cloth itself is the subject.

In consideration of the costs of real-world data generation, we kept the size of the simulation dataset modest, to ensure that the system can be trained effectively on realistically generatable amounts of real-world data. A system for automated data generation on robot hardware is currently under development (Tanaka et al., 2018).

TRAINING

We train the network on our Blender-generated dataset. Recall that the purpose of training here is not to teach the net to plan, but to let it acquire a forward model of the cloth's dynamics w.r.t. the manipulation repertoire. Hence whereas we generated sequences of length 3, the training process uses individual manipulation examples. Each 7-tuple $<s_0, m_0, s_1, m_1, s_2, m_2, s_3>$ provides 3 training examples of the form $<s_i, m_i, s_{i+1}>$. The net is trained on 1,250,000 batches of 16 such

manipulations each. Batches are composed randomly, but with some weighing of the manipulation steps. As all sequences start from the same default state, there is less cloth shape variation over the first-of-sequence manipulations in the dataset than over the third-of-sequence manipulations. The later cloth shapes in a manipulation sequences are also taller (i.e., have more vertices with higher z-coordinates) on average, as repeated manipulation often produces shapes in which the cloth is folded over itself. To counter-balance this bias in shape variation across steps we pick the first, second and third manipulation step with probabilities of 1/7, 2/7, and 4/7, respectively.

Loss Functions

We use two loss functions, which we denote as loss_s and loss_c. Loss_s is the mean squared error between network output, i.e., $D(M(E(s_0),m_0))$ and the (voxel representation of) the actual outcome, i.e., s_1.

$$loss_s = \frac{\sum (D(M(E(s_i),m_i)) - s_{i+1})^2}{size_s} \tag{5}$$

Here size_s is the size (in voxels) of the state representation. The second loss function, loss_c, is introduced to enforce consistency of state encoding format between the input and output layers of the manipulation module.

$$loss_c = \frac{\sum (M(E(s_i),m_i) - E(s_{i+1}))^2}{size_c} \tag{6}$$

Where size_c is the size of a latent representation (512 with our settings). Loss_c serves to enable multi-step planning. Multi-step planning involves recurrent use of the manipulation module. For recurrent application to make sense, the input and output of the manipulation module must be in the same encoding format, i.e., the latent representation of a given cloth state should not differ (much) depending on whether it is read at the input or output of the manipulation module. When the encoding format is inconsistent, looping the manipulation module's output back into its input will not produce a meaningful subsequent output. Hence, consistency of representation format between the manipulation network's input and output must be enforced explicitly. However, we do not want to impose any format in particular; finding a suitable encoding is up to the learning process.

To achieve this we compare two differently obtained encodings of s_1. The first is simply $\hat{c}_1$ as above, i.e., $M(E(s_0),m_0)$. The second is obtained by application of the encoder directly on s_1, i.e. $E(s_1)$. Loss_c quantifies encoding inconsistency as the mean squared error over these two encodings of s_1. Regardless of how the encoder module encodes states, this loss will be low if the manipulation module preserves the encoding format between its input and output. A functionally similar loss term was used in Watter et al. (2015).

There is overlap in function between the two loss terms. Minimizing loss_s trains the net to compress states into an easily manipulable format and to apply manipulations, whereas loss_c trains the net to keep the encoding consistent over the course of manipulation application and to apply manipulations in this encoding.

It proved difficult to balance the losses effectively. Loss_s and loss_c derive from different representation formats, and do not necessarily decrease in tandem over the training process. Balancing them with fixed weight parameters will often have one dominate the other. We resolved this issue as follows: instead of combining the losses into a compound loss function, we compute the gradients for both losses separately, and then combine the gradients on a per-weight basis by averaging over their signs. As only the signs of the gradients are used, the resulting update rule can be considered a variant of the Manhattan update rule. With combining gradients, the update rule takes the following form:

$$\Delta w_i = 0.5 \cdot \eta \cdot \left[sign\left(g_i^s\right) + sign\left(g_i^c\right) \right] \tag{7}$$

Where Δw_i is the change in weight for connection i, η is the learning rate, and g_i^s and g_i^c are the gradients for connection i w.r.t. loss_s and loss_c, respectively. Using this rule, weights are updated by η in the direction of the sign of the gradients when the gradients agree in sign. If the signs oppose, they cancel out, and the weight is not updated. This update rule is used on the weights of the manipulation module only. No gradients for the decoder module can be derived from loss_c, and whereas gradients for the encoder module can be derived they may actually be harmful: with respect to the encoder, loss_c would favor trivial encodings that map every state to the same representation, for this maximizes encoding consistency and makes manipulation application trivial. Restricting loss_c to the manipulation module bars this dead-end solution. For the encoder and decoder modules we use the Manhattan update rule on the gradients derived from loss_s only:

$$\Delta w_i = \eta \cdot sign\left(g_i^s\right) \tag{8}$$

Learning rate η is initialized to $5 \cdot 10^{-5}$, and reduced dynamically (see Learning rate adjustment and overfitting counter-measures).

Use of the Manhattan update rule is unusual. In general, it is by no means the fastest weight update rule. However, our network proved hard to train with the more common update rules. We expect that this problem is related to the strong zero-bias in our data (zeros outnumber ones by a large margin in the voxel representations of all cloth states). Experiments with more advanced rules invariably saw the net devolve into producing all-zero outputs (a fine first approximation, but hard to escape from). We never observed this problem with the Manhattan update rule. Additionally, the Manhattan update rule affords the easy combination of dissimilar losses shown in equation (7), and circumvents the vanishing gradient problem, both without additional hyperparameters to tune.

Since the use of the Manhattan rule is atypical, we include results for a configuration using standard Stochastic Gradient Descent (SGD) (configuration C7). Loss_s and loss_c are combined by simple summing. Using a modestly high initial learning rate of $5 \cdot 10^{-3}$, training first converges upon the all-zero solution mentioned above and temporarily stagnates there, but eventually escapes and achieves some level of prediction ability.

Alternative Encoding Consistency Enforcement

An alternative training scheme combines loss_s with a loss computed over s_i and $D(E(s_i))$, i.e., the typical autoencoder loss. Wahlström et al. (2015) adopt a loss term to this effect. To compare these variants, we define a reconstruction loss as follows:

$$loss_r = \frac{\sum (D(E(s_i)) - s_i)^2 + \sum (D(E(s_{i+1})) - s_{i+1})^2}{2 \cdot size_s} \quad (9)$$

For convenience, we let loss_r combine pre- and post-manipulation states (as our training examples provide both). Our experiments below include a configuration (configuration C6) that replaces loss_c with loss_r (combined with loss_s in the same manner as loss_c). This configuration, too, is trained with the Manhattan update rule.

Learning Rate Adjustment and Overfitting Counter-Measures

To appropriately adjust the learning rate as the net trains, and to avoid overfitting, we use a validation set of 900 examples (300 sequences). Every 10,000 batches, we evaluate prediction performance (i.e., loss_s) on the entire validation set. When validation set performance has not increased for 5 such evaluations in a row (i.e., over 50,000 batches) at the same learning rate, the learning rate is reduced by a factor 2. To avoid overfitting on the training data, we store a copy of the net whenever the validation score is improved, and perform all performance assessments below on these "validation-best" networks. This strategy can be considered a simple variant of early stopping (Morgan and Bourlard, 1990; Prechelt, 2012). This learning rate adjustment scheme was applied identically in all system configurations.

RESULTS—OUTCOME PREDICTION

Once trained, the network can fairly well predict the result of applying a given manipulation to a given cloth state. **Table 3** gives results over the test (top panel) and training (bottom panel) sets for all experiments, and **Figure 6** shows representative example results for configurations C4, C5, and C6. For generating these scores, all data augmentation types were enabled except for the failure-to-grasp augmentation (including this would artificially improve the scores). The binary scores are computed by rounding the values of all voxels to the nearest binary value, then taking the absolute difference w.r.t. the target state and dividing by the total number of voxels (16384).

We also include $D(E(s_i))$ and $D(E(s_{i+1}))$, i.e., the result of encoding and then directly decoding the initial state and goal state. Note however that except for configuration C6, these pathways are not trained directly. In general, the pre-manipulation state (s_i) is reconstructed more accurately than post-manipulation state (s_{i+1}). This is unsurprising: $s_i \in \{s_0, s_1, s_2\}$ whereas $s_{i+1} \in \{s_1, s_2, s_3\}$, and in general states occurring later in a sequence will have more complex shapes. In particular, s_0 is the same initial state in every sequence (albeit rotated by a random angle). Being simple and common, it is generally reconstructed with very high accuracy. This state never occurs in the $D(E(s_{i+1}))$ and $D(M(E(s_i),m_i)$ targets.

We trained a total of 8 system configurations. Configuration C4 is the default configuration discussed so far. Configuration C0 feeds the manipulation input only at the first layer of the manipulation network, to assess the effect of feeding it anew at every layer. Configuration C1 drops the residual connectivity in the manipulation module, to assess the effect of this connectivity on performance. Configuration C2 reduces the number of layers in the manipulation module from 10 to 5, to assess whether 10 layers is overkill for this task. Configuration C3 drops the discretization layers, meaning latent representations are continuous. Configuration C5 and C6 are included to investigate the role of encoding consistency enforcement. Configuration C5 drops loss_c, whereas configuration C6 features an alternative consistency enforcement scheme that replaces loss_c with loss_r (see section Alternative Encoding Consistency Enforcement). Lastly, configuration C7 replaces the Manhattan update rule with regular SGD as discussed above.

We compare the various configurations to our base configuration C4. Looking at the scores in **Table 3** we observe that C4 outperforms C0, C1, and C2 on both test and training data, indicating that feeding the manipulation input at all layers of the manipulation network and the inclusion of residual connections is beneficial, and (at a rough granularity) that the depth of the manipulation module is warranted. C4 is also seen to outperform C3 on both sets, but the difference is marginal at best, suggesting the contribution of discretization was limited.

Despite high prediction accuracy (exceeding C4 on the training set), configuration C5 produces by far the worst direct reconstructions. High prediction accuracy despite dismal reconstruction may seem contradictory at first glance. In a standard autoencoder, low-quality reconstruction would strongly imply low-quality latent representations, and it is hard to see how accurate prediction could be achieved with low-quality latent representations. However, the low reconstruction quality observed here is no indication of poor latent representation quality, but of inconsistency of representation format between the representations produced by the encoder and the manipulation network. The decoder can only meaningfully decode representations produced by the latter, hence prediction succeeds but reconstruction fails. This result indicates that the consistency of representation format achieved by the default configuration (evidenced by the combination of high prediction accuracy and high reconstruction accuracy) is indeed due to the inclusion of loss_c, Configuration C6 excels in reconstruction, exceeding all other configurations on both the test and training set. This is to be expected, as it is the only configuration explicitly trained to reconstruct. Overall, prediction ability is close between C3, C4, C5, and C6. Configuration C7 falls short of all other configurations in terms of prediction ability, and short of all other configurations except C5 in terms of reconstruction ability, showing that our variant of the Manhattan update rule was more effective for training this particular network architecture than standard SGD.

TABLE 3 | Prediction results on test and training data.

Configuration		Error Measure	$D(E(s_i))$	$D(E(s_{i+1}))$	$D(M(E(s_i),m_i))$
TEST SET					
C0	Single input	MSE	0.00567 (0.0043)	0.00852 (0.0026)	0.0114 (0.0045)
		binary	0.00781 (0.0062)	0.0119 (0.0038)	0.0158 (0.0060)
C1	No residual connectivity	MSE	0.00724 (0.0055)	0.0107 (0.0030)	0.0120 (0.0044)
		binary	0.00990 (0.0078)	0.0148 (0.0044)	0.0165 (0.0058)
C2	Shallow M	MSE	0.00633 (0.0046)	0.00939 (0.0027)	0.0122 (0.0045)
		binary	0.00869 (0.0067)	0.0130 (0.0040)	0.0167 (0.0060)
C3	Continuous	MSE	0.00537 (0.0042)	0.00812 (0.0026)	0.0109 (0.0045)
		binary	0.00741 (0.0061)	0.0113 (0.0038)	0.0150 (0.0059)
C4	Default	MSE	0.00522 (0.0041)	0.00790 (0.0025)	**0.0107** (0.0047)
		binary	0.00722 (0.0059)	0.0110 (0.0037)	**0.0148** (0.0060)
C5	$Loss_s$ only	MSE	0.0272 (0.0058)	0.0311 (0.0044)	0.0108 (0.0049)
		binary	0.0384 (0.0035)	0.0386 (0.0040)	0.0149 (0.0062)
C6	$Loss_s$ & $loss_r$	MSE	**0.00151** (0.0014)	**0.00245** (0.0011)	0.0110 (0.0047)
		binary	**0.00204** (0.0019)	**0.00335** (0.0015)	0.0151 (0.0060)
C7	SGD	MSE	0.0153 (0.0050)	0.0185 (0.0045)	0.0191 (0.0051)
		binary	0.0194 (0.0077)	0.0245 (0.0060)	0.0249 (0.0070)
TRAINING SET					
C0	Single input	MSE	0.00562 (0.0043)	0.00843 (0.0027)	0.0110 (0.0042)
		binary	0.00772 (0.0062)	0.0117 (0.0039)	0.0152 (0.0056)
C1	No residual connectivity	MSE	0.00719 (0.0055)	0.0107 (0.0032)	0.0117 (0.0041)
		binary	0.00983 (0.0078)	0.0147 (0.0046)	0.0160 (0.0055)
C2	Shallow M	MSE	0.00629 (0.0047)	0.00931 (0.0028)	0.0120 (0.0045)
		binary	0.00860 (0.0067)	0.0129 (0.0041)	0.0164 (0.0060)
C3	Continuous	MSE	0.00531 (0.0042)	0.00802 (0.0026)	0.0101 (0.0038)
		binary	0.00733 (0.0060)	0.0112 (0.0038)	0.0140 (0.0051)
C4	Default	MSE	0.00515 (0.0041)	0.00781 (0.0026)	0.00982 (0.0039)
		binary	0.00711 (0.0059)	0.0109 (0.0038)	0.0136 (0.0052)
C5	$Loss_s$ only	MSE	0.0271 (0.0057)	0.0309 (0.0043)	**0.00979** (0.0039)
		binary	0.0385 (0.0036)	0.0386 (0.0040)	**0.0135** (0.0051)
C6	$Loss_s$ & $loss_r$	MSE	**0.00146** (0.0014)	**0.00239** (0.0011)	0.0101 (0.0039)
		binary	**0.00198** (0.0019)	**0.00327** (0.0015)	0.0138 (0.0052)
C7	SGD	MSE	0.0152 (0.0049)	0.0184 (0.0046)	0.0191 (0.0052)
		binary	0.0192 (0.0077)	0.0244 (0.0061)	0.0249 (0.0072)

Mean squared and binary error scores, for reconstruction, $D(E(s_i))$ and $D(E(s_{i+1}))$, and outcome prediction, $D(M(E(s_i),m_i))$, for test (top panel) and training (bottom panel) datasets, under various system configurations. Standard deviations in brackets. Best scores for each item bolded.

Looking at **Figure 6A** we can observe that configuration C4 produces appropriate (though moderately diffuse) reconstructions and predictions of the target states. Incidental prediction failures are observed on examples with one or both grasp points lying very close to the edge of the cloth. The cloth's edges correspond to sharp discontinuities in the relation between manipulation inputs and outcomes: grasping the cloth right at its edge produces a very different outcome from failing to grasp the cloth by a millimeter. The observed failures can often be understood as mistaking one of these situations for the other.

Figure 6B shows reconstructions and predictions from configuration C5. The "reconstructions" here bear no discernible resemblance to the targets at all. We initially thought that even in absence of $loss_c$, the residual connectivity of the manipulation module may produce some degree of encoding consistency, but the outcomes do not support this notion. Nonetheless, prediction accuracy is high. **Figure 6C** shows reconstructions and predictions from configuration C6. Here too, prediction is close to configuration C4, whereas reconstruction is by far the most accurate across all configurations.

Figure 6D shows prediction results w.r.t. two common folds: folding in half along one of the cloth's axes, and folding in half along a diagonal. The results are centered in view for ease of interpretation. These examples were generated for purpose of illustration, and are not part of the dataset. We observe that prediction quality on these examples is in line with prediction on the random examples in the test set.

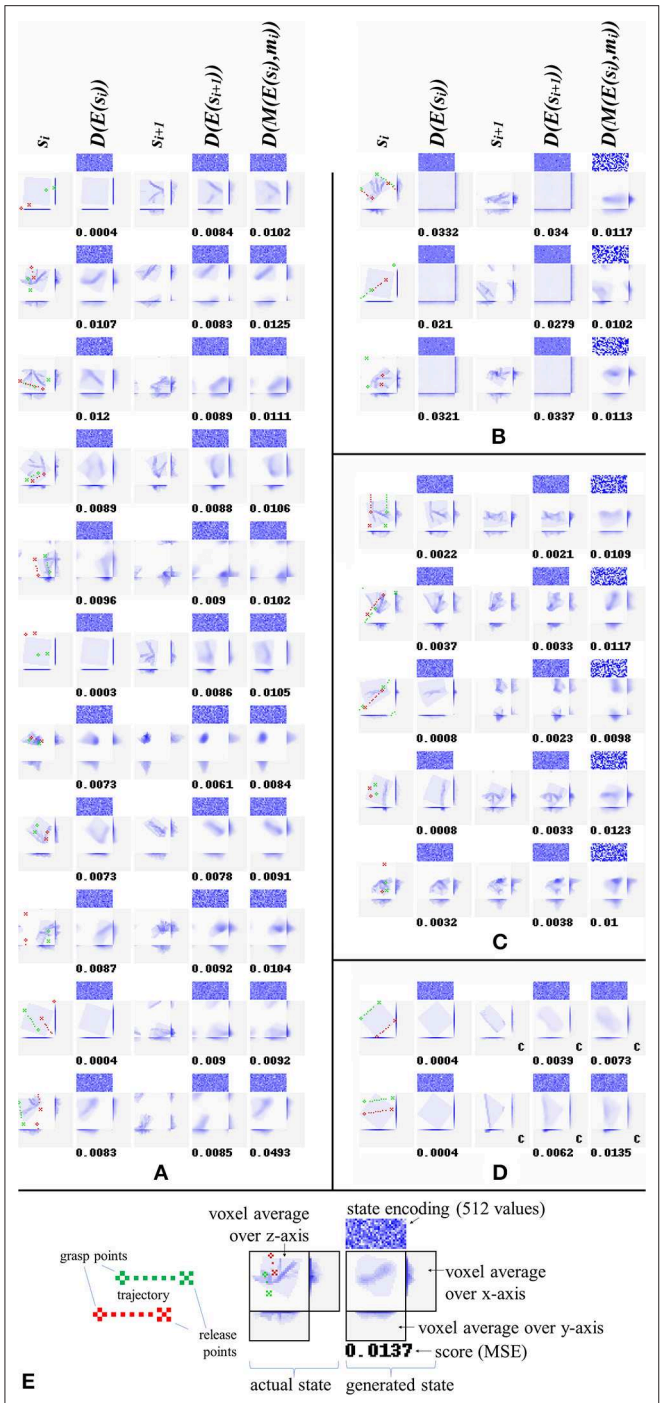


FIGURE 6 | Reconstructions and predictions generated by various configurations. Each row corresponds to one result, showing (from left to right): start states s_i, their reconstructions $D(E(s_i))$, manipulation outcomes s_{i+1}, their reconstructions $D(E(s_{i+1}))$, and outcome prediction results $D(M(E(s_i),m_i))$. Along with each generated state is shown a direct visualization of its encoding (512 real values, shown as blue tones) and its MSE. All examples are from the test set. **(A)** Results for configuration C4. The last example in panel a shows a typical failure case (the net mistakes a grasp point right on the cloth edge for a miss, and consequently predicts no change of shape). **(B)** Results for configuration C5. Training without $loss_C$ produces

(Continued)

FIGURE 6 | accurate predictions, but the "reconstructions" show no obvious resemblance to their targets. **(C)** Results for configuration C6. Training with $loss_I$ instead of $loss_C$ produces highly accurate reconstruction, and prediction accuracy similar to configurations C3, C4, and C5. **(D)** Prediction results for two common manipulations: folding in two along one of the cloth's axes, and folding in two along a diagonal. For ease of interpretation, we manually centered the outcomes for these examples (centered outcomes are marked with a letter *c* in their lower right corner). **(E)** Legend explaining the figure format.

RESULTS—PLANNING

Next we assess the system's planning performance. From a manipulation sequence of length 3, represented as a 7-tuple $<s_0, m_0, s_1, m_1, s_2, m_2, s_3>$, we can extract a total of six subsequences $<s_i, m_i, \ldots, s_{i+n}>$: $(n,i) \in [(1,0), (1,1), (1,2), (2,0), (2,1), (3,0)]$. Note that subsequences (n,i) for which $i = 0$ start from the fully spread state, whereas subsequences for which $i > 0$ start from states generated by application of i random manipulations. For $n = 3$ the only subsequence is the full sequence, and consequently planning for 3 steps always starts at the fully spread state. As in training and prediction evaluation, random rotation and mirroring is applied. For planning we apply rotation and mirroring identically to each state and manipulation in a sequence, in order to maintain sequence coherence. The other data augmentation operators are not applicable for planning.

For each sub-sequence we run the system on 100 examples, following the closed-loop planning procedure detailed in section Closed-Loop Planning. **Table 4** shows the scores obtained by configurations C3, C4, C5, and C6 for both test (top panel) and training (bottom panel) datasets (C0, C1, C2, and C7 were excluded as they evidently fell short in prediction ability). Scores represent the mean absolute errors between the goal state and the state actually obtained by performing the planned manipulations in simulation, both in voxel representation (since both are binary representations, this is identical to the MSE). Medians were included because the presence of occasional failures sometimes skews the mean upward. As is to be expected, there is some falloff in accuracy as plans get longer, but recognizable approximations of the goal state are obtained for all plan lengths tested here. What performance gap there is between training and test data appears to be below the noise level of this assessment, suggesting that the net did not overfit substantially and generalizes well to unseen data.

Configuration C4 succeeds in both single and multi-step planning. **Figure 7** shows representative examples of plans and outcomes. **Figure 8** shows a few iterations of the generation process of a 3-step plan. On a single NVIDIA GTX1080 GPU, plan generation took <2.5 s on average. Plan generation times per plan length are given in **Table 5**. **Figure 9** shows plans and outcomes obtained for a small number of common folds. The goal states here are not part of the dataset, but were modeled manually in simulation for the purpose of illustration. We observe that the trained network is capable of generating sensible plans that produce adequate approximations of these goal states.

TABLE 4 | Planning results for test and training data.

Test set			Sequence					
Configuration			1-0	1-1	1-2	2-0	2-1	3-0
C3	Continuous	μ	0.0134	0.0166	0.0181	0.0210	0.0254	0.0243
		σ	0.0063	0.0069	0.0099	0.0066	0.011	0.0083
		M	0.0130	0.0174	0.0170	0.0210	0.0237	**0.0224**
C4	Default	μ	0.0130	**0.0153**	**0.0177**	**0.0206**	**0.0250**	**0.0236**
		σ	0.0080	0.0064	0.010	0.0063	0.0082	0.0063
		M	**0.0123**	**0.0151**	0.0163	**0.0206**	**0.0232**	0.0226
C5	$Loss_S$ only	μ	0.0401	0.0439	0.0430	0.0512	0.0519	0.0557
		σ	(0.019)	(0.016)	(0.018)	0.013	0.012	0.011
		M	0.0406	0.0442	0.0438	0.0513	0.0518	0.0561
C5*	$Loss_S$ only	μ	**0.0128**	0.0167	0.0183	0.0335	0.0374	0.0385
		σ	0.0065	0.0070	0.011	0.012	0.016	0.015
		M	0.0127	0.0162	**0.0156**	0.0312	0.0316	0.0330
C6	$Loss_S$ and $loss_r$	μ	0.0346	0.0403	0.0397	0.0438	0.0460	0.0468
		σ	0.018	0.016	0.017	0.015	0.012	0.013
		M	0.0364	0.0378	0.0385	0.0447	0.0437	0.0466
C6*	$Loss_S$ and $loss_r$	μ	0.0131	0.0175	0.0180	0.0244	0.0277	0.0292
		σ	0.0067	0.0095	0.010	0.0090	0.012	0.0093
		M	0.0125	0.0161	0.0168	0.0233	0.0251	0.0272
Training set			**Sequence**					
Configuration			**1-0**	**1-1**	**1-2**	**2-0**	**2-1**	**3-0**
C3	Continuous	μ	0.0135	0.0156	0.0177	**0.0212**	0.0221	**0.0238**
		σ	0.0073	0.0055	0.012	0.0068	0.0081	0.0072
		M	0.0123	0.0150	0.0158	0.0213	0.0205	**0.0229**
C4	Default	μ	0.0124	**0.0154**	0.0158	0.0217	**0.0201**	0.0248
		σ	0.0060	0.0073	0.0086	0.0078	0.0076	0.0088
		M	0.0123	0.0148	0.0143	**0.0211**	**0.0197**	0.0240
C5	$Loss_S$ only	μ	0.0405	0.0466	0.0440	0.0505	0.0511	0.0535
		σ	0.017	0.015	0.018	0.014	0.014	0.012
		M	0.0412	0.0468	0.0450	0.0522	0.0528	0.0553
C5*	$Loss_S$ only	μ	**0.0117**	0.0165	**0.0154**	0.0308	0.0328	0.0360
		σ	0.0058	0.0094	0.074	0.012	0.015	0.014
		M	**0.0107**	**0.0142**	0.0145	0.0281	0.0282	0.0311
C6	$Loss_S$ and $loss_r$	μ	0.0344	0.0414	0.0380	0.0432	0.0448	0.0474
		σ	0.016	0.017	0.018	0.015	0.014	0.012
		M	0.0349	0.0410	0.0384	0.0419	0.0438	0.0463
C6*	$Loss_S$ and $loss_r$	μ	0.0129	0.0163	0.0157	0.0229	0.0268	0.0288
		σ	0.0066	0.0083	0.010	0.0080	0.011	0.011
		M	0.0132	0.0144	**0.0142**	0.0225	0.0240	0.0262

*Average (μ), standard deviation (σ), and median (M) of binary errors for each type of subsequence, for the test (top panel) and training (bottom panel) datasets. For subsequence types, n-i indicates a subsequence of length n, starting at manipulation i of the source sequence. Each score is an average over 100 examples from the relevant set. Example sets used for different subsequence types are non-overlapping. Best scores (per subsequence type) in bold. Results with a * mark in the configuration column were obtained using an alternative planning algorithm (see text).*

Plan generation with configuration C5 fails when we use the algorithm given in section Planning Algorithm. However, this is in part a consequence of speed optimization (see **Figure 4B**). The planning process aims to reduce the loss between latent representations of the goal state and the expected outcome for the manipulation input. Comparing the two only makes sense if there is sufficient encoding consistency between the two. Configuration C5 does not enforce consistency during training, so failure here is not unexpected. To perform planning with C5, we can follow the strategy given in **Figure 4A**, using the loss between the voxel representations of the goal state and the expected outcome for the manipulation input. This strategy is slower (taking roughly twice as long), as we need to run the decoder every time the manipulation inputs are updated, but

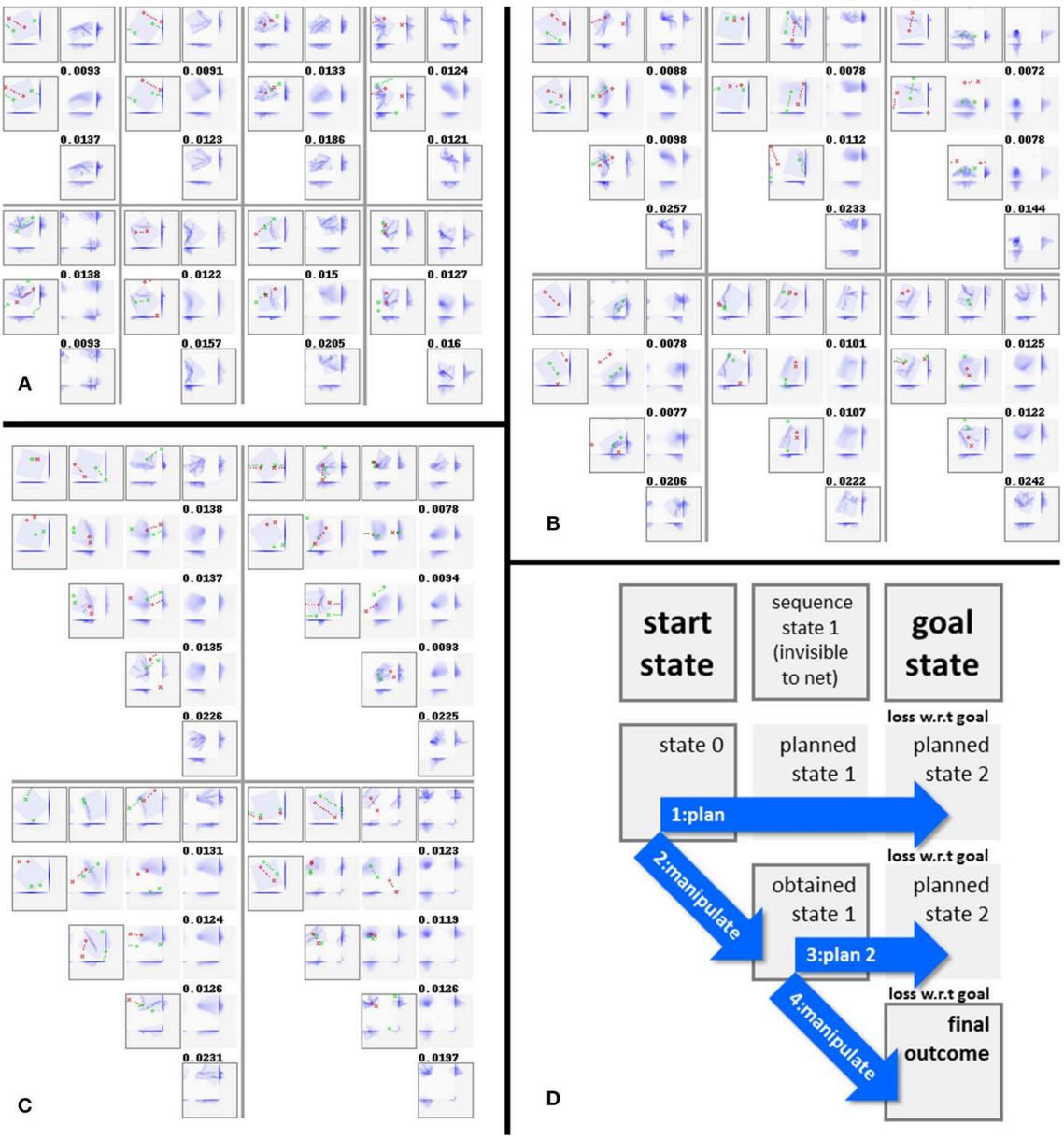


FIGURE 7 | (A–C) Representative 1, 2, and 3-step manipulation results, respectively, for the default configuration (C4). Manipulations and intermediate states of the original sequence (top row in each example) shown for reference only; the net sees the start and goal states only. Framed states are voxelisations of actual (i.e., simulation-generated) states, non-framed states are network-generated predictions. Numbers above states indicate MSE loss w.r.t. the goal state. Note that manipulation trajectories can wrap around the edges of the viewport, and that the camera shifts along with the manipulation trajectory. **(D)** Legend explaining the figure format.

otherwise similarly effective. Results for C5 obtained with this alternative planning strategy are given in **Table 4** as C5*.

Configuration C6 had encoding consistency enforced via loss_r and indeed fares better than C5, but still falls short of C3 and C4 by a large margin on both test and training data. Assuming that loss_r was not quite effective for this purpose, we ran C6 with the alternative planning scheme as well, and give the results as C6*.

The alternative planning scheme brings C5* scores for 1-step plans up to par with C4, with C5* doing slightly better on the training set and C4 doing slightly better on the test set (repeating the pattern seen for prediction). However, scores for multi-step planning remain poor. This is to be expected: without an encoding consistency enforcing loss, we obtain an M module that predicts the results of individual steps accurately, but cannot read its own output. This makes it unfit for recurrent application, leading to failure in multi-step planning.

The alternative planning scheme brings C6* scores for 1-step plans close to C4 as well for both test and training data. Scores

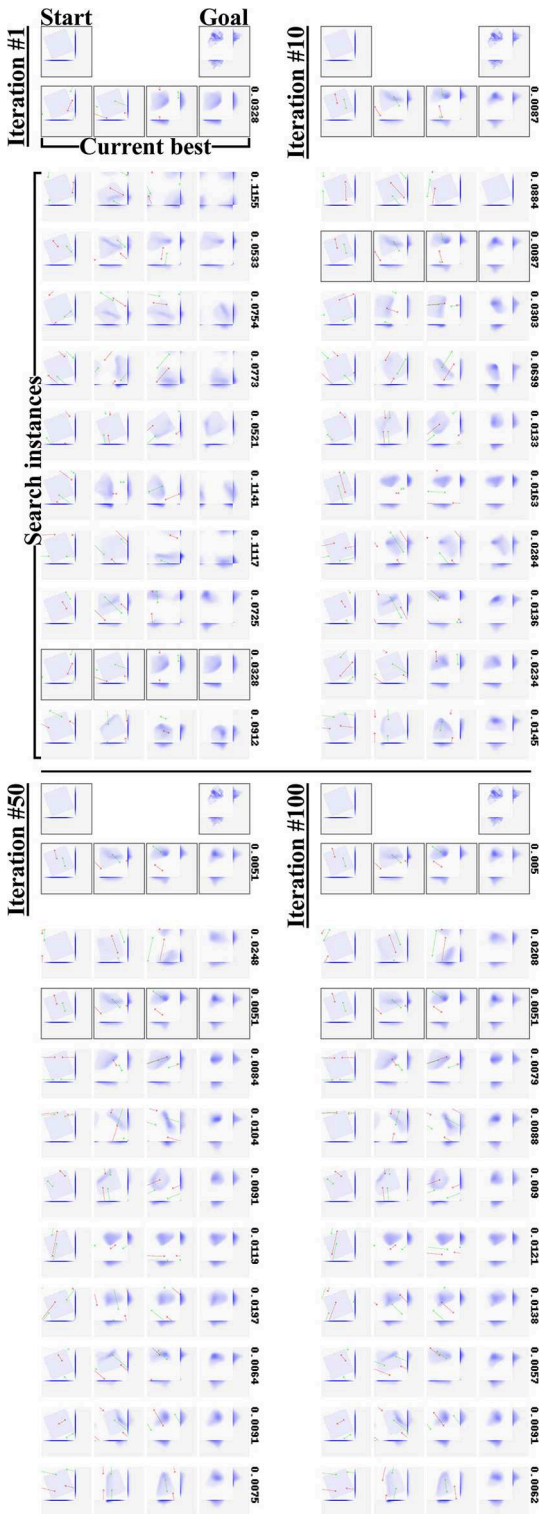


FIGURE 8 | Iterations 1, 10, 50, and 100 from the generation process for a 3-step plan. Each panel shows the task, the best plan up to the iteration, and the current plan in each of the 10 parallel search instances. Numbers to the right of a plan indicate its residual error w.r.t. the goal state. We see the search instances quickly converging on a variety of plans for approximating the target outcome.

TABLE 5 | Mean plan generation times per plan length.

		Plan length		
	Configuration	**1**	**2**	**3**
C3	continuous	1.85s (0.039)	2.06s (0.047)	2.27s (0.041)
C4	default	1.90s (0.050)	2.10s (0.070)	2.30s (0.52)
C5*	$loss_s$ only	4.33s (0.082)	4.51s (0.10)	4.65s (0.066)
C6*	$loss_s$ and $loss_r$	4.29s (0.067)	4.49s (0.083)	4.65s (0.068)

*Times in seconds. All times measured over the test set. Numbers in brackets are standard deviations. Results marked with a * used an alternative planning scheme (see text).*

for multi-step plans are improved too, but fall short of C4. This again suggests that $loss_r$ did not enforce encoding consistency as effectively as $loss_c$. We hypothesize as follows: configuration C6 trains to minimize $loss_s$ and $loss_r$. When both are near zero, this implies that D can decode latent representation *c* of a given state *s* into an approximation of *s*, regardless whether c was produced by E or M. This suggests encoding consistency between E and M, but does not guarantee it: D may be decoding different latent representations into similar state approximations. $Loss_c$ on the other hand, directly and specifically enforces encoding consistency with no such wiggling room.

Whereas C4 outperforms C3 by a slight margin on most sequences in the test set, the scores are again very close, indicating that the effect of the discretization layers was marginal at best. As seen in **Table 5**, plan generation with C3 was faster by a similarly diminutive margin.

It bears emphasizing why back-propagation (BP) can arrive at good solutions within 2.5 s. During plan generation, we only search for input values, not connection weights. The number of input values for a plan of length *n* is *6n*, so BP for generating a plan of a few steps has vastly fewer variables to optimize than BP in the training process has. Furthermore, planning does not work on batches of examples; only one input state and one output state need to be considered (in our implementation the batch dimension is instead used to run multiple search instances in parallel). The small scale of the problem allows us to zoom in on solutions quickly with aggressive use of iRprop-.

The plan generation process need not be implemented with BP. In preliminary experimentation, we found plan generation by means of a genetic algorithm to be viable as well (the trained network then serves as the evaluation function for solutions generated by the genetic algorithm). However, the fact that the BP machinery is already present in the system for training makes the BP approach particularly convenient to implement.

It is worth noting the planning process we employ is not deterministic. Where multiple solutions exist, different runs of the search process (and different search instances in parallel search, as seen in **Figure 8**) can produce different solutions.

SELF-OCCLUSION

So far, we have considered the case where the cloth shape is fully visible (albeit at low resolution) to the network. Full visibility

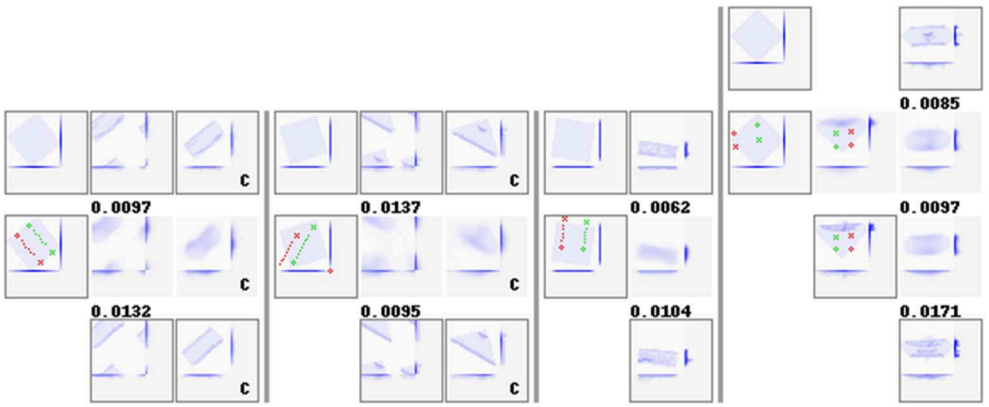


FIGURE 9 | Manipulation plans and outcomes obtained for four hand-crafted goal states. See **Figure 7D** for the figure format. For examples that wrap around the view border (first two examples), we include manually centered views of the goal, prediction, and result states. Centered states are marked with a letter *c* in their lower right corner.

would be hard to achieve in a physical implementation on robotic hardware. To be applicable in a physical setup, the system must be able to handle self-occlusion. In this section we evaluate the performance impact of self-occlusion. We assume to have a single top-down view of the cloth, recorded using a depth camera. To replicate this view limitation on simulation data, we occlude all voxels below the top-most 1-voxel in each z-column of the voxel space. Occluded voxels are given the value 1, same as known-occupied voxels. We evaluate the impact of occlusion by training the default configuration from scratch with occlusion enabled and all other settings unchanged. All states presented to the network, in training, prediction, and planning, are given with occlusion applied. Consequently, predictions are also generated with occlusion present.

Since the state representation differs from the preceding experiments, prediction results cannot be compared straightforwardly to results obtained without occlusion. However, planning performance can be evaluated as before, since it involves comparison of actual states only. As before, we compute the binary error scores over non-occluded voxel representations of the outcome and goal state. **Table 6** shows the results for the occlusion experiment, for both test and training data. Comparing these results to the results obtained by configuration C4 without occlusion (**Table 4**), we observe that the performance impact of occlusion is small. The use of incomplete state representations carries a risk of inducing more overfitting, but we do not observe a widening of the gap between training and test scores. These results indicates that even with the occlusion incurred by a single static top-down view, the cloth state representation generally still provides sufficient information to allow effective planning.

TABLE 6 | Planning results with self-occlusion.

Occlusion		Sequence					
		1-0	1-1	1-2	2-0	2-1	3-0
Test set	μ	0.0136	0.0162	0.0187	0.0210	0.0249	0.0252
	σ	0.0068	0.0073	0.010	0.0068	0.010	0.0089
	M	0.0141	0.0160	0.0173	0.0218	0.0230	0.0229
Training set	μ	0.0131	0.0165	0.0171	0.0213	0.0230	0.0236
	σ	0.0066	0.0061	0.0081	0.0062	0.0088	0.0070
	M	0.0127	0.0160	0.0164	0.0220	0.0221	0.0225

Average (μ), standard deviation (σ), and median (M) of binary errors for each type of subsequence, for the test (top) and training (bottom) datasets. For subsequence types, n-i indicates a subsequence of length n, starting at manipulation i of the source sequence. Each score is an average over 100 examples from the relevant set. Example sets used for different subsequence types are non-overlapping.

DISCUSSION

The EM*D network functions as a forward model that is differentiable, and therefor searchable, w.r.t. the manipulation repertoire it is trained on. Given an economically defined manipulation repertoire, planning is not a high-dimensional problem. In the present work, planning a manipulation sequence of length n is reduced to gradient descent search in a $6n$-dimensional space.

The task of multi-step cloth folding could also be cast as a model-free reinforcement learning (RL) problem, so it is worth noting the merits afforded by each approach. First off, while both approaches generate goal-directed behavior, ours generates such behavior in the form of explicit plans. There is a large conceptual difference between learning to pursue a given goal state, and learning the deformation and movement characteristics of a given task environment. This has practical consequences for the training procedure. The former requires evaluation of manipulation outputs during training (which for our task would be computationally very costly). The latter, as demonstrated here, allows training from a static database of examples. Furthermore, as this style of training is goal-agnostic, the goal state for planning can be set freely, and the planning process can be constrained using additional loss terms without having to retrain the net (which is presently proving useful in integrating the system

with a physical robot platform with a limited range of motion). The cost of these merits is that the planning process is slow (taking seconds) compared to the action-generation time cost of model-free RL systems.

We found that the system emphasizes overall shape (what we might call the 3D silhouette of the cloth) over the details of how that shape is realized. For example, given as target a cloth folded neatly in two, the system will produce a plan that produces the same rectangular shape, but not necessarily with the fold on the same side. This is to be expected (as the loss used in plan generation quantifies the difference between voxel representations), but not ideal. Marking or patterning of the cloth (e.g., adding an additional color channel to distinguish the cloth's hem) can likely improve this issue, but would also constrain the applicability of the system to cloths adhering to the marking scheme.

A related issue is that predictions are rather diffuse. Some degree of diffusion is theoretically appropriate. The voxelisation introduces some ambiguity with respect to the cloth mesh. Predictions from a network trained to perfection would be distributions over the manipulation outcomes for all the mesh configurations that would produce the given input voxel representation. This diffusion could be reduced (though not eliminated) by simply using higher resolution voxel representations, at the cost of slower training and operation. However, there are also some unwanted sources of blur affecting the results presented here. One is that autoencoders tend to replicate low-frequency features better than high-frequency features. This is a well-known problem (Snell et al., 2015). Low-frequency features generally have a bigger impact on the loss function, causing them to dominate the training process. However, the fact that the reconstruction results of configuration C6 show only limited diffusion suggests that the root cause of the blur in our prediction results is not just due to an emphasis on low-frequency features. Further supporting a different root cause, we experimented with 3D SSIM (Snell et al., 2015) loss functions in hopes of improving prediction clarity, but did not obtain notable improvements over the MSE loss used in the experiments here. Adversarial training (Goodfellow et al., 2014) has been shown capable of improving the quality of generated images. It is imaginable that sharper predictions could be obtained with an adversarial approach, but it is not a given that this would contribute to better multi-step planning. Sharpness only contributes to planning performance insofar it reflects increased prediction accuracy, whereas the sharpness obtained by adversarial training is obtained by the objective of reducing distinguishability between real and generated states. Also, introducing a discriminator net would substantially complicate the system. A third factor is the somewhat stochastic nature of the data itself, caused by limitations of the cloth simulation. The simulated cloth generally does not stabilize entirely. After the grasp is released and the cloth has settled into a folded state, it continues to jitter slightly, which slowly changes the cloth shape (left to run for a long time, this jittering can even cause the cloth to unfold entirely). Our data generation procedure stops the simulation 16 frames after the grasp is released, which includes variable amounts of such jitter. This introduces a source of noise, which is likely reflected in the predictions. Adoption of a more stable cloth simulation algorithm should resolve this issue. However, the system is developed with the aim of operating on real cloth, which introduces a different set of noise sources, so exact predictions will likely remain hard to obtain.

Perhaps somewhat counter-intuitively, we see that manipulation outcomes often resemble the goal state more closely than their predictions do. Evidently the planning process is robust to some level of blur. This can be understood as follows. What is required for planning to function is that the prediction's match w.r.t. the goal state should improve (i.e., planning loss should decrease) as the manipulation input approaches the correct action. Differently put, prediction ability does not need to produce the goal state, it only needs to provide predictions that are good enough to identify the goal state among other possible outcomes. This permits a fair level of noise and blur.

Next, we position our approach among related work in generative models, control, and planning. The EM*D net could be considered a relative of transforming autoencoders (Hinton et al., 2011) and variational autoencoders (Kingma and Welling, 2014; Rezende et al., 2014). Whereas typical autoencoders are mainly used for compression and feature extraction, transforming and variational autoencoders are used to generate novel outputs. Dosovitskiy et al. (2014) introduced techniques to make individual values in an autoencoder's latent encoding control specific features of the representation, and demonstrated how this can be used to change specific features (e.g., color, view angle) in a controlled manner. The EM*D net, too, compresses and modifies representations in a controlled manner, although not by changing specific features or view angles but by applying specific manipulations.

As noted in the introduction, the use of neural networks as forward models can also be found in research on model-based control. Wahlström et al. (2015), Watter et al. (2015) similarly use neural networks to map states into a latent space, in which a variety of control problems is then solved. Both studies map high-dimensional states (images) into latent space using encoder networks, and then solve control problems in latent space. A first point to note is that whereas both these studies solve control tasks, we focused on a planning task. This is fundamentally a difference in the size of the time-step: our step units are full manipulations (representing large jumps in state-space), whereas the control studies consider smaller and more granular transitions. For convenience, we will use the term "actions" here to refer to the object of inference (i.e., control signals in the control studies, and manipulation plans in the present work). Both (Wahlström et al., 2015) and (Watter et al., 2015) employ high-dimensional observations (images) of low-dimensional tasks (2D in the former, 2D to 6D in the latter). Both systems successfully reduce these observations back to low dimensionality. However, neither compresses state representations beyond the original task dimensionality. The present work employs high-dimensional observations of an intrinsically high-dimensional task. The actual state dimensionality is, strictly speaking, 19200D (x, y and z coordinates for 80 × 80 cloth vertices). Observations are

16384D voxel representations, which the encoder compresses into 512D latent representations. The encoder's task here is not to recover or approach the task's actual dimensionality; the actual dimensionality is too large to plan on effectively. Rather, the encoder must learn a manifold of low-dimensional representations of a high-dimensional state space, that also allows for easy manipulability. Our work shows that the concept of action inference in latent space can be applied effectively under the demanding conditions of intrinsically high-dimensional real-world problems.

In terms of action inference (Watter et al., 2015) stay close to the variational auto-encoder paradigm. Transitions are performed by means of linear transformations in latent space. This is effective for small time-steps in low-dimensional control problems and has advantages in terms of solution search, but it is not clear how well imposed linearity would play in inherently high-dimensional problems with large time-steps, such as treated here. Our approach is close to Wahlström et al. (2015), combining non-linear latent state transformations with backpropagation-based solution search. The concept of planning by means of back-propagation has also been discussed and demonstrated in Henaff et al. (2017), with the purpose of extending this approach to discrete state and action spaces. A shallow recurrent neural network architecture was used there, and no mapping to latent spaces was employed. In a wider scope, planning by means of back-propagation is a special case of planning by means of gradient descent, which has a history outside the context of neural networks. The present work shows that this concept is effective on moderately deep neural architectures and in combination with manifold learning.

Sergeant et al. (2015) propose an interesting autoencoder-like architecture for control of a mobile robot which generates control signals along with a reconstruction of the robot's sensor input (laser range scan measurements). Sensor input is associated with control signals using a mix of supervised and unsupervised learning. This approach, too, targets control, and is not applicable for planning as-is, as it does not accommodate variable goal states.

Another closely related work is (Finn et al., 2016). This work combined autoencoders with RL to accomplish a variety of robotic manipulation tasks, including some on deformable objects. Here too the role of the autoencoder is to extract compact representations suitable for driving control, but in contrast to other work, the autoencoder's architecture is designed specifically to extract feature points that indicate the locations of objects in the scene. The cost function used in the RL process bears notable resemblance to our loss_c: the encoder is applied to obtain the latent representation of the goal state, which can then be compared to the latent representation of the current state to compute the cost. However, the use of RL requires that the goal state be set at training time. Consequently the trained system does not accommodate goal variability.

Koganti et al. (2017) also employ automatically acquired low-dimensional latent representations of cloth states, but in contrast to the autoencoder-based architectures above a Bayesian Gaussian Process Latent Variable Model (BGPLVM) is used. After training on motion capture and depth sensor data, the latent variable model is used to map noisy and high-dimensional depth sensor readings to cloth configurations in a task-specific, low-dimensional manifold. This approach was demonstrated on a dressing assistance task. This example illustrates the effectiveness of aggressive dimensionality reduction and manifold learning for cloth manipulation.

Finally, Erickson et al. (2018) presents a conceptually close example of Model Predictive Control (MPC) applied in cloth manipulation (a dressing assistance task). The forward model employed here operates in the haptic domain instead of the visuospatial domain, predicting the forces an action would exert (indirectly, through the clothing item) on the subject being dressed. No mapping to a latent space is performed (as cloth shape is not explicitly represented high dimensionality is less of a hurdle), and the task is again one of control rather than planning, but the approach is close to ours in its use of a recurrent neural network architecture for prediction, and its goal-agnostic training procedure. Like in our approach, the latter allows for goal definition at run-time, and hence the approach can in principle accommodate goal variability without retraining.

When training a forward model with the intent to use it recurrently, encoding consistency is crucial. Our results demonstrate that training with a loss computed over the encoding-prediction pathway (loss_c) results in better planning ability than training with a loss computed over the encoding-decoding pathway (loss_r).

Our network architecture is specialized for the purpose of modeling state transformations. It consists of a section of neurons designated to hold the state representation (passed on via residual connections), a section of regular neurons, and repeated action input at every layer. Our results demonstrate that this architecture is beneficial for modeling cloth's forward dynamics in latent space. As the reasoning behind this architecture is not specific to cloth manipulation, its benefits potentially extend to other task domains as well, although this remains to be investigated.

With the eye on future practical application, it is important to consider how goal state representations could be set. Although we have not focused on this aspect yet, we can outline a few ways forward. A cloth item already in the goal state could be used to specify the goal state. This would be practical when folding a number of items when at least one similar item already in the intended goal configuration can be observed. Targets could also be acquired by having a human user produce the goal state once and storing it for later recall. Either approach could build up a database of goal states on the side for quick selection of a suitable goal state at a later time (either by the user or by a high-level planning process).

Higher flexibility could be obtained by relaxing the definition of a "goal state." There is no need for the goal state to be a literal cloth state. The planning process tries to maximize the similarity between the expected outcome and the goal. Preliminary experiments suggest that it is possible to set the goal as (a voxel representation of) the space we want to fit a given cloth item into. This allows us to consider scenarios like the following: A household robot is tasked with tidying up a room. A

high-level planning process decides to store some scattered cloth items in a closet drawer. The drawer's internal sizes are estimated and passed to the cloth manipulation planning system for use as goal state. The manipulation planning system then returns a suitable manipulation sequence, allowing the robot to achieve its high-level goal.

FUTURE WORK

A limitation of the current system is that we need to set the length of the sequence to plan, which in practice will generally be unknown. Along with the manipulation plan, the system outputs the plan's expected loss w.r.t. the target state, providing a natural quality assessment of the plan. This can be used to automatically search for the appropriate sequence length sequentially. Alternatively, it may be possible to search over variable plan lengths simultaneously. One could use an appropriately defined loss over the sequence of latent representations generated by the propagation loops through the manipulation module (with a small penalty term for plan length). Implementation and evaluation of such procedures remains as future work.

Related to the above, dynamic plan length adjustment during operation could be exploited for failure recovery. Discrepancy between predicted and observed outcome can be used to infer failure. For example, in the case of a failure to grasp, one will want to add one step in order to allow a retry of the failed manipulation.

Another avenue for improvement is expansion of the manipulation repertoire. In particular, the present manipulation format enforces that both grasp points are moved by the same displacement vector. This restriction should be relaxed, as there are common manipulations in manual cloth folding that involve different movement vectors for different grasping points. However, allowing diverging trajectories for the grasp points also introduces manipulations into the repertoire that would pull the cloth apart, so this expansion requires some careful consideration.

Concurrent to further development of the planning system, we are integrating the system with a dual-armed robot. Initial results are reported in Tanaka et al. (2018). The system as discussed in the present paper assumes certain idealizations that do not carry over to real-world application. One difference to account for is the assumption of an infinite desk and point-sized non-colliding actuators with infinite range of motion. The constraints imposed by a finite desk and real robot hands can be accounted for with constraints on the planning process in the form of additional loss terms, but work remains in defining these loss terms efficiently and balancing them with the main planning loss.

CONCLUSIONS

We proposed the EM*D neural network architecture for generating multi-step cloth manipulation plans, and experimentally demonstrated its viability on simulated cloth. This approach to manipulation planning combines flexibility (variable start and goal states) speed (plans are generated in seconds), and robustness to cloth self-occlusion, core prerequisites for practical application in household robotics. Future work will focus on accuracy improvement, expansion of the manipulation repertoire, and continued integration with robotic hardware.

AUTHOR CONTRIBUTIONS

SA and KY contributed conception and design of the study and contributed to manuscript revision and approved the submitted version. SA implemented the system, performed the experiments, and drafted the paper.

FUNDING

This work was supported in part by the Japanese Society for the Promotion of Science (JSPS) (grant nr.: 26700024) and the New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Available online at: www.tensorflow.org

Arnold, S., and Yamazaki, K. (2017). "Cloth manipulation planning by back-propagation using a 3D convolutional auto-encoder and a recurrent neural network," in *The 35th Annual Conference of the Robotics Society of Japan* (Kawagoe).

Blender (2017). Available online at: www.blender.org.

Dosovitskiy, A., Springenberg, J. T., Tatarchenko, M., and Brox, T. (2014). *Learning to Generate Chairs, Tables and Cars with Convolutional Networks.* arXiv: 1411.5928. doi: 10.1109/CVPR.2015.7298761

Erickson, Z., Clever, H. M., Turk, G., Liu, C. K., Kemp, C. C. (2018). "Deep haptic model predictive control for robot-assisted dressing," in *2018 IEEE International Conference on Robotics and Automation (ICRA).* (Brisbane).

Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., Abbeel, P. (2016). "Deep spatial autoencoders for visuomotor learning," in *2016 IEEE International Conference on Robotics and Automation (ICRA).* (Stockholm).

Gläscher, J., Dew, N., Dayan, P., and O'Doherty, J. P. (2010). States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. *Neuron* 66, 585–595. doi: 10.1016/j.neuron.2010.04.016

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. et al. (2014). "Generative adversarial networks," in *Advances in Neural Information Processing Systems* (Montreal, QC: Curran Associates, Inc.) 27.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). *Deep Residual Learning for Image Recognition.* Tech Report. arXiv:1512.03385. doi: 10.1109/CVPR.2016.90

Henaff, M., Whitney, W. F., and Lecun, Y. (2017). *Model-Based Planning in Discrete Action Spaces.* arXiv:1705.07177.

Hinton, G. E., Krizhevsky, A., and Wang, S. D. (2011). "Transforming auto-encoders," in *Artificial Neural Networks and Machine Learning - ICANN 2011,*

Lecture Notes in Computer Science, eds T. Honkela, W. Duch, M. Girolami, and S. Kaski (Berlin, Heidelberg: Springer), 44–51.

Igel, C., and Hüsken, M. (2000). "Improving the rprop learning algorithm," in *Proceedings of the Second International Symposium on Neural Computation, NC'2000., 2000.* (Berlin: ICSC Academic Press).

Jang, E., Gu, S., and Poole, B. (2017). "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations (ICLR)* (Toulon).

Kingma, D., and Welling, M. (2014). "Auto-encoding variational bayes," in *International Conference on Learning Representations - ICLR* (Banff, AB; Ithaca).

Kita, K., Kanehiro, F., Ueshiba, T., and Kita, N. (2014). "Strategy for folding clothes on the basis of deformable models," in *Image Analysis and Recognition (ICIAR2014). Lecture Notes in Computer Science,* volume 8815, eds A. Campilho, and M. Kamel (Springer International Publishing).

Koganti, N., Tamei, T., Ikeda, K., and Shibata, T. (2017). Bayesian nonparametric learning of cloth models for real-time state estimation. *IEEE Trans Robotics* 33, 916–931. doi: 10.1109/TRO.2017.2691721

Koishihara, Y., Arnold, S., and Matsubara, T. Y. K. (2017). "Hanging work of T-shirt in consideration of deformability and strechability," in *IEEE International Conference on Information and Automation.* (Macau).

Lee, A. X., Gupta, A., Lu, H., Levine, S., Abbeel, P. (2015). "Learning from multiple demonstrations using trajectory-aware non-rigid registration with applications to deformable object manipulation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Hamburg).

Lee, S. W., Shimojo, S., and O'Doherty, J. P. (2014). Neural computations underlying arbitration between model-based and model-free learning. *Neuron* 81, 687–699. doi: 10.1016/j.neuron.2013.11.028

Lenz, I., Lee, H., and Saxena, A. (2015). Deep learning for detecting robotic grasps. *Int. J. Robot. Res.* 34, 705–724. doi: 10.1177/0278364914549607

Levine, S., Finn, , Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* 17, 1334–1373.

Li, Y., Yue, Y., Xu, D., Grinspun, E., Allen, P. K. (2015). "Folding deformable objects using predictive simulation and trajectory optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Hamburg).

Liljeholm, M., Wang, S., Zhang, J., and O'Doherty, J. P. (2013). Neural correlates of the divergence of instrumental probability distributions. *J. Neurosci.* 33, 12519–12527. doi: 10.1523/JNEUROSCI.1353-13.2013

Maddison, C. J., Mnih, A., and Teh, Y. W. (2017). "The concrete distribution: a continuous relaxation of discrete random variables," in *International Conference on Learning Representations (ICLR)* (Toulon).

Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., and Abbeel, P. (2010). "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," in *Proceedings of the International Conference on Robotics and Automation (ICRA)* (Anchorage, AK).

Morgan, N., and Bourlard, H. (1990). Generalization and parameter estimation in feedforward nets: some experiments. *Adv. Neural Inform. Process. Syst.* 2, 630–637.

Prechelt, L. (2012). "Early stopping - but when?" in *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science* (Berlin; Heidelberg: Springer).

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). *Stochastic Backpropagation and Approximate Inference in Deep Generative Models.* arXiv:1401.4082.

Riedmiller, M., and Braun, H. (1992). "Rprop - a fast adaptive learning algorithm," in *Proceedings of the International Symposium on Computer and Information Science VII* (Antalya).

Seita, D., Jamali, N., Laskey, M., Berenstein, R., Tanwani, A. K., Baskaran, P. et al. (2018). *Robot Bed-Making: Deep Transfer Learning Using Depth Sensing of Deformable Fabric.* arXiv:1809.09810.

Sergeant, J., Sünderhauf, N., Milford, M., and Upcroft, B. (2015). "Multimodal deep autoencoders for control of a mobile robot," in *Proceedings of the Australasian Conference on Robotics and Automation 2015 (ACRA 2015).* (Canberra, ACT).

Snell, J., Ridgeway, K., Liao, R., Roads, B. D., Mozer, M. C., Zemel, R. S. (2015). *Learning to Generate Images with Perceptual Similarity Metrics.* arXiv:1511.06409.

Tanaka, D., Arnold, S., and Yamazaki, K. (2018). EMD Net: An Encode-Manipulate-Decode Network for Cloth Manipulation. *IEEE Robot. Automat. Let.* 3, 1771–1778. doi: 10.1109/LRA.2018.2800122

Van den Oord, A., Vinyals, O., and Kavukcuoglu, K. (2017). *Neural Discrete Representation Learning.* arXiv:1711.00937.

Wahlström, N., Schön, T., and Deisenroth, M. (2015). "From pixels to torques: policy learning with deep dynamical models," in *Deep Learning Workshop at the 32nd International Conference on Machine Learning (ICML).* (Lille).

Watter, M., Springenberg, J. T., Boedecker, J., and Riedmiller, M. (2015). "Embed to control: a locally linear latent dynamics model for control from raw images," in *Advances in Neural Information Processing Systems 28 (NIPS2015)* (Montreal, QC).

Yang, P.-C., Sasaki, K., Suzuki, K., Kase, K., Sugano, S., Ogata, T. (2017). Repeatable folding task by humanoid robot worker using deep learning. *IEEE Robot. Automat. Lett.* 2, 397–403. doi: 10.1109/LRA.2016.2633383

Yuba, H., Arnold, S., and Yamazaki, K. (2017). Unfolding of a rectangular cloth from unarranged starting shapes by a Dual-Armed robot with a mechanism for managing recognition error and uncertainty. *Adv. Robot.* 31, 544–556. doi: 10.1080/01691864.2017.1285722

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 18 June 2019
doi: 10.3389/fnbot.2019.00039

Design and Validation of a Modular One-To-Many Actuator for a Soft Wearable Exosuit

Michele Xiloyannis[1,2†], Eugenio Annese[3†], Marco Canesi[4], Anil Kodiyan[5], Antonio Bicchi[6,7], Silvestro Micera[8,9], Arash Ajoudani[10] and Lorenzo Masia[11]*

[1] Robotics Research Centre, Interdisciplinary Graduate School, Nanyang Technological University, Singapore, Singapore, [2] Sensory-Motor Systems Lab, Department of Mechanical and Process Engineering, Institute of Robotics and Intelligent Systems, ETH Zürich, Zurich, Switzerland, [3] Moveo Walks, Inc., Cambridge, MA, United States, [4] Egicon S.R.L., Modena, Italy, [5] Gait Up S.A., Lausanne, Switzerland, [6] SoftRobotics Lab for Human Cooperation and Rehabilitation, Istituto Italiano di Tecnologia, Genoa, Italy, [7] Department of Information Engineering, Research Center "E. Piaggio", Università di Pisa, Pisa, Italy, [8] Bertarelli Foundation Chair in Translational Neuroengineering, Center for Neuroprosthetics and Institute of Bioengineering, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, [9] Director of the Neuro-X Center, Head of Translational Neural Engineering Area, The BioRobotics Institute Scuola Superiore Sant'Anna, Pisa, Italy, [10] Human-Robot Interfaces and Physical Interaction Lab, Istituto Italiano di Tecnologia, Genoa, Italy, [11] Institut für Technische Informatik (ZITI), Heidelberg University, Heidelberg, Germany

The size, weight, and power consumption of soft wearable robots rapidly scale with their number of active degrees of freedom. While various underactuation strategies have been proposed, most of them impose hard constrains on the kinetics and kinematics of the device. Here we propose a paradigm to independently control multiple degrees of freedom using a set of modular components, all tapping power from a single motor. Each module consists of three electromagnetic clutches, controlled to convert a constant unidirectional motion in an arbitrary output trajectory. We detail the design and functioning principle of each module and propose an approach to control the velocity and position of its output. The device is characterized in free space and under loading conditions. Finally, we test the performance of the proposed actuation scheme to drive a soft exosuit for the elbow joint, comparing it with the performance obtained using a traditional DC motor and an unpowered-exosuit condition. The exosuit powered by our novel scheme reduces the biological torque required to move by an average of 46.2%, compared to the unpowered condition, but negatively affects movement smoothness. When compared to a DC motor, using the our paradigm slightly deteriorates performance. Despite the technical limitations of the current design, the method proposed in this paper is a promising way to design more portable wearable robots.

Keywords: soft exosuit, underactuation, assistive robots, unidrive, soft robotics, exoskeletons, PWM control, wearable robotic suit

OPEN ACCESS

Edited by:
Yongping Pan,
National University of Singapore, Singapore

Reviewed by:
Zhao Guo,
Wuhan University, China
Igor Gaponov,
Korea University of Technology and Education, South Korea
Zhenglong Sun,
The Chinese University of Hong Kong, China

****Correspondence:***
Michele Xiloyannis
michele.xiloyannis@hest.ethz.ch

† These authors have contributed equally to this work

Received: *24 February 2019*
Accepted: *27 May 2019*
Published: *18 June 2019*

Citation:
Xiloyannis M, Annese E, Canesi M, Kodiyan A, Bicchi A, Micera S, Ajoudani A and Masia L (2019) Design and Validation of a Modular One-To-Many Actuator for a Soft Wearable Exosuit. Front. Neurorobot. 13:39. doi: 10.3389/fnbot.2019.00039

1. INTRODUCTION

One of the earliest attempts to develop a wearable robotic device to assist human motion, dating back to 1967, failed because of the excessive weight and size of the system (Mosher, 1967). Since then, advancements in material science, power supplies and computing power have fundamentally broadened the boundaries of what we can achieve.

Exoskeletons have been used for a plethora of applications, ranging from performance augmentation in industry (de Looze et al., 2016) to neuro-rehabilitation in medicine

(Louie and Eng, 2016). Despite exciting achievements, there are still substantial technical limitations preventing wearable powered devices from becoming a ubiquitous part of our daily lives. Among others, power requirements and weight of the actuation stage play a key role, confining most of the existing exoskeletons to research laboratories or specialized clinics.

A significant step forwards in this direction has been recently taken by using fabric and polymers to transmit forces and torques to the human body. Soft materials limit the magnitude and accuracy of assistive forces but allow to engineer lighter, less power-demanding, and svelter exoskeletons, resembling our everyday clothes more than the rigid machines portrayed by science-fiction movies (Asbeck et al., 2014).

Although fundamental research is being carried out to design efficient, controllable and robust new actuators (Cappello et al., 2018), most exosuits are powered by traditional electric motors, transferring power to the joints through flexible transmissions (Asbeck et al., 2013). Using one motor to assist each Degrees of Freedom (DoF) of the human body is the most common design choice. This strategy is hardly scalable to complex systems: the human arm alone has at least 7 DoFs and the complexity, size, and weight of a device using such a high number of motors would make it impractical.

A common way to address this problem is to use fewer motors than DoF: underactuation strategies include differential mechanism (In and Cho, 2015), mechanical implementation of kinematic synergies (Catalano et al., 2014; Xiloyannis et al., 2016) and routing of the driving cables along multiple joints (Asbeck et al., 2015). However, these approaches impose hard constrains on the kinetics and/or kinematics of the wearer, allowing only a finite number of predefined moving patterns. This idea is conceptually shown in **Figure 1A**, where multiple DoF are mechanically coupled to be driven by a single motor.

An interesting, yet less investigated, method involves using a set of modules, each one moving a DoFs, all tapping mechanical energy from a single drive. In literature, this paradigm is known as One-To-Many (OTM) (Hunt et al., 2013), Unidrive (Karbasi et al., 2004), or Single-Motor-Driven (SMD) system (Chen and Xie, 1999).

The idea of having a prime mover delivering motion to many subunits is gracefully illustrated in Dante Alighieri's picture of the structure of the universe: he imagined the existence of an outer rotating spheric "sky" that generates energy and transfers it trough its motion to its inner circles, each rotating at a fraction of its speed:

> *"Non è suo moto per altro distinto,*
> *ma li altri son mensurati da questo,*
> *sì come diece da mezzo e da quinto."*[1]

A One-To-Many transmission is the mechanical equivalent of this idea: one electric motor, that we shall call prime mover, transfers power to many modules, each driving a DoF. This paradigm is shown in **Figure 1B**. The prime mover rotates at a constant speed and the trajectory of each DoF is modulated locally by its corresponding module. Such a setup, unlike underactuation mechanisms, allows independent control of each DoF and scales nicely with increasingly complex systems.

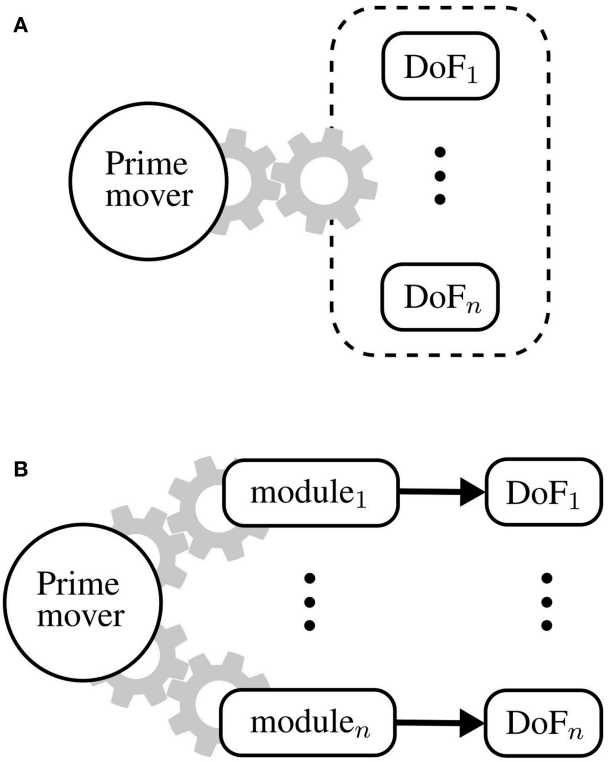


FIGURE 1 | Underactuation mechanisms and One-To-Many (OTM) paradigm. **(A)** Typical underactuated mechanisms rely on some form of mechanical coupling between degrees of freedom (DoF), imposing a hard constraint on the kinematics and/or dynamics of the system. **(B)** The OTM paradigm consists in actuating many DoF, each tapping power from the same drive but controlled independently by a local module.

The challenge when designing an OTM system comes down to engineering a module that is smaller and less power-consuming than a motor but can control the position, velocity, and/or torque of a joint just as well.

Performance-wise, Infinite Variable Transmissions (IVT), allowing to continuously modulate their transmission ratio within a range of positive and negative values, are the ideal candidates for an OTM module. The input velocity remains constant while the output velocity of each DoF can be modulated and even reversed by changing the transmission ratio of each IVT. The recently-published work from Kembaum et al. proposed a novel compact design (Kembaum et al., 2017), but traditionally, the size and mass of IVTs would not justify their use over a simple additional motor. Hunt et al.

Abbreviations: DoF, Degree of freedom; OTP, One-to-many; EM, Electromagnetic; MR, Magneto-rheological; PWM, Pulse width modulation; MJT, Minimum jerk trajectory; EMG, Electromyography; MVC, Maximum voluntary contraction; SPARC, Spectral arc length; RMS, Root mean square; RMSE, Root mean square error.

[1] "Its motion is not measured by another, but all the others are by this, as ten is measured by its half and by its fifth" (Alighieri, 1555).

proposed an OTM system where each module consisted of a linear spring and a clutchable ratchet, the former used to store energy and the latter to selectively release it (Hunt et al., 2013).

The group led by Kermani (Kermani and Alex, 2014; Yadmellat et al., 2014), proposed a mechanism employing magnetorheological (MR) clutches to design a 2 DoF manipulator for safe human-robot interaction. A single electric motor was placed a the base of the robot, providing power to both joints, while three MR clutches were controlled to limit the output torque's magnitude and direction at each joint. The authors showed that this Distributed Active-Semi Active (DASA) actuation paradigm can achieve smooth and accurate tracking of joint positions while benefiting from key advantages of MR clutches such as backdrivability and low impedance.

Finally, a handful of research groups have presented designs based on a similar and powerful principle: the module consists of at least two gears, constantly rotating in opposite directions, and the output is coupled with either one of them using ElectroMagnetic (EM) clutches, or locked using a brake (Li et al., 2003; Karbasi et al., 2004; Yadmellat et al., 2014). The group led by Xie was probably one of the first to propose such arrangement to drive a 9 DoF robotic hand (Chen and Xie, 1999) and a 6 DoF serial manipulator (Li et al., 2011) with a single electric motor.

The encouraging results of OTM systems, applied to drive manipulators and robotic hands, led us to investigate their feasibility in the field of wearable assistive devices, increasingly in need of novel, efficient actuation paradigms. We previously proposed an OTM system, consisting of 2 clutchable modules, to actuate the tendons driving a soft exosuit for the elbow joints (Canesi et al., 2017), shown in **Figure 2**. Each module followed a working principle similar to the one presented in Chen and Xie (1999): this design has the advantage, over the spring-ratchet design in Hunt et al. (2013), of not being limited in the amount of energy that can be stored in each module and it results in a simpler and lighter architecture than the designs employing MR clutches or IVTs.

Our first work was controlled using a heuristically-designed state machine that didn't guarantee stability. Because of this lack of robustness we limited our first assessments to a testbench. Even with a simple controller, however, we demonstrated the feasibility of this approach to achieve independent control of multiple DoF, using a single drive. In this manuscript, we bring the assessment one step further, by including the human in the validation process.

We propose a refined version of our module's design and presents a novel PID-modulated PWM controller to finely adjust the velocity of each DoF independently. We thoroughly characterize the system on a test-bench and then use it as the low-level layer of an admittance-based scheme to control our soft exosuit.

The performance of the new actuation unit is finally compared to that of a traditional DC motor by assessing their impact on kinetics and kinematics of human movement. This human-in-the-loop validation highlights the limitations of our approach and points out avenues for improvement.

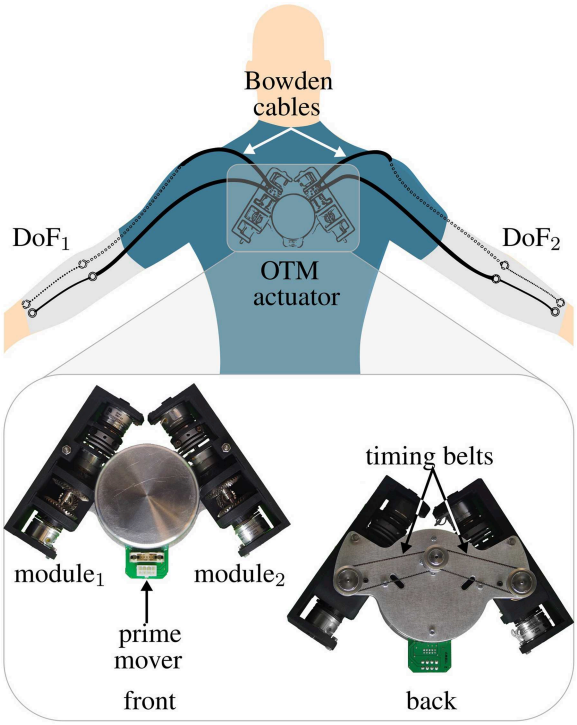


FIGURE 2 | OTM actuator to power two elbow exosuits with one electric motor. This system, that we proposed in Canesi et al. (2017), consists in one prime mover, transmitting power to two clutchable DOF Modules connected to two elbow exosuits via Bowden cables. Each module independently controls the position of its corresponding DoF.

2. OTM DESIGN AND CONTROL

The working principle of the clutchable OTM module is shown in **Figure 3**. The device consists of three EM clutches (SO11, Inertia Dynamics, 5 W) used for coupling the output to either a forward-rotating gear (red), a reverse-rotating gear (blue), or to lock it (black). Depending on which clutch is engaged, the module works in four possible states:

- Free: when all the clutches are disengaged, the output velocity at the pulley is undefined, the output is back-drivable.
- Lock: when the brake is engaged.
- Forward: when the "forward" clutch is engaged, the output velocity equals the input velocity times the reduction ratio between the input and output ports.
- Reverse: when the "reverse" clutch is engaged, the output velocity equals the reverse of the input velocity times the reduction ratio between the input and output ports.

Table 1 summarizes the states of the module, where the engagement of each clutch is represented by a binary variable. Any other pattern of activation is avoided as it would result in mechanical stall of the prime mover.

2.1. Mechanical Design

Figure 4 shows an exploded view of the functional components of the module. A pinion is continuously rotated by the prime

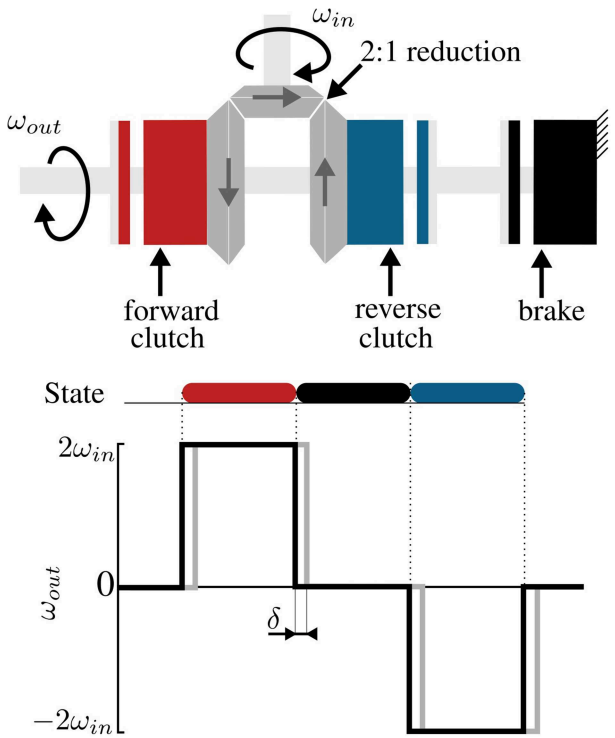


FIGURE 3 | Working principle of a DOF module. ElectroMagnetic (EM) clutches are used to convert a unidirectional input to a bidirectional output by coupling an output shaft with either one of two counter-rotating gears; a brake locks the output in place. Black lines in the plot show the ideal behavior of the clutches, gray lines the actual one, caused by a delay in engagement and disengagement of the armature and rotor.

mover and meshes orthogonally with two bevel gears facing each other. Thanks to this arrangements, the bevel gears rotate in opposite directions. An aluminum link couples each gear to the armature of an EM clutch, whose rotor is rigidly linked to a passing countershaft. When power is applied to either one of the EM clutches, the armature is coupled to the rotor, thus effectively linking the countershaft to either one of the gears. A third, identical clutch, acts as a brake, locking the countershaft in a static position by coupling it with the frame. A flexible coupling joins the countershaft to a pulley (the output of the module); the pulley houses two cables wrapped in opposite directions and is used to transmit motion to the exosuit through flexible Bowden cables. **Figure 5** shows a photograph of an assembled module, enclosed in a 3D printed casing.

2.2. Control

Figure 3 shows that the module has three working states: forward, reverse, and brake. In Canesi et al. (2017), we proposed a heuristic control approach that consisted in a feedback state machine to keep the measured trajectory as close as possible to a reference profile. This strategy, although being simple, presented fundamental limitations in accuracy and stability.

TABLE 1 | Possible states of the module.

State	Forward clutch	Reverse clutch	Brake
Free	OFF	OFF	OFF
Forward	ON	OFF	OFF
Reverse	OFF	ON	OFF
Lock	OFF	OFF	ON

In this work, we propose a more principled paradigm to control this discrete system, based on the work of Karbasi et al. (2004). The controller is based on Pulse Width Modulation (PWM) of the three discrete states of the OTM, regulated by a feedback PI controller, to continuously adjust the average velocity of the module's output. The non-linear PI controller translates the difference between the desired and actual velocity of the module to a rectangular pulse signal, having value -1 (reverse clutch), 0 (brake), or 1 (forward clutch), whose duration in time is dependent on the magnitude of the error.

In the next sections we describe how a tracking error is converted to a discrete control signal for the clutches for an ideal PWM-regulated system.

Using a state-space representation, we can describe a non-linear PWM-controlled system with the following equations:

$$\begin{cases} \frac{dx}{dt} = u(t)\omega_{in} \\ y(t) = x(t) \\ e(t) = r(t) - y(t) \\ u(t) = PWM(e(t_k)) \end{cases}$$

where

- $u(t)$ is the control signal for the clutches, bonded to have values -1, 0, or 1.
- $x(t)$, the state variable; in our case, the angular position of the module's pulley, θ.
- $e(t)$ is the error between the reference $r(t)$ and the output $y(t)$.
- ω_{in} is the fixed input velocity of the module.
- t_k represents the instant of initiation of the k-th PWM period (**Figure 6**, top).
- $PWM(e(t_k))$ is the PWM control operator, defined as:

$$PWM = \begin{cases} \text{sgn}(e(t_k)) & \text{for } t_k \leq t \leq t_k + \alpha \\ 0 & \text{elsewhere} \end{cases}$$

where we have used $\alpha = \tau(e(t_k))T_{PWM}$, for simplifying the notation, with $\tau(e(t_k))$ being the duty ratio function and T_{PWM} the period of the PWM signal.

The PWM operator basically sets the control signal to 1, 0, or -1 for a period of time defined by $\tau(e(t_k))$.

This duty ratio function, $\tau(e(t_k))$, needs to output a value between 0 and 1, representing the ratio for which the control signal will be equal to the sign of the error. Sira-Ramirez

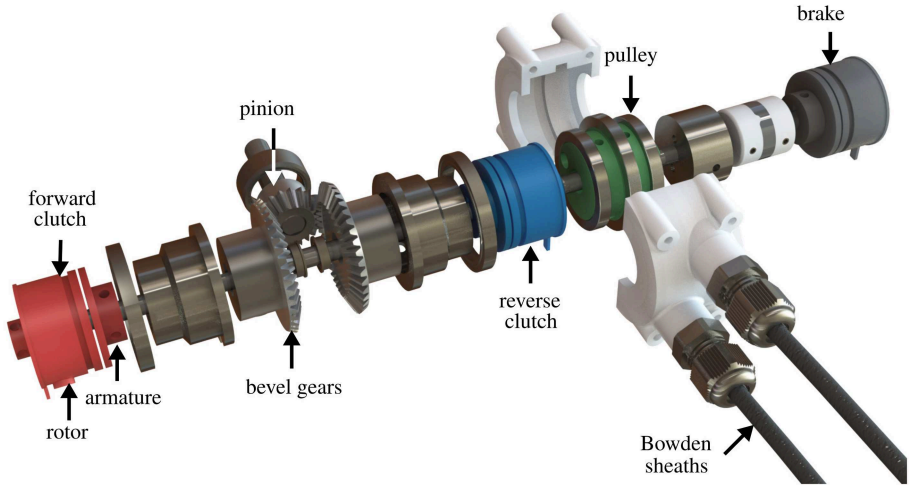


FIGURE 4 | Mechanical design of a One-To-Many (OTM) module. Each module receives power from the prime mover through a pinion, connected to two, opposite-facing, bevel gears that rotate in opposite directions. Each bevel gear can be coupled to the passing countershaft by engaging an ElectroMagnetic (EM) clutch, selectively rotating it in either direction. A third EM clutch can connect the countershaft to the frame, acting as a break. The shaft drives a pulley housing the antagonistic Bowden cables that actuate the soft exosuit.

suggests the following as a good choice for non-linear systems (Siraramirez, 1989):

$$\tau(e(t_k)) = \begin{cases} \beta|e(t_k)| & \text{for } |e(t_k)| \leq 1/\beta \\ 1 & \text{for } |e(t_k)| > 1/\beta, \end{cases} \quad (1)$$

that makes the duty ratio proportional to the magnitude of the error if the error is relatively small (smaller than a threshold $1/\beta$) and saturates it to 1 if the magnitude of the error is larger than the threshold.

The big assumption of this controller is that the clutches behave like ideal switches, i.e., the output velocity of the module instantaneously equals a multiple of the input velocity when the clutch is engaged (**Figure 3**, black). In practice, this is not the case: the clutches have an intrinsic delay when engaging (shown in gray in **Figure 3**) and the velocity of the output decays in an exponential-like fashion when the clutch is disengaged (Karbasi et al., 2004).

Skoog and Blankenship (1970) and colleagues have shown that such phenomenon can be mitigated with the addition of an integral feedback term. Adding this to Equation (1) makes the output of the duty ratio function, in the region $|e(t_k)| \leq 1/\beta$, proportional not only on the magnitude of the error but also to its history, effectively behaving like a non-linear PI controller.

2.3. Performance

We characterized the performance of the DOF module for varying control parameters and tested its limits in speed and torque transmission.

Specifically, we evaluated the ability of the controller to modulate the output velocity for varying PMW periods T_{PWM}, we tested its ramp response and bandwidth for varying input velocities of the prime mover and, lastly, its maximum load rating. The DOF module was equipped with an incremental encoder (AMS, AS5047P, 1,000 pulses/rev), monitoring the position of the cable pulley.

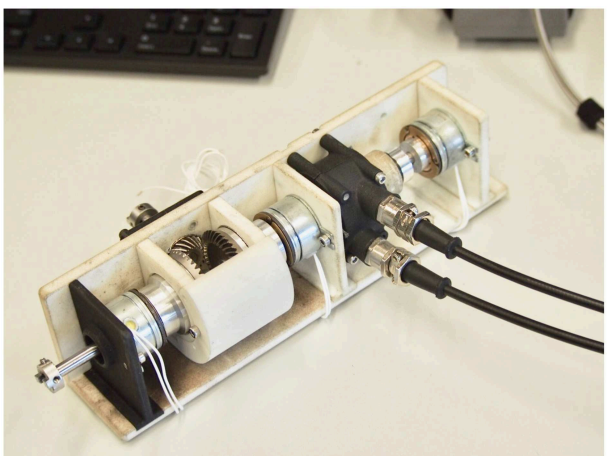


FIGURE 5 | A single DoF assembled Module.

2.3.1. Velocity Modulation

Figure 6B shows the ability of the PWM controller to adjust the velocity of the module as its duty cycle changes. The plot shows the normalized velocity (ω_{out}/ω_{in}), for varying duty ratio of the PWM signal, where the duty ratio expresses the percentage of time in T_{PWM}, where the control signal is non-zero, i.e., 1 or -1. Lines of different color represent different frequencies of the PWM signal.

The T_{PWM} should be set as small as possible, to ensure stability, but bigger than the time delay δ, necessary for the clutches to physically engage once powered. Indeed, very high

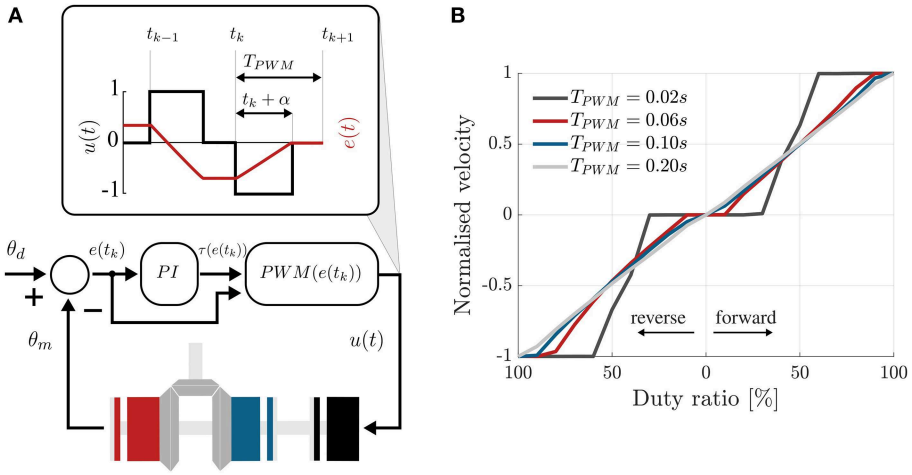


FIGURE 6 | PI-regulated PWM control of the clutchable module. **(A)** The error between the desired and measured position of the countershaft is passed through a traditional PI controller that computes the duty cycle of the control signal, according to Equation (2), in cascade with the PWM function defined in Equation (1). This sets the control signal to 1 or −1 for a fraction of the PWM period that depends on the magnitude of the error. 1 engages the forward clutch, −1 the reverse clutch and 0 the brake. **(B)** Choosing the period of the PWM signal. Velocity of the output shaft, normalized by the input speed, vs. the duty ratio of the PWM signal controlling the forward and reverse clutches. Different colors show different PWM frequencies. For high frequency, low and high duty ratios do not affect the output velocity.

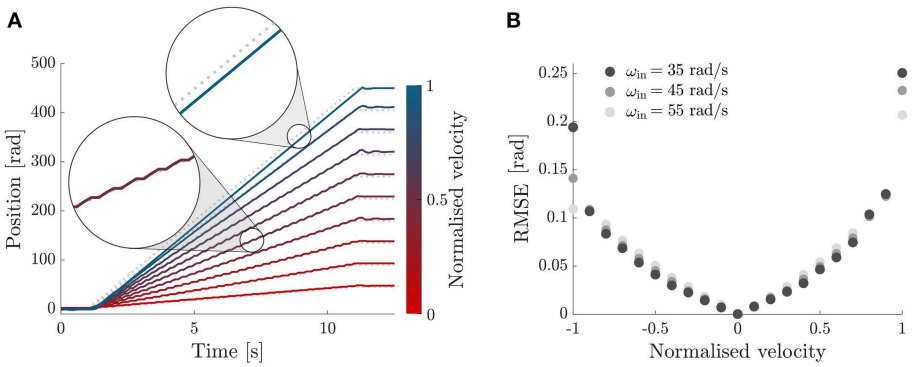


FIGURE 7 | Ramp response. **(A)** Tracking performance of the DOF module with the PI-regulated PWM control. Ramp response for velocities between 10 and 100% of the input velocity. **(B)** RMSE between the desired and measured positions of the module's output, for desired output velocities between -ω_{in} and ω_{in} (x-axis) and different input velocities (grayscale). The behavior of the controller is symmetric, with increasing error for higher absolute velocities.

frequencies of the PWM do not result in a higher ability to modulate the output velocity. This is caused by the intrinsic delay, δ, necessary for the rotor and the armature of the EM clutches to engage upon the application of power. We chose $T_{PWM} = 0.20$ s, being the highest PWM frequency showing a near-linear trend.

2.3.2. Ramp Response

Figure 7A shows the tracking ability of our PID-modulated PWM controller to track a ramp position profile of increasing slope, between 10% and 100% of the input velocity, in steps of 10%. The experiment was repeated for three fixed input velocities and showed a consistent behavior, with the tracking ability of the controller deteriorating as the normalized speed approached one.

2.3.3. Bandwidth

Figure 8 shows the tracking of sinusoidal trajectories and Bode plot of the module with the controller proposed in section 2.2. The module was commanded to follow a sinusoidal trajectory of the form:

$$\theta^d_{out}(t) = A \sin(2\pi f_0 t) \tag{2}$$

with A corresponding to the amplitude required for a flexion/extension motion of the elbow of 90 deg and a frequency f_0 evaluated between 0.01 and 1.51 Hz, in incremental steps of 0.02 Hz. For each frequency we collected data for 20 s at a sampling rate of 1 kHz; the first second was discarded for further analysis to evaluate the performance of the system at steady-state. This procedure was repeated for three different input velocities of the prime mover 35, 45, and 55 rad/s.

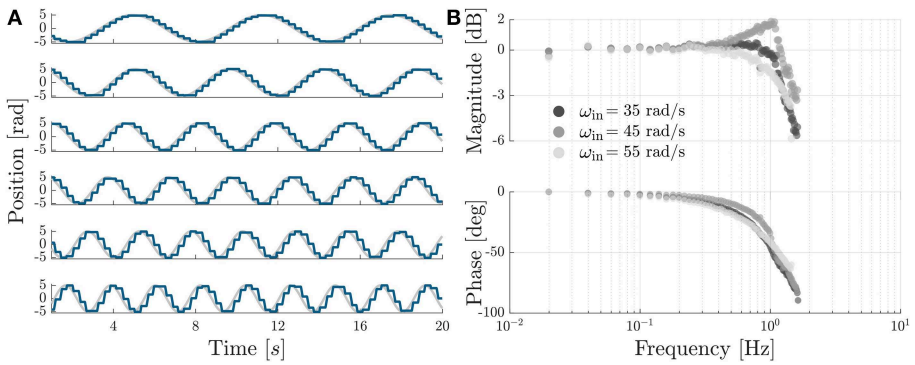


FIGURE 8 | Sinusoidal tracking and Bode plot. **(A)** Sinusoidal tracking accuracy of the OTM module with the PI-regulated PWM control described in section 2.2, shown for six equally-spaced frequencies, between 0.15 and 0.4 Hz. **(B)** Bode plot of the transfer function between desired and measured position of the module's output, shown for three different velocities of the prime mover. The system has a cut-off frequency of 1.26, 1.51, and 1.30 Hz, for a speed of the prime mover of 35, 45, and 55 rad/s, respectively.

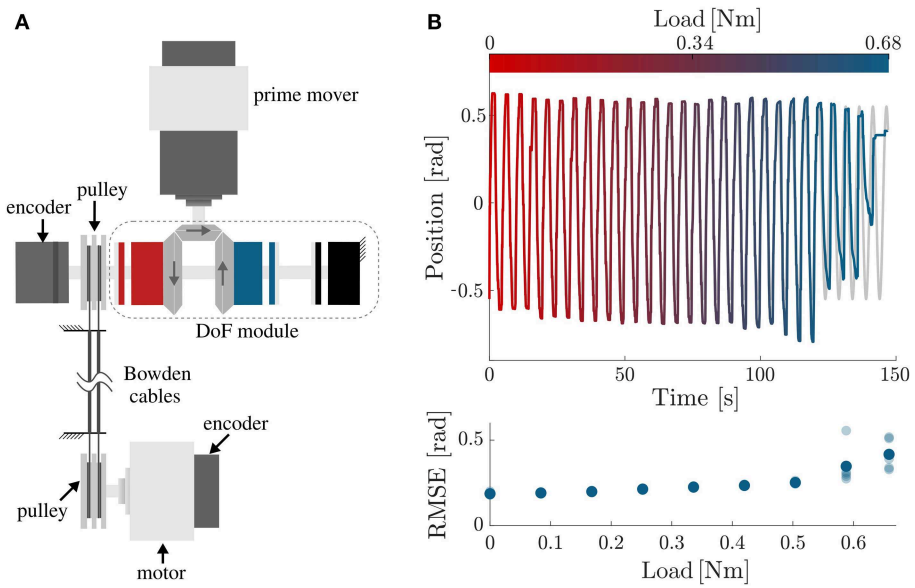


FIGURE 9 | Performance testing under load. **(A)** Testbench used for assessing the performance of the DoF while moving a load. The first motor, i.e., the prime mover, drives the DoF module; the second motor is used to simulate a load on the output. Mechanical power is transmitted from the DoF module to the load via Bowden cables. **(B)** Tracking accuracy of the OTM module, controlled with the PI-regulated PWM controller, under increasing load. The top plot shows the desired (gray) and measured (colored) position of the module's output as the load increases between 0 and 0.68 N m. The clutches start to slip just before 0.6 N m, as quantified by the increase in RMSE shown in the bottom plot.

For each frequency we performed an analysis in the Fourier domain to evaluate the amplitude ratio and the phase lag between the desired $s(t)$ and the measured θ_{out} signal. The nth complex coefficient of the Fourier series has the form:

$$C_n = \frac{f_0}{N} \int_0^{\frac{N}{f_0}} s(t) e^{-j2\pi n f_0 t} dt \tag{3}$$

where f_0 is the sampling rate and N is the number of cycles whereby the signal is repeated. For each driving frequency we evaluated the response as the ratio between the coefficients of the fundamental frequency of the measured and desired signals:

$$H(f_0) = \frac{C_1^{\text{measured}}}{C_1^{\text{desired}}}. \tag{4}$$

Figure 8B shows a Bode plot of the system, representing the transfer function between the measured and desired position of the module. The device and controller show a cut-off frequency of 1.26, 1.51, and 1.30 Hz, for a speed of the prime mover of 35, 45, and 55 rad/s, respectively. We chose a velocity of 45 rad/s for all following tests.

2.3.4. Loaded Behavior

Finally, we tested the tracking accuracy of the module in the presence of a load. For this testing, we attached the Bowden cables on a second pulley, driven by a DC motor (Maxon EC-i 40, 70 W, 3.7:1 reduction ratio), to apply a load on the end-effector (shown in **Figure 9A**).

The module was set to follow a sinusoidal trajectory while the load linearly increased between 0 and 0.68 N m (maximum rated torque for the EM clutches). The test was repeated 5 times. Indeed, the clutches started slipping, causing a steep deterioration in tracking accuracy, around 0.6 N m (**Figure 9B**). This is lower than their rated value, probably due to friction losses in the transmission.

Table 2 summarizes the technical characteristics of our OTM module assembly shown in **Figures 4**, **5**, where the maximum rated torque and velocity have been mapped to the elbow joint.

TABLE 2 | Technical specifications.

Characteristics	Values		
Module			
Weight [Kg]	0.534		
Dimensions [cm]	15 × 6 × 5		
	35 rad/s	45 rad/s	55 rad/s
Bandwidth [Hz]	1.26	1.51	1.30
Assembly			
DOF	2		
Weight [Kg]	2.2		
Dimensions [cm]	26 × 18 × 12		
Motor power [W]	90		
Max torque[a] [Nm]	3.4		
Max velocity[a] [deg/s]	424		

[a] *At the elbow joint.*

3. TESTING ON HUMAN MOVEMENTS

To test the feasibility of the OTM mechanism for assistive purposes, we compared the performance of an exosuit driven by the OTM mechanism to that of the exosuit driven by a traditional DC motor. The performance of the device was assessed through its effect on the the kinematics and muscular activity of its wearer.

3.1. Exosuit Design and Control

The exosuit is shown in **Figure 10**. The device consists of a frame of soft material that wraps around the arm and forearm and transmits torque to the elbow through artificial tendons. A pair of Bowden cables transmits power from the actuation unit to the joint (**Supplementary Material**).

The suit is equipped with a force sensor (Futek, LCM300), secured on the distal anchor point, that measures the tension in the flexing tendon, and an absolute encoder (AMS, AS5047P, 1,000 pulses/rev), mounted on a 3D-printed joint between the arm and forearm straps, that monitors the angular position of the joint.

The device provides assistance to its wearer through an admittance-based controller, detailed in Chiaradia et al. (2018). The control paradigm is designed to have the dual purpose of compensating for gravitational forces acting on the forearm and allowing the exosuit to move in concert with its wearer.

A schematic diagram of the controller is shown in **Figure 10B**. It comprises an outer torque loop and an inner position loop. The former is responsible for tracking the position-dependent torque profile at the elbow, equal and opposite to gravity. A PID-admittance converts the error between the desired and assistive torque at the elbow to a position reference for the actuation stage.

The torque acting on the elbow joint as a result of gravity is estimated using a simple single-joint model and assuming that the arm is adducted on the side of the trunk:

$$\tau_g = mgl_c \sin\theta_e, \tag{5}$$

with m being the combined mass of the forearm and hand, l_c the distance of the center of gravity of the forearm and hand from the center of rotation of the elbow joint, g the acceleration of gravity and θ_e the elbow angle, assumed to be zero in the fully-extended configuration.

The assistive torque is estimated by multiplying the tension measured by the load cell, f, by its moment arm $P(\theta_e)$ (refer to Xiloyannis et al., 2017 for a full formulation):

$$\tau_{exo} = P(\theta_e)f. \tag{6}$$

Using the notation shown in **Figure 10B**, the difference between τ_g and τ_{exo} is converted to a reference position for the actuation stage, by a specified admittance. The admittance assumes the form of a PID controller (Yu et al., 2011):

$$Y(s) = \frac{\theta_e}{\tau_g - \tau_{exo}} = P + \frac{I}{s} + Ds, \tag{7}$$

with the P, I, and D constants governing the characteristics of the relation between the interaction force and the exosuit's kinematics. An additional positive feedback term, proportional to the speed of the elbow joint, increases the sensitivity of the device to its wearer's movements.

3.2. Experiments and Data Analysis

The aim of this experiment was to compare the effect on human movements of the exosuit driven by an OTM module to that of the exosuit driven by a traditional DC motor. We included a third condition, consisting in unpowered movements, to have a physiological baseline. In all three conditions, we evaluated smoothness and accuracy of movement, biological torque and muscular activation patterns of a healthy subject performing controlled movements of the elbow. The experimental procedure was similar to the one we used in Xiloyannis et al. (2019) to quantify the effects of a soft wearable exosuit on movements of the upper limbs.

The testing was done on one male subject presenting no evidence or known history of skeletal or neurological diseases, and exhibiting intact joint range of motion and muscle strength.

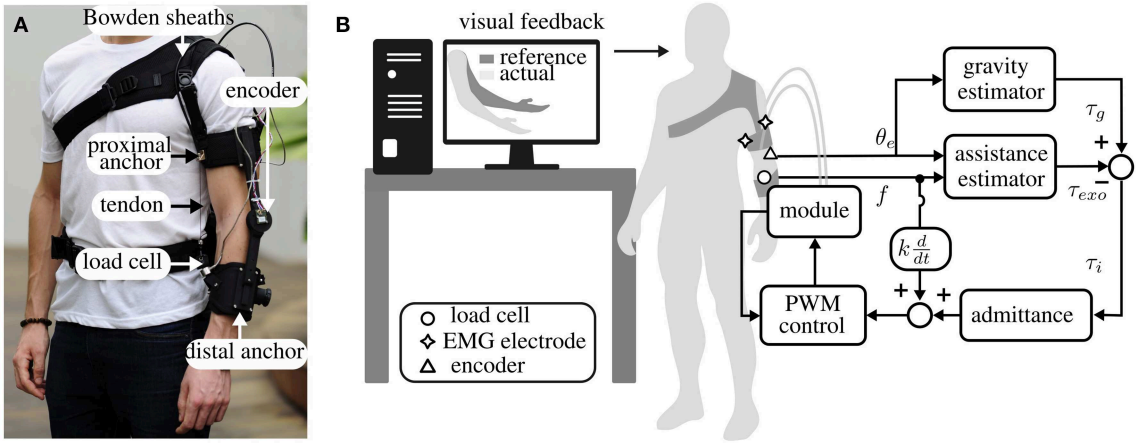


FIGURE 10 | Design and control of the exosuit for the elbow, actuated by a DoF module. **(A)** The suit is driven by Bowden cables that pull together two anchor points on the arm and forearm, generating a torque on the elbow joint. A load cell senses the tension on the cables and an encoder measures the join position. **(B)** Experimental setup and high-level control of the suit. The participant was asked to follow the movement of an avatar on a screen while the suit followed the subject and provided a torque on the elbow equal and opposite to gravity. This was achieved with an admittance controller, using the PWM controller described in section 2.2 as an inner position loop.

At the beginning of each experimental session the participant was informed of the procedure and he signed an informed consent. The procedures, in agreement with the Declaration of Helsinki, was approved by the Institutional Review Board at Nanyang Technological University.

The participant had to follow a reference movement performed by a dummy character on a screen. The position of his own elbow was displayed as a superimposed replica of the reference one to provide visual feedback. In the unpowered condition, the exosuit's tendons were unhooked from the distal anchor point and the motor's power source was turned off.

The reference motion consisted of series of 10 Minimum Jerk Trajectories (MJT), known to correspond well to the movements of healthy subjects (Flash and Hogan, 1985), with amplitude of 80 deg and a peak velocity of 60 deg/s, corresponding to approximally 50% of the average speed in activities of daily living (Buckley et al., 1996).

Raw data from the suit's absolute encoder and load cell was low-pass filtered (second order Butterworth filter, 10 Hz cut-off frequency) and segmented to isolate the 10 movements comprising each condition. The accuracy of movement was quantified by evaluating the Root Mean Square Error (RMSE) between the measured and reference trajectory.

To quantify kinematic smoothness, we used the SPectral ARC length (SPARC) index proposed in Balasubramanian et al. (2015). The SPARC index was estimated on the norm of the elbow's speed.

The measured force on the flexing tendon was mapped to a torque on the joint using Equation (6), this was used as an estimate of the assistive moment delivered by the exosuit, τ_{exo}. The total torque, τ, required to perform the movement was derived from the inverse dynamics of the human elbow, represented as a simple pendulum using a second order model. The difference between the total and assistive torque,

$$\tau_{bio} = \tau - \tau_{exo}, \tag{8}$$

was used to estimate the biological torque exerted by the subject to perform the movement or hold the position.

The output EMG signal of the Delsys Trigno system was processed to extract its linear average envelope using the procedure suggested in Clancy et al. (2002). The Root Mean Square (RMS) of the processed EMG signal was used as an index of the level of activation of a muscle.

3.3. Results

Figure 11 compares the characteristics of the elbow's trajectory for the three tested conditions. In **Figure 11A**, the zoomed-in area shows the overlapping trajectories for the unpowered (gray), powered by an OTM module (red), and powered by a DC (blue) cases: the second condition clearly being more jerky than movements with the DC motor. This is confirmed by a quantitative analysis of the smoothness of movement, evaluated through the SPARC index and shown in **Figure 11B**.

Figure 11C shows the mean and standard error of the mean of the root mean square error between the reference MJT trajectory and the measured one, in the three conditions. When assisted by the exosuit, whether using the OTM module or a traditional DC motor, movements are less accurate. A smaller difference in mean values exists between the OTM and DC motor cases.

An analysis of the forces transmitted by the exosuit during movement gives further insight on the performance of the OTM module. **Figures 11D–F** shows the profile and average values of the biological torque, calculated using Equation (8). In the unpowered case, the entire moment required for movement is exerted by the subject; in the OTM case, the positive torque required for movement is reduces, but the exosuit

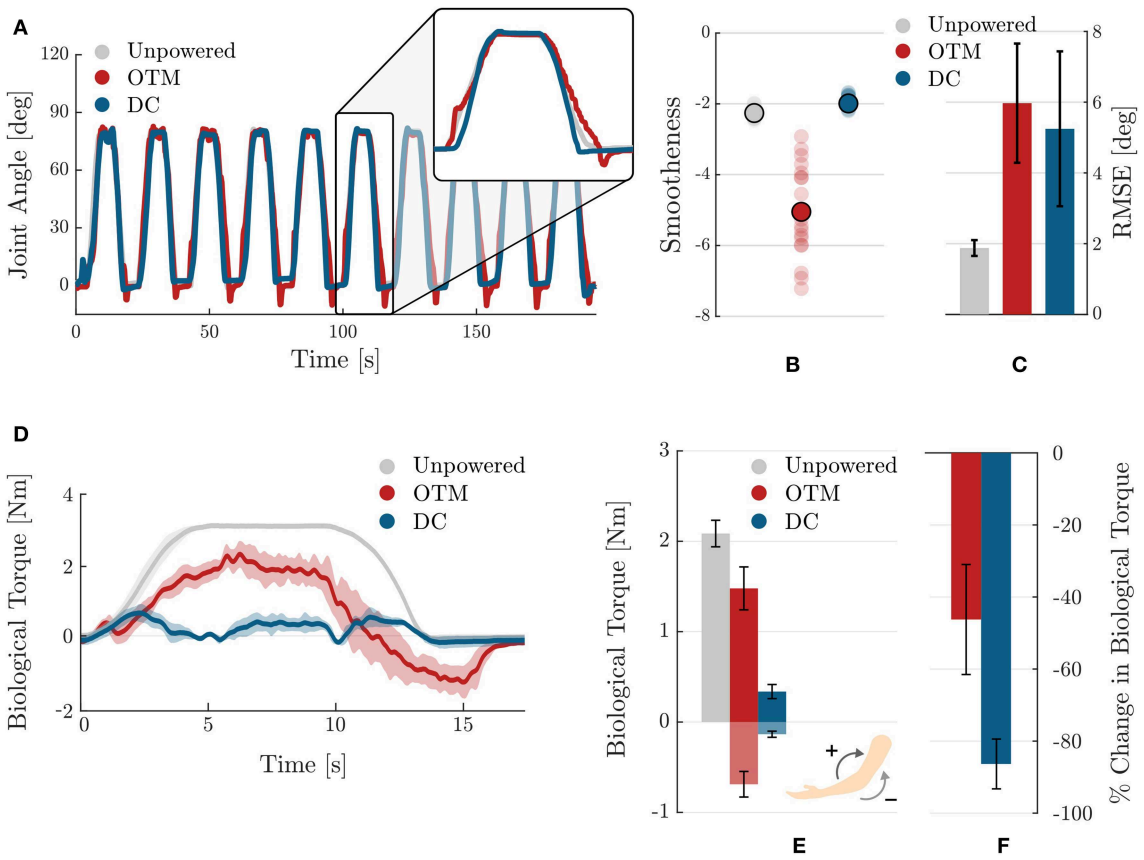


FIGURE 11 | Comparison of the effects of the OTM module and DC motor actuation strategies on the kinetics and kinematics of human movement. **(A)** Trajectories of the elbow when wearing the exosuit in the three tested conditions: unpowered (gray), powered by the OTM module (red), and powered by a traditional DC motor (blue). **(B)** Smoothness of movements as measured by the SPARC coefficient in the three tested conditions. Translucent dots are individual values over the 20 sub-movements of the task, while opaque dots show their mean. **(C)** Accuracy in tracking the reference trajectory, measured by calculating the RMSE, for the three conditions. Movements assisted by the OTM module have similar accuracy to movements assisted by a traditional DC motor. **(D)** Average (solid) and standard deviation across repetitions of the biological torque profiles for the unpowered (gray) condition, for the powered by an OTM module (red) and powered by a traditional DC motor (blue). **(E)** Mean and standard error of the biological torque, averaged across repetitions, for the three tested conditions. **(F)** Change in the overall biological torque required for lifting the arm when assisted by the exosuit. Values are in percentage change of the unpowered case.

introduces a substantial negative component in the descending phase (**Figure 11D**).

Figure 11E shows the mean and standard error of the mean, for the positive and negative biological torques exerted in the three conditions. Wearing the exosuit introduces a negative component, required to initiate the downwards motion, both when powered by the DC motor and OTM module. In the latter case, however, the magnitude of negative torque is substantially higher. Overall, the OTM reduces the biological torque at the joint of 46.2%, compared to the unpowered condition, while the DC motor achieves an average −86.3% change.

Finally, **Figure 12** analyses the activity of the two major antagonistic muscles involved in delivering power to the elbow joint, namely the biceps brachii and the long head of the triceps brachii. An average and standard deviation of the profile of activation of these muscles is shown in **Figure 12A**, for the three conditions.

When the suit is unpowered, the biceps brachii reaches a peak activation of 7% of his MVC, while the triceps muscles is relaxed for most of the duration of the movement. When the exosuit, powered by a traditional DC motor, assists its wearer, the activation of the biceps muscles is substantially reduced, with a peak activation during the transient lifting phase. When the device is powered by the OTM module, the agonist muscle's activity is lower than the unpowered condition but presents a slightly different activation pattern, while the triceps shows a peak during the descending phase.

Figure 12B shows the average over repetitions and time, of the RMS of the activity of both muscles for the OTM and DC conditions, expressed in percentage change of the unpowered case. Using a traditional DC motor to actuate the exosuit reduces the activation of the biceps muscle by 57.1% and increases that of the triceps muscle of 8.84%. When the OTM powered the suit, the activation of the biceps muscle was reduced by an average 30% while its antagonist increased by 74%.

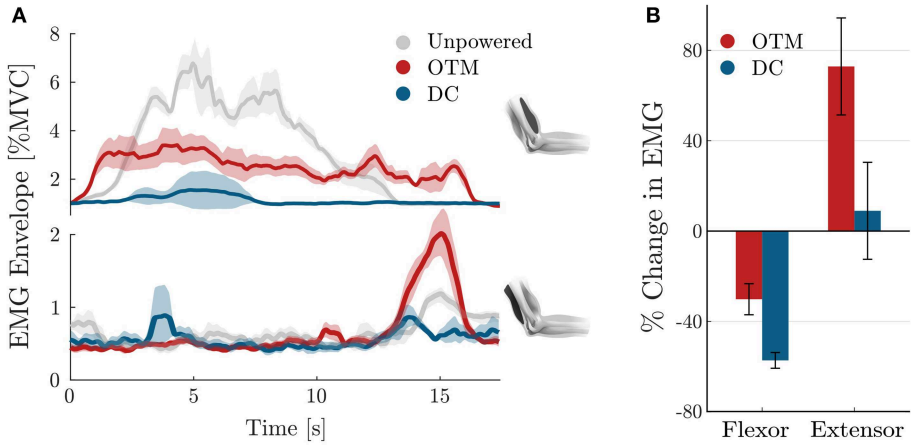


FIGURE 12 | Muscular activation of the biceps brachii and long head of the triceps brachii. **(A)** Average (solid) and standard deviation (shaded) across repetitions of the muscular activation profile during one lifting repetition of the forearm, shown for the unpowered condition (gray), for the exosuit powered by the OTM condition (red), and the exosuit powered by a traditional DC motor (blue). **(B)** Percentage change in the RMS of the muscular activity of the antagonistic muscles, averaged over repetitions.

4. DISCUSSION

Power consumption, weight and size of wearable robots have a fundamental impact on their performance. Soft exosuits, adopting clothing-like materials instead of rigid frames, represent a significant step toward making assistive wearable devices ubiquitous in the near future. Yet, because of the unparalleled complexity of our bodies' biomechanics, we are bounded to either use a high number of motors or underactuation solutions that constrain the natural kinematics of human movements.

The implementation of an actuator that can exploit the torque generated by a single motor to drive multiple DoF would cut down cost, weight and size of the actuation module, reducing the complexity of the control and increasing the overall autonomy of device.

In this study we presented the design and control, and proposed a human-in-the-loop validation of a clutchable modular unit to implement the OTM paradigm for a soft exosuit. The following subsections discuss the implications of our findings first by looking purely at the technical performance of the device and then evaluating its effects on human biomechanics.

4.1. On the Performance of the PWM Controller

Figure 6B shows that as the frequency of the PWM controller increases, the output velocity stays null for low duty cycles and saturates to the input velocity earlier, effectively reducing the range of duty cycle where one can modulate the output velocity. This effect is caused by the delay δ, necessary for the armature and rotor of the EM clutches to engage/disengage upon the application of power. Our results are consistent with the observations of Karbasi et al. (2004), who found a similar behavior with wrap-spring clutches: when increasing the PWM frequency, the period T_{PWM} decreases and, for small duty-ratios, Δt becomes smaller than δ, effectively causing the clutch never to engage. For high duty ratios, on the other hand, the disengagement time, $T_{PWM} - \Delta t$ is smaller than the minimum time required for the clutch to disengage, causing a permanent engagement.

The bandwidth of each module, with a PI-regulated PWM controller, varied for input velocities of 35, 45, and 55 rad/ s, with the maximum 1.51 Hz obtained for 45 rad/ s. This result echoes our previous findings, where Canesi et al. (2017) tested the same device for higher input velocity and found a saturation effect of its bandwidth for higher velocities of the prime mover. This suggests that the limiting dynamic at high frequencies is not given by the rotating speed of the prime mover but by the engaging dynamics of the clutches (Karbasi et al., 2004).

Assuming a sinusoidal motion of the elbow with a peak to peak movement of 90 deg, 1.51 Hz corresponds to a maximum velocity of the joint of 426 deg/s, which is sufficient for everyday tasks such as drinking from a glass (269 deg/s), eating with a spoon (126 deg/s), and pouring from a bottle (92 deg/s), but not for more demanding working tasks such as hammering (842 deg/s) (Buckley et al., 1996).

The maximum load that each module can handle is bounded by the maximum rated torque of the EM clutches. This can be mapped, using Equation (6), to an estimated average torque at the elbow joint of 3.4 N m, which is just above the torque required to keep the average human male forearm in a static posture of 90 deg.

It is worth comparing the performance achieved here with EM clutches with the results reported in Yadmellat et al. (2014), using Magneto-Rheological ones. Yadmellat and colleagues present custom-built clutches, each able to transmit a maximum torque of 15 N m and shown to accurately track a 2 Hz sinusoidal trajectory. Their clutches weigh 2.2 kg each, with a volume of over 650 cm^2 but, unlike EM cluthces, allow to achieve a smooth and

continuous regulation of the output torque, solving the problem of jerky movements typical of discrete systems.

4.2. On the Effect on Human Movements

It is increasingly important to evaluate novel wearable assistive technologies based on their effect on the end-user. We believe that this human-in-the-loop validation approach can provide important insights on the limitations and advantages of the innovation, allowing a data-driven design process.

In this work, we compared the performance of of the OTM module with that of a traditional DC motor by looking at its effect on the kinematics and physiology of healthy movements.

Overall, when driven by the OTM module, the assistive device provided fewer benefits in terms of muscular activation and had a more marked effect on movement kinematics than when driven by a traditional DC motor. Specifically, wearing the exosuit powered by the OTM module resulted in more jerky movements, as measured by the SPARC index. This is probably a direct consequence of the discrete nature of the mechanism, driving the assisted joint with small step-like motions along the desired trajectory (see **Figure 7** or **Figure 8**). These fast-changing dynamics are filtered by the compliant transmission between the pulley and the joint but still affect the wearer's movement. We believe that switching to particle-based megneto-rheological clutches, that allow a continuous modulation of the torque being transferred between their input and output shafts, would attenuate or even solve this problem. Previous studies support this proposition (Yadmellat et al., 2014).

Movements assisted by the OTM module resulted in an overall 46.2% reduction in biological torque, vs. an 86.3% reduction achieved with a traditional DC motor, when compared to the unpowered case. This poorer performance of the OTM module is partly caused by: (1) the upper torque limit of the EM clutches, restraining the magnitude of assistance that the actuation can provide; (2) the lower bandwidth of the module, that does not allow it to move fast enough to accommodate the user's intention. This last point is clearly visible in **Figure 11D**, in red, where the biological torque shows a significant negative peak during the descending phase of the joint (15 s): the user had to push down against the exosuit to extend the elbow.

A corollary of the increase in negative torque, when using the OTM actuation module, is an increase in the activation of the long head of the triceps brachii. **Figure 12A** shows a peak in the activation of the extensor muscle (bottom plot, in red), of up to 2% of the MVC, during the descending phase of movement (15 s). This peak is present also in the other two conditions, but with much smaller amplitude. Overall, the activity of the triceps brachii increased notably when wearing the exosuit powered by the OTM module, while the biceps brachii was reduced by 30%, when compared to the unpowered condition.

It is useful to compare the technical characteristics of the proposed OTM system with the traditional approach of using one motor per elbow. In the latter case one would require at least a 70 W motor with an appropriate reduction stage and a motor controller per DoF, which would result in no more than 600 g and 560 cm^3 per DoF (e.g., Maxon EC-i 40, 70 W, 2-stage planetary gearhead and an ESCON 50/5 module motor driver). **Figure 13** shows how the volume and weight of the actuation stage scale with increasing DoF, for a typical one-to-one paradigm and for the OTM strategy proposed here. With the current hardware, the actuation stage would weigh less only for 4 or more DoF. It is worth highlighting, however, that the OTM system only requires one motor controller overall and three simple relay switches per DoF, thus reducing the cost and idle power consumption of the electronics compared to the traditional approach. There is, moreover, ample room for improvement of the size and weight of the OTM modules.

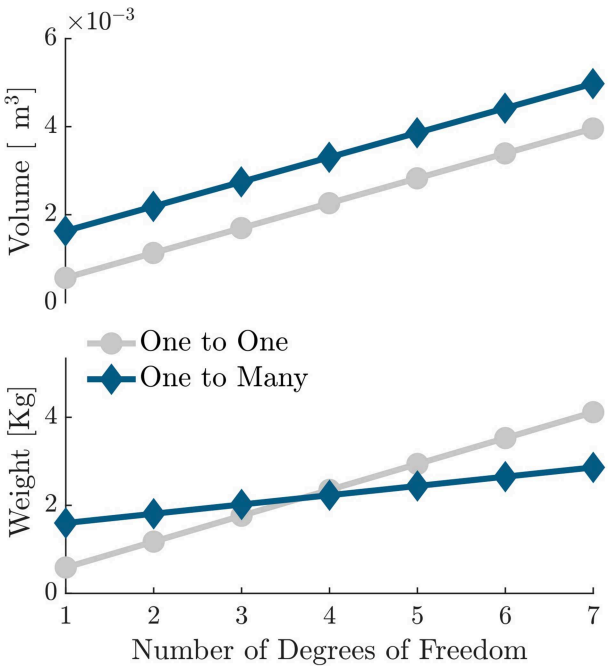


FIGURE 13 | Scalability of the OTM design proposed herein, compared to a traditional one-to-one approach. Volume and weight of the actuation strategy using an OTM paradigm, compared to a setup having similar torque and velocity rating but using one electric motor per DoF.

In the present work we explored the potential of an OTM module and compared its performance with a direct drive DC motor, using a wearable device as a test bench for human testing. The DC motor showed higher benefits in assisting human kinematics thanks to its higher bandwidth and smoother motion. However, the OTM paradigm lends itself well to scalable applications, that require independent control of multiple DoFs. This is a common problem in wearable technology, where the complexity of human biomechanics results in an increase in weight, power consumption and control intricacy of the device. If on one side the OTM startegy showed limitations in trajectory tracking, on the other, multiple OTM modules can be driven using a single drive and their control strategy might be optimized reducing the task manifold. In our experiment we used a wearable device for upper limb, and it is known that replicating the dexterity of a human arm is a complex task; we believe that with further improvement on the hardware design,

the OTM paradigm has the potential to allow independent control of multiple DoF in a power- and size-effective manner. A different result, for example, might be obtained for well-defined tasks with periodic velocity profiles such as walking, where joint coordination can be achieved by means of synergistic and intermittent assistance on each leg (Asbeck et al., 2015).

DATA AVAILABILITY

The datasets generated and analyzed for this study are available upon reasonable request.

ETHICS STATEMENT

This study was carried out in accordance with the recommendations of the Institutional Review Board at Nanyang Technological University with written informed consent from all subjects. All subjects gave written informed consent in accordance with the Declaration of Helsinki. The protocol was approved by the Institutional Review Board at Nanyang Technological University.

AUTHOR CONTRIBUTIONS

LM conceived the mechanism. MC, MX, EA, and LM designed the OTM module. AK, EA, MX, and LM developed the controller. EA, MX, AK, MC, and LM designed and performed the experiments. AB, AA, and SM helped drafting the paper. All authors analyzed and interpreted the data. All authors provided critical feedback on the manuscript. All authors read and approved the final manuscript.

FUNDING

This work was supported by the Rehabilitation Research Institute of Singapore (RRIS) (Grant #: M4062142).

ACKNOWLEDGMENTS

We acknowledge financial support by Deutsche Forschungsgemeinschaft within the funding programme Open Access Publishing, by the Baden-Württemberg Ministry of Science, Research and the Arts and by Ruprecht-Karls-Universität Heidelberg.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2019.00039/full#supplementary-material

REFERENCES

Alighieri, D. (1555). "Canto XXVII," in *Divina Commed. - Parad.* (Venice).

Asbeck, A. T., De Rossi, S. M., Galiana, I., Ding, Y., and Walsh, C. J. (2014). Stronger, smarter, softer: next-generation wearable robots. *IEEE Robot. Autom. Mag.* 21, 22–33. doi: 10.1109/MRA.2014.2360283

Asbeck, A. T., Dyer, R. J., Larusson, A. F., and Walsh, C. J. (2013). "Biologically-inspired soft exosuit," in *IEEE International Conference on Rehabilitation Robotics* (Seattle, WA), 1–8.

Asbeck, A. T., Schmidt, K., Galiana, I., Wagner, D., and Walsh, C. J. (2015). "Multi-joint soft exosuit for gait assistance," *IEEE International Conference on Robotics and Automation* (Seattle, WA), 6197–6204.

Balasubramanian, S., Melendez-Calderon, A., Roby-Brami, A., and Burdet, E. (2015). On the analysis of movement smoothness. *J. Neuroeng. Rehabil.* 12, 1–11. doi: 10.1186/s12984-015-0090-9

Buckley, M. A., Yardley, A., Johnson, G. R., and Cams, D. A. (1996). Dynamics of the upper limb during performance of the tasks of everyday living: a review of the current knowledge base. *J. Eng. Med.* 210, 241–247. doi: 10.1243/PIME_PROC_1996_210_420_02

Canesi, M., Xiloyannis, M., Ajoudani, A., Biechi, A., and Masia, L. (2017). Modular one-to-many clutchable actuator for a soft elbow exosuit. *IEEE Int. Conf. Rehabil. Robot.* 2017, 1679–1685. doi: 10.1109/ICORR.2017.8009489

Cappello, L., Galloway, K. C., Sanan, S., Wagner, D. A., Granberry, R., Engelhardt, S., et al. (2018). Exploiting textile mechanical anisotropy for fabric-based pneumatic actuators. *Soft Robot.* 5, 662–674. doi: 10.1089/soro.2017.0076

Catalano, M. G., Grioli, G., Farnioli, E., Serio, a., Piazza, C., and Bicchi, A. (2014). Adaptive synergies for the design and control of the Pisa/IIT SoftHand. *Int. J. Robot. Res.* 33, 768–782. doi: 10.1177/0278364913518998

Chen, W., and Xie, M. (1999). "On the design of a novel dexterous hand," in *9th International Conference on Advanced Robotics* (Tokyo).

Chiaradia, D., Xiloyannis, M., Antuvan, C. W., Frisoli, A., and Masia, L. (2018). "Design and embedded control of a soft elbow exosuit," in *Proceedings of IEEE International Conference on Soft Robotics* (Livorno).

Clancy, E. A., Morin, E. L., and Merletti, R. (2002). Sampling, noise-reduction and amplitude estimation issues in surface electromyography. *J. Electromyogr. Kinesiol.* 12, 1–16. doi: 10.1016/S1050-6411(01)00033-5

de Looze, M. P., Bosch, T., Krause, F., Stadler, K. S., and O'Sullivan, L. W. (2016). Exoskeletons for industrial application and their potential effects on physical work load. *Ergonomics* 59, 671–681. doi: 10.1080/00140139.2015.1081988

Flash, T., and Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *J. Neurosci.* 5, 1688–1703. doi: 10.1523/JNEUROSCI.05-07-01688.1985

Hunt, T. R., Berthelette, C. J., and Popovic, M. B. (2013). "Linear One-to-Many (OTM) system," in *IEEE Conference on Technologies for Practical Robot Applications* (Woburn, MA: IEEE), 1–6.

In, H., and Cho, K. J. (2015). Exo-Glove : Soft wearable robot for the hand using soft tendon routing system. *IEEE Robot. Autom.* 22, 97–105. doi: 10.1109/MRA.2014.2362863

Karbasi, H., Huissoon, J. P., and Khajepour, A. (2004). Uni-drive modular robots: theory, design, and experiments. *Mech. Mach. Theory* 39, 183–200. doi: 10.1016/S0094-114X(03)00113-7

Kembaum, A. S., Kitchell, M., and Crittenden, M. (2017). "An ultra-compact infinitely variable transmission for robotics," in *IEEE International Conference on Robotics and Automation* (Singapore, SG), 1800–1807.

Kermani, M., and Alex S. (2014). *Magneto-and Electro-Rheological Based Actuators for Human Friendly Manipulators.* U.S. Patent No. 9,566,715. Western University (accessed February 14, 2017).

Li, S., Liu, Y., and Xie, M. (2011). Implementation of a single motor driven manipulator with multiple joints. *Ind. Robot Int. J.* 38, 48–57. doi: 10.1108/01439911111097841

Li, Y., Ma, P., Qin, C., Gao, X., Wang, J., and Zhu, H. (2003). Design and study of a novel hyper-redundant manipulator. *Robotica* 21, 505–509. doi: 10.1017/S0263574703005125

Louie, D. R., and Eng, J. J. (2016). Powered robotic exoskeletons in post-stroke rehabilitation of gait: a scoping review. *J. Neuroeng. Rehabil.* 13:53. doi: 10.1186/s12984-016-0162-5

Mosher, R. S. (1967). Handyman to Hardiman. *SAE Trans.* 76, 588–597. doi: 10.4271/670088

Siraramirez, H. (1989). Sliding regimes in pulse-width-modulation systems-design. *Proceedings of the 28th IEEE Conference on Decision and Control, Vols. 1–3* (Tampa, FL), 2199–2204.

Skoog, R., and Blankenship, G. L. (1970). Generalized pulse-modulated feedback systems: norms, gains, lipschitz constants, and stability. *IEEE Trans. Automat. Contr.* 15, 300–315

Xiloyannis, M., Cappello, L., Binh, D. K., Antuvan, C. W., and Masia, L. (2017). Preliminary design and control of a soft exosuit for assisting elbow movements and hand grasping in activities of daily living. *J. Rehabil. Assist. Technol. Eng.* 4, 1–15. doi: 10.1177/2055668316680315

Xiloyannis, M., Cappello, L., Khanh, D. B., Yen, S. C., and Masia, L. (2016). "Modelling and design of a synergy-based actuator for a tendon-driven soft robotic glove," in *Proceedings of the First IEEE / RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics* (Singapore, SG), 1213–1219.

Xiloyannis, M., Chiaradia, D., Frisoli, A., and Masia, L. (2019). Physiological and kinematic effects of a soft exosuit on arm movements. *J. Neuroeng. Rehabil.* 16:29. doi: 10.1186/s12984-019-0495-y

Yadmellat, P., Shafer, A. S., and Kermani, M. R. (2014). Design and development of a single-motor, two-DOF, safe manipulator. *IEEE/ASME Trans. Mechatron.* 19, 1384–1391. doi: 10.1109/TMECH.2013.2281598

Yu, W., Rosen, J., and Li, X. (2011). "PID admittance control for an upper limb exoskeleton," in *Proceedings of 2011 American Control Conference* (San Francisco, CA), 1124–1129.

Conflict of Interest Statement: EA was employed by company Moveo Walks, Inc., Cambridge, Massachusetts, USA. MC was employed by company Egicon S.R.L., Modena, IT. AK was employed by company Gait Up S.A., Lausanne, Vaud, CH.

The remaining authors declare that the research was conducted in the absence of any commercial or finan-cial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 16 July 2019
doi: 10.3389/fnbot.2019.00049

Autonomous Development of Active Binocular and Motion Vision Through Active Efficient Coding

Alexander Lelais[1], Jonas Mahn[1], Vikram Narayan[1], Chong Zhang[2], Bertram E. Shi[2] and Jochen Triesch[1]*

[1] *Frankfurt Institute for Advanced Studies, Frankfurt, Germany,* [2] *Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong*

We present a model for the autonomous and simultaneous learning of active binocular and motion vision. The model is based on the Active Efficient Coding (AEC) framework, a recent generalization of classic efficient coding theories to active perception. The model learns how to efficiently encode the incoming visual signals generated by an object moving in 3-D through sparse coding. Simultaneously, it learns how to produce eye movements that further improve the efficiency of the sensory coding. This learning is driven by an intrinsic motivation to maximize the system's coding efficiency. We test our approach on the humanoid robot iCub using simulations. The model demonstrates self-calibration of accurate object fixation and tracking of moving objects. Our results show that the model keeps improving until it hits physical constraints such as camera or motor resolution, or limits on its internal coding capacity. Furthermore, we show that the emerging sensory tuning properties are in line with results on disparity, motion, and motion-in-depth tuning in the visual cortex of mammals. The model suggests that vergence and tracking eye movements can be viewed as fundamentally having the same objective of maximizing the coding efficiency of the visual system and that they can be learned and calibrated jointly through AEC.

Keywords: autonomous learning, active perception, binocular vision, optokinetic nystagmus, smooth pursuit, efficient coding, intrinsic motivation

OPEN ACCESS

Edited by:
Florian Röhrbein, Technical University of Munich, Germany

Reviewed by:
Huajin Tang, Sichuan University, China
Jeffrey L. Krichmar, University of California, Irvine, United States

****Correspondence:***
Alexander Lelais
lelais@fias.uni-frankfurt.de

Received: *10 January 2019*
Accepted: *24 June 2019*
Published: *16 July 2019*

Citation:
Lelais A, Mahn J, Narayan V, Zhang C, Shi BE and Triesch J (2019) Autonomous Development of Active Binocular and Motion Vision Through Active Efficient Coding. Front. Neurorobot. 13:49. doi: 10.3389/fnbot.2019.00049

1. INTRODUCTION

The development of sensorimotor and cognitive skills in humans and other animals provides a rich source of inspiration for research in robotics and artificial intelligence. For example, how can we build robots that acquire intelligent behavior in an autonomous and open-ended developmental process mimicking that of human infants? And, in turn, can we use such robotic models to better understand the computational principles underlying human development?

Early stages of human development are largely concerned with learning to control various sensorimotor systems. These systems form the foundation for the later development of higher cognitive functions. Specifically, some of the earliest sensorimotor skills developing in human infants are related to active visual perception. The infant needs to make sense of the signals arriving at her eyes and she needs to learn how to move her eyes to facilitate perception of the world around her. For the development of visual representations (in particular early visual representations) the Efficient Coding Hypothesis has been the most influential theory. Inspired by the development of

information theory, Attneave (1954) and Barlow (1961) have argued in their pioneering works that the visual system exploits the statistical regularities of visual input in order to encode the visual scene efficiently. Furthermore, Barlow (1961) conjectured that early sensory systems have evolved to maximize the amount of information about the visual scene passed to successive processing stages with a constraint of minimizing the associated metabolic costs. Later, the work of Olshausen and Field (1996) established a relation between the statistical structure of natural images and the response properties of cortical simple cells. They proposed to represent natural image patches as linear combinations of sparsely activated basis functions in order to encode the regularities in the images efficiently. Their experiments revealed that a model which learns sparse codes of natural scenes succeeds in developing receptive fields similar to those in the visual cortex. Since then, many experiments have supported the idea that efficient coding is a ubiquitous strategy employed in multiple modalities across diverse organisms (Olshausen and Field, 2004).

A recent extension of the efficient coding hypothesis is Active Efficient Coding (AEC). AEC postulates that biological sensory systems do not just seek to encode the sensory input efficiently, but that they also utilize motor behaviors, such as eye movements, to further improve their coding efficiency (Zhao et al., 2012; Lonini et al., 2013b). Thus, AEC studies efficient coding in the context of behavior and considers the full perception-action cycle and how the organism's behavior shapes the statistics of the sensory signals. AEC works by combining a sparse coding model with a reinforcement learner, which is responsible for generating actions. The sparse coding model learns to efficiently encode the visual input, which serves as a state representation for the reinforcement learner. The reinforcement learner generates actions in order to increase the coding efficiency of the sparse coder.

In previous work, we have successfully applied the AEC approach to model the development of disparity tuning and vergence eye movements using both discrete (Zhao et al., 2012; Lonini et al., 2013b) and continuous actions (Klimmasch et al., 2017). In addition, we have shown that the AEC framework can also be used to model the development of other eye movements such as smooth pursuit (Zhang et al., 2014) and the optokinetic nystagmus (Zhang et al., 2016). Furthermore, the approach has been extended to also learn attention shifts via overt saccadic eye movements (Zhu et al., 2017). In the present study, we present an integrated model of the autonomous learning of active depth and 3-D motion perception using the AEC framework. The model autonomously learns to generate vergence and smooth pursuit eye movements in the presence of a stimulus moving in 3-D. Learning is driven by the agent's intrinsic motivation to maximize its coding efficiency. The advancement to our previous work is the integration of learning to perceive and fixate stimuli located in 3-D and to perceive and track the 3-D motion of respective stimuli. Our results show that the model self-calibrates its eye movement control, improving its performance until it either hits a physical constraint (camera or motor resolution) or runs out of internal resources (capacity of the sparse coding model). Thereby we show and explain the limitations of the model. Furthermore, we show that the model's learned representation of the visual input matches recent findings on the tuning properties of neurons in visual cortex coding for 3-D motion. Thus, the model offers an explanation of how these tuning properties develop in biological vision systems.

2. MATERIALS AND METHODS

2.1. Model Overview

Our model consists of three distinct parts (see **Figure 1**) explained in detail below. At first, one image per camera is preprocessed and dissected into sets of patches. These are encoded by spatio-temporal basis functions of a sparse coding model. This forms a state representation of the sensory input. The state information is processed by a reinforcement learner, which generates camera movements. The negative reconstruction error of the sparse coding stage serves as an indicator of the efficiency of sensory encoding and is used as the reward signal of the reinforcement learner. After execution of the calculated camera movement, the next image pair is sensed and the perception-action cycle starts anew.

2.2. Simulation

We simulate the perception-action cycle by using Gazebo[1], a well known open-source robot simulation platform. Our agent operates the iCub[2] robot in a rendered virtual environment by moving its cameras (see **Figure 2**). The two cameras have a horizontal field of view (FOV) of 90° and a resolution of 320 px × 240 px. The distance between the cameras is $d_E = 0.068$ m. The visual stimuli presented to the agent were taken from the *man made* section of the McGill Calibrated Color Image Database (Olmos and Kingdom, 2004), which contains natural images of urban scenes. Each stimulus had a resolution of 600 px × 600 px. The stimuli were placed on a 1.5 m × 1.5 m plane, perpendicular to the gaze direction. The plane moved within ±30° vertically and horizontally from the agent's center of FOV and [1, 2.5] m in depth. The background image in our virtual environment was taken from Frank Schwichtenberg[3] and is licensed under CreativeCommons (CC BY-SA 4.0).

2.3. Image Processing

At first the images from the left and right camera are gray scaled and convolved with a combined whitening/low-pass filter, a method described by Olshausen and Field (2004). The frequency response of that filter is defined by $R(f) = fe^{-(\frac{f}{f_0})^n}$, where we set the cutoff frequency $f_0 = 96$ px/image and the steepness parameter $n = 4$. Olshausen and Field (2004) stated that such a filter not only reduces various challenges in operating on digitized natural images but roughly resembles the spatial-frequency response characteristic of retinal ganglion cells. Following Lonini et al. (2013b) we use multiple spatial scales to increase the operating range and robustness of our

[1] http://gazebosim.org/

[2] http://www.icub.org/

[3] https://commons.wikimedia.org/wiki/File:Uetersen_Langes_Tannen_Panorama_02.jpg

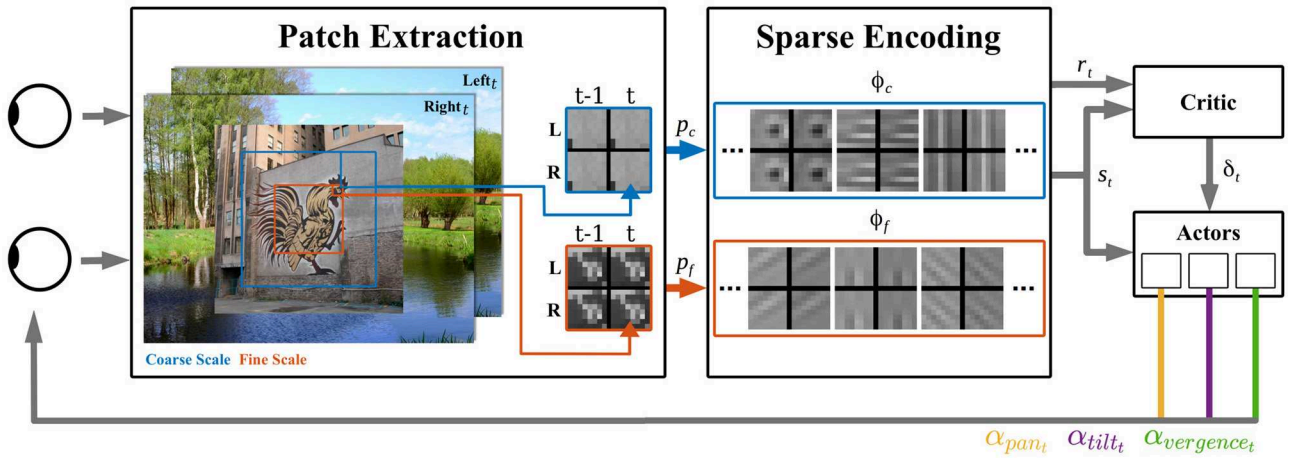


FIGURE 1 | Overview of the active vision architecture. From the binocular visual input at time points $t-1$ and t, patches of different resolutions are extracted for the coarse p_C (blue) and fine p_f (red) scale. These patches are encoded by spatio-temporal basis functions of the coarse scale (blue) and fine scale (red) sparse coders. The activations of both sparse coders' basis functions ϕ_C and ϕ_f form the state vector s_t. The negative reconstruction error indicates the encoding efficiency and is used as the reward signal r_t for the reinforcement learner. The Critic computes from r_t and s_t a TD-error δ_t and three distinct actors generate from s_t movement actions $\alpha_{\text{pan},t}$, $\alpha_{\text{tilt},t}$, $\alpha_{\text{vergence},t}$ for the respective camera joints.

FIGURE 2 | The agent operating the iCub robot inside the virtual environment rendered by the Gazebo simulator.

model. Specifically, we extract two sub-windows with different resolutions from the preprocessed left and right camera image. The first sub-window is a coarse scale 128 px × 128 px image, which covers 36° FOV and corresponds approximately to the human near peripheral vision. To simulate the coarser resolution in the peripheral parts of the human FOV, this image is down-sampled by a factor of 4 via a Gaussian pyramid, resulting in a 32 px × 32 px coarse scale input. The second sub-window is a fine scale 64 px × 64 px image, which covers 18° FOV and corresponds approximately to human macular vision. This image is not down-sampled. Each pixel in the coarse (fine) scale image corresponds to 1.125 (0.281) degrees. Subsequently, patches of 8 px × 8 px size with a stride of 4 px are cut for each scale and camera and normalized to zero mean and unit norm. At each point in time of the simulation, respective patches of the left and right camera image for the current and last iteration are combined to a 16 px × 16 px binocular spatio-temporal patch. This is conducted for each scale and the sets of patches are then encoded by the respective sparse coders.

2.4. Sparse Coding

The sensory input is encoded by sparse coding models for the two scales. For each scale $S \in \{c, f\}$ there is a corresponding dictionary $\mathcal{B}_S$ of spatio-temporal basis functions $\phi_{S,i}$ with $|\mathcal{B}_S| = 600$. For the coarse scale, there are $|p_c| = 49$ spatio-temporal patches and for the fine scale there are $|p_f| = 225$. Each spatio-temporal patch $p_{S,j}$ is encoded by a linear combination of 10 basis functions $\phi_{S,i}$ to form an approximation $\hat{p}_{S,j}$ of the respective patch by

$$\hat{p}_{S,j} = \sum_{i=1}^{|\mathcal{B}_S|} \kappa_{S,i}^{j} \phi_{S,i} \,. \tag{1}$$

This is accomplished by the matching pursuit algorithm (Mallat and Zhang, 1993), where we restrict that 10 $\kappa_{S,i}^{j}$ are non-zero. Hence, we ensure a sparse encoding by using only a subset of 10 basis functions from the dictionary to approximate each image patch. The error of this approximation is the reconstruction error E_S (Lonini et al., 2013b), defined as

$$E_S = \sum_{j=1}^{|p_S|} \frac{||p_{S,j} - \hat{p}_{S,j}||^2}{||p_{S,j}||^2} \,. \tag{2}$$

We use the negative of the total reconstruction error $E = E_c + E_f$ as the reinforcement signal in the next stage of the procedure. The sparse coding model creates as the last step a feature vector s_t of size $2|B_S|$, which serves as the sensory state information

for the reinforcement learner. Every entry in s_t corresponds to the mean squared $\kappa^j_{S,i}$ over all patches. This state representation is motivated by the results of Freeman and Ohzawa (1990) who demonstrated that the response of complex cells could be modeled by summing the squared outputs of preceding simple cells. In our case one can interpret the $\phi_{S,i}$ as receptive fields of simple cells in the visual cortex and the entries in the feature vector as activities of complex cells which pool the activities of simple cells over a larger portion of the visual field.

The receptive field (RF) of a neuron in the visual system refers to the visual attributes of a stimulus it is confronted with which generate a response in that cell. The attributes encoded cover a wide range, such as location within the visual field, orientation, disparity, motion direction, velocity and contrast to name a few. Jones and Palmer (1987) have shown that the RFs of neurons in cat striate cortex are particularly well characterized by 2D Gabor filters. The idea that visual input is encoded by elementary components resembling Gabor functions is supported by Bell and Sejnowski (1997). They demonstrated that orthogonal decompositions of natural scenes lead to filters which are best characterized by Gabor-like functions. How RFs arise in living organisms remains a big topic of investigation. In her review of retinal waves Wong (1999) provides support that these patterns of coordinated activity of the premature retina mediate the shaping of structure and function of the visual system in animals already before birth. The current point of view is that the foundations of the visual system are established by spontaneous activity and molecular cues before eye opening (Huberman et al., 2008; Hagihara et al., 2015). Subsequently, the system is fine tuned by visual experience, especially in the so called *critical period* of development (Thompson et al., 2017). Chino et al. (1997) have quantified the fine tuning of response properties of disparity selective V1 neurons in macaque monkeys during the first four postnatal weeks. They found that a coarse disparity selectivity was already present at the sixth postnatal day. In recent studies it has been shown that RF properties such as orientation and direction sensitivity are even established in mice when they are dark-reared (Ko et al., 2014). In view of this background we initialize our basis functions as Gabor wavelets. Specifically, we assume that neurons in the visual system have RFs resembling 2D Gabor functions already before visual experience is gained. However, we do not assume any correlations in time representation of pairs of RFs or space representations of left and right eye encoding RFs. Thus, we initialize each of the four subfields of all $\phi_{S,i}$ with independent random 2D Gabor functions, defined by

$$g(x,y)_\Delta = \exp(-\frac{x'^2+\beta^2 y'^2}{2\sigma^2})\cos(2\pi\frac{x'}{\lambda}+\psi) \tag{3}$$

$$\Delta = \{\lambda, \theta, \psi, \sigma, \beta, x_c, y_c\} \tag{4}$$

$$x' = (x-x_c)\cos\theta + (y-y_c)\sin\theta \tag{5}$$

$$y' = -(x-x_c)\sin\theta + (y-y_c)\cos\theta\ , \tag{6}$$

where λ is the wavelength of the sinusoidal factor, θ represents the orientation, ψ is the phase offset, σ is the standard deviation of the Gaussian envelope, β is the spatial aspect ratio which specifies the ellipticity, and x_c, y_c are the coordinates of the center. The parameters were drawn from uniform distributions over the following intervals: $\lambda \sim [\frac{8}{3}, 16]$ px, $\theta \sim [0, 180]$ deg, $\psi \sim [0, 360]$ deg, $x_c, y_c \sim [\frac{8}{3}, 8]$ px. The aspect ratio of the Gaussian envelope was set to $\beta = \frac{\lambda}{0.8 \cdot 8}$px and the envelope's standard deviation was kept constant $\sigma = 2.5$px.

The basis functions are adapted during the training to represent the visual input in the best way with respect to its reconstruction. Therefore, the basis functions are updated through gradient descent on the reconstruction error (Olshausen and Field, 1996):

$$\Delta\phi_S = \eta\kappa_S(p_S - \hat{p}_S)\frac{1}{|p_S|}\ , \tag{7}$$

where η is the learning rate, which we set to 0.5 for both scales. After each update step all basis functions are normalized by their energy.

2.5. Reinforcement Learning

In the course of training our agent learns to use the sensory state representation to generate camera movements. For this we use a reinforcement learning approach (Sutton and Barto, 1998) named natural-gradient actor-critic (NAC) with advantage parameters (Bhatnagar et al., 2009). The critic learns to approximate the value function given the current state s_t, which is represented by the sensory state vector provided by the sparse coding model. The actor is generating movement commands on the basis of the current state, which results in a new state and a reward. The goal of the reinforcement learning is to select actions which maximize the discounted cumulative future reward, defined by $R(t) = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$, where we set the reward $r_t = -E_t$ and the discount factor $\gamma = 0.3$. The value function is learned by computing the temporal difference (TD) error δ_t and approximating the average reward $\hat{J}_t$. The TD-error is defined by Equation 8, where $\hat{V}_t(s_t) = \langle \theta^V_t, s_t \rangle$ is the critic's current value function approximation with θ^V_t being the respective parameter vector and $\langle , \rangle$ indicating the inner product of two vectors. The approximation of $\hat{J}_t$ is defined by Equation 9 which is equivalent to low-pass filtering r_t, where $\xi = 0.01$ is the smoothing factor. For the value function approximation we use a two layer artificial neural network (ANN) with $|s_t|$ input neurons, one output neuron and θ^V as weights between the layers. The weights are updated by Equation 10, where $\alpha = 0.4$ is the learning rate of the critic. The low value of γ was found empirically to produce good performance. As the agent receives a reward in every iteration there is no issue of delayed rewards and therefore a fairly strong discounting of future rewards does not disadvantage the learning or performance.

$$\delta_t = r_t - \hat{J}_t + \gamma\hat{V}_t(s_t) - \hat{V}_{t-2}(s_{t-2}) \tag{8}$$

$$\hat{J}_t = (1-\xi)\hat{J}_{t-1} + \xi r_t \tag{9}$$

$$\Delta\theta^V_{i,t} = \alpha\delta_t s_{t-2} \tag{10}$$

The movement commands are generated by three individual actors which control the agent's pan, tilt, and vergence joints of

the cameras, respectively. Each actor maps the current s_t to an action $a \in \mathcal{A} = \{-16, -8, -4, -2, -1, -0.5, 0, 0.5, 1, 2, 4, 8, 16\}$. The actions of the pan and tilt joint controlling actors are interpreted as acceleration commands of the cameras, whereas the vergence joint controlling actor's output is interpreted as change in the vergence angle of the cameras. Therefore, the units for the pan and tilt actions are deg/s^2 and deg for the vergence actions. Each actor is implemented as a two layer ANN with $|s_t|$ input neurons, $|\mathcal{A}|$ output neurons and θ^A as weights between the layers. The activation $z_{a,t}$ of the output neuron corresponding to the respective action a is computed by $z_{a,t} = \langle \theta^A_{a,t}, s_t \rangle$. The actions applied are chosen by sampling from a Softmax policy, where the probability $\pi_{a,t}$ for action a is

$$\pi_{a,t} = \frac{\exp(z_{a,t}T^{-1})}{\sum_{n=1}^{|\mathcal{A}|} \exp(z_{n,t}T^{-1})}\,, \tag{11}$$

where T is the temperature parameter, which controls the exploration vs. exploitation behavior of the agent. We set $T = 1$ to ensure the agent explores while learning. The actors' weights θ^A are updated by

$$\zeta_t = \nabla_\theta \log \pi_\theta(a_{t-2} \mid s_{t-2})\,, \tag{12}$$

$$\Delta w_t = \beta(\delta_t \zeta_t - \zeta_t(\zeta_t^T w_{t-1}))\,, \tag{13}$$

$$\Delta \theta^A_t = \eta w_t\,, \tag{14}$$

where ζ_t are the policy derivatives, w_t are the advantage parameters, β is the learning rate of the natural gradient and η is the learning rate of the actor. The family of NAC algorithms are reinforcement learning approaches, which combine learning from the TD-error δ_t and a policy gradient. However, instead of following the regular (vanilla) policy gradient, NAC algorithms are following the *natural* gradient to update the actor's weights θ^A. A thorough derivation and discussion of the natural gradient is provided, e.g., by Peters et al. (2005). The NAC algorithm with advantage parameters w_t does not explicitly store an estimate of the inverse Fisher information matrix, which the other members of the NAC family are using to follow the natural gradient as Bhatnagar et al. (2009) point out. This makes the NAC algorithm with advantage parameters computationally cheaper and the approximation of the natural gradient through the w_t is comparable to the other members of the NAC family. The interested reader is referred to Bhatnagar et al. (2009) for derivations of Equations 8–10 and Equations 12–14 and convergence analysis and discussion of various NAC algorithms. We set for all actors $\beta = 0.16$ and $\eta = 0.4$. Due to the model's architecture, it takes two iteration steps until an action has its full effect on the state representation. Therefore, we update the critic and the actors with respect to s_{t-2} and a_{t-2}.

2.6. Experimental Procedure

In our experiments we probe the agent's capability to learn to fixate and track a moving stimulus. Each experiment consists of $5 \cdot 10^5$ training iterations, each corresponding to 100 ms. Experiments are repeated 10 times with different randomization seeds. Training is divided into intervals, each lasting 40 iterations. At the start of each interval, a stimulus is drawn at random from a set of 100 images from the McGill dataset and centered in the agent's FOV. The stimulus is positioned at a distance to the agent drawn from a uniform distribution over $[1, 2.5]$ m. The agent initially fixates on a point directly in front of it at a distance chosen at random from the interval $[0.3, 3]$ m. During the interval, the stimulus moves according to velocities drawn from uniform distributions over $[-7.5, 7.5]$ deg/s in the horizontal and vertical directions and $[-0.375, 0.375]$ m/s in depth. The agent updates the pan and tilt velocities of its eyes and the vergence angle between them according to the policy. In case any joint exceeds a pre-defined angle boundary (± 15 deg for the pan/tilt joint, $[0.2, 16.3]$ deg for the vergence joint), the joint velocities are set to zero and the agent's gaze is reset as described afore.

3. RESULTS

We start by presenting the quality of sensory state encoding of our approach. **Figure 3A** shows the reconstruction error of both sparse coders vs. training time in solid lines. The improvement of stimulus reconstruction in both scales over the course of training clearly shows an increase in coding efficiency. As we enforce the encoding to be sparse (see Equation 1), the agent works with the same small amount of resources throughout training. Hence, by improving the encoding result using the same amount of resources as at the start of training, the agent increases its encoding efficiency. We also tested the encoding performance of the sparse coding model in a testing procedure (which is described further below) with a stimulus set disjoint from the training set. As the agent showed similar reconstruction capabilities in both training and testing procedures (compare **Figure S3**), the learned sparse coding dictionary can be considered generic (at least for urban scenes and man-made objects as they occur in the data base). In a control experiment we used the same model but exchanged the action generation of the reinforcement learning (RL) by a uniform sampling at random of the pan, tilt and vergence actions from the same action sets we used before. The encoding performance of both sparse coders in this control experiment is shown in **Figure 3A** in dashed lines. The sparse coders' coding efficiency does not significantly improve in this setup in the course of training compared to the model using RL for action generation. This shows that the RL does improve the coding efficiency in our AEC framework. In **Figure 3B** six representative spatio-temporal basis functions of the coarse scale dictionary are depicted at initialization time and at the end of the training. The fine scale bases look similar. The basis functions were initialized by random Gabor wavelets, but the sparse coding model has adjusted the bases to properly encode the stimulus statistics it was confronted with.

For a qualitative impression of the reconstruction performance, a stimulus is shown at different processing and training stages in **Figure 4**. The comparison of the preprocessed input images and the respective reconstructions thereof, shows

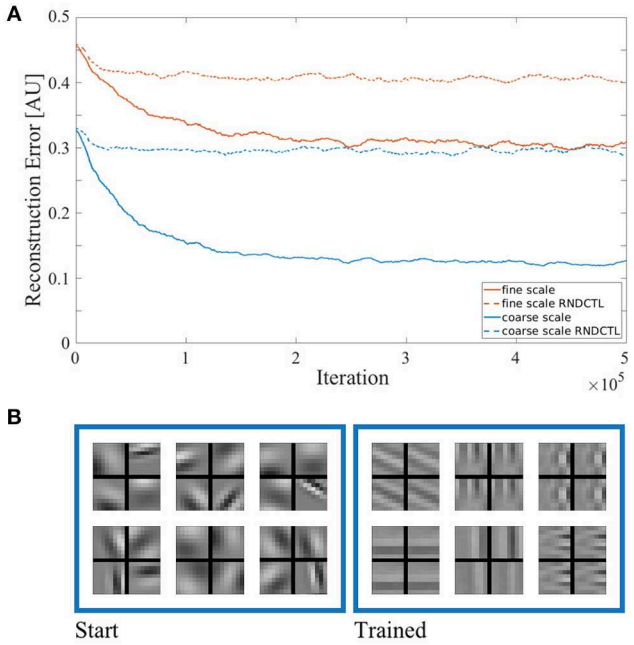


FIGURE 3 | (A) Reconstruction error of the sparse coding model. The error is plotted in arbitrary units vs. training time for the coarse scale (blue) and fine scale (red) sparse coder in solid lines. The model's encoding performance in a control experiment, where the actions were uniformly sampled at random from the same action sets (RNDCTL) is plotted in dashed lines. **(B)** Coarse scale basis functions. Six representative spatio-temporal basis functions of a coarse scale dictionary are shown at the start (left) and the end of training (right). Every basis function consists of 4 parts. The rows show the corresponding patch for the left (top) and right (bottom) eye. The columns represent the patch for time $t-1$ (left) and t (right).

a clear improvement of the reconstruction quality between the sparse coding model at initialization time and at the end of training. All images are shown for the left eye and its respective basis parts at time t are used for encoding and reconstruction. For a fair comparison between the trained and the untrained agent the joint angles of the cameras are set to perfectly fixate the center of the stimulus. The image reconstruction is already at initialization time fairly decent due to the size of the sparse coding dictionary, the amount of basis functions used for individual patch encoding and the perfect fixation of the stimulus. Though, the encoding and therefore the reconstruction improves as the basis functions are adapted to the stimulus statistics. The image reconstruction with white noise initialized basis functions looks more noisy at initialization time but similar at the end of training (see **Supplementary Material**).

We tested the policy at 10 points during training for $50 \cdot 6^3 = 10800$ trials, each corresponding to one of the possible combinations of 50 stimuli chosen from a set of images from the McGill database disjoint from the training set and 6 velocities in each of the three directions (horizontal, vertical and depth). The velocities were chosen from $\{\pm 0.1, \pm 0.5, \pm 1\}$ times the maximum velocities in each direction. Each trial lasted for 10 iterations, as no performance improvement was gained after that. To correctly track the stimulus, the agent needs to rotate its eyes with the same speed as the stimulus is moving in the respective direction. Therefore, the errors for the pan and tilt joints Δv were measured in deg/iteration as the difference between the speeds of the object and the eyes at the last iteration of the trial. The error for the vergence joint $\Delta\xi$ was defined as the difference between the actual and desired vergence angle, which was computed by

$$\xi^* = 2 \arctan\left(\frac{d_E}{2d_O}\right), \tag{15}$$

where d_E is the horizontal separation between the eyes and d_O is the object distance. During the performance assessment, the learning of the sparse coders and the reinforcement learner was switched off and the actors applied a greedy policy. The testing performance is depicted in **Figure 5A**. For each of the respective joints the median of the absolute error at the last iteration of a testing trial is plotted in solid lines and one IQR is indicated by shading. Statistics are computed over all testing trials.

We also examined the influence of the sparse coder's basis function dictionary size on the testing performance. **Figure 5B** shows the testing performance after training for $5 \cdot 10^5$ iterations for $|\mathcal{B}_S| \in \{100, 200, 400, 600\}$ on the same test stimulus set used before. Each experiment was repeated 3 times with different randomization seeds. A student's t-test revealed a significant improvement (p-values $< 10^{-8}$) for all comparisons marked in **Figure 5B**. The errors decrease with increasing dictionary size. Calculation of the effect sizes by Cohen's d (Cohen, 1988) showed that increasing the dictionary size to 800 results in a neglectable effect of $d < 0.045$ compared to $|\mathcal{B}_S| = 600$. Therefore, we conclude that the model's performance saturates when $\sim$ 600 basis functions are present. Initializing the basis functions with white noise yielded similar results (see **Supplementary Material**), though the learning progress was less robust, as the IQRs were bigger before convergence.

In **Figure 6** we provide a more detailed view of the learned policies averaged over 10 agents and the 50 stimuli of the test set. It shows the probability distributions of the action sets of the respective pan, tilt and vergence actor over a range of errors in the corresponding state space. The ideal policy π^* is a diagonal in each case. The pan and tilt actor's policy was probed by moving the stimulus only along the respective dimension. For the vergence actor the stimulus's distance was varied but the object remained static. Thereby, we avoided any interference between the actors. The pan and tilt actors perform more accurately the bigger the absolute speed errors are. For small speed errors the ideal action is not uniquely identified. The vergence policy shows the desired diagonal structure only for negative and small positive vergence errors $\Delta\xi$. This is due to the ranges of initial eye fixations and stimulus depths in our experimental setup. Specifically, the agent is rarely confronted with big positive vergence errors and never with $\Delta\xi > 3$ deg (see Equation 15). An accurate vergence policy for large positive $\Delta\xi$ would require a training setup where such vergence errors are encountered regularly.

For a qualitative impression of the behavior we show in **Figure 7** good examples of movement trajectories of an agent for one stimulus. For the pan and tilt dimension the stimulus

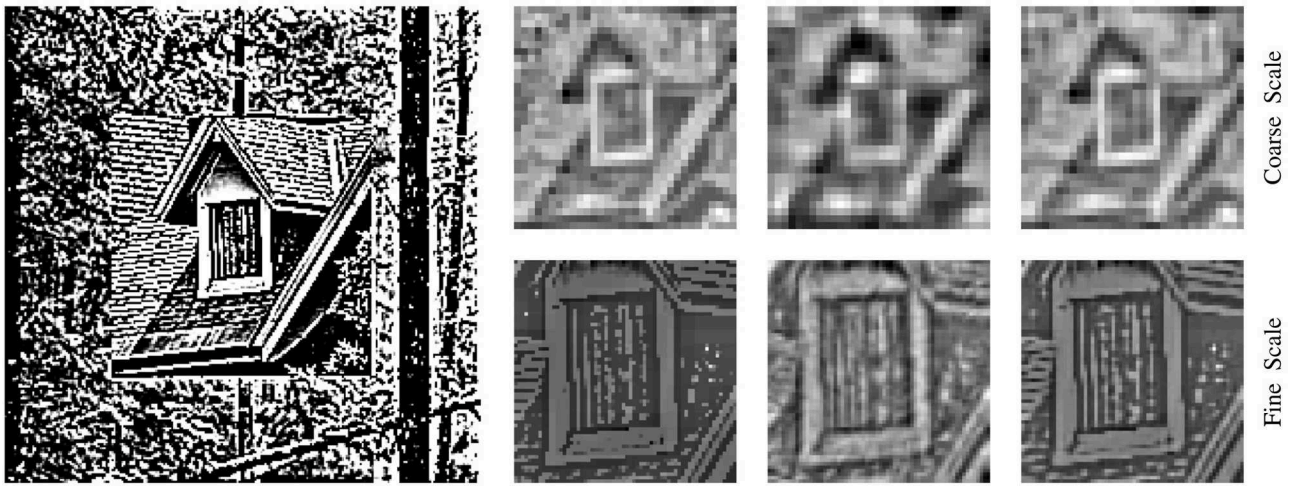


FIGURE 4 | Input image reconstruction. Depicted are column wise from left to right the original whitened camera image, the cropped, down-sampled, and normalized input image for the coarse scale (top row) and fine scale (bottom row) sparse coders. Right to the preprocessed images are the respective images reconstructed with random Gabor wavelets at initialization time and the images reconstructed with learned basis functions at the end of training.

speed changed and the respective joint speed was reset to 0 deg/s every 10 iterations. For the vergence dimension the eyes were initialized with varying $\Delta\xi$ errors every 10 iterations. We demonstrate the agent's performance additionally in a video (see **Supplementary Material**) for various stimuli and movement speeds. The object tracking is driven by the movement of the object, as the agent can best encode the visual input stream of the moving object if the object remains static on the retina (camera images). The agent manages to establish a static retinal image by moving its eyes with the same speed as the fixated object moves.

In two additional experiments we investigated the limits of our model. Both were conducted the same way as described before. In the first experiment we reduced the camera resolution by providing no fine scale sparse coder (NFS) to the agent. In the second experiment we reduced the motor resolution by providing a coarser action set (CAS) to the agent. The coarser action set was defined by $\mathcal{A} = \{-16, -8, -4, -2, -1, 0, 1, 2, 4, 8, 16\}$ for all actors, i.e., the actions ± 0.5 have been removed. **Figure 8** shows the results of this analysis. A student's t-test was used to compare the performance between the agents with NFS and CAS and an agent with standard configuration (STD). The difference between NFS and STD was significant (p-values $< 10^{-57}$) with an effect size of Cohen's $d > 0.622$ for all joints. The comparison between the agent with the CAS and STD showed significant differences for the tilt and vergence actor (p-values $< 10^{-57}$) with effect sizes of $d > 0.219$. The difference between the pan actors was also significant ($p = 0.003$), but the effect size of $d = 0.041$ was relatively small. These results demonstrate that the agent keeps improving until it hits physical constraints such as camera or motor resolution, or limits on its internal coding capacity as shown in **Figure 5B**.

3.1. Analysis of Basis Function Properties

We investigated whether the learned basis functions maintained a Gabor-like structure and compared their properties to biological data. For that we fitted 2D Gabor functions (see Equations 3–6) to the four sub-fields of the basis functions. The squared norm of the residual of the basis functions r had a mean of $\mu = 0.003 \pm 0.006$ SD at initialization time. After training the mean of r was $\mu = 0.038 \pm 0.034$ SD for the coarse scale and $\mu = 0.011 \pm 0.016$ SD for the fine scale basis functions. Basis functions initialized with white noise have a mean of r of $\mu = 0.188 \pm 0.022$ SD. Hence, the basis functions remained Gabor-like. The histograms of orientation preferences θ of the coarse scale (blue) and fine scale (red) basis functions are depicted in **Figure 9A**. Vertical ($\sim 42\%$) and horizontal ($\sim 22\%$) orientations are most common. This is in line with biological findings on the so-called *oblique effect*, which show an over-representation of vertical and horizontal RFs in many species such as cats, monkeys and humans (Appelle, 1972; Li et al., 2003). This bias is strongly shaped by the stimulus statistics the agent is facing during training, as there is a prevalence of horizontal and vertical edges. We have investigated RF properties which arise from normal and abnormal rearing conditions in more detail in Klimmasch et al. (2018).

We further analyze the disparity preferences $\hat{d}$ of the basis functions for the respective basis sub-parts representing time t for both scales (see **Figure 9B**). The disparity preference at time t is computed by

$$\hat{d} = \frac{\lambda(\psi_{t,\text{left}} - \psi_{t,\text{right}})}{2\pi \cos\theta} . \tag{16}$$

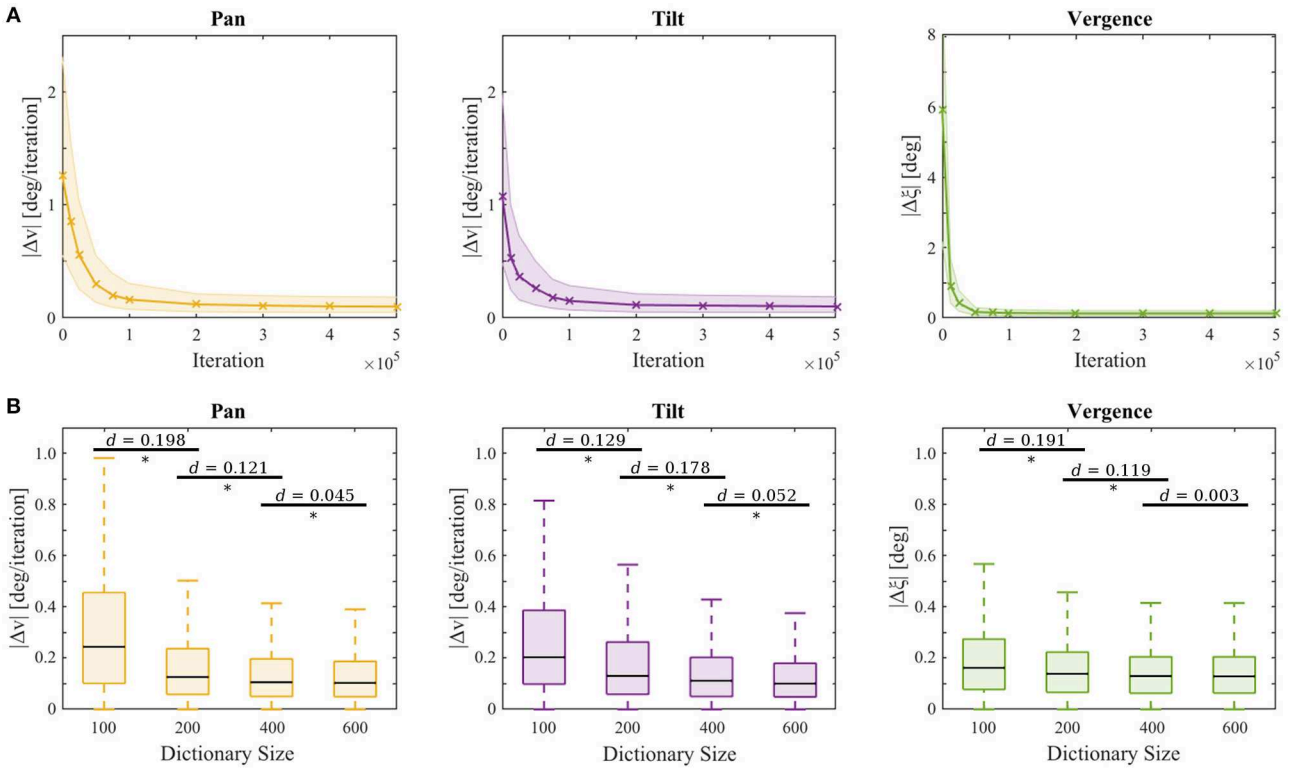


FIGURE 5 | (A) Testing performance vs. training iteration. Depicted are the respective errors in the pan Δv (yellow), tilt Δv (purple), and vergence $\Delta\xi$ (green) joint of the testing procedures for all test stimuli and movement speeds over 10 trials at the respective points in time during the training procedure. The lines represent the median errors and the shaded areas show one inter quartile range. **(B)** Testing performance at the end of training for agents with different sizes of sparse coding dictionaries over 3 experiment repetitions. Significant differences ($p < 0.05$) between two sets of data are assessed by a t-test and marked (*). Horizontal bars indicate effect size as measured by Cohen's d.

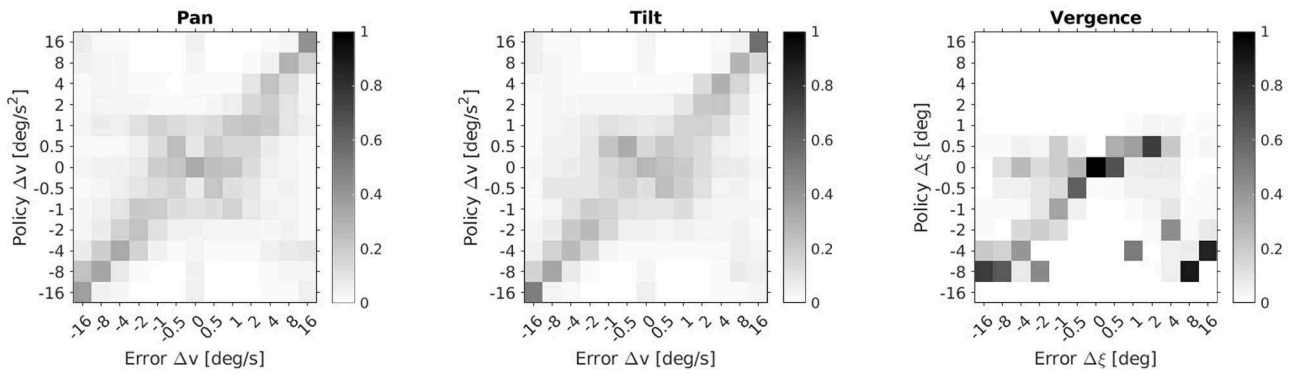


FIGURE 6 | Learned policy distributions averaged over 10 agents and 50 stimuli. Depicted are action probabilities for the respective pan, tilt and vergence actor as a function of state errors.

The distribution of preferred disparities is centered at zero degrees and covers a range of about $\pm 2^\circ$ for the fine scale and $\pm 8^\circ$ for the coarse scale. This is consistent with the biological finding that the majority of receptive fields in macaque V1 and middle temporal (MT) visual cortex are tuned to near zero disparities (Prince et al., 2002; DeAngelis and Uka, 2003).

The velocity preference $\hat{v}$ for a given eye, say, the left eye, is computed by

$$\hat{v} = \frac{\lambda |\psi_{t,\text{left}} - \psi_{t-1,\text{left}}|}{2\pi} . \tag{17}$$

Figure 9C shows that the basis functions have a preference for encoding low velocities at both coarse and fine scale. Orban

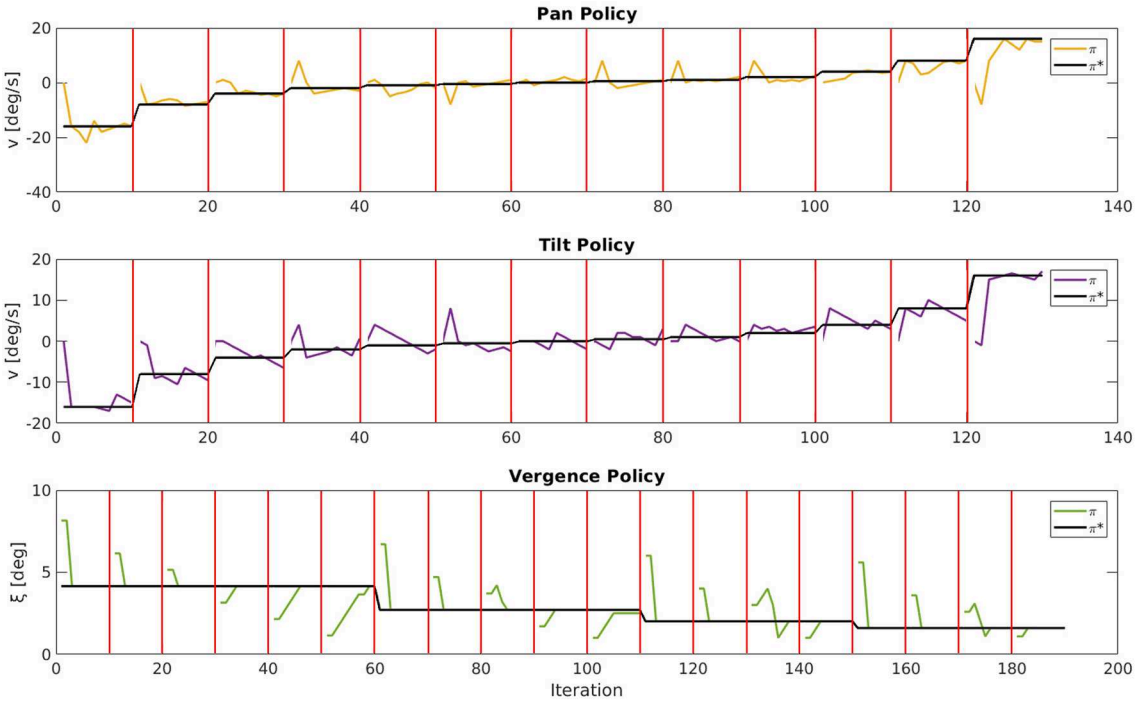


FIGURE 7 | Movement trajectories of an agent for one stimulus. For pan and tilt the respective joint speed was reset to 0 deg/s every 10 iterations as indicated by the red bars. For the vergence joint the fixation angle ξ was initialized with varying vergence errors every 10 iterations. The actual policy π is plotted, respectively, in yellow (pan), purple (tilt), and green (vergence) and the desired policy $\pi*$ in black.

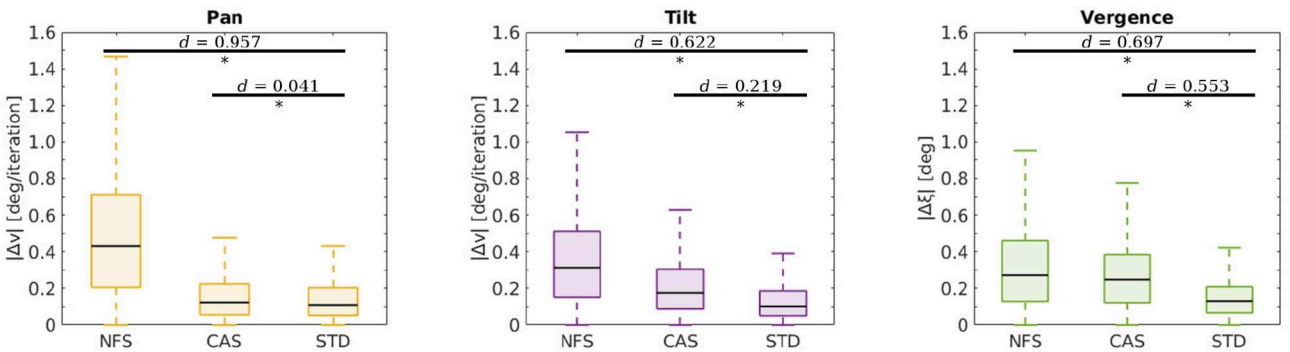


FIGURE 8 | Testing performance at the end of training for agents with different configurations. Depicted are the respective errors in the pan Δv (yellow), tilt Δv (purple) and vergence $\Delta\xi$ (green) joint for all test stimuli and movement speeds. The configurations span the situations when there is no fine scale sparse coder (NFS), a coarser action set (CAS), and when the standard configuration is used (STD). Horizontal bars indicate comparisons between two sets of data as assessed by a *t*-test. Significant differences (p-values < 0.05) are marked (*) and effect sizes are indicated as measured by Cohen's d.

et al. (1986) analyzed the velocity preference of V1 and V2 neurons in macaque monkeys and Felleman and Kaas (1984) have shown for the further visual processing path in cortex of owl and macaque monkeys that neurons in the MT cortex are also encoding stimulus velocities but typically higher velocities than neurons in V1 and V2. This is most likely due to the increased receptive field size of MT neurons compared to RF sizes of neurons in lower areas. The stimulus selectivity of our basis function sub-parts show similar $\hat{v}$ distributions to V1 and V2 velocity preference of neurons encoding the central visual field (compared to Orban et al., 1986). Therefore, our results provide support for interpreting the *sub-parts* of our basis functions, i.e., the columns, as RFs of binocular simple cells in V1/V2 and a *complete* basis as the response of a complex cell pooling activities from multiple simple cells.

Figure 10A shows the disparity preference $\hat{d}$ of the basis functions at time t vs. $t - 1$. This illustrates that the agent has learned representations for all situations it was confronted

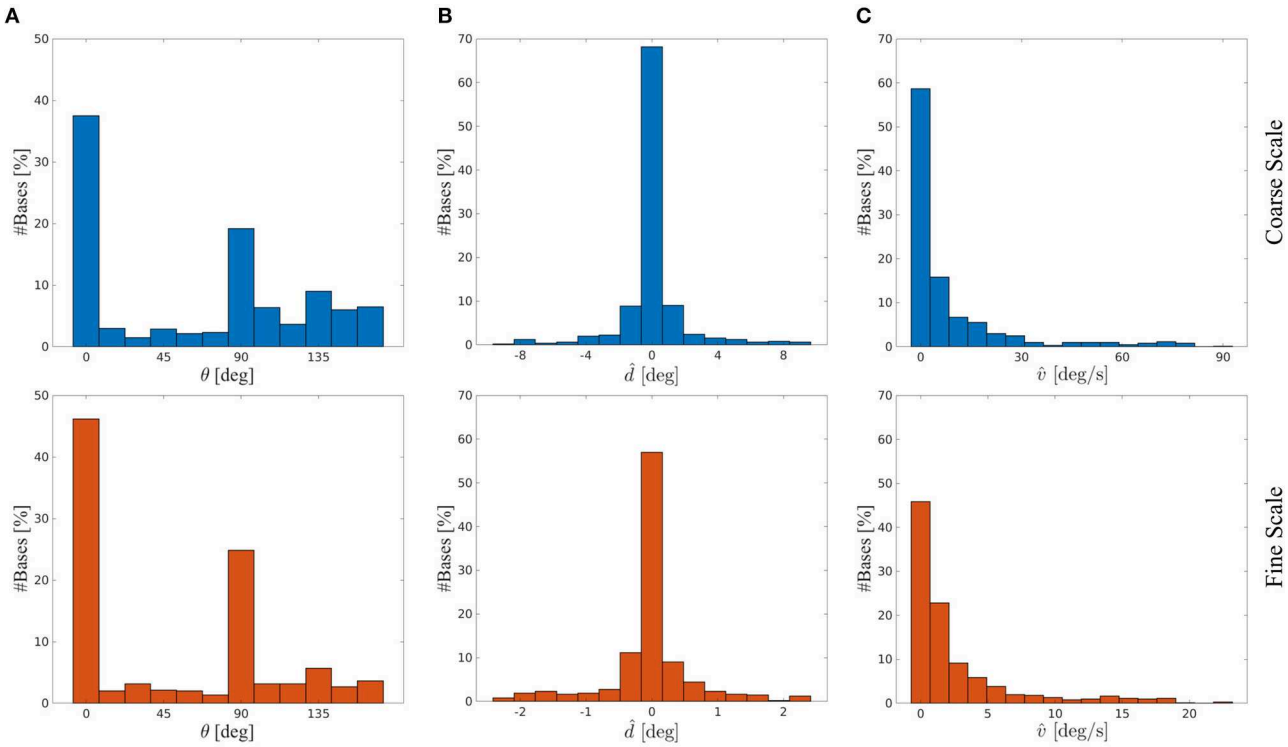


FIGURE 9 | Basis functions' stimulus preferences for the coarse scale (blue) and fine scale (red) from a typical experiment. **(A)** Histogram of orientation preferences θ. **(B)** Histogram of disparity preferences $\hat{d}$. **(C)** Histogram of velocity preferences $\hat{v}$.

with during the training phase. Specifically, 45.5 % of the basis functions are representing a disparity of $|\hat{d}| \leq 1.125\,\text{deg} \mathrel{\hat{=}} 1\,\text{px}$ in the coarse scale and 40 % of the basis functions encode $|\hat{d}| \leq 0.281\,\text{deg} \mathrel{\hat{=}} 1\,\text{px}$ in the fine scale, respectively. These basis functions represent the situations where the agent was fixating the stimulus within 1 px accuracy at time $t-1$ and kept on fixating it at time t. Other basis functions show, e.g., tuning for close to zero disparity at time $t-1$ but not at time t. Such basis functions can detect object movement in depth, where the object leaves the current fixation plane.

In general, various kinds of motion can be encoded with our basis functions, such as fronto-parallel and 3-D motion. In case of equal velocity representation for left and right eye, a fronto-parallel motion is encoded. Whereas different velocity preferences in both eyes represent a motion in depth (Czuba et al., 2014). **Figure 10B** depicts the results of this analysis for both scales. The linear correlation between the basis parts representing the left and right eye shows a correlation coefficient of $\rho = 0.215$ for the coarse and $\rho = 0.289$ for the fine scale. This indicates that most basis functions are encoding motion in depth, nevertheless a considerable amount of basis functions are representing fronto-parallel motion.

Electrophysiological recordings from neurons in the MT area of macaque visual cortex show that most MT neurons are tuned to both binocular disparity and the direction of stimulus motion, and many MT neurons have their disparity and motion tuning independent of each other (DeAngelis and Newsome, 2004). A more recent study of Sanada and DeAngelis (2014) has shown that about a half of the neurons in macaque MT cortex are selective for the direction of motion in depth with some contribution of disparity cues. In this context we analyzed the average velocity preference of both eyes vs. the average disparity preference for t and $t-1$ in **Figure 10C** to study the results of joint encoding of both velocity and disparity. It is evident from **Figure 10C** that the velocity and disparity preferences have no linear correlation and thus they respond to a combination of specific disparity and motion. Despite a peak in near zero velocity and disparity, as already seen in **Figure 9**, one can clearly observe that the learned basis functions are encoding a wide range of velocities and disparities.

The distribution of preferred disparities in the model (see **Figure 9B**) has less variance compared to biological data (DeAngelis and Uka, 2003; Sprague et al., 2015). We investigated whether the agent encounters a too narrow range of disparities during training, as the range of object distances is small and the objects are planar textures. Hence, we trained our agent as described before, but at each training interval the stimulus was at random either tilted or slanted by an angle uniformly drawn from ± 45 deg and the stimulus distance was uniformly drawn from $[0.3, 1.5]$ m. As suspected, this manipulation resulted in a larger variance of the distribution of preferred disparities of the basis functions for both scales. We verified this by applying a Brown-Forsythe test on distributions of preferred disparities trained in the standard and the afore

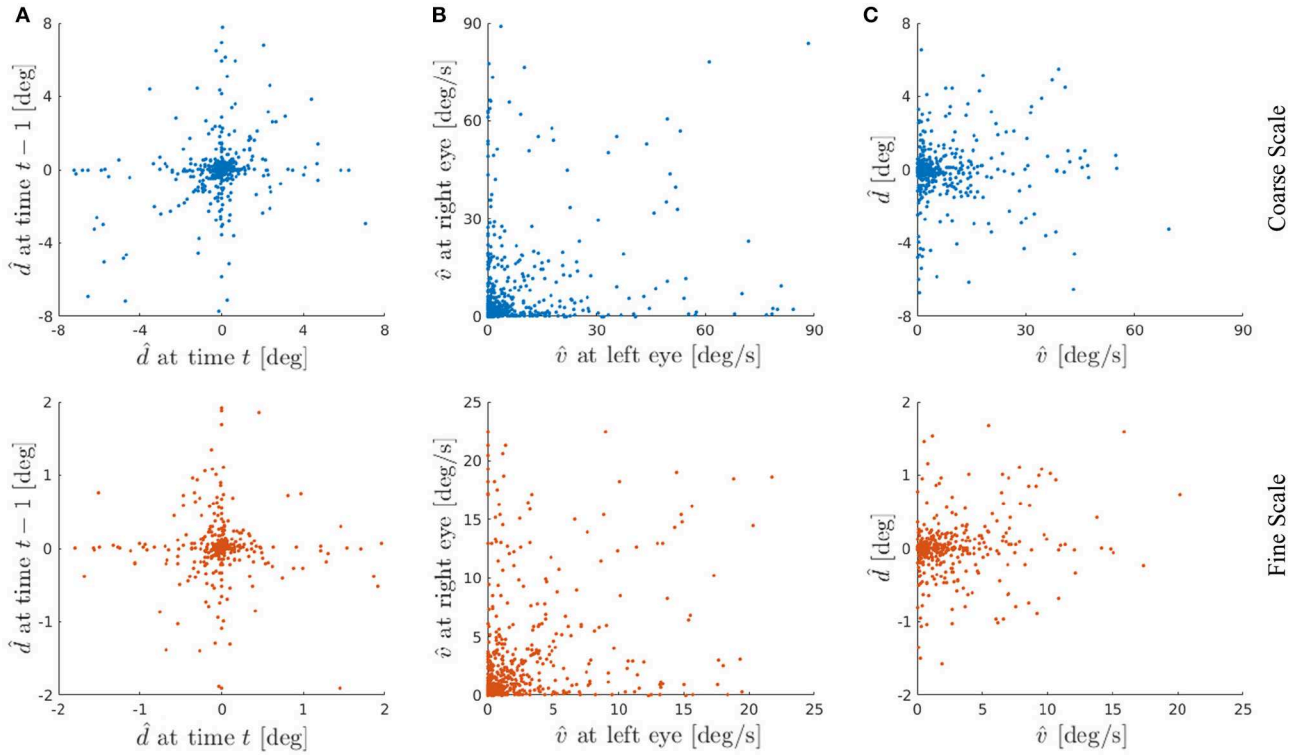


FIGURE 10 | (A) Basis functions' disparity preference $\hat{d}$ at time t vs. $t-1$ from a typical experiment. Each dot represents one basis function of the coarse scale (blue) or the fine scale (red). **(B)** Basis functions' velocity preference $\hat{v}$ at left eye vs. right eye. **(C)** Basis functions' velocity $\hat{v}$ vs. disparity $\hat{d}$ preference averaged over left and right eye and time t and $t-1$, respectively. Here we show the basis functions' joint encoding of velocity and disparity. The basis functions are sensitive to a wide range of combinations of preferred velocities and preferred disparities. The velocity and disparity preferences of basis functions are not correlated.

mentioned modified scenario. For both the coarse scale ($p = 1.39 \cdot 10^{-3}$) and the fine scale ($p = 2.81 \cdot 10^{-2}$) the test indicated a significant increase in the variance of the distributions of preferred disparities. The testing performance over 10 repetitions with different randomization seeds in this modified scenario was similar to the standard scenario. Hence, with our approach the agent can encode and track non-fronto-parallel objects as well. The study of Zhu et al. (2017) has demonstrated that within our AEC framework an agent can also learn to fixate 3D objects. In an additional control experiment we tested the standard agent with a sphere-shaped object instead of the fronto-parallel plane and projected the same natural textures on top of the sphere as in the standard testing procedure. In a video (see **Supplementary Material**) we demonstrate that an agent which was trained with a fronto-parallel plane can also fixate and track a sphere-shaped object.

The shape of the stimulus does not limit our approach, but the size of the stimulus does. The extent of the FOV of the agent (see Seq. 2.3), the amount of patches in the coarse and fine scale, the formalization of the reconstruction error (see Equation 2) and the resulting reward signal determine the minimum size of the stimulus which can still be tracked by our agent. As the agent strives to minimize the total reconstruction error E, it is fixating and tracking the image regions that contribute most to E. Hence, if the stimulus is covering more of the FOV than the background, it is encoded by more patches of the sparse coders and therefore the stimulus contributes more to E than the background does. Hence, if <50 % of E is accounted for by the stimulus, the agent will focus on the background instead. Considering the number of patches in the fine and coarse scale regions and their overlap, one can estimate that successful tracking requires that the stimulus covers $\sim$ 80 % of the area of the fine scale. This means that if the stimulus width is 0.5 m it is not rewarding for the agent to fixate and track it when the distance to the stimulus is $\geq$ 1 m. We show in a video (see **Supplementary Material**) the agent's behavior in the discussed situations where it is confronted with a 1.5 m, 1.2 m, and a 0.3 m wide object.

4. DISCUSSION

The fixation of an object in depth and its pursuit with the eyes when it moves are two elementary visual capabilities that emerge early during human development. We have demonstrated that Active Efficient Coding is well suited as a model for the joint learning of these two basic visual abilities, which were learned separately in our previous works. Our model learns an efficient representation of depth and motion via sparse coding. In parallel, a reinforcement learning component learns to generate a behavior which facilitates the efficient encoding of the scene by the sparse coding component via an intrinsic motivation

for coding efficiency. Thereby the agent simultaneously learns a representation of the visual scene and the fixation and pursuit behavior in a completely autonomous fashion. To the best of our knowledge, the joint learning of both sensory representation and behavior is unique to our approach. For example, the recent approach by Konda and Memisevic (2014) also learns disparity and motion representations, but it does so from a fixed set of training videos via supervised learning and it does not include the learning of any behavior which would change the statistics of the sensory signals. Conversely, the approach by Gibaldi et al. (2015) learns to execute vergence eye movements, but the set of filter banks which are used to process the input images is predefined and does not adapt to the statistics of the visual input. Indeed, the majority of existing models for learning vergence or smooth pursuit have a much narrower focus than our work. Early models detected only specific velocities or disparities (Rashbass and Westheimer, 1961; Krishnan and Stark, 1977). Some works only used synthetic and not natural images (Patel et al., 1997; Gibaldi et al., 2010). The studies of Hoyer and Hyvärinen (2000), Hunter and Hibbard (2015), and Chauhan et al. (2018) used unsupervised approaches to learn binocular disparity selectivity from natural stereoscopic images. In the work of Burge and Geisler (2014) disparity selectivity was learned by optimizing disparity discrimination in natural images. Importantly, the focus of these studies was on learning representations of still images and these models do not learn or produce any behavior and none addresses motion selectivity. Beyeler et al. (2016) show a model how the motion signal from MT cortex could be further processed by medial superior temporal (MSTd) cortex. They present an alternative approach how sparse basis functions, which show similar tuning properties as macaque MSTd neurons, could emerge from MT units through a dimensionality reduction technique. In contrast to our work, their MT units are predefined and the model does not generate any behavior. Other works required the engineering of specific image features, knowledge of the intrinsic parameters of the camera, or a predefined model of object velocity or disparity. In addition, most works on motion vision do not address the issue of binocular vision, because they only consider monocular visual input.

The tasks learned by our model, vergence control and smooth pursuit, are similar to those learned by the model of Zhang et al. (2016), vergence control and the optokinetic nystagmus (OKN). Both smooth pursuit and the OKN are minimizing the retinal slip, but smooth pursuit is associated with smaller targets and more voluntary eye movements. The architectures presented here and in Zhang et al. (2016) are similar in that they show the same sparse coding based perceptual stage and the same reinforcement learner for the vergence commands. However, they differ in the learning of the smooth pursuit/OKN. Here we use reinforcement learning, but Zhang et al. (2016) use Hebbian learning combined with scaffolding by a subcortical pathway. The work here provides a more parsimonious model, but Zhang et al. (2016) is more consistent with the observed developmental interactions between the cortical and subcortical pathways underlying the OKN.

Many experimental studies on binocular disparity tuning in the brain have found evidence suggesting that the primary visual cortex (V1) optimally processes the natural binocular disparity statistics. In this regard, the efficient coding hypothesis conjectures that the disparity tuning of V1 binocular neurons reflects the natural range of disparities (Read and Cumming, 2004; Liu et al., 2008) and that eye movement strategy is such that it minimizes the binocular disparity and motor inefficiency (Tweed, 1997; Schreiber et al., 2001). These findings are consistent with our model.

The work of Yu et al. (2005) has shown that neurons in primary visual cortex exhibit higher coding efficiency when responding to correlated signals compared to uncorrelated ones. Our AEC framework similarly exploits correlations in sensory signals that are generated through its own motor behavior. Specifically, as our model learns vergence eye movements it learns to reduce disparities between the eyes and therefore increases the redundancy between left and right camera input. Similarly, our model increases the redundancy between successive images of its cameras as it learns to perform pursuit eye movements. The agent's actions ultimately result in a more efficient encoding of the visual scene, because the model adapts its basis functions to efficiently exploit the redundancies in the sensory signals that it is creating through its own learned motor behavior.

It has been well established in the neuroscience literature that the RFs in primary visual cortex of certain mammalian species already have a Gabor-like structure before visual experience is gathered, i.e., before eye opening. Therefore, we also initialized the basis functions in our model to already have Gabor shapes at the start of learning. Importantly, however, as seems to be the case in biology, the left and right subfields of the basis functions were statistically independent. In addition to the experiments presented above, we also tested if the model can still learn successfully without such a Gabor initialization of the basis functions. We observed that the model still learns successfully, when the basis functions are initialized as independent Gaussian white noise (see **Supplementary Material**).

The analysis of the basis functions confirms the findings of Qian (1994) and Smolyanskaya et al. (2013) that disparity and motion tuning are largely independent of each other. Czuba et al. (2014) have shown that MT neurons encode 3-D motion and in this regard we also observe the presence of basis functions which have different velocity preferences between left and right eye, thus being sensitive to 3-D motion. Furthermore, some basis functions are also encoding fronto-parallel movement and overall a broad range of velocities and disparities. Therefore, they resemble the encoding properties of real neurons in the visual system.

To the best of our knowledge, apart from our work (Zhang et al., 2016), research on vergence eye movements and research on pursuit eye movements and the optokinetic nystagmus has been progressing independently. In stark contrast to this tradition, our new model suggests that these phenomena can be unified and seen as special cases of the general idea of Active Efficient Coding, i.e., the idea of a sensory system exploiting its motor degrees of freedom to support the efficient encoding of information from the environment. In fact, recent work suggests that torsional eye movements (Zhu et al., 2018) and the control of accommodation (Triesch et al., 2017) are just two further instances of this very general idea.

In previous studies we have shown that our AEC approach also works on the iCub robot in a real life scenario (Lonini et al., 2013a; Teulière et al., 2015). As our model presented in this study shows good performance on the simulated iCub, we are confident that future studies will prove its robustness on the real iCub. This should be tested in future work.

The present model may also have implications for developmental disorders of the visual system such as strabismus and amblyopia (Eckmann et al., 2018). As a first model of how sensory and motor aspects of binocular and motion vision jointly develop and self-calibrate, it may be a useful testbed for studying what factors can derail this development in developmental disorders and what treatments may bring it back on track.

AUTHOR CONTRIBUTIONS

JT, AL, and JM designed the experiments. AL and JM conducted the experiments. VN and CZ implemented initial versions of the model and AL and JM implemented the final version of it. CZ implemented a first version of the Gabor wavelet fitting algorithm and AL and JM implemented the final version of all analysis related algorithms. All authors contributed to the paper writing.

FUNDING

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant 713010 and was supported in part by the German Federal Ministry of Education and Research under grants 01GQ0840, 01GQ1414, and 01EW1603A and by the Hong Kong Research Grants Council under grant 16213617. JT is supported by the Johanna Quandt Foundation.

ACKNOWLEDGMENTS

We would like to thank Benoit R. Cottereau, Samuel Eckmann, Lukas Klimmasch, Johann Schneider, and Charles Wilmot for helpful suggestions and fruitful discussions.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2019.00049/full#supplementary-material

Video 1 | Active binocular and motion vision—testing performance.

Video 2 | Active binocular and motion vision—testing with 3D stimuli and limitations.

REFERENCES

Appelle, S. (1972). Perception and discrimination as a function of stimulus orientation: the "oblique effect" in man and animals. *Psychol. Bull.* 78:266.

Attneave, F. (1954). Some informational aspects of visual perception. *Psychol. Rev.* 61, 183–193.

Barlow, H. B. (1961). "Chapter 13: Possible principles underlying the transformations of sensory messages," in *Sensory Communication,* ed W. Rosenblith (Cambridge: MIT Press), 217–234.

Bell, A. J., and Sejnowski, T. J. (1997). The "independent components" of natural scenes are edge filters. *Vision Res.* 37, 3327–3338.

Beyeler, M., Dutt, N., and Krichmar, J. L. (2016). 3d visual response properties of mstd emerge from an efficient, sparse population code. *J. Neurosci.* 36, 8399–8415. doi: 10.1523/JNEUROSCI.0396-16.2016

Bhatnagar, S., Sutton, R. S., Ghavamzadeh, M., and Lee, M. (2009). Natural actor-critic algorithms. *Automatica* 45, 2471–2482. doi: 10.1016/j.automatica.2009.07.008

Burge, J., and Geisler, W. S. (2014). Optimal disparity estimation in natural stereo images. *J. Vision* 14, 1–1. doi: 10.1167/14.2.1

Chauhan, T., Masquelier, T., Montlibert, A., and Cottereau, B. R. (2018). Emergence of binocular disparity selectivity through hebbian learning. *J. Neurosci.* 38, 9563–9578. doi: 10.1523/JNEUROSCI.1259-18.2018

Chino, Y. M., Smith III, E. L., Hatta, S., and Cheng, H. (1997). Postnatal development of binocular disparity sensitivity in neurons of the primate visual cortex. *J. Neurosci.* 17, 296–307.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences, 2nd ed.* Hillsdale: Erlbaum Associates.

Czuba, T. B., Huk, A. C., Cormack, L. K., and Kohn, A. (2014). Area mt encodes three-dimensional motion. *J. Neurosci.* 34, 15522–15533. doi: 10.1523/JNEUROSCI.1081-14.2014

DeAngelis, G. C., and Newsome, W. T. (2004). Perceptual "read-out" of conjoined direction and disparity maps in extrastriate area mt. *PLoS Biol.* 2:e77. doi: 10.1371/journal.pbio.0020077

DeAngelis, G. C., and Uka, T. (2003). Coding of horizontal disparity and velocity by mt neurons in the alert macaque. *J. Neurophysiol.* 89, 1094–1111. doi: 10.1152/jn.00717.2002

Eckmann, S., Klimmasch, L., Shi, B., and Triesch, J. (2018). A model of the development of anisometropic amblyopia through recruitment of interocular suppression. *J. Vision* 18, 942–942. doi: 10.1167/18.10.942

Felleman, D. J., and Kaas, J. H. (1984). Receptive-field properties of neurons in middle temporal visual area (mt) of owl monkeys. *J. Neurophysiol.* 52, 488–513.

Freeman, R. D., and Ohzawa, I. (1990). On the neurophysiological organization of binocular vision. *Vision Res.* 30, 1661–1676.

Gibaldi, A., Canessa, A., Solari, F., and Sabatini, S. P. (2015). Autonomous learning of disparity–vergence behavior through distributed coding and population reward: Basic mechanisms and real-world conditioning on a robot stereo head. *Robot. Auton. Syst.* 71, 23–34. doi: 10.1016/j.robot.2015.01.002

Gibaldi, A., Chessa, M., Canessa, A., Sabatini, S. P., and Solari, F. (2010). A cortical model for binocular vergence control without explicit calculation of disparity. *Neurocomputing* 73, 1065–1073. doi: 10.1016/j.neucom.2009.11.016

Hagihara, K. M., Murakami, T., Yoshida, T., Tagawa, Y., and Ohki, K. (2015). Neuronal activity is not required for the initial formation and maturation of visual selectivity. *Nat. Neurosci.* 18:1780–1788. doi: 10.1038/nn.4155

Hoyer, P. O., and Hyvärinen, A. (2000). Independent component analysis applied to feature extraction from colour and stereo images. *Netw. Comput. Neural Syst.* 11, 191–210. doi: 10.1088/0954-898X-11-3-302

Huberman, A. D., Feller, M. B., and Chapman, B. (2008). Mechanisms underlying development of visual maps and receptive fields. *Annu. Rev. Neurosci.* 31, 479–509. doi: 10.1146/annurev.neuro.31.060407.125533

Hunter, D. W., and Hibbard, P. B. (2015). Distribution of independent components of binocular natural images. *J. Vision* 15, 6–6. doi: 10.1167/15.13.6

Jones, J. P., and Palmer, L. A. (1987). An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *J. Neurophysiol.* 58, 1233–1258.

Klimmasch, L., Lelais, A., Lichtenstein, A., Shi, B. E., and Triesch, J. (2017). "Learning of active binocular vision in a biomechanical model of the oculomotor system," in *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (Lisbon), 21–26.

Klimmasch, L., Schneider, J., Lelais, A., Shi, B. E., and Triesch, J. (2018). "An active efficient coding model of binocular vision development under normal

and abnormal rearing conditions," in *International Conference on Simulation of Adaptive Behavior* (Frankfurt: Springer), 66–77.

Ko, H., Mrsic-Flogel, T. D., and Hofer, S. B. (2014). Emergence of feature-specific connectivity in cortical microcircuits in the absence of visual experience. *J. Neurosci.* 34, 9812–9816. doi: 10.1523/JNEUROSCI.0875-14.2014

Konda, K. R., and Memisevic, R. (2014). "A unified approach to learning depth and motion features," in *Indian Conference on Computer Vision, Graphics and Image Processing* (Bangalore), 78.

Krishnan, V., and Stark, L. (1977). A heuristic model for the human vergence eye movement system. *IEEE Trans. Biomed. Eng.* 1, 44–49.

Li, B., Peterson, M. R., and Freeman, R. D. (2003). Oblique effect: a neural basis in the visual cortex. *J. Neurophysiol.* 90, 204–217. doi: 10.1152/jn.00954.2002

Liu, Y., Bovik, A. C., and Cormack, L. K. (2008). Disparity statistics in natural scenes. *J. Vision* 8, 19.1–14. doi: 10.1167/8.11.19

Lonini, L., Forestier, S., Teulière, C., Zhao, Y., Shi, B. E., and Triesch, J. (2013a). Robust active binocular vision through intrinsically motivated learning. *Front. Neurorobot.* 7:20. doi: 10.3389/fnbot.2013.00020

Lonini, L., Zhao, Y., Chandrashekhariah, P., Shi, B. E., and Triesch, J. (2013b). "Autonomous learning of active multi-scale binocular vision," in *2013 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (Osaka), 1–6.

Mallat, S., and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* 41, 3397–3415.

Olmos, A., and Kingdom, F. A. (2004). A biologically inspired algorithm for the recovery of shading and reflectance images. *Perception* 33, 1463–1473. doi: 10.1068/p5321

Olshausen, B. A., and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381, 607–609.

Olshausen, B. A., and Field, D. J. (2004). Sparse coding of sensory inputs. *Curr. Opin. Neurobiol.* 14, 481–487. doi: 10.1016/j.conb.2004.07.007

Orban, G. A., Kennedy, H., and Bullier, J. (1986). Velocity sensitivity and direction selectivity of neurons in areas v1 and v2 of the monkey: influence of eccentricity. *J. Neurophysiol.* 56, 462–480.

Patel, S. S., Öğmen, H., White, J., and Jiang, B. C. (1997). Neural network model of short-term horizontal disparity vergence dynamics. *Vision Res.* 37, 1383–1399.

Peters, J., Vijayakumar, S., and Schaal, S. (2005). "Natural actor-critic," in *European Conference on Machine Learning* (Porto: Springer), 280–291.

Prince, S. J., Pointon, A. D., Cumming, B. G., and Parker, A. J. (2002). Quantitative analysis of the responses of v1 neurons to horizontal disparity in dynamic random-dot stereograms. *J. Neurophysiol.* 87, 191–208. doi: 10.1152/jn.00465.2000

Qian, N. (1994). Computing stereo disparity and motion with known binocular cell properties. *Neural Comput.* 6, 390–404.

Rashbass, C., and Westheimer, G. (1961). Disjunctive eye movements. *J. Physiol.* 159, 339–360.

Read, J. C., and Cumming, B. G. (2004). Understanding the cortical specialization for horizontal disparity. *Neural Comput.* 16, 1983–2020. doi: 10.1162/0899766041732440

Sanada, T. M., and DeAngelis, G. C. (2014). Neural representation of motion-in-depth in area mt. *J. Neurosci.* 34, 15508–15521. doi: 10.1523/JNEUROSCI.1072-14.2014

Schreiber, K., Crawford, J. D., Fetter, M., and Tweed, D. (2001). The motor side of depth vision. *Nature* 410, 819–822. doi: 10.1038/35071081

Smolyanskaya, A., Ruff, D. A., and Born, R. T. (2013). Joint tuning for direction of motion and binocular disparity in macaque mt is largely separable. *J. Neurophysiol.* 110, 2806–2816. doi: 10.1152/jn.00573.2013

Sprague, W. W., Cooper, E. A., Tošić, I., and Banks, M. S. (2015). Stereopsis is adaptive for the natural environment. *Sci. Adv.* 1:e1400254. doi: 10.1126/sciadv.1400254

Sutton, R. S., and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*, vol 1. Cambridge: MIT press Cambridge.

Teulière, C., Forestier, S., Lonini, L., Zhang, C., Zhao, Y., Shi, B., and Triesch, J. (2015). Self-calibrating smooth pursuit through active efficient coding. *Robot. Auton. Syst.* 71, 3–12. doi: 10.1016/j.robot.2014.11.006

Thompson, A., Gribizis, A., Chen, C., and Crair, M. C. (2017). Activity-dependent development of visual receptive fields. *Curr. Opin. Neurobiol.* 42, 136–143. doi: 10.1016/j.conb.2016.12.007

Triesch, J., Eckmann, S., and Shi, B. (2017). A computational model for the joint development of accommodation and vergence control. *J. Vision* 17, 162–162. doi: 10.1167/17.10.162

Tweed, D. (1997). Visual-motor optimization in binocular control. *Vision Res.* 37, 1939–1951.

Wong, R. O. (1999). Retinal waves and visual system development. *Ann. Rev. Neurosci.* 22, 29–47.

Yu, Y., Romero, R., and Lee, T. S. (2005). Preference of sensory neural coding for 1/f signals. *Phys. Rev. Lett.* 94:108103. doi: 10.1103/PhysRevLett.94.108103

Zhang, C., Triesch, J., and Shi, B. E. (2016). An active-efficient-coding model of optokinetic nystagmus. *J. Vision* 16:10. doi: 10.1167/16.14.10

Zhang, C., Zhao, Y., Triesch, J., and Shi, B. E. (2014). "Intrinsically motivated learning of visual motion perception and smooth pursuit," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong: IEEE), 1902–1908.

Zhao, Y., Rothkopf, C. A., Triesch, J., and Shi, B. E. (2012). "A unified model of the joint development of disparity selectivity and vergence control," in *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (San Diego, CA: IEEE), 1–6.

Zhu, Q., Triesch, J., and Shi, B. E. (2017). "Autonomous, self-calibrating binocular vision based on learned attention and active efficient coding," in *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (Lisbon), 27–32.

Zhu, Q., Zhang, C., Triesch, J., and Shi, B. E. (2018). "Autonomous learning of cyclovergence control based on active efficient coding," in *2018 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (Tokyo).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

TECHNOLOGY AND CODE
published: 14 August 2019
doi: 10.3389/fnbot.2019.00065

Reachy, a 3D-Printed Human-Like Robotic Arm as a Testbed for Human-Robot Control Strategies

Sébastien Mick[1]*, *Mattieu Lapeyre*[2], *Pierre Rouanet*[2], *Christophe Halgand*[1], *Jenny Benois-Pineau*[3], *Florent Paclet*[1], *Daniel Cattaert*[1], *Pierre-Yves Oudeyer*[4] and *Aymar de Rugy*[1,5]*

[1] Institut de Neurosciences Cognitives et Intégratives d'Aquitaine, UMR 5287 CNRS & Univ. Bordeaux, Bordeaux, France, [2] Pollen Robotics, Bordeaux, France, [3] Laboratoire Bordelais de Recherche en Informatique, UMR 5800, CNRS & Univ. Bordeaux & Bordeaux INP, Talence, France, [4] Inria Bordeaux Sud-Ouest, Talence, France, [5] Centre for Sensorimotor Performance, School of Human Movement and Nutrition Sciences, University of Queensland, Brisbane, QLD, Australia

To this day, despite the increasing motor capability of robotic devices, elaborating efficient control strategies is still a key challenge in the field of humanoid robotic arms. In particular, providing a human "pilot" with efficient ways to drive such a robotic arm requires thorough testing prior to integration into a finished system. Additionally, when it is needed to preserve anatomical consistency between pilot and robot, such testing requires to employ devices showing human-like features. To fulfill this need for a biomimetic test platform, we present Reachy, a human-like life-scale robotic arm with seven joints from shoulder to wrist. Although Reachy does not include a poly-articulated hand and is therefore more suitable for studying reaching than manipulation, a robotic hand prototype from available third-party projects could be integrated to it. Its 3D-printed structure and off-the-shelf actuators make it inexpensive relatively to the price of an industrial-grade robot. Using an open-source architecture, its design makes it broadly connectable and customizable, so it can be integrated into many applications. To illustrate how Reachy can connect to external devices, this paper presents several proofs of concept where it is operated with various control strategies, such as tele-operation or gaze-driven control. In this way, Reachy can help researchers to explore, develop and test innovative control strategies and interfaces on a human-like robot.

Keywords: robotic arm, humanoid robot, research testbed, 3D printing, open-source, rehabilitation engineering

OPEN ACCESS

Edited by:
Alex Pitti,
Université de Cergy-Pontoise, France

Reviewed by:
Arnaud Blanchard,
École Nationale Supérieure de l'électronique et de ses Applications, France
Davide Marocco,
University of Naples Federico II, Italy

***Correspondence:**
Sébastien Mick
sebastien.mick@u-bordeaux.fr
Aymar de Rugy
aymar.derugy@u-bordeaux.fr

Received: *03 April 2019*
Accepted: *29 July 2019*
Published: *14 August 2019*

Citation:
Mick S, Lapeyre M, Rouanet P, Halgand C, Benois-Pineau J, Paclet F, Cattaert D, Oudeyer P-Y and de Rugy A (2019) Reachy, a 3D-Printed Human-Like Robotic Arm as a Testbed for Human-Robot Control Strategies. Front. Neurorobot. 13:65. doi: 10.3389/fnbot.2019.00065

1. INTRODUCTION

While robotic systems keep improving in terms of motor capabilities thanks to progress in mechatronics, developing control strategies and interfaces allowing a human to harness the full potential of an advanced robotic arm proves to be a key challenge in the field of humanoid robotics and in particular, rehabilitation engineering. Indeed, user surveys and reviews (Biddiss and Chau, 2007; Cordella et al., 2016) have already revealed that the lack of functionality and the necessity of a long and difficult training were some main reasons behind upper-limb prosthesis abandonment. As examples drawn from some of the most advanced devices currently on the prosthesis market, Michelangelo (Ottobock) and i-limb quantum (Touch Bionics) hands include too many actuators for an amputee to operate them independently, and their control relies a lot

on pre-programmed grip patterns. Even in the case of an able-bodied human, the gap between robotic devices' complexity and available command signals highlights the need for efficient and usable control interfaces and strategies.

To bridge this gap, researchers have investigated techniques to retrieve additional input data from the human. One of these solutions is the sensor fusion approach, which intends to combine measurements from multiple sensors running at once. This approach can be used with various devices and sensing modalities (Novak and Riener, 2015), whether vision-based, kinematic, or physiological. In particular, as object recognition from egocentric videos can help grasping actions for neuroprostheses (de San Roman et al., 2017), recent works explored how a robotic system could be controlled by fusing eye-tracking with EMG (Corbett et al., 2013, 2014; Markovic et al., 2015; Gigli et al., 2017) or ElectroEncephaloGraphy (EEG) signals (McMullen et al., 2014; Wang et al., 2015). Other works also investigated how Augmented Reality (AR) can be employed to provide relevant visual feedback about a robotic arm's state (Markovic et al., 2014, 2017), with the aim of improving the control loop.

Another approach to overcome this limit is to reduce the need for command signals, by making the robotic system take charge of part of its own complexity. In this way, techniques are developed to allow a human to drive a robot through higher-level, task-relevant commands instead of operating the robot directly in actuator space. A common implementation of this approach is to perform endpoint control through Inverse Kinematics (IK), which convert command signals from the 3D operational space into the actuator space. IK solving is a key research topic in the whole field of robotics, including autonomous humanoid robotics (Bae et al., 2015; Rakita et al., 2018), but can also be employed to manage the kinematic redundancy of human-driven robots (Zucker et al., 2015; Rakita et al., 2017; Meeker et al., 2018).

To evaluate the performance of control techniques, virtual reality (VR) has been employed for more than a decade (Hauschild et al., 2007; Kaliki et al., 2013; Phelan et al., 2015; Blana et al., 2016). Recently, this approach also benefits from the increasing availability of VR development kits on the market, e.g., Oculus (Facebook Technologies, LLC) and VIVE (HTC Corporation), making it easier for researchers to integrate a virtual test environment into their experimental setup. However, a robotic system simulated within a virtual environment would not behave the same way as a physical device, inherently subject to mechanical limits and imperfections. Indeed, VR setups usually implement a simplified device (e.g., ideal, friction-less actuators) in a simplified context (e.g., ignoring gravity). As a result, conclusions drawn from assessments performed in a virtual test environment may not be directly applicable to an actual robot.

On the other hand, some researchers use actual robotic arms to get more realistic data from the testing phase. Works from the literature are found to employ both commercially available devices (Rakita et al., 2017; Meeker et al., 2018) and prototype systems (McMullen et al., 2014; Bae et al., 2015) in their research. More specifically, in the field of prosthetics, many multi-DoF devices have been developed as experimental prototypes, such as the UNB hand (Losier et al., 2011), the Yale hand (Belter and Dollar, 2013), and the SmartHand (Cipriani et al., 2011).

Among such research devices, the ones developed by Dawson et al. (2014) and Krausz et al. (2016) were designed with the aim of being inexpensive and easily available to other researchers, as open-source systems including 3D-printed parts. Indeed, as 3D-printing allows to produce complex and custom shapes in small batches at a low cost, this manufacturing technique is useful when developing products at the prototype stage. Besides, the fact that the same parts can be produced by many different 3D printers participates notably in the shareablity of these designs.

In this paper, we present Reachy, a life-size test platform to be used by researchers to explore, develop, and test control strategies and interfaces for human-driven robotics. Relying on technical solutions drawn from similar works, we aimed at designing a robot that would be **affordable**, **shareable**, and "**hackable**" compared to high-end prototypes or commercially available robotic arms; but also more **human-like** than industrial-grade robots. Indeed, Reachy benefits from its closeness to a human arm in terms of scale and shape, as well as motor features and joint structure. Additionally, even though its use cases are not limited to this field, this robotic platform is primarily intended for applications in prosthetics and rehabilitation engineering.

2. ROBOT DESIGN

2.1. Design Principles

Reachy was created with the aim of providing researchers with a robotic platform on which to test control interfaces and strategies that would be employed to drive a robotic arm. In order to make the robot a relevant tool in the field of rehabilitation technologies, its structure puts the emphasis on human-likeness. Indeed, Reachy is meant to emulate the behavior of a life-size human upper limb, while being fixed at shoulder level on an unmovable support.

Besides, another major requirement of Reachy's design was to ensure that the robot is suitable for a variety of applications ranging from neuroprostheses to teleoperated manipulators. Thus, in order for Reachy to be a versatile platform, we intended to make it extensively customizable, as well as easily and broadly connectable. Ensuring extensive experimental reproducibility in this context requires the platform to allow for thorough hardware modifications, as well as the sharing of said modifications within the scientific community. Therefore, we chose to develop Reachy's design on the following principles and technical solutions: **3D-printed** plastic skeleton parts; **off-the-shelf actuators**, mechanical components and electronics; **free and open-source sharing** of both software and hardware resources.

Reachy was designed by the creators of the Poppy project (Lapeyre et al., 2014), a family of robots for research, art and education that rely on a common software and hardware architecture, but display a variety of shapes, features and purposes. In particular, the first robot of this family, Poppy Humanoid, was originally designed to investigate the role of morphology in biped locomotion (Lapeyre et al., 2013), thus generating the need for a platform whose parts could easily

be redesigned, produced then assembled. The aforementioned design principles directly stem from the philosophy and technical solutions that drove Poppy's development.

2.2. Hardware

Reachy was initially developed as a "full-length" arm, that is to say, a prototype comprising the three segments of the human upper-limb, from shoulder to hand. In this "standard" version, Reachy weighs 1.4 kg and measures 60 cm from shoulder to wrist, with dimensions and proportions similar to those of a human adult's right arm. These prototypes have been equipped and tested with various end-effectors (see **Figure 1B**): a basic sphere, a jointless anthropomorphic hand, or a two-prong clamp providing a minimal grasping feature. Furthermore, as the robot is meant to be customized and "hacked", Reachy users can adapt its distal end to fit an existing robotic hand chosen among available research prototypes (Losier et al., 2011; Belter and Dollar, 2013; Krausz et al., 2016). For instance, a new prototype featuring the Brunel hand (OpenBionics) as the end-effector has undergone development in order to expand the robot's features and capabilities.

In its standard version, the robot comprises seven independent DoF, each of them actuated by a dedicated motor. The first three motors operate the gleno-scapulo-humeral joint in a simplified way, by performing three consecutive rotations: shoulder flexion-extension, shoulder abduction-adduction, and humeral lateral-medial rotation. The three motors' rotation axes intersect at a single point, located at the center of the second motor's shaft (see **Figure 1A**). The shoulder assembly also comprises two roller bearings to facilitate the operation of the first and third DoF. In order for the robot to remain reasonably easy to model and build, this assembly does not reproduce the system of interdependent rotations and translations forming the gleno-scapulo-humeral complex, but still allows for a wide range of motion. The fourth and fifth motors operate respectively elbow flexion-extension and forearm pronation-supination, the latter being mounted with a ball bearing. Finally, the last two motors operate the wrist joint by performing consecutively radial-ulnar deviation and flexion-extension. Their respective rotation axes are orthogonal, however they do not intersect; instead, the two motors are linked by a short piece joining the forearm and end-effector. This interval between rotation axes shares some resemblance to that which separates a human's wrist joint axes, even though it is slightly too large because of the actuator size.

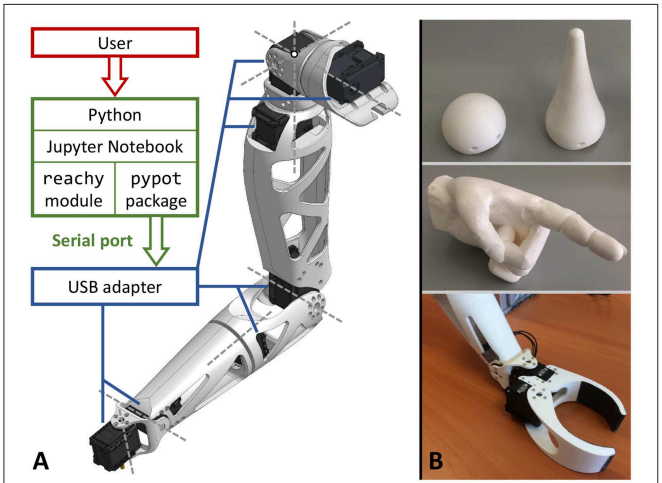


FIGURE 1 | Reachy in its standard version. **(A)** Architectural diagram describing the software stack, from the high-level software interface to the motors. Dashed lines indicate the joints' rotation axes. **(B)** Currently available end-effectors. From top to bottom: spherical, conical, hand-shaped, and articulated clamp.

We chose to employ Robotis Dynamixel servomotors[1] to actuate Reachy's seven DoF. These motors are all-in-one modules that provide a good trade-off between accuracy, speed and robustness in mechanical terms, as well as embedded sensors monitoring angular speed and position. They also allow the individual tuning of an internal Proportional-Integral-Derivative (PID) controller, maximum torque and mechanical compliance. Due to these features, Dynamixel servomotors enable a high level of modularity while being able to produce rich motor behaviors. For that matter, this range of actuators is commonly used in the field of robotics, including humanoid robots (Ha et al., 2011; Ly et al., 2011; Hild et al., 2012; Schwarz et al., 2013; Dawson et al., 2014). Apart from actuators, only few additional mechanical components are needed to assemble Reachy's joints, namely: the three aforementioned bearings, and transmission wheels to insert on each servomotor's shaft. In particular, as all these actuators include an individual gearbox, Reachy's assembly does not require extra reduction mechanisms for joint actuation.

Three different models of Dynamixel motors are included in Reachy's standard version. The most powerful one is an MX-106 and operates the most proximal DoF of the robot, shoulder flexion-extension, while the shoulder's remaining DoF and elbow joint are actuated by MX-64. As these four joints support the heaviest loads while the robot is put in motion, the corresponding motors were chosen accordingly. The forearm and wrist joints, which do not require as much power, are operated by AX-18, lighter and smaller than MX servomotors, so that the robot's weight distribution leans toward the proximal end.

Regarding the robot's skeleton, the limbs' design relies on a trellis-like structure to reduce the weight and keep assembly simple, by providing easy access to screw holes. This open, low-density structure also improves motor heat dissipation thanks to freer air circulation. Prototypes and current versions of Reachy were printed in polyamide or Poly-Lactic Acid (PLA), two materials commonly used in the additive manufacturing industry. Their low cost, availability and compatibility with most desktop 3D printers make them ideal for prototyping, while their durability and printing resolution make them adequate for finished products with good quality standards.

2.3. Electronics and Software

Reachy's motors are connected with each other in a series using three-pin connectors and powered by a pair of 12 V×5 A power

[1]http://en.robotis.com/subindex/dxl_en.php

supply units, for a total power of 120 W. At one end of the series, a USB adapter allows for plugging into a computer. The robot is then controlled through a serial port with a software interface called Pypot, which handles the communication with Dynamixel servomotors to drive the robot, e.g., sending motor commands, retrieving data from embedded sensors. This architecture is illustrated in **Figure 1A**.

Developed as part of the Poppy project, this software base is common to the whole Poppy-Reachy family of Dynamixel-powered robots. Following an open-source approach, Pypot was entirely written in Python in order to enable cross-platform deployment, as this language is compatible with most desktop operating systems as well as some embedded systems for single-board computers. Python programming also allows for fast development by emphasizing code readability and conciseness, so that developers can efficiently produce clear programs whether their project is of small or large scale. Besides, Reachy users can take advantage of numerous Python libraries dedicated to scientific computing, and already in use within the scientific community. This allows them to combine Reachy's features with techniques such as signal processing or machine learning, without having to resort to other languages or software.

While its open-source nature provides expert programmers with extensive freedom over the system, Pypot is also intended to be accessible to beginners. In particular, it provides high-level motor commands over the joints' angular positions and mechanical compliance, so that any user can program a trajectory and put the robot in motion with only a few lines of code (see **Supplementary Material**). Additionally, tutorials are provided to Reachy users in the form of Jupyter notebooks (Kluyver et al., 2016), which are interactive development supports combining source code, formatted text, plots, and graphical input/output widgets. Jupyter notebooks can be created from a Web navigator and don't require any dedicated code editor. As a result, this software environment is accessible enough to allow Poppy robots to be currently used as educational platforms in several middle and high schools[2].

Pypot also includes features to operate a virtual robot within the simulator V-REP (Freese, 2015), as illustrated in **Figure 2**. In this way, users can experiment and verify their developments on a simulated Reachy before deploying them on an actual robot in a physical setup. Migrating from a simulated to an actual robot, and vice versa, does not require any modification on the source code apart from a single keyword while configuring the connection to a robot.

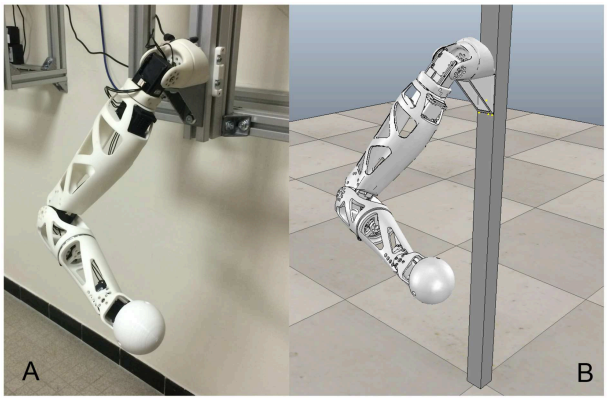


FIGURE 2 | An actual Reachy robot **(A)** and its simulated counterpart **(B)**, set in the same posture.

2.4. Features

2.4.1. Motor Performance

Reachy's motors can sustain up to 10 min of continuous operation and are able to work for as long as a full day when tasked with short, out-of-charge movements alternating with short resting periods. They provide a payload capacity of about 500 g at endpoint level, that the robot can handle for a few minutes. Their embedded load and temperature sensors also allow to automatically trigger resting phases, in order to prevent overheating during prolonged operation or after exposing the robot to significant strain. As a result, Reachy can be programmed to work autonomously for extended periods of time without putting the actuators at risk.

Out of charge, Dynamixel motors can reach a maximum speed of 500°/s and a maximum acceleration of 10,000°/s^2. When they operate in their nominal angular speed range, their performance allows the robot's joint to reach their goal positions with a delay from 50 to 100 ms. This responsiveness makes it thus possible to develop real-time control schemes within which a human is continuously driving the robot. As a consequence, the robot can move its endpoint safely at a speed up to 2 m/s, with an acceleration of 10 m/s^2.

Thanks to its three DoF at shoulder level, Reachy's full-length version can perform complex movements and postures in a wide range of action in the 3D space. As a result, Reachy benefits from having a workspace similar to that of a human adult's arm, within a 65 cm-radius sphere centered on its shoulder.

2.4.2. Application in Prosthetics

Thanks to its human-like shape and joints, Reachy is suitable for applications in the field of upper-limb rehabilitation engineering, as a life-size test platform. Indeed, Reachy can emulate the behavior of a prosthetic arm in order to test and validate control schemes before implementing them on a genuine prosthesis. In this context, it also benefits from being notably cheaper than most commercially available upper-limb prostheses, thanks to its hardware architecture.

Indeed, 3D-printing technology has already been employed to create numerous arm prosthesis prototypes, whose designs are being developed by creators ranging from DIY enthusiasts and hobbyists, to researchers and engineers, as detailed in ten Kate et al. (2017). The fact that more than half of these 3D-printed devices' designs are shared online and available for free, shows how these creators take advantage of the interoperability of most desktop printers. This review also highlights how the production cost of these devices is one of the decisive aspects that sparkled the growth of this category of prosthetic devices, so much so

[2]http://perseverons.espe-aquitaine.fr/sp6-robotique-inria/

that some 3D-printed prosthetic arms have recently went beyond the prototype stage and entered the market, such as the Hero Arm (OpenBionics).

Compared to the devices listed in this review, Reachy stands as one of the few models to address amputation above the elbow. Additionally, even though the predominance of transradial amputations among upper-limb disabilities explains the rarity of this type of prosthesis, Reachy is intended to enable research at multiple amputation levels. Indeed, the robot can be employed as a mockup device for any level of upper-limb amputation, when training a patient to produce muscle activity before being fit with a myoelectric prosthesis. This allows a patient to begin training even before being able to wear a prosthesis, e.g., while the stump is still cicatrizing. Obviously, such a training cannot replace experience with an actual prosthesis, especially because of the differences in terms of point of view, embodiment and perception of weight and inertia. Nonetheless, it can take place in a patient's rehabilitation as a complementary or preliminary training, with the aim of getting familiar with myoelectric control as well as motors' responsiveness and accuracy.

In this context, the patient's lost motor functions are emulated with Reachy's corresponding joints while the robot's more proximal actuators are locked in a given posture. In this way, the patient can practice performing appropriate muscle contractions and receive relevant feedback from the robot moving accordingly, following a given prosthesis control scheme. In a more advanced setup, the patient's residual limb movements can even be tracked and reproduced on these motors, instead of being locked. Such a setup could turn out to be useful as well for testing control strategies using residual limb motion as input signal to drive the prosthesis (Kaliki et al., 2013; Merad et al., 2016).

Regarding the end-effector, in respect to grasping with an arm prosthesis, a fixed wrist often requires the patient to perform extra shoulder and elbow movements to compensate for the lack of distal mobility. Thus, enabling wrist motion proves to be quite useful for a patient (Kanitz et al., 2018), as it enables a more natural and comfortable use. In this way, Reachy's 2-DoF wrist makes it suitable to address this aspect of prosthesis control. In combination with forearm rotation, these motor functions at wrist level allow the robot to put its endpoint in a wide variety of 3D orientations, enabling different grasping types depending on the item of interest.

Finally, Reachy's customizable architecture allows users to design, print and assemble custom fixings, so that a part of the robot can be mounted on an actual prosthetis socket or harness, and worn by an amputee (see **Figure 3**). Whether at transradial or transhumeral level, the robot's skeleton parts can be modified so that its dimensions are adjusted to the wearer's morphology, to fit best with the stump's anatomy as well as the sound limb's proportions. Obviously, Reachy is not meant to replace a prosthesis for daily use, but a socket-mounted version of Reachy could as well be employed for training patients with myoelectric control.

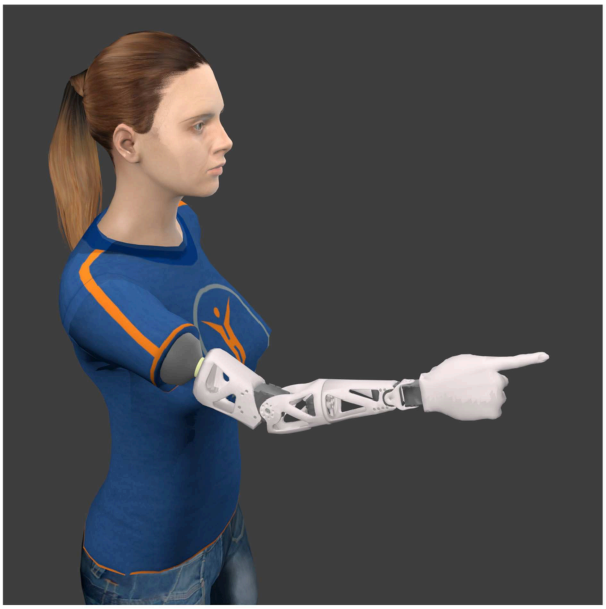


FIGURE 3 | Virtual illustration of a possible evolution of Reachy: socket-mounted version worn by a transhumeral amputee.

2.5. Comparison With Existing Robotic Arms

In order to put Reachy's performance and abilities in perspective with existing robotic devices, we compared it with several related robotic arms, considering various aspects and features.

The Bento Arm (Dawson et al., 2014) is a robotic arm employed in research and upper-limb rehabilitation to emulate a myoelectric prosthesis. It includes five joints from humerus to end-effector, each actuated by a Dynamixel motor. Its mechanical structure follows human-like shapes and proportions, and relies mostly on 3D-printed plastic parts. In this sense, this robot is very similar to Reachy, although the upper arm includes only a single DoF at humeral level, and none at shoulder level. As a consequence, its workspace is limited to a 22 cm-wide circular area centered on the elbow, for a payload capacity of only 300 g. This narrow range of motion and limited upper-arm actuation make the Bento Arm unsuitable for research on whole-arm movements, unless it is mounted on a prosthesis socket. As a rehabilitation device, it focuses on emulating a transradial prosthesis but appears to be inappropriate with respect to higher levels of amputation. In particular, it cannot be employed to study or reproduce coordinations between upper-arm joints.

The GummiArm (Stoelen et al., 2016) is an experimental bio-inspired robotic arm comprising 10 tendon-driven joints, actuated by 19 Dynamixel motors. With eight of its joints enabling variable stiffness, this robot can perform movements in a workspace similar to that of a human arm while being safe to physically interact with. Similarly to Reachy, its 3D-printed skeleton parts and open-source approach allow for replication and modification by users. However, its higher number of actuators and tendon-based mechanics make it more expensive

(over 5,000$ in spare parts, over 11,500$ as a kit) as well as more suitable for research on bio-inspired actuation and compliant motor control than on rehabilitation engineering. Indeed, most upper-arm, elbow and forearm prostheses are actuated by a single motor per joint in a non-compliant fashion, instead of emulating human biomechanical structures.

The Modular Prosthetic Limb (MPL) (Johannes et al., 2011) is an experimental robotic upper-limb prosthesis, comprising 26 joints actuated by 17 motors. Its high-grade components and anthropomorphic design allow it to reach human-like strength in the wide range of motion offered by its joints. Compared to Reachy, this robot offers much better motor performance, such as a payload capacity of 15 kg and a joint speed of 120°/s. However, these abilities also come with a higher power consumption (24 vs. 5 V for Reachy) and a heavier weight (4.7 vs. 1.2 kg for Reachy).

The DLR Hand Arm System (Grebenstein et al., 2011) is an experimental bio-inspired robotic arm, now integrated to the humanoid robot David as its upper limb. It includes six DoF in the arm and 19 in the hand, actuated by a total of 52 motors. Its tendon-driven mechanical structure allows the robot to operate dexterously at a speed and in a workspace comparable to those of a human, making it clearly more capable than Reachy in terms of motor performance. However, its bidirectional antagonist joints require numerous motors and mechanical components, a dedicated transmission architecture and a dense electronics network managing both actuation and sensing.

Due to their price and complexity in terms of electronics and mechanics, these advanced robotic devices are much more difficult to replicate or customize in depth. In this sense, their users depend significantly on the robot's designers and makers to assemble, modify and repair it, whereas Reachy's design allows users to handle every step of the fabrication process. Its architecture is simple enough to allow non-experts to build it and connect it to a computer. Regarding control and interfacing, both of these robots rely on complex control architectures (Bridges et al., 2011; Grebenstein et al., 2011) running in Simulink, proprietary software owned by MathWorks, Inc. In this regard, the pieces of software operating these devices are less open and more difficult for a user to modify or adapt to a given use case. Conversely, Reachy benefits from its open-source software architecture providing many interfacing options, with a variety of command signals and external software tools.

Although Reachy does not compare to these advanced robots in terms of performance, its connectability and highly modifiable structure make it a suitable research platform. In this sense, we wish to promote Reachy as a complete platform combining both hardware and software characteristics fostering replication, customization and versatility. We are not aware of a similar robotic system that would offer as many possibilities, based on the comparison detailed in this section.

2.6. Sharing Philosophy

Reachy is developed in partnership with and distributed by Pollen Robotics[3] as a fully open-source project. Besides, users willing to assemble the robot by themselves can buy all the hardware in spare parts, at a total price below 4,000$ for the standard version. Any laboratory can build their own Reachy, modify its components and customize it at will, on both hardware and software sides. This allows researchers to adapt the robot to their specific needs and interface it with their own devices and tools.

The source files from the Computer-Aided Design (CAD) models of the different printable parts are shared under the Creative Commons BY-SA license[4] and available online[5]. The bill of materials and software components that are specific to Reachy are shared under the Lesser GNU General Public License[6] and available online in the project's repository[7]. The Pypot library is shared under the GNU General Public License[8] and available online in a dedicated repository[9].

As Reachy relies on the same software and hardware architecture as Poppy robots, it is worth noting that its users can benefit from the help and contributions shared by the Poppy project's community on its repository[10]. Indeed, this community hub gives access to many tips regarding the different aspects of the robot, from configuring and assembling the servomotors to setting up the software tools and troubleshooting.

3. PROOFS OF CONCEPT

In order to illustrate Reachy's interfacing capabilities, we developed several proofs of concept where the robot's features are combined or expanded with various external devices and software tools. All the proofs of concept described below were developed in Python, to further demonstrate the interfacing potential provided by this language.

3.1. Inverse Kinematics for Endpoint Position Control

Determining a set of motor angles that put a robot's endpoint at a target position in its operational space is a common problem in the field of robotic arms, and is usually referred to as the *Inverse Kinematics* (IK) problem. As it comprises seven independent DoF, Reachy typically displays *kinematic redundancy*, implying that there is an infinite number of distinct solutions to this problem for each reachable target position. Thus, in order to drive the robot's endpoint position to a given target, one needs to determine which set of angles to apply, among the infinity of possible sets. However, the numerical expression of this under-constrained geometrical problem is non-linear, which makes analytic solving impractical and costly in terms of computation.

3.1.1. Local Optimization

Instead, a widespread method used by roboticists to solve IK problems is to employ local optimization. This method relies on

[3]https://www.pollen-robotics.com/en/our-products/humanoid-robot-arm-reachy/

[4]http://creativecommons.org/licenses/by-sa/4.0/

[5]http://cad.onshape.com/documents/66388ae9c63cef53d76acd77/w/68c2411483d5bc65c7f54234/e/581d46ba9b8ee98de9d636ee

[6]http://gnu.org/licenses/lgpl-3.0.html

[7]http://github.com/pollen-robotics/reachy

[8]http://gnu.org/licenses/gpl.html

[9]http://github.com/poppy-project/pypot

[10]http://github.com/poppy-project

a cost function, attributing a scalar value to any set of angles to quantify to what extent it is a good solution with respect to the IK problem: a lower cost means a better solution. Usually, this cost function is based on the distance between the target and the endpoint position, which can be analytically determined with the geometrical model of the robot. Then, through a step-by-step process, the optimization finds and returns a local minimum of this cost function, that should correspond to one of the sets of angles putting the endpoint at the required position.

We used the Python library IKPy (Manceron, 2015), a generic IK solver, to apply this method on Reachy. The robot's software resources include a Universal Robot Description Format (URDF) file describing Reachy's mechanical properties, such as the relative position and orientation of each joint and skeleton part. These geometrical data can then be imported with IKPy to build the corresponding kinematic chain, by going through the sequence of joints from the robot's base to its endpoint. In this way, this interfacing between IKPy and Reachy's software interface can be performed straight out of the box, and works as a standalone, without requiring any external device or specific hardware. As a result, combining IKPy's features with the motors' command options provides a new and easy way to control Reachy by sending 3D coordinates as commands, instead of joint angles. A code sample showing how to use IKPy with Reachy is available online[11].

IKPy allows to set parameters for the optimization process (e.g., maximum iteration number, convergence tolerance) when calling it from another program. Thanks to these options available in the code, users can fine-tune the process in respect to the intended use case and available computing power. As an example, after fine-tuning our setup through a trial-and-error process, the model was able to reach a sub-centimetric accuracy with a computing time below 100 ms on a desktop computer. However, the kinematic chain employed with this method is a **theoretical** model of the robot and does not take into account the robot's weight and joints' mechanical play. On a physical robot, as actual motors are unable to reach the *exact* angular positions determined by IKPy, the endpoint tends to undershoot when driven with this method. To assess endpoint accuracy, the distance between the endpoint's actual position and its target was measured for eighty postures distributed in the robot's range; each measurement was performed after the robot moves for 1.5 s then is asked to hold the posture for 3.5 s. We obtained a mean distance to target of 87 mm (SD 23 mm), and also observed that the endpoint usually reaches positions located under the target. Indeed, position errors along the two cartesian horizontal axes are roughly centered on zero (mean < 5 mm) whereas along the vertical axis, this error is subject to a notable offset (mean $= 84$ mm).

Nevertheless, this flaw is not blatantly noticeable if no visible object materializes the target position in the operational space. Besides, the vertical offset proves to be fairly consistent over time and reachable space. Therefore, in the context of a continuous endpoint position control, it can be dealt with during a calibration phase performed prior to the control phase. In this way, this interfacing between IKPy and Reachy can be conveniently employed in applications where there is no strong need for endpoint accuracy in the operational space.

To assess repeatability, the robot was tasked to perform several times the same movement while the endpoint's position was recorded with a motion tracking system (Optitrack V120 Trio, Natural Point Inc.). Firstly, the robot was tasked to travel accross a 40 cm-wide circle in a frontal plane, in 3.5 s. A comparison of the recorded trajectories showed that on keyframes distributed along the movement, for a given set of motor goals, the robot's resulting endpoint positions were spread within a 12 mm-radius sphere. Then, the robot was tasked to reach a given posture and hold it for several seconds before its endpoint position was recorded. Over ten repetitions of this movement, the positions were spread within a 5 mm-radius sphere. These results illustrate Reachy's ability to reach the same point in space in response to the same motor commands, in both static and dynamic contexts.

3.1.2. Supervised Learning With an Artificial Neural Network

On another hand, this gap between a theoretical model and Reachy's actual functioning can be reduced by employing modeling techniques that do not intend to simulate the robot's *ideal* behavior. One of them consists in recording actual movements performed by the robot and using them as "ground truth" examples on which to perform supervised learning. The goal is to build a set of movements where both motor angles and endpoint coordinates are synchronously recorded, so that a supervised learning algorithm can emulate the actual relationship between these two quantities.

To apply this technique with Reachy, we first defined a set of robot postures through physical demonstration: with its motors set as compliant, the robot was manually placed in various configurations while embedded sensors recorded the joints' angles. Then, based on the recorded angles, the robot performed movements going from one of such demonstrated postures to another, while the Optitrack V120 recorded the actual endpoint position. We used an Artificial Neural Network (ANN) to perform supervised learning on the captured joint and endpoint data. ANNs are computational tools relying on elementary logical units called "neurons" and connected between them by weighted links, generally following a specific network architecture (Reed and Marks, 1998). For several decades, these tools have been used to perform supervised learning by tuning the weights of these links based on the training data. In the field of robotics, ANNs are typically employed to perform environment sensing or effector control, including IK solving (Bouganis and Shanahan, 2010; Duka, 2014; Almusawi et al., 2016).

Our results were obtained with a feed-forward multi-layer perceptron including two fully connected hidden layers of, respectively 64 and 128 neurons. We employed the TensorFlow (Abadi et al., 2015) backend and Keras (Chollet, 2015), a Python programming interface for ANNs to implement and train this network to perform IK solving, that is to say: take endpoint coordinates as input and return corresponding joint angles as output.

[11] http://github.com/pollen-robotics/reachy/blob/master/doc/notebook/Kinematics.ipynb

Relatively to using IKPy, implementing this method is more demanding and requires to carry out the previously described two-phase data acquisition process using motion capture equipment. However, as this technique is based on movements performed by the physical robot instead of a mechanically perfect model, the ANN implicitly takes into account the deviations between the motor commands sent to the robot and the angles actually reached by the motors. As a result, this method proved to be more accurate than the local optimization method with an actual robot (mean distance to target = 25 mm, SD 11 mm). In particular, it does not suffer from the aforementioned vertical offset, as the position error along the vertical axis isn't more off-centered than along horizontal axes (mean < 14 mm for all three axes). On another hand, the computing time required to perform a single IK solving with this method remained consistently under 1ms, proving it to be much faster than local optimization.

Besides, building the training set through manual demonstration of postures allows users to deliberately introduce a bias in favor of a certain type of posture. In this way, such a bias would be implicitly learned and emulated by the network, as its output would be, by design, similar to the training set's postures. For instance, if one only records postures with a horizontal hand and palm facing downwards, virtually all joint angles returned by the network should correspond to postures displaying that same characteristic.

Regarding network structure, we noticed that adding more hidden layers or increasing their size does not draw significant benefits and can even result in the network overfitting the examples, while notably increasing the time required to train it. Based on these observations, we hypothesize that more complex network architectures, such as convolutional or recurrent networks, may not be appropriate for the solving of this IK problem.

As a conclusion on the topic of IK solving for Reachy, we presented here two methods with notable differences regarding accuracy, practicality or convenience. These methods also illustrate how Reachy benefits from being connectable and customizable, in the way that various solutions can be employed to provide similar features, so that users can choose a solution suitable for their needs. Other approaches could be employed to perform endpoint position control, either based on existing techniques from the literature, or developed *ad hoc* with more specific requirements.

3.2. Tele-operation

Based on the endpoint position control feature made available by these IK solving techniques, we developed a second proof of concept, which we refer to as "tele-operation." The goal of this proof of concept is to provide users with an intuitive and transparent way to drive the robot in real time, that would not require them to send explicit, quantitative commands such as joint angles or endpoint coordinates. With this aim, the tele-operation driving mode works by continuously tracking a subject's hand trajectory and simultaneously mapping it on the robot's hand, considered its endpoint.

Our implementation of this driving mode makes use of the Optitrack V120 Trio as the motion tracking system, to determine the 3D position of a marker placed on the hand. We interfaced the device with a Python program using OptiRX (Astanin, 2016) to retrieve marker data in real time at 120 Hz. Before the control phase, a calibration is performed to set the relation between the subject's reference frame, in which marker data is expressed, and the robot's reference frame, in which endpoint target coordinates must be expressed. Then, both the robot and subject's arms are placed in the same initial posture: humerus along the body and elbow flexed at 90° (see **Figure 4**). In this posture, the subject's and robot's hand positions are saved in order to work as origin points in their respective frames. At each instant of a 10 Hz control loop, the former is used to compute the instantaneous displacement vector of the subject's hand, then the latter is used to compute the robot's hand target position, by mapping this vector in the robot's frame. Finally, using an IK solving method, Reachy is put in motion so that its endpoint goes toward this instantaneous target.

As a result, the subject can drive the robot by performing natural arm movements, observing how Reachy mirrors them and using this visual feedback to adjust the robot's motion. Obviously, the processing time as well as the fact that the motors cannot instantly reach the goal angles sent as commands, introduce a lag between its endpoint's movement and the subject's hand trajectory. In the current setup, this lag is usually comprised between 350 and 450 ms. This proof of concept illustrates how one can implement a control strategy with Reachy, that is: a way to put it in motion based on data acquired by external devices. It also demonstrates how Reachy can be controlled in a real-time fashion, while performing smooth and steady movements.

A video clip showing a subject driving the robot in tele-operation mode is available online[12].

3.3. Gaze-Driven Control

Following on from vision-based assistive devices, we developed a second proof of concept to explore how eye movements and gaze behavior could be employed as a source of commands to put Reachy in motion. With this aim, we tried to interface the robot with eye-tracking and image processing tools, in order to allow a subject to drive Reachy by moving only their eyes instead of their limbs. Eye tracking is a category of techniques aiming at measuring eye movements or gaze direction, whether for observation purposes or as input in an interactive setup (Duchowski, 2003). In the field of robotics, eye-tracking techniques have recently been employed to control robotic arms, especially with applications in rehabilitation and assistive technologies (Frisoli et al., 2012; McMullen et al., 2014; Hortal et al., 2015).

The resulting setup relies on a camera filming a scene in front of the robot, and a computer screen displaying its video stream to a subject. The camera is placed so that the scene matches with the robot's reachable space, and hand-sized objects of various colors and shapes are located within its range. They are placed so that no visual occlusion occurs from the point of

[12]https://www.youtube.com/watch?v=Oa9mHMoDtYI

FIGURE 4 | Tele-operation setup, shown during the calibration phase. Subject and robot are placed in the same posture while the Optitrack system (on the left) records the coordinates of the reflecting marker placed on the subject's hand. See this driving mode in operation at https://www.youtube.com/watch?v=Oa9mHMoDtYI.

view of the camera, and no physical obstruction occurs when the robot moves its endpoint toward an object. In this setup, the screen acts as a 2-dimensional proxy between the robot's operational space and the subject's field of view, in order to use eye-tracking technology in a simpler context than 3D space. We employed the GP3 HD eye tracker (Gazepoint) to locate the focus of the subject's gaze on the plane of the screen, and identify the corresponding object in the robot's reaching space (see **Figure 5**). Then, the robot can be put in motion toward this object's position, either using pre-recorded postures, or a combination of inverse kinematics and computer vision in the scene in front of the robot.

Finally, using a Myo armband (Thalmic Labs Inc.), we also integrated a basic form of myoelectric control to this setup. This measuring device allows for the detection of a specific muscle activation pattern, that can be interpreted as a command signal. In this way, the subject can perform, for instance, a voluntary co-contraction of forearm muscles to trigger a movement by the robot. Whenever such a signal is detected, the object on which the subject's gaze is focused is identified by the eye-tracking system, and Reachy is put in motion accordingly.

It is worth noting that the processing of gaze data performed to identify the object of interest remains very basic in this simplified setup. In a daily life context, the subject's posture would be unconstrained. Furthermore, the environment the subject acts in, performing its instrumental activities of daily living, is cluttered. The distractors and scene changes provoke saccades. When maintaining gaze on the target object, the geometry in a dynamic scene is also unstable due to micro-saccades. This is why a filtering of gaze fixation signal along the time is needed (González-Díaz et al., 2019). Moreover, today a localization of objects in a gaze-predicted area can be solved together with an object-recognition task, employing powerful deep CNN classifiers. This allows for more precise object localization and also adaptation to the scene dynamics due to the unconstrained motion of the subject.

The source code employed in this proof of concept is available online[13]. A video clip showing a subject performing gaze-driven control is available online[14].

4. CONCLUSION AND PERSPECTIVES

Reachy, a seven-DoF human-like robotic arm, was developed to act as a test platform for research on human-driven robotic arms. Following an open-source approach, its design was elaborated to allow for easy sharing and low fabrication cost, with the purpose of enabling extensive customization in a wide variety of applications. Software and hardware resources were made available online so that researchers and laypeople can build a Reachy robot and integrate it in their own experiments and projects.

In the short term, immediate applications of Reachy include the exploration, development and testing of control strategies and interfaces for robotic arms. In this way, several prototypes were produced and proofs of concepts were developed in order to illustrate potential use cases in various fields in relation with human-driven robotics. As a broadly connectable platform, it allows to investigate hybrid control strategies, combining biomechanical signals with motion- or eye-tracking tools and computer vision techniques (de San Roman et al., 2017; González-Díaz et al., 2019). Reachy can also help study how vision-based control strategies would help driving rehabilitation

[13]http://github.com/pollen-robotics/reachy/blob/master/applications/video_gaze.py

[14]https://www.youtube.com/watch?v=qloR67AaqQ4

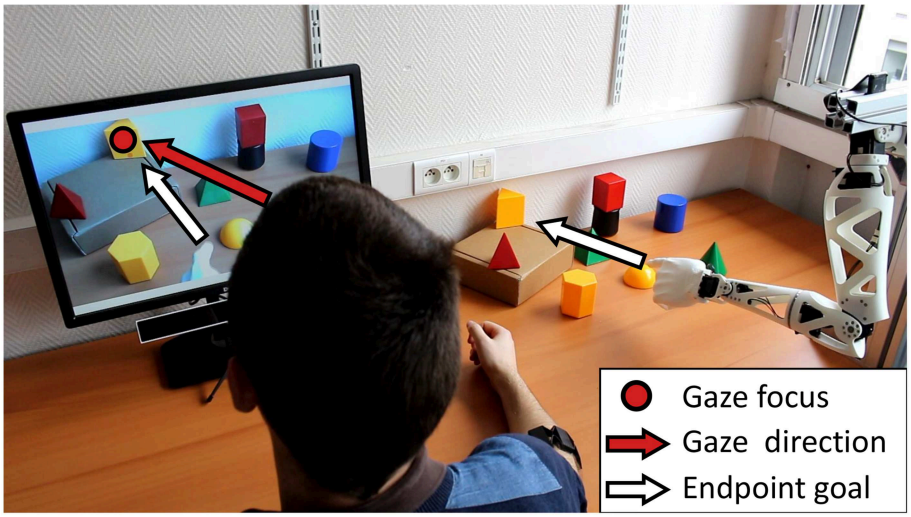


FIGURE 5 | Gaze-driven control setup. On the right, various objects are located in front of a left-handed version of Reachy, and the resulting scene is filmed by the camera placed over the robot's shoulder. Its video feed is shown on the screen on the left, under which the eye-tracker is placed. The subject wears the Myo armband on the right forearm to trigger a movement by the robot. See this driving mode in operation at https://www.youtube.com/watch?v=qloR67AaqQ4.

devices, such as an assistive arm fixed to a wheelchair, for use by patients suffering from Spinal Cord Injury (SCI) (Corbett et al., 2013, 2014).

On the longer term, Reachy can be employed as a mockup device for research and training with upper-limb neuroprostheses. In particular, it can help patients get familiar with muscle signal production and myoelectric control prior to being fit with an actual arm prosthesis. Additionally, thanks to its motors' control options, Reachy is suitable to address different levels of amputation, by employing separate control modes to drive proximal and distal joints. For instance, as a way to emulate transhumeral amputation, Reachy can be controlled through a "hybrid" teleoperation mode where shoulder joints reproduce a patient's actual shoulder motion while the other motors are driven with a separate, artificial control strategy. Similar approaches were investigated in recent works (Kaliki et al., 2013; Merad et al., 2016), where natural shoulder motion (performed by a subject) is used to infer artificial elbow and/or wrist motion (performed by a virtual avatar or a wearable prosthesis). As a general conclusion, Reachy can prove to be a versatile device suitable for applications with multiple approaches for the control of an upper-limb neuroprosthesis.

DATA AVAILABILITY

All datasets generated for this study are included in the manuscript and/or the **Supplementary Files**.

AUTHOR CONTRIBUTIONS

SM participated in the development of proofs of concept under the supervision of P-YO and AR, and wrote the paper. ML and PR designed the robot and participated in the development of proofs of concept. CH participated in the development of proofs of concept. JB-P, FP, and DC provided inputs for paper writing. All authors approved the final version.

FUNDING

This work was funded by two PEPS CNRS/IdEx Bordeaux grants (2014 and 2015), and a CNRS Osez l'interdisciplinarité grant (2017–2019).

ACKNOWLEDGMENTS

The authors would like to thank Lionel Parra-Iglesias for designing and printing skeleton parts, Philippe Chauvet for designing and assembling the robot's support, and Lucas Dure for creating **Figure 4**.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2019.00065/full#supplementary-material

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). *TensorFlow: Large-scale Machine Learning on Heterogeneous Systems.*

Almusawi, A. R., Dülger, L. C., and Kapucu, S. (2016). A new artificial neural network approach in solving inverse kinematics of robotic arm (Denso VP6242). *Comput. Intell. Neurosci.* 2016:5720163. doi: 10.1155/2016/5720163

Astanin, S. (2016). *A Pure Python Library to Receive Motion Capture Data From OptiTrack Streaming Engine*.

Bae, J.-H., Park, J.-H., Oh, Y., Kim, D., Choi, Y., and Yang, W. (2015). Task space control considering passive muscle stiffness for redundant robotic arms. *Intell. Serv. Robot.* 8, 93–104. doi: 10.1007/s11370-015-0165-2

Belter, J. T., and Dollar, A. M. (2013). "Novel differential mechanism enabling two DoF from a single actuator: application to a prosthetic hand," in *2013 IEEE International Conference on Rehabilitation Robotics (ICORR)* (Seattle, WA: IEEE), 1–6.

Biddiss, E. A., and Chau, T. T. (2007). Upper limb prosthesis use and abandonment: a survey of the last 25 years. *Prosthet. Orthot. Int.* 31, 236–257. doi: 10.1080/03093640600994581

Blana, D., Kyriacou, T., Lambrecht, J. M., and Chadwick, E. K. (2016). Feasibility of using combined EMG and kinematic signals for prosthesis control: a simulation study using a virtual reality environment. *J. Electromyogr. Kinesiol.* 29, 21–27. doi: 10.1016/j.jelekin.2015.06.010

Bouganis, A., and Shanahan, M. (2010). "Training a spiking neural network to control a 4-DoF robotic arm based on spike timing-dependent plasticity," in *2010 International Joint Conference on Neural Networks (IJCNN)* (Barcelona: IEEE), 1–8.

Bridges, M. M., Para, M. P., and Mashner, M. J. (2011). Control system architecture for the Modular Prosthetic Limb. *Johns Hopkins APL Techn. Digest* 30, 217–222.

Chollet, F. (2015). *Keras*. Available online at: https://github.com/fchollet/keras

Cipriani, C., Controzzi, M., and Carrozza, M. C. (2011). The SmartHand transradial prosthesis. *J. Neuroeng. Rehabil.* 8:29. doi: 10.1186/1743-0003-8-29

Corbett, E. A., Körding, K. P., and Perreault, E. J. (2013). Real-time evaluation of a noninvasive neuroprosthetic interface for control of reach. *IEEE Trans. Neural Syst. Rehabil. Eng.* 21, 674–683. doi: 10.1109/TNSRE.2013.2251664

Corbett, E. A., Sachs, N. A., Körding, K. P., and Perreault, E. J. (2014). Multimodal decoding and congruent sensory information enhance reaching performance in subjects with cervical spinal cord injury. *Front. Neurosci.* 8:123. doi: 10.3389/fnins.2014.00123

Cordella, F., Ciancio, A. L., Sacchetti, R., Davalli, A., Cutti, A. G., Guglielmelli, E., et al. (2016). Literature review on needs of upper limb prosthesis users. *Front. Neurosci.* 10:209. doi: 10.3389/fnins.2016.00209

Dawson, M. R., Sherstan, C., Carey, J. P., Hebert, J. S., and Pilarski, P. M. (2014). Development of the Bento Arm: an improved robotic arm for myoelectric training and research. *Proc. MEC* 14, 60–64. doi: 10.13140/2.1.3118.4640

de San Roman, P. P., Benois-Pineau, J., Domenger, J.-P., Paclet, F., Cattaert, D., and de Rugy, A. (2017). Saliency driven object recognition in egocentric videos with deep CNN: toward application in assistance to neuroprostheses. *Comput. Vis. Image Understand.* 164, 82–91. doi: 10.1016/j.cviu.2017.03.001

Duchowski, A. T. (2003). *Eye Tracking Methodology: Theory and Practice*. London, UK: Springer.

Duka, A.-V. (2014). Neural network based inverse kinematics solution for trajectory tracking of a robotic arm. *Proc. Technol.* 12, 20–27. doi: 10.1016/j.protcy.2013.12.451

Freese, M. A. (2015). *Virtual Robot Experimentation Platform*. Zürich: Coppelia Robotics.

Frisoli, A., Loconsole, C., Leonardis, D., Banno, F., Barsotti, M., Chisari, C., et al. (2012). A new gaze-bci-driven control of an upper limb exoskeleton for rehabilitation in real-world tasks. *IEEE Trans. Syst. Man Cybern. Part C* 42, 1169–1179. doi: 10.1109/TSMCC.2012.2226444

Gigli, A., Gijsberts, A., Gregori, V., Cognolato, M., Atzori, M., and Caputo, B. (2017). Visual cues to improve myoelectric control of upper limb prostheses. *arXiv[Preprint]. arXiv:1709.02236*. doi: 10.1109/BIOROB.2018.8487923

González-Díaz, I., Benois-Pineau, J., Domenger, J.-P., Cattaert, D., and de Rugy, A. (2019). Perceptually-guided deep neural networks for ego-action prediction: object grasping. *Pattern Recogn.* 88, 223–235. doi: 10.1016/j.patcog.2018.11.013

Grebenstein, M., Albu-Schäffer, A., Bahls, T., Chalon, M., Eiberger, O., Friedl, W., et al. (2011). "The DLR hand arm system," in *2011 IEEE International Conference on Robotics and Automation* (Shanghai: IEEE), 3175–3182.

Ha, I., Tamura, Y., Asama, H., Han, J., and Hong, D. W. (2011). "Development of open humanoid platform DARwIn-OP," in *2011 Proceedings of SICE Annual Conference (SICE)* (Tokyo: IEEE), 2178–2181.

Hauschild, M., Davoodi, R., and Loeb, G. E. (2007). A virtual reality environment for designing and fitting neural prosthetic limbs. *IEEE Trans. Neural Syst. Rehabil. Eng.* 15, 9–15. doi: 10.1109/TNSRE.2007.891369

Hild, M., Siedel, T., Benckendorff, C., Thiele, C., and Spranger, M. (2012). "Myon, a new humanoid," in *Language Grounding in Robots*, eds L. Steels and M. Hild (New York, NY: Springer), 25–44.

Hortal, E., Iáñez, E., Úbeda, A., Perez-Vidal, C., and Azorín, J. M. (2015). Combining a brain–machine interface and an electrooculography interface to perform pick and place tasks with a robotic arm. *Robot. Auton. Syst.* 72, 181–188. doi: 10.1016/j.robot.2015.05.010

Johannes, M. S., Bigelow, J. D., Burck, J. M., Harshbarger, S. D., Kozlowski, M. V., and Van Doren, T. (2011). An overview of the developmental process for the Modular Prosthetic Limb. *Johns Hopkins APL Techn. Digest* 30, 207–216.

Kaliki, R. R., Davoodi, R., and Loeb, G. E. (2013). Evaluation of a noninvasive command scheme for upper-limb prostheses in a virtual reality reach and grasp task. *IEEE Trans. Biomed. Eng.* 60, 792–802. doi: 10.1109/TBME.2012.2185494

Kanitz, G., Montagnani, F., Controzzi, M., and Cipriani, C. (2018). Compliant prosthetic wrists entail more natural use than stiff wrists during reaching, not (necessarily) during manipulation. *IEEE Trans. Neural Syst. Rehabil. Eng.* 26, 1407–1413. doi: 10.1109/TNSRE.2018.2847565

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., et al. (2016). "Jupyter notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, eds L. Fernando and S. Birgit (Göttingen: IOS Press), 87–90.

Krausz, N. E., Rorrer, R. A., and Weir, R. F. (2016). Design and fabrication of a six degree-of-freedom open source hand. *IEEE Trans. Neural Syst. Rehabil. Eng.* 24, 562–572. doi: 10.1109/TNSRE.2015.2440177

Lapeyre, M., Rouanet, P., Grizou, J., Nguyen, S., Depraetre, F., Le Falher, A., and Oudeyer, P.-Y. (2014). "Poppy project: open-source fabrication of 3D-printed humanoid robot for science, education and art," in *Digital Intelligence 2014* (Nantes), 6.

Lapeyre, M., Rouanet, P., and Oudeyer, P.-Y. (2013). "Poppy humanoid platform: experimental evaluation of the role of a bio-inspired thigh shape," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (Atlanta, GA: IEEE), 376–383.

Losier, Y., Clawson, A., Wilson, A., Scheme, E., Englehart, K., Kyberd, P., et al. (2011). "An overview of the UNB hand system," *Presented at the Myoelectric Controls Symposium* (Fredericton, NB).

Ly, O., Lapeyre, M., and Oudeyer, P.-Y. (2011). "Bio-inspired vertebral column, compliance and semi-passive dynamics in a lightweight humanoid robot," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (San Francisco, CA: IEEE), 1465–1472.

Manceron, P. (2015). *An Inverse Kinematics Library Aiming Performance and Modularity*.

Markovic, M., Dosen, S., Cipriani, C., Popovic, D., and Farina, D. (2014). Stereovision and augmented reality for closed-loop control of grasping in hand prostheses. *J. Neural Eng.* 11:046001. doi: 10.1088/1741-2560/11/4/046001

Markovic, M., Dosen, S., Popovic, D., Graimann, B., and Farina, D. (2015). Sensor fusion and computer vision for context-aware control of a multi degree-of-freedom prosthesis. *J. Neural Eng.* 12:066022. doi: 10.1088/1741-2560/12/6/066022

Markovic, M., Karnal, H., Graimann, B., Farina, D., and Dosen, S. (2017). GLIMPSE: Google Glass interface for sensory feedback in myoelectric hand prostheses. *J. Neural Eng.* 14:036007. doi: 10.1088/1741-2552/aa620a

McMullen, D. P., Hotson, G., Katyal, K. D., Wester, B. A., Fifer, M. S., McGee, T. G., et al. (2014). Demonstration of a semi-autonomous hybrid brain–machine interface using human intracranial EEG, eye tracking, and computer vision to control a robotic upper limb prosthetic. *IEEE Trans. Neural Syst. Rehabil. Eng.* 22, 784–796. doi: 10.1109/TNSRE.2013.2294685

Meeker, C., Rasmussen, T., and Ciocarlie, M. (2018). Intuitive hand teleoperation by novice operators using a continuous teleoperation subspace. *arXiv[Preprint] arXiv:1802.04349*. doi: 10.1109/ICRA.2018.8460506

Merad, M., de Montalivet, E., Roby-Brami, A., and Jarrassé, N. (2016). "Intuitive prosthetic control using upper limb inter-joint coordinations and IMU-based shoulder angles measurement: a pilot study," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon: IEEE), 5677–5682.

Novak, D., and Riener, R. (2015). A survey of sensor fusion methods in wearable robotics. *Robot. Auton. Syst.* 73, 155–170. doi: 10.1016/j.robot.2014.08.012

Phelan, I., Arden, M., Garcia, C., and Roast, C. (2015). "Exploring virtual reality and prosthetic training," in *Virtual Reality (VR), 2015* (Arles: IEEE), 353–354.

Rakita, D., Mutlu, B., and Gleicher, M. (2017). "A motion retargeting method for effective mimicry-based teleoperation of robot arms," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction* (Vienna: ACM), 361–370.

Rakita, D., Mutlu, B., and Gleicher, M. (2018). "RelaxedIK: Real-time synthesis of accurate and feasible robot arm motion," in *Proceedings of Robotics: Science and Systems* (Pittsburgh, PA).

Reed, R. D., and Marks, R. J. (1998). *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA: MIT Press.

Schwarz, M., Pastrana, J., Allgeuer, P., Schreiber, M., Schueller, S., Missura, M., et al. (2013). "Humanoid teen-size open platform NimbRo-OP," in *Robot Soccer World Cup*, eds S. Behnke, M. Veloso, A. Visser and R. Xiong (Berlin: Springer), 568–575.

Stoelen, M. F., Bonsignorio, F., and Cangelosi, A. (2016). "Co-exploring actuator antagonism and bio-inspired control in a printable robot arm," in *International Conference on Simulation of Adaptive Behavior* (Aberystwyth: Springer), 244–255.

ten Kate, J., Smit, G., and Breedveld, P. (2017). 3D-printed upper limb prostheses: a review. *Disabil. Rehabil.* 12, 300–314. doi: 10.1080/17483107.2016.1253117

Wang, H., Dong, X., Chen, Z., and Shi, B. E. (2015). "Hybrid gaze/EEG brain computer interface for robot arm control on a pick and place task," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE* (Milan: IEEE), 1476–1479.

Zucker, M., Joo, S., Grey, M. X., Rasmussen, C., Huang, E., Stilman, M., et al. (2015). A general-purpose system for teleoperation of the DRC-HUBO humanoid robot. *J. Field Robot.* 32, 336–351. doi: 10.1002/rob.21570

Conflict of Interest Statement: ML and PR are affiliated to Pollen Robotics, a company with financial interests in the system presented in this work, as its primary supplier.

The remaining authors declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 17 January 2020
doi: 10.3389/fnbot.2019.00114

A Piezoresistive Array Armband With Reduced Number of Sensors for Hand Gesture Recognition

Daniele Esposito[1,2], Emilio Andreozzi[1,2], Gaetano D. Gargiulo[3], Antonio Fratini[4], Giovanni D'Addio[2], Ganesh R. Naik[5] and Paolo Bifulco[1,2]**

[1] Department of Electrical Engineering and Information Technologies, Polytechnic and Basic Sciences School, University of Naples Federico II, Naples, Italy, [2] Department of Neurorehabilitation, IRCCS Istituti Clinici Scientifici Maugeri, Pavia, Italy, [3] School of Computing, Engineering and Mathematics, Western Sydney University, Penrith, NSW, Australia, [4] School of Life and Health Sciences, Aston University, Birmingham, United Kingdom, [5] MARCS Institute for Brain, Behaviour and Development, Western Sydney University, Penrith, NSW, Australia

Human machine interfaces (HMIs) are employed in a broad range of applications, spanning from assistive devices for disability to remote manipulation and gaming controllers. In this study, a new piezoresistive sensors array armband is proposed for hand gesture recognition. The armband encloses only three sensors targeting specific forearm muscles, with the aim to discriminate eight hand movements. Each sensor is made by a force-sensitive resistor (FSR) with a dedicated mechanical coupler and is designed to sense muscle swelling during contraction. The armband is designed to be easily wearable and adjustable for any user and was tested on 10 volunteers. Hand gestures are classified by means of different machine learning algorithms, and classification performances are assessed applying both, the 10-fold and leave-one-out cross-validations. A linear support vector machine provided 96% mean accuracy across all participants. Ultimately, this classifier was implemented on an Arduino platform and allowed successful control for videogames in real-time. The low power consumption together with the high level of accuracy suggests the potential of this device for exergames commonly employed for neuromotor rehabilitation. The reduced number of sensors makes this HMI also suitable for hand-prosthesis control.

Keywords: muscle sensors array, piezoresistive sensor, human–machine interface, hand gesture recognition, support vector machine, exergaming

OPEN ACCESS

Edited by:
Keum-Shik Hong,
Pusan National University,
South Korea

Reviewed by:
Hassan Elahi,
Sapienza University of Rome, Italy
Amad Zafar,
University of Wah, Pakistan

****Correspondence:***
Ganesh R. Naik
ganesh.naik@westernsydney.edu.au
Paolo Bifulco
paolo.bifulco@unina.it

Received: *27 August 2019*
Accepted: *17 December 2019*
Published: *17 January 2020*

Citation:
Esposito D, Andreozzi E, Gargiulo GD, Fratini A, D'Addio G, Naik GR and Bifulco P (2020) A Piezoresistive Array Armband With Reduced Number of Sensors for Hand Gesture Recognition. Front. Neurorobot. 13:114. doi: 10.3389/fnbot.2019.00114

INTRODUCTION

Human machine interfaces (HMIs) are becoming increasingly widespread with applications spanning from assistive devices for disability, muscle rehabilitation, prosthesis control, remote manipulation, and gaming controllers (McKirahan and Guccione, 2016; Boy, 2017; Beckerle et al., 2018). Being the hand extremely important in one's life, an entire field of HMI is dedicated to hand gesture recognition applications (Arapi et al., 2018; Shukla et al., 2018). Generally, visual, electromyographic, or inertial sensors are the most used technologies for detecting hand gestures (Cho et al., 2017; Ghafoor et al., 2017; Bisi et al., 2018; Polfreman, 2018). Visual-based hand gesture recognition systems do not need any device to wear, allowing for extreme freedom of use. Such remote sensing is very attractive, but its performances are heavily influenced by many factors such

as camera field of view, challenging image processing, illumination conditions, objects overlapping, etc. (Chakraborty et al., 2017; Abraham et al., 2018). Devices based on surface electromyography (sEMG or simply EMG) recordings (Geng et al., 2016; Du et al., 2017) need electrodes in steady contact with the skin, and they are prone to motion artifacts, electromagnetic noise, and crosstalk with other biopotentials. They also require real-time processing of the raw sEMG signals to extrapolate useful features (e.g., sEMG envelope/RMS) (Parajuli et al., 2019). As example, Myo Armband by Thalmic Labs[1], a commercial device based on eight sEMG sensors and an inertial platform, allows the user to interface via Bluetooth with PCs or mobile devices to control supported applications (Nymoen et al., 2015; Sathiyanarayanan and Rajan, 2016; Myoband, 2019) including robot motion (Bisi et al., 2018).

As an alternative to sEMG, other sensors can monitor the mechanical muscular activity, and some are briefly presented below. A pressure sensors array coupled to air-bladders mounted on an armband was proposed to detect hand motion (accuracy of 90%) by monitoring the swelling of muscles (Jung et al., 2015). The air bladders are cumbersome, uncomfortable, and not widely adaptable. A wristband composed of an array of barometric pressure sensors was proposed to estimate tendons and muscle motions during gestures (Zhu et al., 2018), reaching a classification accuracy of wrist gestures of 98%. A combination of sEMG electrodes and microphones (Caramiaux et al., 2015) was used to detect both electrical muscle activity and the mechanomyogram (MMG – i.e., mechanical vibrations produced during muscle contraction). The microphones presented high sensitivity to noise and motion artifacts, in addition to the aforementioned EMG problems. A conventional ultrasound probe fixed to the forearm was proposed for finger motion recognition, proving accuracy of 96% (Huang et al., 2017). This approach resulted very cumbersome, uncomfortable, and required a complex image processing for gestures features extraction. Furthermore, piezoelectric sensors were used to estimate finger gestures (accuracy of 97%) by recording the vibrations and shape changes that occur at the wrist due to muscles and tendons motions (Booth and Goldsmith, 2018). These kinds of sensors could also be employed to harvest energy from body movements, including upper limb motion (Elahi et al., 2018).

Other recent studies (Giovanelli and Farella, 2016) presented devices for gesture recognition based on an array of force-sensitive resistors (FSRs[2]) (Interlink Electronics, 2019). A combination of two sEMG and four FSR sensors, mounted on a wrist strap, can be used to classify finger movements scoring accuracy of 96% (McIntosh et al., 2016). An armband equipped with 16 FSR sensors positioned on both wrists and forearms (Jiang et al., 2017) allowed the classification of several hand gestures with an accuracy of about 97%. A similar device equipped with eight FSR sensors, tested on amputees (Cho et al., 2016) while trying to mirror different hand grips in their residual forearm muscles, yielded an accuracy of 70%. Moreover, a high-density grid of 126 FSR sensors (Radmand et al., 2016) embedded in a forearm prosthetic socket and tested on healthy subjects to recognize arm positions, yielded an accuracy of 99.7%.

However, the approaches proposing pressure sensors wrapped around the wrist do not directly monitor muscle contraction, but rather tension of tendons. Moreover, even in the cases of FSR arrays applied on the forearm, to the best of our knowledge, the detected signals were not proven to be equivalent to EMG.

The aim of this study was to investigate the possibility to recognize hand gestures by monitoring the contractions of a reduced number of specific forearm muscles, via the bespoke FSR-based sensors, which demonstrated to provide signals quite similar to the EMG linear envelope (EMG-LE) (Esposito et al., 2018). To reach this goal, a new gesture recognition armband is presented; it is equipped with only three FSR-based sensors, applied on specific forearm muscles to recognize eight hand gestures. The armband is designed to be easily wearable and adjustable for any user. Thanks to the similarity with the EMG-LE (Esposito et al., 2018), the device could be reconfigured to resemble previous, well-established EMG-based HMIs (e.g., exergaming applications for patients during neuromotor rehabilitation) (Ma and Bechkoum, 2008).

MATERIALS AND METHODS

Piezoresistive Array Armband Design

The armband consists of three piezoresistive FSRs (Interlink FSR 402) mounted on an inextensible band by means of 3D printed rigid supports (**Figure 1**). An FSR changes its electrical resistance in the function of the applied force (Interlink Electronics, 2019). The FSR active area is suitably mechanically coupled to the muscle through a rigid dome, which enables the measurement of muscle volume changes during contraction (Esposito et al., 2018). The support was designed with a housing site for the FSR, and an opening to allow sensor sliding along the band and precise positioning on a target muscle. The armband can be wrapped around user's forearm and fastened with a Velcro strip in order to measure muscle contractions and recognize hand gestures. Indeed, each gesture generates a characteristic force distribution on the sensors, and this allows discriminating the intentional movements.

Given the similarity between the FSR-based sensor output and the EMG-LE (Esposito et al., 2018), the muscle sensors should be positioned above the muscle belly as for EMG detection. The chosen muscles should be superficial to allow advantageous signal to noise ratio. Moreover, since the FSR-based sensors are embedded in an armband, the pick-up points should belong to a circumference that wraps around the forearm. Three forearm muscles were preferred to better discriminate the different hand gestures. In detail, FSR1 was applied on flexor carpi ulnaris, FSR2 on flexor carpi radialis, and FSR3 on extensor digitorum. The armband was positioned proximally at 25% of the distance between the olecranon and the process styloideus ulnae of the right forearm (**Figure 2**). Indeed, a functional–anatomical analysis of the forearm muscles (Drake et al., 2014) revealed that flexor carpi ulnaris is mainly involved in wrist flexion and

[1]https://support.getmyo.com

[2]https://www.interlinkelectronics.com/request-data-sheets

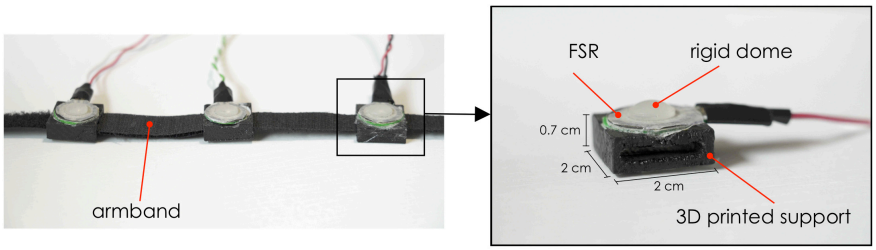


FIGURE 1 | Piezoresistive array armband: **Left**, the armband with three FSRs; **Right**, an enlargement of the FSR sensor mounted on its 3D printed support with actual dimensions.

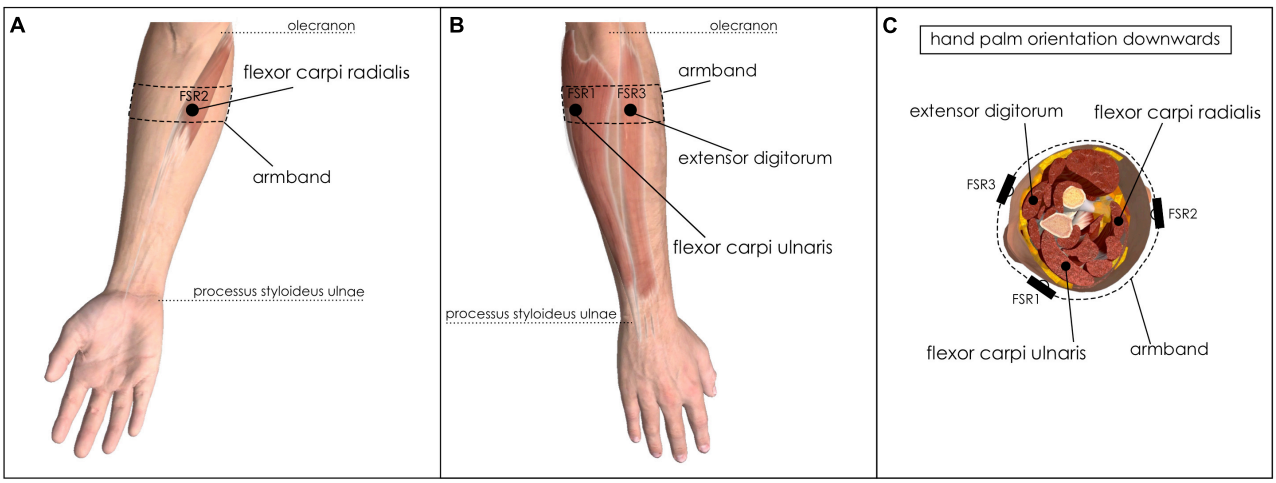


FIGURE 2 | Placements of FSRs on forearm muscles. **(A)** Ventral view of right forearm: FSR2 sensor on flexor carpi radialis; **(B)** Dorsal view of the right forearm: FSR1 on flexor carpi ulnaris and FSR3 on extensor digitorum; **(C)** Right forearm cross-section: FSRs placement onto the aforementioned muscles.

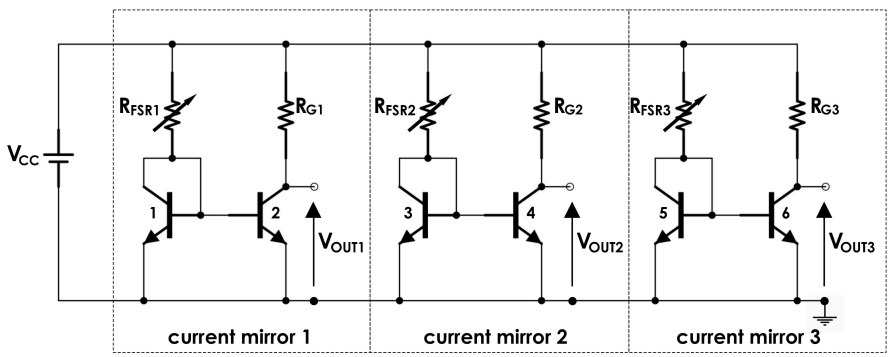


FIGURE 3 | FSR sensors conditioning circuit based on mirror current circuits.

wrist adduction; flexor carpi radialis in wrist flexion and wrist abduction; and extensor digitorum in fingers extension, fingers abduction, and wrist extension.

A current mirror (**Figure 3**) was used as a conditioning circuit for each FSR sensor (Esposito et al., 2019a,b). It was made of a pair of common npn BJT (2N2222), positioned very close to each other. Basically, the current mirror replicates the FSR sensor (R_{FSR}) current in the gain resistor (R_G), thus providing a linear load-to-voltage response and allowing the output voltage to swing through the full voltage supply range. The sensibility of each muscle sensor can be varied by changing the R_G value. Thanks to its low energy consumption, this conditioning circuit can be directly supplied by microcontrollers or ADC boards (e.g., 3.3 or 5 V). V_{CC} was set to 5 V, and the gain resistors R_{G1}, R_{G2}, and R_{G3} were set to 850, 790, and 960 Ω, respectively, to equalize the gains of the three channels.

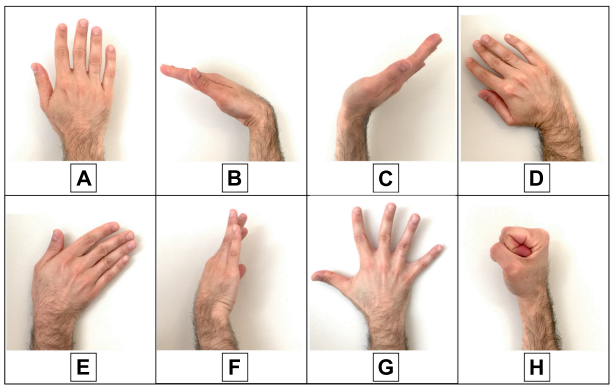


FIGURE 4 | Performed hand gestures: **(A)** rest; **(B)** wrist flexion; **(C)** wrist extension; **(D)** wrist adduction; **(E)** wrist abduction; **(F)** wrist rotation (supination); **(G)** fingers abduction; **(H)** clenched fist.

Static calibrations were performed for each FSR sensor to evaluate the relationship between the muscular force exerted on the FSR, reported in kilograms, and the voltage output V_{OUT} (**Figure 3**; Esposito et al., 2018). Each sensor was placed on a precision electronic scale, then different weights were applied on active sensor area perpendicularly to the dome, and the corresponding output voltages were recorded. The output signals were acquired at 1 kHz sampling frequency with 12-bit precision by means of National Instruments NI USB-6008 acquisition board.

Machine Learning Algorithms Applied to Hand Gesture Classification

The experimental tests involved 10 subjects (eight men and two women aged from 25 to 64 years), who provided their informed and written consent. Each participant comfortably sat on an adjustable height chair, leaning against its fixed seatback, in front of a desk with a computer screen. He was asked to place his elbow on the desk, forming an angle of about 45° between the forearm and the desktop. The armband was appropriately positioned on the forearm, and the pressure at rest was recorded by the sensors and resulted 100 g/cm^2 on average. The subjects were asked to perform 10 repetitions of each hand gesture class (**Figure 4**) in the following order: rest; wrist flexion; wrist extension; wrist adduction; wrist abduction; wrist rotation (supination); finger abduction; clenched fist; holding the final hand posture for a couple of seconds; and resting for a few seconds before the next movement. After the 10 repetitions of each hand gesture class, the participant was allowed to rest for about a minute. Simultaneous recordings from the three FSR sensors ($V_{OUT\ 1-2-3}$) were collected via the NI USB-6008 board at 1 kHz sampling frequency with 12-bit precision.

The raw signals were firstly pre-processed, by subtracting the minimum signal values recorded at rest (FSR offsets due to the armband fastening pressure) and normalizing to the absolute maximum value (**Figure 5**). In order to avoid manual selection of each hand gesture, pre-processed data were automatically segmented to extract the time intervals corresponding to the final hand postures. Segmentation was achieved by selecting the FSR signal with maximum variation (peak-to-peak amplitude) and applying a heuristically chosen threshold set at 40% of this value, which guaranteed appropriate segmentation of all gestures. Means and standard deviations (SDs) of the three FSR signals were computed for each segment. Then, for each gesture instance, the three means and the three SDs computed in the corresponding segment were considered as features. In detail, the features extracted from all the gestures instances in a single trial of a subject were assembled in a database consisting of an 80 × 7 matrix (10 repetitions for each of the eight hand gestures); each row corresponded to a single gesture instance and was composed by the following seven elements: (FSR1_mean, FSR2_mean, FSR3_mean, FSR1_SD, FSR2_SD, FSR3_SD, and GESTURE_LABEL).

Then, different machine learning algorithms (linear/polynomial/radial basis function-support vector machines; linear discriminant analysis; quadratic discriminant analysis; random forest; K-nearest neighbors, and neural networks) were used for model training and data classification, by means of "Weka" software (Frank et al., 2016). The conceptual scheme of the entire process of hand gestures classification is depicted in **Figure 5**.

Classification performances were assessed by applying the 10-fold and leave-one-out cross validations on each of the 10

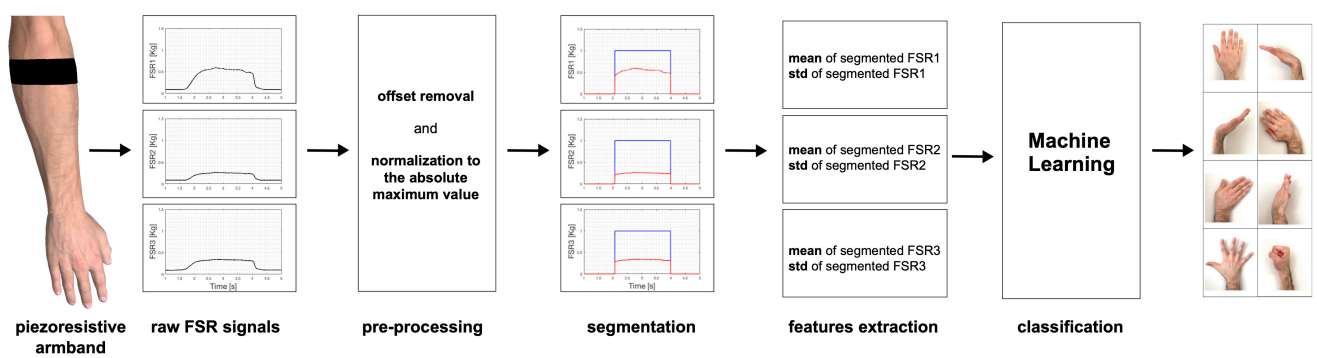


FIGURE 5 | Schematic illustration of the hand gesture recognition system.

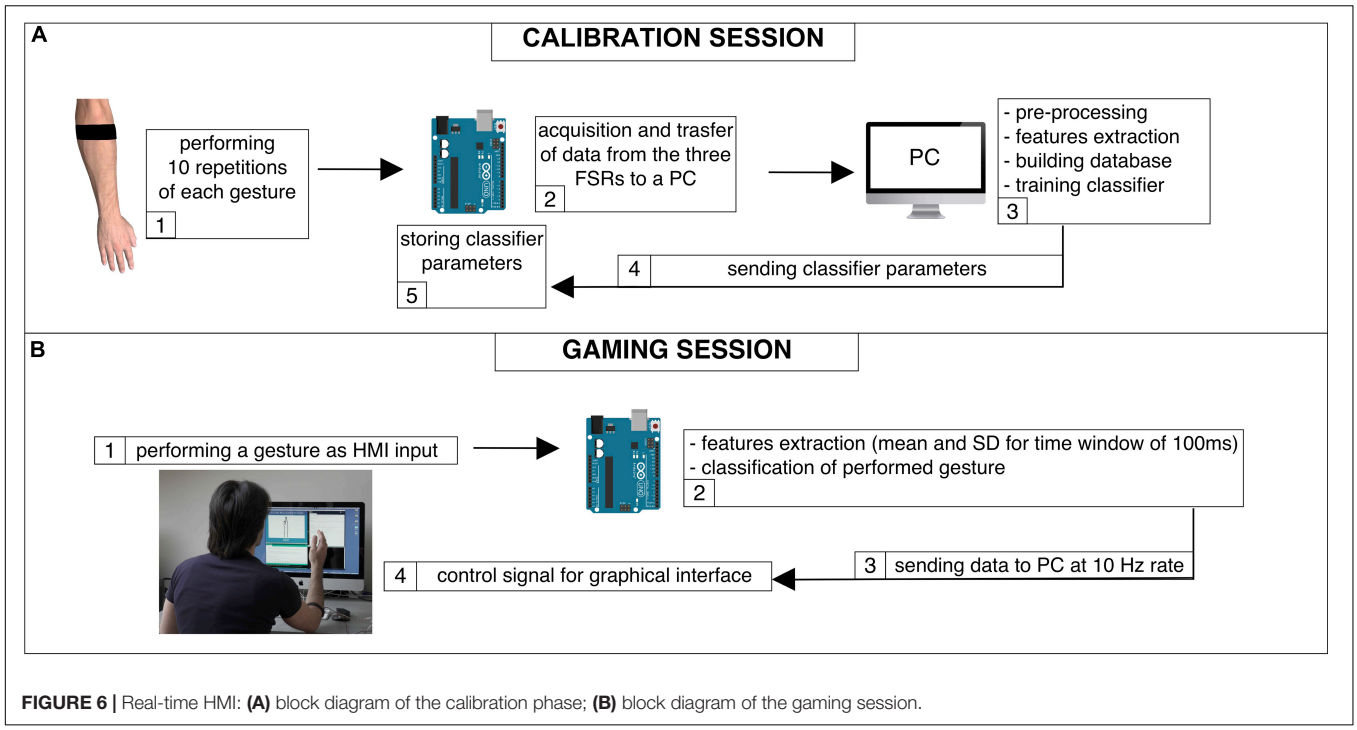


FIGURE 6 | Real-time HMI: **(A)** block diagram of the calibration phase; **(B)** block diagram of the gaming session.

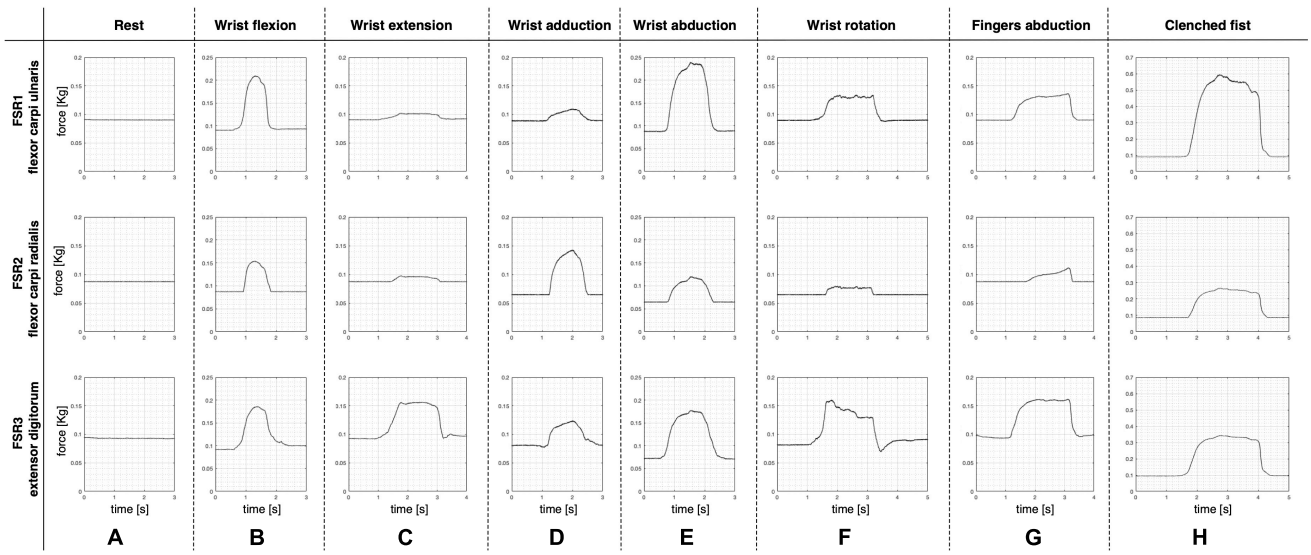


FIGURE 7 | Examples of raw signals (subject #3) recorded by the three FSRs for each performed gesture: **(A)** rest; **(B)** wrist flexion, **(C)** wrist extension, **(D)** wrist adduction, **(E)** wrist abduction, **(F)** wrist rotation (supination); **(G)** fingers abduction; **(H)** clenched fist. Signal amplitudes are expressed in kilograms and different force scales were used.

subjects' databases. In 10-fold cross-validation, the dataset is randomly divided into 10 subsets of equal size, and then each subset is tested using the classifier trained on the remaining nine subsets. Then, the obtained 10 classification accuracies were averaged to provide an overall classification accuracy. Instead, leave-one-out cross-validation is simply *n*-fold cross-validation, where *n* is the number of instances in the dataset. Each instance, in turn, is left out, and the learning method is trained on all the remaining instances. Finally, all the *n* classification accuracies were averaged to yield an overall classification accuracy (Witten et al., 2016).

Furthermore, the classification performances of the different machine learning algorithms were also tested on a combined database, obtained by joining all subjects' databases.

Finally, the possibility to classify gestures with less than three sensors was tested by considering features from different sensors

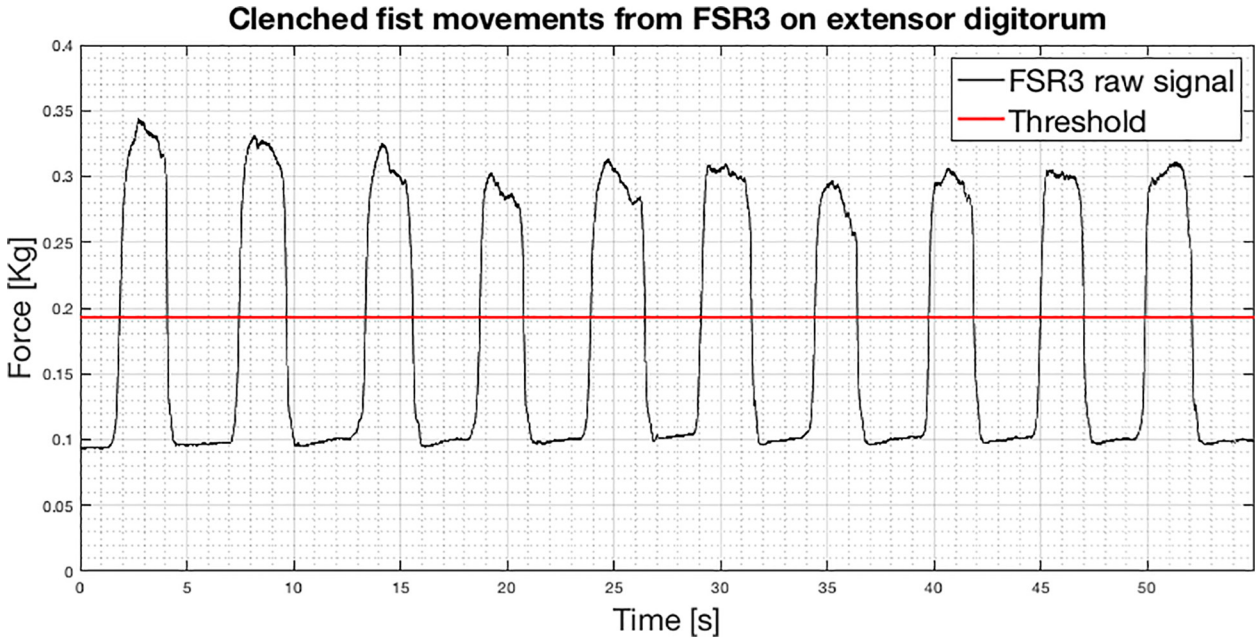


FIGURE 8 | Recording of 10 consecutive clenched fist movements from FSR3 (subject #3): FSR3 raw signal with the superimposed threshold (red line).

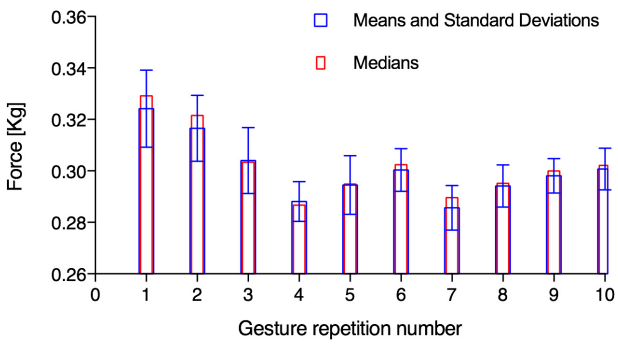


FIGURE 9 | Means, standard deviations, and medians related to the segmented FSR3 signals of 10 clenched fist movements showed in **Figure 8**.

pairs (FSR1-FSR2, FSR1-FSR3, and FSR2-FSR3) and even from a single sensor (FSR1, FSR2, and FSR3). In the case of sensors pairs, each instance is characterized by four features (two means and two SDs), while for a single sensor, the features reduced to two.

Reproducibility Test

A reproducibility test was also performed to assess the possibility to use a model trained in a previous trial to classify gestures performed in a subsequent trial. The data acquired by the 10 subjects (10 repetitions for each of the 8 gestures, as described in the section "Machine Learning Algorithms Applied to Hand Gesture Classification") were used to construct the "linear SVM" prevision model for each subject. Then, in a subsequent trial, the same subjects wore again the device and performed a randomized gestures sequence guided by a video. The video showed a sequence of icons representing the gestures to be performed (50 randomly chosen gestures separated by the rest condition). For each subject, the data collected in this last trial were classified using the model obtained from the previous trial. The entire procedure for the reproducibility test was repeated using an LDA classifier.

Real-Time Implementation of Hand Gesture Recognition

A linear SVM classifier was implemented on an Arduino UNO board[3] (D'Ausilio, 2012; Arduino, 2019), equipped with an ATmega328 (Atmel) microcontroller, to provide real-time gesture recognition. The three outputs of the FSR sensors conditioning circuit were directly connected to the analog inputs of the board. In addition, custom graphical user interfaces (GUI) were designed by means of "Processing" software[4] (Processing, 2019) to facilitate interactive armband calibration and to allow real-time user interaction with a computer. The real-time application involved the steps described below. The subject was asked to wear the armband and to perform the same sequence of gestures described in the section "Machine Learning Algorithms Applied to Hand Gesture Classification," for device calibration. Data were sent to the PC and used to train a linear SVM classifier by means of Weka software; the trained classifier parameters were sent to the Arduino board, and the calibration phase was completed (**Figure 6A**). The videogame started on the PC screen and the Arduino board performed real-time classification of the current gesture: extracting gesture features (mean and

[3]https://store.arduino.cc/arduino-uno-rev3

[4]https://processing.org

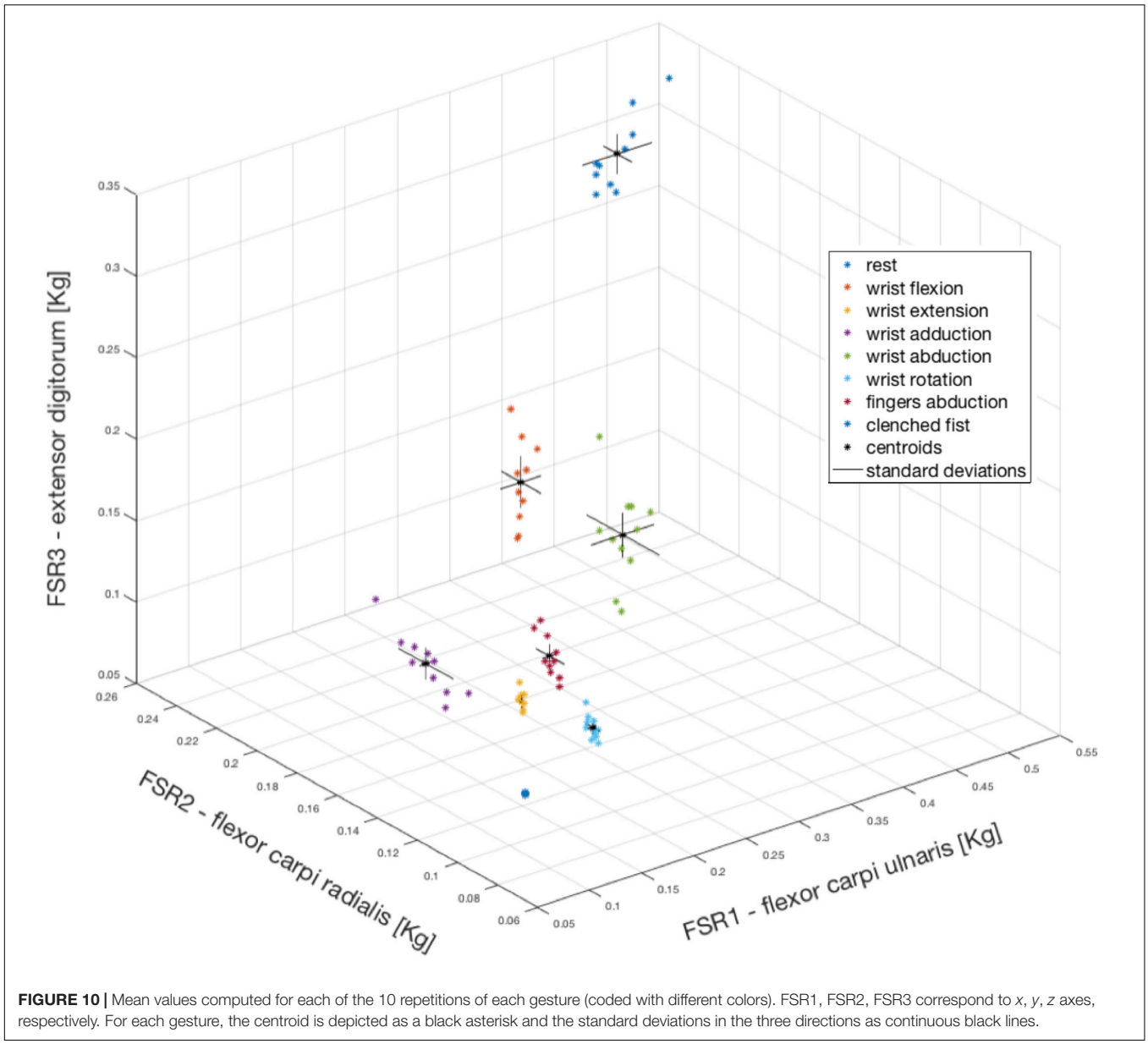


FIGURE 10 | Mean values computed for each of the 10 repetitions of each gesture (coded with different colors). FSR1, FSR2, FSR3 correspond to *x*, *y*, *z* axes, respectively. For each gesture, the centroid is depicted as a black asterisk and the standard deviations in the three directions as continuous black lines.

SD) every 100 ms, making a classification and sending this information (coded in 1 byte) to the PC at a 10 Hz rate, via USB communication (**Figure 6B**). The subject started to play, and the Arduino board output was used to replace the keyboard and mouse controls. The subject never removed the armband between these steps. For each gaming session, the gestures correctly recognized in real-time were annotated and then their percentages were computed. Each user was also asked to evaluate the comfort and effectiveness of the device on a 0-to-10 scale. The implementation of a real-time LDA classifier was further tested, repeating the same procedure described for the linear SVM (Hong et al., 2018).

Moreover, in order to verify a viable real-time classification, the mean and standard deviation parameters were computed using shorter FSR signal tracts than the segmented ones (the section "Machine Learning Algorithms Applied to Hand Gesture Classification"). However, due to the stationarity properties of the FSR signal during a particular gesture, these concise statistical parameters do not differ from those computed on larger time windows and used to train the classifier.

RESULTS

Signals Pre-processing and Hand Gestures Classification

Figure 7 shows an example of the FSRs raw signals for each performed hand gesture (subject #3). Different intensity force scales were used to better appreciate the signals shapes.

TABLE 1 | Classification accuracies (in percentage) on 10 different subjects, using different machine learning algorithms [linear SVM (L-SVM), polynomial SVM (P-SVM), radial basis function SVM (RBF-SVM), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), random forest (RF), K-nearest neighbors (K-NN), and neural networks (NN)] and different cross-validation methods [10-fold (CV1) and leave-one-out (CV2)].

	L-SVM		P-SVM		RBF-SVM		LDA		QDA		RF		KNN		NN	
	CV1	CV2	CV1	CV2	CV1	CV2	CV1	CV2	CV1	CV2	CV1	CV2	CV1	CV2	CV1	CV2
S1	95	95	87.5	81.25	93.25	91.25	97.5	97.5	97.5	97.5	96.25	96.25	91.25	91.25	90	92.5
S2	92.5	87.5	73.75	76.25	90	88.75	95	96.25	96.25	96.25	89	88.75	88.50	88.75	83.75	86.25
S3	98.75	98.75	82.5	78.75	97.5	96.25	97.5	96.25	100	100	97.5	97.5	95	93.75	97	96.25
S4	96.25	96.25	80	83.75	96.25	96.25	96.25	96.25	98.75	98.75	97.5	97.5	100	100	100	100
S5	90	88.75	75	73.75	90	97.5	93.75	92.5	97.5	97.5	93.5	93.75	91	90	91.5	90
S6	100	100	85	86.25	100	100	100	100	98.75	98.75	97.5	97.5	98.75	98.75	98.75	98.75
S7	97.5	97.5	97.5	93.75	97.5	97.5	100	100	98.75	98.75	97.5	95	100	100	98.75	97.5
S8	97.5	96.25	92.5	92.5	97.5	96.25	98.75	98.75	98.75	98.75	100	100	97.5	97.5	98.75	98.75
S9	96.25	96.25	83.75	86.25	97.5	96.25	96.25	96.25	98.75	98.75	98.75	98.75	98.75	98.75	100	100
S10	96.25	95	75	86.25	96.25	96.25	97.5	96.25	93.75	93.75	93.75	93.75	98.75	98.75	97.5	97.5

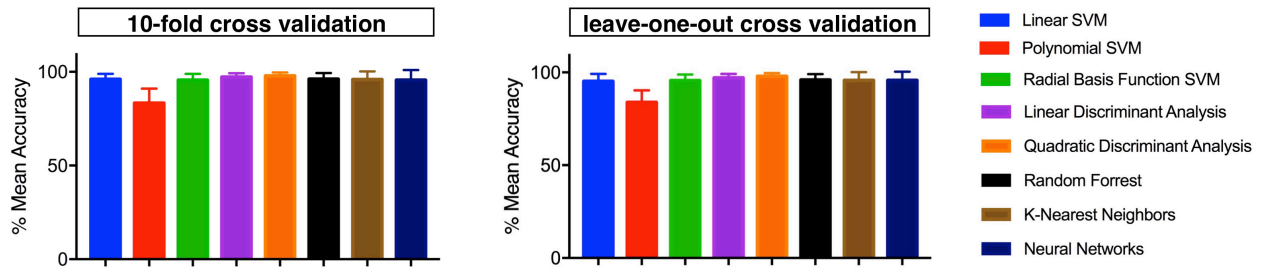


FIGURE 11 | Means and standard deviations of the accuracies achieved across the 10 participants, by means of the different machine learning algorithms and for each tested cross-validation method (**Left**, 10-fold and **Right**, leave-one-out).

TABLE 2 | Means and standard deviations of classification accuracies (across all participants) by using linear SVM and LDA algorithms for all sensors combinations.

Selected sensor/s	Linear SVM mean (SD) accuracy (%)	LDA mean (SD) accuracy (%)
FSR1	80.25 (9.89)	80.62 (8.00)
FSR2	76 (9.12)	82.37 (10.38)
FSR3	73.88 (13.25)	82.05 (8.58)
FSR1 and FSR2	91.75 (8.70)	92.27 (6.96)
FSR1 and FSR3	92.25 (5.26)	91.62 (6.18)
FSR2 and FSR3	90.38 (5.68)	92.87 (4.41)
FSR1 and FSR2 and FSR3	96 (2.93)	97.25 (2.02)

TABLE 3 | Classification accuracies reached on the combined database by using linear SVM and LDA for all sensor combinations.

Selected sensor/s	Linear SVM accuracy (%)	LDA accuracy (%)
FSR1	41.5	32.62
FSR2	38	28.5
FSR3	37.1	33.87
FSR1 and FSR2	49.6	44
FSR1 and FSR3	49.6	37.37
FSR2 and FSR3	51.9	41.12
FSR1 and FSR2 and FSR3	58.5	44.50

An example of raw signal segmentation is showed in **Figure 8**. The segmentation function was achieved by applying a threshold set at 40% of the FSR3 maximum signal variation. The segmentation allowed us to extract only the samples associated with the fully reached gesture while discarding the initial and final transients.

Moreover, analyzing the values of the segmented signals for each clenched fist movement in **Figure 8**, it was found that the distributions of the occurrences do not seem Gaussian. These probability distributions showed up also from the segmented signals related to the other gestures. The median, as an alternative to the mean, would be another possible feature. As an example, **Figure 9** shows the means, the standard deviations, and the medians referred to the segmented signals depicted in **Figure 8**. In this case, the percentage variation between the mean and the median was <2% for each repetition. Comparable percentages were also found in the segmented signals related to the other gestures. Hence, there is not practical convenience in using medians instead of means because it would increase the computational burden (critical for real-time applications).

As an example, **Figure 10** shows the means corresponding to the 10 repetitions of each gesture (subject #3) with different colors (see legend of **Figure 10**) in a three-dimensional space (x, y, and z axes correspond to FSR1, FSR2, and FSR3, respectively). In addition, data were enriched by reporting centroids and standard

TABLE 4 | Linear SVM classification accuracies (in percentage) on 10 different subjects in recognizing eight hand gestures (classes).

Gesture (class)	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Rest	100	100	100	100	100	100	100	100	100	100
Wrist flexion	90	100	100	100	100	100	100	100	100	100
Wrist extension	100	90	100	100	90	100	100	100	100	90
Wrist adduction	100	90	100	100	100	100	90	90	90	80
Wrist abduction	80	70	100	80	60	100	100	90	100	100
Wrist rotation	90	100	90	100	80	100	90	100	100	100
Fingers abduction	100	90	100	90	90	100	100	100	80	100
Clenched fist	100	100	100	100	100	100	100	100	100	100

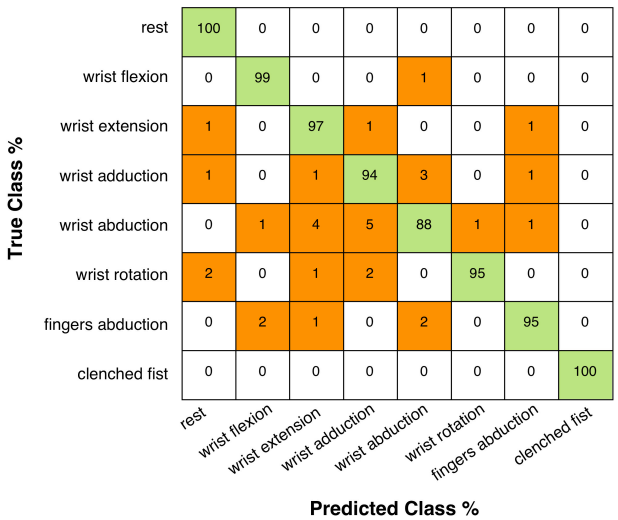


FIGURE 12 | Confusion matrix (across all participants) presenting the linear SVM classification accuracies (in percentages): rows correspond to true performed hand gestures and columns to predicted hand gestures.

deviations (computed in the three directions). Gestures appeared to be confined in specific regions, which did not overlap with each other. It is interesting to note that the rest condition was located around a point that represented the grip force of the armband (here about 0.1 kg).

TABLE 5 | LDA classification accuracies (in percentage) on 10 different subjects in recognizing eight hand gestures (classes).

Gesture (class)	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Rest	100	100	100	100	100	100	100	100	100	100
Wrist flexion	100	100	100	100	90	100	100	100	100	90
Wrist extension	100	90	100	100	90	100	100	100	100	100
Wrist adduction	100	100	90	80	90	100	100	90	80	90
Wrist abduction	90	70	100	90	90	100	100	100	100	100
Wrist rotation	90	100	90	100	90	100	100	100	100	100
Fingers abduction	100	100	100	100	100	100	100	100	90	100
Clenched fist	100	100	100	100	100	100	100	100	100	100

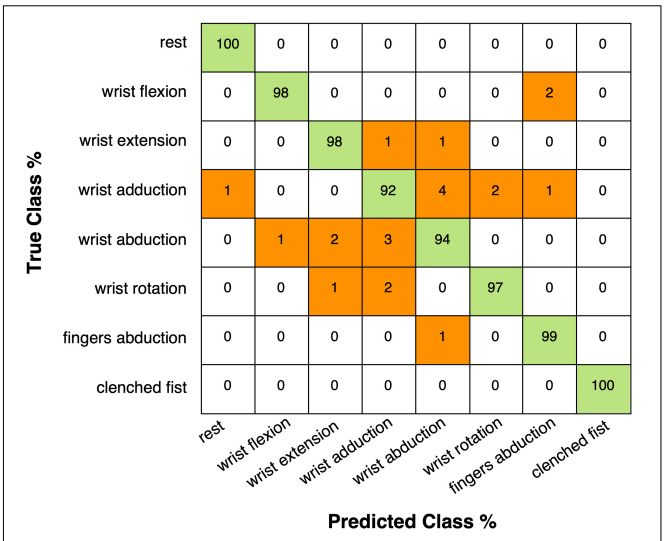


FIGURE 13 | Confusion matrix (across all participants) presenting the LDA classification accuracies (in percentages): rows correspond to true performed hand gestures and columns to predicted hand gestures.

Considering all three FSRs, the classification accuracy achieved for each subject, by means of the different algorithms and cross-validation methods, are shown in **Table 1**.

Figure 11 shows the means and the standard deviations of the accuracies achieved across all participants, using the aforementioned machine learning algorithms and the two cross-validation methods.

Table 1 shows that linear SVM and LDA algorithms allow to obtain higher classification accuracies with lower computational complexities, compared to all the other evaluated machine learning algorithms. Therefore, more extended analysis was focused on these classifiers, considering the 10-fold cross-validation.

Table 2 summarizes the classification performances achieved by considering all sensors combinations, reporting means and standard deviations of the related accuracies (across all participants). Using a single sensor, the mean classification accuracy was about 77% for linear SVM, while about 82% for LDA. Moreover, using two sensors the accuracy increased to about 91% for linear SVM, while about 92% for LDA.

Table 3 outlines the classification performances obtained for the combined database (all subjects) by using linear SVM and LDA for all sensors combinations.

Table 4 shows the classification accuracies reached with linear SVM, for each subject and hand gesture class. The average accuracy across all participants resulted 96% (SD: 2.93%), and the confusion matrix (right and wrong average recognition percentages across all 10 subjects) is shown in **Figure 12**.

Table 5 shows the classification accuracies reached with LDA, for each subject and hand gesture class. The average accuracy across all participants resulted 97.25% (SD: 2.02%), and the confusion matrix (right and wrong average recognition percentages across all 10 subjects) is shown in **Figure 13**.

Reproducibility Test

During the reproducibility test, the mean classification accuracy (across all users) was 78.8% with linear SVM, while 60.25% with LDA.

Graphical Interfaces for Practical HMI Applications

The custom graphical interface that displays icons corresponding to the recognized hand gestures was used both for calibration purposes and for quick assessment of real-time classifier performances (**Figure 6**). The real-time gesture recognition system was used to play various games (e.g., "Pong" videogame[5]) by replacing the mouse and keyboards commands with those provided by the Arduino board (Pong-Game, 2019). The average percentage (across all users) of correctly recognized gestures resulted 93% with linear SVM, while 90% with LDA. Subjects reported that this HMI was comfortable to wear and intuitive to use, not requiring long training to achieve good results. The mean "comfort score" was 8.3/10. The "effectiveness score" was 8.1/10 for linear SVM, and 7.8/10 for LDA.

DISCUSSION AND CONCLUSION

A novel piezoresistive array armband for hand gesture recognition was presented. It was based on a reduced number of muscle contraction sensors, appropriately positioned on specific forearm muscles. Nevertheless, it allowed discriminating eight classes of hand gestures with remarkable accuracy, regardless of the specific classifier (**Table 1**). Classifiers based on linear SVM and LDA have low computational complexities and can be easily implemented in hardware. Therefore, more extended analysis was focused on these classifiers. The average classification accuracy across all subjects, resulted 96% for linear SVM and 97.25% for LDA. These performances were achieved by separately considering the databases associated with each user and averaging the accuracies. Instead, considering the combined database (all subjects) the linear SVM classification achieved a maximum accuracy of 58.5%, while LDA scored 44.5%. A significant classification accuracy was also achieved by considering combinations of only two sensors: the mean accuracy resulted 91.46% for linear SVM and 92.25% for LDA. As expected, the use of a single sensor led to a significant reduction in mean classification accuracy (about 77% for linear SVM and 82% for LDA). With regard to the reproducibility test (described in the section "Reproducibility Test"), the mean classification accuracy (across all subjects) was 78.8% for linear SVM and 60.25% for LDA. This reduction in accuracy suggests that each time the device is used, a new calibration (i.e., classifier training) is advisable for optimal performances. It could be interesting to extend this study to a much larger cohort of subjects, in order to obtain more reliable classification results, and also to investigate the possibility to discover common muscle activation strategies, to identify pathological behaviors, etc.

[5]https://it.wikipedia.org/wiki/pong

The proposed armband is extremely lightweight, simple to wear, and easily adjustable for any user. It is comfortable and unobtrusive, as proved by the low grip force values recorded at rest, and it allows to simultaneously monitor the contractions of multiple specific forearm muscles. It is also scalable in the number of sensors, thus giving the opportunity to avoid their precise positioning onto specific muscles (e.g., full sensors covered armband could be used). The extreme simplicity of FSR sensors and their conditioning circuits, along with the straightforward usability of the output signals (no additional processing required), allow to easily implement this system on low-performing, commercial platforms, also with wireless capabilities (Gargiulo et al., 2010; Bifulco et al., 2011).

The proposed HMI could be applied in "exergaming" applications: graphical interfaces can provide patients with real-time feedback on the quality of the performed gestures, inducing self-corrections of their movements. Moreover, the possibility to monitor the contractions of specific muscles would provide additional clinical information about patients' progress. Thus, the exergaming could be used in clinical practice to make neuromotor rehabilitation processes more stimulating and enjoyable (Ordnung et al., 2017).

The encouraging results obtained with few sensors suggest the possibility to adopt this HMI also in hand prosthesis control (Polisiero et al., 2013; Bifulco et al., 2017; Sreenivasan et al., 2018), thanks to the similarity of the FSR-based sensors outputs and the EMG-LE. Indeed, the small size and flatness of the sensors make it possible to embed them inside the prosthesis socket. More generally, the muscle contraction sensors could be potentially adapted to monitor other muscles (e.g., muscles of arms, legs, shoulders, etc.), allowing them to develop a wide range of EMG-based HMI applications.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by the Department of Electrical Engineering and Information Technologies, University Naples of Federico II, Napoli, Italy. The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

DE, EA, and PB: conceptualization and methodology. DE and EA: software and investigation. AF, GG, and GD: validation. GD: resources. DE: writing – original draft preparation. EA, PB, AF, GG, and GN: writing, review, and editing. PB: supervision.

REFERENCES

Abraham, L., Urru, A., Normani, N., Wilk, M., Walsh, M., and O'flynn, B. (2018). Hand tracking and gesture recognition using lensless smart sensors. *Sensors (Basel)* 18:2834. doi: 10.3390/s18092834

Arapi, V., Della Santina, C., Bacciu, D., Bianchi, M., and Bicchi, A. (2018). DeepDynamicHand: a deep neural architecture for labeling hand manipulation strategies in video sources exploiting temporal information. *Front. Neurorobot.* 12:86. doi: 10.3389/fnbot.2018.00086

Arduino, (2019). *Arduino UNO* [Online]. Available at: https://store.arduino.cc/arduino-uno-rev3 (accessed May 16, 2019).

Beckerle, P., Kõiva, R., Kirchner, E. A., Bekrater-Bodmann, R., Dosen, S., Christ, O., et al. (2018). Feel-good robotics: requirements on touch for embodiment in assistive robotics. *Front. Neurorobot.* 12:84. doi: 10.3389/fnbot.2018.00084

Bifulco, P., Cesarelli, M., Fratini, A., Ruffo, M., Pasquariello, G., and Gargiulo, G. (2011). "A wearable device for recording of biopotentials and body movements," in *Proceedings of the IEEE International Symposium on Medical Measurements and Applications* (Bari: IEEE), 469–472.

Bifulco, P., Esposito, D., Gargiulo, G., Savino, S., Niola, V., Iuppariello, L., et al. (2017). "A stretchable, conductive rubber sensor to detect muscle contraction for prosthetic hand control," in *Proceedings of the 2017 E-Health and Bioengineering Conference (EHB)* (Sinaia: IEEE), 173–176.

Bisi, S., De Luca, L., Shrestha, B., Yang, Z., and Gandhi, V. (2018). Development of an EMG-controlled mobile robot. *Robotics* 7, 36. doi: 10.3390/robotics7030036

Booth, R., and Goldsmith, P. (2018). A wrist-worn piezoelectric sensor array for gesture input. *J. Med. Biol. Eng.* 38, 284–295. doi: 10.1007/s40846-017-0303-8

Boy, G. A. (2017). *The Handbook of Human-Machine Interaction: A Human-Centered Design Approach*. Boca Raton, FL: CRC Press.

Caramiaux, B., Donnarumma, M., and Tanaka, A. (2015). Understanding gesture expressivity through muscle sensing. *ACM Trans. Comput. Hum. Interact.* 21:31.

Chakraborty, B. K., Sarma, D., Bhuyan, M. K., and Macdorman, K. F. (2017). Review of constraints on vision-based gesture recognition for human–computer interaction. *IET Comput. Vis.* 12, 3–15. doi: 10.1049/iet-cvi.2017.0052

Cho, E., Chen, R., Merhi, L. K., Xiao, Z., Pousett, B., and Menon, C. (2016). Force myography to control robotic upper extremity prostheses: a feasibility study. *Front. Bioeng. Biotechnol.* 4:18. doi: 10.3389/fbioe.2016.00018

Cho, H., Lee, H., Kim, Y., and Kim, J. (2017). Design of an optical soft sensor for measuring fingertip force and contact recognition. *Int. J. Control Autom. Syst.* 15, 16–24. doi: 10.1007/s12555-016-0470-3

D'Ausilio, A. (2012). Arduino: a low-cost multipurpose lab equipment. *Behav. Res. Methods* 44, 305–313. doi: 10.3758/s13428-011-0163-z

Drake, R., Vogl, A. W., and Mitchell, A. (2014). *Gray's Anatomy for Student*, 3rd Edn. London, UK: Churchill Livingstone Elsevier.

Du, Y., Jin, W., Wei, W., Hu, Y., and Geng, W. (2017). Surface EMG-based inter-session gesture recognition enhanced by deep domain adaptation. *Sensors (Basel)* 17:458. doi: 10.3390/s17030458

Elahi, H., Eugeni, M., and Gaudenzi, P. (2018). A review on mechanisms for piezoelectric-based energy harvesters. *Energies* 11:1850. doi: 10.3390/en11071850

Esposito, D., Andreozzi, E., Fratini, A., Gargiulo, G., Savino, S., Niola, V., et al. (2018). A piezoresistive sensor to measure muscle contraction and mechanomyography. *Sensors (Basel)* 18:2553. doi: 10.3390/s18082553

Esposito, D., Cosenza, C., Gargiulo, G. D., Andreozzi, E., Niola, V., Fratini, A., et al. (2019a). "Experimental study to improve "Federica" prosthetic hand and its control system," in *Proceedings of the 15th Mediterranean Conference on Medical and Biological Engineering and Computing*, eds J. Henriques, P. de Carvalho, and N. Neves, (Coimbra: Springer International Publishing), 586–593. doi: 10.1007/978-3-030-31635-8_70

Esposito, D., Savino, S., Cosenza, C., Gargiulo, G. D., Fratini, A., Cesarelli, G., et al. (2019b). "Study on the activation speed and the energy consumption of "Federica" prosthetic hand," in *Proceedings of the XV Mediterranean Conference on Medical and Biological Engineering and Computing – MEDICON 2019*, Vol. 76, eds J. Henriques, N. Neves, and P. de Carvalho, (Cham: Springer International Publishing), 594–603. doi: 10.1007/978-3-030-31635-8_71

Frank, E., Hall, M. A., and Witten, I. H. (2016). "The WEKA workbench," in *Data Mining: Practical Machine Learning Tools and Techniques*, 4th Edn, ed. M. Kaufmann, (Burlington, MA: Morgan Kaufmann Publishers).

Gargiulo, G., Bifulco, P., Mcewan, A., Tehrani, J. N., Calvo, R. A., Romano, M., et al. (2010). "Dry electrode bio-potential recordings," in *Proceedings of the 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, (Buenos Aires: IEEE), 6493–6496.

Geng, W., Du, Y., Jin, W., Wei, W., Hu, Y., and Li, J. (2016). Gesture recognition by instantaneous surface EMG images. *Sci. Rep.* 6:36571. doi: 10.1038/srep36571

Ghafoor, U., Kim, S., and Hong, K.-S. (2017). Selectivity and longevity of peripheral-nerve and machine interfaces: a review. *Front. Neurorobot.* 11:59. doi: 10.3389/fnbot.2017.00059

Giovanelli, D., and Farella, E. (2016). Force sensing resistor and evaluation of technology for wearable body pressure sensing. *J. Sens.* 2016:9391850.

Hong, K.-S., Khan, M. J., and Hong, M. (2018). Feature extraction and classification methods for hybrid fNIRS-EEG brain-computer interfaces. *Front. Hum. Neurosci.* 12:246. doi: 10.3389/fnhum.2018.00246

Huang, Y., Yang, X., Li, Y., Zhou, D., He, K., and Liu, H. (2017). Ultrasound-based sensing models for finger motion classification. *IEEE J. Biomed. Health Inform.* 22, 1395–1405. doi: 10.1109/JBHI.2017.2766249

Interlink Electronics (2019). *FSR 400 – Interlink Electronics*. Camarillo, CA: Interlink Electronics.

Jiang, X., Merhi, L. K., Xiao, Z. G., and Menon, C. (2017). Exploration of force myography and surface electromyography in hand gesture classification. *Med. Eng. Phys.* 41, 63–73. doi: 10.1016/j.medengphy.2017.01.015

Jung, P.-G., Lim, G., Kim, S., and Kong, K. (2015). A wearable gesture recognition device for detecting muscular activities based on air-pressure sensors. *IEEE Trans. Industr. Inform.* 11, 485–494.

Ma, M., and Bechkoum, K. (2008). "Serious games for movement therapy after stroke," in *Proceedings of 2008 IEEE International Conference on Systems, Man and Cybernetics* (Singapore: IEEE).

McIntosh, J., Mcneill, C., Fraser, M., Kerber, F., Löchtefeld, M., and Krüger, A. (2016). "EMPress: practical hand gesture classification with wrist-mounted EMG and pressure sensing," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, NY: Association for Computing Machinery).

McKirahan, J., and Guccione, S. (2016). *Human Machine Interface: Concepts and Projects*. Norwalk, CT: Industrial Press.

Myoband, (2019). *Myo Band for EMG Analysis*. Available at: https://support.getmyo.com (accessed May 16, 2019).

Nymoen, K., Haugen, M. R., and Jensenius, A. R. (2015). "Mumyo–evaluating and exploring the myo armband for musical interaction," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Baton Rouge, LA.

Ordnung, M., Hoff, M., Kaminski, E., Villringer, A., and Ragert, P. (2017). No overt effects of a 6-week exergame training on sensorimotor and cognitive function in older adults. A preliminary investigation. *J. Front. Hum. Neurosci.* 11:160. doi: 10.3389/fnhum.2017.00160

Parajuli, N., Sreenivasan, N., Bifulco, P., Cesarelli, M., Savino, S., Niola, V., et al. (2019). Real-time EMG based pattern recognition control for hand prostheses: a review on existing methods, challenges and future implementation. *Sensors (Basel)* 19:4596. doi: 10.3390/s19204596

Polfreman, R. (2018). "Hand posture recognition: IR, sEMG and IMU," in *Proceedings of the Conference on New Interfaces for Musical Expression 2018*, Blacksburg, VA, 6.

Polisiero, M., Bifulco, P., Liccardo, A., Cesarelli, M., Romano, M., Gargiulo, G. D., et al. (2013). Design and assessment of a low-cost, electromyographically controlled, prosthetic hand. *Med. Devices (Auckl)* 6, 97–104. doi: 10.2147/MDER.S39604

Pong-Game, (2019). *Pong Game*. Available at: https://en.wikipedia.org/wiki/Pong (accessed May 21, 2019).

Processing, (2019). *Processing*. Available at: https://processing.org/ (accessed May 10, 2019).

Radmand, A., Scheme, E., and Englehart, K. (2016). High-density force myography: a possible alternative for upper-limb prosthetic control. *J. Rehabil. Res. Dev.* 53, 443–456. doi: 10.1682/JRRD.2015.03.0041

Sathiyanarayanan, M., and Rajan, S. (2016). "MYO armband for physiotherapy healthcare: a case study using gesture recognition application," in *Proceedings of*

the 2016 8th International Conference on Communication Systems and Networks (Bengaluru: IEEE), 1–6.

Shukla, D., Erkent, , Ö, and Piater, , J. (2018). Learning semantics of gestural instructions for human-robot collaboration. *Front. Neurorobot.* 12:7. doi: 10.3389/fnbot.2018.00007

Sreenivasan, N., Ulloa Gutierrez, D. F., Bifulco, P., Cesarelli, M., Gunawardana, U., and Gargiulo, G. D. (2018). Towards ultra low-cost myo-activated prostheses. *Biomed Res. Int.* 2018:9634184. doi: 10.1155/2018/9634184

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA: Morgan Kaufmann.

Zhu, Y., Jiang, S., and Shull, P. B. (2018). "Wrist-worn hand gesture recognition based on barometric pressure sensing," in *Proceedings of the 2018 IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks* (Las Vegas, NV: IEEE), 181–184.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

REVIEW
published: 15 April 2020
doi: 10.3389/fnbot.2020.00019

A Review of Robot-Assisted Lower-Limb Stroke Therapy: Unexplored Paths and Future Directions in Gait Rehabilitation

*Bradley Hobbs and Panagiotis Artemiadis**

Human-Oriented Robotics and Control Laboratory, Department of Mechanical Engineering, University of Delaware, Newark, DE, United States

Stroke affects one out of every six people on Earth. Approximately 90% of stroke survivors have some functional disability with mobility being a major impairment, which not only affects important daily activities but also increases the likelihood of falling. Originally intended to supplement traditional post-stroke gait rehabilitation, robotic systems have gained remarkable attention in recent years as a tool to decrease the strain on physical therapists while increasing the precision and repeatability of the therapy. While some of the current methods for robot-assisted rehabilitation have had many positive and promising outcomes, there is moderate evidence of improvement in walking and motor recovery using robotic devices compared to traditional practice. In order to better understand how and where robot-assisted rehabilitation has been effective, it is imperative to identify the main *schools of thought* that have prevailed. This review intends to observe those perspectives through three different lenses: the goal and type of interaction, the physical implementation, and the sensorimotor pathways targeted by robotic devices. The ways that researchers approach the problem of restoring gait function are grouped together in an intuitive way. Seeing robot-assisted rehabilitation in this unique light can naturally provoke the development of new directions to potentially fill the current research gaps and eventually discover more effective ways to provide therapy. In particular, the idea of utilizing the human inter-limb coordination mechanisms is brought up as an especially promising area for rehabilitation and is extensively discussed.

Keywords: gait rehabilitation, rehabilitation robotics, review, stroke therapy, therapeutic devices

OPEN ACCESS

Edited by:
Yongping Pan,
National University of Singapore, Singapore

Reviewed by:
Andrés Úbeda,
University of Alicante, Spain
Giacinto Barresi,
Italian Institute of Technology (IIT), Italy

****Correspondence:***
Panagiotis Artemiadis
partem@udel.edu

Received: *25 January 2020*
Accepted: *16 March 2020*
Published: *15 April 2020*

Citation:
Hobbs B and Artemiadis P (2020) A Review of Robot-Assisted Lower-Limb Stroke Therapy: Unexplored Paths and Future Directions in Gait Rehabilitation. Front. Neurorobot. 14:19. doi: 10.3389/fnbot.2020.00019

1. INTRODUCTION

Stroke is typically caused by a long-term lack of oxygen to the brain through a blood vessel bursting or clotting. Since this event usually occurs on one side of the brain, the effects are generally seen on the contralateral half of the body in the form of hemiparesis. This partial paralysis is common after stroke and makes a significant impact on daily life. After initial onset, recovery in the early stages is crucial to mitigate the long-term effects of stroke. More people are in need of stroke rehabilitation every year, and the cost for post-stroke patients with a need for continuous care is still high and projected to substantially increase in the next decade (Benjamin et al., 2019). In order to reduce the cost and increase the efficacy of post-stroke rehabilitation, it is crucial to determine and use the methods that prove to provide the best outcomes.

In recent years, robotic and electromechanical systems have gained increased interest in the rehabilitation community for their ability to automate the tedious and time intensive therapy needed for beneficial patient outcomes (Sale et al., 2012; Calabrò et al., 2016). Because locomotion is the result of complex dynamic interactions between feedback mechanisms and a central controller in the brain, the rehabilitation methods that work the best use a fundamental understanding of this coordination of human gait (Gassert and Dietz, 2018). It is well-known that in order to be effective, therapy should begin as soon as possible and provide an intensive training that incorporates multiple sensory mechanisms in a structured way (Poli et al., 2013). Robotic and electromechanical systems for rehabilitation purposes are designed with the intent of evoking the muscle activation synergies and neural plasticity through specific repetitive motor coordination exercises. Because brain tissue cannot simply be repaired in the exact way as before the damage, in order to regain a physical ability such as walking, the brain must be *rewired* along intact, active neural pathways. This influences therapies that incorporate various sensory inputs, experiences, learning, and especially motor training (Poli et al., 2013), showing there is a link between vigorous multisensory rehabilitation and recovery in stroke patients. Therefore, neural pathways that are not normally in use might be triggered to make up for the lost pathways. The intensity of stimulating those pathways can be drastically increased by introducing robotic devices to aid the physical therapists.

Because of the fast pace in which rehabilitation robotics has grown, robots and autonomous systems are longing to be the standard in rehabilitation. Due to both a rapid increase in technological improvements (Reinkensmeyer et al., 2004; Schmidt et al., 2005b; Hogan et al., 2006; Johnson, 2006; Patton et al., 2006) and a rapid increase in neurological understanding of rehabilitation (Kwakkel et al., 2008; Carter et al., 2010; Albert and Kesselring, 2012), there is a need to summarize where we are currently at with popular and emerging methods. This paper is an attempt to organize and categorize the ways in which we think about stroke rehabilitation, in order to produce more effective approaches to be developed in the future, while making sure to learn from past mistakes.

Moreover, there is a significant disparity between engineers that create devices for rehabilitation, and the underlying neuroscience related to motor deficits and rehabilitation after stroke. While this gap is certainly closing, it can be further bridged by understanding the underlying mechanisms for gait, gait adaptation, and gait therapy and by connecting promising technological advances in robotics with promising, related underlying neural pathways. Many of the studies and methods shown in this paper have produced promising results, but the proof of long-term benefits is required for the proper use of the word rehabilitation. The critical difference of this paper compared to previous reviews (Dickstein, 2008; Vallery et al., 2008; Marchal-Crespo and Reinkensmeyer, 2009; Schwartz et al., 2009; Díaz et al., 2011; Horno et al., 2011; Morone et al., 2011; Conesa et al., 2012; Mehrholz and Pohl, 2012; Pennycott et al., 2012; Chang and Kim, 2013; Kelley et al., 2013; Viteckova et al., 2013; Waldner et al., 2013; Zhang et al., 2013; Swinnen et al., 2014; Venkatakrishnan et al., 2014; Mehrholz et al., 2017; Agostini et al., 2018; Bruni et al., 2018), is that each individual point of view behind the creation of these methods is grouped into schools of thought, or approaches, based on a fundamental understanding of rehabilitation. This paper systematically reviews the different methods used by scientists to study and rehabilitate gait in humans and discusses the gaps in research that have yet to be filled, prompting potential new directions in the field. These schools of thought are the desired goal and type of interaction, the physical implementation of the method, and the neural mechanisms that are intended to be targeted or evoked, as depicted in **Figure 1**. There are many different tools and ways of thinking about gait rehabilitation, so within each school of thought, some of the gaps left behind are put forth.

2. INTERACTION GOAL AND TYPE

Each rehabilitation technique and system known today can fall under many different categories depending on differing points of view. One distinction can be made in the area of how the method will interact with the subject based on the goal of the therapy. For example, the most popular developments use some sort of physical connection between the subject and the machine. Furthermore, there is another distinction made in the objective of this interaction. In all cases, this physical interaction can be grouped according to if the interaction is intended to correct the subject's pathological gait, or if it is intended to perturb, or induce error into the gait. There has also been some research into methods that do not directly interact with the patient's gait in a physical way, but offer a form of rehabilitation in a strictly informational or communicative way. The majority of the current methods should fit into these three ways of thinking sub-categories, as it is further discussed below.

2.1. Error Correction via Physical Interaction

The human gait is often thought to be the result of complex sensori-motor neuro-mechanical systems that use real-time feedback to control the different muscle groups. The main objective of this controller is to produce a steady state walking pattern, considering any errors produced by the internal sources such as muscle spasm or overshoot, and external sources such as ground stiffness changes or interaction forces. In patients with post-stroke hemiplegia, there is a loss of coordination in muscle activations in certain muscle groups, or synergies on one side of the body. This causes the gait patterns in those individuals to exhibit common undesired traits such as drop-foot (Krebs et al., 2008). An idea that has been proposed in the past is to try to minimize those undesired activation patterns through robot-assisted rehabilitation. The long-term objective of the robotic intervention in that framework is to minimize the difference (error) between the normal and the paretic movement of the limb, while increasing repeatability and intensity of training. In general, this is through augmenting and precisely automating movements that would normally be followed through manually by a physical therapist. Thus, use of a robotic system minimizes

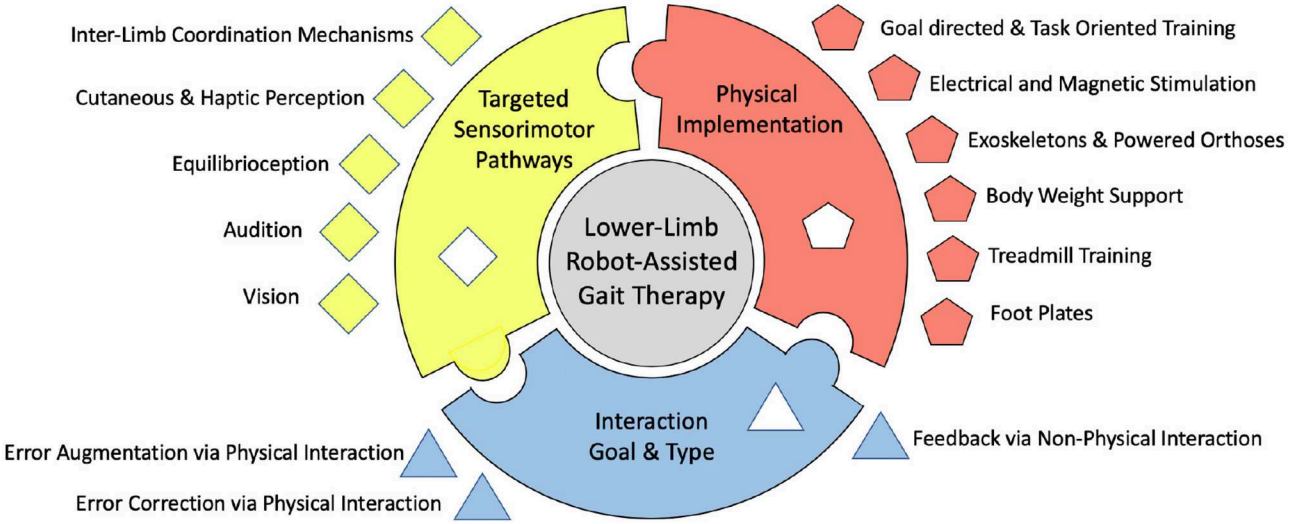


FIGURE 1 | Depiction of the proposed organizational chart of existing robot-assisted stroke rehabilitation methods.

therapist fatigue and increases repeatability (Freivogel et al., 2009; Peurala et al., 2009).

Early methods developed were in the form of systems intended to allow for prolonged training sessions and reduced therapist workload by automating the process of facilitating gait patterns (Colombo et al., 2000, 2001; Belforte et al., 2001). Many researchers use a trajectory tracking based approach to gait training (Beyl et al., 2008). A robot with upper and lower limb connections that allowed for walking velocity updates through generated spatial motions on the sagittal plane for each foot was developed by authors in Emken et al. (2005). For ankle assistive devices, authors in Agrawal et al. (2005) developed an ankle-foot orthosis to assist the tibialis anterior muscle in maintaining proper foot position for subjects with ankle flexion/extension control and inversion/eversion control. To avoid imposing constraints on naturalistic walking due to a robot's kinematic structure, the work in Aoyagi et al. (2007) suggests assisting the pelvic motion during stepping, and providing a type of compliant assistance to avoid perturbed rather than assisted stepping. This was intended to be a compliant robot that could act either in aid of the trainers, or in place of them if desired, and tailored the desired trajectory for each subject. Authors in Bharadwaj et al. (2005) and Bharadwaj and Sugar (2006) also emphasized repetitive task training as an effective form of rehabilitation for people suffering from stroke and presented an ankle rehabilitation method based on a tripod mechanism which moves the ankle in dorsiflexion/plantarflexion and inversion/eversion.

Some implementations for the seated position were presented in Bouri et al. (2009) and Chisholm et al. (2014) and then for the standing position through a deambulator mechanically interfaced with the verticalized orthoses (Bouri et al., 2006). Many methods use velocity or moment control (Chen et al., 2009) or an idea of feedback control of joint trajectories through modulated friction brakes (Farris et al., 2009a), which are used in conjunction with electrical stimulation. This unidirectionally couples hip to knee flexion and aids hip and knee flexion with a spring assist (Farris et al., 2009b). Specifically used for stair ascent and descent, powered assistance in the sagittal plane at both hip and knee joints and can be used in conjunction with an ankle foot orthosis (Farris et al., 2012) or functional electrical stimulation (FES) (Ha et al., 2012, 2016). Studies suggest walking with error correcting devices such as an exoskeleton provides increase in walking speed and a concomitant decrease in required exertion relative to walking with other knee-ankle-foot-orthoses (Farris et al., 2014).

Another ankle robot (Forrester et al., 2013) uses an internal model-based adaptive controller that both accommodates individual deficit severities and adapts to changes in patient performance. In general, the main purpose of an ankle-based system is to prevent slapping the foot after heel strike, and to control the ankle joint to actively minimize the fore foot collision with the ground (Hwang et al., 2006). This can be achieved by lifting the foot during swing but supporting further gait movements by controlling of the center of mass (Hesse et al., 2000). This system was later adapted to simulate level floor walking as well as climbing up and down stairs (Hesse et al., 2010). These systems intend to simulate gait-like movement through simulating stance and swing phases.

2.2. Error Augmentation via Physical Interaction

Many researchers have approached rehabilitation with the notion of a subject learning his or her own walking pattern through unexpected physical contact made to him or her, which elicits a reaction response in order to correct for the disturbance. It has been shown that individuals with cerebral damage from stroke have a normal capacity to make both reactive and predictive locomotor adaptations during walking (Choi and Bastian, 2007). The idea here is that neural plasticity is evoked through the brain

attempting the gait correction in response to the disturbance. The brain perceives the error through various senses and neural pathways in the body and corrects based on this feedback. Typically, this still involves repetitive and frequent trials in order for a long-term effect to be realized in many cases.

An early example of this interaction method is the work presented in Girone et al. (2001), where a Stewart platform supplies resistive forces in response to virtual reality-based exercises. Error inducing methods are meant to manipulate human stepping, but can be used to study the mechanical properties of different joints as well (Roy et al., 2007). Another early implementation that emphasizes back-driveability and force generation capability shows that this method can induce motor adaptation and long-term after-effects (Reinkensmeyer et al., 2003). Many robotic tools provide different assistance levels, but also may have modes that challenge the subject's posture (Peshkin et al., 2005), elicit a stumbling like response (Schmidt et al., 2005a), regulate force feedback (Barkan et al., 2014), or induce perturbations (Schmidt and Werner, 2007) and resistance (Saglia et al., 2009; Klarner, 2010). In some cases, force-field-based perturbations can cause a subject to adapt to the applied field and follow normal gait pattern until it is turned off (Koopman et al., 2013). In other cases, these perturbation-based methods attempt to induce error by unexpectedly removing the perturbations and observing the after-effects (Reinkensmeyer et al., 2014), during treadmill training (Skidmore et al., 2015) or over-ground walking (Martelli et al., 2019). The idea of augmenting the error feedback is also shown to reduce some asymmetries in gait (Bishop et al., 2017). Even after therapeutic intervention, counteracting force perturbations can lead to improved responses for real-world loss of balance in regular life (Matjačić et al., 2018) by applying these force perturbations in a controlled setting (Olenšek et al., 2018).

2.3. Feedback via Non-physical Interaction

The third distinction is made for methods of rehabilitation that do not directly interact with the subject in a physical manner. This means that there are no corrections or perturbations evoked directly or indirectly through the senses. This is much less common but is emerging and can have the benefit of a greater patient independence. One of the ways to do this is by having a socially assistive robot that will give some sort of informational feedback to the subject through audio or visual means (Matarić et al., 2007). While this informational feedback is often coupled with physical contact in some way, it is worth mentioning briefly on the aspects of the non-physical method. Self-training is a relatively new method for rehabilitation that is augmented with robotic assistants that guide and observe patients during tasks (Gross et al., 2014). This comes with the challenge of the navigation and perception of humans and human behavior (Losey and O'Malley, 2019). Described as socially assistive, these robotic platforms describe a modern thought process for the role of robots in stroke therapy for survivors that have standing and walking mobility (Feil-Seifer and Matarić, 2005).

Another non-physical method for rehabilitation could be communicating information to the subject about how the subject is progressing in terms of his or her gait by making gait data accessible for self-correction. This informational feedback can be data provided to a subject to influence the rehabilitation process in some way. The distinction is made when the information is in the form of meaningful data that the subject can use in independent, self-driven rehabilitation, potentially post-therapy. This can include displaying movement patterns throughout the day and comparing with daily goals or informing the overall physical progress statistics of the subject. To achieve improvements in all phases of therapy, supplementing the patient with his or her own data is a potential addition in stroke therapy.

3. PHYSICAL IMPLEMENTATION

The second of the three categories that this paper uses for grouping the robot-assisted gait rehabilitation approaches, is the physical way in which each method is implemented. Even if the type of interaction is non-physical, the implementation of this method still has a physical attribute. Any robot-assisted rehabilitation method should fit into one of the following physical implementations. Many of these implementations assist physical therapists either in determining progress or in alleviating the strain of high intensity and repetitions by automating the therapy process. It should be noted that one physical implementation can use more than one idea from the Type and Goal of Interaction subcategory; consequently, this is where many gaps can be found.

3.1. Body Weight Support

Body weight supported training has been used in many studies in the past. Typically coupled with a treadmill, a body weight support allows the therapist to take a varying degree of weight off of the subject (Bouri et al., 2006; Seo and Lee, 2009). This is often used to minimize the effects of balance impairments or when the patient is unable to independently support his or her own weight (Stauffer, 2009). Perhaps the earliest form of mechanically augmented rehabilitation is the use of an over-ground body weight support (Hesse et al., 1999) as opposed to other techniques that may use body weight support in treadmill training. This includes a body weight support system that is connected to a base with wheels for mobility when a treadmill may not be desired (Peshkin et al., 2005). In general, proper body weight support is provided to reduce the balance and postural control mechanisms for both treadmill (Skidmore et al., 2014) and over ground studies (Peshkin et al., 2005).

3.2. Foot Plates

Many early physical approaches use only a foot attachment for rehabilitation and for exercise studies (Homma and Usuba, 2007). This has the benefit of being applied while a patient is in the early stages of stroke who may be bedridden (Monaco et al., 2009) or wheelchair bound (Hesse and Werner, 2009). Early systems may call these "haptic interfaces" (Girone et al., 2001), however this term is too broad for modern robotic classifications. With physical therapist assistance, high intensity training with these can lead to better gait ability (Hoölig et al., 2007). These can also apply assistive and resistive training (Saglia et al., 2009). The foot plate approach differs from other robotic methods in that only the foot is attached to the external device (Schmidt et al., 2007). This can have the impression of acting as a robot that actuates

the foot in a Cartesian space, rather than in a joint space (Tsoi and Xie, 2008). These are typically actuated to move the foot into a trajectory (Freivogel et al., 2008) but can also perturb the normal walking pattern by using error inducing ideas. Since only the interaction with the foot is controlled, the foot plates have the advantage of being able to simulate different walking surfaces such as stairs (Hesse et al., 2000, 2010; Yano et al., 2010), inclines and uneven surfaces (Iwata et al., 2002). Some foot plates are also in the form of stationary platforms that allow the patient to sit (Bouri et al., 2009) or stand (Boian et al., 2005) in place and can aid in balance (Ding et al., 2010). Individuals who are not able to support their full body weight onto the legs benefit from the foot plate approach.

3.3. Exoskeletons and Powered Orthoses

The largest subcategory and the one that has received the most attention in the area of rehabilitation for many neurological disabilities including stroke is the exoskeletons and powered orthoses category. Many of the ways rehabilitation attempts are made is through an exoskeleton or a powered orthosis. These attach to the leg of the subject and induce an actuated torque directly on one or more of the joints (Veneman et al., 2007). This is most often used in conjunction with motors and some form of a controller to actively provide precise joint torques. Depending on the desired interaction goal, these can be used over ground (Murray et al., 2014) or in conjunction with a treadmill (Jezernik et al., 2003), and can have improved functional outcomes results in different aspects of motor recovery (Hornby et al., 2005; Heller et al., 2007; Mayr et al., 2007; Hidler et al., 2009; Chang et al., 2012; Krishnan et al., 2012; Nilsson et al., 2014; Kim et al., 2015). With over-ground systems, much of the focus has been on compactness (Farris et al., 2011a), and trajectory (Farris et al., 2011b). It should also be noted that some exoskeletons focus on spinal cord injured patients and have mechanical design and control attributes that can transfer to stroke rehabilitation as well. Seated implementations also have potential for reducing ankle impairment, as well as studying the effects of varied feedback on lower extremity motor learning (Forrester et al., 2011).

Recently, interest has been drawn in the design of interfaces that can use internal body measurements or intentions while walking, such as electroencephalography (EEG) signals (He et al., 2014) or surface electromyography (EMG) signals (Ferris et al., 2006). These types of systems, such as a brain-computer interface (BCI), offer an alternative, internal-based method for accessing information about the human body. The goal of using these implementations is to use this neuro-physiological information to provide control-relevant information for a rehabilitation robot to make decisions considering force and timing for movement. Surface EMG has been used in the control of trajectories of full leg manipulation systems (Kawamoto and Sankai, 2007) and ankle foot systems (Ferris et al., 2005) for rehabilitation. These provided non-FES aided gait, as opposed to earlier methods (Goldfarb et al., 2003) that were mainly designed to just provide significantly better trajectory control and to reduce muscle fatigue when compared to FES-only gait. Current and future implementations of this neuro-physiological information have the challenge of overcoming signal variability, classification algorithm robustness, and quantifiable performance feedback indicators (Tariq et al., 2018). Current advances in EMG and EEG analysis have led to broad applications of this control approach in rehabilitation robotics, however these challenges still require solving for these methods to become viable parts of rehabilitation, especially in exoskeleton and orthosis implementations (Ison and Artemiadis, 2014).

Soft actuation has the advantage of providing a more compliant way to interact with natural human morphology and biomechanics (Ortiz et al., 2017). Some successful robotic exoskeletons have even been redesigned as a version using cable routing instead of rigid links (Hidayah et al., 2018). Utilizing soft robotics techniques, a reduction in size and weight can also be achieved (Jin et al., 2018). As demonstrated in Awad et al. (2017), a low assistance soft exosuit that functions in synchrony with a wearer's paretic limb could facilitate an immediate increase in the paretic ankle's swing phase dorsiflexion and increase in the paretic limb's generation of forward propulsion. These improvements can result in a significant reduction in forward propulsion inter-limb asymmetry and reduced the energy cost of walking in ambulatory individuals after stroke, which is an important factor in both lower- and upper-limb soft rehabilitation robot designs (Xiloyannis et al., 2019). These compliance and comfort based systems have shown considerable advantages over traditional rigid exoskeleton designs, and have shown the ability to have similar beneficial outcomes such as increased foot clearance in stroke patient studies (Di Natali et al., 2019). Soft actuation and interfaces have a very promising future in lower-limb robot-assisted rehabilitation.

A powered leg orthosis applies suitable forces to move the leg on a desired trajectory using an assist as needed force-field controller and linear actuators at hip joint and knee joints in Banala et al. (2007b,a). This approach resists undesirable gait motion and provides assistance toward the desirable motion by applying forces at the foot of the subject (Banala et al., 2009). Furthermore, the work in Banala et al. (2010) showed that subjects with a force-field based control and with visual guidance produced considerable adaptation of their normal gait pattern toward the prescribed gait pattern when compared to a separate group receiving only visual guidance. An assist-as-needed paradigm with visual feedback is also a promising application for force-field based control methods in exoskeletons (Srivastava et al., 2015). Another active ankle-foot orthosis presented in Blaya and Herr (2004) tests the idea of modulating impedance of the orthotic joint throughout the gait cycle to treat drop-foot gait. Implementing an adaptive trajectory control to guide a patient's limb within a desired path (Bortole et al., 2013) allowed a deviation based on torque of interaction between the user and the system. This also used an admittance control strategy that allows the robotic platform to capture the user's movements during assistive training and replicates it during active training. Experimental results show that an exoskeleton can adapt a pre-recorded gait pattern of a specific user that can be adjusted by clinicians, then updated (Bortole et al., 2015) for future experiments.

3.4. Treadmill Training

Many of the exoskeleton implementations also make use of a treadmill for training in order to keep certain variables consistent, such as average walking speed. However, treadmill training can be used without use of any direct attachment or robotic device. The treadmill allows for the execution of many walking cycles in a relatively small and controlled space (Hesse, 2008). This allows for any sensors, motion capture camera systems or other data gathering systems to be placed near the subject for local experiments and trials. Training with a split-belt treadmill gives the ability to study of short-term motor adaptations when walking (Skidmore and Artemiadis, 2015), which have been shown to have improved long-term effects in post-stroke gait (Reisman et al., 2012).

3.5. Goal-Directed and Task-Oriented Training

Many treadmill-based systems have specialized functions that apply changes to the walking surface. Goal-directed movements that force the subject to produce specific movements can evoke muscle activity that may not be shown during normal, level ground walking. One type of non-conventional surface change used in rehabilitation is speed or direction variation. This is shown by authors in Choi and Bastian (2007) that set both sides of a split-belt treadmill to different speeds or in opposing directions, showing the ability of human motor adaptation. The authors in Choi and Bastian (2007) used a split-belt treadmill to induce motor adaptation by setting both sides of the belt to different speeds and in opposing directions. This technique has been shown to also increase gait speed when coupled with a Virtual Reality (VR) environment (Fung et al., 2006). Furthermore, inducing an unexpected acceleration of the trailing limb can have an increase in propulsive forces, which is a common metric for assessing walking ability (Farrens et al., 2019). This study also allowed the user to actively change the treadmill speed in real time, which has also shown promise of higher walking speed in stroke patients (Ray et al., 2020). Another split-belt treadmill training method that unilaterally changes the walking surface compliance has been shown to provide insight into the role of sensory feedback in perturbed gait, while highlighting mechanisms of inter-leg coordination (Skidmore and Artemiadis, 2016b,a). A change in slope, whether simulated with a tether (Hollerbach et al., 2001), implemented in foot plates (Iwata et al., 2002), or an actual change in level of a treadmill (Eng and Fang Tang, 2011), can be used to manipulate intensity of gait training and give another way to offer task-specific, eccentric therapy (Basso et al., 2018). Stair climbing is another intensive training method that has been tested (Hesse et al., 2010). A goal-directed or task-oriented therapy can be coupled with visual feedback to produce resulting muscle changes through obstacle avoidance or through targeting muscle activation objectives on a screen.

3.6. Electrical and Magnetic Stimulation

Instead of implementing robotic systems to interact with the subject, Functional Electrical Stimulation (FES) has been proposed to implement electrical excitation directly onto the muscle. When coordinated, induced muscle contractions can be useful for drop-foot prevention (Peckham and Knutson, 2005). This is used in some exoskeletons for spinal cord injured individuals (Schmitt et al., 2004; Farris et al., 2009a,b; Quintero et al., 2010, 2012; Ha et al., 2012, 2016), and to study the effects of synchronization while walking (Dohring and Daly, 2008). FES may improve the fitness and strength of stroke patients who still have a level of voluntary control (Tong et al., 2006). Moreover, it has been shown to produce positive results when used in conjunction with a treadmill (Hesse et al., 1995).

Transcranial Magnetic Stimulation (TMS) depolarizes cortical nerve membranes and discharges groups of neurons by an induced magnetic field near the cortex of the brain (Lamontagne et al., 2007). The Motor Evoked Potential (MEP) recorded in muscles has been studied in the past (Lotze et al., 2003; Forrester et al., 2006, 2009). This method has been used early on to study H-reflex (Petersen et al., 1998), stretch reflex (Shemmell et al., 2009; Zuur et al., 2009), and transcortical reflexes (Christensen L.O.D. et al., 2000). This disruption of electrical transmissions in the brain is generally considered safe and reversible (O'Dell et al., 2009). TMS and other transcranial stimulations have shown limited use in long-term post-stroke gait rehabilitation, but they can provide new opportunities to study supraspinal mechanisms and cortical activations that might provide useful insight for gait rehabilitation (Lamontagne et al., 2007).

4. TARGETED SENSORIMOTOR PATHWAYS

Stroke rehabilitation relies on the ability of the brain to recover through neuroplasticity. Neuroplasticity occurs when brain cells regenerate, re-establish, and rearrange neural connections in response to the damage inflicted by a stroke. Specifically on motor rehabilitation, physical therapy that engages sensori-motor mechanisms sparks neuroplasticity, encouraging the brain to correct mental and physical deficits (Morton and Bastian, 2006). This naturally places the third piece of the stroke rehabilitation puzzle: finding, evoking, and manipulating the neural mechanisms that take advantage of the brain's plasticity. Effective rehabilitation techniques maximize this neuroplasticity to achieve an optimal outcome for each patient (Gassert and Dietz, 2018). All rehabilitation methods should use ideas from this category in order to close the gap between neuroscience-based problems and engineering solutions.

4.1. Vision

Visual feedback has been utilized as a way provide sensory input to supraspinal mechanisms related to either the subject's position or motion in space. Modalities that have been used in the past focus on displaying spatial feedback (Unluhisarcikli et al., 2011) such as position, trajectory, progress, or statistics about movements, and typically entail moving a mechanical device attached to the subject's limb in order to hit some type of on screen target (Forrester et al., 2013), to maintain desired force (Forrester et al., 2006), or center of rotation (Nalam and Lee, 2019). This encourages patients to improve

their movements (Lunenburger et al., 2004) by activation of targeted muscle groups in order to improve functional outcomes. Those methods usually include video game-based therapy methods to enhance visuo-motor coordination while increase patient's engagement (Deutsch et al., 2009). Lately, Virtual Reality (VR) has been proposed as a more engaging and effective way to stimulate visuo-motor pathways and induce plasticity. This is a fast-growing way to implement a Virtual Environment (VE) most commonly by attaching a headset over the eyes, covering the entire visual space of the subject with the virtual environment. With VR, only a program and headset are required to interact with any physical setting, and can be even implemented in tele-rehabilitation training (Deutsch et al., 2007). People with disabilities including stroke show promise of motor learning within virtual environments (Holden, 2005), as well as increasing gait speed (Fung et al., 2006). However, there is a need to better understand the neural mechanisms that validates VR in the stroke rehabilitation field (Fluet and Deutsch, 2013). With a trained therapist, these systems can be used to "monitor, manipulate, and augment the users' interaction with their environment" toward functional recovery (Wade and Winstein, 2011). This has recently been implemented as a method for studying the effects of perturbations during gait over ground (Martelli et al., 2019), and showed promise when coupled with robotic implementations (Boian et al., 2002, 2005; Mirelman et al., 2009). Typically supplementing rehabilitation techniques already used, VR provides an environment that would normally require a real world setting, increasing the complexity in material set up. If a real-world environment is desired, distinguished from virtual reality, augmented reality places animated objects into the real-world environment. This has the advantage of appearing more realistic to the user and removes any disorientation stemming from VR environments. Since this technology is relatively new, implementations for rehabilitation that are coupled with various other physical implementations and interaction goals are still waiting to be discovered.

4.2. Audition

Supplementary feedback such as auditory could supplement or replace vision for wearable systems (Roby-Brami and Jarrassé, 2018), but is also used for socially assistive robots. These can use vocal cues to facilitate movement or provide encouragement and discouragement behavior, and when combined with robotic gestures, vocal grammar is an important part of interactions with the real world (Feil-Seifer and Matarić, 2005). Another application of auditory feedback is rhythmic auditory cueing. This is an approach that synchronizes gait to a rhythm to improve gait measures. There is moderate evidence of improved velocity and stride length in stroke patients after gait training with rhythmic auditory cueing (Winstein et al., 2016).

4.3. Equilibrioception

The sense of balance is another mechanism that is important for walking that uses visual and auditory feedback, as well as proprioception (Peshkin et al., 2005). The proprioceptive sense includes various muscle afferents with origin in muscle spindles and Golgi Tendon Organs (GTO's). Proprioception is the sense of having a known position of body parts relative to other parts of the body through regulation of the muscle activation amplitude during and in the switch between the gait phases (Rossignol et al., 2006). Because the muscle spindles are in parallel with the muscle, they provide accurate muscle length and velocity feedback through neural channels. Similarly, GTO's are in series with the tendons of the muscles and sense the muscle force. While the patient is performing either a static or dynamic activity the authors in Khan et al. (2018) show a system for posture training to reduce balance abnormalities by providing proprioceptive haptic feedback. Center of pressure, ground reaction forces and center of mass have been proven to be used by the brain during locomotion. Center of pressure is studied in a powered limb orthosis (Goldfarb et al., 2011) for the control interface to offer effective ways of providing sitting, standing, and walking functionality (Matjačić et al., 2018). The interplay of visual and proprioceptive feedback has also been shown through VR systems (Frost et al., 2015) with promising results.

4.4. Cutaneous and Haptic Perception

Haptic feedback is growing in popularity as a possible way to stimulate brain plasticity (Poli et al., 2013). The responses elicited during haptic resistance exercises for healthy individuals (Stegall et al., 2017) suggest that this feedback modality could be utilized for rehabilitation. In fact, haptic feedback may even allow for an increase in motor learning when compared to visual based error amplification (Marchal-Crespo et al., 2019). This unique modality has been shown to activate specific brain structures involved in error-processing (Milot et al., 2018). Haptic feedback has been used in lower limb exoskeletons for posture control (Khan et al., 2018) and conveying feedback information about a desired movement (Olenšek et al., 2018). This type of feedback is also useful for training in bedridden patients (Chisholm et al., 2014) and in this case, is especially useful in maintaining patient engagement (Berezny et al., 2019).

4.5. Inter-limb Coordination Mechanisms

Human walking requires coordination of muscle activation patterns between both legs, which seems to be achieved by a flexible neuronal coupling at a spinal level, with each limb affecting the behavior of the other (Swinnen et al., 2013). Typically, the initiation of the swing phase of one leg requires the contralateral leg to simultaneously be in the stance phase. This inter-limb coordination has been shown to be supraspinal based on muscle activation latency (Seiterle et al., 2015). From previous works, it is evident that inter-leg coordination in gait is a process that involves multiple feedback channels and processing of those signals in multiple levels (Christensen L. et al., 2000; Kuo, 2002; Dietz, 2003; Grillner, 2003; Nielsen, 2003; Rossignol et al., 2006; Yang and Gorassini, 2006; Choi and Bastian, 2007; Field-Fote and Dietz, 2007; Forrester et al., 2009; Grillner et al., 2008; Guertin, 2009; Norton, 2010; Petersen et al., 2012). Even though hemiparesis is typically seen as unilateral, almost all of the leg function is bilaterally organized

through neural circuitry explained by inter-limb coordination (Kautz and Patten, 2005).

Motivated by early studies of upper inter-limb coordination (Dietz and Berger, 1984; Berger et al., 1987; Sparrow et al., 1987; Kelso et al., 1979), quadrupedal inter-limb coordination (Forssberg et al., 1980), running (Whitall, 1989), and intra-limb coordination (Barela et al., 2000; Haddad et al., 2006; Presacco et al., 2012), unilateral treadmill-based perturbations have been used to study contralateral muscle responses (Dietz et al., 1989; Artemiadis and Krebs, 2011a,b; Skidmore and Artemiadis, 2017). Adults show adapted motor patterns of inter-limb coordination when experimented on split-belt treadmills with varying speeds on each side (Reisman et al., 2005). In the context of hemiparetic gait rehabilitation, the study of inter-limb coordination mechanisms might be of great significance. In fact, it has been shown that neural coupling exists in poststroke patients as it does in healthy subjects (Arya and Pandian, 2014) and for the upper limbs as well (Yoon et al., 2010). In studies with poststroke subjects with hemiparesis, it was found that neural decoupling between the lower limbs perturbs the paretic lower limb function (Kautz and Patten, 2005). It has been also shown that forceful interaction with the non-paretic leg elicits involuntary tension of the resting paretic leg where subjects are supine (Poskanzer, 1972). The central controller requires both locomotion patterns from spinal circuits, as well as neural drive through a multitude of descending pathways, such as proprioception (Poppele et al., 2003), that trigger desired gait corrections from various sensory modalities (Frost et al., 2015). Therefore, both the modeling of muscle activations (Skidmore and Artemiadis, 2016c) and mapping of the brain areas that seem to be involved (Debaere et al., 2001) in inter-limb coordination should be delved into further. From these principles, it is evident that understanding the sensorimotor network of inter-limb coordination is of paramount importance toward providing targeted rehabilitation to hemiparesis and improving the quality of life of patients suffering from it.

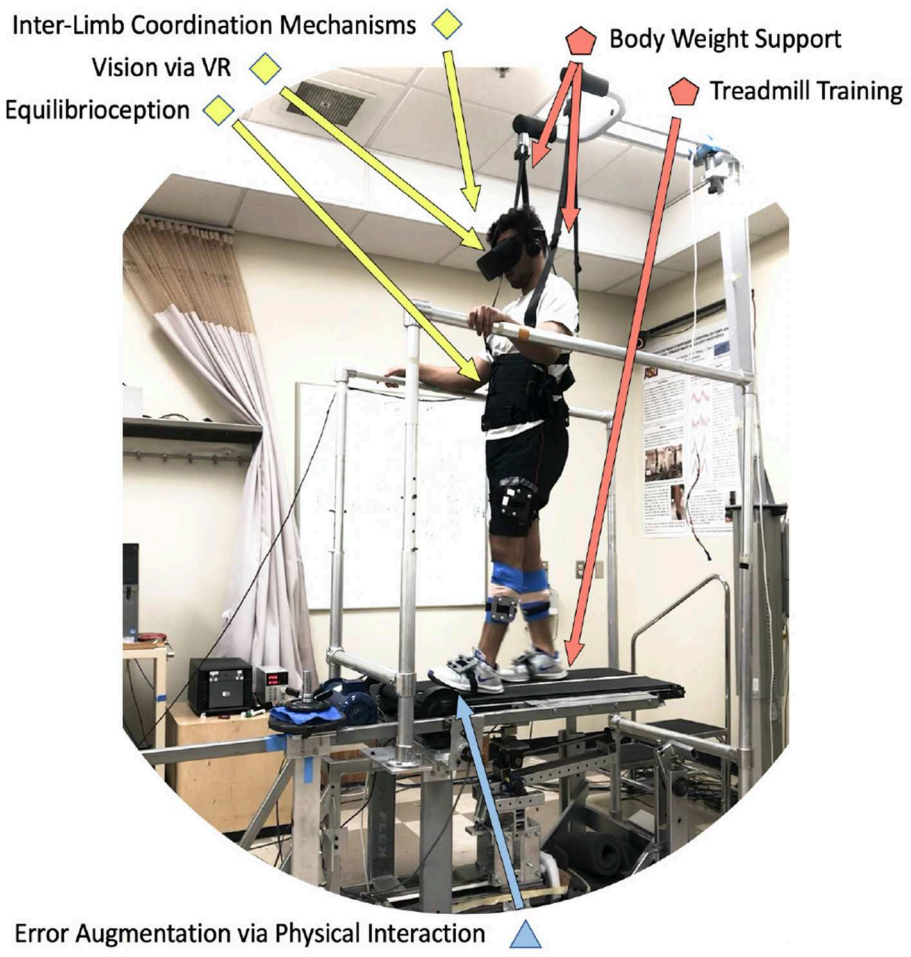


FIGURE 2 | Example of a protocol that uses six components of the proposed organizational chart. A subject wearing a virtual reality headset (*visual*) while walking on a split-belt treadmill with body-weight support (*treadmill training*, *body weight support*), is experiencing unexpected unilateral walking surface stiffness perturbations (*error augmentation*), which specifically evoke contralateral leg responses (*inter-limb coordination mechanisms*) by disturbing proprioceptive and balance feedback mechanisms (*equilibrioception*).

5. SYNOPSIS AND FUTURE DIRECTIONS

The goal of the paper is to consider previous research on robot-assisted rehabilitation through three different perspectives: the goal and type of interaction, the physical implementation, and the sensorimotor pathways targeted by the robotic devices utilized in the past. Combinations of approaches across groups that have not been attempted yet could lead to new approaches with improved outcomes. As new technologies are developed and new neural links to stroke affected patients are found, a increasingly large number of combinations for implementing these discoveries can be made using the proposed categorization. In other words, the categorization method presented allows for future scientists to fill research gaps with a more universal thought process.

An example of how this categorization could lead to new methods and approaches is illustrated in **Figure 2**. In this method, *inter-limb coordination mechanisms* are targeted via *error augmentation* disturbances in experimental setups that include *treadmill training* with *body weight support* through the interplay of *visual* and *equilibrioception*-based feedback. The method above is based on preliminary studies that are already being conducted with a novel device called the variable stiffness treadmill (VST), shown in **Figure 2**. The VST is a split-belt treadmill with which the compliance of the walking surface can be interactively and dynamically controlled. The VST consists of a spring-loaded lever mounted on a translational linear track that can change the effective stiffness under the foot by moving the linear track. An optical motion capture system monitors the location of the foot in real-time to control the timing of the stiffness perturbations throughout the gait cycle. The effective stiffness of each side/belt of the treadmill can range from 61.7 N/m to theoretically infinite (i.e., rigid walking surface), in 0.13 s. Furthermore, the resolution of the VST stiffness control is about 0.038 N/m (Skidmore et al., 2014, 2015).

According to the protocol followed in this study, a subject wearing a virtual reality headset (*visual*) while walking on a split-belt treadmill with body-weight support (*treadmill training*, *body weight support*), is experiencing unexpected unilateral walking surface stiffness perturbations (*error augmentation*), that specifically evoke contralateral leg responses (*inter-limb coordination mechanisms*) by disturbing proprioceptive and balance feedback mechanisms (*equilibrioception*). Preliminary results with these tools suggest that muscle and brain activity

TABLE 1 | Literature summary categorized via the proposed organization.

Device name and Studies	Augmentation	correction	Non-physical	Goal/Task-oriented	Stimulation	Exoskeleton	Body weight support	Treadmill	Footplates	Inter-limb coordination	Haptic/Cutaneous	Equilibrioception	Audition	Vision	Patient tested
MIT-Skywalker—Artemiadis and Krebs, 2011a,b; Seiterle et al., 2015	•			•			•	•		•	•	•			
Ankle robot—Saglia et al., 2009	•								•		•				
BAR-TM—Matjačić et al., 2018; Olenšek et al., 2018	•							•			•	•		•	•
VST–Barkan et al., 2014; Skidmore et al., 2014; Frost et al., 2015; Skidmore and Artemiadis, 2015; Skidmore et al., 2015; Skidmore and Artemiadis, 2016a,b,c,d, 2017	•			•			•	•		•	•	•		•	•
LOPES—Veneman et al., 2007; Koopman et al., 2013	•	•		•		•	•	•			•	•		•	•
Active/Passive AFO—Barela et al., 2000; Blaya and Herr, 2004; Agrawal et al., 2005; Hwang et al., 2006	•	•		•		•				•	•	•			•
Anklebot—Roy et al., 2007; Forrester et al., 2011, 2013	•	•		•		•					•			•	•
KineAssist—Peshkin et al., 2005	•	•		•			•				•	•			
BWS treadmill—Hesse et al., 1999; Poppele et al., 2003; Haddad et al., 2006; Choi and Bastian, 2007; Field-Fote and Dietz, 2007; Petersen et al., 2012; Presacco et al., 2012; Reisman et al., 2012	•	•		•			•	•		•	•	•			•
NUVABAT—Ding et al., 2010	•	•							•		•	•		•	
Rutgers ankle—Girone et al., 2001; Boian et al., 2002; Deutsch et al., 2007	•	•		•					•		•		•	•	
The gait master—Iwata et al., 2002	•	•		•					•		•	•			
Lokomat—Colombo et al., 2001; Jezernik et al., 2003; Hornby et al., 2005; Lunenburger et al., 2004; Mayr et al., 2007; Heller et al., 2007; Dohring and Daly, 2008; Hidler et al., 2009; Klarner, 2010; Chang et al., 2012; Krishnan et al., 2012	•	•			•	•	•	•			•	•		•	•
ARTHuR—Reinkensmeyer et al., 2003; Emken et al., 2005	•	•				•					•	•			

TABLE 2 | Literature summary categorized via the proposed organization (continued).

Device name and studies	Augmentation	Correction	Non-physical	Goal/task-oriented	Stimulation	Exoskeleton	Body weight support	Treadmill	Footplates	Inter-limb coordination	Haptic/Cutaneous	Equilibrioception	Audition	Vision	Patient tested
HapticWalker—Schmidt et al., 2005a; Schmidt and Werner, 2007	•	•					•		•		•				•
RMA—Boian et al., 2005	•	•					•		•		•	•		•	
Trunk Support Trainer – Khan et al., 2018	•	•									•	•			
Lambda—Bouri et al., 2009		•							•		•				
Gait Trainer—Hesse et al., 2000; Peurala et al., 2009; Hoölig et al., 2007		•					•		•		•	•			•
Vanderbilt lower limb exoskeleton—Farris et al., 2012; Goldfarb et al., 2003; Ha et al., 2012, 2016; Farris et al., 2014		•		•	•	•					•	•			•
ALEX—Banala et al., 2007a,b, 2009, 2010; Srivastava et al., 2015; Stegall et al., 2017; Hidayah et al., 2018; Jin et al., 2018		•		•		•	•	•			•	•		•	•
HAL—Nilsson et al., 2014; Kawamoto and Sankai, 2007		•		•		•	•				•	•			•
Lokohelp—Freivogel et al., 2008, 2009		•		•			•	•	•		•	•			•
G-EO-Systems Robot—Hesse et al., 2010		•		•			•		•		•				•
ViGGR—Chisholm et al., 2014		•		•							•			•	
JCO—Farris et al., 2009a,b; Quintero et al., 2010		•			•	•					•	•			
Motion Maker—Schmitt et al., 2004		•			•	•					•				
DGO—Colombo et al., 2000		•				•	•	•			•	•			
PAM/POGO—Aoyagi et al., 2007		•				•	•	•			•	•			
WALKBOT—Kim et al., 2015		•				•	•	•			•	•		•	•
WalkTrainer—Bouri et al., 2006; Stauffer, 2009		•				•	•				•	•			•
RGT—Bharadwaj et al., 2005; Bharadwaj and Sugar, 2006		•				•					•				
ANdROS—Unluhisarcikli et al., 2011		•				•					•			•	
Gait Rehabilitation Exoskeleton—Beyl et al., 2008		•				•					•	•			
LLRR– Chen et al., 2009		•					•		•		•	•			
NEUROBike—Monaco et al., 2009		•							•		•				
ROREAS—Gross et al., 2014			•	•										•	•

is evoked on one leg when the stiffness of the walking surface for the other leg is perturbed (Skidmore et al., 2014; Skidmore and Artemiadis, 2015, 2016a,d). The specific outcomes targeted in this study include evoked activity on the ankle muscles of the unperturbed leg, which is very encouraging since this can provide solutions to the problem of drop-foot that most impaired walkers suffer from, and it is the leading cause of after-stroke falls. Recent findings with hemiplegic walkers provide strong evidence that a new method for providing gait rehabilitation could entail evoking activity on the paretic side by introducing unilateral perturbations on the healthy side of hemiplegic walkers (Skidmore and Artemiadis, 2016b, 2017). Therefore, the combination of this type of interactive treadmill system with a variety of specifically timed physical perturbations can significantly broaden our scientific understanding of gait and can open new avenues of research in rehabilitation focusing on the neural and mechanical coupling of the legs, while going beyond the single-leg intervention approaches currently followed.

The above combination of methods and approaches is not unique, and by no means exclusive to what needs to be included in a comprehensive approach in gait rehabilitation. Ideally, selection of approaches ought to be done in the context of a review of gaps and weaknesses found in the empirical evidence. These gaps can be identified using the categorization of approaches this paper introduces. More specifically, **Tables 1**, **2** provide a comprehensive list of devices and methods used in the past for gait rehabilitation, and how these previous studies can be categorized based on the proposed perspective[1]. Moreover, the tables show if the devices have been tested with patients or not. Although the effectiveness of each approach is not mentioned—and is quite difficult to be assessed and compared across studies—it is important and useful to the future researcher

[1]It must be noted that the entries in the **Tables 1**, **2** are organized in terms of the device used, and how each device makes use of the components discussed in this review paper. When a device has been used in multiple previous works or studies, the cumulative set of components across all papers listed is shown at the tables.

to be able to see how each device or method uses the components of the three categories discussed here, and identify gaps and potential opportunities.

In conclusion, this paper provides a potential solution to the overwhelming number of gait therapy methods based from the need for utilizing the methods that work and combining them in organized ways to produce new methods, which can potentially have improved outcomes. An example of using this categorization to come up with new methods for rehabilitation, such as perturbation-based approaches using inter-limb coordination mechanisms, is demonstrated. However, this is only one of the possible seeds of new approaches that could sprout from this framework. The authors strongly believe that this new perspective of mixing and matching hardware, procedures, algorithms, and intended neural pathways could lead to more focused research and eventually significant advances in lower-limb robot-assisted stroke rehabilitation.

AUTHOR CONTRIBUTIONS

BH and PA contributed conception of the paper. BH organized the database of the references included and wrote the first draft of the manuscript. PA wrote sections of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This material was based upon work supported by the National Science Foundation under Grants No. #1727838 and #2015786.

REFERENCES

Agostini, V., Ghislieri, M., Taborri, J., Roh, J., Jacobs, D. A., Rossi, S., et al. (2018). Feasibility of muscle synergy outcomes in clinics, robotics, and sports: a systematic review. *Appl. Bion. Biomech.* 2018:3934698. doi: 10.1155/2018/3934698

Agrawal, A., Banala, S. K., Agrawal, S. K., and Binder-Macleod, S. A. (2005). "Design of a two degree-of-freedom ankle-foot orthosis for robotic rehabilitation," in *Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics* (Chicago, IL), 41–44. doi: 10.1109/ICORR.2005.1501047

Albert, S. J., and Kesselring, J. (2012). Neurorehabilitation of stroke. *J. Neurol.* 259, 817–832. doi: 10.1007/s00415-011-6247-y

Aoyagi, D., Ichinose, W., Harkema, S. J., Reinkensmeyer, D. J., and Bobrow, J. E. (2007). A robot and control algorithm that can synchronously assist in naturalistic motion during body-weight-supported gait training following neurologic injury daisuke. *IEEE Trans. Neural Syst. Rehabil. Eng.* 15:387. doi: 10.1109/TNSRE.2007.903922

Artemiadis, P. K., and Krebs, H. I. (2011a). "Interlimb coordination evoked by unilateral mechanical perturbation during body-weight supported gait," in *2011 IEEE International Conference on Rehabilitation Robotics* (Boston, MA), 1–5. doi: 10.1109/ICORR.2011.5975513

Artemiadis, P. K., and Krebs, H. I. (2011b). "On the interlimb coordination and synchronization during gait," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Shanghai), 1571–1574. doi: 10.1109/IEMBS.2011.6090457

Arya, K. N., and Pandian, S. (2014). Interlimb neural coupling: implications for poststroke hemiparesis. *Ann. Phys. Rehabil. Med.* 57, 696–713. doi: 10.1016/j.rehab.2014.06.003

Awad, L. N., Bae, J., O'donnell, K., De Rossi, S. M. M., Hendron, K., Sloot, L. H., et al. (2017). A soft robotic exosuit improves walking in patients after stroke. *Sci Transl Med.* 9:eaai9084. doi: 10.1126/scitranslmed.aai9084

Banala, S. K., Agrawal, S. K., Kim, S. H., and Scholz, J. P. (2010). Novel gait adaptation and neuromotor training results using an active leg exoskeleton. *IEEE/ASME Trans. Mechatron.* 15, 216–225. doi: 10.1109/TMECH.2010.2041245

Banala, S. K., Agrawal, S. K., and Scholz, J. P. (2007a). "Active Leg Exoskeleton (ALEX) for gait rehabilitation of motor-impaired patients," in *2007 IEEE 10th International Conference on Rehabilitation Robotics* (IEEE), 401–407. doi: 10.1109/ICORR.2007.4428456

Banala, S. K., Kim, S. H., Agrawal, S. K., and Scholz, J. P. (2009). Robot assisted gait training with Active Leg Exoskeleton (ALEX). *Rehabilitation* 17, 2–8. doi: 10.1109/TNSRE.2008.2008280

Banala, S. K., Kulpe, A., and Agrawal, S. K. (2007b). "A powered leg orthosis for gait rehabilitation of motor-impaired patients," in *Proceedings 2007 IEEE International Conference on Robotics and Automation* (Rome: IEEE), 4140–4145. doi: 10.1109/ROBOT.2007.364115

Barela, J. A., Whitall, J., Black, P., and Clark, J. E. (2000). An examination of constraints affecting the intralimb coordination of hemiparetic gait. *Hum. Mov. Sci.* 19, 251–273. doi: 10.1016/S0167-9457(00)00014-2

Barkan, A., Skidmore, J., and Artemiadis, P. (2014). "Variable Stiffness Treadmill (VST): a novel tool for the investigation of gait," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong: IEEE), 2838–2843. doi: 10.1109/ICRA.2014.6907266

Basso, D. M., Lang, C. E., Sciences, R., State, T. O., and Therapy, O. (2018). Consideration of dose and timing after stroke. 41(Suppl. 3), S24–S31. doi: 10.1097/NPT.0000000000000165

Belforte, G., Gastaldi, L., and Sorli, M. (2001). Pneumatic active gait orthosis. *Mechatronics* 11, 301–323. doi: 10.1016/S0957-4158(00)00017-9

Benjamin, E. J., Muntner, P., Alonso, A., Bittencourt, M. S., Callaway, C. W., Carson, A. P., et al. (2019). Heart disease and stroke statistics-2019 Update: A report from the american heart association. *Circulation* 139:e56–e528. doi: 10.1161/CIR.0000000000000659

Berezny, N., Dowlatshahi, D., and Ahmadi, M. (2019). Interaction control and haptic feedback for a lower-limb rehabilitation robot with virtual environments. *Int. Conf. Control Dyn. Syst. Robot.* 10:11159. doi: 10.11159/cdsr19.145

Berger, W., Dietz, V., and Horstmann, G. (1987). *Interlimb Coordination of Posture in Man, Vol. 390*. Cambridge New York, NY: University Press.

Beyl, P., Van Damme, M., Van Ham, R., Versluys, R., Vanderborght, B., and Lefeber, D. (2008). "An exoskeleton for gait rehabilitation: prototype design and control principle," in *Proceedings - IEEE International Conference on Robotics and Automation* (Pasadena, CA), 2037–2042. doi: 10.1109/ROBOT.2008.4543506

Bharadwaj, K., and Sugar, T. G. (2006). "Kinematics of a robotic gait trainer for stroke rehabilitation," in *Proceedings - IEEE International Conference on Robotics and Automation* (Orlando, FL), 3492–3497. doi: 10.1109/ROBOT.2006.1642235

Bharadwaj, K., Sugar, T. G., Koeneman, J. B., and Koeneman, E. J. (2005). Design of a robotic gait trainer using spring over muscle actuators for ankle stroke rehabilitation. *J. Biomech. Eng.* 127:1009. doi: 10.1115/1.2049333

Bishop, L., Khan, M., Martelli, D., Quinn, L., Stein, J., and Agrawal, S. (2017). Exploration of two training paradigms using forced induced weight shifting with the tethered pelvic assist device to reduce asymmetry in individuals after stroke case reports. *Am. J. Phys. Med. Rehabil.* 96, S135–S140. doi: 10.1097/PHM.0000000000000779

Blaya, J., and Herr, H. (2004). Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait. *IEEE Trans. Neural Syst. Rehabil. Eng.* 12, 24–31. doi: 10.1109/TNSRE.2003.823266

Boian, R., Lee, C., Deutsch, J., Burdea, G., and Lewis, J. (2002). "Virtual reality-based system for ankle rehabilitation post stroke," in *Proceedings of 1st International Workshop on Virtual Reality Rehabilitation (Mental Health, Neurological, Physical, Vocational)* (Lausanne), 77–86.

Boian, R. F., Bouzit, M., Burdea, G. C., Lewis, J., and Deutsch, J. E. (2005). "Dual stewart platform mobility simulator," in *Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics*, 550–555. doi: 10.1109/ICORR.2005.1502023

Bortole, M., Del Ama, A., Rocon, E., Moreno, J. C., Brunetti, F., and Pons, J. L. (2013). "A robotic exoskeleton for overground gait rehabilitation," in *Proceedings - IEEE International Conference on Robotics and Automation* (Karlsruhe), 3356–3361. doi: 10.1109/ICRA.2013.6631045

Bortole, M., Venkatakrishnan, A., Zhu, F., Moreno, J. C., Francisco, G. E., Pons, J. L., et al. (2015). The H2 robotic exoskeleton for gait rehabilitation after stroke: early findings from a clinical study. *J. NeuroEng. Rehabil.* 12:54. doi: 10.1186/s12984-015-0048-y

Bouri, M., Le Gall, B., and Clavel, R. (2009). "A new concept of parallel robot for rehabilitation and fitness: the Lambda," in *2009 IEEE International Conference on Robotics and Biomimetics, ROBIO 2009* (Guilin), 2503–2508. doi: 10.1109/ROBIO.2009.5420481

Bouri, M., Stauffer, Y., Schmitt, C., Allemand, Y., Gnemmi, S., Metrailler, P., et al. (2006). "The WalkTrainerTM : a robotic system for walking," in *Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics* (Kunming), 1616–1621. doi: 10.1109/ROBIO.2006.340186

Bruni, M. F., Melegari, C., De Cola, M. C., Bramanti, A., Bramanti, P., and Calabrò, R. S. (2018). What does best evidence tell us about robotic gait rehabilitation in stroke patients: a systematic review and meta-analysis. *J. Clin. Neurosci.* 48, 11–17. doi: 10.1016/j.jocn.2017.10.048

Calabrò, R. S., Cacciola, A., Bertè, F., Manuli, A., Leo, A., Bramanti, A., et al. (2016). Robotic gait rehabilitation and substitution devices in neurological disorders: where are we now? *Neurol. Sci.* 37, 503–514. doi: 10.1007/s10072-016-2474-4

Carter, A. R., Connor, L. T., and Dromerick, A. W. (2010). Rehabilitation after stroke: current state of the science. *Curr. Neurol. Neurosci. Rep.* 10, 158–166. doi: 10.1007/s11910-010-0091-9

Chang, W. H., Kim, M. S., Huh, J. P., Lee, P. K. W., and Kim, Y.-H. (2012). Effects of robot-assisted gait training on cardiopulmonary fitness in subacute stroke patients. *Neurorehabil. Neural Repair* 26, 318–324. doi: 10.1177/1545968311408916

Chang, W. H., and Kim, Y.-H. (2013). Robot-assisted therapy in stroke rehabilitation. *J. Stroke* 15, 174–181. doi: 10.5853/jos.2013.15.3.174

Chen, S., Wang, Y., Li, S., Wang, G., Huang, Y., and Mao, X. (2009). "Lower limb rehabilitation robot," in *2009 ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots* (London, UK), 439–443.

Chisholm, K. J., Klumper, K., Mullins, A., and Ahmadi, M. (2014). A task oriented haptic gait rehabilitation robot. *Mechatronics* 24, 1083–1091. doi: 10.1016/j.mechatronics.2014.07.001

Choi, J. T., and Bastian, A. J. (2007). Adaptation reveals independent control networks for human walking. *Nat. Neurosci.* 10, 1055–1062. doi: 10.1038/nn1930

Christensen, L., Johannsen, P., Sinkjær, T., Petersen, N., Pyndt, H., and Nielsen, J. (2000a). Cerebral activation during bicycle movements in man. *Exp. Brain Res.* 135, 66–72. doi: 10.1007/s002210000493

Christensen, L. O. D., Petersen, N., Andersen, J. B., Sinkjær, T., and Nielsen, J. B. (2000). Evidence for transcortical reflex pathways in the lower limb of man. *Prog. Neurobiol.* 62, 251–272. doi: 10.1016/S0301-0082(00)00007-1

Colombo, G., Joerg, M., Schreier, R., and Dietz, V. (2000). Treadmill training of paraplegic patients using a robotic orthosis. *J Rehabil. Res. Dev.* 37:693–700.

Colombo, G., Wirz, M., and Dietz, V. (2001). Driven gait orthosis for improvement of locomotor training in paraplegic patients. *Spinal Cord* 39, 252–255. doi: 10.1038/sj.sc.3101154

Conesa, L., Costa, Ú., Morales, E., Edwards, D. J., Cortes, M., León, D., et al. (2012). An observational report of intensive robotic and manual gait training in sub-acute stroke. *J. NeuroEng. Rehabil.* 9:13. doi: 10.1186/1743-0003-9-13

Debaere, F., Swinnen, S. P., Béatse, E., Sunaert, S., Van Hecke, P., and Duysens, J. (2001). Brain areas involved in interlimb coordination: a distributed network. *NeuroImage* 14, 947–958. doi: 10.1006/nimg.2001.0892

Deutsch, J. E., Lewis, J. A., and Burdea, G. (2007). Technical and patient performance using a virtual reality-integrated telerehabilitation system: preliminary finding. *IEEE Trans. Neural Syst. Rehabil. Eng.* 15, 30–35. doi: 10.1109/TNSRE.2007.891384

Deutsch, J. E., Robbins, D., Morrison, J., and Guarrera Bowlby, P. (2009). "WII-based compared to standard of care balance and mobility rehabilitation for two individuals post-stroke," in *2009 Virtual Rehabilitation International Conference, VR 2009* (Haifa), 117–120.

Di Natali, C., Poliero, T., Sposito, M., Graf, E., Bauer, C., Pauli, C., et al. (2019). Design and evaluation of a soft assistive lower limb exoskeleton. *Robotica*. doi: 10.1017/S0263574719000067

Díaz, I., Gil, J. J., and Sánchez, E. (2011). Lower-limb robotic rehabilitation: literature review and challenges. *J. Robot.* 2011:759764. doi: 10.1155/2011/759764

Dickstein, R. (2008). Rehabilitation of gait speed after stroke: a critical review of intervention approaches. *Neurorehabil. Neural Repair* 22, 649–660. doi: 10.1177/1545968308315997

Dietz, V. (2003). Spinal cord pattern generators for locomotion. *Clin. Neurophysiol.* 114, 1379–1389. doi: 10.1016/S1388-2457(03)00120-2

Dietz, V., and Berger, W. (1984). Interlimb coordination of posture in patients with spastic paresis: impaired function of spinal reflexes. *Brain* 107, 965–978. doi: 10.1093/brain/107.3.965

Dietz, V., Horstmann, G. A., and Berger, W. (1989). Interlimb coordination of leg-muscle activation during perturbation of stance in humans. *J. Neurophysiol.* 62, 680–693. doi: 10.1152/jn.1989.62.3.680

Ding, Y., Sivak, M., Weinberg, B., Mavroidis, C., and Holden, M. K. (2010). "NUVABAT: Northeastern University Virtual Ankle and Balance Trainer," in *2010 IEEE Haptics Symposium, HAPTICS 2010* (Waltham, MA), 509–514.

Dohring, M. E., and Daly, J. J. (2008). Automatic synchronization of functional electrical stimulation and robotic assisted treadmill training. *IEEE Trans. Neural Syst. Rehabil. Eng.* 16, 310–313. doi: 10.1109/TNSRE.2008.920081

Emken, J. L., Wynne, J. H., Harkema, S. J., and Reinkensmeyer, D. J. (2005). A robotic device for measuring and manipulating stepping. *IEEE Trans. Robot.* 22, 185–189. doi: 10.1109/TRO.2005.861481

Eng, J. J., and Fang Tang, P. (2011). Gait training strategies to optimize walking ability in people with stroke: a synthesis of the evidence. *Expert Rev. Neurother.* 7, 1417–1436. doi: 10.1586/14737175.7.10.1417

Farrens, A. J., Marbaker, R., Lilley, M., and Sergi, F. (2019). Training propulsion: locomotor adaptation to accelerations of the trailing limb. *IEEE Int. Conf. Rehabil. Robot.* 2019, 59–64. doi: 10.1109/ICORR.2019.8779374

Farris, R. J., Quintero, H. A., and Goldfarb, M. (2011a). Preliminary evaluation of a powered lower limb orthosis to aid walking in paraplegic individuals. *IEEE Trans. Neural Syst. Rehabil. Eng.* 19, 652–659. doi: 10.1109/TNSRE.2011.2163083

Farris, R. J., Quintero, H. A., and Goldfarb, M. (2012). "Performance evaluation of a lower limb exoskeleton for stair ascent and descent with Paraplegia," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* (San Diego, CA), 1908–1911.

Farris, R. J., Quintero, H. A., Goldfarb, M., Farris, R. J., Goldfarb, M., Quintero, H. A., et al. (2011b). Control and implementation of a powered lower limb orthosis to aid walking in paraplegic individuals. *IEEE Int. Conf. Rehabil. Robot.* 19, 1–6. doi: 10.1109/ICORR.2011.5975481

Farris, R. J., Quintero, H. A., Murray, S. A., Ha, K. H., Hartigan, C., and Goldfarb, M. (2014). A preliminary assessment of legged mobility provided by a lower limb exoskeleton for persons with paraplegia. *IEEE Trans. Neural Syst. Rehabil. Eng.* 22, 482–490. doi: 10.1109/TNSRE.2013.2268320

Farris, R. J., Quintero, H. A., Withrow, T. J., and Goldfarb, M. (2009a). "Design and simulation of a joint-coupled orthosis for regulating FES-aided gait," in *2009 IEEE International Conference on Robotics and Automation* (Kobe), 1916–1922. doi: 10.1109/ROBOT.2009.5152634

Farris, R. J., Quintero, H. A., Withrow, T. J., and Goldfarb, M. (2009b). "Design of a joint-coupled orthosis for FES-aided gait," in *2009 IEEE International Conference on Rehabilitation Robotics, ICORR 2009* (Kyoto), 246–252.

Feil-Seifer, D., and Matarić, M. J. (2005). "Defining socially assistive robotics," in *Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics* (Chicago, IL), 465–468. doi: 10.1109/ICORR.2005.1501143

Ferris, D., Sawicki, G., Domingo, A., Daniel, P. F. P., Gregory, S. S. M., and Antoinette, R. D. M. P. T. (2005). Powered lower limb orthoses for gait rehabilitation. *Top. Spinal Cord Inj. Rehabil.* 11, 34–49. doi: 10.1310/6GL4-UM7X-519H-9JYD

Ferris, D. P., Gordon, K. E., Sawicki, G. S., and Peethambaran, A. (2006). An improved powered ankle–foot orthosis using proportional myoelectric control. *Gait Posture* 23, 425–428. doi: 10.1016/j.gaitpost.2005.05.004

Field-Fote, E. C., and Dietz, V. (2007). Single joint perturbation during gait: preserved compensatory response pattern in spinal cord injured subjects. *Clin. Neurophysiol.* 118, 1607–1616. doi: 10.1016/j.clinph.2007.03.022

Fluet, G. G., and Deutsch, J. E. (2013). Virtual reality for sensorimotor rehabilitation post-stroke: the promise and current state of the field. *Curr. Phys. Med. Rehabil. Rep.* 1, 9–20. doi: 10.1007/s40141-013-0005-2

Forrester, L. W., Hanley, D. F., and Macko, R. F. (2006). Effects of treadmill exercise on transcranial magnetic stimulation-induced excitability to quadriceps after stroke. *Arch. Phys. Med. Rehabil.* 87, 229–234. doi: 10.1016/j.apmr.2005.10.016

Forrester, L. W., Roy, A., Goodman, R. N., Rietschel, J., Barton, J. E., Krebs, H. I., et al. (2013). Clinical application of a modular ankle robot for stroke rehabilitation. *NeuroRehabilitation* 33, 85–97. doi: 10.3233/NRE-130931

Forrester, L. W., Roy, A., Krebs, H. I., and Macko, R. F. (2011). Ankle training with a robotic device improves hemiparetic gait after a stroke. *Neurorehabil. Neural Repair* 25, 369–377. doi: 10.1177/1545968310388291

Forrester, L. W., Wheaton, L. A., and Luft, A. R. (2009). Exercise-mediated locomotor recovery and lower-limb neuroplasticity after stroke. *J. Rehabil. Res. Dev.* 45, 205–220. doi: 10.1682/JRRD.2007.02.0034

Forssberg, H., Grillner, S., Halbertsma, J., and Rossignol, S. (1980). The locomotion of the low spinal cat. II. Interlimb coordination. *Acta Physiol. Scand.* 108, 283–295. doi: 10.1111/j.1748-1716.1980.tb06534.x

Freivogel, S., Mehrholz, J., and Husak-Sotomayor, T. (2008). *Gait Training With the Newly Developed 'LokoHelp'-System Is Feasible for Non-ambulatory Patients After Stroke, Spinal Cord and Brain Injury. A Feasibility Study*. Taylor & Francis.

Freivogel, S., Schmalohr, D., and Mehrholz, J. (2009). Improved walking ability and reduced therapeutic stress with an electromechanical gait device. *J. Rehabil. Med.* 41, 734–739. doi: 10.2340/16501977-0422

Frost, R., Skidmore, J., Santello, M., and Artemiadis, P. (2015). Sensorimotor control of gait: a novel approach for the study of the interplay of visual and proprioceptive feedback. *Front. Hum. Neurosci.* 9:14. doi: 10.3389/fnhum.2015.00014

Fung, J., Richards, C. L., Malouin, F., McFadyen, B. J., and Lamontagne, A. (2006). A treadmill and motion coupled virtual reality system for gait training post-stroke. *CyberPsychol. Behav.* 9, 157–162. doi: 10.1089/cpb.2006.9.157

Gassert, R., and Dietz, V. (2018). Rehabilitation robots for the treatment of sensorimotor deficits: a neurophysiological perspective. *J. NeuroEng. Rehabil.* 15, 1–15. doi: 10.1186/s12984-018-0383-x

Girone, M., Burdea, G., Bouzit, M., Popescu, V., and Deutsch, J. E. (2001). A Stewart Platform-Based System for Ankle Telerehabilitation. *Auton. Robot.* 10:203–212. doi: 10.1023/A:1008938121020

Goldfarb, M., Korkowski, K., Harrold, B., and Durfee, W. (2003). Preliminary evaluation of a controlled-brake orthosis for FES-aided gait. *IEEE Trans. Neural Syst. Rehabil. Eng.* 11, 241–248. doi: 10.1109/TNSRE.2003.816873

Goldfarb, M., Quintero, H., Farris, R., Hartigan, C., Clesson, I., and Goldfarb, M. (2011). A powered lower limb orthosis for providing legged mobility in paraplegic individuals. *Top. Spinal Cord Injury Rehabil.* 17, 25–33. doi: 10.1310/sci1701-25

Grillner, S. (2003). The motor infrastructure: from ion channels to neuronal networks. *Nat. Rev. Neurosci.* 4:573. doi: 10.1038/nrn1137

Grillner, S., Wallén, P., Saitoh, K., Kozlov, A., and Robertson, B. (2008). Neural bases of goal-directed locomotion in vertebrates–An overview. *Brain Res. Rev.* 57, 2–12. doi: 10.1016/j.brainresrev.2007.06.027

Gross, H. M., Debes, K., Einhorn, E., Mueller, S., Scheidig, A., Weinrich, C., et al. (2014). "Mobile robotic rehabilitation assistant for walking and orientation training of stroke patients: a report on work in progress," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics* (San Diego, CA), 1880–1887.

Guertin, P. A. (2009). The mammalian central pattern generator for locomotion. *Brain Res. Rev.* 62, 45–56. doi: 10.1016/j.brainresrev.2009.08.002

Ha, K. H., Murray, S. A., and Goldfarb, M. (2016). An approach for the cooperative control of FES with a powered exoskeleton during level walking for persons with paraplegia. *IEEE Trans. Neural Syst. Rehabil. Eng.* 24, 455–466. doi: 10.1109/TNSRE.2015.2421052

Ha, K. H., Quintero, H. A., Farris, R. J., and Goldfarb, M. (2012). "Enhancing stance phase propulsion during level walking by combining fes with a powered exoskeleton for persons with paraplegia," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* (San Diego, CA), 344–347.

Haddad, J. M., van Emmerik, R. E. A., Whittlesey, S. N., and Hamill, J. (2006). Adaptations in interlimb and intralimb coordination to asymmetrical loading in human walking. *Gait Posture* 23, 429–434. doi: 10.1016/j.gaitpost.2005.05.006

He, Y., Nathan, K., Venkatakrishnan, A., Rovekamp, R., Beck, C., Ozdemir, R., et al. (2014). "An integrated neuro-robotic interface for stroke rehabilitation using the NASA X1 powered lower limb exoskeleton," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014* (Chicago, IL), 3985–3988.

Heller, S., Müller, F., Koenig, E., Husemann, B., Krewer, C., Müller, F., et al. (2007). Effects of locomotion training with assistance of a robot-driven gait orthosis in hemiparetic patients after stroke. *Stroke* 38, 349–354. doi: 10.1161/01.STR.0000254607.48765.cb

Hesse, S. (2008). Treadmill training with partial body weight support after stroke: a review. *NeuroRehabilitation* 23, 55–65. doi: 10.3233/NRE-2008-23106

Hesse, S., Konrad, M., and Uhlenbrock, D. (1999). Treadmill walking with partial body weight support versus floor walking in hemiparetic subjects. *Arch. Phys. Med. Rehabil.* 80, 421–427. doi: 10.1016/S0003-9993(99)90279-4

Hesse, S., Malezic, M., Schaffrin, A., and Mauritz, K. H. (1995). Restoration of gait by combined treadmill training and multichannel electrical stimulation in non-ambulatory hemiparetic patients. *Scand. J. Rehabil. Med.* 27, 199–204.

Hesse, S., Uhlenbrock, D., Werner, C., and Bardeleben, A. (2000). A mechanized gait trainer for restoring gait in nonambulatory subjects. *Arch Phys Med Rehabil* 81:1158–1161. doi: 10.1053/apmr.2000.6280

Hesse, S., Waldner, A., and Tomelleri, C. (2010). Innovative gait robot for the repetitive practice of floor walking and stair climbing up and down in stroke patients. *J. NeuroEng. Rehabil.* 7:30. doi: 10.1186/1743-0003-7-30

Hesse, S., and Werner, C. (2009). Connecting research to the needs of patients and clinicians. *Brain Res. Bull.* 78, 26–34. doi: 10.1016/j.brainresbull.2008.06.004

Hidayah, R., Jin, X., Chamarthy, S., Fitzgerald, M. M., and Agrawal, S. K. (2018). "Comparing the performance of a cable-driven active leg exoskeleton (C-ALEX) over-ground and on a treadmill," in *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics* (Enschede), 299–304. doi: 10.1109/BIOROB.2018.8487771

Hidler, J., Nichols, D., Pelliccio, M., Brady, K., Campbell, D. D., Kahn, J. H., et al. (2009). Multicenter randomized clinical trial evaluating the effectiveness of the lokomat in subacute stroke. *Neurorehabil. Neural Repair* 23, 5–13. doi: 10.1177/1545968308326632

Hogan, N., Krebs, H. I., Rohrer, B., Palazzolo, J. J., Dipietro, L., Fasoli, S. E., et al. (2006). Motions or muscles? Some behavioral factors underlying robotic assistance of motor recovery. *J. Rehabil. Res. Dev.* 43, 605–618. doi: 10.1682/JRRD.2005.06.0103

Holden, M. K. (2005). Virtual environments for motor rehabilitation: review. *CyberPsychol. Behav.* 8, 187–211. doi: 10.1089/cpb.2005.8.187

Hollerbach, J. M., Mills, R., Tristano, D., Christensen, R. R., Thompson, W. B., and Xu, Y. (2001). Torso force feedback realistically simulates slope on treadmill-style locomotion interfaces. *Int. J. Robot. Res.* 20, 939–952. doi: 10.1177/02783640122068209

Homma, K., and Usuba, M. (2007). "Development of ankle dorsiflexion/plantarflexion exercise device with passive mechanical joint," in *2007 IEEE 10th International Conference on Rehabilitation Robotics, ICORR'07* (Noordwijk), 292–297. doi: 10.1109/ICORR.2007.4428440

Hoölig, G., Koch, R., Wingendorf, I., Pohl, M., Werner, C., Hesse, S., et al. (2007). Repetitive locomotor training and physiotherapy improve walking and basic activities of daily living after stroke: a single-blind, randomised multicentre trial (DEutsche GAngtrainerStudie, DEGAS). *Clin. Rehabil.* 21, 17–27. doi: 10.1177/0269215506071281

Hornby, T. G., Zemon, D. H., and Campbell, D. D. (2005). Robotic-assisted, body-weight–supported treadmill training in individuals following motor incomplete spinal cord injury. *Phys. Ther.* 85, 52–66. doi: 10.1093/ptj/85.1.52

Horno, S. M.-D., Bermejo-bosch, I., Moreno, J. C., Pons, J. L., Farina, D., Iosa, M., et al. (2011). Rehabilitation of gait after stroke: a review towards a top-down approach. *J. Neuroeng. Rehabil.* 8:66. doi: 10.1186/1743-0003-8-66

Hwang, S., Kim, J., Yi, J., Tae, K., Ryu, K., and Kim, Y. (2006). "Development of an active ankle foot orthosis for the prevention of foot drop and toe drag," in *ICBPE 2006 - Proceedings of the 2006 International Conference on Biomedical and Pharmaceutical Engineering* (Singapore), 418–423.

Ison, M., and Artemiadis, P. (2014). The role of muscle synergies in myoelectric control: trends and challenges for simultaneous multifunction control. *J. Neural Eng.* 11:051001. doi: 10.1088/1741-2560/11/5/051001

Iwata, H., Yano, H., and Nakaizumi, F. (2002). "Gait master: a versatile locomotion interface for uneven virtual terrain," in *Proceedings IEEE Virtual Reality 2001* (Yokohama), 131–137.

Jezernik, S., Colombo, G., Keller, T., Frueh, H., and Morari, M. (2003). Robotic orthosis lokomat: a rehabilitation and research tool. *Neuromodulation* 6, 108–115. doi: 10.1046/j.1525-1403.2003.03017.x

Jin, X., Prado, A., and Agrawal, S. K. (2018). Retraining of human gait - are lightweight cable-driven leg exoskeleton designs effective? *IEEE Trans. Neural Syst. Rehabil. Eng.* 26, 847–855. doi: 10.1109/TNSRE.2018.2815656

Johnson, M. J. (2006). Recent trends in robot-assisted therapy environments to improve real-life functional performance after stroke. *J. NeuroEng. Rehabil.* 3:29. doi: 10.1186/1743-0003-3-29

Kautz, S. A., and Patten, C. (2005). Interlimb influences on paretic leg function in poststroke hemiparesis. *J. Neurophysiol.* 93, 2460–2473. doi: 10.1152/jn.00963.2004

Kawamoto, H., and Sankai, Y. (2007). "Power assist system HAL-3 for gait disorder person," in *International Conference on Computers for Handicapped Persons* (Linz), 196–203.

Kelley, C. P., Childress, J., Boake, C., and Noser, E. A. (2013). Over-ground and robotic-assisted locomotor training in adults with chronic stroke: a blinded randomized clinical trial. *Disabil. Rehabil. Assist. Technol.* 8, 161–168. doi: 10.3109/17483107.2012.714052

Kelso, J. A. S., Southard, D. L., and Goodman, D. (1979). On the nature of human interlimb coordination. *Science* 203, 1029–1031. doi: 10.1126/science.424729

Khan, M. I., Santamaria, V., and Agrawal, S. K. (2018). Improving trunk-pelvis stability using active force control at the trunk and passive resistance at the pelvis. *IEEE Robot. Autom. Lett.* 3, 2569–2576. doi: 10.1109/LRA.2018.2809919

Kim, S. Y., Yang, L., Park, I. J., Kim, E. J., Park, M. S., You, S. H., et al. (2015). Effects of innovative WALKBOT robotic-assisted locomotor training on balance and gait recovery in hemiparetic stroke: a prospective, randomized, experimenter blinded case control study with a four-week follow-up. *IEEE Trans. Neural Syst. Rehabil. Eng.* 23, 636–642. doi: 10.1109/TNSRE.2015.2404936

Klarner, T. (2010). *Contribution of load and length related manipulations to muscle responses during force perturbations* (Ph.D. thesis), The University of British Columbia, Vancouver, Canada.

Koopman, B., van Asseldonk, H., and van der Kooij, H. (2013). Selective control of gait subtasks in robotic gait training: foot clearance support in stroke survivors with a powered exoskeleton. *J. Neuroeng. Rehabil.* 10:3. doi: 10.1186/1743-0003-10-3

Krebs, H. I., Dipietro, L., Levy-Tzedek, S., Fasoli, S. E., Rykman-Berland, A., Zipse, J., et al. (2008). A paradigm shift for rehabilitation robotics. *IEEE Eng. Med. Biol. Mag.* 27, 61–70. doi: 10.1109/MEMB.2008.919498

Krishnan, C., Ranganathan, R., Kantak, S. S., Dhaher, Y. Y., and Rymer, W. Z. (2012). Active robotic training improves locomotor function in a stroke survivor. *J. NeuroEng. Rehabil.* 9:57. doi: 10.1186/1743-0003-9-57

Kuo, A. D. (2002). The relative roles of feedforward and feedback in the control of rhythmic movements. *Motor Control* 6, 129–145. doi: 10.1123/mcj.6.2.129

Kwakkel, G., Kollen, B. J., and Krebs, H. I. (2008). Effects of robot-assisted therapy on upper limb recovery after stroke: a systematic review. *Neurorehabil. Neural Repair* 22, 111–121. doi: 10.1177/1545968307305457

Lamontagne, A., Stephenson, J. L., and Fung, J. (2007). Physiological evaluation of gait disturbances post stroke. *Clin. Neurophysiol.* 118, 717–729. doi: 10.1016/j.clinph.2006.12.013

Losey, D. P., and O'Malley, M. K. (2019). Enabling robots to infer how end-users teach and learn through human-robot interaction. *IEEE Robot. Autom. Lett.* 4, 1956–1963. doi: 10.1109/LRA.2019.2898715

Lotze, M., Braun, C., Birbaumer, N., Anders, S., and Cohen, L. G. (2003). Motor learning elicited by voluntary drive. *Brain* 126, 866–872. doi: 10.1093/brain/awg079

Lunenburger, L., Colombo, G., Riener, R., and Dietz, V. (2004). Biofeedback in gait training with the robotic orthosis Lokomat. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 2004, 4888–4891. doi: 10.1109/IEMBS.2004.1404352

Marchal-Crespo, L., and Reinkensmeyer, D. J. (2009). Review of control strategies for robotic movement training after neurologic injury. *J. NeuroEng. Rehabil.* 6:20. doi: 10.1186/1743-0003-6-20

Marchal-Crespo, L., Tsangaridis, P., Obwegeser, D., Maggioni, S., and Riener, R. (2019). Haptic error modulation outperforms visual error amplification when learning a modified gait pattern. *Front. Neurosci.* 13:61. doi: 10.3389/fnins.2019.00061

Martelli, D., Xia, B., Prado, A., and Agrawal, S. K. (2019). Gait adaptations during overground walking and multidirectional oscillations of the visual field in a virtual reality headset. *Gait Posture* 67, 251–256. doi: 10.1016/j.gaitpost.2018.10.029

Matarić, M. J., Eriksson, J., Feil-Seifer, D. J., and Winstein, C. J. (2007). Socially assistive robotics for post-stroke rehabilitation. *J. NeuroEng. Rehabil.* 4:5. doi: 10.1186/1743-0003-4-5

Matjačić, Z., Zadravec, M., and Olenšek, A. (2018). Feasibility of robot-based perturbed-balance training during treadmill walking in a high-functioning chronic stroke subject: a case-control study. *J. NeuroEng. Rehabil.* 15, 1–15. doi: 10.1186/s12984-018-0373-z

Mayr, A., Kofler, M., Quirbach, E., Matzak, H., Fröhlich, K., and Saltuari, L. (2007). Prospective, blinded, randomized crossover study of gait rehabilitation in stroke patients using the lokomat gait orthosis. *Neurorehabil. Neural Repair* 21, 307–314. doi: 10.1177/1545968307300697

Mehrholz, J., and Pohl, M. (2012). Electromechanical-assisted gait training after stroke: a systematic review comparing end-effector and exoskeleton devices. *J. Rehabil. Med.* 44, 193–199. doi: 10.2340/16501977-0943

Mehrholz, J., Thomas, S., Werner, C., Kugler, J., Pohl, M., and Elsner, B. (2017). Electromechanical–assisted training for walking after stroke. *Cochrane Database Syst. Rev.* 5:CD006185. doi: 10.1002/14651858.CD006185.pub4

Milot, M. H., Marchal-Crespo, L., Beaulieu, L. D., Reinkensmeyer, D. J., and Cramer, S. C. (2018). Neural circuits activated by error amplification and haptic guidance training techniques during performance of a timing-based motor task by healthy individuals. *Exp. Brain Res.* 236, 3085–3099. doi: 10.1007/s00221-018-5365-5

Mirelman, A., Bonato, P., and Deutsch, J. E. (2009). Effects of training with a robot-virtual reality system compared with a robot alone on the gait of individuals after stroke. *Stroke* 40, 169–174. doi: 10.1161/STROKEAHA.108.516328

Monaco, V., Galardi, G., Jung, J. H., Bagnato, S., Boccagni, C., and Micera, S. (2009). "A new robotic platform for gait rehabilitation of bedridden stroke patients," in *2009 IEEE International Conference on Rehabilitation Robotics, ICORR 2009*, (Kyoto), 383–388. doi: 10.1109/ICORR.2009.5209548

Morone, G., Bragoni, M., Iosa, M., De Angelis, D., Venturiero, V., Coiro, P., et al. (2011). Who may benefit from robotic-assisted gait training? *Neurorehabil. Neural Repair* 25, 636–644. doi: 10.1177/1545968311401034

Morton, S. M., and Bastian, A. J. (2006). Cerebellar contributions to locomotor adaptations during splitbelt treadmill walking. *J. Neurosci.* 26, 9107–9116. doi: 10.1523/JNEUROSCI.2622-06.2006

Murray, S. A., Ha, K. H., and Goldfarb, M. (2014). "An assistive controller for a lower-limb exoskeleton for rehabilitation after stroke, and preliminary assessment thereof," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014* (Chicago, IL), 4083–4086. doi: 10.1109/EMBC.2014.6944521

Nalam, V., and Lee, H. (2019). Development of a two-axis robotic platform for the characterization of two-dimensional ankle mechanics. *IEEE/ASME Trans. Mechatron.* 24, 459–470. doi: 10.1109/TMECH.2019.2892472

Nielsen, J. B. (2003). How we walk: central control of muscle activity during human walking. *Neuroscientist* 9, 195–204. doi: 10.1177/1073858403009003012

Nilsson, A., Vreede, K., Häglund, V., Kawamoto, H., Sankai, Y., and Borg, J. (2014). Gait training early after stroke with a new exoskeleton – the hybrid assistive limb: a study of safety and feasibility. *J. NeuroEng. Rehabil.* 11:92. doi: 10.1186/1743-0003-11-92

Norton, J. (2010). Changing our thinking about walking. *J. Physiol.* 588:4341. doi: 10.1113/jphysiol.2010.200204

O'Dell, M. W., Lin, C.-C. D., and Harrison, V. (2009). Stroke rehabilitation: strategies to enhance motor recovery. *Annu. Rev. Med.* 60, 55–68. doi: 10.1146/annurev.med.60.042707.104248

Olenšek, A., Zadravec, M., Rudolf, M., Humar, M. G., Tomšič, I., Bizovičar, N., et al. (2018). "A novel approach to robot-supported training of symmetry, propulsion and balance during walking after stroke: a case study," in *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics* (Enschede), 408–413.

Ortiz, J., Rocon, E., Power, V., de Eyto, A., O'Sullivan, L., Wirz, M., et al. (2017). XoSoft - A vision for a soft modular lower limb exoskeleton. 16, 83–88. doi: 10.1007/978-3-319-46532-6_14

Patton, J. L., Stoykov, M. E., Kovic, M., and Mussa-Ivaldi, F. A. (2006). Evaluation of robotic training forces that either enhance or reduce error in chronic hemiparetic stroke survivors. *Exp. Brain Res.* 168, 368–383. doi: 10.1007/s00221-005-0097-8

Peckham, P. H., and Knutson, J. S. (2005). Functional electrical stimulation for neuromuscular applications. *Annu. Rev. Biomed. Eng.* 7, 327–360. doi: 10.1146/annurev.bioeng.6.040803.140103

Pennycott, A., Wyss, D., Vallery, H., Klamroth-Marganska, V., and Riener, R. (2012). Towards more effective robotic gait training for stroke rehabilitation: a review. *J. Neuroeng. Rehabil.* 9:65. doi: 10.1186/1743-0003-9-65

Peshkin, M., Brown, D. A., Santos-Munné, J. J., Makhlin, A., Lewis, E., Colgate, J. E., et al. (2005). "KineAssist: a robotic overground gait and balance training device," in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005* (Chicago, IL), 241–246.

Petersen, N., Christensen, L. O. D., and Nielsen, J. (1998). The effect of transcranial magnetic stimulation on the soleus H reflex during human walking. *J. Physiol.* 513, 599–610. doi: 10.1111/j.1469-7793.1998.599bb.x

Petersen, T. H., Willerslev-Olsen, M., Conway, B. A., and Nielsen, J. B. (2012). The motor cortex drives the muscles during walking in human subjects. *J. Physiol.* 590, 2443–2452. doi: 10.1113/jphysiol.2012.227397

Peurala, S. H., Airaksinen, O., Huuskonen, P., Jäkälä, P., Juhakoski, M., Sandell, K., et al. (2009). Effects of intensive therapy using gait trainer or floor walking exercises early after stroke. *J. Rehabil. Med.* 41, 166–173. doi: 10.2340/16501977-0304

Poli, P., Morone, G., Rosati, G., and Masiero, S. (2013). Robotic technologies and rehabilitation: new tools for stroke patients' therapy. *BioMed Res. Int.* 2013:153872. doi: 10.1155/2013/153872

Poppele, R. E., Rankin, A., and Eian, J. (2003). Dorsal spinocerebellar tract neurons respond to contralateral limb stepping. *Exp. Brain Res.* 149, 361–370. doi: 10.1007/s00221-003-1378-8

Poskanzer, E. H. (1972). Movement therapy in hemiplegia: a neurophysiologic approach. *J. Gerontol.* 27:290. doi: 10.1093/geronj/27.2.290

Presacco, A., Forrester, L. W., and Contreras-Vidal, J. L. (2012). Decoding intra-limb and inter-limb kinematics during treadmill walking from scalp electroencephalographic (EEG) signals. *IEEE Trans. Neural Syst. Rehabil. Eng.* 20, 212–219. doi: 10.1109/TNSRE.2012.2188304

Quintero, H. A., Farris, R. J., Durfee, W. K., and Goldfarb, M. (2010). "Feasibility of a hybrid-FES system for gait restoration in paraplegics," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10* (Buenos Aires), 483–486.

Quintero, H. A., Farris, R. J., Ha, K., and Goldfarb, M. (2012). "Preliminary assessment of the efficacy of supplementing knee extension capability in a lower limb exoskeleton with FES," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* (San Diego, CA), 3360–3363. doi: 10.1109/EMBC.2012.6346685

Ray, N. T., Reisman, D. S., and Higginson, J. S. (2020). Walking speed changes in response to user-driven treadmill control after stroke. *J. Biomech.* 101:109643. doi: 10.1016/j.jbiomech.2020.109643

Reinkensmeyer, D., Wynne, J., and Harkema, S. (2003). "A robotic tool for studying locomotor adaptation and rehabilitation," in *Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society* (Houston, TX), 2353–2354.

Reinkensmeyer, D. J., Aoyagi, D., Emken, J. L., Galvez, J. A., Ichinose, W., Kerdanyan, G., et al. (2014). Tools for understanding and optimizing robotic gait training. *J. Rehabil. Res. Dev.* 43, 657–670. doi: 10.1682/JRRD.2005.04.0073

Reinkensmeyer, D. J., Emken, J. L., and Cramer, S. C. (2004). Robotics, motor learning, and neurologic recovery. *Annu. Rev. Biomed. Eng.* 6, 497–525. doi: 10.1146/annurev.bioeng.6.040803.140223

Reisman, D. S., Block, H. J., and Bastian, A. J. (2005). Interlimb coordination during locomotion: what can be adapted and stored? *J. Neurophysiol.* 94, 2403–2415. doi: 10.1152/jn.00089.2005

Reisman, D. S., McLean, H., and Bastian, A. J. (2012). Split-belt treadmill training post-stroke: a case study background and purpose. *J. Neurol. Phys. Ther.* 34, 202–207. doi: 10.1097/NPT.0b013e3181fd5eab

Roby-Brami, A., and Jarrassé, N. (2018). *Wearable Robotic Systems and Their Applications for Neurorehabilitation*. Elsevier Ltd.

Rossignol, S., Dubuc, R., and Gossard, J.-P. (2006). Dynamic sensorimotor interactions in locomotion. *Physiol. Rev.* 86, 89–154. doi: 10.1152/physrev.00028.2005

Roy, A., Krebs, H. I., Patterson, S. L., Judkins, T. N., Khanna, I., Forrester, L. W., et al. (2007). "Measurement of human ankle stiffness using the anklebot," in *2007 IEEE 10th International Conference on Rehabilitation Robotics, ICORR'07* (Noordwijk), 356–363.

Saglia, J. A., Tsagarakis, N. G., Dai, J. S., and Caldwell, D. G. (2009). A high-performance redundantly actuated parallel mechanism for ankle rehabilitation. *Int. J. Robot. Res.* 28, 1216–1227. doi: 10.1177/0278364909104221

Sale, P., Franceschini, M., Waldner, A., and Hesse, S. (2012). Use of the robot assisted gait therapy in rehabilitation of patients with stroke and spinal cord injury. *Eur. J Phys. Rehabil. Med.* 48:111–21.

Schmidt, H., Hesse, S., Bernhardt, R., and Krüger, J. (2005a). HapticWalker—a novel haptic foot device. *ACM Trans. Appl. Percept.* 2, 166–180. doi: 10.1145/1060581.1060589

Schmidt, H., Hesse, S., Werner, C., and Bardeleben, A. (2005b). "Upper and lower extremity robotic devices to promote motor recovery after stroke - recent developments," in *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (San Francisco, CA), 4825–4828.

Schmidt, H., and Werner, C. (2007). *Gait Rehabilitation Machines Based on Programmable Footplates Foot Plate*. Available online at: jneuroengrehab.biomedcentral.com

Schmidt, H., Werner, C., Bernhardt, R., Hesse, S., and Krüger, J. (2007). Gait rehabilitation machines based on programmable footplates. *J. NeuroEng. Rehabil.* 4:2. doi: 10.1186/1743-0003-4-2

Schmitt, C., Métrailler, P., Al-Khodairy, A., Brodard, R., Fournier, J., Bouri, M., et al. (2004). "The motion maker™: a rehabilitation system combining an orthosis with closed-loop electrical muscle stimulation": in *Proceedings of the 8th Vienna International Workshops on Functional Electrical Stimulation* (Vienna), 117–120.

Schwartz, I., Sajin, A., Fisher, I., Neeb, M., Shochina, M., Katz-Leurer, M., et al. (2009). The effectiveness of locomotor therapy using robotic-assisted gait training in subacute stroke patients: a randomized controlled trial. *PM&R* 1, 516–523. doi: 10.1016/j.pmrj.2009.03.009

Seiterle, S., Susko, T., Artemiadis, P. K., Riener, R., and Igo Krebs, H. (2015). Interlimb coordination in body-weight supported locomotion: a pilot study. *J. Biomech.* 48, 2837–2843. doi: 10.1016/j.jbiomech.2015.04.042

Seo, K.-H., and Lee, J.-J. (2009). The development of two mobile gait rehabilitation systems. *IEEE Trans. Neural Syst. Rehabil. Eng.* 17, 156–166. doi: 10.1109/TNSRE.2009.2015179

Shemmell, J., An, J. H., and Perreault, E. J. (2009). The differential role of motor cortex in stretch reflex modulation induced by changes in environmental mechanics and verbal instruction. *J. Neurosci.* 29, 13255–13263. doi: 10.1523/JNEUROSCI.0892-09.2009

Skidmore, J., and Artemiadis, P. (2015). "Leg muscle activation evoked by floor stiffness perturbations: a novel approach to robot-assisted gait rehabilitation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA: IEEE), 6463–6468.

Skidmore, J., and Artemiadis, P. (2016a). On the effect of walking surface stiffness on inter-limb coordination in human walking: toward bilaterally informed robotic gait rehabilitation. *J. NeuroEng. Rehabil.* 13:32. doi: 10.1186/s12984-016-0140-y

Skidmore, J., and Artemiadis, P. (2016b). "Sudden changes in walking surface compliance evoke contralateral EMG in a hemiparetic walker: a case study of inter-leg coordination after neurological injury," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Orlando, FL: IEEE), 4682–4685.

Skidmore, J., and Artemiadis, P. (2016c). Unilateral floor stiffness perturbations systematically evoke contralateral leg muscle responses: a new approach to robot-assisted gait therapy. *IEEE Trans. Neural Syst. Rehabil. Eng.* 24, 467–474. doi: 10.1109/TNSRE.2015.2421822

Skidmore, J., and Artemiadis, P. (2016d). "Unilateral walking surface stiffness perturbations evoke brain responses: toward bilaterally informed robot-assisted gait rehabilitation," in *2016 IEEE International Conference on*

Robotics and Automation (ICRA), Vol. 2016 (Stockholm: IEEE), 3698–3703. doi: 10.1109/ICRA.2016.7487555

Skidmore, J., and Artemiadis, P. (2017). Unilateral changes in walking surface compliance evoke dorsiflexion in paretic leg of impaired walkers. *J. Rehabil. Assist. Technol. Eng.* 4:2055668317738469. doi: 10.1177/2055668317738469

Skidmore, J., Barkan, A., and Artemiadis, P. (2014). "Investigation of contralateral leg response to unilateral stiffness perturbations using a novel device," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Chicago, IL: IEEE), 2081–2086.

Skidmore, J., Barkan, A., and Artemiadis, P. (2015). Variable stiffness treadmill (VST): system development, characterization, and preliminary experiments. *IEEE/ASME Trans. Mechatron.* 20, 1717–1724. doi: 10.1109/TMECH.2014.2350456

Sparrow, W. A., Donovan, E., Van Emmerik, R., and Barry, E. B. (1987). Using relative motion plots to measure changes in intra-limb and inter-limb coordination. *J. Motor Behav.* 19, 115–129. doi: 10.1080/00222895.1987.10735403

Srivastava, S., Kao, P. C., Kim, S. H., Stegall, P., Zanotto, D., Higginson, J. S., et al. (2015). Assist-as-needed robot-aided gait training improves walking function in individuals following stroke. *IEEE Trans. Neural Syst. Rehabil. Eng.* 23, 956–963. doi: 10.1109/TNSRE.2014.2360822

Stauffer, Y. (2009). "Overground gait rehabilitation: first clinical investigation with the walkTrainer, Yves Allemand, Fondation Suisse pour les Cyberthèses, Switzerland Yves Stauffer, EPFL, Switzerland," in *European Conference on Technically Assisted Rehabilitation (TAR)* (Berlin), 1–4.

Stegall, P., Zanotto, D., and Agrawal, S. K. (2017). Variable damping force tunnel for gait training using ALEX III. *IEEE Robot. Autom. Lett.* 2, 1495–1501. doi: 10.1109/LRA.2017.2671374

Swinnen, E., Beckwée, D., Meeusen, R., Baeyens, J.-P., and Kerckhofs, E. (2014). Does robot-assisted gait rehabilitation improve balance in stroke patients? A systematic review. *Top. Stroke Rehabil.* 21, 87–100. doi: 10.1310/tsr2102-87

Swinnen, S. P., Massion, J., Heuer, H., Casaer, P., and Heuer, H. (2013). *Interlimb Coordination: Neural, Dynamical, and Cognitive Constraints*. Cambridge, MA: Academic Press.

Tariq, M., Trivailo, P. M., and Simic, M. (2018). EEG-based BCI control schemes for lower-limb assistive-robots. *Front. Hum. Neurosci.* 12:312. doi: 10.3389/fnhum.2018.00312

Tong, R. K., Ng, M. F., and Li, L. S. (2006). Effectiveness of gait training using an electromechanical gait trainer, with and without functional electric stimulation, in subacute stroke: a randomized controlled trial. *Arch. Phys. Med. Rehabil.* 87, 1298–1304. doi: 10.1016/j.apmr.2006.06.016

Tsoi, Y. H., and Xie, S. Q. (2008). "Impedance control of ankle rehabilitation robot," in *2008 IEEE International Conference on Robotics and Biomimetics, ROBIO 2008* (Bangkok), 840–845.

Unluhisarcikli, O., Pietrusinski, M., Weinberg, B., Bonato, P., and Mavroidis, C. (2011). "Design and control of a robotic lower extremity exoskeleton for gait rehabilitation," in *IEEE International Conference on Intelligent Robots and Systems* (San Francisco, CA), 4893–4898.

Vallery, H., Veneman, J., Van Asseldonk, E., Ekkelenkamp, R., Buss, M., and Van der Kooij, H. (2008). Compliant actuation of rehabilitation robots - Benefits and limitations of series elastic actuators. *IEEE Robot. Autom. Mag.* 15, 60–69. doi: 10.1109/MRA.2008.927689

Veneman, J. F., Kruidhof, R., Hekman, E. E. G., Ekkelenkamp, R., Van Asseldonk, E. H. F., and Van Der Kooij, H. (2007). Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation. *IEEE Trans. Neural Syst. Rehabil. Eng.* 15, 379–386. doi: 10.1109/TNSRE.2007.903919

Venkatakrishnan, A., Francisco, G. E., and L. Contreras-Vidal, J. (2014). Applications of brain–machine interface systems in stroke recovery and rehabilitation. *Curr. Phys. Med. Rehabil. Rep.* 2, 93–105. doi: 10.1007/s40141-014-0051-4

Viteckova, S., Kutilek, P., and Jirina, M. (2013). Wearable lower limb robotics: a review. *Biocybern. Biomed. Eng.* 33, 96–105. doi: 10.1016/j.bbe.2013.03.005

Wade, E., and Winstein, C. J. (2011). Virtual reality and robotics for stroke rehabilitation: where do we go from here? *Top. Stroke Rehabil.* 18, 685–700. doi: 10.1310/tsr1806-685

Waldner, A., Picelli, A., Molteni, F., Mazzoleni, S., Gasperini, G., Munari, D., et al. (2013). Systematic review of outcome measures of walking training using electromechanical and robotic devices in patients with stroke. *J. Rehabil. Med.* 45, 987–996. doi: 10.2340/16501977-1234

Whitall, J. (1989). A developmental study of the interlimb coordination in running and galloping. *J. Motor Behav.* 21, 409–428. doi: 10.1080/00222895.1989.10735492

Winstein, C. J., Stein, J., Arena, R., Bates, B., Cherney, L. R., Cramer, S. C., et al. (2016). Guidelines for adult stroke rehabilitation and recovery: a guideline for healthcare professionals from the American Heart Association/American Stroke Association. *Stroke* 47, e98–e169. doi: 10.1161/STR.0000000000000098

Xiloyannis, M., Annese, E., Canesi, M., Kodiyan, A., Bicchi, A., Micera, S., et al. (2019). Design and validation of a modular one-to-many actuator for a soft wearable exosuit. *Front. Neurorobot.* 13:39. doi: 10.3389/fnbot.2019.00039

Yang, J. F., and Gorassini, M. (2006). Spinal and brain control of human walking: implications for retraining of walking. *Neuroscientist* 12, 379–389. doi: 10.1177/1073858406292151

Yano, H., Tamefusa, S., Tanaka, N., Saitou, H., and Iwata, H. (2010). "Gait rehabilitation system for stair climbing and descending," in *2010 IEEE Haptics Symposium, HAPTICS 2010* (Waltham, MA), 393–400.

Yoon, J., Novandy, B., Yoon, C., and Park, K. (2010). A 6-DOF gait rehabilitation robot with upper and lower limb connections that allows walking velocity updates on various terrains. *IEEE/ASME Trans. Mechatron.* 15, 201–215. doi: 10.1109/TMECH.2010.2040834

Zhang, M., Davies, T. C., and Xie, S. (2013). Effectiveness of robot-assisted therapy on ankle rehabilitation - A systematic review. *J. NeuroEng. Rehabil.* 10:30. doi: 10.1186/1743-0003-10-30

Zuur, A. T., Christensen, M. S., Sinkjær, T., Grey, M. J., and Nielsen, J. B. (2009). Tibialis anterior stretch reflex in early stance is suppressed by repetitive transcranial magnetic stimulation. *J. Physiol.* 587, 1669–1676. doi: 10.1113/jphysiol.2009.169367

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

REVIEW
published: 03 June 2020
doi: 10.3389/fnbot.2020.00025

Current Status, Challenges, and Possible Solutions of EEG-Based Brain-Computer Interface: A Comprehensive Review

Mamunur Rashid[1]*, *Norizam Sulaiman*[1], *Anwar P. P. Abdul Majeed*[2], *Rabiu Muazu Musa*[3], *Ahmad Fakhri Ab. Nasir*[2], *Bifta Sama Bari*[1] *and Sabira Khatun*[1]

[1] *Faculty of Electrical & Electronics Engineering Technology, Universiti Malaysia Pahang, Pekan, Malaysia,* [2] *Innovative Manufacturing, Mechatronics and Sports Laboratory, Faculty of Manufacturing and Mechatronic Engineering Technology, Universiti Malaysia Pahang, Pekan, Malaysia,* [3] *Centre for Fundamental and Continuing Education, Universiti Malaysia Terengganu, Kuala Nerus, Malaysia*

Brain-Computer Interface (BCI), in essence, aims at controlling different assistive devices through the utilization of brain waves. It is worth noting that the application of BCI is not limited to medical applications, and hence, the research in this field has gained due attention. Moreover, the significant number of related publications over the past two decades further indicates the consistent improvements and breakthroughs that have been made in this particular field. Nonetheless, it is also worth mentioning that with these improvements, new challenges are constantly discovered. This article provides a comprehensive review of the state-of-the-art of a complete BCI system. First, a brief overview of electroencephalogram (EEG)-based BCI systems is given. Secondly, a considerable number of popular BCI applications are reviewed in terms of electrophysiological control signals, feature extraction, classification algorithms, and performance evaluation metrics. Finally, the challenges to the recent BCI systems are discussed, and possible solutions to mitigate the issues are recommended.

Keywords: brain-computer interface (BCI), electroencephalogram (EEG), machine learning, classification, feature extraction

OPEN ACCESS

Edited by:
Alois C. Knoll,
Technical University of Munich, Germany

Reviewed by:
Jing Jin,
East China University of Science and Technology, China
Feng Duan,
Nankai University, China

***Correspondence:**
Mamunur Rashid
mamun110218@gmail.com

Received: *26 November 2019*
Accepted: *08 April 2020*
Published: *03 June 2020*

Citation:
Rashid M, Sulaiman N, P. P. Abdul Majeed A, Musa RM, Ab. Nasir AF, Bari BS and Khatun S (2020) Current Status, Challenges, and Possible Solutions of EEG-Based Brain-Computer Interface: A Comprehensive Review. Front. Neurorobot. 14:25. doi: 10.3389/fnbot.2020.00025

INTRODUCTION

Communication, or social interaction, is one of the key principles of human civilization. This quality enables one to share emotions, expectations, and creative thoughts amongst human beings. In the event that this communication is established through speech, gesture, or writing, human communication becomes easier and devoid of constraints. Nonetheless, people who are suffering from locked-in syndrome do not have the aforementioned options for interaction. Patients with locked-in syndrome could not interact or express themselves, although they are well-cognizant of things around them (Ashok, 2017). Amyotrophic lateral sclerosis (ALS), cerebral palsy, brain stem stroke, multiple sclerosis, cerebral palsy, and spinal cord injury are the main causes of locked-in syndrome (Holz et al., 2013). It is almost impossible for a person who is affected by the locked-in syndrome to communicate with other persons, and hence, Brain–Computer Interface (BCI) is a promising means to furnish them with basic communication abilities. Fundamentally, the human brain and devices are interfaced through the concept of BCI in which the users will have to generate a variety of brain waves

that will be recognized and converted into commands to the devices (Volosyak et al., 2017). In its earlier days, researchers intended to use this technology to develop assistive devices for medical purposes only. Nonetheless, the employment of this technology has expanded, and it has found its way into non-medical applications. It is discernible that over the last 15 years, a considerable number of original articles as well as reviews have been published on BCI. An excellent review article on BCI spelling systems was published in Rezeika et al. (2018), giving a concise description of some successful recent BCI spelling models, including their categories, methodologies, and results. The authors also listed some limitations of the current systems as well as making recommendations for directions that could be pursued to overcome the issues. However, it is worth mentioning that the content of the review emphasized and was restricted to only BCI spellers. There are other notable BCI reviews that cater to specific applications, for instance, wheelchair control (Fernández-Rodríguez et al., 2016; Al-qaysi et al., 2018), BCI mobile robot (Bi et al., 2013), emotion recognition using EEG (Al-Nafjan et al., 2017), biometrics (Del Pozo-Banos et al., 2014; Alariki et al., 2018), and virtual reality and gaming (Kaplan et al., 2013; Ahn et al., 2014; Cattan et al., 2018). Nevertheless, some pertinent information was missing or not duly reported, for instance, descriptions of methodology and evaluation metrics employed, and/or future directions of the research.

Electroencephalogram (EEG) control signals and their classifications have been briefly discussed in an excellent review (Ramadan and Vasilakos, 2017). The authors reviewed state-of-the-art BCI solutions with regards to both hardware and software; however, it was noticeable that the applications, as well as the signal processing methods, were not taken into consideration. Likewise, Hwang et al. (2013) have summarized articles related to EEG-based BCI systems published from 2007 to 2011. Notwithstanding, the review did not entirely reflect the current state-of-the-art and did not provide any future directions for the research. Conversely, in Abdulkader et al. (2015), the fundamental aspects that cover the wide spectrum of EEG-based BCI systems were reviewed; however, the number of articles reviewed was rather limited. Lotte et al. (2007) provided a review of the classification algorithms used in EEG-based BCI systems up to 2006. In their second review (Lotte et al., 2018), the application of machine learning algorithms used on BCI systems from 2007 to 2017 was reviewed. In both articles, the authors surveyed EEG control signals, features, classification methods, and classification accuracy. Moreover, the authors provided some guidelines on selecting the appropriate classification algorithms; nonetheless, the articles lacked evaluation of the performance metrics. A review on portable and non-invasive modalities such as EEG-, functional transcranial Doppler (fTCD)-, and near-infrared spectroscopy (NIRS)-based hybrid BCI was reported in Banville and Falk (2016). Twenty-two items were investigated from 55 journal articles published between 2008 and 2014. The authors reviewed non-invasive modalities, EEG control signal, experiment protocol, signal processing methods, and system evaluation, as well as shedding some light on future directions for EEG-based BCI research. However, a comparison of the experimental results between the BCI applications or EEG modalities were not made available, and a similar observation was also noticed in Abiri et al. (2019).

Therefore, the objectives of this article are to review EEG-based BCI systems with regards to the different brain control signals, feature extraction methods, classification algorithms, and evaluation metrics utilized. Moreover, a concise overview of EEG-based BCI systems is presented here so that the reader(s) may select the most appropriate method for a specific BCI system. In addition, related research gaps that warrant further exploration are also presented in this paper. Of note, salient problems associated with EEG-based BCI systems are listed in terms of its applications, and possible solutions to mitigate the issues are also recommended. Moreover, this review, unlike other published review articles with regards to EEG-based BCI systems that were specific in nature, particularly with respect to either its specific applications or part of the methodology employed (e.g., feature extraction, signal processing, and classification, amongst others), provides a more comprehensive overview that can easily be comprehended by the readers to identify the gaps in the body of knowledge. This article is structured in the following manner: section Essential Components of BCI Technology presents a brief discussion on BCI overview, section Popular EEG Based BCI Applications Aspect illustrates the review of popular EEG-based BCI applications, section Current Challenges and Directions discusses the challenges, giving recommendations, section Conclusion draws the conclusion of the present review paper.

ESSENTIAL COMPONENTS OF BCI TECHNOLOGY

Brain–Computer Interface (BCI) is an effective as well as a powerful tool for user-system communication. Through this system, from the issuance of the commands to the completion of the interaction, no external devices or muscle intervention is required (Van Erp et al., 2012). Nicolas-Alonso and Gomez-Gil (2012) defined brain–computer interface (BCI) or brain–machine interface (BMI) as a hardware and software communications strategy that empowers humans to interact with their surroundings with no inclusion of peripheral nerves or muscles by utilizing control signals produced from electroencephalographic activity. Every BCI system essentially consists of five components: brain activity measurement, preprocessing, feature extraction, classification, and translation into a command (Mason and Birch, 2003). **Figure 1** depicts a typical block diagram that illustrates the different stages of EEG signal processing for BCI. In the brain activity acquisition phase, the brain activity from the targeted user is captured through the various types of EEG sensors (Wolpaw et al., 2006). The raw EEG data includes a variety of artifacts, and these artifacts are eliminated in the pre-processing phase (Bashashati et al., 2007a). Feature extraction aims at describing the signals by a few relevant values called "features;" often, at this stage, the selection of significant features is also investigated (Bashashati et al., 2007a). The extracted features are then classified through different machine learning and deep learning algorithms in the

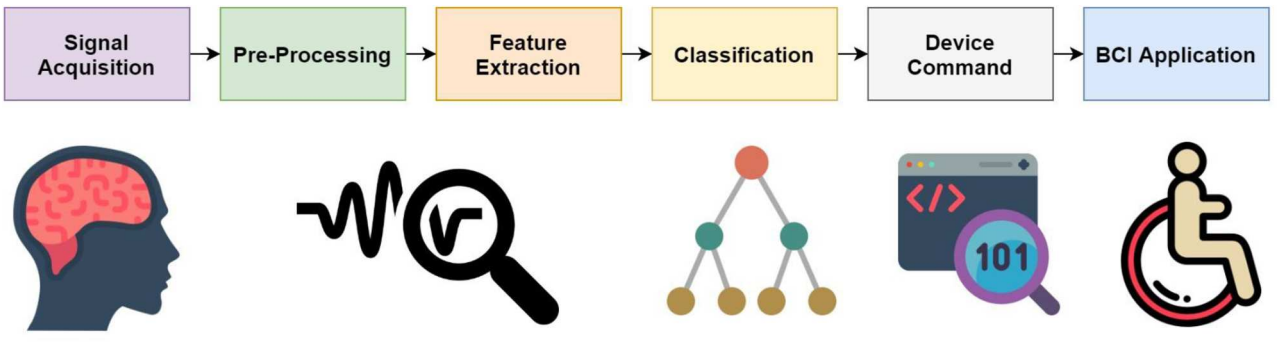


FIGURE 1 | General architecture of a brain-computer interface.

classification phase (Lotte et al., 2007). Finally, the classified outcomes are translated into device commands to develop real-life BCI application (Kubler et al., 2006).

Branches of BCI Technology

Generally, BCI frameworks may be separated into a number of classes. **Figure 2** illustrates the three categorization schemes, namely by dependability, recording technique, and method of operation (Lotte et al., 2015). Regarding dependability, BCI can be classed as either dependent or independent BCI. Dependent BCIs require some form of motor control by the user or healthy subjects, for instance, gaze control (Lalor et al., 2005). MI-based BCIs are an ideal example of dependent BCI systems and have been extensively utilized. Conversely, independent BCIs do not require any form of motor control by the user; this type of BCI is ideal for stroke patients or severely impaired patients. In Tello et al. (2016), an SSVEP-based independent BCI system was proposed to identify two different targets, and it was demonstrated to be successful.

With regard to recording method, BCI can be categorized into invasive and non-invasive. Microelectrode arrays are often required to be implanted inside the skull for invasive BCIs. Two common invasive modalities that have been reported in BCI research are intracortical recording and electrocorticography (ECoG). Conversely, if the brain signals are acquired by means of sensors placed on the scalp, it is known as non-invasive BCI. Amongst the non-invasive modalities often utilized are EEG, MEG, PET, fMRI, and fNIRS. In BCIs, EEG is the most widely employed non-invasive modality, where a variety of control signals, including SCP, SSVEP, MI, ErrP, and P300, can be evoked.

Finally, BCI can have either a synchronous or asynchronous mode of operation. The interaction between the user and the system may be either time-dependent or time-independent. In the event that the interaction is carried out within a certain period of time upon a cue imposed by the system, then the system is known as synchronous BCI. In contrast, in asynchronous BCI, the subject can generate a mental task at any period of time to interact with the application. In comparison with asynchronous BCI, synchronous BCIs are not user-friendly, but designing such a system is much easier than for asynchronous BCI (Bashashati et al., 2007b).

Brain Activity Measurement Modalities

To avoid the risk of surgery, most BCI researchers prefer the non-invasive approach. EEG, MEG, PET, fMRI, and fNIRS are among the non-invasive modalities that are often utilized. The selection of the measurement method depends on a variety of parameters, for instance, spatial resolution, temporal resolution, invasiveness, measured activity, cost, and portability, amongst others. Owing to its desirable traits, namely high temporal resolution, low cost, ease of portability, and non-invasiveness, EEG is the most commonly employed neuroimaging modality among BCI researchers.

EEG records voltage fluctuations due to the flow of ionic current during synaptic excitations in the neurons of the brain (Baillet et al., 2001). In this modality, electrodes are attached to the scalp to obtain brain signals. Its non-invasive and inexpensive characteristics have made EEG the most popular modality among the BCI research community. The electrode number varies from 1 to 256 for different EEG headsets. The measured EEG signal is the voltage difference between the active and reference electrode over time, with its amplitude in micro-volts (μV). Generally, the EEG amplitude ranges from -100 to $+100$ microvolts. The EEG signals can be categorized according to frequency bands, and each of these bands has specific biological significance. The EEG frequency bands with relevant characteristics are listed in **Table 1** (Wang et al., 2016).

EEG Control Signals Used in BCI Applications

Some neurophysiological EEG signals have been decoded to enable the BCI to understand the subject's intentions, and these signals are known as EEG control signals. BCI aims to identify the specific neurophysiological signals of a given subject in order to associate a command to each of these signals. Some of these control signals are relatively easy to identify, as well as being relatively easy to control by the user. The extensively utilized EEG control signals include SCP, P300, MI, MRCP, ErrP, SSVEP, SSAEP, and SSSEP.

The movement-related cortical potential (MRCP) is a low-frequency negative shift in the EEG recording that takes place $\sim$2 s prior to the production of voluntary movement. MRCP replicates the cortical processes employed in the planning and

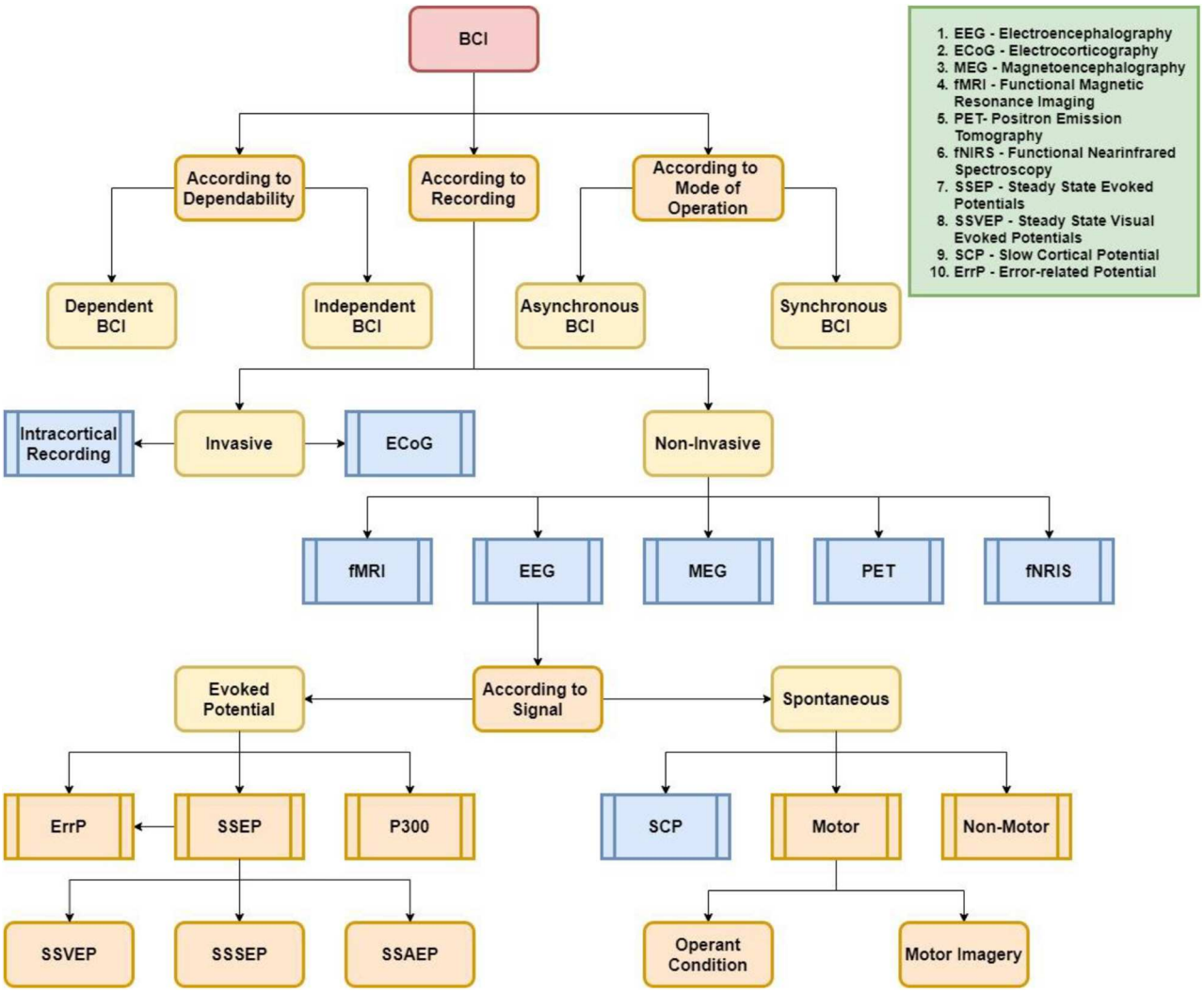


FIGURE 2 | Classification of BCI systems in terms of dependability, recording method, and mode of operation.

preparation of the movement (Shakeel et al., 2015). It is mainly beneficial for those BCI applications where the delay between the intention to act and the feedback from the system is crucial to induce plasticity.

The error-related potential (ErrP) has recently been utilized as an ERP component that can be used to correct BCI errors. The ErrP occurs when there is a mismatch between a subject's intention to perform a given task and the response provided by the BCI (Abiri et al., 2019). For instance, if a user wishes to move a cursor from the middle of a monitor to the left side of the monitor but the cursor erroneously moves to the right, an error-related potential will be generated. The ErrP is most pronounced at the frontal and central lobes. The delay and non-stationarity characteristics of this signal are still a challenge for real-time BCI implementation (Abiri et al., 2019).

Spontaneous signals are generated voluntarily by the user, without external stimulation, following an internal cognitive process. The most typical spontaneous signals used are undoubtedly sensorimotor rhythms. However, other neurophysiological signals have been used, such as slow cortical potentials or non-motor cognitive signals.

Slow Cortical Potentials (SCP) are very slow variations in cortical activity that can last from hundreds of milliseconds (ms) to several seconds (s) (Kleber and Birbaumer, 2005). It is possible to make these variations positive or negative via operant conditioning. As the control of SCP is achieved by operant conditioning, mastering such a signal generally requires a very long training time. This training by operant conditioning is even longer for SCP than for motor rhythms (Birbaumer, 2006). However, it seems that SCP would be a more stable signal.

Non-motor cognitive processing tasks are also extensively used to operate a BCI. These tasks are, for instance, mental mathematical computations, mental rotation of geometric figures, visual counting, mental generation of words, and music

TABLE 1 | EEG frequency bands with properties.

Band	Frequency (Hz)	Amplitude (μV)	Location	Activity
Delta	0.5–4 Hz	100–200	Frontal	Deep sleep
Theta	4–8 Hz	5–10	Various	Drowsiness, light sleep
Alpha	8–13 Hz	20–80	Posterior region of head	Relaxed
Beta	13–30 Hz	1–5	Left and right side, symmetrical distribution, most evident frontally	Active thinking, alert
Gamma	>30 Hz	0.5–2	Somatosensory cortex	Hyperactivity

imagination, amongst others (Chiappa and Bengio, 2004). All of these mental tasks generate specific EEG signal variations in specific cortical regions and frequency bands, which make them relatively easy to identify.

Steady-State Evoked Potentials (SSEP)

SSEP appears when the user perceives a periodic stimulus like a flickering photo or an amplitude-modulated sound. An important characteristic of SSEP is that the stimulation frequency or harmonics is equivalent to the EEG signal frequencies (Gouy-Pailler et al., 2007). The stimulation of a fixed frequency evokes SSEP by yielding EEG activity of the identical frequency as the stimulation is generated (Maye et al., 2011). According to visual, auditory, and somatosensory stimulation, SSEP can be further divided into Steady-State Visually Evoked Potentials (SSVEP) (Valbuena et al., 2010), Steady-State Auditory Evoked Potentials (SSAEP) (Fairclough and Gilleade, 2014), and Steady-State Somatosensory Evoked Potentials (SSSEP) (Muller-Putz et al., 2006).

Every SSVEP-based BCI needs a specific number of visual stimuli that indicate specific BCI output commands. These stimuli flicker continuously, with distinguishable frequency bands ranging from 6 to 30 Hz. In the event that a subject concentrates on a particular flickering stimulus, an SSVEP with an identical frequency to that of the target flicker is generated. For example, if the frequency of the targeted stimulus is 15 Hz, the frequency of the generated SSVEP will also be 15 Hz. Therefore, the user pays attention visually to a target, and the BCI determines the target through analyzing the SSVEP features.

SSAEP are commonly extracted by trains of click stimuli, tone pulses, or amplitude-modulated tones, with a repetition or modulation rate between 20 and 100 Hz. The resulting brain response can be localized at the primary auditory cortex (Hill and Schölkopf, 2012). Although the SSAEP-based BCI system yielded promising results, only highly experienced users could maintain the high level of attention needed in order to obtain high accuracy (Punsawad and Wongsawat, 2017).

In the SSSEP paradigm, vibrotactile sensors are placed at pre-determined parts of the body, and these sensors generate stimulations at different frequencies (Hamada et al., 2014). The stimulations of these sensors will then be reflected in EEG signals recorded from the scalp. In comparison to visual- or sensorimotor rhythm-based BCI research, limited studies of SSSEP have been published. This is primarily due to the lack of a well-designed standard tactile stimulator with which to extract the SSSEP signals.

P300

BCI systems with P300 rely upon stimuli that flash in succession. These stimuli may be symbols or letters and are used for different BCI applications, for instance, controlling a robot arm, cursor, or mobile robot. P300 is generated in the Pz areas of the brain, ∼300 ms after the stimulus is presented (Farwell and Donchin, 1988). It has been reported in the literature that the response's peak amplitude is much larger, even with less probable stimulus (John et al., 1996). Amongst the advantageous features of P300-based BCIs is that they do not require any form of training. However, it is worth mentioning that in the event that infrequent stimulus decreases the amplitude of P300 that, in turn, reduces the overall performance of the system.

Motor Imagery

Moving a limb or even contracting a single muscle changes brain activity in the cortex. Preparation for the movement or imagining movement [also known as motor imagery (MI)] generates oscillations in the brain motor areas known as sensorimotor rhythms (SMR). Increase and decrease of oscillatory activity in a particular frequency band are referred to as event-related synchronization (ERS) and event-related desynchronization (ERD), respectively. The most influential frequency bands for motor imagery are the alpha and beta brain waves. Activity invoked by the left and right hand MI is generated from the C3 and C4 areas of the brain, respectively, whereas the foot movement imagery is originated from Cz. Left and right foot movements are almost impossible to distinguish in EEG due to the fact that the corresponding cortical regions are extremely close. The cortical areas must be large enough to produce detectable patterns in the background EEG. The cortical areas of the left hand, right hand, tongue, and foot, are large and distinguishable. Thus, the movement of those body limbs via imagination can be controlled by BCI applications (Schlögl et al., 2005).

EEG Acquisition Framework for BCI Application

The human brain consists of two main parts, i.e., the cerebral cortex and subcortical regions. The essential and vital

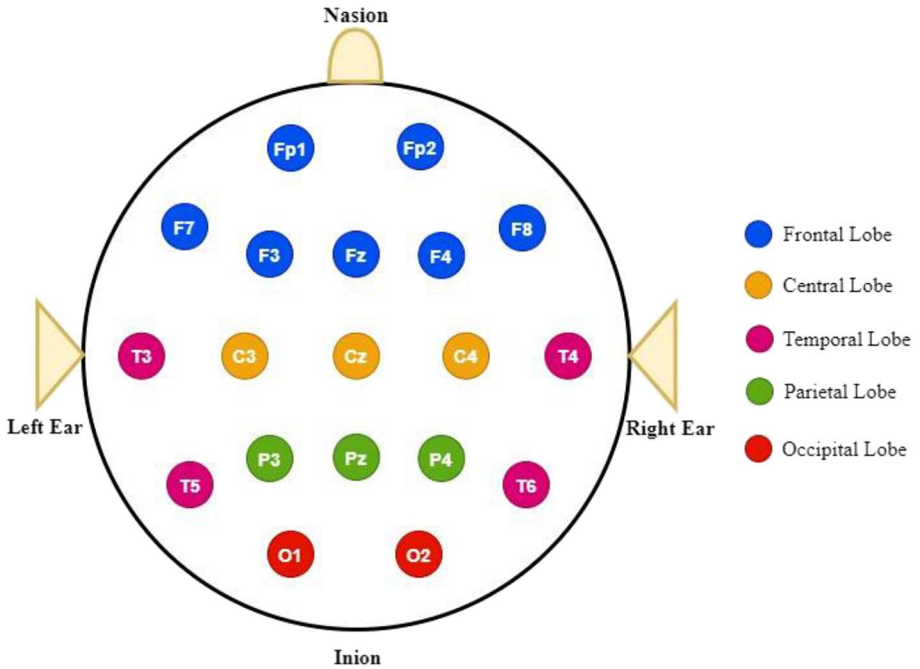


FIGURE 3 | Standardized electrode placement scheme.

functions, including body temperature, respiration, heart rate, and emotional responses, including reflexes, fear, learning, and memory, are controlled by the subcortical regions. Conversely, the cerebral cortex, commonly known as the cerebrum, regulates sensory and motor processing as well as higher-level functions, for example, language processing, pattern recognition, reasoning, and planning. The cerebral cortex is partitioned into two hemispheres, in which every hemisphere is classified into four lobes, namely the parietal, occipital, frontal, and temporal lobes. The parietal lobe is in charge of numerous functions, for instance, spelling, objects, manipulation, perception, and spatial awareness. Conversely, the language, memory, recognizing faces, and generating emotions are the main functions of the temporal lobe. The third lobe, i.e., the frontal lobe, involves organizing, social skills, planning, flexible thinking, problem-solving, conscious movement, attention, and emotional and behavioral control. The occipital lobe is related to interpreting visual stimuli. Additionally, another essential system of the human body is the nervous system, which is classified into two main parts: the central and peripheral systems. The spinal cord and the brain are the two parts of the central nervous system. In contrast, the peripheral nervous system includes the autonomic nervous system, which controls functions such as digestion, secretion of hormones, breathing, and heart rate.

The 10/20 system is a universally recognized method that indicates the locations of electrodes on the scalp. The system depends on the connection between the electrode location and the underlying area of the cerebral cortex. The numbers 10 and 20 indicate that the distances among adjacent electrodes are either 10 or 20% of the total front-back or right-left distance of the skull. In each site, a letter is used to denote the lobe, whereas the hemisphere location is represented by a number. In the 10/20 system, the frontal, parietal, temporal, and occipital lobes can be denoted by the letters F, P, T, and O, respectively, as depicted in **Figure 3**. The central lobe is not included; the letter C is utilized only for identification purposes. Z (zero) implies that an electrode is placed on the midline. Even numbers (2, 4, 6, 8) are utilized to indicate the right hemispheres electrode positions, whereas left-hemisphere electrode positions are denoted by odd numbers (1, 3, 5, 7) (Rojas et al., 2018).

Hardware Technology for EEG Signal Acquisition

There are two main methods of acquiring EEG signals: wireless or wired. Typically, EEG signal measurements are performed using a number of electrodes varying from 1 to about 256. These electrodes are generally attached using an elastic cap. The contact between the electrodes and the skin is commonly enhanced through the utilization of a conductive gel or paste. However, this makes the electrode embedding procedure a generally tedious and lengthy operation. Nonetheless, it is worth noting that the use of dry electrodes, which do not require conductive gels or pastes, has been proposed and validated (Popescu et al., 2007). In spite of this success, it is worth pointing out that the performance of this method in terms of maximum information rate is, on average, 30% lower than that obtained with a BCI based on electrodes that employ conductive gels or pastes. Though the wired system is well-established, it has some notable limitations. It is evident that connection using wires between the electrodes

TABLE 2 | Summary table of recent EEG devices.

Device name	No. of channels	Sampling frequency	Communication	No. of publications
NeuroScan	SynAmps:64 Grael:32 NuAmps:40 Siesta:32	SynAmps:20 kHz Grael:4,096 Hz NuAmps:1,000 Hz Siesta:1,024 Hz	Wired	12,300
Brain Products	LiveAmp: 8/16/32	Between 250, 500, and 1,000 Hz	Wireless	6,690
BioSemi	16, 32, or 64	2/4/8/16 KHz	Wired	5,750
Emotiv	INSIGHT: 5 EPOC+: 14 EPOC FLEX: 32	128 Hz	Wireless	3,990
NeuroSky	1	512 Hz	Wireless	2,290
Advanced brain monitoring	ABM B-Alert X24: 24	256 Hz	Wireless	790
g.tec nautilus	64	500 Hz	Wireless	430
AntNeuro eego	64	2,048 Hz	Wireless	340
Neuroelectrics Enobio 32	32	500 Hz	Wireless	317
Muse	4	256 Hz	Wireless	207
OpenBCI	Up to 16 channels	256 Hz	Wireless	201
Cognionics Mobile	72	500–1,000 Hz	Wireless	128
mBrainTrain	24	250–500 Hz	Wireless	38
MyndBand EEG headset	3	512 Hz	Wireless	
Enobio	8, 20, or 32	500 Hz	Wireless	

and the acquisition part is often complicated, as it is a rather time-consuming procedure. Moreover, the user's movement is restricted owing to the tethered nature of cable constraints. Hence, wireless BCI systems have gained due attention, primarily owing to their ability to mitigate the aforesaid restrictions. One of the attractive natures of the wireless EEG headset is that it is non-invasive. Moreover, it does not hinder the motion of the user. **Table 2** lists the types of EEG devices that have been reported in the literature with their specifications. It is evident that the selection of the type of EEG headset or device is dependent on the BCI application itself[1].

EEG Data Pre-processing Strategies

A small SNR and different noise sources are amongst the greatest challenges in EEG-based BCI application studies. Unwanted signals contained in the main signal can be termed noise, artifacts, or interference. There are two sources of EEG artifacts: external or environmental source and physiological source. The external sources of noise include AC power lines, lighting, and a large array of electronic equipment (from computers, displays, and TVs to wireless routers, notebooks, and mobile phones, amongst others). Physiological noise arises from an assortment of body activities due to movement, other bioelectrical potentials, or skin resistance fluctuations. The predominant physiological noises include electrooculographic activity (EOG, eye), electrocardiographic activity (ECG, heart), scalp-recorded electromyographic activity (EMG, muscle), ballistocardiographic activity (heart-related pulsatile motion), and respiration (Somers et al., 2018).

Pre-processing is a non-trivial process, as it is carried out to remove any unwanted components embedded within the EEG signal. Good preprocessing leads to an increase in the signal quality, which in turn results in better feature separability and classification performance. Simple low, high, and band-pass filters are the primary attempts to attenuate artifacts in the measured EEG. However, these are only effective when the frequency bands of the signal do not overlap (Sweeney et al., 2012). In case of spectral overlap, where artifacts are recorded with the EEG, alternative artifact removal techniques are required such as adaptive filtering, Wiener filtering, Bayes filtering (Sweeney et al., 2012), surface Laplacian transforms (Fitzgibbon et al., 2013), regression (Gratton et al., 1983), Common Average Referencing (CAR) (Zaizu Ilyas et al., 2015), EOG correction (Croft and Barry, 2000), and blind source separation (BSS) (Oosugi et al., 2017), as well as more modern attempts, for instance, the wavelet transform (WT) method (Punsawad and Wongsawat, 2017), empirical mode decomposition (EMD) (Zhang et al., 2008), Canonical Correlation Analysis (CCA) (de Clercq et al., 2006), and non-linear mode decomposition (NMD) (Iatsenko et al., 2015). It is worth noting that the BSS methods are also called component-based techniques, as they employ principal component analysis (PCA) or independent component analysis (ICA). Kilicarslan et al. (1976) proposed a quick artifact segment identification technique through the combination of dynamic time warping (DTW) and temporal motifs. Chavez et al. (2018) proposed a

[1] *Top 14 EEG Hardware Companies [Ranked].* Available online at: https://imotions.com/blog/top-14-eeg-hardware-companies-ranked/ (accessed February 03, 2019).

data-driven algorithm, namely surrogate-based artifact removal (SuBAR), to remove muscular and ocular artifacts effectively from EEG. A joint approach combining BSS and REG, the online EEG artifact attenuation technique, has also been proposed for BCI applications (Guarnieri et al., 2018). Although there is no single gold standard in EEG artifact removal algorithms, the authors of Urigüen and Garcia-Zapirain (2015) recommend using an ICA algorithm based upon second-order blind identification (SOBI) due to its success in removing different types of EEG contaminants.

Real-time BCI applications require artifact removal methods that are automatic and of low computational cost. Regression and filtering approaches can be executed automatically when they have a reference signal. Moreover, BSS methods will be automatic when there is a subsequent procedure. Although ICA is the most commonly used technique among these BSS methods, it disregards the temporal or spatial relations within sources, which will result in the loss of relevant information. Nevertheless, a CCA algorithm can mitigate this problem as it takes little computational time, which makes the algorithm applicable for real-time performance. Another factor that should be taken into account is the number of measurement channels. It should be noted that for home healthcare environments, fewer channels are often expected. BBS algorithms cannot be utilized in such a situation, due to the principle of BSS that it requires more channels in order to allow it to be more accurate. However, it should be noted that wavelet transform and EMD-based methods can be executed with a single channel, as they can decompose from a single record into multiple components (Chen et al., 2014). However, a reduction in the number of measurement channels will cause an increase in computational complexity, which will not be suitable for BCI applications.

In addition, it is worth noting that automatic methods are not commonly used for artifact removal, as there are multiple types of artifacts that exist in the recordings. Hence, the availability of reference signals will improve the accuracy and robustness of artifact removal by providing satisfactory complementary information. Also, the information on the artifact epochs obtained by the reference channel will reduce the computational cost. However, having a reference channel for each muscle contributing to EEG muscle artifacts is not feasible. Apart from the aforementioned methods, there are plenty of innovative and efficient approaches for artifact removal that have been recently proposed. One recently emerging BSS algorithm, independent vector analysis (IVA), integrates the advantages of CCA and ICA into one single framework. This technique could remove muscle artifacts by synchronously extracting the sources with maximal independence and maximal autocorrelation (Chen et al., 2017a,b). In addition, the combination of EEMD and IVA has been demonstrated to outperform other existing methods in a situation where there are few channels (Xu X. et al., 2018). More recently, a modified joint BSS approach and quadrature regression IVA (q-IVA) provided a more effective artifact removal technique in both the time and frequency domains, paving the way for future research (Lee S. et al., 2019). Dhindsa (2017) proposed a filter-bank-based supervised machine learning approach to detect artifacts from a single channel, and the approach outperformed statistical thresholding for EEG artifact rejection due to its ability to identify small artifacts in the presence of high-amplitude EEG. Mohammadpour and Rahmani (2017) have utilized an HMM architecture to remove eyeblink artifacts. Contrary to conventional algorithms, machine learning-based approaches have gained due attention, particularly for their ability to identify artifacts. To attenuate eye blink artifacts, a multichannel Weighted Weiner filter has been proposed (Manojprabu and Sarma Dhulipala, 2020), where Hierarchical Fully Connected Topology (HFCT) and *Ad-hoc* Nearest-Neighbor Topology (ANNT) are utilized. The proposed approach provides 5% better results for artifact attenuation when compared with the other existing approaches like PCA and ICA. However, it should be noted that the proposed approach has not been employed in real medical devices.

Feature Extraction Approaches in EEG-Based BCI Systems

After the noise removal phase, the most discriminative and non-redundant information within the EEG is extracted through different feature extraction techniques. Time-domain, frequency-domain, time-frequency domain, and spatial domain are the popular types of feature extraction techniques in EEG-based BCIs.

A typical time-domain-based feature extraction approach, autoregressive (AR) modeling, is a linear regression of the current observation of the series against one or more earlier observations. A combination strategy of feature extraction, where each feature vector consists of AR coefficients and approximate entropies, was also proposed. In many recent articles (Lawhern et al., 2012; Zhang and Xiaomin, 2015; Chai et al., 2017b), the AR model has been implemented as the strategy of feature extraction in EEG-based BCI systems. AR models are preferred by researchers due to their resolution, smoother spectra, and applicability to short segments of data. Lower model orders represent the signal poorly, while higher orders increase noise. Hence, identifying the appropriate AR modeling order is an open challenge. Conventional ways of modeling order estimation incorporate a Bayesian information criterion, Final prediction error, or Akaike Information Criterion (AIC). It was hypothesized in Atyabi et al. (2016) that an adequate mixture of AR features derived from various AR modeling orders is a better representative of the underlying signal compared with any fixed modeling order. For the detection of drowsiness state from EEG signals, the analysis of respiratory rate variability from EEG (Guede-Fernández et al., 2019), adaptive Hermite decomposition (Taran and Bajaj, 2018), and RR time series (Tripathy and Rajendra Acharya, 2018) have been employed to extract features. In emotion recognition using an EEG signal, the fractal dimension of raw signals has been implemented to extract the feature by using the Higuchi technique (Anh et al., 2012; Kaur et al., 2018). In AydIn et al. (2009), the authors proposed the use of log energy entropy to extract EEG features; this approach could investigate how much randomness is captured in the signal. A

hybrid feature extraction technique consists of PCA, and the cross-covariance technique has been developed in Zarei et al. (2017) to excerpt discriminatory information from the mental states of EEG.

Frequency-domain analysis has also been employed to extract features from different EEG-based BCIs. Among frequency-domain-based techniques, there are those that use fast Fourier transform (FFT) (Hortal et al., 2015; Djamal et al., 2017; Bousseta et al., 2018; Yang C. et al., 2018), power spectral density (PSD) (Chiappa and Bengio, 2004; Carlson and Millan, 2013; Mara et al., 2013; Pham et al., 2013; Bascil et al., 2016; Liu Y. et al., 2017; Nguyen et al., 2017; Chakladar and Chakraborty, 2018b), band power (Mandel et al., 2009; Serdar Bascil et al., 2015; Kreilinger et al., 2016), and spectral centroid (Murugappan et al., 2014). The PSD of a signal can be calculated through the FFT and Welch's method (Oikonomou et al., 2017). Welch's method reduces the artifacts in the PSD, in contrast to FFT, but produces a poorer frequency resolution. Another frequency domain-based feature extraction technique that does not require FFT to compute the PSD is local characteristic-scale decomposition (Liu A. et al., 2017). This procedure disintegrates the raw data into inherent segments that convey the properties of the primary signal. Fourier analysis decomposes the signal into its frequency components and determines their relative strengths. Due to the non-stationarity and non-Gaussianity properties of the EEG signals, classic spectral analysis techniques are not suitable for extracting useful and important information. Gursel Ozmen et al. (2018) introduced a biologically inspired frequency domain-based feature extraction approach that extracts the most discriminative spectral features from the PSD of the EEG signals. Meziani et al. (2019) presented novel spectral estimators, namely the quantile periodogram, and the lasso quantile periodogram, which are based on quantile regression and L1-norm regularization, respectively.

The use of spectral characteristics for feature extraction is sometimes ineffective due to the absence of temporal characteristics. Similarly, time-domain interpretation occasionally neglects spectral characteristics that may be important to the classifier. To overcome the shortcomings of a single domain that is either time domain or frequency domain, time-frequency analysis is assumed to be able to mitigate the issue as it leverages both domains. This approach could be promising for EEG-based BCIs. A variety of time-frequency-based feature extraction approaches have been employed in EEG-based BCIs. The most widespread approaches are short-time Fourier transform (STFT) (Tabar and Halici, 2017; Chaudhary et al., 2019; Ha and Jeong, 2019; Tian and Liu, 2019), continuous wavelet transform (CWT) (Borisoff et al., 2004; Lee and Choi, 2019; Ieracitano et al., 2020), discrete wavelet transform (DWT) (Guo et al., 2015; Bajwa and Dantu, 2016; Djamal and Lodaya, 2017; Ji et al., 2019; Lin and She, 2020), and wavelet packet decomposition (WPD) (Bong et al., 2017; Dhiman et al., 2018; Wang et al., 2019). CWT (Ortiz-Echeverri et al., 2019; Mammone et al., 2020), and STFT (Dai et al., 2019) have been utilized to generate spectral images that can be classified through deep learning approaches. An EEG-based motor planning exercise was investigated by Mammone et al. (2020), where a time-frequency map, generated through beamforming and CWT, was utilized as input to the CNN. Decomposition techniques, for instance, DWT and WPD, are efficacious because significant information is carried in different EEG bands (Kevric and Subasi, 2017), and these approaches are capable of decomposing the brain waves at multiresolution and multiscale (Li et al., 2016a). Moreover, they are able to extract dynamic features, which is crucial for EEG signals due to their non-stationary and non-linear characteristics (Kevric and Subasi, 2017). In Kevric and Subasi (2017), three distinct decomposition techniques, namely, WPD, EMD, and DWT, have been investigated to gain optimum accuracy. Higher-order statistics (HOS) features have been extracted from the decomposed EEG sub-bands. The frequency resolution of DWT coefficients is comparatively lower than that of WPD, and the deficiencies of wavelet strategies could be neutralized by HOS.

Zhou et al. (2018) combined the utilization of DWT and Hilbert transform (HT) for feature extraction. The EEG signal is decomposed through DWT, and the wavelet envelope of the decomposed sub-bands was computed through HT. They utilized both time-series and envelope information, which assisted in obtaining optimum accuracy. Göksu (2018) proposed wavelet packet analysis (WPA) to extract features from an EEG-SCP response. The WPA sub-images were further studied through log energy entropy. Yang et al. (2016), proposed Fisher wavelet packet decomposition (WPD)-CSP for extracting features, in which EEG channels are decomposed by WPA, the average power of each subband is calculated, and then, finally, CSP is employed to the selected subbands.

Another powerful feature extraction approach known as the common spatial pattern is extensively utilized in EEG-based BCI (Zhang R. et al., 2019). This technique utilizes a spatial filtering method that converts brain waves into a unique space where the variance of one group is magnified, while lower variance is seen in the remaining group. The pure CSP approach sometimes cannot achieve sufficient performance due to the subject-specific optimal frequency band. Hence, the choice of an optimized filter band may enhance performance, but the selection of the optimal sub-band through pure CSP takes a large amount of time. To overcome this issue, numerous changes have been applied to the CSP. The common spatio-spectral pattern approach (CSSP) combines an FIR filter with a CSP algorithm and was observed to improve performance relative to pure CSP (Reddy et al., 2019). Common sparse spatio-spectral patterns (CSSSP) (Dornhege et al., 2006) are a comparatively more advanced procedure where the common spectral patterns across channels are investigated. In sub-band common spatial pattern (SBCSP) (Khan et al., 2019), EEG is first filtered at different sub-bands, and then CSP features are calculated for each of the bands. Frequency bands of the CSP, for instance, filter bank CSP (FBCSP), were implemented in Korik et al. (2019), whereas wavelet CSP (WCSP) has been implemented (Lin et al., 2019) by considering the effect of frequency resolutions. However, these strategies are not significantly relevant to EEG data from selected electrodes. To mitigate this issue, a new technique for feature extraction from selected channels known as regularized CSP (RCSP) has been proposed (Jin et al., 2019).

The efficient frequency recognition algorithm in SSVEP-based BCIs performs a crucial role in overall system performance. Among these algorithms, the most prevalent are based on multivariate statistical algorithms, for instance, canonical correlation analysis (CCA) (Chen et al., 2015) and multivariate synchronization index (MSI) (Zhang Y. et al., 2016). Recently, another MSI-based frequency recognition approach known as CORRCA, which is based on correlated component analysis (COCA), has been proposed (Zhang et al., 2018). The CORRCA approach performs substantially better than the state-of-art CCA process. Authors in Zhang Y. et al. (2019) proposed a hierarchical feature fusion architecture that consists of the spatial dimension and frequency dimension to improve the performance of frequency identification techniques in SSVEP-based BCI.

Batres-Mendoza et al. (2016) proposed a novel feature extraction approach based on quaternions, which represent objects within a three-dimensional space with regards to their orientation and rotation. Islam et al. (2018) proposed a multiband tangent space mapping with sub-band selection (MTSMS) approach to improve EEG-MI classification accuracy, and the authors claim that the proposed framework outperforms state-of-the-art methods. Authors in Lee S. B. et al. (2019) investigated the comparative analysis of EEG features in three different domains, namely spectral, temporal, and spatial, to classify multi-class MI data. According to their investigation, the time-domain parameter (TDP) has been observed to be superior as compared to the CSP and PSD. Another study explored the use of tunable Q-factor wavelet transform (TQWT) for the identification of drowsiness EEG signals (Al Ghayab et al., 2019). With this approach, TQWT decomposes the EEG signals into band-limited sub-bands, and the drowsiness and alertness EEG signal characteristics from TQWT-provided sub-bands are extracted using time-domain measures. These measures are based on the statistics of Hjorth mobility. Moreover, a novel hybrid feature extraction technique has been proposed (Asadur Rahman et al., 2019), which consists of PCA and t-statistics. In Guede-Fernández et al. (2019), the analysis of respiratory rate variability of EEG has been utilized to monitor the state of drowsiness in a driver.

Classification Methods

To operate a BCI system, the subject needs to create various brain activity patterns that can be identified by the system and translated into commands. It is worth noting that either regression or classification algorithms could be utilized to achieve the said objective. However, the usage of classification algorithms is presently reported to be the most popular approach (Lotte et al., 2007). The design of the classification step includes the choice of one or several classification algorithms from many alternatives. Numerous classification algorithms have been presented in the published EEG-based BCI literature, for instance, support vector machine (SVM), neural network (NN), linear discriminant analysis (LDA), Bayesian classifier, k-nearest neighbor (*k*-NN), as well as deep learning and its iterations. The aforesaid classifiers are described briefly, and their essential properties for BCI applications are highlighted.

Conventional Machine Learning Approaches in EEG-Based BCIs

The *k*-NN algorithm depends on the principle that the features corresponding to the several classes will form individual clusters in feature space. The features that are closer to each other are recognized as neighbors and are therefore grouped together. This classifier takes k metric distances into account between the test sample features and those of the nearest classes in order to classify a test feature vector. The metric distances are a measure of the similarities between the features of the test vector and the features of each class (Nicolas-Alonso and Gomez-Gil, 2012). It is worth highlighting that the *k*-NN algorithms are not exceptionally popular in the BCI community because they are known to be very sensitive to the dimensionality of the feature vector (Borisoff et al., 2004). Nonetheless, when the algorithm is utilized in BCI systems with low-dimensional feature vectors, the algorithm could be useful. Thus, the *k*-NN algorithm can provide good results when it is combined with other efficient feature selection and reduction algorithms. In *k*-NN architecture, the number of neighbors and the type of distance metrics are the key factors.

LDA is employed to find the linear combinations of feature vectors that describe the characteristics of the corresponding signal. The LDA seeks to separate two or more classes of objects or events representing different classes. It utilizes hyperplanes to accomplish this mission. The isolating hyperplane is achieved by searching for the projection that maximizes the distance among the classes' means and minimizes the interclass variance (Abdulkader et al., 2015). This technique has a very low computational requirement, and it is simple to use. The LDA has been successfully applied in a variety of BCI systems, for example, motor imagery-based BCI, P300 speller, multiclass, or asynchronous BCI (Long et al., 2012a). However, while it provides expected outcomes due to its immunity to non-stationary issues, its linearity can cause performance degradation in a few circumstances containing complex non-linear EEG data. Moreover, numerous updated algorithms have been presented depending on LDA, for example, Fisher LDA (FLDA) as well as Bayesian LDA (BLDA) (Hoffmann et al., 2008). FLDA does not work well if the number of features becomes too large in relation to the number of training examples. This issue is called the small sample size problem (Hoffmann et al., 2008). On the other hand, the BLDA is considered as an expansion of FLDA that mitigates the small sample size problem through the incorporation of a statistical method called regularization. The regularization is estimated through Bayesian analysis of training data and is utilized to prevent the overfitting problem of high-dimensional as well as possibly noisy datasets. Overfitting means that the classifier has lost generality and is therefore undesirable in a classifier. If a classifier is overfitted, it is only able to classify the training data or similar data. Unlike FLDA, the BLDA algorithm gives higher classification accuracy and bitrates, particularly in those situations where the size of the training sample is large (Hoffmann et al., 2008). Furthermore, BLDA requires slightly

more computational time, which is a crucial constraint in real BCI systems.

SVM is a classifier that builds a hyperplane or set of hyperplanes for separating the feature vectors into several classes. However, in contrast to LDA, SVM selects the hyperplanes that maximize the margins, that is, the distance between the nearest training samples and the hyperplanes (Burges, 1998). SVM that empowers classification by utilizing linear decision boundaries is called linear SVM. This type of classifier has been applied successfully to a moderately large number of synchronous BCI problems (Garrett et al., 2003; Rakotomamonjy et al., 2005). However, it is advantageous to make non-linear decision boundaries with a low increment of the classifier's complexity by utilizing the "kernel trick." The kernel usually utilized for BCI research is the Gaussian or Radial Basis Function (RBF) kernel. The corresponding SVM is called Gaussian SVM or RBF SVM. The RBF SVM has also shown to be robust for achieving good results in BCI applications (Garrett et al., 2003; Kaper et al., 2004; Rakotomamonjy et al., 2005). Generally, the SVM has been broadly recognized as the simpler algorithm used in BCI applications. In addition, the algorithm is shown to be robust with a high-dimensional dataset, which implies that a large training set is not required for good outcomes, even with high-dimensional feature vectors (Kaper et al., 2004). It is worth noting that these favorable circumstances do not hinder the execution speed during the integration of real-time BCIs (Thulasidas et al., 2006).

The neural network (NN) has the special capacity to extract patterns and identify trends that seem to be complicated, either by humans or by computerized techniques. A trained NN system can be considered as an "expert" in the classification of information that it has been provided to analyze. This algorithm is one of the fundamental tools utilized in machine learning. The term "neural" denotes that it is considered to be a brain-inspired system that is intended to replicate the way that humans learn. A NN consists of input, output, and hidden layers. A hidden layer consists of units that transform the input into something that the output layer can utilize. It is an excellent tool for discovering patterns that are too complex or numerous for a human programmer to extract and to teach the machine to perceive. One of the most well-known ANN structures is the multilayer perceptron (MLP) introduced by Rumelhart et al. (1986). MLPs are flexible classifiers with the ability to classify any number of classes as well as to adapt to various different sets of problems. In BCIs, the MLP has been applied to classify two, three, and five tasks and to design synchronous (Haselsteiner and Pfurtscheller, 2000) as well as asynchronous (Millan and Mourino, 2003) BCIs. Additionally, the MLP has been utilized to preprocess EEG signals prior to the feature extraction step rather than the classification step to improve the separability of EEG features (Coyle et al., 2010). Other than MLP, numerous sorts of NN architecture have been utilized in the design of BCI systems, including Fuzzy ARTMAP Neural Networks, Finite Impulse Response Neural Networks (FIRNN) or Probability estimating Guarded Neural Classifiers (PeGNC), Probabilistic Neural Networks (PNN), Time-Delay Neural Networks (TDNN) or Gamma dynamic Neural Networks (GDNN), Learning Vector Quantization (LVQ) Neural Networks, Bayesian Logistic Regression Neural Networks (BLRNN), RBF Neural Networks, and Adaptive Logic Networks (ALN), amongst others.

A Hidden Markov Model (HMM) is a Bayesian classifier that produces non-linear decision boundaries. An HMM is a sort of probabilistic automaton that gives the likelihood of observing a given sequence of feature vectors (Rabiner, 1989). For BCI, generally, these probabilities are Gaussian Mixture Models (GMM) (Obermaier et al., 2001). HMMs are perfectly appropriate algorithms for the classification of time series (Rabiner, 1989). As the EEG components that are used to operate BCI have specific time courses, HMM is applicable to the classification of temporal sequences of BCI features (Obermaier et al., 2001; Cincotti et al., 2003), even for classifying raw EEG signals.

Deep Learning Approaches in EEG-Based BCIs

The ability to acquire a robust automatic classification of EEG signals is an essential step toward making the use of EEG more practical in many applications and less reliant on trained professionals (Alexander et al., 2018). It is worth noting that although conventional BCI systems have made tremendous advances in the past few decades, nonetheless, the research still faces significant challenges in EEG classification. The challenges include various biological and environmental artifacts in EEG, a low SNR, and dependency on human expertise for extracting meaningful features. In addition, most existing machine learning research, if not all, centers on static data and, hence, is not able to classify rapidly changing brain signals accurately (Lotte et al., 2018). Of late, the availability of large EEG data sets has led to the utilization of Deep Learning (DL) architectures, especially to uncover relevant information from the signals that were not possible to acquire via conventional approaches and has shown success in addressing the aforesaid challenges. Fundamentally, DL is a specific machine learning algorithm in which the features and the classifier are jointly learned directly from data (Zhang X. et al., 2019). DL algorithms have been explored for almost all major types of EEG-based BCI systems, namely P300, SSVEP, motor imagery (MI), SCP, and passive BCI (for emotions and workload detection). Here, a number of prevalent DL models including convolutional neural networks (CNN), deep belief networks (DBN) restricted Boltzmann machines (RBM), recurrent neural networks (RNN), a stacked autoencoder (SAE), and generative adversarial networks (GAN) will be discussed briefly with regards to their employment in BCI research.

A Convolutional Neural Network (CNN) is a special type of neural network architecture that is specialized in spatial information exploration. CNN contains at least one convolutional layer, and this layer maps input to an output through a convolution operator (Fan et al., 2019; Zhang X. et al., 2019). In BCI research, CNN is assumed to capture the distinctive dependencies amongst the patterns associated with different brain signals (Lotte et al., 2018). Recently, a considerable amount of studies (Tang et al., 2017; Aznan et al., 2018; Dose et al., 2018; El-Fiqi et al., 2018; Shojaedini et al., 2018; Wang et al., 2018; Waytowich et al., 2018; Amber et al.,

2019; Amin et al., 2019; Nguyen and Chung, 2019; Olivas-Padilla and Chacon-Murguia, 2019; Tayeb et al., 2019; Xu et al., 2019) on the employment of CNN architecture in EEG-based BCI systems have been published. In Olivas-Padilla and Chacon-Murguia (2019), the classification of multiple MI using CNN was explored, with the features being extracted by a variety of Discriminative Filter Bank Common Spatial Patterns (DFBCSP). Conversely, the authors in Xu et al. (2019) presented a wavelet transform time-frequency image coupled with a CNN-based approach in classifying EEG MI, and a classification accuracy of 92.75% was attained. Tayeb et al. (2019) classified raw EEG MI signals using a CNN architecture, achieving an accuracy of 84%, and this model has been successfully adopted in a real-time robotic arm control system. A CNN-based multilevel feature fusion model was proposed in Amin et al. (2019) for motor imagery EEG classification. Three other studies also employed CNN for EEG MI classification with reported the classification accuracies of 80, 93, and 86%, respectively (Tang et al., 2017; Dose et al., 2018; Wang et al., 2018). It should also be noted that the CNN model has also been employed in SSVEP-based BCI systems. A novel CNN approach for the classification of raw SSVEP EEG signals was proposed in Aznan et al. (2018). Here, the CNN architecture achieved a classification accuracy of 96%, which is significantly better than other competing DL approaches. In El-Fiqi et al. (2018), raw SSVEPs were classified using CNN for person identification and verification. In addition, a 1-D CNN was employed for SSVEP frequency detection with an average accuracy of 97.4% (Nguyen and Chung, 2019). A compact-CNN approach was proposed in Waytowich et al. (2018), and it was able to decode signals from a 12-class SSVEP dataset with a mean accuracy of ∼80%. With regard to the use of CNN on P300, Amber et al. (2019) presented a lie detection system from the P300 signals with an accuracy of 99.6%. In addition, a new adaptive hyperparameter-tuning method is proposed in Shojaedini et al. (2018) to improve the training of CNN in P300 signal detection. It was established from the study that the proposed method is able to improve the classification accuracy by 6.44% against the conventional Naive hyperparameter tuning method.

A deep belief network (DBN) is a probabilistic generative model consisting of a sequence of restricted Boltzmann machine (RBM) architectures (Abbas et al., 2019). The top two layers in DBN are connected without directions, while the lower layers are connected with directions. The RBM consists of a visible layer and a hidden layer, and the connection lines between these two layers are undirectional (Abbas et al., 2019). Several studies have explored MI classification with DBN (An et al., 2014; Tang et al., 2015; Lu et al., 2017; Ortega et al., 2017). In Lu et al. (2017), a novel deep learning scheme based on RBM was proposed for EEG MI classification in which FFT and wavelet package decomposition are obtained to train three RBMs. These RBMs are then stacked up with an extra output layer to form a four-layer frequential DBN. The authors of Tang et al. (2015) proposed an EEG MI data recognition technique using DBN. The findings from the study showed that the recognition rate of EEG MI data based on a DBN is better than that with the conventional SVM model. A novel technique of classification of imagined speech in EEG was proposed in Lee and Sim (2015), where the classification accuracy obtained was 87.96% with DBN. A P300-based Guilty Knowledge Test system was proposed in Kulasingham et al. (2016). Here, the DBN architecture was used to classify the P300 wave with an accuracy of 86.9%, and the input to this classifier was the filtered EEG signal without any feature extraction. Another P300 potential detection method based on DBN has been proposed in Lu et al. (2018), where the average accuracy attained was 84.3%. A DBN architecture has also been exploited successfully for EEG-based emotion recognition (Zheng and Lu, 2015; Huang et al., 2017). An EEG-based emotion classification framework based on combining emotional patches and a DBN model was proposed in Huang et al. (2017), and it was reported that a classification accuracy of 94.92% was achieved, outperforming other traditional methods. The authors of Kawde and Verma (2017) implemented an effective recognition system to examine the emotional state of a human being based on DBN. The experiment was performed on a benchmark DEAP database, and the accuracies achieved were 78.28, 70.33, and 70.16% for valence, arousal, and dominance, respectively. In Bablani et al. (2018), a system for identifying deceit from EEG has also been proposed. A DBN was developed with four RBMs stacked together, and EEG data in the form of time-frequency was fed to this DBN. The accuracy of this system was recorded at 81%. In Chai et al. (2017a), an EEG-based driver fatigue classification between fatigue and alert states was investigated. The system employs an AR model as the feature extraction algorithm and a sparse-DBN as the classification algorithm. It was shown from the study that a classification accuracy of 93.1% was attained.

RNN architecture is a powerful deep learning classification method that is specifically applied to sequential data. This type of DL architecture is able to analyze the overall logical sequence between the input information. These logical sequences are rich in content and possess a complex time relationship with each other. The key concept of RNN is that the hidden state of the current network will retain the previous input information, and it is used for the next current network (Li et al., 2019). There are two typical RNN architectures that have attracted much attention and achieved great success: long short-term memory (LSTM) and gated recurrent units (GRU). Two notable studies have been carried out to recognize the EEG-based sleep stage using RNN architecture (Michielli et al., 2019; Wang and Wu, 2019; Zhang T. et al., 2019). A novel cascaded RNN architecture based on LSTM blocks was proposed in (Michielli et al., 2019) for the automated scoring of sleep stages using EEG, and an average classification accuracy of 86.7% was achieved. The authors of Wang and Wu (2019) also developed an automatic sleep stage classification system where they proposed an RNN based on the attention mechanism and bidirectional LSTM. This architecture provided better performance than the C-CNN model but requires more training time. A novel DL framework called spatial-temporal RNN (STRNN) was proposed in Zhang T. et al. (2019), where both spatial and temporal information were integrated for feature learning. The authors claimed that the experimental results based on STRNN were more competitive than the state-of-the-art methods for emotion recognition. In another study (Liu et al., 2018), the combination of temporal attention and band attention

mechanisms based on multi-layer LSTM-RNN architecture was proposed for emotion recognition. Another study (Jawed et al., 2018) distinguished visual and non-visual learners by considering the wavelet features of EEG alpha and beta bands. The LSTM-based RNN framework was also used for classification purposes, and the mean training accuracy was 87.5 and 86% for beta and alpha bands, respectively. In relation to EEG MI classification, Ma et al. (2018) proposed a pure RNN-based parallel method for encoding spatial and temporal raw data with bidirectional LSTM and standard LSTM, respectively, reporting an average accuracy of 68.20%. A deep RNN with a sliding window cropping strategy (SWCS) to classify EEG MI signals was investigated in Luo et al. (2018). In addition, an LSTM-RNN architecture for an EEG MI classification model was proposed in Li et al. (2016b), where DWT was applied to extract the time-frequency features. A novel system for cross-day workload estimation using EEG has also been proposed by Hefron et al. (2017), where the authors applied an LSTM-based RNN architecture, and the average classification accuracy achieved was 93.0%.

An autoencoder (AE) is a DL approach used for unsupervised feature learning with efficient data encoding and decoding. In the encoding phase, the input samples are often mapped in the lower dimensional feature space with a constructive feature representation (Alom et al., 2019). This approach can be repeated until the desired feature dimensional space is reached. Conversely, in the decoding phase, actual features are regenerated from the lower-dimensional features with reverse processing (Alom et al., 2019). It should be pointed out that the target output of the autoencoder is the autoencoder input itself. There are a number of notable AE architectures, i.e., Stacked Autoencoder (SAE), Variational Autoencoder (VAE), and Generative Adversarial Networks (GAN), that have been employed in EEG signal processing investigations (Tsinalis et al., 2016; Vareka and Mautner, 2017; Yin and Zhang, 2017; Ditthapron et al., 2018; Idowu et al., 2018; Nair et al., 2018; Rundo et al., 2019). The authors of Rundo et al. (2019) developed a drowsiness detection system from EEG using stacked AE and achieved an accuracy of 100% in discriminating drowsy from wakeful. Idowu et al. (2018) proposed a DL-based classification of EEG signals for given visual stimuli by showing familiar and unfamiliar faces. The preprocessed signal was fed to an AE that yielded a mean accuracy of 82.21%. In Nair et al. (2018), five-class EEG MI data was classified, where SAE was applied to generate the features, and a softmax layer was then used for classification purposes. The proposed method produced an overall accuracy of 98.9%. In Ditthapron et al. (2018), the authors proposed a multitask autoencoder-based model known as the ERP encoder network (ERPENet) that can be applied to any ERP-related tasks. In Vareka and Mautner (2017), an SAE architecture was proposed for P300 wave detection, and the trained SAE achieved a classification accuracy of 69.2%. An automatic sleep stage scoring model that uses a single channel of EEG was proposed in Tsinalis et al. (2016). Here, the methodology is based on time-frequency analysis and stacked sparse autoencoders (SSAEs). The overall accuracy attained was 78%. With regard to the mental workload (MW) classification, several studies have been carried out (Yin and Zhang, 2017; Yang et al., 2019; Yin et al., 2019). An adaptive Stacked Denoising Auto Encoder (SDAE) was developed in Attia et al. (2018) to tackle cross-session MW classification from EEG, and it was reported that the proposed classifier achieved an accuracy of 95.5%.

Apart from the aforesaid standalone DL models, researchers have attempted to hybridize different DL models in EEG-based BCI investigations (Narejo et al., 2016; Attia et al., 2018; Yang J. et al., 2018; Dai et al., 2019; Kanjo et al., 2019), with encouraging classification accuracies. In Narejo et al. (2016), the authors developed a system for predicting eye state from EEG signals using a hybrid DL architecture consisting of DBN and SAE. The accuracy of this hybrid model was reported to be as high as 98.9%. Another hybrid DL architecture based on CNN-RNN was proposed in Attia et al. (2018) to classify SSVEP signals in the time domain directly, and it achieved an accuracy of 93.59%. Kanjo et al. (2019) proposed a hybrid approach that applied CNN and LSTM-RNN on the raw sensor data. Through this method, the need for manual feature extraction is eliminated. The results show that the adoption of DL approaches is effective in human emotion classification when a large number of sensor inputs are utilized (with an average classification accuracy of 95%). Dai et al. (2019) proposed a hybrid DL model where a CNN architecture was combined with a VAE for EEG MI classification. In addition, an LSTM-CNN-based hybrid model has also been proposed by Yang J. et al. (2018) for EEG MI classification.

The fundamental idea of a Riemannian geometry classifier (RGC) is to map the data directly onto a geometrical space equipped with a suitable metric (Lotte et al., 2018). In such a space, data can be easily manipulated for several purposes, such as averaging, smoothing, interpolating, extrapolating, and classifying. In the case of EEG data, the power and the spatial distribution of EEG sources can be considered fixed for a given mental state, and such information can be coded by a covariance matrix (Lotte et al., 2018). Riemannian geometry studies smooth curved spaces that can be locally and linearly approximated. The curved space is named manifold, and its linear approximation at each point is known as the tangent space (Lotte et al., 2018). Riemannian geometry has been successfully utilized in many BCI classification problems (Kalunga et al., 2016; Congedo et al., 2017; Wu et al., 2017; Yger et al., 2017; Gaur et al., 2018; Guan et al., 2019; Han et al., 2019; Majidov and Whangbo, 2019) and has demonstrated superior performance. In Han et al. (2019), the authors implemented an EEG-based endogenous BCI system for online binary communication by a completely paralyzed patient. An online classification accuracy of 87.5% was achieved when the Riemannian geometry-based classification was applied to real-time EEG data. A number of investigations (Gaur et al., 2018; Guan et al., 2019; Majidov and Whangbo, 2019) have employed Riemannian geometry for EEG MI classification purposes. The authors of Majidov and Whangbo (2019) proposed a Riemannian geometry-based architecture for EEG MI classification. They combined the PSD features with covariance matrices mapped onto the tangent space of a Riemannian manifold, and an average classification accuracy of 87.94% was obtained. The use of a Riemannian geometry framework for EEG MI classification has also been presented in Gaur et al. (2018), where the EEG signals were preprocessed using a subject-specific multivariate empirical

mode decomposition (SS-MEMD)-based filtering method. They achieved a mean Kappa value of 0.60. Kalunga et al. (2016) investigated the efficiency of Riemannian geometry on SSVEP wave classification for a four-class BCI application. In the study, the minimum distance to Riemannian mean (MDRM) algorithm achieved an average classification accuracy of 90.47+/7.8% and an ITR of 16.37 ± 5.3 bits/min. A novel feature extraction approach based on Riemannian geometry was proposed in Wu et al. (2017), in which a spatial filter is first used to increase the EEG signal quality and to reduce the dimensionality of the covariance matrix, and then, finally, the Riemannian tangent space features are extracted. Moreover, it is worth noting that there are two review articles (Congedo et al., 2017; Yger et al., 2017) on the application of Riemannian geometry for BCI systems, which may be an excellent source of information for interested readers.

Performance Evaluation of BCI Systems

Overall, BCI performance depends entirely on classifier performance (Nicolas-Alonso et al., 2015). For the classification algorithm, the most basic performance measure is classification accuracy. Sometimes, the Kappa metric or the confusion matrix are also used to provide further information on the performance of a classifier (Fatourechi et al., 2008). The sensitivity-specificity pair or precision can be calculated from the confusion matrix. When the classification relies on a continuous parameter, the Receiver Operating Characteristic (ROC) curve, as well as the Area Under the Curve (AUC), is often utilized. Classifier performance is generally computed offline on pre-recorded data utilizing a hold-out strategy: some datasets are set aside to be utilized for the evaluation and are not part of the training dataset. However, some authors also reported cross-validation measures estimated on training data, which may over-rate the performance (Lotte et al., 2018; Raschka, 2018). A number of researchers (Farwell and Donchin, 1988; Iturrate et al., 2009a; Yeom et al., 2014; Obeidat et al., 2015; Ansari and Singla, 2016; Chang et al., 2016; Cao et al., 2017) also reported either the information transfer rate (ITR) or the practical bit rate (PBR) (Farwell and Donchin, 1988). Many articles (Allison et al., 2012; Long et al., 2012a; Li Y. et al., 2013; Cao et al., 2014; Wang H. et al., 2014) used task-specific metrics, including task completion time and the number of successful trials. These metrics are tailored for each BCI paradigm and/or application, and therefore do not allow comparisons between studies.

POPULAR EEG-BASED BCI APPLICATIONS

In BCI technology, human brain signals can be detected and translated into device commands for controlling assistive devices. Besides medical applications, the area of this technology has been expanded to non-medical applications. Recently, the possibility of a variety of BCI applications is being investigated by researchers around the world. The most important achievements in EEG-based BCIs include spelling systems, wheelchair control, robot control, mental workload, virtual reality, and gaming, environment control, driver fatigue monitoring, biometrics system, and emotion recognition.

BCI Wheelchair Control

One of the essential objectives of a BCI wheelchair is to upgrade the life quality as well as the autonomy of people affected by motor neuron diseases (MND), for instance, amyotrophic lateral sclerosis (ALS). This innovation assists the disabled users to operate the wheelchair using their brain activity, granting autonomy to travel through an experimental environment. In 2013, Bi et al. (2013) conducted a survey of BCI-controlled mobile robots, which is partially connected to this field. However, two articles (Fernández-Rodríguez et al., 2016; Al-qaysi et al., 2018; Tariq et al., 2018) have been published that contain extensive reviews on BCI wheelchairs. Four types of EEG control signal are used to handle BCI wheelchairs, which are MI (Li J. et al., 2013; Varona-Moya et al., 2015; Tang et al., 2018), P300 (Rebsamen et al., 2007; Iturrate et al., 2009a; Alqasemi and Dubey, 2010; Shin et al., 2010; Lopes et al., 2013), SSVEP (Mandel et al., 2009; Xu et al., 2012; Mara et al., 2013; Duan et al., 2014; Ng et al., 2014), and hybrid (Li Y. et al., 2013; Cao et al., 2014) signals. The feature extraction methods are quite heterogeneous; however, CSP is the most used EEG feature in BCI wheelchair applications (Li J. et al., 2013; Li Z. et al., 2013; Cao et al., 2014; Wang H. et al., 2014; Zhang R. et al., 2016). Other researchers used methods such as PSD (Varona-Moya et al., 2015; Tang et al., 2018), FFT (Duan et al., 2014), logarithmic band power (Arabnia and Tran, 2011; Duan et al., 2014), signal averaging techniques (Alqasemi and Dubey, 2010; Shin et al., 2010; Zhang R. et al., 2016), the amplitude of the target frequency (Mandel et al., 2009; Mara et al., 2013; Ng et al., 2014), CCA (Xu et al., 2012; Duan et al., 2014), and other methods (Rebsamen et al., 2007; Iturrate et al., 2009a; Lopes et al., 2013). With regard to the classification techniques, the most widely used algorithm is SVM (Rebsamen et al., 2007; Shin et al., 2010; Li J. et al., 2013; Li Y. et al., 2013; Zhang R. et al., 2016), followed by LDA (Iturrate et al., 2009a; Cao et al., 2014). Performance evaluation is the most challenging part in BCI research, as it is the most heterogeneous area. However, the most common metrics reported are success rate, classification accuracy, information transfer rate, path length, time required, path length optimality ratio, time optimality ratio, number of user commands, and number of collisions. Cao et al. (2014) used the highest number of metrics, namely ITR, CA, Collision, Time Required, Useful Command, Useless Command, and Stopping Task Time to evaluate their research. **Table 3** shows a summary of some articles regarding EEG-based wheelchair BCI.

BCI Cursor Control

The first attempt to control a cursor by EEG signal was described in Wolpaw et al. (1991). Here, vertical movement of a cursor on a video screen was maintained by changing mu-rhythm amplitude such that the cursor was moved upward by large amplitude mu-rhythm, while downward movement required small mu-rhythm amplitudes. The second experiment conducted by some of the same authors (McFarland et al., 1993) achieved a target hit rate of 54.85%. The promising result obtained encouraged other researchers to develop this research further.

TABLE 3 | Summary of EEG-based BCI wheelchair studies.

References	No. of subjects	EEG control signal	No. and types of control command	EEG features	Classification algorithm	Performance evaluation
Cao et al. (2014)	3	MI+SSVEP	8; Left, Right, Forward, Acceleration, Deceleration, Uniform Velocity, Turn ON, Turn OFF	CSP for MI; CCA for SSVEP	RBF SVM	ITR: 295.20; CA: 90.63%; Collision: 0; Time required: 370 ± 41; Useful command: 5 ± 3; Useless command: 2 ± 2; Stopping task time: 35 ± 4;
Iturrate et al. (2009a)	5	P300	18; Fifteen Locations, Left, Right, Validate selection	Moving average technique	Stepwise LDA (SWLDA)	Success rate: 100%, Time: 659 s, Path length: 39.3 m.
Long et al. (2012a)	2	MI+P300	4; Left, Right, Acceleration, Deceleration	CSP	LDA	Accuracy: 100%, Path length (pixel): 2843.46 ± 105.41, Time: 84.42 ± 4.63 s, Collisions: 0.
Mara et al. (2013)	9	SSVEP	4; Right, left, Forward, Stop	PSD	Decision tree	Success rate: 83 ± 15%, ITR: 70.3 ± 28.8 bits/min.
Li J. et al. (2013)	3	MI	3; Right, Left, Forward	CSP	SVM	Success rate: 82.56%
Li Y. et al. (2013)	5	P300+SSVEP	4; Forward, Stop, Turn ON, Turn OFF	Statistic average Minimum energy combination	SVM	Task duration: 4.30 s/command; TPR: 14.18 event/min, FPR: 0.49 event/min, ITR: 21.11 bit/min.

TABLE 4 | Summary of EEG-based BCI cursor control studies.

References	No. of subjects	Control signal	No. and types of control command	EEG features	Classification algorithm (CA)	Performance evaluation
Serdar Bascil et al. (2015)	2	MI	2; Left/Right	BP	PNN	CA: PNN: 93.05%
Long et al. (2012b)	11	MI+P300	4; Left, Right, Up, Down	CSP	SVM	Success rate: 93.99% Duration per trial: 18.19 s
Bascil et al. (2016)	5	MI	4; Left, Right, Up, Down	PSD	SVM	CA: 81.22%;
Chakladar and Chakraborty (2018b)	1	MI	4; Left, Right, Up, Down	PSD	DB-Scan	Execution time: 4.663 min, Success rate: 70.36%

Single control signals and hybrid control signals have both been suggested. Li et al. (2010) presented a BCI that enabled the subjects to control vertical movement as well as horizontal movement through P300 and ERD activity, respectively. The subjects could hit one of the four targets with hit rates between 82 and 96%, with average selection times between 25 and 26 s. In their second (Long et al., 2012b) experiment, the trial duration and average accuracy of the target selection were 18.19 s and 93.99%, respectively. The overall outcomes of the experiments were excellent, and the subjects could move the cursor diagonally by executing both sorts of activities simultaneously. However, the cursor control achieved by this system is not continuous. Allison et al. (2012), meanwhile, introduced MI/SSVEP-based hybrid BCI for simultaneous cursor control in two dimensions. The features and classification techniques utilized in this application are heterogeneous. **Table 4** shows a summary of some recent investigations in the domain of BCI cursor control.

BCI Spellers

An excellent review article (Rezeika et al., 2018) have been published where almost all types of recent BCI spellers have been summarized. **Table 5** tabulates the methodology that has

TABLE 5 | Summary of EEG-based BCI speller studies.

References	No. of subjects	Control signal	Method	Classification algorithm	Typing speed	Success RATE
Obeidat et al. (2015)	14	P300	Stimuli variation	Bayesian BLDA	13.7 bits/min	93.3%
Cao et al. (2017)	3	MI	Oct-O-spell	SVM	67.33 bits/min	98.23%
Ansari and Singla (2016)	20	SSVEP	Multi-phase spellers	SVM	13 chars/min	96.04%
Chang et al. (2016)	10	SSVEP+P300	Hybrid speller	CCA, SWLDA	31.8 bits/min	93%
Käthner et al. (2015)	19	P300	Familiar faces and symbols	SWLDA	15.85 bits/min	95%

been applied along with the performance of some recent EEG-based BCI speller system. In 1988, Farwell and Donchin (1988) presented a P300 speller is known as the matrix speller. It was the first BCI speller and had a maximum accuracy of 95% and a speed of 12 bits/min. An adjacency problem arises in the matrix speller, in that it is difficult to identify a target with a lot of similar objects surrounding it. This problem was avoided by a random-set representation and edges paradigm (Obeidat et al., 2015). Another sort of P300 base speller flashed up a familiar face over a character to improve the speed of spelling (Kaufmann and Kübler, 2014). The checkerboard paradigm (CBP) was presented in Townsend et al. (2010) for avoiding the adjacency-distraction problem and double flash issues. This paradigm achieved better accuracy and was more comfortable than a row-column paradigm (RCP) in ALS patients. A different sort of P300 base paradigm, namely a gaze-independent block speller, was also introduced (Pires et al., 2011), which has the ability to be utilized without ocular movement. Additionally, this paradigm creates almost the same information transfer rates as standard to a row-column speller. Acqualagnav et al. (2010) proposed rapid serial visual presentation (RSVP) with the aim of forming an efficient gaze-independent ERP speller. The subject showed better performance with the colored letters than with monochrome ones. The accuracy of the RSUP speller is better than that of the matrix speller. It is worth noting that one of the earliest high-speed SSVEP-based BCI spellers is the Bremen BCI speller (Volosyak et al., 2009). The efficacy of this speller was examined on both healthy and disabled people. The average ITR reported was 25.67 bits/min, with an accuracy of 93.27%. Furthermore, Cao et al. (2011) proposed a multi-phage SSVEP-based speller system that allows the input of 42 characters comprising letters, digits, and symbols. The mean ITR and mean accuracy of this speller are 61.64 bits/min and 98.78%, respectively. However, this speller did not include MND patients as subjects to test the system. In the same vein, Nakanishi et al. (2018) proposed a multi-target one-phase SSVEP speller by achieving an 89.83% accuracy and 325.33 bits/min ITR. In general, it is worth noting that a higher number of targets in SSVEP-based BCI is shown to increase the spelling speed but also to increase eye fatigue and target misclassification.

A Hex-O-Spell, which depends on imaginary movement, has also been reported on Blankertz et al. (2006). The speller was demonstrated to offer higher performance than the conventional matrix speller. A recent MI-based speller, namely Oct-O-Spell, was introduced in Cao et al. (2017), involving an octagon divided equally into eight sections. These sections contained a total of 26 letters, characters, digits, or symbols. The interface showed a similar performance to hybrid BCI spellers. Similarly, a hybrid BCI speller based on SSVEP and P300 was presented in Chang et al. (2016), again featuring an octagon divided equally into eight sections. Each section consisted of a total of 26 letters, characters, digits, or symbols. The interface showed a similar performance to hybrid BCI spellers. The authors of Nguyen et al. (2018) proposed a high-speed BCI spelling system using eye closure and double-blink EEG by means of an SVM classification algorithm. It was demonstrated from the investigation that the proposed system is able to achieve an average classification accuracy of 92.5% with an ITR of five letters per minute.

BCI Biometrics

Biometrics is the process of identifying one individual among others by biological means. Biometrics, including iris, face, and fingerprint recognition, is frequently applied to avoid security breaches. Nonetheless, the possibility of imitating, replicating, or stealing original information has made these tools unreliable. As a result, there has been a growing interest in finding a better biometric system and brain activity-based biometrics. An EEG system-based biometric has been identified to have the advantage of being quite impossible to mimic (Alariki et al., 2018). Using 15 human participants, Ruiz Blondet et al. (2015) studied the stability of EEG brainwaves over a 6-month period. Based on their findings, it was shown that the accuracy of EEG signals for biometric systems and the stability of human brain activities could remain stable over a long time. Bashar et al. (2016) proposed a method for human identification using EEG signals. The authors used a bandpass FIR filter to remove noise and then divided the EEG signals into two sections. Multi-scale Wavelet Packet Statistics (WPS), Multi-scale Shape Description (MSD), and multi-scale Wavelet Packet Energy Statistics (WPES) were utilized as the features in this

TABLE 6 | Summary of EEG-based BCI biometrics research.

References	Application	No. of subject	EEG features	Classification algorithm	Performance evaluation
Pham et al. (2013)	Person authentication system	9	PSD	SVM	Multiple matched policy is highly secured
Bajwa and Dantu (2016)	Cancelable biometrics-based key generation	Data-1: 7, Data-2: 120	DFT, DWT	SVM	CA: Data-set-1: 98.46%; Data-set-2: 91.05%
Hu (2018)	EEG-based gender recognition	28	Fuzzy entropy	Vote classifier	Average accuracy: 99.8%
Nguyen et al. (2017)	Cryptographic key generation	125	PSD	Enroll and KeyGen	Success rate: 99%
Bashar et al. (2016)	Human identification	9	MSD, WPS, WPES	SVM	The highest accuracy: 94.44%
Ruiz-Blondet et al. (2016)	Biometric identification	50	Average ERP	SVM	Accuracy: 100%

method, which were in the time-frequency domain, and SVM was the classifier. A true positive rate of 94.44% in the aforesaid method was achieved using nine subjects in an experiment. Ruiz-Blondet et al. (2016) proposed an ERP-based highly accurate biometric recognition system designed to extract unique individual responses from the brain. The authors reportedly achieved 100% identification accuracy using 50 subjects. **Table 6** shows a summary of some BC-based biometrics and other related research.

BCI Emotion Recognition

Data acquisition from brain signals connected to human emotion is one of the core steps toward emotional intelligence. Picard stated that emotions play a vital role in rational decision-making, learning, and perception, as well as in a variety of functions (Picard, 2003). The identification of emotion changes from EEG signals has recently achieved attention among BCI researchers in the process of developing different BCI devices. There are three excellent review articles (Al-Nafjan et al., 2017; Soroush et al., 2018; Xu T. et al., 2018) on EEG-based emotion recognition. These articles might be of help to the interested readers to further have an insight of the EEG-based emotion recognition.

A number of emotion-recognition studies have been carried out by BCI researchers in the last 20 years. **Table 7** summarizes some recent emotion recognition approaches in terms of the number of subjects, stimulation, emotion types, feature extraction, classification method, and performance. Mu Li and Bao-Liang Lu (2009) utilized EEG signals for emotion recognition in response to emotional pictures. Their study gained a recognition rate of 93.5% for two emotional states. Petrantonakis and Hadjileontiadis (2010) presented a user-independent emotion recognition strategy that achieved an 83.33% recognition rate for six emotion categories. Wei et al. (2017) proposed a combination of features for achieving higher accuracy. The experimental outcomes showed that the combination of Power Spectral Density (PSD), Signal Power (SP), and Common Spatial Pattern (CSP) as the features achieved the highest accuracy of 86.83% with LDA as the classification algorithm, whereas, the accuracy of individual features was 64.73%. Chakladar and Chakraborty (2018a) introduced a correlation-based subset selection technique for dimensional reduction and used higher-order statistical features (mean, skewness, kurtosis, etc.) for classification. The authors classified four classes of emotion through employing an LDA algorithm, and an overall accuracy of 82% was attained. Moreover, Anh et al. (2012) proposed an emotion identification scheme to identify 2 valence classes and 2 arousal classes, which resulted in a combination of 4 fundamental emotions (happy, sorrowful, angry, and relaxed) and the neutral state. The authors affirmed the fractal dimension for feature selection and SVM as a classifier, where the average accuracy across all subjects was 70.5%. Liu Y. et al. (2017) designed a movie-induced feelings recognizer using EEG. This framework reached 92.26% accuracy in recognizing neutrality from high arousal with valence emotions and 86.63% to classify negative from positive emotions. The authors similarly classified 3 positive emotions and 4 negative emotions with 86.43 and 65.09% accuracy, respectively. Here, STFT was for feature extraction and SVM for classification. Another study reported the recognition of three classes of human emotion, namely sorrowful, excited, and relaxed, in real time using Wavelet and Learning Vector Quantization (LVQ) with an accuracy of 72–87% (Djamal and Lodaya, 2017). Moreover, a group of features, namely power, standard deviation, variance, and entropy, were classified by utilizing the k-NN algorithm. Happiness, anger, and calm were categorized in Kaur et al. (2018). Here, the fractal dimension feature was classified by utilizing RBF SVM with 60% accuracy. In another study, a human-vehicle collaborative driving (HVCD) simulation system was designed by integrating visual information and human intentions to achieve a comfortable and safe driving experience (Li et al., 2018). The average accuracy and ITR achieved were 91.1% and 85.80 bit/min, respectively.

BCI Virtual Reality and Gaming

This section has been inspired by some excellent review articles on BCI-based VR and games that are reported in Cattan et al. (2018), Ahn et al. (2014), and Kaplan et al. (2013). The research on BCI systems for healthy subjects has attracted

TABLE 7 | Summary of EEG-based BCI emotion recognition studies.

References	No. of subjects	Stimulation	Emotion types	EEG features	Classification algorithm	Performance evaluation
Wei et al. (2017)	12	Pictures	Positive and negative	PSD, SP, CSP	LDA	Average accuracy: 86.83%
Wang X.-W. et al. (2014)	6	Movie clips	Positive and negative	PSD	LDA	CA: 91.77%
Liu Y. et al. (2017)	30	Movie clips	Joy, anger, fear, sadness, disgust, and neutrality	PSD	LIBSVM	Average accuracy: 89.45%
Kaur et al. (2018)	10	Video clips	Calm, angry, and happy	FD	RBF SVM	CA: 60%
Özerdem and Polat (2017)	32	Music clips	Positive and negative	DWT	MLPNN	CA: 77.14%
Pan et al. (2016)	6	Photos	Happiness and sadness	CSP	SVM	CA: 74.17%
Djamal and Lodaya (2017)	10	Music	Excited, relaxed, and sad	WT	LVQ	CA: 87%
Murugappan (2011)	20	Video clips	Disgust, happiness, fear, surprise, and neutral	Entropy	K-NN	CA: 82.87%

TABLE 8 | Summary of EEG-based BCI gaming and VR studies.

References	No. of subjects	Control signal	EEG feature	Classification algorithm	Performance evaluation
Kreilinger et al. (2016)	10	MI	BP	LDA	"Upper 10%" MI detection rates: >70%
Bonnet et al. (2013)	10	MI	CSP	LDA	CA: >70%
Maby et al. (2012)	2	P300	Shannon entropy	LDA	Average accuracy: 82%
Djamal et al. (2017)	10	MI	FFT	LVQ	Average accuracy: 70%

considerable interest. The prototypes of BCI-based video games in existence are based on three BCI paradigms: steady-state evoked potential (SSVEP), P300 event-related potential (ERP), and mental imagery (MI). **Table 8** shows a summary of some recent EEG-based virtual reality and gaming systems. Finke et al. presented a P300-based BCI game known as Mind Game in which the user moves a character from one field to another on a game board (Finke et al., 2009). For Mind Game, the authors reported a 66% mean accuracy (specifically, this was the rate at which the correct target was selected out of 12 possible targets). Other P300-based BCI games have also been proposed (Mühl et al., 2010; Congedo et al., 2011; Angeloni et al., 2012; Ganin et al., 2013).

Some famous video games, including Pong, Pacman, and similar games, can be played with motor imagery (Krepki et al., 2007). The Pacman produces one step every 1.5–2 s with the aim of giving the gamer enough time to perform a control command. In another study, a pinball game was developed in order to illustrate that it is possible to successfully apply non-invasive recording techniques for complex control tasks (Krauledat et al., 2009).

Moreover, external evoked potentials have been utilized for game implementations. Middendorf et al. (2000) designed a simple flight simulator that is controlled by a BCI based on Steady-State Visual-Evoked Response (SSVER). This simulator was very modest, and only two control actions were possible. The position could only be moved to the left or right. Two methods were tested over the airplane control trials. On the other hand, the control command (right or left) was detected according to the strength of the SSVERs. The selection was identified by reference to the frequency of SSVER. The results of the trials with able-bodied participants showed that the last one was preferred because it required little or no training since the system capitalized on naturally occurring responses. Lalor et al. (2005) proposed a game named Mind Balance in which healthy subjects needed to hold a tightrope walker in balance. The application was based on SSVEP, which is generated as a response to phase-reversing checkerboard patterns.

A final interesting example of this application was recently presented at the Cybathlon 2016, a competition for participants with disabilities who compete against each other using assistive

TABLE 9 | Summary of EEG-based BCI robotic arm studies.

References	No. of subjects	Control signal	EEG feature	Classification algorithm	Performance evaluation
Yang C. et al. (2018)	2	SSVEP	FFT	CCA	Five tasks performed
Hortal et al. (2015)	2	MI	FFT	SVM	CA: 70%
Bhattacharyya et al. (2015)	11	MI	MFDFA	ANFIS	Success rate: >60%
Bousseta et al. (2018)	4	MI	FFT	RBF SVM	Success rate: 85.45%
Roy et al. (2016)	5	MI	WT, PSD	SVM	CA: 75.77%

technologies. In the BCI discipline, 11 participants with tetraplegia competed against each other in a virtual environment where their avatars raced along a virtual obstacle course (Novak et al., 2018). Since external visual stimuli were not allowed at the Cybathlon, the participants could not make use of SSVEPs and P300; instead, they relied on motor control and/or mental imagery to control their avatars. As expected, the results varied strongly between the 11 participants, with the best participant completing the race in the 90 s and the worst completing it in 196 s. In this competition, every team used gelled electrodes, indicating that they did not consider dry or water-based electrodes reliable enough for use in uncontrolled environments. Similarly, every team used laboratory-grade EEG amplifiers, suggesting that no team trusted consumer-grade devices to provide sufficiently good performance. Furthermore, the competition emphasized the importance of effective BCI training for the users, as the teams all had very different participant-training strategies and the winning team stated that their effective BCI training regimen likely had a major effect on their success.

BCI Robotic Arm

Because of the advances in robot control (RC) systems, they are playing an increasingly important role in a wide range of fields. The relationship between humans and robots has become increasingly intimate, and many human-robot collaboration systems have been developed. However, it is hard for a disabled person to operate a robot because of their loss of motion capacity or reduced sensing ability (Yang C. et al., 2018). Many studies have been dedicated to solving this problem (see **Table 9**). Yang C. et al. (2018) presented a shared control system by combining an SSVEP-based BCI and visual servoing (VS) technology to enable mind control of a robot manipulator. To enhance the intelligence and accuracy of the shared control system, the authors proposed an adaptive color adjustment for object detection, the least squares method (LSM) for camera calibration, and the coordination of task motion and self-motion (CTS) for obstacle avoidance. However, the authors tested the system using only two healthy subjects. Furthermore, Bousseta et al. (2018) proposed a preliminary outcome for the movement control of a robot arm for four directions: left, right, up, and down through mental tasks. Spectral analysis with FFT transform was combined with a PCA strategy to produce optimal features to feed into an RBF Kernel SVM classifier aimed at distinguishing the four movements. The experiments performed by four volunteers produced an average accuracy of 85.45%. A hybrid BCI system consisting of SSVEP and MI was proposed in Duan et al. (2015) to provide control commands to a brain-actuated robot. The SSVEP responses were utilized to tell the robot to move forward, turn left, and turn right, whereas the MI was utilized to perform the grasp motion. The authors also developed a visual servo module to perform the grasping with higher accuracy.

BCI Environmental Control

An important application of EEG-based BCIs is environmental control, which can improve the quality of life and increase the independence of paralyzed patients. Various EEG-based environmental control systems have been developed in recent years.

Aloise et al. developed a P300-based BCI home electronics control system that included a DVD player, electric lights, etc. (Aydin et al., 2018) where the subjects suffered from chronic neurological disorders. Shyu et al. (2010) proposed a steady-state visual evoked potential (SSVEP)-based BCI multimedia control system. Specifically, four flickering buttons were utilized to perform four different control commands: play or pause the selected multimedia file, scroll through displayed items, stop the multimedia system, and adjust the volume up and down. In another study, the authors proposed the application of an SSVEP-based BCI hospital bed nursing system (Kleber and Birbaumer, 2005) that enabled the users to move the entire bed up and down and control the power of the massaging cushion. In a study carried out by Corralejo et al. (2014), several patients with different degrees of motor impairment were able to operate home electronics, including a TV, DVD player, and lights, via a P300-based BCI. Edlinger and Guger (2012) suggested a non-invasive subject dependent P300 stimulus-based BCI system that provides the subject with a stimulus and records his reaction as an input. The user is provided with a GUI with various icons representing different tasks like turning a light on/off, opening a door/window, switching TV channels, etc. In the worst-case scenario, this system obtained 30% accuracy with 12 participants,

TABLE 10 | Summary of EEG-based BCI environmental control studies.

References	BCI application	Control signal	No. of subject	Method	Performance evaluation
Shyu et al. (2013)	Hospital bed nursing system	SSVEP	15	FPGA	Accuracy: 92.5%, ITR: 5.22 s/command
Zhang et al. (2017)	Environmental control system	ERP	3	Classifier: BLDA	Accuracy: 89.2%
Aydin et al. (2018)	Environmental control system	P300	10	Classifier: LDA	Accuracy: 93.71%
Kosmyna et al. (2016)	Control of a smart home with a BCI	P300, SSVEP	12	Minimum Distance Classifiers	Average accuracy: 81–77%

whereas, in the best-case scenario, a 100% accuracy was achieved from one of the subjects.

In another perspective, Carabalona et al. (2010) suggested a non-invasive visual P300-based BCI system for physically impaired users to control a smart home environment. The user is provided with a 6 × 6 matrix of icons. Each icon represents a command related to an everyday device. The icons flash on the computer screen one by one. Once the desired icon is reached, a peak is observed in the neural signals of the user. This peak is considered to be icon selection. The system was tested using four participants, and the accuracy rate observed varied from 33 to 100% among different users. Kim et al. (2013) also suggested a non-invasive P300 stimulus-based BCI system to switch TV channels from a viewing distance of 3 meters and a 46-inch TV screen. A total of eight subjects were provided with a visual stimulus in the form of a flashing green cursor in the top left corner of each channel icon. Once the desired channel was reached, a peak in the subject's neural signals was considered as input for channel selection. An average of 92.3% accuracy was attained from the system. In two other studies (Edlinger and Guger, 2012; Lin et al., 2014), the authors proposed non-invasive subject-dependent BCI to control electrical home appliances. Two physiological states (drowsiness or alertness) of users were used and translated into commands to interact with different electronic appliances. Similarly, Akman Aydin et al. (2015) suggested a region-based selection paradigm for a smart home control system for physically impaired users. The proposed system is a non-invasive P300 stimulus-based BCI system that flashes each region five times on the screen to invoke a response from the user. Upon acquiring a peak in the neural signals of the subject, the system considers it as a selection command for that particular region. The proposed system was able to achieve 95% accuracy for 49 household tasks using five subjects without any physical impairment. Masood et al. (2016) suggested a non-invasive BCI that uses the blink of an eye as a control input to interact with home appliances. The selection and control of the device are performed using a GUI. The system has the potential to be enhanced by adding more devices. However, the currently proposed system has only 70% accuracy. **Table 10** summarizes some recent findings with regard to EEG-based BCI environmental control.

Recent Achievements and Innovations in EEG-Based BCIs

In its early days, BCI technology was regarded as unattractive for genuine scientific investigation, primarily owing to its restricted resolution and the unreliability of the information from the brain as well as its high variability. Furthermore, the data acquisition of such technology was somewhat expensive. Therefore, brain activity research was typically constrained to medical use or exploration of brain functions in the laboratory. However, this scenario has changed dramatically over the past two decades due to advances in technology, which have brought down its associated costs to some degree. Meanwhile, BCI research has expanded to non-medical applications as well. Over the last 15 years, the number of studies regarding BCI has increased substantially. **Figure 4** depicts the number of companies that manufacture BCI-based products around the globe[2]. The two most well-known technology providers for BCI devices are Neurosky and Emotive.

The leading social media platform, Facebook, funded a research project at UCSF that aims to restore the communication ability of disabled people through their thoughts at a speed of 100 words per minute. Eventually, this company is planning to launch an EEG headset that lets users control music or interact in virtual reality using their thoughts. Some findings of this project were published in Moses et al. (2019), where the authors demonstrated that high-resolution recordings directly from the cortical surface can be used to decode both perceived and produced speech in real time. By integrating what participants hear and say, they leveraged an interactive question-and-answer behavioral paradigm that can be used in a real-world assistive communication setting. Together, these results represent an important step in the development of a clinically viable speech neuroprosthesis.

SpaceX and Tesla CEO Elon Musk founded Neuralink, which is aiming to design ultrafine threads (more tenuous than a human hair) that can be embedded into the brain to recognize neural activity. It builds arrays of small and flexible electrode "threads," with as many as 3,072 electrodes per array distributed across 96 threads Musk (2019). This company has also built

[2] *Companies - BNCI Horizon 2020* Available online at: at: http://bnci-horizon-2020.eu/community/companies (accessed February 02, 2019).

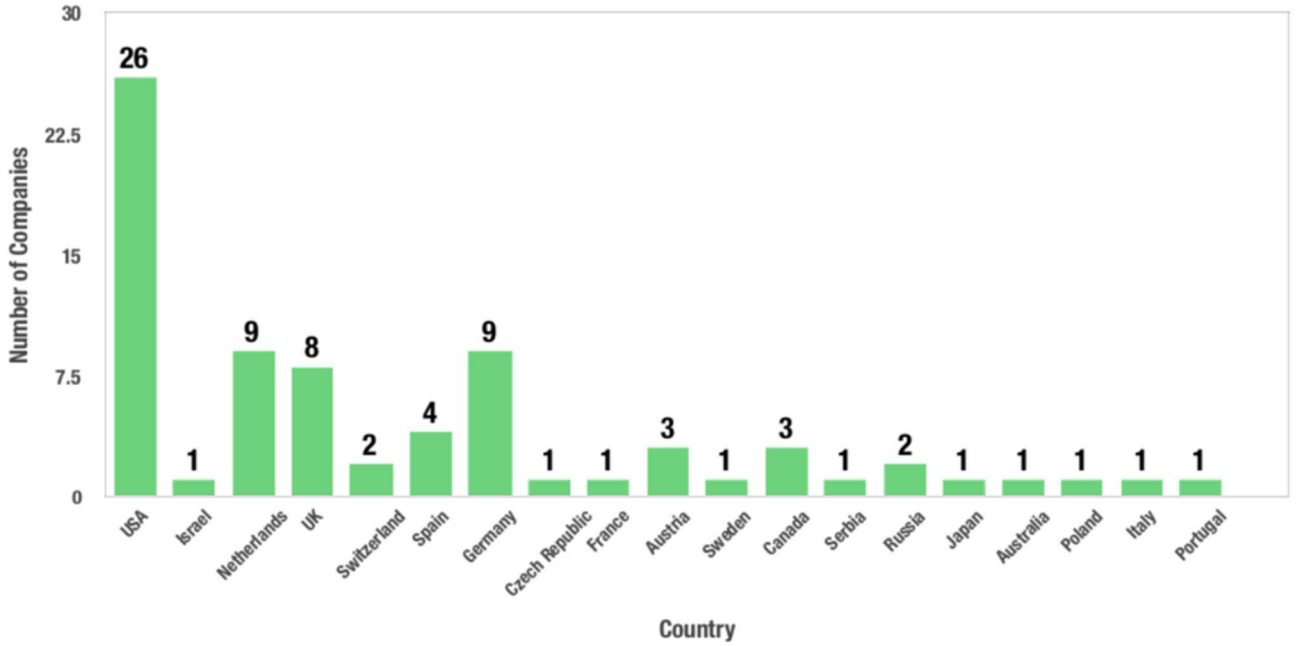


FIGURE 4 | Distribution of BCI companies around the world.

a neurosurgical robot that is capable of inserting six threads (192 electrodes) per minute. Each thread can be individually inserted into the brain with micron precision for the avoidance of surface vasculature and to target specific brain regions. The electrode array is packaged into a small implantable device that contains custom chips for low-power onboard amplification and digitization: the package for 3,072 channels occupies $<23 \times 18.5 \times 2$ mm^3. The size and composition of the thin-film probes are a better match for the material properties of brain tissue than commonly used silicon probes and may, therefore, exhibit enhanced biocompatibility (Chung et al., 2019). A single USB-C cable provides full-bandwidth data streaming from the device, recording from all channels simultaneously. This system has achieved a spiking yield of up to 70% in chronically implanted electrodes.

Jiang et al. (2019) have presented BrainNet, which is the first multi-person non-invasive direct brain-to-brain interface for collaborative problem-solving. The interface combines EEG to record brain signals and transcranial magnetic stimulation (TMS) to deliver information non-invasively to the brain. Among the three human subjects, two subjects are designated as "Senders" whose brain signals are decoded using real-time EEG data analysis. The decoding process extracts each Sender's decision about whether to rotate a block in a Tetris-like game before it is dropped to fill a line. The Senders' decisions are transmitted via the Internet to the brain of a third subject, the "Receiver," who cannot see the game screen. The Receiver integrates the information received from the two Senders and uses an EEG interface to make a decision about either turning the block or keeping it in the same orientation. Five groups, each with three human subjects, successfully used BrainNet to perform the collaborative task, with an average accuracy of 81.25%. This result points the way to future brain-to-brain interfaces that enable cooperative problem solving by humans using a "social network" of connected brains.

Boston area company Neurable has created a brain-controlled game called Awakening, the central character of which is a child with telekinetic powers. The character is set the task of escaping from a laboratory by using mind power to pick up toys such as a balloon dog and rainbow rings. Players wear a headband studded with electrodes that connect to a virtual reality headset. Their brain signals are picked up and analyzed by software that determines how the character will move. Players are then able to train their brains to produce the right signals to pick up the toys (iHuman, 2019).

Over the last two decades, evidence has accumulated on the capacity of neural interfaces such as those using transcranial direct current stimulation (tDCS) to enhance performance in cognitive areas such as working memory and attention as well as in physical activities such as cycling (Ke et al., 2019). tDCS involves the use of a headset and electrodes that are typically contained in sponge bags, with saline solution used to conduct electricity from the electrodes to the scalp (Cincotti et al., 2003). Adverse effects are rare—users have reported mild tingling sensations and occasionally headaches or fatigue. One study showed how tDCS improved the ability of US Air Force personnel to "multitask" (Nelson et al., 2016). The participants, all stationed at the Wright-Patterson Air Force Base in Ohio, USA, were asked to monitor and respond to four independent tasks on one computer screen. Their tasks were to: keep dial

markers centered in a "system monitoring" box; change the communications channel frequencies as requested by an audible prompt; keep a target centered in a "targeting" box, and keep fluids moving in a "resource management" box by turning tanks on and off. The 10 participants who received active tDCS stimulation from the headsets, provided by Wales-based company Magstim, performed about 30% better than those who did not.

CURRENT CHALLENGES AND DIRECTIONS

In the past few years, substantial BCI research has been performed to invent some potential assistive technologies. From the current status of BCIs, it can be predicted unquestionably that BCI technology will very soon be launched in the market. In fact, a few commercial BCI appliances have already been launched in the market. A remarkable program, namely the BNCI Horizon 2020 project, has proposed a future agenda regarding BCIs. However, some crucial issues and challenges exist in every component of the BCI paradigm, and these issues should be addressed by the BCI community to make further advances in BCIs.

Issues in EEG Modalities for BCI Applications

The most preferred EEG modalities, namely MI, SSVEP, and P300, are continuously facing signal processing issues, especially the identification of the most applicable approaches for feature extraction and feature reduction. This is primarily owing to the nature of the EEG signals, namely extremely non-linear, non-stationary, and artifact-prone. Other notable issues involve data fusion, in particular how the data from numerous electrodes are merged to be able to lessen the data dimensionality as well as to make improvements to the classification results. It is evident that once individuals have been taught properly, MI strategies often deliver remarkable outcomes. For an MI-based BCI to be handled by a targeted user, plenty of training trials are needed from the targeted user, causing the calibration period to be unacceptably long for a realistic model. Thus, investigations should be concentrated on cutting down the calibration period as well as effective training strategies. P300 shows greater average ITRs, and it does not call for a training process, although the degree of severity and variety of the disease may considerably affect the performance of this modality. Nevertheless, a good number of studies have also found that even individuals with LIS are capable of handling a P300-based BCI for long durations. However, with regards to ITR, it is worth noting that the healthy individual group yields higher bit rates than disabled subjects in practically all the previous findings regarding P300-BCI (Lazarou et al., 2018). The method of stimulation is so difficult that the experimental process could not be carried out by the patients. Moreover, a wide range of instructions in a P300-based BCI system increases the number of trials, which, in turn, causes reduced overall performance. The coupling of the generic models with online training could be an excellent alternative to reduce the calibration period and enhance P300-based BCI system performance together with consumer satisfaction (Jin et al., 2020). The SSVEP approach obviates the calibration or subject training. Hence, this type of speller should be faster in comparison with P300 spellers. However, a certain number of individuals generate extremely poor SSVEP responses, which is tough to explore. Therefore, a hybrid approach could be an excellent alternative to the use of a single EEG modality. Further investigations are necessary in order to identify which EEG modality is the most appropriate type for BCI application. For instance, MI-based approaches are considered the most appropriate to handle a prosthetic arm or leg. However, in the operation of digital radio, an SSVEP-based approach with a 4-option menu including volume-up, volume down, station-up, and station-down options may be much intuitive than an MI-based technique with imagined movements being associated with television controls. Intuitiveness should be a crucial factor at the time of the initial decision stage when selections of technology and BCI category are being made. Additional investigations are also needed to figure out the more appropriate feature set and classification frameworks for specific EEG modalities. Research with regards to features and classifiers also need emphasize figuring out the optimal options to be employed for individuals affected by CNS injury.

Issues With EEG Headsets

The quality of EEG data for BCI application mostly depends on the EEG headset. There are some issues regarding EEG headsets that need to be resolved. First of all, most of the EEG headsets require gel or liquid on electrodes, which are very uncomfortable to the user. For practical applications, users prefer dry electrodes, as it is not mandatory to use any conductive gel between the scalp skin and electrode pad. However, it is a matter of open debate as to whether this type of electrode offers identical signal properties. It has been reported that the EEG from dry electrodes contains considerably more artifacts and noise than wet electrodes, whereas another study reported that the signal properties are almost identical for wet and dry electrodes. Hence, an in-depth investigation should be carried out to further validate the efficacy of dry electrodes. It is worth noting that newly proposed water-based EEG electrodes are now being investigated. In Mihajlovi and Peuscher (2012), the authors pointed out that the performance of dry and water-based electrodes with shorter hair is comparably superior to gel-based electrodes. They also recommended continuing further investigation for the refinement of electrodes to make them applicable with longer hair and further suggested that dry and water-based electrodes have the potential to replace gel electrodes. To date, headsets with dry electrodes are available in the market, but these contain very few electrodes, and the procedure has not yet been standardized. There are thousands of EEG headsets that have already been launched by different companies for BCI applications. The number of electrodes (for example, 1, 3, 4, 8, 14, 16, 20, 24, 32, or 64) is varied from headset to headset, and these headsets are not compatible with each other. Hence, the minimum number of electrodes for specific EEG

modalities should be standardized. High ownership cost of an EEG headset is another challenge in allowing BCI technology to become affordable for the general public around the world (Abiyev et al., 2016). Hence, the reduction of the price of such EEG headsets is much lauded. The portability of the BCI model is also a crucial matter: headsets that transfer data wirelessly permit individuals to move freely, whereas a wired headset somewhat limits movements.

Lack of Ideal Data Analysis Methods

Although most of the artifact removal algorithms offer good performance, the methods listed in section EEG Data Preprocessing Strategies suffer from different limitations when utilized in a particular EEG-based application. Indeed, some methods are only focused on the detection and removal of particular artifacts. Some methods need reference channels to enhance the accuracy of artifact removal, which is not feasible for some specific applications. ICA-based algorithms can deal with all kinds of artifacts occurring in EEG recordings. Regression and adaptive filters are more feasible choices when the reference channels for specific artifacts are available. Apart from ICA, CCA and its combination with other methods seem to be a good choice for the removal of muscle artifacts. For application to a few channels, EMD, IVA, and its hybrid methods with BSS or WT can be an ideal choice. Moreover, EMD can significantly improve the signal quality by eliminating noise with fewer data (Zhang Z. et al., 2019). However, the requirement for a reference signal limits adaptive filter or regression methods to the removal of particular types of artifact. Wavelet transform fails to completely identify artifacts that overlap with spectral properties. EMD also suffers from the drawback of mode-mixing. Therefore, it is quite difficult to find a single method that is both efficient and accurate enough to satisfy all the conditions perfectly. Thus, one of the future objectives of the effective attenuation of artifacts is to develop an application-specific algorithm with better time-efficiency and accuracy. Also, from the current trend of artifact removal, it can be concluded that future directions will combine machine learning and traditional approaches for effective automatic artifact removal. Apart from that, new artifact removal algorithms for numerous types of artifacts in multiple scenarios still need to be identified.

Regarding feature extraction techniques, a wide range of features have been extracted to figure out the significant information from the EEG. From previous studies, it has been seen that the performance of CSP and its modified algorithms is comparatively encouraging when applied to EEG motor imagery data. In the case of emotion recognition or mental workload classification, spectral features are more suitable, whereas frequency-domain based features perform better with SSVEP data. However, it is too early to state the optimum feature extraction technique for a specific EEG control signal modality.

Based on previous studies, SVM is the most robust classifier for classifying high-dimensionality feature vectors. Sometimes, HMM may deal with a high-dimensionality feature set by analyzing the sequence of feature vectors. In order to avoid computational complexity, a low-dimensional feature vector should be preferred. However, feature reduction or selection algorithms could be considered in the case of high-dimensional feature sets. Several deep learning approaches have been implemented to classify MI, SSVEP, emotion recognition, and ERP data. The architecture of deep networks entirely depends on network structure and input formalization. Previous studies showed that CNN- and RNN-based deep learning approaches outperformed other deep learning methods. Moreover, CNN offered optimum accuracy when time-series values or spectral images were utilized as inputs. Hence, it has been suggested that there should be comprehensive study of the combination of network architectures, especially the structure and number of distinctive layers, for example, RBMs, convolutional layers, fully connected layers, and recurrent layers. Besides network structure, further studies need to be conducted to distinguish how deep learning approaches interpret raw EEG against artifact-prone EEG, as these sorts of studies have not yet been explicitly conducted. In this article, a wide range of classifiers have been surveyed that are evaluated in an offline manner. As every BCI application is essentially an online scenario, the classifiers should be validated online. Moreover, the classifier model should be tested to assure lower computational complexity and calibrated quickly in real-time operation. To design calibration-free BCIs, domain adaptation, and transfer learning can be an effective alternative where the combination of superior feature sets, for example, covariance matrices and domain adaptation algorithms, may enhance the invariance capability of BCIs. The design of a stable estimator of the Riemannian median is an open challenge. This property could make the Riemannian geometry classifiers more robust.

Performance Evaluation Metrics

It is evident from the previous studies that a variety of performance evaluation metrics are employed to evaluate BCI systems. It is almost impossible to compare the same types of BCI systems when they are evaluated by dissimilar performance metrics. Hence, the BCI research community should recommend a standard and systematic approach or a single metric to quantify a specific BCI application. For example, number of control commands, types of control commands, distance covered, time required, number of collisions, classification accuracy, average success rate, amongst others, should be utilized to evaluate the performance of a BCI wheelchair control. If the same performance metrics are used for a given BCI application, then a direct comparison between different BCI experiments is possible. Not only are training models not liable to limit the overall performance of the system, but testing data can also be utilized. A large number of BCIs are designed to substitute CNS functionality, and these BCIs are actually assessed on healthy individuals inside a controlled laboratory environment. This may misguide to defective outcomes if the targeted users of these systems would be disabled patients; for example, BCI performance seems to be poorer for patients affected by spinal cord injuries when compared with healthy subjects. Thus, researchers should emphasize specifying similar performance evaluation

metrics and, in the meantime, ensure system assessment by valid data.

Trends in Lab-Based BCI Technology

One of the major concerns regarding BCIs is that almost all of BCI experiments have been conducted in a controlled lab, regardless of the realistic environment of the targeted users. Therefore, these EEG data are a crucial factor in the initial evaluation of signal processing approaches as well as the advancement of considerably more robust systems. As pointed out before, heart rate and cortisol may considerably affect the characteristics of brain waves. Outside the lab, different sensory stimulations found in the surroundings, like sounds, movements, and smells, may affect the quality of EEG signals. Hence, during the design of any BCI system, engineers should consider the particular environment where the proposed technology will be employed. For example, the design criteria may not be identical for the operation of home appliances in a home environment as compared to the detection of the attention level of a pilot using EEG while flying a plane. Thus, at the time of system design, it is important to examine the basic criteria of the system, environmental aspects, situation, and target users in-depth.

Low ITR of BCI Systems

Higher ITR is the primary requirement of any effective BCI system. ITR is the metric that is most employed in the BCI community to assess the performance of BCI prototypes. The ITR of a given BCI system depends upon three criteria: the number of classes, target detection accuracy, and target detection time (Wolpaw et al., 2002; Ramadan and Vasilakos, 2017). The target detection accuracy can be boosted with the aid of the enhancement of the Signal-to-Noise Ratio (SNR). Several strategies are usually employed in the preprocessing phase in order to increase the SNR, as described in section EEG Control Signals Used in BCI Applications. Once high ITR is achieved, more sophisticated applications may be developed by increasing the variety of classes. A variety of stimulus coding strategies, for instance, CDMA, TDMA, and FDMA, have already been adapted for BCI systems (Bin et al., 2011; Jin et al., 2011). In order to code the intended stimuli, TDMA has been utilized with P300, whereas CDMA and FDMA have been employed in those BCIs that deal with VEP. Finally, minimizing the target recognition period, which assists in boosting the ITR, is another crucial aspect of BCIs. To accomplish this, adaptive approaches, namely "dynamic stopping," may be an excellent alternative. Moreover, single-trail classifications using machine learning frameworks and optimized stimulus demonstrations can also contribute to reducing the target recognition period (Schreuder et al., 2013).

Commercialization of EEG-Based BCI Technology

The main consumer of BCI technology is this particular group of disabled patients, and commercialization is the only way to spread this technology all over the world. Hence, the BCI research community should identify the actual causes that prevent the commercialization of this technology. To further illustrate the obstacles that are preventing the commercialization of BCI technology, Vansteensel et al. (2017) supplied a set of questions to more than 3,500 BCI researchers in the field of EEG or EMG technologies throughout the world. Almost all of the experts were confident that a particular BCI application would be launched to the market within the next 5–10 years. This survey suggests that upcoming research ought to concentrate on sensor development as well as the overall system performance of non-invasive BCIs. Kristo et al. (2013) also concluded that the BCI research community should focus on boosting ITR (Speier et al., 2013) and exploring the signal processing platform. In addition to these, there are also some major issues that prevent the commercialization of an EEG-based BCI system among the targeted users. First of all, existing BCIs cannot be handled by disabled patients. Expert assistance is mandatory to set up the signal-receiving electrodes of current BCIs. Moreover, almost all of the BCI devices are still under investigation and, hence, are not readily available for home-usage. Therefore, future BCI systems should cater for disabled patients lacking any assistance from experts. BCI illiteracy is an additional barrier to the widespread use of EEG-based BCI systems. When an individual is unable to operate the BCI device due to low-quality brain signals being produced, this phenomenon is regarded as BCI illiteracy. EEG signal quality may be improved with the aid of a collaborative co-learning strategy that gives the end-user audio or visual feedback. Continuous use of BCIs makes frequent use of specific neural pathways (Padfield et al., 2019), and additional study needs to figure out the possible health risks or variations in brain functionality due to such prolonged exposure. The BCI system should be similar to other ideal off-the-shelf technologies with regards to its ease-of-use. The BCI devices should also be user-friendly and have in-built safeguards to prevent untoward situations. Furthermore, advanced BCI technology should also be capable of providing stable results when it is used in multisensory environments such as in a noisy family home. **Table 11** lists the essential features that enable the commercialization of BCI systems. These features are deemed

TABLE 11 | Essential features for BCI systems.

Key feature	Description
Effectiveness	The BCI system should be really helpful to users
Robustness	The system must be stable during regular use and robust with respect to anomalies
Quick operation	Task execution time should be as low as possible
More functionality	The system should allow the user to perform as many tasks as possible to increase the autonomy of the user
Safety	The system must pose no danger to user health
Comfort	The EEG cap should be comfortable to wear for several hours
Mobility	To ensure mobility, the BCI system should be wireless, lightweight, and compact
User-friendliness	The system should be simple to operate and need no expert help for daily use
Cost-effectiveness	The price should be affordable for all kinds of users

necessary to increase the quality of BCI systems (Miralles et al., 2015).

Issues in Specific EEG-Based BCI Applications

The use of P300 in a BCI wheelchair has numerous drawbacks, for example, when a patient suffers from a neural disease, specifically ALS (Bashashati et al., 2007a; Kodi et al., 2013). Usually, BCIs that use a P300 modality have a poor ITR. Additionally, several studies have stated in their findings that performance may drop after a long period owing to the reduction of the P300 amplitude due to familiarization (Iturrate et al., 2009a; Choi, 2012; Amiri et al., 2013). In a real-time system, the user needs to concentrate on the mission without interruption (Iturrate et al., 2009b), which is another weakness of such a scheme. If users focus on such visual stimuli for a significant period, they feel fatigued or suffer from sore eyes (Puanhvuan and Wongsawat, 2012; Chai et al., 2014). Therefore, such physiological control signals are neither suitable nor effective for wheelchair operation. Despite the advantages of using navigation systems that assist the control of the wheelchair with shared control, limited findings have been reported. However, it is worth noting that several cases, for instance, wheelchair operation in a corridor or a free area with unfamiliar impediments, need to be addressed. In such cases, a complete wheelchair control system with its main navigation components including mapping, localization, path planning, and obstacle avoidance is highly recommended (Widyotriatmo and Suprijanto, 2015).

With regards to BCI cursor control, only three articles have reported a success rate of above 90% (Long et al., 2012b). This information somewhat suggests that an insufficient number of experiments have yet been carried out in this area. Therefore, more experiments are needed to explore the exact limitation behind the low success rate. Although MI, P300, and hybrid approaches (MI+P300, MI+SSVEP) have been used in BCI cursor control, it was observed that the hybrid approach yields the best results (Long et al., 2012b). However, it is too early to remark on which hybrid strategy is optimal for multidimensional cursor control, and, hence, more experiments are needed to evaluate efficacy. Different factors should be taken into consideration to make a robust system, such as literacy rate, target size, timeout interval, user preferences, volume of electrodes, training period, invasiveness, and movement time.

Most of the previous studies of BCI spellers have been carried out utilizing the P300 modality owing to its credibility among researchers. Hence, there are more development opportunities in the other BCI paradigms, for example, MI, SSVEP, and hybrid. Almost all of the studies employed mean accuracy and ITR to evaluate their experiments. Nonetheless, it is noteworthy to mention that amongst a selected paradigm, the performance of one study varies from another owing to the utilization of different experimental resources. The aforesaid resources include EEG caps, electrodes, GUI, software, and data from healthy or disabled subjects, as well as the number of subjects. Hence, for conducting a comparison between different BCI spellers, the first step should be that the identical hardware and software must be used. Hitherto, to operate a BCI speller, users require expert help to set up electrodes at specific positions. Portability is another challenge to the current BCI spellers. Typing errors are another issue, as it necessitates a correction period and influences the spelling rate. Hence, further enhancements should be made to provide faster, accurate, and user-friendly spellers.

Some constituents, including data preprocessing strategy, classification model design, have a considerable impact on the performance of BCI-based person identification. Although a variety of features and classifiers have been employed to figure out the superior approach, the optimum technique has not been explored. Hence, more experiments should be carried out to identify the most appropriate technique. Moreover, more attention should be given to removing artifacts from EMG, EOG, and ECG. In an EEG-based authentication system, different paradigms like P300, SSVEP, and MI can be used, although each paradigm has its own merits and demerits. Thus, the best paradigm for EEG should be identified based on the person's authentication. Moreover, the EEG acquisition protocols should be user-friendly. Brain activity acquisition for biometric usage is a crucial matter that has not received enough attention from researchers (Campisi and La Rocca, 2014). Few partial studies have been conducted, and these studies emphasized session-wise EEG stability. Moreover, in these studies, the variation in data acquisition time lay between 1 week and 5 months (La Rocca et al., 2013; Lee et al., 2013; Palaniappan and Revett, 2014). Additionally, the human brain is very sensitive to emotion (Marcel and Millan, 2007; Lee et al., 2013). No publication has clarified the stability of EEG regarding the emotional diversity of BCI-based biometrics. Hence, the emotional diversity of the human mind should be carefully considered for a realistic BCI biometric system. Moreover, permanence in terms of elicitation strategy, applied protocol, and feature selection should also be investigated critically.

Researchers have attempted to recognize human emotion or mental state using EEG. However, there are still many challenges in this particular research domain. The EEG signal is very weak and is easily disturbed by external factors, such as subjects' movements and environment noise. Noise-free and accurate EEG is the greatest challenge in recognizing emotional states. The way in which emotions are evoked contributes significantly in emotion recognition systems. There are numerous methods of stimulation, i.e., pictures, video clips, music, memories, self-induction, environmental elicitation like light, humidity, and temperature, and games, amongst others. By using good and strong stimulation, emotion recognition is more likely to be performed with better results and higher accuracy. A number of emotion types also sometimes affect the system's performance. Another crucial matter that significantly influences the performance of the system is the signal processing platform, which includes data denoising, filtering, suitable feature set selection, and classification model design. A variety of feature extraction and classification algorithms have been

carried out by the researchers, but it is still difficult to identify which model is optimal. However, PSD and SVM have been most widely used for feature extraction and classification, respectively.

Most BCI games demonstrate very low accuracy and speed as compared to conventional interfaces, suggesting that there are issues that must be addressed to facilitate the acceptance of BCI games. One of the most prevalent issues is the selection of the EEG control signal. Among SSVEP, P300, and MI, the P300 paradigm is favored by researchers for BCI-based games. With regards to tagging problems and hardware impediments, an optimal setup should be employed, for instance, an EEG headset, VR machine, and motion sensor can be integrated, and this joint system may able to trace the rotation together with the position of the targeted users (De Vos et al., 2014). A number of notable BCI-based games have been recommended in Marshall et al. (2013). Although the P300 modality has been employed to good effect in puzzle games, this system may be upgraded by motion-based technology. A turn-based strategy allows the users to pick options at their own pace, whereas a variety of simulation and adventure games are more relevant for a BCI+VR game. The performance of real-time BCI with moving users is a controversial issue. When the users are walking or moving, the P300 modality can be stimulated, but the overall performance drops (Debener et al., 2012). A camera and accelerators can be used for identifying user movement as well as removing the corresponding EEG signal from the analysis (Ahn et al., 2014). To ensure noise-free EEG, a real-time artifact elimination strategy has been recommended (Barachant et al., 2013). Additionally, some novel features, including the weighted phase lag index, can be employed when the subjects are walking (Lau et al., 2012). However, we have noticed that there is still a lack of studies outside of a controlled environment, such as Debener et al. (2012).

A limited number of studies have reported on the use of BCI in robotic arm control, suggesting that there is more room for exploration. Moreover, it is apparent that the performance of these reported systems is still at an unsatisfactory level, indicating that more experiments should be carried out. Due to a lack of necessary adaptation and training, the time taken for such activity to transpire is somewhat lengthened; nonetheless, this problem can be mitigated through proper training. Some BCI-based home appliances may able to handle only one application a time, which does not reflect effectiveness in real-life scenarios. Hence, the activities of daily living (ADL) of paralyzed patients will be easier when many home appliances at a time can be utilized through an independent system. Most of the proposed environmental control systems have been assessed on healthy users. Thus, there is a crucial question as to whether the performance of healthy users is identical to that of disabled people. Until now, the possibility of the use of environmental control systems by severely disabled people has been investigated in very few studies, and, hence, more effort should be applied to evaluating data obtained from disabled patients.

CONCLUSION

A thorough review analysis has been carried out in this article on EEG-based BCI, particularly to investigate its methodological advantages and disadvantages and the essential contributions required in this field of research. Each BCI application has been explored in terms of data acquisition technique, control signals, EEG feature extraction, classification methods, and performance evaluation metrics. Finally, potential complications with EEG-based BCI systems have been addressed, and promising alternative options have been recommended. Patients with CNS injury may be able to rehabilitate their motor function through the progress of emerging BCIs. Owing to wireless recording, portability of EEG headsets, cost-effective amplifiers, significant temporal resolution, and proficient signal analysis strategies, there is keen attention on EEG-based BCI technology for such purposes. In spite of the many outstanding breakthroughs that have been achieved in BCI research, some issues still need to be resolved. First of all, a general BCI standard is currently the main issue. The BCI community should declare a general BCI standard that must be adhered to by BCI researchers. Secondly, the existing BCIs offer somewhat poor ITR for any type of effectual BCI application. Hence, future research should concentrate on increasing the ITR of BCI systems. Moreover, matching the most relevant EEG control signal with the intended BCI application is another important issue in EEG-based BCI research. Owing to the present accessibility of computational resources, researchers have begun to move away from conventional machine learning models to deep learning approaches. The use of such contemporary techniques allows for the classification of non-stationary EEG. Most of the studies on BCI have used different evaluation metrics on their own as per their convenience without any uniformity, which makes it difficult to choose the most efficient method, especially for new researchers in this field. Hence, it could be of interest to establish standardized BCI metrics so that all related works may follow the standard. These metrics would be formed based on BCI application or EEG modalities. Through this, a fair comparison could be made between related works accurately and conveniently, which would help new researchers to pursue their intended research direction. Finally, the majority of BCI applications are at the investigative phase, and they are not readily available for the ADL of the general populace. In addition, the lack of commercialization of BCI technology is also partly responsible for impeding its popularity around the world. If the abovementioned concerns can be addressed, BCI systems could be an emerging means of human-machine interaction in the foreseeable near-future.

AUTHOR CONTRIBUTIONS

MR and NS: conceptualization. MR and BB: methodology. MR: investigation and writing—original draft preparation. MR, AA, and BB: resources. MR, AP, RM, BB, and SK: writing—review and editing. NS: supervision and project administration. NS, AP, AA, RM, and SK: funding acquisition.

FUNDING

This work was supported by the Universiti Malaysia Pahang, Malaysia, through research grant FRGS/1/2018/TK04/UMP/02/3 (RDU190109).

ACKNOWLEDGMENTS

The authors would like to acknowledge support from the Faculty of Electrical & Electronics Engineering Technology, Universiti Malaysia Pahang, Malaysia.

REFERENCES

Abbas, Q., Ibrahim, M. E. A., and Jaffar, M. A. (2019). A comprehensive review of recent advances on deep vision systems. *Artif. Intell. Rev.* 52, 39–76. doi: 10.1007/s10462-018-9633-3

Abdulkader, S. N., Atia, A., and Mostafa, M. S. M. (2015). Brain computer interfacing: applications and challenges. *Egypt. Informatics J.* 16, 213–230. doi: 10.1016/j.eij.2015.06.002

Abiri, R., Borhani, S., Sellers, E. W., Jiang, Y., and Zhao, X. (2019). A comprehensive review of EEG-based brain–computer interface paradigms. *J. Neural Eng.* 16:011001. doi: 10.1088/1741-2552/aaf12e

Abiyev, R. H., Akkaya, N., Aytac, E., Günsel, I., and Çagman, A. (2016). Brain-computer interface for control of wheelchair using fuzzy neural networks. *Biomed Res. Int.* 2016:9359868. doi: 10.1155/2016/9359868

Acqualagnav, L., Treder, M. S., Schreuder, M., and Blankertz, B. (2010). "A novel brain-computer interface based on the rapid serial visual presentation paradigm," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology* (Buenos Aires: IEEE), 2686–2689. doi: 10.1109/IEMBS.2010.5626548

Ahn, M., Lee, M., Choi, J., and Jun, S. (2014). A review of brain-computer interface games and an opinion survey from researchers, developers and users. *Sensors* 14, 14601–14633. doi: 10.3390/s140814601

Akman Aydin, E., Bay, O. F., and Guler, I. (2015). "Region based Brain Computer Interface for a home control application," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Milan: IEEE), 1075–1078. doi: 10.1109/EMBC.2015.7318551

Al Ghayab, H. R., Li, Y., Siuly, S., and Abdulla, S. (2019). A feature extraction technique based on tunable Q-factor wavelet transform for brain signal classification. *J. Neurosci. Methods* 312, 43–52. doi: 10.1016/j.jneumeth.2018.11.014

Alariki, A. A., Ibrahimi, A. W., Wardak, M., and Wall, J. (2018). A review study of brian activity-based biometric authentication. *J. Comput. Sci.* 14, 173–181. doi: 10.3844/jcssp.2018.173.181

Alexander, C., Yongtian, H., and Jose, L., C.-V. (2018). Deep learning for electroencephalogram (EEG) classification tasks: a review. *J. Neural Eng.* 16:031001. doi: 10.1016/S0165-0114(98)00202-4

Allison, B. Z., Brunner, C., Altstätter, C., Wagner, I. C., Grissmann, S., and Neuper, C. (2012). A hybrid ERD/SSVEP BCI for continuous simultaneous two dimensional cursor control. *J. Neurosci. Methods* 209, 299–307. doi: 10.1016/j.jneumeth.2012.06.022

Al-Nafjan, A., Hosny, M., Al-Ohali, Y., Al-Wabil, A., Al-Nafjan, A., Hosny, M., et al. (2017). Review and classification of emotion recognition based on EEG brain-computer interface system research: a systematic review. *Appl. Sci.* 7:1239. doi: 10.3390/app7121239

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., et al. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics* 8:292. doi: 10.3390/electronics8030292

Alqasemi, R., and Dubey, R. (2010). "A 9-DoF wheelchair-mounted robotic arm system: design, control, brain-computer interfacing, and testing," in *Advances in Robot Manipulators* (InTech). doi: 10.5772/9678

Al-qaysi, Z. T., Zaidan, B. B., Zaidan, A. A., and Suzani, M. S. (2018). A review of disability EEG based wheelchair control system: coherent taxonomy, open challenges and recommendations. *Comput. Methods Progr. Biomed.* 164, 221–237. doi: 10.1016/J.CMPB.2018.06.012

Amber, F., Yousaf, A., Imran, M., and Khurshid, K. (2019). "P300 based deception detection using convolutional neural network," in *2019 2nd International Conference on Communication, Computing and Digital Systems (C-CODE)* (Islamabad: IEEE), 201–204. doi: 10.1109/C-CODE.2019.8681025

Amin, S. U., Alsulaiman, M., Muhammad, G., Bencherif, M. A., and Hossain, M. S. (2019). Multilevel weighted feature fusion using convolutional neural networks for EEG motor imagery classification. *IEEE Access* 7, 18940–18950. doi: 10.1109/ACCESS.2019.2895688

Amiri, S., Fazel-Rezai, R., and Asadpour, V. (2013). A review of hybrid brain-computer interface systems. *Adv. Human-Computer Interact.*2013:187024. doi: 10.1155/2013/187024

An, X., Kuang, D., Guo, X., Zhao, Y., and He, L. (2014). A deep learning method for classification of EEG data based on motor imagery," in *Intelligent Computing in Bioinformatics. ICIC 2014. Lecture Notes in Computer Science, Vol. 8590,* D. S. Huang, K. Han, and M. Gromiha (Cham: Springer), 203–210. doi: 10.1007/978-3-319-09330-7_25

Angeloni, C., Salter, D., Corbit, V., Lorence, T., Yu, Y.-C., and Gabel, L. A. (2012). "P300-based brain-computer interface memory game to improve motivation and performance," in *2012 38th Annual Northeast Bioengineering Conference (NEBEC)* (Philadelphia, PA: IEEE), 35–36. doi: 10.1109/NEBC.2012.6206949

Anh, V. H., Van, M. N., Ha, B. B., and Quyet, T. H. (2012)." A real-time model based Support Vector Machine for emotion recognition through EEG," in *2012 International Conference on Control, Automation and Information Sciences (ICCAIS)* (Ho Chi Minh City: IEEE), 191–196. doi: 10.1109/ICCAIS.2012.6466585

Ansari, I. A., and Singla, R. (2016). BCI: an optimised speller using SSVEP. *Int. J. Biomed. Eng. Technol.* 22:31. doi: 10.1504/IJBET.2016.078988

Arabnia, H. R., and Tran, Q.-N. (eds.). (2011). *Software Tools and Algorithms for Biological Systems.* New York, NY: Springer. doi: 10.1007/978-1-4419-7046-6

Asadur Rahman, M., Foisal Hossain, M., Hossain, M., and Ahmmed, R. (2019). Employing PCA and t-statistical approach for feature extraction and classification of emotion from multichannel EEG signal. *Egypt. Inform. J.* 21, 23-45. doi: 10.1016/j.eij.2019.10.002

Ashok, S. (2017). High-level hands-free control of wheelchair–a review. *J. Med. Eng. Technol.* 41, 46-64. doi: 10.1080/03091902.2016.1210685

Attia, M., Hettiarachchi, I., Hossny, M., and Nahavandi, S. (2018). "A time domain classification of steady-state visual evoked potentials using deep recurrent-convolutional neural networks," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)* (Washington, DC: IEEE), 766–769. doi: 10.1109/ISBI.2018.8363685

Atyabi, A., Shic, F., and Naples, A. (2016). Mixture of autoregressive modeling orders and its implication on single trial EEG classification. *Expert Syst. Appl.* 65, 164–180. doi: 10.1016/J.ESWA.2016.08.044

Aydin, E. A., Bay, O. F., and Guler, I. (2018). P300-based asynchronous brain computer interface for environmental control system. *IEEE J. Biomed. Heal. Inform.* 22, 653–663. doi: 10.1109/JBHI.2017.2690801

AydIn, S., Saraoglu, H. M., and Kara, S. (2009). Log energy entropy-Based EEG classification with multilayer neural networks in seizure. *Ann. Biomed. Eng.* 37, 2626–2630. doi: 10.1007/s10439-009-9795-x

Aznan, N. K. N., Bonner, S., Connolly, J. D., Al Moubayed, N., and Breckon, T. P. (2018). "On the classification of SSVEP-based dry-EEG signals via convolutional neural networks," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC2018)* (Miyazaki; Piscataway, NJ: IEEE), 3726–3731. doi: 10.1109/SMC.2018.00631

Bablani, A., Edla, D. R., and Kuppili, V. (2018). "Deceit identification test on EEG data using deep belief network," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (Bangalore: IEEE), 1–6. doi: 10.1109/ICCCNT.2018.8494124

Baillet, S., Mosher, J. C., and Leahy, R. M. (2001). Electromagnetic brain mapping. *IEEE Signal Process. Mag.* 18, 14–30. doi: 10.1109/79.962275

Bajwa, G., and Dantu, R. (2016). Neurokey: towards a new paradigm of cancelable biometrics-based key generation using electroencephalograms. *Comput. Secur.* 62, 95–113. doi: 10.1016/J.COSE.2016.06.001

Banville, H., and Falk, T. H. (2016). Recent advances and open challenges in hybrid brain-computer interfacing: a technological review of non-invasive human research. *Brain Comput. Interf.* 3, 9–46. doi: 10.1080/2326263X.2015.1134958

Barachant, A., Andreev, A., Congedo, M., Barachant, A., Andreev, A., and Congedo, M. (2013). "The riemannian potato: an automatic and adaptive artifact detection method for online experiments using Riemannian geometry," in *TOBI Workshop lV* (Sion), 19–20.

Bascil, M. S., Tesneli, A. Y., and Temurtas, F. (2016). Spectral feature extraction of EEG signals and pattern recognition during mental tasks of 2-D cursor movements for BCI using SVM and ANN. *Australas. Phys. Eng. Sci. Med.* 39, 665–676. doi: 10.1007/s13246-016-0462-x

Bashar, M. K., Chiaki, I., and Yoshida, H. (2016). "Human identification from brain EEG signals using advanced machine learning method EEG-based biometrics," in *2016 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)* (Kuala Lumpur: IEEE), 475–479. doi: 10.1109/IECBES.2016.7843496

Bashashati, A., Fatourechi, M., Ward, R. K., and Birch, G. E. (2007a). A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals. *J. Neural Eng.* 4, R32–R57. doi: 10.1088/1741-2560/4/2/R03

Bashashati, A., Ward, R. K., and Birch, G. E. (2007b). Towards development of a 3-state self-paced brain-computer interface. *Comput. Intell. Neurosci.* 2007:84386. doi: 10.1155/2007/84386

Batres-Mendoza, P., Montoro-Sanjose, C. R., Guerra-Hernandez, E. I., Almanza-Ojeda, D. L., Rostro-Gonzalez, H., Romero-Troncoso, R. J., et al. (2016). Quaternion-based signal analysis for motor imagery classification from electroencephalographic signals. *Sensors* 16:336. doi: 10.3390/s16030336

Bhattacharyya, S., Basu, D., Konar, A., and Tibarewala, D. N. (2015). Interval type-2 fuzzy logic based multiclass ANFIS algorithm for real-time EEG based movement control of a robot arm. *Rob. Auton. Syst.* 68, 104–115. doi: 10.1016/J.ROBOT.2015.01.007

Bi, L., Fan, X.-A., and Liu, Y. (2013). EEG-based brain-controlled mobile robots: a Survey. *IEEE Trans. Hum. Mach. Syst.* 43, 161–176. doi: 10.1109/TSMCC.2012.2219046

Bin, G., Gao, X., Wang, Y., Li, Y., Hong, B., and Gao, S. (2011). A high-speed BCI based on code modulation VEP. *J. Neural Eng.* 8:025015. doi: 10.1088/1741-2560/8/2/025015

Birbaumer, N. (2006). Breaking the silence: brain?computer interfaces (BCI) for communication and motor control. *Psychophysiology* 43, 517–532. doi: 10.1111/j.1469-8986.2006.00456.x

Blankertz, B., Dornhege, G., Krauledat, M., Schröder, M., Williamson, J., Murray-Smith, R., et al. (2006). "The berlin brain-computer interface presents the novel mental typewriter Hex-O-Spell," in *Proceedings of the 3rd International Brain-Computer Interface Workshop and Training Course* (Graz).

Bong, S. Z., Wan, K., Murugappan, M., Ibrahim, N. M., Rajamanickam, Y., and Mohamad, K. (2017). Implementation of wavelet packet transform and non linear analysis for emotion classification in stroke patient using brain signals. *Biomed. Signal Process. Control* 36, 102–112. doi: 10.1016/J.BSPC.2017.03.016

Bonnet, L., Lotte, F., and Lécuyer, A. (2013). Two brains, one game: design and evaluation of a multiuser bci video game based on motor imagery. *IEEE Trans. Comput. Intell. AI Games* 5, 185–198. doi: 10.1109/TCIAIG.2012.2237173

Borisoff, J. F., Mason, S. G., Bashashati, A., and Birch, G. E. (2004). Brain–computer interface design for asynchronous control applications: improvements to the LF-ASD asynchronous brain switch. *IEEE Trans. Biomed. Eng.* 51, 985–992. doi: 10.1109/TBME.2004.827078

Bousseta, R., El Ouakouak, I., Gharbi, M., and Regragui, F. (2018). EEG based brain computer interface for controlling a robot arm movement through thought. *IRBM* 39, 129–135. doi: 10.1016/J.IRBM.2018.02.001

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* 2, 121–167. doi: 10.1023/A:1009715923555

Campisi, P., and La Rocca, D. (2014). Brain waves for automatic biometric-based user recognition. *IEEE Trans. Inf. Forensics Secur.* 9, 782–800. doi: 10.1109/TIFS.2014.2308640

Cao, L., Li, J., Ji, H., and Jiang, C. (2014). A hybrid brain computer interface system based on the neurophysiological protocol and brain-actuated switch for wheelchair control. *J. Neurosci. Methods.* 229, 33–43.doi: 10.1016/j.jneumeth.2014.03.011

Cao, L., Xia, B., Maysam, O., Li, J., Xie, H., and Birbaumer, N. (2017). A synchronous motor imagery based neural physiological paradigm for brain computer interface speller. *Front. Hum. Neurosci.* 11:274. doi: 10.3389/fnhum.2017.00274

Cao, T., Wang, X., Wang, B., Wong, C. M., Wan, F., Mak, P. U., et al. (2011). "A high rate online SSVEP based brain-computer interface speller," in *2011 5th International IEEE/EMBS Conference on Neural Engineering* (Cancun: IEEE), 465–468. doi: 10.1109/NER.2011.5910587

Carabalona, R., Grossi, F., Tessadri, A., Caracciolo, A., Castiglioni, P., and Don Carlo Gnocchi Onlus, I. D. M. F. (2010). "Home smart home: brain-computer interface control for real smart home environments," in *International Convention on Rehabilitation Engineering and Assistive Technology* (Shanghai).

Carlson, T., and Millan, J. D. R. (2013). Brain-controlled wheelchairs: a robotic architecture. *IEEE Robot. Autom. Mag.* 20, 65–73. doi: 10.1109/MRA.2012.2229936

Cattan, G., Mendoza, C., Andreev, A., and Congedo, M. (2018). Recommendations for integrating a P300-based brain computer interface in virtual reality environments for gaming. *Computers* 7:34. doi: 10.3390/computers7020034

Chai, R., Ling, S. H., Hunter, G. P., Tran, Y., and Nguyen, H. T. (2014). Brain-computer interface classifier for wheelchair commands using neural network with fuzzy particle swarm optimization. *IEEE J. Biomed. Heal. Informatics*. 18, 1614-1624. doi: 10.1109/JBHI.2013.2295006

Chai, R., Ling, S. H., San, P. P., Naik, G. R., Nguyen, T. N., Tran, Y., et al. (2017a). Improving EEG-based driver fatigue classification using sparse-deep belief networks. *Front. Neurosci.* 11:103. doi: 10.3389/fnins.2017.00103

Chai, R., Naik, G. R., Nguyen, T. N., Ling, S. H., Tran, Y., Craig, A., et al. (2017b). Driver fatigue classification with independent component by entropy rate bound minimization analysis in an EEG-based system. *IEEE J. Biomed. Heal. Informatics* 21, 715–724. doi: 10.1109/JBHI.2016.2532354

Chakladar, D. D., and Chakraborty, S. (2018a). EEG based emotion classification using "correlation based subset selection." *Biol. Inspired Cogn. Archit.* 24, 98–106. doi: 10.1016/j.bica.2018.04.012

Chakladar, D. D., and Chakraborty, S. (2018b). Multi-target way of cursor movement in brain computer interface using unsupervised learning. *Biol. Inspired Cogn. Archit.* 25, 88–100. doi: 10.1016/J.BICA.2018.06.001

Chang, M. H., Lee, J. S., Heo, J., and Park, K. S. (2016). Eliciting dual-frequency SSVEP using a hybrid SSVEP-P300 BCI. *J. Neurosci. Methods* 258, 104–113. doi: 10.1016/j.jneumeth.2015.11.001

Chaudhary, S., Taran, S., Bajaj, V., and Sengur, A. (2019). Convolutional neural network based approach towards motor imagery tasks EEG signals classification. *IEEE Sens. J.* 19, 4494–4500. doi: 10.1109/JSEN.2019.2899645

Chavez, M., Grosselin, F., Bussalb, A., de Vico Fallani, F., and Navarro-Sune, X. (2018). Surrogate-based artifact removal from single-channel EEG. *IEEE Trans. Neural Syst. Rehabil. Eng.* 26, 540–550. doi: 10.1109/TNSRE.2018.2794184

Chen, X., Liu, A., Chen, Q., Liu, Y., Zou, L., and McKeown, M. J. (2017a). Simultaneous ocular and muscle artifact removal from EEG data by exploiting diverse statistics. *Comput. Biol. Med.* 88, 1–10. doi: 10.1016/j.compbiomed.2017.06.013

Chen, X., Liu, A., Peng, H., and Ward, R. K. (2014). A preliminary study of muscular artifact cancellation in single-channel EEG. *Sensors* 14, 18370–18389. doi: 10.3390/s141018370

Chen, X., Peng, H., Yu, F., and Wang, K. (2017b). Independent vector analysis applied to remove muscle artifacts in EEG data. *IEEE Trans. Instrum. Meas.* 66, 1770–1779. doi: 10.1109/TIM.2016.2608479

Chen, X., Wang, Y., Gao, S., Jung, T.-P., and Gao, X. (2015). Filter bank canonical correlation analysis for implementing a high-speed SSVEP-based brain-computer interface. *J. Neural Eng.* 12:046008. doi: 10.1088/1741-2560/12/4/046008

Chiappa, S., and Bengio, S. (2004). "HMM and IOHMM modeling of EEG rhythms for asynchronous BCI systems," in *European Symposium on Artificial Neural Networks* (Bruges), 199–204.

Choi, K. (2012). Control of a vehicle with EEG signals in real-time and system evaluation. *Eur. J. Appl. Physiol.* 112, 755–766. doi: 10.1007/s00421-011-2029-6

Chung, J. E., Joo, H. R., Fan, J. L., Liu, D. F., Barnett, A. H., Chen, S., et al. (2019). High-density, long-lasting, and multi-region electrophysiological

recordings using polymer electrode arrays. *Neurona* 101, 21-31.e5. doi: 10.1016/j.neuron.2018.11.002

Cincotti, F., Scipione, A., Timperi, A., Mattia, D., Marciani, A. G., Millan, J., et al. (2003). "Comparison of different feature classifiers for brain computer interfaces," in *Proceedings of Conference on First International IEEE EMBS Conference on Neural Engineering, 2003* (Capri Island: IEEE), 645–647. doi: 10.1109/CNE.2003.1196911

Congedo, M., Barachant, A., and Bhatia, R. (2017). Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review. *Brain Comput. Interf.* 4, 155–174. doi: 10.1080/2326263X.2017.1297192

Congedo, M., Goyat, M., Tarrin, N., Ionescu, G., Varnet, L., Rivet, B., et al. (2011). "Brain invaders: a prototype of an open-source P300- based video game working with the OpenViBE platform," in *5th International Brain-Computer Interface Conference* (Graz), 280–283.

Corralejo, R., Nicolás-Alonso, L. F., Álvarez, D., and Hornero, R. (2014). A P300-based brain–computer interface aimed at operating electronic devices at home for severely disabled people. *Med. Biol. Eng. Comput*. 52, 861–872. doi: 10.1007/s11517-014-1191-5

Coyle, D., McGinnity, T. M., and Prasad, G. (2010). Improving the separability of multiple EEG features for a BCI by neural-time-series-prediction-preprocessing. *Biomed. Signal Process. Control* 5, 196–204. doi: 10.1016/J.BSPC.2010.03.004

Croft, R. J., and Barry, R. J. (2000). Removal of ocular artifact from the EEG: a review. *Neurophysiol. Clin. Neurophysiol*. 30, 5–19. doi: 10.1016/S0987-7053(00)00055-1

Dai, M., Zheng, D., Na, R., Wang, S., Zhang, S., Dai, M., et al. (2019). EEG classification of motor imagery using a novel deep learning framework. *Sensors* 19:551. doi: 10.3390/s19030551

de Clercq, W., Vergult, A., Vanrumste, B., van Paesschen, W., and van Huffel, S. (2006). Canonical correlation analysis applied to remove muscle artifacts from the electroencephalogram. *IEEE Trans. Biomed. Eng.* 53, 2583–2587. doi: 10.1109/TBME.2006.879459

De Vos, M., Kroesen, M., Emkes, R., and Debener, S. (2014). P300 speller BCI with a mobile EEG system: comparison to a traditional amplifier. *J. Neural Eng.* 11:036008. doi: 10.1088/1741-2560/11/3/036008

Debener, S., Minow, F., Emkes, R., Gandras, K., and de Vos, M. (2012). How about taking a low-cost, small, and wireless EEG for a walk? *Psychophysiology* 49, 1617–1621. doi: 10.1111/j.1469-8986.2012.01471.x

Del Pozo-Banos, M., Alonso, J. B., Ticay-Rivas, J. R., and Travieso, C. M. (2014). Electroencephalogram subject identification: a review. *Expert Syst. Appl*. 41, 6537–6554. doi: 10.1016/j.eswa.2014.05.013

Dhiman, R., Priyanka, N. A., and Saini, J. S. (2018). Motor imagery classification from human EEG signatures. *Int. J. Biomed. Eng. Technol*. 26, 101–110. doi: 10.1504/IJBET.2018.089265

Dhindsa, K. (2017). Filter-bank artifact rejection: high performance real-time single-channel artifact detection for EEG. *Biomed. Signal Process. Control* 38, 224–235. doi: 10.1016/j.bspc.2017.06.012

Ditthapron, A., Banluesombatkul, N., Ketrat, S., Chuangsuwanich, E., and Wilaiprasitporn, T. (2018). Universal joint feature extraction for P300 EEG classification using multi-task. *Autoencoder* 2019:2919143. doi: 10.1109/ACCESS.2019.2919143

Djamal, E. C., Abdullah, M. Y., and Renaldi, F. (2017). Brain computer interface game controlling using fast fourier transform and learning vector quantization. *J. Telecommun. Electron. Comput. Eng.* 9, 71–74.

Djamal, E. C., and Lodaya, P. (2017). "EEG based emotion monitoring using wavelet and learning vector quantization" in *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)* (Yogyakarta: IEEE). doi: 10.1109/EECSI.2017.8239090

Dornhege, G., Blankertz, B., Krauledat, M., Losch, F., Curio, G., and Müller, K. R. (2006). Combined optimization of spatial and temporal filters for improving brain-computer interfacing. *IEEE Trans. Biomed. Eng.* 53, 2274–2281. doi: 10.1109/TBME.2006.883649

Dose, H., Møller, J. S., Iversen, H. K., and Puthusserypady, S. (2018). An end-to-end deep learning approach to MI-EEG signal classification for BCIs. *Expert Syst. Appl*. 114, 532–542. doi: 10.1016/J.ESWA.2018.08.031

Duan, F., Lin, D., Li, W., and Zhang, Z. (2015). Design of a multimodal EEG-based hybrid BCI system with visual servo module. *IEEE Trans. Auton. Ment. Dev*. 7, 332–341. doi: 10.1109/TAMD.2015.2434951

Duan, J., Li, Z., Yang, C., and Xu, P. (2014). "Shared control of a brain-actuated intelligent wheelchair," in *Proceeding of the 11th World Congress on Intelligent Control and Automation* (Shenyang: IEEE), 341–346. doi: 10.1109/WCICA.2014.7052737

Edlinger, G., and Guger, C. (2012). "A hybrid Brain-Computer Interface for improving the usability of a smart home control," in *2012 ICME International Conference on Complex Medical Engineering (CME)* (Kobe: IEEE), 182–185. doi: 10.1109/ICCME.2012.6275714

El-Fiqi, H., Wang, M., Salimi, N., Kasmarik, K., Barlow, M., and Abbass, H. (2018). "Convolution neural networks for person identification and verification using steady state visual evoked potential," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (Miyazaki: IEEE), 1062–1069. doi: 10.1109/SMC.2018.00188

Fairclough, S. H., and Gilleade, K. (eds.). (2014). *Advances in Physiological Computing*. London: Springer. doi: 10.1007/978-1-4471-6392-3

Fan, J., Ma, C., and Zhong, Y. (2019). *A Selective Overview of Deep Learning*. Available online at: http://arxiv.org/abs/1904.05526 (accessed July 15, 2019).

Farwell, L. A., and Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalogr. Clin. Neurophysiol.* 70, 510–523.

Fatourechi, M., Ward, R. K., Mason, S. G., Huggins, J., Schlögl, A., and Birch, G. E. (2008). "Comparison of evaluation metrics in classification applications with imbalanced datasets," in *2008 Seventh International Conference on Machine Learning and Applications* (San Diego, CA: IEEE), 777–782. doi: 10.1109/ICMLA.2008.34

Fernández-Rodríguez, Á., Velasco-Álvarez, F., and Ron-Angevin, R. (2016). Review of real brain-controlled wheelchairs. *J. Neural Eng.* 13:061001. doi: 10.1088/1741-2560/13/6/061001

Finke, A., Lenhardt, A., and Ritter, H. (2009). The mindgame: a P300-based brain–computer interface game. *Neural Netw.* 22, 1329–1333. doi: 10.1016/J.NEUNET.2009.07.003

Fitzgibbon, S. P., Lewis, T. W., Powers, D. M. W., Whitham, E. W., Willoughby, J. O., and Pope, K. J. (2013). Surface laplacian of central scalp electrical signals is insensitive to muscle contamination. *IEEE Trans. Biomed. Eng.* 60, 4–9. doi: 10.1109/TBME.2012.2195662

Ganin, I. P., Shishkin, S. L., and Kaplan, A. Y. (2013). A P300-based brain-computer interface with Stimuli on moving objects: four-session single-trial and triple-trial tests with a game-like task design. *PLoS ONE* 8:e77755. doi: 10.1371/journal.pone.0077755

Garrett, D., Peterson, D. A., Anderson, C. W., and Thaut, M. H. (2003). Comparison of linear, nonlinear, and feature selection methods for eeg signal classification. *IEEE Trans. Neural Syst. Rehabil. Eng.* 11, 141–144. doi: 10.1109/TNSRE.2003.814441

Gaur, P., Pachori, R. B., Wang, H., and Prasad, G. (2018). A multi-class EEG-based BCI classification using multivariate empirical mode decomposition based filtering and Riemannian geometry. *Expert Syst. Appl.* 95, 201–211. doi: 10.1016/J.ESWA.2017.11.007

Göksu, H. (2018). BCI Oriented EEG analysis using log energy entropy of wavelet packets. *Biomed. Signal Process. Control* 44, 101–109. doi: 10.1016/j.bspc.2018.04.002

Gouy-Pailler, C., Achard, S., Rivet, B., Jutten, C., Maby, E., Souloumiac, A., et al. (2007). "Topographical dynamics of brain connections for the design of asynchronous brain-computer interfaces," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Lyon: IEEE), 2520–2523. doi: 10.1109/IEMBS.2007.4352841

Gratton, G., Coles, M. G., and Donchin, E. (1983). A new method for off-line removal of ocular artifact. *Electroencephalogr. Clin. Neurophysiol.* 55, 468–484. doi: 10.1016/0013-4694(83)90135-9

Guan, S., Zhao, K., and Yang, S. (2019). Motor imagery EEG classification based on decision tree framework and riemannian geometry. *Comput. Intell. Neurosci.* 2019:5627156. doi: 10.1155/2019/5627156

Guarnieri, R., Marino, M., Barban, F., Ganzetti, M., and Mantini, D. (2018). Online EEG artifact removal for BCI applications by adaptive spatial filtering. *J. Neural Eng.* 15:056009. doi: 10.1088/1741-2552/aacfdf

Guede-Fernández, F., Fernández-Chimeno, M., Ramos-Castro, J., and García-González, M. A. (2019). Driver drowsiness detection based on respiratory signal analysis. *IEEE Access* 7, 81826–81838. doi: 10.1109/ACCESS.2019.2924481

Guo, S., Lin, S., and Huang, Z. (2015). "Feature extraction of P300s in EEG signal with discrete wavelet transform and fisher criterion," in *2015 8th International Conference on Biomedical Engineering and Informatics (BMEI)* (Shenyang: IEEE), 200–204. doi: 10.1109/BMEI.2015.7401500

Gursel Ozmen, N., Gumusel, L., and Yang, Y. (2018). A biologically inspired approach to frequency domain feature extraction for EEG classification. *Comput. Math. Methods Med*. 2018, 19–24. doi: 10.1155/2018/9890132

Ha, K. W., and Jeong, J. W. (2019). Motor imagery EEG classification using capsule networks. *Sensors* 19 19:2854. doi: 10.3390/s19132854

Hamada, K., Mori, H., Shinoda, H., and Rutkowski, T. M. (2014). Airborne ultrasonic tactile display brain-computer interface paradigm. *Front. Hum. Neurosci.* 9:18. doi: 10.3217/978-3-85125-378-8-18

Han, C.-H., Kim, Y.-W., Kim, D. Y., Kim, S. H., Nenadic, Z., and Im, C.-H. (2019). Electroencephalography-based endogenous brain–computer interface for online communication with a completely locked-in patient. *J. Neuroeng. Rehabil*. 16:18. doi: 10.1186/s12984-019-0493-0

Haselsteiner, E., and Pfurtscheller, G. (2000). Using time-dependent neural networks for EEG classification. *IEEE Trans. Rehabil. Eng*. 8, 457–463. doi: 10.1109/86.895948

Hefron, R. G., Borghetti, B. J., Christensen, J. C., and Kabban, C. M. S. (2017). Deep long short-term memory structures model temporal dependencies improving cognitive workload estimation. *Pattern Recognit. Lett*. 94, 96–104. doi: 10.1016/J.PATREC.2017.05.020

Hill, N. J., and Schölkopf, B. (2012). An online brain–computer interface based on shifting attention to concurrent streams of auditory stimuli. *J. Neural Eng.* 9:026011. doi: 10.1088/1741-2560/9/2/026011

Hoffmann, U., Vesin, J.-M., Ebrahimi, T., and Diserens, K. (2008). An efficient P300-based brain–computer interface for disabled subjects. *J. Neurosci. Methods* 167, 115–125. doi: 10.1016/j.jneumeth.2007.03.005

Holz, E. M., Höhne, J., Staiger-Sälzer, P., Tangermann, M., and Kübler, A. (2013). Brain-computer interface-controlled gaming: evaluation of usability by severely motor restricted end-users. *Artif. Intell. Med*. 59, 111-120. doi: 10.1016/j.artmed.2013.08.001

Hortal, E., Planelles, D., Costa, A., Iáñez, E., Úbeda, A., Azorín, J. M., et al. (2015). SVM-based brain–machine interface for controlling a robot arm through four mental tasks. *Neurocomputing* 151, 116–121. doi: 10.1016/J.NEUCOM.2014.09.078

Hu, J. (2018). An approach to EEG-based gender recognition using entropy measurement methods. *Knowledge Based Syst*. 140, 134–141. doi: 10.1016/J.KNOSYS.2017.10.032

Huang, J., Xu, X., and Zhang, T. (2017). "Emotion classification using deep neural networks and emotional patches," in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (Kansas City, MO: IEEE), 958–962. doi: 10.1109/BIBM.2017.8217786

Hwang, H. J., Kim, S., Choi, S., and Im, C. H. (2013). EEG-based brain-computer interfaces: a thorough literature survey. *Int. J. Hum. Comput. Interact*. 29, 814–826.doi: 10.1080/10447318.2013.780869

Iatsenko, D., McClintock, P. V. E., and Stefanovska, A. (2015). Nonlinear mode decomposition: a noise-robust, adaptive decomposition method. *Phys. Rev. E* 92:032916. doi: 10.1103/PhysRevE.92.032916

Idowu, O. P., Fang, P., Li, X., Xia, Z., Xiong, J., and Li, G. (2018). "Towards control of EEG-based robotic arm using deep learning via stacked sparse autoencoder," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (Kuala Lumpur: IEEE), 1053–1057. doi: 10.1109/ROBIO.2018.8665089

Ieracitano, C., Mammone, N., Hussain, A., and Morabito, F. C. (2020). A novel multi-modal machine learning based approach for automatic classification of EEG recordings in dementia. *Neural Netw.* 123, 176–190. doi: 10.1016/j.neunet.2019.12.006

iHuman (2019). *Blurring Lines between Mind and Machine.* London. Available at: https://royalsociety.org/-/media/policy/projects/ihuman/report-neural-interfaces.pdf (accessed November 05, 2019).

Islam, M. R., Tanaka, T., and Molla, M. K. I. (2018). Multiband tangent space mapping and feature selection for classification of EEG during motor imagery. *J. Neural Eng.* 15:046021. doi: 10.1088/1741-2552/aac313

Iturrate, I., Antelis, J., and Minguez, J. (2009b)." Synchronous EEG brain-actuated wheelchair with automated navigation," in *2009 IEEE International Conference on Robotics and Automation* (Kobe: IEEE), 2318–2325. doi: 10.1109/ROBOT.2009.5152580

Iturrate, I., Antelis, J. M., Kubler, A., and Minguez, J. (2009a). A noninvasive brain-actuated wheelchair based on a P300 neurophysiological protocol and automated navigation. *IEEE Trans. Robot*. 25, 614–627. doi: 10.1109/TRO.2009.2020347

Jawed, S., Amin, H. U., Malik, A. S., and Faye, I. (2018). "EEG visual and non-visual learner classification using LSTM recurrent neural networks," in *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)* (Sarawak: IEEE), 467–471. doi: 10.1109/IECBES.2018.8626711

Ji, N., Ma, L., Dong, H., and Zhang, X. (2019). EEG signals feature extraction based on DWT and EMD combined with approximate entropy. *Brain Sci.* 9:201. doi: 10.3390/brainsci9080201

Jiang, L., Stocco, A., Losey, D. M., Abernethy, J. A., Prat, C. S., and Rao, R. P. N. (2019). BrainNet: a multi-person brain-to-brain interface for direct collaboration between brains. *Sci. Rep*. 9:7. doi: 10.1038/s41598-019-41895-7

Jin, J., Allison, B. Z., Sellers, E. W., Brunner, C., Horki, P., Wang, X., et al. (2011). Optimized stimulus presentation patterns for an event-related potential EEG-based brain-computer interface. *Med. Biol. Eng. Comput*. 49, 181–191. doi: 10.1007/s11517-010-0689-8

Jin, J., Li, S., Daly, I., Miao, Y., Liu, C., Wang, X., et al. (2020). The study of generic model set for reducing calibration time in P300-based brain-computer interface. *IEEE Trans. Neural Syst. Rehabil. Eng*. 28, 3–12. doi: 10.1109/TNSRE.2019.2956488

Jin, J., Miao, Y., Daly, I., Zuo, C., Hu, D., and Cichocki, A. (2019). Correlation-based channel selection and regularized feature optimization for MI-based BCI. *Neural Networks* 118, 262–270. doi: 10.1016/j.neunet.2019.07.008

John, P., Ellerson, P. C., and Cohen, J. C. (1996). P300, stimulus intensity, modality, and probability. *Int. J. Psychophysiol*. 23, 55–62.

Kalunga, E. K., Chevallier, S., Barthélemy, Q., Djouani, K., Monacelli, E., and Hamam, Y. (2016). Online SSVEP-based BCI using Riemannian geometry. *Neurocomputing* 191, 55–68. doi: 10.1016/J.NEUCOM.2016.01.007

Kanjo, E., Younis, E. M. G., and Ang, C. S. (2019). Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection. *Inf. Fusion* 49, 46–56. doi: 10.1016/J.INFFUS.2018.09.001

Kaper, M., Meinicke, P., Grossekathoefer, U., Lingner, T., and Ritter, H. (2004). BCI competition 2003—data set IIb: support vector machines for the P300 speller paradigm. *IEEE Trans. Biomed. Eng*. 51, 1073–1076. doi: 10.1109/TBME.2004.826698

Kaplan, A. Y., Shishkin, S. L., Ganin, I. P., Basyul, I. A., and Zhigalov, A. Y. (2013). Adapting the P300-based brain–computer interface for gaming: a review. *IEEE Trans. Comput. Intell. AI Games* 5, 141–149. doi: 10.1109/TCIAIG.2012.2237517

Käthner, I., Kübler, A., and Halder, S. (2015). Rapid P300 brain-computer interface communication with a head-mounted display. *Front. Neurosci*. 9:207. doi: 10.3389/fnins.2015.00207

Kaufmann, T., and Kübler, A. (2014). Beyond maximum speed—a novel two-stimulus paradigm for brain–computer interfaces based on event-related potentials (P300-BCI). *J. Neural Eng*. 11:056004. doi: 10.1088/1741-2560/11/5/056004

Kaur, B., Singh, D., and Roy, P. P. (2018). EEG based emotion classification mechanism in BCI. *Proc. Comput. Sci*. 132, 752–758. doi: 10.1016/J.PROCS.2018.05.087

Kawde, P., and Verma, G. K. (2017). "Deep belief network based affect recognition from physiological signals," in *2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON)* (Mathura: IEEE), 587–592. doi: 10.1109/UPCON.2017.8251115

Ke, Y., Wang, N., Du, J., Kong, L., Liu, S., Xu, M., et al. (2019). The effects of transcranial direct current stimulation (tDCS) on working memory training in healthy young adults. *Front. Hum. Neurosci*. 13:19. doi: 10.3389/fnhum.2019.00019

Kevric, J., and Subasi, A. (2017). Comparison of signal decomposition methods in classification of EEG signals for motor-imagery BCI system. *Biomed. Signal Process. Control* 31, 398–406. doi: 10.1016/j.bspc.2016.09.007

Khan, J., Bhatti, M. H., Khan, U. G., and Iqbal, R. (2019). Multiclass EEG motor-imagery classification with sub-band common spatial patterns. *Eurasip J. Wirel. Commun. Netw*. 2019:174. doi: 10.1186/s13638-019-1497-y

Kilicarslan, A., Grossman, R. G., and Contreras-vidal, J. L. (1976). A robust motifs based artifacts removal technique from EEG. *Biomed. Phys. Eng. Express* 3, 1–18.

Kim, M., Hwang, T., Oh, E., and Hwangbo, M. (2013). "Toward realistic implementation of brain-computer interface for TV channel control," in *2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE)* (Tokyo: IEEE), 394–396. doi: 10.1109/GCCE.2013.6664868

Kleber, B., and Birbaumer, N. (2005). Direct brain communication: neuroelectric and metabolic approaches at Tübingen. *Cogn. Process*. 6, 65–74. doi: 10.1007/s10339-004-0045-8

Kodi, A., Kumar, D., Kodali, D., and Pasha, I. A. (2013). "EEG-controlled wheelchair for ALS patients," in *2013 International Conference on Communication Systems and Network Technologies* (Gwalior: IEEE), 879–883. doi: 10.1109/CSNT.2013.190

Korik, A., Sosnik, R., Siddique, N., and Coyle, D. (2019). Decoding imagined 3D arm movement trajectories from EEG to control two virtual arms—a pilot study. *Front. Neurorobot*. 13: 94. doi: 10.3389/fnbot.2019.00094

Kosmyna, N., Tarpin-Bernard, F., Bonnefond, N., and Rivet, B. (2016). Feasibility of BCI Control in a Realistic Smart Home Environment. *Front. Hum. Neurosci*. 10:416. doi: 10.3389/fnhum.2016.00416

Krauledat, M., Grzeska, K., Sagebaum, M., Blankertz, B., Vidaurre, C., Müller, K.-R., et al. (2009). "Playing Pinball with non-invasive BCI," in *21st International Conference on Neural Information Processing Systems* (Vancouver, BC), 1641–1648.

Kreilinger, A., Hiebel, H., and Muller-Putz, G. R. (2016). Single versus multiple events error potential detection in a BCI-controlled car game with continuous and discrete feedback. *IEEE Trans. Biomed. Eng.* 63, 519–529. doi: 10.1109/TBME.2015.2465866

Krepki, R., Blankertz, B., Curio, G., and Müller, K.-R. (2007). The Berlin brain-computer interface (BBCI) – towards a new communication channel for online control in gaming applications. *Multimed. Tools Appl*. 33, 73–90. doi: 10.1007/s11042-006-0094-3

Kristo, G., Real, R., Blefari, M. L., Brunner, C., Blankertz, B., Höhne, J., et al. (2013). *Contribution to Roadmap*. Available online at: https://pdfs.semanticscholar.org/5cb4/11de3db4941d5c7ecfc19de8af9243fb63d5.pdf (accessed November 12, 2019).

Kubler, A., Mushahwar, V. K., Hochberg, L. R., and Donoghue, J. P. (2006). BCI meeting 2005—workshop on clinical issues and applications. *IEEE Trans. Neural Syst. Rehabil. Eng*. 14, 131–134. doi: 10.1109/TNSRE.2006.875585

Kulasingham, J. P., Vibujithan, V., and De Silva, A. C. (2016). "Deep belief networks and stacked autoencoders for the P300 Guilty Knowledge Test," in *2016 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)* (Kuala Lumpur: IEEE), 127–132. doi: 10.1109/IECBES.2016.7843428

La Rocca, D., Campisi, P., and Scarano, G. (2013). "On the repeatability of EEG features in a biometric recognition framework using a resting state protocol," in *6th International Conference on Bio-inspired Systems and Signal Processing BIOSIGNALS 2013*.

Lalor, E. C., Kelly, S. P., Finucane, C., Burke, R., Smith, R., Reilly, R. B., et al. (2005). Steady-state VEP-based brain-computer interface control in an immersive 3D gaming environment. *EURASIP J. Adv. Signal Process*. 2005, 3156–3164. doi: 10.1155/ASP.2005.3156

Lau, T. M., Gwin, J. T., McDowell, K. G., and Ferris, D. P. (2012). Weighted phase lag index stability as an artifact resistant measure to detect cognitive EEG activity during locomotion. *J. Neuroeng. Rehabil*. 9:47. doi: 10.1186/1743-0003-9-47

Lawhern, V., Hairston, W. D., McDowell, K., Westerfield, M., and Robbins, K. (2012). Detection and classification of subject-generated artifacts in EEG signals using autoregressive models. *J. Neurosci. Methods* 208, 181–189. doi: 10.1016/j.jneumeth.2012.05.017

Lazarou, I., Nikolopoulos, S., and Petrantonakis, P. C. (2018). EEG-based brain – computer interfaces for communication and rehabilitation of people with motor impairment: a novel approach of the 21 st Century. *Front. Hum. Neurosci.* 12:14. doi: 10.3389/fnhum.2018.00014

Lee, H. J., Kim, H. S., and Park, K. S. (2013). "A study on the reproducibility of biometric authentication based on electroencephalogram (EEG)," in *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)* (San Diego, CA: IEEE), 13–16. doi: 10.1109/NER.2013.6695859

Lee, H. K., and Choi, Y.-S. (2019). Application of continuous wavelet transform and convolutional neural network in decoding motor imagery brain-computer interface. *Entropy* 21:1199. doi: 10.3390/e21121199

Lee, S., McKeown, M. J., Wang, Z. J., and Chen, X. (2019). Removal of high-voltage brain stimulation artifacts from simultaneous EEG recordings. *IEEE Trans. Biomed. Eng*. 66, 50–60. doi: 10.1109/TBME.2018.2828808

Lee, S. B., Kim, H. J., Kim, H., Jeong, J. H., Lee, S. W., and Kim, D. J. (2019). Comparative analysis of features extracted from EEG spatial, spectral and temporal domains for binary and multiclass motor imagery classification. *Inf. Sci. (Ny)*. 502, 190–200. doi: 10.1016/j.ins.2019.06.008

Lee, T.-J., and Sim, K.-B. (2015). Vowel classification of imagined speech in an electroencephalogram using the deep belief network. *J. Inst. Control. Robot. Syst*. 21, 59–64. doi: 10.5302/J.ICROS.2015.14.0073

Li, G., Lee, C. H., Jung, J. J., Youn, Y. C., and Camacho, D. (2019). Deep learning for EEG data analytics: a survey. *Concurr. Comput. Pract. Exp*. e5199. doi: 10.1002/cpe.5199

Li, J., Liang, J., Zhao, Q., Li, J., Hong, K., and Zhang, L. (2013). Design of assistive wheelchair system directly steered by human thoughts. *Int. J. Neural Syst*. 23:1350013. doi: 10.1142/S0129065713500135

Li, M., Luo, X., Yang, J., and Sun, Y. (2016a). Applying a locally linear embedding algorithm for feature extraction and visualization of MI-EEG. *J. Sens*. 2016:7481946. doi: 10.1155/2016/7481946

Li, M., Zhang, M., Luo, X., and Yang, J. (2016b). "Combined long short-term memory based network employing wavelet coefficients for MI-EEG recognition," in *2016 IEEE International Conference on Mechatronics and Automation* (Harbin: IEEE), 1971–1976. doi: 10.1109/ICMA.2016.7558868

Li, W., Duan, F., Sheng, S., Xu, C., Liu, R., Zhang, Z., et al. (2018). A human-vehicle collaborative simulated driving system based on hybrid brain-computer interfaces and computer vision. *IEEE Trans. Cogn. Dev. Syst*. 10, 810–822. doi: 10.1109/TCDS.2017.2766258

Li, Y., Long, J., Yu, T., Yu, Z., Wang, C., Zhang, H., et al. (2010). An EEG-based BCI system for 2-D cursor control by combining Mu/Beta rhythm and P300 potential. *IEEE Trans. Biomed. Eng*. 57, 2495-2505.doi: 10.1109/TBME.2010.2055564

Li, Y., Pan, J., Wang, F., and Yu, Z. (2013). A hybrid BCI system combining P300 and SSVEP and its application to wheelchair control. *IEEE Trans. Biomed. Eng*. 60, 3156–3166. doi: 10.1109/TBME.2013.2270283

Li, Z., Lei, S., Su, C.-Y., and Li, G. (2013). "Hybrid brain/muscle-actuated control of an intelligent wheelchair," in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (Shenzhen: IEEE), 19–25. doi: 10.1109/ROBIO.2013.6739429

Lin, C.-T., Lin, B.-S., Lin, F.-C., and Chang, C.-J. (2014). Brain computer interface-based smart living environmental auto-adjustment control system in UPnP home networking. *IEEE Syst. J.* 8, 363–370. doi: 10.1109/JSYST.2012.2192756

Lin, J., Liu, S., Huang, G., Zhang, Z., and Huang, K. (2019). "The recognition of driving action based on EEG signals using wavelet-CSP algorithm," in *International Conference on Digital Signal Processing, DSP* (Shanghai: Institute of Electrical and Electronics Engineers Inc.). doi: 10.1109/ICDSP.2018.8631540

Lin, J. S., and She, B. H. (2020). A BCI system with motor imagery based on bidirectional long-short term memory. *IOP Conf. Ser. Mater. Sci. Eng*. 719:012026. doi: 10.1088/1757-899X/719/1/012026

Liu, A., Chen, K., Liu, Q., Ai, Q., Xie, Y., and Chen, A. (2017). Feature selection for motor imagery EEG classification based on firefly algorithm and learning automata. *Sensors* 17:2576. doi: 10.3390/s17112576

Liu, J., Su, Y., and Liu, Y. (2018). Multi-Modal Emotion Recognition with Temporal-Band Attention Based on LSTM-RNN," in *Advances in Multimedia Information Processing – PCM 2017. PCM 2017. Lecture Notes in Computer Science, Vol. 10735,* eds B. Zeng, Q. Huang, A. El Saddik, H. Li, S. Jiang, and X. Fan (Cham: Springer), 194–204. doi: 10.1007/978-3-319-77380-3_19

Liu, Y., Yu, M., Zhao, G., Song, J., Ge, Y., and Shi, Y. (2017). Real-time movie-induced discrete emotion recognition from EEG signals.. *IEEE Trans. Affect. Comput*. 9:2660485. doi: 10.1109/TAFFC.2017.2660485

Long, J., Li, Y., Wang, H., Yu, T., Pan, J., and Li, F. (2012a). A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair. *IEEE Trans. Neural. Syst. Rehabil. Eng*. 20, 720–729. doi: 10.1109/TNSRE.2012.2197221

Long, J., Li, Y., Yu, T., and Gu, Z. (2012b). Target selection with hybrid feature for BCI-based 2-D cursor control. *IEEE Trans. Biomed. Eng.* 59, 132–140. doi: 10.1109/TBME.2011.2167718

Lopes, A. C., Pires, G., and Nunes, U. (2013). Assisted navigation for a brain-actuated intelligent wheelchair. *Rob. Auton. Syst.* 61, 245-258. doi: 10.1016/j.robot.2012.11.002

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., et al. (2018). A review of classification algorithms for EEG-based brain-computer interfaces: a 10-year update. *J. Neural Eng.* 15:031005. doi: 10.1088/1741-2552/aab2f2

Lotte, F., Bougrain, L., and Clerc, M. (2015). "Electroencephalography (EEG)-based brain-computer interfaces," in *Wiley Encyclopedia of Electrical and Electronics Engineering*, ed J. Webster (Hoboken, NJ: John Wiley and Sons, Inc.), 1–20. doi: 10.1002/047134608X.W8278

Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., and Arnaldi, B. (2007). A review of classification algorithms for EEG-based brain–computer interfaces. *J. Neural Eng.* 4, R1–R13. doi: 10.1088/1741-2560/4/2/R01

Lu, N., Li, T., Ren, X., and Miao, H. (2017). A deep learning scheme for motor imagery classification based on restricted boltzmann machines. *IEEE Trans. Neural Syst. Rehabil. Eng.* 25, 566–576. doi: 10.1109/TNSRE.2016.2601240

Lu, Z., Gao, N., Liu, Y., and Li, Q. (2018). "The detection of P300 potential based on deep belief network," in *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)* (Beijing: IEEE), 1–5. doi: 10.1109/CISP-BMEI.2018.8633147

Luo, T., Zhou, C., and Chao, F. (2018). Exploring spatial-frequency-sequential relationships for motor imagery classification with recurrent neural network. *BMC Bioinform.* 19:344. doi: 10.1186/s12859-018-2365-1

Ma, X., Qiu, S., Du, C., Xing, J., and He, H. (2018). "Improving EEG-based motor imagery classification via spatial and temporal recurrent neural networks," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Honolulu, HI: IEEE), 1903–1906. doi: 10.1109/EMBC.2018.8512590

Maby, E., Perrin, M., Bertrand, O., Sanchez, G., and Mattout, J. (2012). BCI could make old two-player games even more fun: a proof of concept with "connect Four." *Adv. Hum.Comput. Interact.* 2012:124728. doi: 10.1155/2012/124728

Majidov, I., and Whangbo, T. (2019). Efficient classification of motor imagery electroencephalography signals using deep learning methods. *Sensors* 19:1736. doi: 10.3390/s19071736

Mammone, N., Ieracitano, C., and Morabito, F. C. (2020). A deep CNN approach to decode motor preparation of upper limbs from time–frequency maps of EEG signals at source level. *Neural Networks* 124, 357–372. doi: 10.1016/j.neunet.2020.01.027

Mandel, C., Luth, T., Laue, T., Rofer, T., Graser, A., and Krieg-Bruckner, B. (2009). "Navigating a smart wheelchair with a brain-computer interface interpreting steady-state visual evoked potentials," *in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE),* 1118–1125. doi: 10.1109/IROS.2009.5354534

Manojprabu, M., and Sarma Dhulipala, V. R. (2020). Improved energy efficient design in software defined wireless electroencephalography sensor networks (WESN) using distributed architecture to remove artifact. *Comput. Commun.* 152, 266–271. doi: 10.1016/j.comcom.2019.12.056

Mara, S., Müller, T., Freire, T., Mário, B., and Filho, S. (2013). Proposal of a SSVEP-BCI to Command a Robotic Wheelchair. *J. Control Autom. Electr. Syst.* 24, 97–105. doi: 10.1007/s40313-013-0002-9

Marcel, S., and Millan, J. D. R. (2007). Person Authentication Using Brainwaves (EEG) and maximum a posteriori model adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 743–752. doi: 10.1109/TPAMI.2007.1012

Marshall, D., Coyle, D., Wilson, S., and Callaghan, M. (2013). Games, gameplay, and BCI: the state of the Art. *IEEE Trans. Comput. Intell. AI Games* 5, 82–99. doi: 10.1109/TCIAIG.2013.2263555

Mason, S. G., and Birch, G. E. (2003). A general framework for brain-computer interface design. *IEEE Trans. Neural Syst. Rehabil. Eng.* 11, 70–85. doi: 10.1109/TNSRE.2003.810426

Masood, M. H., Ahmad, M., Kathia, M. A., Zafar, R. Z., and Zahid, A. N. (2016). "Brain Computer Interface Based Smart Home Control," in *2013 IEEE International Symposium on Consumer Electronics (ISCE)* (Hsinchu), 35–36, doi: 10.1109/ISCE.2013.6570240

Maye, A., Zhang, D., Wang, Y., Gao, S., and Engel, A. K. (2011). Multimodal brain-computer interfaces. *Tsinghua Sci. Technol.* 16, 133–139. doi: 10.1016/S1007-0214(11)70020-7

McFarland, D. J., Neat, G. W., Read, R. F., and Wolpaw, J. R. (1993). An EEG-based method for graded cursor control. *Psychobiology* 21, 77–81. doi: 10.3758/bf03327130

Meziani, A., Djouani, K., Medkour, T., and Chibani, A. (2019). A Lasso quantile periodogram based feature extraction for EEG-based motor imagery. *J. Neurosci. Methods* 328:108434. doi: 10.1016/j.jneumeth.2019.108434

Michielli, N., Acharya, U. R., and Molinari, F. (2019). Cascaded LSTM recurrent neural network for automated sleep stage classification using single-channel EEG signals. *Comput. Biol. Med.* 106, 71–81. doi: 10.1016/J.COMPBIOMED.2019.01.013

Middendorf, M., McMillan, G., Calhoun, G., and Jones, K. S. (2000). Brain-computer interfaces based on the steady-state visual-evoked response. *IEEE Trans. Rehabil. Eng.* 8, 211–214. doi: 10.1109/86.847819

Mihajlovi, V., and Peuscher, J. (2012). *To What Extent Can Dry and Water-Based EEG Electrodes Replace Conductive Gel Ones? A Steady State Visual Evoked Potential Brain-Computer Interface Case Study*, 14–26. doi: 10.5220/0003726000140026

Millan, J. D. R., and Mourino, J. (2003). Asynchronous bci and local neural classifiers: an overview of the adaptive brain interface project. *IEEE Trans. Neural Syst. Rehabil. Eng.* 11, 159–161. doi: 10.1109/TNSRE.2003.814435

Miralles, F., Vargiu, E., Dauwalder, S., Sol,à, M., Müller-Putz, G., Wriessnegger, S. C., et al. (2015). Brain computer interface on track to home. *Sci. World J.* 2015:623896. doi: 10.1155/2015/623896

Mohammadpour, M., and Rahmani, V. (2017). "A hidden markov model-based approach to removing EEG artifact," in *5th Iranian Joint Congress on Fuzzy and Intelligent Systems - 16th Conference on Fuzzy Systems and 14th Conference on Intelligent Systems, CFIS 2017* (Qazvin: Institute of Electrical and Electronics Engineers Inc.), 46–49. doi: 10.1109/CFIS.2017.8003655

Moses, D. A., Leonard, M. K., Makin, J. G., and Chang, E. F. (2019). Real-time decoding of question-and-answer speech dialogue using human cortical activity. *Nat. Commun.* 10:3096. doi: 10.1038/s41467-019-10994-4

Mu Li, M., and Bao-Liang Lu, B.-L. (2009). "Emotion classification based on gamma-band EEG," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Minneapolis, MN: IEEE), 1223–1226. doi: 10.1109/IEMBS.2009.5334139

Mühl, C., Gürkök, H., Bos, D. P.-O., Thurlings, M. E., Scherffig, L., Duvinage, M., et al. (2010). "Bacteria hunt: a multimodal, multiparadigm BCI game," in *5th International Summer Workshop on Multimodal Interfaces, eNTERFACE 2009* (Genova: University of Genua), 41–62.

Muller-Putz, G., Scherer, R., Neuper, C., and Pfurtscheller, G. (2006). Steady-state somatosensory evoked potentials: suitable brain signals for brain–computer interfaces? *IEEE Trans. Neurol. Syst. Rehabil. Eng.* 14, 30–37. doi: 10.1093/acprof:oso/9780198528272.003.0001

Murugappan, M. (2011). "Human emotion classification using wavelet transform and KNN," in *Proceedings of the 2011 International Conference Pattern Analysis. Intelligence Robot ICPAIR 2011 Vol. 1* (Putrajaya), 148–153. doi: 10.1109/ICPAIR.2011.5976886

Murugappan, M., Murugappan, S., Balaganapathy, B., and Gerard, C. (2014). "Wireless EEG signals based neuromarketing system using Fast Fourier Transform (FFT)," in *Proceedings - 2014 IEEE 10th International Colloquium on Signal Processing and Its Applications, CSPA 2014* (Kuala Lumpur), 25–30. doi: 10.1109/CSPA.2014.6805714

Musk, E. (2019). An integrated brain-machine interface platform with thousands of channels. *J. Med. Internet Res.* 21:e16194. doi: 10.2196/16194

Nair, A. V., Kumar, K. M., and Mathew, J. (2018). An improved approach for EEG signal classification using autoencoder," in *2018 8th International Symposium on Embedded Computing and System Design (ISED)* (Cochin: IEEE), 6–10. doi: 10.1109/ISED.2018.8704011

Nakanishi, M., Wang, Y., Chen, X., Wang, Y.-T., Gao, X., and Jung, T.-P. (2018). Enhancing detection of SSVEPs for a high-speed brain speller using task-related component analysis. *IEEE Trans. Biomed. Eng.* 65, 104–112. doi: 10.1109/TBME.2017.2694818

Narejo, S., Pasero, E., and Kulsoom, F. (2016). EEG based eye state classification using deep belief network and stacked autoEncoder. *Int. J. Electr. Comput. Eng.* 6:3131. doi: 10.11591/ijece.v6i6.12967

Nelson, J., McKinley, R. A., Phillips, C., McLntire, U., Goodyear, C., Kreiner, A., et al. (2016). The effects of transcranial direct current stimulation (tDCS) on multitasking throughput capacity. *Front. Hum. Neurosci.* 10:589. doi: 10.3389/fnhum.2016.00589

Ng, D. W.-K., Soh, Y.-W., and Goh, S.-Y. (2014). "Development of an Autonomous BCI Wheelchair," in *2014 IEEE Symposium on Computational Intelligence in Brain Computer Interfaces (CIBCI)* (Orlando, FL: IEEE), 1–4. doi: 10.1109/CIBCI.2014.7007784

Nguyen, D., Tran, D., Sharma, D., and Ma, W. (2017). On the study of EEG-based cryptographic key generation. *Proc. Comput. Sci.* 112, 936–945. doi: 10.1016/JPROCS.2017.08.126

Nguyen, T.-H., and Chung, W.-Y. (2019). A single-channel SSVEP-based BCI speller using deep learning. *IEEE Access* 7, 1752–1763. doi: 10.1109/ACCESS.2018.2886759

Nguyen, T.-H., Yang, D.-L., and Chung, W.-Y. (2018). A high-rate BCI speller based on eye-closed EEG signal. *IEEE Access* 6, 33995–34003. doi: 10.1109/ACCESS.2018.2849358

Nicolas-Alonso, L. F., Corralejo, R., Gomez-Pilar, J., Álvarez, D., and Hornero, R. (2015). Adaptive semi-supervised classification to reduce intersession non-stationarity in multiclass motor imagery-based brain-computer interfaces. *Neurocomputing* 159, 186–196. doi: 10.1016/j.neucom.2015.02.005

Nicolas-Alonso, L. F., and Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors* 12, 1211–1279. doi: 10.3390/s120201211

Novak, D., Sigrist, R., Gerig, N. J., Wyss, D., Bauer, R., Götz, U., et al. (2018). Benchmarking brain-computer interfaces outside the laboratory: the cybathlon 2016. *Front. Neurosci.* 11:756. doi: 10.3389/fnins.2017.00756

Obeidat, Q. T., Campbell, T. A., and Kong, J. (2015). Introducing the edges paradigm: a P300 brain–computer interface for spelling written words. *IEEE Trans. Hum. Mach. Syst.* 45, 727–738. doi: 10.1109/THMS.2015.2456017

Obermaier, B., Guger, C., Neuper, C., and Pfurtscheller, G. (2001). Hidden Markov models for online classification of single trial EEG data. *Pattern Recognit. Lett.* 22, 1299–1309. doi: 10.1016/S0167-8655(01)00075-7

Oikonomou, V. P., Georgiadis, K., Liaros, G., Nikolopoulos, S., and Kompatsiaris, I. (2017). "A comparison study on EEG signal processing techniques using motor imagery EEG data," in *Proceedings - IEEE Symposium on Computer-Based Medical Systems* (Thessaloniki: Institute of Electrical and Electronics Engineers Inc.), 781–786. doi: 10.1109/CBMS.2017.113

Olivas-Padilla, B. E., and Chacon-Murguia, M. I. (2019). Classification of multiple motor imagery using deep convolutional neural networks and spatial filters. *Appl. Soft Comput.* 75, 461–472. doi: 10.1016/J.ASOC.2018.11.031

Oosugi, N., Kitajo, K., Hasegawa, N., Nagasaka, Y., Okanoya, K., and Fujii, N. (2017). A new method for quantifying the performance of EEG blind source separation algorithms by referencing a simultaneously recorded ECoG signal. *Neural Netw.* 93, 1–6. doi: 10.1016/J.NEUNET.2017.01.005

Ortega, J., Ortiz, A., Martín-Smith, P., Gan, J. Q., and González-Peñalver, J. (2017) "Advances in computational intelligence," in *IWANN 2017. Lecture Notes in Computer Science*, *Vol. 10305*, eds I. Rojas, G. Joya, and A. Catala (Cham: Springer), 28-39. doi: 10.1007/978-3-319-59153-7_3

Ortiz-Echeverri, C. J., Salazar-Colores, S., Rodríguez-Reséndiz, J., and Gómez-Loenzo, R. A. (2019). A new approach for motor imagery classification based on sorted blind source separation, continuous wavelet transform, and convolutional neural network. *Sensors* 19:4541. doi: 10.3390/s19204541

Özerdem, M. S., and Polat, H. (2017). Emotion recognition based on EEG features in movie clips with channel selection. *Brain Inform.* 4, 241–252. doi: 10.1007/s40708-017-0069-3

Padfield, N., Zabalza, J., Zhao, H., Masero, V., Ren, J., Padfield, N., et al. (2019). EEG-based brain-computer interfaces using motor-imagery: techniques and challenges. *Sensors* 19:1423. doi: 10.3390/s19061423

Palaniappan, R., and Revett, K. (2014). PIN generation using EEG: a stability study. *Int. J. Biometrics* 6, 95–105. doi: 10.1504/IJBM.2014.060960

Pan, J., Li, Y., and Wang, J. (2016). "An EEG-Based brain-computer interface for emotion recognition," in *2016 International Joint Conference on Neural Networks (IJCNN)* (Vancouver, BC: IEEE), 2063–2067. doi: 10.1109/IJCNN.2016.7727453

Petrantonakis, P. C., and Hadjileontiadis, L. J. (2010). Emotion recognition from EEG using higher order crossings. *IEEE Trans. Inf. Technol. Biomed.* 14, 186–197. doi: 10.1109/TITB.2009.2034649

Pham, T., Ma, W., Tran, D., Nguyen, P., and Phung, D. (2013). "EEG-based user authentication in multilevel security systems" in *Advanced Data Mining and Applications. ADMA 2013. Lecture Notes in Computer Science, Vol 8347*, eds H. Motoda, Z. Wu., L. Cao, O. Zaiane, M. Yao, and W. Wang (Berlin, Heidelberg: Springer), 513–523. doi: 10.1007/978-3-642-53917-6_46

Picard, R. W. (2003). Affective computing: challenges. *Int. J. Hum. Comput. Stud.* 59, 55–64. doi: 10.1016/S1071-5819(03)00052-1

Pires, G., Nunes, U., and Castelo-Branco, M. (2011). "GIBS block speller: toward a gaze-independent P300-based BCI," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Boston, MA: IEEE), 6360–6364. doi: 10.1109/IEMBS.2011.6091570

Popescu, F., Fazli, S., Badower, Y., Blankertz, B., and Müller, K.-R. (2007). Single trial classification of motor imagination using 6 dry EEG electrodes. *PLoS ONE* 2:e637. doi: 10.1371/journal.pone.0000637

Puanhvuan, D., and Wongsawat, Y. (2012). "Semi-automatic P300-based brain-controlled wheelchair," in *2012 ICME International Conference on Complex Medical Engineering (CME)* (Kobe: IEEE), 455–460. doi: 10.1109/ICCME.2012.6275713

Punsawad, Y., and Wongsawat, Y. (2017). Multi-command SSAEP-based BCI system with training sessions for SSVEP during an eye fatigue state. *IEEJ Trans. Electr. Electron. Eng.* 12, S72–S78. doi: 10.1002/tee.22441

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 257–286. doi: 10.1109/5.18626

Rakotomamonjy, A., Guigue, V., Mallet, G., and Alvarado, V. (2005). "Ensemble of SVMs For Improving Brain Computer Interface P300 Speller Performances," in *Artificial Neural Networks: Biological Inspirations – ICANN 2005. ICANN 2005. Lecture Notes in Computer Science, Vol 3696,* eds W. Duch, J. Kacprzyk, E. Oja and S. Zadrozny (Berlin, Heidelberg: Springer), 45–50. doi: 10.1007/11550822_8

Ramadan, R. A., and Vasilakos, A. V. (2017). Brain computer interface: control signals review. *Neurocomputing* 223, 26–44. doi: 10.1016/j.neucom.2016.10.024

Raschka, S. (2018). *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning.* Available online at: http://arxiv.org/abs/1811.12808 (accessed November 17, 2019).

Rebsamen, B., Burdet, E., Guan, C., Teo, C. L., Zeng, Q., Ang, M., et al. (2007). "Controlling a wheelchair using a BCI with low information transfer rate," in *2007 IEEE 10th International Conference on Rehabilitation Robotics* (Noordwijk: IEEE), 1003–1008. doi: 10.1109/ICORR.2007.4428546

Reddy, T. K., Arora, V., Behera, L., Wang, Y. K., and Lin, C. T. (2019). Multiclass fuzzy time-delay common spatio-spectral patterns with fuzzy information theoretic optimization for EEG-based regression problems in Brain-Computer Interface (BCI). *IEEE Trans. Fuzzy Syst.* 27, 1943–1951. doi: 10.1109/TFUZZ.2019.2892921

Rezeika, A., Benda, M., Stawicki, P., Gembler, F., Saboor, A., and Volosyak, I. (2018). Brain–computer interface spellers: a review. *Brain Sci.* 8:57. doi: 10.3390/brainsci8040057

Rojas, G. M., Alvarez, C., Montoya, C. E., de la Iglesia-Vay,á, M., Cisternas, J. E., and Gálvez, M. (2018). Study of resting-state functional connectivity networks using EEG electrodes position as seed. *Front. Neurosci.* 12:235. doi: 10.3389/fnins.2018.00235

Roy, R., Mahadevappa, M., and Kumar, C. S. (2016). Trajectory path planning of EEG controlled robotic arm using GA. *Procedia Comput. Sci.* 84, 147–151. doi: 10.1016/J.PROCS.2016.04.080

Ruiz Blondet, M. V., Laszlo, S., and Jin, Z. (2015). "Assessment of permanence of non-volitional EEG brainwaves as a biometric," in *IEEE International Conference on Identity, Security and Behavior Analysis (ISBA 2015) (IEEE),* 1–6. doi: 10.1109/ISBA.2015.7126359

Ruiz-Blondet, M. V., Jin, Z., and Laszlo, S. (2016). CEREBRE: a novel method for very high accuracy event-related potential biometric identification. *IEEE Trans. Inf. Forensics Secur.* 11, 1618–1629. doi: 10.1109/TIFS.2016.2543524

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). "Learning internal representations by error propagation," in *Parallel Distributed Processing:*

Explorations in the Microstructure of Cognition. Vol. 1, eds D. E. Rumelhart, J. L. McClelland, and the PDP Research Group (MIT Press).

Rundo, F., Rinella, S., Massimino, S., Coco, M., Fallica, G., Parenti, R., et al. (2019). An innovative deep learning algorithm for drowsiness detection from EEG signal. *Computation* 7:13. doi: 10.3390/computation7010013

Schlögl, A., Lee, F., Bischof, H., and Pfurtscheller, G. (2005). Characterization of four-class motor imagery EEG data for the BCI-competition 2005. *J. Neural Eng.* 2, L14–L22. doi: 10.1088/1741-2560/2/4/L02

Schreuder, M., Höhne, J., Blankertz, B., Haufe, S., Dickhaus, T., and Tangermann, M. (2013). Optimizing event-related potential based brain-computer interfaces: a systematic evaluation of dynamic stopping methods. *J. Neural Eng.* 10:036025. doi: 10.1088/1741-2560/10/3/036025

Serdar Bascil, M., Tesneli, A. Y., and Temurtas, F. (2015). Multi-channel EEG signal feature extraction and pattern recognition on horizontal mental imagination task of 1-D cursor movement for brain computer interface. *Australas. Phys. Eng. Sci. Med.* 38, 229–239. doi: 10.1007/s13246-015-0345-6

Shakeel, A., Navid, M. S., Anwar, M. N., Mazhar, S., Jochumsen, M., and Niazi, I. K. (2015). A review of techniques for detection of movement intention using movement-related cortical potentials. *Comput. Math. Methods Med.* 2015:346217. doi: 10.1155/2015/346217

Shin, B.-G., Kim, T., and Jo, S. (2010). "Non-invasive brain signal interface for a wheelchair navigation," in *International Conference on Control Automation and Systems (ICCAS)* (Gyeonggi-do), 2257–2260. doi: 10.1109/ICCAS.2010.5669830

Shojaedini, S., Morabbi, S., and Keyvanpour, M. (2018). A new method for detecting P300 signals by using deep learning: hyperparameter tuning in high-dimensional space by minimizing nonconvex error function. *J. Med. Signals Sens.* 8, 205–214. doi: 10.4103/jmss.JMSS_7_18

Shyu, K.-K., Chiu, Y.-J., Lee, P.-L., Lee, M.-H., Sie, J.-J., Wu, C.-H., et al. (2013). Total design of an FPGA-based brain–computer interface control hospital bed nursing system. *IEEE Trans. Ind. Electron.* 60, 2731–2739. doi: 10.1109/TIE.2012.2196897

Shyu, K.-K., Lee, P.-L., Lee, M.-H., Lin, M.-H., Lai, R.-J., and Chiu, Y.-J. (2010). Development of a low-cost FPGA-based SSVEP BCI multimedia control system. *IEEE Trans. Biomed. Circ. Syst.* 4, 125–132. doi: 10.1109/TBCAS.2010.2042595

Somers, B., Francart, T., and Bertrand, A. (2018). A generic EEG artifact removal algorithm based on the multi-channel Wiener filter. *J. Neural Eng.* 15:036007. doi: 10.1088/1741-2552/aaac92

Soroush, M. Z., Maghooli, K., Setarehdan, S. K., and Nasrabadi, A. M. (2018). A review on EEG signals based emotion recognition. *Int. Clin. Neurosci. J.* 4, 118–129. doi: 10.15171/icnj.2017.01

Speier, W., Fried, I., and Pouratian, N. (2013). Improved P300 speller performance using electrocorticography, spectral features, and natural language processing. *Clin. Neurophysiol.* 124, 1321–1328. doi: 10.1016/j.clinph.2013.02.002

Sweeney, K. T., Ward, T. E., and McLoone, S. F. (2012). Artifact removal in physiological signals—practices and possibilities. *IEEE Trans. Inf. Technol. Biomed.* 16, 488–500. doi: 10.1109/TITB.2012.2188536

Tabar, Y. R., and Halici, U. (2017). A novel deep learning approach for classification of EEG motor imagery signals. *J. Neural Eng.* 14:016003. doi: 10.1088/1741-2560/14/1/016003

Tang, J., Liu, Y., Hu, D., and Zhou, Z. (2018). Towards BCI-actuated smart wheelchair system. *Biomed. Eng. Online* 17:111. doi: 10.1186/s12938-018-0545-x

Tang, X., Zhou, J., Zhang, N., and Liu, L. (2015). Recognition of motor imagery EEG based on deep belief network. *Inf. Control* 44, 717–721. doi: 10.13976/j.cnki.xk.2015.0717

Tang, Z., Li, C., and Sun, S. (2017). Single-trial EEG classification of motor imagery using deep convolutional neural networks. *Optik* 130, 11–18. doi: 10.1016/J.IJLEO.2016.10.117

Taran, S., and Bajaj, V. (2018). Drowsiness detection using adaptive hermite decomposition and extreme learning machine for electroencephalogram signals. *IEEE Sens. J.* 18, 8855–8862. doi: 10.1109/JSEN.2018.2869775

Tariq, M., Trivailo, P. M., and Simic, M. (2018). EEG-based BCI control schemes for lower-limb assistive-robots. *Front. Hum. Neurosci.* 12:312. doi: 10.3389/fnhum.2018.00312

Tayeb, Z., Fedjaev, J., Ghaboosi, N., Richter, C., Everding, L., Qu, X., et al. (2019). Validating deep neural networks for online decoding of motor imagery movements from EEG signals. *Sensors* 19:210. doi: 10.3390/s19010210

Tello, R. M. G., Müller, S. M. T., Hasan, M. A., Ferreira, A., Krishnan, S., and Bastos, T. F. (2016). An independent-BCI based on SSVEP using Figure-Ground Perception (FGP). *Biomed. Signal Process. Control* 26, 69–79. doi: 10.1016/J.BSPC.2015.12.010

Thulasidas, M., Guan, C., and Wu, J. (2006). Robust classification of EEG signal for brain–computer interface. *IEEE Trans. Neural Syst. Rehabil. Eng.* 14, 24–29. doi: 10.1109/TNSRE.2005.862695

Tian, G., and Liu, Y. (2019). Simple convolutional neural network for left-right hands motor imagery EEG signals classification. *Int. J. Cogn. Informatics Nat. Intell.* 13, 36–49. doi: 10.4018/IJCINI.2019070103

Townsend, G., LaPallo, B. K., Boulay, C. B., Krusienski, D. J., Frye, G. E., Hauser, C. K., et al. (2010). A novel P300-based brain–computer interface stimulus presentation paradigm: moving beyond rows and columns. *Clin. Neurophysiol.* 121, 1109–1120. doi: 10.1016/j.clinph.2010.01.030

Tripathy, R. K., and Rajendra Acharya, U. (2018). Use of features from RR-time series and EEG signals for automated classification of sleep stages in deep neural network framework. *Biocybern. Biomed. Eng.* 38, 890–902. doi: 10.1016/j.bbe.2018.05.005

Tsinalis, O., Matthews, P. M., and Guo, Y. (2016). Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders. *Ann. Biomed. Eng.* 44, 1587–1597. doi: 10.1007/s10439-015-1444-y

Urigüen, J. A., and Garcia-Zapirain, B. (2015). EEG artifact removal—state-of-the-art and guidelines. *J. Neural Eng.* 12:031001. doi: 10.1088/1741-2560/12/3/031001

Valbuena, D., Volosyak, I., and Gräser, A. (2010). "SBCI: fast detection of steady-state visual evoked potentials," *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Buenos Aires: IEEE), 2010:3966–3969. doi: 10.1109/IEMBS.2010.5627990

Van Erp, J. B. F., Lotte, F., and Tangermann, M. (2012). Brain-computer interfaces: beyond medical applications. *Computer* 45, 26–34. doi: 10.1109/MC.2012.107

Vansteensel, M. J., Kristo, G., Aarnoutse, E. J., and Ramsey, N. F. (2017). The brain-computer interface researcher's questionnaire: from research to application. *Brain Comput. Interf.* 4, 236-247. doi: 10.1080/2326263X.2017.1366237

Vareka, L., and Mautner, P. (2017). Stacked autoencoders for the P300 component detection. *Front. Neurosci.* 11:302. doi: 10.3389/fnins.2017.00302

Varona-Moya, S., Velasco-Alvarez, F., Sancha-Ros, S., Fernandez-Rodriguez, A., Blanca, M. J., and Ron-Angevin, R. (2015). "Wheelchair navigation with an audio-cued, two-class motor imagery-based brain-computer interface system," in *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)* (Montpellier: IEEE), 174–177. doi: 10.1109/NER.2015.7146588

Volosyak, I., Cecotti, H., Valbuena, D., and Graser, A. (2009). "Evaluation of the Bremen SSVEP based BCI in real world conditions," in *2009 IEEE International Conference on Rehabilitation Robotics* (Kyoto: IEEE), 322–331. doi: 10.1109/ICORR.2009.5209543

Volosyak, I., Gembler, F., and Stawicki, P. (2017). Age-related differences in SSVEP-based BCI performance. *Neurocomputing* 250, 57–64. doi: 10.1016/j.neucom.2016.08.121

Wang, H., Li, Y., Long, J., Yu, T., and Gu, Z. (2014). An asynchronous wheelchair control by hybrid EEG–EOG brain–computer interface. *Cogn. Neurodyn.* 8, 399-409. doi: 10.1007/s11571-014-9296-y

Wang, L., Lan, Z., Wang, Q., Yang, R., and Li, H. (2019). ELM_kernel and wavelet packet decomposition based EEG classification algorithm. *Autom. Control Comput. Sci.* 53, 452–460. doi: 10.3103/S0146411619050079

Wang, M., Abbass, H. A., Hu, J., and Merrick, K. (2016). "Detecting rare visual and auditory events from EEG using pairwise-comparison neural networks," in *Advances in Brain Inspired Cognitive Systems. BICS 2016. Lecture Notes in Computer Science, Vol. 1002,* eds C. L. Liu, A. Hussain, B. Luo, K.Tan, Y. Zeng, and Z. Zhang (Cham: Springer), 90–101. doi: 10.1007/978-3-319-49685-6_9

Wang, X.-W., Nie, D., and Lu, B.-L. (2014). Emotional state classification from EEG data using machine learning approach. *Neurocomputing* 129, 94–106. doi: 10.1016/J.NEUCOM.2013.06.046

Wang, Y., and Wu, D. (2019). "Deep Learning for sleep stage classification," in *Proceedings 2018 Chinese Automation Congress, CAC 2018* (Xi'an), 3833–3838. doi: 10.1109/CAC.2018.8623637

Wang, Z., Cao, L., Zhang, Z., Gong, X., Sun, Y., and Wang, H. (2018). Short time fourier transformation and deep neural networks for motor imagery brain computer interface recognition. *Concurr. Comput. Pract. Exp.* 30:e4413. doi: 10.1002/cpe.4413

Waytowich, N. R., Lawhern, V., Garcia, J. O., Cummings, J., Faller, J., Sajda, P., et al. (2018). Compact convolutional neural networks for classification of asynchronous steady-state visual evoked potentials. *J. Neural Eng.* 15:aae5d8. doi: 10.1088/1741-2552/aae5d8

Wei, Y., Wu, Y., and Tudor, J. (2017). A real-time wearable emotion detection headband based on EEG measurement. *Sensors Actuators A Phys.* 263, 614–621. doi: 10.1016/J.SNA.2017.07.012

Widyotriatmo, A., Suprijanto, and Andronicus, S. (2015). "A collaborative control of brain computer interface and robotic wheelchair," in *2015 10th Asian Control Conference (ASCC)* (Kota Kinabalu: IEEE), 1–6. doi: 10.1109/ASCC.2015.7244838

Wolpaw, J. R., Birbaumer, N., Mcfarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. *Clin. Neurophysiol.* 113, 767–791. doi: 10.1016/S1388-2457(02)00057-3

Wolpaw, J. R., Loeb, G. E., Allison, B. Z., Donchin, E., Do Nascimento, O. F., Heetderks, W. J., et al. (2006). BCI meeting 2005—workshop on signals and recording methods. *IEEE Trans. Neural Syst. Rehabil. Eng.* 14, 138–141. doi: 10.1109/TNSRE.2006.875583

Wolpaw, J. R., Mcfarland, D. J., Neat, G. W., and Forneris, C. A. (1991). An EEG-based brain-computer interface for cursor control. *Electroencephalogr. Clin. Neurophysiol.* 78, 252–259. doi: 10.1016/0013-4694(91)90040-B

Wu, D., Lance, B. J., Lawhern, V. J., Gordon, S., Jung, T.-P., and Lin, C.-T. (2017). EEG-based user reaction time estimation using Riemannian geometry features. *IEEE Trans. Neural Syst. Rehabil. Eng.* 25, 2157–2168. doi: 10.1109/TNSRE.2017.2699784

Xu, B., Zhang, L., Song, A., Wu, C., Li, W., Zhang, D., et al. (2019). Wavelet transform time-frequency image and convolutional network-based motor imagery EEG classification. *IEEE Access* 7, 6084–6093. doi: 10.1109/ACCESS.2018.2889093

Xu, T., Zhou, Y., Wang, Z., and Peng, Y. (2018). Learning emotions EEG-based recognition and brain activity: a survey study on BCI for intelligent tutoring system. *Procedia Comput. Sci.* 130, 376–382. doi: 10.1016/j.procs.2018.04.056

Xu, X., Chen, X., and Yu, Z. (2018). Removal of muscle artefacts from few-channel EEG recordings based on multivariate empirical mode decomposition and independent vector analysis. *Electron. Lett.* 54, 866–868. doi: 10.1049/el.2018.0191

Xu, Z., Li, J., Gu, R., and Xia, B. (2012). "*Steady-State Visually Evoked Potential (SSVEP)-Based Brain-Computer Interface (BCI): A Low-Delayed Asynchronous Wheelchair Control System* (Berlin; Heidelberg: Springer), 305–314. doi: 10.1007/978-3-642-34475-6_37

Yang, B., Li, H., Wang, Q., and Zhang, Y. (2016). Subject-based feature extraction by using fisher WPD-CSP in brain-computer interfaces. *Comput. Methods Progr. Biomed.* 129, 21–28. doi: 10.1016/j.cmpb.2016.02.020

Yang, C., Wu, H., Li, Z., He, W., Wang, N., and Su, C.-Y. (2018). Mind control of a robotic arm with visual fusion technology. *IEEE Trans. Ind. Informatics* 14, 3822–3830. doi: 10.1109/TII.2017.2785415

Yang, J., Yao, S., and Wang, J. (2018). Deep fusion feature learning network for MI-EEG classification. *IEEE Access* 6, 79050–79059. doi: 10.1109/ACCESS.2018.2877452

Yang, S., Yin, Z., Wang, Y., Zhang, W., Wang, Y., and Zhang, J. (2019). Assessing cognitive mental workload via EEG signals and an ensemble deep learning classifier based on denoising autoencoders. *Comput. Biol. Med.* 109, 159–170. doi: 10.1016/J.COMPBIOMED.2019.04.034

Yeom, S.-K., Fazli, S., and Lee, S.-W. (2014). "P300 visual speller based on random set presentation," in *2014 International Winter Workshop on Brain-Computer Interface (BCI)* (Jeongsun-kun: IEEE), 1–2. doi: 10.1109/iww-BCI.2014.6782567

Yger, F., Berar, M., and Lotte, F. (2017). Riemannian approaches in brain-computer interfaces: a review. *IEEE Trans. Neural Syst. Rehabil. Eng.* 25, 1753–1762. doi: 10.1109/TNSRE.2016.2627016

Yin, Z., and Zhang, J. (2017). Cross-session classification of mental workload levels using EEG and an adaptive deep learning model. *Biomed. Signal Process. Control* 33, 30–47. doi: 10.1016/J.BSPC.2016.11.013

Yin, Z., Zhao, M., Zhang, W., Wang, Y., Wang, Y., and Zhang, J. (2019). Physiological-signal-based mental workload estimation via transfer dynamical autoencoders in a deep learning framework. *Neurocomputing* 347, 212–229. doi: 10.1016/J.NEUCOM.2019.02.061

Zaizu Ilyas, M., Saad, P., and Imran Ahmad, M. (2015). "A survey of analysis and classification of EEG signals for brain-computer interfaces," in *2nd International Conference on Biomedical Engineering (ICoBE)* (Penang), 1–6. doi: 10.1109/ICoBE.2015.7235129

Zarei, R., He, J., Siuly, S., and Zhang, Y. (2017). A PCA aided cross-covariance scheme for discriminative feature extraction from EEG signals. *Comput. Methods Programs Biomed.* doi: 10.1016/j.cmpb.2017.05.009

Zhang, D.-X., Wu, X.-P., and Guo, X. (2008). "The EEG signal preprocessing based on empirical mode decomposition," in *2008 2nd International Conference on Bioinformatics and Biomedical Engineering* (Shanghai: IEEE), 2131–2134. doi: 10.1109/ICBBE.2008.862

Zhang, R., Li, X., Wang, Y., Liu, B., Shi, L., Chen, M., et al. (2019). Using brain network features to increase the classification accuracy of MI-BCI inefficiency subject. *IEEE Access* 7, 74490–74499. doi: 10.1109/ACCESS.2019.2917327

Zhang, R., Li, Y., Yan, Y., Zhang, H., Wu, S., Yu, T., et al. (2016). Control of a wheelchair in an indoor environment based on a brain-computer interface and automated navigation. *IEEE Trans. Neural. Syst. Rehabil. Eng.* 24, 128–39. doi: 10.1109/TNSRE.2015.2439298

Zhang, R., Wang, Q., Li, K., He, S., Qin, S., Feng, Z., et al. (2017). A BCI-based environmental control system for patients with severe spinal cord injuries. *IEEE Trans. Biomed. Eng.* 64, 1959–1971. doi: 10.1109/TBME.2016.2628861

Zhang, T., Zheng, W., Cui, Z., Zong, Y., and Li, Y. (2019). Spatial–temporal recurrent neural network for emotion recognition. *IEEE Trans. Cybern.* 49, 839–847. doi: 10.1109/TCYB.2017.2788081

Zhang, X., Yao, L., Wang, X., Monaghan, J., Mcalpine, D., and Zhang, Y. (2019). *A Survey on Deep Learning based Brain Computer Interface: Recent Advances and New Frontiers.* Available online at: http://arxiv.org/abs/1905.04149 (accessed July 15, 2019).

Zhang, Y., Guo, D., Xu, P., Zhang, Y., and Yao, D. (2016). Robust frequency recognition for SSVEP-based BCI with temporally local multivariate synchronization index. *Cogn. Neurodyn.* 10, 505–511. doi: 10.1007/s11571-016-9398-9

Zhang, Y., Xiaomin, J., and Zhang, Y. (2015). "Classification of EEG signals based on AR model and approximate entropy," in *2015 International Joint Conference on Neural Networks (IJCNN)* (Killarney: IEEE), 1–6. doi: 10.1109/IJCNN.2015.7280840

Zhang, Y., Yin, E., Li, F., Zhang, Y., Guo, D., Yao, D., et al. (2019). Hierarchical feature fusion framework for frequency recognition in SSVEP-based BCIs. *Neural Netw.* 119, 1–9. doi: 10.1016/j.neunet.2019.07.007

Zhang, Y., Yin, E., Li, F., Zhang, Y., Tanaka, T., Zhao, Q., et al. (2018). Two-stage frequency recognition method based on correlated component analysis for SSVEP-based BCI. *IEEE Trans. Neural Syst. Rehabil. Eng.* 26, 1314–1323. doi: 10.1109/TNSRE.2018.2848222

Zhang, Z., Duan, F., Sole-Casals, J., Dinares-Ferran, J., Cichocki, A., Yang, Z., et al. (2019). A novel deep learning approach with data augmentation to classify motor imagery signals. *IEEE Access* 7, 15945–15954. doi: 10.1109/ACCESS.2019.2895133

Zheng, W. L., and Lu, B. L. (2015). Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. *IEEE Trans. Auton. Ment. Dev.* 7, 162–175. doi: 10.1109/TAMD.2015.2431497

Zhou, J., Meng, M., Gao, Y., Ma, Y., and Zhang, Q. (2018). "Classification of motor imagery EEG using wavelet envelope analysis and LSTM networks," in *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018* (Shenyang: Institute of Electrical and Electronics Engineers Inc.), 5600–5605. doi: 10.1109/CCDC.2018.8408108

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: frontiersin.org/about/contact

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership