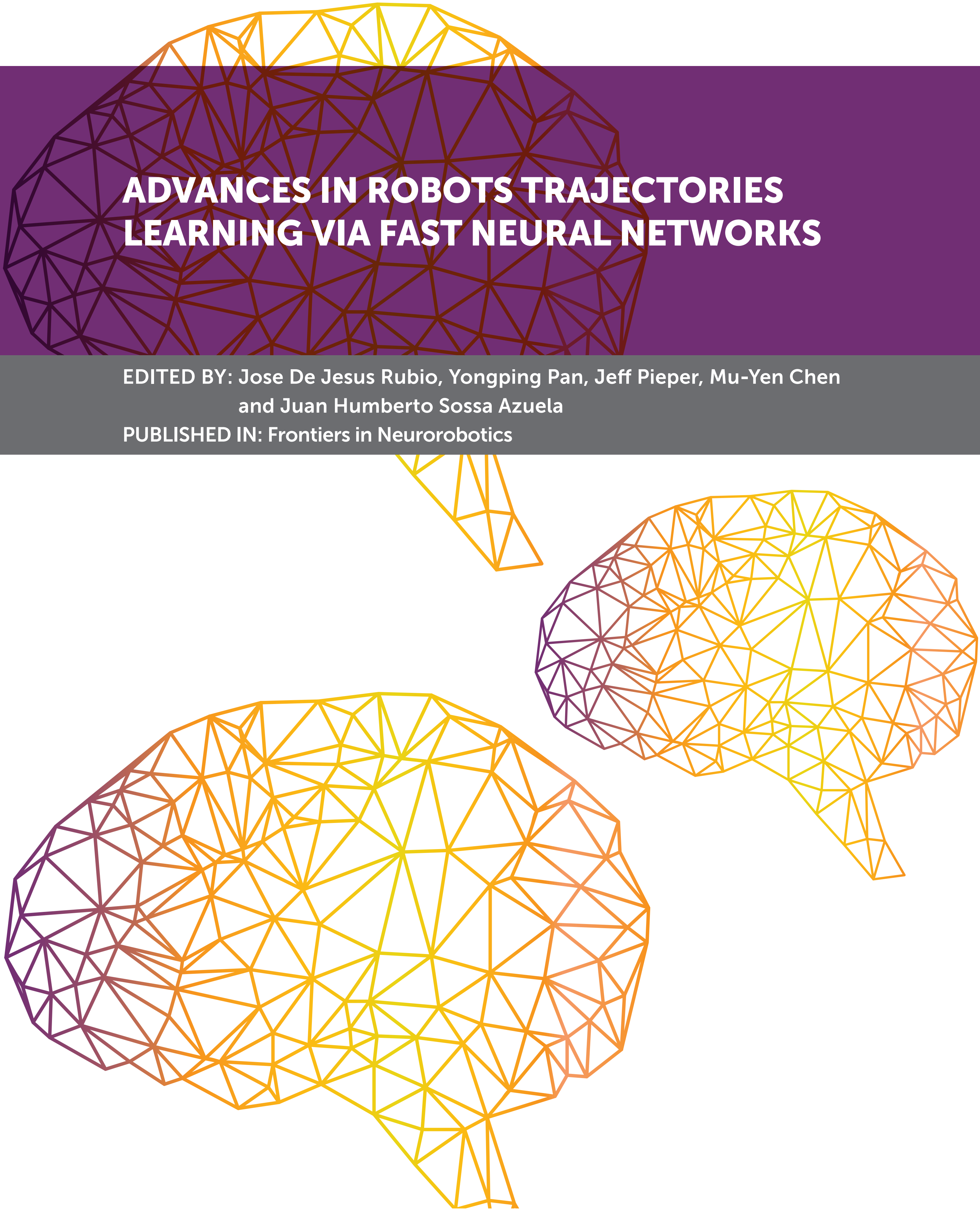
ADVANCES IN ROBOTS TRAJECTORIES LEARNING VIA FAST NEURAL NETWORKS



EDITED BY: Jose De Jesus Rubio, Yongping Pan, Jeff Pieper, Mu-Yen Chen
and Juan Humberto Sossa Azuela

PUBLISHED IN: Frontiers in Neurorobotics

frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-88966-768-0
DOI 10.3389/978-2-88966-768-0

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

ADVANCES IN ROBOTS TRAJECTORIES LEARNING VIA FAST NEURAL NETWORKS

Topic Editors:
Jose De Jesus Rubio, Instituto Politécnico Nacional (IPN), Mexico
Yongping Pan, National University of Singapore, Singapore
Jeff Pieper, University of Calgary, Canada
Mu-Yen Chen, National Cheng Kung University, Taiwan
Juan Humberto Sossa Azuela, Instituto Politécnico Nacional (IPN), Mexico

Citation: De Jesus Rubio, J., Pan, Y., Pieper, J., Chen, M.-Y., Azuela, J. H. S., eds. (2021). Advances in Robots Trajectories Learning via Fast Neural Networks. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88966-768-0

Table of Contents

frontiers in Neurorobotics

EDITORIAL
published: 22 March 2021
doi: 10.3389/fnbot.2021.671519

Editorial: Advances in Robots Trajectories Learning via Fast Neural Networks

Jose de Jesus Rubio [1]*, *Yongping Pan* [2], *Jeff Pieper* [3], *Mu-Yen Chen* [4] *and Juan Humberto Sossa Azuela* [5]

[1] Sección de Estudios de Posgrado e Investigación, ESIME Azcapotzalco, Instituto Politécnico Nacional, Ciudad de Mexico, Mexico, [2] Department of Biomedical Engineering, National University of Singapore, Singapore, Singapore, [3] Department of Mechanical and Manufacturing Engineering, University of Calgary, Calgary, AB, Canada, [4] Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan, [5] Laboratorio de Robótica y Mecatrónica, Centro de Investigación en Computación, Instituto Politécnico Nacional, Ciudad de Mexico, Mexico

Keywords: robots, neural networks, control, trajectory, learning

Editorial on the Research Topic

Advances in Robots Trajectories Learning via Fast Neural Networks

Motion planning, also known as the navigation problem, is a term used in robotics to find a sequence of valid configurations that moves the robot from the source to a destination. From **Figure 1**, a robot trajectory may be specified as a sequence of discrete points of a temporal sequence. For this Research Topic, we hope to see focused manuscripts that use artificial intelligence for robots to learn how to develop a trajectory specifically using fast neural networks. Fast neural networks are algorithms which accelerate trajectory learning in robots. Some examples of fast neural networks, and their variants, are long short-term memory, convolutional neural network, recurrent neural network, deep deterministic policy gradient, cascade neural network, genetic algorithm, machine learning, and fuzzy model. The goal of this Research Topic is to welcome research on different types of robots that use fast neural networks to learn and retrieve trajectories in order to perform a positioning task of a robot and contribute solutions to the navigation problems. This Research Topic collected twelve high quality papers reporting the performance results related to some of the previously mentioned emerging research directions how to realize fast neural networks for robots trajectory learning.

The paper titled "Using Long Short-Term Memory for Building Outdoor Agricultural Machinery" by Wu et al. describes an outdoor agricultural robot that uses Long Short-Term Memory (LSTM). The key features of this innovation consider that the robot is portable and uses green power to reduce installation cost, the system combines the current environment with weather forecasts through LSTM to predict the correct timing for watering, and the robot is mainly for outdoor applications.

The paper titled "Quantum-Based Creative Generation Method for a Dancing Robot" by Mei et al. introduces a creative generation process model based on the quantum modeling simulation method. This model is mainly aimed at generating the running trajectory of a dancing robot and the execution plan of the dancing action. They use digital twin technology to establish the robot trajectory and the dance movements, they use regions with convolutional neural networks to extract character bones and movement features to form a movement library, and the system then render scenes that match the actions through generative adversarial networks.

Edited and reviewed by:
Jae-Ho Han,
Korea University, South Korea

****Correspondence:***
Jose de Jesus Rubio
rubio.josedejesus@gmail.com

Received: *24 February 2021*
Accepted: *01 March 2021*
Published: *22 March 2021*

Citation:
Rubio JdJ, Pan Y, Pieper J, Chen M-Y and Sossa Azuela JH (2021) Editorial: Advances in Robots Trajectories Learning via Fast Neural Networks. Front. Neurorobot. 15:671519. doi: 10.3389/fnbot.2021.671519

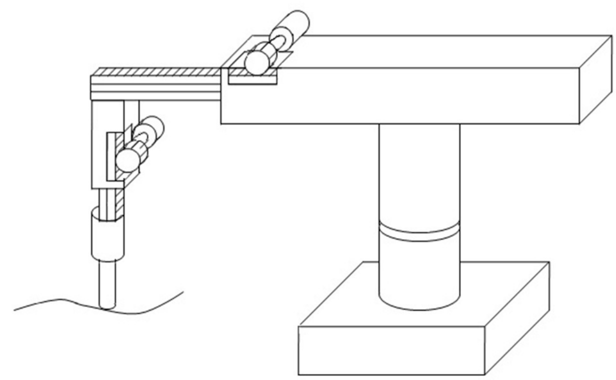


FIGURE 1 | A robot trajectory.

The paper titled "Bi-criteria Acceleration Level Obstacle Avoidance of Redundant Manipulator" by Zhao et al. presents a recurrent neural network based neural dynamic solver for the improved obstacle-avoidance-scheme-based kinematic control problem in acceleration level for a redundant robot manipulator. The distance between the manipulator and an obstacle is described as the point-to-point distance, and the collision avoidance strategy is formulated as an inequality. From the perspective of optimization, therefore, an acceleration level quadratic programming (QP) problem is formulated to solve the resultant QP minimization problem.

The paper titled "The Path Planning of Mobile Robot by Neural Networks and Hierarchical Reinforcement Learning" by Yu et al. proposes a Deep Deterministic Policy Gradient (DDPG) as a hierarchical reinforcement learning of neural networks for the path planning of mobile robots. Specifically, when compare with Double Deep Q-Learning (DDQN), DDPG has a shorter path planning time and a reduced number of path steps.

The paper titled "PD Control Compensation Based on a Cascade Neural Network Applied to a Robot Manipulator" by Soriano et al. introduces a nominal control law to achieve a sub-optimal performance, and implements a scheme based on a cascade neural network to act as a non-linear compensation of a two-degree-of-freedom robot manipulator. The main contributions of this work are neural compensation based on a cascade neural networks and the function to update the weights.

The paper titled "A Custom EOG-Based HMI Using Neural Network Modeling to Real-Time for the Trajectory Tracking of a Manipulator Robot" by Reynoso et al. considers the generation of points in a Cartesian space (X, Y, Z) in order to control a manipulator robot that follows a desired trajectory by means of the movement of the user eyeball. For this purpose, a multilayer neural network (MNN) is used to model the EOG signal as a mathematical function, which is optimized using genetic algorithms, and the machine learning is customized for the classification in order to reduce the domain time of the system without the need of a database.

The paper titled "CNN Based Detectors on Planetary Environments: A Performance Evaluation" by Furlán et al. presents a convolutional neural network algorithm and an exploration robot for the detection of rocks in environments similar to Mars. The methodology proposed here is based on the use of a Single-Shot-Detector (SSD) network architecture, which has been modified to detect rocks in planetary images.

The paper titled "Optimal UAV's Deployment and Transmit Power Design for Two Users Uplink NOMA Systems" by Zhao describes a simplified setup with two ground users to draw some insightful results in the UAV deployment location. Authors formulate an optimization problem employing the Karush-Kuhn-Tucher (KKT) conditions that maximizes the sum throughput subject to each user transmit power constraint.

The paper titled "Adoption of Machine Learning Algorithm-Based Intelligent Basketball Training Robot in Athlete Injury Prevention" by Xu and Tang proposes a machine learning-based improved Q-Learning algorithm for the path planning obstacle avoidance in an intelligent robot. First, combined with the basketball motion trajectory model, the sport recognition in basketball training was analyzed. Second, the mathematical model of the basketball motion trajectory of the shooting motion is established, and the factors affecting the shooting are analyzed.

The paper titled "Intelligent Badminton Training Robot in Athlete Injury Prevention based on Machine Learning" by Xie et al. presents a machine learning algorithm to explore the role of intelligent badminton training robot (IBTR) in the prevention of badminton player injuries. An IBTR is designed from the perspectives of hardware and software systems, and the movements of the athletes are recognized and analyzed with the hidden Markov model (HMM) in the machine learning.

The paper titled "The Industrial Robot based on Machine Vision and Artificial Intelligence in 5G Environment" by Jin et al. introduces a visual recognition system of industrial robots based on improved Fast R-CNN target detection model. The image layers are convolved and pooled through the deep learning model of artificial intelligence.

To end up, the paper titled "The Analysis of Trajectory Control of Nonholonomic Mobile Robots based on Internet of Things Target Image Enhancement Technology and Backpropagation Neural Network" by Zhao proposes the trajectory tracking and control of incomplete mobile robots. First, the mathematical kinematics model of the nonholonomic mobile robot is studied. Then, the improved Backpropagation Neural Network (BPNN) based on combining the fuzzy algorithm and the neural network is applied to the robot controller.

We conclude this editorial by expressing our sincere gratitude and appreciation to the Specialty Chief Editors of Frontier in Neurorobotics, Alois C. Knoll and Florian Röhrbein, for their great support throughout the compilation of this Research Topic, for their guidance when proposing this Research Topic. Last but not least, we thank the authors for their contributions to this Research Topic and to all reviewers for their voluntary contribution in the peer-review process to maintain a high standard of our Research Topic.

AUTHOR CONTRIBUTIONS

JR, YP, and JP: writing—original draft. M-YC and JS: review and editing. All authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGMENTS

JR and JS thank the Instituto Politecnico Nacional, the Secretaria de Investigacion y Posgrado, the Comision de Operacion y Fomento de Actividades Academicas, and the Consejo Nacional de Ciencia y Tecnologia for their help in this research.

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

ORIGINAL RESEARCH
published: 29 May 2020
doi: 10.3389/fnbot.2020.00027

Using Long Short-Term Memory for Building Outdoor Agricultural Machinery

Chien-Hung Wu[1], *Chun-Yi Lu*[2], *Jun-We Zhan*[3] *and Hsin-Te Wu*[4]*

[1] *Department of Marine Recreation, National Penghu University of Science and Technology, Magong, Taiwan,* [2] *Department of Information Management, National Penghu University of Science and Technology, Magong, Taiwan,* [3] *Department of Computer Science and Information Engineering, National Penghu University of Science and Technology, Magong, Taiwan,* [4] *Department of Computer Science and Information Engineering, National Ilan University, Yilan City, Taiwan*

Today, climate change has caused a decrease in agricultural output or overall yields that are not as expected; however, with the ongoing population explosion, many undeveloped countries have transformed into emerging countries and have transformed farmland to be used in other types of applications. The resulting decline in agricultural output further increases the severity of the food crisis. In this context, this study proposes an outdoor agricultural robot that uses Long Short-Term Memory (LSTM). The key features of this innovation include: (1) the robot is portable, and it uses green power to reduce installation cost, (2) the system combines the current environment with weather forecasts through LSTM to predict the correct timing for watering, (3) detecting the environment and utilizing information from weather forecasts can help the system to ensure that growing conditions are suitable for the crops, and (4) the robot is mainly for outdoor applications because such farms lack sufficient electricity and water resources, which makes the robot critical for environmental control and resource allocation. The experimental results indicate that the robot developed in this study can detect the environment effectively to control electricity and water resources. Additionally, because the system is planned to increase agricultural output significantly, the study predicts the variables through multivariate LSTM, which controls the power supply from the solar power system.

Keywords: artificial intelligence, robot, deep learning, intelligent agriculture, automation equipment, long short-term memory

OPEN ACCESS

Edited by:
Mu-Yen Chen,
National Taichung University of Science and Technology, Taiwan

Reviewed by:
Pei-Wei Tsai,
Swinburne University of Technology, Australia
Tsung-Ju Lee,
Feng Chia University, Taiwan

****Correspondence:***
Hsin-Te Wu
hsinte@niu.edu.tw

Received: *28 February 2019*
Accepted: *16 April 2020*
Published: *29 May 2020*

Citation:
Wu C-H, Lu C-Y, Zhan J-W and Wu H-T (2020) Using Long Short-Term Memory for Building Outdoor Agricultural Machinery. Front. Neurorobot. 14:27. doi: 10.3389/fnbot.2020.00027

1. INTRODUCTION

The world is currently facing energy and food crises; moreover, many countries are transforming farmland into industrial land for related usage because they are transforming from being undeveloped to being emerging countries. All of these factors have reduced agricultural output tremendously. The literature (Liu et al., 2018) has mentioned that rapid economic development and urbanization are consuming the resources of the planet extremely rapidly and that, especially due to the impact of climate change, the yields of many crops are declining or are not as expected. Therefore, the huge challenge arises of rethinking our exploitation of food and energy. On the other hand, traditional agriculture requires labor and machinery to irrigate and harvest; yet, with the impact of an aging agricultural labor population, a large amount of farmland is deserted or fallow. Consequently, agriculture needs to be upgraded to intelligent production, which could increase output while reducing labor because farmers would then only need to calibrate the smart equipment and set relevant parameters.

Precision Agriculture (PA) works to detect relevant environmental information around the farmland to enhance automated production; the PA system controls automated machinery and the related irrigation equipment while farmers will only need to calibrate the equipment and confirm correctness; this will reduce agricultural labor and ensure a good crop-growing environment Narvaez et al. (2017). Hu et al. (2019) utilized an intelligent agriculture system to handle the tasks of irrigation and production, as well as doing the watering work according to the environmental conditions of the farmland to manage the growing environment. In another study Ayaz et al. (2019), the farmland environment was detected via the Internet of Things (IoT), which supports equipment, such as lighting and watering equipment, to make sure that the crops grow in a suitable environment. Furthermore, the article pointed out that for farmland in extreme climate areas, there are difficulties over a huge amount of farmland in planting proper crops to grow in such an environment; thus, the development of smart agriculture can re-analyze farmland environments to find suitable species of crops for the farmers. Finally, a study by Chebrolu et al. (2018) used unmanned aerial vehicles to check and monitor farms; these can detect the environment and ease the agricultural labor issue.

Based on the aforementioned issues, this study offers an approach for building outdoor agricultural machinery that utilizes LSTM. Additionally, because most farmland is located beyond the reach of power equipment or where there are not enough water resources, this study proposes a system that uses solar power as the key electricity supply and makes the equipment movable. Due to the limited electricity offered by the solar power system, the approach detects the equipment and conducts watering tasks by LSTM to avoid energy waste, the detectors used in the experiment monitor the soil temperature and humidity, pH value, and sunlight conditions in the farmland. The study also collects forecast information, such as temperature and the Probability of Precipitation, from the Central Weather Bureau and uses this to predict soil humidity and sunlight conditions via LSTM. The robot will forecast the best time to activate the watering equipment, and when the prediction result exceeds the set suitable conditions, the server will notify the farmer. This approach is an application designed for outdoor usage and is shown to be practical on a farm. The experimental result proves that the proposed method is feasible; moreover, the equipment presented is cost-efficient, which makes it well-suited to widespread distribution and massive adoption for general applications.

2. RELATED WORK

To date, many studies have offered intelligent agriculture-based approaches for improving the quality of crops. A study by Liu et al. (2018) mainly used natural resources to support farm planting; for example, it used overproduced energy to provide watering and lighting or utilized solar energy and rainwater to take care of the farm. Using natural resources can significantly reduce resource waste and achieve eco-friendliness. Narvaez et al. (2017) used spectral data to monitor the growing conditions of crops and decide whether to trim or spray pesticides. Hu et al. (2019) utilized wireless sensor networks to transfer detector data and reduce installation cost as well as using the network to lower the energy consumption of the detectors. Ayaz et al. (2019), meanwhile, detailed the structures of intelligent agriculture performed through IoT and cloud networks. IoT equipment supports the delivery of the environmental data collected to a cloud server, which enables agricultural experts to analyze data for decision-making. Two other studies, Chebrolu et al. (2018) and Farooq et al. (2019) utilized unmanned aerial vehicles to monitor farmland regularly to detect the growing conditions of the crops; additionally, Shadrin et al. (2019) used a Convolutional Neural Network to judge whether the growing conditions had reached expectations, and Bayrakdar (2019) combined detectors with wireless sensor networks to check whether there were holes underneath the farmland and used sonar to determine whether the area was suffering from plant diseases and pest damage. The method presented in this article can detect the growing environment of the crop regularly to ensure the quality of the crop.

Because Internet equipment cannot be robustly installed in outdoor farms, it is better to develop a Wi-Fi system to transmit the data and ensure delivery quality. A study by Chen and Yang (2019) offers an intelligent agriculture method to improve growing for farmers significantly by using relevant data to build decision systems or Knowledge-Based Systems. Lozoya et al. (2016) constructed modular agriculture that helps diverse farms to adjust system modules based on their requirements, which ensures an optimal automated process. Herrera et al. (2016) combine four-wheeled vehicles with sensors to tour around the farm; with fixed routes, the vehicle would collect the growth information of each selected part and transfer the data back for analysis. The method presented by Murugan et al. (2017) can be used in latifundios (large farms) because it uses aerial shots to identify crop maturity; this method can substantially reduce agricultural labor pressures. Another study, Elijah et al. (2018) developed a wireless sensor network system for farmland that can collect farm information effectively; with data analytics, the system could activate the equipment rapidly and ensure that crops were growing under the most suitable environment. In Liu et al. (2019) study combining IoT and cloud computing to record the relevant environmental factors of the farm, the crop quality was increased because the system protected the farm from pollution. Lin et al. (2020) used R-CNN to conduct pest control and detection; instead, this study suggests using a 4G network to improve transmission quality.

Palangi et al. (2016) searched keywords online through LSTM; due to the massive amount of data on the Internet, LSTM helps find the required files rapidly by utilizing the concept of keywords. Wang et al. (2018) examined the production yield of wafers via bilateral LSTM. Park et al. (2018) mainly used LSTM to develop keyword recognition in speech recognition for drivers; because the level of device computation available in a car is usually low, keywords are required to boost searches to fulfill the needs of drivers. Zhou et al. (2019) presented a system that predicts sunlight and estimates the electricity produced by solar power systems through LSTM, which becomes the basis of solar power equipment management. Finally, Zhang et al. (2019) utilized LSTM in edge computing; when video files are too large, an LSTM technique helps predict the buffer memory for each edge computing, node. Guardo et al. (2018) mainly used Fog Computing to reduce the workload of centralized servers. Additionally, Arachchi et al. (2019) scheduled video time arrangements via LSTM, which is beneficial for composing various videos chronologically. The current article uses the LSTM technique to detect the experimental environment and effectively control water resources and electrical power conditions.

In this study, outdoor agricultural machinery was built that has a solar power system to provide electricity because outdoor farms usually lack sufficient electricity and water resources. Due to limited solar power energy, for achieving the performance of monitoring the farm environment and maintaining good growing conditions, the experiment used LSTM to monitor the environment and control the watering system. It combined information on sunlight, soil humidity, temperature, and weather forecasts from the Central Weather Bureau with LSTM to set system schedules and avoid wastage of electricity and water resources.

3. THE PROPOSED SCHEME

This chapter introduces the following details: the system model; signal delivery and control; data normalization; long short-term memory prediction.

3.1. System Model

The article proposes the system model shown in **Figure 1**. The equipment includes lighting and sensors that detect the barometric pressure, soil temperature and humidity, and pH value in an outdoor farm, and also a sprinkler motor. We attached an IoT development board to the machinery (I_0) that uses stored electricity from the solar power system; the sensors on the equipment deliver the collected data to the server (S_0) via 4G networks. Furthermore, the server updates the weather forecast data through a web crawler and combines this with the sensor data to conduct LSTM analysis. The LSTM system will set watering schedules based on the analytic results to ensure that the growing environment provides a suitable humidity and temperature for the crops. On the other hand, outdoor farmland usually lacks sufficient water resources and electricity; in particular, the installation cost of electrical equipment is a huge burden for farmers. Therefore, this research presents a solar power system for reducing the cost stress; additionally, the designed equipment is movable to any location on the farm, which also reduces the workload of farmers. The system uses LSTM to predict soil conditions and weather changes, the equipment sets the schedules for activating the machinery, and the design saves electricity to avoid waste; all of these improve the growing environment for crops.

The research developed an outdoor robot for detecting environmental conditions and controlling the watering system. The robot transmits the data to the server for further analysis, and the server can use environmental factors and weather forecasts to predict the environmental conditions. Different kinds of crops grow under different conditions; hence, through the analysis, the result can help to judge whether the environmental conditions are beyond expectations. If the growing conditions are not suitable for growing the crop, the system will warn the farmers. Moreover, because weather data exhibits linear growth, the study applies LSTM to calculate environmental variables so as to predict the environmental factors and conditions of the farmland to increase output.

3.2. Signal Delivery and Control

This study uses Extensible Messaging and Presence Protocol (XMPP) to collect information from the sensors on the machine and uses LSTM to predict soil temperature and humidity so as to judge the activation time of the equipment. As shown in **Figure 2**, the machinery and the IoT development board will maintain a low power consumption status. When the LSTM system suggests starting the equipment, the server will send a signal to the IoT development board to activate the sprinkler system and the sensor. When the task is finished, the IoT development board will also send a signal back to the server and provide the sensor

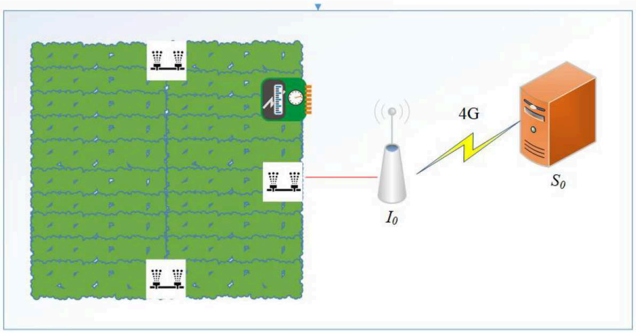


FIGURE 1 | System model.

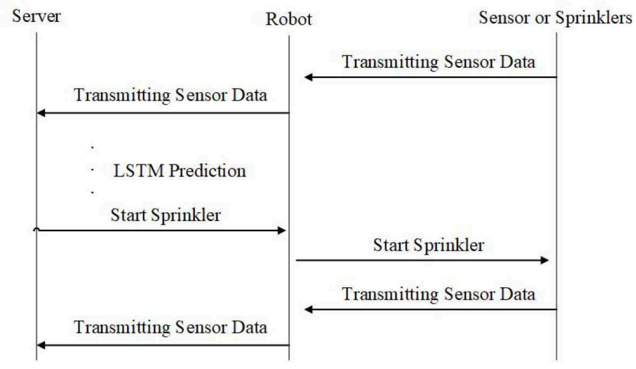


FIGURE 2 | Signal delivery and control model.

data for analysis. Due to the low power consumption, the IoT development board can avoid wastage of electricity; hence, when the server receives the sensor data, it will further conduct LSTM prediction analysis and set the next schedule for activation.

3.3. Data Normalization

The machinery with sensors can detect soil temperature and humidity, sunlight, and related information, which will be delivered to the server through 4G networks. Nonetheless, packet loss might occasionally occur, causing data error, and insects might also cause the sensors to detect incorrect information; thus, it is necessary to judge the data correctness. Firstly, to check the correctness of the detected temperature, we input the lowest and highest temperatures from the Central Weather Bureau and check the current time to judge the rationality. The algorithm is shown below:

Algorithm 1: Temperature detection algorithm.

```
//If the value of the current temperature minus the estimated temperature is smaller than a set threshold.
if ($|Te_{ct} - PTe_s|$) $< Threshold$ then
    Correct
else
    Re-detect
end if
```

Te_{ct} means the current soil temperature detected from the sensor, and PTe_s is the temperature loaded from the Central Weather Bureau. If the temperature between 06:00 and 17:00 is set to be the highest temperature, the temperature during the rest of the day will be set to be the lowest temperature. The threshold presents the defined value; the detected temperature is correct if the difference is smaller than the threshold; otherwise, the system will re-detect the temperature.

Regarding humidity, the study uses the evaporation equation presented by Priestley (1970) to judge whether the soil evaporation fulfills the prediction. Moreover, we combine weather forecast data to check whether it will rain. The evaporation equation is $ET_0 = \alpha S\,(Rn - G)\,/\,(s + \gamma)$, where Rn is the net radiation, α is the equilibrium evaporation parameter, G is the sensible heat flux, and γ is the humidity. A large evaporation value means that the soil humidity is low, and the condition is normal. The study uses weather forecast information to judge whether the sensor is detecting correctly and further improve the correctness of LSTM.

3.4. Long Short-Term Memory Prediction

The study conducts Multivariate LSTM to manage the watering and detect the farm conditions. Firstly, the experiment judges whether the targeted crop can grow under the weather conditions as forecasted; if not (for example, when the temperature is too low or too high or when it will rain heavily), the system will send an alarm to notify the farmer. The equation is shown below:

Algorithm 2: Farm Environmental Detection algorithm.

```
//If the temperature is suitable for growing the targeted crop.
if ($ATe < PTe_h$ and $ATe > PTe_l$) then
    Normal
    if (QPF $< Threshold$) then
        Normal
    else
        Alarm
    end if
end if
```

The aforementioned equation checks whether the growing temperature (ATe) is set between the predicted highest (PTe_h) and lowest temperatures (PTe_l); if not, the system will notify the farmer to take care of the issue. On the other hand, the system will also confirm whether the Quantitative Precipitation Forecast (QDF) is higher than the threshold; if so, this means that the rain will be too heavy and will damage the crops.

The study utilizes soil humidity, temperature, the weather forecasted temperature, UV index, and QDF, and the calculated evaporation to make a watering prediction. Firstly, the evaporation equation, $ET_0 = \alpha S\,(Rn - G)\,/\,(s + \gamma)$, is used to calculate the evaporation of the soil. The second step is to calculate the temperature curve. Because time and temperature are proportional, assuming that 12:00 to 14:00 is the period of the highest temperature, the percentage for that period is set as 100%; as the temperature decreases from 14:00 to 17:00, the percentages would be 70, 40, and 10%, and as the temperature increases between 07:00 and 12:00, the percentages would be 20, 40, 60, 80, and 100%. Therefore, the slopes of the current and predicted temperatures can be calculated from the equation $m = \Delta y/\Delta x$, where Δy is the time percentage of the current temperature and Δx is the predicted time percentage of the estimated temperature. After the calculation, using Multivariate LSTM for further prediction, the variable is calculated by the equation $x_i = \{Hu_{ct}, m, QDF, ET_0, UV\}$, where Hu_{ct} is the soil humidity and UV is the UV index. x_i is entered to conduct the LSTM analysis, and tanh is using to transfer the output value into a number between 1 and -1, as shown in **Figure 3**. According to the predicted time and the output value, the system will judge whether it is necessary to activate the sprinkler and environmental detection equipment. The system offered in this research can reduce electricity waste significantly and control water resources effectively to ensure a good growing environment for the crops. Building outdoor agricultural machinery based on LSTM enables farmers to set the desired watering system and environmental management parameters according to different crops, which can improve agricultural output and reduce agricultural labor use.

4. PERFORMANCE

The performance of the system is described in sections Experiment Results of the System's Functions and LSTM Experimental Results.

4.1. Experiment Results of the System's Functions

The experimental area used in this research is an outdoor farm, as shown in **Figure 4**. The experiment uses a solar power system to store energy because there is no other source of electricity at the farm. The area is 0.2 hectares, and the equipment used is pictured in **Figure 5**; the machinery can be flexibly moved around the farmland, and the hardware and software equipped on the system are listed in **Table 1**. The machinery connects with the sensors and the development board through IoT; there are various types of sensors, such as barometric pressure and light sensors on the machinery, as well as a mini pumping motor, a 3-in-1 soil tester, and a solar power system. We set up a remote server that not only collects sensor data but also conducts LSTM analysis; the server further transfers the analytic results into readable signals for judging whether the system should activate the sprinkler system and initiate the environmental monitor.

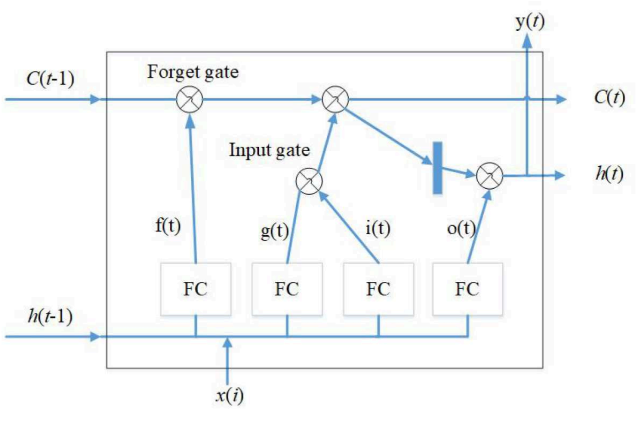


FIGURE 3 | LSTM diagram.

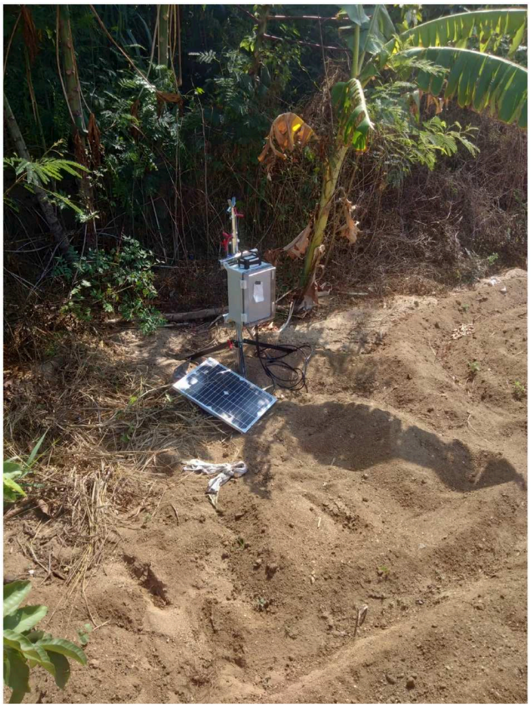


FIGURE 4 | Experimental area.

4.2. LSTM Experimental Results

The study utilized LSTM to monitor the farm and select the timing for watering. **Figure 6** shows the curves from the XMPP server and **Figure 7** pictures the activation of the sprinkler system after the signals have been judged. The data from the Central Weather Bureau and the detected data from the environment were combined for a further prediction. The result is shown in **Figure 8**, where the blue line is the actual data, the orange line is the prediction result after training, and the green line is the result of the test data. In **Figure 8**, the number of samples is shown in the X-axis, while the Y-axis represents the soil humidity. The experimental results prove that the prediction approach offered in this study is extremely accurate. The LSTM prediction value is between 0 and 1, as **Figure 9** shows; the blue line represents the training sample, the orange line shows the test sample, the X-axis is the number of samples, and the Y-axis is the prediction value. The threshold value is set to 0.4, and the sprinkler system will be activated when the value is lower than 0.4.

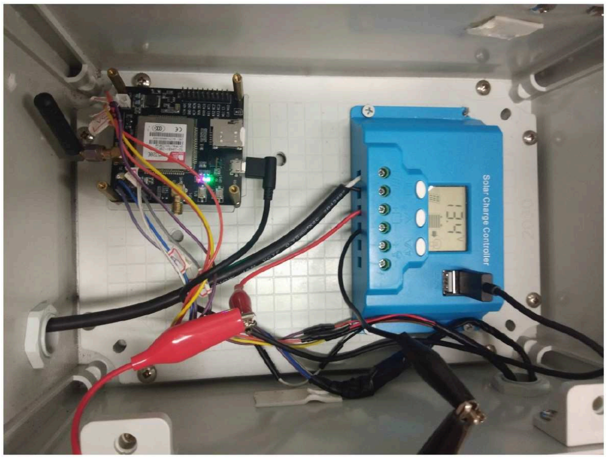


FIGURE 5 | Experimental hardware.

TABLE 1 | The software and hardware the proposed system.

Hardware	Software
IoT Development Board	XMPP Platform
Barometric Pressure Sensor	Python
Light Sensor	MySQL Server
Mini Pumping Motor	Windows 10
3-in-1 Soil Tester	
Solar Power System	

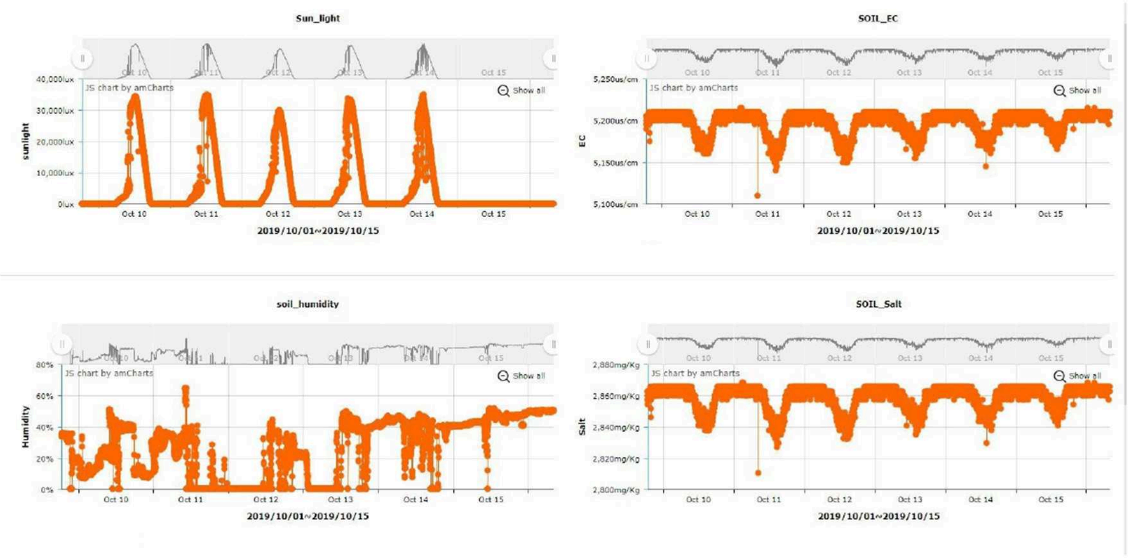


FIGURE 6 | XMPP platform.

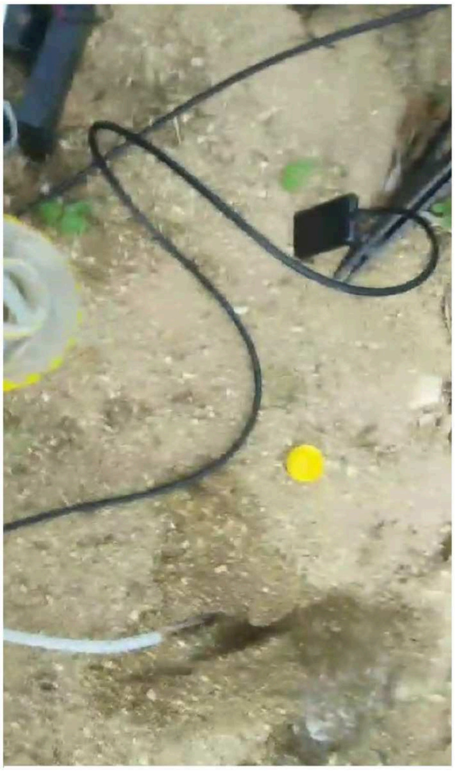


FIGURE 7 | Activation of the sprinkler systems.

5. CONCLUSIONS

This study built outdoor agricultural machinery incorporating LSTM; the system can carry out watering automatically and

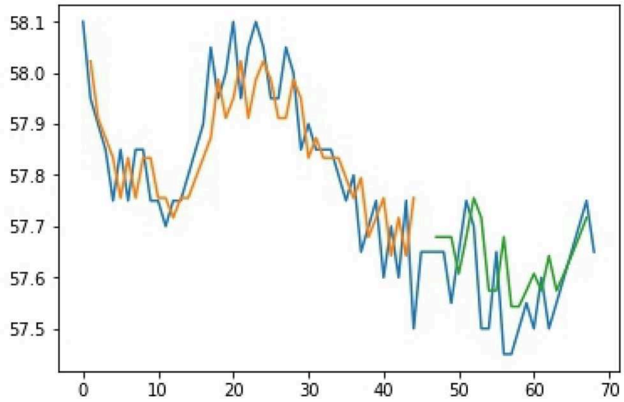


FIGURE 8 | Prediction result of humidity.

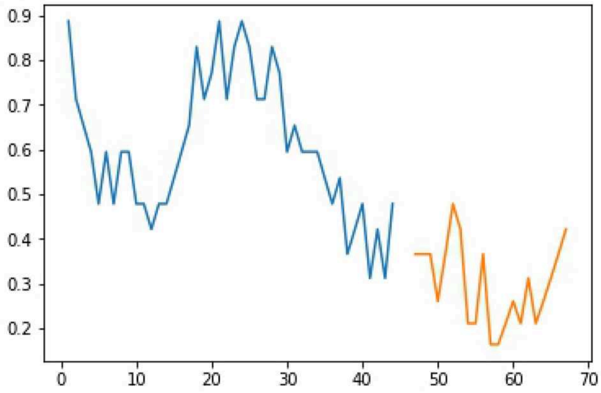


FIGURE 9 | Prediction result of LSTM.

monitor the farm conditions, fulfilling the purpose of an intelligent machine. The approach presented in this study can achieve the purpose of green energy effectively and reduce the waste of water resources. Through the implementation of LSTM, the system can analyze and predict the best timing for watering, which can avoid wastage of stored solar power and ensure an optimal growth environment for crops. Furthermore, the method can resolve the problem of an aging agricultural labor population and improve production output. The study successfully accomplishes the following functions. (1) The LSTM technique can predict the temperature and humidity of the soil precisely. (2) The suggested approach uses a solar power system and stores the electricity in a battery to reduce the workload of the power installation on the farm. (3) Through an IoT platform, the presented method is capable of predicting the best time to initiate the watering function and control water resources effectively. (4) The system presented in this study is mainly for outdoor application; the experimental results have proved that practical usage of the equipment is feasible. The experimental results show that the proposed approach was practical under testing; the LSTM experimental data demonstrated decent prediction performance. In the future, the authors aim to conduct a further study on monitoring multiple farms with a single server, which is expected to reduce the installation costs for farmers and enable the commercialization of the relevant equipment.

DATA AVAILABILITY STATEMENT

The datasets presented in this article are not readily available because the data also forms part of an ongoing study. Requests to access the datasets should be directed to hsinte@niu.edu.tw.

AUTHOR CONTRIBUTIONS

J-WZ proposed the experiment, compiled the data, and performed most of the analysis. C-HW and C-YL wrote most of the main text and contributed to the theoretical analysis and implications of the work. H-TW contributed to the analysis and discussion of the data. C-HW contributed to the theoretical analysis and implications of the work. All authors contributed ideas, discussed the results, and wrote the manuscript.

REFERENCES

Arachchi, S. P. K., Shih, T. K., Lin, C.-Y., and Wijayarathna, G. (2019). Deep learning-based firework video pattern classification. *J. Internet Technol.* 20, 2033–2042. doi: 10.3966/160792642019122007002

Ayaz, M., Ammad-Uddin, M., Sharif, Z., Mansour, A., and Aggoune, E.-H. M. (2019). Internet-of-things (iot)-based smart agriculture: toward making the fields talk. *IEEE Access* 7, 129551–129583. doi: 10.1109/ACCESS.2019.2932609

Bayrakdar, M. E. (2019). A smart insect pest detection technique with qualified underground wireless sensor nodes for precision agriculture. *IEEE Sens. J.* 19, 10892–10897. doi: 10.1109/JSEN.2019.2931816

Chebrolu, N., Labe, T., and Stachniss, C. (2018). Robust long-term registration of UAV images of crop fields for precision agriculture. *IEEE Robot. Automat. Lett.* 3, 3097–3104. doi: 10.1109/LRA.2018.2849603

Chen, J., and Yang, A. (2019). Intelligent agriculture and its key technologies based on internet of things architecture. *IEEE Access* 7, 77134–77141. doi: 10.1109/ACCESS.2019.2921391

Elijah, O., Rahman, T. A., Orikumhi, I., Leow, C. Y., and Hindia, M. N. (2018). An overview of internet of things (IoT) and data analytics in agriculture: benefits and challenges. *IEEE Internet Things J.* 5, 3758–3773. doi: 10.1109/JIOT.2018.2844296

Farooq, M. S., Riaz, S., Abid, A., Abid, K., and Naeem, M. A. (2019). A survey on the role of iot in agriculture for the implementation of smart farming. *IEEE Access* 7, 156237–156271. doi: 10.1109/ACCESS.2019.2949703

Guardo, E., Stefano, A. D., Corte, A. L., Sapienza, M., and Scata, M. (2018). A fog computing-based iot framework for precision agriculture. *J. Internet Technol.* 19, 1401–1411. doi: 10.3966/160792642018091905012

Herrera, D., Tosetti, S., and Carelli, R. (2016). Dynamic modeling and identification of an agriculture autonomous vehicle. *IEEE Latin Am. Trans.* 14, 2631–2637. doi: 10.1109/TLA.2016.7555230

Hu, Z., Xu, L., Cao, L., Liu, S., Luo, Z., Wang, J., et al. (2019). Application of non-orthogonal multiple access in wireless sensor networks for smart agriculture. *IEEE Access* 7, 87582–87592. doi: 10.1109/ACCESS.2019.2924917

Lin, T.-L., Chang, H.-Y., and Chen, K.-H. (2020). The pest and disease identification in the growth of sweet peppers using faster R-CNN and mask R-CNN. *J. Internet Technol.* 21, 605–614. doi: 10.3966/160792642020032102027

Liu, J., Chai, Y., Xiang, Y., Zhang, X., Gou, S., and Liu, Y. (2018). Clean energy consumption of power systems towards smart agriculture: roadmap, bottlenecks and technologies. *CSEE J. Power Energy Syst.* 4, 273–282. doi: 10.17775/CSEEJPES.2017.01290

Liu, S., Guo, L., Webb, H., Ya, X., and Chang, X. (2019). Internet of things monitoring system of modern eco-agriculture based on cloud computing. *IEEE Access* 7, 37050–37058. doi: 10.1109/ACCESS.2019.2903720

Lozoya, C., Aguilar, A., and Mendoza, C. (2016). Service oriented design approach for a precision agriculture datalogger. *IEEE Latin Am. Trans.* 14, 1683–1688. doi: 10.1109/TLA.2016.7483501

Murugan, D., Garg, A., and Singh, D. (2017). Development of an adaptive approach for precision agriculture monitoring with drone and satellite data. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* 10, 5322–5328. doi: 10.1109/JSTARS.2017.2746185

Narvaez, F. Y., Reina, G., Torres-Torriti, M., Kantor, G., and Cheein, F. A. (2017). A survey of ranging and imaging techniques for precision agriculture phenotyping. *IEEE/ASME Trans. Mechatron.* 22, 2428–2439. doi: 10.1109/TMECH.2017.2760866

Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., et al. (2016). Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval. *IEEE/ACM Trans. Audio Speech Lang. Process.* 24, 694–707. doi: 10.1109/TASLP.2016.2520371

Park, J., Son, H., Lee, J., and Choi, J. (2018). Driving assistant companion with voice interface using long short-term memory networks. *IEEE Trans. Indus. Inform.* 15, 582–590. doi: 10.1109/TII.2018.2861739

Priestley, J. (1970). *Autobiography of Joseph Priestley*, ed J. Lindsay. Teaneck, NJ: Fairleigh Dickinson University Press.

Shadrin, D., Menshchikov, A., Ermilov, D., and Somov, A. (2019). Designing future precision agriculture: detection of seeds germination using artificial intelligence on a low-power embedded system. *IEEE Sens. J.* 19, 11573–11582. doi: 10.1109/JSEN.2019.2935812

Wang, J., Zhang, J., and Wang, X. (2018). Bilateral lstm: A two-dimensional long short-term memory model with multiply memory units for short-term cycle time forecasting in re-entrant manufacturing systems. *IEEE Trans. Indus. Inform*. 14, 748–758. doi: 10.1109/TII.2017.2754641

Zhang, C., Pang, H., Liu, J., Tang, S., Zhang, R., Wang, D., et al. (2019). Toward edge-assisted video content intelligent caching with long short-term memory learning. *IEEE Access* 7, 152832–152846. doi: 10.1109/ACCESS.2019.2947067

Zhou, H., Zhang, Y., Yang, L., Liu, Q., Yan, K., and Du, Y. (2019). Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism. *IEEE Access* 7, 78063–78074. doi: 10.1109/ACCESS.2019.2923006

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

ORIGINAL RESEARCH
published: 29 September 2020
doi: 10.3389/fnbot.2020.578834

A Custom EOG-Based HMI Using Neural Network Modeling to Real-Time for the Trajectory Tracking of a Manipulator Robot

Francisco D. Perez Reynoso[1], Paola A. Niño Suarez[1], Oscar F. Aviles Sanchez[2], María B. Calva Yañez[3], Eduardo Vega Alvarado[3] and Edgar A. Portilla Flores[3]*

[1] Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, Mexico City, Mexico, [2] Departamento de Ingeniería Mecatrónica, Universidad Militar Nueva Granada, Bogotá, Colombia, [3] Centro de Innovación Tecnológica en Computo, Instituto Politécnico Nacional, Mexico City, Mexico

Although different physiological signals, such as electrooculography (EOG) have been widely used in the control of assistance systems for people with disabilities, customizing the signal classification system remains a challenge. In most interfaces, the user must adapt to the classification parameters, although ideally the systems must adapt to the user parameters. Therefore, in this work the use of a multilayer neural network (MNN) to model the EOG signal as a mathematical function is presented, which is optimized using genetic algorithms, in order to obtain the maximum and minimum amplitude threshold of the EOG signal of each person to calibrate the designed interface. The problem of the variation of the voltage threshold of the physiological signals is addressed by means of an intelligent calibration performed every 3 min; if an assistance system is not calibrated, it loses functionality. Artificial intelligence techniques, such as machine learning and fuzzy logic are used for classification of the EOG signal, but they need calibration parameters that are obtained through databases generated through prior user training, depending on the effectiveness of the algorithm, the learning curve, and the response time of the system. In this work, by optimizing the parameters of the EOG signal, the classification is customized and the domain time of the system is reduced without the need for a database and the training time of the user is minimized, significantly reducing the time of the learning curve. The results are implemented in an HMI for the generation of points in a Cartesian space (X, Y, Z) in order to control a manipulator robot that follows a desired trajectory by means of the movement of the user's eyeball.

Keywords: EOG, HMI, customization calibration, MNN, optimization, robots trajectories

OPEN ACCESS

Edited by:
Jeff Pieper,
University of Calgary, Canada

Reviewed by:
Luis Arturo Soriano,
National Polytechnic Institute of Mexico (IPN), Mexico
Dante Mujica-Vargas,
Centro Nacional de Investigación y Desarrollo Tecnológico, Mexico
Genaro Ochoa,
Instituto Tecnológico Superior de Tierra Blanca, Mexico

****Correspondence:***
Paola A. Niño Suarez
pninos@ipn.mx

Received: *01 July 2020*
Accepted: *18 August 2020*
Published: *29 September 2020*

Citation:
Perez Reynoso FD, Niño Suarez PA, Aviles Sanchez OF, Calva Yañez MB, Vega Alvarado E and Portilla Flores EA (2020) A Custom EOG-Based HMI Using Neural Network Modeling to Real-Time for the Trajectory Tracking of a Manipulator Robot. Front. Neurorobot. 14:578834. doi: 10.3389/fnbot.2020.578834

INTRODUCTION

The development of human–machine Interfaces (HMI) has been on the rise due to the incorporation of physiological signals as inputs to the control algorithms. Currently, robots are collaborative and interact with humans to improve their quality of life, which has allowed the development of intuitive interfaces for human–robot collaboration, in tasks, such as assistance and robotic rehabilitation. One of the study objectives in these systems is shared control, where a robotic system and human control the same body, tool, mechanism, etc. Shared control has originated in research fields, such as human–robot co-adaptation, where the two agents can benefit by each other's skills or must adapt to the other's behavior, to achieve the execution of effective cooperative tasks.

In this paper, it was considered that the human and individual characteristics affect the execution of the task that the HMI perform; these parameters are highly variable, and it is required to analyze and reduce the effects on the efficiency of the system. It is difficult to determine the level of adaptability or personalization of an HMI; however, calibrating a system looking for it to adapt to the personal parameters of a user has been shown to decrease the learning curve, improving the level of acceptance of inexperienced users. The proposed HMI will be implemented in the future to assist people with severe disabilities, where a manipulator robot will be adapted to a wheelchair, so that the user can control the movements of the robot by means of orientation of the gaze with the ability of taking objects and increasing their autonomy.

The work presented proposes to develop an intelligent calibration system to personalize the use of an HMI, where using EOG signals controls the trajectory tracking of a manipulator robot in its workspace. To achieve this, a fuzzy inference system is calibrated using the EOG signal of each user. The individual EOG signal was modeled by means of an MNN, implementing descending backpropagation using the Widrow–Hoff technique, obtaining a mathematical function that describes the waveform of the signal discrete EOG. The objective function obtained by means of the neural network is optimized using genetic algorithms to obtain the maximum and minimum voltage threshold of the EOG signal corresponding to each person. Once the variability range is obtained by optimizing the EOG signal, the fuzzy classifier is calibrated for the generation of coordinates in the Cartesian space (X, Y, Z). Gaussian membership functions define position in space by detecting EOG signal voltage thresholds; each threshold corresponds to a point in space defined precisely by calibration for each individual. In this case, a database is not required for the system to work; in most interfaces, they have a set of signals stored, and through training the user it is expected to reach the expected values, which only then does the system respond.

In section Overview of Related Work of this document, a summary of related works is presented; section Materials and Methods provides an overview of the neural network for non-linear regression of discrete EOG signal samples and details of the method used to implement the calibration system using genetic algorithms. The experimental procedure and analysis of results are presented in section Experiments and Results Analysis, and section Conclusion concludes the current work and discusses the advantages and limitations of the proposed system.

OVERVIEW OF RELATED WORK

People with severe disabilities cannot move their lower and upper extremities, so designing interfaces with custom features has become a technological challenge (Lum et al., 2012); for this reason, controllers have been implemented that can adapt to the needs of the user using haptic algorithms, multimodal human–machine interfaces (mHMI) and incorporation of artificial intelligence algorithms (Dipietro et al., 2005) among others. In Gopinathan et al. (2017), a study is presented that describes the physical human–robot interaction (pHRI) using a custom rigidity control system of a 7-DOF KUKA industrial robot; the system is calibrated using a force profile obtained through each user and validates their performance by 49 participants using a heuristic control. A similar control system is applied in Buchli et al. (2011), where the level of force of each user is adapted to the control of a 3-DOF robot by haptics and is adjusted to the biomechanics of the user, in order to work on cooperative environments with humans (Gopinathan et al., 2017).

To customize assistive systems, Brain–Computer Interface (BCI) systems have also been developed in combination with electroencephalography (EEG), electromyography (EMG), and electrooculography (EOG) signals (Ang et al., 2015). In Zhang et al. (2019), a multimodal system (mHMI) is presented that can achieve a classification accuracy of physiological signals with an average of 93.83%, which is equivalent to a control speed of 17 actions per minute; the disadvantage that it presents is the long training time and the excessive use of sensors placed on the user. In Rozo et al. (2015), Gaussian functions are used to classify and learn cooperative human–robot skills in the context of object transport. In Medina et al. (2011), a method is proposed using Markov models to increase the experience of a manipulator robot in collaborative tasks with humans; the control adapts and improves cooperation through user speech commands and repetitive haptic training tasks.

The disadvantages of handling EEG for the development of Brain–Computer Interfaces (BCI) are described in Xiao and Ding (2013), since EEG signals do not have sufficient resolution because they attenuate during transmission; however, detection is reported in the EEG bandwidth using artificial intelligence that decodes individual finger movements for the control of prostheses. Advanced methods have been used for the detection, processing, and classification of EMG signals generated by muscle movements. In Gray et al. (2012), a comprehensive review was conducted on the changes that occur in the muscle after clinical alterations and how it affects the characteristics of the EMG signal, emphasizing the adaptability of the signal classification due to muscle injuries.

In the case of wheelchair control in Djeha et al. (2017), they use wavelet transform and an MNN for the classification of EOG and EEG signals, obtaining a classification accuracy rate of 93%; the classifier works in a control system for a virtual wheelchair. In Kumar et al. (2018), a review of human–computer interface systems based on EOG is presented; the work of 41 authors is explained, where the interfaces used implement artificial intelligence algorithms for signal classification. To calibrate the classifier, they use databases that contain an average of the signal threshold; they are characterized by implementing pattern search algorithms so that the machines designed to provide assistance have a response.

The HMI system presented in this paper has the property of being calibrated in real time, so it can be adapted to an EOG signal of any user, without the need for a database, unlike most of the systems reported in Kumar et al. (2018). The HMI works with any inexperienced user because of its capability of adapting to personal characteristics after a brief training. This is mainly due to the use of a calibration system designed from a multilayer neural network (MNN) to model the EOG signal as a mathematical function. The proposal in this work is that the user is not the one that adapts to previously acquired signals to generate a response in the system and that the system is the one that adapts to personal parameters of any user with severe motor disability. The preliminary results, obtained with 60 different users without disabilities in order to measure the adaptability of the system, showed that it was possible to generate trajectories to control a robot by means of ocular commands.

MATERIALS AND METHODS

The developed system is presented in **Figure 1**; there is an analog acquisition stage of the EOG signal and two parallel processing of the signal. One is for classification, where the EOG signal is divided by means of voltage thresholds, and a fuzzy inference system is implemented to establish the relationship between the EOG signal and the workspace of an assistance robot. The classification using fuzzy logic requires a working threshold to generate points in the Cartesian space; these data represent the desired position to which a manipulating robot must arrive.

The other is for calibration of the fuzzy inference system. The proposed method is to obtain a mathematical model that describes the behavior of the EOG signal for each individual, and an algorithm that detects the optimal thresholds with which the classifier can be modified and adapted to any user. Customizing the control of the assistance system reduces time training necessary for mastering it. Thus, the objective function for each individual is obtained by optimizing the range of signal variability. These data are the input to the fuzzy classifier, adapting the interface to the personal properties of each user.

The proposed methodology for the development of the HMI consists of parallel processing, while acquisition digital processing and classification by means of fuzzy inference is carried out with a time from the generation of the eye movement to the articular movement of the robot of 1.1 s. Calibration consisting of modeling and optimization of the EOG signal is carried out every 3 min. When the optimal range data is obtained, it is transmitted through a virtual port, communicating the optimization results with the fuzzy classifier. So the range of the classifier is constantly updating, adapting to changes in either the signal, due to user changes, or in the variability of the voltage threshold due to fatigue and clinical alterations.

EOG Acquisition

By generating an eye movement through the direct central position toward the periphery, the retina approaches an electrode while the cornea approaches the electrode on the opposite side. This change in the orientation of the dipole is reflected as a change in the amplitude and polarity of the EOG signal (**Figure 2**) so that by registering these changes the movement of the eyeball can be determined. EOG signals have been determined to show amplitudes ranging from 5 to 20 μV per degree of displacement, with a bandwidth between 0 and 50 Hz (Lu et al., 2018).

The EOG signal is obtained using two pairs of electrodes connected near the eyes, plus a reference electrode on the forehead and another to eliminate muscle noise in the earlobe, thus generating two channels that record horizontal movement and vertical of the eyeball. In total, six silver/silver chloride electrodes are connected (Ag/AgCl), as presented in **Figure 3A**.

For the acquisition of the reliable EOG signal, an analog processing stage was designed, which includes amplification, isolation, and filtering for each channel (horizontal and vertical) and was complemented by a digital filtering module. The pre-amplification and amplification stage has a 100-dB CMRR, an analog low-pass filter in Butterworth configuration of 40 dB/decade, and a capacitive isolation system for user safety. The designed acquisition system is embedded on a PCB board placed in portable glasses (**Figure 3B**), to provide security and comfort to the user.

To remove the D.C. level, an integrator circuit is used for feedback of the EOG signal at the reference terminal of the instrumentation amplifier. It acts as a high-pass filter preventing instrumentation amplifiers from being saturated. The muscle signal is considered as noise, and it does not allow obtaining a good interpretation of the EOG signal. To eliminate it, the output of the common-mode circuit of the instrumentation amplifier is connected to the earlobe through an electrode for return noise of the muscle signal at the input of the amplifier, thus subtracting the noise signal of the EOG signal affected by noise. Additionally, the electrode placed on the user's forehead is connected to the isolated ground of the circuit. Through these connections, the D.C. component, generated by involuntary movements and poor electrode connection, is eliminated.

In summary, each type of noise is eliminated and the additive noise is eliminated by means of an integrating circuit that works as a 0.1-Hz high-pass filter that eliminates the DC component that is added to the EOG signal. Impulsive noise caused by muscle movement is eliminated by a common rejection mode circuit connected to the earlobe that is fed back to the instrumentation amplifier in its differential configuration. Due to this property, this noise is subtracted and eliminated. The multiplicative noise is eliminated by means of a second-order digital Notch filter tunable in real time on the device's test platform.

For digital processing of the obtained EOG signal, the horizontal and vertical channels were connected to the differential voltage input of a DAQ6009 acquisition card that communicates with a PC through a USB port of a 25-s sample. The DAQ6009 card is used for the acquisition of the EOG signal because it has a maximum input frequency of 5 MHz; the electrooculography has a bandwidth of DC at 50 Hz, so for the purposes of this work the sample frequency is ideal, complying with the Nyquist sampling theorem. This theorem indicates that the exact reconstruction of a continuous periodic signal from its samples is mathematically possible if the signal is band-limited and the sampling rate is more than double its bandwidth.

In **Figure 4A**, the waveform of the EOG signal of a user is observed when the movement of the gaze to the right and left is performed. This acquisition is done in 25-s time

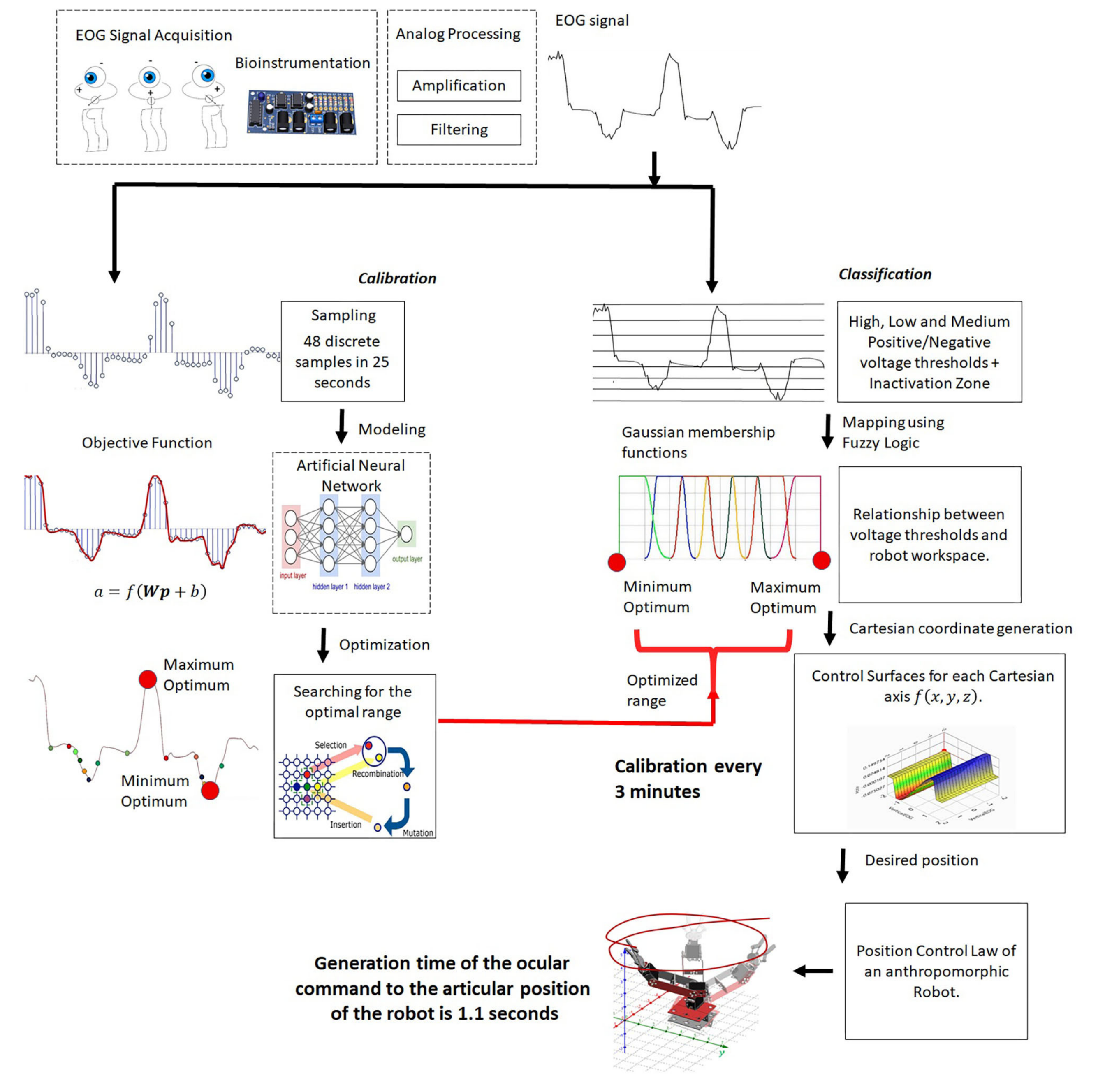

EOG Signal Acquisition
Bioinstrumentation
Analog Processing
Amplification
Filtering
EOG signal
Calibration
Sampling 48 discrete samples in 25 seconds
Modeling
Objective Function
Artificial Neural Network
input layer
hidden layer 1
hidden layer 2
output layer
$a = f(\boldsymbol{Wp} + b)$
Optimization
Maximum Optimum
Minimum Optimum
Searching for the optimal range
Selection
Recombination
Insertion
Mutation
Classification
High, Low and Medium Positive/Negative voltage thresholds + Inactivation Zone
Mapping using Fuzzy Logic
Gaussian membership functions
Relationship between voltage thresholds and robot workspace.
Minimum Optimum
Maximum Optimum
Cartesian coordinate generation
Optimized range
Control Surfaces for each Cartesian axis $f(x, y, z)$.
Calibration every 3 minutes
Desired position
Position Control Law of an anthropomorphic Robot.
Generation time of the ocular command to the articular position of the robot is 1.1 seconds

FIGURE 1 | HMI system.

windows generating 48 discrete samples. This signal is digitized by convolution as a function of time with a Dirac pulse train at a frequency of 100 Hz (Equation 1), and the result of signal sampling is presented in **Figure 4B**.

$$x_p(t) = \sum x[nT]\delta(t - nT) \tag{1}$$

The EOG signal (**Figure 4A**) is the input to the fuzzy classifier to generate points according to the workspace of an assistance system that can be a mobile robot, a robotic arm, or a cursor on the screen.

The nature of the EOG signal behavior is non-linear, there is no pattern, and thresholds vary from one individual to another; if this signal is used as input to an HMI system, the classification system must be calibrated for each user or recalibrated if there is a disturbance in the environment. Assistive systems controlled by physiological signals regularly use a database for the system to generate a response to a particular signal; in this type of case, the user must have a training that makes their eye movements

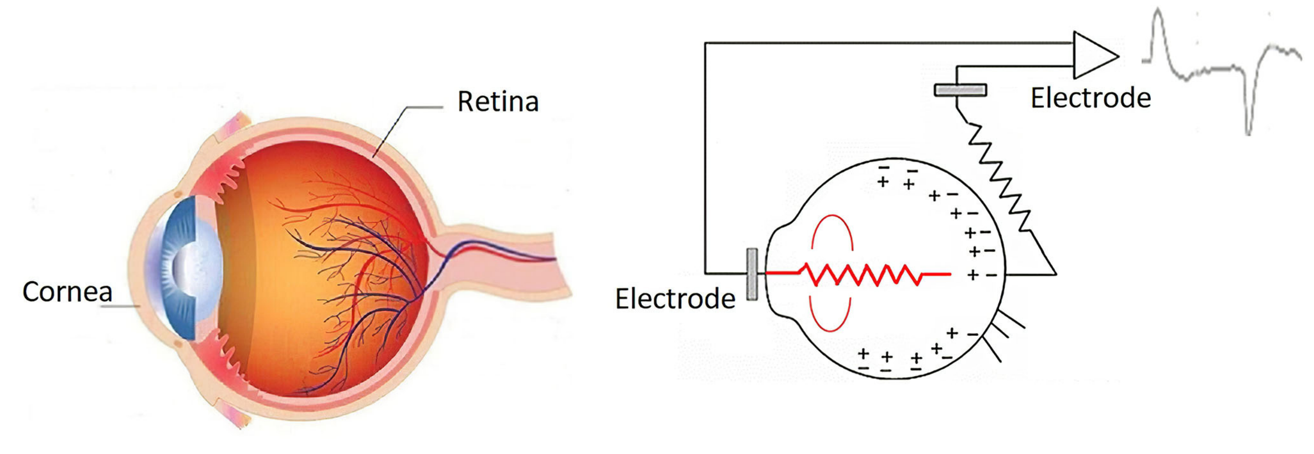


FIGURE 2 | Retina–cornea action potential (Ding and Lv, 2020).

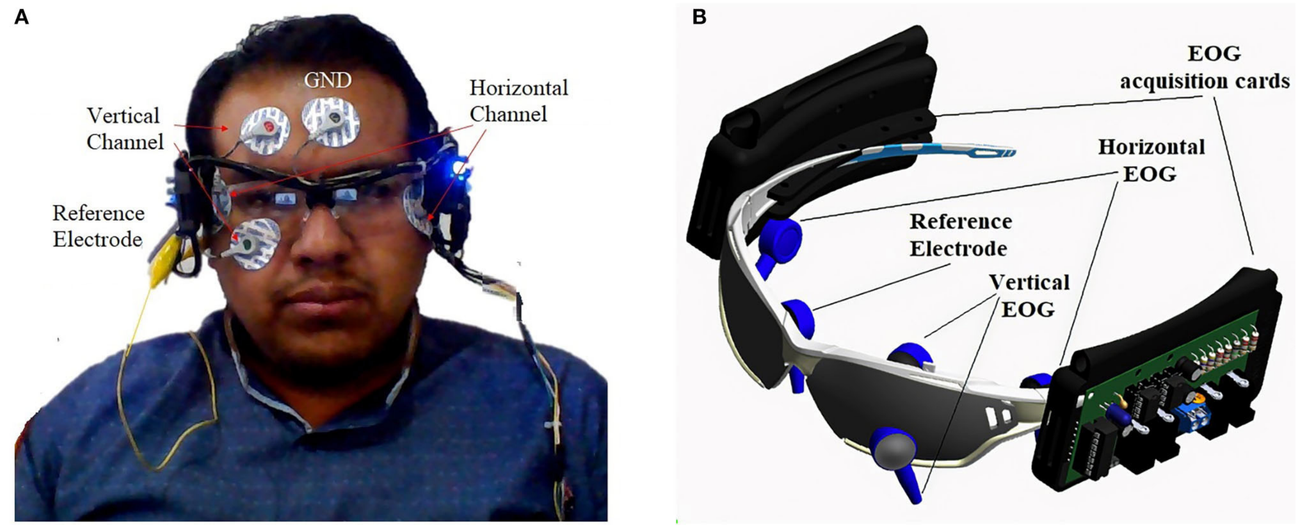


FIGURE 3 | (A) Placement of the electrodes. **(B)** Portable EOG signal acquisition and instrumentation system.

generate a signal similar to those stored in the database, thus generating a longer response time in the system. In this case, a database or previous training will not be necessary, because a process of the discrete samples (**Figure 4B**) of the EOG signal is performed in parallel, which are the input to the MNN designed to perform the interpolation of the discrete data in order to calibrate the system. The objective is to obtain the maximum and minimum values of the voltage threshold; this range is important because it delimits the operation of the fuzzy inference system.

In the next section, the design of the intelligent calibration system is explained first, followed by the operation of the fuzzy inference system.

Intelligent Calibration System

Due to the need to determine the working threshold of the fuzzy classifier for each person in this section, the modeling of the EOG signal is presented, which allows obtaining the required values of the optimal operating range of the fuzzy inference system. First, the mathematical model of the signal is obtained by means of an MNN; the result of this stage provides an objective function. Then, using genetic algorithms, the voltage thresholds were calculated which, without falling into local values, represent the maximum and minimum values of the signal amplitude when the user guides the gaze. Finally, the custom EOG signal is classified based on its optimal range. This data is sent as the user's optimal thresholds.

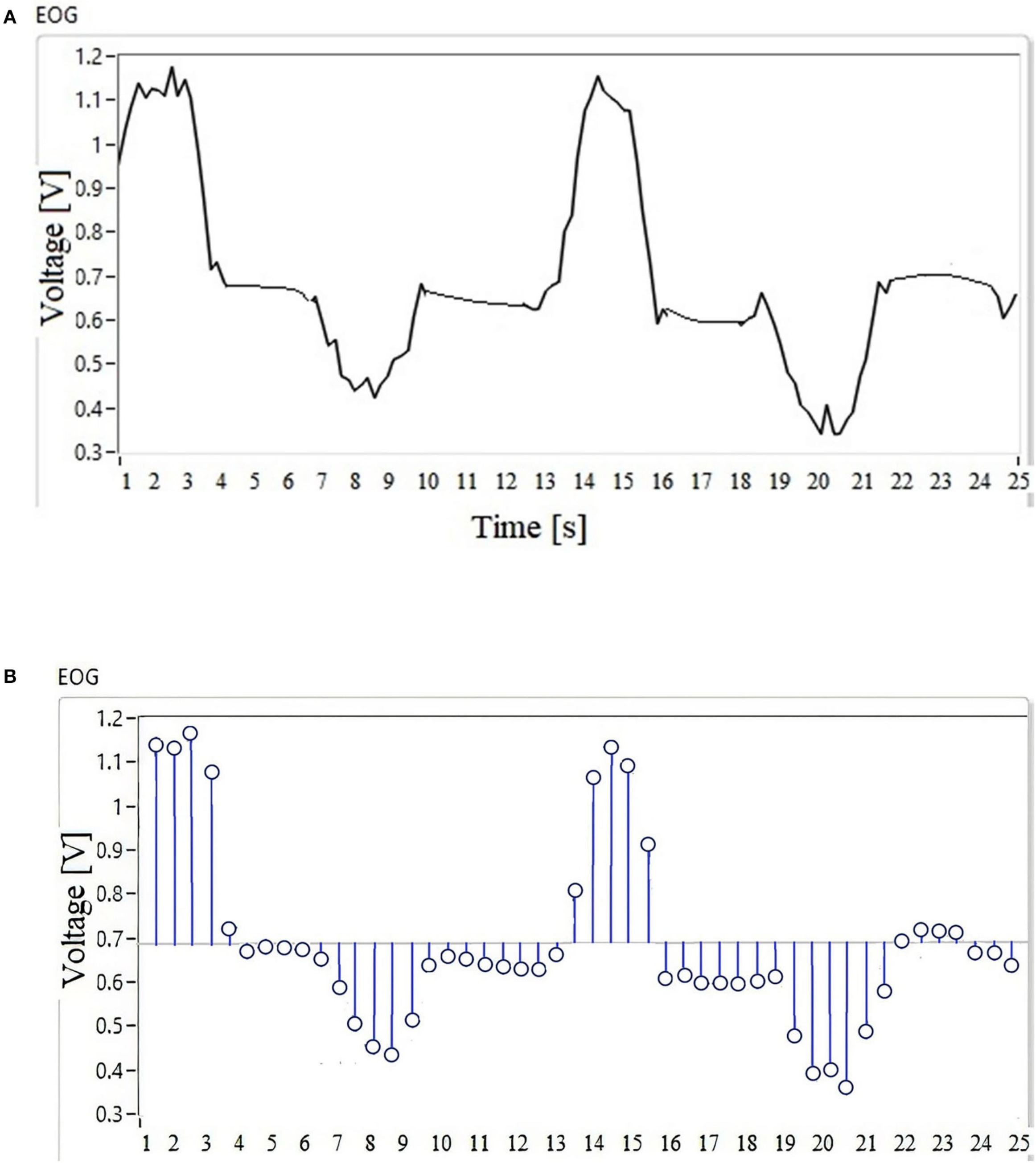


FIGURE 4 | (A) Signal waveform. **(B)** Discrete samples.

Multilayer Neural Network

The Widrow–Hoff learning is a training algorithm for an MNN, with the objective of determining synaptic weights; polarization for the classification of data is not linearly separable (An et al., 2020). Given these characteristics, this algorithm was selected for the training of the neural network developed to model the EOG signal.

In **Figure 5**, a monolayer neural network is represented, where the vector of the R inputs is $\boldsymbol{p} = \left[p^T\right]$, $\boldsymbol{W} = \left[W_{SR}{}^T\right]$ is the synaptic weight matrix, $\boldsymbol{b} = \left[b^T\right]$ represents the polarization of the S neurons, $\boldsymbol{n} = \left[n^T\right]$ represents the net inputs of each of the S neurons, and $\boldsymbol{a} = \left[a^T\right]$ is the vector of the S outputs of the neurons (An et al., 2020).

The output of the monolayer neural network is represented in Equation (2):

$$a = f\left(Wp + b\right) \tag{2}$$

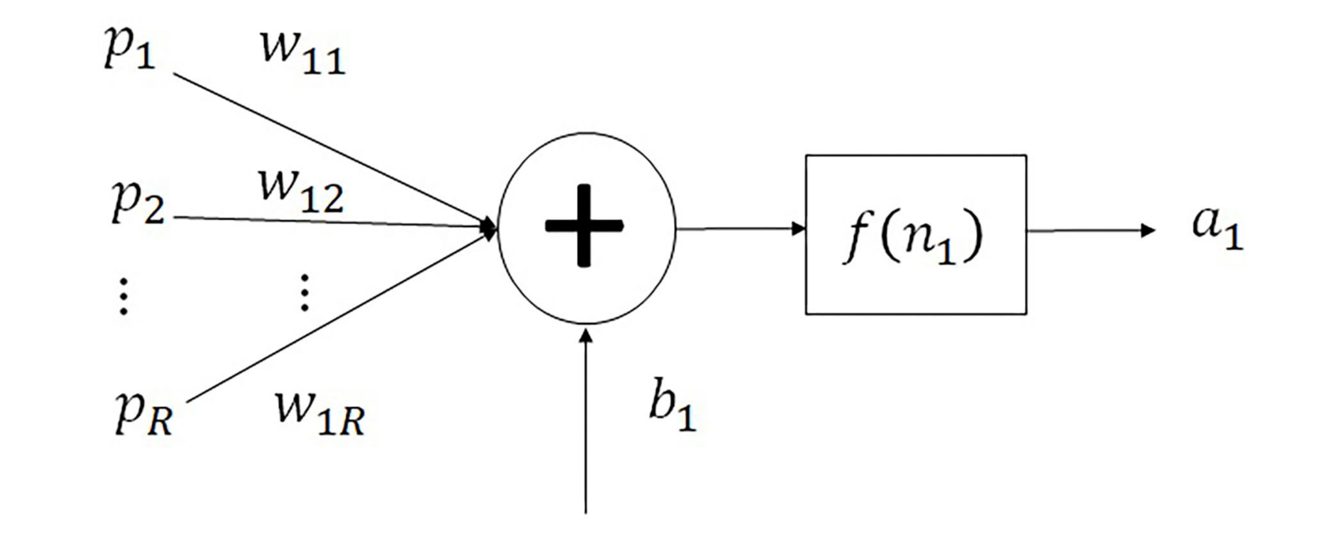


FIGURE 5 | Representation of a monolayer neural network.

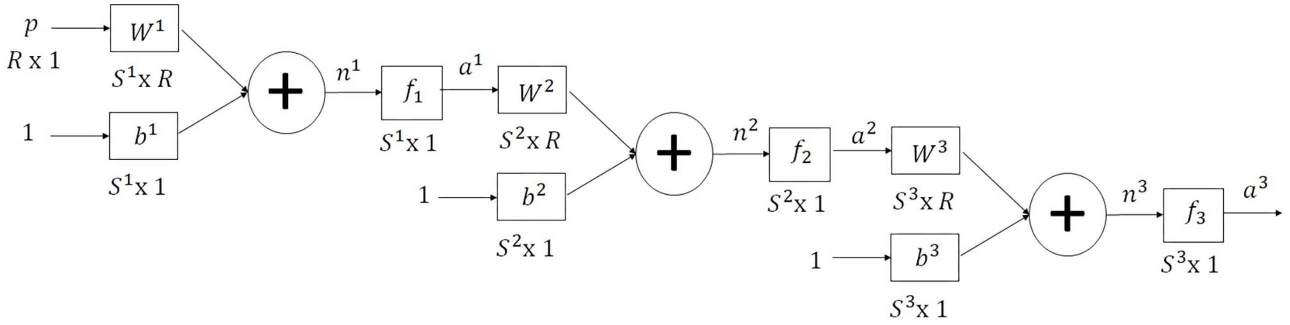


FIGURE 6 | MNN for the calculation of the objective function that describes the behavior of the EOG signal.

The neural network employs activation functions, using a least squares method for its training. The weights are adjusted using the Widrow–Hoff rule to minimize the difference between the output and the objective. This algorithm is an iterative implementation of linear regression, reducing the square error of a linear fit.

A pattern p_q is presented as the input to a network; it responds with an output a_q. Due to this, an error vector e_q is formed, which is the subtraction of the desired answer t_q, and the neuron's response a_q so that $e_q = t_q - a_q$. Square error is defined as the dot product $e_q^T e_q$ of the error vector that provides the sum of the square errors of each neuron. In order to minimize the square error, the gradient descent is used, whose objective problem is to find x_0 which minimizes function $F(x)$. In Equation (3), the descending gradient equation is presented to minimize the square error.

$$x_0 = x_0 - \alpha \left. \frac{dF}{dx} \right|_{x=x_0} \tag{3}$$

The value of $F(x)$ is defined as $e_q^T e_q$ whose objective is to minimize the square error by means of an iterative Widrow–Hoff learning. There is a set of test patterns (p_Q, t_Q), and with these data, the synaptic weights and polarization are found so that the multilayer network responds as desired.

The neural network multilayer is implemented to calculate the function that describes the behavior of the EOG signals. It is a neural network that has three layers; it is represented in **Figure 6**.

The multilayer neural network is used for linear regression; the structure is made up as follows: The input layer has a sigmoid activation function, the hidden layer has a sigmoid activation function, and the output layer has an activation function linear. This is the reason why using non-linear activation functions at the input corresponds to the smooth transition between one sampling point and another, while a linear output activation function allows obtaining numerical values that correspond to the exact value of the sample. In this way, a smooth transition is achieved, and all the sampling points are covered for a correct modeling. The output of the multilayer network takes values that the EOG signal registers which vary according to each person;

using a sigmoid function in layer 3 does not allow to reach these values.

The recursive equation that describes the output of the multilayer neural network represented by a^M with input patterns p through q, for a neural network with M layers, is presented in Equation (4), where X^M represents the polarization and synaptic weights of neuron M. The solution is more complex because these parameters must be calculated for each of the neurons that make up the multilayer.

$$\begin{aligned} a^M &= f^M\left(W^M f^{M-1}\left(W^{M-1} \cdots f^2\left(W^2 f^1\left(W^1 p + b^1\right) + b^2\right) \cdots + b^{M-1}\right) + b^M\right) \Rightarrow \\ a^m &= f^m\left(W^m a^{m-1} + b^m\right) \end{aligned} \tag{4}$$

The objective is to minimize the square error, which is a function of X^M arrays (Equation 5).

$$F\left(X^1, \cdots, X^M\right) = e_q^T e_q \ \ where X^m = \left[W^m \ \ b^m\right]^T \tag{5}$$

The function obtained $F\left(X^1, \cdots, X^M\right) = e_q^T e_q$; the gradient descent method is used to find synaptic weights and polarizations that minimize square error. An optimization method has been obtained that is found when defining the error gradient and is minimized with respect to the parameters of the neural network, as indicated in Equation (6).

$$\begin{aligned} X^m &= X^m - \alpha \left.\frac{dF}{dX^m}\right|_{x^m = x_0^m} \\ X^m &= \left[W^m \ \ b^m\right]^T \end{aligned} \tag{6}$$

The variation of the mean square error with respect to the synaptic weights and the polarization of the corresponding neuron is described in Equation (7).

$$\frac{dF}{dX^m} = \left[\frac{dF}{dx_1^m} \ \cdots \ \frac{dF}{dx_{S^m}^m}\right] \tag{7}$$

To calculate the gradient $\frac{dF}{dX^m}$, the function can be decomposed using the chain rule as the variation of F with respect to net input n_i, and the product of the variation of the net input with respect to the variation of the neuron's polarization and synaptic weights i is represented by X_i. The net input is represented as the product of the vectors $n_i = x_i^T z$; the results is seen in Equation (8).

$$\frac{dF}{dx_i} = \frac{dn_i}{dx_i}\frac{dF}{dn_i} = zs_i \tag{8}$$

There is variation of function F, which is the square error with respect to any net input; in any layer of the neuron, it is represented with an **s** and is called sensitivity. In the sensitivity gradients in Equation (8), the vector z is factored and is replaced by the input augmented with 1 in the last element; the input of layer n is the output of the previous layer, so $z^m = \begin{bmatrix} a^{m-1} \\ 1 \end{bmatrix}$ and applying the transposed operator the Equation (9) is obtained.

$$\begin{aligned} \frac{dF}{dX^m} &= \left[zs_i \ \ldots \ zs_{s^m}\right]^m = z^m\left(s^m\right)^T = \begin{bmatrix} a^{m-1} \\ 1 \end{bmatrix}\left(s^m\right)^T \\ &= \begin{bmatrix} a^{m-1}\left(s^m\right)^T \\ \left(s^m\right)^T \end{bmatrix} \\ \frac{dF^T}{dX^m} &= \left[s^m\left(a^{m-1}\right)^T \quad s^m\right] \end{aligned} \tag{9}$$

If Equation (9) in the formula for the descending gradient of Equation (3) and the vector X^m is replaced in terms of synaptic weights W and polarization b, Equations (10) and (11) are obtained which determine the iterative method for learning a multilayer neural network by Widrow–Hoff.

$$\left(X^m\right)^T = \left(X^m\right)^T - \alpha\frac{dF^T}{dX^m} \tag{10}$$

$$\begin{aligned} W^m &= W^m - \alpha s^m\left(a^{m-1}\right)^T \\ b^m &= b^m - \alpha s^m \\ &For\ \forall m \in [1, \ \ldots, \ M] \end{aligned} \tag{11}$$

Now the calculation of the sensitivities must be carried out, which is the basis of the backpropagation algorithm. Sensitivity is defined as the derivative of the function, which is the square error, with respect to the net input of the neuron (Equation 12).

$$\frac{dF}{dn^{m-1}} = \frac{dn^m}{dn^{m-1}}\frac{dF}{dn^m} \tag{12}$$

In Equation (12) applying the chain rule, we have the variation of F with respect to the net input of the layer m, as well as the variation of the net input of the layer m with respect to the net input of the previous layer n^{m-1}. If the nomenclature of sensitivities is used, Equation (13) is obtained.

$$s^{m-1} = \frac{dn^m}{dn^{m-1}}s^m \tag{13}$$

Equation (13) indicates the sensitivity of the previous layer s^{m-1} which is calculated from the sensitivity of the back layer s^m. This relationship is what gives it the name of backpropagation because the sensitivity will be propagated from the last layer to the first layer of the neural network to calculate the sensitivity in each one. The net inputs of two consecutive neural networks are related by Equation (14).

$$n^m = W^m f^{m-1}\left(n^{m-1}\right) + b^m \tag{14}$$

There is an equation where the net input n^m depends on f, which is the activation function of the net input n^{m-1} from the previous layer. Using the chain rule, the result is expressed in Equation (14), where the derivative of the activation function f^{m-1} with respect to the net input of the previous layer n^{m-1} is expressed as $\dot{F}^{m-1}\left(n^{m-1}\right)$. The second derivative of the net input n^m in relation to the activation function f^{m-1} is obtained by deriving Equation (13) which results in the transposed vector of the synaptic weights of layer m using the equation $(W^m)^T$, where Equation (15) is obtained.

$$\begin{aligned}\frac{dn^m}{dn^{m-1}} = \frac{df^{m-1}}{dn^{m-1}}\frac{dn^m}{df^{m-1}} &= \dot{F}^{m-1}\left(n^{m-1}\right)\left(W^m\right)^T \\ s^{m-1} &= \dot{F}^{m-1}\left(n^{m-1}\right)\left(W^m\right)^T s^m \\ For\forall m &\in [M, \ldots, 2] \qquad (15)\end{aligned}$$

Equation (15) calculates each of the sensitivities; in order to carry out this process, it is necessary to calculate the sensitivity of the last layer M (where $s^m = s^M$). Applying the chain rule to deduce the sensitivity s^M, the last layer of the sensitivity definition is known to be the derivative of the objective function to be minimized with respect to the net input of the last layer n^m, F is the square error $F = e_q^T e_q$, and the error is the difference of the desired response t_q and the response of the last layer, defined as the activation function $f^M\left(n^M\right)$, that is, $e_q = t_q - f^M\left(n^M\right)$; the result of the said process is observed in Equation (16).

$$\begin{aligned}s^M &= \frac{dF}{dn^M} = \frac{df^M}{dn^M}\frac{de_q}{df^M}\frac{dF}{de_q} = \frac{df^M}{dn^M}(-1)\left(2e_q\right) \\ &= -2\dot{F}^M\left(n^M\right)e_q \qquad (16) \\ s^M &= -2\dot{F}^M\left(n^M\right)e_q \qquad (17)\end{aligned}$$

The definition of $\dot{F}^M\left(n^M\right)$ which is the derivative of the activation functions with respect to the net input; this process is represented in Equation (17). The derivative generates a matrix containing the gradients of each of the activation functions of the neural network with respect to its net input, and so on, until it reaches the last neuron.

$$\begin{aligned}\dot{F}^M\left(n^M\right) &= \dot{F}^m\left(n^m\right)\frac{df^m}{dn^m} = \left[\frac{df_1^m}{dn^m}\ \cdots\ \frac{df_{s^m}^m}{dn^m}\right] \qquad (18) \\ &= \begin{bmatrix}\frac{df_1^m}{dn_1^m} & 0 & \cdots & 0 \\ 0 & \frac{df_2^m}{dn_2^m} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{df_{s^m}^m}{dn_{s^m}^m}\end{bmatrix} = diag\left(\frac{df_1^m}{dn_1^m}\right)\end{aligned}$$

The derivative must exist for any value of the net input that is a continuous function, and there are three activation functions to which the operation $\dot{F}^M\left(n^M\right)$ must be calculated. The MNN is made up of three neurons, but two activation functions are used: the sigmoidal and the linear. The result of implementing Equation (17) in the activation functions of the neural network is presented in Equation (18) for the sigmoid activation function and in Equation (19) for the linear activation function.

- Logistics sigmoid.

$$a_i = f_i\left(n_i\right) = \frac{1}{1+e^{-ni}} \rightarrow \frac{df_i}{dn_i} = \left(1-a_i\right)a_i$$

If all neurons have the same function:

$$\dot{F}^m\left(n^m\right) = diag\left(\left(1-a_i^m\right)a_i^m\right) \qquad (19)$$

- Lineal function.

$$a_i = f_i\left(n_i\right) = ni \rightarrow \frac{df_i}{dn_i} = 1$$

If all neurons have the same function:

$$\dot{F}^m\left(n^m\right) = 1 \qquad (20)$$

The descending backpropagation algorithm for calculating an objective function that models the behavior of the EOG signal by discrete samples is presented in **Listing 1**.

Listings 1 | Backpropagation algorithm for interpolation of an EOG signal using a multilayer neural network, through discrete samples.

```
Pseudocode: algorithm for interpolation of an EOG
Random initialization of $W^m$ and $b^m$ for $\forall m$
Since epochs = 1 to $N_{epochs}$ repeat
Since $q = 1$ to $Q$ repeat $(Q =$ Sample vector size$)$
        1. Forward propagation.
           $a^m = f^m\left(W^m a^{m-1} + b^m\right)$ for $\forall m$
        2. Backpropagation.
           $e_q = t_q - a_0^M$
           $s^M = -2\dot{F}^M\left(n^M\right)e_q$
           $s^{m-1} = \dot{F}^{m-1}\left(n^{m-1}\right)\left(W^m\right)^T s^m$ for $\forall m \in [M, \ldots, 2]$
        3. Update for $\forall$ m
           $W^m = W^m - \alpha s^m\left(a^{m-1}\right)^T$
           $b^m = b^m - \alpha s^m$
    End
End
```

From the acquisition of the EOG signal, 48 discrete samples are obtained that are stored as a data vector Q; the non-linear regression is applied on these data. The algorithm calculates a function that passes through all the discrete points. **Figure 7** shows a trend line resulting from the neural network when interpolating the signal samples, this being the output of the last layer.

The function $f(x) = a^m = f^m\left(W^m x^{m-1} + b^m\right)$ describes the behavior of the EOG signal of each individual, depending on the variability of the value of the synaptic weights W^m and polarization b^m. Through this method, the analytical description of the physiological signal is obtained.

By having a mathematical function that describes the individual characteristics of each user, information is obtained that allows a classification system to adapt to the variability that physiological signals present. As can be seen in **Figure 7**, this signal has several positive and negative data on a threshold; in order to determine the operating range of a system, it is necessary to know the amplitude in which the signal varies for each user. The objective is to record the maximum and the minimum value of the signal threshold to calibrate the fuzzy inference system.

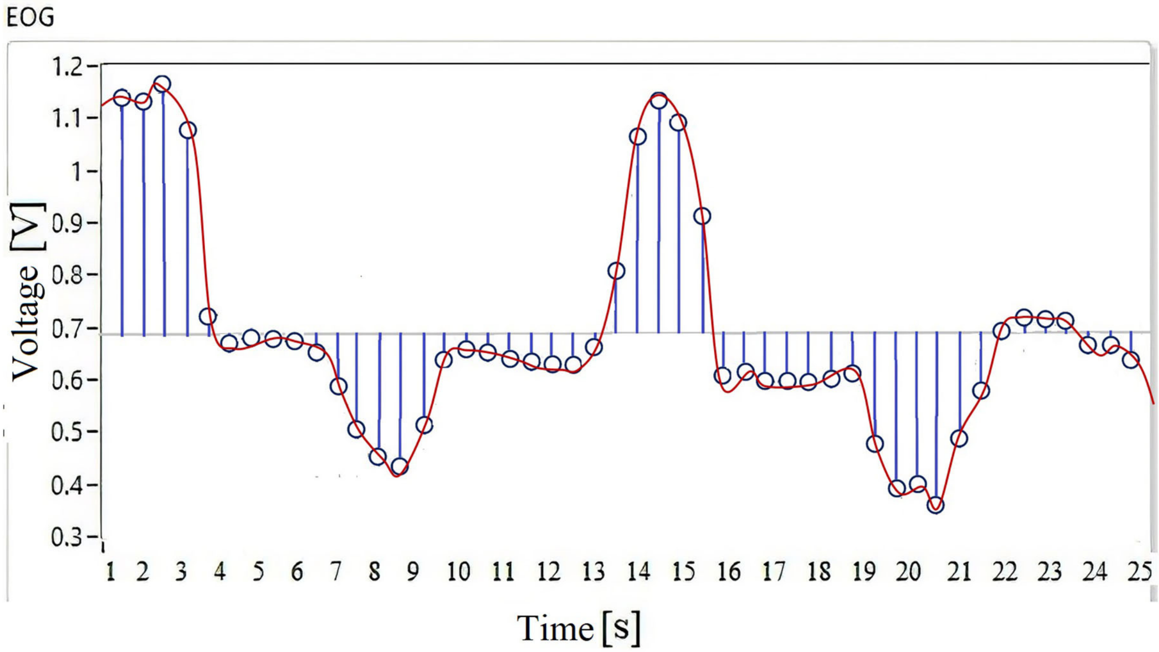


FIGURE 7 | Interpolation using a multilayer neural network, to calculate an objective function to be optimized.

Genetic Algorithm

An optimization problem can be formulated as a process where the optimal value x which minimizes or maximizes the objective function is found. In this case, the objective function is the result from interpolation performed with a multilayer neural network and it is determined by Equation (20). Considering that W^m represents the value of the synaptic weights, b^m is the value of polarization of each layer, m is the maximum number of layers, and $\mathbf{X}$ represents the vector of decision variables. $\mathbf{X}$ represents the candidate solution set, also known as the search space or solution space, such that $\mathbf{x} \in \mathbf{X}$. The search space is limited by the lower (l) or upper (u) limits of each of the d variables, as indicated in Equation (20).

$$f(x) = f^m\left(W^m x^{m-1} + b^m\right) \in \mathbb{R}, \qquad (21)$$
$$X = \left\{x \in \mathbb{R} \mid l_i \leq x_i \leq u_i,\ d = 1\right\}$$

The objective function obtained is a complex problem to solve using classical optimization methods, because it contains a set of local optimums. Therefore, an evolutionary method like genetic algorithms is a good alternative for its solution. When the mathematical model of the EOG signal is obtained as a result of the processing of the neural network, it is represented as a mathematical function with local positive and negative thresholds; using classical optimization techniques, it is not possible to determine the range of the signal because it presents different ridges and valleys of different amplitudes, so the two objectives sought are to obtain the maximum and minimum optima regardless of the variable characteristic of the signal. For the maximum optimal value, the 15 iterations of the genetic algorithm are applied, and to obtain the minimum optimal value, a negative sign multiplies the objective function and the 15 iterations of the genetic algorithm are performed again.

For genetic algorithms (GA), each candidate solution is considered an individual that belongs to a population, and its level of "adaptation" is the value obtained when evaluating each of the candidate solutions with the objective function (Leardi, 2003). Basically, a GA is an algorithm that generates a random population of parents; during each generation, it selects pairs of parents considering their value $f(x)$ and exchanges of genetic material or crosses are made to generate pairs of children; such children will be mutated with a certain probability and will ultimately compete to survive the next generation with the parents, a process known as elitism.

The algorithm corresponding to a GA is indicated below:
Number of dimensions d = 1
Search space limits l = 0 y u = 25
Number of iterations Niter = 15
Population size Np = 48
Number of bits per dimension Nb = 11
Initialization by the equation: $x_n = l + rand(.)^* \left(u - l\right)$
Selection of parents (Roulette Method): Each individual is evaluated considering the objective function.

The cumulative of the objective function is calculated as E, as indicated in Equation (21).

$$E = \sum_{i=1}^{Nb} f(x_i) \qquad (22)$$

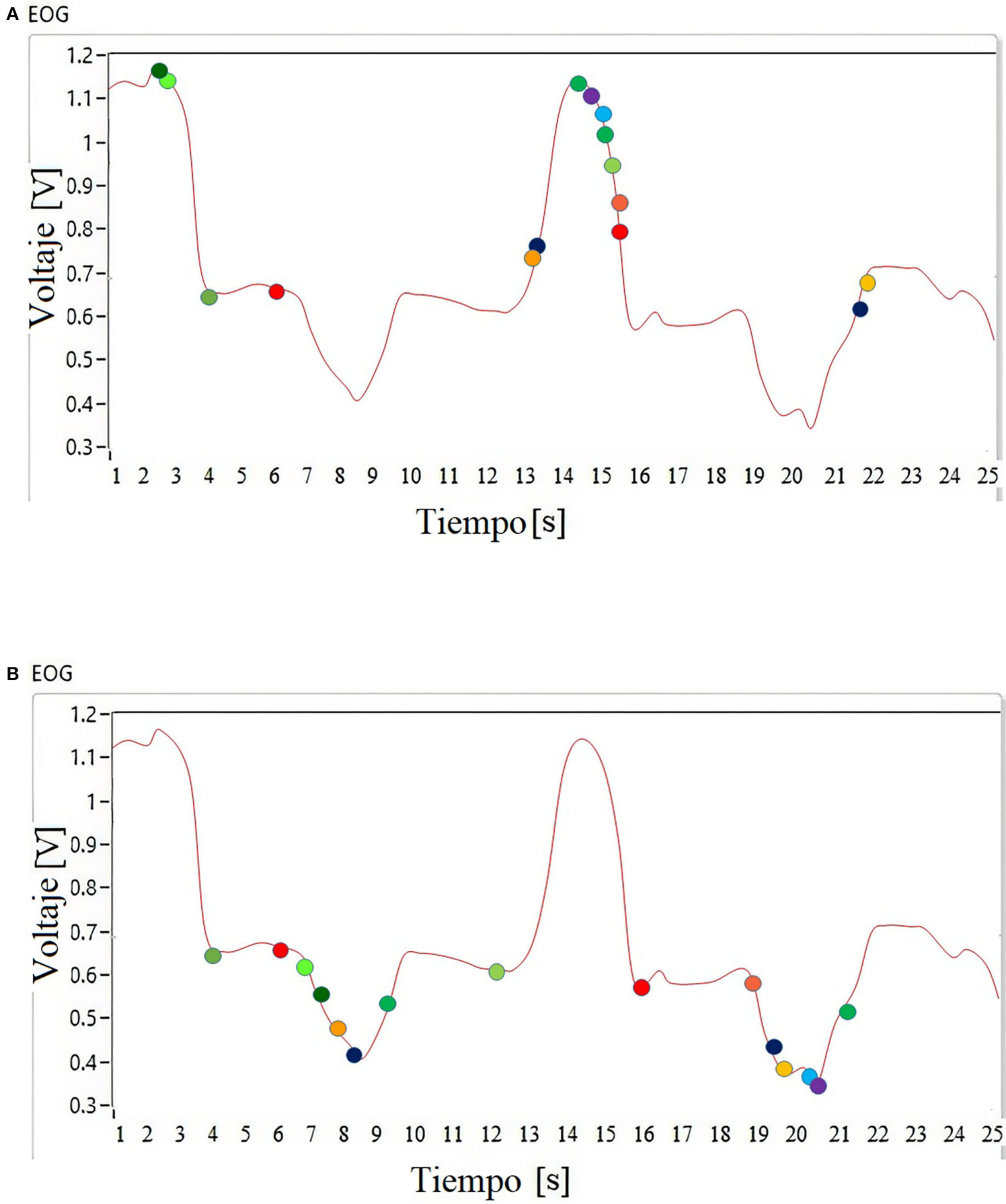


FIGURE 8 | (A) Genetic algorithm at optimal maximum. **(B)** Genetic algorithm at optimal minimum.

The possibility of selection of each individual is calculated, as shown in Equation (22).

$$p_i = \frac{f\,(x_i)}{E} \tag{23}$$

The cumulative probability of each individual is calculated, represented in Equation (23).

$$q_i = \sum_{j=1}^{i} p_j \tag{24}$$

Then, the selection of the parents is made:

A uniformly distributed random number is generated.

The parent that satisfies the condition $q_i > r$ is selected, where r is a random value between 0 and 1.

In a GA, it is necessary to determine certain parameters for its design; these are as follows.

Cross: It consists of randomly generating a location within each individual which will serve as a reference for the exchange of genetic information, previously converting to binary values. We consider a parent pair of 11 binary data and an initial cross point with a value of 7 generated randomly. Each individual is divided into two parts: one of 5 bits and the other of 6 bits; later, the complementary parts of each individual will be united, to form the descendants.

Mutation: The individuals in the population are made up of binary chains; the mutation is carried out by changing with some probability the bits of each descendant individual, generating a population of mutated children, although it is also necessary to convert the said population to real numbers in order to evaluate them in the objective function.

Selection of the fittest: It is necessary to select the fittest individuals who will survive the next generation. This is achieved through the competition of the parents with the children that were generated after applying the cross and mutation operators; in the case of the binary AG, the original populations of parents are simply mixed, and that of the children generated.

In **Figure 8A**, the data is observed in each period of the genetic algorithm cycle in the case of obtaining the optimal maximum, while in **Figure 8B** the data is observed in the same period of the genetic algorithm, but calculating the optimal minimum.

Table 1 records the data in each period in which the optimization algorithm is evaluated. As an example for a specific user, the data in **Table 1** are obtained.

Fuzzy Inference System

To characterize the EOG signal and to be able to use it to generate Cartesian coordinates, a classification system with fuzzy logic was implemented. This method uses a set of mathematical principles based on degrees of belonging and is performed based on linguistic rules that approximate a mathematical function. The input is the signals of the two previously calibrated EOG horizontal and vertical channels, and the output of the system are Cartesian coordinates within the working space of an assistance system, in this case an anthropomorphic manipulator robot.

The Mamdani-type inference method was implemented to design the fuzzy classifier because it allows the intuitive relationship through syntactic rules between the workspace and the voltage thresholds; this feature is very useful when generating a point in Cartesian space in real time.

In the mathematical interpretation of Mamdani's fuzzy controller, there are two fuzzy antecedents expressed by membership functions of the linguistic variables A' and B', with a first premise or a valid fact: If x is A' and y is B', then we have a set of fuzzy rules expressed in the form; if x is A_i and y is B_i then z is C_i. Where x is the voltage range of the calibrated EOG signal for the input or the robot workspace for the output, A' and B' are the antecedents of the linguistic variables expressed by membership functions and C_i is the consequent of a fuzzy set z. In the end, when evaluating all the fuzzy rules, we have a conclusion set z which is C'; this approach is represented in Equation (24) using the Mamdani inference model.

TABLE 1 | Values delivered by the genetic algorithm in each period.

Iteration number	Maximum optimum	Minimum optimum
1	0.7328	0.6983
2	0.8102	0.6452
3	0.8392	0.6012
4	0.8583	0.5932
5	0.8873	0.5874
6	0.8921	0.5320
7	0.9012	0.4943
8	0.9058	0.4832
8	0.9134	0.48032
10	0.9239	0.4786
11	0.9323	0.4632
12	0.9532	0.4324
13	0.9832	0.4132
14	1.0983	0.3932
15	1.1193	0.3172

The values highlighted in gray in ***Table 1*** *provide the range to which the EOG classifier should be calibrated for this user, thus assigning to each voltage threshold a coordinate in Cartesian space using fuzzy logic.*

$$
\begin{array}{lr}
x \text{ is } A' \text{ and } y \text{ is } B' & \text{True Premise} \\
\quad \text{If } x \text{ is } A_1 \text{ and } y \text{ is } B_1,\ \text{ then } z \text{ is } C_1 & \text{Fuzzy rule 1} \\
\quad \text{If } x \text{ is } A_2 \text{ and } y \text{ is } B_2,\ \text{ then } z \text{ is } C_2 & \text{Fuzzy rule 2} \\
\qquad \vdots & \\
\quad \text{If } x \text{ is } A_i \text{ and } y \text{ is } B_i,\ \text{ then } z \text{ is } C_i & \text{Fuzzy rule i} \\
\hline
z \text{ is } C' & \text{Set Conclusion.}
\end{array}
\tag{25}
$$

By classifying the EOG signal by thresholds from positive to negative, leaving an inactivation zone as indicated in **Figure 9,** the response relationship is performed in the Cartesian space of the anthropomorphic robot. The entire workspace is mapped according to the threshold registered in the classifier.

The membership function used to verify the performance of the fuzzy classifier is a Gaussian-type function, such as that presented in Equation (25). The implemented membership functions are Gaussian, because the transition between one membership function to another is smooth; it also helps to generate trajectories from one point to another without using cubic polynomials like the Spline technique.

$$
\text{Gaussian}: f\left(x;\, a, b\right) = \left\{ e^{\left(\frac{x-a}{b}\right)^2} \right. \tag{26}
$$

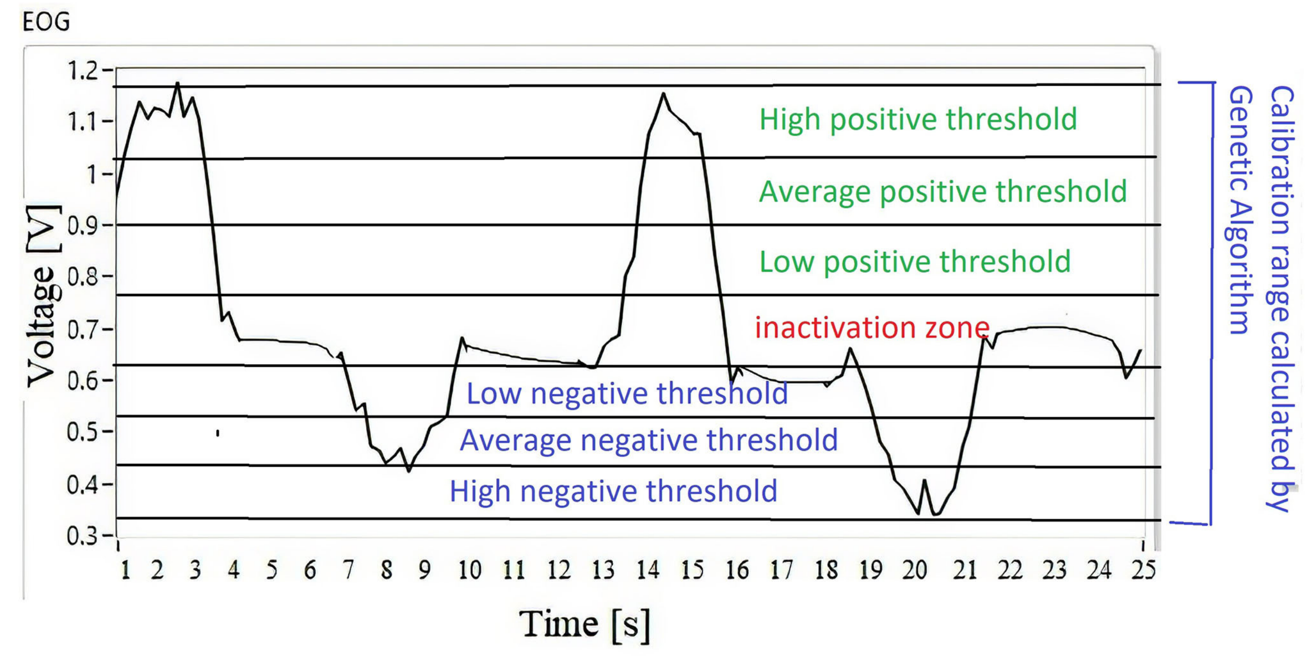


FIGURE 9 | Classification using thresholds of the EOG signal.

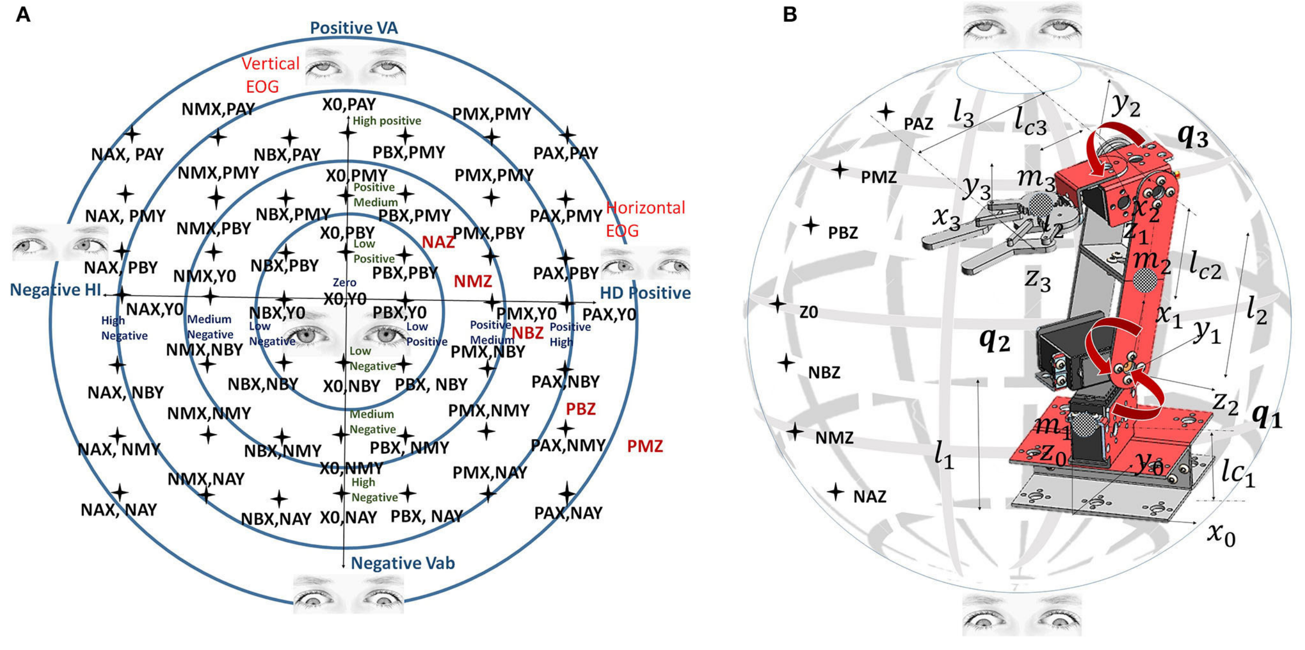


FIGURE 10 | (A) Relation of the robot workspace and the EOG signals. **(B)** Robot workspace and dimensions.

where a defines the mean value of the Gaussian bell, while b determines how narrow the bell is.

The output is the work space of the anthropomorphic robot, which is represented as a hollow sphere; previously, studies of direct and inverse kinematics were performed to calculate its work space, as well as robot dynamics to apply control algorithms for monitoring of the paths generated by the fuzzy classifier. The robot with which experimental tests were carried out is presented in **Figure 10B**.

According to the voltage level that each linguistic variable represents, the inputs are defined, x = EOG vertical/EOG horizontal, for fuzzy classifier inputs that use Gaussian membership functions $T(x)$; the names of these functions are presented in **Algorithm 1.**

Algorithm 1: Name of each of the Classifier input Functions.

$T(x) =$
Negative Vertical/Horizontal High *(NAV/NAH),*
Vertical/Horizontal Medium Negative (NMV/NMH)
Vertical/Horizontal Low Negative (NBV/NBH)
Vertical/Horizontal Zero (ZV/ZH),
Positive Vertical/Horizontal Low (PBV/PBH),
Positive Vertical/Horizontal Medium (PMV/PMH),
Vertical/Horizontal High Positive (PAV/PAH).

According to the voltage level represented by each linguistic variable, the workspace is defined with $y =$ Position $X/Y/Z$, for the outputs of the fuzzy classifier, using the Gaussian membership functions $T(y)$. The name of the functions is indicated in **Algorithm 2.**

Algorithm 2: Name of each of the Output Functions of the Classifier.

$T(y) =$
Negative High X/Y/Z (NAX/NAY/NAZ)
Negative Medium X/Y/Z (NMX/NMY/NMZ)
Negative Low X/Y/Z (NBX/NBY/NBZ)
Zero X/Y/Z (X0/Y0/Z0)
Positive Low X/Y/Z (PBX/PBY/PBZ)
Positive Medium X/Y/Z (PMX/PMY/PMZ)
Positive High X/Y/Z (PAX/PAY/PAZ).

Algorithm 3 establishes the range of the membership functions of the optimization of the modeling of the EOG signal from the results obtained in **Table 1**, corresponding to each vertical and horizontal channel modeled by Gaussian membership functions.

Algorithm 4 establishes the range of the membership functions of the fuzzy classifier outputs for each of the Cartesian coordinates in $f(p_x,\ p_y,\ p_z)$; the Gaussian membership functions are used depending on the workspace of any assistance system, whose positions are expressed in Cartesian coordinates, in this case that of an anthropomorphic robot with three degrees of freedom.

The relationship between the voltage thresholds and the robot workspace is indicated by 113 inference rules. The workspace is classified on dividing surfaces by slices from the positive to the negative threshold; the horizontal EOG is related to the positions on the x-axis, while the vertical EOG is related to the positions on the z-axis and y-axis. By having the membership functions for the inputs and outputs of the fuzzy classifier, syntactic rules are implemented that indicate the points that form the trajectory to be followed by the manipulator robot. **Figure 10A** represents the relationship of the workspace in the XY plane by means of a Cartesian axis; the horizontal EOG channel is represented by the abscissa, and the vertical EOG channel is represented by the ordinates. Each of the concentric circles represents a layer of the plane that encompasses the position correlation in the XY plane and the EOG signal voltage threshold value from positive values above the baseline and values below this reference which take negative values. **Figure 10B** represents the relationship of the workspace on the z-axis with respect to the vertical EOG channel and the three degrees of freedom robot $\left(q_1,\ \ q_2\ and\ q_3\ \right)$.

All fuzzy syntactic rules for generating positions through eye movement interaction were introduced into the LabVIEW Design Manager (see **Figure 11A**) V2019. The horizontal EOG signal corresponds to the x-axis or abscissa, while the vertical EOG corresponds to the y-axis or ordered y for the z coordinates, generating a trajectory in the Cartesian space using a function $f\left(x, y, z\right)$. The LabVIEW V.2019 Design Manager presents the control surfaces for each X, Y, and Z position output relative to the horizontal and vertical channel input data. In each trend of the surfaces, it is observed that while the voltage value in the horizontal/vertical EOG is positive, the graph has a blue color. The direction of position in each of the axes is explained; the graph indicates the positions in X to the right (see **Figure 11A**), in Y it indicates the position upward on the XY plane (see **Figure 11B**), and in Z the robot's position is higher than the XY plane (see **Figure 11C**). In contrast, if the graph tends toward negative values, it has a red color tone; the position values in X are to the left (see **Figure 11A**), in Y it indicates the position down on the XY plane (see **Figure 11B**), and in Z the robot position is lower than the XY plane (see **Figure 11C**), covering the entire workspace.

The classifier has the property of being variable in the input membership functions to be adaptive to any user, while the output membership functions are variable in order to adjust the classifier to any navigation system with coordinates in the Cartesian space. This system can be adapted to the generation of trajectories for autonomous aerial vehicles, a pointer for a personal computer in order to write letterforms and for home automation systems; however, for the purposes of this work the fuzzy output classifier adapts to the workspace of an anthropomorphic robot. **Table 2** describes the relationship of the horizontal and vertical EOG signals and the robot workspace represented as a hollow sphere. The position is determined by the membership functions of the semantic rules $\left(p_x, p_y,\ and\ p_z\right)$. The coordinates of the manipulator robot are previously defined for each value of the acquisition potential of the EOG signal, covering the entire workspace of the robot. For example, the horizontal EOG input is defined in the membership function ZH, the vertical EOG input is defined in the membership function ZV, and the output values are in Cartesian coordinates; they are delimited by the membership functions $X0$, $Y0$, and $Z0$ set as the robot home position.

Robot Position Control Scheme

The result of the classifier provides the position of the robot in Cartesian coordinates (X, Y, Z); to convert these results into desired joint coordinates $\left(q_{1d}, q_{2d}, q_{3d}\right)$, the inverse kinematics of the robot are used. These values are the input of the control PD+ algorithm that orders the robot to path tracking.

Algorithm 3: Input membership functions.

EOG input signal: $\left[0.3172\ \left(optimal\ minimum\right)\ \ to\ 1.1923\ \left(optimal\ maximum\right)\right]$ *Volts*. Peak-to-peak voltage of an EOG signal (Optimized Variable Rating Range).

x=EOG Vertical	**x=EOG Horizontal**
$M\left(NAV\right)=Gaussian\left(x,[0.31\ \ 0.31\ \ 0.4\ \ 0.49]\right)$	$M\left(NAH\right)=Gaussian\left(x,[0.31\ \ 0.31\ \ 0.4\ \ 0.49]\right)$
$M\left(NMV\right)=Gaussian\left(x,[0.4\ \ 0.49\ \ 0.58\ \ 0.67]\right)$	$M\left(NMH\right)=Gaussian\left(x,[0.4\ \ 0.49\ \ 0.58\ \ 0.67]\right)$
$M\left(NBV\right)=Gaussian\left(x,[0.49\ \ 0.58\ \ \ 0.67\ \ 0.76]\right)$	$M\left(NBH\right)=Gaussian\left(x,[0.49\ \ 0.58\ \ \ 0.67\ \ 0.76]\right)$
$M\left(ZV\right)=Gaussian\left(x,[0.67\ \ 0.76\ \ 0.85\ \ 0.94]\right)$	$M\left(ZH\right)=Gaussian\left(x,[0.67\ \ 0.76\ \ 0.85\ \ 0.94]\right)$
$M\left(PBV\right)=Gaussian\left(x,\ \ [0.76\ \ \ 0.85\ \ \ 0.94\ \ 1.03]\right)$	$M\left(PBH\right)=Gaussian\left(x,\ \ [0.76\ \ \ 0.85\ \ \ 0.94\ \ 1.03]\right)$
$M\left(PMV\right)=Gaussian\left(x,[0.85\ \ \ 0,94\ \ \ 1.03\ \ \ 1.10]\right)$	$M\left(PMH\right)=Gaussian\left(x,[0.85\ \ \ 0,94\ \ \ 1.03\ \ \ 1.10]\right)$
$M\left(PAV\right)=Gaussian\left(x.[1.03\ \ \ 1.10\ \ \ 1.\ 19\ \ 1.19]\right)$	$M\left(PAH\right)=Gaussian\left(x.[1.03\ \ \ 1.10\ \ \ 1.\ 19\ \ 1.19]\right)$

Algorithm 4: Output membership functions.

Robot workspace: $[-30\ a\ \ 30]\ \ cm$ Distance in centimeters within the workspace (this range may vary depending on the device workspace).

x=Position p_x	**x=Position p_y**	**x=Position p_z**
$M\left(NAX\right)=$ $Gaussian\left(x,[-30-30-22.5\ \ -15]\right)$	$M\left(NAY\right)=$ $Gaussian\left(x,[-30-30-22.5\ \ -15]\right)$	$M\left(NAZ\right)\ =\ Gaussian\ \left(x,[-30\ -\ 30\ -$ $22.5\ \ -15]\right)$
$M\left(NMX\right)=$ $Gaussian\left(x,[-22.5-18.7\ \ -11.2\ \ -7.5]\right)$	$M\left(NMY\right)=$ $Gaussian\left(x,[-22.5-18.7\ \ -11.2\ \ -7.5]\right)$	$M\left(NMZ\right)\ \ =\ \ Gaussian\ \ \left(x,[-22.5\ \ \ -$ $18.7\ \ \ -11.2\ \ \ -7.5]\right)$
$M\left(NBX\right)=$ $Gaussian\left(x,[-15\ \ -11.2\ \ \ -3.7\ \ \ 0]\right)$	$M\left(NBY\right)=$ $Gaussian\left(x,[-15\ \ -11.2\ \ \ -3.7\ \ \ 0]\right)$	$M\left(NBZ\right)=Gaussian\left(x,[-15\ -11.2\ \ -$ $3.7\ \ \ 0]\right)$
$M\left(X0\right)=$ $Gaussian\left(x,[-7.5\ \ -3.7\ \ 3.7\ \ 7.5]\right)$	$M\left(Y0\right)=$ $Gaussian\left(x,[-7.5\ \ -3.7\ \ 3.7\ \ 7.5]\right)$	$M\left(Z0\right)\ \ \ =\ \ \ Gaussian\ \ \ \left(x,[-7.5\ \ \ \ -$ $3.7\ \ 3.7\ \ 7.5]\right)$
$M\left(PBX\right)=$ $Gaussian\ \ \left(x,[0\ \ \ 3.7\ 11.2\ \ 1.5]\right)$	$M\left(PBY\right)=$ $Gaussian\ \ \left(x,[0\ \ \ 3.7\ 11.2\ \ 1.5]\right)$	$M\left(PBZ\right)\qquad\qquad\qquad\qquad\qquad=$ $Gaussian\ \ \left(x,[0\ \ \ 3.7\ 11.2\ \ 1.5]\right)$
$M\left(PMX\right)=$ $Gaussian\left(x,[7.5\ \ \ 11.2\ \ 18.7\ \ 22.5]\right.$	$M\left(PMY\right)=$ $Gaussian\left(x,[7.5\ \ \ 11.2\ \ 18.7\ \ 22.5]\right.$	$M\left(PMZ\right)\qquad\qquad\qquad\qquad\qquad=$ $Gaussian\left(x,[7.5\ \ \ 11.2\ \ 18.7\ \ 22.5]\right.$
$M\left(PAX\right)=$ $Gaussian\left(x,[15\ \ 22.5\ \ 30\ \ 30]\right)$	$M\left(PAY\right)=$ $Gaussian\left(x,[15\ \ 22.5\ \ 30\ \ 30]\right)$	$M\left(PAZ\right)=$ $Gaussian\left(x,[15\ \ 22.5\ \ 30\ \ 30]\right)$

The control law is expressed in Equation (26).

$$\tau_{PD+}=K_p\tilde{q}+K_v\dot{\tilde{q}}+M\left(q\right)\ddot{q_d}+C\left(q,\dot{q}\right)\dot{q}_d+B\dot{q_d}+g\left(q\right) \tag{27}$$

This algorithm requires the dynamic robot model, so $M\left(q\right)$ is a positive defined symmetric matrix $n\ x\ n$ which corresponds to the robot's inertia matrix, $C(q,\dot{q})$ is an array of $n\ x\ n$ which corresponds to the matrix of centrifugal forces or Coriolis, B is a vector $n\ x\ 1$ which determines the viscous friction coefficients, $g\left(q\right)$ is a vector $n\ x\ 1$ representing the effect of gravitational force, τ is a vector $n\ x\ 1$ indicating torque applied to joint actuators, K_p and K_v are the proportional and derivative constants of the controller, $\tilde{q}$ is joint position error, $\dot{\tilde{q}}$ is the joint speed error, $\ddot{q_d}$ is the desired joint acceleration, and $\dot{q_d}$ is the desired joint speed.

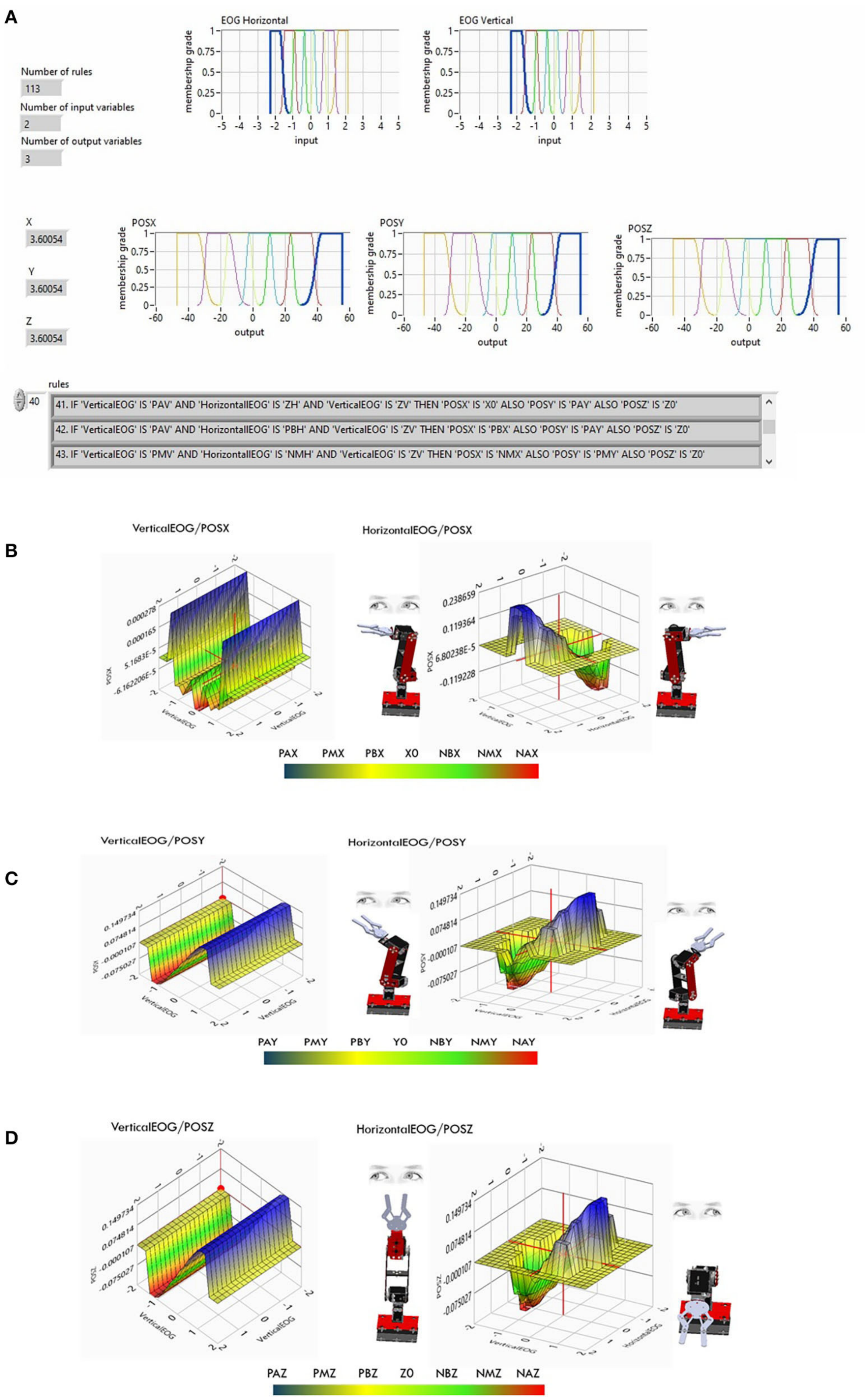
EXPERIMENTS AND RESULTS ANALYSIS

To perform different experiments to validate the operation of the designed HMI system, a graphical interface was developed that allows the operator to visualize the EOG signals of both channels, the movement of a virtual robot that emulates the movements generated by the interaction of the gaze, a graph showing the position in Cartesian coordinates of the data generated by the fuzzy classifier, and a visual feedback of the object to be taken by means of the image acquired by an external camera placed on the end effector. In addition, the response of the control algorithm, the position error, and the torque graph in each of the robot's joints are presented in the graphic interface (see **Figure 12**).

The characteristics of the robot used in the experiments are shown in **Table 3**.

To evaluate the performance of the HMI system, experiments were conducted with 60 individuals inexperienced in the use of this type of system. The purpose was to demonstrate that a system that adapts to the user allows a learning curve that requires fewer repetitions and therefore less time to perform a defined task, with the advantage of reducing the training time of a user to become an expert.

The performance of the HMI is verified by obtaining the time it takes the user, using the orientation of his eyeball (see



Number of rules
113
Number of input variables
2
Number of output variables
3

X
3.60054
Y
3.60054
Z
3.60054

rules
40
41. IF 'VerticalEOG' IS 'PAV' AND 'HorizontallEOG' IS 'ZH' AND 'VerticalEOG' IS 'ZV' THEN 'POSX' IS 'X0' ALSO 'POSY' IS 'PAY' ALSO 'POSZ' IS 'Z0'
42. IF 'VerticalEOG' IS 'PAV' AND 'HorizontallEOG' IS 'PBH' AND 'VerticalEOG' IS 'ZV' THEN 'POSX' IS 'PBX' ALSO 'POSY' IS 'PAY' ALSO 'POSZ' IS 'Z0'
43. IF 'VerticalEOG' IS 'PMV' AND 'HorizontallEOG' IS 'NMH' AND 'VerticalEOG' IS 'ZV' THEN 'POSX' IS 'NMX' ALSO 'POSY' IS 'PMY' ALSO 'POSZ' IS 'Z0'

FIGURE 11 | (A) Fuzzy classifier of the EOG signal using the 113 inference rules previously calibrated with genetic algorithms. **(B)** *X*-axis control surface. **(C)** *Y*-axis control surface. **(D)** *Z*-axis control surface.

TABLE 2 | Correspondence of the ocular displacement and the workspace of an anthropomorphic robot with three degrees of freedom.

Vertical EOG	**NAH**	**NMH**	**NBH**	**ZH**	**PBH**	**PMH**	**PAH**	**Horizontal EOG**	
PBV				X0, PBY, PAZ				PAV	Vertical EOG
ZV				X0, Y0, PAZ					
NBV				X0, NBY, PAZ					
Vertical EOG	**NAH**	**NMH**	**NBH**	**ZH**	**PBH**	**PMH**	**PAH**	**Horizontal EOG**	
PMV				X0, PMY, PMZ				PMV	Vertical EOG
PBV			NBX, PBY, PMZ	X0, PBY, PMZ	PBX, PBY, PMZ				
ZV			NBX, Y0, PMZ	X0, Y0, PMZ	PBX, Y0, PMZ				
NBV			NBX, NBY, PMZ	X0, NBY, PMZ	PBX, NBY, PMZ				
NMV				X0, NMY, PMZ					
Vertical EOG	**NAH**	**NMH**	**NBH**	**ZH**	**PBH**	**PMH**	**PAH**	**Horizontal EOG**	
PAV				X0, PAY, PBZ				PBV	Vertical EOG
PMV			NBX, PMY, PBZ	X0, PMY, PBZ	PBX, PMY, PBZ				
PBV		NMX, PBY, PBZ	NBX, PBY, PBZ	X0, PBY, PBZ	PBX, PBY, PBZ	PBX, PBY, PBZ			
ZV		NMX, Y0, PBZ	NBX, Y0, PBZ	X0, Y0, PBZ	PBX, Y0, PBZ	PMX, Y0, PBZ			
NBV		NMX, NBY, PBZ	NBX, NBY, PBZ	X0, NBY, PBZ	PBX, NBY, PBZ	PMX, NBY, PBZ			
NMV			NBX, NMY, PBZ	X0, NMY, PBZ	PBX, NMY, PBZ				
NAV				X0, NAY, PBZ					
Vertical EOG	**NAH**	**NMH**	**NBH**	**ZH**	**PBH**	**PMH**	**PAH**	**Horizontal EOG**	
PAV			NBX, PAY, Z0	X0, PAY, Z0	PBX, PAY, Z0			ZV	Vertical EOG
PMV		NMX, PMY, Z0	NBX, PMY, Z0	X0, PMY, Z0	PBX, PMY, Z0	PMX, PMY, Z0			
PBV	NAX, PBY, Z0	NMX, PBY, Z0	NBX, PBY, Z0	X0, PBY, Z0	PBX, PBY, Z0	PBX, PBY, Z0	PAX, PBY, Z0		
ZV	NAX, Y0, Z0	NMX, Y0, Z0	NBX, Y0, Z0	X0, Y0, Z0	PBX, Y0, Z0	PMX, Y0, Z0	PAX, Y0, Z0		
NBV	NAX, NBY, Z0	NMX, NBY, Z0	NBX, NBY, Z0	X0, NBY, Z0	PBX, NBY, Z0	PMX, NBY, Z0	PAX, NBY, Z0		
NMV		NMX, NMY, Z0	NBX, NMY, Z0	X0, NMY, Z0	PBX, NMY, Z0	PMX, NMY, Z0			
NAV			NBX, NAY, Z0	X0, NAY, Z0	PBX, NAY, Z0				
Vertical EOG	**NAH**	**NMH**	**NBH**	**ZH**	**PBH**	**PMH**	**PAH**	**Horizontal EOG**	
PAV				X0, PAY, NBZ				NBV	Vertical EOG
PMV			NBX, PMY, NBZ	X0, PMY, NBZ	PBX, PMY, NBZ				
PBV		NMX, PBY, NBZ	NBX, PBY, NBZ	X0, PBY, NBZ	PBX, PBY, NBZ	PBX, PBY, NBZ			
ZV		NMX, Y0, NBZ	NBX, Y0, NBZ	X0, Y0, NBZ	PBX, Y0, NBZ	PMX, Y0, NBZ			
NBV		NMX, NBY, NBZ	NBX, NBY, NBZ	X0, NBY, NBZ	PBX, NBY, NBZ	PMX, NBY, NBZ			
NMV			NBX, NMY, NBZ	X0, NMY, NBZ	PBX, NMY, NBZ				
NAV				X0, NAY, NBZ					
Vertical EOG	**NAH**	**NMH**	**NBH**	**ZH**	**PBH**	**PMH**	**PAH**	**Horizontal EOG**	
PMV				X0, PMY, NMZ				NMV	Vertical EOG
PBV			NBX, PBY, NMZ	X0, PBY, NMZ	PBX, PBY, NMZ				
ZV			NBX, Y0, NMZ	X0, Y0, NMZ	PBX, Y0, NMZ				
NBV			NBX, NBY, NMZ	X0, NBY, NMZ	PBX, NBY, NMZ				
NMV				X0, NMY, NMZ					
Vertical EOG	**NAH**	**NMH**	**NBH**	**ZH**	**PBH**	**PMH**	**PAH**	**Horizontal EOG**	
PBV				X0, PBY, NAZ				NAV	Vertical EOG
ZV				X0, Y0, NAZ					
NBV				X0, NBY, NAZ					

Figure 13A), to control the robot to follow a trajectory defined by 5 points (**Figure 13B**). Each user performs twenty repetitions. A camera is placed on the end effector, and a program for detecting red color is added to the interface in real time. Each point has an internal number that defines the order that the robot must follow to indicate them; when the first red color point is detected, a timer is activated to take the time of the execution of the task. For evaluating adaptability of the classifier, it was necessary to compare the time of the execution of the twenty repetitions of the 30 users, in each experiment.

In the first experiment, the glasses are placed on each user and a sample of the EOG signal is taken for 25 s to generate a database with the 30 users, and the average voltage of the EOG signals is calculated for the maximum and minimum threshold

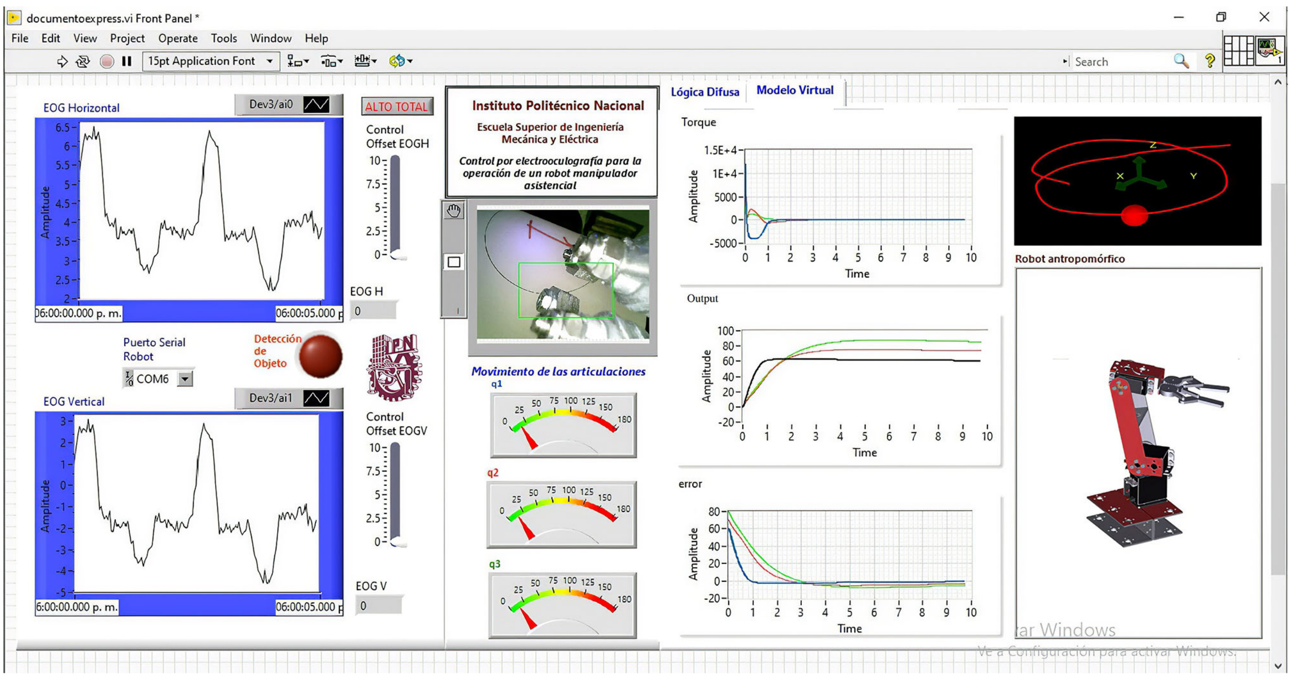


FIGURE 12 | Graphical interface that provides system information to the user.

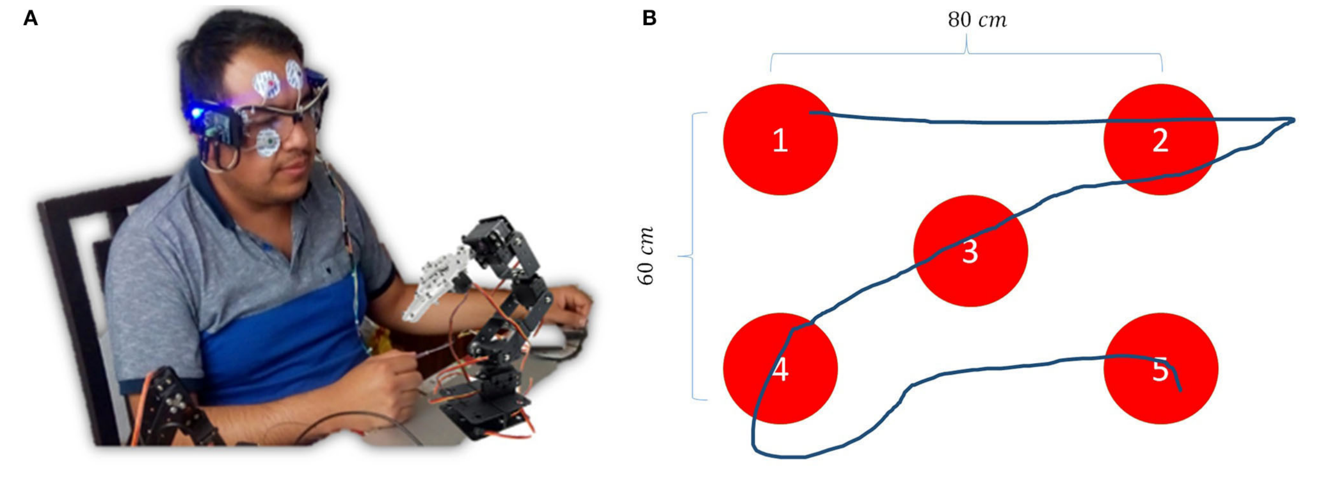


FIGURE 13 | **(A)** Control of the physical manipulator robot by EOG. **(B)** Proposed trajectory for the validation of the system by following the gaze.

values. The system is calibrated once, and all users need to do a workout to reach the required thresholds. In other words, in this first experiment, the user must adapt to the HMI in order to operate the assistance system.

In the second experiment, the system, by optimizing the signal thresholds, is automatically calibrated every 3 min, adapting the fuzzy classifier to the parameters of the EOG signal of each individual. The optimal range data becomes the input of the classifier; the calibration process is imperceptible to the user and does not affect the operation control of the assistance system since it only lasts 0.53 s. In this second experiment, the HMI adapts to each user and the variability of its EOG signal.

A third experiment was realized with the users of the second experiment, who had previous training to analyze the performance of users with experience in executing the task and evaluate if with only 20 training tests the execution time of this is considerably reduced.

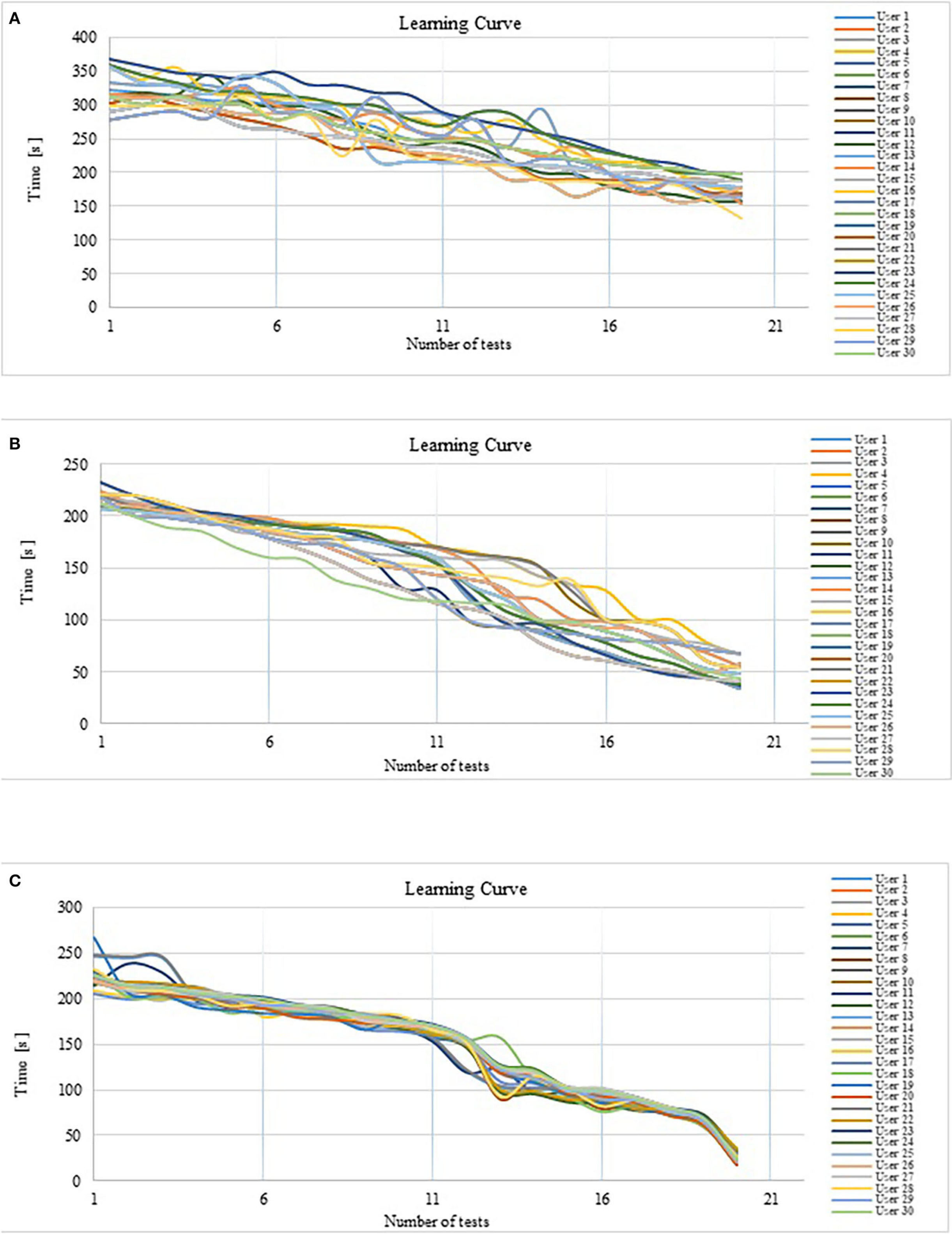
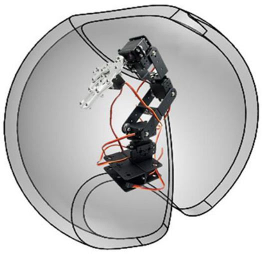


FIGURE 14 | (A) Response time trend without optimization of the fuzzy classifier. **(B)** Response time trend with fuzzy classifier optimization. **(C)** Response time trend with fuzzy classifier optimization and expert users.

TABLE 3 | Physical characteristics of the robot.

Physical robot dimensions	Position range of robot joints	Robot implemented
$l_1 = 0.18\,m$	$q_1 = 20 - 70°$	
$l_2 = 0.28\,m$	$q_2 = 30 - 230°$	
$l_3 = 0.\ 15\,m$	$q_3 = 20 - 270°$	

Experiment 1: Standard Calibration With Inexpert and Expert Users

In this experiment, 30 different EOG signals were obtained. The average of the maximum and minimum thresholds of the user voltage was calculated; the result gave a value of 1.123 volts for the maximum threshold and 0.3212 volts for the minimum threshold. The fuzzy classifier was calibrated with these databases, and the same 30 users were asked to perform the test. When making the first attempts, the users were unable to control the operation of the robot and complete the trajectory; it was necessary to do prior training in the use of the HMI and to manually adjust the thresholds of the fuzzy classifier for each user on the average value obtained for get them to complete the test. When they had the necessary training, time was taken in 20 repetitive tests.

As seen in **Figure 14A** when starting the experiment, the average execution time of the task was 322.22 s; after 20 repetitions, the average was 175.7 s. In **Figure 14A**, the tendency to decrease the execution time to realize the path tracking is observed. Task execution time average was reduced by 45.5% after twenty tests. The standard deviation of the recorded time is 55.56, which indicates that there is considerable variation in relation to the average. This is because each user tries to adapt to the thresholds already preestablished in the system.

Table 4 shows the response time of a sample of 7 users out of 30 users who carried out the experiment. The time it takes to perform the 5-point tracking experiment is represented in the rows for each user. While in the column, the number z test is indicated.

In the analysis of the results, it is observed that there is a decrease in the response time resulting after test number 6, but a dispersion in the trend is observed, that is, the user did not achieve a good control of the operation of the robot until the repetition number 16. In test 16, it is observed how this dispersion decreases. This variability explains why the system does not respond adequately until the user reaches the voltage thresholds at which the classifier is calibrated. Most HMIs work on this principle; they are calibrated using information stored in a database, even if the user has different parameters from those stored, the system responds with close values, increasing the time in which a control command is

TABLE 4 | Value of the response trend of inexperienced users by manual calibration of the fuzzy classifier.

	Response in time of experimental tests																			
	Test 1	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Test 6**	**Test 7**	**Test 8**	**Test 9**	**Test 10**	**Test 11**	**Test 12**	**Test 13**	**Test 14**	**Test 15**	**Test 16**	**Test 17**	**Test 18**	**Test 19**	**Test 20**
User 1	322.90	319.20	315.32	307.38	306.38	300.12	298.13	280.34	268.38	250.34	249.28	239.83	218.39	200.38	198.38	180.38	170.39	168.38	158.82	158.39
User 5	358.38	343.38	332.39	320.29	318.38	314.38	309.39	300.28	298.39	278.38	268.39	288.29	287.38	258.39	239.29	228.38	218.39	200.39	198.39	188.39
User 10	278.39	286.45	291.38	281.39	329.69	290.89	289.98	270.14	312.45	267.89	254.32	278.98	214.38	220.86	218.43	200.78	176.45	187.89	167.56	165.90
User 15	368.38	358.38	348.28	344.38	339.30	349.38	330.39	329.38	318.38	315.38	289.38	278.38	268.38	258.38	248.38	232.38	219.28	213.29	200.29	198.28
User 20	300.89	312.33	299.80	289.08	278.65	268.54	253.40	234.57	236.78	228.12	217.45	214.37	212.38	190.29	189.76	187.45	188.09	189.45	170.45	169.02
User 25	357.38	332.90	328.38	316.38	343.38	332.29	298.38	288.39	218.29	216.39	215.39	214.59	213.38	210.12	208.32	198.34	189.54	184.38	182.18	178.38
User 30	312.76	300.45	312.40	300.75	299.89	278.13	288.12	267.34	258.54	248.80	246.50	248.90	236.78	228.76	219.48	213.45	208.54	206.43	200.12	198.34

generated because a search must be made for the closest parameter and then generate a response by activating the actuators of the system to be controlled. In addition, extensive user training is required to adapt as quickly as possible to the HMI calibration parameters.

Experiment 2: Customized Calibration With Inexpert Users

For the second experiment, the times it took to perform the test for 30 new users were obtained, but in this case, the intelligent calibration system developed from the modeling of the voltage thresholds was used. The fuzzy inference system is automatically calibrated for each user every 3 min, from the first EOG signal acquisition until the test ends. This is a parallel process, and interrupting the control routine for a period of 0.53 s, the genetic algorithm obtains the thresholds and the optimal range is the new fuzzy classifier input, customizing the system and adapting the control to individual parameters, including when there are disturbances in the EOG signal due to external interference. The interruption time for calibration is imperceptible by the user and negligible, compared to the response time of the controlled device. The user does not require prior training to generate some skill in controlling the device, because the classifier is constantly calibrating.

In this experiment, the average time it took the 30 users to follow the path when starting the test was 215.53 s; after a series of 20 repetitions, the average execution time was 48.51 s. In **Table 5**, a summary of the response time of Experiment 2 is presented and it can be observed that the execution time is much less than the average obtained in the first experiment. As seen in **Figure 14B**, from the first test there was a tendency to decrease the average time by 77.55%, a value considerably higher than that observed in the first experiment. The standard deviation of the recorded time is 41.3, which indicates less variability in the response of different users using the optimal calibration for the classifier. In **Figure 14B**, can be see that the standard deviation is reduced after of test number 18, this indicates that the dispersion of the data is decreased, which suggests that all the users adapted to the system at the end of the Experiment 3: Customized calibration with expert users.

A third experiment was carried out with the 30 users who carried out the second experiment, and very significant results were obtained. In **Table 6**, a summary of the response time of Experiment 3 is presented. The average response time when starting the test of the 30 users is 224.15 s, after a series of 20 repetitions, it is verified that the tendency to decrease the execution time has an average of 24.09 s. With previous training, the average time to follow a new path decreased the response time of the robot by 89.26%, it can be observed in the graphs in **Figure 14C**. When analyzing the results presented, a significant improvement in the dispersion of the response is observed due to the decrease in the standard deviation of 29.8, which indicates a greater domain in the control of the system by users, especially after test 16. In **Figure 14C**, in the last two tests in the 30

TABLE 5 | Response trend value of inexperienced users through automated calibration of the fuzzy classifier.

	Response in time of experimental tests																			
	Test 1	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Test 6**	**Test 7**	**Test 8**	**Test 9**	**Test 10**	**Test 11**	**Test 12**	**Test 13**	**Test 14**	**Test 15**	**Test 16**	**Test 17**	**Test 18**	**Test 19**	**Test 20**
User 1	210.23	208.27	206.32	200.23	196.34	194.76	190.56	188.90	178.56	170.67	155.23	120.23	100.23	89.34	78.34	68.57	58.34	49.47	46.45	34.33
User 5	232.87	219.89	208.45	204.33	200.56	193.45	188.49	186.43	178.45	166.45	154.32	123.38	97.33	96.34	78.66	66.34	54.33	46.72	44.22	36.40
User 10	232.87	219.89	208.45	204.33	200.56	193.45	188.49	186.43	178.45	166.45	154.32	123.38	97.33	96.34	78.66	66.34	54.33	46.72	44.22	36.40
User 15	217.38	214.34	210.32	202.34	185.54	183.34	178.02	173.43	164.33	162.34	160.34	158.22	157.39	142.57	133.67	99.59	89.32	80.78	76.32	66.12
User 20	212.57	200.45	199.37	193.45	189.45	178.45	167.34	154.45	138.45	129.34	115.45	110.39	100.98	78.49	65.45	60.45	54.45	50.56	43.23	40.45
User 25	207.33	205.43	201.33	198.47	189.32	188.39	183.22	180.42	176.59	168.93	160.46	133.58	120.78	100.47	98.49	89.43	79.78	65.43	53.39	48.34
User 30	213.34	200.23	189.34	185.89	170.34	160.34	158.48	138.34	130.45	120.23	118.45	116.23	114.43	100.09	98.34	89.34	78.34	65.43	50.54	43.23

users, a significant decrease in the execution time of the task is observed.

With this HMI, the user does not have to worry about reaching the required voltage levels or need prior training to control the robot, on the contrary, the HMI adapts to the operating thresholds of each user, generating a response from the robot throughout its workspace.

CONCLUSION

In this work, an intelligent calibration system is presented by means of which an HMI interface whose control input is the EOG signal adapts to the characteristics of the signals of different users and generates trajectories in the workspace of an anthropomorphic robot manipulator in real time. The difference from other HMIs is that the proposed system does not need a database for its calibration. The innovation is the intelligent system capable of calibrating the HMI from the use of fast neural networks to model the physiological signal and its optimization with genetic algorithms to obtain amplitude thresholds that allow easy adaptation of the HMI to the EOG signal of the user. It is verified that the use of artificial intelligence to generate trajectories from signals with high variability, such as EOG results in a decrease in the execution time of a task and the sensation of real-time control of the robot. It was shown, from the observation of data obtained by experimentation, that the adaptive calibration system generates response times in the robotic system to be controlled less than when the user is trained to use standard calibrated systems. When comparing **Figure 14C** with **Figure 14A**, the decrease in task execution time is observed. For the first experiment (users with manual calibration experience), the average decrease in task execution time is 44%, and for the third experiment (users with adaptive calibration experience) it is 82%. In addition, using an intelligent system reduces training time, since the user does not have to adapt to the HMI if not the HMI adapting to the user. In **Figure 14A**, a large dispersion of data is observed (standard deviation), indicating that each user tries to adapt differently to the HMI. In contrast, **Figure 14C** shows a significant reduction in data dispersion (standard deviation), since each user manages to control the system adequately, since the HMI adapts to the characteristics of its EOG signal.

In the graphs of **Figure 14**, it is observed how the adaptability of the system improves; in the first experiment, the calibration was done manually, although it presents a decrease in response time, where it takes the user more time to reach the objective set; however, the fuzzy logic allows adaptability to personal characteristics. The second experiment has been worked with inexperienced users who had no control over the system, but calibrating the system from modeling the signal and optimizing the range of signal variability, it is observed that the response time is less and the level of adaptability is verified by decreasing the measure of dispersion of each of the responses. The system tends to standardize the learning curve to the same pattern regardless of the individual who uses the HMI; this property

TABLE 6 | Response trend value of expert users by automated calibration of the fuzzy classifier.

	Response in time of experimental tests																			
	Test 1	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Test 6**	**Test 7**	**Test 8**	**Test 9**	**Test 10**	**Test 11**	**Test 12**	**Test 13**	**Test 14**	**Test 15**	**Test 16**	**Test 17**	**Test 18**	**Test 19**	**Test 20**
User 1	245.49	243.48	243.93	200.23	196.34	194.76	190.56	188.9	178.56	170.67	155.23	120.23	102.23	99.99	100.34	97.56	86.46	76.12	66.59	30.19
User 5	220.19	209.38	208.45	204.33	200.56	193.45	188.49	186.43	178.45	172.03	166.28	151.23	122.98	119.28	98.38	97.32	88.28	76.38	65.38	21.99
User 10	224.35	212.24	210.78	204.57	200.35	198.45	190.78	186.78	179.49	174.45	168.89	154.34	124.45	119.56	100.34	98.56	89.46	78.34	68.43	23.38
User 15	219.19	206.38	197.45	212.33	200.56	189.45	178.49	176.43	169.45	170.03	164.28	150.23	119.98	117.28	93.38	94.32	86.28	72.38	63.38	17.99
User 20	221.35	210.24	204.78	200.57	193.35	189.45	179.78	177.78	173.49	172.45	168.89	151.34	119.45	116.56	96.34	93.56	86.46	72.34	62.43	18.38
User 25	220.19	209.38	208.45	204.33	200.56	193.45	188.49	186.43	178.45	172.03	166.28	151.23	122.98	119.28	98.38	97.32	88.28	76.38	65.38	21.99
User 30	225.52	213.41	211.95	205.74	201.52	199.62	191.95	187.95	180.66	175.62	170.06	155.51	125.62	120.73	101.51	99.73	90.63	79.51	69.6	24.55

of the modeling of the EOG signal to customizing the fuzzy classifier can be seen in the results of Experiment 3, in the graph of **Figure 14C** where the response time decreases to an average value of 24 s and the standard deviation measure is reduced.

The experiment was performed using an anthropomorphic robot to validate the HMI response, but since the fuzzy classifier generates coordinates in a Cartesian space (in three dimensions), it can be adapted to any navigation system by modifying only the mapping in the workspace, generating trajectories for example for autonomous vehicles or intelligent spatial location systems for the control of wheelchairs or any type of mobile robot.

In a future work, this HMI would be implemented in assistance systems for people with severe disabilities, by implementing an eye joystick system in order to accomplish everyday tasks, such as taking objects.

DATA AVAILABILITY STATEMENT

All datasets generated for this study are included in the article/supplementary material.

ETHICS STATEMENT

Ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements. The patients/participants provided their written informed consent to participate in this study. Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

AUTHOR CONTRIBUTIONS

FP, PN, OA, MC, EV, and EP conceived, designed, performed the experiments, analyzed the data, and wrote the paper. All authors contributed to the article and approved the submitted version.

FUNDING

The present research has been partially financed by Instituto Politécnico Nacional and by Consejo de Ciencia y Tecnología, CONACYT, co-financed with FEDER funds.

REFERENCES

An, L., Tang, Y., Wang, D., Jia, S., Pei, Q., Wang, Q., et al. (2020). Intrinsic and synaptic properties shaping diverse behaviors of neural dynamics. *Front. Comp. Neurosci.* 14:26. doi: 10.3389/fncom.2020.00026

Ang, K. K., Chua, K. S., Phua, K. S., Wang, C., Chin, Z. Y., Kuah, C. W., et al. (2015). A randomized controlled trial of EEG-based motor imagery brain-computer interface robotic rehabilitation for stroke. *Clin. EEG Neurosci.* 46, 310–320. doi: 10.1177/1550059414522229

Buchli, J., Theodorou, E., Stulp, F., and Schaal, S. (2011). "Variable impedance control a reinforcement learning approach," in *Robotics: Science and Systems VI*, eds Y. Matsuoka, H. Durrant-Whyte, and J. Neira (Cambridge, MA: MIT Press), 153–160.

Ding, X., and Lv, Z. (2020). Design and development of an EOG-based simplified Chinese eye-writing system. *Biomed. Sign. Proc. Control* 57:101767. doi: 10.1016/j.bspc.2019.101767

Dipietro, L., Ferraro, M., Palazzolo, J. J., Krebs, H. I., Volpe, B. T., and Hogan, N. (2005). Customized interactive robotic treatment for stroke: EMG-triggered therapy. *IEEE Trans. Neural Syst. Rehabil. Eng.* 13, 325–334. doi: 10.1109/TNSRE.2005.850423

Djeha, M., Fazia, S., Mohamed, G., Khaled, F., and Noureddine, A. (2017). "A combined EEG and EOG signals based wheelchair control in virtual environment," in *2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)* (Boumerdes: IEEE), 1–6. doi: 10.1109/ICEE-B.2017.8192087

Gopinathan, S., Ötting, S. K., and Steil, J. J. (2017). A user study on personalized stiffness control and task specificity in physical human–robot interaction. *Front. Robot.* 4:58. doi: 10.3389/frobt.2017.00058

Gray, V., Rice, C. L., and Garland, S. J. (2012). Factors that influence muscle weakness following stroke and their clinical implications: a critical review. *Physiother. Can.* 64, 415–426. doi: 10.3138/ptc.2011-03

Kumar, K., Rajkumar, D. T., Ilayaraja, M., and Shankar, K. (2018). A review-classification of electrooculogram based human computer interfaces. *Biomed. Res.* 29, 1079–1081. doi: 10.4066/biomedicalresearch.29-17-2979

Leardi, R. (2003). *Nature-Inspired Methods in Chemometrics: Genetic Algorithms and Artificial Neural Networks*. Genova: Elsevier.

Lu, Y., Zhang, C., Zhou, X.-P., Gao, X, P., and Lv, Z. (2018). A dual model approach to EOG-based human activity recognition. *Biomed. Signal Process. Control* 45, 50–57. doi: 10.1016/j.bspc.2018.05.011

Lum, P. S., Godfrey, S. B., Brokaw, E. B., Holley, R. J., and Nichols, D. (2012). Robotic approaches for rehabilitation of hand function after stroke. *Am. J. Phys. Med. Rehabil.* 91, S242–S254. doi: 10.1097/PHM.0b013e31826bcedb

Medina, J. R., Lawitzky, M., Mörtl, A., Lee, D., and Hirche, S. (2011). "An experience-driven robotic assistant acquiring human knowledge to improve haptic cooperation," in *International Conference on Intelligent Robots and Systems (IROS)* (San Francisco, CA), 2416–2422. doi: 10.1109/IROS.2011.6095026

Rozo, L., Bruno, D., Calinon, S., and Caldwell, D. G. (2015). "Learning optimal controllers in human-robot cooperative transportation tasks with position and force constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Hamburg: IEEE), 1024–1030. doi: 10.1109/IROS.2015.7353496

Xiao, R., and Ding, L. (2013). Evaluation of EEG features in decoding individual finger movements from one hand. *Comput. Math. Methods Med.* 2013:243257. doi: 10.1155/2013/243257

Zhang, J., Wang, B., Zhang, C., Xiao, Y., and Wang, M. Y. (2019). An EEG/EMG/EOG-based multimodal human-machine interface to real-time control of a soft robot hand. *Front. Neurorobot.* 13:7. doi: 10.3389/fnbot.2019.00007

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.

The reviewer LS declared a shared affiliation, though no other collaboration, with several of the authors FP, PN, MC, EV, and EP to the handling Editor.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 02 October 2020
doi: 10.3389/fnbot.2020.00063

The Path Planning of Mobile Robot by Neural Networks and Hierarchical Reinforcement Learning

Jinglun Yu, Yuancheng Su and Yifan Liao*

Chongqing University-University of Cincinnati Joint Co-op Institute, Chongqing University, Chongqing, China

Existing mobile robots cannot complete some functions. To solve these problems, which include autonomous learning in path planning, the slow convergence of path planning, and planned paths that are not smooth, it is possible to utilize neural networks to enable to the robot to perceive the environment and perform feature extraction, which enables them to have a fitness of environment to state action function. By mapping the current state of these actions through Hierarchical Reinforcement Learning (HRL), the needs of mobile robots are met. It is possible to construct a path planning model for mobile robots based on neural networks and HRL. In this article, the proposed algorithm is compared with different algorithms in path planning. It underwent a performance evaluation to obtain an optimal learning algorithm system. The optimal algorithm system was tested in different environments and scenarios to obtain optimal learning conditions, thereby verifying the effectiveness of the proposed algorithm. Deep Deterministic Policy Gradient (DDPG), a path planning algorithm for mobile robots based on neural networks and hierarchical reinforcement learning, performed better in all aspects than other algorithms. Specifically, when compared with Double Deep Q-Learning (DDQN), DDPG has a shorter path planning time and a reduced number of path steps. When introducing an influence value, this algorithm shortens the convergence time by 91% compared with the Q-learning algorithm and improves the smoothness of the planned path by 79%. The algorithm has a good generalization effect in different scenarios. These results have significance for research on guiding, the precise positioning, and path planning of mobile robots.

Keywords: neural network, hierarchical reinforcement learning, mobile robot, path planning, fusion algorithm

OPEN ACCESS

Edited by:
Mu-Yen Chen,
National Taichung University of Science and Technology, Taiwan

Reviewed by:
Yinyan Zhang,
Jinan University, China
Kuan-Yu Lin,
Ling Tung University, Taiwan

****Correspondence:***
Jinglun Yu
yujinglun2016@163.com

Received: *28 June 2020*
Accepted: *05 August 2020*
Published: *02 October 2020*

Citation:
Yu J, Su Y and Liao Y (2020) The Path Planning of Mobile Robot by Neural Networks and Hierarchical Reinforcement Learning. Front. Neurorobot. 14:63. doi: 10.3389/fnbot.2020.00063

INTRODUCTION

Mobile robot autonomous navigation can be divided into three subsystems: information perception, behavior decision-making, and manipulation control. Path planning is the basis of mobile robot navigation and control (Ghosh et al., 2017; Orozco-Rosas et al., 2019). The goal of mobile robot path planning is to find a path from the current position to the target position. The path should be as short as possible, the smoothness of the path should meet the dynamics of the mobile robot, and the safety of the path should be collision-free (Han and Seo, 2017).

Depending on how much information is known about the environment in the path planning process, path planning can be divided into global path planning and local path planning (Li and Chou, 2018). There are many methods of path planning. According to specific algorithms and strategies, path planning algorithms can be roughly divided into four types: template matching, artificial potential field, map construction, and artificial intelligence (Zhao et al., 2018). Each type of path planning algorithm has an optimal application scenario and limitations. The current path planning of mobile robots relies heavily on the surrounding environment. In addition to the limitations of traditional path planning, robots cannot complete their learning and judgment in complex environments, a bottleneck in the development of research in this field (Bakdi et al., 2017). It is therefore particularly important to develop a path planning method with low reliance on the environment, which can quickly adapt to the surrounding environment.

The Deep Q-Learning Network (DQN) is a way of modeling the environment and calculating the collision energy function, which is the main cause of a loss in functionality (Ohnishi et al., 2019). To realize the path planning process, the neural network is trained to minimize the loss function through the gradient descent method. To enable better generalization ability in the neural network, various sample data are needed for learning and training, however, an over large data sample will increase the training time (Shen et al., 2019a; Sung et al., 2020).

Deep Reinforcement Learning (DRL), as an important machine learning method, has received more attention and there are increasing applications of it in robot path planning DRL (Arulkumaran et al., 2017). The agent obtains knowledge through the exploration of an environment and learns using a process of trial and error. The DRL method has obvious advantages in path planning and requires less prior information about the environment (Wulfmeier et al., 2017; Zheng and Liu, 2020).

Unlike the supervised learning method, reinforcement learning does not require much sample data for training, like neural network methods, and acquires sample data during the training process. In recent years, scholars have focused on using new algorithms or fusion algorithms to improve the performance of mobile robots (Yan and Xu, 2018). Lei et al. found that adding the Q-Learning algorithm to the reinforcement learning path enhances the ability of robots to dynamically avoid obstacles and local planning in the environment (Lei et al., 2018; Liu et al., 2019). Wang et al. found that compared with Distributed DQN (DDQN) algorithm, the Tree Double Deep Network (TDDQN) has the advantages of fast convergence speed and low loss (Wang P. et al., 2020). By using a neural network to strengthen the learning path planning system, Wen et al. suggested that the mobile robot can be navigated to a target position without colliding with any obstacles and other mobile robots, and this method was successfully applied to the physical robot platform (Wen et al., 2020). Botteghi et al. introduced a reward function training strategy in the fusion algorithm, which not only outperformed the standard reward function in terms of convergence speed but also reduced the number of collisions by 36.9% of iteration steps (Shen et al., 2019b; Botteghi et al., 2020). Therefore, the fusion algorithm has obvious advantages in path planning and algorithm performance. However, the path planning performance of current fusion algorithms is not outstanding.

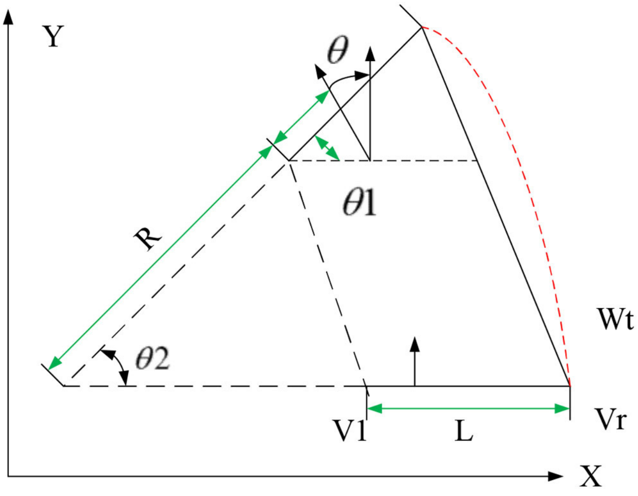


FIGURE 1 | The path planning motion model of mobile robots.

Taking into account the shortcomings of these research results, we designed a mobile robot path planning system based on neural networks and hierarchical reinforcement learning. Through neural networks, this system perceives the environment and performs feature extraction to realize the fitting from the environment to the state action function (Chen, 2018). The mapping of the current state to the action of the hierarchical reinforcement learning is satisfied through the enhancement function, thereby realizing the demand for mobile robots. Theoretically, the organic combination of the two can improve the performance of mobile robots in path planning. Therefore, in this study, the algorithm was embedded into a mobile robot, and the designed algorithm was verified by comparing it with other path planning algorithms in different environments and scenarios. The initial *Q*-value of the proposed algorithm sped up the convergence speed, redefined the number of states, as well as the direction of motion, and step length. The real-time performance of the mobile robot's path planning and smoothness was significantly improved, and could be used to guide robot movement, and improve algorithm mobility (Liu and Wang, 2019).

METHODS

Mobile Robot Path Planning Model

The path planning task explored in this study is based on a two-wheel differential mobile robot. The robot can control the speed of its two driving wheels to achieve arbitrary trajectory movements such as linear movement, turning, and turning around in circles. **Figure 1** shows the pose of the robot at adjacent time intervals, based on which kinematic model is established.

The world coordinate system pose of the mobile robot at time t is set to $Wt = \left[x_t, y_t, \theta_t\right]^T$; if the world coordinate pose of the mobile robot at time $t + \Delta t$ is$W_{t+\Delta \mathrm{t}} = \left[x_{t+\Delta \mathrm{t}}, y_{t+\Delta \mathrm{t}}, \theta_{t+\Delta \mathrm{t}}\right]^T$, the distance between the left and right driving wheels is L, the speeds of the left and right driving wheels are v_l and [[Mathtype-mtef1-eqn-5.mtf]], and the robot linear speed and angular speed are respectively v and ω, the speed v of the mobile robot in the ideal motion state is:

$$v = \frac{v_l + v_r}{2} \tag{1}$$

The angular velocity of the robot is:

$$\omega = \frac{v_l \text{-} v_r}{L} \tag{2}$$

The instantaneous curvature radius R is:

$$R = \frac{v}{\omega} \tag{3}$$

As shown in **Figure 1**, θ_1=θ_2=θ, after Δt, the heading angle of the robot changes as follows:

$$\theta_{\mathrm{t}+\Delta \mathrm{t}} = \theta_t + \theta \tag{4}$$

The motion from position $Wt = \left[x_t, y_t, \theta_t\right]^T$ to $W_{t+\Delta \mathrm{t}} = \left[x_{t+\Delta \mathrm{t}}, y_{t+\Delta \mathrm{t}}, \theta_{t+\Delta \mathrm{t}}\right]^T$ can be regarded as a circular arc with radius R. If the arc is used to approximate the actual trajectory of the robot, the geometric relationship should be:

$$\begin{bmatrix} x_{t+\Delta t} \\ y_{t+\Delta t} \\ \theta_{t+\Delta t} \end{bmatrix} = \begin{bmatrix} x_t + R(\sin(\theta_t + \theta) - \sin\theta_{\mathrm{t}}) \\ y_t + R(\cos(\theta_{\mathrm{t}} + \theta) - \cos\theta_{\mathrm{t}}) \\ \theta_t + \theta \end{bmatrix}, \theta \neq 0 \tag{5}$$

Combining the above equations, the motion equation of the differential mobile robot can be obtained as:

$$\begin{bmatrix} x_{t+\Delta t} \\ y_{t+\Delta t} \\ \theta_{t+\Delta t} \end{bmatrix} = \begin{bmatrix} x_t + \frac{L(v_r+v_l)}{2(v_r-v_l)}(R(\sin(\theta_t + \theta) - \sin\theta_{\mathrm{t}}) \\ y_t + \frac{L(v_r+v_l)}{2(v_r-v_l)}R(\cos(\theta_{\mathrm{t}} + \theta) - \cos\theta_{\mathrm{t}}) \\ \theta_t + \theta \end{bmatrix}, \theta \neq 0 \tag{6}$$

ANN

ANN is a mathematical or computational model that simulates the structure and function of biological neural networks, which is used to estimate or approximate functions. With the continuous deepening of research works on ANNs, it has made great breakthroughs in the fields of speech recognition, pattern recognition, automatic control, and predictive estimation. ANN has successfully solved many problems that are difficult for computers to solve, showing good performance.

In the practical application of ANN, most neural network models use a backpropagation neural network (BPNN) and its transformations, which have good nonlinear mapping ability, self-learning ability, and fault tolerance. It mainly uses many aspects such as pattern recognition, function approximation, data compression, prediction estimation, and classification. Therefore, the most representative BPNN is chosen as the basis of modeling to analyze the robot path. An ANN is usually composed of multiple BPNN layers and multiple neurons, which are mainly divided into an input layer, a hidden layer, and an output layer, where the input vector should be:

$$x = \left[x_1, x_2, x_3 ... x_i, ... x_m\right], i = 1, 2, ...m \tag{7}$$

The output vector should be:

$$y = \left[y_1, y_2, y_3 ... y_{\mathrm{k}}, ... y_n\right], \mathrm{k} = 1, 2, ...n \tag{8}$$

The neuron input of the hidden layer should be:

$$\mathrm{h}^{(1)} = \left[\mathrm{h}^{(1)}{}_1, \mathrm{h}^{(1)}{}_2, \mathrm{h}^{(1)}{}_3 ... \mathrm{h}^{(1)}{}_j, ... \mathrm{h}^{(1)}{}_{sl}\right], j = 1, 2, ...sl \tag{9}$$

Where: sl is the number of neurons in layer 1; assuming that $\mathrm{w}^{(1)}{}_{\mathrm{ij}}$ is the connection weight between the j-th neuron in layer 1-1, $\mathrm{b}^{(1)}{}_{\mathrm{i}}$ is the threshold of the i-th neuron in layer 1, and $net^{(l)}{}_i$ is the input of the i-th neuron in layer 1, then the following equation is obtained:

$$\mathrm{h}^{(1)}{}_{\mathrm{i}} = f(net^{(l)}{}_i) \tag{10}$$

$$net^{(l)}{}_i = \sum_{j=1}^{sl-1} w^{(l)}{}_{ij} \mathrm{h}^{(l\text{-}1)}{}_{\mathrm{j}} + \mathrm{b}^{(l)}{}_i \tag{11}$$

Here, the functions of the input layer to the output layer use the S-type corresponding TANSIG function, the output layer uses the PURELIN linear function, the learning rules use the TRINGDX function, and the performance evaluation uses the MES function, where the model number is set to 1,000 times and the accuracy is set to 0.0001. The rest are the default parameters of the system, and the specific structure is shown in **Figure 2**.

Reinforcement Learning

Reinforcement learning is a machine learning method that learns by interacting with the environment. An Agent uses reinforcement learning methods to learn, which is to acquire knowledge from a sequence of actions obtained by exploration. Its sample data is not existing, meaning it is different from the supervised learning process. After an Agent executes an action, it will get feedback from the environment. This feedback is the evaluation of the action made by the environment and is a process of "trial and error." The evaluation of the action made

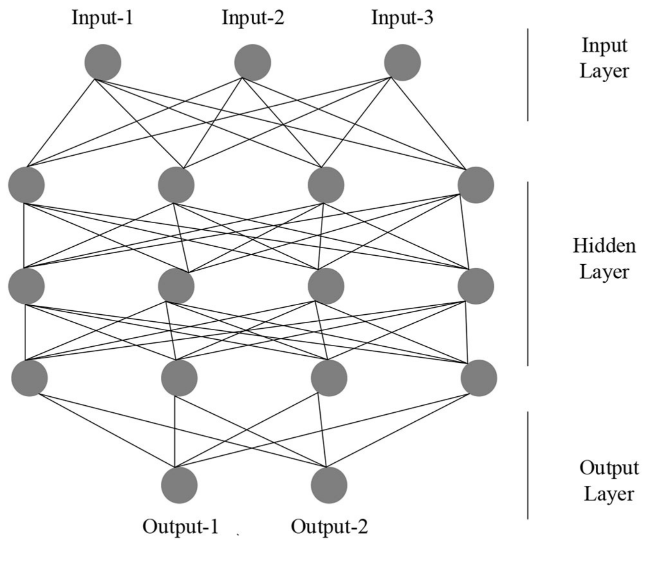


FIGURE 2 | Structure of ANN.

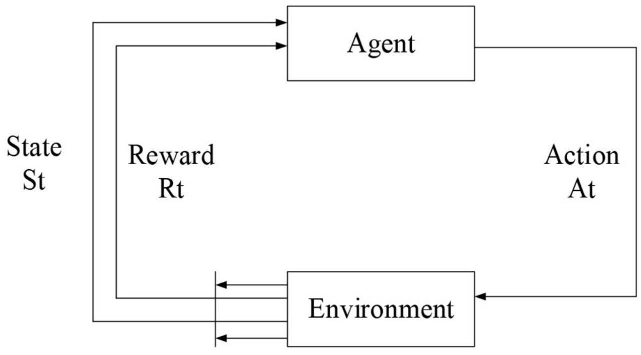


FIGURE 3 | The schematic diagram of the reinforcement learning model.

by the environment is the immediate reward value received by the Agent. The immediate reward is an enhanced signal, which indicates the impact of the execution of this action on the result. The larger the value is, the better the effect is, otherwise it will have a poor impact. The reinforcement learning model is shown in **Figure 3**. The learning process of the reinforcement learning method is a heuristic process. It continuously tries through random units, searches for the optimal action to obtain the enhanced signal of the environment, and increases the probability that the optimal action is selected by the iterative update, thereby finding a set of optimal solutions (a set of action sequences with the highest reward value).

The reinforcement signal in reinforcement learning comes from the immediate reward of environmental feedback. This reward value indicates the quality of the action performed instead of telling the machine what the correct action is. The process of the machine interacting with the environment can be regarded as a Markov Decision Processing (MDP). As long as the random variable set $\{X_1, X_2, X_3 .... \ X_t\}$ satisfies the following equation, the set will have Markov attributes:

$$\begin{aligned} \mathrm{P}_r(X_{t+1} &= x\,|X_t = x_t, X_{t-1} = x_{t-1}, ...X_1 = x_1) \\ &= \mathrm{P}_r(X_{t+1} = x \big| \, X_t = x_t) \end{aligned} \tag{12}$$

Once the state x is determined, the actions before the state are not correlated to the actions after the state and are independent of each other. Among them, the state set S, the action set A, the reward function R, the state transition function T, and the objective function constitute the MDP. The state process of the transition is as follows:

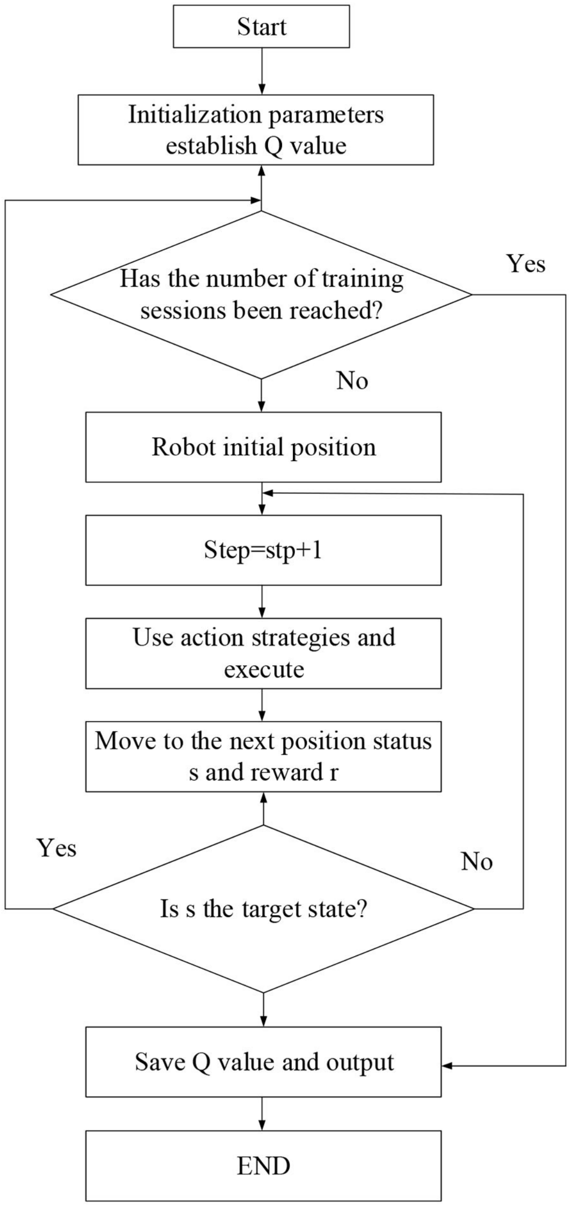

Start
Initialization parameters establish Q value
Has the number of training sessions been reached?
Yes
No
Robot initial position
Step=stp+1
Use action strategies and execute
Move to the next position status s and reward r
Yes
No
Is s the target state?
Save Q value and output
END

FIGURE 4 | Schematic diagram of Q-Learning path planning method.

$$T(s,\alpha,s')=P_r(s_{t+1}=s'\,|s_t=s,\alpha_t=a) \tag{13}$$

The process of Markov's decision is mainly to realize a set of action sequences $\alpha=\pi(s)$ so that the cumulative discount reward $\sum_{t=0}^{\infty}\gamma' R(s_t,a_t)$ reaches the maximum value. Through the iteration of values, the optimization problem of MDP can be realized. The function of the optimal value is defined as:

$$V*(s)=\max(\mathrm{R}(s,\alpha)+\gamma\sum_{s'\in \mathrm{S}}\mathrm{T}(s,a,s')\mathrm{V}*(s')),\forall \mathrm{s}\in \mathrm{S} \tag{14}$$

Then, the optimal strategy is calculated as follows:

$$\pi(s)=\arg|\max\;(\mathrm{R}(s,\alpha)+\gamma\sum_{s'\in \mathrm{S}}\mathrm{T}(s,a,s')\mathrm{V}*(s')) \tag{15}$$

The reinforcement learning system is mainly composed of three parts: reward function, value function, and action selection strategy. Among them, reinforcement function is divided into continuous reward function. By establishing a mathematical model between the state and environmental feedback perceived by the Agent at each moment, the Agent can obtain the evaluation of the environment in each state, giving more guidance information during the Agent training process, and the Agent can find the optimal strategy faster. The calculation is as follows:

$$R_t=f(s_t,i_t) \tag{16}$$

The discrete reward functions require less a-priori information and are simple to construct, which have better applications in exploration and learning in unknown environments. The calculation is as follows:

$$R_t=\begin{cases}\text{1 Perform optimal actions}\\ \text{-1 Perform the worst action}\\ \text{0 Other situations}\end{cases} \tag{17}$$

The reward function only gives the reward of the currently executed action, but this does not guarantee that each action can get a reward. As the training progresses, the value function continuously optimizes and converges, and the action is selected by strategy in a state, which ensures that each action will get not only the largest reward but also the largest cumulative discount reward, of which the limited non-discount cumulative reward function is:

$$V^{\pi}(s_t)=\sum_{t=0}^{h}r_t \tag{18}$$

Where: r_t is the reward immediately obtained by the machine at time t, and the cumulative reward is the accumulation of the immediate rewards obtained from the starting state to the target state. The unlimited discount reward function is:

$$V^{\pi}(s_t)=\sum_{t=0}^{h}\gamma' r_{t+1}0\le\gamma\le 1 \tag{19}$$

Where: γ' is the discount factor, and the value range is $0\le\gamma\le 1$, which represents the limit of reinforcement learning. The value function pays more attention to future rewards. The average reward function is:

$$V^{\pi}(s_t)=\lim_{h\to\infty}(\frac{1}{h}\sum_{t=0}^{h}r_t) \tag{20}$$

After learning, the optimal strategy can use the value function obtained by training to select the action strategy. The equation is as follows:

$$\pi^{*}=\arg\max \mathrm{V}^{\pi}(\mathrm{s}),\forall \mathrm{s}\in \mathrm{S} \tag{21}$$

The action selection strategy of Softmax is used to analyze the probability of the action, which is generally described by the Boltzmann distribution function. The mathematical model is as follows, where T is the temperature control coefficient.

$$p(at/s)=)=\frac{k^{Vi/T}}{\sum_{\alpha\in A}k^{Vi/T}} \tag{22}$$

Different Path Planning Recognition Algorithms

Here, different algorithms are compared to determine the advantages of the proposed algorithm. There are many recognition algorithms for the path planning of mobile robots. These path planning algorithms are all based on the principle of feature point positioning, which changes in any direction of the images mainly through a Gaussian window. Through this movement, the correlation matrix of different windows is calculated and the image data of the environment are obtained.

(1) The Q-Learning algorithm is a table-valued learning algorithm because the state-action Q value table is established during the interaction between the machine and the environment. The reward in the environment will affect the *Q*-value corresponding to the state-action. The *Q*-value of the correct behavior is gradually increased under the positive reward, and the *Q*-value corresponding to the wrong behavior will also be reduced under the negative reward. The optimal action is selected in the action selection strategy to make the Agent obtain the optimal behavior strategy (Wei et al., 2016; Zhu et al., 2017). The method of updating the *Q*-value is as follows in **Figure 4**:

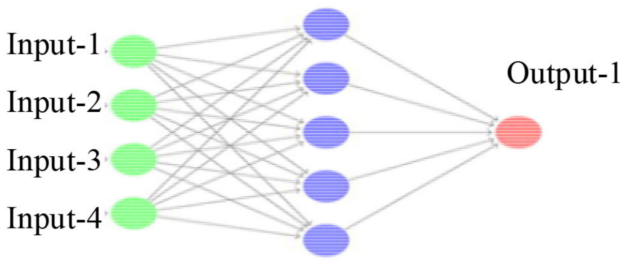


FIGURE 5 | Schematic diagram of the neural network approximation function structure.

(2) The DQN algorithm is a process of using the neural network to approximate the value function. As shown in **Figure 5**, the optimal value function $Q(s, \alpha, \theta)$ is approximated by adjusting the weight of the neural network. The update value function changes the parameters. After the neural network training is completed, the parameters are determined, and the corresponding function value will not change anymore. The training process then converges (Liu and Hodgins, 2017; Zhu et al., 2017). The location update equation is:

$$\theta_{t+1} = \theta_t + \alpha \left[r + \gamma \max Q(s, \alpha, \theta \right] \Delta Q(s, \alpha, \theta) \tag{23}$$

(3) The Potential DQN (PDQN) algorithm is an improvement to the DQN algorithm. Its major purpose is to accelerate the running speed of the algorithm. On this basis, the artificial potential field method is added (Gupta et al., 2019). The gravitational field is calculated as follows:

$$U(X) = \frac{1}{2} k (X - X_g)^2 \tag{24}$$

Where: k is the gain coefficient, X is the current position of the mobile robot, Xg is the target position, j is the planning adjustment reward, and the relationship between reward and gravity is as follows:

$$r = jU(X) \tag{25}$$

(4) Actor-Critic (A3C) algorithm is a way of reinforcement learning. It introduces an evaluation mechanism to solve the high variance problem. It utilizes a neural network to predict the selected action and directly passes the prediction result back to increase the probability that the action is selected next time. If the reward function shows that the selected action is not optimal, the probability that the action is selected next time will be reduced (Haarnoja et al., 2018). The strategy gradient equation is as follows:

$$\Delta_\theta J\theta = \frac{1}{T} \sum_t^T \Delta_\theta \log \pi (\alpha_t \,|s_t; \theta)(\sum_{i=1}^{n} \gamma^{i-1} rt + 1 + v(s_t + n) - v(s_t)) + \beta \Delta_\theta H(\pi(s_t; \theta) \tag{26}$$

(5) The Deep Deterministic Policy Gradient (DDPG) algorithm is an algorithm with a lot of improvements to DQN, in which the A3C algorithm is added. It is a fusion algorithm of neural network and reinforcement learning. The specific improvement details are shown in **Figure 6**.

(6) THE double DQN (DDQN) algorithm estimates the maximum action in the target network through the network and uses this estimated action to select $Q(s)$ in the target network (Zhang et al., 2018; Han et al., 2019). Then, the goals of TD should be:

$$Y_t^{DoubleDQN} = R_{t+1} + \gamma Q(s_{t+1}, \arg\max Q(s_{t+1}, \theta_t{'})) \tag{27}$$

Construction and Monitoring of Simulation Environment

The simulation environment mainly uses the multimedia framework pyglet under Python to design the interactive applications as the simulation platform. A 200*200-pixel static environment is built in the experiment. In the environment, the mobile robot is no longer a particle but is represented by a blue circle of 10*10 size. The green circle S represents the starting coordinate. The pixel coordinate of the starting position is (5, 5). The purple circle represents the target position and the five black areas of different sizes in the figure are the positions of obstacles. The white area indicates that there are no obstacles in the map, in which the robot can move freely. Since the robot has size in the real world when the boundary of the mobile robot is in contact with the boundary of the obstacle area, it is considered to have collided, and after the collision, it is considered to have failed and is returned to the starting position. The state of the experiment is represented by the rasterized state.

The detection mainly uses the summary.value.add () function in TensorFlow to add variables to the monitoring log. The changes in training process data can be viewed through TensorFlow. After learning, the neural network parameters are saved by using the tf.train.Saver () function and the neural network is reloaded and run again to indicate the effect after the learning is completed. The experimental results show that the mobile robot can avoid dynamic obstacles in time and find an optimal path to reach the target position after avoiding the dynamic obstacles. In the experiment, the copy network value function, the average number of steps used to reach the target position, and the average cumulative reward of the copy network are saved. At the end of the learning and training process, the changing process of the three data can be viewed through TensorBoard.

RESULTS AND DISCUSSIONS

Experimental Results of Different Path Planning Algorithms of Mobile Robot

Figure 7 shows the experimental results of the path planning of mobile robot under different algorithms. As shown in **Figure 7**, under the same starting and ending conditions, all algorithms can effectively avoid obstacles. Comparing **Figures 7A,B**, it was found that in the traditional Q-Learning and A3C algorithms,

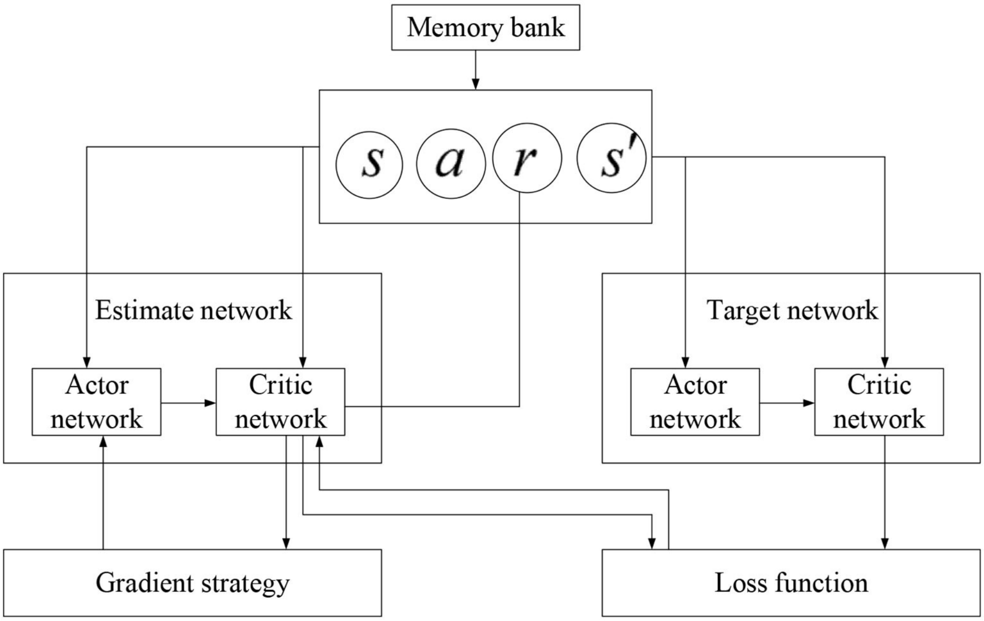


FIGURE 6 | Structure of DDPG network algorithm.

the reinforcement learning algorithm effectively reduces the number of path steps. Comparing **Figures 7A,C**, it was found that the introduction of a neural network algorithm based on the traditional Q-Learning algorithm can greatly reduce the number of paths and achieve the same effect as the reinforcement learning algorithm. Comparing **Figures 7C,D**, it was found that the introduction of the force field based on the neural network has greatly accelerated the running speed of the algorithm, causing a significant reduction in the number of steps. Although the algorithm can effectively avoid obstacles, it has taken many useless paths. Therefore, the DDQN algorithm of Q value accumulation was added. As shown in **Figure 7E**, the algorithm can effectively utilize the neural network to learn and achieve the minimum number of steps. Compared to the DQN algorithm, the running speed of DDQN was improved and compared to the PDQN algorithm, the DDQN can find the optimal path. As shown in **Figure 7F**, a reinforcement learning algorithm was added based on the neural network. It was found that compared to the DDQN algorithm, it runs faster and has an optimal path. According to the above results, the fusion algorithm using a neural network and reinforcement learning has better performance in the path experiment.

Performance Evaluation of Different Path Planning Algorithms of Mobile Robot

Figure 8A illustrates the path planning time of different algorithms under different path lengths. The results show that as the path length increases, the path planning time is also increasing, where the time required is proportional to the path length. As far as different algorithms are concerned, the traditional Q-Learning algorithm takes the longest time, with an average of 78.35 s. The PDQN takes the shortest time because the algorithm introduces a force field, causing the algorithm to be improved continuously. The DDPG algorithm based on neural networks and HRL marks the second position, which takes an average of 40.7 s and is 48.05% higher than the traditional algorithm, 31.01% higher than the DQN algorithm of the neural network, and 40.1% higher than the reinforcement algorithm.

Figure 8B illustrates the number of path steps of different algorithms at different iteration times. As the number of iterations increases, it does not affect the Q-Learning and A3C algorithms because these two algorithms do not have deep learning capabilities. With the increase in the number of iterations, in terms of other algorithms, the number of path steps continues to decrease under the same path. Of the different algorithms, the reinforcement learning algorithm is significantly better than the traditional Q-Learning algorithm, with a 20.56% improvement. Of the different neural network algorithms, the DDPG algorithm has the best performance, which has an average path step of 63 steps; compared to the DQN algorithm, it has an increase of 20.25%. When compared to the DDQN algorithm, the number of path steps is increased by 8.69%. According to the above results, the PDQN algorithm is more efficient under the same path conditions, as the learning continues, the fusion algorithm performs better in terms of path steps.

Figure 9A illustrates the convergence time of different algorithms under different path steps. The results show that as the path steps continue to increase, the convergence time of each algorithm is continuously increasing. Compared to the Q-Learning and A3C algorithms, after adding reinforcement learning, the convergence time of robot path planning is increased by 13.54%; compared to the Q-Learning and DQN algorithms, after adding the neural network algorithm, the convergence time of robot path planning is increased by 33.85%,

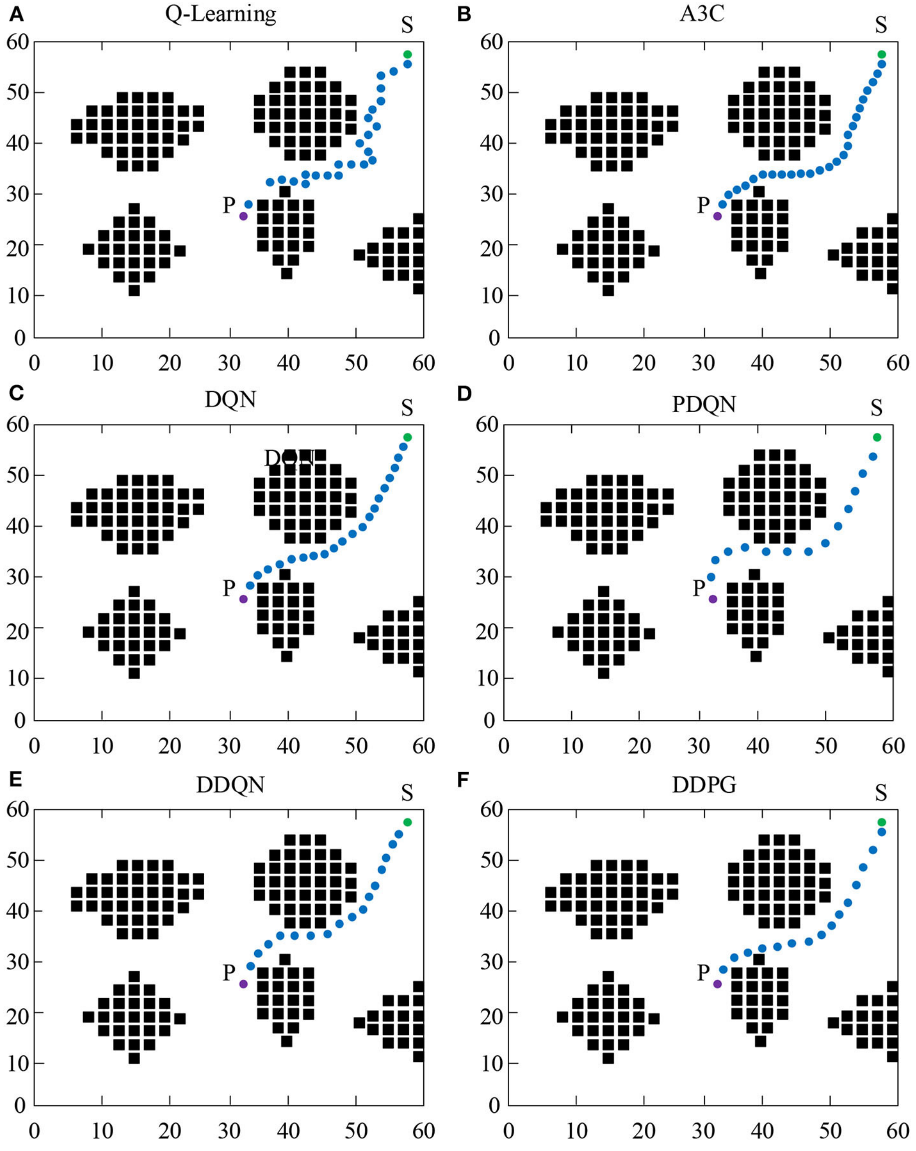


FIGURE 7 | Experimental results of different path planning algorithms of mobile robot.

which is the most obvious improvement. Comparing different neural networks, it was found that the convergence time of the DDQN algorithm with increased *Q*-value is greatly improved, and the convergence time of path planning is improved by 94.44% compared with the previous Q-Learning algorithm. For the DDPG algorithm based on neural network and HRL, the convergence time of the algorithm under the unsynchronized number is 1.34 s on average, which is 55.52% faster than the optimal DDQN algorithm.

Figure 9B illustrates the cumulative rewards of different algorithms under different path steps. Since the designed reward rules are more stringent, the reward results are all negative, but this does not affect the obtained results. As shown in **Figure 9B**, as the number of path steps continues to increase, the cumulative rewards continue to increase. For different algorithms, comparing the Q-Learning and A3C algorithms, the cumulative reward is significantly improved by 29.64%. Compared to the Q-Learning algorithm, the neural network DQN has increased significantly. Under the same neural network, it was found that the PDQN algorithm that introduces the force field has less cumulative rewards. The reason may be that the purpose of the algorithm is to enhance the running speed of the algorithm. The mechanism for rewards is not very complete; thus, the rewards are less. Among the neural network algorithms, the DDQN algorithm has the best cumulative reward. However, compared to the fusion algorithm DDPG, the performance of the DDQN algorithm is not very good. The cumulative reward of DDPG is increased by 41.5% compared to DDQN. According to

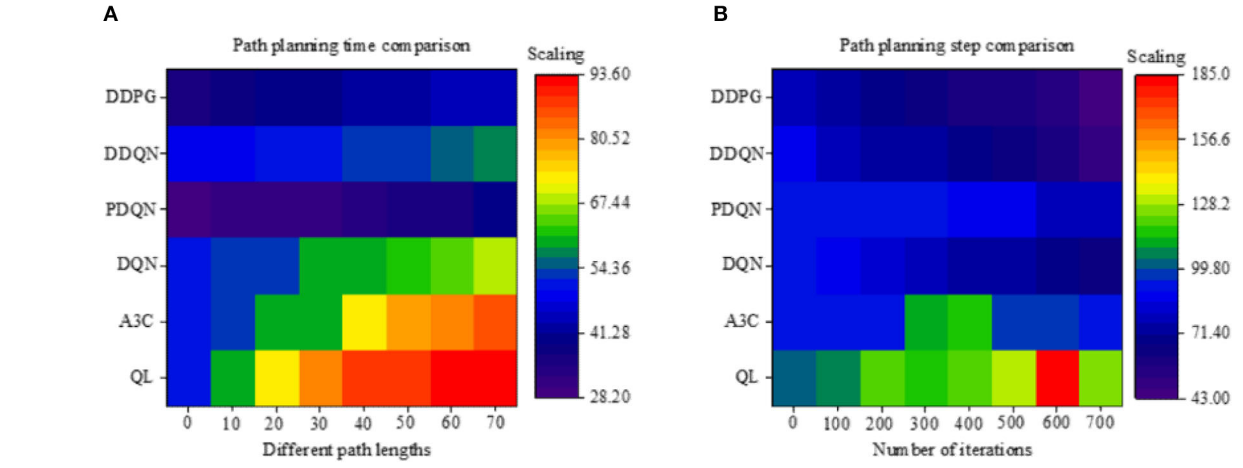


FIGURE 8 | Performance evaluation of time and steps of different mobile robot path planning algorithms (QL algorithm represents the Q-Learning algorithm).

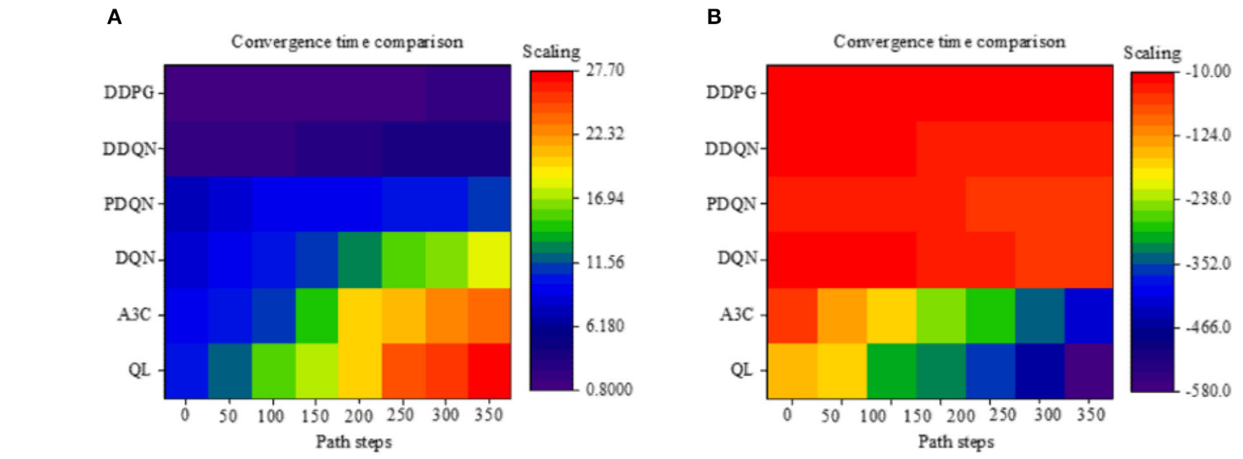


FIGURE 9 | Evaluation of convergence time and cumulative reward performance of different path planning algorithms of the mobile robot (QL algorithm represents the Q-Learning algorithm).

the above results, it is concluded that under different path steps, the convergence time of the algorithm is the fusion algorithm; at the same time, the algorithm can also obtain the most rewards.

Analysis of Performance Changes in Neural Network and HRL Algorithms Under Different Environmental Conditions

To explore the impact of different environmental conditions on the performance of the algorithm, the performance of the DDPG algorithm was tested under different action sets, grid numbers, state sets, and force values. Under the premise of the same starting point and ending point, the average value of the algorithm was obtained after running 30 times. The results are shown in **Table 1**. As shown in the table, the comparison between M1 and M2 indicates that when the action set is doubled, the convergence time of the algorithm will increase by 41%, and the smoothness of the planned path is also increased by 53%. Comparing M2 and M3, it is found that when the number of grids is increased three times, the convergence of the algorithm will be reduced by 69%, and the smoothness will be increased by 45%. Comparing M3 and M4, it was found that increasing the number of state sets will slow down the convergence speed of the algorithm, but by adjusting the direction of the action set, the right angles and corners in the path can be avoided, and the smoothness with which it navigates the planned path is increased by 18%. Comparing M4 and M5, it is found that the introduction of the force field will reduce the convergence time of the algorithm by 49%, which can increase the action step size, thereby adjusting the number of state sets and the direction of the action set. Therefore, when the action set is 4, the number of grids is 3, and the state set is 40*40*8, with the introduction of the force value, the algorithm can reduce the convergence time by 91% compared with the traditional Q-learning algorithm, and the smoothness of the path increased by 79%.

TABLE 1 | Effect of different environmental conditions on algorithm performance.

Numbering	Number of states	Number of actions	Action step	Potential field/s	Convergence time	Convergence round	Path length	Total corner/rad
M1	40*40	4	1	N0	1.9254	682.6	38.1	21.677
M2	40*40	8	1	N0	2.7139	629.7	32.7	10.210
M3	40*40	8	3	N0	0.8515	274.6	34.3	5.655
M4	40*40*8	4	3	N0	1.4259	340.1	32.8	4.616
M5	40*40	4	1	Yes	0.9848	559.8	38.0	21.834
M6	40*40*8	4	3	Yes	0.1735	155.3	32.1	4.555

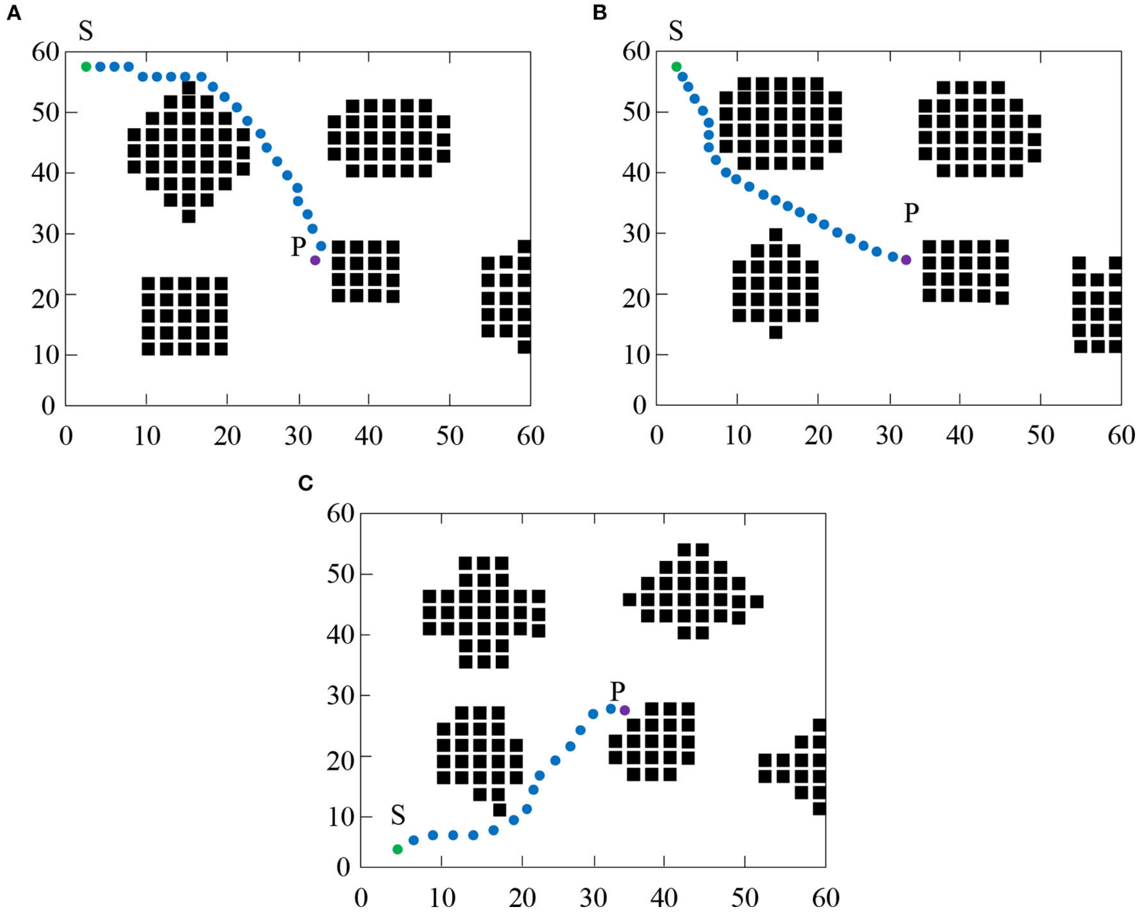


FIGURE 10 | Path changes of algorithms in different scenarios.

Analysis of Changes in Paths Based on Neural Networks and HRL Under Different Scenario Conditions

Figure 10 and **Table 2** indicate the path changes and quantitative data of the algorithm under different scene conditions. As shown in **Figure 10**, by comparing **Figures 10A,B**, it was found that at the same starting point and ending point, under the condition of different obstacles, the algorithm system can effectively avoid obstacles and design the optimal paths. In addition, the convergence time is maintained at about 0.15 s, the number of convergence rounds is maintained at 145, and the total rotation angle is 4.8 rad. By comparing **Figures 10A,C**, it was found that under different environments and different starting points and ending points, the system can still avoid collisions with obstacles, maintain a high convergence time, and design an optimal path. Simulation results show that the proposed path planning algorithm for mobile robots based on neural networks and HRL has a good generalization effect in different scenarios.

DISCUSSION

The neural network DQN can perceive the environment and perform feature extraction to realize the fitting from the environment to the state action function. This has been

TABLE 2 | Statistical results of algorithm path changes under different scenario conditions.

Scenes	Position	Convergence time/s	Convergence round	Path length	Total corner/rad
P1	(1, 39)	0.1615	144.5	32.1	4.869
P2	(1, 39)	0.1468	147.0	31.6	4.712
P3	(39, 39)	0.1724	147.4	31.8	4.641

mentioned in the literature. Qiao et al. (2018) proposed an adaptive DQN strategy and applied it to text recognition. These results showed that the DQN algorithm is significantly better than other algorithms, which also indicated the advantages of the DQN algorithm in image recognition (Qiao et al., 2018). Compared with the deep learning algorithm DQN, the DDQN algorithm is better than DQN in terms of value accuracy and strategy, which is also consistent with previous reports (Qu et al., 2020). The hierarchical reinforcement learning technology is utilized to achieve the mapping from state to action and meet the mobile needs of mobile robots. The data have also proven that the robot path planning method based on deep reinforcement learning is an effective end-to-end mobile robot path planning method, which has also been confirmed in a study by Wang B. et al. (2020). The above results illustrate the feasibility of the proposed method in the path planning of mobile robots.

The DDPG algorithm was developed based on the DQN algorithm. The biggest improvement is that the action strategy of the DQN algorithm can only select actions in discrete action space, while the DDPG algorithm can select actions in continuous action space. The results show that the algorithm is significantly better than other algorithms in terms of operating efficiency. This is consistent with the results of Shen X. et al. (2019), in which it was found that when compared with the exponential moving average the effective variance of DDPG and average DDQN were reduced, which explained the efficient runtime of the algorithm further (Shen X. et al., 2019). The results also found that after reinforcement learning is added, the convergence time of robot path planning is increased by 13.54%. Low et al. used the flower pollination algorithm to properly initialize the *Q*-value, which could speed up the convergence of mobile robots (Low et al., 2019). The principle is similar to reinforcement learning, therefore, the research results here are also supported. The comparison between the Q-Learning and DQN algorithms found that the convergence time of robot path planning is increased by 33.85% after adding the neural network algorithm. Some scholars have improved the convergence performance of the model significantly by using two natural heuristic algorithms in unknown or partially known environments (Saraswathi et al., 2018). This natural heuristic algorithm is similar to the neural network structure, further proving the effectiveness of the proposed algorithm.

In summary, the proposed DDQN algorithm has been proven to be applicable to image feature extraction, and the neural network algorithm has also been proven to effectively improve the performance and convergence of the algorithm. The data obtained are consistent with previous research. However, in terms of algorithm performance, the performance of mobile robot path planning based on neural networks and hierarchical reinforcement learning has been significantly improved. This algorithm can significantly reduce path planning time and improve smoothness, enabling mobile robots to move more conveniently and flexibility.

CONCLUSIONS

Through neural networks, the fitting from the environment to the state action function was realized by perceiving the environment and performing feature extraction. Through the enhancement function, the mapping of the current state to the action of the hierarchical reinforcement learning was satisfied, thereby enabling the robot to become more mobile. The two were organically combined to improve the performance of mobile robots during path planning. The mobile robot path planning algorithm based on neural networks and hierarchical reinforcement learning has better performance than other algorithms in all aspects. In addition, the proposed algorithm reduces the planning time, decreases the number of path steps, shortens the convergence time, and increases the smooth and efficient recognition and movement functions of the mobile robots. Although the performance of each algorithm has been analyzed as comprehensively as possible, the following aspects need to be improved in the future. First, it is impossible for the neural network learning method of the mobile robot's motion path planning to perform multiple "trial and error" processes in actual operations, which makes it difficult to apply the proposed algorithm. It is therefore necessary to implement the application on the physical platform before applying the algorithm to the actual robots. Second, the path planning only involves static scenarios. Whether the algorithm can show the same performance when encountering dynamic environmental changes is yet to be explored. The path planning capabilities of mobile robots were improved, laying a theoretical foundation for practical applications.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by Chongqing University Ethics Committee. The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

REFERENCES

Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: a brief survey. *IEEE Signal Process. Magazine* 34, 26–38. doi: 10.1109/MSP.2017.2743240

Bakdi, A., Hentout, A., Boutami, H., Maoudj, A., Hachour, O., and Bouzouia, B. (2017). Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. *Robot. Autonomous Syst*. 89, 95–109. doi: 10.1016/j.robot.2016.12.008

Botteghi, N., Sirmacek, B., Mustafa, K. A., Poel, M., and Stramigioli, S. (2020). On reward shaping for mobile robot navigation: a reinforcement learning and SLAM based approach. *arXiv:200204109*. 2020, 1025–1037.

Chen, M. (2018). "The research of human individual's conformity behavior in emergency situations,"(Library Hi Tech). doi: 10.1108/LHT-08-2018-0113

Ghosh, S., Panigrahi, P. K., and Parhi, D. R. (2017). Analysis of FPA and BA meta-heuristic controllers for optimal path planning of mobile robot in cluttered environment. *IET Sci. Measure. Technol*. 11, 817–828. doi: 10.1049/iet-smt.2016.0273

Gupta, U., Mandal, S. K., Mao, M., Chakrabarti, C., and Ogras, U. Y. (2019). A deep Q-learning approach for dynamic management of heterogeneous processors. *IEEE Comp. Architect. Lett*. 18, 14–17. doi: 10.1109/LCA.2019.2892151

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., et al. (2018). Soft actor-critic algorithms and applications. *arXiv:181205905*. 2018, 26–32.

Han, J., and Seo, Y. (2017). Mobile robot path planning with surrounding point set and path improvement. *Appl. Soft Comp*. 57, 35–47. doi: 10.1016/j.asoc.2017.03.035

Han, X., He, H., Wu, J., Peng, J., and Li, Y. (2019). Energy management based on reinforcement learning with double deep Q-learning for a hybrid electric tracked vehicle. *Appl. Energy* 254, 113708–113725. doi: 10.1016/j.apenergy.2019.113708

Lei, X., Zhang, Z., and Dong, P. (2018). Dynamic path planning of unknown environment based on deep reinforcement learning. *J. Robot*. 2018, 25–32. doi: 10.1155/2018/5781591

Li, G., and Chou, W. (2018). Path planning for mobile robot using self-adaptive learning particle swarm optimization. *Sci. China Inform. Sci*. 61, 052204–052213. doi: 10.1007/s11432-016-9115-2

Liu, L., and Hodgins, J. (2017). Learning to schedule control fragments for physics-based characters using deep q-learning. *ACM Trans. Graph*. 36, 1–14. doi: 10.1145/3083723

Liu, Q., Cheng, Z., and Chen, M. (2019). Effects of environmental education on environmental ethics and literacy based on virtual reality technology. *Electron. Lib*. 37, 860–877. doi: 10.1108/EL-12-2018-0250

Liu, Z., and Wang, C. (2019). Design of traffic emergency response system based on internet of things and data mining in emergencies. *IEEE Access* 7, 113950–113962. doi: 10.1109/ACCESS.2019.2934979

Low, E. S., Ong, P., and Cheah, K. C. (2019). Solving the optimal path planning of a mobile robot using improved Q-learning. *Robot. Autonomous Syst*. 115, 143–161. doi: 10.1016/j.robot.2019.02.013

Ohnishi, S., Uchibe, E., Nakanishi, K., and Ishii, S. (2019). Constrained Deep Q-learning gradually approaching ordinary Q-learning. *Front. Neurorobot*. 13, 103–112. doi: 10.3389/fnbot.2019.00103

Orozco-Rosas, U., Montiel, O., and Sepúlveda, R. (2019). Mobile robot path planning using membrane evolutionary artificial potential field. *Appl. Soft Comp*. 77, 236–251. doi: 10.1016/j.asoc.2019.01.036

Qiao, J., Wang, G., Li, W., and Chen, M. (2018). An adaptive deep Q-learning strategy for handwritten digit recognition. *Neural Netw*. 107, 61–71. doi: 10.1016/j.neunet.2018.02.010

Qu, Z., Hou, C., Hou, C., and Wang, W. (2020). Radar signal intra-pulse modulation recognition based on convolutional neural network and deep Q-learning network. *IEEE Access* 8, 49125–49136. doi: 10.1109/ACCESS.2020.2980363

Saraswathi, M., Murali, G. B., and Deepak, B. (2018). Optimal path planning of mobile robot using hybrid cuckoo search-bat algorithm. *Proc. Comp. Sci*. 133, 510–517. doi: 10.1016/j.procs.2018.07.064

Shen, C.-,w., Ho, J.-,t., Ly, P. T. M., and Kuo, T.-,c. (2019a). Behavioural intentions of using virtual reality in learning: perspectives of acceptance of information technology and learning style. *Virtual Reality* 23, 313–324. doi: 10.1007/s10055–018-0348–1

Shen, C.-w., Min, C., and Wang, C.-,c. (2019b). Analyzing the trend of O2O commerce by bilingual text mining on social media. *Comp. Human Behav*. 101, 474–483. doi: 10.1016/j.chb.2018.09.031

Shen, X., Yin, C., Chai, Y., and Hou, X. (2019). "Exponential moving averaged Q-network for DDPG," in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. (Beijing: Springer), 562–572. doi: 10.1007/978-3-030-31654-9_48

Sung, I., Choi, B., and Nielsen, P. (2020). On the training of a neural network for online path planning with offline path planning algorithms. *Int. J. Inform. Manage*. 102142–102150. doi: 10.1016/j.ijinfomgt.2020.102142

Wang, B., Liu, Z., Li, Q., and Prorok, A. (2020). Mobile robot path planning in dynamic environments through globally guided reinforcement learning. *arXiv:200505420*. 2020, 22–29.

Wang, P., Li, X., Song, C., and Zhai, S. (2020). Research on dynamic path planning of wheeled robot based on deep reinforcement learning on the slope ground. *J. Robot*. 9, 36–39. doi: 10.1155/2020/7167243

Wei, Q., Lewis, F. L., Sun, Q., Yan, P., and Song, R. (2016). Discrete-time deterministic $ Q $-learning: a novel convergence analysis. *IEEE Trans. Cybernet*. 47, 1224–1237. doi: 10.1109/TCYB.2016.2542923

Wen, S., Zhao, Y., Yuan, X., Wang, Z., Zhang, D., and Manfredi, L. (2020). Path planning for active SLAM based on deep reinforcement learning under unknown environments. *Intelligent Service Robot*. 1–10. doi: 10.1007/s11370-019-00310-w

Wulfmeier, M., Rao, D., Wang, D. Z., Ondruska, P., and Posner, I. (2017). Large-scale cost function learning for path planning using deep inverse reinforcement learning. *Int. J. Robot. Res*. 36, 1073–1087. doi: 10.1177/0278364917722396

Yan, Z., and Xu, Y. (2018). Data-driven load frequency control for stochastic power systems: a deep reinforcement learning method with continuous action search. *IEEE Trans. Power Syst*. 34, 1653–1656. doi: 10.1109/TPWRS.2018.2881359

Zhang, Q., Lin, M., Yang, L. T., Chen, Z., Khan, S. U., and Li, P. (2018). A double deep Q-learning model for energy-efficient edge scheduling. *IEEE Trans. Services Comp*. 12, 739–749. doi: 10.1109/TSC.2018.2867482

Zhao, Y., Zheng, Z., and Liu, Y. (2018). Survey on computational-intelligence-based UAV path planning. *Knowledge Based Syst*. 158, 54–64. doi: 10.1016/j.knosys.2018.05.033

Zheng, Y., and Liu, S. (2020). "Bibliometric analysis for talent identification by the subject–author–citation three-dimensional evaluation model in the discipline of physical education," (Library Hi Tech). doi: 10.1108/LHT-12-2019-0248

Zhu, J., Song, Y., Jiang, D., and Song, H. (2017). A new deep-Q-learning-based transmission scheduling mechanism for the cognitive Internet of Things. *IEEE Internet Things J*. 5, 2375–2385. doi: 10.1109/JIOT.2017.2759728

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

ORIGINAL RESEARCH
published: 15 October 2020
doi: 10.3389/fnbot.2020.00054

Bi-criteria Acceleration Level Obstacle Avoidance of Redundant Manipulator

Weifeng Zhao[1], *Xiaoxiao Li*[2], *Xin Chen*[1], *Xin Su*[1] *and Guanrong Tang*[2]*

[1] Foshan Longshen Robotics LTD., Foshan, China, [2] Guangdong Key Laboratory of Modern Control Technology, Guangdong Institute of Intelligent Manufacturing, Guangzhou, China

In this paper, an improved obstacle-avoidance-scheme-based kinematic control problem in acceleration level for a redundant robot manipulator is investigated. Specifically, the manipulator and obstacle are abstracted as mathematical geometries, based on the vector relationship between geometric elements, and the Cartesian coordinate of the nearest point to an obstacle on a manipulator can be found. The distance between the manipulator and an obstacle is described as the point-to-point distance, and the collision avoidance strategy is formulated as an inequality. To avoid the joint drift phenomenon of the manipulator, bi-criteria performance indices integrating joint-acceleration-norm minimization and repetitive motion planning is adopted by assigning a weighing factor. From the perspective of optimization, therefore, an acceleration level quadratic programming (QP) problem is eventually formulated. Considering the physical structure of robot manipulators, inherent joint angle, speed, and acceleration limits are also incorporated. To solve the resultant QP minimization problem, a recurrent neural network based neural dynamic solver is proposed. Then, simulation experiments performing on a four-link planar manipulator validate the feasibility and effectiveness of the proposed scheme.

Keywords: recurrent neural network, path planning, redundant manipulator, acceleration level obstacle avoidance, bi-criteria

OPEN ACCESS

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

Reviewed by:
Ameer Hamza Khan,
Hong Kong Polytechnic University,
Hong Kong
Baofang Wang,
Qingdao University, China
Shuai Li,
Swansea University, United Kingdom

****Correspondence:***
Guanrong Tang
gr.tang@giim.ac.cn

Received: *26 May 2020*
Accepted: *08 July 2020*
Published: *15 October 2020*

Citation:
Zhao W, Li X, Chen X, Su X and Tang G (2020) Bi-criteria Acceleration Level Obstacle Avoidance of Redundant Manipulator. Front. Neurorobot. 14:54. doi: 10.3389/fnbot.2020.00054

1. INTRODUCTION

With the advances of society, ranging from industry to military, home furnishing, service, medical treatment, *etc.*, robot technology has already become gradually mature. Simultaneously, the high demand on the execution abilities of a robot manipulator working in complicated environment also poses a challenge to robotic control. Due to its degrees of freedom (DOF) exceeding ones required by the robot to complete the given tasks, a redundant manipulator shows better flexibility, multifunction, and wide universality than the traditional non-redundant robot.

As a fundamental problem in robotic control, the kinematic motion planning problem of the redundant manipulator has already been widely investigated in recent years. Series of related products have been reported, e.g., in Li et al. (2016), from the perspective of game theory, and a distributed recurrent neural network (RNN)-based dynamic controller was proposed for the coordination control of multi-robot system. In Li et al. (2020), based on the RNN, Li et al. investigated the kinematic control problem of the multi-robot system under neighbor-to-neighbor communication. To access the desired global command, an observer was developed for estimating the velocity information of the desired motion trajectory. A distributed RNN scheme was proposed in Jin et al. (2018) for the consensus and cooperative control of a multiple manipulator under limited communication, achieving the global cooperation of PUMA 560 manipulators. The kinematic control of a redundant manipulator disturbed by the periodic input was investigated in Zhang et al. (2019a) based on the RNN. Moreover, Zhang et al. proposed an RNN control scheme incorporating the joint acceleration constraint for the redundant manipulator in Zhang et al. (2019b), which is solved in acceleration level. In Xu et al. (2019a) and Xu et al. (2020), the RNN was used to the kinematic control of redundant manipulator with model uncertainties and coupling of motion and contact force, respectively. In Chen et al. (2019), the RNN was applied to the motion control of a mobile robot. In Li et al. (2018a) and Li et al. (2018b), a modified RNN-based controller was proposed for motion control of the manipulator disturbed by noises. In Chen et al. (2020a), a time-varying noise disturbance rejection constraint was established. In addition, Chen et al. proposed a joint velocity, acceleration, and joint jerk three-level simultaneous minimization scheme in Chen et al. (2020b). The abovementioned involve single and multiple robot systems. Following them, the RNN can in principle handle the kinematic problem of a redundant manipulator. In addition, in most of the abovementioned literature, the consensus is that the quadratic programming (QP) method, i.e., where the manipulator kinematic control problem is described as a QP minimization problem, is adopted, owing to which can incorporate physical constraints such as joint angle and joint velocity limits.

When performing a desired task, the success of the motion planning task may not be guaranteed if the manipulator encounters a sudden obstacle, and even the robot manipulator will be damaged due to the collision. The obstacle avoidance problem of a redundant manipulator is thus worthy of investigation. Obstacle avoidance, called collision avoidance, always plays an important role and is continuously investigated among redundant manipulators. For collision avoidance, in general, two aspects need to be considered: one is robot-to-environment, and the other is robot-to-robot. Especially for a multi-robots system, the obstacle avoidance scheme should include not only the collision avoidance between robot arms but also the collision avoidance between robots and environmental obstacles. Many obstacle avoidance methods have been proposed, such as pseudo-inverse-based ones (Zlajpah and Nemec, 2002; Lee and Buss, 2007; Guo et al., 2018), random-sampling-based methods such as rapidly exploring random tree (RRT) (Ju et al., 2014; Zhang et al., 2018), artificial potential field (Volpe and Khosla, 1990; Kim and Khosla, 1992), and QP-based optimization methods (Zhang and Wang, 2004; Guo and Zhang, 2014, 2019; Zhou et al., 2019; Xu et al., 2019b). In general, pseudo-inverse methods have no ability of handling physical structure constraints of a manipulator. The RRT methods are very effective for high-dimensional and complicated environments, which makes the generated path approaches a collision-free region by randomly sampling unknown space. For ones aided with an artificial potential field, different environments need specialized potential functions. Among such a method, the robot is assumed to move within a virtual force field where the target and the obstacle are denoted as an attractive pole and a repulsive surface, respectively. Although effective, these two methods are accompanied by higher computational costs; for the latter, the computational complexity is exponentially increasing to the DOFs of the robot.

Generally speaking, for QP-based methods, the obstacle avoidance strategy is usually formulated as an attachment constraint of the resultant QP minimization problem. For their works in Zhang and Wang (2004), Guo and Zhang (2014, 2019), etc., the collision avoidance constraints were set inner and outer thresholds for safety. In Xu et al. (2019b) and Zhou et al. (2019), a relatively simple inequality that can avoid collision with the obstacle was proposed. In their works, both the obstacle and manipulator are abstracted as point sets. A safe distance is given by ensuring the distance between the manipulator and obstacle is always greater than the safe distance, and the safety is ensured. However, as points representing the manipulator are chosen in a uniform way, this method carries a possible risk that the chosen points do not collide with the obstacle; in practice, the collision has already happened due to the distance from the chosen point to the obstacle may be greater than the shortest distance between the manipulator and the obstacle.

In this study, therefore, we provide an improved obstacle avoidance scheme that can determine the nearest point on every link of the manipulator to the obstacle. By always keeping the minimal distance between them outside the non-safety region, the safety is ensured. In addition, if the acceleration vector is quite different at the front and back time, it will produce excessive velocity, which will enable the manipulator to shake, critically impact, or even cause damage to the manipulator or potential safety accident. Moreover, if not taking the joint acceleration into account, the generated joint velocity command may be discontinuous (Guo and Zhang, 2014, 2019). Consequently, in this study, the kinematic control problem of a redundant manipulator is investigated in terms of acceleration level. Specifically, the robot manipulator and obstacle are first abstracted as mathematical geometries based on the vector relationship between geometric elements in the search for the Cartesian coordinates of the points whose distance from every link of the manipulator to the obstacle is shortest. The distance between a robot manipulator and an obstacle is described as point-to-point distance, and an inequality constraint is thus constructed, which is built in acceleration level, to avoid the obstacle. To avoid the joint drift problem and improve the stability and reliability of robots in periodic tasks such as palletizing, welding, etc., the bi-criteria performance indices integrating joint-acceleration-norm minimization (MAN) and repetitive motion planning (RMP) is considered by assigning a weighing factor. The kinematic control problem of the manipulator is transformed into an equality constraint mapping from Cartesian space to joint acceleration space. To sum up, an acceleration-level quadratic programming (QP) problem is obtained, combining the joint angle, joint velocity, and joint acceleration limits rebuilt in the acceleration level. Then, utilizing the real-time property of the RNN, we designed an RNN based neural dynamic controller to solve the QP problem. Finally, simulative experiments are performed on a four-link planar manipulator, validating the feasibility of the proposed control scheme and obstacle avoidance strategy by simulative results.

The ensuing part of this paper is arranged around the following aspects: preliminaries such as kinematic description of redundant manipulator, the nearest point selection as well as the formulation of the inequality obstacle avoidance strategy, and problem statement are introduced in section 2. Section 3 shows the QP problem reformulation and the design of RNN controller. Simulation results are given in section 4, where both the static and dynamic obstacle are considered. Section 5 summarizes the whole paper with a final remark. The main contributions of this paper are summarized as follows:

1) The acceleration-level kinematic control problem of redundant manipulator with the obstacle collision avoidance is investigated. Bi-critic performance indices consisting of joint-acceleration-norm minimization and repetitive motion planning are considered in order to avoid the joint drift and improve the stability and reliability of robots in periodic tasks.
2) An improved obstacle avoidance strategy that can return the nearest point of every link of a manipulator to the obstacle is proposed. By keeping minimal distance between the robot and the obstacle outside the non-safety region all the times, the safety is ensured.
3) An RNN-based dynamic controller combining the motion planner, obstacle avoidance and joint angles, joint speed, as well as joint acceleration constraints is proposed. Under its control, the robot achieves the desired trajectory tracking task with a desired tracking error, and it successfully avoids collision with static and dynamic obstacles.

2. PRELIMINARIES AND PROBLEM STATEMENT

2.1. Kinematics Description of Redundant Manipulator

For path planning task of a robot manipulator, the position of its end-effector is only determined by its joint space vector $\theta(t)$, and the relationship between them is usually described as

$$r(t) = f(\theta(t)), \tag{1}$$

where $r(t) \in \mathbb{R}^m$ are Cartesian coordinate of the end-effector at time t, and $\theta(t) \in \mathbb{R}^n$ are the coordinate of the end-effector in joint space. $f(\cdot)$: $\mathbb{R}^n \rightarrow \mathbb{R}^m$, is a non-linear mapping determined by the physical structure and parameters of the used manipulator. For a redundant manipulator, $m < n$; this means that when $r(t)$ is given and known, infinite corresponding $\theta(t)$ may exist. Moreover, due to the non-linear property of redundant manipulator, directly solving Equation (1) is extremely difficult. On the contrast, solving Equation (1) in velocity level or acceleration level gives a simpler way. For the velocity level, Equation (1) can be transformed into

$$\dot{r}(t) = J(\theta(t))\dot{\theta}(t), \tag{2}$$

where $J(\theta(t)) \in \mathbb{R}^{m\times n}$ is Jacobian matrix. $\dot{r}(t)$ and $\dot{\theta}(t)$ correspond to the derivatives of $r(t)$ and $\theta(t)$, respectively, denoting Cartesian and joint velocity, respectively.

Computing the derivatives of Equation (2), the acceleration level kinematics is described as

$$\ddot{r}(t) = J(\theta(t))\ddot{\theta}(t) + \dot{J}(\theta(t))\dot{\theta}(t), \tag{3}$$

where $\dot{J}(\theta(t))$ is a time derivative of $J(\theta(t))$. $\ddot{r}(t)$ and $\ddot{\theta}(t)$ are the derivatives of $\dot{r}(t)$ and $\dot{\theta}(t)$, respectively, denoting acceleration of the manipulator in Cartesian and joint space, respectively. For simplicity, in the following sections, $\dot{J}(\theta(t))$, $J(\theta(t))$, $r(t)$, $\dot{r}(t)$, $\ddot{r}(t)$, $\theta(t)$, $\dot{\theta}(t)$, and $\ddot{\theta}(t)$ are abbreviated to $\dot{J}$, J, r, $\dot{r}$, $\ddot{r}$, θ, $\dot{\theta}$, and $\ddot{\theta}$, respectively.

2.2. Obstacle Avoidance

2.2.1. Basic Description

Based on the bound box theory, the robot manipulator and obstacle can be simplified as the mathematical geometry. For example, the plane manipulator can be abstracted as a combination of cylinders, and the obstacles are abstracted as spheres, cylinders, cuboids, or a combination of them (Yue et al., 2015). By describing the manipulator and obstacle as point sets, the distance between them is transformed into the point-to-point distance. Assume that A and B are Cartesian coordinates of one of the points on a manipulator and an obstacle, respectively, given a safety distance d; in principle, if $||AB|| \geq d$ is always satisfied during robot movement, the safety (collision-free) between the robot and obstacle will be ensured, where $||AB|| = \sqrt{(A-B)^T(A-B)}$ denotes the Euclidean norm.

As the DH parameters of a manipulator are given, Cartesian coordinates of the critical points located in manipulator joint centers are easier to compute. By setting certain criteria, critical points on links of a manipulator can also be obtained. However, how to select the representative points on both the manipulator and obstacles is challenging. Selecting abundant points will increase the computational costs and is not necessary. A method is to uniformly choose critical points on a manipulator (as shown in **Figure 1A**), which is introduced in Xu et al. (2019b) and Zhou et al. (2019). The basic idea is that the critical point is chosen in the center of a link of a manipulator, based on the joint angle information and link length, and Cartesian coordinates of the critical point can be computed. Although we reduced the computational complexity, we found that this way caused a possibility that the chosen points did not collide with the obstacle; in practice, the collision had already happened due to the distance from the chosen points to the obstacle being greater than the shortest distance between them.

As shown in **Figure 1B**, when the obstacle is located in Position III based on the uniform point selection, B will be adopted to determine whether the collision with the obstacle occurred. If the safety distance is just set as $d + a$ and $d < d + a < d_s$, where a is a positive constant, for B, the collision-free will be determined. However, for point A, the manipulator will collide with the obstacle. Therefore, motivated by it, in this study, we aim to find the nearest points on every link of the manipulator to the obstacle to ensure the collision-free. Utilizing

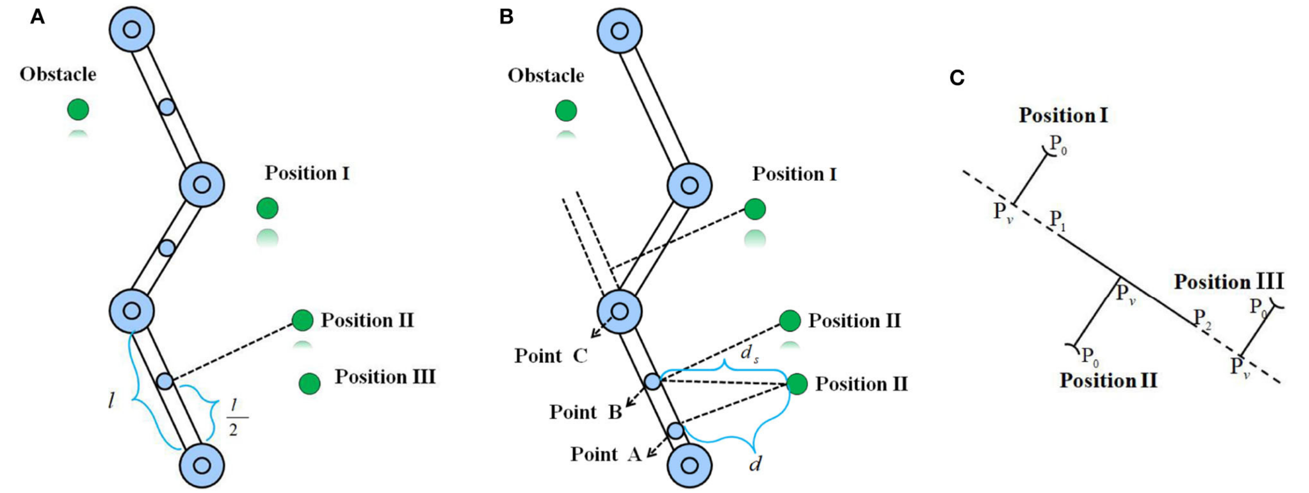


FIGURE 1 | The basic idea of two obstacle avoidance schemes where the green denotes the obstacle, and the blue denotes the critical point selected from a link of a manipulator. **(A)** Uniform point selection. **(B)** The proposed nearest point selection method in this paper. **(C)** Three possible situations corresponding to the nearest point collision avoidance strategy.

the vector relationship between geometric elements, the method is simple and easy to implement. Assuming that P_1 and P_2 are the coordinates of two critical points in the center of two joints connecting a link of the manipulator, P_0 are Cartesian coordinates of the detected critical point from the encountered obstacle (as shown in **Figure 1C** where the manipulator link and the obstacle are simplified as a segment and a point, respectively). Assume that P_v is a projection of P_0 to segment P_1P_2. If $\lambda = \overrightarrow{P_1P_v}/\overrightarrow{P_1P_2} \in [0, 1]$ (that is to say, P_0 is located in Position II), where $\overrightarrow{\cdot}$ denotes the directional vector, then the nearest point is P_v with the minimal distance $d_{\min} = ||P_0P_v||$. Otherwise, $d_{\min} = \{||P_0P_1||, ||P_0P_2||\}_{\min}$. For $\lambda < 0$, the nearest point is P_2 (i.e, P_0 is located in Position III), for $\lambda > 1$, the nearest point P_1 will be returned.

*Remark*1: Note that, in this paper, Cartesian coordinates of the critical points on the obstacle are known by default. In real life, the real-time measurement of the surrounding obstacles is easy to achieve by use of a camera, and the related achievements have been reported in Carloni et al. (2013) and Zhang et al. (2015).

2.2.2. Inequality Formalization on an Acceleration Level

Assume that A is the nearest point on a link of a manipulator to the obstacle and B denotes the mass center of the obstacle. To ensure safety between them, the inequality $||AB|| \geq d$ is required to hold. For this purpose, define $e = ||AB|| - d$, and an inequality in velocity level is constructed as follows:

$$\frac{d||AB||}{dt} \geq -k_1 e, \tag{4}$$

where k_1 is a positive constant that is used to scale the convergence rate of the error. Due to $\dot{A} = J\dot{\theta}$ and

$$\frac{d||AB||}{dt} = \frac{d}{dt}\sqrt{(A-B)^T(A-B)} = \overrightarrow{||BA||}^T(\dot{A} - \dot{B}), \tag{5}$$

where $\overrightarrow{||BA||} = (A-B)^T/||A-B|| \in \mathbb{R}^{1\times m}$ is the unit vector of $\overrightarrow{A-B}$, $\dot{A}$ is the velocity of point A in joint space, and $J \in \mathbb{R}^{m\times n}$ is the Jacobian matrix of A; we can obtain

$$\begin{aligned} &\overrightarrow{||BA||}^T(\dot{A} - \dot{B}) \geq -k_1 e, \\ &\overrightarrow{||BA||}^T(J\dot{\theta} - \dot{B}) \geq -k_1 e, \\ &\overrightarrow{||BA||}^T(J\dot{\theta}) \geq -k_1 e + \overrightarrow{||BA||}^T\dot{B}, \end{aligned} \tag{6}$$

let $-\overrightarrow{||BA||}^T J = J_o \in \mathbb{R}^{1\times n}$, $k_1 e - \overrightarrow{||BA||}^T\dot{B} = C$, Equation (6) can be summarized as

$$J_o\dot{\theta} \leq C. \tag{7}$$

The velocity-level collision avoidance inequality, i.e., Equation (7), is obtained, and it has been proven to have the ability to avoid collision between the static and dynamic obstacles in Zhou et al. (2019) and Xu et al. (2019b). Much like the velocity level, by constructing

$$\frac{d}{dt}(\frac{d||AB||}{dt} + k_1 e) \geq -k_2(\frac{d||AB||}{dt} + k_1 e), \tag{8}$$

then,

$$\begin{aligned} &\frac{d}{dt}(\frac{d||AB||}{dt} + k_1 e) \geq -k_2(\frac{d||AB||}{dt} + k_1 e) \\ &\frac{d}{dt}(\overrightarrow{||BA||}^T(J\dot{\theta} - \dot{B}) + k_1 e) \geq -k_2(\overrightarrow{||BA||}^T(J\dot{\theta} - \dot{B}) + k_1 e) \\ &- J_o\ddot{\theta} - \dot{J}_o\dot{\theta} - \overrightarrow{||BA||}^T\ddot{B} + k_1(-J_o\dot{\theta} - \overrightarrow{||BA||}^T\dot{B}) \geq \\ &- k_2(-J_o\dot{\theta} - \overrightarrow{||BA||}^T\dot{B} + k_1 e) \\ &J_o\ddot{\theta} + \dot{J}_o\dot{\theta} + \overrightarrow{||BA||}^T\ddot{B} + k_1(J_o\dot{\theta} + \overrightarrow{||BA||}^T\dot{B}) \leq \\ &- k_2(J_o\dot{\theta} + \overrightarrow{||BA||}^T\dot{B} - k_1 e), \end{aligned} \tag{9}$$

therefore, we can obtain the obstacle avoidance inequality in acceleration level:

$$\begin{aligned} J_o\ddot{\theta} \le & -k_2(J_o\dot{\theta} + \overrightarrow{||BA||}^T\dot{B} - k_1 e) \\ & - k_1(J_o\dot{\theta} + \overrightarrow{||BA||}^T\dot{B}) - \dot{J}_o\dot{\theta} - \overrightarrow{||BA||}^T\ddot{B}. \end{aligned} \tag{10}$$

where $-\overrightarrow{||BA||}^T J = J_o$. Let the right side of inequality (10) be denoted by μ; Equation (10) is then equivalent to

$$J_o\ddot{\theta} \le \mu. \tag{11}$$

So far, the construction of the inequality collision avoidance strategy on the acceleration level, i.e., Equation (11), is completed.

2.3. QP Problem Statement

For a redundant manipulator, due to the redundancy, it is possible to perform the primary and secondary tasks simultaneously. In view of $m < n$, many solutions satisfying Equation (11) exist. To choose a better solution from them, the secondary task can be set as the optimization of some performance indices such as joint velocity minimization, joint acceleration minimization, joint jerk minimization, etc. In this study, the acceleration level kinematic control of the redundant manipulator was considered, and the joint-acceleration-norm minimization was thus chosen. On one hand, in terms of practical industrial applications, the robot is often expected to perform some repetitive tasks such as palletizing and welding. To make the kinematic control of manipulator repetitive, the RMP scheme was proposed and investigated in Zhang et al. (2009), Xiao and Zhang (2013), and Jin et al. (2018), and it was constructed as the minimization of the displacements between the $\theta(t)$ and $\theta(0)$, where $\theta(0)$ denotes the initial joint angle. On the other hand, to avoid the joint drift problem, another performance index, i.e., the repetitive motion planning, was also adopted in this paper:

1) Minimum acceleration norm (MAN):

$$U = \ddot{\theta}^T\ddot{\theta}/2. \tag{12}$$

2) Repetitive motion planning (RMP):

$$U = (\ddot{\theta} + d_1(\theta - \theta(0)))^T(\ddot{\theta} + d_1(\theta - \theta(0)))/2, \tag{13}$$

where $d_1 > 0$ is designed as a positive constant determined by the designer based on the experimental results, which is used to scale the magnitude of the displacements $\theta - \theta(0)$. Parameters θ and $\theta(0)$ denote the current joint angle and the initial joint angle of the manipulator, respectively.

Let $\eta = d_1(\theta - \theta(0))$, assigning a weight $\omega_1 = 0.5$ and $\omega_2 = 0.5$ to the MAN and the RMP schemes, respectively; $\omega_1 + \omega_2 = 1$, the bi-criteria acceleration level obstacle avoidance, and kinematic control problem of a redundant manipulator are formulated as an QP problem as follows:

$$\begin{aligned} \min \quad & \ddot{\theta}^T\ddot{\theta}/4 + (\ddot{\theta} + \eta)^T(\ddot{\theta} + \eta)/4, && \text{(14a)} \\ \text{s.t.} \quad & J\ddot{\theta} = \ddot{r}_d - \dot{J}\dot{\theta}, && \text{(14b)} \\ & J_o\ddot{\theta} \le \mu, && \text{(14c)} \\ & \theta^- \le \theta \le \theta^+, && \text{(14d)} \\ & \dot{\theta}^- \le \dot{\theta} \le \dot{\theta}^+, && \text{(14e)} \\ & \ddot{\theta}^- \le \ddot{\theta} \le \ddot{\theta}^+, && \text{(14f)} \end{aligned}$$

where Equation (14a) denotes the objective function to be minimized. Equations (14b) and (14c) denote the motion planning scheme and obstacle avoidance scheme, respectively. Equations (14d)–(14f) are the physical constraints. Parameters $\ddot{\theta}, \dot{\theta}, \theta$ denote joint acceleration vector, joint velocity vector and joint angle vector of the robot manipulator, respectively. θ^-, $\dot{\theta}^-$, $\ddot{\theta}^-$ and θ^+, $\dot{\theta}^+$, $\ddot{\theta}^+$ are lower bound and upper bound of θ, $\dot{\theta}$, $\ddot{\theta}$, respectively. r_d are the desired trajectory that the robot is expected to track. $\ddot{r}_d$ is the time derivation of $\dot{r}_d$, and $\dot{r}_d$ is the derivation of r_d.

3. QP REFORMULATION AND RNN CONTROLLER

3.1. QP Reformulation

For Equation (14b), to achieve a higher tracking accuracy to the desired trajectory, a feedback is introduced, and Equation (14b) is rewritten as

$$J\ddot{\theta} = \ddot{r}_d - \dot{J}\dot{\theta} - \beta(J\dot{\theta} - \dot{r}_d) - \gamma(r - r_d), \tag{15}$$

where $\beta > 0 \in \mathbb{R}$ and $\gamma > 0 \in \mathbb{R}$ are the feedback gains, and r is the actual trajectory achieved by manipulator under the designed controller. In addition, For Equations (14d)–(14f), it is obvious that they are located at different levels, which makes it impossible to directly solve Equation (14). Following Guo and Zhang (2014, 2019), Equations (14d)–(14f) can be incorporated in the acceleration level, i.e.,

$$\begin{aligned} \xi^+ &= \min\{\kappa_1(\theta^+ - \vartheta - \theta), \kappa_2(\dot{\theta}^+ - \dot{\theta}), \ddot{\theta}^+\}, \\ \xi^- &= \max\{\kappa_1(\theta^- + \vartheta - \theta), \kappa_2(\dot{\theta}^- - \dot{\theta}), \ddot{\theta}^-\}, \end{aligned} \tag{16}$$

where $\vartheta > 0 \in \mathbb{R}, \kappa_1 > 0 \in \mathbb{R}$, and $\kappa_2 > 0 \in \mathbb{R}$. The QP problem (14) can thus be reformulated:

$$\begin{aligned} \min \quad & \ddot{\theta}^T\ddot{\theta}/4 + (\ddot{\theta} + \eta)^T(\ddot{\theta} + \eta)/4, && \text{(17a)} \\ \text{s.t.} \quad & J\ddot{\theta} = \ddot{r}_d - \dot{J}\dot{\theta} - \beta(J\dot{\theta} - \dot{r}_d) - \gamma(r - r_d), && \text{(17b)} \\ & J_o\ddot{\theta} \le \mu, && \text{(17c)} \\ & \xi^+ = \min\{\kappa_1(\theta^+ - \vartheta - \theta), \kappa_2(\dot{\theta}^+ - \dot{\theta}), \ddot{\theta}^+\}, && \text{(17d)} \\ & \xi^- = \max\{\kappa_1(\theta^- + \vartheta - \theta), \kappa_2(\dot{\theta}^- - \dot{\theta}), \ddot{\theta}^-\}. && \text{(17e)} \end{aligned}$$

*Remark*2: The weight factors of both the MAN and RMP schemes are set at the same value, meaning that the MAN and RMP schemes are viewed as equally important. For different weights ω_1 and $\omega_2 = 1 - \omega_1$, the minimized objective function is different.

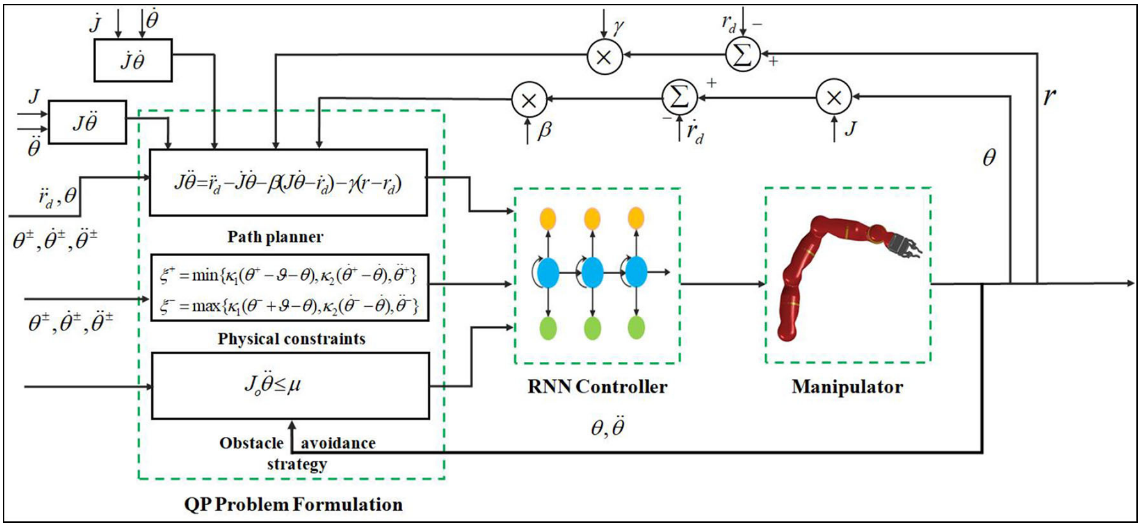

$\dot{J}$
$\dot{\theta}$
$\dot{J}\dot{\theta}$
J
$\ddot{\theta}$
$J\ddot{\theta}$
$\ddot{r}_d,\theta$
$\theta^{\pm},\dot{\theta}^{\pm},\ddot{\theta}^{\pm}$
$J\ddot{\theta}=\ddot{r}_d-\dot{J}\dot{\theta}-\beta(J\dot{\theta}-\dot{r}_d)-\gamma(r-r_d)$
Path planner
$\xi^{+}=\min\{\kappa_1(\theta^{+}-\vartheta-\theta),\kappa_2(\dot{\theta}^{+}-\dot{\theta}),\ddot{\theta}^{+}\}$
$\xi^{-}=\max\{\kappa_1(\theta^{-}+\vartheta-\theta),\kappa_2(\dot{\theta}^{-}-\dot{\theta}),\ddot{\theta}^{-}\}$
$\theta^{\pm},\dot{\theta}^{\pm},\ddot{\theta}^{\pm}$
Physical constraints
$J_o\ddot{\theta}\leq\mu$
Obstacle avoidance strategy
QP Problem Formulation
γ
r_d
$-$
$+$
β
$+$
$-$
$\dot{r}_d$
J
r
θ
RNN Controller
$\theta,\ddot{\theta}$
Manipulator

FIGURE 2 | Block diagram of the acceleration-level kinematic motion control of redundant manipulator with obstacle avoidance (17c) and physical constraints (17d)-(17e) under the designed RNN controller (21).

3.2. RNN Controller

In this part, we would design an RNN-based dynamic controller to solve Equation (17) recursively. Specifically, for Equation (17), a lagrange function is defined as

$$L=\ddot{\theta}^T\ddot{\theta}/4+(\ddot{\theta}+\eta)^T(\ddot{\theta}+\eta)/4+\lambda_1^T(B_{right}-J\ddot{\theta})+\lambda_2^T(J_o\ddot{\theta}-\mu), \tag{18}$$

where $B_{right}=\ddot{r}_d-\dot{J}\dot{\theta}-\beta(J\dot{\theta}-\dot{r}_d)-\gamma(r-r_d)$, λ_1 and λ_2 is the Lagrange multiplier. Based on the KKT conditions, the optimal solution of Equation (18) can be equivalently rewritten as

$$\ddot{\theta}=P_{\Omega}(\ddot{\theta}-\frac{\partial L}{\partial\ddot{\theta}}), \tag{19a}$$

$$J\ddot{\theta}=B_{right}, \tag{19b}$$

$$\begin{cases}\lambda_2=0 & \text{if } J_o\ddot{\theta}\leq\mu,\\ \lambda_2>0 & \text{othewise.}\end{cases} \tag{19c}$$

where P_{Ω} is a projection operation to a set Ω, and $P_{\Omega}(x)=\text{argmin}_{y\in\Omega}||y-x||$ Li et al. (2016). Equation (19c) can be further written as

$$\lambda_2=\max((\lambda_2+J_o\ddot{\theta}-\mu),0). \tag{20}$$

The designed RNN controller is:

$$\epsilon\dddot{\theta}=-\ddot{\theta}+P_{\Omega}(-\frac{1}{2}\eta+J^T\lambda_1-J_o^T\lambda_2), \tag{21a}$$

$$\epsilon\dot{\lambda}_1=J\ddot{\theta}-B_{right}, \tag{21b}$$

$$\epsilon\dot{\lambda}_2=\max((J_o\ddot{\theta}-\mu+\lambda_2),0)-\lambda_2, \tag{21c}$$

where $\epsilon>0$ is a constant that is used to scale the convergence rate of the neural network. **Figure 2** shows a block diagram of the acceleration-level kinematic motion control of a redundant manipulator with obstacle avoidance (17c) and physical constraints (17d)-(17e) under the designed RNN controller (21).

TABLE 1 | The D-H parameter of the robot manipulator employed in this paper and simulation parameters setup.

Link	a(m)	α(rad)	d(m)	Parameter	Value	Parameters	Value
1	0.296	0	0	k_1	7	k_2	7
2	0.296	0	0	ϵ	0.002	ϑ	0.1
3	0.296	0	0	κ_1	20	κ_2	20
4	0.212	0	0	β	20	γ	20
				θ^-	-2(rad)	θ^+	2(rad)
				$\dot{\theta}^-$	-2(rad/s)	$\dot{\theta}^+$	2(rad/s)
				$\ddot{\theta}^-$	-2(rad/s)	$\ddot{\theta}^+$	2(rad/s^2)
				d	0.1(m)	d_1	10

The left side is the DH parameter. The right side is simulation parameters involved in the simulative experiments.

4. SIMULATION

In this paper, the simulation experiment was performed on a plane four-DOF robot manipulator to validate the feasibility of the control scheme Equation (21). **Table 1** gives the corresponding DH parameters of the employed manipulator and the parameter values involved in the simulative experiment, respectively.

4.1. Circle Trajectory Tracking

4.1.1. Static Obstacle

In this experiment, the robot is expected to track a circle trajectory with definition of $r_d=[0.6470+0.1\cos(0.5t),0.3125+0.1\sin(0.5t)]^T$ whose radius is 0.1. Assume that position of the obstacle is centered at $[-0.1,0.3]^T$m. The initial joint angle is chosen as $q(0)=[\pi/2,-\pi/3,-\pi/4,0]^T$rad with the joint velocity and joint acceleration initializing as zero. The simulation time is set as 25s with step size being 0.001. Collision avoidance and trajectory tracking results are shown in **Figure 3**, where the single static obstacle is considered. **Figures 3A,B** show the trajectory tracked by the manipulator under RNN controller without and with the obstacle avoidance strategy Equation (17c), respectively, where the corresponding tracking results are used at $t = 1, 6, 9$ and $t = 13s$. Following them, when not considering the collision avoidance, though the manipulator successfully tracks the desired circle trajectory, distances between the obstacle and both the first and second links of the manipulator are small, and this allows the collision between them to happen. For practical industrial applications, this control method will inevitably lead to the tracking failure of the expected behavior. After introducing the obstacle avoidance strategy, as the distance between the nearest point on the manipulator and the encountered environmental obstacle, and as the obstacle is less than the setting safety distance 0.1, the inequality Equation (17c) comes in the control command, enabling the manipulator to escape the obstacle (see **Figure 3B**); under the path-tracking controller Equation (17b), the robot moves along the desired trajectory as expected with a promising tracking error being the 10^{-3} order (see **Figure 3C**). As the initial point of the end-effector of the manipulator coincides with the expected tracking trajectory, the tracking error is always

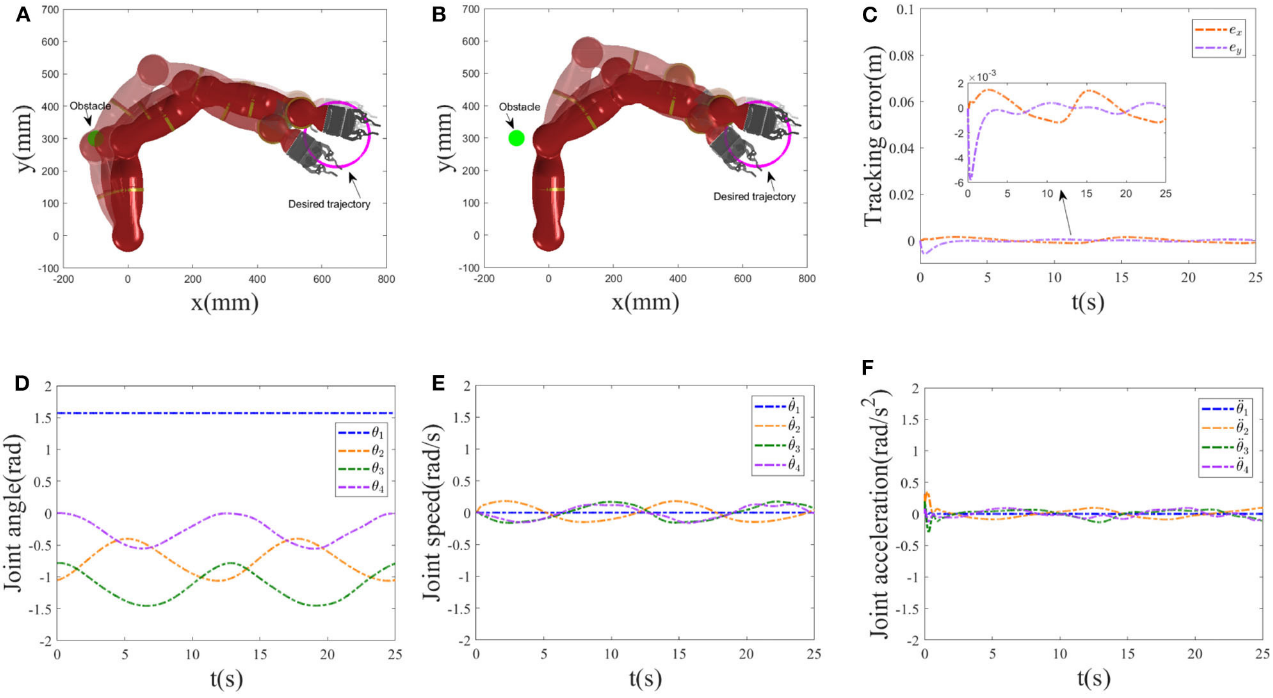


FIGURE 3 | Static obstacle collision avoidance results. **(A)** Trajectory tracked by the manipulator without Equation (17c) at $t = 1, 6, 9$ and $t = 13s$. **(B–F)** Simulation results achieved by the manipulator with Equation (17c). **(B)** Tracked trajectory. **(C)** Tracking error at *x*-axis, *y*-axis. **(D)** Joint angle profiles. **(E)** Joint velocity profiles. **(F)** Joint acceleration profiles.

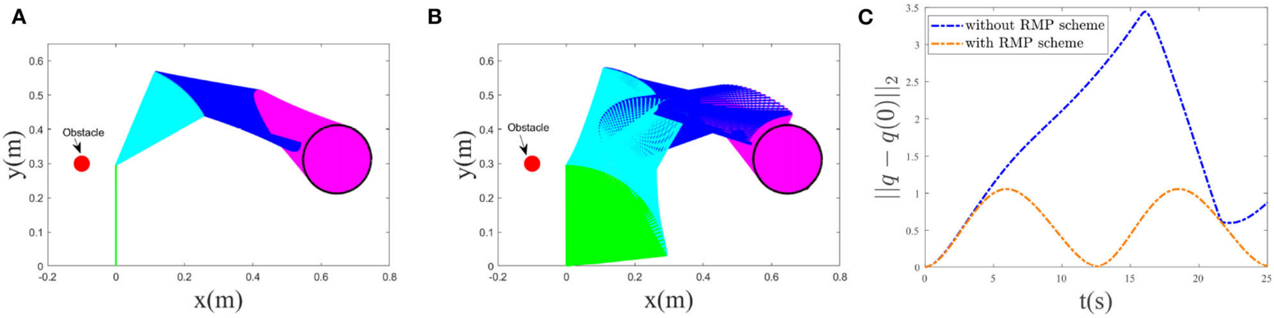


FIGURE 4 | Comparative results between bi-criteria scheme considering the MAN and RMP and the MAN scheme. **(A)**: Tracking trajectory result corresponding to bi-criteria scheme; **(B)** Tracking result corresponding to MAN scheme. **(C)**: Comparison of $||q - q(0)||_2$ with and without RMP scheme.

satisfying from the start to the end of simulation. **Figures 3D–F** show the joint angles, joint velocities, and joint accelerations profiles, respectively. Among them, the lines are relatively smooth and not sharp, and they do not exceed the setting the bound constrains.

4.1.2. Verification of RMP

Now, we start to validate the effectiveness of the RMP scheme. In this paper, the performance index was chosen as a bi-criteria optimization, i.e., a weighted combination of the MAN scheme Equation (12) and the RMP scheme Equation (13). The desired trajectory tracking result corresponding to the bi-criteria scheme is shown in **Figure 4A**, and the one corresponding to the MAN scheme is illustrated in **Figure 4B**. Comparing **Figures 4A,B**, the joint-drift problem at the acceleration level can be seen to be solved by considering the RMP scheme. In addition, a comparison between $||q - q(0)||_2$ with and without the RMP scheme is illustrated in **Figure 4C**, showing that, for the scheme considering RMP, $||q - q(0)||_2$ would be guaranteed to converge to zero when $t = T, 2T$ and change periodically. If not considering the RMP scheme, $||q - q(0)||_2$ increases when $t = T, 2T$ and is haphazard. Moreover, based on the simulative results shown in **Figures 3D,E**, we observe that when $t = T, 2T$, joint angles and joint velocities of the manipulator are guaranteed to return to their initial configurations. The RMP scheme can therefore be said to be effective.

4.1.3. Dynamic Obstacle

Pedestrians or other objects with dynamic property may break into the motion range of the robot. In this part, we consider the collision avoidance between the robot and a dynamic obstacle, and snapshots of the manipulator avoiding a dynamic obstacle at different time t are given in **Figure 5**, where the real shadow denotes the collision avoidance result achieved by the manipulator under the RNN dynamic controller with the inequality collision avoidance strategy Equation (17c), and the virtual shadow corresponds to ones without Equation (17c). The motion trajectory of the considered dynamic obstacle is set as $[-0.1 + 0.01t, 0.3]^T$ with simulation time being 15s. Macroscopically, when $t = 3s, 6s, 9s$ and 12s, if not considering the collision avoidance, the manipulator collides with the

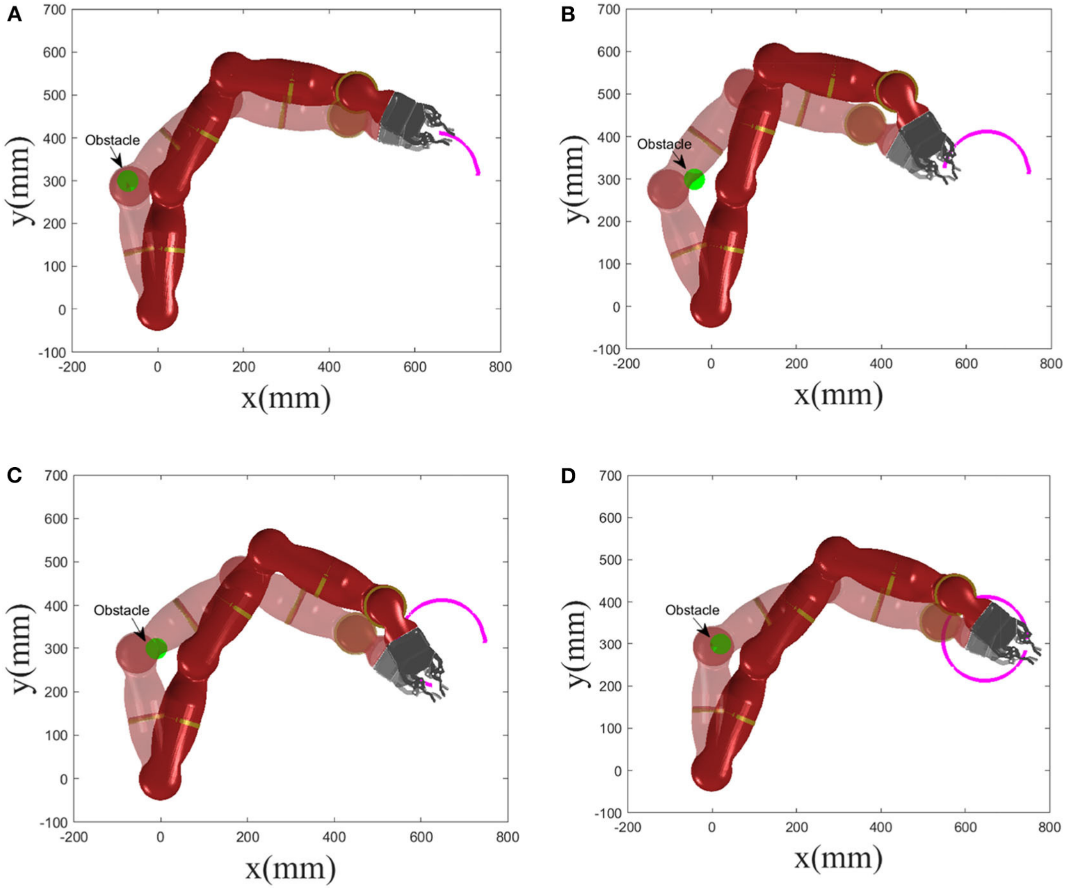


FIGURE 5 | Snapshots of manipulator avoiding a dynamic obstacle at different time t, where the real shadow denotes collision avoidance result with Equation (17c), and the virtual shadow corresponds to ones without Equation (17c). **(A)** t = 3 s. **(B)** t = 6 s. **(C)** t = 9 s. **(D)** t = 12 s.

TABLE 2 | A dynamic obstacle is considered: the nearest point on the manipulator to the obstacle obtained by the controller with and without the obstacle avoidance scheme Equation (17c) at different time *t* and the distance between the nearest point and the obstacle.

Time (s)	Nearest point with Equation (17c)	Distance (m)	Nearest point without Equation (17c)	Distance (m)
$t = 1$	$[0.0089, 0.2959]^T$	0.0990	$[-0.0107, 0.2958]^T$	0.0794
$t = 3$	$[0.0285, 0.2903]^T$	0.0990	$[-0.0616, 0.2910]^T$	0.0123
$t = 6$	$[0.0570, 0.2803]^T$	0.0990	$[-0.0677, 0.3215]^T$	0.0351
$t = 9$	$[0.0845, 0.2704]^T$	0.0991	$[-0.0210, 0.3151]^T$	0.0187
$t = 12$	$[0.1112, 0.2612]^T$	0.0991	$[0.0178, 0.3039]^T$	0.0045
$t = 15$	$[0.1450, 0.2722]^T$	0.0989	$[0.0136, 0.3417]^T$	0.0554

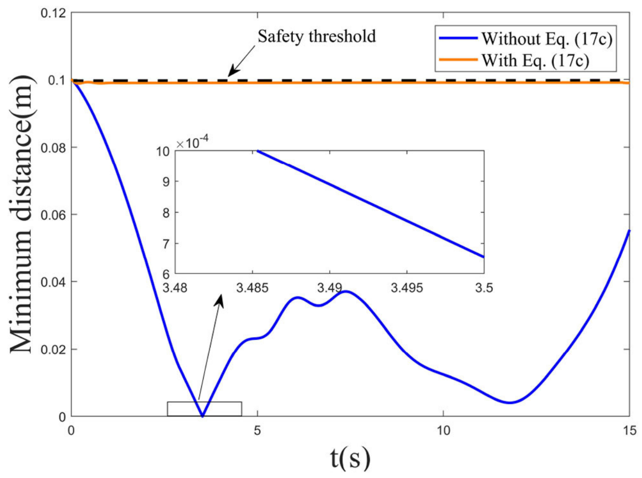


FIGURE 6 | Minimum distance profile between the robot and the considered dynamic obstacle achieved by the controller without and with Equation (17c).

dynamic obstacle. After introducing the collision avoidance strategy, under the control of the controller, the robot escapes the obstacle by changing its joint angles and being maintained outside the non-safety distance. To further show the effectiveness of the collision avoidance scheme Equation (17c), **Table 2** gives the corresponding Cartesian coordinates of the nearest point on the manipulator to the obstacle obtained by the controller with and without Equation (17c) at different time t and the distance between the nearest point and the obstacle. Obviously, without Equation (17c), the distance is significantly less than the safety distance 0.1, meaning that the collision will happen with high probability. By contrast, after introducing Equation (17c), the collision avoidance scheme comes in the control command,and enables the manipulator to escape the obstacle and maintain a safe distance. Based on **Table 2**, under the control of the dynamic controller, the distances between the nearest point on the manipulator and the obstacle maintain 0.099, which is very close to 0.1. In addition, we give the minimum distance profile between the manipulator and the dynamic obstacle achieved by the controller without and with Equation (17c) for illustration, as shown in **Figure 6**. It is more obvious and intuitive than **Table 2**, if not taking the collision avoidance strategy Equation (17c) into account, and the robot would collide with the obstacle at $t = 3.5s$ owing to the distance being 7×10^{-4}m. It is therefore concluded that the proposed collision avoidance strategy is effective.

Remark 3: Compared to the setting safety threshold 0.1m, the distances between the nearest point to the obstacle on the manipulator and the obstacle at different time were maintained at 0.099 in the dynamic obstacle avoidance experiment. We observed that the minimal distance achieved by the controller was a somewhat smaller than 0.1, and this is attributed to the sampling interval adopted in the simulation experiment. Compared to the generated results without Equation (17c), the proposed collision avoidance strategy can help the robot to avoid collision with the obstacle on the whole, and the slight difference can thus be ignored.

4.2. Three-Ring Trajectory Tracking

To further validate the feasibility of the dynamic controller [Equation (17)] integrating path tracking and obstacle avoidance strategies, in this experiment, the robot tracks a three-ring trajectory. The tracked trajectory is defined as $r_d = [0.05\cos(\pi t/10) - 0.025\cos(4\pi t/10) + 0.4888, 0.05\sin(\pi t/10) - 0.025\sin(4\pi t/10) + 0.0040]^T$. Position of the obstacle is located on $[-0.025, 0.25]^T$m. The initial joint angle is valued as $q(0) = [\pi/2.5, -\pi/3, -\pi/4, -\pi/2]^T$rad, and the initial joint velocity is set as $[-0.0210, -0.0101, 0.0032, 0.0092]^T$rad/s. Other experimental parameters are the same as the previous circle trajectory tracking. The simulation time is set as 20s with step size being 0.001. Static obstacle collision avoidance and trajectory tracking results are illustrated in **Figure 7**. **Figures 7A–C** show snapshots of manipulator avoiding a static obstacle at different time $t = 4s$, $t = 8s$, and $t = 20s$, respectively, where the real shadow denotes collision avoidance result with Equation (17c), the virtual shadow corresponds to ones without Equation (17c). We can observe that when not considering the collision avoidance scheme Equation (17c), the distance between the first link of the manipulator and the obstacle is tiny (e.g., when for $t = 8$, the minimal distance is 0.0785 with the nearest point being $[0.0516, 0.2330]^T$). After introducing Equation (17c), compared to the previous, the distance between them is enlarged and maintained outside the non-safety region (see **Figure 7D**). **Figure 7D** gives the distance between the nearest point on the first link of the manipulator to the obstacle and the obstacle. When $t = 4s$ and $t = 8s$, the distance between them is maintained as 0.1, i.e., the setting safety distance. For $t = 8$, the neatest point returned by the computer is $[0.0702, 0.2195]^T$. Except when successfully avoiding the obstacle, the robot also accomplishes the desired three-ring path tracking as expected. Based on **Figures 7E,F**, it is obvious that the actual trajectory achieved by the manipulator is coincident with the desired trajectory, and the tracking errors at x-axis and y-axis reach 10^{-3} level. **Figures 7G,I** show the joint-angle profiles, joint-velocity profiles, and joint-acceleration profiles of the manipulator, respectively. Following them, therefore, we can say that the

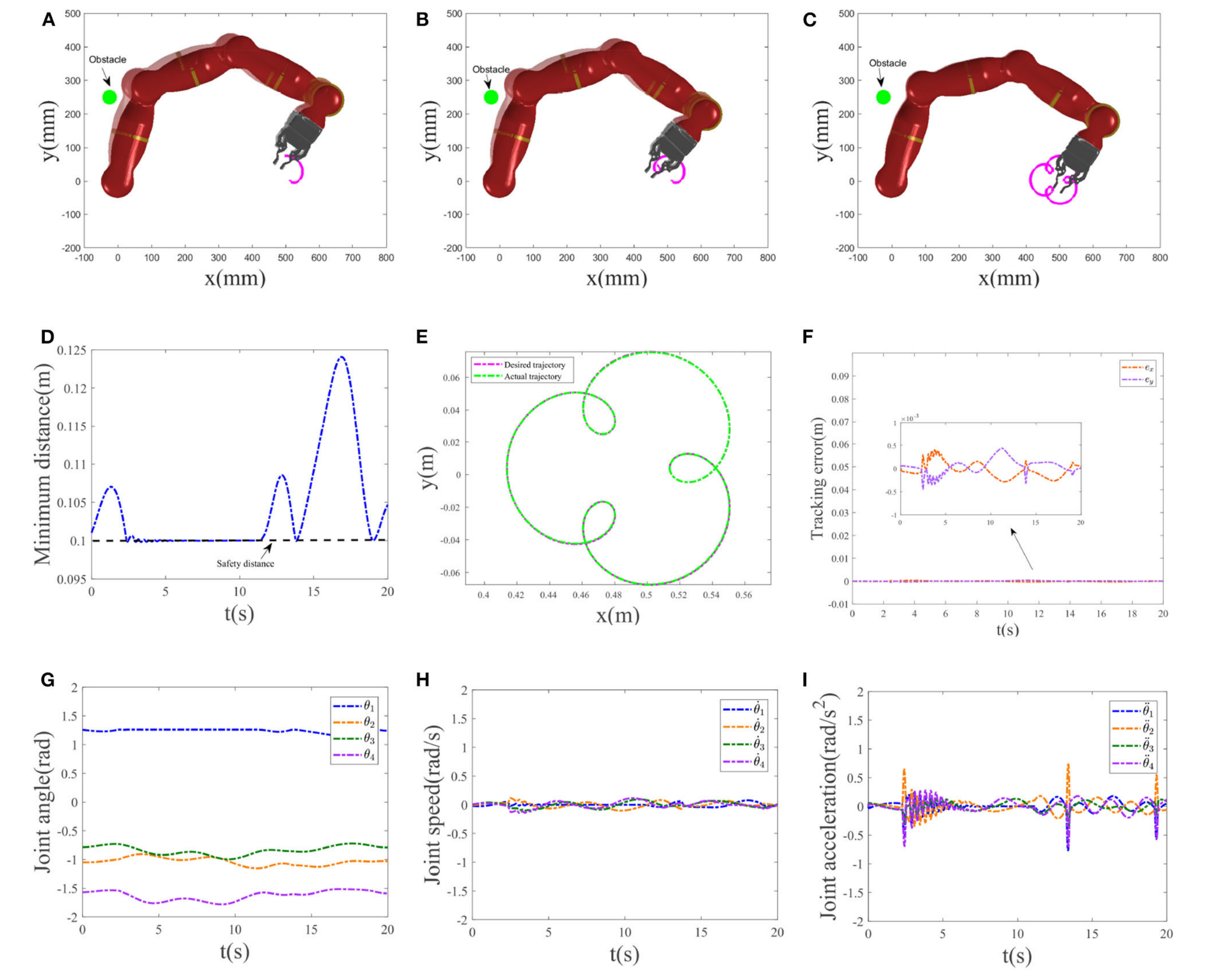


FIGURE 7 | Obstacle collision avoidance results. **(A–C)** Snapshots of the manipulator avoiding a static obstacle at $t = 4s$, $t = 8s$, and $t = 20s$, respectively. **(D)** Minimum distance profile. **(E)** Trajectory tracked by the manipulator and the desired trajectory. **(F)** Tracking error at *x*-axis, *y*-axis. **(G)** Joint-angle profiles. **(H)** Joint-velocity profiles. **(I)** Joint-acceleration profiles.

proposed obstacle avoidance scheme Equation (17c) and the designed RNN dynamic controller Equation (17) are effective for solving the kinematic motion problem of a redundant manipulator at the acceleration level.

Comparative results between the bi-criteria scheme and the MAN scheme are shown in **Figure 8**. As the movement period is $20s$, in this experiment, the simulation time is set as $40s$. In addition, $d_1 = 15$. **Figures 8A,B** show the tracking trajectory result corresponding to bi-criteria scheme and MAN scheme, respectively. A comparison between $||q-q(0)||_2$ with and without the RMP scheme is illustrated in **Figure 8C**. For the scheme considering RMP, $||q - q(0)||_2$ would be guaranteed to converge to zero when $t = T, 2T$ and changes periodically. If is not considering the RMP scheme, when $t = T, 2T$, the current joint-angle state q does not return the initial joint-angle state $q(0)$.

Remark 4: As one of the important performance indices, here, we start to show the effectiveness of the MAN scheme. **Figure 9** gives comparative results of the joint-acceleration norm $||\ddot{\theta}||$ achieved by the pseudo-inverse method and the RNN-based method proposed in this paper for two trajectories. **Figure 9A** corresponds to the circle trajectory tracking experiment. **Figure 9B** corresponds to the three-ring trajectory tracking experiment. Note that, in this experiment, the cost function only considers the MAN scheme to show the effectiveness of the MAN scheme, not involving the RMP and the collision avoidance. In general, the pseudo-inverse method is deemed as a persuasive solution, therefore, it is employed to compare with our RNN-based method. Following **Figure 9A**, it is observed that $||\ddot{\theta}||$ quickly coincides with one achieved by the pseudo-inverse method although it is slightly different at initial time.

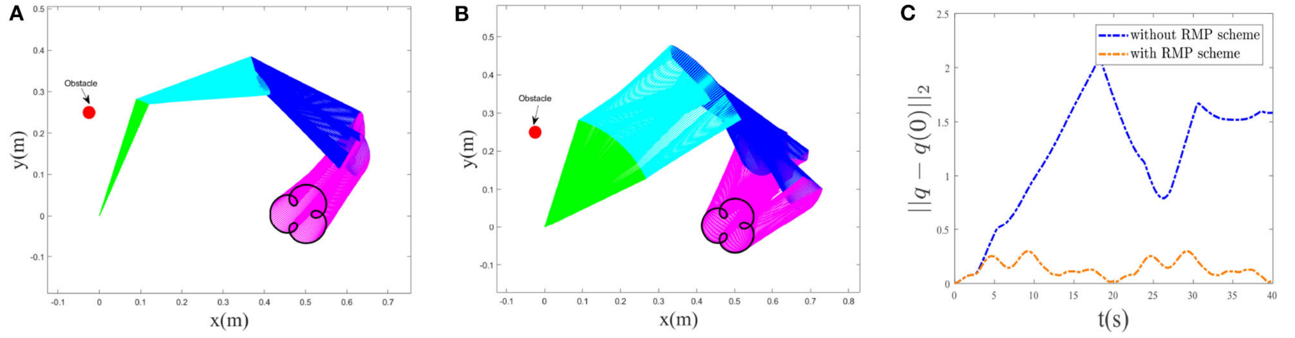


FIGURE 8 | Comparative results between bi-criteria scheme considering the MAN and RMP, and the MAN scheme. **(A)** Tracking trajectory result corresponding to bi-criteria scheme; **(B)** Tracking result corresponding to MAN scheme. **(C)** Comparison of $||q - q(0)||_2$ with and without RMP scheme.

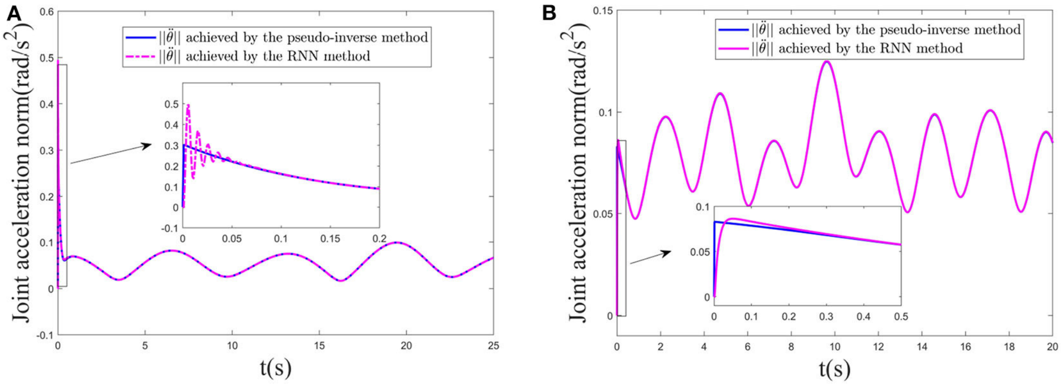


FIGURE 9 | Joint acceleration norm $||\ddot{\theta}||$ comparison achieved by the pseudo-inverse method and the RNN-based method proposed in this paper, respectively. **(A)** Circle trajectory tracking. **(B)** Three-ring trajectory tracking.

TABLE 3 | Comparison between the proposed scheme in this paper and the existing QP-based acceleration level obstacle avoidance schemes.

Methods	Performance indices	Physical constraints	Static obstacle	Dynamic obstacle
This paper	Bi-criteria	Yes	Yes	Yes
Guo and Zhang (2019)	MAN	Yes	Yes	No
Guo and Zhang (2014)	MAN	Yes	Yes	No
Xiao and Zhang (2013)	RMP	Yes	/	/
Guo and Li (2016)	MAN	No	Yes	No

/ denotes the obstacle avoidance is not considered in Xiao and Zhang (2013).

The conclusion is also same for the three-ring trajectory (as shown in **Figure 9B**), consequently, the effectiveness of the MAN scheme is validated. In addition, the pseudo-inverse method does not handle the physical constraints such as joint angles, joint accelerations, consequently, the RNN-based method in this paper is adopted.

4.3. Comparison

As described in the previous sections, obstacle avoidance of the redundant manipulator has been investigated for decades, and the research has been fruitful. However, the existing products mainly focus on the velocity level. At present, only a small amount of attention is paid to the obstacle avoidance of the redundant manipulator at the acceleration level (not to mention the bi-criteria acceleration-level obstacle avoidance). There are few related works that have been reported. In this study based on the QP optimization, we investigated the bi-criteria acceleration-level obstacle avoidance of the redundant manipulator. To highlight the proposed controller scheme in this paper, comparisons between our scheme and the existing QP-based acceleration level obstacle avoidance schemes were conducted, and the comparative results are illustrated in **Table 3**. In Xiao and Zhang (2013), the obstacle avoidance scheme was not considered. For their works proposed by Guo and Zhang (2014, 2019) and Guo and Li (2016), in their collision-avoidance schemes, the inner and outer safety thresholds were considered. In Guo et al. (2018), a noise-tolerant obstacle avoidance strategy was introduced by proposing an integration-enhanced error function. As this paper is not investigated from the perspective of QP, it is not listed in **Table 3**. Following **Table 3**, this study first investigated the bi-criteria performance optimization-based acceleration-level obstacle avoidance of the redundant manipulator from the QP perspective. Moreover, the dynamic obstacle as also considered.

Combining all simulative results, in summary, the proposed collision avoidance scheme has the ability to find the nearest point on the manipulator to obstacle, and it can enable the manipulator to avoid collision with the environmentally static and dynamic obstacles. Under the designed RNN controller, the manipulator also accurately achieves the desired trajectory tracking task.

5. CONCLUSIONS

We shed some light on the acceleration-level kinematic motion control problem of the redundant manipulator with obstacle avoidance in this paper. An improved inequality obstacle avoidance method is introduced, and it can find the nearest point on every link of a manipulator to an obstacle. By keeping the minimal distance between them outside the non-safety region at all times, the safety is ensured. Minimizing the combination integrating the joint-acceleration norm and repetitive motion planning as the objective function, a QP optimization problem is established where the desired motion behavior and obstacle avoidance are formulated as equality and inequality constraints rebuilt at the acceleration level. The inherent physical constraints of the manipulator are also incorporated. An RNN-based neural dynamic controller is designed to solve the resultant QP problem. Simulative results performing on four-link planer manipulator validate the feasibility of the designed control scheme, when the minimal distance between robot and obstacle violates the setting safety criticality, the collision avoidance strategy come in the control command, the robots successfully avoid collision with the environmental obstacles. If no collision is detected, the robot performs the desired trajectory tracking task with a promising tracking error. In this paper, we only considered the obstacle avoidance problem of a single redundant manipulator. For the multiple robot system, the obstacle avoidance scheme should not only consider collision between the manipulator and the environment, but also collision between the manipulators each other. This is a challenge problem. In the future work, the obstacle avoidance problem of multiple robot manipulators system will be considered.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

WZ developed the idea for this study, derived mathematical equations, and wrote to the paper. Some drawings in this paper and the program code were completed by XL. XC and XS completed the corresponding MATLAB simulation. GT contributed to refining the paper and proposed amendments. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the Guangdong Province Key Areas R&D Program (Grant Nos. 2020B090925001 and 2019B090919002), Foshan Key Technology Research Project (Grant No. 1920001001148), Natural Science Foundation of Guangdong Province (Grant No. 2020A1515010631), Guangzhou Science Research Plan—Major Project (Grant No. 201804020095), and GDAS' Project of Science and Technology Development (Grant No. 2018GDASCX-0115).

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their valuable comments and suggestions.

REFERENCES

Carloni, R., Lippiello, V., D'Auria, M., and Fumagalli, M. (2013). Robot vision: obstacle-avoidance techniques for unmanned aerial vehicles. *IEEE Robot. Automat. Mag.* 20, 22–31. doi: 10.1109/MRA.2013.2283632

Chen, D., Li, S., Li, W., and Wu, Q. (2020b). A multi-level simultaneous minimization scheme applied to Jerk-Bounded redundant robot Manipulators. *IEEE Trans. Automat. Sci. Eng.* 17, 463–474. doi: 10.1109/TASE.2019.2931810

Chen, D., Li, S., and Liao, L. (2019). A recurrent neural network applied to optimal motion control of mobile robots with physical constraints. *Appl. Soft Comput.* 85, 1568–4946. doi: 10.1016/j.asoc.2019.105880

Chen, D., Li, S., Wu, Q., and Luo, X. (2020a). New disturbance rejection constraint for redundant robot manipulators: an optimization perspective. *IEEE Trans. Indus. Inform.* 16, 2221–2232. doi: 10.1109/TII.2019.2930685

Guo, D., and Li, K. (2016). "Acceleration-level obstacle-avoidance scheme for motion planning of redundant robot manipulators," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (Qingdao), 1313–1318. doi: 10.1109/ROBIO.2016.7866508

Guo, D., Xu, F., Yan, L., Nie, Z., and Shao, H. (2018). A new noise-tolerant obstacle avoidance scheme for motion planning of redundant robot manipulators. *Front. Neurorobot.* 12:51. doi: 10.3389/fnbot.2018.00051

Guo, D., and Zhang, Y. (2014). Acceleration-level inequality-based man scheme for obstacle avoidance of redundant robot manipulators. *IEEE Trans. Indus. Electron.* 61, 6903–6914. doi: 10.1109/TIE.2014.2331036

Guo, D., and Zhang, Y. (2019). Acceleration-level obstacle avoidance of redundant manipulators. *IEEE Access* 7, 183040–183048. doi: 10.1109/ACCESS.2019.2960399

Jin, L., Li, S., Hu, B., and Yi, C. (2018). Dynamic neural networks aided distributed cooperative control of manipulators capable of different performance indices. *Neurocomputing* 291, 50–58. doi: 10.1016/j.neucom.2018.02.059

Ju, T., Liu, S., Yang, J., and Sun, D. (2014). Rapidly exploring random tree algorithm-based path planning for robot-aided optical manipulation of biological cells. *IEEE Trans. Automat. Sci. Eng.* 11, 649–657. doi: 10.1109/TASE.2013.2289311

Kim, J., and Khosla, P. K. (1992). Real-time obstacle avoidance using harmonic potential functions. *IEEE Trans. Robot. Automat.* 8, 338–349. doi: 10.1109/70.143352

Lee, K. K., and Buss, M. (2007). "Obstacle avoidance for redundant robots using Jacobian transpose method," in *IEEE/RSJ International Conference on Intelligent Robots & Systems* (San Diego, CA: IEEE), 3509–3514.

Li, S., He, J., Li, Y., and Rafique, M. U. (2016). Distributed recurrent neural networks for cooperative control of manipulators: a game-theoretic perspective. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 415–426. doi: 10.1109/TNNLS.2016.2516565

Li, S., Wang, H., and Rafique, M. (2018b). A novel recurrent neural network for manipulator control with improved noise tolerance. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 1908–1918. doi: 10.1109/TNNLS.2017.2672989

Li, S., Zhou, M., and Luo, X. (2018a). Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 4791–4801. doi: 10.1109/TNNLS.2017.2770172

Li, X., Zhihao, X., Shuai, L., Hongmin, W., and Xuefeng, Z. (2020). Cooperative kinematic control for multiple redundant manipulators under partially known information using recurrent neural network. *IEEE Access* 8, 40029–40038. doi: 10.1109/ACCESS.2020.2974248

Volpe, R., and Khosla, P. (1990). Manipulator control with superquadric artificial potential functions: theory and experiments. *IEEE Trans. Syst. Man Cybernet.* 20, 1423–1436. doi: 10.1109/21.61211

Xiao, L., and Zhang, Y. (2013). Acceleration-level repetitive motion planning and its experimental verification on a six-link planar robot manipulator. *IEEE Trans. Control Syst. Technol.* 21, 906–914. doi: 10.1109/TCST.2012.2190142

Xu, Z., Li, S., Zhou, X., Yan, W., Cheng, T., and Huang, D. (2019a). Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties. *Neurocomputing* 329, 255–266. doi: 10.1016/j.neucom.2018.11.001

Xu, Z., Li, S., Zhou, X., Zhou, S., and Cheng, T. (2020). Dynamic neural networks for motion-force control of redundant manipulators: an optimization perspective. *IEEE Trans. Indus. Electron.* doi: 10.1109/TIE.2020.2970635

Xu, Z., Zhou, X., and Li, S. (2019b). Deep recurrent neural networks based obstacle avoidance control for redundant manipulators. *Front. Neurorobot.* 13:47. doi: 10.3389/fnbot.2019.00047

Yue, S., Jia, Q., Gang, C., Wang, Y., and Sun, H. (2015). "Study of rapid collision detection algorithm for manipulator," in *IEEE Conference on Industrial Electronics and Applications (ICIEA)* (Auckland), 934–938.

Zhang, H., Wang, Y., Zheng, J., and Yu, J. (2018). Path planning of industrial robot based on improved RRT algorithm in complex environments. *IEEE Access* 6, 53296–53306. doi: 10.1109/ACCESS.2018.2871222

Zhang, Y., Li, S., Kadry, S., and Liao, B.(2019a). Recurrent neural network for kinematic control of redundant manipulators with periodic input disturbance and physical constraints. *IEEE Trans. Cybernet.* 49, 4194–4205. doi: 10.1109/TCYB.2018.2859751

Zhang, Y., Li, S., and Zhou, X. (2019b). Recurrent-neural-network-based velocity-level redundancy resolution for manipulators subject to a joint acceleration limit. *IEEE Trans. Indus. Electron.* 66, 3573–3582. doi: 10.1109/TIE.2018.2851960

Zhang, Y., Tan, Z., Ke, C., Zhi, Y., and Lv, X. (2009). Repetitive motion of redundant robots planned by three kinds of recurrent neural networks and illustrated with a four-link planar manipulator's straight-line example. *Robot. Auton. Syst.* 57, 645–651. doi: 10.1016/j.robot.2009.01.002

Zhang, Y., and Wang, J. (2004). Obstacle avoidance for kinematically redundant manipulators using a dual neural network. *IEEE Trans. Syst. Man Cybernet. B* 34, 752–759. doi: 10.1109/TSMCB.2003.811519

Zhang, Z., Xu, H., Chao, Z., Li, X., and Wang, C. (2015). A novel vehicle reversing speed control based on obstacle detection and sparse representation. *IEEE Trans. Intell. Transport. Syst.* 16, 1321–1334. doi: 10.1109/TITS.2014.2360337

Zhou, X., Xu, Z., and Li, S. (2019). Collision-free compliance control for redundant manipulators: an optimization case. *Front. Neurorobot.* 13:50. doi: 10.3389/fnbot.2019.00050

Zlajpah, L., and Nemec, B. (2002). "Kinematic control algorithms for on-line obstacle avoidance for redundant manipulators," in *IEEE/RSJ International Conference on Intelligent Robots & Systems* (Lausanne: IEEE), 1898–1903.

Conflict of Interest: WZ, XC, and XS are employed by the company Foshan Longshen Robotics LTD.

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: 30 October 2020
doi: 10.3389/fnbot.2020.590371

CNN Based Detectors on Planetary Environments: A Performance Evaluation

Federico Furlán, Elsa Rubio*, Humberto Sossa and Víctor Ponce*

Instituto Politécnico Nacional, Centro de Investigación en Computación, Ciudad de México, México

An essential characteristic that an exploration robot must possess is to be autonomous. This is necessary because it will usually do its task in remote or hard-to-reach places. One of the primary elements of a navigation system is the information that can be acquired by the sensors of the environment in which it will operate. For this reason, an algorithm based on convolutional neural networks is proposed for the detection of rocks in environments similar to Mars. The methodology proposed here is based on the use of a Single-Shot-Detector (SSD) network architecture, which has been modified to evaluate the performance. The main contribution of this study is to provide an alternative methodology to detect rocks in planetary images because most of the previous works only focus on classification problems and used handmade feature vectors.

Keywords: convolutional neural network (CNN), rock detection, machine learning, planetary exploration, remote sensing

OPEN ACCESS

Edited by:
Ganesh R. Naik,
Western Sydney University, Australia

Reviewed by:
Enrique Garcia-Trinidad,
Tecnológico de Estudios Superiores de Huixquilucan, Mexico
Genaro Ochoa,
Instituto Tecnológico Superior de Tierra Blanca, Mexico
Jesus Alberto Meda Campaña,
Escuela Superior de Ingeniería Mecánica y Eléctrica (IPN), Mexico

****Correspondence:***
Federico Furlán
ffurlan_b11@sagitario.cic.ipn.mx
Elsa Rubio
erubio@cic.ipn.mx

Received: *01 August 2020*
Accepted: *24 September 2020*
Published: *30 October 2020*

Citation:
Furlán F, Rubio E, Sossa H and Ponce V (2020) CNN Based Detectors on Planetary Environments: A Performance Evaluation. Front. Neurorobot. 14:590371. doi: 10.3389/fnbot.2020.590371

1. INTRODUCTION

Research interest in planetary missions centered on exploring on-site regions of Mars or the Moon is increasing. Remarkable examples of this are the next NASA mission to Mars with a new Rover generation (NASA, 2020) or the recent Chinese (Amos, 2020) and Arabian launches. Projects that have reached singular success are the exploration missions performed by geologist robots. Their main task is to retrieve samples that could give clues about the past of the terrain conditions of vital importance for future missions. A serious problem in these missions originates from data transmission latency, which is the time needed to send information from the robot location back to Earth, in contrast to a reduced time window for this assignment. Therefore, the robot must be able to detect objects of interest like rocks autonomously. A typical method used for object detection is through image processing. But conditions typically encountered in planetary environments, like arid terrains devoid of any kind of vegetation, as well as similar color and texture scenarios, results in poor performance of conventional image processing methods that usually are not adequate to different lighting conditions. This makes it necessary to experiment with models capable of handling information with uncertainty and effective in recognizing objects of interest with tolerance to the disturbances present in the captured images, such as Artificial Neural Networks (ANN).

In Gao et al. (2014) several approaches to detect objects in planetary terrains are introduced, suggesting that neural networks could provide promissory results. In many research works found in literature, ANN and recently, Convolutional Neural Networks (CNN), have demonstrated astonishing results in a diversity of problems related to object recognition, surpassing the performance of other approaches. Typical works deal meanly with images that focus on scenes taken from houses, offices, or cities. Other works are specialized in medical or biological images. However, the number of articles that employ CNN to process planetary terrain images or lands with characteristics alike is reduced.

Results from testing two CNN architectures, along with a Visual Geometry Group Neural Network (VGG) type and a Residual Neural Network (Resnet) for rock classification are reported in Li et al. (2020), where an approach called transfer learning is employed, which consists of using the trained weights of a model processed over a large amount of data as the initial weights of the CNN. The second training model named fine-tuning adjusts the CNN weights with a smaller dataset of the object of interest. They reported extraordinary results with an accuracy of 100%, by using a VGG16. Also, they compare the results with conventional methodologies, like Histogram Oriented Gradients (HOG) or Scale-Invariant Feature Transform (SIFT), plus a Support Vector Machine (SVM) that reaches a humble accuracy of around 63% and 75%. They used, as the dataset, images captured from the Curiosity mission, Nevertheless, images are trim and show only a rock.

In Furlán et al. (2019), a methodology to detect rocks using a CNN is presented, where a U-net, which is a convolutional neural network introduced in Ronneberger et al. (2015), was adapted to segment panoramic images taken in a Mars-like environment located on Earth. An F1-score of 78% while improving the inference latency of the algorithm is reported. The results were satisfactory and similar to other methodologies.

This work is aimed to evaluate the performance of some CNN's models for rock detection tasks, in a Mars-like environment, demonstrating that a CNN can be an alternative to conventional image detection techniques, due to their inherent advantage for handling the uncertainty found typically in unexplored terrains, paving the way for ambitious exploration traversals. Indeed, a combination of CNNs with neuromorphic computing, based on memristor technologies are gaining attention as future intelligent computing platforms for image detection due to their ultra-low power consumption and implementation on integrated circuits (Amravati et al., 2018) and (Chen et al., 2019). A combination of CMOS-camera with a neuromorphic chip, running CNN based algorithms for image recognition is expected to become the next step for planetary Rover missions.

2. MATERIALS AND METHODS

Recent advances in object detection that use CNN models have achieved successful results with different datasets, like COCO (Lin et al., 2014) or Pascal VOC (Everingham et al., 2010). COCO and Pascal VOC are datasets consisting of images taken in different scenarios, focused on detecting objects like cars, people, cats, dogs, among other daily life objects. Due to those promising results, we considered testing the performance of such CNN architectures with unstructured objects typical in outdoor environments.

We are interested in experimenting with the CNN architectures in a Mars-like environment where the main task is to detect rocks. The methodology proposed uses a Single-shot Multibox Detector (SSD) to detect rocks, which are objects of interest in an exploration mission.

2.1. Single-shot Multibox Detector

The Single-shot Multibox Detector was introduced in 2016 (Liu et al., 2016). The architecture is formed by three parts, a backbone followed by a series of convolutional feature extraction layers and the detection layers. It is required to apply a non-maxima suppression process to obtain the correct output, which are the corresponding predicted boxes in the image, see **Figure 1**.

In the original paper, the backbone corresponds to a truncated VGG-16 network that works as a feature extraction phase. The extra feature extraction layers decrease gradually in size to make predictions on different scales. The detection layers align the top feature layers with bounding boxes that have multiple predefined scales and ratios. The predictions obtain for each bounding box related to the feature layer are the offset of the position of the bounding box and a confidence value that means whether a class is present in the region or not. An advantage of this architecture is that it makes predictions at multiple scales, which improves the detection in comparison with other models like Faste R-CNN.

The function of the extra feature extraction layers is to generate default bounding boxes using convolution filters. For each feature layer, a small kernel operates to obtain a membership value for each class or an offset measured relative to the default bounding box position, depending on the convolutional layer.

A set of default bounding boxes is associated with each feature layer where the predictions will be made. So, each bounding box will produce c score values, where c is the number of classes to be detected, and 4 location offsets relative to the initial box position. For example, an $m \times n$ feature map produces a total of $(c + 4)k$ filters that are applied around each location, where k is the number of boxes, generating $(c + 4)kmn$ outputs.

The loss function is a weighted sum of two functions.

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \tag{1}$$

The localization loss function ($L_{loc}(x, l, g)$) estimates the closeness between the predicted box (l) and the ground truth box (g). It measures the difference in the center location (cx, cy) and the width (w) and height (h) of the predicted box relative to the ground truth box. It uses a Smooth L1 norm and α is a weight term.

The confidence loss function ($L_{conf}(x, c)$) compares the predicted classes with the ground truth classes for each bounding box. It uses a softmax function. For a detailed explanation of the loss functions consult (Liu et al., 2016).

2.2. Dataset

The images used with the CNN model are from (Furgale et al., 2012) created by the Autonomous Space Robotics Lab (ASRL) from the University of Toronto. The original dataset is a compilation of more than 50,000 images captured during a 10-kilometer traverse in the Mars analog site on Devon Island located in Canada. The dataset is not labeled. Hence to avoid the

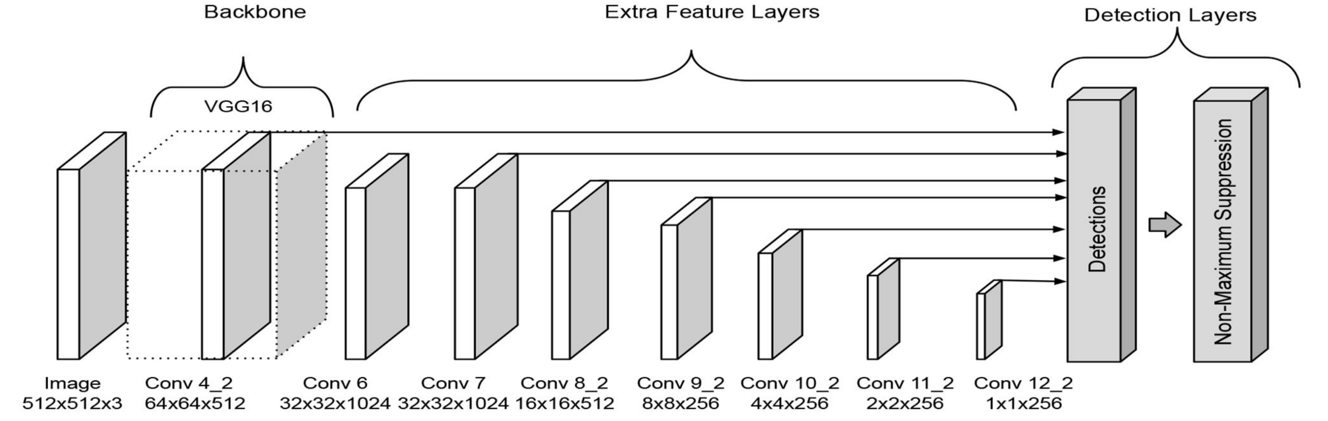


FIGURE 1 | SSD architecture as presented in Liu et al. (2016).

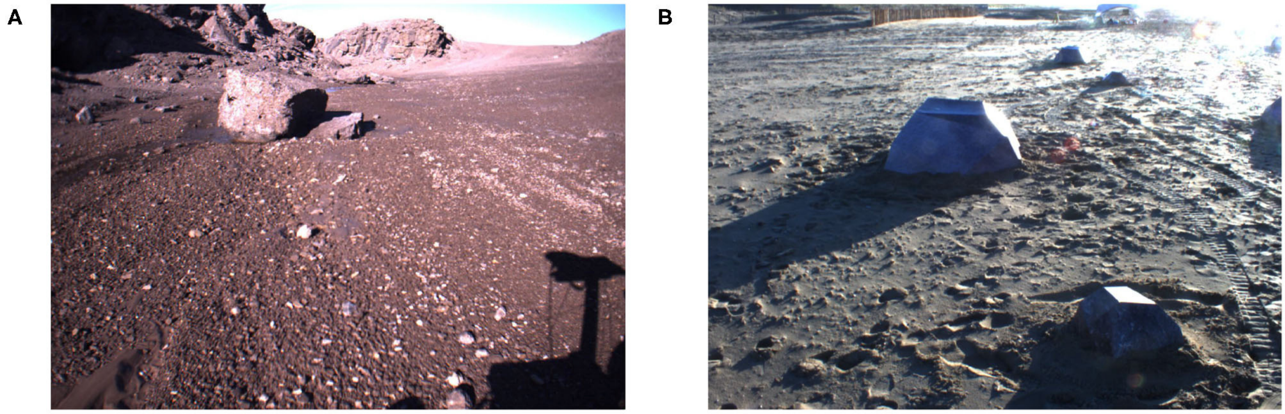


FIGURE 2 | Sample images form dataset **(A)** Devon Island (Furgale et al., 2012) and **(B)** Katwijk beach (Hewitt et al., 2018).

laborious task of manually mark every image, we separated an image each five frames ending with a dataset of 5172 images.

During the labeling process, we discarded images that didn't display rocks. In the end, the final dataset has 1,600 labeled images that include a total of 8,372 objects labeled as rocks. Then, we divided the dataset into 1,280 images for training and 320 for validation. To examine the performance of the CNN model, we selected a different dataset for testing.

We used The Katwijk beach planetary rover dataset (Hewitt et al., 2018) that uses artificial models of rocks of different sizes and distribute them around a beach to resemble a planetary terrain. We manually labeled 331 images to estimate the generalization ability of the models. In **Figure 2**, we show an image from each dataset.

2.3. Proposed CNN

We introduce two modified versions of the original SSD architecture presented in Liu et al. (2016). We resized the dataset images to 512 × 512, which is the input size of the models. ReLu is the activation function used in all convolution operations.

The scales are parameters required in the detection layers, which are obtained using the next equation:

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m-1}(i-1) \tag{2}$$

where $s_{min} = 0.1$ and $s_{max} = 1.06$, m is the number of predictions layers for all models. In this work, $m = 7$ is considered, the first scale is set as 0.04, and i is the number of scales needed in the model. The scales used in all models are $[0.04, 0.1, 0.26, 0.42, 0.58, 0.74, 0.9, 1.06]$.

The models proposed makes predictions over 7 layers, the aspect ratios used for all models are the same as in the original paper (Liu et al., 2016). The aspect ratios for prediction layers $1, 6$ and 7 are $[1, 2, \frac{1}{2}]$ and for prediction layers $2, 3, 4$ and 5 are $[1, 2, \frac{1}{2}, 3, \frac{1}{3}]$.

The first introduced model is a modified version of the original SSD architecture that reduces the number of filters in the VGG16 backbone in half. This backbone has 13 convolutional layers with 3 × 3 kernels. The input size is reduced from 512 to 32 due to 5 max-pooling operations. A detailed diagram of the Backbone

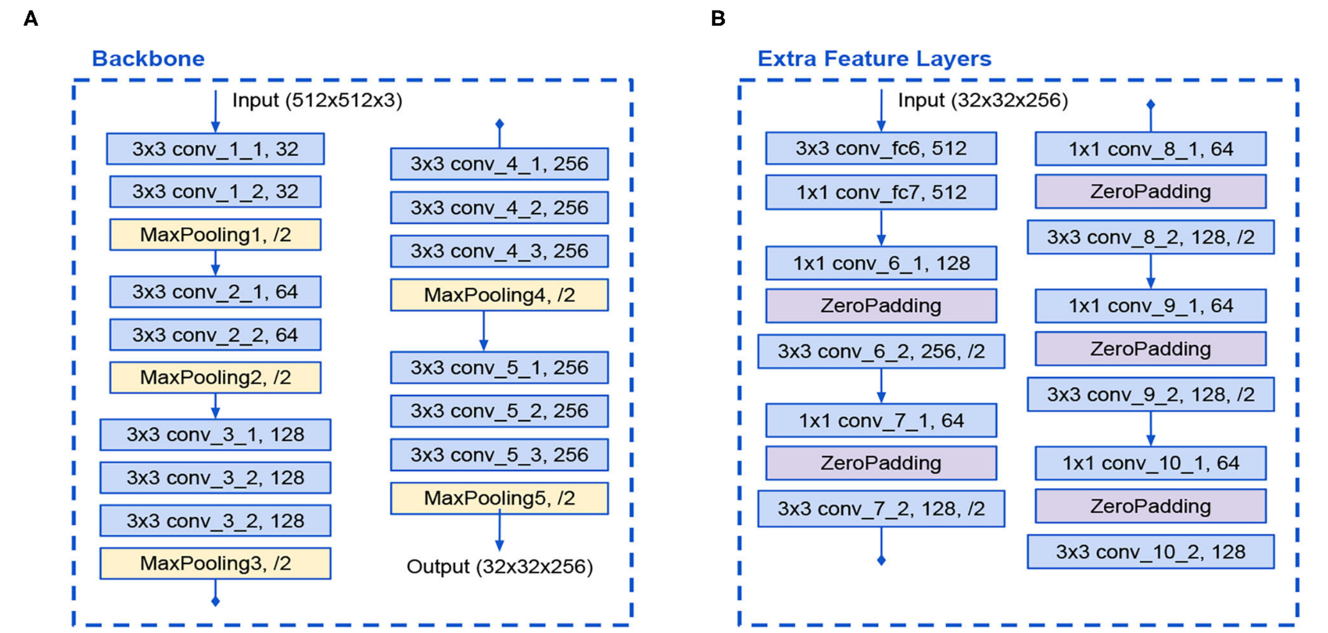


FIGURE 3 | (A) Modified VGG16 backbone and **(B)** Modified Extra Feature Layers in SSD A architecture.

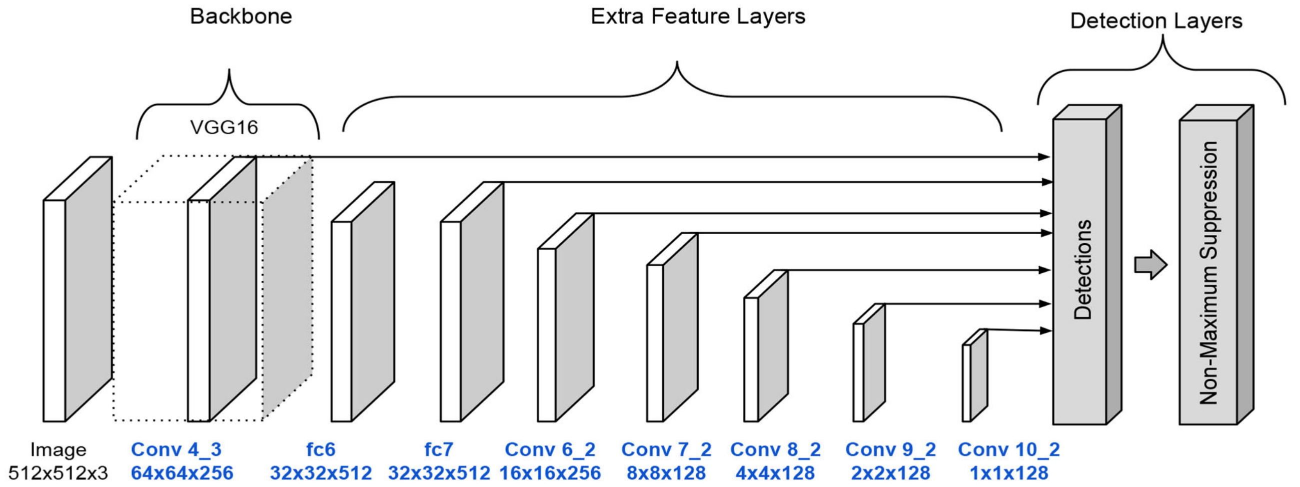


FIGURE 4 | SSD architecture version A.

is presented in **Figure 3**. Additionally, the Extra Feature Layers also reduced its filters in half and is formed by 12 convolutional layers with 1 × 1 and 3 × 3 kernels with strides of 2 that caused feature maps dimension reduction. A detailed diagram of the Extra Feature Layers is presented in **Figure 3**.

These modifications lessen the number of parameters. The full SSD A architecture is shown in **Figure 4**.

Previous works like (He et al., 2017) used ResNet configurations as a backbone to improve the performance for detections and instance segmentation tasks but require big datasets for training since its large number of trainable parameters. In the second model, the VGG16 net is replaced with a convolutional network inspire in ResNet50. The new backbone uses two types of building blocks known as identity block and convolutional block. Their unique property is the shortcut connection, which consists of an add operation between an early convolution and the final convolution. A detailed diagram of these blocks is shown in **Figure 5**. The identity block has 3 convolutional layers and the convolutional block has 4 layers.

The backbone architecture is similar to the ResNet50, but we use the same number of filters in all convolutions in each building

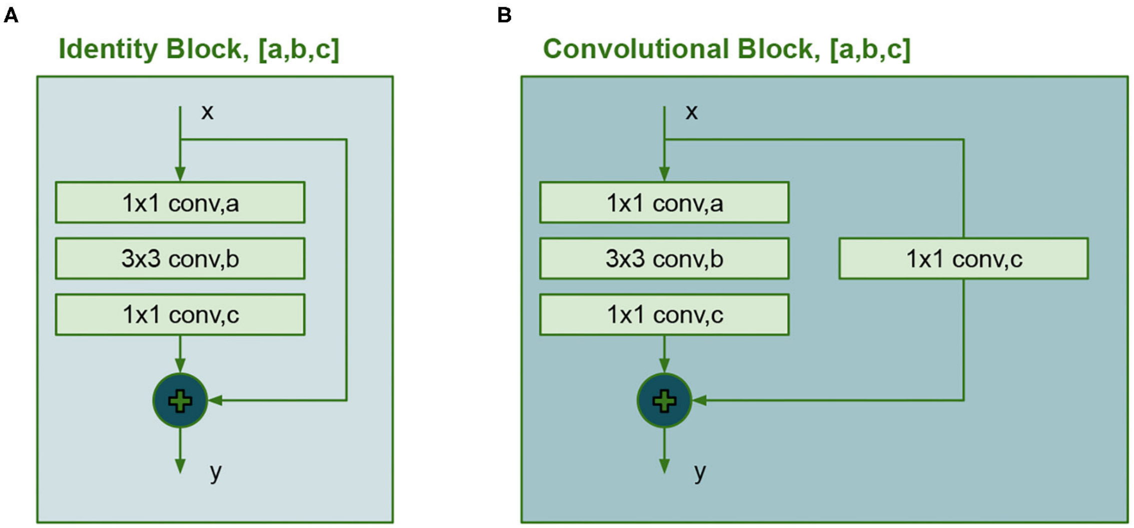


FIGURE 5 | Building blocks for the ResNet50, **(A)** Identity block and **(B)** Convolutional block.

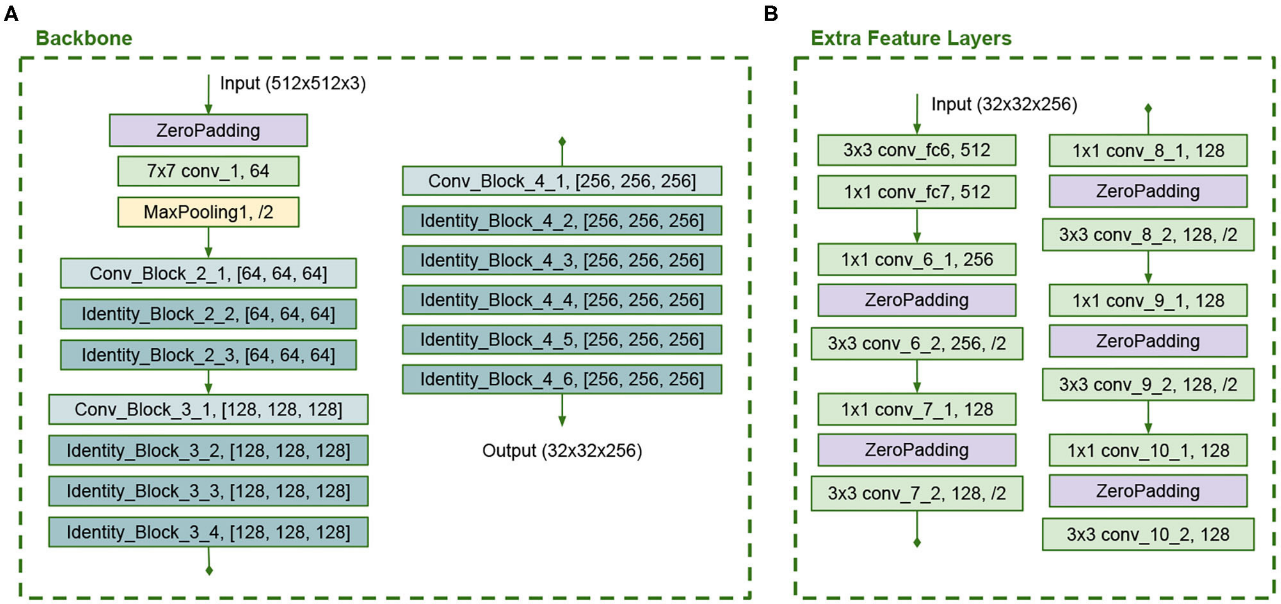

A
Backbone
Input (512x512x3)
ZeroPadding
7x7 conv_1, 64
MaxPooling1, /2
Conv_Block_2_1, [64, 64, 64]
Identity_Block_2_2, [64, 64, 64]
Identity_Block_2_3, [64, 64, 64]
Conv_Block_3_1, [128, 128, 128]
Identity_Block_3_2, [128, 128, 128]
Identity_Block_3_3, [128, 128, 128]
Identity_Block_3_4, [128, 128, 128]
Conv_Block_4_1, [256, 256, 256]
Identity_Block_4_2, [256, 256, 256]
Identity_Block_4_3, [256, 256, 256]
Identity_Block_4_4, [256, 256, 256]
Identity_Block_4_5, [256, 256, 256]
Identity_Block_4_6, [256, 256, 256]
Output (32x32x256)
B
Extra Feature Layers
Input (32x32x256)
3x3 conv_fc6, 512
1x1 conv_fc7, 512
1x1 conv_6_1, 256
ZeroPadding
3x3 conv_6_2, 256, /2
1x1 conv_7_1, 128
ZeroPadding
3x3 conv_7_2, 128, /2
1x1 conv_8_1, 128
ZeroPadding
3x3 conv_8_2, 128, /2
1x1 conv_9_1, 128
ZeroPadding
3x3 conv_9_2, 128, /2
1x1 conv_10_1, 128
ZeroPadding
3x3 conv_10_2, 128

FIGURE 6 | (A) Modified ResNet50 backbone and **(B)** Modified Extra Feature Layers in SSD B architecture.

block and truncate it at stage 4. In the original ResNet50 model, the last convolution has more filters than the other convolutions in each block. **Figure 6** presents the backbone configuration. This backbone has 43 layers. The Extra Feature Layers are configured as in **Figure 6** and have 12 layers. These modifications in the model reduce the number of trainable parameters. The full SSD B architecture is shown in **Figure 7**.

An original SSD configuration serves as a baseline for comparing performance with the introduced models. The original model is named SSD O in tables and graphics and shares the same configuration as the SSD A model. The only difference is the number of filters in the convolutional operations.

The code from (Ferrari, 2018), which is a Keras implementation of the original SSD architecture, was modified to run with Tensorflow 2.2. The generator function was transformed to read CSV files from the label datasets. The corresponding architectures presented in this article were developed as functions for the training process. Each training process took about 18 h of time execution, using an Intel i9 computer equipped with 64 Gb of RAM and two GPU cards installed, to complete the job with a learning rate of 0.001.

We utilized stochastic gradient descent (SGD) during 500 epochs to adjust the parameters during the training process. We used data augmentation to change the images with one of four transformations, which could be photometric distortion, expansion, random crop, or random horizontal flip. The intention of using data augmentation is to evade overfitting while training the models. The training process of a model requires only one execution to generate a weights file, that later will be loaded in the model to implement the inference task. Each execution will produce similar results, but not the same since the weights are randomly initialized using a He normal distribution. The resulting learning curves are shown in **Figure 8**.

3. RESULTS

Table 1 shows a comparison of the number of parameters within the architectures. The number of parameters is associated with the complexity of the net and the inference time. The inference time denotes how long does the CNN take

TABLE 1 | Comparison of the number of parameters and inference time.

Model	Number of parameters	Inference time (milliseconds)	FPS
SSD Original - VGG16	24,088,664	55.36	18
SSD A - VGG16	6,320,632	38.70	25
SSD B - ResNet50	10,088,664	39.01	25

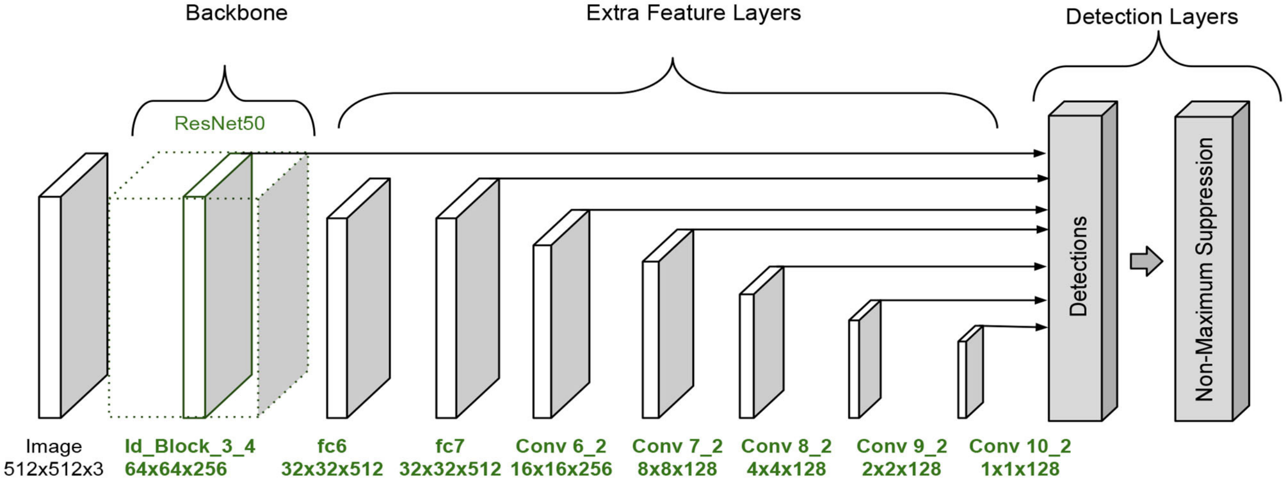


FIGURE 7 | SSD architecture version B.

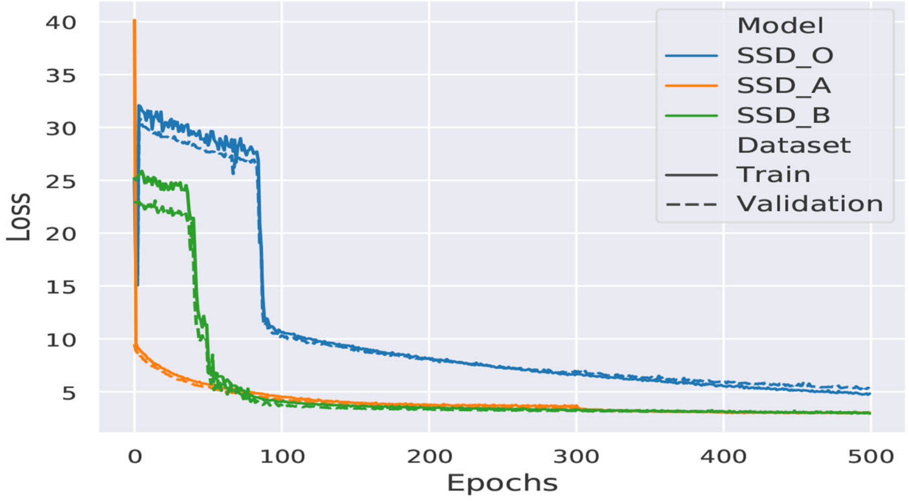


FIGURE 8 | SSD architectures training curves.

to produce a prediction. A remarkable characteristic of the SSD architecture is that it delivers what can be considered real-time performance. **Table 1** shows the average inference times for each model for each model running over the training computer.

The mean average precision per dataset (Train, Validation, and Testing) is listed in **Table 2**. This value represents how many target objects are predicted or detected by a CNN. The higher the number obtained, the network performance is better. This value is bounded to the [0,1] interval.

Additionally, the graphs of the mAP for each model and dataset are shown in **Figure 9**.

The original architecture shows signs of overfitting, caused by a large number of trainable parameters, more than twice the number of parameters of the proposed models. Another factor that contributes to the overfitting is the reduced amount of images of the dataset. Since this model posses a large number of parameters, it shows an undesired behavior conducting to memorize the training data, which results in a high mAP for training and validation but a significant drop for the testing dataset.

The results showed that there is plenty of room for improvement. Model A achieved better results for training and validation, while model B scored better in testing. Hence to determine which model is better, we need to remember that most of the planetary applications are focus on exploring unknown environments to find valuable scientific information.

Therefore we need a model capable of generalizing, which means, be capable of achieving high-grade performance with unknown data slightly different from the training data. Model B has a lower standard deviation among its mAP over all datasets.

We show some testing images with their corresponding predictions and ground truths in **Figures 10**, **11**. The predictions made by the network are depicted with a red square along with its confidence value, which means the grade of accuracy that the boxed object is a rock. Lastly, the ground truth is labeled with a green square.

TABLE 2 | Comparison of the mean average precision.

	mAP			
Model	**Train**	**Validation**	**Testing**	**Standard deviation**
SSD Original -VGG16	0.815	0.604	0.233	29.46%
SSD A -VGG16	0.627	0.520	0.174	23.68%
SSD B - ResNet50	0.451	0.353	0.253	9.90%

4. DISCUSSION

Previous methodologies employed to detect rocks in planetary environments require algorithms that need handmade feature vectors, which are complicated to design and are dependant on expert knowledge and the feature extractors applied that sometimes are not robust. This study evaluates an alternative solution adopting a supervised learning algorithm to avoid selecting feature extractors. Since CNN's are tolerant of translation transformations, and also trained appropriately permit small rotations or scale transformations, it adds a factor of robustness. It could become part of an autonomous navigation system because rocks are the main obstacles for rovers traversals, and with the same algorithm fulfill two functions detecting valuable samples and obstacles.

This methodology can improve while operating in an unknown environment by collecting new images and adding them to the training dataset. The training process can be performed remotely in a high-performance computer and then transmit the weights file to be updated on the operation site. The expected result would be an enhanced performance caused

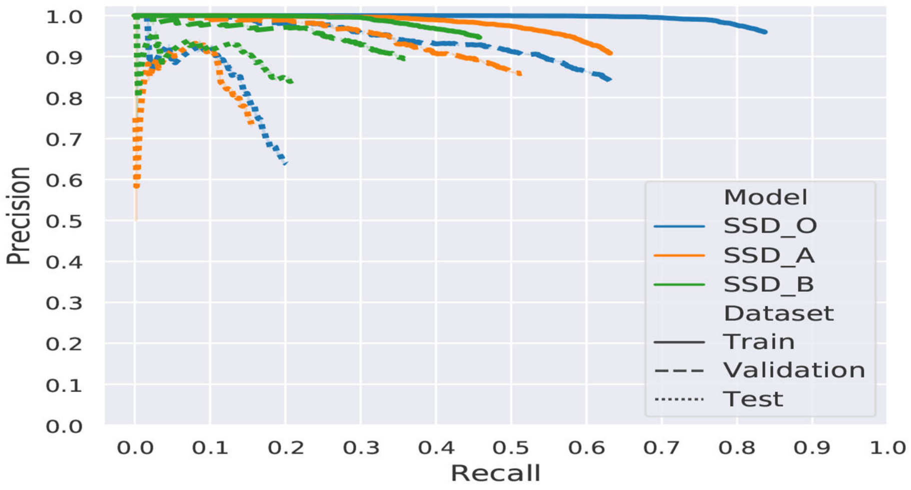


FIGURE 9 | Comparative graphic of the mAP.

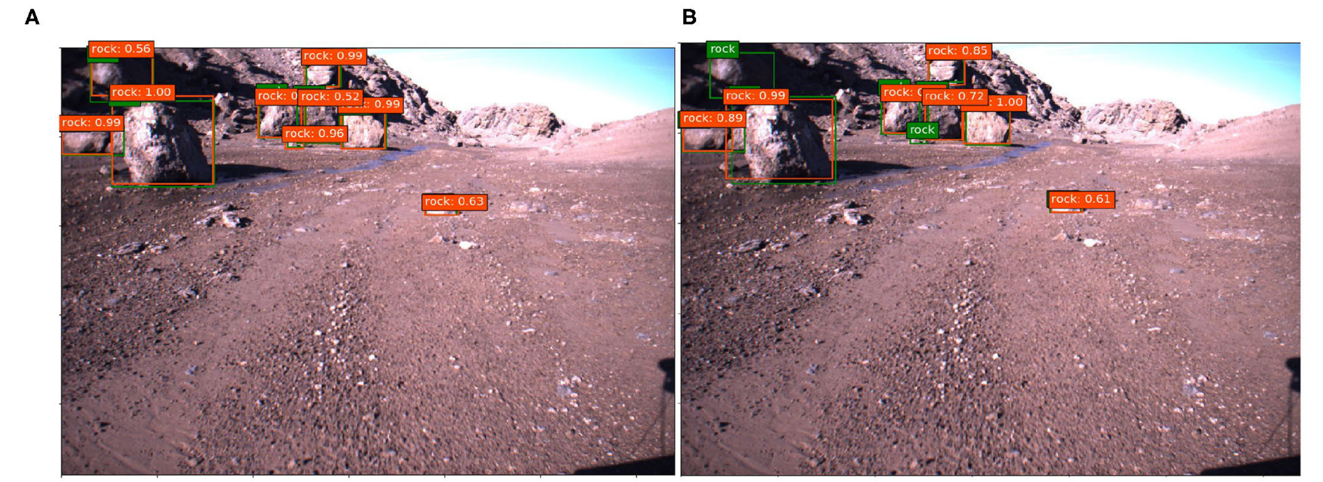


FIGURE 10 | Examples of predictions with Devon Island dataset, **(A)** SSD A and **(B)** SSD B.

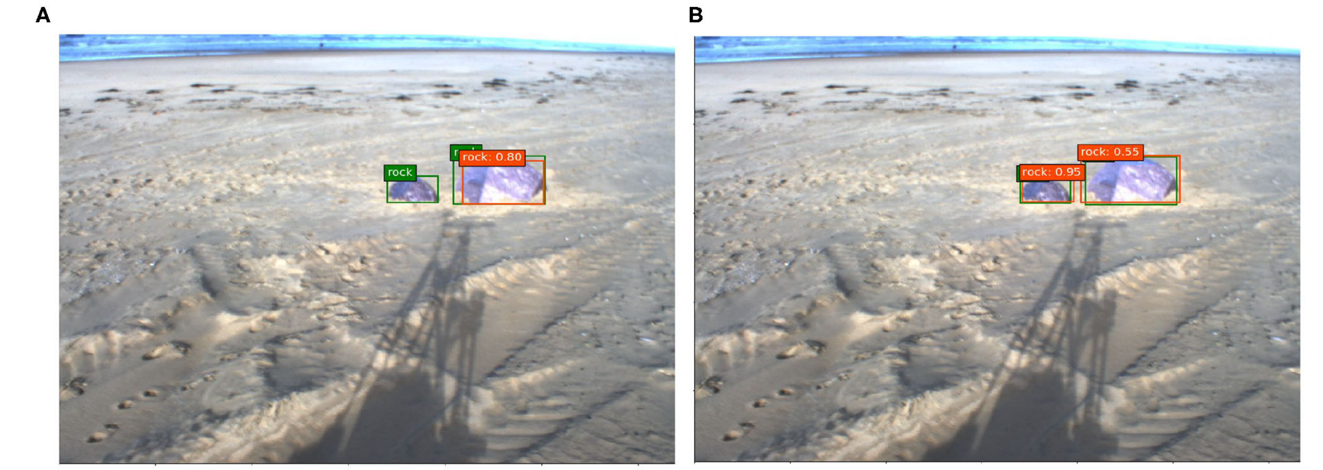


FIGURE 11 | Examples of predictions with Katwijk beach dataset, **(A)** SSD A and **(B)** SSD B.

by the new knowledge acquired from the unexplored area. Space exploration missions use remote sensing equipment to broadcast information to a control center. Hence this methodology would be suitable for object detection process.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

All authors contributed to the conception and design of the study. FF selected the datasets. HS and ER proposed modifications to the CNN architectures. VP wrote the first draft of the manuscript. FF developed the code required for the experiments. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

The authors would like to thank the economic support by the projects SIP 20180943, 20190007, 20195835, 20200630, 20200569, and 20201397, COFAA and CONACYT-México under project 65 within the framework of call: Frontiers of Science 2015.

ACKNOWLEDGMENTS

The authors would like to thank the support provided by Instituto Politécnico Nacional (IPN), Secretaría de Investigación y Posgrado (SIP-IPN), Comisión de Operación y Fomento de Actividades Académicas (COFAA-IPN) and CONACYT-México for the support to carry out this research.

REFERENCES

Amos, J. (2020). *China's Tianwen-1 Mars rover rockets away from Earth*. Available online at: https://www.bbc.com/news/science-environment-53504797

Amravati, A., Nasir, S. B., Thangadurai, S., Yoon, I., and Raychowdhury, A. (2018). "A 55nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots," in *2018 IEEE International Solid - State Circuits Conference* (San Francisco, CA: ISSCC), 124–126. doi: 10.1109/ISSCC.2018.8310215

Chen, G., Bing, Z., Rohrbein, F., Conradt, J., Huang, K., Cheng, L., et al. (2019). Toward brain-inspired learning with the neuromorphic snake-like robot and the neurorobotic platform. *IEEE Trans. Cogn. Dev. Syst*. 11, 1–12. doi: 10.1109/TCDS.2017.2712712

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis*. 88, 303–338. doi: 10.1007/s11263-009-0275-4

Ferrari, P. (2018). SSD: *Single-Shot MultiBox Detector implementation in Keras*. Available online at: https://github.com/pierluigiferrari/ssd_keras

Furgale, P., Carle, P., Enright, J., and Barfoot, T. D. (2012). The Devon island rover navigation dataset. *Int. J. Robot. Res*. 31, 707–713. doi: 10.1177/0278364911433135

Furlán, F., Rubio, E., Sossa, H., and Ponce, V. (2019). "Rock detection in a mars-like environment using a CNN," in *Pattern Recognition*, eds J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, J. A. Olvera-López, and J. Salas (Cham: Springer International Publishing), 149–158. doi: 10.1007/978-3-030-21077-9_14

Gao, Y., Spiteri, C., Pham, M.-T., and Al-Milli, S. (2014). A survey on recent object detection techniques useful for monocular vision-based planetary terrain classification. *Robot. Auton. Syst*. 62, 151–167. doi: 10.1016/j.robot.2013.11.003

He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)* (Venice), 2980–2988. doi: 10.1109/ICCV.2017.322

Hewitt, R. A., Boukas, E., Azkarate, M., Pagnamenta, M., Marshall, J. A., Gasteratos, A., et al. (2018). The Katwijk beach planetary rover dataset. *Int. J. Robot. Res*. 37, 3–12. doi: 10.1177/0278364917737153

Li, J., Zhang, L., Wu, Z., Ling, Z., Cao, X., Guo, K., et al. (2020). Autonomous Martian rock image classification based on transfer deep learning methods. *Earth Sci. Inform*. 13, 951–963. doi: 10.1007/s12145-019-00433-9

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). "Microsoft COCO: common objects in context," in *Computer Vision-ECCV 2014*, eds D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars (Cham: Springer International Publishing), 740–755. doi: 10.1007/978-3-319-10602-1_48

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). "SSD: single shot multibox detector," in *Computer Vision-ECCV 2016*, eds B. Leibe, J. Matas, N. Sebe, and M. Welling (Cham: Springer International Publishing), 21–37. doi: 10.1007/978-3-319-46448-0_2

NASA (2020). *Mars 2020 Mission Perseverance Rover*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-Net: convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015*, eds N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi (Cham: Springer International Publishing), 234–241. doi: 10.1007/978-3-319-24574-4_28

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.

The reviewer JAMC declared a shared affiliation, though no other collaboration, with the authors to the handling Editor.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 01 December 2020
doi: 10.3389/fnbot.2020.559366

Quantum-Based Creative Generation Method for a Dancing Robot

Peng Mei[1], *GangYi Ding*[1], *QianKun Jin*[1], *FuQuan Zhang*[2]* *and YangFan Jiao*[3]

[1] Digital Performance and Simulation Technology, School of Computer Science & Technology, Beijing Institute of Technology, Beijing, China, [2] Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou, China, [3] Beijing Wanshide Technology Co., Ltd., Beijing, China

In this paper, we propose a creative generation process model based on the quantum modeling simulation method. This model is mainly aimed at generating the running trajectory of a dancing robot and the execution plan of the dancing action. First, we used digital twin technology to establish data mapping between the robot and the computer simulation environment to realize intelligent controllability of the robot's trajectory and the dance movements described in this paper. Second, we conducted many experiments and carried out a lot of research into information retrieval, information fidelity, and result evaluation. We constructed a multilevel three-dimensional spatial quantum knowledge map (M-3DQKG) based on the coherence and entangled states of quantum modeling and simulation. Combined with dance videos, we used regions with convolutional neural networks (R-CNNs) to extract character bones and movement features to form a movement library. We used M-3DQKG to quickly retrieve information from the knowledge base, action library, and database, and then the system generated action models through a holistically nested edge detection (HED) network. The system then rendered scenes that matched the actions through generative adversarial networks (GANs). Finally, the scene and dance movements were integrated, and the creative generation process was completed. This paper also proposes the creativity generation coefficient as a means of evaluating the results of the creative process, combined with artificial brain electroenchalographic data to assist in evaluating the degree of agreement between creativity and needs. This paper aims to realize the automation and intelligence of the creative generation process and improve the creative generation effect and usability of dance movements. Experiments show that this paper has significantly improved the efficiency of knowledge retrieval and the accuracy of knowledge acquisition, and can generate unique and practical dance moves. The robot's trajectory is novel and changeable, and can meet the needs of dance performances in different scenes. The creative generation process of dancing robots combined with deep learning and quantum technology is a required field for future development, and could provide a considerable boost to the progress of human society.

Keywords: creative generation, quantum simulation, information fidelity, M-3DQKG, QGAN, robot trajectory

OPEN ACCESS

Edited by:
Mu-Yen Chen,
National Taichung University of Science and Technology, Taiwan

Reviewed by:
Baran Cürüklü,
Mälardalen University College, Sweden
Jun-Hong Shen,
Asia University, Taiwan

****Correspondence:***
FuQuan Zhang
8528750@qq.com

Received: *06 May 2020*
Accepted: *11 September 2020*
Published: *01 December 2020*

Citation:
Mei P, Ding G, Jin Q, Zhang F and Jiao Y (2020) Quantum-Based Creative Generation Method for a Dancing Robot.
Front. Neurorobot. 14:559366.
doi: 10.3389/fnbot.2020.559366

INTRODUCTION

Robot trajectory calculations have always been an essential subject of scientific research. Nevertheless, directly programming a robot takes much debugging time and development costs. In this paper, computer simulation technology and a quantum modeling method are used to generate dance movement creativity that meets the performance and calculation of the robot trajectory data. The data are simulated and rehearsed by a digital twin to realize the robot's dance performance. Simulation technology has been widely used since the 20th century (Zeigler et al., 2000). With the maturity of computer technology, simulation technology has been further popularized through simulation software. Now, simulation technology has been widely used in stage performance (Bilbao, 2009; Niedenthal et al., 2010; Jaber et al., 2019), event scheduling (Colella, 2011; Arima, 2015), emergency response (Merién et al., 2010), military training (Liang et al., 2001; Machado et al., 2015; Ma et al., 2016), aerospace (Zipfel and Schiehlen, 2001; Jha et al., 2015), industrial manufacturing (Buyya and Murshed, 2010; Yamaguchi et al., 2016; Santipanusopon and Worawattanaparinya, 2019; Taheripour et al., 2019), technology research and development (Boyd and Bruns, 2001; Rapaport et al., 2002; IEEE, 2010; Binder and Heermann, 2014; Moin, 2018; Yingying et al., 2018), and many other fields. The simulation process has many commonalities, and these commonalities can be found in the application of simulation technology in various fields (Bucklew, 2010). As one of the tools to assist us in completing the planning, it has an essential and irreplaceable role in insignificant events. For example, the US military simulation system (Zhou et al., 2015) to the global strategic military defense system evaluation (Burger and Jenkins, 2013). Simulation is used to construct mathematical and physical models through computer modeling tools (Kasztenny and Kezunovic, 2002). Its purpose is to help people discover the problems that may be encountered during the development of an event (Biolek et al., 2019). Therefore, this paper attempts to use simulation technology to solve the problem of dance creativity for a dancing robot in order to assist people in the creative generation process to make decisions that meet their needs. Simulation technology can realize data interconnections between the real world and the virtual world by establishing a digital twin relationship with the robot. This paper uses simulation technology to generate dance robot data, including body movement data on points and spatial trajectory movement data. This paper establishes the usefulness of the multilevel three-dimensional spatial quantum knowledge map (M-3DQKG) knowledge retrieval model by determining the domain of the requirement ontology: to retrieve creative points according to creative needs to generate a creative pool. Using the creative pool as the original data, M-3DQKG is updated through quantum generative adversarial networks (QGANs) learning. The retrieval weight of this model is allocated according to the probability branch model. The knowledge retrieval process is performed according to weight. The generated ideas are evaluated by means of the subjects' electroencephalographic (EEG) data and creative generation coefficient (CGC) calculation results. The whole process described in this paper simulates the creative generation process, which provides a quick and effective method of enhancing the richness of creativity. Migrating the simulation data to the dancing robot can operate the robot's dance performance and trajectory movement.

Section Related Work is an introduction to related work. Section Method mainly introduces the critical technologies of dance creative movement generation, including the M-3DQKG model, probability branch model, creative generation model, information fidelity calculation method, and system creative generation ability evaluation method. Section Results presents an analysis and discussion of the experimental results. The last section is the conclusion and future prospects.

RELATED WORK

This section will introduce related work and the key technologies involved in this field. As a relatively new research direction, creativity generation is not widely understood by many people. This section first introduces the concepts of originality, creativity, and creativity generation, as well as the cognition of these concepts by related researchers. Next, this section will introduce the essential means of realizing creative generation–computer simulation technology. This technology can complete the creative process in a virtual environment and transfer the simulated data to the dancing robot to complete dance performances and trajectory movements. In the simulation process, the three links of knowledge map construction, machine learning network, and result evaluation are all integrated into quantum modeling methods. This accelerates the creative generation process from a new perspective and enriches the creative results.

Originality, Creativity, and Creative Generation

Originality is new original ideas. Creativity is the ability to generate originality. And creative generation is a flow of operations that span time to create originality.

Originality is a new abstract thinking or action based on people's cognition of things. It cannot get rid of cognition and exists alone, and human perceptions are very different. Therefore, there is no unified rule for the definition of creativity or for the evaluation criteria of creativity. At this stage, although the creative process is complicated, and the tasks are numerous, people's demand for it is becoming greater. Not only that, but with the development of society in the future, this demand will continue to increase. This paper starts with meeting people's creative needs, breaking the single boundary of people's cognition, and proposes a method that can be quantitatively evaluated.

Originality arises as a new thing, but it is in fact a recombination of old elements. People's insight into the relationship between elements is the basis for generating new combinations. The creative generation process of the brain is the perfect cooperation between knowledge and neurons. Simulating the creative generation process of the human brain requires two conditions. Pezzulo et al. (2013) proposed that one is widely generated and frequently changed, and the other is summarized and filtered from it. The focus of the former is to continuously acquire knowledge, analyze knowledge, use trial and error, and correct. The focus of the latter is to emphasize the method of selection. In 1960 Campbell proposed the theory that creativity requires blind change and selective retention, exploring the generation of creativity (Simonton, 2012). These ideas have become the key technical points for computers to complete the creative generation process.

The process of creative generation is differently affected by a person's personality (Shamay-Tsoory et al., 2011), intelligence level (Kim, 2016), educational situation (Calignano and Jsendal, 2018), and creative environment (Saorín et al., 2017). One study (Steele et al., 2018) proposed neural and cognitive models to balance the influence of cognition on creativity. This paper explores the automation and intelligence of the creative generation process so that the creativity of the system is maintained at a stable level, and the results are free from the influence of individual differences. Therefore, a dance movement creative generation model is proposed to complete the creative generation process without human intervention and environmental influence. Studies have found that the internal role of emotion is the basis for the association between people's creativity and intuitive thinking (Haijuan et al., 2018). Positive emotions can enhance people's original ideas (Rooij et al., 2017) and make people work hard on simple tasks that are fun and stupid. Negative emotions make people work harder on serious and essential complex tasks (Friedman et al., 2007). Therefore, people use a creative process with emotion as the dominant position (Agnoli and Corazza, 2019). This paper simulates the emergence of emotion-led creativity based on the random nature of quantum modeling. The purpose is to increase the novelty of originality. The improvement of this technology has become one of the motivations for computers to complete the design of creative generative models.

This paper takes creative needs as the starting point and realizes the unity of opposites between creative thinking and critical thinking (Diyanni, 2014; Rivas, 2017): jump out of the shackles of creativity and standardization (Oliver et al., 2019) to give freedom to people's creative ability (Robinson, 2011; Jonason et al., 2015). The computer has become a capable means of giving full freedom to creativity with its plasticity, stability, upgradeability, and supercomputing power. Knowledge retrieval and learning networks break the upper limit of human creativity so that creativity is not affected by individual education, intelligence, experience, environment, needs, and other factors. The movement data generated by dance creativity take the virtual human in the simulation system as the carrier and are digitally twinned to the physical robot to drive the robot's limb movement and trajectory calculations to enhance the expressive power of the dancing robot.

Computer Simulation Technology

Computer simulation technology emerged in the 1950's and was transformed from analog to digital simulation in the 1960's (Gould and Tobochnik, 2007). The process of simulation transforms the uncertainty of an event into a representation of the overall level of the event through modeling (Roux et al., 1973). Its purpose is to clarify the development trend of things and eliminate the vague understanding of the connections between things (Mustafee et al., 2018). Simulation is used to simulate and map real events. The modeling method is the primary tool for computer simulation to enhance people's understanding of things (Rozenblit, 2015).

Based on the advantages of computer simulation, researchers have made many attempts in the field of creative generation. In their research, Li proposed combining original thinking models with intuitive thinking methods such as the Theory of Inventive Problem Solving (TRIZ) and brainstorming (Li et al., 2003), through computer-aided product innovation. Computer simulation technology uses a neural network to process voice (Hinton et al., 2012; Graves et al., 2013; Muckenhirn et al., 2017), image (Sjöström et al., 2015; Lin et al., 2016; Hu et al., 2018; Sana et al., 2018), and video (Matta, 2008; Han et al., 2017) data and extracts the characteristics of the simulated object. The simulation system performs three-dimensional modeling based on the characteristic data. The application of GANs in the field of computer simulation has also accelerated the process of data processing and learning. Although different GANs methods are based on various convolutional neural network (CNN) frameworks (Wang et al., 2019), experiments by Goodfellow et al. have proved the potential of this framework through qualitative and quantitative evaluation of the generated samples (Goodfellow et al., 2014). Similarly, the apparent advantage of computer simulation technology improves the credibility of evaluating simulation results. Zheng et al. (2017) made some attempts in their paper, assigning recognition results into five categories and generating results through six-category detection. Recently, the University of California, Berkeley, has made significant progress (Xu et al., 2020). They proved the advantages of GANs evaluation results in the simulation system. In addition, an essential method of computer simulation technology is the modeling method. Moore and Cain (William and Ralph, 2015) and others pointed out the vital role of mathematical ability and logical methods in the process of computer simulation creative generation. Zhang et al. (2018) proposed a computable model of creative generation, which provides a theoretical basis for the digitization of the creative generation process. The construction of the simulation environment of computational simulation technology provides a simulation platform for the performance of a dancing robot. This is a great help in saving development costs and shortening the time required for choreography.

Originality needs to be evaluated in a consensus model (Mumford and Mcintosh, 2017). The computer simulation system uses the measurement of brain waves (EEG) (Matthias et al., 2010) to obtain people's quantitative indicators of creative results. At the same time, the scope of innovation is defined in artificial intelligence (Gazoni, 2016). The process of creative generation is the divergence of thinking information, but it does not mean that its expression function does not converge. When the system complexity reaches a saturation point, from the perspective of thermodynamics, energy will be released at this time. At this time, the function tends to converge. This paper defines the release range of this idea in the simulation system as the "emergence value range" (EVR). It also proposes the CGC as a means of measuring the creativity of the system. The evaluation method is realized by the quantum modeling method.

This paper gives full play to the advantages of computer simulation technology in visualization, modeling, and evaluation. It generates robot dance creativity and trajectory calculation results that meet people's needs. It drives dance robot performance through digital twin technology. This process realizes the point data sharing of robot dance performance and trajectory movement. Computer simulation technology combined with quantum modeling methods quantitatively evaluate the robot dance creativity and trajectory calculation capabilities.

Quantum Modeling Method

Since the introduction of quantum theory, it has been widely concerned in all walks of life. Nowadays, it is not only a modeling method and calculation theory in traditional physics but also an essential tool for problem-solving in fields such as mathematics (Jayantika et al., 2019), chemistry (Nalewajski and Roman, 2017), the life sciences (Han et al., 2001), and computer science (Bennett and DiVincenzo, 2000; Schramm, 2017; Aerts et al., 2019). For computer simulation, the superposition and entanglement of quantum theory mean that it plays a pivotal role. Quantum computing can be realized by using the properties of quantum superposition (Gyongyosi and Imre, 2019), and the applications of quantum entanglement are mainly in the fields of quantum teleportation, quantum encryption, and quantum dense coding. With the help of thermodynamic principles, the complexity of a quantum system can be calculated by entropy. The complexity theory of quantum computing is one of the fundamental theories of quantum computer science (Zhang et al., 2016a). This paper uses the quantum superposition property and GANs to construct QGANs to accelerate the data-processing and -learning process. This paper uses quantum theory, the superposition state, and entangled states in building the knowledge storage and retrieval structure of M-3DQKG. At the same time, by calculating the complexity of the quantum system, combined with CGC, it is possible to evaluate the creative generation ability of the simulation system.

If this paper transforms information into quantum state information, the information at this time is a superposition state. Then the form of the quantum state information is stored in each qubit. The superposition state makes the storage capacity of data much more important than traditional storage methods. This can prompt M-3DQKG to store more information in a shallower structure. Moreover, the qubit in the calculation is the superposition of 0 and 1, which makes quantum computation much faster than other classical algorithms (Knill et al., 2001). This can promote the QGAN based on this to have a better learning ability. More information stored and retrieved by the algorithm can make the trajectory calculation of the dancing robot faster, and the body language more abundant. We use creative analogy (Christensen and Ball, 2016) to combine newly acquired innovative thinking skills with normal scientific thinking processes and use proven techniques to expand the ability of humans to generate original ideas (Ness, 2012). Kitaev et al. (2011) and others introduced the new theory of quantum computing. The information of classical systems can be calculated with the help of quantum systems and their properties (Nielsen and Chuang, 2007; Roy et al., 2017), so as to realize the research of classical computing theory and quantum mapping. This also provides new ideas for improving the computing power of classical systems.

In the process of studying quantum entanglement, we were surprised to find that the coherence of the input state and the entanglement of the output state are quantitatively equivalent (Berta et al., 2015). The conversion between the coherent state and the entangled state means that the quantum coherent state can be measured by the entangled state (Plenio and Virmani, 2011). Nevertheless, we also found that quantum entanglement and quantum teleportation (Prakash, 2010) are effective means of quantum information transmission. Before the development and popularization of quantum computers, we could use computer simulation technology to simulate quantum models.

The entropy of the complexity of a quantum system is equivalent to the quantum information entropy (Edward, 2015). The linear characteristics of quantum systems make it possible to calculate the information entropy of quantum systems. As the quantum system complexity can be calculated through entropy (Deng et al., 2012), if the mapping relationship between the microscopic quantum system and the macroscopic classical system can be found (Neill et al., 2016), it may be possible to solve the problem that the complexity of the classical system cannot be evaluated owing to non-linear characteristics (Jami and Labbafi, 2017; Madhok et al., 2018).

According to the viewpoint of quantum information theory, quantum entanglement is a kind of physical existence with a measurement effect (Yu and Song, 2004; Li et al., 2010; Shota et al., 2017). Through the measurement and calculation of multiple dimensions of quantum theory (Heydari, 2006; Mintert and Buchleitner, 2007; Chernyavskiy, 2009), the purpose of system complexity evaluation can be achieved. Therefore, an entangled quantum system with strong operability, useful calculations, and containing mixed and pure states is valuable. This is conducive to quantitative evaluation to ensure the effectiveness of the dance robot's trajectory and the practicality of the creative action process.

Summary

The main content of this paper is to use computer simulation technology to realize the generation and trajectory calculation of dance robot dance creativity. The system builds the M-3DQKG structure based on the quantum modeling simulation model to achieve information storage and retrieval. Under the premise of information fidelity, a variety of neural networks are used to accelerate the creative generation process and increase its richness. This paper uses the quantum system linear method and EEG test to quantify and evaluate the creative ability and creative results of the system. The simulation system uses digital twin technology to copy the simulation output results that meet the needs of the dancing robot. These data will drive the trajectory and dance movements of the dancing robot.

METHOD

This paper proposes a modeling and simulation method to enhance the creative generation effect and uses this method to realize the action arrangement and trajectory calculation of the dancing robot in a particular scene. As shown in **Figure 1**, the general process of idea generation has many links and requires a lot of human resources and time to complete. This leads to very low efficiency in the proposal of creative generation schemes. Programming the creative generation process will be an essential means of improving efficiency and reducing costs.

The generation of dance creativity mainly involves four main links: skeleton movement extraction, movement template generation, three-dimensional (3D) model production, and the integration of characters and scenes, as shown in **Figure 2**. The region with convolutional neural networks (R-CNNs) recognizes characters' actions and movement tracks in text, sound, and images, and extracts bone information and displacement point information. The holistically nested edge detection (HED) network generates abstract models of virtual human limbs as action templates by extracting information. In the 3D engine, we bind the generated action template and bone attributes to the character model and drive the character model to complete the dance motion simulation. The QGAN network generates performance scenes based on the information extracted by R-CNNs.

To enhance the practicality and novelty of the results of dance creativity, we have made new attempts to increase the capacity of information storage and the efficiency of knowledge retrieval. We start from the creative needs, extract the keywords of the needs, and form the ontology domain, abstract model, and the weight index of the demand knowledge. The ontology domain is used to define the object to be creative and the constraint range of the object's actions, scenes, and expressions. The abstract model is a data model generated by the ontology domain to describe the virtual person. The demand knowledge weight index is a weight table defined according to the frequency and importance of the keywords in demand. Moreover, we extract dance videos, images, and music information from R-CNNs to form an action library. The action library stores the body movement data and the running track. Based on the properties of the quantum superposition state and entangled state, we build M-3DQKG result storage and retrieval knowledge based on action feature tags.

To ensure the accuracy of the body template and the rationality of the fusion of characters and scenes, we have carried out some research into information fidelity and optimizing the creative generation process. We put forward the calculation method of information fidelity and related parameters for each link. Moreover, we optimize the closure of the information flow of the two processes of action generation and scene generation in the creative generation process. Our purpose in doing this is to keep information from being disturbed and lost by the external environment. We use the information degree to continuously measure the information changes of the system and provide an available reference for the final creative result.

To evaluate the degree of creativity of the creative generation results, we propose a CGC linear evaluation method based on measuring the information entropy of the quantum system. This combines EEG results to evaluate creative results. CGC is the mapping value of the quantum system information complexity in the classical system. We calculate the complexity of the quantum system based on the equivalence relationship between quantum entanglement entropy and information entropy. We use the principle of thermodynamics to calculate the relationship between the information entropy of the quantum system and the classical system and realize the mapping of linear quantum systems to non-linear classical systems. We defined the critical value of the convergence of the CGC description function as the EVR to evaluate the creative generation ability of the system. In addition, we invited volunteers to take an EEG test while watching creative generation videos. We marked creative points based on EEG peaks. Creative points are used as knowledge labels for neural network learning. The result of learning is an update of the structure and weight of M-3DQKG. Finally, the purpose is to achieve the convergence of the CGC description function.

This paper takes the generation of dance movements as an experimental case and visualizes the generated dance movement videos in a 3D simulation engine through character modeling and scene fusion. The system will meet the simulation results of the needs assessment to produce dance robot bones, joints, and displacement data. These data are transferred to the robot, and, finally, used to realize the dance performance and trajectory calculation of the dancing robot.

Knowledge Retrieval Model

Knowledge retrieval of the creative generation is based on the knowledge graph. We propose the M-3DQKG model, which is used to construct the knowledge graph of the research object. Knowledge is a description of the attributes of the research object. Each node is defined as a qubit to store a unit of knowledge. The knowledge of a qubit is in a superposition state, which increases the data capacity of the knowledge graph. We divide knowledge into levels and determine the attribution or relationship of the upper and lower levels. Each layer forms an array of qubits. With the increase and connection of the qubits storing the knowledge unit points, the system creates an intricate knowledge network, as shown in **Figure 3**. When acquiring knowledge, knowledge units are retrieved one by one from the first level to deeper levels according to the connection relationship between qubits. Among them, the knowledge unit is locally entangled. A red dotted line connects the two entangled qubits in **Figure 3**. The generation of entanglement is determined based on the coherent state of the knowledge points stored in the qubit. When a knowledge unit in an entangled state is retrieved, we think that a measurement has been made. At this time, the state of the qubit is determined, and the superposition state of the qubit disappears. The information from the state change is instantly transmitted to the qubit at the other end through entanglement. The new search starts from this qubit, which realizes the cross-level search of the knowledge unit. The M-3DQKG model increases the diversity of knowledge unit

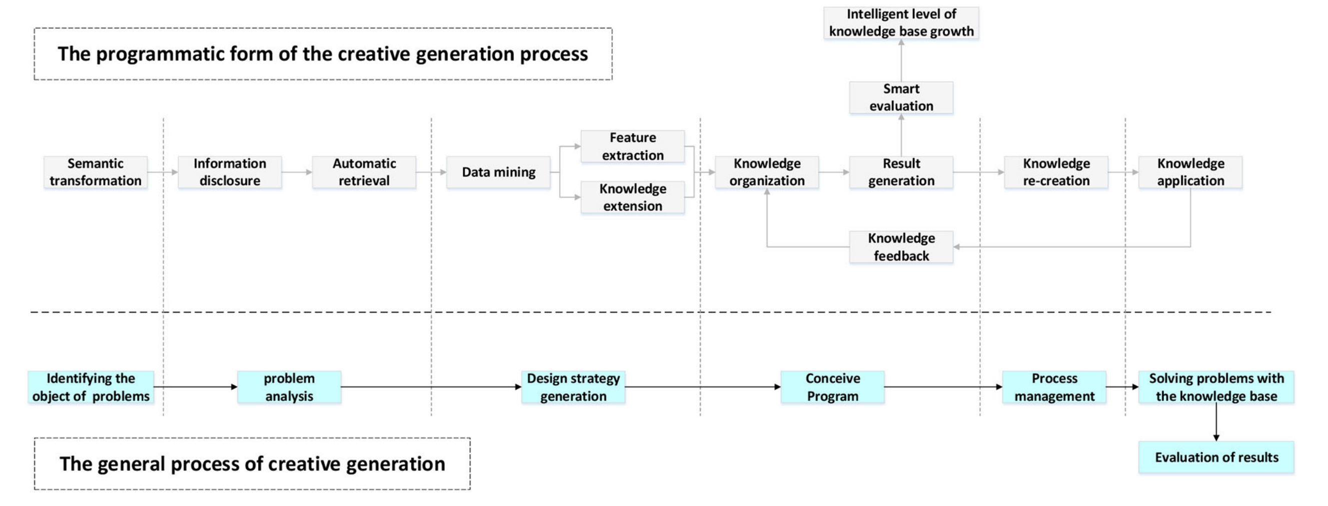


FIGURE 1 | Comparison between the general process of creative generation and the programmatic form.

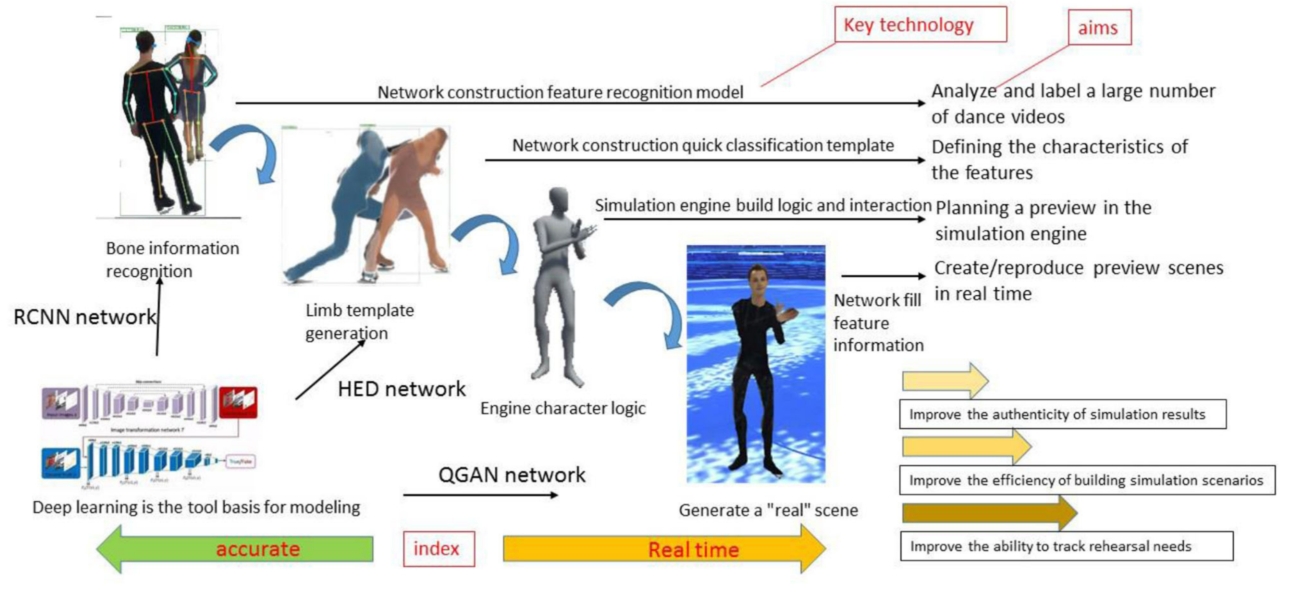

Key technology
aims
Network construction feature recognition model
Analyze and label a large number of dance videos
Network construction quick classification template
Defining the characteristics of the features
Simulation engine build logic and interaction
Planning a preview in the simulation engine
Create/reproduce preview scenes in real time
Network fill feature information
Bone information recognition
RCNN network
Limb template generation
HED network
Engine character logic
Deep learning is the tool basis for modeling
QGAN network
Generate a "real" scene
Improve the authenticity of simulation results
Improve the efficiency of building simulation scenarios
Improve the ability to track rehearsal needs
accurate
index
Real time

FIGURE 2 | The main links and key techniques of creative dance movements. HED, holistically nested edge detection; R-CNN, regional convolutional neural network; QGAN, quantum generative adversarial network.

retrieval. This model not only enriches the knowledge structure but also promotes faster information transmission.

In the process of retrieving knowledge based on the M-3DQKG model, the connection relationship between qubits is assigned weights according to the characteristics of the research object. Furthermore, we define the knowledge unit of the qubits in an entangled state. If one of the qubits is retrieved, the other qubit must be extracted. Therefore, the connection between them is assigned a weight of 1. The initial structure of the knowledge graph is based on research-on demand ontology (Jing et al., 2013; Zhang et al., 2016b). This ensures that the creative generation results remain valid. We extract the demand ontology and calculate the ontology domain of the idea generation so that the idea generation process always meets the constraints of the demand.

The learning network extracts keywords from the input demand information. The main research object is defined as the central ontology or primary ontology, and the secondary object is defined as the related ontology or secondary ontology. The ontology is the carrier of creative generation, and the keywords are bound to the ontology as attributes of the ontology. After that, the keywords are sorted according to their importance, and the sort number is marked. Take the expression rules in **Table 1** as an example of designing a sorting number labeling algorithm.

After the labels of the sorting numbers are assigned, each keyword is tested once. The test is based on the C-index in

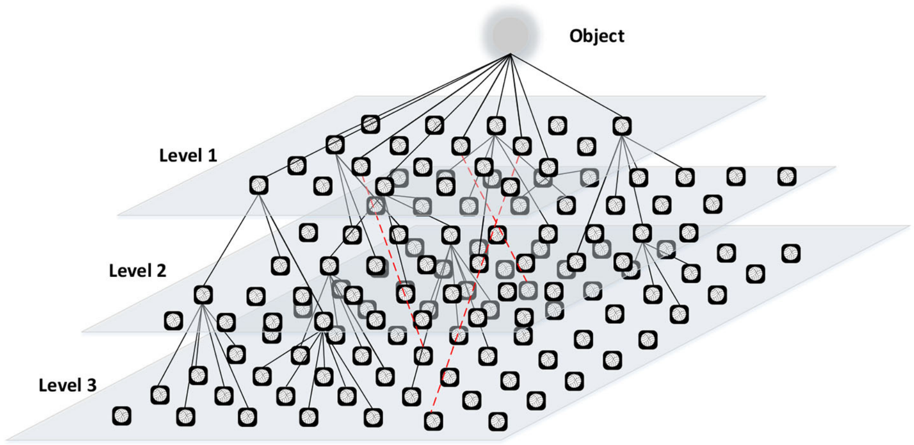


FIGURE 3 | Examples of quantum knowledge graphs.

TABLE 1 | Ontology and keyword sorting number labeling.

Object	**Person**						**Environment**		
Keyword	**Duet-dance**	**Slide**	**Whirl**	**Jump**	**Fast**	**Hold**	**Ice field**	**Audience**	**Light**
Number	1	2	3	4	5	6	7	8	9
C-index	4	3.32	2.86	2.71	2.15	1.88	1.27	1.03	0.54

Table 1. The purpose of the detection is to determine whether the labeling of the sorting number is accurate. Otherwise, the sorting number labeling of the main body and the sub-body will be reassigned. If multiple sub-ontologies and more keywords are determined from the demand information, the ranking numbers are marked sequentially. **Table 1** is used to find dance clips from the action library.

The generation of the action library relies on the R-CNN network to extract and label the features of videos, pictures, and audios. The tags are constructed into a knowledge network, as shown in **Figure 4**. In the network, the demand layer is the ontology information in **Table 1**. The retrieval layer is the knowledge network in the action library. The resulting layer is the conclusion data generated by knowledge reorganization. The system sequentially calculates the connection weight of the upper level knowledge and the lower level knowledge in the knowledge network. Take the hierarchical structure and node distribution in **Figure 4** as an example.

First, determine the interpretation matrix *A* of the primary ontology. According to the searching results of the row labels constructs the matrix, where B_1, B_2, and B_3 connected to it and $B_2 > B_3 > B_1$. The subordinate ranking numbers are marked as $B_1 = 5, B_2 = 2, B_3 = 3$.

$$\mathbf{A=\begin{bmatrix} 1 & |1/(B_1-B_2)| & |1/(B_1-B_3)| \\ |B_1-B_2| & 1 & |B_3-B_2| \\ |B_1-B_3| & |1/(B_3-B_2)| & 1 \end{bmatrix} \Leftrightarrow A=\begin{bmatrix} 1 & 1/3 & 1/2 \\ 3 & 1 & 1 \\ 2 & 1 & 1 \end{bmatrix}} \quad (1)$$

Calculate the primary weight *W* by the square root method,

$$\mathbf{W=\begin{bmatrix} \sqrt[3]{1^*\left|\frac{1}{(B_1-B_2)}\right|^*\left|\frac{1}{(B_1-B_3)}\right|} \\ \sqrt[3]{|B_1-B_2|^*1^*|B_3-B_2|} \\ \sqrt[3]{|B_1-B_3|^*\left|\frac{1}{(B_3-B_2)}\right|^*1} \end{bmatrix} = \begin{bmatrix} 0.338 \\ 1.442 \\ 1.260 \end{bmatrix}} \quad (2)$$

Normalize *W* using dispersion normalization and map the value between [0, 1],

$$\mathbf{w_i^* = \frac{w_i}{\sum_{i=1}^{N} w_i}} \quad (3)$$

$$\mathbf{W=\begin{bmatrix} w_1^* \\ w_2^* \\ w_3^* \end{bmatrix} = \begin{bmatrix} 0.111 \\ 0.475 \\ 0.414 \end{bmatrix}} \quad (4)$$

where w_i^* is the normalized value of each weight and W° is the normalized matrix. We can use this method to obtain the retrieval weight between each layer of tags. Mark the weights of each layer on the graph, as shown in **Figure 5**. It can be seen that, after knowledge retrieval and reorganization, the result layer data are output, and the weight that the knowledge unit can adopt is marked. The weight value can reflect the admissibility of the knowledge unit. The dotted line in the figure indicates that the intermediate knowledge retrieval level is omitted.

The knowledge units obtained in the result layer are used to generate a creative library through clustering. The generation of the creative scheme adopts the "probability

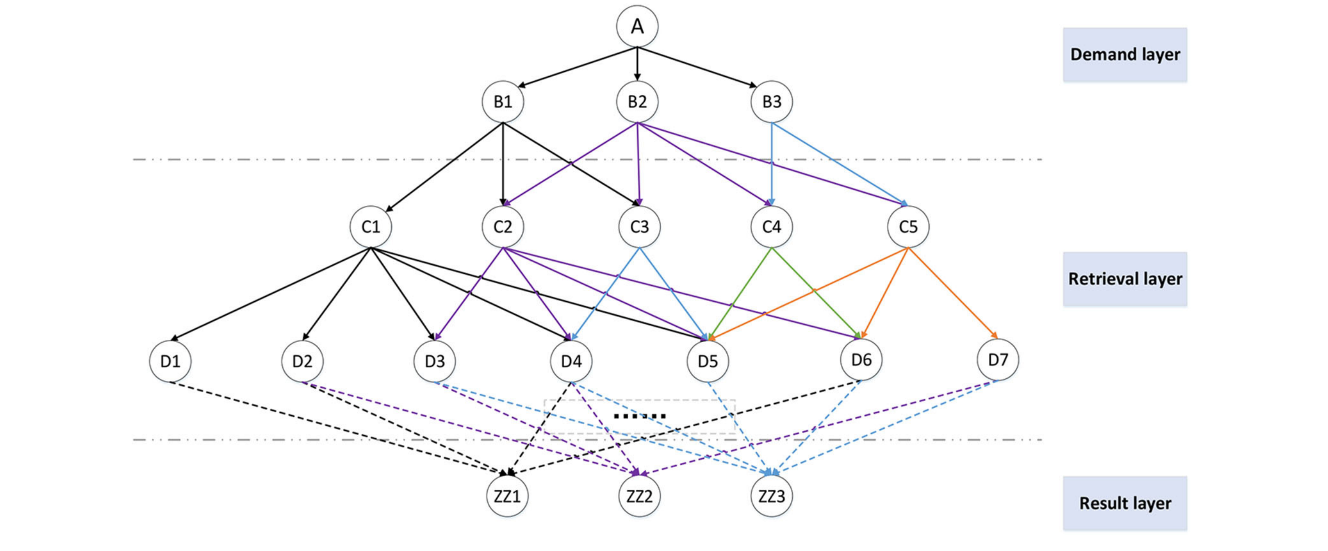


FIGURE 4 | Hierarchical structure of the M-3DQKG model.

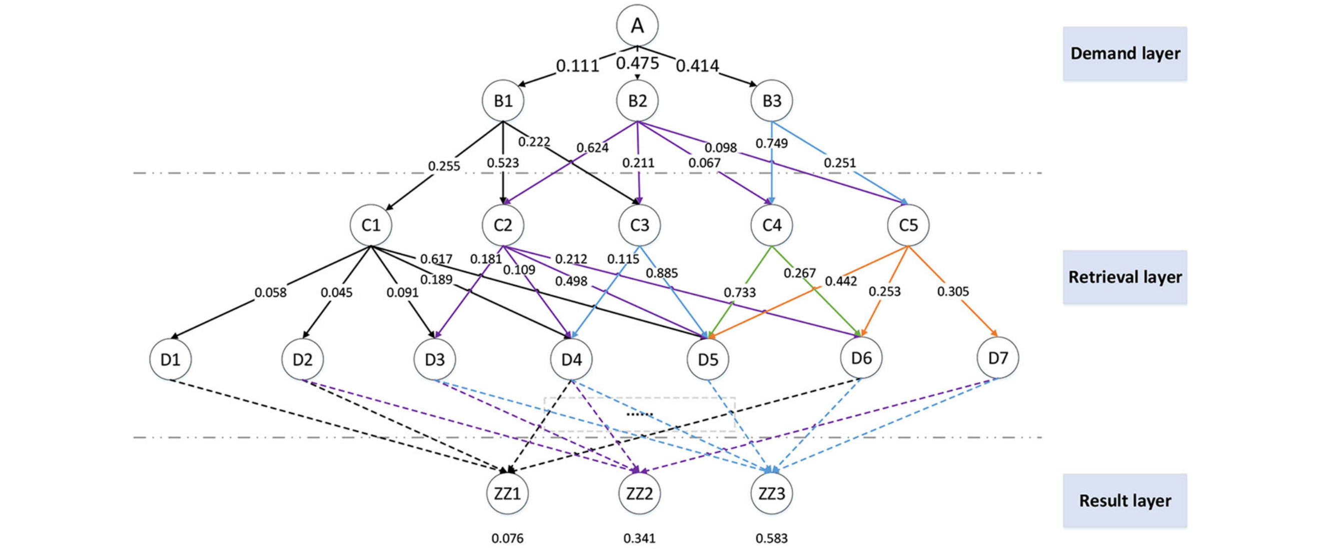


FIGURE 5 | M-3DQKG model retrieval weight calculation and creative library generation.

branch model" to generate the knowledge units in the creative library through combination, as shown in **Figure 6**. The final scheme judges its validity according to the value of the sum of weights. The M-3DQKG model is repeatedly reconstructed through the continuous "learning–improvement–re-learning–re-improvement" process of the learning network to obtain the most reasonable knowledge unit storage and retrieval structure.

The "probabilistic branching model" in **Figure 6** can reorganize the knowledge units of the result layer according to probability. The creative scheme generated after the reorganization is a random combination of "hot knowledge" with a relatively large weight and "cold knowledge" with minimal weight. The model is based on the creative knowledge unit from reorganization at the two levels of quantity and quality, so it improves the novelty and diversity of creative generation schemes.

Information Fidelity and Creative Generation Model

This paper proposes the concept of "information fidelity" in the process of creative generation. "Information fidelity" means that the information in the system should be as authentic and complete as possible. The information fidelity in this paper includes the reliability of the demand ontology information and the integrity of the information in the information-processing procedure. The former ensures that the results of the creative

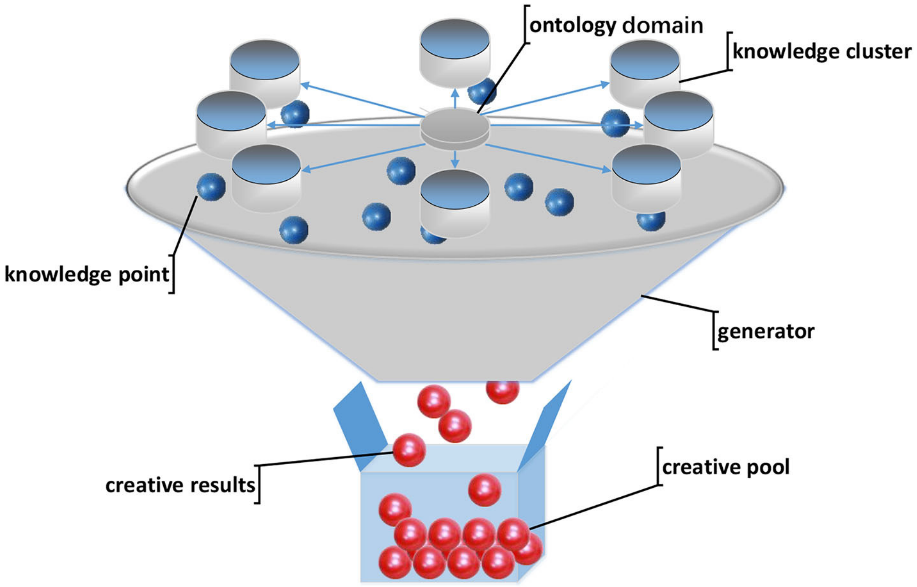


FIGURE 6 | The structure of the probability branch model in the knowledge base.

generation meet the constraints of the ontology domain, while the latter can maximize the creative ability of the system through rich information.

The input system of the required information in the creative generation process is the starting point; it ends with the generation of the creative plan. The process always follows the principle of information fidelity, and, under the premise of satisfying the ontology domain, it emphasizes that creativity should follow natural laws and social reality. Its purpose is to enable creative results to be used to solve a social problem.

Information fidelity needs to reduce the process of extraction and visualization in the process of information processing and transmission. Moreover, based on the current information-processing environment, even if we use multiple threads to process information, we cannot adequately meet the impact of big data. Information will be lost in a specific link, causing information loss. The advantage of parallel quantum computing lies in solving these problems. Creative computing has been able to satisfy the composition of music (Hugill, 2014; Giraud et al., 2016) in music creation, but it is a new endeavor in scene generation and dance performance. Because of the numerous elements involved in the creative calculation and the enormous amount of calculation required, the calculation cannot solve all creative problems in a short time, and the process of composing music is relatively simple.

To ensure the fidelity of the information, we have carried out a lot of research work mainly on the credibility of the information source, the closure of the information-processing method, the integrity of the information-processing results, and the timeliness of the information. Define the information fidelity $I(x)$ as

$$\boldsymbol{I(x) = f(Sc, Cl, Inte, Tim)} \tag{5}$$

where *Sc* is the credibility of the information source, *Cl* is the degree of closure of the information-processing method, *Inte* is the completeness of the information result, and *Tim* is the timeliness evaluation value of the information.

$$\boldsymbol{Sc(x) = q(1-p)\sqrt{\sum_{i=1}^{q}\frac{(x_i - x_0)^2}{q}}} \tag{6}$$

where p represents the error tolerance rate of the dataset and q is the amount of the subset of the dataset. Sources of information include the initial graphics, audio–video datasets, databases, and requirements documents. All of these must form an exact information index database, which is convenient for checking that the information is correct.

$$\boldsymbol{Cl(m,n) = dec(m,n)\frac{deep^{*}num}{sum}{}^{*}rate^{*}100\%} \tag{7}$$

where $dec(m, n)$ is the information loss function affected by the environment, m is the amount of quantum parallel computation, n is the number of the structural unit of each parallel computing process, *deep* represents the depth of knowledge graph retrieval, *num* is the cumulative number of knowledge nodes that the process goes through, and *sum* is the total number of nodes of knowledge points. Furthermore, *rate* is the coverage rate of knowledge points. During information processing, the database must not only cover enough demand information but also form a complete expression of the characteristics of the data. Based on the M-3DQKG model construction method in the previous section, the system assigns weights to the knowledge units containing the information of the ontology and realizes information retrieval.

$$\boldsymbol{Inte(t) = \frac{sets(t) \cap (dm \cup F)}{dm \cup F}{}^{*}com(t)^{*}100\%} \tag{8}$$

where *sets* (t) is the fitting function of the dataset containing the ontology domain and feature labels returned at a random time, *dm* is the ontology domain set, F is the feature label set, and *com* () is the proper function of the knowledge nodes used. This paper proposes a comparison method to compensate for information loss. The method is a combination of information feedback and a suitable comparison. The feedback information is the ontology domain and feature label of the current creative generation at a random time in the process of generating results. Based on the feedback from these two characteristics, the deep learning network compares them with the original demand data. If the ontology domain or the feature labels are different during the fitting process, it is considered that a suitable comparison has failed. At this time, the system considers that the result of the creative calculation has deviated. This deviation is part of the unsuccessful suitable comparison segment. The successful part of fitting and matching forms the starting value of the creative evaluation score (between [0, 100]). This value will also be used as the evaluation value for the fidelity of the information when the creative generation result is evaluated. Also, all information in the creative process should be kept up to date.

$$\boldsymbol{Tim}\,(\boldsymbol{t}) = \frac{\Delta \boldsymbol{q}}{\Delta \boldsymbol{Q}} \frac{\left(\boldsymbol{t}-\boldsymbol{t}'\right)}{\boldsymbol{t}} \int_{\boldsymbol{t}}^{\boldsymbol{t}'} \boldsymbol{f}\,(\boldsymbol{x})\, \boldsymbol{dx} \tag{9}$$

where $t-t'$ is the difference between the current time and the data update time, q is the sum of the absolute value of the increase and decrease of the data from t' to t, Q is the total amount of current information, and $f\,(x)$ is the quantization function of the information.

In the process of generating ideas, the data are updated continuously. The arrows in **Figure 7** indicate the input of new data, and the numbers indicate the order of information input. Each row represents the progress of the data calculation. Different colors indicate the module unit of varying processing times. The first line is the primary process, which is used to interact with the data and integrate the data directly. The module unit processes integrated data based on the last completed process. Processes completed before this base point choose to wait. Moreover, the integrated data are used for idea generation.

Creative generation is based on information fidelity calculations. This paper proposes a basic framework for the creative generation model based on the flow of information. The creative generation model was mentioned at the outset of digital creative technology (Lee, 2016), but, unfortunately, the definition of the creative model is relatively one-sided and not universal. Moreover, the method is single and cannot complete the creative generation process quickly and intelligently. This paper uses a variety of deep learning networks and quantum technologies to reconstruct the creative generation model. Compared with previous studies, the framework has the characteristics of universality, high performance, and intelligence.

It can be seen from **Figure 8** that this paper proposes a four-stage simulation framework for the creative generation process. The demand analysis module determines the ontology domain and feature labels through demand analysis. The system generates abstract models with unified characteristics and stores knowledge weight tables. The data-processing module extracts the action library and knowledge network through the R-CNN network and establishes a connection between them. The M-3DQKG structure is constructed using quantum modeling and simulation methods to realize entangled interconnection and the teleportation of information. The information stored in the qubit forms a backtracking table, which compares the requirements with the update of the knowledge network. In the creative generation module, action learning extracts dance action models through R-CNN and HED and builds simulation models in a 3D engine. At the same time, QGAN generates an expression environment that meets the characteristics of the dance and realizes the integration of dance movements and the expression environment. The mature performance data are twinned on the robot to drive the robot's dance expression. The result evaluation module uses EEG tests to mark the creative tags that trigger excitement on the creative generation results and uses information fidelity and CGC to evaluate the novelty, practicality, and intelligence of the creative results. Under the framework of this creative generation model, computer simulation technology reduces the event cost of robot debugging. It ensures the closure of the creative generation process and the practical value of the creative results.

Linear Evaluation of System Creative Ability

Three crucial indicators comprehensively evaluate the creative ability of the system. The first one is the amount of information that the system may need to generate originality. The second one is the timeliness of the results, and the last one is the diversity of results. The M-3DQKG model solves the problem of result diversity. Information fidelity is used to evaluate the timeliness of the generated results. Moreover, we study the problem of the calculation of the amount of information in this section. The evaluation of creative results should conform to four criteria. First, the semantic extracted features and demands need to match with others to confirm that the learning network in the creative generation model satisfies the characteristic value of the ontology domain. The M-3DQKG model does this work. Second, the abstract model and dataset training results also need to match to generate an abstract model based on the dataset that has the same attributes as the research object. The creative generation model solves this problem. Third, the output results and realization conditions need to match to ensure that creative solutions can be realized under existing conditions. Information fidelity computing achieves this goal. Finally, the creative ability and psychological expectations need to match to develop a creative generation system that meets people's expectations. This issue will be studied in this section.

This paper proposes a method to evaluate the creative ability of the system by calculating the complexity of the system to quantify the creative capacity of the system. This method maps the non-linear relationship of the complex macroscopic system into the linear space of the microscopic quantum system. The complexity of the system is quantitatively analyzed by calculating the changes in system energy during information transmission

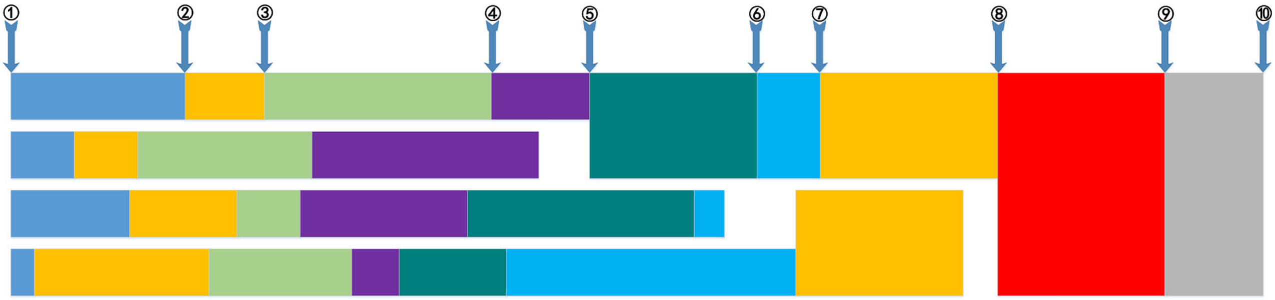


FIGURE 7 | Information-processing timelines.

and conversion. The amount of information calculates the measurement of information. Entropy is used as a unified calculation standard to quantify the energy change in the system from the thermodynamics perspective (Nalewajski, 2016). Thereby, humans can obtain system complexity (Burdon et al., 2018). As shown in **Figure 9**, the entropy is used to measure the consistency between macro-complex systems and micro-quantum systems. At the same time, this paper proposes the concept of the "creative generation coefficient" (CGC), which is a standardized value used to map the system's current complexity and information volume. The value continually changes with time. What is more, the creativity generation coefficient is just like the value of evaluating people's IQs. It can also reflect the level of intelligence and creativity of the system. When the CGC reaches a specific value, it means that the complexity of the system has reached a high level. According to thermodynamic theory, the increasing trend is suppressed by quantum effects, although quantum chaos can lead to an increase in entanglement, that is, an increase in quantum entanglement entropy. In other words, the entanglement entropy cannot be infinite. It tends to converge. Humans cannot make the complex system lose its original emergent properties by increasing the entanglement of the quantum system. The entropy of the system cannot be increased indefinitely, which also coincides with the emergence of complex systems (Morrison, 2003). If the system emerges strongly, it gradually changes from the disordered state to the ordered state when the emergence process occurs. Then, the entropy of the system decreases. Therefore, we define that the system will emerge and complete the creative generation process when the CGC grows to a higher value.

Information is stored in qubits, which are particles. They are in a linear superposition state. This can be expressed as

$$\Psi = \mathbf{c_1}\Psi_\mathbf{1} + \mathbf{c_2}\Psi_\mathbf{2} \tag{10}$$

where Ψ_1 and Ψ_2 are the possible states of particles and c_1 and c_2 are constant complex numbers. If entanglement occurs in the quantum system, the possible state of the particles can be expressed as a spin form, which is shown with the Dirac mark

$$|\psi\rangle = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}(|\uparrow\rangle \otimes |\downarrow\rangle - |\downarrow\rangle \otimes |\uparrow\rangle) \tag{11}$$

where $|\uparrow\rangle$ and $|\downarrow\rangle$ indicate that the spin of the particles is up or down, respectively. In order to facilitate calculation, quantum entanglement is expressed mathematically. Assuming that a composite system is composed of two subsystems A and B, the Hilbert spaces of these two subsystems are H_A and H_B, respectively, and the Hilbert space H_{AB} of the composite system is a tensor product

$$\mathbf{H_{AB}} = \mathbf{H_A} \otimes \mathbf{H_B} \tag{12}$$

Set the quantum states of subsystems A and B to $|\alpha\rangle_A$, $|\beta\rangle_B$, respectively. If the quantum state of the composite system $|\psi\rangle_{AB}$ cannot be written as a tensor product $|\alpha\rangle_A \otimes |\beta\rangle_B$, the compound system is called the entanglement system of subsystems A and B, and the two subsystems A and B are entangled with each other.

The state vector $|\psi\rangle$ or the density operator $\rho = |\psi\rangle\langle\psi|$ can describe the quantum state of a pure state. Density operators can only describe statistical mixed states

$$\rho = \sum_\mathbf{k} \mathbf{P_k} |\psi_\mathbf{k}\rangle\langle\psi_\mathbf{k}| \tag{13}$$

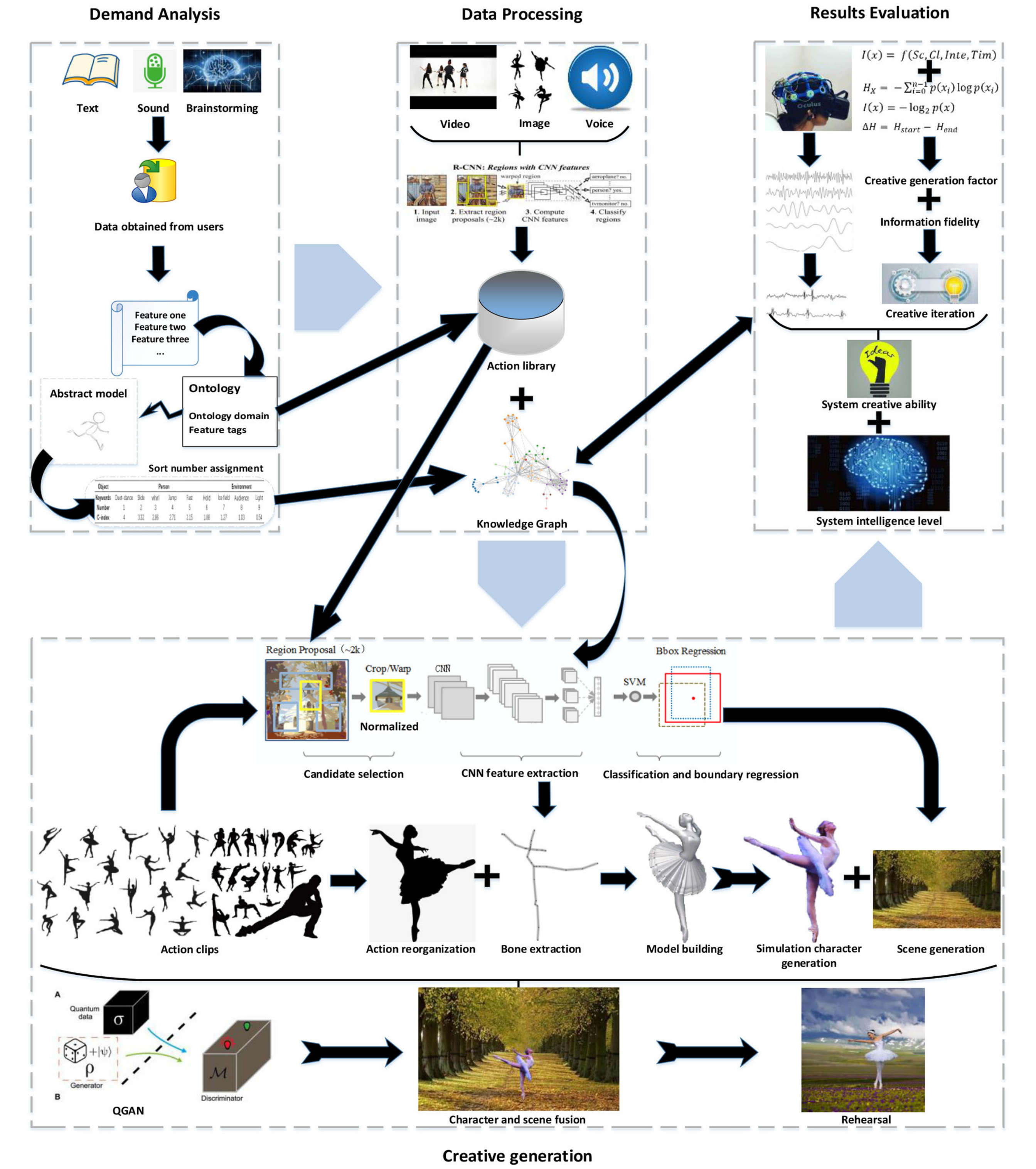
where P_k is the pure state $|\psi_k\rangle$ probability of occurrence in the statistical mixed state ρ. They satisfy the normalization condition

$$\mathbf{Tr}\rho = \sum_\mathbf{k} \mathbf{P_k} = \mathbf{1} \tag{14}$$

Because of the inevitable effect of the environment, the system is converted from the initial pure state (the coherent superposition state of the coherent state) to the mixed state (the incoherent superimposed state of the coherent state) when passing through the cavity field in the process of information propagating using the qubit as a carrier. The pure state of this composite system can be expressed as

$$|\psi(\mathbf{0})\rangle \propto (|\alpha\rangle + |-\alpha\rangle) \tag{15}$$

where $|\alpha\rangle$ is an assumption that the cavity field is initially prepared in a coherent state and defines that the environment is



Object	Person						Environment		
Keywords	Duet-dance	Slide	whirl	Jump	Fast	Hold	Ice field	Audience	Light
Number	1	2	3	4	5	6	7	8	9
C-index	4	3.32	2.86	2.71	2.15	1.88	1.27	1.03	0.54

FIGURE 8 | The creative generation model of dance movements integrated with performance scenes. CNN, convolutional neural network; SVM, support vector machines; QGAN, quantum generative adversarial network.

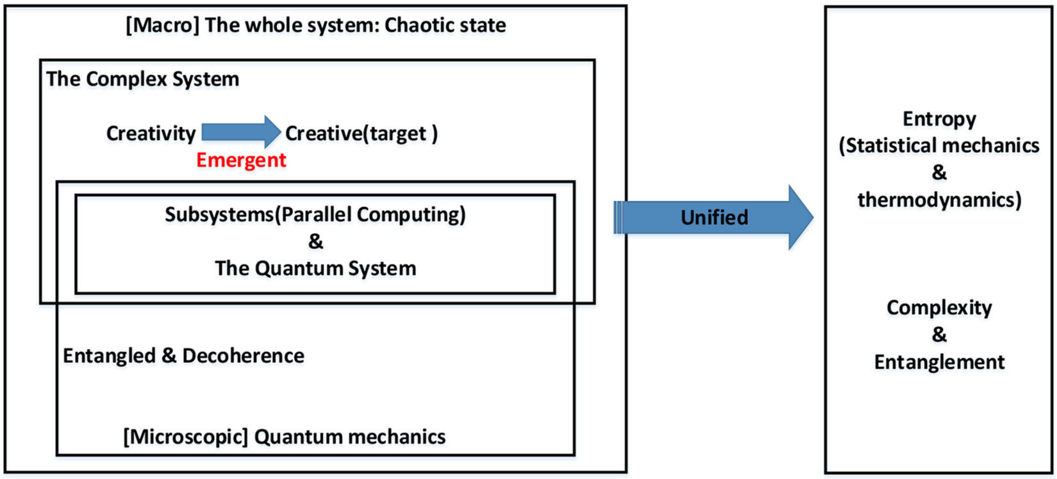


FIGURE 9 | The relationship between macroscopic complex systems and microscopic quantum systems.

initially in the state $| \varepsilon\rangle$, then the state of the composite system at time *t* becomes

$$| \psi (\mathbf{t})\rangle \propto (| \beta (\mathbf{t})\rangle | \varepsilon_1\rangle + | -\beta (\mathbf{t})\rangle | \varepsilon_2\rangle) \tag{16}$$

where

$$\beta (\mathbf{t}) = \alpha \mathbf{e}^{-\frac{\gamma \mathbf{t}}{2}} \tag{17}$$

γ represents the information loss rate of the cavity field and $| \psi\rangle$ is the quantum state of the cavity field. To trace the environmental variables, the reduced density operator of the cavity field is

$$\rho_{\mathbf{F}} = \mathbf{Tr}_{\mathbf{e}}\rho = \sum_{\mathbf{j}} \langle \varepsilon_{\mathbf{j}} | \psi (\mathbf{t})\rangle \langle \psi (\mathbf{t}) | \varepsilon_{\mathbf{j}}\rangle \propto | \beta (\mathbf{t})\rangle \langle \beta (\mathbf{t})| + |- \beta (\mathbf{t})\rangle \langle - \beta (\mathbf{t})| \tag{18}$$

It can be seen that, because of the effect of the environment, the pure state becomes a mixed state. When $t \rightarrow \infty$, the cavity field information will decay to the vacuum state $|0\rangle$. For linear entropy, the nature of the trace of the density operator ρ is

$$\mathbf{Tr}\rho = \begin{cases} \mathbf{1}, & \textbf{pure state} \\ < \mathbf{1}, & \textbf{mixed state} \end{cases} \tag{19}$$

Tracing the density operator can obtain the moisture entropy of the system, then the entropy value of the system is calculated according to the entanglement entropy

$$\mathbf{S}(\rho) = -\mathbf{Tr}\left(\rho \log \rho\right) \tag{20}$$

The rate of change of entropy with time is

$$\dot{\mathbf{S}} = -\frac{\mathbf{dTr}\rho^{2}}{\mathbf{dt}} = -\mathbf{Tr}\left(\dot{\rho}\rho + \rho\dot{\rho}\right) \tag{21}$$

Since the entanglement entropy is not additive, the amount of information reflecting the two entangled subsystems is

$$\mathbf{I}(\mathbf{A},\mathbf{B}) = \mathbf{S}(\mathbf{A}\cup\mathbf{B}) - \mathbf{S}(\mathbf{A}) - \mathbf{S}(\mathbf{B}) + \mathbf{S}(\mathbf{A}\cap\mathbf{B}) \tag{22}$$

This is Shannon's proposed definition of information entropy. It is seen that the information entropy can be expressed as the mathematical expectation of the amount of information $I(x_i)$ provided in the system when an event x_i appears in set X, which is

$$\mathbf{H}_{\mathbf{X}} = -\sum_{\mathbf{i}=0}^{\mathbf{n}-1} \mathbf{p}(\mathbf{x}_{\mathbf{i}}) \log \mathbf{p}(\mathbf{x}_{\mathbf{i}}) \tag{23}$$

For information, it is a question of probability. A large probability only requires one bit to be transmitted. In other words, the smaller the probability, the more bits need to be transmitted. Using the number of bits to measure the amount of information, the lower probability, the enormous amount of information. The formula based on this information is

$$\boldsymbol{I}(\boldsymbol{x}) = -\mathbf{log}_{2}\boldsymbol{p}(\boldsymbol{x}) \tag{24}$$

It can be clearly seen that there is a correlation between entanglement entropy and information entropy. The change process of the system from the pure state to the mixed state also exactly meets the necessary environment for creative computing (Liu and Yang, 2014). The amount of information on the two entangled subsystems can also be calculated. When the input information is stored in the qubit unit of the system, the amount of information is known. Using this as the cardinal number, humans can change the system information through the calculation used in the process of information transmission. This change is mainly the result of the increasing amount of information. The added information is the process data and this results in the dataset generated by the learning network after learning. It also includes a knowledge graph that represents the relationship between the data, an abstract model representing the ontology of things, and point information, representing the movement of the model. When the amount of information reaches a specific value, the information entropy of the system also reaches a higher value. At this time, as a subsystem of a complex system, each quantum system can map the result to the information entropy of the complex system through the calculation of information entropy. Based on this, the complexity of a complex system can be measured. For this measure of complexity, the value can be mapped to a value between [0, 100]. This is the "creative generation factor" (CGC) that we proposed. The emergence phenomenon will occur when the system information is saturated. At this time, the system information is continuously released to the outside system, and the information entropy of the system will increase slowly, and even tend to converge. The quantity of emergence of the system is defined as the total value from the beginning of emergence to the end of it, that is,

$$\Delta \boldsymbol{H} = \boldsymbol{H}_{\boldsymbol{start}} - \boldsymbol{H}_{\boldsymbol{end}} \tag{25}$$

This can more intuitively represent the system's ability to generate creativity. The CGC $C\,(x,t)$ is

$$\mathbf{C}\,(\mathbf{x},\mathbf{t}) \rightarrow \mathbf{f}\left(\mathbf{I},\, \Delta\mathbf{H},\, \mathbf{H}_{\mathbf{X}}, \dot{\mathbf{S}}\right) \tag{26}$$

When the system emerges, the CGC current value is called the "emergence value range" (EVR). After the CGC reaches the EVR, the representation function of the CGC tends to converge. From the perspective of system evaluation, we believe that the system has a high creative generation ability. We use this method to achieve a linear evaluation of the system's ability to generate originality. This method, combined with EEG, can reasonably evaluate the results of the creative generation. We will explain the evaluation experiment process in detail in the next section.

RESULTS

The experiment was deployed on an IBMAC922 machine, using four Tesla V100 floating-point arithmetic cards. It was compiled with Python3.5+ and OpenCV 3.4 under the environment of IBMpcc64 ubuntu18.04. The experiment used the cityscape database for 1,000 epochs of deep learning. It was able to obtain 3D scenes and task models, generate action libraries, and perform fragmented creative simulations of dance movements. The engine achieves dance performances by combining creative dance moves and generated simulation scenes. Then, the system gets the relevant results, showing that the generated actions are not the same as the learned video library. The rearrangement of the combined sequence of actions and the movement path of the skeleton of the simulation model can achieve creative generation. Dance movements can be viewed through VR devices, and humans can experience the viewing effects in the performance scene.

According to different task requirements and files, the learning network extracts data feature values and labels the data to form the original feature label table, as shown in **Table 2**. Dance creativity is classified according to the type of dance, such as folk dance, tap dance, and ballet. We assign keyword tags that represent the characteristics of each type of dance. These tags are used to match with the requirements to determine the actions needed for that dance action. Moreover, dance performances combined with dance moves need to have corresponding scenes to express the artistic conception. Therefore, the environmental label is also stored in this table. For example, mountains, plants, landscapes, and street scenes. The data extracted from this table are as accurate as possible. As the only constant table, it is the standard for feedback comparison in the system learning network's learning process, and it is also the basis for data review. Similarly, this table contains not only the primary data for defining the ontology domain of ideas but also provides the basis for evaluating the results of creative generation.

For the creative process of dance movements, an initial knowledge graph is formed, according to **Table 2**. This knowledge graph is the accumulation of experience in the formation of traditional ideas. The system must search and form a new knowledge graph–M-3DQKG search structure, first based on the task feature label. Then, the system stores the knowledge in the qubit. For each knowledge point, the weight is determined according to the degree of primary and secondary requirements, as shown in **Table 3**. M-3DQKG retrieves the hierarchical management of the structure and defines the weight of the parent node to the lower child node. The sum of the weights is 1. There may be entangled nodes each time, defining the entangled pairs for cross-level retrievals, such as B2 and D9.

According to the weight distribution of the knowledge graph, dance movements are recombined from the action library formed by the learning network through learning to form dance fragments, which are combined with the probabilistic branch model. The system generates simulation scenes based on dance clips and demand ontology domains. Then, the simulation scene and dance movements are combined to complete the creative process. **Figure 10** shows an example of generating a character abstract model and the dance movements.

The subjects watched the creative generation plan and recorded EEG data through EEG equipment, as shown in **Figure 11**. The participants were creative needs proposers and student volunteers (a total of 26 people). The former assessed how well the creativity needs are met, while the latter pretended to be an audience watching the creativity. When the subject was interested in the creative points generated by the system, the EEG data obtained from the test had a peak. The time corresponding to the peak was defined as the creative time point. In **Figure 11**, for example, the creative time points for the first subject were the 12th and 69th s. The subjects watched the creative plan again. This time, they compared their EEG data and marked the creative tag at the point of the stimulus (the peak point during the EEG). We analyzed these times in the subjects' EEG data and marked keywords that were of interest to the subjects. We organized these keywords to form an evaluation form, as shown in **Table 4**. We added the tags for each subject to the knowledge graph one by one as the negative feedback knowledge of M-3DQKG. If the label already existed in the knowledge graph, then a new round of

TABLE 2 | Original feature labels.

Ontology	Feature label	Attributes	Layer number	Node number	Retrieval rate
Folk dance	Dai people	Peacock, drums, performance...	6	541	0.673
	Dong nationality	Chorus, lusheng, love...	6	532	
	Korean	Sword dance, fan dance, chic, cheerful...	5	344	
	Uighur	Plate dance, tambourine dance, head, shoulders, waist, and eyes...	8	892	
	…	…			
Tap dance	American	Free, relaxed, complex rhythm, jazz...	3	87	0.522
	British	Rotate, slide, beautiful, split sound, repeat...	5	381	
	Irish	No upper body movements, crossed feet, soft shoes...	3	98	
	…	…			
Ballet	Banquet ballet	Song, dance, recitation...	2	22	0.735
	Court ballet	Drama, acrobatics, music...	2	19	
	Plot ballet	Education, story...	4	235	
	Romantic ballet	Lighting, big screen, lightness, poetry...	3	106	
	Russian ballet	Women, elves, ghosts, gods, longing...	4	269	
	Contemporary ballet	Life, stretch, jump, straight...	3	112	
	…	…			
		...			
Mountains	Green hills	Green, vitality, birds, and flowers...	5	401	0.449
	Snow mountain	White, cold, steep...	3	115	
	Distant mountain	Small, rolling...	5	384	
	……	…			
Plants	Flower	Colorful, petals, stamens, chrysanthemums, sunflowers...	4	298	0.907
	Grass	Green, four-leaf clover, mimosa...	7	754	
	Tree	Colors, leaves, branches, trunks, pine trees, willows...	5	372	
	…	…			
Landscapes	Bright moon	Bright, curved, round...	2	13	0.846
	Mountains	High, overlapping...	6	557	
	Morning glow	Red, big, halo...	5	369	
	…	…			
Street scenes	Street	Width, length, car, lights...	8	933	0.633
	Building	Shape, floor...	11	1,421	
	Shops	Signboard, display...	10	1,275	
	…	…			
		...			

machine learning updated its weight W. After repeated iterations, we perfected the structure of M-3DQKG.

As shown in **Figure 11**, different subjects have different points of interest in creativity. For example, the top subject only showed peaks on both sides, while the middle subject showed peaks that appeared very frequently. However, it can be seen that they have coincident peak points. **Table 4** reflects the intersection of the creative points.

According to the evaluation results, the previous knowledge graph is readjusted to assign weights to highlight the creative points of the test-takers' focus. The results of one adjustment are shown in **Table 5**. Compared with **Table 3**, the connection node of the A1 node is increased by B31, and the connection weight of each node has changed. Similarly, the number of connected child nodes of other parent nodes has been adjusted. As the creative label in **Table 4** was added on the one layer, the number of changed nodes increased, and the numbering sequence was changed. Such adjustments may need to be made hundreds or thousands of times. All are done automatically by the system. **Figure 11** shows the EEG test results of the first three test-takers in **Table 4**.

Finally, the creative results that meet the creative needs are evaluated for their information fidelity. Meanwhile, in the creative generation process, we have to evaluate the

TABLE 3 | Original table of the knowledge graph and weights.

	Label		Connection nodes						Entangled nodes
Name	A	A1	B3	B7	B10				
W			0.241	0.533	0.226				
Name		A2	B2	B5	B9	B12			
W			0.321	0.552	0.109	0.018			
Name		A3	B1	B3	B4	B6	B8	B11	
W			0.116	0.284	0.485	0.061	0.033	0.021	
Name	B	B1	C5	C8	C13	C17			
W			0.702	0.135	0.098	0.065			
Name		B2	C1	C6	C7	C16	C22		D9
W			0.238	0.299	0.374	0.018	0.071		
Name		B3	C3	C4	C11	C15	C16		
W			0.096	0.167	0.355	0.219	0.163		
Name		B4	C6	C8	C12	C15	C20		
W			0.457	0.166	0.182	0.064	0.131		
Name		B5	C4	C9	C14	C18	C24	C31	
W			0.205	0.611	0.034	0.142	0.005	0.003	
…						…			
Name		B11	C2	C7	C19				F11
W			0.857	0.032	0.111				
Name		B12	C5	C11	C27	C32			
W			0.629	0.174	0.062	0.135			
Name	C	C1	D3	D12	D44	D47			
W			0.525	0.261	0.103	0.111			
Name		C2	D2	D23	D39				
W			0.732	0.233	0.035				
Name		C3	D9	D11	D52	D69	D73		E7
W			0.494	0.268	0.127	0.042	0.069		
…						…			

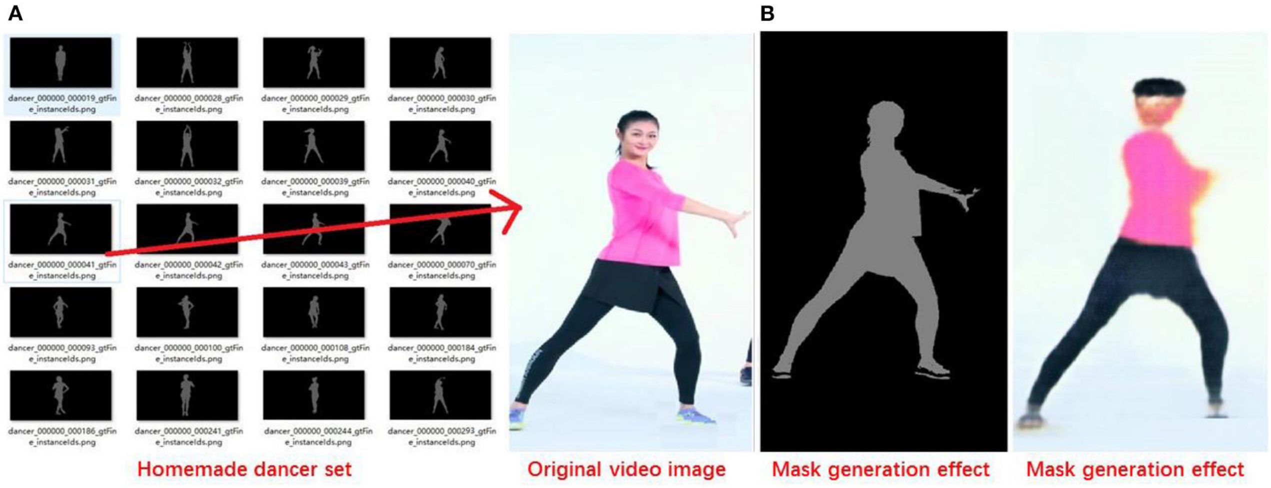

A
B
dancer_000000_000019_gtFine_instanceIds.png
dancer_000000_000028_gtFine_instanceIds.png
dancer_000000_000029_gtFine_instanceIds.png
dancer_000000_000030_gtFine_instanceIds.png
dancer_000000_000031_gtFine_instanceIds.png
dancer_000000_000032_gtFine_instanceIds.png
dancer_000000_000039_gtFine_instanceIds.png
dancer_000000_000040_gtFine_instanceIds.png
dancer_000000_000041_gtFine_instanceIds.png
dancer_000000_000042_gtFine_instanceIds.png
dancer_000000_000043_gtFine_instanceIds.png
dancer_000000_000070_gtFine_instanceIds.png
dancer_000000_000093_gtFine_instanceIds.png
dancer_000000_000100_gtFine_instanceIds.png
dancer_000000_000108_gtFine_instanceIds.png
dancer_000000_000184_gtFine_instanceIds.png
dancer_000000_000186_gtFine_instanceIds.png
dancer_000000_000241_gtFine_instanceIds.png
dancer_000000_000244_gtFine_instanceIds.png
dancer_000000_000293_gtFine_instanceIds.png
Homemade dancer set
Original video image
Mask generation effect
Mask generation effect

FIGURE 10 | Dance movement generation **(A)**. Deep learning process **(B)**. Generation of virtual people.

system's creative generation ability. As shown in **Figure 12**, the entanglement entropy and information entropy of the microscopic system are tested, and the two are mutual proof that errors in the measurement data have been prevented. Similarly, the information entropy of a complex system is calculated through the change in information volume of the macro-system.

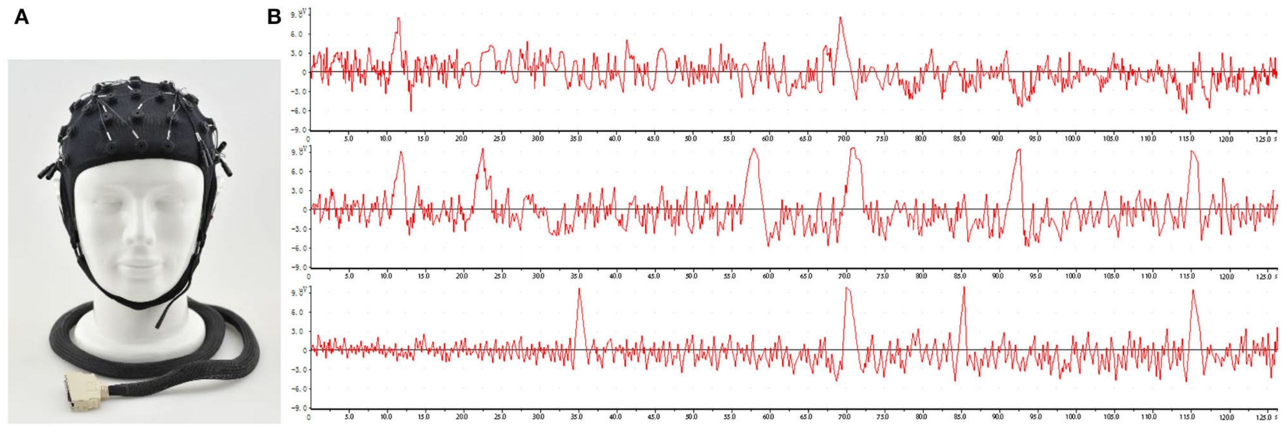


FIGURE 11 | Electroencephalographic (EEG) experiment and waveform. **(A)** Schematic of the EEG equipment used. **(B)** EEG waveforms of three subjects.

TABLE 4 | Test takers' creative input label.

Creative time	$T = 12$ s		$T = 35$ s			$T = 84$ s	$T = 92$ s	
Insertion time		$T = 23$ s		$T = 57$ s	$T = 69$ s			$T = 115$ s
Subject 1	Gesture				Spinning			
Subject 2	Cross	Light		Imagination	Passion		Dynamic	Warm
Subject 3			Bounce		Spinning	Expression		Squat-up
Subject 4	Fly	Light	Power			Pleasant		Elegant
Subject 5				Imagination	Spinning		Music	
Subject 6			Power		Spinning	Pleasant		Soothing
Subject 7		Light			Spinning	Expression	Music	Warm
…								
Subject 26				Imagination	Spinning	Expression	Dynamic	Elegant

It can be seen that there is a correlation between the three. It can be proved that the calculation of the information entropy of the complex macroscopic system can be obtained by the linear calculation of the microscopic quantum system. Based on the calculation method of the CGC proposed in this paper, the information entropy of the complex system is mapped to the interval [0, 100], as shown in **Figure 13**. It is not difficult to see that there is also a mapping relationship between the CGC, information entropy, and system complexity entropy. Furthermore, it is used as the basis to calculate the creative generation ability of complex systems.

As shown in **Figures 12**, **13**, the mapping relationships among the CGC, the system information quantity, and the system complexity have provided the quantitative calculation basis for the assessment of the system's creative ability. When the CGC reaches an absolute value, the amount of information will no longer increase significantly, and tends to converge. As shown in **Figure 13**, after 15 iterations per month, the value is stable at around 95. This phenomenon can be explained by the heating effect of the metering subsystem. Currently, the system completes the creative generation process and releases some storage space. In other words, the process of releasing energy in a complex system is the process of completing the emergence of information and realizing the creation of ideas. In this way, the pressure on the system caused by process data storage during the calculation process is relieved. The performance of the system is much improved. It can also be seen that the creative generation ability of the system has been maintained at a high level in the subsequent creative generation process.

The EEG waveform and the subject's creative label annotations reflect the popularity of the current creative system very well. As the tag knowledge is learned again, the structure of M-3DQKG tends to be reasonable. The next creative plan will still be marked by the subject's EEG test results and creative tags as one of the criteria. The loop of this process may require multiple iterations. In the process of system iteration, the information entropy of the system increases continuously and tends to be saturated. Therefore, CGC steadily approaches 100. When the creative generation results are satisfactory to the subjects, the current CGC is fixed as the system's creative emergence value. At this time, the dance creative M-3DQKG structure that addresses these needs is set. When the system faces new dance creation needs, the dance creativity can be simplified to observe whether the CGC value reaches the creative emergence value, and the process of EEG testing and creative labeling is omitted. Therefore, the whole process of dance creativity can be completed more

TABLE 5 | Change table of the knowledge graph and weights.

Label			Connection nodes						Entangled nodes
Name	A	A1	B3	B7	B10	B31			
W			0.203	0.515	0.126	0.156			
Name		A2	B2	B5	B12				C6
W			0.333	0.585	0.082				
Name		A3	B1	B3	B4	B6	B8	B11	
W			0.116	0.284	0.485	0.061	0.033	0.021	
Name	B	B1	C5	C8	C14	C19	C31		
W			0.513	0.089	0.114	0.065	0.219		
Name		B2	C1	C6	C11				D21
W			0.388	0.305	0.307				
Name		B3	C3	C4	C11	C15	C16		
W			0.096	0.167	0.355	0.219	0.163		
Name		B4	C12	C23					
W			0.698	0.302					
Name		B5	C4	C9	C14	C18	C24	C31	
W			0.205	0.611	0.034	0.142	0.005	0.003	
…					…				
Name		B11	C2	C7	C19	C39			F16
W			0.646	0.014	0.125	0.215			
Name		B12	C5	C11	C27	C32	C33		D54
W			0.352	0.109	0.262	0.135	0.142		
…					…				
Name		B16	C33	C36					
W			0.732	0.268					
Name	C	C1	D3	D12	D44	D47	D66		G78
W			0.455	0.198	0.117	0.102	0.128		
Name		C2	D2	D23	D52				
W			0.227	0.532	0.241				
Name		C3	D12	D16	D56	D71	D75	D81	E11
W			0.387	0.206	0.158	0.033	0.012	0.204	
…					…				

quickly and effectively. Because of the small number of subjects, the coincidence of EEG peaks between different subjects needs to be strengthened. The time points of some creative point tags obtained from the test may be optimized. Furthermore, in order to accelerate the speed and accuracy of creative generation iterations, the subject technology can be expanded, which is one of the important means of achieving this.

CONCLUSION AND DISCUSSION

This paper builds a creative generation process based on a quantum modeling simulation framework and proposes a method that can generate multiple creative schemes for dance movements. This method can realize the reasonable construction and reorganization of a quantum knowledge graph and assign knowledge retrieval weights according to creative needs dynamically. The method includes the M-3DQKG model, information fidelity method, the creative generation model, and the linear evaluation method of the creative ability. Furthermore, the learning results from the creative network are novel, useful, and diverse based on this method. The practicality of the dancing robot's movements and running trajectories can also be guaranteed. The results of the creative generation can be viewed through a computer simulation engine and VR devices so that people can feel immersed in the experience. For the results of the creative generation process, this paper proposes CGC to judge the creative generation ability of the system. Moreover, the evaluation result data are input to the dancing robot to realize the robot's dancing performance and trajectory calculation in a specific scene. In this paper, the robot data are generated by simulation technology, and the writing and debugging work is completed in the simulation engine. This is helpful in reducing the time cost and hardware loss of the dance robot's multiple runs and debugging, and can map the real robot in the model of the simulation engine through digital twin technology. The benefits of this are to facilitate the analysis and preservation of

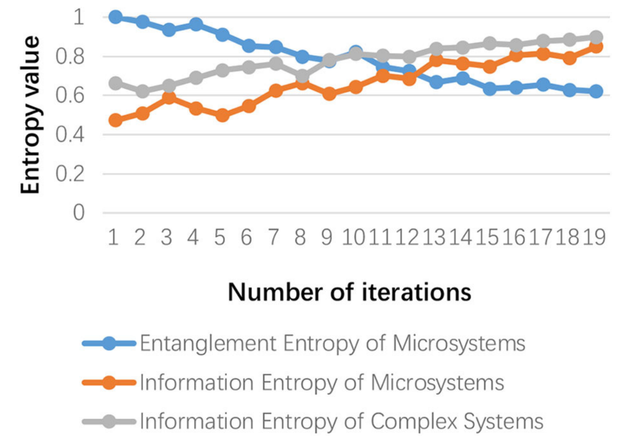


FIGURE 12 | Entanglement entropy, quantum system information entropy, and complex system information entropy.

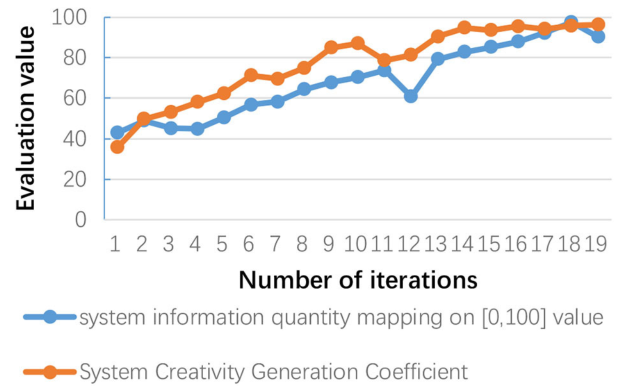


FIGURE 13 | Information volume and creative generation coefficient.

data and to ensure information fidelity and intelligent control of the dancing robot. At present, in terms of the performance of dance robots, we have found that, in the process of generating creative dance movements, the continuity of movements needs to be optimized by algorithms because the creative movements are fragmented and the dynamic model movements are rigid. The goal of the next stage is to be able to generate a complete set of coherent and smooth dance movements. In terms of the experimental data of the system, because of the small number of subjects, the EEG and creative label data are not sufficient, resulting in a slow system iteration speed, which affects the current generation effect of dance creativity. The next step will be to invest more research in the process of increasing test data and accelerating iteration.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

ETHICS STATEMENT

This study was reviewed and approved by the Medical Ethics Committee of Laizhou People's Hospital, 2020040700001. Written informed consent was obtained from all participants for their participation in this study.

AUTHOR CONTRIBUTIONS

PM was responsible for article writing and experimental work. GD directs the article structure and designs experiments. QJ conducts a lot of research work. FZ was the corresponding author and participates in the revision of the methodology. YJ was responsible for data analysis. All authors contributed to the article and approved the submitted version.

FUNDING

Special thanks to Beijing Wanshide Technology Co., Ltd. for its financial support for scientific research projects.

ACKNOWLEDGMENTS

We thank the supervisor for his guidance in the subject research. Thanks also to the 26 volunteers who took part in the test. We have promised to keep their information completely confidential, and the test data will only be used for the evaluation of the creative generation results. They will not be used for any other purposes.

REFERENCES

Aerts, D., Beltran, L., Geriente, S., and Sozzo, S. (2019). Quantum-theoretic modeling in computer science. *Int. J. Theor. Phys.* 1–17. doi: 10.1007/s10773-019-04155-y

Agnoli, S., and Corazza, G. E. (2019). "Emotions: the spinal cord of the creative thinking process," in *Dynamic Perspectives on Creativity. Creativity Theory and Action in Education*, *Vol. 4,* eds R. Beghetto and G. Corazza (Cham: Springer). doi: 10.1007/978-3-319-99163-4_4

Arima, T. (2015). Next generation em simulation technique. *Games* 4, 624–647.

Bennett, C. H., and DiVincenzo, D. P. (2000). Quantum information and computation. *Nature* 404, 247–255. doi: 10.1038/35005001

Berta, M., Wehner, S., and Wilde, M. M. (2015). Entropic uncertainty and measurement reversibility. *New J. Phys.* 18:7. doi: 10.1088/1367-2630/18/7/073004

Bilbao, S. (2009). *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*. Hoboken, NJ: Wiley.

Binder, K., and Heermann, D. W. (2014). *Monte Carlo Simulation in Statistical Physics.* Melville, NY: American Institute of Physics.

Biolek, D., Thompson, A. V., Kolka, Z., and Biolkova, V. (2019). Smoothing technique for simulation of switched-capacitor filters using general-purpose circuit simulation programs. *2019 Communication and Information Technologies (KIT)* (Slovakia: Vysoke Tatry) 1–5. doi: 10.23919/KIT.2019.8883307

Boyd, J. C., and Bruns, D. E. (2001). Quality specifications for glucose meters: assessment by simulation modeling of errors in insulin dose. *Clin. Chem.* 2:2. doi: 10.1093/clinchem/47.2.209

Bucklew, J. (2010). *Introduction to Rare Event Simulation.* Beijing: Springer Publishing Company, Incorporated.

Burdon, D., Barnard, S., Boyes, S. J., and Elliott, M. (2018). Oil and gas infrastructure decommissioning in marine protected areas: system complexity, analysis and challenges. *Mar. Pollut. Bull.* 135, 739–758. doi: 10.1016/j.marpolbul.2018.07.077

Burger, J., and Jenkins, T. (2013). Assessment of alternative strategic defense system support concepts with a simulation model. *Home Healthc. Nurse.* 16:7. doi: 10.1097/00004045-199807000-00012

Buyya, R., and Murshed, M. (2010). Gridsim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Concurr. Comput. Pract. Exp.* 14, 1175–1220. doi: 10.1002/cpe.710

Calignano, G., and Jsendal, K. (2018). Does the nature of interactions with higher education institutions influence the innovative capabilities of creative firms? The case of a south-western norwegian county. *Quaest. Geogr.* 37, 69–79. doi: 10.2478/quageo-2018-0040

Chernyavskiy, A. Y. (2009). The computable measure of quantum entanglement of multiqubit states. *Russ. Microelectr.* 38, 199–205. doi: 10.1134/S106373970903007X

Christensen, B. T., and Ball, L. J. (2016). Creative analogy use in a heterogeneous design team: the pervasive role of background domain knowledge. *Design Stud.* 46, 38–58. doi: 10.1016/j.destud.2016.07.004

Colella, H. V. (2011). Multi-event simulations of slow slip events for a cascadia-like subduction zone. *Geophys. Res. Lett.* 38, 239–255. doi: 10.1029/2011GL048817

Deng, X. L., Wang,. B., Wu, B., and Yang, S. Q. (2012). Modularity modeling and evaluation in community detecting of complex network based on information entropy. *J. Comput. Res. Dev.* 49, 725–734.

Diyanni, R. (2014). *The Pearson Guide to Critical and Creative Thinking.* Upper Saddle River, NJ: Pearson Education.

Edward, W. (2015). *Quantum field Theory and the Jones Polynomial.* Beijing: Braid Group, Knot Theory And Statistical Mechanics II.

Friedman, R. S., Förster, J., and Denzler, M. (2007). Interactive effects of mood and task framing on creative generation. *Creat. Res. J.* 19, 141–162. doi: 10.1080/10400410701397206

Gazoni, R. M. (2016). "Creative thinking in artificial intelligence: a peircean account," in *2016 International Conference on Computational Science and Computational Intelligence (CSCI)* (Las Vegas, NV). doi: 10.1109/CSCI.2016.0107

Giraud, M., Groult, R., and Florence, L. (2016). "Computational analysis of musical form," in *Computational Music Analysis*, ed D. Meredith (Cham: Springer). doi: 10.1007/978-3-319-25931-4_5

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial networks. *Adv. Neural Inf. Process. Syst.* 3, 2672–2680.

Gould, H., and Tobochnik, J. (2007). An introduction to computer simulation methods. *Comput. Phys.* 10, 652–653. doi: 10.1063/1.4822415

Graves, A., Mohamed, A. R., and Hinton, G. (2013). "Speech recognition with deep recurrent neural networks," in *IEEE International Conference on Acoustics. Speech and Signal Processing* (Vancouver, BC). doi: 10.1109/ICASSP.2013.6638947

Gyongyosi, L., and Imre, S. (2019). A survey on quantum computing technology. *Comput. Sci. Rev.* 31, 51–71. doi: 10.1016/j.cosrev.2018.11.002

Haijuan, Y., Jinxia, W., Qingli, S. U., and Xuejun, B. (2018). The effects of embodied emotion and creative thinking: situational regulatory focus as moderator. *Stud. Psychol. Behav.* 16, 441–448. doi: 10.3969/j.issn.1672-0628.2018.04.002

Han, M. Y., Gao, X. H., Su, J. Z., et al. (2001). Quantum-dot-tagged microbeads for multiplexed optical coding of biomolecules. *Nat. Biotechnol.* 19, 631–635. doi: 10.1038/90228

Han, Y., Zhang, P., Zhuo, T., Huang, W., and Zhang, Y. (2017). "Video action eecognition based on deeper convolution networks with pair-wise frame motion concatenation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Honolulu, HI). doi: 10.1109/CVPRW.2017.162

Heydari, H. (2006). Quantum entanglement measure based on wedge product. *Quan. Inform. Comput.* 6:2. doi: 10.1142/9789812773210_0010

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process. Magaz.* 29, 82–97. doi: 10.1109/MSP.2012.2205597

Hu, Y. P., Modat, M., Gibson, E., Li, W. Q., Ghavami, N., et al. (2018). Weakly-supervised convolutional neural networks for multimodal image registration. *Med. Image Anal.* 49, 1–13. doi: 10.1016/j.media.2018.07.002

Hugill, A. (2014). "Creative computing processes: musical composition," in *2014 IEEE 8th International Symposium on Service Oriented System Engineering* (Oxford). doi: 10.1109/SOSE.2014.80

IEEE B. E. (2010). "IEEE standard for modeling and simulation (M&S) high level architecture (HLA)– framework and rules - redline," in *IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000) - Redline* (New York, NY).

Jaber, R. M., Abu-Hassan, H. H., Raie, J. A., Tayseer, A., and Massad, I. (2019). Merging drama with medical education: simulation in learning breaking bad news pre- and post-study. *Open J. Nurs.* 9, 620–628. doi: 10.4236/ojn.2019.97050

Jami, S., and Labbafi, Z. (2017). Entanglement dynamics in two-mode gaussian systems. *Indian J. Phys.* 91, 391–401. doi: 10.1007/s12648-016-0931-9

Jayantika, I. G. A. N. T., Parmithi, N. N., and Dyanawati, N. P. A. (2019). Quantum teaching learning model as solution to improve learning activity and mathematics learning outcome. *J. Phys. Conf. Ser.* 1321:022119. doi: 10.1088/1742-6596/1321/2/022119

Jha, P. K., Shekhar, C., and Kumar, L. S. (2015). Visual simulation application for hardware in-loop simulation (hils) of aerospace vehicles. *Adv. Intell. Syst. Comput.* 337, 473–480. doi: 10.1007/978-3-319-13728-5_53

Jing, D., Yang, H., and Tian, Y. (2013). "Abstraction based domain ontology extraction for idea creation," in *13th International Conference on Quality Software* (Najing). doi: 10.1109/QSIC.2013.31

Jonason, P. K., Richardson, E. N., and Potter, L. (2015). Self-reported creative ability and the dark triad traits: an exploratory study. *Psychol. Aesth. Creat. Arts* 9, 485–487. doi: 10.1037/aca0000037

Kasztenny, B., and Kezunovic, M. (2002). A method for linking different modeling techniques for accurate and efficient simulation. *IEEE Trans. Power Syst.* 15, 65–72. doi: 10.1109/59.852102

Kim, M. (2016). "Importance of creative thinking for paradigm shifts that foster scientific advances," in *Interplay of Creativity and Giftedness in Science. Advances in Creativity and Giftedness,* eds M. K. Demetrikopoulos and J. L. Pecore (Rotterdam: SensePublishers). doi: 10.1007/978-94-6300-163-2_2

Kitaev, A. Y., Shen, A. H., and Vyalyi, M. N. (2011). Classical and quantum computation. *Am. Math. Monthly.* 47:257. doi: 10.2307/3647986

Knill, E., Laflamme, R., and Milburn, G. J. (2001). A scheme for efficient quantum computation with linear optics. *Nature* 409, 46–52. doi: 10.1038/35051009

Lee, K. C. (2016). *Digital Creativity Model and Its Relationship with Corporate Performance*. Zurich: Springer

Li, M., Fei, S. M., and Li, J. X. (2010). Quantum entanglement: separability, measure, fidelity of teleportation, and distillation. *Adv. Math. Phys.* 2010:301072. doi: 10.1155/2010/301072

Li, Y., Wang, J., Li, X. L., Zhao, W., and Hu, W. (2003). Creative thinking and computer aided product innovation. *Comput. Integr. Manuf. Syst.* 9, 1092–1096. doi: 10.3969/j.issn.1006-5911.2003.12.009

Liang, B. C., Wang, H. L., and Zheng, Y. H. (2001). Military simulation technology development trend and prospects. *Acta Simulata Systemat. Sinica.* 13, 18–21.

Lin, S. J., Chao, M. H., Lee, C. Y., and Yang, C. S. (2016). Human action recognition using motion history image based temporal segmentation. *Int. J. Pattern Recogn. Artif. Intell.* 30, 36–46. doi: 10.1142/S021800141655017X

Liu, R., and Yang, H. (2014). "Chaos and fractal for creative computing," in *2014 IEEE 8th International Symposium on Service Oriented System Engineering* (Oxford). doi: 10.1109/SOSE.2014.83

Ma, H., Yao, Y., and Tang, W. (2016). "Warship reusable component model development approach for parallel and distributed simulation," in *Asian Simulation Conference Scs Autumn Simulation Multi-conference*. Springer.

Machado, A. F. A., Costa, F. A. C. R., and Rezende, J. L. D. (2015). "Use of simulation to achieve better results in cyber military training," in *MILCOM 2015 - 2015 IEEE Military Communications Conference.* Tampa, FL: IEEE. doi: 10.1109/MILCOM.2015.7357620

Madhok, V., Dogra, S., and Lakshminarayan, A. (2018). Quantum correlations as probes of chaos and ergodicity. *Optics Commun.* 420, 189–193. doi: 10.1016/j.optcom.2018.03.069

Matta, F. (2008). Video person recognition strategies using head motion and facial appearance. *Bibliogr.* Available online at: https://theses.eurasip.org/theses/196/video-person-recognition-strategies-using-head/

Matthias, M., Marshall, L., Wolf, B., Fehm, H. L., and Born, J. (2010). Eeg complexity and performance measures of creative thinking. *Psychophysiology* 36, 95–104. doi: 10.1017/S0048577299961619

Merién, A. E. R., van de Ven, J., Mol, B. W., Houterman, S., and Oei, S. G. (2010). Multidisciplinary team training in a simulation setting for acute obstetric emergencies. *Obstetr. Gynecol.* 115, 1021–1031. doi: 10.1097/AOG.0b013e3181d9f4cd

Mintert, F., and Buchleitner, A. (2007). Observable entanglement measure for mixed quantum states. *Phys.rev. Lett.* 9:140505. doi: 10.1103/PhysRevLett.98.140505

Moin, P. (2018). "Numerical simulation of turbulent flows," in *APS March Meeting 2018.* Los Angeles, CA: American Physical Society.

Morrison, K. (2003). Complexity theory and curriculum reforms in Hong Kong. *Pedag. Cult. Soc.* 11, 279–302. doi: 10.1080/14681360300200174

Muckenhirn, H., Magimai-Doss, M., and Marcel, S. (2017). "End-to-end convolutional neural network-based voice presentation attack detection," in *IEEE International Joint Conference on Biometrics.* Denver, CO: IEEE. doi: 10.1109/BTAS.2017.8272715

Mumford, M. D., and Mcintosh, T. (2017). Creative thinking processes: the past and the future. *J. Creat. Behav.* 51, 317–322. doi: 10.1002/jocb.197

Mustafee, N., Brailsford, S., Djanatliev, A., Eldabi, T., and Tolk, A. (2018). "Purpose and benefits of hybrid simulation: contributing to the convergence of its definition," *2017 Winter Simulation Conference (WSC)*, (Las Vegas, NV), 1631–1645. doi: 10.1109/WSC.2017.8247903

Nalewajski, R. F. (2016). Complex entropy and resultant information measures. *J. Math. Chem.* 54, 1777–1782. doi: 10.1007/s10910-016-0651-6

Nalewajski, and Roman, F. (2017). Quantum information measures and their use in chemistry. *Curr. Phys. Chem.* 7:2. doi: 10.2174/1877946806666160622075208

Neill, C., Roushan, P., Fang, M., Chen, Y., Kolodrubetz, M., and Chen, Z., et al. (2016). Ergodic dynamics and thermalization in an isolated quantum system. *Nat. Phys.* 12, 1037–1041. doi: 10.1038/nphys3830

Ness, R. B. (2012). *Innovation Generation: How to Produce Creative and Useful Scientific Ideas.* Oxford: Oxford University Press.

Niedenthal, P. M., Mermillod, M., Maringer, M., Hess, U. (2010). The simulation of smiles (SIMS) model: embodied simulation and the meaning of facial expression. *Behav. Brain Sci.* 33, 447–448. doi: 10.1017/S0140525X10000865

Nielsen, M. A., and Chuang, I. L. (2007). Quantum computation and quantum information. *Math. Struct. Comput. Sci.* 17, 1115–1115. doi: 10.1017/S0960129507006317

Oliver, A. L., April, S., and Kalish, Y. (2019). Organizational creativity-innovation process and breakthrough under time constraints: mid-point transformation. *Creat. Innov. Manag.* 28, 318–328. doi: 10.1111/caim.12326

Pezzulo, G., Candidi, M., Dindo, H., and Barca, L. (2013). Action simulation in the human brain: twelve questions. *New Ideas Psychol.* 31, 270–290. doi: 10.1016/j.newideapsych.2013.01.004

Plenio, M. B., and Virmani, S. (2011). An introduction to entanglement measures. *Quan. Inform. Comput.* 7, 1–51.

Prakash, H. (2010). Quantum teleportation. *Theor. Math. Phys.* 84, 1021–1029. doi: 10.1007/s12648-010-0096-x,

Rapaport, D. C., Blumberg, R. L., Mckay, S. R., and Christian, W. (2002). The art of molecular dynamics simulation. *Comput. Sci. Eng.* 1, 70–71. doi: 10.1109/5992.743625

Rivas, P. G. (2017). Strategies for teaching and dissemination of artistic heritage by promoting critical and creative thinking among future primary education teachers. *Proc. Soc. Behav. Sci.* 237, 717–722. doi: 10.1016/j.sbspro.2017.02.112

Robinson, K. (2011). *Out of our Minds: Learning to be Creative*. Mankato, MN: Wiley & Sons, 232–234.

Rooij, A. D., Corr, P. J., and Jones, S. (2017). "Creativity and emotion: enhancing ereative thinking by the manipulation of computational feedback to determine emotional intensity," in *2017 Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*. New York, NY: Association for Computing Machinery. doi: 10.1145/3059454.3059469

Roux, N. L., Amar, M., Moreau, A., and Fossier, P. (1973). A general purpose computer program for the dynamic simulation of vehicle-guideway interactions. *AIAA J.* 11, 278–282. doi: 10.2514/3.50469

Roy, S., Petersen, I. R., and Huntington, E. H. (2017). Coherent-classical estimation for linear quantum systems. *Automatica* 82, 109–117. doi: 10.1016/j.automatica.2017.04.034

Rozenblit, J. W. (2015). Experimental frame specification methodology for hierarchical simulation modeling. *Int. J. Gen. Syst.* 19, 317–336. doi: 10.1080/03081079108935180

Sana, S., Mohammad, A. B., Kamran, S., Iqbal, N. T., Moskaluk, C. A., Asad, A., et al. (2018). Convolutional neural networks image analysis of duodenal biopsies robustly distinguishes environmental enteropathy from healthy controls and identifies secretory cell lineages as high activation locations. *Gastroenterology* 154:S52. doi: 10.1016/S0016-5085(18)30638-3

Santipanusopon, S., and Worawattanaparinya, S. (2019). Simulation model for commercial innovation management guideline in industrial business for thailand competitiveness in global market. *Acad. Strateg. Manag. J.* 18

Saorín, J. L., Bonnet, D. M., Meier, C. A., and Cantero, J. D. L. T. (2017). Makerspace teaching-learning environment to enhance creative competence in engineering students. *Think. Skills Creat*. 23, 188–198. doi: 10.1016/j.tsc.2017.01.004

Schramm, S. (2017). Computer science: data analysis meets quantum physics. *Nature* 550, 339–340. doi: 10.1038/550339a

Shamay-Tsoory, S. G., Adler, N., Aharon-Peretz, J., Perry, D., and Mayseless, N. (2011). The origins of originality: the neural bases of creative thinking and originality. *Neuropsychologia* 49, 178–185. doi: 10.1016/j.neuropsychologia.2010.11.020

Shota, Y., Nicola, D. P., Takahiro, S., et al. (2017). The quantum entanglement of measurement. *Front. Optics.* doi: 10.1364/FIO.2017.FTh3E.6

Simonton, D. K. (2012). Creative thought as blind variation and selective retention: why creativity is inversely related to sightedness. *J. Theor. Philos. Psychol.* 33, 253–266. doi: 10.1037/a0030705

Sjöström, P. J., Frydel, B. R., and Wahlberg, L. U. (2015). Artificial neural network-aided image analysis system for cell counting. *Cytometry Part B Clin. Cytometry* 36, 18–26. doi: 10.1002/(SICI)1097-0320(19990501)36:1<18::AID-CYTO3>3.0.CO;2-J

Steele, L. M., Johnson, G., and Medeiros, K. E. (2018). Looking beyond the generation of creative ideas: confidence in evaluating ideas predicts creative outcomes. *Pers. Ind. Diff.* 125, 21–29. doi: 10.1016/j.paid.2017.12.028

Taheripour, S., Saffarian, M. R., and Daneh-Dezfuli, A. (2019). Heat transfer simulation in an industrial journal bearing using vof method. *J. Braz. Soc. Mech. Ences Eng.* 41:248. doi: 10.1007/s40430-019-1751-6

Wang, S. Y., Wang, O., Zhang, R., Owens, A., and Efros, A. A. (2019). "Cnn-generated images are surprisingly easy to spot… for now", in *Computer Vision and Pattern Recognition (CVPR)* (Washington). doi: 10.1109/CVPR42600.2020.00872

William, J. M., and Ralph, W. C. (2015). The new mathematics and logical reasoning and creative thinking abilities. *School Sci. Math.* 68, 731–733. doi: 10.1111/j.1949-8594.1968.tb08341.x

Xu, K., Qin, M., Sun, F., Wang, Y., Chen, Y. K., and Ren, F. (2020). "Learning in the frequency domain", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA: IEEE). doi: 10.1109/CVPR42600.2020.00181

Yamaguchi, T., Kaga, T., Donzé, A., and Seshia, S. (2016). "Combining requirement mining, software model checking and simulation-based verification for industrial automotive systems", in *2016 Formal Methods in Computer-Aided Design (FMCAD)* (Mountain View, CA), 201–204. doi: 10.1109/FMCAD.2016.7886680

Yingying, Z., Luhui, W., Yanan, W., and Yafei, D. (2018). A non-label and enzyme-free sensitive detection method for thrombin based on simulation-assisted dna assembly. *Sensors* 18:2179. doi: 10.3390/s18072179

Yu, C. S., and Song, H. S. (2004). Free entanglement measure of multiparticle quantum states. *Phys. Lett. A* 330, 377–383. doi: 10.1016/j.physleta.2004.07.054

Zeigler, B. P., Kim, T. G., Praehofer, H. (2000). *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. San Francisco, CA: Academic Press.

Zhang, F. Q., Mao, Z., Ding, G., Zhang, R., and Lai, J. (2016b). "Idea creation of creative computing based on embedding features and ontology," *The Multidisciplinary International Social Networks Conference on Social informatics 2016,* Data Science, ACM. doi: 10.1145/2955129.2955191

Zhang, H. G., Mao, S. W., Wu, W. Q., Wu, S. M. (2016a). Overview of quantum computation complexity theory. *Chin. J. Comput.* 39, 2403–2428. doi: 10.11897/SP.J.1016.2016.02403

Zhang, F., Ding, G., Ma, L., Zhu, Y., and Xu, L. (2018). "Research on stage creative scene model generation based on series key algorithms," in *Advances in Smart Vehicular Technology, Transportation, Communication and Applications. VTCA 2018. Smart Innovation, Systems and Technologies*, *Vol 128.* eds Y. Zhao, T. Y. Wu, T. H. Chang, J. S. Pan, and L. Jain (Cham: Springer), 170–177. doi: 10.1007/978-3-030-04585-2_20

Zheng, D. Z., Zheng, L., and Yang, Y. (2017). Unlabeled samples generated by gan improve the person re-identification baseline *in vitro*. *ICCV* 2017, 3774–3782. doi: 10.1109/ICCV.2017.405

Zhou, L., Sun, M. F., Peng, Z. Y., and Yuan, S. J. (2015). "Architecture and simulation of US Military Tactical Internet", in *Thesis of School of Communication & Information Engineering, Shanghai University, Luoyang Electronic Equipment Test Center*. Shanghai: Communications Technology. Available online at: http://en.cnki.com.cn/Article_en/CJFDTotal-TXJS201510013.htm

Zipfel, P., and Schiehlen, W. (2001). Modeling and simulation of aerospace vehicle dynamics. *Appl. Mech. Rev*. 54, B101–B102. doi: 10.1115/1.1421113

Conflict of Interest: YJ was employed by the company Beijing Wanshide Technology Co.

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: 03 December 2020
doi: 10.3389/fnbot.2020.577749

PD Control Compensation Based on a Cascade Neural Network Applied to a Robot Manipulator

Luis Arturo Soriano[1]*, Erik Zamora[2], J. M. Vazquez-Nicolas[3], Gerardo Hernández[2], José Antonio Barraza Madrigal[4] and David Balderas[5]

[1] Departamento de Ingeniería Mecánica Agrícola, Universidad Autónoma Chapingo, Texcoco, Mexico, [2] Laboratorio de Robótica y Mecatrónica, Instituto Politécnico Nacional, CIC, Ciudad de México, Mexico, [3] Unidad Mixta Internacional, French-Mexican Laboratory of Informatics and Automatic Control, 3175 French National Research Council, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Deparment of Control of Dynamic Systems, Ciudad de México, Mexico, [4] Unidad Profesional Adolfo López Mateos, Escuela Superior de Ingeniería Química e Industrias Extractivas del Instituto Politécnico Nacional, Academia de Física, Ciudad de México, Mexico, [5] Tecnologico de Monterrey, Escuela de Ingeniería y Ciencias, Ciudad de México, Mexico

OPEN ACCESS

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

Reviewed by:
Heng Liu,
Guangxi University for Nationalities,
China
Huiming Wang,
Chongqing University of Posts and
Telecommunications, China

***Correspondence:**
Luis Arturo Soriano
lsorianoa@chapingo.mx

Received: 29 June 2020
Accepted: 14 September 2020
Published: 03 December 2020

Citation:
Soriano LA, Zamora E,
Vazquez-Nicolas JM, Hernández G,
Barraza Madrigal JA and Balderas D
(2020) PD Control Compensation
Based on a Cascade Neural Network
Applied to a Robot Manipulator.
Front. Neurorobot. 14:577749.
doi: 10.3389/fnbot.2020.577749

A Proportional Integral Derivative (PID) controller is commonly used to carry out tasks like position tracking in the industrial robot manipulator controller; however, over time, the PID integral gain generates degradation within the controller, which then produces reduced stability and bandwidth. A proportional derivative (PD) controller has been proposed to deal with the increase in integral gain but is limited if gravity is not compensated for. In practice, the dynamic system non-linearities frequently are unknown or hard to obtain. Adaptive controllers are online schemes that are used to deal with systems that present non-linear and uncertainties dynamics. Adaptive controller use measured data of system trajectory in order to learn and compensate the uncertainties and external disturbances. However, these techniques can adopt more efficient learning methods in order to improve their performance. In this work, a nominal control law is used to achieve a sub-optimal performance, and a scheme based on a cascade neural network is implemented to act as a non-linear compensation whose task is to improve upon the performance of the nominal controller. The main contributions of this work are neural compensation based on a cascade neural networks and the function to update the weights of neural network used. The algorithm is implemented using radial basis function neural networks and a recompense function that leads longer traces for an identification problem. A two-degree-of-freedom robot manipulator is proposed to validate the proposed scheme and compare it with conventional PD control compensation.

Keywords: cascade neural networks, robot manipulator, PD control, radial basis function, control compensation

1. INTRODUCTION

An industrial robot manipulator frequently works at high velocities to reach its desired position. Common tasks performed by robot manipulators include trajectory tracking, reaching positions, and picking and dropping objects. These tasks need the robot controllers to maintain satisfactory dynamic behavior in spite of possible external perturbations, unknown dynamic parameters, and sensor information loss (Armendariz et al., 2014). Several controllers that are often implemented to manage these features are also mentioned (Luo and Kuo, 2016; Makarov et al., 2016; Nicolis et al., 2016; Pan et al., 2018; Hwang and Yu, 2020). Over time, the Proportional Integral Derivative

(PID) control has been used to design industrial robots due to their simple structure and simple hardware implementation. However, during operation, the PID integral gain provokes the controller to reduce its bandwidth and stability (Rahimi Nohooji, 2020). PD control with uncertainty compensation has been proposed to manage the increase of integral gain due to the steady-state error. The PD controllers are also limited without gravity compensation, which requires a dynamic model (Wen Yu and Rosen, 2013). In practice, the non-linearities of a dynamic robot system are generally unknown.

To solve this issue, different approaches have been developed in order to compensate for unmodeled uncertainties (i.e., noise, gravity, and friction). Intelligent compensation is a model free and it has been applied to well-known algorithms such as the neural networks (NNs) and fuzzy logic (FL) (Krishna and Vasu, 2018; Wang et al., 2019b). In Liu et al. (2020), the authors propose an adaptive NN backstepping control design for fractional-order non-linear systems with actuator faults whose parameters and patterns are fully unknown. Baek et al. (2016) present an adaptive sliding mode control scheme that implements the time-delay estimation. In Xu et al. (2018), a fuzzy NN sliding mode control is designed to improve controller performance against system uncertainty and external disturbances. Kumar et al. (2012) proposed a hybrid trajectory tracking controller for redundant robot manipulators. The adaptive controller is implemented to estimate unstructured uncertainties and error reconstruction. In He et al. (2018), one Radial Basis Function Neural Network (RBFNN) is used to estimate the unknown dynamics robotic manipulator. Jung and Hsia (2000) proposed two NN control schemes for a non-model-based robot manipulator, which show advantages over feedback error learning. In Zhang et al. (2018), the authors proposed a gravity compensation based on an RBFNN and robustness analysis, and the results were compared with a classic PID and PD with fixed gravity compensation. Gandolfo et al. (2019) propose a control scheme that combines a classical PD and a robust adaptive compensator based on NNs.

Although adaptive controllers are addressed for systems with non-linear and uncertainties dynamics, thus their slow convergence can lead to performance degradation or even affect operational safety. In Liu et al. (2019), an adaptive NN control with optimal number of hidden nodes and less computation is formulated for approximating the trajectory of robot manipulator. Similarly, Yang et al. (2018) develop a control and identification scheme in order to identify the unknown robot parameters with an enhanced convergence rate. Another approach is to relax the linear parameterized assumption and the requirements of system knowledge, thus, NNs have been used as function approximators. In time series modeling, RBFNN is commonly used for function approximation, since its value is different from zero in infinite space, and its approximation can avoid the local minimum (Wang et al., 2019a). An RBFNN uses a Gauss function as its activation function. In general, RBFNN controllers waste less computational resources in comparison to other NN controllers (He et al., 2018). In Wang et al. (2012), the authors proposed an RBFNN to compensate for non-linear dynamics of the robotic manipulator and a robust control designed to suppress the modeling error of NN.

However, update laws commonly increase the weight magnitudes until the output error has been mitigated, without a robust design continued training can lead to excessive control effort. In order to avoid this, adaptive controls frequently update the neural weights according to robust adaptive laws, which are computed with Lyapunov methods (Razmi and Macnab, 2020). In this work, a robust adaptive control design to compensate a nominal controller for robot manipulator with uncertainties and external perturbations is formulated. Moreover, a scheme based on two RBFNN in cascade is proposed in order to improve the response of the nominal controller. In the scheme aforementioned, the first NN is used to estimate the error and the second uses the estimation error value to improve the output of the nominal controller. NN weights are online updated by developing new adaptive laws. The adaptive law based on the gradient is modified by introducing a recompense function of the online error in order to improve the convergence of the NN weights.

In this work, a PD control with a scheme based on NNs in cascade is designed to manage the compensation of uncertainties in a robot manipulator. The main contributions of this paper are summarized as follows:

- In order to improve the robustness of the system against external disturbance, and unknown system parameters, a scheme of cascade NNs is proposed.
- A recompense function for neural weights updates is proposed in order to improve the NNs' weights convergence.
- The response of the nominal controller is improved.

To validate the proposed scheme and compare it with conventional neural compensation, a two-degree-of-freedom robot manipulator (TDOFRM) is proposed. This paper is organized as follows: Section 2 presents the preliminary mathematical model of the TDOFRM, the conventional PD compensation with RBFNN. Section 3 describes the controller design based on NNs in cascade. Section 4 presents simulation experiments and compared with the conventional compensation, and finally, section 5 presents the conclusions.

2. ADAPTIVE CONTROL TO ROBOTIC MANIPULATOR

2.1. Dynamic Model of Robotic Manipulator

The dynamic model of an n degree of freedom robot manipulator can be described as follows (Spong and Vidyasagar, 1989):

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau + d \tag{1}$$

$M(q)$ is a $n \times n$ inertia matrix, $C(q,\dot{q})$ is a $n \times n$ is a centrifugal and Coriolis matrix, and $g(q)$ is a $n \times 1$ vector of gravity. $q, \dot{q}, \ddot{q}$ are the position, velocity, and acceleration of each link, respectively. $\tau \in R^n$ is the control input and d denotes disturbances.

2.2. Proportional Derivative Control Scheme

In industrial application, the exact model is difficult to obtain and external disturbances are always present in practice. According to Liu (2013), a nominal model of robot manipulator can be computed as $M_0(q)$, $C_0(q,\dot{q})$, and $g_0(q)$. Considering $\Delta M = M_0 - M$, $\Delta C = C_0 - C$, and $\Delta g = g_0 - g$, Equation (1) is reordered as follows:

$$(M_0(q) - \Delta M)\ddot{q} + (C_0(q,\dot{q}) - \Delta C)\dot{q} + (g_0(q) - \Delta g) = \tau + d \quad (2)$$

Thus,

$$M_0(q)\ddot{q} + C_0(q,\dot{q})\dot{q} + g_0(q) = \tau + \Delta M\ddot{q} + \Delta C\dot{q} + \Delta g + d \quad (3)$$

Defining $f(\cdot) = \Delta M\ddot{q} + \Delta C\dot{q} + \Delta g + d$ and if $f(\cdot)$ is known, the control law is designed as

$$M_0(q)(\ddot{q_d} - \kappa_v\dot{e} - \kappa_p e) + C_0(q,\dot{q})\dot{q} + g_0(q) - f(\cdot) = \tau \quad (4)$$

Submitting Equation (4) into Equation (3), the close loop system can be expressed as follows:

$$\ddot{e} + \kappa_v\dot{e} + \kappa_p e = 0 \quad (5)$$

where $e = q - q_d$, $\dot{e} = \dot{q} - \dot{q}_d$, and $\ddot{e} = \ddot{q} - \ddot{q}_d$. Frequently, $f(\cdot)$ in industrial applications is unknown, hence, $f(\cdot)$ requires to be estimated and compensated.

2.3. Radial Basis Function Neural Network Approximation

The NNs approximates $M(q)$, $C(q)$, and $g(q)$ when they are unknown. The Radial Basis Function (RBF) algorithm can approximate a continuous function and it is defined as

$$\phi_i = exp^{\left(-\frac{\|x-c_i\|^2}{b_i^2}\right)}, i = 1, 2, \ldots, n \quad (6)$$

$$y = W\phi \quad (7)$$

where x is the input vector, $\phi = [\phi_1, \phi_2, \ldots, \phi_n]$ is the output of the Gaussian function, y is the output of the NN, W is the weight values matrix, and c_i is the center and b_i is the width of the Gaussian function. In Gandolfo et al. (2019), Liu et al. (2019), it has been shown that an RBFNN can approximate a non-linear function $f(\cdot)$ under the following assumptions

1. The output $\hat{f}(x, W^*)$ is a continuous function.
2. Given a small positive constant ϵ_0 and a continuous function $f(\cdot)$, a weight vector W^* exists so that $\hat{f}(\cdot)$ satisfies.

$$max\|f(\cdot) - \hat{f}^*(\cdot)\| \leq \varepsilon_0 \quad (8)$$

and $W^* = \underset{W\in\beta(M_W)}{\text{argmin}} \left\{ \underset{W\in\beta(M_W)}{\sup} \|f(\cdot) - \hat{f}^*(\cdot)\| \right\}$, where W^* is $n \times n$ matrix that denotes the optimal weigh values for $f(\cdot)$ approximation.

2.4. Adaptive Law to Compensation Control

In the controller scheme proposed (Feng, 1995), the close loop system is given by the following equation:

$$\tau = M_0(q)(\ddot{q_d} - \kappa_v\dot{e} - \kappa_p e) + C_0(q,\dot{q})\dot{q} + g_0(q) - \hat{f}(\cdot) \quad (9)$$

where $\hat{f}(\cdot) = \hat{W}\phi(x)$ and $\hat{W}$ is an estimation of W^*. Equations (1) and (9) have the same term, and the substitution result is shown in the following equation:

$$\begin{aligned} &M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) - d \\ &\quad = M_0(q)(\ddot{q_d} - \kappa_v\dot{e} - \kappa_p e) + C_0(q,\dot{q})\dot{q} + g_0(q) - \hat{f}(\cdot) \quad (10)\end{aligned}$$

Then, the equation $M_0(q)\ddot{q} + C_0(q,\dot{q})\dot{q} + g_0(q)$ is subtracted with Equation (10) in both sides as follows:

$$\begin{aligned} &M_0(q)\ddot{q} + C_0(q,\dot{q})\dot{q} + g_0(q) - \left[M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) - d\right] \\ &\quad = M_0(q)\ddot{q} + C_0(q,\dot{q})\dot{q} + g_0(q) - \left[M_0(q)(\ddot{q_d} - \kappa_v\dot{e} - \kappa_p e)\right. \\ &\quad \left. + C_0(q,\dot{q})\dot{q} + g_0(q) - \hat{f}(\cdot)\right] \quad (11)\end{aligned}$$

The result is given as follows:

$$\begin{aligned} &M_0^{-1}(q)\left[\Delta M\ddot{q} + \Delta C\dot{q} + \Delta g(q) + d\right] \\ &\quad = \ddot{e} + \kappa_v\dot{e} + \kappa_p e + M_0^{-1}(q)\left[\hat{f}(\cdot)\right] \quad (12)\end{aligned}$$

Equation (12) can be rewritten as follows:

$$\ddot{e} + \kappa_v\dot{e} + \kappa_p e = M_0^{-1}(q)\left[f(\cdot) - \hat{f}(\cdot)\right] \quad (13)$$

Select to $x = (\,e\ \ \dot{e}\,)^T$. Equation (13) turns into

$$\dot{x} = Ax + B(f(\cdot) - \hat{f}(\cdot)) \quad (14)$$

where

$$A = \begin{pmatrix} 0 & I \\ -\kappa_p & -\kappa_v \end{pmatrix}; B = \begin{pmatrix} 0 \\ M_0^{-1}(q) \end{pmatrix} \quad (15)$$

Setting $f(\cdot) - \hat{f}(\cdot) = f(\cdot) - \hat{f}^*(\cdot) + \hat{f}^*(\cdot) - \hat{f}(\cdot) = \zeta + W^{*T}\phi - \hat{W}^T\phi = \zeta - \tilde{W}^T\phi$ and $\tilde{W} = \hat{W} - W^*, \zeta = f(\cdot) - \hat{f}^*(\cdot)$, where ζ denotes the modeling error due to the use of the NN. The modeling error ζ is bounded by a finite constant ζ_0, where $\zeta_0 = \underset{t\geq 0}{\sup}\|f(\cdot) - \hat{f}^*(\cdot)\|$. Finally,

$$\dot{x} = Ax + B(\zeta - \tilde{W}^T\phi) \quad (16)$$

The Lyapunov function is given by the following equation:

$$L = \frac{1}{2}x^T Px + \frac{1}{2\varphi}\|\tilde{W}\|^2; \varphi > 0 \quad (17)$$

where $\tilde{W} = \hat{W} - W^*$ is a definition that describes the estimation error. In Equation (17), P is a positive definite matrix that satisfies the Lyapunov equation

$$PA + A^T P = -Q; Q \geq 0 \quad (18)$$

Equation (17) can be rewritten in terms of the next definition

$$\| \tilde{W} \|^2 = \sum_{i,j} |w_{ij}|^2 = tr(\tilde{W}\tilde{W}^T) = tr(\tilde{W}^T\tilde{W}) \tag{19}$$

Thus, the derivative of V is given as follows:

$$\dot{L} = \frac{1}{2}[x^T P\dot{x} + \dot{x}^T Px] + \frac{1}{\varphi} tr(\dot{\tilde{W}}^T \tilde{W}) \tag{20}$$

Substituting Equation (16) for (20), the results is given by

$$\begin{aligned}
\dot{L} \; &= \; \frac{1}{2}[x^T P(Ax + B(\zeta \, - \, \tilde{W}^T\phi)) \\
&\quad +(Ax + B(\zeta \, - \, \tilde{W}^T\phi))^T Px] + \frac{1}{\varphi} tr(\dot{\tilde{W}}^T \tilde{W}) \\
&= \; \frac{1}{2}[x^T PAx + x^T PB(\zeta \, - \, \tilde{W}^T\phi) + x^T A^T Px) \\
&\quad +(\zeta \, - \, \tilde{W}^T\phi)^T B^T Px] + \frac{1}{\varphi} tr(\dot{\tilde{W}}^T \tilde{W}) \\
&= \; \frac{1}{2}[x^T PAx + x^T A^T Px + x^T PB(\zeta \, - \, \tilde{W}^T\phi)) \\
&\quad +(\zeta \, - \, \tilde{W}^T\phi)^T B^T Px] + \frac{1}{\varphi} tr(\dot{\tilde{W}}^T \tilde{W}) \\
&= \; \frac{1}{2}[x^T (PA + A^T P)x + x^T PB\zeta \, - x^T PB\tilde{W}^T\phi) \\
&\quad +\zeta^T B^T Px - \phi^T \tilde{W} B^T Px] + \frac{1}{\varphi} tr(\dot{\tilde{W}}^T \tilde{W}) \\
&= \; -\frac{1}{2}[x^T Qx] + \zeta^T B^T Px - B^T \tilde{W}\phi^T Px + \frac{1}{\varphi} tr(\dot{\tilde{W}}^T \tilde{W})(21)
\end{aligned}$$

Considering that $x^T PB\zeta \; = \; \zeta^T B^T Px$, $x^T PB\tilde{W}^T\phi \; = \; \phi^T \tilde{W} B^T Px$ and $\phi^T \tilde{W} B^T Px = tr[B^T Px\phi^T \tilde{W}]$ in Equation (21), this results in

$$\dot{L} = -\frac{1}{2}[x^T Qx] + \zeta^T B^T Px + \frac{1}{\varphi} tr(-\varphi B^T Px\phi^T \tilde{W} + \dot{\tilde{W}}^T \tilde{W}) \tag{22}$$

2.5. Adaptive Control Based on Cascade Neural Network

The PD control has been widely implemented in robot control to deal with the drawbacks presented by the integer gain in PID control. However, the PD control can have similar deficiencies if the derivative gain has high values (Wen Yu and Rosen, 2013). The PD control with compensation presents positive results to avoid high derivative gains and identify uncertainties that occur in the real operation of robot manipulators. In this work, a compensation of the nominal controller is proposed. The adaptive control scheme is shown in **Figure 1**. Two RBFNN in cascade are proposed to deal with the tracking error and the NN weight estimation error.

2.5.1. NN Weight Estimation Error

The input of NN is an error vector defined as follows:

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} q_i - q_{\vartheta i} \\ \dot{q}_i - \dot{q}_{\vartheta i} \end{bmatrix} \tag{23}$$

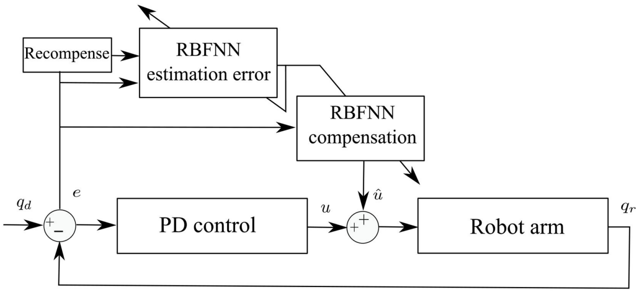


FIGURE 1 | Proposed adaptive control scheme.

where i and ϑ denote the real and desired position and velocity of the n-link of the robot. The NN estimation error anticipates which action to take in order to improve the output of the nominal control. Thus, the prediction of estimation error is also given as the output of RBFNN as follows:

$$\hat{x} = \sum_{j=1}^{J} W_e \phi_e \tag{24}$$

where ϕ_e is the Gaussian function given by Equation (6). The criterion for the weight update is proposed in the following equation:

$$\dot{W}_e = \gamma_e r \phi_e(x)^T PB \tag{25}$$

where $r \; = \| e^{-x_e^2} \|$ is a function that represents the recompense signal. The goal of the recompense function is to lead longer traces for an identification problem.

2.5.2. Adaptive Law to Compensate the Nominal Controller

A novel adaptive control scheme is proposed to ensure that the output of the nominal controller for the system defined in Equation (1) reaches the position desired, and the estimated NN can converge an ideal weight. An RBFNN is selected to approximate the system dynamics and deals with uncertainties and external disturbances, which is given as follows:

$$\hat{f}^* = \sum_{j=1}^{J} \hat{W}\phi_c \tag{26}$$

where ϕ_c is a Gaussian function given by the equation form 6. The NN actor provides compensation to the PD controller to, that is, $\hat{f}(\cdot) = \hat{f}^*$. The weigh update of the NN actor is proposed according to the following equation:

$$\dot{\hat{W}} = \gamma_c \phi_c (\hat{x} + (\hat{x} - x))^T PB \tag{27}$$

If we select the parameter update law as $\dot{\tilde{W}} \; = \; \dot{\hat{W}}$, we assume that the value $x \, \approx \, \hat{x} + (\hat{x} - x)$. Substituting Equation (27) into Equation (22), the result is given as

$$\dot{L} = -\frac{1}{2}[x^T Qx] + \zeta^T B^T Px \tag{28}$$

As it is well-known, $\|\zeta^T\| \leq \|\zeta_0\|$ and from Equation (15), $\|B\| = \|M_0^{-1}(q)\|$, and Equation (28) can be written as follows:

$$\dot{L} \leq -\tfrac{1}{2}\lambda_{min}(Q)\|x\|^2 + \|\zeta_0\|\|M_0^{-1}(q)\|\lambda_{max}(P)\|x\| \quad (29)$$

$$\dot{L} = -\tfrac{1}{2}\|x\|\left[\lambda_{min}(Q)\|x\| - 2\|\zeta_0\|\|M_0^{-1}(q)\|\lambda_{max}(P)\right] \quad (30)$$

where λ_{min} denotes the minimum eigenvalues of matrix Q and λ_{max} denotes the maximum eigenvalues of matrix P. In order to satisfy $\dot{L} \leq 0$, the value of $\|x\|$ should be satisfied as follows:

$$\|x\| \leq \frac{2\|M_0^{-1}(q)\|\lambda_{max}(P)}{\lambda_{min}Q}\|\zeta_0\| \quad (31)$$

According to Equation (28), $\dot{L}$ is negative semidefinite, that is $L(x, W, t) \leq L(x, W, 0)$. It implies that x, and W are bounded. Let function $\Omega = \dot{L}$ and integrate Ω with respect to time as follows:

$$\int_0^t \Omega(s)ds \leq L(x, W, t) \leq L(x, W, 0) \quad (32)$$

Due to $L(x, W, 0)$ is bounded, and $L(x, W, t)$ is non-increasing and bounded, the following result can be computed:

$$\lim_{t\to\infty}\int_0^t \Omega(s)ds < \infty \quad (33)$$

Since $\dot{\Omega}(s)$ is bounded, by Barbalat's lemma (Slotine and Li, 1991), $\lim_{t\to\infty}\Omega(s) = 0$, that is $x \to 0$ as $t \to \infty$.

3. RESULTS

In order to validate the proposed scheme of control, a set of simulations was carried out. Two simulations are proposed, the first considers an adaptive control using an RBFNN, and the second is based on the proposed scheme using two RBFNNs in cascade. The controllers were also implemented in a TDOFRM, which is shown in **Figure 2**. The main objective of controllers is the position tracking in the presence of external disturbances.

$$\begin{bmatrix}\boldsymbol{\tau}_1\\ \boldsymbol{\tau}_2\end{bmatrix} = \begin{bmatrix}M_{11}(q) & M_{12}(q)\\ M_{21}(q) & M_{22}(q)\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix} + \begin{bmatrix}C_{11}(q,\dot{q}) & C_{12}(q,\dot{q})\\ C_{21}(q,\dot{q}) & C_{22}(q,\dot{q})\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix} + \begin{bmatrix}g_1(\dot{q})\\ g_2(\dot{q})\end{bmatrix} \quad (34)$$

where:

$$\begin{aligned}
M_{11}(q) &= m_1l_{c1}^2 + m_2l_1^2 + m_2l_{c2}^2 + 2m_2l_1l_{c2}cos(q_2) + I_1 + I_2\\
M_{12}(q) &= m_2l_{c2}^2 + m_2l_1l_{c2}cos(q_2) + I_2\\
M_{21}(q) &= m_2l_{c2}^2 + m_2l_1l_{c2}cos(q_2) + I_2\\
M_{22}(q) &= m_2l_{c2}^2 + I_2\\
C_{11}(q,\dot{q}) &= -m_2l_1l_{c2}sin(q_2)\dot{q}_2\\
C_{12}(q,\dot{q}) &= m_2l_1l_{c2}sin(q_2)(\dot{q}_1 + \dot{q}_2)\\
C_{21}(q,\dot{q}) &= m_2l_1l_{c2}sin(q_2)\dot{q}_1\\
C_{22}(q,\dot{q}) &= 0
\end{aligned}$$

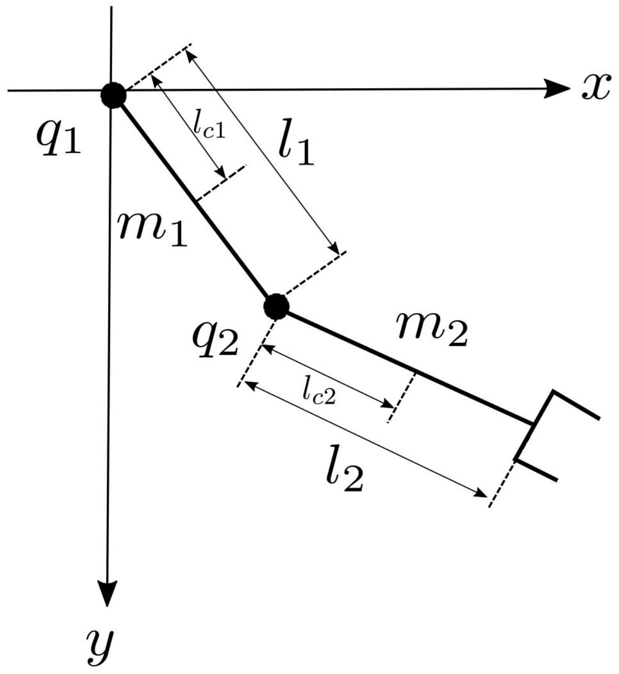


FIGURE 2 | Model of two degree-of-freedom (DOF) robot manipulator.

TABLE 1 | Robot manipulator parameters.

Parameter	Value	Units
l_1	0.45	m
l_2	0.45	m
l_{c1}	0.091	m
l_{c2}	0.048	m
m_1	23.902	kg
m_2	3.880	kg
I_1	1.266	kgm^2
I_1	0.093	kgm^2
g	9.81	$\frac{m}{s^2}$

$$\begin{aligned}
g_1(q) &= (m_1l_{c1} + m_2l_1)gsin(q_1) + m_2l_{c2}gsin(q_1 + q_2)\\
g_2(q) &= m_2l_{c2}gsin(q_1 + q_2)
\end{aligned} \quad (35)$$

The parameters came from Kelly and Santibáñez (2003) and presented in **Table 1**.

The desired position vector is defined as follows:

$$\boldsymbol{q}_d = \begin{bmatrix}\boldsymbol{q}_{1d}\\ \boldsymbol{q}_{2d}\end{bmatrix} = \begin{bmatrix}\frac{\pi}{6}sin(\frac{1}{4}\pi t)\\ \frac{\pi}{6}sin(\frac{2}{3}\pi t)\end{bmatrix} \quad (36)$$

The initial positions are given by $q_0 = \begin{bmatrix}\frac{\pi}{12} & \frac{\pi}{12}\end{bmatrix}^T$. The uncertainties and unknown disturbances are defined as follows:

$$d = \begin{bmatrix}sin(2\pi t) + cos(\frac{1}{4}\pi t)\\ sin(2\pi t) + cos(\frac{1}{4}\pi t)\end{bmatrix} \quad (37)$$

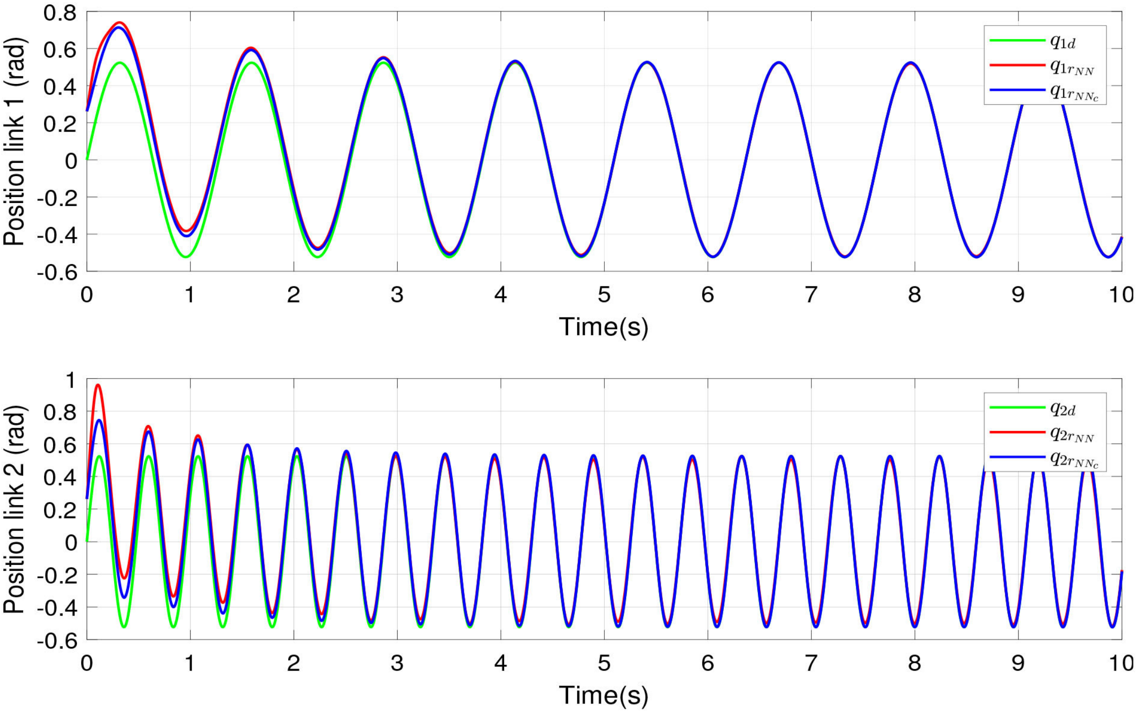


FIGURE 3 | Real and desired links positions of the robot manipulator.

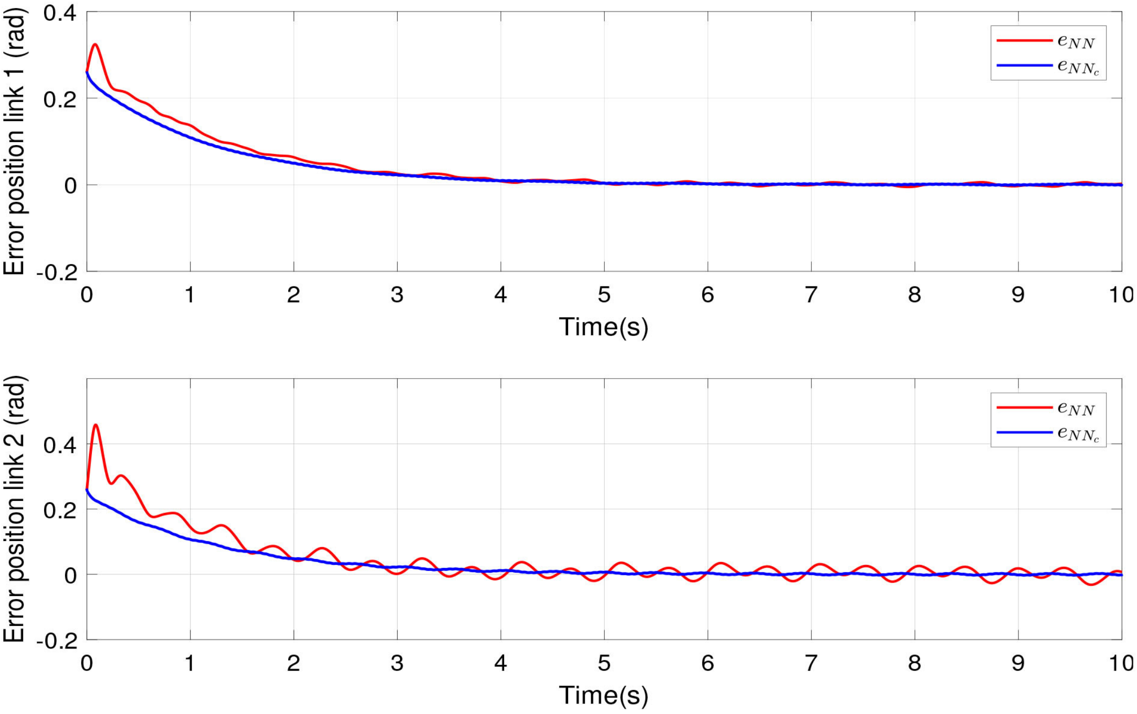


FIGURE 4 | Tracking error of the robot manipulator joints by scheme proposed and conventional compensation.

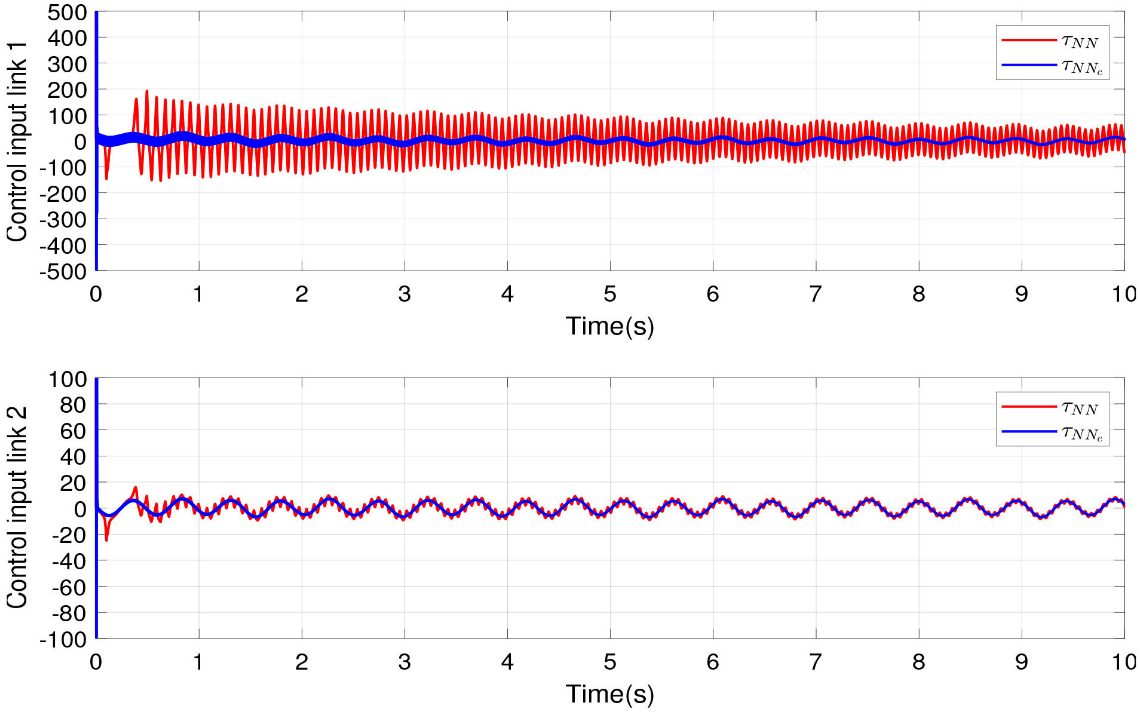


FIGURE 5 | Control inputs of links 1 and 2.

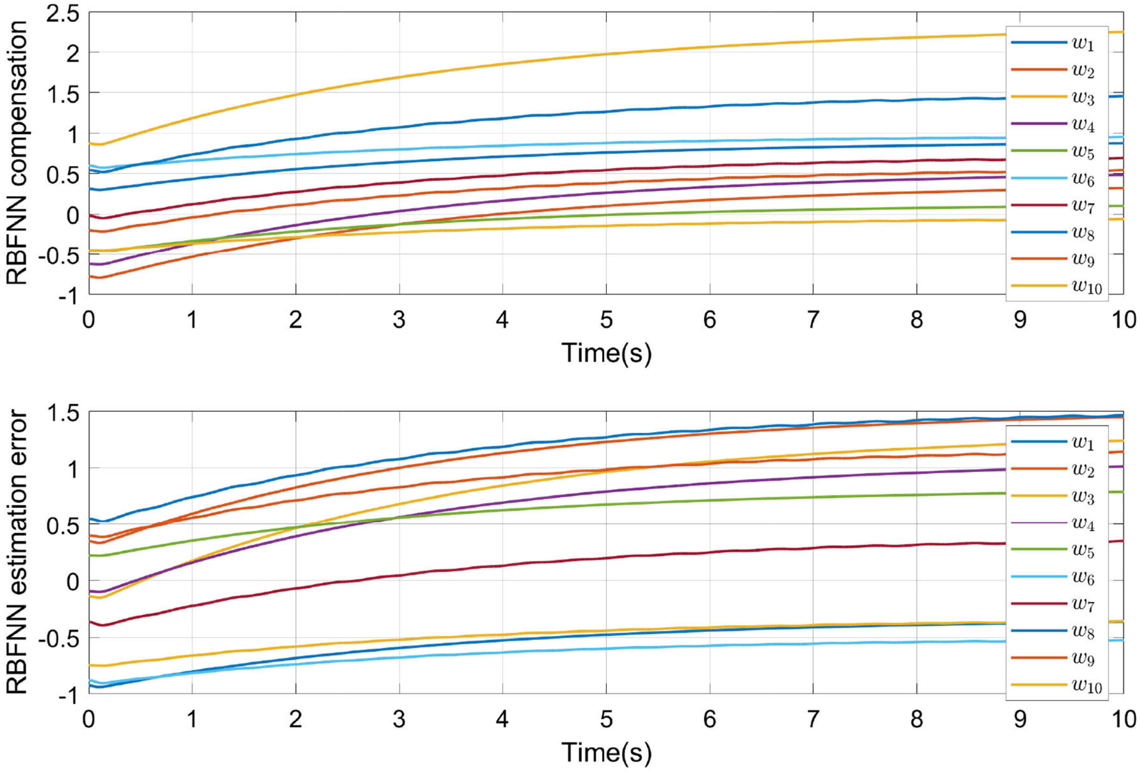


FIGURE 6 | Convergence of neural networks parameters.

The matrix Q, A, and B are

$$Q = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix};$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.25 & 0 & -1 & 0 \\ 0 & -0.25 & 0 & -1 \end{bmatrix}; B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (38)$$

Two RBFNNs are proposed for critic and actor agents. Here, γ_e is the learning rate of the RBFNN estimation error and γ_c is the learning rate of the RBFNN tracking error; c_i is the center vector of neural net i and b is the width value of Gaussian function for neural net i. The next values are proposed due to the optimal weight for the actor NN and critic NN could take arbitrary large values, but in order to avoid any numerical problems, this work considers $\gamma_e = 0.5$, $\gamma_c = 0.4$, $c_i = \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \end{bmatrix}$, $b_i = 0.5$, and $i = 4$.

Figure 3 shows the tracking position of links 1 and 2, where *NN* and NN_c indicate an RBFNN compensation and a compensation based on two RBFNN in cascade, respectively. The green lines show the desired tracking position. The red lines indicate the tracking position of NN compensation. The blue lines show the tracking position of the proposed controller. The uncertainties and disturbances were added to the controller. According to the RBFNN in cascade, an RBFNN predicts the NN estimation error, and this value is included in adaptive laws to update the RBFNN compensation in order to take adequate action for the disturbances and guarantee the convergence of tracking error. The proposed recompense function helps to maintain longer traces for the identification task over time.

The compensation of proposed algorithm is compared with an RBFNN compensation. **Figure 4** shows tracking errors for links 1 and 2. The red line represents the tracking error of an RBFNN for compensation, which presents overshoot to reach the desired positions and oscillations in steady state. The blue line indicates the tracking error of the proposed algorithm, which present robustness against uncertainties and disturbances. The green line indicates the desired tracking positions for links 1 and 2.

The simulations were proposed in order to show the difference between adaptive conventional control and the proposed scheme. In this sense, two desired tracking signal was proposed that goes at different velocities, and link 1 follows a slow signal and link 2 follows a fast signal. **Figure 5** shows the control inputs to links 1 and 2, and it also exhibits the improvement of the nominal controller under our scheme proposed in comparison with adaptive conventional control.

In **Figure 6**, the NN weights convergence process of the two RBFNN in cascade are shown. **Figure 6** also denoted as the identification process in order to deal with the uncertainties and external disturbances is reached.

The important factors that usually must be considered together are time and error. A performance index is a measure that indicates those features of the response that are regarded to be important. In order to evaluate the performance of the proposed controller, a comparison of different performance indices is shown in **Table 2**. Hence, **Table 2** is based on the next four equations, integral absolute error (IAE), integral square error (ISE), integral time absolute error (ITAE), and integral time square error (ITSE).

TABLE 2 | Comparison of different errors, ITAE, ITSE, IAE, and ISE as performance indices.

Controller	Indices	Link 1	Link 2
PD+NN	IAE	0.3758	0.4992
	ISE	0.0554	0.0869
	ITAE	0.5162	1.0030
	ITSE	0.0340	0.0578
PD+NN$_C$	IAE	0.3057	0.3077
	ISE	0.0373	0.0370
	ITAE	0.3898	0.4148
	ITSE	0.0232	0.0231

$$ISE = \int e(t)^2 dt \quad (39)$$

$$IAE = \int \mid e(t) \mid dt \quad (40)$$

$$ITSE = \int t\, e(t)^2 dt \quad (41)$$

$$ITAE = \int t \mid e(t) \mid dt \quad (42)$$

Table 2 shows that the proposed controller presents a better response than the conventional PD control compensation based on an RBFNN. According to the performance indices of links 1 and 2, the proposed scheme presents an adequate response against external disturbance. Moreover, it presents less oscillation in steady state and less time in the transient response than the conventional compensation based on an RBFNN.

4. CONCLUSIONS

The algorithm proposed has been implemented to compensate for the PD control of a TDOFRM. A PD control was selected because the common knowledge that if designed with gravity compensation, it can reach asymptotic stability. The cascade scheme was implemented by two RBFNN in cascade, which compensates for the control input in order to deal with the estimation error, uncertainties, and external disturbances. The proposed algorithm was validated using a simulation of a TDOFRM. Two adaptive algorithms for compensating for the controller of robot manipulators were implemented. The first was based on a conventional RBFNN, and the second is the proposed algorithm that uses two RBFNN. An adaptive law is proposed to deal with the simultaneous convergence of both NNs, which are used to estimate for tracking error and estimation error. The results showed that the proposed compensation scheme based on RBFNN presents robustness against external uncertainties and disturbances, also an adequate convergence for the NN weights. Position tracking has been reached without overshoots and oscillations in steady state in comparison to compensations with a conventional RBFNN scheme.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

JB and GH proposed a two-degree-of freedom robot manipulator to validate the scheme proposed and they also design the robot manipulator simulation. JV-N and DB contributed to the analysis and discussion of results. LS and EZ contributed with the theoretical analysis, simulations, and also written the main text. All authors participated and contributed to the final version manuscript.

REFERENCES

Armendariz, J., Parra-Vega, V., Garcia-Rodriguez, R., and Rosales, S. (2014). Neuro-fuzzy self-tuning of PID control for semiglobal exponential tracking of robot arms. *Appl. Soft Comput*. 25, 139–148. doi: 10.1016/j.asoc.2014.08.037

Baek, J., Jin, M., and Han, S. (2016). A new adaptive sliding-mode control scheme for application to robot manipulators. *IEEE Trans. Indus. Electron*. 63, 3628–3637. doi: 10.1109/TIE.2016.2522386

Feng, G. (1995). A compensating scheme for robot tracking based on neural networks. *Robot. Auton. Syst*. 15, 199–206. doi: 10.1016/0921-8890(95)00023-9

Gandolfo, D. C., Rossomando, F. G., Soria, C. M., and Carelli, R. O. (2019). Adaptive neural compensator for robotic systems control. *IEEE Latin Am. Trans*. 17, 670–676. doi: 10.1109/TLA.2019.8891932

He, W., Huang, B., Dong, Y., Li, Z., and Su, C.-Y. (2018). Adaptive neural network control for robotic manipulators with unknown deadzone. *IEEE Trans. Cybernet*. 48, 2670–2682. doi: 10.1109/TCYB.2017.2748418

Hwang, C.-L., and Yu, W.-S. (2020). Tracking and cooperative designs of robot manipulators using adaptive fixed-time fault-tolerant constraint control. *IEEE Access* 8, 56415–56428. doi: 10.1109/ACCESS.2020.2979795

Jung, S., and Hsia, T. (2000). Neural network inverse control techniques for PD controlled robot manipulator. *Robotica* 18, 305–314. doi: 10.1017/S0263574799002064

Kelly, R., and Santibáñez, V. (2003). *Control de Movimiento de Robots Manipuladores*. Madrid: Pearson Educación; Prentice Hall.

Krishna, S., and Vasu, S. (2018). Fuzzy PID based adaptive control on industrial robot system. *Mater. Tdy Proc*. 5(5 Pt 2), 13055–13060. doi: 10.1016/j.matpr.2018.02.292

Kumar, N., Borm, J.-H., Panwar, V., and Chai, J. (2012). Tracking control of redundant robot manipulators using RBF neural network and an adaptive bound on disturbances. *Int. J. Precis. Eng. Manufactur*. 13, 1377–1386. doi: 10.1007/s12541-012-0181-5

Liu, C., Zhao, Z., and Wen, G. (2019). Adaptive neural network control with optimal number of hidden nodes for trajectory tracking of robot manipulators. *Neurocomputing* 350, 136–145. doi: 10.1016/j.neucom.2019.03.043

Liu, H., Pan, Y., Cao, J., Wang, H., and Zhou, Y. (2020). Adaptive neural network backstepping control of fractional-order nonlinear systems with actuator faults. *IEEE Trans. Neural Netw. Learn. Syst*. doi: 10.1109/TNNLS.2020.3027335. [Epub ahead of print].

Liu, J. (2013). *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems*. Berlin; Heidelberg: Springer. doi: 10.1007/978-3-642-34816-7

Luo, R. C., and Kuo, C.-W. (2016). Intelligent seven-DoF robot with dynamic obstacle avoidance and 3-D object recognition for industrial cyber-physical systems in manufacturing automation. *Proc. IEEE* 104, 1102–1113. doi: 10.1109/JPROC.2015.2508598

Makarov, M., Grossard, M., Rodriguez-Ayerbe, P., and Dumur, D. (2016). Modeling and preview H_∞ control design for motion control of elastic-joint robots with uncertainties. *IEEE Trans. Indus. Electron*. 63, 6429–6438. doi: 10.1109/TIE.2016.2583406

Nicolis, D., Zanchettin, A. M., and Rocco, P. (2016). Constraint-based and sensorless force control with an application to a lightweight dual-arm robot. *IEEE Robot. Automat. Lett*. 1, 340–347. doi: 10.1109/LRA.2016.2517206

Pan, Y., Wang, H., Li, X., and Yu, H. (2018). Adaptive command-filtered backstepping control of robot arms with compliant actuators. *IEEE Trans. Control Syst. Technol*. 26, 1149–1156. doi: 10.1109/TCST.2017.2695600

Rahimi Nohooji, H. (2020). Constrained neural adaptive PID control for robot manipulators. *J. Franklin Instit*. 357, 3907–3923. doi: 10.1016/j.jfranklin.2019.12.042

Razmi, M., and Macnab, C. (2020). Near-optimal neural-network robot control with adaptive gravity compensation. *Neurocomputing* 389, 83–92. doi: 10.1016/j.neucom.2020.01.026

Slotine, J.-J. E., and Li, W. (1991). *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice Hall.

Spong, M. W., and Vidyasagar, M. (1989). *Robot Dynamics and Control*. New York, NY: Wiley.

Wang, F., Chao, Z.-Q., Huang, L.-B., Li, H.-Y., and Zhang, C.-Q. (2019a). Trajectory tracking control of robot manipulator based on RBF neural network and fuzzy sliding mode. *Cluster Comput*. 22, 5799–5809. doi: 10.1007/s10586-017-1538-4

Wang, H., Lei, Z., Zhang, X., Zhou, B., and Peng, J. (2019b). A review of deep learning for renewable energy forecasting. *Energy Convers. Manage*. 198, 1–16. doi: 10.1016/j.enconman.2019.111799

Wang, L., Chai, T., and Yang, C. (2012). Neural-network-based contouring control for robotic manipulators in operational space. *IEEE Trans. Control Syst. Technol*. 20, 1073–1080. doi: 10.1109/TCST.2011.2147316

Xu, J., Wang, Q., and Lin, Q. (2018). Parallel robot with fuzzy neural network sliding mode control. *Adv. Mech. Eng*. 10, 1–8. doi: 10.1177/1687814018801261

Yang, C., Jiang, Y., He, W., Na, J., Li, Z., and Xu, B. (2018). Adaptive parameter estimation and control design for robot manipulators with finite-time convergence. *IEEE Trans. Indus. Electron*. 65, 8112–8123. doi: 10.1109/TIE.2018.2803773

Yu, W., and Rosen, J. (2013). Neural PID control of robot manipulators with application to an upper limb exoskeleton. *IEEE Trans. Cybernet*. 43, 673–684. doi: 10.1109/TSMCB.2012.2214381

Zhang, H., Du, M., Wu, G., and Bu, W. (2018). PD control with RBF neural network gravity compensation for manipulator. *Eng. Lett*. 26, 236–244.

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.

ORIGINAL RESEARCH
published: 15 January 2021
doi: 10.3389/fnbot.2020.599344

Optimal UAV's Deployment and Transmit Power Design for Two Users Uplink NOMA Systems

*Fayong Zhao**

School of Physics and Electronic Engineering, Fuyang Normal University, Fuyang, China

In order to fully utilize the spectrum resources, this work considers a unmanned aerial vehicle (UAV) uplink communication system based on non-orthogonal multiple access technology (NOMA), in which the UAV receives information from the ground users with a certain flying altitude. As an initial study, we consider a simplified setup with two ground users to draw some insightful results. Explicitly, we first formulate an optimization problem that maximizes the sum throughput subject to each user's transmit power constraint and their corresponding minimum transmission rate requirement. Then, both the optimal transmit power and UAV's deployment location are derived with the aid of employing the Karush-Kuhn-Tucher (KKT) conditions. Simulation results show that the proposed UAV's deployment scheme with the users' power allocation can achieve a higher sum throughput compared with two existing benchmark schemes.

Keywords: UAV network, uplink NOMA, power control, UAV's deployment, transmit power design

Edited by:
Mu-Yen Chen,
National Taichung University of Science and Technology, Taiwan

Reviewed by:
Jun-Hong Shen,
Asia University, Taiwan
Hsin-Te Wu,
National Ilan University, Taiwan

***Correspondence:**
Fayong Zhao
zhaofy_72@fynu.edu.cn

Received: *27 August 2020*
Accepted: *17 December 2020*
Published: *15 January 2021*

Citation:
Zhao F (2021) Optimal UAV's Deployment and Transmit Power Design for Two Users Uplink NOMA Systems.
Front. Neurorobot. 14:599344.
doi: 10.3389/fnbot.2020.599344

INTRODUCTION

Non-orthogonal multiple access (NOMA) is one of the key technologies for future wireless networks, which meets the heterogeneous demands on low latency, massive connectivity, high throughput, etc. (Dai et al., 2015; Ding et al., 2017). Technically, NOMA combined several modern wireless technologies, including multiple-input multiple-output (MIMO), massive MIMO and millimeter wave communications was studied in Vaezi et al. (2019) and Wang et al. (2020). Besides, the intelligent reflecting surface (IRS) aided NOMA systems were investigated in recent work (de Sena et al., 2020). The apparent benefit of NOMA which blends those compelling techniques is that it has ability in improving scalability, spectral efficiency and energy efficiency. Compared with traditional orthogonal multiple access (OMA) schemes, such as frequency division multiple access (FDMA), time division multiple access (TDMA) and code division multiple access (CDMA), NOMA simultaneously share the time, frequency and code resources. Consequently, the inter-user interference is introduced actively. Notably, correct demodulation is achieved at the receiver through successive interference cancellation (SIC) (Saito et al., 2013; Chen et al., 2017). In Ding et al. (2014) and Timotheou and Krikidis (2015), downlink NOMA networks were studied, where the authors have demonstrated that NOMA can achieve better outage performance than that of OMA schemes, when both the users' rate and power allocation are carefully designed. In Zhang et al. (2016) and Al-Imari et al. (2014), uplink NOMA networks were discussed, where they showed that the uplink NOMA can improve both the spectrum efficiency and fairness index compared with OMA technique.

The researches related to unmanned aerial vehicles (UAVs) has become a hot topic due to their wide application prospects, such as goods delivery, search and rescue, aerial photography, and telecommunications (Zeng et al., 2016). For example, Jiao et al. (2020) presented an intuitive end-to-end interaction system between a human and an UAV in which the UAV can be commanded by natural human poses. Moreover, a brain-inspired decision-making spiking neural network (BDM-SNN) was proposed in Zhao et al. (2018), which can help UAV making decisions in some tasks. In realistic communications, UAVs can be regarded as aerial stations for serving ground users within certain areas. In order to prolong the network lifetime, energy efficient of UAV networks was studied in Amoiralis et al. (2014), Zeng and Zhang (2017), and Zhan et al. (2018). In addition, the authors of Wu et al. (2019) investigated the fundamental throughput, delay, and energy tradeoffs in UAV networks. Furthermore, the authors in Wu and Zhang (2017, 2018) and Wu et al. (2017) pointed out that the fundamental tradeoff between the delay and the throughput in multi-user UAV networks with OMA. For reducing the access latency and improving the communication quality of UAV-based networks, it is reasonable to graft the NOMA technique into UAV networks, which is termed as UAV-enabled NOMA networks (Sharma and Kim, 2017; Cui et al., 2018; Sohail et al., 2018; Liu et al., 2019; Nasir et al., 2019; Zhao et al., 2019; Do et al., 2020). Against this background, a number of works related to the UAV-enabled downlink NOMA networks have been comprehensively studied in Sharma and Kim (2017), Cui et al. (2018), and Sohail et al. (2018). Specifically, a power allocation scheme that maximizes the sum-rate of the UAV networks for reducing the energy consumption was studied in Sohail et al. (2018). In Cui et al. (2018) and Sharma and Kim (2017), the authors proposed a novel algorithm to maximize the minimum average rate by jointly optimizing the UAV's trajectory and its transmit power. Additionally, an UAV-enabled NOMA network with user pairing was studied in Nasir et al. (2019), where one user having the minimum throughput was maximized. Furthermore, to maximize the sum rate of the ground users, the authors in Liu et al. (2019) studied both the UAV's location and its transmit power. Moreover, the UAV-enabled relay NOMA networks were investigated in Do et al. (2020), and it was demonstrated that full-duplex mode can provide better outage performance than half-duplex mode. However, the extension from the downlink NOMA to uplink NOMA is not trivial because the decoding order of SIC in uplink NOMA is completely opposite to that of downlink NOMA. It is worth mentioning that the aforementioned literatures only considered the downlink scenarios, hence these existing contributions are unsuitable for the uplink scenarios, such as the data collection in the upcoming Internet of Things (IoT).

Motivated by above-mentioned reasons, this paper considers an UAV-enabled uplink NOMA with power multiplexing network, where an UAV is deployed to collect the messages transmitted from the ground users. We note that UAV-enabled uplink NOMA systems with multi-user is difficult to obtain the optimal design since the formulated optimization problem is generally difficult to tackle directly. As an initial study, similar to Wu et al. (2018), which explores the capacity of UAV-enabled/aided two user communication systems, as shown in **Figure 1** we consider the optimal UAV's deployment and each user's power allocation in UAV-enabled uplink NOMA systems with two ground users to get some insightful results. Our goal is to maximize the sum rate by jointly designing the UAV's deployment location and each user's transmit power subject to the transmit power constraints and the quality of service (QoS) constraints. We should point out that our proposed algorithm is significantly different from the recent work (Duan et al., 2019) and (Du et al., 2020). Specifically, Duan et al. (2019) studies the multi-UAV aided uplink NOMA systems, where the transmit power is solved by the proposed SCA-based iterative algorithm, but the UAVs' deployment locations are not optimized. Although Du et al. (2020) designed the UAV deployment location, the proposed algorithm has high computational complexity and only obtains a sub-optimal solution. The main contributions of this work are summarized as follows.

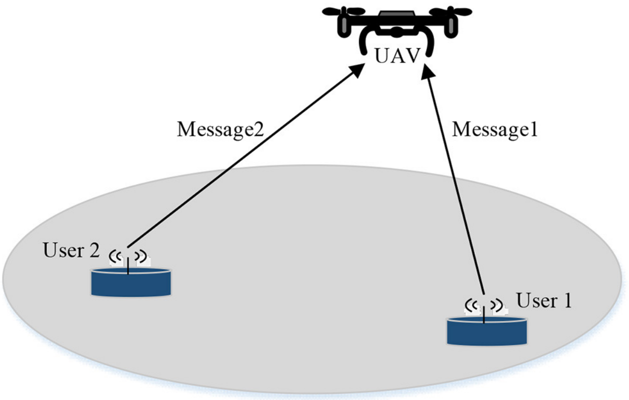


FIGURE 1 | UAV-enabled uplink NOMA network with two users.

- The analytical solution to the transmit power allocation policy that maximizes the sum rate for the considered dual-user systems is derived. Besides, the result can be further extended to general multi-user systems in a similar way.
- We prove that the optimal UAV deployment location lies on the line segment connected by the two users. Following this fact, the formulated optimization problem is transformed into a univariate quadratic optimization problem. Then the optimal UAV deployment location can be achieved.
- Numerical results confirm the validity of the analytical solution to the optimal UAV deployment location. In addition, our examinations demonstrate that our proposed scheme significantly outperforms the baseline schemes in terms of the sum rate.

The rest of this paper is organized as follows. In section System Model and Problem Formulation we present the system model for an UAV-enabled uplink NOMA network with two users and formulate the optimization problem. The corresponding algorithm for solving problem is introduced in section Proposed

Algorithm for Problem (P1). In section Numerical Results, simulation results are provided to demonstrate the performance gain of our proposed algorithm. Finally, our conclusion is provided in section Conclusion.

SYSTEM MODEL AND PROBLEM FORMULATION

In this work, we consider a two user UAV-enabled uplink NOMA network, where the UAV is adopted to collect the messages transmitted from the ground users. We consider a 3-D Cartesian coordinate system where the origin is the geometric center of the two users and the x-axis is the straight line connecting them. Assume that the distances between the two users and the origin as D, then the horizontal coordinates of the two users can be denoted as $\mathbf{u}_1 = [D,\ 0]^T$ and $\mathbf{u}_2 = [-D,\ 0]^T$, respectively. It is also assumed that the UAV flies at a fixed altitude H, and the horizontal coordinate of the UAV is denoted as $\mathbf{Q} = [x, y]^T$.

To capture the essential characteristics of dual-user UAV-enabled NOMA systems, following Zeng et al. (2019) and the recent 3GPP specification[1], we assume that the air-to-ground or ground-to-air channel is mainly dominated by line of sight (LoS) link. Thus, the channel gain from user $i(i = 1, 2)$ to the UAV is given by

$$h_i = \sqrt{\frac{\eta_0}{||Q - u_i||^2 + H^2}}, i = 1, 2 \tag{1}$$

where η_0 denotes the channel power gain at the reference distance $d_0 = 1$ m. Since the NOMA transmission scheme is adopted in this work, the received signal at the UAV is a series of superimposed message, which can be expressed as

$$y = \sqrt{P_1}h_1x_1 + \sqrt{P_2}h_2x_2 + n \tag{2}$$

where x_1 and x_2 denote the message transmitted by user 1 and user 2, respectively. P_1 and P_2 are the corresponding transmit power. n denotes the zero-mean additive white Gaussian noise (AWGN) with the variance σ^2 at the UAV. To manage the inter-user interference, the transmit power constraints are given by

$$\begin{aligned} &P_1 + P_2 \leq P_{\max} \\ &P_i \geq 0, i = 1, 2 \end{aligned} \tag{3}$$

where $P_{\max}$ denotes the maximum total transmit power of the two users. For symmetry, we only consider the scenario of $x \geq 0$ in this work. Consequently, the channel gain of user 1 is greater than that of user 2. According to the principle of NOMA, the SIC is employed at the UAV to decode the messages received from different users. In particular, the UAV first decodes the message from user 1 while treating the message from user 2 as inter-user interference. Then, the decoded message from user 1 is subtracted from the superimposed received signal. Finally, the UAV decodes the message from user 2 without inter-user interference. As a result, the achievable rate of these two users can be expressed as

$$R_1 = \log_2(1 + \frac{P_1\tilde{h}_1}{1 + P_2\tilde{h}_2}) \tag{4}$$

$$R_2 = \log_2(1 + P_2\tilde{h}_2) \tag{5}$$

where $\tilde{h}_i = \frac{h_i^2}{\sigma^2} = \frac{\zeta_O}{||Q-\boldsymbol{u}_i||^2+H^2}$, and $\zeta_0 = \frac{\eta_0}{\sigma^2}$. As a result, the sum rate of the both users is given by

$$R_{sum} = R_1 + R_2 = \log_2(1 + P_1\tilde{h}_1 + P_2\tilde{h}_2) \tag{6}$$

Our goal is to maximize $R_{\max}$ by jointly optimizing the UAV deployment location and the transmit power of the both users with QoS constraints

$$R_i \geq r^*, i = 1, 2 \tag{7}$$

where r^* denotes the minimum rate for reliable communication. As a result, the optimization problem can be written as

$$(\text{P1}): \max_{Q,P_1,P_2} R_{sum} \tag{8}$$

$$s.t. (3), (7) \tag{9}$$

Problem (P1)[2] is a non-convex optimization problem due to the non-concavity of the objective function (8) and the non-convexity of the constraint (7), which is, in general, difficult to solve. In the next section, we develop an algorithm to solve this problem.

PROPOSED ALGORITHM FOR PROBLEM (P1)

This section devises an algorithm to solve problem (P1) based on the solution to the transmit power of both the users. The analytical solution to problem (P1) is given as follows.

Solution to Transmit Power

Denote Q^* as the optimal UAV deployment location and let h_1^* and h_2^* be the corresponding channel gains of the user 1 and user 2, respectively. We note that the transmit power optimization problem is a convex problem, which can be efficiently solved the by Lagrangian. More explicitly, the corresponding Karush-Kuhn-Tucher (KKT) conditions are listed as

$$\lambda \geq 0, v_i \geq 0, i = 1, 2 \tag{10}$$

[1] 3GPP. TR 36.777: Enhanced LTE Support for Aerial Vehicles. Available online at: https://www.3gpp.org/ftp/Specs/archive/36_series/36.777 2017.

[2] In problem (P1), we assume that $x \geq 0$. However, there is also a symmetric solution to problem (P1) for $x \leq 0$, where the optimal UAV deployment is symmetric with that for $x \geq 0$ with respect to y-axis, and the optimal transmit power policy reverses.

$$\lambda(P_1 + P_2 - P_{\max}) = 0 \tag{11}$$

$$\nu_1 \left[r^* - \log_2(1 + \frac{P_1 \tilde{h}_1^*}{1 + P_2 \tilde{h}_2^*}) \right] = 0 \tag{12}$$

$$\nu_2 \left[r^* - \log_2(1 + P_2 \tilde{h}_2^*) \right] = 0 \tag{13}$$

$$\lambda - \frac{(1+\nu_1)\tilde{h}_1^*/\ln 2}{1 + P_1 \tilde{h}_1^* + P_2 \tilde{h}_2^*} = 0 \tag{14}$$

$$\lambda - \frac{\tilde{h}_2^*/\ln 2}{1 + P_1 \tilde{h}_1^* + P_2 \tilde{h}_2^*} - \frac{\nu_2 \tilde{h}_2^*/\ln 2}{1 + P_2 \tilde{h}_2^*} + \frac{\nu_1 P_1 \tilde{h}_1^* \tilde{h}_2^*/\ln 2}{(1 + P_2 \tilde{h}_2^*)(1 + P_1 \tilde{h}_1^* + P_2 \tilde{h}_2^*)} = 0 \tag{15}$$

where λ, ν_1 and ν_2 are the Lagrange multipliers. As per Equation (14), we can obtain

$$\lambda = \frac{(1+\nu_1)\tilde{h}_1^*/\ln 2}{1 + P_1 \tilde{h}_1^* + P_2 \tilde{h}_2^*} > 0 \tag{16}$$

According to Equation (11), we have $P_1 + P_2 = P_{\max}$. Upon substituting Equation (16) into Equation (15), we have

$$\frac{\nu_2 \tilde{h}_2^*}{1 + P_2 \tilde{h}_2^*} = \frac{\tilde{h}_1^* - \tilde{h}_2^* + \tilde{h}_1^* \nu_1}{1 + P_1 \tilde{h}_1^* + P_2 \tilde{h}_2^*} + \frac{\nu_1 P_1 \tilde{h}_1^* \tilde{h}_2^*}{(1 + P_2 \tilde{h}_2^*)(1 + P_1 \tilde{h}_1^* + P_2 \tilde{h}_2^*)} > 0 \tag{17}$$

where $\nu_2 > 0$. As per Equation (13), we have

$$\log_2(1 + P_2 \tilde{h}_2^*) = r^* \tag{18}$$

Based on the above derivation, the optimal solution to transmit power can be expressed as

$$P_2^* = \frac{2^{r^*} - 1}{\tilde{h}_2^*} \tag{19}$$

$$P_1^* = P_{max} - P_2^*. \tag{20}$$

Solution to UAV Deployment Location

To determine the optimal UAV deployment location, we have the following Lemma 1.

Lemma 1. The optimal horizontal coordinate of the UAV for maximizing R_{sum} should be on the line segment that linked by the two users.

Proof of Lemma 1. Assume $Q^* = [x^*, y^*]^T, (y^* \neq 0)$, i.e. the optimal deployment location deviates from the line segment. Let us define the achievable rates of the two users at the optimal solution as ${R_1}^*$ and ${R_2}^*$, respectively. Then, aided with the results in the above subsection, we can obtain

$${R_1}^* = \log_2(1 + \frac{P_1^* \tilde{h}_1^*}{2^{r^*}}) \geq r^* \tag{21}$$

$${R_2}^* = \log_2(1 + P_2^* \tilde{h}_2^*) = r^* \tag{22}$$

However, if we deploy the UAV at $Q' = [x^*, 0]^T$, the corresponding channel gains $\tilde{h}_1{}'$ and $\tilde{h}_2{}'$ will be larger than $\tilde{h}_1^*$ and $\tilde{h}_2^*$, respectively. Assuming that $P_1{}'$ and $P_2{}'$ are the optimal transmit power at this time, then we have

$${R_1}' = \log_2(1 + \frac{{P_1}'{\tilde{h}_1}'}{2^{r^*}}) \overset{(a)}{>} \log_2(1 + \frac{P_1^* \tilde{h}_1^*}{2^{r^*}}) = R_1^* \geq r^* \tag{23}$$

$${R_2}' = \log_2(1 + {P_2}'{\tilde{h}_2}') = r^* = R_2^* \tag{24}$$

where (a) holds since ${P_2}' = \frac{2^{r^*}-1}{{\tilde{h}_2}'} < {P_2}^*$, and thus having ${P_1}' = P_{\max} - P_2^{'} > P_{\max} - P_2^* = P_1^*$. As a result, $Q' = [x^*, 0]^T$ is also a feasible deployment location. Apparently, the corresponding sum rate will be larger than that at Q^*.

Similarly, if $x^* > D$, we can deploy the UAV at $Q'' = [D, 0]^T$, and the corresponding sum rate is larger. Consequently, the optimal UAV deployment location has to be located at one point on the line segment linked by the two users. This completes the proof.

Based on the above results, problem (P1) can be further reformulated as

$$(\text{P2}): \max_{x^*} \phi(x^*) \tag{25}$$

$$s.t. 0 \leq x^* \leq D \tag{26}$$

where

$$\phi(x^*) = P_1^* \tilde{h}_1^* = \frac{\zeta_0 P_{\max}}{H^2 + (D - x^*)^2} - \frac{(2^{r^*} - 1)[H^2 + (D - x^*)^2]}{H^2 + (D + x^*)^2} \tag{27}$$

Clearly, problem (P2) is a univariate quadratic optimization problem. The derivative of $\phi(x^*)$ is given by

$$\frac{d\phi(x^*)}{dx^*} = \frac{2[2D - (x^*)^2 - \Gamma x^* + \Gamma D - 2D(H^2 + D^2)]}{[H^2 + (D - x^*)^2]^2} \tag{28}$$

where $\Gamma = \frac{\zeta_0 P_{\max}}{2^{r^*} - 1}$. We note that $\frac{d\phi(x^*)}{dx^*}$ is equivalent to

$$2D(x^*)^2 - \Gamma x^* + \Gamma D - 2D(H^2 + D^2) = 0 \tag{29}$$

It can be observed that Equation (29) is a quadric equation. To proceed, we define $\Delta = \Gamma^2 - 8\Gamma D^2 + 16D^2(H^2 + D^2)$. If $\Delta < 0$, $\phi(x^*)$ is a monotonically increasing function with respect to x^*, and therefore the optimal deployment location is arrived when $x_{opt} = D$. By contrast, as $\Delta \geq 0$, the two stationary points x_1^* and x_2^* can be derived as

$$x_1^* = \frac{\Gamma - \sqrt{\Delta}}{4D} \tag{30}$$

$$x_2^* = \frac{\Gamma + \sqrt{\Delta}}{4D} \tag{31}$$

Based on the relation between x_1^*,x_2^* and the interval $(0, D)$, the solution to the optimal UAV deployment location x_{opt} can be obtained via Algorithm 1. Finally, we have to check these obtained solutions until finding the one meets the condition of $R_1^* \geq r^*$.

```
Algorithm 1 Solution to the optimal UAV deployment location.
Input: $D, H, r^*, P_{\max}$, and $\zeta_0$.
    Calculate $x_1^*$ and $x_2^*$ according to Equations (30) and (31), respectively.
    If $x_1^* \in (0, D), x_2^* \in (0, D)$, then $x_{\mathrm{opt}} = \arg\max\{\phi(0), \phi(x_1{}^*), \phi(x_2{}^*), \phi(D)\}$.
    If $x_1^* \in (0, D), x_2^* \notin (0, D)$, then $x_{\mathrm{opt}} = \arg\max\{\phi(0), \phi(x_1{}^*), \phi(D)\}$.
    If $x_1^* \notin (0, D), x_2^* \in (0, D)$, then $x_{\mathrm{opt}} = \arg\max\{\phi(0), \phi(D)\}$.
    If $x_1^* \notin (0, D), x_2^* \notin (0, D)$, then $x_{\mathrm{opt}} = \arg\max\{\phi(0), \phi(D)\}$.
Output: The optimal UAV deployment location $x_{\mathrm{opt}}$.
```

Solution to Problem (P1)

According to the results of the previous two subsections, the optimal solution to problem (P1) can be obtained via Algorithm 2 given below, and the flow chart of the proposed algorithm is shown in **Figure 2**. Specifically, the optimal UAV deployment location is first obtained via Algorithm 1. Then, the corresponding channel gains of user 2 is calculated. Finally, the optimal solution to the transmit power can be obtained based on Equations (19) and (20). It is easy to calculate that, in the worst case, the proposed algorithm requires 59 multiplications and 44 additions.

```
Algorithm 2 Solution to Problem (P1).
Input: $D, H, r^*, P_{max}$, and $\zeta_0$.
    Calculate the optimal UAV deployment position $x_{\mathrm{opt}}$ via Algorithm 1.
    Calculate the corresponding channel gain of user 2 $\tilde{h}_2$.
    Calculate the optimal transmit power $P_1^*$ and $P_2^*$ based on Equations (19) and (20).
Output: $x_{\mathrm{opt}}$, $P_1^*$, and $P_2^*$.
```

NUMERICAL RESULTS

In this section, simulation results are provided to demonstrate the effectiveness of our proposed algorithm (denoted as N-LPJO). Referring to the existing the literatures (Wu and Zhang, 2017, 2018; Wu et al., 2017, 2019; Zeng and Zhang, 2017; Cui et al., 2018; Sohail et al., 2018; Zhan et al., 2018), the simulation parameters, unless otherwise specified, are set as: the maximum total transmit power of the two users $P_{\max} = 10$dBm, the UAV altitude $H = 200$ m, the distance between the two users and the origin $D = 400$ m, and the reference signal-to-noise ratio (SNR) $\zeta_0 = 80$ dB.

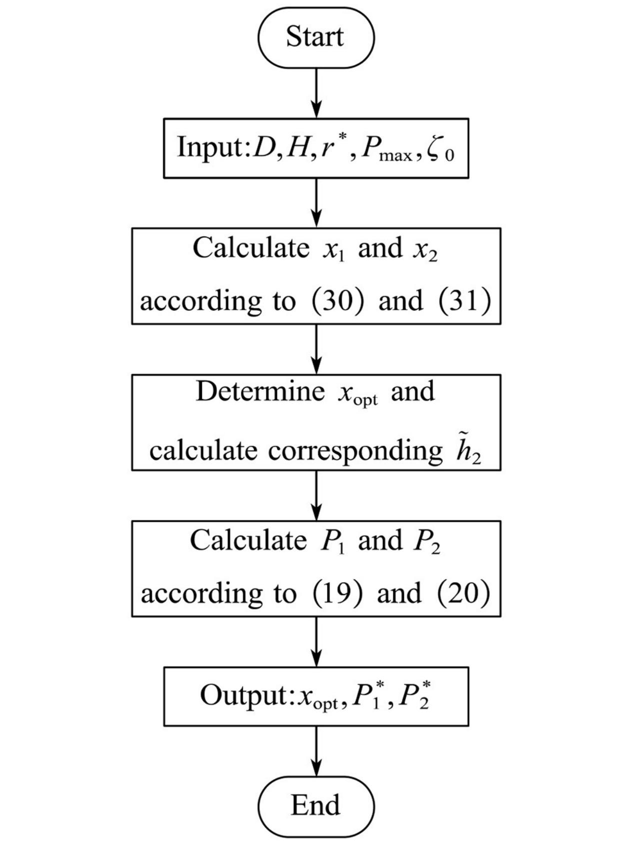

Start
Input: $D, H, r^*, P_{\max}, \zeta_0$
Calculate x_1 and x_2 according to (30) and (31)
Determine x_{opt} and calculate corresponding $\tilde{h}_2$
Calculate P_1 and P_2 according to (19) and (20)
Output: $x_{\mathrm{opt}}, P_1^*, P_2^*$
End

FIGURE 2 | Flow chart of the proposed algorithm.

For comparison, the following three baseline schemes are invoked:

1) FDMA: The UAV collects the messages in FDMA manner, where both the UAV deployment location and transmit power are jointly optimized;
2) N-FLPO: The scheme in Duan et al. (2019), where only the transmit power is optimized while the UAV is fixed at the geometric center of two users, i.e., $[0, 0]^T$;
3) N-LOFP: The UAV collects the messages in NOMA manner, where only the UAV deployment location is optimized while the transmit power is fixed as $P_1 = 2$ mW and $P_2 = 8$ mW.

Figure 3 plots the optimal UAV deployment location of the N-LPJO scheme vs. r^*, where the numerical results (obtained by 1-D search method) are invoked to reveal the optimality of our proposed algorithm. It can be noted that the sum rates achieved by 1-D search method meets that arrived by our developed analytical solution. This phenomenon implies the optimality of the analytical solution. We also observe that the optimal UAV deployment location is close to user 1, which is beneficial for improving the achievable rate of user 1. Moreover, we observe that the optimal UAV deployment location moves toward the

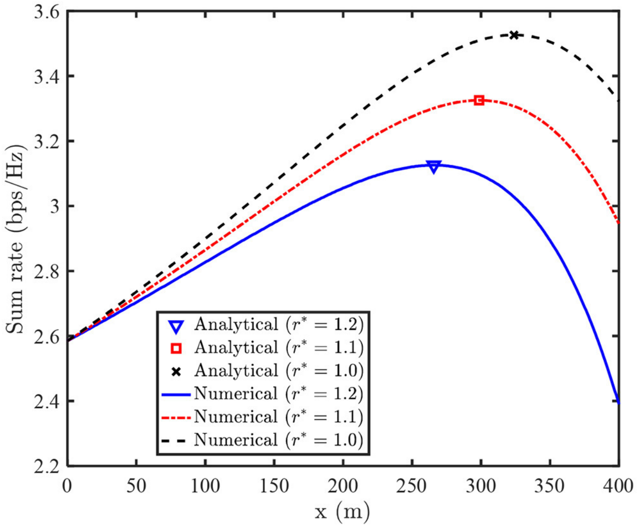


FIGURE 3 | Optimal UAV's deployment location for N-LPJO scheme vs. r^*.

origin as r^* increases. This is due to the fact that it will cost less transmit power to meet user 2's QoS constraint if the UAV is deployed close to the origin.

Figure 4 plots the achievable sum rates of the four different schemes vs. r^*. We observe that the sum rates of the N-LPJO and the FDMA schemes decreases as r^*increases. This is mainly because the poorer user has to increase the transmit power for meeting QoS requirement. Meanwhile, the stronger user has to decrease its transmit power. However, the sum rate of the N-FLPO scheme remains unchanged regardless of r^*, this is due to the fact that $\tilde{h}_1 = \tilde{h}_2$ and thus $R_{sum} = \log_2(1 + P_1\tilde{h}_1 + P_2\tilde{h}_2) = \log_2(1+(P_1 + P_2)\tilde{h}_2) = \log_2(1 + P_{\max}\tilde{h}_2)$ is a constant. Besides, different from the other three schemes, the sum rate of the N-LOFP scheme remains unchanged when $r^* \leq 1.1$ bps and decreases when $r^* = 1.2$ bps. This is because the optimal UAV deployment location can naturally meet the QoS constraints if the QoS constraints are not very tight. It is clearly shown that our proposed N-LPJO scheme outperforms both the N-FLPO scheme and the N-LOFP scheme, which demonstrates the necessity of optimizing the UAV deployment position and the transmit power, respectively. Furthermore, our proposed N-LPJO scheme also outperforms the FDMA scheme since NOMA can provide higher spectral efficiency than OMA.

Figure 5 shows that the sum rate of our proposed N-LPJO scheme vs. the distance D between the two users and the origin. Firstly, we observe that the sum rate decreases as D increases for each QoS requirement r^*. This can be explained as: the channel gain of user 2 sharply decreases once D increases. As a result, user 2 has to improve the transmit power to meet the minimum rate requirement, while user 1 has to decrease its transmit power, thus leading to the decrease in the sum rate. Secondly, we note that the difference of the sum rate achieved by the different r^* becomes apparent with the increasing of D.

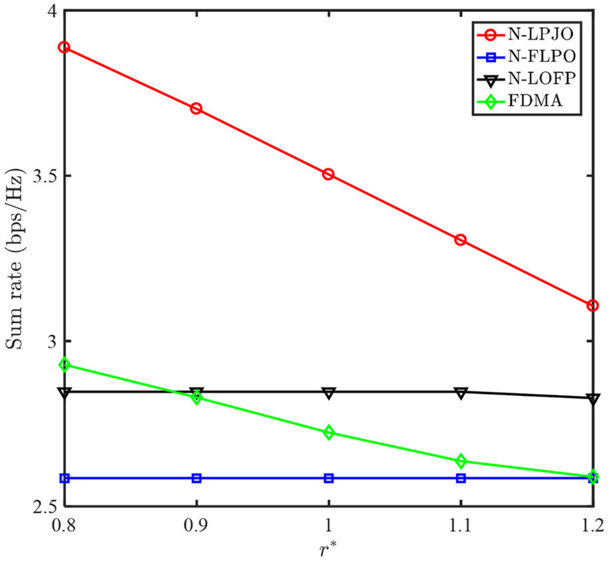


FIGURE 4 | Sum rates of four schemes vs. r^*.

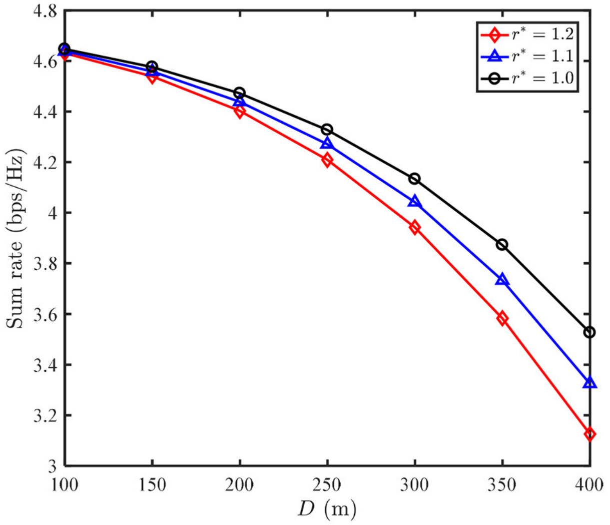


FIGURE 5 | Sum rate of N-LPJO scheme vs. distance D between both users.

Figure 6 shows the sum rate of our proposed N-LPJO scheme vs. UAV's altitude H. It is observed that all the sum rates decrease as H increases for each QoS requirement. This is due to the fact that both the channel gains of the two users decrease as H increases, hence user 2 has to increase its transmit power for satisfying the QoS requirement, leading to the received power from user 1 decreases. Additionally, we observe that all the performance gains attained by the different r^* decrease as H increases. This is because that the effect of H on the two users' channel gains will be rather small as H becomes sufficiently large.

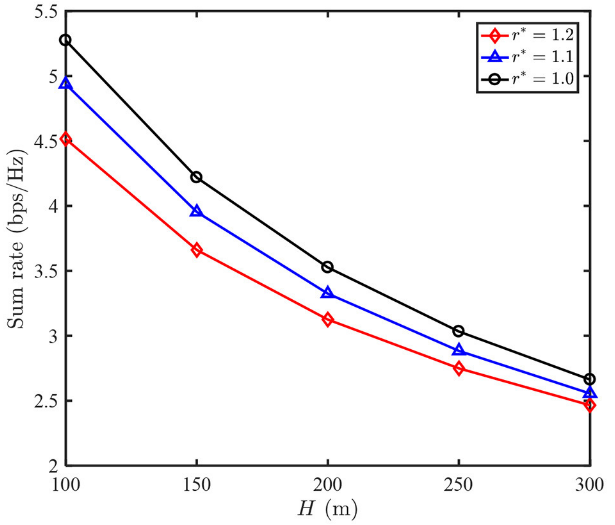


FIGURE 6 | Sum rate of N-LPJO scheme vs. UAV altitude H.

CONCLUSIONS

In this paper, we have investigated an UAV-enabled uplink NOMA system under a total power constraint. To maximize the sum rate of two users, we have demonstrated that the UAV should be deployed at a certain point over the line segment linked by the two users. Then, we have translated the corresponding optimization into a univariate quadratic optimization problem, which can be efficiently solved by our developed scheme. Simulation results showed that our proposed scheme significantly outperforms both the two benchmarks in terms of the sum rate. It should be pointed out that the proposed power allocation policy can be extended to general multi-user systems in a similar way, but extending the proposed UAV deployment scheme to general multi-user systems is not trivial. Our future works will commit to solving the sum rate maximization problem in general multi-user networks.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and has approved it for publication.

FUNDING

This research was funded in part by the Natural Science Research Project of Education Department of Anhui Province of China under Grant KJ2019A1002, in part by horizontal cooperation project under Grant XDHX201705, and in part by the Key scientific and technological projects of Anhui Province under Grand 1501031114.

REFERENCES

Al-Imari, M., Xiao, P., Imran, M. A., and Tafazolli, R. (2014). "Uplink non-orthogonal multiple access for 5G wireless networks," in *2014 11th International Symposium on Wireless Communications Systems* (Barcelona), 781–785. doi: 10.1109/ISWCS.2014.6933459

Amoiralis, E. I., Tsili, M. A., Spathopoulos, V., and Hatziefremidis, A. (2014). Energy efficiency optimization in UAVs: a review. *Mater. Sci. Forum* 792, 281–286. doi: 10.4028/www.scientific.net/MSF.792.281

Chen, Z., Ding, Z., Dai, X., and Zhang, R. (2017). An optimization perspective of the superiority of NOMA compared to conventional OMA. *IEEE Trans. Signal Process.* 65, 5191–5202. doi: 10.1109/TSP.2017.2725223

Cui, F., Cai, Y., Qin, Z., Zhao, M., and Li, G. Y. (2018). "Joint trajectory design and power allocation for UAV-enabled non-orthogonal multiple access systems," in *2018 IEEE Global Communications Conference (GLOBECOM)* (Abu Dhabi), 1–6. doi: 10.1109/GLOCOM.2018.8647352

Dai, L., Wang, B., Yuan, Y., Han, S. I. C., and Wang, Z. (2015). Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends. *IEEE Commun. Mag.* 53, 74–81. doi: 10.1109/MCOM.2015.7263349

de Sena, A. S., Carrillo, D., Fang, F., Nardelli, P. H. J., Costa, D. B., Dias, U. S., et al. (2020). What role do intelligent reflecting surfaces play in multi-antenna non-orthogonal multiple access? *IEEE Wireless Commun.* 27, 24–31. doi: 10.1109/MWC.001.2000061

Ding, Z., Lei, X., Karagiannidis, G. K., Schober, R., and Bhargava, V. A. (2017). Survey on non-orthogonal multiple access for 5G networks: research challenges and future trends. *IEEE J. Sel. Areas Commun.* 35, 2181–2195. doi: 10.1109/JSAC.2017.2725519

Ding, Z., Yang, Z., Fan, P., and Poor, H. V. (2014). On the performance of non-orthogonal multiple access in 5G systems with randomly deployed users. *IEEE Signal Process. Lett.* 21, 1501–1505. doi: 10.1109/LSP.2014.2343971

Do, D., Nguyen, T., Le, C., Voznak, M., Kaleem, Z., and Rabie, K. M. (2020). UAV relaying enabled NOMA network with hybrid duplexing and multiple antennas. *IEEE Access* 8, 186993–187007. doi: 10.1109/ACCESS.2020.3030221

Du, J., Wang, Z., Fan, Z., and Wan, X. (2020). "Sum rate maximization for UAV-enabled wireless powered NOMA systems," in *2020 IEEE/CIC International Conference on Communications in China (ICCC)* (Chongqing), 753–757. doi: 10.1109/ICCC49849.2020.9238850

Duan, R., Wang, J., Jiang, C., Yao, H., Ren, Y., and Qian, Y. (2019). Resource allocation for multi-UAV aided IoT NOMA uplink transmission systems. *IEEE Internet Things J.* 6, 7025–7037. doi: 10.1109/JIOT.2019.2913473

Jiao, R., Wang, Z., Chu, R., Dong, M., Rong, Y., and Chou, W. (2020). An intuitive end-to-end human-UAV interaction system for field exploration. *Front. Neurorobot.* 13:117. doi: 10.3389/fnbot.2019.00117

Liu, X., Wang, J., Zhao, N., Chen, Y., Zhang, S., Ding, Z., et al. (2019). Placement and power allocation for NOMA-UAV networks. *IEEE Wireless Commun. Lett.* 8, 965–968. doi: 10.1109/LWC.2019.2904034

Nasir, A. A., Tuan, H. D., Duong, T. Q., and Poor, H. V. (2019). UAV-enabled communication using NOMA. *IEEE Trans. Commun.* 67, 5126–5138. doi: 10.1109/TCOMM.2019.2906622

Saito, Y., Kishiyama, Y., Benjebbour, A., Nakamura, T., Li, A., and Higuchi, K. (2013). "Non-orthogonal multiple access (NOMA) for cellular future radio access," in *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)* (Dresden), 1–5. doi: 10.1109/VTCSpring.2013.6692652

Sharma, P. K., and Kim, D. I. (2017). "UAV-enabled downlink wireless system with non-orthogonal multiple access," in *IEEE Globecom Workshops (GC Wkshps)* (Singapore), 1–6. doi: 10.1109/GLOCOMW.2017.8269066

Sohail, M. F., Leow, C. Y., and Won, S. (2018). Non-orthogonal multiple access for unmanned aerial vehicle assisted communication. *IEEE Access* 6, 22716–22727. doi: 10.1109/ACCESS.2018.2826650

Timotheou, S., and Krikidis, I. (2015). Fairness for non-orthogonal multiple access in 5G systems. *IEEE Signal Process. Lett.* 22, 1647–1651. doi: 10.1109/LSP.2015.2417119

Vaezi, M., Amarasuriya, G., Liu, Y., Arafa, A., Fang, F., and Ding, Z. (2019). Interplay between NOMA and other emerging technologies: a survey. *IEEE Trans. Cogn. Commun. Netw*. 5, 900–919. doi: 10.1109/TCCN.2019.2933835

Wang, Y., Tian, Z., and Cheng, X. (2020). Enabling technologies for spectrum and energy efficient NOMA-MmWave-MaMIMO systems. *IEEE Wireless Commun.* 7, 53–59. doi: 10.1109/MWC.001.2000055

Wu, Q., Liu, L., and Zhang, R. (2019). Fundamental trade-offs in communication and trajectory design for UAV-enabled wireless network. *IEEE Wireless Commun.* 26, 36–44. doi: 10.1109/MWC.2018.1800221

Wu, Q., Xu, J., and Zhang, R. (2018). Capacity characterization of UAV-enabled two-user broadcast channel. *IEEE J. Sel. Areas Commun.* 36, 1955–1971. doi: 10.1109/JSAC.2018.2864421

Wu, Q., Zeng, Y., and Zhang, R. (2017). "Joint trajectory and communication design for UAV-enabled multiple access," in *LOBECOM 2017 - 2017 IEEE Global Communications Conference* (Singapore), 1–6. doi: 10.1109/GLOCOM.2017.8254949

Wu, Q., and Zhang, R. (2017). "Delay-constrained throughput maximization in UAV-enabled OFDM systems," in *2017 23rd Asia-Pacific Conference on Communications (APCC)* (Perth, WA), 1–6. doi: 10.23919/APCC.2017.8304088

Wu, Q., and Zhang, R. (2018). Common throughput maximization in UAV-enabled OFDMA systems with delay consideration. *IEEE Trans. Commun.* 66, 6614–6627. doi: 10.1109/TCOMM.2018.2865922

Zeng, Y., Wu, Q., and Zhang, R. (2019). Accessing from the sky: a tutorial on UAV communications for 5G and beyond. *Proc. IEEE* 107, 2327–2375. doi: 10.1109/JPROC.2019.2952892

Zeng, Y., and Zhang, R. (2017). Energy-efficient UAV communication with trajectory optimization. *IEEE Trans. Wireless Commun.* 16, 3747–3760. doi: 10.1109/TWC.2017.2688328

Zeng, Y., Zhang, R., and Lim, T. J. (2016). Wireless communications with unmanned aerial vehicles: opportunities and challenges. *IEEE Commun. Mag.* 54, 36–42. doi: 10.1109/MCOM.2016.7470933

Zhan, C., Zeng, Y., and Zhang, R. (2018). Energy-efficient data collection in UAV enabled wireless sensor network. *IEEE Wireless Commun. Lett.* 7, 328–331. doi: 10.1109/LWC.2017.2776922

Zhang, N., Wang, J., Kang, G., and Liu, Y. (2016). Uplink non-orthogonal multiple access in 5G systems. *IEEE Commun. Lett.* 20, 458–461. doi: 10.1109/LCOMM.2016.2521374

Zhao, F., Zeng, Y., and Xi, B. (2018). A brain-inspired decision-making spiking neural network and its application in unmanned aerial vehicle. *Front. Neurorobot*. 12:56. doi: 10.3389/fnbot.2018.00056

Zhao, N., Pang, X., Li, Z., Chen, Y., Li, F., Ding, Z., et al. (2019). Joint trajectory and precoding optimization for UAV-assisted NOMA networks. *IEEE Trans. Commun.* 67, 3723–3735. doi: 10.1109/TCOMM.2019.2895831

Conflict of Interest: The author 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

BRIEF RESEARCH REPORT
published: 15 January 2021
doi: 10.3389/fnbot.2020.620378

Adoption of Machine Learning Algorithm-Based Intelligent Basketball Training Robot in Athlete Injury Prevention

Teng Xu[1] *and Lijun Tang*[2*]

[1] Department of Physical Education, Shanghai Jiao Tong University, Shanghai, China, [2] Physical Education College, Shanghai Normal University, Shanghai, China

In order to effectively prevent sports injuries caused by collisions in basketball training, realize efficient shooting, and reduce collisions, the machine learning algorithm was applied to intelligent robot for path planning in this study. First of all, combined with the basketball motion trajectory model, the sport recognition in basketball training was analyzed. Second, the mathematical model of the basketball motion trajectory of the shooting motion was established, and the factors affecting the shooting were analyzed. Thirdly, on this basis, the machine learning-based improved Q-Learning algorithm was proposed, the path planning of the moving robot was realized, and the obstacle avoidance behavior was accomplished effectively. In the path planning, the principle of fuzzy controller was applied, and the obstacle ultrasonic signals acquired around the robot were taken as input to effectively avoid obstacles. Finally, the robot was able to approach the target point while avoiding obstacles. The results of simulation experiment show that the obstacle avoidance path obtained by the improved Q-Learning algorithm is flatter, indicating that the algorithm is more suitable for the obstacle avoidance of the robot. Besides, it only takes about 250 s for the robot to find the obstacle avoidance path to the target state for the first time, which is far lower than the 700 s of the previous original algorithm. As a result, the fuzzy controller applied to the basketball robot can effectively avoid the obstacles in the robot movement process, and the motion trajectory curve obtained is relatively smooth. Therefore, the proposed machine learning algorithm has favorable obstacle avoidance effect when it is applied to path planning in basketball training, and can effectively prevent sports injuries in basketball activities.

Keywords: machine learning algorithm, intelligent robot, basketball training, sports injury, athlete injury prevention

OPEN ACCESS

Edited by:
Mu-Yen Chen,
National Taichung University of Science and Technology, Taiwan

Reviewed by:
Jing-Wei Liu,
National Taiwan University of Physical Education and Sport, Taiwan
Ferozkhan A. B,
C. Abdul Hakeem College of Engineering and Technology, India

Correspondence:
Lijun Tang
ljtang1976@shnu.edu.cn

Received: *22 October 2020*
Accepted: *10 December 2020*
Published: *15 January 2021*

Citation:
Xu T and Tang L (2021) Adoption of Machine Learning Algorithm-Based Intelligent Basketball Training Robot in Athlete Injury Prevention. Front. Neurorobot. 14:620378. doi: 10.3389/fnbot.2020.620378

INTRODUCTION

Basketball is widely accepted as one of the world's top sports. In the process of training, there are more comprehensive requirements on the physical and competitive ability of athletes (Coglianese and Lehr, 2017; Liu and Hodgins, 2018; Ji, 2020). Sports injury is an inevitable problem faced by athletes in all sports, and it is also an unavoidable situation in training and competition. Basketball increases the risk of injury due to the high intensity and high-speed collisions between players in the sport. Especially, excessive use of joints, muscles, and ligaments will lead to joint sprains and muscle strains (Li et al., 2016). Sports injuries not only damage the physical and mental health of individual athletes, but also affect their competitive ability and restrict the development of the whole team (Li et al., 2019). To prevent sports injuries caused by collisions in training or competitions, it is necessary to optimize the offensive and defensive paths by intelligent basketball training robots.

In modern basketball, the advantages of intelligent robots based on artificial intelligence, machine learning, and other algorithms in sports training gradually appear (Zhao et al., 2018). In the context of the Internet of Things, Lv (2020) combined high-tech achievements in multiple fields to realize human-computer interaction in a natural and intelligent way. Information processing and operation through computer programs can form an interactive and simulated natural state and three-dimensional environment on the display terminal, which can make people feel immersed. This is also of great reference value for intelligent basketball training (Lv, 2020). Toyota has developed a basketball robot called "Cue," with the internal visual feedback system as its core. Before shooting the shot, "Cue" will determine the 3D image of the surrounding environment via its limbs and torso, calculate the height and force of the basketball through the algorithm, and thus achieve the goal of accuracy (Narayanan et al., 2020). Shooting results show that it is considerably more accurate than professional basketball players, because it can continue to learn and improve itself under strong support of artificial intelligence technology. Some scholars combined the labVIEW software development platform and visual development module of the American company. In relative motion state , the basketball could achieve the goal of precise positioning and meet the adoption requirements of basketball robots in actual games (Bing et al., 2020). At present, there is an ultrasonic-based multi-degree-of-freedom basketball robot complex path tracking control system. Ultrasound can accurately provide the distance information of obstacles encountered by the robot. If there is an obstacle, the received information is converted and fed back to the main control board in the form of an electrical signal, which ultimately effectively improves the recognition rate of the basketball robot to the obstacle. Since most sports injuries in basketball activities are caused by failure to avoid obstacles and errors in cooperation between attack and defense, the machine learning algorithm is utilized to plan the movement posture and autonomous path of basketball players, so as to effectively avoid obstacles and prevent sports injuries.

In this study, a robot system for basketball training based on machine learning algorithm was proposed from the perspective of preventing athletes from training injuries and the characteristics of fierce confrontation in basketball. The autonomous path planning problem of the robot system applies the improved Q-Learning algorithm. On this basis, combined with the basketball trajectory model, the motion robot controller system based on fuzzy control principle was analyzed. Finally, the feasibility of the improved algorithm to solve the autonomous motion was verified through simulation experiments. According to the obstacle avoidance ability of the sports robot based on machine learning algorithm in basketball training, the occurrence of sports injuries was effectively prevented.

MATERIALS AND METHODS

Sport Recognition in Basketball Training

The movements involved in basketball constitute a relatively complex and comprehensive movement system. The recognition of basic basketball movements is of great value to further improve the training efficiency of athletes, and the division of basketball posture composition is shown in **Figure 1**. In ordinary training, basic training movements mainly include dribbling, ball control, passing, catching, shooting, and pace adjustment (Lobos-Tsunekawa et al., 2018). In specific training, basketball posture can be divided into two states: static and moving based on the different states of the body. The basketball action can also be divided into two kinds according to whether it is periodic or not: instantaneous and continuous. During continuous movement, the athletes' upper and lower limbs keep a continuous periodic transformation. Therefore, unit actions can be divided according to specific action data (Chau et al., 2019; Yoon et al., 2019; Stübinger et al., 2020). When the body movement is described, angular velocity can be taken as the reference base of data division due to its intuitive advantage of data.

After data division, the unit action data composed of acceleration and angular velocity can be obtained. The acceleration vector sum and the angular velocity vector sum are expressed as a_n and g_n, respectively.

$$a_n = \sqrt{\left(a_n^x\right)^2 + \left(a_n^y\right)^2 + \left(a_n^z\right)^2} \tag{1}$$

$$g_n = \sqrt{\left(g_n^x\right)^2 + \left(g_n^y\right)^2 + \left(g_n^z\right)^2} \tag{2}$$

where, a_n^x, a_n^y, and a_n^z represent the acceleration of the three axes of the n^{th} sampling point x, y, and z, respectively; and g_n^x, g_n^y, and g_n^z represent the angular velocity of the three axes of the nth sample point x, y, and z, respectively.

Each unit action was taken as a sample in this study. Among them, N represents the number of sampling points in each unit action, then each sample is a dimensional matrix (the composition of the 8-dimensional vector includes the acceleration and angular velocity of the three axes, as well as the combined acceleration and the combined angular velocity). The calculation of the mean value and variance of a certain component of the acceleration in the unit action are as Equations (3) and (4), respectively.

$$\mu_a = E\left(a\right) = \frac{1}{N}\sum\nolimits_{i=1}^{N} a_i \tag{3}$$

$$\delta^2 = \frac{1}{N}\sum\nolimits_{i=1}^{N} \left(a_i - \mu_a\right)^2 \tag{4}$$

where, a represents a certain component of acceleration.

The signal is converted from the time domain to the frequency domain through discrete Fourier transform, Fourier transform result $S_{DFT}\left(n\right)$ of the n^{th} sampling point is calculated as Equation (5), and the corresponding frequency f after Fourier transform is

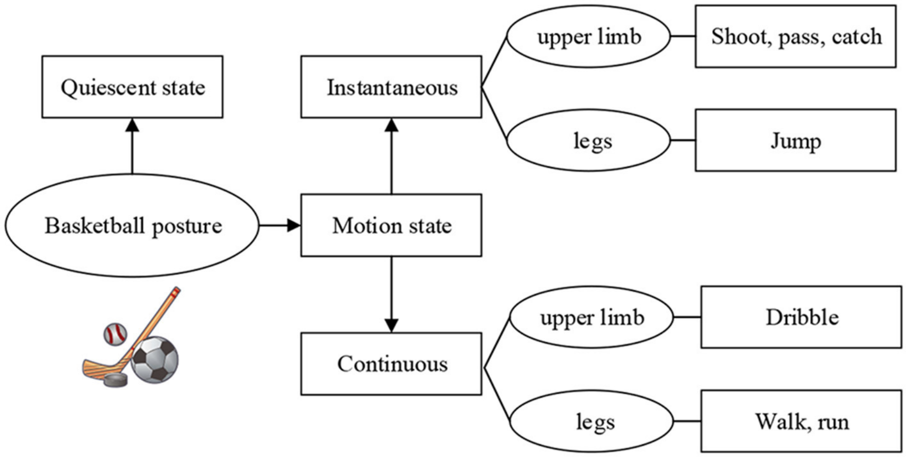


FIGURE 1 | The composition of the sports posture of basketball training.

calculated as Equation (6).

$$S_{DFT}(n) = \sum_{i=1}^{N} a_i e^{-j\frac{2\pi}{N} in} \tag{5}$$

$$f = k \times \frac{f_s}{N} \tag{6}$$

where, j represents the imaginary unit, k represents the sampling point corresponding to the peak value of the Fourier transform, and f_s represents the sampling frequency of the sensor.

After a series of operations such as data collection, preprocessing, and feature extraction, a set of feature vectors describing basketball actions can be obtained, and the classification to which these abstract features belong can be obtained using the classifier model (Hildebrandt, 2019).

The process of basketball sport recognition mainly includes data acquisition, data preprocessing, data division, feature extraction, and classifier training, as shown in **Figure 2**. During data acquisition, the physiological signal or physical signal of individual athletes is generally collected through sensor equipment. During data preprocessing, the data collected are desiccated and normalized to obtain more accurate signals. During data partition, the data extraction of single action in time domain and frequency domain is realized, and the data characteristics are analyzed separately. In the next step of feature extraction, the unit action is analyzed, and the attribute features are extracted and taken as samples. At the final classifier stage, the sample data are constructed into a classification model according to different classification principles to complete the classification of the sample data (Mejia-Ruda et al., 2018; Mullard, 2019; Starke et al., 2020).

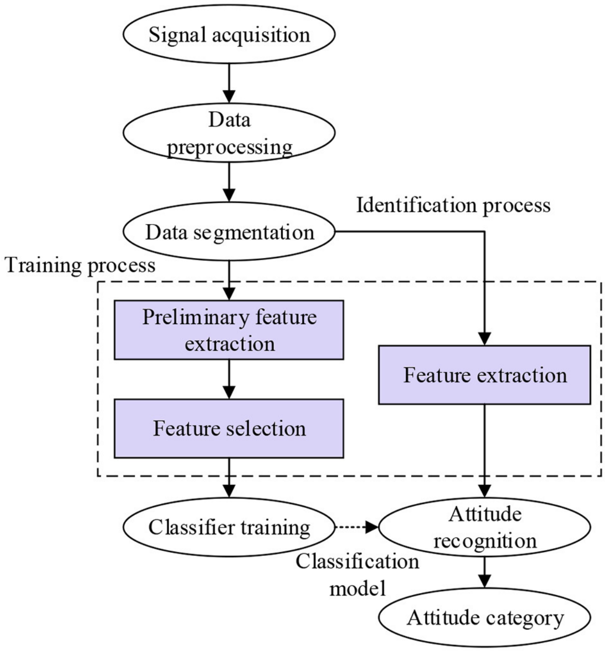


FIGURE 2 | The process of basketball sport recognition.

Intelligent Path Autonomous Planning Based on Machine Learning Algorithm

From the perspective of sports cognition, the decision-making of basketball players directly affects the tactical performance and game result of the whole team (Žemgulys et al., 2018; Kim and Lee, 2020). Therefore, players are required to be able to capture the basketball target in real time and complete relevant information processing when make decisions. Not only do they need to lock in the basketball target that is active, but they also need to make decisions as quickly as possible. Planning the trajectory of real-time change can improve the hit rate. The shooting action was taken as an example in this study, and a clear route should be established before athlete shot a basketball shot. With the parabolic form taken as the premise, a reasonable angle needs to be adjusted in the global path planning (Pan and Li,

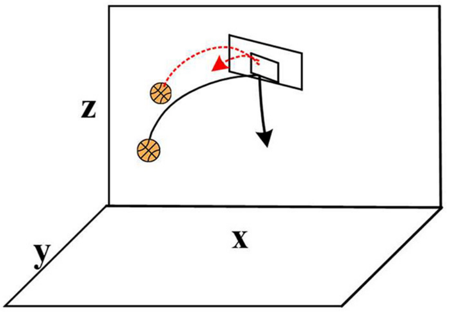


FIGURE 3 | Correct and incorrect basketball path planning results.

TABLE 1 | The best shot angle and minimum shot speed for different shot heights and vertical distances.

Basketball shooting height (m)	Vertical distance needed to move (m)	Minimum release angle (°)	Minimum release speed (m/s)
2.89	0.16	46.09	6.55
2.68	0.37	47.58	6.72
2.25	0.80	50.49	7.06
2.06	0.98	51.56	7.21
1.94	1.10	52.26	7.31
1.76	1.29	53.55	7.45
1.72	1.34	53.78	7.50

2020). The correct and incorrect basketball path planning results are shown in **Figure 3**.

The learning system mainly implements the decision-making and motion planning of the intelligent basketball robot through environmental map construction and path planning. Machine learning includes supervised learning, unsupervised learning, and reinforcement learning (Zhu et al., 2019; Zhang et al., 2020). Among which, reinforcement learning emphasizes how to evaluate selection actions based on the environment to maximize the expected benefits. Through continuous trial and error, the system gradually improves the action selection strategy, and finally achieves the purpose of learning. Q-Learning learning algorithm is one of the most widely used important algorithms in reinforcement learning. A robot based on the Q-Learning algorithm only knows the set of actions that can be selected at the moment. To represent the action reward value from one to the next state, a corresponding matrix is usually constructed, from which the Q matrix that can guide the robot's activities is obtained (Hung et al., 2018; Tieck et al., 2019; Wong et al., 2019).

First, the value *Q(s,a)* should be initialized. In the current state, the robot selected a strategy according to the action, the next state was obtained after the strategy-guided exercise, and then the above selection process was repeated according to the *Q(s,a)* value corrected by the update rule until the end of the learning. The Q-Learning algorithm needed to choose through continuous trial and error and action selection. Therefore, only through continuous correction of feedback information could the final suitable strategy be obtained, and that's why the algorithm had slow update speed (Sombolestan et al., 2019).

In the Q-Learning algorithm, the ε-greedy strategy was utilized to balance the exploration, but the value of the exploration factor ε would decrease as the algorithm training time increased (Maryasin et al., 2018). The purpose of balancing the exploration process was obtained by dynamically updating the value of the exploration factor ε. The number of successful path-finding and the number of different paths found by the robot were taken as the basis for controlling the ε value. The robot had a low number of successful pathfinding initially, so it should maintain a high ε value at the beginning and keep trying new actions. When the robot was gradually familiar with the environment, the number of successful pathfinding would exceed the threshold, and then the robot state action value function would stabilize.

When the Q-Learning algorithm was applied to the robot path planning, in the initial state s_t, a certain action a_t was selected according to the strategy; when the state was transferred to the state s_{t+1}, an immediate return could be obtained. The Q reality can be expressed as Equation (7).

$$Q\left(s_t, a_t\right) = r_{t+1} + \gamma \underset{\alpha}{max} \left(s_{t+1}, a_{t+1}\right) \tag{7}$$

When the robot chose the next state, the probability method could be adopted to select the action to avoid maximizing the deviation. The calculation of probability is expressed as Equation (8).

$$P\left(s|a_k\right) = \frac{Q\left(s, a_k\right)}{\sum_j Q\left(s, a_j\right)} \tag{8}$$

As for the slow update speed of Q-Learning algorithm, a counting threshold was supplemented. The update of the Q value was determined according to the cumulative access times to the "state-action pair" <s, a>. When the cumulative access times reached the threshold, the Q value of "state-action pair" would be updated to improve the efficiency of the algorithm.

Basketball Trajectory Model

A key indicator for evaluating basketball players is shooting accuracy, which is also the primary issue for most basketball players to improve their competitive ability. If the basketball movement is quantified and the data of basketball movement is calculated from the perspective of mechanics, the athletes can be trained scientifically according to the numerical results (Gonzalez et al., 2018). This not only improves the training efficiency of athletes, but also avoids fatigue sports injuries caused by additional training. In the shooting action, the basketball shot from the optimal shooting corner not only consumes the least effort, but also the basketball has the farthest flight distance and the highest scoring rate (El-Shamouty et al., 2019). In basketball training, if athletes can find their optimal shooting angle and form muscle inertia through repeated training, the athlete's training level can be greatly improved.

The main external influence on basketball trajectories was air resistance, so the basketball trajectory in the absence of is mainly

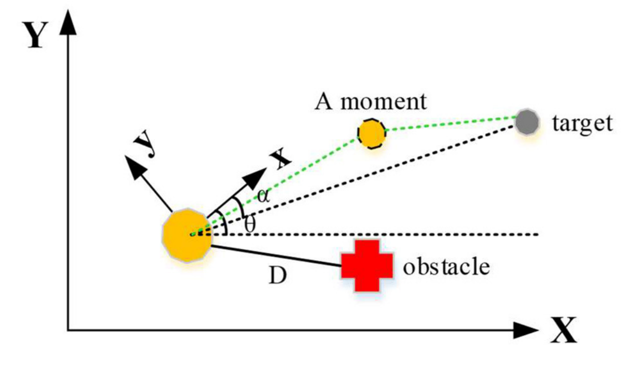


FIGURE 4 | The coordinate system of the moving robot system.

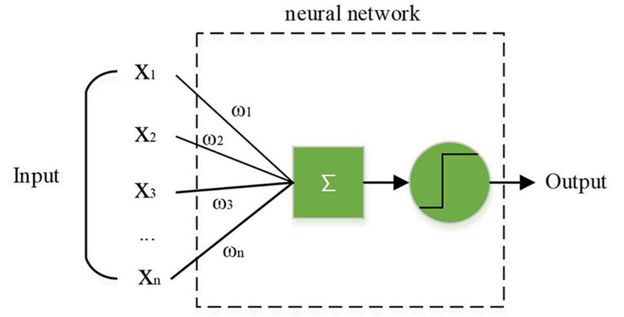


FIGURE 5 | Artificial neuron model.

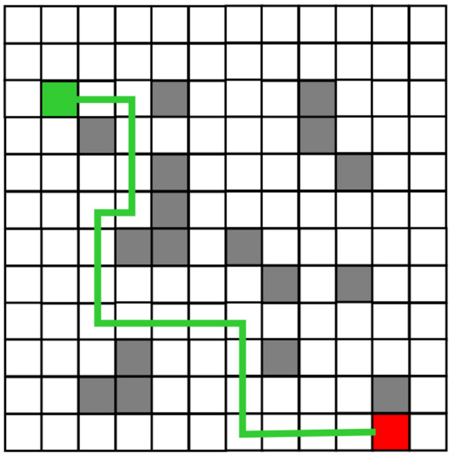


FIGURE 6 | Obstacle Avoidance Path Planning of the Traditional Q-Learning Algorithm before Improvement.

analyzed in this study. The basketball and the athletes with arms outstretched was taken as a whole for measurement. Among this, x and y represent the uniform movement in the horizontal and vertical directions, respectively, and the Equation (9) can be obtained.

$$\begin{cases} x = vt\cos\alpha \\ y = -gt/2 + vt\sin\alpha \end{cases} \quad (9)$$

where, v represents the speed when the basketball is shot, and α represents the angle when the basketball is shot.

The common diameter standard for basketball is 24.6 cm, the diameter of the standard basket is 0.45 meters, and the basket is 3.05 meters above the ground. The backboard specifications are 1.80 meters in length, 1.05 meters in vertical height, and 0.03 meters in thickness. The horizontal distance from the center of the basket is 4.21 m for free throws, and 6.25 m for three-pointers. On the free throw line, the optimal shooting angle and the minimum shooting speed can be calculated according to the height of the basketball shot and the vertical distance needed to move, as shown in **Table 1**.

It is obvious that the optimal shooting angle of the basketball decreases with increasing shooting height and increases with increasing shooting speed. Therefore, for the taller players, the basketball shooting angle or speed should be reduced appropriately during the jump shot. By shortening the flight arc of basketball and reducing the external influence of environmental factors as much as possible, the players can improve their shooting percentage. In addition, due to the increase in the height of shooting, the optimal shooting angle of basketball has a great change. In the jump shot training, the sensitivity of the athlete's fingers should be improved.

Design of Obstacle Avoidance Algorithm Based on Fuzzy Control of Motion Robot

Fuzzy control simulates human behavior and then makes decisions based on fuzzy reasoning. The core of fuzzy control is fuzzy controller, which is mainly responsible for fuzzy processing of input variables, rule design and reasoning, and de-fuzzy. Generally, a complete fuzzy controller consists of four modules: fuzzy interface, knowledge base, reasoning mechanism, and de-fuzzy. The precision of the fuzzy controller completely determines the performance of the entire fuzzy control system to a certain extent (Alonso-Martín et al., 2017). First of all, the precise amount of input can only be utilized by the fuzzy controller after the fuzzy transformation of the variable. The finer the fuzzy subset division, the more the fuzzy rules. The inference machine realizes fuzzy inference by selecting a certain inference algorithm and combining fuzzy rules, and finally completes the output of the control quantity. After fuzzy reasoning, a fuzzy set can be obtained, and it can be applied by the control system only by defuzzing the set into precise values.

The control structure of the moving robot designed in this work is divided into two parts, that is, to approach the target point and to avoid obstacles. First, a two-dimensional coordinate system is established for the mobile environment of the moving robot, including global coordinate system and local coordinate system, as shown in **Figure 4**. The distance between the robot and the target point is D, the direction angle of the robot is θ, and the direction angle between the robot and the target point is α.

The activities of a moving robot are divided into two kinds: the obstacle-free tendency behavior and the obstacle-avoiding behavior, both of which are realized by fuzzy controller. During the barrier-free approach to the target point, the distance of the robot from the target point, the difference between the direction angle of the mobile robot and the direction angle of the target point, and the difference between the heading angle and the direction angle of the target are taken as input, then the angular velocity and linear velocity of the robot are output. The linear speed and angle are controlled, so that the robot can constantly adjust posture and approach the target. In the obstacle avoidance behavior with obstacles, the ultrasonic signals acquired around the robot are taken as input to ensure that the robot can effectively avoid obstacles.

The obstacles are set to be at the right front and left front of the robot, respectively, and two fuzzy controllers are applied. When the robot detects an obstacle, it first judges the specific orientation and then chooses to enter the obstacle-avoidance fuzzy controller to calculate the speed that can avoid the obstacle. When the robot leaves the obstacle, it switches to the fuzzy controller which tends to the target point and moves toward the target. According to the fuzzy segmentation of each input variable, the fuzzy control rules are as follows. If the obstacle is at the right of the robot, it is set to turn counterclockwise. If the obstacle is at the left of the robot, it is set to turn clockwise. The linear velocity and angular velocity of the robot are affected by the distance from the obstacle. When the robot is far away from the obstacle, its output linear velocity and angular velocity are larger. When the robot is close to the obstacle, the output linear velocity and angular velocity are smaller.

In the actual basketball game, since the defender is dynamic, dynamic obstacles should be adopted in the training strategy. An artificial nerve is an abstract model that can be implemented by a circuit or control program, as shown in **Figure 5**. In this model, neurons receive signals from more than one neuron and use them as input. After these signals are compared through the threshold, they are finally processed by the activation function to produce the output of the neuron (Fu et al., 2019; Lesort et al., 2020). Artificial neuron model can simulate the working principle of neuron cells, and get better decision-making data through constant intelligent training, so as to improve the decision-making efficiency and level.

Robot Simulation Experiment

Windows 10 operating system and Inter Core I7-2600 were adopted for the simulation experiment in this study. The path planning algorithm of the robot system was written in Matlab programming language. The traditional Q-Learning algorithm adopted Markov decision process for modeling. The learning rate was initialized to 0.01, the deduction factor was set to 0.8, and the initial value of the exploration factor was set to 0.4. Adopting convolutional neural network to approximate the Q value function, that is, utilizing Q-Network to represent the Q value. The aim of marking the deviation value of the signal and the output of the convolutional neural network is to minimize the loss function. Therefore, a certain training sample is needed, including a large amount of labeled data. Then, backpropagation and gradient descent are adopted to update the parameters of the convolutional neural network. The experience playback is added. First, the sample information discovered by the system is stored in experience pool D. The sample information is a four-tuple consisting of the current state S_t, the current state action value a_t, the immediate reward r_t obtained by the current action, and the next state S_{t+1}. During training, a set of samples is randomly selected from the samples stored in the experience pool D through the experience playback mechanism, and then the gradient descent method is adopted for iterative learning.

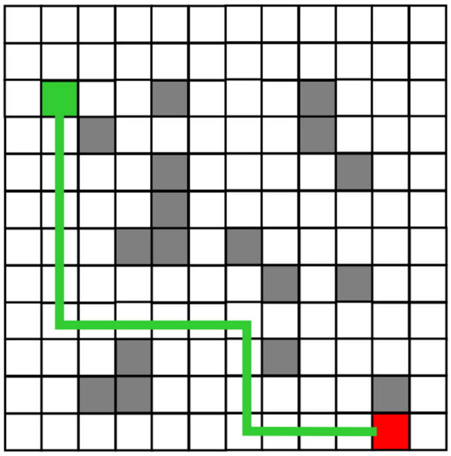


FIGURE 7 | Obstacle Avoidance Path Planning of the Improved Q-Learning algorithm.

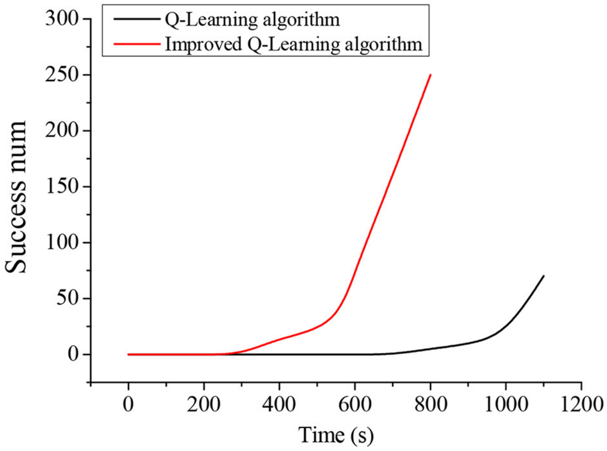


FIGURE 8 | Comparison of the Convergence time of Q-Learning Algorithm before and after Improvement.

$s^i = \left(s_1^i, s_2^i, \dots s_n^i\right)$ represents the obstacle avoidance path of the i-th robot. However, since the number of steps (length) of the path is different, the robot with few steps is expanded, and the target state is filled into the static obstacle avoidance path of a single robot. In addition, since the state vector is adopted in the multi-robot system, the static obstacle avoidance path of each robot is combined into a state vector for representation. First, the Q value is initialized to 0. When the multi-robot system performs $\left(s_1^1, s_1^2, s_1^3\right)$ to $\left(s_2^1, s_2^2, s_2^3\right)$, the action value function of

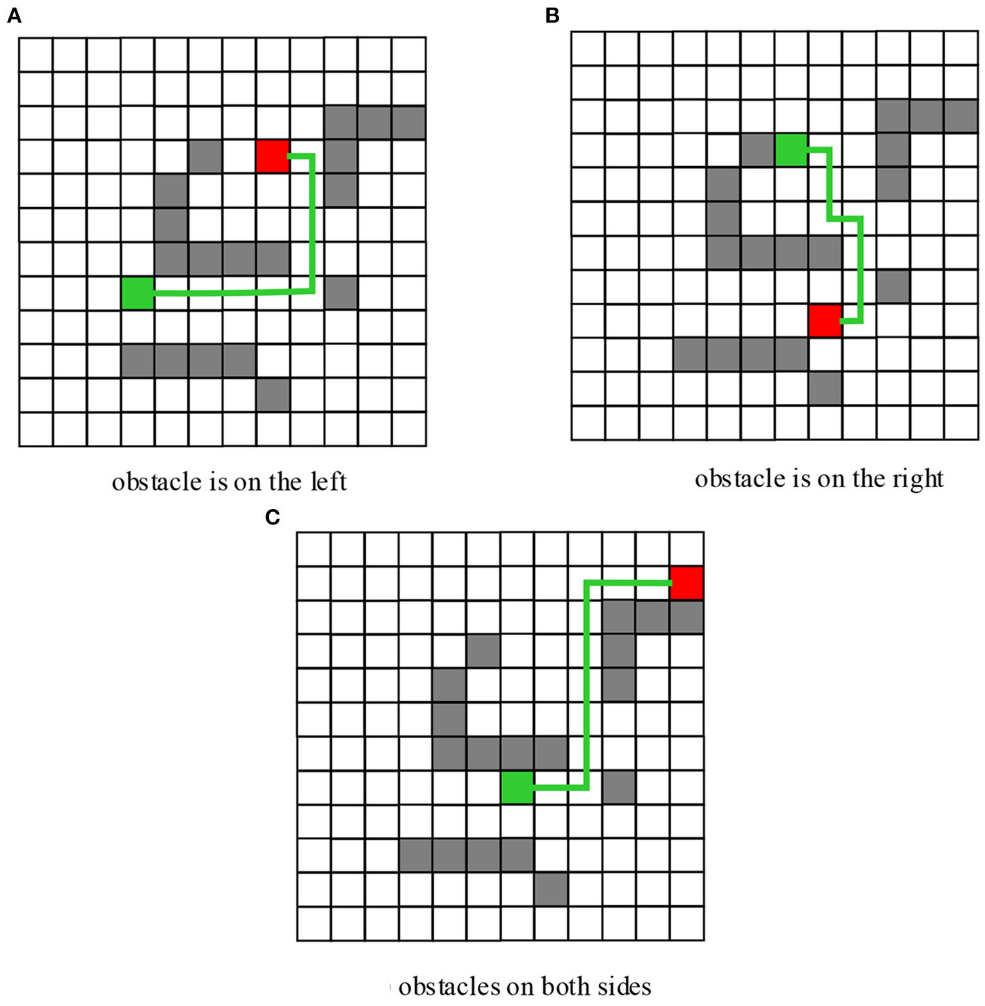


FIGURE 9 | Simulation Results of Fuzzy Obstacle Avoidance of the Robot. **(A)** Obstacle is on the left. **(B)** Obstacle is on the right. **(C)** Obstacles on both sides.

state transition is set to a reasonable value >0. It can force the multi-robot system to have a certain understanding of the environment and tend to choose the optimal static obstacle avoidance path instead of randomly trying actions.

Besides, the 12 × 12 grid map was adopted, and the location of obstacles was random. The traditional Q-Learning algorithm and the improved Q-Learning algorithm were utilized in this study to plan the motion path of the robot, and the two algorithms were compared with regard to convergence performance.

RESULTS

The Algorithm Performance Analysis of the Improved Robot in Path Planning

The robot planning path obtained from the simulation experiment of the traditional Q-Learning algorithm and the improved Q-Learning algorithm are shown in **Figures 6**, **7**, respectively. Among these, black represents obstacles, and yellow grid represents the robot movement path. There are more obstacle avoidance path turning points obtained by the traditional Q-Learning algorithm. The obstacle avoidance path based on the improved algorithm is smoother, indicating that the improved Q-Learning algorithm is more suitable for robots to solve obstacle avoidance problems.

The time it takes for the traditional Q-Learning algorithm and the improved Q-Learning algorithm to converge in the training process is shown in **Figure 8**. It takes about 700 s for the robot to find the obstacle avoidance path to the target for the first time under the traditional Q-Learning algorithm, but the improved Q-Learning algorithm-based robot only needs about 250 s. Moreover, it is difficult for the traditional Q-Learning algorithm to find a path to the target in the initial training stage, while the improved Q-Learning algorithm can find a path to the target faster during the initial training.

Obstacle Avoidance Function Test of the Robot

Matlab programming was adopted to simulate obstacle avoidance behavior of the robot. The simulation results of the obstacle avoidance behavior when there are obstacles exist on the left, right, and left and right sides at the same time were shown in **Figure 9**. The starting heading angle was set to 60°, and the starting position and final target were randomly selected. According to the simulation experiment results, the basketball robot fuzzy controller established in this work can effectively avoid obstacles encountered during the robot movement. Moreover, since multiple fuzzy controllers were combined, the fuzzy rules were simplified, it was easier for the fuzzy control to implement, and the obtained motion trajectory curve was relatively smooth.

CONCLUSION

In order to improve the practical adoption efficiency of basketball training strategies, and avoid chronic or acute injuries caused by blind training and collisions in basketball training, in this study, the scientific and effective training of athletes were improved based on the adoption of basketball intelligent robots. Firstly, according to the basic training actions in basketball training, the sport recognition in basketball training was analyzed, and in-depth analysis of the shooting action was implemented. In the training of the shooting action, a mathematical model was utilized to simulate the flying situation of the basketball without resistance, and the improved Q-Learning algorithm based on machine learning was proposed to plan the path of the sports robot. The improved algorithm supplemented a count threshold, and adopted the cumulative access times to the "state-action pair" <s, a> to determine the update of the Q value. In path planning, the fuzzy controller was applied to make the robot complete the approach to the target point and avoid obstacles at the same time. Sun et al. (2018) and Zheng and Liu (2020) showed that an optimized fuzzy control algorithm based on path planning could overcome the problem of excessively subjective fuzzy boundary selection and generate the optimal path, which was basically consistent with this research.

In the simulation experiment, the path planning of the robot through the improved machine learning algorithm shows that the improved Q-Learning algorithm can find a path to the target faster during initial training (Zheng and Ke, 2020). Moreover, the convergence time of the algorithm is considerably shorter than that of the traditional algorithm. In the subsequent obstacle avoidance performance test, the established basketball robot fuzzy controller can effectively avoid obstacles encountered during the robot movement. In addition, due to the combination of multiple fuzzy controllers, the fuzzy rules are simplified and the fuzzy control is easier to implement, making the obtained motion trajectory curve relatively smooth. In the research of the motion robot system, this work only discusses the motion planning of a single robot, but doesn't analyze the motion planning of the multi-robot system, which will be the focus of the next research.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

REFERENCES

Alonso-Martín, F., Gamboa-Montero, J. J., Castillo, J. C., CastroGonzalez, A., and Salichs, M. (2017). Detecting and classifying human touches in a social robot through acoustic sensing and machine learning. *Sensors* 17:1138. doi: 10.3390/s17051138

Bing, Z., Lemke, C., Morin, F. O., Jiang, Z., Cheng, L., Huang, K., et al. (2020). Perception-action coupling target tracking control for a snake robot via reinforcement learning. *Front. Neurorobot*. 14:79. doi: 10.3389/fnbot.2020.591128

Chau, V. H., Vo, A. T., and Le, B. T. (2019). The effects of age and body weight on powerlifters: An analysis model of powerlifting performance based on machine learning. *Int. J. Comput. Sci. Sport* 18, 89–99. doi: 10.2478/ijcss-2019-0019

Coglianese, C., and Lehr, D. (2017). Regulating by robot: administrative decision making in the machine-learning era. *Soc. Sci. Electr. Publ*. 105, 1147–1223.

El-Shamouty, M., Kleeberger, K., and Lämmle, A. (2019). Simulation-driven machine learning for robotics and automation. *Technisches Messen* 86, 673–684. doi: 10.1515/teme-2019-0072

Fu, Y., Jha, D. K., Zhang, Z., Yuan, Z., and Ray, A. (2019). Neural network-based learning from demonstration of an autonomous ground robot. *Machines* 7:24. doi: 10.3390/machines7020024

Gonzalez, R., Fiacchini, M., and Iagnemma, K. (2018). Slippage prediction for off-road mobile robots via machine learning regression and proprioceptive sensing. *Rob. Auton. Syst*. 105, 85–93. doi: 10.1016/j.robot.2018.03.013

Hildebrandt, M. (2019). Privacy as protection of the incomputable self: from agnostic to agonistic machine learning. *Theoret. Inquiries Law* 20, 83–121. doi: 10.1515/til-2019-0004

Hung, A. J., Chen, J., Che, Z., and Nilanon, T. (2018). Utilizing machine learning and automated performance metrics to evaluate robot-assisted radical prostatectomy performance and predict outcomes. *J. Endourol*. 32, 438–444. doi: 10.1089/end.2018.0035

Ji, R. (2020). Research on basketball shooting action based on image feature extraction and machine learning. *IEEE Access* 8, 138743–138751. doi: 10.1109/ACCESS.2020.3012456

Kim, J., and Lee, H. (2020). Adaptive human–machine evaluation framework using stochastic gradient descent-based reinforcement learning for dynamic competing network. *Appl. Sci*. 10:2558. doi: 10.3390/app10072558

Lesort, T., Lomonaco, V., Stoian, A., Maltonic, D., Filliata, D., and Díaz-Rodríguez, N. (2020). Continual learning for robotics: definition, framework, learning strategies, opportunities and challenges. *Inform. Fusion* 58, 52–68. doi: 10.1016/j.inffus.2019.12.004

Li, T. H. S., Kuo, P. H., Chang, C. Y., Hsu, H. P., Chen, Y. C., Chang, C. H., et al. (2019). Deep belief network–based learning algorithm for humanoid robot in a pitching game. *IEEE Access* 7, 165659–165670. doi: 10.1109/ACCESS.2019.2953282

Li, T. H. S., Kuo, P. H., Ho, Y. F., Liu, C. Y., and Fang, N. C. (2016). Robots that think fast and slow: an example of throwing the ball into the basket. *IEEE Access* 4, 5052–5064. doi: 10.1109/ACCESS.2016.2601167

Liu, L., and Hodgins, J. (2018). Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Transac. Graph*. 37, 1–14. doi: 10.1145/3197517.3201315

Lobos-Tsunekawa, K., Leiva, F., and Ruiz-del-Solar, J. (2018). Visual navigation for biped humanoid robots using deep reinforcement learning. *IEEE Robot. Automat. Lett*. 3, 3247–3254. doi: 10.1109/LRA.2018.2851148

Lv, Z. (2020). Virtual reality in the context of Internet of Things. *Neural Comput. Appl*. 32, 9593–9602. doi: 10.1007/s00521-019-04472-7

Maryasin, B., Marquetand, P., and Maulide, N. (2018). Machine learning for organic synthesis: are robots replacing chemists? *Angew. Chem. Int. Ed*. 57, 6978–6980. doi: 10.1002/anie.201803562

Mejia-Ruda, E., Jimenez-Moreno, R., and Hernandez, R. D. (2018). Topological data analysis for machine learning based on kernels: a survey. *Int. J. Appl. Eng. Res*. 13, 13268–13271.

Mullard, A. (2019). Machine learning brings cell imaging promises into focus. *Nat. Rev. Drug Discov*. 18, 653–656. doi: 10.1038/d41573-019-00144-2

Narayanan, A., Desai, F., and Stewart, T. (2020). Application of raw accelerometer data and machine-learning techniques to characterize human movement behavior: a systematic scoping review. *J. Phys. Activity Health* 17, 360–383. doi: 10.1123/jpah.2019-0088

Pan, Z., and Li, C. (2020). Robust basketball sports recognition by leveraging motion block estimation. *Signal Proc. Image Commun*. 83:115784. doi: 10.1016/j.image.2020.115784

Sombolestan, S. M., Rasooli, A., and Khodaygan, S. (2019). Optimal path-planning for mobile robots to find a hidden target in an unknown environment based on machine learning. *J. Ambient Intell. Humaniz. Comput*. 10, 1841–1850. doi: 10.1007/s12652-018-0777-4

Starke, S., Zhao, Y., Komura, T., and Zaman, K. (2020). Local motion phases for learning multi-contact character movements. *ACM Transac. Graph*. 39, 1–13. doi: 10.1145/3386569.3392450

Stübinger, J., Mangold, B., and Knoll, J. (2020). Machine learning in football betting: prediction of match results based on player characteristics. *Appl. Sci*. 10:46. doi: 10.3390/app10010046

Sun, B., Zhu, D., and Yang, S. X. (2018). An optimized fuzzy control algorithm for three-dimensional AUV path planning. *Int. J. Fuzzy Syst*. 20, 597–610. doi: 10.1007/s40815-017-0403-1

Tieck, J. C. V., Schnell, T., Kaiser, J., Mauch, F., Roennau, A., and Dillmann, R. (2019). Generating pointing motions for a humanoid robot by combining motor primitives. *Front. Neurorobot*. 13:77. doi: 10.3389/fnbot.2019.00077

Wong, N. C., Lam, C., Patterson, L., and Shayegan, B. (2019). Use of machine learning to predict early biochemical recurrence after robot-assisted prostatectomy. *BJU Int*. 123, 51–57. doi: 10.1111/bju.14477

Yoon, Y., Hwang, H., Choi, Y., Joo, M., and Hwang, J. (2019). Analyzing basketball movements and pass relationships using realtime object tracking techniques based on deep learning. *IEEE Access* 7, 56564–56576. doi: 10.1109/ACCESS.2019.2913953

Žemgulys, J., Raudonis, V., Maskeliunas, R., and DamazEviaIus, R. (2018). Recognition of basketball referee signals from videos using Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM). *Proc. Comput. Sci*. 130, 953–960. doi: 10.1016/j.procs.2018.04.095

Zhang, J., Lu, C., Wang, J., Yue, X. G., and Tolba, A. (2020). Training convolutional neural networks with multi-size images and triplet loss for remote sensing scene classification. *Sensors* 20:1188. doi: 10.3390/s20041188

Zhao, Y., Yang, R., Chevalier, G., Shah, R. C., and Romijnders, R. (2018). Applying deep bidirectional LSTM and mixture density network for basketball trajectory prediction. *Optik* 158, 266–272. doi: 10.1016/j.ijleo.2017.12.038

Zheng, Y., and Ke, H. (2020). The adoption of scale space hierarchical cluster analysis algorithm in the classification of rock-climbing teaching evaluation system. *J Ambient Intell. Hum. Comput*. doi: 10.1007/s12652-020-01778-6

Zheng, Y., and Liu, S. (2020). Bibliometric analysis for talent identification by the subject–author–citation three-dimensional evaluation model in the discipline of physical education. *Library Hi Tech*. doi: 10.1108/LHT-12-2019-0248

Zhu, Y., Zhou, S., Gao, D., and Liu, Q. (2019). Synchronization of nonlinear oscillators for neurobiologically inspired control on a bionic parallel waist of legged robot. *Front. Neurorobot*. 13:59. doi: 10.3389/fnbot.2019.00059

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

BRIEF RESEARCH REPORT
published: 25 February 2021
doi: 10.3389/fnbot.2021.624466

Target Recognition of Industrial Robots Using Machine Vision in 5G Environment

Zhenkun Jin[1], *Lei Liu*[2], *Dafeng Gong*[3]* *and Lei Li*[4]

[1] Department of Information Engineering, Wuhan Business University, Wuhan, China, [2] Graduate School, Gachon University, Seoul, South Korea, [3] Department of Information Technology, Wenzhou Polytechnic, Wenzhou, China, [4] Huawei Technologies Co. Ltd., Shenzhen, China

The purpose is to solve the problems of large positioning errors, low recognition speed, and low object recognition accuracy in industrial robot detection in a 5G environment. The convolutional neural network (CNN) model in the deep learning (DL) algorithm is adopted for image convolution, pooling, and target classification, optimizing the industrial robot visual recognition system in the improved method. With the bottled objects as the targets, the improved Fast-RCNN target detection model's algorithm is verified; with the small-size bottled objects in a complex environment as the targets, the improved VGG-16 classification network on the Hyper-Column scheme is verified. Finally, the algorithm constructed by the simulation analysis is compared with other advanced CNN algorithms. The results show that both the Fast RCN algorithm and the improved VGG-16 classification network based on the Hyper-Column scheme can position and recognize the targets with a recognition accuracy rate of 82.34%, significantly better than other advanced neural network algorithms. Therefore, the improved VGG-16 classification network based on the Hyper-Column scheme has good accuracy and effectiveness for target recognition and positioning, providing an experimental reference for industrial robots' application and development.

Keywords: machine vision, artificial intelligence, deep learning, industrial robot, 5G environment

OPEN ACCESS

Edited by:
Mu-Yen Chen,
National Taichung University of Science and Technology, Taiwan

Reviewed by:
Hsin-Te Wu,
National Ilan University, Taiwan
Chia-Chen Chen,
National Chung Hsing University, Taiwan

****Correspondence:***
Dafeng Gong
gongdafeng@gmail.com

Received: *31 October 2020*
Accepted: *03 February 2021*
Published: *25 February 2021*

Citation:
Jin Z, Liu L, Gong D and Li L (2021) Target Recognition of Industrial Robots Using Machine Vision in 5G Environment.
Front. Neurorobot. 15:624466.
doi: 10.3389/fnbot.2021.624466

INTRODUCTION

In the early 1960s, the United States was the first country in the world to manufacture industrial robots. After that, industrial robot technology and its products developed rapidly. At present, industrial robots have provided productivity tools for automation in various industries (Li et al., 2018). At the same time, the growing use of artificial intelligence technology in industrial robots that have been widely used has dramatically changed the way people produce and live.

An industrial robot is a machine that automatically performs industrial production tasks. It can accept commands, perform corresponding functions according to programming procedures, or operate according to the principles of artificial intelligence technology. Its main applications include plundering, assembly, acquisition, placement, product inspection, and testing. The ability of industrial robots to perform these functions relies on accurate detection and identification of targets (Dönmez et al., 2017; Fernandes et al., 2017; Song et al., 2017; Zhang et al., 2017). The basis for realizing the above functions is the industrial robot vision system, which realizes the visual function of the robot through a computer, enabling the robot to recognize the objective world. Therefore, the vision system plays an important role in improving the performance of industrial robots. Industrial robot vision systems are used for target detection and task identification. Image feature extraction technology is the basis of machine vision. It refers to the extraction of certain key information that can be used to represent images from image data for machine vision tasks such as subsequent recognition and classification (Ertuğrul and Tağluk, 2017; Sung et al., 2017; Wang et al., 2018) With the rapid development of communication technology, artificial intelligence, and machine learning, the precondition of the 5G environment has matured. The new generation of wireless communication technology not only means faster transmission speed but also better security performance for industrial robots (Bogue, 2017). The application of these intelligent algorithms to the development of industrial robots with higher service levels will bring new opportunities for the development of industrial robots.

However, a large amount of existing research is at the level of simple industrial devices. With the diversification of industrial devices, the traditional positioning and recognition algorithm has the problems of serious positioning errors, slow recognition speed, and low accuracy, which leads to the fact that the sorting work in complex environments is still in the manual operation stage. Therefore, to improve the accuracy of industrial robots for object recognition and positioning in complex environments, a visual recognition and localization algorithm based on artificial intelligence deep learning is proposed. The deep learning network model is used to convolve and pool the image layers, and the target classification algorithm is used to optimize the industrial robot visual recognition system. Besides, the Fast R-CNN algorithm and the improved VGG-16 classification network based on the Hyper-Column scheme can identify and classify the target objects in a complex background. The simulation results have shown that the proposed algorithm has good stability and accuracy.

LITERATURE REVIEW

The Development Trend of Deep Learning

With rapid science and technology development, artificial intelligence (AI) has been applied in more fields. As a new intelligent algorithm in the 21st century, DL has been studied by many scientific researchers. In 2006, Hinton, the expert in the field of artificial intelligence of the University of Toronto, proposed the concept of deep learning. He proposed an algorithm for fast training of deep neural networks, which opened up the craze for deep learning in the field of artificial intelligence (Hinton et al., 2017). Deep learning mathematical models generally include deep convolution neural networks using supervised learning methods, and superposition self-coding networks and deep belief networks using hybrid supervised learning methods. Stani et al. (2017) proved that the deep learning algorithm has superior non-linear approximation ability and generalization ability than BP neural network, support vector machine (SVM), and other network models, and can exhibit very powerful performance in complex pattern recognition occasions. To overcome the limitations of existing hybrid precoding schemes, Huang et al. (2019) proposed a mmWave massive MIMO hybrid precoding framework based on deep learning, which regarded each selected pre-coder as the mapping relationship in Deep Neural Network (DNN). Simulations found that the DNN-based method could minimize the bit error rate and improve the spectrum efficiency of mmWave's massive MIMO. While significantly reducing the computational complexity, hybrid precoding could provide better performance than traditional schemes. Sun et al. (2020) proposed an adaptive Deep Learning-aided Digital Pre-Distorter (DL-DPD) using optimized deep Recurrent Neural Networks (RNN). The experimental results proved the effectiveness of the proposed adaptive DL-DPD and revealed that the online system switched sub-DPD modules more frequently than expected.

Research of Target Recognition Technology

Traditional detection and recognition technologies include segmentation-based method, feature analysis method, image recognition, and decision-classification method, and pattern learning and shape matching method, which are widely used in the industrial fields. With AI technologies' development, such as machine learning, the target recognition accuracy has attracted many scholars' attention. You et al. (2017) built a SCARA robot automatic recognition and positioning plug-in system platform based on monocular vision. Through the calibration of the crawling system parameter model and the establishment of camera parameters, the color recognition of the workpiece and the position information of the workpiece could be achieved. Besides, the robot's jaws could be controlled to accurately grasp the target workpiece, meeting the real-time requirements of grasping the workpiece in general industrial production. Sampedro et al. (2019) proposed a fully autonomous aerial robot scheme for performing complicated Search And Rescue (SAR) tasks in an unstructured indoor environment. The algorithm integrated a new type of deep reinforcement learning method to identify and interact with targets in multiple environments. Wang et al. (2020) proposed a small humanoid combat robot design scheme based on the target recognition algorithm for the low recognition rate of mobile robots in complicated working environments. This scheme could fuse information by adding visual information, which was simpler and ran faster and more effectively. Finally, the simulation experiment proved its effectiveness. Li et al. (2020) investigated the control performance of the visual servo system under planar cameras and RGB-D cameras. They segmented the color images collected by RGB-D cameras based on optimized normalized cut sets. A control cycle could be completed by calculating the end angle and speed of the robot, and the entire process was iterated until the servo task was completed. Finally, experiments verified that the proposed RGB-D image processing algorithm had an excellent performance in the above aspects of the visual servo system.

According to the above works, the DL algorithm is applied in many fields, and its application in industrial robot target recognition is scarce. Therefore, based on the DL algorithm, CNN is improved based on the Hyper-Column scheme. The

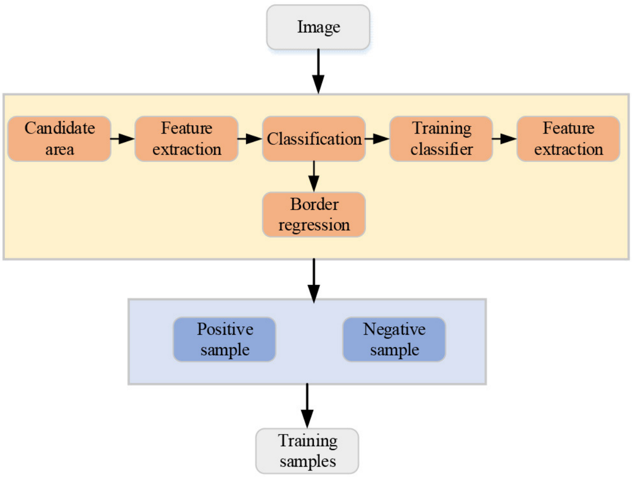


FIGURE 1 | The basic flow chart of target detection technology.

constructed algorithm's performances are analyzed to study industrial robot target recognition, which has significant research value (Wen, 2020).

PROPOSED METHOD

Target Detection Technology Based on Deep Learning of Artificial Intelligence

Deep learning (DL) is one of the technical and research fields of machine learning. By establishing an artificial neural network with a hierarchical structure, artificial intelligence is realized in the computing system. The artificial intelligence target detection technology of images refers to the application of techniques such as machine vision to determine the type, position, size, and confidence of the target object, and the predetermined target object is automatically detected from the image (Wang et al., 2019). The basic process of target detection technology is shown in **Figure 1**.

Researchers have proposed a Fast Region Convolution Neural Network (Fast R-CNN) detection method in recent years, which transfers feature extraction to the convolution feature map of the last layer (Matson et al., 2019). The region convolution neural networks (CNN) need to repeat the problem of multiple convolution calculations for the same image; at the same time, the discrimination of the proposed area and the bounding box regression are integrated into one framework, which effectively improves the accuracy and efficiency of target detection. Fast R-CNN uses a single network joint training convolution neural network, classifiers, and bounding box regenerators to extract image features (CNN), classification (SVM), and compact bounding boxes (regressors). The Fast R-CNN network takes the entire image and a set of candidate frames as input, first using several convolution layers and a maximum pooling layer to process the entire image, producing a convolved feature map, and then for each candidate box, the pooling layer of regions of interest extracts a fixed-length feature vector from the feature map. Each feature vector is fed into a series of fully connected layers, which are finally branched into two output layers; one outputs k object categories and the probability estimation for a background category, and the other one outputs four real values for k object categories. Fast R-CNN solves many problems of R-CNN with faster training speed; however, there are still many problems. Fast R-CNN uses the Selective Search algorithm to extract the candidate regions, while the target detection consumes a lot of time. The candidate region extraction takes 2∼3 s, and the single image detection time reaches 0.32 s. The extraction efficiency of the recommended region is low, which still cannot meet the demand of the real-time application.

Convolution and Pooling of Image Layers by the Artificial Intelligence-Based Deep Learning Network Model

The deep learning network receives the recorded visual feedback image in real-time and performs deep learning processing on each frame of the received image, i.e., the multi-level convolution, pooling operation, and classification processing to obtain the coordinates, angles, and time of the detection target on the image in the image coordinate system, focuses on the coordinates and angle of the detection target on the captured visual feedback image, and sends the processed image and the coordinates, angles, and time information of the detection target on the image to the intermediate result synthesis processing unit; the processed image sent to the human error correction interface is shown in **Figure 2**.

The time mentioned in the above content refers to the image capturing time, the angle is the angle between the target re-axis and the image coordinate system X; the image coordinate system describes the position coordinates of a single-pixel in the whole image, and the pixel point in the upper left corner of the coordinate system is the origin coordinate (0,0), the abscissa axis of the image coordinate system is horizontal, the maximum value is 1,400, the ordinate axis of the image coordinate system is vertical, and the maximum value is 1,040, i.e., the image width is 1400 × 1050.

It is assumed that the numerical matrix of the input image is P_0 and the size is $P \times Q$. In the scheme, 1400 × 1050 is used, and the numerical matrix P_0 and the convolution kernel M_{k1} is convoluted:

$$P_{1k} = P_0 \otimes M_{k1} \tag{1}$$

In the equation, $\otimes$ is the matrix convolution; M_{k1} is the k-th output in the first layer of the deep network, $k = 1\ldots256$, i.e., the first layer consists of 256 feature convolution kernels, parameters of M_{k1} is obtained from VGG-16 model. P_{1k} is the convolution output of the first layer in the deep network, with 256 outputs in total.

The convolution result of the first layer P_{1k} is pooled. In the scheme, the maximum pooling method is adopted, i.e., every 2×2 local matrices in P_{1k} are merged into one element, and the maximum value among the four elements is used as the result;

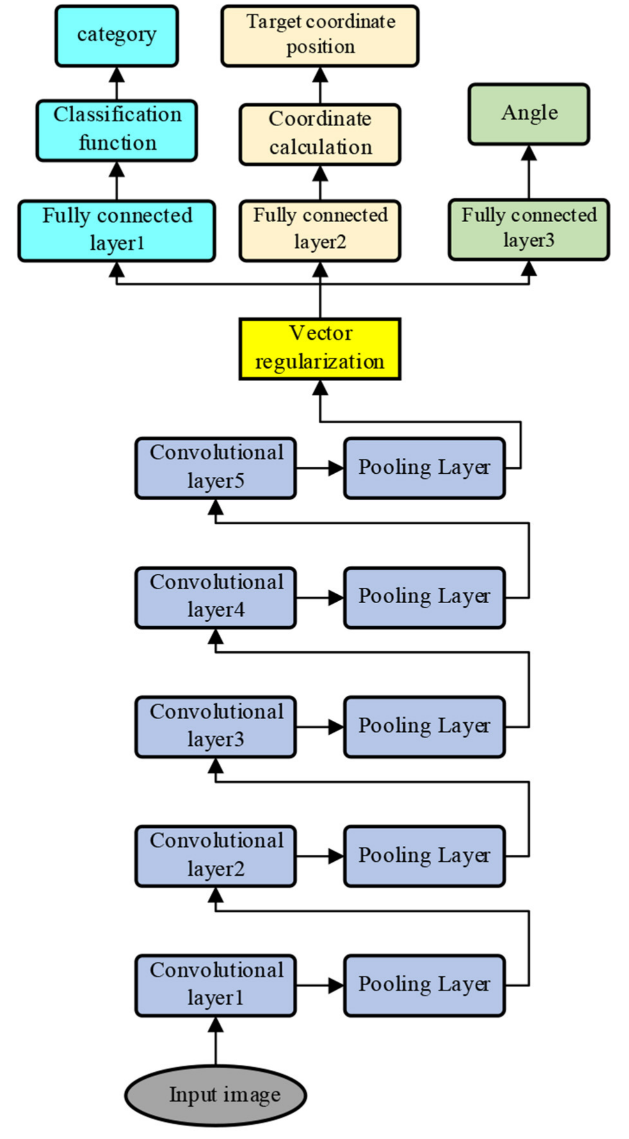


FIGURE 2 | The model of layered convolution kernel pooling of deep learning network.

the step size of pooling is 2. The pooling result P_{1k} is P_{1k} the size of P_{1k} is 1/2 of the original size.

The convolution and pooling result of the first layer P_{1kc} is input to the second layer, and the result of the second layer is P_{2kc}, the results of the l-1 layer $P_{(l-1)kc}$ are sequentially obtained.

The operations of convolution and pooling of the first layer are obtained through recurrence:

$$P_{lk} = P_{(l-1)kc} \otimes M_{kl} \tag{2}$$

In the equation, M_{kl} is the convolution kernel matrix of the k-th feature of the first layer, the parameters M_{kl} are obtained. P_{lk} is the k-th output of the first layer in the deep network.

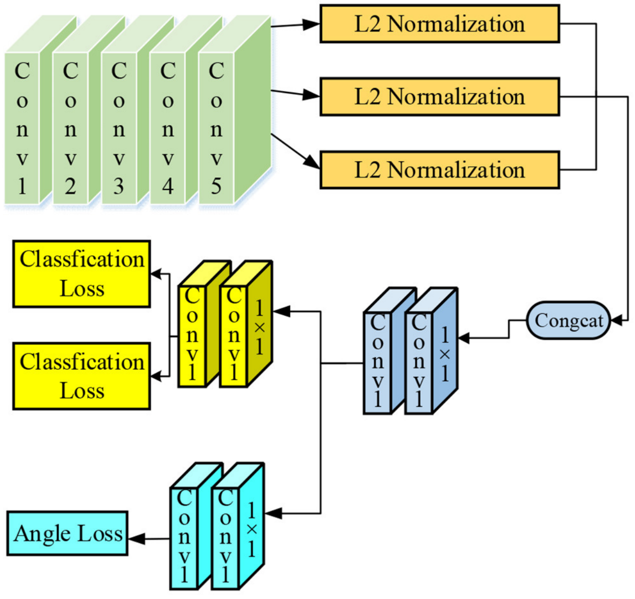


FIGURE 3 | The improved VGG-16 classification model.

Improvements in the Target Classification Algorithm

To accurately identify objects, the Hyper-Column scheme is used. The Hyper-Column scheme refers to a way to fuse the responses of neurons at various levels. It overcomes the information loss caused by multiple convolutions in the high-level convolution network, which makes the small target information in the high-level features be seriously missed and the detection effects on small targets are poor. Meanwhile, due to the addition of the underlying features, the bottom layer features have rich direction information, which can help the angle prediction (Capparelli et al., 2019; Toğaçar et al., 2020). Based on the original VGG-16 architecture, the 3rd, 4th, and 5th layers of the VGG-16 convolution layer are fused so that the model can have good detection robustness of smaller bottle objects on the conveyor belt; meanwhile, the information on the angle prediction of the object is increased by introducing a low-level neural response, thereby reducing the error of the angle prediction.

In summary, the following three improvements are made to the original VGG16 classification network: (1) the VGG-16 network structure with angle prediction is increased, and the response images of Conv3-4, Conv4-4, and Conv5-4 layers are fused to enhance the system performance through the feature fusion method; (2) the traditional classification regression loss is decoupled with angle prediction so that they are independently trained and predicted; and (3) the full connection layer is canceled, and the whole network is trained by using a full convolution network so that the network parameters are reduced to prevent over-fitting, and the training speed of the system is also improved. The improved VGG-16 classification model is shown in **Figure 3**.

TABLE 1 | Parameter settings used in training in the laboratory environment.

Item	Results	Item	Results
Training set	1620 images (image resolution 1400 × 1050)	The test set	162 images (image resolution 1400 × 1050)
Learning rate	0.00114	Momentum	0.9
Number of anchor nodes	901	Weight-decay	0.0005
Dimension of anchor nodes	[600, 1000]	Iterations	Fine-tune: 40000 times
Number of images selected	VGG16 classification network: 2	NMS parameter	0.7
Number of Minni-batch in RPN	64	Detection threshold	0.99
The ratio of the prospect to the backdrop in RPN	[0, 0.3]	Flipped image	The first shift
Intersection-over-Union range of RPN prospect samples	No	Number of anchor nodes	901
Weight of criteria function	Position weight: 1; angle weight: 5	Training the updated layer	The 1st time: update the convolution layer and the FC layer; the 2nd time: update the FC layer. Shared convolution strategy applied

EXPERIMENT

To verify the correctness of the target recognition algorithm, in terms of the experimental method, the design adopts the method of pushing the input image, and the system performs the recognition and detection to verify whether the system can identify the target in the system. In the experiment, the bottle objects are used as the test objects.

Verification of Fast R-CNN Algorithm

In the laboratory environment, 1,800 images are acquired and the collected data are grouped according to the ratio of the training set and test set, which is 9:1. The image resolution of images used in the test is 1400 × 1050. Hyperparameter settings are significant for CNN performance. Many optimization techniques can reduce the neural network model's hyperparameter setting difficulty and manually set hyperparameters. **Table 1** shows the hyperparameter settings adopted in training to enable the CNN framework to obtain qualified prediction results.

In the training process, the proportional selection of positive and negative samples will also have a great impact on training performance. In the task of detecting bottle-shaped objects, the identification of bottles is taken as the research object; thus, the samples are divided into two categories, i.e., bottles and non-bottles. Based on the above situation, the classified output layer of RPN and VGG-16 is set to 2 nodes, and the corresponding VGG-16 regression layer includes $4 \times 2 + 1 = 9$ nodes. The batch size of the RPN network is set to 64. Besides, the ratio of positive and negative samples is 1:1, which can ensure that the image has enough positive samples.

In the experiment, 1,800 images are used for the test, in which the total number of bottles is 260. During the measurement, whether the bottle is successfully detected is determined by the Intersection-over-Union (IoU) of the test result to the true position of the bottle being >0.5. The evaluation criteria used in the experiment are the false detection rate and the missed detection rate, which are defined as Equations (3) and (4), respectively:

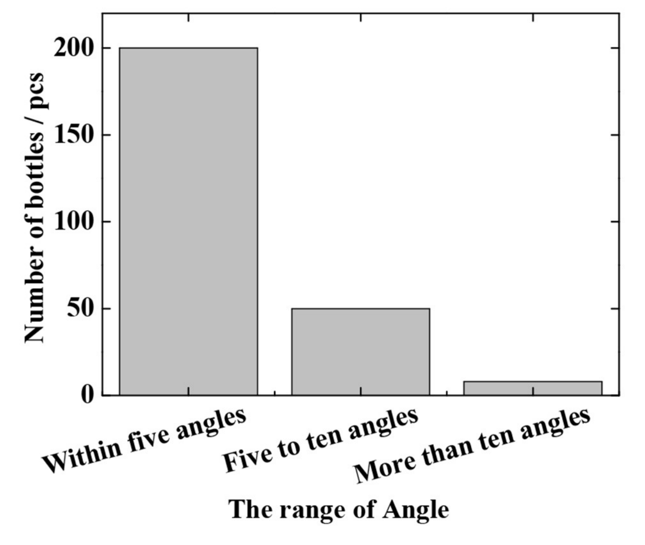


FIGURE 4 | The angle detection result of the improved Fast R-CNN model in the laboratory environment.

$$The\ false\ \text{detection}\ rate = \frac{number\ of\ false\ \text{detection}\ cases}{number\ of\ t\arg ets\ \text{detected}} \times 100\% \tag{3}$$

$$The\ missed\ \text{detection}\ rate = \frac{number\ of\ missed\ \text{detection}\ cases}{the\ total\ number\ of\ t\arg ets} \times 100\% \tag{4}$$

The detection results of the Fast R-CNN algorithm are shown in **Figure 4** and **Table 2**. As can be seen from **Figure 4**, 200 angles are detected within 5 degrees, 50 are detected from 5 degrees to 10 degrees, and 8 are detected above 10 degrees. It can be seen from the results that in the laboratory environment, the predicted angle of the bottle-shaped article is predicted accurately, and the data of **Table 2** shows that the false detection rate and the missed detection rate are 3.2 and 8.7%, respectively. In terms of the detection time, since the CNN network itself has a long forward propagation time, the time is 220 ms.

Figure 5 shows the detected images of objects in the laboratory environment. As can be seen from **Figure 5**, the Fast R-CNN algorithm can accurately detect the position and angle of the bottle.

TABLE 2 | The detection result of the improved Fast R-CNN model in the laboratory environment.

Average error	Standard deviation	Threshold	False detection rate/%	Missed detection rate/%	Time consumption
7.70	13.56	250	3.3	8.9	210 ms

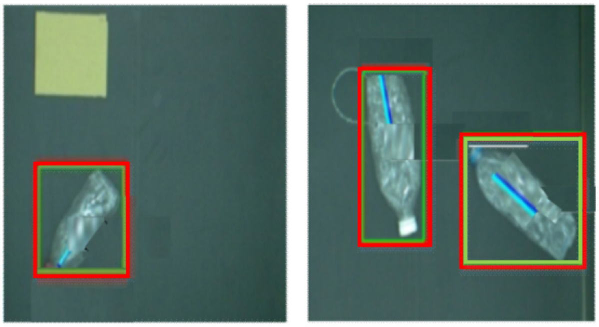


FIGURE 5 | Physical graph of the test in the laboratory environment.

The results in the laboratory environment show that the improved algorithm model of deep learning can accurately predict the position and angle of the bottle, which meets the needs of industrial robot target recognition.

Verification of VGG-16 Classification Network Model Algorithm Based on Hyper-Column Scheme Improvement

The above simulation verification is carried out in the experimental environment. To verify the target recognition effect of the improved VGG-16 classification network based on the Hyper-Column scheme, field verification is carried out through collecting 1,600 bottles in the real environment from a waste treatment plant in Xi'an City, Shaanxi Province. Similar to the above method, a 9:1 training test ratio is used, in which the total target number of the bottle is 160. The parameter settings for the training phase are shown in **Table 3**.

In the experiment, a total of 1,600 images are tested, in which the total number of bottles is 204 (some images have no bottles, and the crossover threshold is 0.5). The evaluation criteria are the false detection rate and the missed detection rate (as defined in section Verification of Fast R-CNN Algorithm), and the test results are shown in **Figure 6** and **Table 4**. It can be seen that the actual missed detection rate and false detection rate are high due to the poor image pixels captured in the real environment. As shown in **Figure 6**, in the decoupling model, 100 are detected for angles within 5 degrees, 42 for angles between 5 and 10 degrees, and 33 for angles >10 degrees. In the coupled model, 98 angles are detected to be within 5 degrees, 45 are detected between 5 and 10 degrees, and 30 are detected above 10 degrees. In the meantime, the results of coupling and decoupling models are compared, and the results are summarized in **Table 4**. The three loss decoupling models can provide lower false detection rate and missed detection rate, thereby obtaining more accurate detection results.

TABLE 3 | Parameter settings used in training in the real environment.

Item	Results	Item	Results
Training set	1440 images (image resolution 1600 × 1200)	The test set	160 images (image resolution 1600 × 1200)
Learning rate	0.00114	Momentum	0.9
Number of anchor nodes	901	Weight-decay	0.0005
Dimension of anchor nodes	[600, 1000]	Iterations	Fine-tune: 40,000 times
Number of images selected	VGG16 classification network: 2	NMS parameter	0.7
Number of Minni-batch in RPN	64	Detection threshold	0.99
The ratio of the prospect to the backdrop in RPN	[0,0.3]	Flipped image	No
Intersection-over-Union range of RPN prospect samples	No	Number of anchor nodes	901
Weight of criteria function	Position weight: 1; angle weight: 5	Training the updated layer	The 1st time: update the convolution layer and the FC layer; the 2nd time: update the FC layer. Shared convolution strategy applied

Figure 7 shows the test results in the real environment. It can be seen that even in the case of poor image quality (even difficult to distinguish by the naked eye), the algorithm can still obtain better detection results.

The test results of the real waste treatment plant images show that the improved Hyper-Column-based VGG-16 model can better detect the position and angle of the bottle, which meets the task requirements.

Comparison of the Improved VGG-16 Classification Network Model Based on the Hyper-Column Scheme and Other Advanced CNNs

The VGG-16 classification network model improved based on the Hyper-Column scheme is verified on the Matlab network simulation platform. The NYU Depth V2 database is introduced (Chen et al., 2019) with the leave-one-out evaluation method adopted; that is, the dataset with a sample space of 1100 is divided into subsets of 1000 and 100, with the subset of 1000 used as the training set and subset of 100 as the test set. The constructed system model is compared with advanced CNNs (AlexNet, GoogleNet, LeNet, ZF-Net, and ResNet) (Wang et al., 2017; Fadlullah et al., 2018; Luo et al., 2019; Hosny et al., 2020; Wang and Jia, 2020). The following equation shows the accuracy.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \qquad (5)$$

In Equation (5), *TP* represents a positive sample with positive prediction, *FP* represents a negative sample with positive prediction, *FN* represents a positive sample with negative prediction, and *TN* represents a negative sample with negative prediction.

Figure 8 shows the accuracy comparison between the improved algorithm and other advanced neural networks.

Figure 8 shows the accuracy comparison of the improved algorithm with other advanced neural networks. The result shows that the improved algorithm's accuracy reaches 82.34%, at least 3% higher than other advanced CNN algorithms such as AlexNet, GoogleNet, LeNet, ZF-Net, and ResNet.

DISCUSSION

Comparison of the Results of Fast R-CNN Algorithm and Improved Network Model Based on Hyper-Column Scheme

The detection effects of the Fast R-CNN algorithm on the detection of bottle-like objects in the laboratory environment and the detection effects of the VGG-16 algorithm based on the

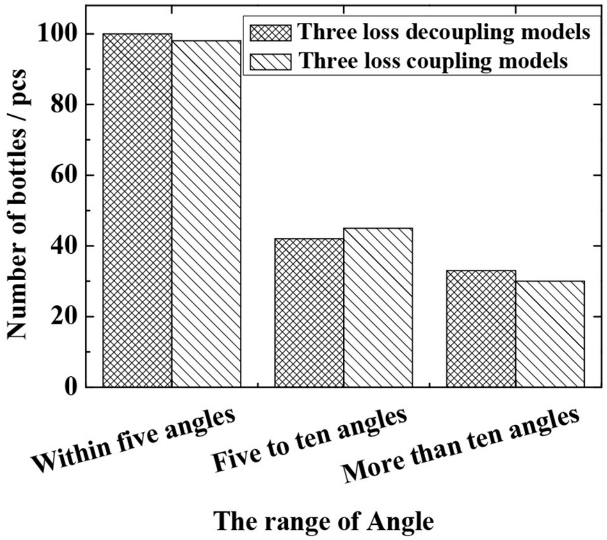


FIGURE 6 | The angle detection result of the improved Fast R-CNN model in the real environment.

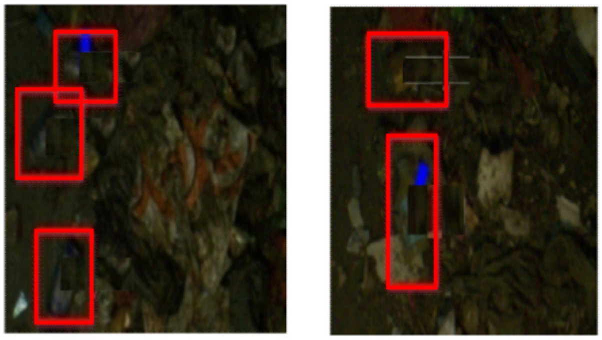


FIGURE 7 | Physical graph of the test in the real environment.

TABLE 4 | The detection result of the improved Fast R-CNN model in the real environment.

Model	Average error/degree	Standard deviation/degree	Numbers within the threshold	False detection rate/%	Missed detection rate/%	Time consumption
3 loss coupling model	7.71	13.53	176	7.4	21.3	230
3 loss decoupling model	6.95	11.41	174	5.8	19.4	260

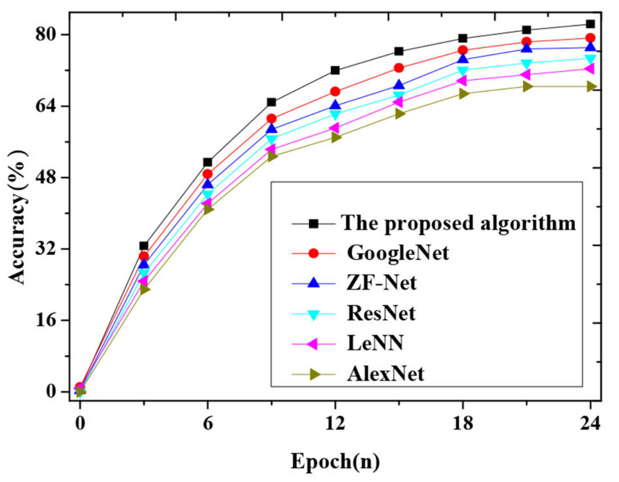


FIGURE 8 | Accuracy curves compared with advanced CNNs.

Hyper-Column scheme on detecting complex bottled objects in a realistic and complex environment are verified. The experimental results show that the Fast R-CNN algorithm can accurately detect bottled objects under the experimental environment, and the distribution of detection rate and the error rate is <10%. The improved VGG-16 algorithm based on the Hyper-Column scheme can accurately detect bottled objects in a complex real-world environment with a missed detection rate of <10% and an error rate of <30%. Both algorithms have better positioned and identified the target objects. In comparison, the Fast R-CNN algorithm in the experimental environment can complete the detection of the target item in 0.2 s, while the VGG-16 algorithm based on the Hyper-Column scheme can complete the detection in 0.24 s in the 3 loss coupling models, and the 3 decoupling models are completed in 0.28 s. Although the false detection rate and the missed detection rate of the 3 loss decoupling models are small, they are slightly inferior to the 3 loss coupling models in terms of detection consumption time. The constructed algorithm's accuracy is compared with that of other advanced neural networks. The analysis shows that the constructed algorithm's accuracy reaches 82.34%, at least 3% higher than other advanced CNN algorithms such as AlexNet, GoogleNet, LeNet, ZF-Net, and ResNet. The reason may be that the original DL algorithm's feature fusion and the training speed increase improve industrial robot target recognition accuracy.

Results Discussions

The applications of the Fast R-CNN algorithm and the improved Hyper-Column-based VGG-16 classification network in industrial robot target recognition and classification are explored. The experimental results show that deep learning plays an important role in the visual recognition of industrial robots. Compared with the previous research, the improvement of the algorithm makes the industrial robots convolve and pool in the visual recognition, leading to the results that the average false detection rate is <5.5% and the average missed detection rate is 17%. The test results have shown the high precision and high accuracy of the algorithm. The deep learning algorithm is suitable for industrial applications. Meanwhile, it also shows that artificial regularization can improve the learning and recognition of the algorithm. It can be seen from the experimental results that the algorithm model proposed in the experiment can meet the requirements of modern industry for industrial robot vision works.

DISCUSSION

Artificial intelligence has superior feature extraction performance and great development potential in the industry 4.0 era. Therefore, to solve these problems, the image layers are convolved and pooled through the deep learning model of artificial intelligence, and the visual recognition system of industrial robots is optimized through the advanced methods of the target classification algorithm. In the aspect of visual recognition, the recognition algorithm of target objects in a complex background environment is studied. The results have shown that both the Fast-RCN algorithm and the improved VGG-16 classification network based on the Hyper-Column scheme can complete the localization and recognition of the target objects, indicating that the Fast-RCN algorithm and the Hyper-Column-based scheme have improved the accuracy and effectiveness of target recognition and positioning of VGG-16 classification network.

At present, the research on industrial robots in artificial intelligence has yet to be improved in China. The application of deep learning in artificial intelligence to the visual system of industrial robots has achieved preliminary results through the experiment. If the relevant algorithms of artificial intelligence can be applied in the research, development, and production process of industrial robots in future research, it will certainly promote the industrial robot technology to develop more rapidly. There are many algorithms in the deep learning theory of artificial intelligence, which can be used to solve a series of problems that occurred in the development of industrial robots. With the rapid development of communication technology, the 5G era has arrived. The new generation of wireless communication technology not only means faster transmission speed but also better security performance for industrial robots. The application of these intelligent algorithms to the development of industrial robots with higher service levels will bring new opportunities for the development of industrial robots.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

ACKNOWLEDGMENTS

This work was supported by the enterprise cooperation project of domestic visiting engineers in Colleges and universities of Department of Education of Zhejiang Province in 2020 (No. FG2020074, Research on the Technology of Intelligent Recognition of Synthetic Forgery Videos) and Wenzhou basic scientific research project of Wenzhou science and Technology Bureau in 2020 (No. G2020033).

REFERENCES

Bogue, R. (2017). Domestic robots: has their time finally come? *Ind. Rob.* 44, 129–136. doi: 10.1108/IR-01-2017-0018

Capparelli, F., Pawelzik, K., and Ernst, U. (2019). Constrained inference in sparse coding reproduces contextual effects and predicts laminar neural dynamics. *PLoS Comput. Biol.* 15:e1007370. doi: 10.1371/journal.pcbi.1007370

Chen, S., Tang, M., and Kan, J. (2019). Predicting depth from single rgb images with pyramidal three-streamed networks. *Sensors* 19:667. doi: 10.3390/s19030667

Dönmez, E., Kocamaz, A. F., and Dirik, M. (2017). A vision-based real-time mobile robot controller design based on gaussian function for indoor environment. *Arab. J. Sci. Eng.* 2017, 1–16. doi: 10.1007/s13369-017-2917-0

Ertuğrul, Ö. F., and Tağluk, M. E. (2017). A novel machine learning method based on generalized behavioral learning theory. *Neural Comput. Appl.* 28, 3921–3939. doi: 10.1007/s00521-016-2314-8

Fadlullah, Z. M., Tang, F., Mao, B., Liu, J., and Kato, N. (2018). On intelligent traffic control for large-scale heterogeneous networks: a value matrix-based deep learning approach. *Commun. Lett. IEEE* 22, 2479–2482. doi: 10.1109/LCOMM.2018.2875431

Fernandes, C. M., Mora, A. M., Merelo, J. J., and Rosa, A. C. (2017). Kants: a stigmergic ant algorithm for cluster analysis and swarm art. *IEEE Trans. Cybern.* 44, 843–856. doi: 10.1109/TCYB.2013.2273495

Hinton, M., He, Z., Xiao, L., et al. (2017). An automatic assembling system for sealing rings based on machine vision. *J. Sens.* 2017, 1–12. doi: 10.1155/2017/4207432

Hosny, K. M., Kassem, M. A., and Foaud, M. M. (2020). Classification of skin lesions into seven classes using transfer learning with alexnet. *J. Digit. Imaging* 33, 1325–1334. doi: 10.1007/s10278-020-00371-9

Huang, H., Song, Y., Yang, J., Gui, G., and Adachi, F. (2019). Deep-learning-based millimeter-wave massive MIMO for hybrid precoding. *IEEE Trans. Veh. Technol.* 68, 3027–3032. doi: 10.1109/TVT.2019.2893928

Li, J., and Yang, F.,. (2018). Research on multi-robot scheduling algorithms based on machine vision. *EURASIP J. Image Video Process.* 2018, 1–11. doi: 10.1186/s13640-018-0355-x

Li, S., Li, D., Zhang, C., Wan, J., and Xie, M. (2020). RGB-D image processing algorithm for target recognition and pose estimation of visual servo system. *Sensors* 20:430. doi: 10.3390/s20020430

Luo, P., Tian, L. P., Ruan, J., and Wu, F. (2019). Disease gene prediction by integrating ppi networks, clinical rna-seq data, and omim data. *IEEE/ACM Trans. Comput. Biol. Bioinfor.* 16, 222–232. doi: 10.1109/TCBB.2017.2770120

Matson, T., Farfel, M., Levin, N., Holm, E., and Wang, C. (2019). Machine learning and computer vision for the classification of carbon nanotube and nanofiber structures from transmission electron microscopy data. *Microsc. Microanal.* 25, 198–199. doi: 10.1017/S1431927619001727

Sampedro, C., Rodriguez-Ramos, A., Bavle, H., Carrio, A., de la Puente, P., and Campoy, P. (2019). A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques. *J. Intell. Robot. Syst.* 95, 601–627. doi: 10.1007/s10846-018-0898-1

Song, Y., Liu, Y., Liu, L., Zhu, D., Jiao, J., and Chen, L. (2017). Extraction method of navigation baseline of corn roots based on machine vision. *Trans. Chin. Soc. Agric. Mach.* 48, 38–44. doi: 10.6041/j.issn.1000-1298.2017.02.005

Stani, I., Musi, J., and Gruji, T. (2017). Gesture recognition system for real-time mobile robot control based on inertial sensors and motion strings. *Eng. Appl. Artif. Intell.* 66, 33–48. doi: 10.1016/j.engappai.2017.08.013

Sun, J., Wang, J., Guo, L., Yang, J., and Gui, G. (2020). Adaptive deep learning aided digital predistorter considering dynamic envelope. *IEEE Trans. Veh. Technol.* 69, 4487–4491. doi: 10.1109/TVT.2020.2974506

Sung, M., Cho, S. H., Kim, J., Lee, J. K., Lee, J. H., and Chung, H. S. (2017). Demonstration of IFoF-based mobile fronthaul in 5G prototype with 28-GHz millimeter wave. *J. Light. Technol.* 36, 601–609. doi: 10.1109/JLT.2017.2763156

Toğaçar, M., Cömert, Z., and Ergen, B. (2020). Classification of brain MRI using hyper column technique with convolutional neural network and feature selection method. *Expert Syst. Appl.* 149:113274. doi: 10.1016/j.eswa.2020.113274

Wang, D., Zhang, M., Li, Z., Li, J., Fu, M., Cui, Y., et al. (2017). Modulation format recognition and OSNR estimation using CNN-based deep learning. *IEEE Photon. Technol. Lett.* 29, 1667–1670. doi: 10.1109/LPT.2017.2742553

Wang, R., and Jia, J. (2020). Design of small humanoid fighting robot based on target recognition algorithm. *Int. J. Adv. Robot. Syst.* 17:1729881420910661. doi: 10.1177/1729881420910661

Wang, X., Lile, H. E., and Zhao, T. (2018). Mobile robot for SLAM research based on lidar and binocular vision fusion. *Chin. J. Sens. Actuators* 31, 394–399. doi: 10.3969/j.issn.1004-1699.2018.03.013

Wang, X., Liu, A., Zhang, Y., and Xue, F. (2019). Underwater acoustic target recognition: a combination of multi-dimensional fusion features and modified deep neural network. *Remote Sens.* 11:1888. doi: 10.3390/rs11161888

Wang, Y., Yang, J., Liu, M., and Gui, G. (2020). LightAMC: lightweight automatic modulation classification via deep learning and compressive sensing. *IEEE Trans. Veh. Technol.* 69, 3491–3495. doi: 10.1109/TVT.2020.2971001

Wen, X. (2020). Using deep learning approach and IoT architecture to build the intelligent music recommendation system. *Soft Comput.* doi: 10.1007/s00500-020-05364-y. [Epub ahead of print].

You, B., Wu, K., Xu, J., Chen, G., Liang, Q., and Tian, J. (2017). Design and research of automatic plug-in system based on machine vision. *Opto Electron. Eng.* 44, 919–926. doi: 10.3969/j.issn.1003-501X.2017.09.009

Zhang, Q., Chen, S., Yu, T., and Wang, Y. (2017). "Cherry recognition in natural environment based on the vision of picking robot," in *IOP Conference Series: Earth and Environmental Science, Vol. 61* (Bristol: IOP Publishing), p. 012021.

Conflict of Interest: LLi was employed by Huawei Technologies Co. Ltd.

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

BRIEF RESEARCH REPORT
published: 12 March 2021
doi: 10.3389/fnbot.2021.621196

Intelligent Badminton Training Robot in Athlete Injury Prevention Under Machine Learning

Jun Xie[1], *Guohua Chen*[1]* *and Shuang Liu*[2]

[1] *School of Physical Education, East China University of Technology, Nanchang, China,* [2] *College of Physical Education, Jinggangshan University, Ji'an, China*

This study was developed to explore the role of the intelligent badminton training robot (IBTR) to prevent badminton player injuries based on the machine learning algorithm. An IBTR is designed from the perspectives of hardware and software systems, and the movements of the athletes are recognized and analyzed with the hidden Markov model (HMM) under the machine learning. After the design was completed, it was simulated with the computer to analyze its performance. The results show that after the HMM is optimized, the recognition accuracy or data pre-processing algorithm, based on the sliding window segmentation at the moment of hitting reaches 96.03%, and the recognition rate of the improved HMM to the robot can be 94.5%, showing a good recognition effect on the training set samples. In addition, the accuracy rate is basically stable when the total size of the training data is 120 sets, after the accuracy of the robot is analyzed through different data set sizes. Therefore, it was found that the designed IBTR has a high recognition rate and stable accuracy, which can provide experimental references for injury prevention in athlete training.

Keywords: intelligent badminton training robot, machine learning, hidden markov model, athlete injury, motion recognition

OPEN ACCESS

Edited by:
Mu-Yen Chen,
National Taichung University of Science and Technology, Taiwan

Reviewed by:
Junwei Wang,
Zhejiang University, China
Yuka Kimura,
Hirosaki University Graduate School of Medicine, Japan
Hsin-Te Wu,
National Ilan University, Taiwan

Correspondence:
Guohua Chen
cghbw@163.com

Received: *25 October 2020*
Accepted: *08 February 2021*
Published: *12 March 2021*

Citation:
Xie J, Chen G and Liu S (2021) Intelligent Badminton Training Robot in Athlete Injury Prevention Under Machine Learning. Front. Neurorobot. 15:621196. doi: 10.3389/fnbot.2021.621196

INTRODUCTION

Today, people's living standards have significantly improved with the rapid development of science and technology. As one of human beings' greatest inventions in the era of artificial intelligence (AI), after years of development, robots already possess a considerable amount of AI. The robot combines high-tech technologies with multiple other technologies, including mechanics, informatics, mechanics, AI, electronics, biology, and control system engineering. Robotics has experienced several stages maturity in its research and development. At this stage, its functions are becoming more mature, and its applicational prospects are also extremely broad (Mizuno et al., 2019; Cao et al., 2020). AI has also begun to be applied in sports. There are some data analyses and intelligent sports products in the popular badminton sport, which initially demonstrate the appeal of applying AI technology to traditional sports (Mutaqin et al., 2020). At present however, the application of AI in badminton sports is limited in both depth and breadth. Most badminton training methods are still realized with the one-to-one or one-to-many manual teaching methods, which can be extremely restrictive to the improvement of a players' own level (Chia et al., 2019; Gao et al., 2020). Therefore, applying machine learning and other related algorithms to the training of badminton players has become the focus of scientific research in recent years.

Badminton is a net-to-net confrontation sport, and the athletes on both sides compete in skill and tactics without physical contact. Therefore, the possibility of injury due to collision is very small in badminton. However, athletes are required to play in a short time, combining the consistency of running and jumping, and the upper limb variability of lower limbs to complete the game (Huang et al., 2019; Wang et al., 2019; Mansec et al., 2020). Therefore, improper operation during training may cause physical injury to the athlete. Preventing physical injury of the athletes is therefore particularly important. Machine learning is an important algorithm in AI. Many scientific researchers have adopted machine learning to study the action and behavior recognition of the human body. For example, standing, walking, running, and lying down in daily actions can be recognized and classified by algorithms (Li and Zhang, 2017; Polydoros and Nalpantidis, 2017; Wang et al., 2018). Similarly, machine learning can also detect abnormalities and falling in the human body, mainly by collecting human electromyographic signals, acceleration signals, and video signals. For badminton, table tennis, tennis, football, and other complex activities, the wearable acceleration sensor can be adopted to collect the data, and then the machine learning and other related algorithms are adopted for recognition and classification (Li et al., 2019b). Machine learning and AI technology therefore have broad applicational prospects in the field of sports.

In summary, the recognition and regulation of actions during sports is of extreme significance, so as to reduce the damage caused to badminton players during training and to further improve their abilities. The innovation of this study lies in the design of the IBTR system, and the use of HMM in the machine learning algorithm to recognize movements of athletes. Finally, computer simulation is adopted to test and analyze its performance in multiple directions, which provides experimental evidence for the future development of sports and injury prevention in athletes during training.

LITERATURE REVIEW

Research Trends in the Application of Machine Learning

As the core content, machine learning has been applied to various branches of AI, and there are many studies based on it. In wireless networks, Jiang et al. (2017) adopted machine learning techniques to propose applying them in attractive applications in 5G networks, including cognitive radio, massive multi input multi output (MIMO), femto /Small cells, heterogeneous networks, smart grids, energy harvesting, and communication between devices, to meet the diverse needs of wireless networks, which provided experimental support for extremely high speeds and new applications of future wireless networks. In the medical field, Walsh et al. (2018) used machine learning to predict the suicidal behavior of adolescents through the collection of clinical data, and screened the risks of non-fatal autotomy attempts of the adolescents through a retrospective analysis of historical data. In electric power, Dalal et al. (2019) proposed a data-driven distributed opportunity-constrained optimization equation and built an agent that predicts the results of the operation process for the power system using machine learning, so as to solve the traction in large-scale power grids; the simulation results revealed that the proposed solution is cheaper and more reliable than other candidate solutions. In the field of physics, Namba et al. (2020) took the three-dimensional alignment control of asymmetric top molecule sulfur dioxide (SO_2) as an example to study a machine learning method for drawing landscape maps in a low-dimensional control parameter space, and studied the control of a set of mutually orthogonal linearly polarized laser pulses; these pulses were parameterized by time delay and fluence ratio; the parameters could be represented by points in the parameter space or time and frequency resolved spectra; and the simulation analysis disclosed that the machine learning model based on convolutional neural network (CNN) is trained using a small number of training samples, and can construct mapping with sufficient accuracy to predict temperature-related control mechanisms.

Based on the analysis of scholars in the above-mentioned related fields, machine learning is an important algorithm in AI. Although its fields of application involve wireless communications, medical and health, power grids and physics, the training and movement analysis in sports basically rely on the traditional coaching experience and training and skill improvement. Therefore, it is of great significance to apply AI technologies, such as machine learning, to sports such as badminton like we have done in this study.

Research Status of Badminton Robot

As a product of multiple disciplines, the badminton robot involves a variety of theoretical knowledge. Generally speaking, the robot is large in size, and requires engineers from multiple disciplines to collaborate and discuss to complete all the work. The badminton robot developed by Sapiee and Annuar (2018) is composed of a robotic arm fixed on the rails of the court and an off-court processor. During training, the shuttlecocks flight trajectory is read through the two off-court cameras; the data is processed in the off-court desktop computer and is then transmitted back to the controller to control the motor-driven manipulator to hit the shuttlecock, which can realize the function of playing against an opponent (Sapiee and Annuar, 2018). Mizuno et al. (2019) developed a fully automatic badminton game robot, which could receive and serve in the same environment as the human athlete; the Kinect sensor and the space shuttle in this system were applied to predict and detect the motion trajectory of the shuttlecock. Liu et al. (2020) collected depth images of the scene using a depth camera and applied the machine vision theory to process the obtained depth images. The position of the badminton camera coordinate system in the three-dimensional space was obtained by combining the image depth information. The position of the field coordinate system was thus realized. Finally, the position information of the shuttlecock in the multi-frame image was adopted to predict the drop point of the shuttlecock; then, the badminton robot quickly ran to the predetermined position to complete the hitting task (Liu et al., 2020).

In summary, the above-mentioned research shows that related machine learning has been applied to many fields, and the development and performance of badminton robots are constantly improving. However, there are relatively few related studies about applying the machine learning to predict the shuttlecock trajectory. Therefore, the machine learning is adopted to recognize the movement of badminton players and to predict the badminton trajectory to realize the hitting task and sparring effect with the players, which has important value for the application of AI in sports.

METHOD

Design Idea of Intelligent Badminton Training Robot

In the era of intelligence, the development and promotion of intelligent robots has become an important direction for the transformation and upgrade of the domestic sports industry. For example, various ball pitching machines, ball pickers, and human-robot sports robots have gradually entered professional sports and ordinary sports practices. Badminton is a national sport that is suitable for all ages and can involve several people synchronously (Sato et al., 2017). With the rapid development of science and technology, automatic machines used for training, fitness, and entertainment (such as badminton ball machines and badminton companion robots) have been successfully developed and have been applied to a certain extent. However, there are not many types of badminton pitching machines, and the pitching system is not perfect. It is not only because the service life of the badminton robot is reduced due to the serious damage caused by the pitching equipment, but also because its intelligence is low.

Therefore, a new type of badminton training robot was researched, developed, and designed from both hardware and software aspects, and was applied in this study in the training of badminton players, enhancing the function and performance of the pitching system.

Hardware Design of the Intelligent Badminton Training Robot

In the designed badminton robot, the hardware mainly includes two aspects: the swing movement of the badminton racket and the movement of the mobile chassis for the robot. During the movement of the badminton racket, the module fixed on the bottom of the racket collects the three-axis acceleration and the three-axis angular acceleration in real time and receives the original data through the integrated digital motion processor (DMP) (Lau et al., 2018) of the module for next posture fusion. Then, the calculated motion posture data is transmitted to the main control chip through mobile communication, and then the main control chip recognizes the motion mode through the machine learning algorithm. Finally, the final result is sent to the upper computer through the Bluetooth module for the user to observe. The hardware framework of the IBTR is shown in **Figure 1**.

The mobile chassis model of badminton requires the speed of each wheel axis and the roller speed to calculate the true speed of the wheel (Matsuo, 2017; Singh and Singh, 2018). First, a plane coordinate system is established to decompose the movement of the chassis into three independent components: translation along the X axis, translation along the Y axis, and rotation along the yaw axis. v_{t_x}, from left to right, refers to the movement speed of the chassis on the X axis, and v_{t_y} from the bottom to the top, refers to the movement speed of the chassis on the Y axis. $\vec{\omega}$ takes clockwise rotation as the positive direction, and refers to the rotational angular velocity of the yaw axis. The speed expression can be written as:

$$\vec{v} = \vec{v}_t + \vec{\omega} \times \vec{r} \tag{1}$$

In the above equation, $\vec{v}$ represents the vector where the center of the chassis points to one of the wheel axis; $\vec{r}$ represents the motion speed vector of the wheel axis; and $\vec{v}_t$ refers to the speed vector at time t. The following equation can be obtained by decomposing $\vec{v}$ along the X and Y axes:

$$\begin{cases} \vec{v}_x = \vec{v}_{t_x} + \omega \times r_y \\ \vec{v}_y = \vec{v}_{t_y} + \omega \times r_x \end{cases} \tag{2}$$

According to the above-mentioned wheel axis speed, the speed $\vec{v}_\perp$ perpendicular to the direction of the roller and the speed $\vec{v}_{||}$ of the roller movement direction can be decomposed. Since $\vec{v}_\perp$ is provided by the friction between the ground and the roller during the movement, its value can be adapted to the needs as long as the maximum friction is large enough. The value of $\vec{v}_{||}$ can be derived according to the following equation:

$$\vec{v}_{||} = \vec{v} \cdot \widehat{u} = -\frac{1}{\sqrt{2}}v_x + \frac{1}{\sqrt{2}}v_y \tag{3}$$

Where, $\widehat{u}$ is the unit vector along the direction of the roller, so the speed of the wheel can be obtained as below:

$$v_\omega = \frac{\vec{v}_{||}}{\cos 45^\circ} = \sqrt{2}(-\frac{1}{\sqrt{2}}v_x + \frac{1}{\sqrt{2}}v_y) = -v_x + v_y \tag{4}$$

In this study, a = 320 mm and b = 290 mm can be obtained based on the installation position of the wheels on the chassis of the vehicle. Based on the above-mentioned wheel axis speed and roller speed, the rotating speeds of four wheels can be expressed as Equation (5) below:

$$\begin{cases} v_{\omega_1} = v_{t_y} - v_{t_x} + \omega(a+b) \\ v_{\omega_2} = v_{t_y} + v_{t_x}\text{-}\omega(a+b) \\ v_{\omega_3} = v_{t_y} - v_{t_x} + \omega(a\text{-}b) \\ v_{\omega_4} = v_{t_y} + v_{t_x}\text{-}\omega(a\text{-}b) \end{cases} \tag{5}$$

In the cylindrical coordinate system, the coordinate (x, y, z) is to represent the point P, P_z refers to the position of the point P on the Z axis, and d represents the projection length of the vector P on the plane XOY. The angle between X axis and the vector P on the XOY axis is expressed with α (Victor et al., 2018). Then, the

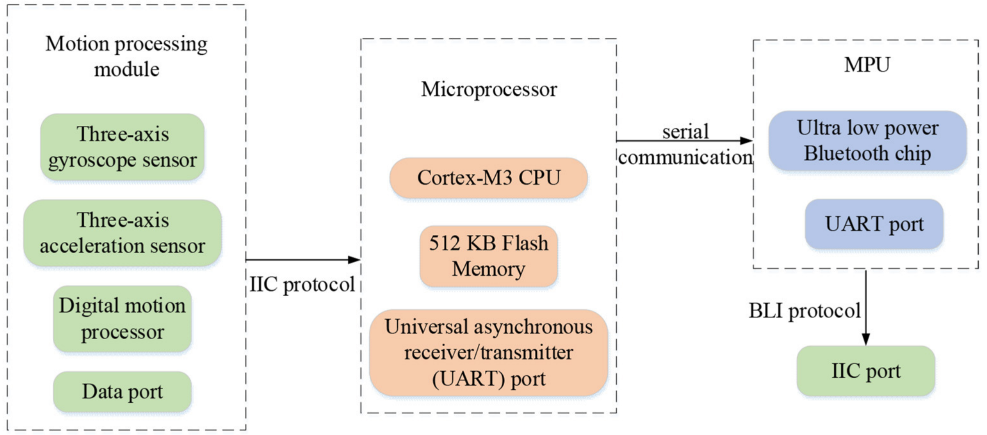


FIGURE 1 | The hardware framework of the intelligent badminton training robot.

cylindrical coordinates of point P can be expressed with z, α, d, as follows:

$$\begin{cases} 0 \leq d \leq +\infty \\ 0 \leq \alpha \leq 2\pi \\ -\infty < z < +\infty \end{cases} \tag{6}$$

The position can be represented by the cylindrical coordinate, which can also be regarded as the x-axis translation for a certain length, rotation around the z-axis, and translation along the z-axis on the Cartesian coordinate system (Chen et al., 2019). In order to facilitate the association of the positions in the Cartesian coordinate system and the cylindrical coordinate system, point P can be expressed as the following equation:

$$Cyl(p_z, \alpha, d) = Trans(0, 0, p_z)Rot(A, \alpha)Trans(d, 0, 0) \tag{7}$$

In the above Equation (7), $Cyl(p_z, \alpha, d)$ refers to the position in cylindrical coordinates, and $Trans(0, 0, p_z)$ refers to its translation transformation. Of which, the translation along the X, Y, and Z axes can be represented by P_x, P_y, and P_z, respectively, and the rotation transformation is represented by $Rot(A, \alpha)$. The rotation axis is the A axis, and α refers to the rotation angle.

When the robot receives and sends instructions for routine operations, the robot's posture changes according to the instant instructions due to the different description instructions for the robot. This is related to the steel body of the robot (the object with fixed dimensions, fixed size, and a fixed form, and its basic properties and volume are not easy to change under the action of force) (Zagatto et al., 2018). By connecting many steel bodies, the movement posture and position of the object can be described using the Cartesian coordinate system, so as to complete the corresponding command actions, which mainly include translational coordinate transformation, rotation coordinate transformation, composite coordinate transformation, and homogeneous coordinate transformation (Kawata et al., 2019).

Software Design of the Intelligent Badminton Training Robot

In the designed badminton robot system, its software system mainly includes four parts: the machine supervised learning system, the posture calculation system, the Bluetooth communication system, and the Android user terminal software. The software design of the control system is mainly developed around the machine learning system. The control chip receives the three-axis posture angle and three-axis acceleration transmitted by the movement processing module through the Inter-Integrated Circuit (IIC) communication protocol and processes the sensor data through machine learning algorithms to analyze the physical fitness and swing movements of the player. The flowchart of the software control system is shown in **Figure 2**.

Data Collection and Pre-processing

After the controller of the intelligent robot is started, the Microprocessor Unit (MPU) data processing module is first initialized, and then the current three-axis acceleration, three-axis angular acceleration, and geomagnetic declination are measured by its own inertial elements and Hall sensors. The internal integrated digital motion processing is adopted for posture fusion, and the three-axis posture angle is calculated. Finally, the obtained data is transmitted to the main control chip through the serial port, thereby completing the data collection of the intelligent feather robot. After the data collection is completed, the signal denoising algorithm (Li et al., 2019a) is adopted to denoise the original data and the data information of a single shot action is extracted with the window segmentation technology (Liang et al., 2019). Next, the single hitting motion is processed by time series and framing to obtain multiple meta-motions. The accuracy of k-means clustering (Mao et al., 2019) is evaluated on multiple waveform features (such as peak values and average values) of each meta-action. The feature with the most obvious clustering characteristics is selected as the characteristic information of each meta-motion. The meta-action is assigned to

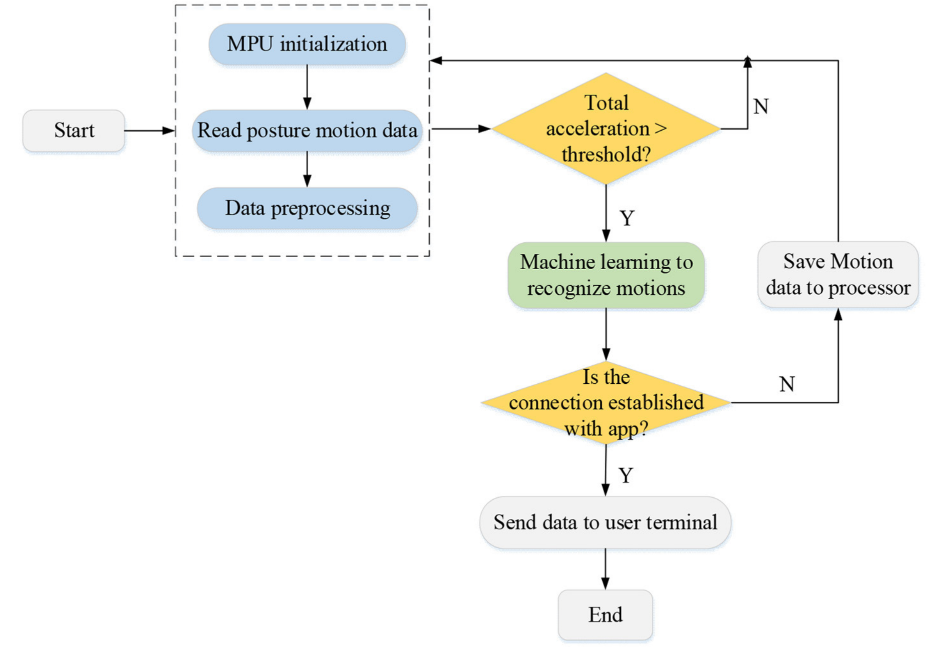

MPU initialization
Start
Read posture motion data
Data preprocessing
Total acceleration > threshold?
N
Y
Machine learning to recognize motions
Save Motion data to processor
Is the connection established with app?
N
Y
Send data to user terminal
End

FIGURE 2 | The flowchart of the software control system of the intelligent badminton training robot.

the corresponding codebook space with vector quantization, and the observation sequence of the hitting motion can be obtained as the input of the subsequent algorithm model. The data pre-processing process is shown in **Figure 3**.

Motion Recognition Based on Machine Learning

Due to the time sequence of the shuttlecock motion, a probability model about time sequence (named as HMM) is adopted to model the hitting motion. The recognition process of the racket motion is to extract the observation sequence of the current hitting motion through the pre-processing algorithm as the input of each established hitting motion HMM. In addition, the Viterbi algorithm (Vu et al., 2018) is utilized to obtain the appearance probability of the best state sequence for the current hitting motion under each model. The hitting motion of the model with the largest probability output result is deemed as the recognition result of the current observation sequence.

HMM modeling mainly relies on three parameters: initial probability, transition probability, and observation probability. A set of applicable HMM is established for each hitting motion, and each hitting action is divided into n meta-motions with sequence. The codebook corresponding to each meta-motion is determined by vector quantization, which is considered to be the observation set defined for HMM. Therefore, each hitting motion can be regarded as a set of observation sequences with the length of n. For this observation sequence, the parameters of each model are trained through the Baum-Welch algorithm (Jalal and Kim, 2020) until the parameters can meet the convergence requirements.

The Baum-Welch algorithm is an unsupervised model learning and training algorithm. It is assumed that a given training data value contains S observation sequences of length T $\{O_1, O_2, ..., O_S\}$ without a corresponding state sequence, then the parameters of the model can be learned with the forward algorithm and the backward algorithm. The observation sequence data is regarded as observation data O, and the state sequence data is regarded as unobservable hidden data I, so the following equation can be obtained:

$$P(O|\lambda) = \sum_{I} P(O|I,\lambda)P(I|\lambda) \tag{8}$$

The parameter learning can be realized by the expectation-maximization algorithm (EM algorithm) (Polydoros and Nalpantidis, 2017), in which, E finds the Q function $Q\left(\lambda, \bar{\lambda}\right)$, as follows:

$$Q\left(\lambda, \bar{\lambda}\right) = \sum_{I} \log P(O,I|\lambda)P(O,I|\bar{\lambda}) \tag{9}$$

In the Equation (9), $\bar{\lambda}$ refers to the current estimated value of the model parameter, and λ refers to the maximized model parameter. The M in the EM algorithm is to obtain the model parameter $\lambda = (A, B, \pi)$ by maximizing the Q function $Q\left(\lambda, \bar{\lambda}\right)$. First, the parameter π_i is calculated using Lagrangian multiplication to find the partial derivative of it, then the following equation can be obtained:

$$\pi_i = \frac{P(O, i_1 = i|\bar{\lambda})}{P(O|\bar{\lambda})} \tag{10}$$

The parameter A can then be obtained by the Lagrangian multiplication of the constraint condition $\sum_{j=1}^{N} a_{ij} = 1$. The a_{ij} can

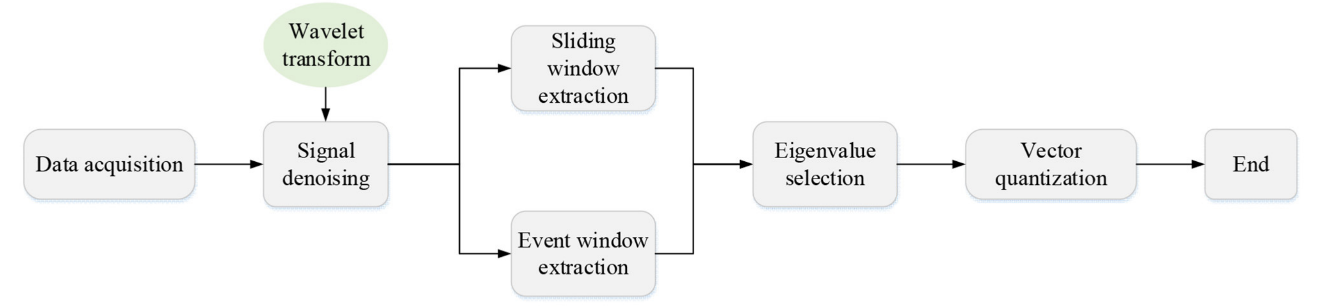


FIGURE 3 | The data pre-processing process.

be expressed as Equation (11):

$$a_{ij} = \frac{\sum\limits_{t=1}^{T-1} P(O, i_t = i, i_{t+1} = j|\bar{\lambda})}{\sum\limits_{t=1}^{T-1} P(O, i_t = i|\bar{\lambda})} \tag{11}$$

Finally, the parameter B is obtained. The constraint condition is set to $\sum\limits_{k=1}^{M} b_j(k) = 1$, and the partial derivative of $b_j(o_t)$ to $b_j(k)$ is not 0 when $o_t = v_k$ is satisfied, which is represented by $I(o_t = v_k)$. The following equation can be obtained:

$$b_j(k) = \frac{\sum\limits_{t=1}^{T} P(O, i_t = j|\bar{\lambda}) I(o_t = v_k)}{\sum\limits_{t=1}^{T} P(O, i_t = j|\bar{\lambda})} \tag{12}$$

However, there are multiple data sets for the same hitting motion in the sample in the designed IBTR, but the training HMM parameter model is based on a single sample, so the model data may fall into the local optimum, and the recognition rate of other samples is low. Therefore, two schemes are proposed for multi-sample training. One scheme refers to the mean training method. The input of the model (the observation sequence of the sample) is treated to realize the data layer fusion, and the model training is realized after the average of multiple sets of training data is obtained under the same model. The other scheme refers to the frequency weighted training method. The frequency of the sample observation sequence in the model is adopted for linear weight, and the parameters in the Baum-Welch algorithm are corrected. It is supposed that the sequence of M observation values $O^{(m)}$ (*m = 1, 2, … m*), and P_m is the frequency of the *mth* observation sequence, then the correction equations are given as follows:

$$\bar{\pi}_i = \sum_{m}^{M} \frac{\alpha_1^{(m)}(i)\beta_1^{(m)}(i)}{P(O^{(m)}|\lambda)} \tag{13}$$

$$\bar{a}_{ij} = \frac{\sum\limits_{m}^{M} P_m \sum\limits_{t=1}^{T_m-1} \alpha_t^{(m)}(i) a_{ij} b_i \left(O_{t+1}^{(m)}\right) \beta_{t+1}^{(m)}(j)}{\sum\limits_{m}^{M} P_m \sum\limits_{t=1}^{T_m} \alpha_t^{(m)}(i)\beta_t^{(m)}(j)} \tag{14}$$

$$\bar{b}_j(k) = \frac{\sum\limits_{m}^{M} P_m \sum\limits_{t=1}^{T_m-1} \alpha_t^{(m)}(i)\beta_t^{(m)}(j) I(o_t = v_k)}{\sum\limits_{m}^{M} P_m \sum\limits_{t=1}^{T_m-1} \alpha_t^{(m)}(i)\beta_t^{(m)}(j)} \tag{15}$$

In the above equations, $\alpha_t^{(m)}(i)$ and $\beta_t^{(m)}(j)$ are the forward probability and backward probability of the *mth* sequence, respectively. $\bar{\pi}_i$, $\bar{a}_{ij}$, and $\bar{b}_j(k)$ are the revised initial, transition, and observation probability parameters, respectively.

Bluetooth Communication Connection

After the control system is started, its working mode is automatically set to allow surrounding devices to initiate connections. After the connection is successful, the control system transmits motion analysis data to the integrated memory through the serial port and sends it to the mobile phone application (APP) through the low-power Bluetooth protocol. If it fails to connect to the APP, the microprocessor will save the data in the integrated Flash memory (Steels et al., 2020) inside the processor. When the control system establishes a connection with the mobile phone APP, it will immediately send the unsent data to the client APP through the Bluetooth function. In addition, the control system will clear the data that has been uploaded to the client APP to avoid duplicate data recording.

Running Simulation of Badminton Robot

The performance of the IBTR was simulated and analyzed in this study. Multiple sets of controlled experiments were designed to test the motion recognition accuracy of the system under different training data sizes. Six representative basic motions were selected based on the rules and playing methods of badminton, including high and far ball, smash, flat driving, flat

TABLE 1 | Multiple-dimensional comparison of recognition rate of the intelligent badminton training robot.

Dimension	Method
Data pre-processing	Sliding window segmentation based on hitting time
	Sliding window segmentation based on the maximal value
	Window segmentation based on the event
Motion recognition	Motion recognition based on support vector machines (SVM)
	Action recognition based on traditional HMM
	Action recognition based on improved HMM
Recognition of training objects	Same athlete
	Different athletes

block, lifting, and chop. These motions were demonstrated by a professional badminton player and 30 records were collected as the training sample set of the system. In the simulation, the accuracy on the different training set sizes (N = 120 and 300 groups, respectively) and recognition rate on different dimensions of the system were mainly determined and analyzed comprehensively. Of which, the recognition rate of the IBTR was compared from three dimensions, as shown in **Table 1**.

RESULTS AND DISCUSSION

Comparison of Recognition Rates in Different Dimensions

The recognition rate of IBTR is compared and analyzed from the three dimensions: data pre-processing, motion recognition, and recognition of training objects.

Analysis on recognition rate of the IBTR in the data pre-processing dimension discloses that the sliding window segmentation based on the hitting motion has a higher recognition rate than the other two dimensions, and the highest recognition rate can reach 96.03% (as shown in **Figure 4**). The recognition rate in the motion recognition dimension is compared and analyzed based on the SVM, traditional HMM, and improved HMM. The results indicate that the traditional HMM training has the lowest recognition rate, followed by SVM, and improved HMM action has the highest recognition rate, which can be up to 94.50% (as given in **Figure 5**). In terms of the recognition of training objects, the recognition rate based on the same athlete is up to 97.98%, and the average recognition rate of the system based on different athletes is up to 96.711%. Thus, it is obvious that recognition rate is highest when the subject is the same athlete (as illustrated in **Figure 6** below).

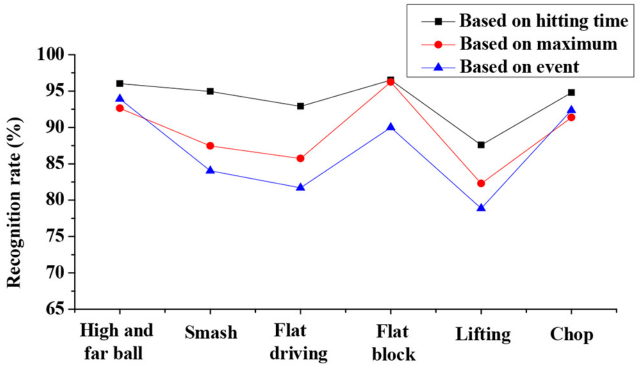


FIGURE 4 | Analysis on recognition rate of the intelligent badminton training robot in the data pre-processing dimension.

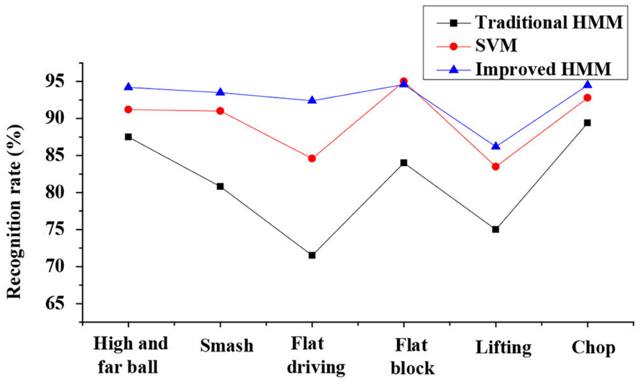


FIGURE 5 | Comparison on recognition rates of the intelligent badminton training robot with the three training algorithms.

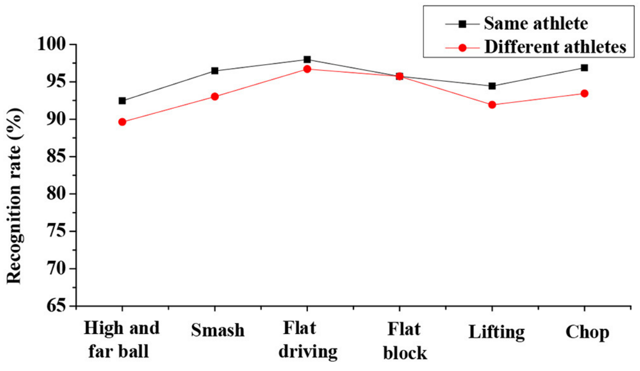


FIGURE 6 | Analysis on recognition rates when the object is a same athlete or different athletes.

Comparative Analysis of Model Simulation Accuracy

In this study, the training data set size N is selected as 120 groups and 300 groups for the simulation test, and the results are shown in **Figure 7**. The simulation test results indicate that the test accuracy for most types of motions reached a higher level when the size of the training data is 120 sets. In addition, the improvement in accuracy is extremely insignificant when the size of the training data is increased to 300. Therefore, the system takes into account the operation efficiency and stability. When the total scale of the training data is 120 groups, that is, the training data scale of each action type is 20 groups, the accuracy rate is basically stable.

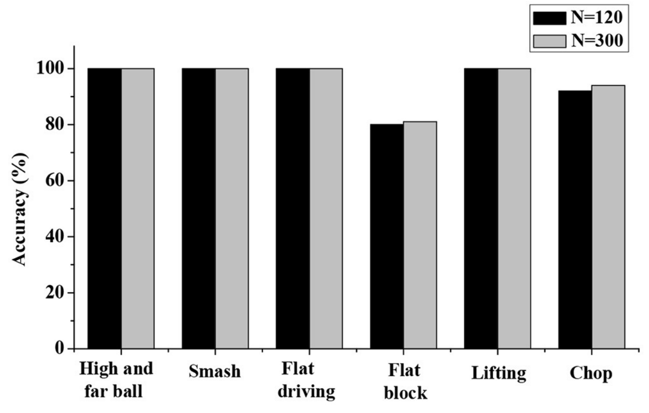


FIGURE 7 | Comparative analysis on accuracy of the model in different data set sizes.

Discussion

Analysis on recognition accuracy of the IBTR reveals that the recognition accuracy of the data pre-processing algorithm, based on the sliding window segmentation, at the moment of hitting can reach 96.03% after the optimization of the HMM model, and the application of the improved HMM algorithm shows a recognition rate for the robot of 94.50%. Thus, the recognition effect to the training set samples is good. In addition, there is no obvious difference between the recognition accuracy in whether the recognition object is the same athlete or not, and the recognition of each action is above 90%. This obviously shows that in the IBTR designed in this study, its recognition rate is significantly improved, and the recognition effect of the training set samples is good after the parameters of each dimension are adjusted and optimized.

Therefore, by applying the designed IBTR to the daily training of badminton players it can better recognize the athletes' movements, and it can significantly reduce the athletes' injuries during the training process.

CONCLUSION

The overall hardware and software framework of the IBTR system was first designed in this study, and the HMM under the machine learning algorithm was adopted to recognize the motions of the athletes. The designed IBTR was then tested and analyzed multi-directionally in terms of motion recognition accuracy, using the computer simulation. Finally, it was found that the recognition rate and accuracy of the designed IBTR are both high, and the system runs stably, which provides an experimental reference for the injury prevention in athlete training. However, there were also some shortcomings in this study. For example, the study focuses on the design of the IBTR and is less concerned with the cooperative training of athletes. In addition, the specific calculation method of the three-axis posture angle is relatively complicated in the hardware design of the IBTR, and is therefore not clearly pointed out in this study. Therefore, a cloud platform for big data analysis will be built in follow-up research for data mining and data analyses on a large amount of player games and training data, so as to analyze the technical advantages and shortcomings of various athletes, thereby promoting the ability of athletes faster and preventing the injury of athletes.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

This work was supported by Educational Science Planning in Jiangxi Province in 2018 Study on Motivation Strategies of College Students' Sports Learning Motivation Based on Self-determination Theory: Taking the Training of Campus Football Referees as an Example (No. 18YB094) and by Research Topics on Teaching Reform of East China University of Technology in 2018 Research and Practice on the Hybrid Teaching Model of Football Course Based on SPOC (No. DHJG-18-29).

REFERENCES

Cao, Z., Liao, T., Song, W., Chen, Z., and Li, C. (2020). Detecting the shuttlecock for a badminton robot: a YOLO based approach. *Exp. Syst. Appl.* 164:113833. doi: 10.1016/j.eswa.2020.113833

Chen, W., Liao, T., Li, Z., Lin, H., Xue, H., Zhang, L., et al. (2019). Using FTOC to track shuttlecock for the badminton robot. *Neurocomputing* 334, 182–196. doi: 10.1016/j.neucom.2019.01.023

Chia, J. S., Chow, J. Y., Barrett, L. A., and Burns, S. F. (2019). Reliability of a novel badminton intermittent exercise protocol. *Res. Q. Exerc Sport* 90, 487–496. doi: 10.1080/02701367.2019.1620911

Dalal, G., Gilboa, E., Mannor, S., and Wehenkel, L. (2019). Chance-constrained outage scheduling using a machine learning proxy. *IEEE Trans. Power Syst.* 34, 2528–2540. doi: 10.1109/TPWRS.2018.2889237

Gao, J., Ye, W., Guo, J., and Li, Z. (2020). Deep reinforcement learning for indoor mobile robot path planning. *Sensors* 20:5493. doi: 10.3390/s20195493

Huang, P., Fu, L., Zhang, Y., Fekete, G., and Gu, Y. (2019). Biomechanical analysis methods to assess professional badminton players' lunge performance. *J. Vis. Exp.* 148:e58842. doi: 10.3791/58842

Jalal, A., and Kim, K. (2020). Wearable inertial sensors for daily activity analysis based on Adam optimization and the maximum entropy Markov model. *Entropy* 22:579. doi: 10.3390/e22050579

Jiang, C., Zhang, H., Ren, Y., Han, Z., Chen, K. C., and Hanzo, L. (2017). Machine learning paradigms for next-generation wireless networks. *IEEE Wirel. Commun.* 24:98–105. doi: 10.1109/MWC.2016.1500356WC

Kawata, J., Morimoto, J., Higuchi, M., and Fujisawa, S. (2019). The educational effects of practical manufacturing activities in graduation research. *J. Robot. Mechatr.* 31, 391–404. doi: 10.20965/jrm.2019.p0391

Lau, B. Y. S., Ting, H. Y., and Tan, Y. W. D. (2018). Cost-benefit analysis reference framework for human motion capture and analysis systems. *Adv. Sci. Lett.* 24, 1249–1253. doi: 10.1166/asl.2018.10726

Li, H., Chang, J., Xu, F., Liu, Z., Yang, Z., Zhang, L., et al. (2019a). Efficient lidar signal denoising algorithm using variational mode decomposition combined with a whale optimization algorithm. *Remote Sens.* 11:126. doi: 10.3390/rs11020126

Li, J., and Zhang, L. (2017). The teleoperation system of service robot based on cloud services. *J. Comput.* 28, 231–245.

Li, T. H. S., Kuo, P. H., Ho, Y. F., and Liou, G. H. (2019b). Intelligent control strategy for robotic arm by using adaptive inertia weight and acceleration coefficients particle swarm optimization. *IEEE Access* 7:126929–126940. doi: 10.1109/ACCESS.2019.2939050

Liang, P., Sun, G., and Wei, S. (2019). Application of deep learning algorithm in cervical cancer MRI image segmentation based on wireless sensor. *J. Med. Syst.* 43:156. doi: 10.1007/s10916-019-1284-7

Liu, L., Wang, L., and Xu, Z. (2020). Design and implementation of badminton robot perception and control system. *Int. J. Adv. Robot. Syst.* 17:1729881420912606. doi: 10.1177/1729881420912606

Mansec, Y. L., Jérôme, P., Rouault, Q., Doron, J., and Jubeau, M. (2020). Impaired performance of the smash stroke in badminton induced by muscle fatigue. *Int. J. Sports Physiol. Perf.* 15, 52–59. doi: 10.1123/ijspp.2018-0697

Mao, Q. C., Sun, H. M., Liu, Y. B., and Jia, R. S. (2019). Fast and efficient non-contact ball detector for picking robots. *IEEE Access* 7, 175487–175498. doi: 10.1109/ACCESS.2019.2955834

Matsuo, K. (2017). Implementation and experimental evaluation of an omnidirectional wheelchair for sports and moving in rooms with narrow spaces. *Int. J. Space Based Situated Comput.* 7, 1–7. doi: 10.1504/IJSSC.2017.084118

Mizuno, N., Makishima, T., Tsuge, K., Kondo, S., and Yamakawa, S. (2019). Development of automatic badminton playing robot with distance image sensor. *IFAC Papers Online* 52, 67–72. doi: 10.1016/j.ifacol.2019.08.050

Mutaqin, A. A. R., Luqman, M., and Heru, A. S. (2020). Aplikasi dan analisis sistem fuzzy untuk pengendalian kecepatan putar motor omnidirectional pada robot badminton. *Jurnal Elektronika Otomasi Industri* 2, 61–67.

Namba, T., Yoshida, M., and Ohtsuki, Y. (2020). Machine-learning approach for constructing control landscape maps of three-dimensional alignment of asymmetric-top molecules. *J. Chem. Phys.* 153:024120. doi: 10.1063/5.0012303

Polydoros, A. S., and Nalpantidis, L. (2017). Survey of model-based reinforcement learning: applications on robotics. *J. Intell Robot. Syst.* 86, 153–173. doi: 10.1007/s10846-017-0468-y

Sapiee, M. R. M., and Annuar, K. A. M. (2018). Synchronous mobile robots formation control. *Telkomnika* 16, 1183–1192. doi: 10.12928/telkomnika.v16i3.8397

Sato, K., Watanabe, K., Mizuno, S., Manabe, M., Yano, H., Iwata, H., et al. (2017). Development and assessment of a block machine for volleyball attack training. *Adv. Robot.* 31:1144–1156. doi: 10.1080/01691864.2017.1388192

Singh, D., Singh, Y. (2018). Development and analysis of a five degrees of freedom robotic manipulator serving as a goalkeeper to train the football players. *MS&E* 402:012092. doi: 10.1088/1757-899X/402/1/012092

Steels, T., Van Herbruggen, B., Fontaine, J., De Pessemier, T., Plets, D., De Poorter, E., et al. (2020). Badminton activity recognition using accelerometer data. *Sensors* 20:4685. doi: 10.3390/s20174685

Victor, T. W., Tivesten, E., Gustavsson, P., Johansson, J., Sangberg, F., and Ljung Aust, M. (2018). Automation expectation mismatch: Incorrect prediction despite eyes on threat and hands on wheel. *Hum. Factors* 60, 1095–1116. doi: 10.1177/0018720818788164

Vu, T. H., Misra, A., Roy, Q., Wei, K. C. T., and Lee, Y. (2018). Smartwatch-based early gesture detection 8 trajectory tracking for interactive gesture-driven applications. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 1–27. doi: 10.1145/3191771

Walsh, C. G., Ribeiro, J. D., and Franklin, J. C. (2018). Predicting suicide attempts in adolescents with longitudinal clinical data and machine learning. *J. Child Psychol. Psychiatry* 59, 1261–1270. doi: 10.1111/jcpp.12916

Wang, L., Niu, W., Wang, K., Zhang, S., and Lu, T. (2019). Badminton players show a lower coactivation and higher beta band intermuscular interactions of ankle antagonist muscles during isokinetic exercise. *Med. Biol. Eng. Comput.* 57, 2407–2415. doi: 10.1007/s11517-019-02040-8

Wang, Y., Chen, M., Wang, X., Chan, R. H. M., and Li, W. J. (2018). IoT for next-generation racket sports training. *IEEE Int. Things J.* 5:4558–4566. doi: 10.1109/JIOT.2018.2837347

Zagatto, A. M., Kondric, M., Knechtle, B., Nikolaidis, P. T., and Sperlich, B. (2018). Energetic demand and physical conditioning of table tennis players. A study review. *J. Sports Sci.* 36, 724–731. doi: 10.1080/02640414.2017.1335957

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

ORIGINAL RESEARCH
published: 22 March 2021
doi: 10.3389/fnbot.2021.634340

The Analysis of Trajectory Control of Non-holonomic Mobile Robots Based on Internet of Things Target Image Enhancement Technology and Backpropagation Neural Network

Lanfei Zhao[1], *Ganlin Wang*[2], *Xiaosong Fan*[2] and *Yufei Li*[3,4]*

[1] The Higher Educational Key Laboratory for Measuring and Control Technology and Instrumentations of Heilongjiang Province, Harbin University of Science and Technology, Harbin, China, [2] Huabei Oil Communication Co., Ltd., Cangzhou, China, [3] Medical Cosmetic Center, Department of Dermatology, Tongji Hospital, Tongji University School of Medicine, Shanghai, China, [4] Department of Plastic Surgery, Huashan Hospital, Fudan University, Shanghai, China

The trajectory tracking and control of incomplete mobile robots are explored to improve the accuracy of the trajectory tracking of the robot controller. First, the mathematical kinematics model of the non-holonomic mobile robot is studied. Then, the improved Backpropagation Neural Network (BPNN) is applied to the robot controller. On this basis, a mobile robot trajectory tracking controller combining the fuzzy algorithm and the neural network is designed to control the linear velocity and angular velocity of the mobile robot. Finally, the robot target image can be analyzed effectively based on the Internet of Things (IoT) image enhancement technology. In the MATLAB environment, the performances of traditional BPNN and improved BPNN in mobile robots' trajectory tracking are compared. The tracking accuracy before and after the improvement shows no apparent differences; however, the training speed of improved BPNN is significantly accelerated. The fuzzy-BPNN controller presents significant improvements in tracking speed and tracking accuracy compared with the improved BPNN. The trajectory tracking controller of the mobile robot is designed and improved based on the fuzzy BPNN. The designed controller combining the fuzzy algorithm and the improved BPNN can provide higher accuracy and tracking efficiency for the trajectory tracking and control of the non-holonomic mobile robots.

Keywords: backpropagation neural network, Internet of Things, image enhancement, non-holonomic mobile robot, trajectory tracking and control

Edited by:
Mu-Yen Chen,
National Taichung University of Science and Technology, Taiwan

Reviewed by:
Hsin-Te Wu,
National Ilan University, Taiwan
Gwo-Jiun Horng,
Southern Taiwan University of Science and Technology, Taiwan

****Correspondence:***
Yufei Li
lufylee@hotmail.com

Received: *27 November 2020*
Accepted: *22 February 2021*
Published: *22 March 2021*

Citation:
Zhao L, Wang G, Fan X and Li Y (2021) The Analysis of Trajectory Control of Non-holonomic Mobile Robots Based on Internet of Things Target Image Enhancement Technology and Backpropagation Neural Network. Front. Neurorobot. 15:634340. doi: 10.3389/fnbot.2021.634340

INTRODUCTION

As human society enters the era of science and technology, computers, and artificial intelligence have developed rapidly; machines to replace human labor to improve production efficiency have become a reality (Ma et al., 2020). However, Lv et al. (2019) also proposed that the existing network structure migrated computing tasks to the cloud, while the increase in cloud data transmission put huge pressure on the core network and affected the quality of service (Lv and Xiu, 2019). Internet of Things (IoT) is a famous object vision.

Information-sensing equipment, such as sensors and electronic tags installed on the object, transmits information collected back and forth through the internet connection according to the agreed protocol. Connecting a simple robot to the internet will become valuable because it can obtain updated information about its environment from sensors or understand the user's whereabouts and the status of nearby devices (Marques et al., 2019). In short, robots integrated with IoT can use IoT data to help machines interact with each other and take necessary actions, enabling robots to communicate effectively and make appropriate decisions by themselves. The core of "IoT+Robot" is the ubiquitous sensors, cameras, and actuators embedded in the environment, as well as autonomous robots that collect data in real-time (Rehman et al., 2019). Sensors provide not only raw data but also interpretation and abstraction to some degree, which can be utilized for decision-making or high-level automation. Lv et al. (2019) applied the sensor technology of the ZigBee wireless network organization, which could provide people with a smarter and more comfortable living environment (Lv et al., 2019). Connecting machine vision systems to IoT can create powerful network functions that can recognize objects from cameras. Such functions can enhance the local nodes' intelligence and autonomy, reducing the processing load on the central server and achieving a better-distributed control architecture.

The IoT-based multi-robot collaborative operation utilizes the intelligent perception inside and outside the robot, makes timely judgments and control decisions according to the signals collected by the network, and timely issues control instructions to the robot to ensure that multiple robots complete tasks safely and efficiently (Özdemir, 2019; Michie et al., 2020). Mobile robots are highly intelligent systems that can continuously obtain information of the surrounding environment and themselves through sensors in real-time, make decisions, analyze, plan for different environments, and control the drive motors to move autonomously toward the targets, thereby completing specific tasks.

Tracking control in mobile robots' motion control has always been a sophisticated problem, which has received extensive attention from researchers in this field. Yang and Pan proposed a sliding mode control method for wheeled mobile robots, established a motion control model for mobile robots, and designed a sliding mode trajectory tracking controller, which effectively reduced the jitter of sliding mode control's input, accelerated the convergence speed, and improved the tracking accuracy (Yang and Pan, 2018). Tinh and Linh improved the online weight adjustment algorithm based on backpropagation and proposed an adaptive tracking controller based on a three-layer neural network, which could ensure the stability of the entire closed-loop system and realize the desired mobile robots' trajectory tracking performance (Tinh and Linh, 2018).

In practical applications, trajectory tracking of mobile robots should ensure versatility while ensuring stability and robustness optimization. Therefore, the focus is on the trajectory control of mobile robots. Here, a trajectory tracking method combining improved Backpropagation Neural Network (BPNN) and fuzzy neural network is proposed, denoted as fuzzy-BPNN. Then this method's feasibility and accuracy in robots' trajectory tracking and control are verified through simulation. On this basis, the IoT multi-sensor data fusion and target infrared image enhancement technology are researched. The multi-robot system can fuse the sensor data, enhance the target images, and provide new ideas for robot obstacle avoidance through fuzzy control.

MATERIALS AND METHODS

Mathematical Kinematics Models of Non-holonomic Mobile Robots

In mechanics, mathematical equations containing coordinate parameters can express constraints. The motion constraints that mobile robots are subjected to include holonomic constraints and holonomic constraints. Holonomic constraints are restrictions on the configuration space, and non-holonomic constraints are restrictions on system motion (Chu et al., 2018). The holonomic constraint reduces the dimensionality of the configuration space. The system can transform the holonomic constraints it receives into the constraints on the position through integration during the motion. The non-holonomic constraints reduce the dimensionality of the velocity. The equations of holonomic constraints and non-holonomic constraints can be expressed as Equations (1) and (2):

$$h\left(q,t\right)=0 \tag{1}$$

$$h\left(q,\dot{q},t\right)=0 \tag{2}$$

In Equations (1) and (2), q represents the coordinate vector of the system, $\dot{q}$ represents the velocity vector of the system, and t is the time parameter.

For practical problems, the motion constraint can be transformed into a linear relationship with the system velocity $\dot{q}$; that is, the Pfaffian constraint:

$$h\left(q,\dot{q},t\right)=\sum_{i=1}^{n}A_i\left(q\right)\cdot\left(\dot{q}\right)=A\left(q\right)\cdot\left(\dot{q}\right)=0 \tag{3}$$

In Equation (3), $A\left(q\right)\in R^{m\times n}$ represents a set of m velocity constraints, and $A_i\left(q\right)\in R^{1\times n}$ is the row vector of $A\left(q\right)$, which is a constraint on the direction of the generalized velocity q of the system.

A wheel moves on the ground, as shown in **Figure 1**. Four parameters can describe the wheel's configuration: the contact points x and y with the ground, the current rotation angle θ, and the forward direction Φ. If the wheel makes a non-slip motion, the direction of the wheel will always be $(cos\phi, sin\phi)$. The non-slip constraint does not reduce the dimension of the wheel configuration space; that is, the wheel can reach any position on the plane. However, this constraint reduces the dimensionality of the wheel's velocity space so that the wheel can only move in direction $(cos\phi, sin\phi)$ at a particular time.

Assuming that the wheel's radius is r, the distance between the wheel's center (x_c, y_c) and the ground track is always the same during the non-slip rolling motion of the linear track. The non-slip rolling between it constrains the wheel's motion and the ground, expressed as the instantaneous velocity v of the wheel's contact point and the ground track is zero. Therefore, the wheel's motion can be regarded as a rapid rotation around the contact point, and the constraint equation can be expressed as:

$$y_c = r \tag{4}$$

$$A(q) \cdot (\dot{q}) = [1\ 0\ \text{-}r] \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\phi} \end{bmatrix} = \dot{x}_c\ -r\ \dot{\phi} = v = 0 \tag{5}$$

Equation (4) is a geometric constraint on the wheel, and Equation (5) represents a linear motion constraint, which can be further expressed by integral:

$$x_c - r\phi = C \tag{6}$$

In Equation (6), C is the integral constant.

The non-holonomic constraint equation does not have a corresponding geometric constraint. Hence, it puts no restriction on the position vector but only restrictions on the particle's velocity at each position. Therefore, the non-holonomic constraint does not reduce the number of independent generalized coordinates; instead, it only reduces the number of independent generalized velocities (Gutiérrez-Giles et al., 2018).

A non-holonomic mobile robot is also a non-holonomic system, and the non-holonomic constraint equation can reflect its motion characteristics. While the robots are moving, two physical phenomena, tire rolling and sliding, will occur when the wheels contact the ground. The structure of a non-holonomic mobile robot is shown in **Figure 2**. One coordinate system is the global coordinate system XOY; the other coordinate system is the local coordinate system xoy (the mass center of the robot is the origin). In **Figure 2**, R is the radius of the robot's driving wheel, θ is the robot's forward direction angle, q is the robot's pose, $[v\,\omega]^T$ is the robot's control quantity (linear velocity and angular velocity), v_L and v_R are the linear velocity of the robot's left and right wheels, and L is the distance between the centers of the robot's two driving wheels.

For error reduction, assuming that the mobile robot moves along a linear track gradually, the wheels do not slide left and right. Then the non-holonomic constraints and the kinematics model can be expressed as in Equations (7) and (8):

$$\dot{y}\ \cos\phi - \dot{x}\ \sin\theta = 0 \tag{7}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{8}$$

The non-holonomic mobile robot's left and right wheels' linear velocities v_L and v_R share the following relationship with its linear velocity v and angular velocity ω :

$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = \begin{bmatrix} 1/R & L/2R \\ 1/R & L/2R \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{9}$$

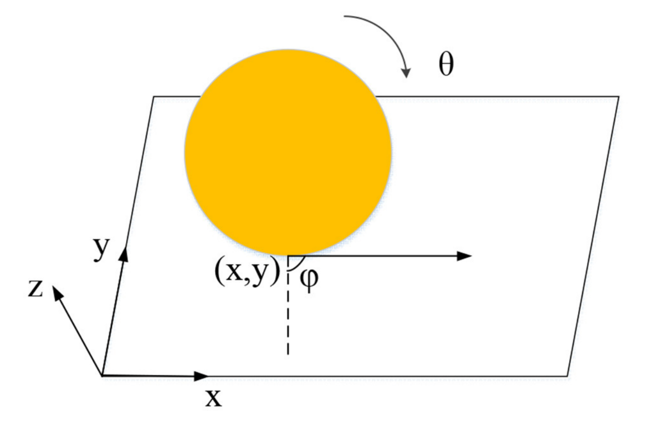


FIGURE 1 | Non-slip rolling constraints of wheels on the ground.

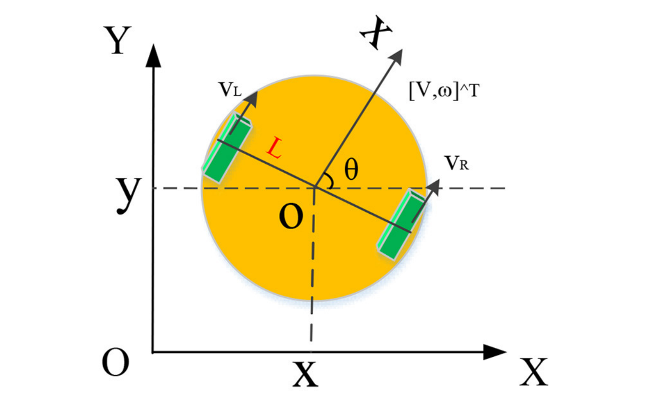


FIGURE 2 | Structure of a non-holonomic mobile robot.

In summary, the kinematics model of the discussed non-holonomic mobile robot is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{R}{2}\cos\theta & \frac{R}{2}\cos\theta \\ \frac{R}{2}\sin\theta & \frac{R}{2}\sin\theta \\ \frac{R}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} v_L \\ v_R \end{bmatrix} \tag{10}$$

Robot Trajectory Tracking Based on Improved BPNN

Mobile robots are widely used in unmodeled spaces and environments; thus, accurate trajectory tracking is the basis of practical applications (Alshakarchi and Al-Maliky, 2018; Singh and Thongam, 2018; Tu et al., 2019; Wang et al., 2019). The traditional control method depends too much on the dynamic model, resulting in low robustness. The intelligent neural networks have strong robustness and adaptability, presenting significant advantages in trajectory tracking and control of mobile robots. As the core of the feedforward network in artificial neural networks, BPNN is widely applied to solve problems such as function approximation and pattern recognition (Yi et al., 2019). The most commonly used transfer functions of backpropagation neurons are log function and tan function, and the output can be expressed as $y = \log sig\left(Wp + b\right)$. Generally, BPNN presents a multi-layer structure. The model of backpropagation neuron and the two-layer structure diagram are shown in **Figure 3**.

BPNN needs to adjust the weights according to each training sample, which requires a massive amount of training data in practical applications, resulting in reduced efficiency of weight adjustment and failure to meet the real-time requirements (Singh and Thongam, 2019). Therefore, the traditional BPNN is divided into several smaller sub-networks, which are trained separately to improve computational efficiency. While dividing the BPNN into n sub-networks, it is also necessary to divide the training samples into n groups of sub-samples. It is also necessary to compare the current training sample with the previous training sample to calculate the corresponding group's mean square error sum. Finally, according to the comparison result, whether to input the data into BPNN for operation is decided. The improved BPNN eliminates the need for repeated calculations, which significantly shortens the calculation time for large-scale neural networks and improves weight adjustment efficiency.

The improved BPNN is adopted to model the unknown parameters of the robot, and a dynamic controller that meets the real-time requirements of mobile robots is designed. First, the kinematic equation of the mobile robot is expressed as:

$$\begin{cases} \dot{x} = v\cos \\ \dot{y} = v\sin \\ \dot{\theta} = \omega \end{cases} \tag{11}$$

In Equation (11), $\left(x, y\right)$ is the actual position of the mobile robot, θ is the azimuth angle, and both the linear velocity v and the angular velocity ω are control inputs in the kinematic model. Robot trajectory tracking is to track the target robot with pose $q_r = \left[x_r, y_r, \theta_r\right]^T$ and velocity $\dot{q}_r = \left[v_r, \omega_r\right]^T$. The tracking error of the mobile robot is expressed as:

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = T_e\left(q_r - q\right) \tag{12}$$

The error change rate of the mobile robot can be expressed as:

$$\dot{e} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \omega e_2 - v + v_r\cos e_3 \\ -\omega e_2 + v_r\sin e_3 \\ \omega_r - \omega \end{bmatrix} \tag{13}$$

The improved BPNN is applied to robot dynamics control. If a vector $P(\cdot)$ is a variable function, a static neural network is utilized for modeling, and the following equation will be obtained:

$$P(\cdot) = \left[\{W_P\}^T \cdot \{\xi_P(\cdot)\}\right] + E_P(\cdot) \tag{14}$$

In Equation (14), $\{W_P\}^T$ and $\{\xi_P(\cdot)\}$ are OpenGL vectors, each element is a model error vector, and $(\cdot)$ represents a general vector or matrix.

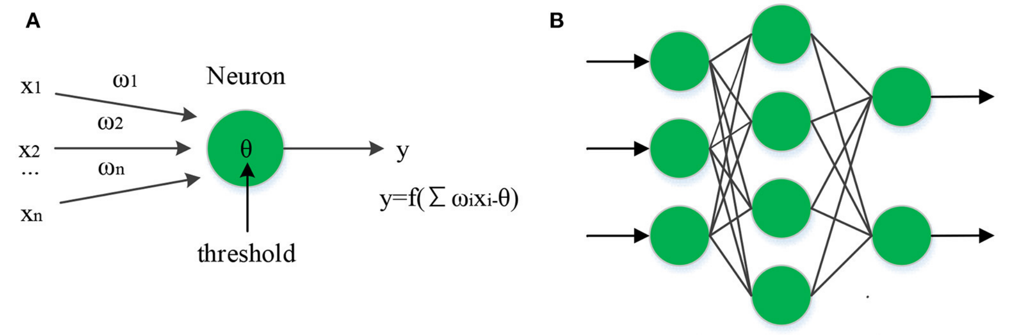


FIGURE 3 | Model of backpropagation neuron and the two-layer structure diagram. **(A)** BP neuronal structure. **(B)** BP neural structure.

Robot Trajectory Tracking Based on the Fuzzy Algorithm Combining Neural Networks

In fuzzy systems, the design of fuzzy sets, membership functions, and fuzzy rules are based on empirical knowledge. This analysis method has a lot of subjectivity (Lu et al., 2018). Hence, the learning mechanism is introduced into the fuzzy system to modify and improve the membership function and fuzzy rules through continuous learning. The connection between the fuzzy system and the fuzzy neural network shows that the fuzzy neural network is essentially the realization of the fuzzy system. The difference between the two reveals that the fuzzy neural network has the characteristics of the neural network. Introducing the learning ability of the neural network into the fuzzy system and representing the fuzzy processing, fuzzy reasoning, and precise calculation of the fuzzy system through a distributed neural network is an important way to realize the self-organization and self-learning of the fuzzy system (Amador-Angulo et al., 2016). In a fuzzy neural network, the input and output nodes are used to fuzzify the input and output signals of the system (Caraveo et al., 2017; Lagunes et al., 2019). The hidden nodes of this neural network express the membership function and fuzzy rules, and the parallel processing capability of the neural network makes the inference ability of the fuzzy system greatly improved. The fuzzy neural network combines fuzzy system and neural network. A fuzzy neural network is essentially a conventional neural network that assigns fuzzy input signals and fuzzy weights. Its learning algorithm is usually a typical neural network's learning algorithm or its extension.

According to the kinematic model of the non-holonomic mobile robot, the current pose of the robot can be obtained as long as $u = [v\, \omega]^T$ is controlled. Assuming that the actual pose of the robot is $p = \left[x\, y\, \theta\right]^T$, the actual motion velocity is $[v\, \omega]^T$; the reference pose is $p_r = \left[x_r\, y_r\, \theta_r\right]^T$, and the reference motion velocity is $[v_r\, \omega_r]^T$. Then the error vector between the actual pose and the reference pose is $p_e = \left[x_e\, y_e\, \theta_e\right]^T$. Essentially, trajectory tracking of a non-holonomic mobile robot is to find a bounded input for any initial pose and velocity error and make:

$$\lim_{t\to\infty} \left\| \left[x_e\, y_e\, \theta_e\right]^T \right\| = 0 \tag{15}$$

The error vector between the actual pose and the reference pose can be expressed as:

$$p_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix}^T = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{16}$$

Equation (16) is derived to obtain the differential equation of the mobile robot's tracking error, expressed as:

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} v_r \cos\theta_e - v + y_e\omega \\ v_r \sin\theta_e - x_e\omega \\ \omega_r - \omega \end{bmatrix} \tag{17}$$

According to the above principles and derivations, a fuzzy-BPNN trajectory tracking controller for mobile robots is designed, as shown in **Figure 4**. It can control the linear velocity and angular velocity of mobile robots.

The actual trajectory tracking process will be disturbed by obstacles and other external environmental factors. Hence, the pose error will change significantly (Boujelben et al., 2017; Lu et al., 2017; Bencherif and Chouireb, 2019). Fuzzy logic can imitate the thinking way of the human brain and process systems with unknown models. Therefore, fuzzy logic can determine the location of obstacles while mobile robots are moving. Its core is to process the pose error data to avoid problems such as an increased number of fuzzy neural network's rules and the repeated change of weights. A data detection step is added, which uses the averaged error changes of the previous pose as a reference and compares it with the new pose error for decision-making. The data detection step can eliminate the error data with apparent mutations in the input. The fuzzy neural network's parameters do not need to be adjusted significantly. Therefore, the stability of the system and the overall calculation efficiency can be improved.

IoT Multi-Sensor Data Fusion and Image Enhancement

A primary function of IoT technology is information perception. The object is connected to the network for information exchange through the installed sensors, electronic tags, and other sensing devices, realizing intelligent identification, positioning, and

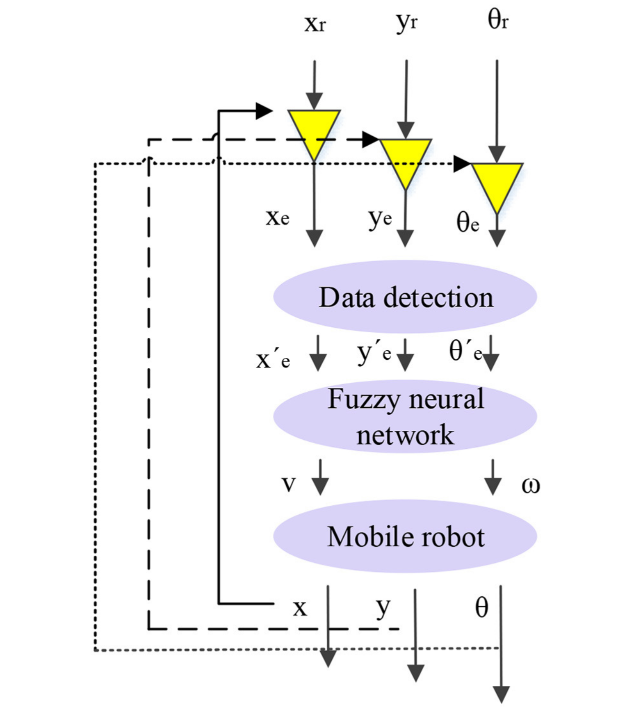


FIGURE 4 | A fuzzy-BPNN trajectory tracking controller for mobile robots.

supervision operations. In the process of information collection, the mobile robots' sensors summarize multiple sensors' data through data fusion, thereby reducing the transmission of redundant information and improving the stability and accuracy of the system (Jing et al., 2017).

IoT intelligent image enhancement establishes an image enhancement model through a wavelet conversion scale, which can adaptively adjust the window of different digital image frequencies to intelligently and adaptively enhance the IoT images. To improve the accuracy of environmental information collection, a mobile robot often uses an adaptive fusion algorithm to process the environmental information collected by its system (Yamashita et al., 2017; Ravi and Krishnan, 2018; Fan et al., 2019). Since the background of the original image collected by the infrared thermal imager and the monitored target robot are not notable, the details of the target image are difficult to identify, and the image feature cannot be clearly extracted. Hence, the image enhancement method is adopted to process the target image. The grayscale transformation method is adopted to enhance the image's contrast and improve its visual effects. Standard methods include (1) direct grayscale transformation; (2) transformation with the help of histogram; (3) transformation with a series of operations between images (such as addition and subtraction) (Chen et al., 2020). The direct grayscale transformation is the most commonly used and most convenient method. First, in the process of negating the gray, the image needs to be negated; that is, to reverse the gray value of the original image. Second, the image's contrast enhances the contrast of each part of the original image. Sometimes the dynamic range of the original image can exceed the allowable range of display devices. Therefore, if the original image is used directly, some details will be lost (Long et al., 2018; Singh et al., 2018). The solution is to compress the original image in grayscale. The principle of grayscale negation, contrast enhancement, and dynamic image compression is shown in **Figure 5**.

Moreover, image noise processing is vital in image enhancement, including impulse noise and Gaussian noise. Usually, image denoising separates the image into two kinds of noise; then, the median filter algorithm and the mean filter algorithm are used to eliminate these noises. The median filter algorithm arranges the to-be-processed pixels' gray values in the neighborhood from large to small. Then it selects the median value to replace the to-be-processed pixel value in the template center. The mean filtering algorithm removes the sudden change by calculating the mean value of a central point and several surrounding points, thereby removing the noises.

Simulations Experiments

MATLAB is chosen as the simulation software to analyze the performances of the improved BPNN and the fuzzy-BPNN in robot trajectory tracking and control. MATLAB is one of the excellent science and technology application software. It has powerful calculation and visualization functions but is simple and easy to operate. In particular, the accompanying toolbox that supports more than 30 different fields has made it the basic tool and preferred platform for computer-aided design and analysis in various fields.

The performance differences between the two algorithms are compared. The reference linear velocity of the target robot is set to $v_r = 3$ m/s, the reference angular velocity is set to $\omega_r = 2$ rad/s, the initial pose is set to $\left[x_r\ y_r\ \theta_r\right] = (-5, 0.25, 0)$, and the initial pose error is set to $[e_1\ e_2\ e_3]^T = [2.6\ 2.4\ \pi/2]^T$. Ten groups of samples are taken for training. Each group of samples is divided into five groups of sub-samples equally. The mean square errors of the corresponding sub-samples of each group of samples (**Table 1**) are summed. The errors eventually converge to 0, the convergence speed is fast, and satisfactory results can be achieved.

RESULTS AND DISCUSSION

Trajectory Tracking Performance of Fuzzy-BPNN

The effects of traditional BPNN and improved BPNN in mobile robots' trajectory tracking are compared, and the results are shown in **Figures 6** and **7**. The improved BPNN algorithm has a slight improvement in tracking accuracy than the traditional BPNN; however, the difference between the two is not notable. On the contrary, the improved BPNN algorithm has a significant improvement in training speed, showing its performance advantages in processing huge samples in reality.

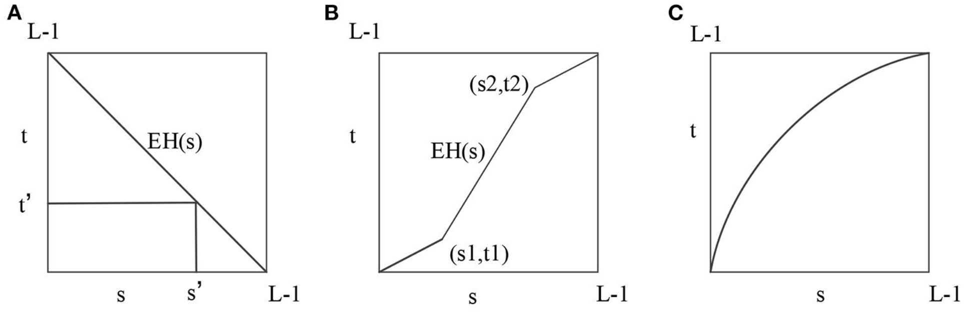


FIGURE 5 | Principle of each image enhancement step. **(A)** Inversion of gray scale. **(B)** Enhance contrast. **(C)** Dynamic image compression.

The fuzzy-BPNN algorithm is applied to track the mobile robot's trajectories, whose effects are compared with the improved BPNN. The simulation results are shown in **Figure 8**. The fuzzy-BPNN controller has significantly improved tracking speed and tracking accuracy compared with the improved BPNN, proving its effectiveness.

Non-holonomic Mobile Robot's Trajectory Tracking and Control Results

The designed fuzzy-BPNN is integrated with the sliding mode trajectory tracking controller to solve the trajectory tracking and control problems of non-holonomic mobile robots. The tracking and control of typical circular and curved trajectories are simulated. **Figures 9** and **10** show the tracking effects and pose error changes in circular trajectory tracking, and **Figures 11** and **12** demonstrate the tracking effects and pose error changes in curve trajectory tracking. The designed algorithm integrating the sliding mode trajectory tracking controller shows good trajectory tracking and control effects for these two different trajectories.

CONCLUSION

As an essential branch of robotics, mobile robots can continuously obtain the surrounding environment's status through sensors to make decisions and complete highly intelligent tasks. Robot intelligence is embodied in planning and executing an optimal path for mobile robots. Its core is trajectory tracking and control. Here, the mathematical kinematics models of non-holonomic mobile robots are analyzed first. Then the advantages of BPNN being widely used in pattern recognition and other problems are discussed. The traditional BPNN is improved, considering loads of data need to be trained in practical applications. The computational

TABLE 1 | The sum of the mean square errors of the sub-samples of each training sample.

Training samples	(1)	(2)	(3)	(4)	(5)
Group 1	0	0	0	0	0
Group 2	3.4	53.5	6.6	12.3	78.9
Group 3	11.3	5.6	5.3	4.3	53.5
Group 4	30.4	11.2	54.5	45.7	6.6
Group 5	6.9	22.3	34.1	23.6	5.9
Group 6	12.5	91.3	1.8	66.5	45.6
Group 7	34.5	11.7	7.2	4.8	12.3
Group 8	88.9	23.5	4.7	67.8	3.4
Group 9	2.2	23.4	44.6	32.4	11.1
Group 10	77.4	32.5	38.8	44.6	65.4

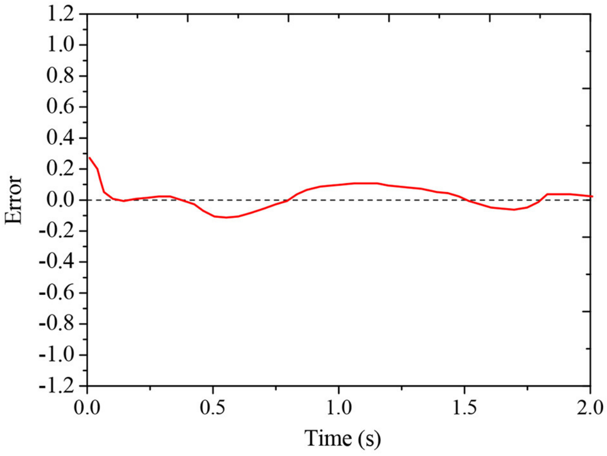


FIGURE 7 | Trajectory tracking error curve of the improved BPNN algorithm.

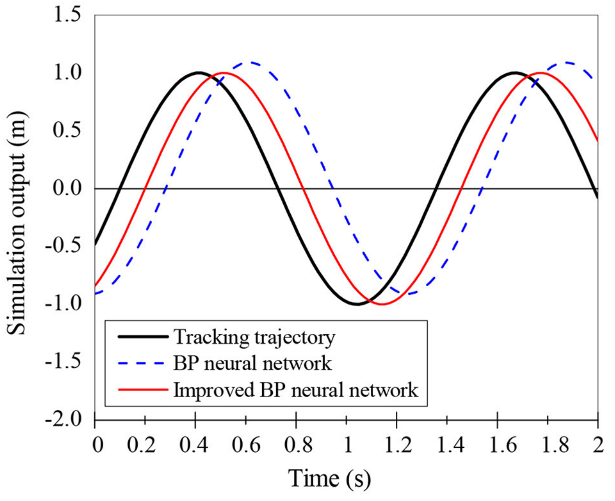


FIGURE 6 | Trajectory tracking effect of the improved BPNN algorithm.

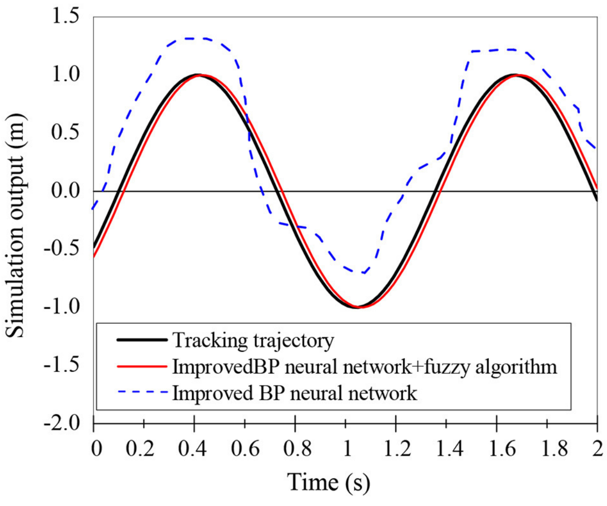


FIGURE 8 | Trajectory tracking effect of fuzzy-BPNN.

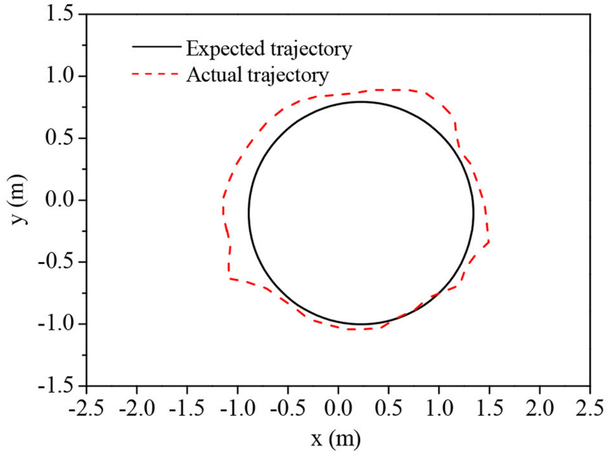


FIGURE 9 | Circular trajectory's tracking effect.

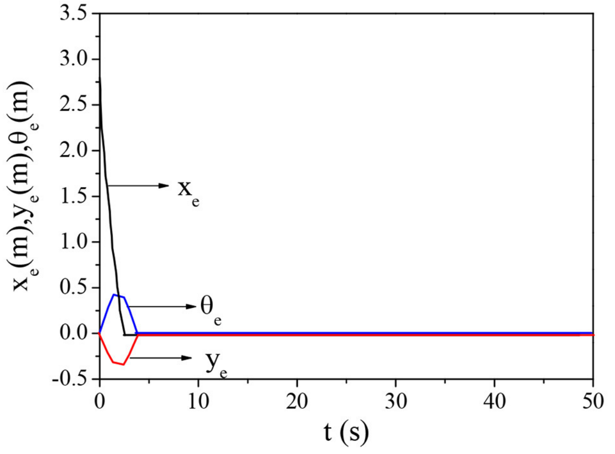


FIGURE 10 | Changes in pose error in circular trajectory tracking.

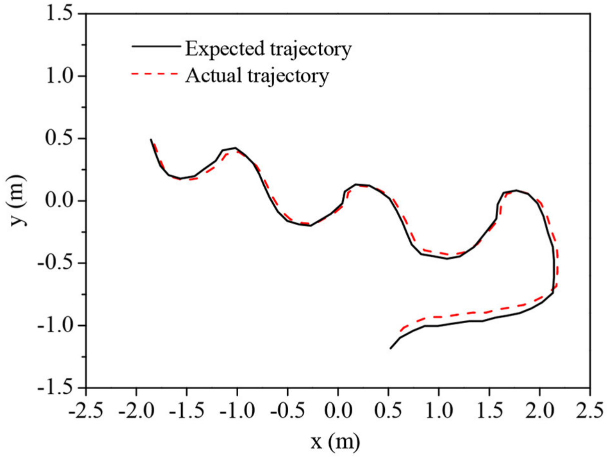


FIGURE 11 | Curve trajectory's tracking effect.

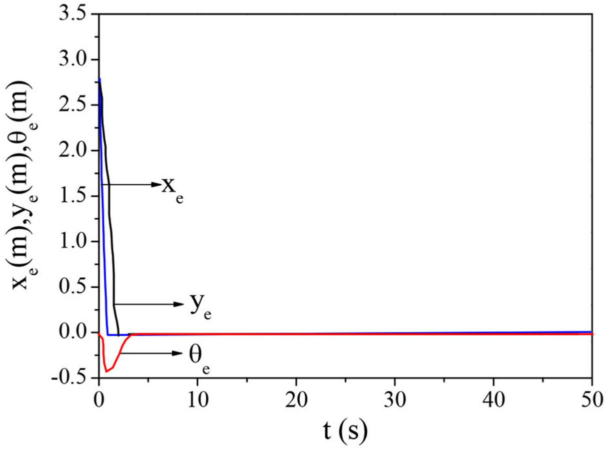


FIGURE 12 | Changes in pose error in curve trajectory tracking.

efficiency is improved by dividing BPNN into several smaller sub-networks for separate training. A fuzzy-BPNN tracking controller for mobile robot trajectories is designed to control the robot's linear velocity and angular velocity, in an effort to improve the accuracy of trajectory tracking. In addition, adaptive fusion algorithms and image enhancement techniques are used to process the environmental information collected by mobile robots to improve the accuracy of the environmental information collection.

A simulation experiment is run in the MATLAB environment to analyze the performances of the improved BPNN and the fuzzy-BPNN in robot trajectory tracking and control. Compared with traditional BPNN, the improved BPNN algorithm has a significant improvement in training speed, which has better application value for the large sample problems in reality. The fuzzy-BPNN controller has notably improved the tracking speed and accuracy compared with the improved BPNN algorithm. The designed fuzzy-BPNN algorithm is integrated with the sliding mode trajectory tracking controller, whose performances in tracking and controlling circular and curve trajectories are simulated. The designed fuzzy BPNN algorithm is integrated with the sliding mode trajectory tracking controller. The circular and curved trajectories are simulated, and both show good trajectory tracking and control effects. However, the external environment is complicated and changeable; maintaining the stability of the mobile robot controller will be the principal direction of the following research.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

This work was supported by the Fundamental Research Foundation for Universities of Heilongjiang Province (No. LGYC2018JC050).

REFERENCES

Alshakarchi, D. A. K., and Al-Maliky, F. T. (2018). Lower Limb Rehabilitation Robotics System Based on BP Neural Network. *Int. J. Appl. Eng. Res.* 13, 5889–5893.

Amador-Angulo, L., Mendoza, O., Castro, J. R., Rodríguez-Díaz, A., Melin, P., and Castillo, O. (2016). Fuzzy sets in dynamic adaptation of parameters of a bee colony optimization for controlling the trajectory of an autonomous mobile robot. *Sensors*16:1458. doi: 10.3390/s16091458

Bencherif, A., and Chouireb, F. (2019). A recurrent TSK interval type-2 fuzzy neural networks control with online structure and parameter learning for mobile robot trajectory tracking. *Appl. Intell.* 49, 3881–3893. doi: 10.1007/s10489-019-01439-y

Boujelben, M. Á., Rekik, C., and Derbel, N. (2017). A reactive approach for mobile robot navigation in static and dynamic environment using fuzzy logic control. *Int. J. Model. Identif. Control* 27, 293–302. doi: 10.1504/IJMIC.2017.084722

Caraveo, C., Valdez, F., and Castillo, O. (2017). A new meta-heuristics of optimization with dynamic adaptation of parameters using type-2 fuzzy logic for trajectory control of a mobile robot. *Algorithms* 10:85. doi: 10.3390/a10030085

Chen, J., Yang, X., Lu, L., Li, Q., Li, Z., and Wu, W. (2020). A novel infrared image enhancement based on correlation measurement of visible image for urban traffic surveillance systems. *J. Intell. Transp. Syst.* 24, 290–303. doi: 10.1080/15472450.2019.1642753

Chu, X., Peng, Z., Wen, G., and Rahmani, A. (2018). Distributed fixed-time formation tracking of multi-robot systems with nonholonomic constraints. *Neurocomputing* 313, 167–174. doi: 10.1016/j.neucom.2018.06.044

Fan, Y. C., Liu, Y. C., and Chen, H. Y. (2019). Image Enhancement Using Back-propagation Neural Network Based Histogram Equalization for Image Sensor. *Sensor Lett.* 17, 124–131. doi: 10.1166/sl.2019.4055

Gutiérrez-Giles, A., Ruggiero, F., Lippiello, V., and et al., (2018). Nonprehensile manipulation of an underactuated mechanical system with second-order nonholonomic constraints: the robotic hula-hoop. *IEEE Robot. Autom. Lett.* 3, 1136–1143. doi: 10.1109/LRA.2018.2792403

Jing, L., Wang, T., Zhao, M., and Wang, P. (2017). An adaptive multi-sensor data fusion method based on deep convolutional neural networks for fault diagnosis of planetary gearbox. *Sensors* 17:414. doi: 10.3390/s17020414

Lagunes, M. L., Castillo, O., Soria, J., Garcia-Valdez, M., and Valdez, F. (2019). Optimization of granulation for fuzzy controllers of autonomous mobile robots using the Firefly Algorithm. *Granul. Comput.* 4, 185–195. doi: 10.1007/s41066-018-0121-6

Long, M., Li, Z., Xie, X., Li, G., and Wang, Z. (2018). Adaptive image enhancement based on guide image and fraction-power transformation for wireless capsule endoscopy. *IEEE Trans. Biomed. Circuits Syst.* 12, 993–1003. doi: 10.1109/TBCAS.2018.2869530

Lu, X., Zhao, Y., and Liu, M. (2018). Self-learning interval type-2 fuzzy neural network controllers for trajectory control of a Delta parallel robot. *Neurocomputing* 283, 107–119. doi: 10.1016/j.neucom.2017.12.043

Lu, X. G., Liu, M., and Liu, J. X. (2017). Design and optimization of interval type-2 fuzzy logic controller for delta parallel robot trajectory control. *Int. J. Fuzzy Syst.* 19, 190–206. doi: 10.1007/s40815-015-0131-3

Lv, Z., Hu, B., and Lv, H. (2019). Infrastructure monitoring and operation for smart cities based on IoT system. *IEEE Trans. Industr. Inform.* 16, 1957–1962. doi: 10.1109/TII.2019.2913535

Lv, Z., and Xiu, W. (2019). Interaction of edge-cloud computing based on SDN and NFV for next generation IoT. *IEEE Internet Things J.* 7, 5706–5712. doi: 10.1109/JIOT.2019.2942719

Ma, X., Wang, R., Zhang, Y., Jiang, C., and Abbas, H. (2020). A name disambiguation module for intelligent robotic consultant in industrial internet of things. *Mech. Syst. Signal Process.* 136:106413. doi: 10.1016/j.ymssp.2019.106413

Marques, G., Pires, I. M., Miranda, N., and Pitarma, R. (2019). Air quality monitoring using assistive robots for ambient assisted living and enhanced living environments through internet of things. *Electronics* 8:1375. doi: 10.3390/electronics8121375

Michie, C., Andonovic, I., Davison, C., Hamilton, A., Tachtatzis, C., Jonsson, N., et al. (2020). The internet of things enhancing animal welfare and farm operational efficiency. *J. Dairy Res.* 87, 20–27. doi: 10.1017/S0022029920000680

Özdemir, V. (2019). The big picture on the "AI turn" for digital health: the internet of things and cyber-physical systems. *OMICS J. Integr. Biol.* 23, 308–311. doi: 10.1089/omi.2019.0069

Ravi, P., and Krishnan, J. (2018). Image enhancement with medical image fusion using multiresolution discrete cosine transform. *Mater. Today Proc.* 5, 1936–1942. doi: 10.1016/j.matpr.2017.11.296

Rehman, M. H., Yaqoob, I., Salah, K., Imran, M., Jayaraman, P. P., and Perera, C. (2019). The role of big data analytics in industrial Internet of Things. *Future Gener. Comput. Syst.* 99, 247–259. doi: 10.1016/j.future.2019.04.020

Singh, H., Kumar, A., Balyan, L. K., and Singh, G. K. (2018). Swarm intelligence optimized piecewise gamma corrected histogram equalization for dark image enhancement. *Comput. Electr. Eng.* 70, 462–475. doi: 10.1016/j.compeleceng.2017.06.029

Singh, N. H., and Thongam, K. (2018). Mobile robot navigation using MLP-BP approaches in dynamic environments. *Arab. J. Sci. Eng.* 43, 8013–8028. doi: 10.1007/s13369-018-3267-2

Singh, N. H., and Thongam, K. (2019). Neural network-based approaches for mobile robot navigation in static and moving obstacles environments. *Intell. Serv. Robot.* 12, 55–67. doi: 10.1007/s11370-018-0260-2

Tinh, N., and Linh, L. E. (2018). Neural network-based adaptive tracking control for a nonholonomic wheeled mobile robot with unknown wheel slips, model uncertainties, and unknown bounded disturbances. *Turk. J. Electr. Eng. Comput. Sci.* 26, 378–392. doi: 10.3906/elk-1705-167

Tu, X., Zhou, Y. F., Zhao, P., and Cheng, X. (2019). Modeling the static friction in a robot joint by genetically optimized BP neural network. *J. Intell. Robot. Syst.* 94, 29–41. doi: 10.1007/s10846-018-0796-6

Wang, F., Chao, Z., Huang, L., Li, H., and Zhang, C. (2019). Trajectory tracking control of robot manipulator based on RBF neural network and fuzzy sliding mode. *Clust. Comput.* 22, 5799–5809. doi: 10.1007/s10586-017-1538-4

Yamashita, H., Sugimura, D., and Hamamoto, T. (2017). Low-light color image enhancement via iterative noise reduction using RGB/NIR sensor. *J. Electron. Imaging* 26:043017. doi: 10.1117/1.JEI.26.4.043017

Yang, L., and Pan, S. (2018). A sliding mode control method for trajectory tracking control of wheeled mobile robot. *J. Phys. Conf.* 1074:012059. doi: 10.1088/1742-6596/1074/1/012059

Yi, Z., Li, G., Chen, S., Xie, W., and Xu, B. (2019). A navigation method for mobile robots using interval type-2 fuzzy neural network fitting

Q-learning in unknown environments. *J. Intell. Fuzzy Syst.* 37, 1113–1121. doi: 10.3233/JIFS-182560

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