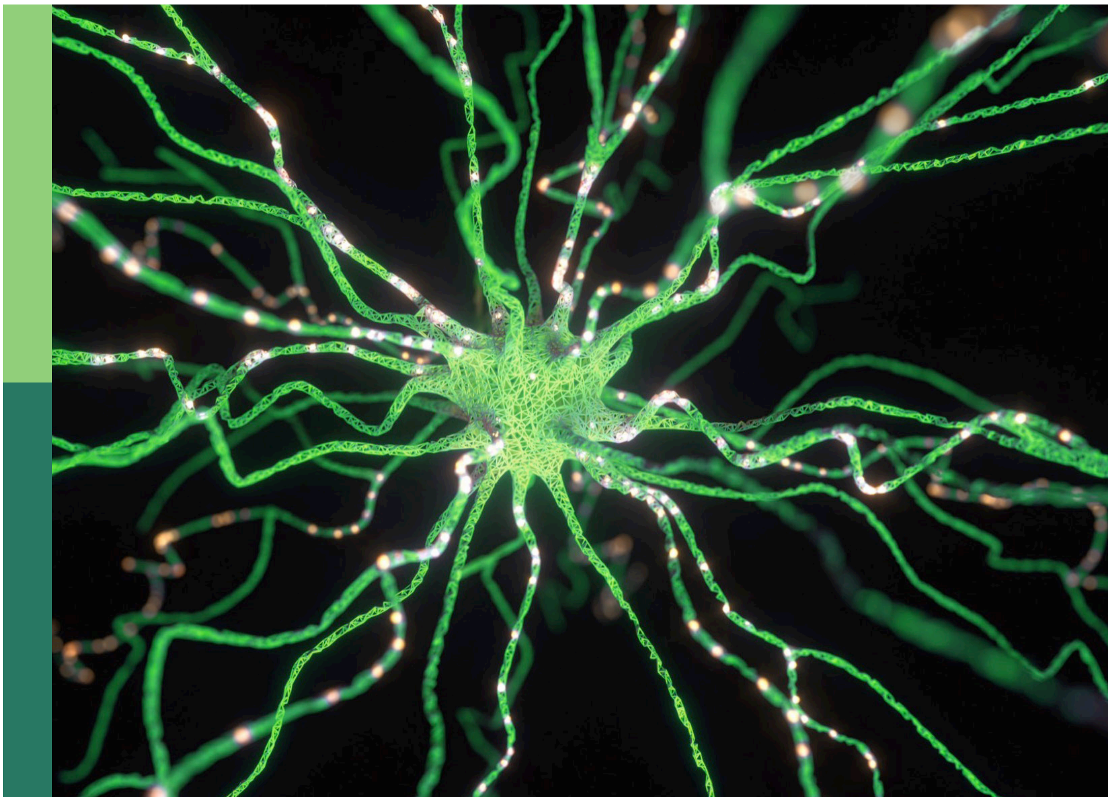
frontiers | Research Topics

Advanced planning, control, and signal processing methods and applications in robotic systems volume II

Edited by
Zhan Li, Shuai Li and Yan Wu

Published in
Frontiers in Neurorobotics



ISSN 1664-8714
ISBN 978-2-8325-2396-4
DOI 10.3389/978-2-8325-2396-4

About Frontiers

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

Frontiers journal series

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

Dedication to quality

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

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

Advanced planning, control, and signal processing methods and applications in robotic systems volume II

Topic editors

Zhan Li — Swansea University, United Kingdom

Shuai Li — Swansea University, United Kingdom

Yan Wu — Institute for Infocomm Research (A*STAR), Singapore

Citation

Li, Z., Li, S., Wu, Y., eds. (2023). *Advanced planning, control, and signal processing methods and applications in robotic systems volume II*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-8325-2396-4

Table of contents

frontiers | Frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 13 June 2022
doi: 10.3389/fnbot.2022.928636

Harmonic Noise-Tolerant ZNN for Dynamic Matrix Pseudoinversion and Its Application to Robot Manipulator

Bolin Liao[1], *Yuyan Wang*[2], *Jianfeng Li*[1]*, *Dongsheng Guo*[3] *and Yongjun He*[1]

[1] *College of Information Science and Engineering, Jishou University, Jishou, China,* [2] *College of Mathematics and Statistics, Jishou University, Jishou, China,* [3] *School of Information Science and Engineering, Huaqiao University, Xiamen, China*

As we know, harmonic noises widely exist in industrial fields and have a crucial impact on the computational accuracy of the zeroing neural network (ZNN) model. For tackling this issue, by combining the dynamics of harmonic signals, two harmonic noise-tolerant ZNN (HNTZNN) models are designed for the dynamic matrix pseudoinversion. In the design of HNTZNN models, an adaptive compensation term is adopted to eliminate the influence of harmonic noises, and a Li activation function is introduced to further improve the convergence rate. The convergence and robustness to harmonic noises of the proposed HNTZNN models are proved through theoretical analyses. Besides, compared with the ZNN model without adaptive compensation term, the HNTZNN models are more effective for tacking the problem of dynamic matrix pseudoinverse under harmonic noises environments. Moreover, HNTZNN models are further applied to the kinematic control of a four-link planar robot manipulator under harmonic noises. In general, the experimental results verify the effectiveness, superiority, and broad application prospect of the models.

Keywords: zeroing neural network, harmonic noise, matrix pseudoinverse, robot manipulator, robustness

OPEN ACCESS

Edited by:
Zhan Li,
Swansea University, United Kingdom

Reviewed by:
Dechao Chen,
Sun Yat-sen University, China
Yinyan Zhang,
Jinan University, China

Correspondence:
Jianfeng Li
ljf_zy@163.com

Received: *26 April 2022*
Accepted: *10 May 2022*
Published: *13 June 2022*

Citation:
Liao B, Wang Y, Li J, Guo D and He Y (2022) Harmonic Noise-Tolerant ZNN for Dynamic Matrix Pseudoinversion and Its Application to Robot Manipulator.
Front. Neurorobot. 16:928636.
doi: 10.3389/fnbot.2022.928636

1. INTRODUCTION

The matrix pseudoinverse (i.e., Moore Penrose generalized inverse) is the generalized inverse of a singular matrix or non-square matrix. Similar to the inverse of the matrix, the matrix pseudoinverse is an important subject in the fields of science and engineering. It is considered a powerful formula and design technology for image reduction (Juang and Wu, 2010), signal processing (Van der Veen et al., 1997), linear classifier (Skurichina and Duin, 2002), and associative memory (Zhang et al., 2004). Due to its importance, numerous efforts have been made in the calculation of matrix pseudoinverse in the past few decades. For instance, Perković and Stanimirović (2011) developed an iterative algorithm to estimate the Moore Penrose generalized inverse. Zhou et al. (2002) presented a sequential recursive formula to calculate the pseudo inverse of the matrix based on the famous Greville formula. Courrieu (2008) proposed an algorithm based on full rank Cholesky decomposition for fast calculation of Moore Penrose inverse matrix. However, most of these serial processing algorithms need to be executed in a single sampling period, and when the system order becomes large, these algorithms may fail. Therefore, these studies are neither feasible nor valid for real-time applications (Zhang and Yi, 2011). While the real-time computing of dynamic matrix pseudoinverse problems often exists in industrial applications. For example, the inverse kinematics control problem of a redundant robot manipulator needs to calculate the pseudoinverse of the dynamic matrix in real time (Liao et al., 2016).

Due to the nature of high-speed parallel, distributed processing and the convenience of hardware implementation, neural networks have been used to replace the numerical algorithm of serial processing, and widely used in the fields of scientific and engineering (Jin et al., 2017; Xu et al., 2019; Jafari, 2020; Liu et al., 2020). Scholars have deeply studied many recurrent neural networks and parallel computing schemes for computing scientific problems including matrix pseudoinverse in real time (Li et al., 2021; Kornilova et al., 2022). In particular, a special kind of recurrent neural network termed zeroing neural network (ZNN), has achieved remarkable success in solving dynamic problems (Zhang et al., 2002; Ding et al., 2017; Xiao et al., 2017; Jin, 2021). Ding et al. (2017) investigated a new neural network based on ZNN to obtain the online solution of complex-valued systems of linear equation in a complex domain with higher precision and higher convergence rate. Xiao et al. (2017) presented a ZNN to solve effectively the unified QP problem and applied it to the coordinated path tracking of dual robot manipulators. Specifically, ZNN successfully solved the lag error generated by traditional recurrent neural networks which uses the indefinite error monitoring function. In addition, the traditional ZNN does not have obvious advantages in real-time calculation of large-scale problems due to its slow convergence speed. Thus, based on the essential nonlinear method of ZNN design, Liao and Zhang (2014b) proposed two finite time convergent ZNN models to solve the real time dynamic matrix pseudoinverse effectively.

In industrial applications, harmonic noises exist widely (Du et al., 2017; Karsli and Dondurur, 2018). In addition, any type of signal can be expressed as the superposition of a sinusoidal signal based on the Fourier transform. However, most of the current studies are aimed at constant noises or linear noises, and there is a lack of an adaptive mechanism to suppress harmonic noises. Furthermore, high frequency and large amplitude harmonic noises will seriously affect the calculation accuracy of the ZNN model. Therefore, by incorporating the dynamics of harmonic signals, we design and use the adaptive compensation term to learn harmonic noises and compensate for the influence of harmonic noises adaptively. Moreover, a well-defined activation function is helpful to accelerate the convergence rate of the ZNN model (Xiao et al., 2018). Therefore, on the basis of harmonic noise adaptation, we employ the Li activation function (Li et al., 2013) to speed up the convergence rate of ZNN and finally design two harmonic noise-tolerant ZNN (HNTZNN) models for solving the dynamic matrix pseudoinverse problem in the harmonic-noise environment. Furthermore, these HNTZNN models are applied to the kinematic control of a four-link planar robot manipulator.

The remainder of this article is organized into six sections. Section 2 introduces the problem formulation and preliminaries. HNTZNN models are developed and studied in Section 3. In Section 4, the convergence and noise suppression ability of HNTZNN are proved theoretically. In Section 5, numerical examples are used to prove the efficacy and superiority of HNTZNN models for calculating the dynamic matrix pseudoinverse. Section 6 shows the application of two ZNN models on a four-link planar robot manipulator. Finally, the conclusion of this article is given in Section 7. The main contributions of this article are as follows.

- In this article, two novel harmonic noise-tolerant ZNN models with a fast convergence rate are first proposed and investigated for computing dynamic matrix pseudoinversion by combining the dynamic properties of harmonic signals and the Li activation function.
- Theoretical analyses are conducted, which deduce the excellent convergence and robustness of these HNTZNN models in coping with single-harmonic and multiple-harmonic noises.
- The experiment results are illustrated, which further substantiate the efficacy and superiority of the proposed HNTZNN models for the dynamic matrix pseudoinversion under low-frequency, high-frequency, periodic, and aperiodic harmonic noises.
- The proposed HNTZNN models are applied to the kinematic control of a four-link planar robot manipulator in the presence of harmonic noise, thereby depicting the application prospect of the proposed models.

2. PROBLEM FORMULATION AND PRELIMINARIES

Definition 1. *(Wang, 1997; Liao and Zhang, 2014a) For a given dynamic matrix $B(t) \in \mathbb{R}^{m\times n}$, its pseudoinverse $Y(t) \in \mathbb{R}^{n\times m}$ satisfies the following Penrose equations:*

$$B(t)Y(t)B(t) = B(t), \quad Y(t)B(t)Y(t) = Y(t),$$
$$B(t)Y(t) = (B(t)Y(t))^{\mathrm{T}}, \quad Y(t)B(t) = (Y(t)B(t))^{\mathrm{T}},$$

where $(\cdot)^{\mathrm{T}}$ denotes the transpose of a matrix, $Y(t)$ is called the dynamic pseudoinverse of $B(t)$, which is often denoted by $B^{\dagger}(t)$.

It is worth noting that the dynamic pseudoinverse $B^{\dagger}(t)$ always exists and is unique (Liao and Zhang, 2014a). In particular, if the matrix $B(t)$ is a full rank matrix at any time instant t, i.e., $\mathrm{rank}[B(t)] = \min\{m,n\}\ \forall t \in [0,\infty)$, the dynamic pseudoinverse of $B(t)$ is obtained by the following lemma.

Lemma 1. *(Wang, 1997; Liao and Zhang, 2014a) For any dynamic matrix $B(t) \in \mathbb{R}^{m\times n}$, if $\mathrm{rank}[B(t)] = \min\{m,n\}\ \forall t \in [0,\infty)$, the unique dynamic pseudoinverse $B^{\dagger}(t)$ can be expressed as:*

$$B^{\dagger}(t) = \begin{cases} \big(B^{\mathrm{T}}(t)B(t)\big)^{-1}B^{\mathrm{T}}(t), & \text{if } m > n, \\ B^{\mathrm{T}}(t)\big(B(t)B^{\mathrm{T}}(t)\big)^{-1}, & \text{if } m \le n. \end{cases}$$

When $m > n$, it represents the left pseudoinverse of the matrix $B(t)$. When $m \le n$, it represents the right pseudoinverse of the matrix $B(t)$. In the case of $m \le n$, the procedure of obtaining the dynamic pseudoinverse $B^{\dagger}(t)$ is similar to that of $m > n$, and thus this paper only considers the case of the left pseudoinverse of the matrix $B(t)$. Therefore, the dynamic pseudoinverse problem studied in this study is in the form of

$$B^{\mathrm{T}}(t)B(t)Y(t) = B^{\mathrm{T}}(t) \in \mathbb{R}^{m\times n}. \tag{1}$$

Our objective in this study is to find the $Y(t)$ of the problem (1).

3. HARMONIC NOISE-TOLERANT ZNN MODELS

In this section, the improved ZNN models against harmonic noises for dynamic pseudoinverse are detailed.

3.1. ZNN Model Against Single-Harmonic Noise

In order to monitor the calculation process of dynamic matrix pseudoinverse (1) in real time, the error function is defined as

$$E(t) = B^{\mathrm{T}}(t)B(t)Y(t) - B^{\mathrm{T}}(t).$$

To force $E(t)$ to converge to zero, we choose

$$\dot{E}(t) = \frac{\mathrm{d}E(t)}{\mathrm{d}t} = -\tau\Psi\big(E(t)\big), \tag{2}$$

where $\tau > 0 \in \mathbb{R}$ is a positive design parameter, and $\Psi(\cdot)$ denotes the activation function of the neural network, which is a monotonically increasing odd function. Then,

$$\begin{aligned}\dot{E}(t) =& B^{\mathrm{T}}(t)B(t)\dot{Y}(t)\\ &+ \big(\dot{B}^{\mathrm{T}}(t)B(t) + B^{\mathrm{T}}(t)\dot{B}(t)\big)Y(t) - \dot{B}^{\mathrm{T}}(t).\end{aligned} \tag{3}$$

In this article, we choose the Li activation function (Li et al., 2013), and its expression is as follows:

$$\psi(e_{ij}) = \mathrm{Lip}^{\lambda}(e_{ij}) + \mathrm{Lip}^{1/\lambda}(e_{ij}), \tag{4}$$

where e_{ij} represents the ijth element of $E(t)$, parameter $\lambda \in (0, 1)$, and the function $\mathrm{Lip}^{\lambda}(\cdot)$ is defined as follows:

$$\mathrm{Lip}^{\lambda}(e_{ij}) = \begin{cases} |e_{ij}|^{\lambda}, & \text{if } e_{ij} > 0, \\ 0, & \text{if } e_{ij} = 0, \\ -|e_{ij}|^{\lambda}, & \text{if } e_{ij} < 0. \end{cases} \tag{5}$$

In view of the noise exiting, formula (2) can be rewritten as

$$\dot{E}(t) = -\tau\Psi\big(E(t)\big) + O(t), \tag{6}$$

where $O(t) \in \mathbb{R}^{m\times n}$ represents the matrix-form harmonic noises and each element is defined as follows:

$$o_{ij}(t) = \Gamma\sin(\omega t + \varphi) = \Gamma\sin(2\pi f t + \varphi), \tag{7}$$

where $i \in \{1, \cdots, m\}$, $j \in \{1, \cdots, n\}$. It is assumed that the frequency f in (7) is known, the amplitude Γ and phase φ are unknown. To adaptively learn the f-frequency harmonics noise, an additional quantity $-X(t) \in \mathbb{R}^{m\times n}$ is introduced into equation (6):

$$\dot{E}(t) = -\tau\Psi\big(E(t)\big) - X(t) + O(t). \tag{8}$$

The purpose of this study is to design and use $-X(t)$ to adaptively learn $O(t)$ and compensate for the impact of $O(t)$. Differentiating (7) twice with respect to time t, the following result is obtained:

$$\ddot{o}_{ij}(t) = -4\pi^2 f^2\Gamma\sin(2\pi f t + \varphi) = -4\pi^2 f^2 o_{ij}(t).$$

Thus, the dynamics of $o_{ij}(t)$ can be formulated as follows:

$$\dot{o}_{ij}(t) = n_{ij}(t), \dot{n}_{ij}(t) = -4\pi^2 f^2 o_{ij}(t). \tag{9}$$

Rewrite formula (9) into matrix form:

$$\dot{O}(t) = N(t), \dot{N}(t) = -4\pi^2 f^2 O(t), \tag{10}$$

where $N(t) \in \mathbb{R}^{m\times n}$ with $n_{ij}(t)$ as its element. The unknown amplitude and phase information of a harmonic signal can be adaptively eliminated from (10) when the frequency f is known.

Therefore, combing formulas (3), (8), and (10), the following single-harmonic noise-tolerant ZNN (HNTZNN) model is obtained:

$$\begin{cases} B^{\mathrm{T}}(t)B(t)\dot{Y}(t) = \dot{B}^{\mathrm{T}}(t) - \big(\dot{B}^{\mathrm{T}}(t)B(t) + B^{\mathrm{T}}(t)\dot{B}(t)\big)Y(t) \\ -\tau\Psi\big(B^{\mathrm{T}}(t)B(t)Y(t) - B^{\mathrm{T}}(t)\big) - X(t) + O(t), \\ \dot{X}(t) = C(t) + 4\pi^2 f^2\lambda E(t), \\ \dot{C}(t) = -4\pi^2 f^2 X(t), \end{cases} \tag{11}$$

where $\lambda \geq 1 \in \mathbb{R}$.

3.2. ZNN Model Against Multiple-Harmonic Noise

In this section, the proposed HNTZNN model (11) is extended to multiple-harmonic noises, and a neural network model against multiple-harmonic noises is obtained. Let $\tilde{O}(t) \in \mathbb{R}^{m\times n}$ be the matrix-form multiple-harmonic noises. Each element of such matrix is defined as

$$\tilde{o}_{ij}(t) = \sum_{k=1}^{l} o_k(t) = \sum_{k=1}^{l}\Gamma_k\sin(2\pi f_k t + \varphi_k). \tag{12}$$

Similarly, in (12), it is assumed that the frequency f_k (with $k \in 1, \cdots, l$) is known, the amplitude Γ_k and phase φ_k are unknown.

Differentiating (12) twice with respect to time t, we have the following results:

$$\begin{aligned}\dot{\tilde{o}}(t) &= \sum_{k=1}^{l}\ddot{o}_k(t)\\ &= \sum_{k=1}^{l} -4\pi^2 f_k^2\Gamma_k\sin(2\pi f_k t + \varphi_k)\\ &= \sum_{k=1}^{l} -4\pi^2 f_k^2 o_k(t).\end{aligned}$$

The dynamics of $\tilde{o}_i j(t)$ are obtained as follows:

$$\dot{\tilde{o}}_{ij}(t) = \sum_{k=1}^{l} n_k(t), \dot{o}_k(t) = n_k(t), \dot{n}_k(t) = -4\pi^2 f_k^2 o_k(t),$$

which is rewritten in the matrix form as:

$$\dot{\tilde{O}}(t) = \sum_{k=1}^{l} N_k(t), \dot{O}_k(t) = N_k(t), \dot{N}_k(t) = -4\pi^2 f_k{}^2 N_k(t), \tag{13}$$

where $N_k(t) \in \mathbb{R}^{m\times n}$ and $O_k(t) \in \mathbb{R}^{m\times n}$ are the matrices composed of $n_k(t)$ and $o_k(t)$, respectively.

Combined with the dynamic analysis in Formula (13), a multiple-harmonic noise-tolerant ZNN model is as follows:

$$\begin{cases} B^{\mathrm{T}}(t)B(t)\dot{Y}(t) = \dot{B}^{\mathrm{T}}(t) - \big(\dot{B}^{\mathrm{T}}(t)B(t) + B^{\mathrm{T}}(t)\dot{B}(t)\big)Y(t) \\ -\tau\Psi\big(B^{\mathrm{T}}(t)B(t)Y(t) - B^{\mathrm{T}}(t)\big) - \sum_{k=1}^{l} X_k(t) + \tilde{O}(t), \\ \dot{X}_k(t) = C_k(t) + 4\pi^2 f_k^2\lambda E(t), \\ \dot{C}_k(t) = -4\pi^2 f_k^2 X_k(t), k = 1, 2, \cdots, l. \end{cases} \tag{14}$$

When $l = 1$, (14) can be reduced to (11).

The corresponding theoretical analysis and proof are carried out in the next section.

4. THEORETICAL ANALYSIS

Three theorems are proposed to verify the convergence and robustness performance of the proposed HNTZNN model.

Theorem 1. *For any smooth dynamic full rank matrix $B(t) \in \mathbb{R}^{m\times n}$ ($m > n$), the state matrix $Y(t)$ of HNTZNN model (11) globally converges to the dynamic theoretical pseudoinverse $B^{\dagger}(t)$ of (1) in the free-noise situation.*

PROOF: In the free-noise situation, (11) can be simplified as

$$\begin{cases} c\dot{E}(t) = -\tau\Psi\big(E(t)\big) - X(t), \\ \dot{X}(t) = C(t) + 4\pi^2 f^2\lambda E(t), \\ \dot{C}(t) = -4\pi^2 f^2 X(t), \end{cases} \tag{15}$$

where $E(t) = B^{\mathrm{T}}(t)B(t)Y(t) - B^{\mathrm{T}}(t)$, (15) is a compact matrix form of the following set of mn-decoupled equations:

$$\begin{cases} c\dot{e}_{ij}(t) = -\tau\psi\big(e_{ij}(t)\big) - x_{ij}(t), \\ \dot{x}_{ij}(t) = c_{ij}(t) + 4\pi^2 f^2\lambda e_{ij}(t), \\ \dot{c}_{ij}(t) = -4\pi^2 f^2 x_{ij}(t). \end{cases} \tag{16}$$

Then, we could define the Lyapunov function candidate (Xiang et al., 2019) to analyze the ijth subsystem (16) as

$$v_{ij}(t) = \frac{e_{ij}^2(t)}{2} + \frac{x_{ij}^2(t)}{8\pi^2 f^2\lambda} + \frac{c_{ij}^2(t)}{32\pi^4 f^4\lambda}, \tag{17}$$

which guarantees $v_{ij}(t) \geq 0$, i.e., $v_{ij}(t) > 0$ for any $e_{ij}^2(t) \neq 0$ or $x_{ij}^2(t) \neq 0$ or $c_{ij}^2(t) \neq 0$, and $v_{ij}(t) = 0$ if and only if $e_{ij}^2(t) = 0$, $x_{ij}^2(t) = 0$, $c_{ij}^2(t) = 0$. The time derivative of (17) could be obtained as

$$\begin{aligned} \frac{\mathrm{d}v_{ij}(t)}{\mathrm{d}t} =& e_{ij}(t)\dot{e}_{ij}(t) + \frac{x_{ij}(t)\dot{x}_{ij}(t)}{4\pi^2 f^2\lambda} + \frac{c_{ij}(t)\dot{c}_{ij}(t)}{16\pi^4 f^4\lambda}, \\ =& -\tau e_{ij}(t)\psi\big(e_{ij}(t)\big) - e_{ij}(t)x_{ij}(t) + \frac{x_{ij}(t)c_{ij}(t)}{4\pi^2 f^2\lambda} \\ & + x_{ij}(t)e_{ij}(t) - \frac{c_{ij}(t)x_{ij}(t)}{4\pi^2 f^2\lambda}, \\ =& -\tau e_{ij}(t)\psi\big(e_{ij}(t)\big). \end{aligned}$$

Substituting (5) into the above equation, we further have

$$\frac{\mathrm{d}v_{ij}(t)}{\mathrm{d}t} = \begin{cases} -\tau|e_{ij}(t)|^{\lambda+1}, & \text{if } e_{ij} \neq 0, \\ 0, & \text{if } e_{ij} = 0. \end{cases}$$

Thus, when $e_{ij}(t) \neq 0$, $\dot{v}_{ij}(t) < 0$, when $e_{ij}(t) = 0$, $\dot{v}_{ij}(t) = 0$. According to Lyapunov stability theory (Khalil, 2001), it can be concluded that $e_{ij}(t)$ globally to 0 for any $i \in \{1, \cdots, m\}$, $j \in \{1, \cdots, n\}$, which means that

$$\lim_{t\to\infty} \|E(t)\|_{\mathrm{F}} = 0.$$

In view of $E(t) = B^{\mathrm{T}}(t)B(t)Y(t) - B^{\mathrm{T}}(t)$ and the nonsingularity of $B^{\mathrm{T}}(t)B(t)$, we can deduce $Y(t) \to \big(B^{\mathrm{T}}(t)B(t)\big)^{-1}B^{\mathrm{T}}(t)$ as $t \to \infty$. Therefore, the state matrix $Y(t)$ of HNTZNN model (11) globally converges to the dynamic theoretical pseudoinverse $B^{\dagger}(t)$ of (1). The proof is, thus, completed.

Theorem 2. *For any smooth dynamic full rank matrix $B(t) \in \mathbb{R}^{m\times n}$ ($m > n$), the state matrix $Y(t)$ of HNTZNN model (11) globally converges to the dynamic theoretical pseudoinverse $B^{\dagger}(t)$ of (1) when the single-harmonic noise is considered.*

PROOF: Let us consider a single harmonic noise $O(t) = [o_{ij}(t)] = [\Gamma\sin(2\pi ft + \varphi)]$ with known frequency, and the unknown amplitude and phase. Model (11) is simplified and rewritten as

$$\begin{cases} c\dot{E}(t) = -\tau\Psi\big(E(t)\big) - X(t) + O(t), \\ \dot{X}(t) = C(t) + 4\pi^2 f^2\lambda E(t), \\ \dot{C}(t) = -4\pi^2 f^2 X(t), \\ \dot{O}(t) = N(t), \\ \dot{N}(t) = -4\pi^2 f^2 O(t). \end{cases} \tag{18}$$

Defining $\Delta(t) = X(t) - O(t)$ and $\Phi(t) = C(t) - N(t)$ yields the following reformulation of (18):

$$\begin{cases} \dot{E}(t) = -\tau\Psi\big(E(t)\big) - \Delta(t), \\ \dot{\Delta}(t) = \Phi(t) + 4\pi^2 f^2\lambda E(t), \\ \dot{\Phi}(t) = -4\pi^2 f^2\Delta(t). \end{cases} \tag{19}$$

Noting that Equations (15) and (19) have exactly the same form, therefore, one can also obtain

$$\lim_{t\to\infty} e_{ij}(t) = 0 \text{ and } \lim_{t\to\infty} \|E(t)\|_{\mathrm{F}} = 0,$$

for any $i \in \{1, \cdots, m\}$, $j \in \{1, \cdots, n\}$. In addition, according to (16), we have

$$\begin{cases} c\dot{x}_{ij}(t) = c_{ij}(t) + 4\pi^2 f^2\lambda e_{ij}(t) \to c_{ij}(t), \\ \dot{c}_{ij}(t) = -4\pi^2 f^2 x_{ij}(t). \end{cases} \tag{20}$$

According to the dynamic description in (9), Equation (20) can generate a harmonic signal $x_{ij}(t)$ that can adaptively compensate for the influence of noise over time. Then, Equation (18) is

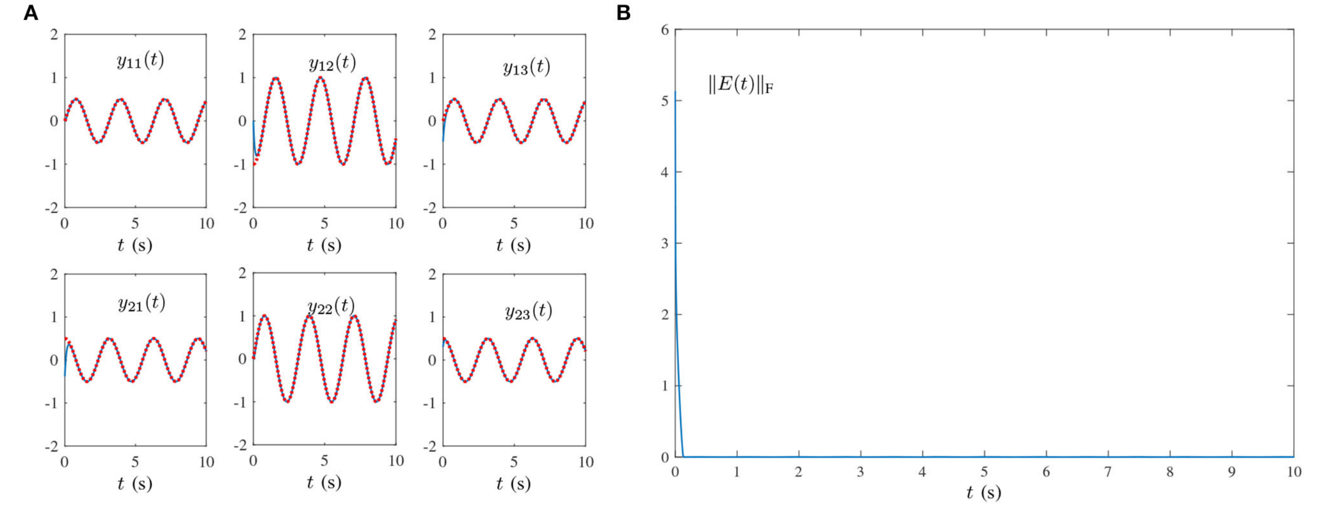


FIGURE 1 | Results synthesized by the HNTZNN model (11) using $\tau = 10$ and $\lambda = 1$ for computing the pseudoinverse of the dynamic matrix (21) under the single-harmonic noise with frequency $f = 0.001$ Hz. **(A)** State trajectory. **(B)** Residual error.

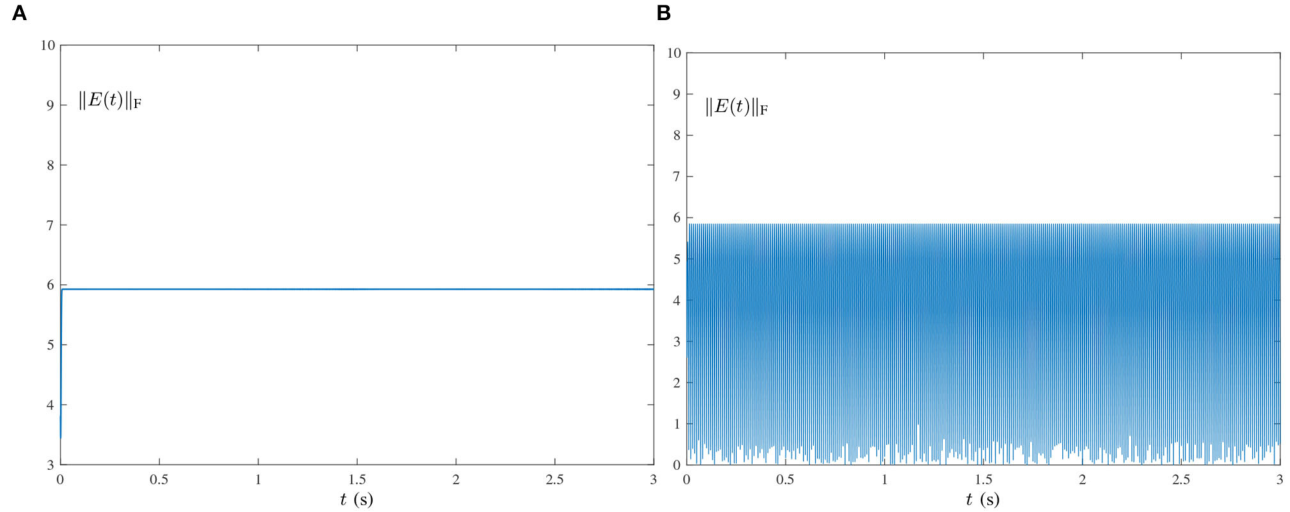


FIGURE 2 | Residual errors of ZNN-1 model (24) for computing the pseudoinverse of the dynamic matrix (21) under the single-harmonic noise with different frequencies. **(A)** $f = 0.001$ Hz. **(B)** $f = 50$ Hz.

a compact matrix form of the following set of mn-decoupled equations:

$$\begin{cases} c\dot{e}_{ij}(t) = -\tau\psi\left(e_{ij}(t)\right) - x_{ij}(t) + o_{ij}(t), \\ \dot{x}_{ij}(t) = c_{ij}(t) + 4\pi^2 f^2 \lambda e_{ij}(t), \\ \dot{c}_{ij}(t) = -4\pi^2 f^2 x_{ij}(t). \end{cases} \tag{21}$$

Because of $\lim_{t\to\infty} e_{ij}(t) = 0$, the harmonic signal $x_{ij}(t)$ is generated based on the analysis of (20), which meets the following results:

$$-x_{ij}(t) + o_{ij}(t) \to 0, \quad \text{as } t \to \infty.$$

In particular, although only the frequency of $o_{ij}(t)$ is known, the second and third dynamics in (21) can adaptively predict the unknown parameters of $o_{ij}(t)$, i.e., amplitude Γ and phase φ. The resultant signal $x_{ij}(t)$ can adaptively compensate the impact of $o_{ij}(t)$.

In summary, the state matrix $Y(t)$ of the HNTZNN model (11) globally converges to the dynamic theoretical pseudoinverse $B^{\dagger}(t)$ of (1) when the single-harmonic noise is considered. The proof is, thus, completed.

Theorem 3. *For any smooth dynamic full rank matrix $B(t) \in \mathbb{R}^{m\times n}$ ($m > n$), the state matrix $Y(t)$ of the HNTZNN model (14) globally converges to the dynamic theoretical pseudoinverse*

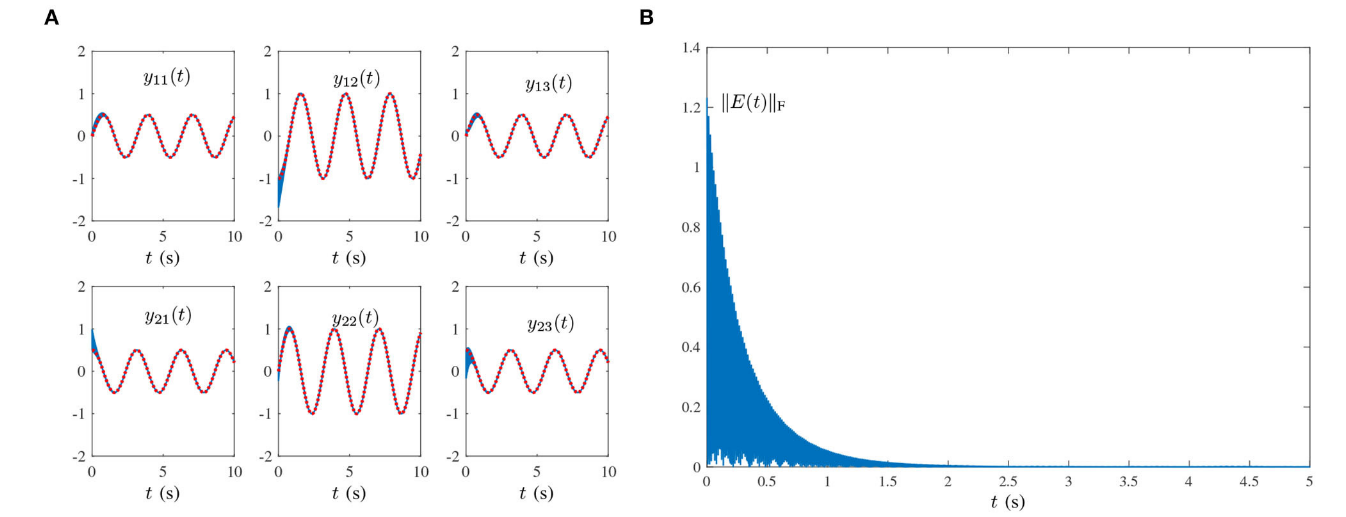


FIGURE 3 | Results synthesized by the HNTZNN model (11) using $\tau = 10$ and $\lambda = 1$ for computing the pseudoinverse of the dynamic matrix (21) under the single-harmonic noise with frequency $f = 50$ Hz. **(A)** State trajectory. **(B)** Residual error.

$B^\dagger(t)$ of (1) when the multiple-harmonic noise is considered, i.e., $\tilde{O}(t) = [o_{ij}(t)] = \left[\sum_{k=1}^{l}\Gamma_k \sin(2\pi f_k t + \varphi_k)\right]$, $i \in \{1, \cdots, m\}$, $j \in \{1, \cdots, n\}$.

PROOF: This theorem can be obtained through the proof of the above two theorems and the superposition principle. Therefore, it is omitted. The proof is, thus, completed.

Remarks: Generally, for any periodic noises which can be decomposed into a series of harmonics through Fourier transform, the proposed model is effective. The limitation of the current work may lie in the suppression of random noises.

5. NUMERICAL VERIFICATIONS

Previously, we analyzed the convergence and noise tolerance of models (11) and (14). In this section, we verify the efficacy and the superiority of the proposed models for solving the following dynamic matrix pseudoinverse:

$$B(t) = \begin{bmatrix} \sin(2t) & \cos(2t) \\ -\cos(2t) & \sin(2t) \\ \sin(2t) & \cos(2t) \end{bmatrix} \in \mathbb{R}^{3\times 2}. \tag{22}$$

To check the correctness of the presented HNTZNN models, the theoretical dynamic matrix pseudoinverse of (22) is directly given as

$$B^\dagger(t) = \begin{bmatrix} 0.5\sin(2t) & -\cos(2t) & 0.5\sin(2t) \\ 0.5\cos(2t) & \sin(2t) & 0.5\cos(2t) \end{bmatrix} \in \mathbb{R}^{2\times 3}. \tag{23}$$

In order to verify the superiority of the proposed model, the ZNN-1 model (Liao and Zhang, 2014a) for dynamic matrix pseudoinverse is directly given in this section:

$$\begin{aligned} B^{\mathrm{T}}(t)B(t)\dot{Y}(t) =& \dot{B}^{\mathrm{T}}(t) - \left(\dot{B}^{\mathrm{T}}(t)B(t) + B^{\mathrm{T}}(t)\dot{B}(t)\right)Y(t) \\ & - \tau\Psi\left(B^{\mathrm{T}}(t)B(t)Y(t) - B^{\mathrm{T}}(t)\right). \end{aligned} \tag{24}$$

The following experiments are carried out to solve the pseudoinverse of the dynamic matrix (22) in the case of four kinds of harmonic noise.

1) Single-harmonic noise with a low-frequency: **Figures 1**, **2** present the experimental result synthesized by HNTZNN model (11) and ZNN-1 model (24) for the dynamic matrix (21) pseudoinverse under the single-harmonic noise with a near-zero frequency, i.e.,

$$O(t) = [o_{ij}(t)] = [10^3 \sin(0.002\pi t + 1)] \in \mathbb{R}^{2\times 3},$$

where frequency $f = 0.002\pi/(2\pi) = 0.001$Hz. In **Figure 1A**, the red dashed-dotted line denotes the theoretical dynamic solution, and the blue solid line denotes the neural-state solution. In addition, we set $\tau = 10$ and $\lambda = 1$. As shown in **Figure 1A**, for a random initial value $Y(0) \in \mathbb{R}^{2\times 3}$, state matrix $Y(t) \in \mathbb{R}^{2\times 3}$ of the proposed HNTZNN model (11) converges to the theoretical pseudoinverse (23) accurately and rapidly in a short period of time. **Figure 1B** shows that the residual error $\|E(t)\|_F$ of the HNTZNN model (11) can converge to zero within 1 s. However, with the same parameters, the residual error $\|E(t)\|_F$ of the ZNN-1 model (24) can not converge to zero displayed in **Figure 2A**. These experimental results verify the efficacy and superiority of model (11) for solving the dynamic matrix pseudoinverse under the single-harmonic noise.

2) Single-harmonic noise with high-frequency: HNTZNN model (11) and ZNN-1 model (24) are tested for the following harmonic noise:

$$O(t) = [o_{ij}(t)] = [10^3 \sin(100\pi t + 1)] \in \mathbb{R}^{2\times 3},$$

where frequency $f = 50$ Hz, which is widely encountered in power systems (Karsli and Dondurur, 2018). The

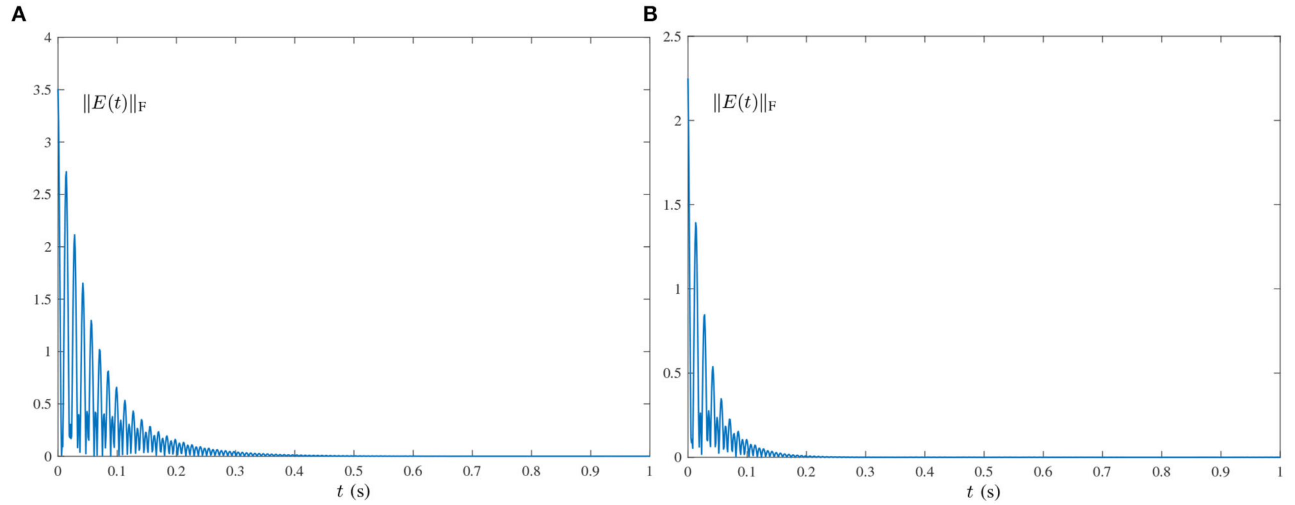


FIGURE 4 | Residual errors of HNTZNN model (11) using different τ for computing the pseudoinverse of the dynamic matrix (21) under the single-harmonic noise with frequency $f = 50$ Hz. **(A)** τ = 50, λ = 1. **(B)** τ = 100, λ = 1.

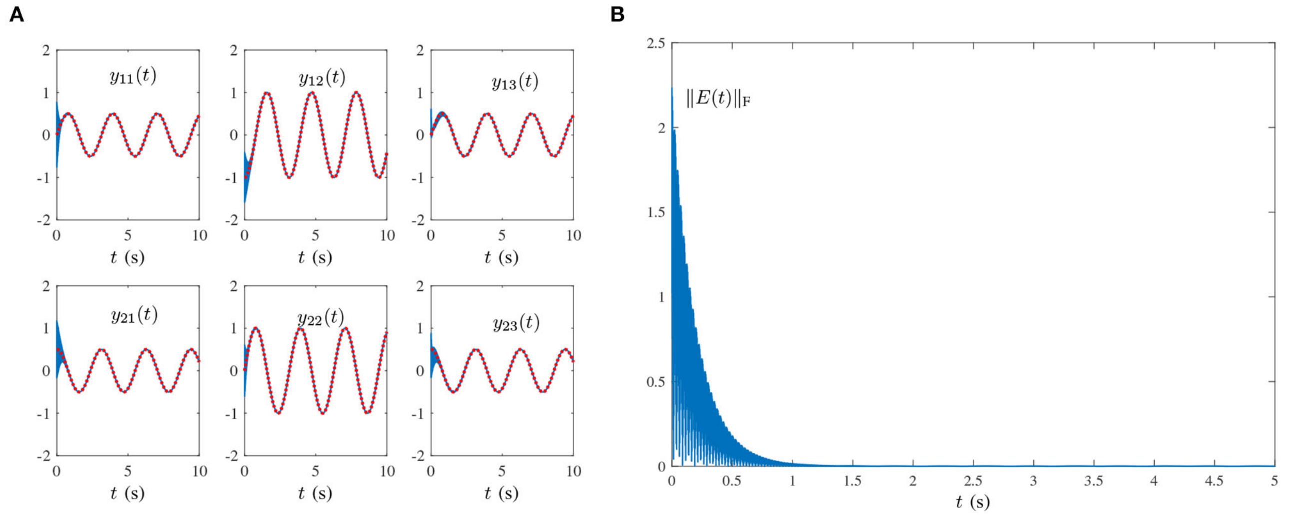


FIGURE 5 | Results synthesized by the HNTZNN model (14) using $\tau = 10$ and $\lambda = 1$ for computing the pseudoinverse of the dynamic matrix (21) under the multiple-harmonic noise with a periodic characteristic. **(A)** State trajectory. **(B)** Residual error.

corresponding results are shown in **Figures 2**, **3**. As shown in **Figure 3A**, the state matrix $Y(t) \in \mathbb{R}^{2\times 3}$ of the proposed HNTZNN model (11) still can converge to the dynamic theoretical pseudoinverse (23) accurately and rapidly. **Figure 3B** shows that the residual error $\|E(t)\|_F$ of HNTZNN model (11) can converge to zero rapidly. On the contrary, **Figure 2B** shows that the residual error $\|E(t)\|_F$ of ZNN-1 model (24) has a large value under harmonic noise and cannot converge to zero.

For further investigation, let $\lambda = 1$ and vary the value of τ (i.e., let $\tau = 50$ and $\tau = 100$). The experimental results are shown in **Figure 4**. It can be seen from **Figures 3B**, **4** that the larger value of τ, the faster convergence rate of HNTZNN model (11).

3) Multiple-harmonic noise with a periodic characteristic: HNTZNN model (14) is tested for the following harmonic noise:

$$\tilde{O}(t) = [\tilde{o}_{ij}(t)]$$
$$= [\sin(2\pi t + 1) + 2\sin(4\pi t + 2) + 5\sin(10\pi t + 4)] \in \mathbb{R}^{2\times 3},$$

where frequencies are denoted by $f_1 = 1$, $f_2 = 2$, $f_3 = 5$ Hz. As shown in **Figure 5**, the state matrix $Y(t)$ of proposed HNTZNN model (14) converges to the dynamic theoretical pseudoinverse (23), and the residual error $\|E(t)\|_F$ of HNTZNN model (11) is convergent to zero.

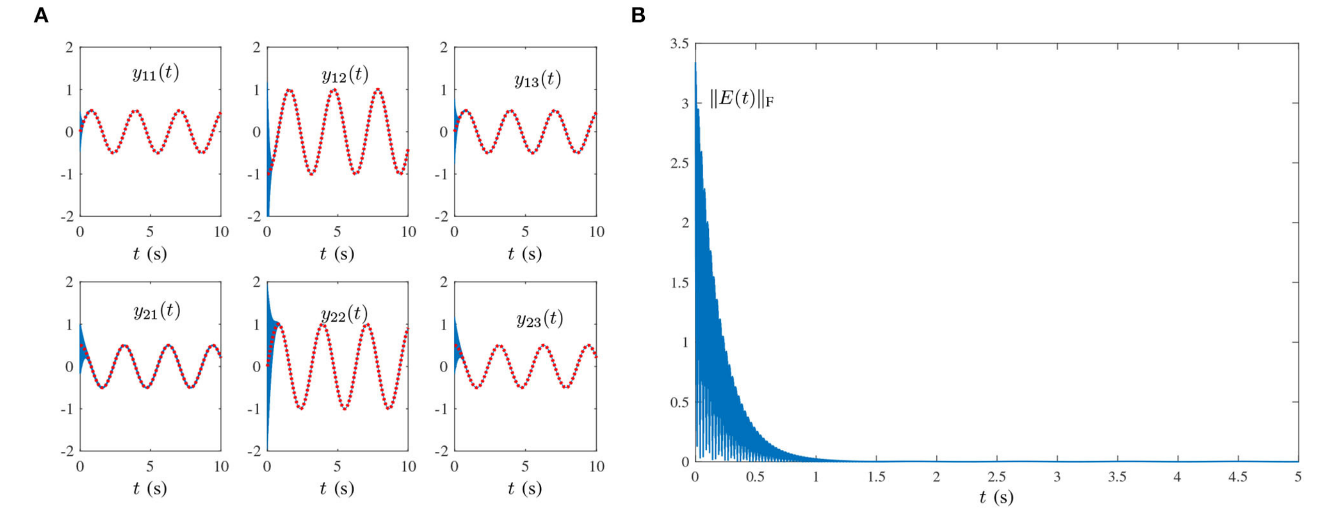


FIGURE 6 | Results synthesized by the HNTZNN model (14) using $\tau = 10$ and $\lambda = 1$ for computing the pseudoinverse of the dynamic matrix (21) under the multiple-harmonic noise with an aperiodic characteristic. **(A)** State trajectory. **(B)** Residual error.

4) Multiple-harmonic noise with an aperiodic characteristic: HNTZNN model (14) is tested for the following harmonic noise:

$$\begin{aligned}\tilde{O}(t) &= [\tilde{o}_{ij}(t)]\\ &= [\sin(1) + 2\sin(5t+2) + 5\sin(10\pi t+3)] \in \mathbb{R}^{2\times 3},\end{aligned}$$

where frequencies are denoted by $f_1 = 0\mathrm{Hz}$, $f_2 = 2.5/\pi\,\mathrm{Hz}$, $f_3 = 5\mathrm{Hz}$. The corresponding results are demonstrated in **Figure 6**. As shown in **Figure 6**, the state matrix $Y(t)$ of proposed HNTZNN model (14) converges to the dynamic theoretical pseudoinverse (23), and the residual error $\|E(t)\|_\mathrm{F}$ of HNTZNN model (11) is convergent to zero.

In summary, the experimental results verify the effectiveness and superiority of HNTZNN models (11) and (14) for solving the dynamic matrix pseudoinverse in the presence of harmonic noises.

6. ROBOT MANIPULATOR APPLICATION

In this section, HNTZNN models (11) and (14) are applied to the kinematic control of a four-link planar robot manipulator under the presence of harmonic noises.

6.1. ZNN-Combined Kinematic Control

The geometry of the four-link planar robot manipulator is shown in Guo et al. (2018a). The Jacobian matrix $J(\theta(t)) \in \mathbb{R}^{2\times 4}$ of this robot manipulator is:

$$J(\theta(t)) = \begin{bmatrix} -\sum_{i=1}^{4} l_i s_i & -\sum_{i=2}^{4} l_i s_i & -\sum_{i=3}^{4} l_i s_i & -l_i s_i \\ -\sum_{i=1}^{4} l_i c_i & -\sum_{i=2}^{4} l_i c_i & -\sum_{i=3}^{4} l_i c_i & -l_i c_i \end{bmatrix},$$

where $\theta(t) \in \mathbb{R}^4$ denotes the joint-angle vector, l_i (with $i = 1, 2, 3, 4$) denotes the length of the ith link, $s_i = \sin(\sum_{j=1}^{i} \theta_j)$, and $c_i = \cos(\sum_{j=1}^{i} \theta_j)$. The redundancy-resolution problem of such a robot manipulator, which corresponds to the kinematic control, is described as follows. Given the desired end-effector path $r_\mathrm{d}(t) \in \mathbb{R}^2$, we need to obtain $\theta(t)$ in real time t. In particular, the following acceleration-level redundancy-resolution problem is formulated by adopting the solutions presented in previous work (Siciliano et al., 2009; Guo et al., 2018b):

$$J(t)\ddot{\theta}(t) = \ddot{r}_\mathrm{a}(t), \tag{25}$$

where $\ddot{\theta}(t) \in \mathbb{R}^4$ is the joint-acceleration vector, $r_\mathrm{a}(t) \in \mathbb{R}^2$ denotes actual end-effector path and $\ddot{r}_\mathrm{a}(t) = \ddot{r}_\mathrm{d}(t) - \dot{J}(t)\dot{\theta}(t) + k_1(\dot{r}_\mathrm{d}(t) - J(t)\dot{\theta}(t)) + k_2(r_\mathrm{d}(t) - \psi(\theta(t)))$ with $\dot{\theta}(t) \in \mathbb{R}^4$ as the joint-velocity vector and $\dot{J}(t)$ as the time derivative of the Jacobian matrix $J(t)$. In addition, k_1 and $k_2 > 0$ are the feedback gains, $\psi(\cdot)$ is a differentiable nonlinear mapping (Siciliano et al., 2009; Guo et al., 2018b), and $\dot{r}_\mathrm{d}(t)$ and $\ddot{r}_\mathrm{d}(t)$ are first-order and second-order time derivatives of $r_\mathrm{d}(t)$, respectively. By effectively solving the acceleration-level kinematic equation in (25), the purpose of kinematic control for robot manipulator is achieved.

According to Siciliano et al. (2009), the following pseudoinverse-type (P-type) scheme, which is a typical solution to (25), is presented for the four-link planar robot manipulator:

$$\ddot{\theta}(t) = J^{\dagger}(t)\ddot{r}_\mathrm{a}(t) = (J^\mathrm{T}J(t))^{-1}J^\mathrm{T}(t)\ddot{r}_\mathrm{a}(t). \tag{26}$$

The dynamic matrix inversion is integrated into (26) to achieve the kinematic control of the four-link planar robot manipulator. By defining $B(t) = J(t)$, combining (11) with (26) yields the following the new ZNN-combined kinematic control method

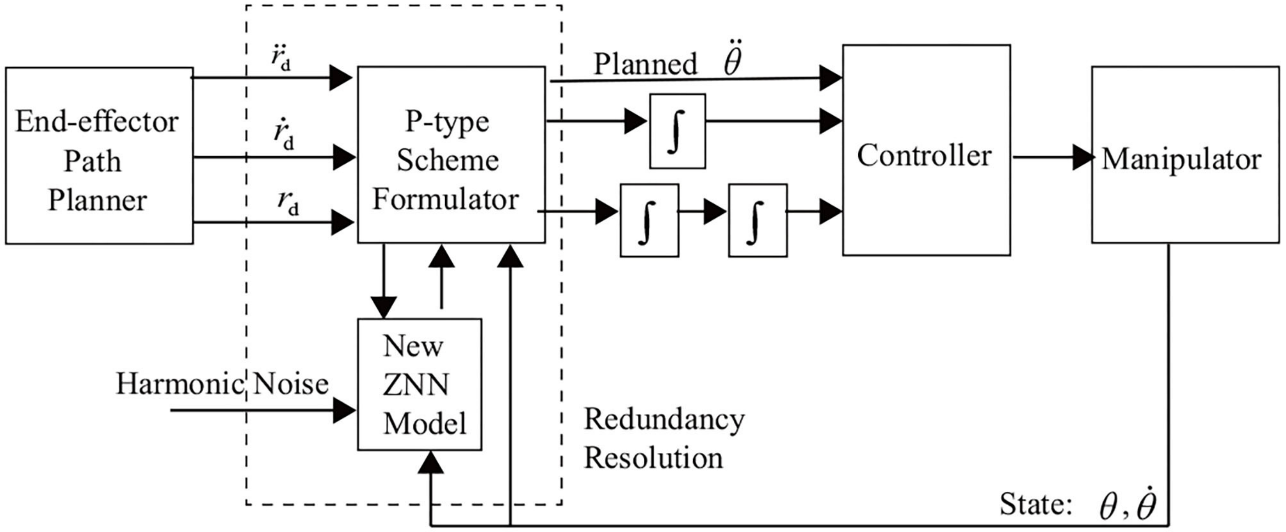


FIGURE 7 | Block diagram of the new ZNN-combined kinematic control method for the four-link planar root manipulator.

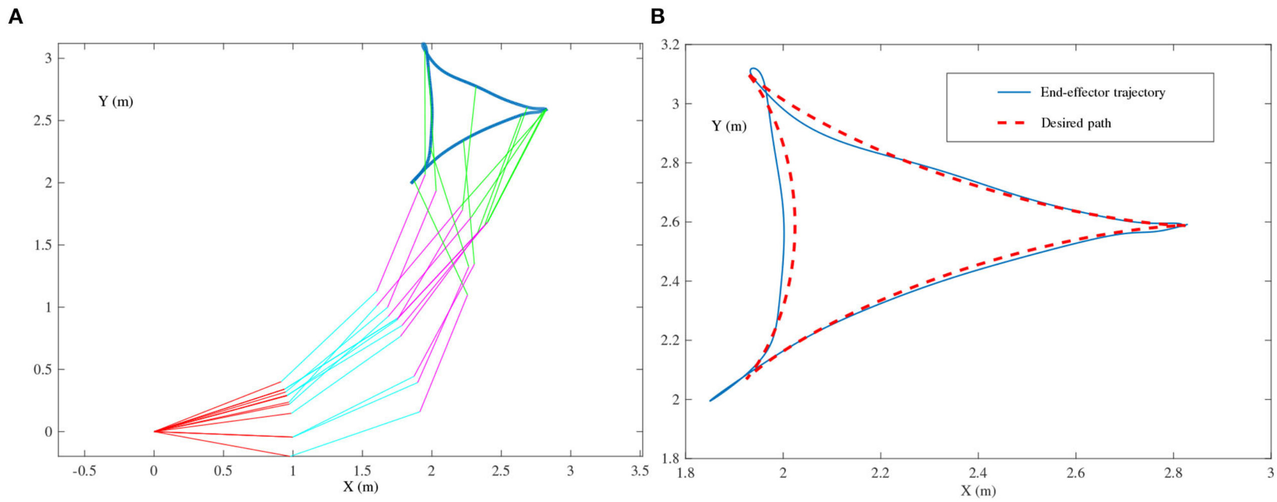


FIGURE 8 | End-effector of the four-link planar robot manipulator tracking the tricuspid path synthesized by the noise-polluted kinematic control method in (29) using $\tau = 10$, where single-harmonic noise is considered. **(A)** Simulated motion trajectories. **(B)** Path and trajectory.

that is tolerant to single-harmonic noise:

$$\begin{cases}\ddot{\theta}(t) = B^{\dagger}(t)\ddot{r}_{\mathrm{a}}(t),\\ B^{\mathrm{T}}(t)B(t)\dot{Y}(t) = \dot{B}^{\mathrm{T}}(t) - \big(\dot{B}^{\mathrm{T}}(t)B(t) + B^{\mathrm{T}}(t)\dot{B}(t)\big)Y(t)\\ -\tau\Psi\big(B^{\mathrm{T}}(t)B(t)Y(t) - B^{\mathrm{T}}(t)\big) - X(t) + O(t),\\ \dot{X}(t) = C(t) + 4\pi^2 f^2\lambda E(t),\\ \dot{C}(t) = -4\pi^2 f^2 X(t).\end{cases} \tag{27}$$

Furthermore, by combining (14) with (26), the new ZNN-combined kinematic control method for multiple-harmonic noise can be obtained as follows:

$$\begin{cases}\ddot{\theta}(t) = B^{\dagger}(t)\ddot{r}_{\mathrm{a}}(t),\\ B^{\mathrm{T}}(t)B(t)\dot{Y}(t) = \dot{B}^{\mathrm{T}}(t) - \big(\dot{B}^{\mathrm{T}}(t)B(t) + B^{\mathrm{T}}(t)\dot{B}(t)\big)Y(t)\\ -\tau\Psi\big(B^{\mathrm{T}}(t)B(t)Y(t) - B^{\mathrm{T}}(t)\big) - \sum_{k=1}^{l} X_k(t) + \tilde{O}(t),\\ \dot{X}_k(t) = C_k(t) + 4\pi^2 f_k^2\lambda E(t),\\ \dot{C}_k(t) = -4\pi^2 f_k^2 X_k(t), k = 1, 2, \cdots, l.\end{cases} \tag{28}$$

To provide a better understanding, we show **Figure 7** to depict the block diagram of the ZNN-combined kinematic control method in (28) [including (27) as a special case] for the four-link planar robot manipulator. The robot application

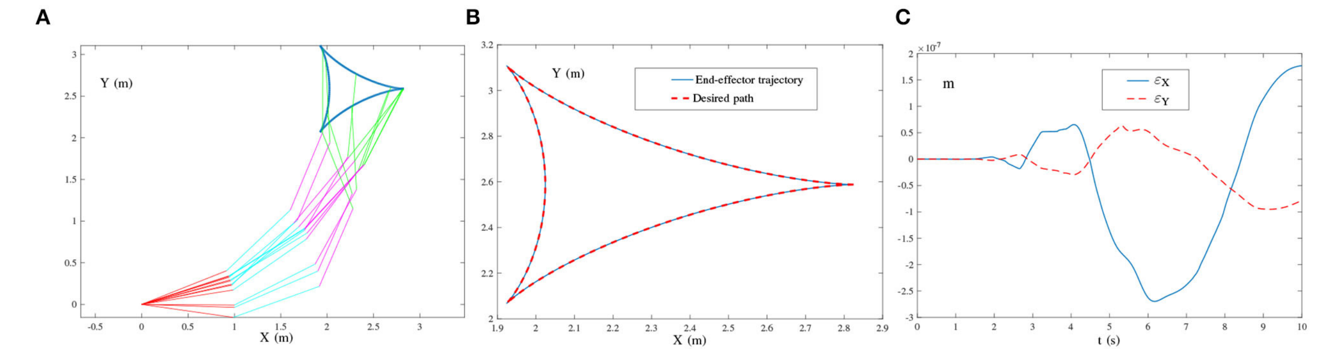


FIGURE 9 | End-effector of the four-link planar robot manipulator tracking the tricuspid path synthesized by the ZNN-combined kinematic control method in (27) using $\tau = 10$ and $\lambda = 1$, where single-harmonic noise is considered. **(A)** Simulated motion trajectories. **(B)** Path and trajectory. **(C)** End-effector positioning errors.

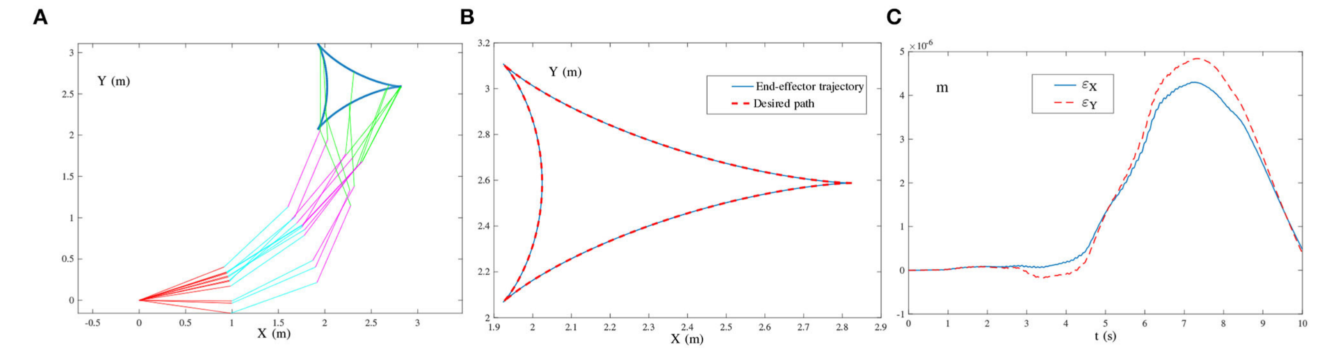


FIGURE 10 | End-effector of the four-link planar robot manipulator tracking the tricuspid path synthesized by the ZNN-combined kinematic control method in (28) using $\tau = 10$ and $\lambda = 1$, where multiple-harmonic noise is considered. **(A)** Simulated motion trajectories. **(B)** Path and trajectory. **(C)** End-effector positioning errors.

in this study is clearly limited by the harmonic noise associated with the ZNN model instead of that of the robot manipulator.

6.2. Simulation Verification

In this subsection, the simulation results are presented to show the effectiveness of the presented ZNN-combined kinematic control methods in (27) and (28) for the four-link planar robot manipulator. In the simulations, the lengths of the robot's links are $l_1 = l_2 = l_3 = l_4 = 1$ m, and the initial joint state is set to $\theta(0) = [\pi/9, \pi/12, \pi/12, \pi/12]^{\mathrm{T}}$ rad.

1) Tricuspid path-tracking example: In this example, the presented ZNN-combined kinematic control methods in (27) and (28) are simulated for the four-link planar robot manipulator with its end-effector tracking a tricuspid path.

★ Single-harmonic noise: The ZNN-combined kinematic control method in (27) is investigated for the single-harmonic noise of $N(t) = [n_{ij}(t)] = [10\sin(2\pi t + 1)] \in \mathbb{R}^{2\times 3}$, in which the harmonic frequency is $f = 2\pi/(2\pi) = 1$ Hz. For comparison, the following noise-polluted kinematic control method based on the original ZNN design in (2) is also simulated:

$$\begin{cases} \ddot{\theta}(t) = B^{\dagger}(t)\ddot{r}_{\mathrm{a}}(t), \\ B^{\mathrm{T}}(t)B(t)\dot{Y}(t) = \dot{B}^{\mathrm{T}}(t) - \left(\dot{B}^{\mathrm{T}}(t)B(t) + B^{\mathrm{T}}(t)\dot{B}(t)\right)Y(t) \\ -\tau\Psi\left(B^{\mathrm{T}}(t)B(t)Y(t) - B^{\mathrm{T}}(t)\right) + O(t). \end{cases} \tag{29}$$

Figure 8 presents the simulation results synthesized by (29) using $\tau = 10$. The simulated end-effector trajectory of the robot manipulator does not match the desired tricuspid path, which means that (29) has failed to achieve the kinematic control of the robot manipulator due to the existence of harmonic noise. As shown in **Figures 9A,B**, the simulated end-effector trajectory is very close to the desired tricuspid path. **Figure 9C** shows that the maximal end-effector positioning error is $<2 \times 10^{-7}$ m, suggesting that the kinematic control is achieved successfully *via* (27) even if harmonic noise exists. These comparative results substantiate the effectiveness and noise-suppression capability of the ZNN-combined kinematic control method in (27) for the four-link planar robot manipulator.

★ Multiple-harmonic noise: The ZNN-combined kinematic control method in (28) is investigated for the

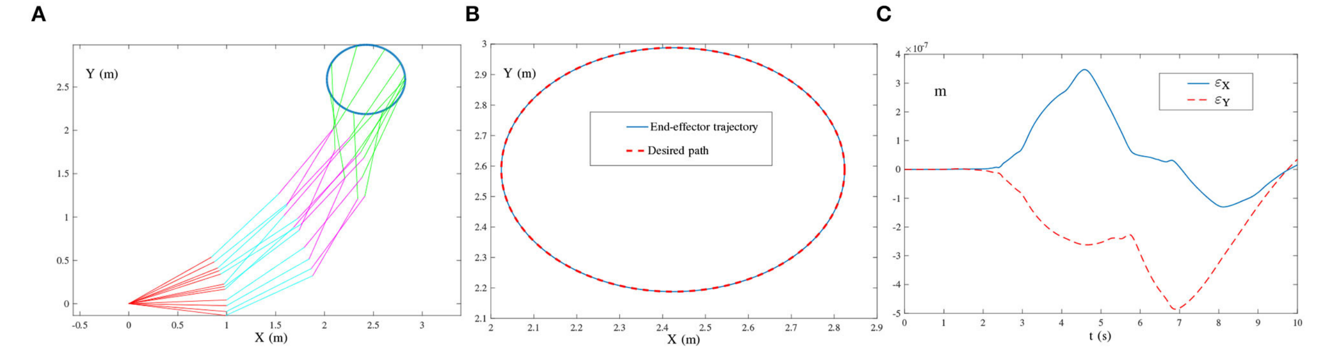


FIGURE 11 | End-effector of the four-link planar robot manipulator tracking the circular path synthesized by the ZNN-combined kinematic control method in (28) using $\tau = 10$ and $\lambda = 1$, where single-harmonic noise is considered. **(A)** Simulated motion trajectories. **(B)** Path and trajectory. **(C)** End-effector positioning errors.

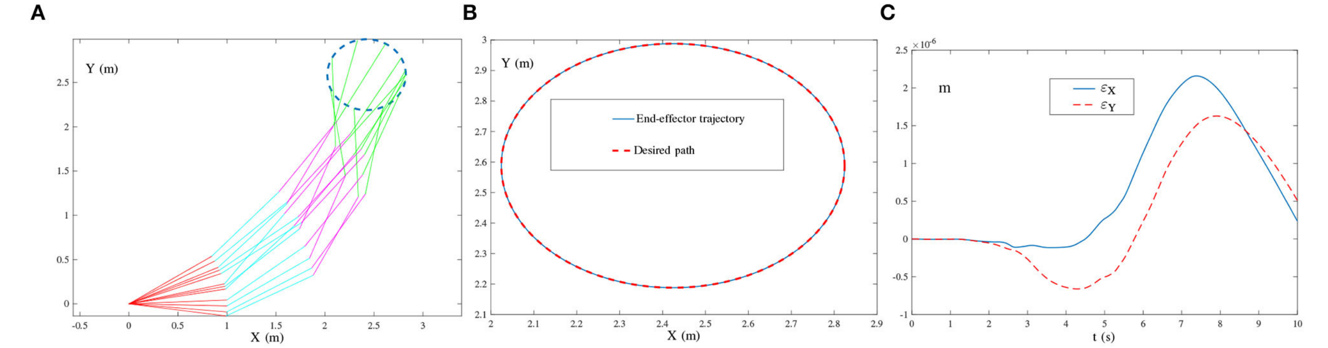


FIGURE 12 | End-effector of the four-link planar robot manipulator tracking the circular path synthesized by the ZNN-combined kinematic control method in (28) using $\tau = 10$ and $\lambda = 1$, where multiple-harmonic noise is considered. **(A)** Simulated motion trajectories. **(B)** Path and trajectory. **(C)** End-effector positioning errors.

multiple-harmonic noise:

$$\begin{aligned}\tilde{O}(t) &= [\tilde{o}_{ij}(t)]\\ &= [10\sin(2\pi t + 1) + 10^2\sin(10\pi t + 2) + 10^3\sin(20\pi t + 3)],\end{aligned}$$

in which the harmonic frequencies are denoted by $f_1 = 1$, $f_2 = 5$, and $f_3 = 10$ Hz. The corresponding simulation results synthesized by (28) are presented in **Figure 10**. The simulated end-effector trajectory is close to the desired tricuspid path, with the maximal positioning error being $<5\times10^{-6}$ m. Owing to the robustness of (28), this method is tolerant of multiple-harmonic noise, and the robot manipulator can track the desired tricuspid path successfully. The effectiveness of the ZNN-combined kinematic control method in (28) for the four-link planar robot manipulator is, thus, demonstrated.

2) Circular path-tracking example: In this example, the presented ZNN-combined kinematic control methods in (27) and (28) are simulated for the four-link planar robot manipulator with its end-effector tracking a circular path. Similarly, the following two harmonic noises are considered in the investigation of (27) and (28):

$$\begin{cases} O(t) = [o_{ij}(t)] = [10\sin(136\pi t + 1)],\\ \tilde{O}(t) = [\tilde{o}_{ij}(t)] = [10\sin(130\pi t + 1) + 10^2\sin(136\pi t + 2)\\ +10^3\sin(140\pi t + 3)] \in \mathbb{R}^{2\times3}. \end{cases}$$

The frequency of the single-harmonic noise approximates the bandwidth of the robot manipulator. The corresponding simulation results are illustrated in **Figures 11**, **12**.

As shown in **Figure 11**, the simulated end-effector trajectory of the robot manipulator matches the desired circular path, and the maximal positioning error is less than 4×10^{-7} m. These results substantiate the successful application of the ZNN-combined kinematic control method in (27) to the robot manipulator. The results presented in **Figure 11** also show that (27) has achieved robustness against single-harmonic noise, even if the noise frequency is nearly the same as the robot's bandwidth. Furthermore, the results presented in **Figure 12** suggest that the path-tracking task has been successfully completed even with the presence of multiple-harmonic noise. The effectiveness of the ZNN-combined kinematic control method in (28) for the robot manipulator is further demonstrated.

In summary, the simulation results (i.e., **Figures 8–12**) prove that the presented ZNN-combined kinematic control methods in (27) and (28) are more effective for the robot manipulator under the presence of harmonic noise compared with the noise-polluted the kinematic control method in (29). More importantly, the results indicate the application prospect of the proposed HNTZNN models in (11) and (14) *via* dynamic matrix pseudoinverse.

7. CONCLUSION

In this article, by combining the harmonic signal dynamics and the Li activation function, two harmonic noises tolerant ZNN models with fast convergence have been proposed. The theoretical analysis demonstrates that the proposed HNTZNN models possess satisfactory convergence and robustness properties. Furthermore, an example has been given to verify its effectiveness in solving the dynamic matrix pseudoinverse under different harmonic noises. Besides, such two HNTZNN models have been applied to the kinematic control of the four-link planar robot manipulator. In summary, the corresponding experimental results substantiate the effectiveness, superiority, and application prospect of the proposed HNTZNN models even with the existence of harmonic noises. Future study may lie in studying the proposed ZNN models activated by different activation functions. Another future research direction is to extend the proposed ZNN models to other application fields.

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

BL and DG presented the scheme. BL and YW designed experiments and wrote the manuscript. YW and JL carried out experiments. JL and YH analyzed the experimental results. All authors contributed to the article and approved the submitted version.

FUNDING

This study was supported in part by the National Natural Science Foundation of China (62066015 and 61962023), the Hunan Natural Science Foundation of China (2020JJ4511), the Research Foundation of Education Bureau of Hunan Province, China (20A396), and the Scientific Research Foundation of Jishou University (Jdy20063).

REFERENCES

Courrieu, P., (2008). Fast Computation of Moore-Penrose inverse matrices. *Comput. Sci*. 8, 25–29. doi: 10.48550/arXiv.0804.4809

Ding, L., Xiao, L., Liao, B., Lu, R., and Peng, H. (2017). An improved recurrent neural network for complex-valued systems of linear equation and its application to robotic motion tracking. *Front. Neurorobot*. 11, 45. doi: 10.3389/fnbot.2017.00045

Du, Z., Sun, Y., Su, R., Gan, Y., Ye, N., Zou, C., and Liu, W. (2017). "Harmonic detection and noise suppression in a magnetic nanoparticle thermometer," in *Proceedings of Chinese Automation Congress*, Jinan, 473–478. doi: 10.1109/CAC.2017.8242814

Guo, D., Nie, Z., and Yan, L. (2018a). The application of noise-tolerant ZD design formula to robots' kinematic control *via* time-varying nonlinear equations solving. *IEEE Trans. Syst. Man Cybern. Syst*. 48, 2188–2197. doi: 10.1109/TSMC.2017.2705160

Guo, D., Xu, F., and Yan, L. (2018b). New pseudoinverse-based path-planning scheme with PID characteristic for redundant robot manipulators in the presence of noise. *IEEE Trans. Control Syst. Technol*. 26, 2008–2019. doi: 10.1109/TCST.2017.2756029

Jafari, M. (2020). System identification of a soil tunnel based on a hybrid artificial neural network-numerical model approach. *IJST-T. Civ. Eng*. 44, 889–899. doi: 10.1007/s40996-020-00405-w

Jin, J. (2021). An improved finite time convergence recurrent neural network with application to time-varying linear complex matrix equation solution. *Neural Process. Lett*. 53, 777–786. doi: 10.1007/s11063-021-10426-9

Jin, L., Liao, B., Liu, M., Xiao, L., Guo, D., and Yan, X. (2017). Different-level simultaneous minimization scheme for fault tolerance of redundant manipulator aided with discrete-time recurrent neural network. *Front. Neurorobot*. 11, 50. doi: 10.3389/fnbot.2017.00050

Juang, L., and Wu, M. (2010). Image noise reduction using Wiener filtering with pseudo-inverse. *Measurement* 43, 1649–1655. doi: 10.1016/j.measurement.2010.09.021

Karsli, H., and Dondurur, D. (2018). A mean-based filter to remove power line harmonic noise from seismic reflection data. *J. Appl. Geophys*. 153, 90–99. doi: 10.1016/j.jappgeo.2018.04.014

Khalil, H. K. (2001). *Nonlinear Systems, 3rd Edn*. Englewood Cliffs, NJ: Prentice Hall.

Kornilova, M., Kovalnogov, V., Fedorov, R., Zamaleev, M., Katsikis, V., Mourtas, S., et al. (2022). Zeroing neural network for pseudoinversion of an arbitrary time-varying matrix based on singular value decomposition. *Mathematics* 10, 1–12. doi: 10.3390/math10081208

Li, S., Chen, S., and Liu, B. (2013). Accelerating a recurrent neural network to finite-time convergence for solving time-varying Sylvester equation by using a sign-BI-power activation function. *Neural Process. Lett*. 37, 189–205. doi: 10.1007/s11063-012-9241-1

Li, X., Li, S., Xu, Z., and Zhou, X. (2021). A vary-parameter convergence-accelerated recurrent neural network for online solving dynamic matrix pseudoinverse and its robot application. *Neural Process. Lett*. 53, 1287–1304. doi: 10.1007/s11063-021-10440-x

Liao, B., and Zhang, Y. (2014a). Different complex ZFs leading to different complex ZNN models for time-varying complex generalized inverse matrices. *IEEE Trans. Neural Netw. Learn. Syst*. 25, 1621–1631. doi: 10.1109/TNNLS.2013.2271779

Liao, B., and Zhang, Y. (2014b). From different ZFs to different ZNN models accelerated *via* Li activation functions to finite-time convergence for time-varying matrix pseudoinversion. *IEEE Trans. Neural Netw*. 133, 512–522. doi: 10.1016/j.neucom.2013.12.001

Liao, B., Zhang, Y., and Jin, L., (2016). Taylor $O(h^3)$ discretization of ZNN models for dynamic equality-constrained quadratic programming with application to manipulators. *IEEE Trans. Neural Netw. Learn. Syst*. 27, 225–237. doi: 10.1109/TNNLS.2015.2435014

Liu, L., Finch, A., Utiyama, M., and Sumita, E., (2020). Agreement on target-bidirectional recurrent neural networks for sequence-to-sequence learning. *J. Artif. Intell. Res*. 67, 581–606. doi: 10.1613/jair.1.12008

Perković, M. D., and Stanimirović, P. S. (2011). Iterative method for computing the Moore-Penrose inverse based on Penrose equations. *J. Comput. Appl. Math*. 235, 1604–1613. doi: 10.1016/j.cam.2010.08.042

Siciliano, B., Sciavicco, L., Luigi, V., and Oriolo, G. (2009). *Robotics: Modelling, Planning and Control*. London: Springer-Verlag. doi: 10.1007/978-1-84628-642-1

Skurichina, M., and Duin, R. P. W. (2002). Bagging, boosting and the random subspace method for linear classifiers. *Pattern Anal. Appl*. 5, 121–135. doi: 10.1007/s100440200011

Van der Veen, A. J., Talwar, S., and Paulraj, A. (1997). A subspace approach to blind space-time signal processing for wireless communication systems. *IEEE Trans. Signal Process*. 45, 173–190. doi: 10.1109/78.552215

Wang, J., (1997). Recurrent neural networks for computing pseudoinverses of rank-deficient matrices. *SIAM J. Sci. Comput*. 18, 1479–1493. doi: 10.1137/S1064827594267161

Xiang, Q., Liao, B., Xiao, L., Lin, L., and Li, S., (2019). Discrete-time noise-tolerant Zhang neural network for dynamic matrix pseudoinversion. *Soft Comput*. 23, 755–766. doi: 10.1007/s00500-018-3119-8

Xiao, L., Liao, B., Li, S., and Chen, K. (2018). Nonlinear recurrent neural networks for finite-time solution of general time-varying linear matrix equations. *Neural Netw*. 98, 102–113. doi: 10.1016/j.neunet.2017.11.011

Xiao, L., Zhang, Y., Liao, B., Zhang, Z., Ding, L., and Jin, L. (2017). A velocity-level bi-criteria optimization scheme for coordinated path tracking of dual robot manipulators using recurrent neural network. *Front. Neurorobot*. 11, 47. doi: 10.3389/fnbot.2017.00047

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

Zhang, B., Zhang, H., and Ge, S. S. (2004) Face recognition by applying wavelet subband representation and kernel associative memory. *IEEE sTrans. Neural Netw*. 15, 166–177. doi: 10.1109/TNN.2003.820673

Zhang, Y., Jiang, D., and Wang, J. (2002). A recurrent neural network for solving Sylvester equation with time-varying coefficients. *IEEE Trans. Neural* Netw. 13, 1053–1063. doi: 10.1109/TNN.2002.1031938

Zhang, Y., and Yi, C., (2011). *Zhang Neural Networks and Neural-Dynamic Method*. New York, NY: Nova.

Zhou, J., Zhu, Y., Li, X., and You, Z., (2002). Variants of the Greville formula with applications to exact recursive least squares. *SIAM J. Matrix Anal. Appl*. 24, 150–164. doi: 10.1137/S0895479801388194

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.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers | Frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 08 July 2022
doi: 10.3389/fnbot.2022.914353

UAV Based Indoor Localization and Objection Detection

Yimin Zhou[1,2]*, *Zhixiong Yu*[2] *and Zhuang Ma*[1,2]

[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, [2] University of Chinese Academy of Sciences, Beijing, China

This article targets fast indoor positioning and 3D target detection for unmanned aerial vehicle (UAV) real-time task implementation. With the combined direct method and feature method, a method is proposed for fast and accurate position estimation of the UAV. The camera pose is estimated by the visual odometer *via* the photometric error between the frames. Then the ORB features can be extended from the keyframes for the map consistency improvement by Bundle Adjustment with local and global optimization. A depth filter is also applied to assist the convergence of the map points with depth information updates from multiple frames. Moreover, the convolutional neural network is used to detect the specific target in an unknown space, while YOLOv3 is applied to obtain the semantic information of the target in the images. Thus, the spatial map points of the feature in the keyframes can be associated with the target detection box, while the statistical outlier filter can be simultaneously applied to eliminate the noise points. Experiments with public dataset, and field experiments on the established UAV platform in indoor environments have been carried out for visual based fast localization and object detection in real-time for the efficacy verification of the proposed method.

Keywords: visual SLAM, self-positioning, real-time localization, convolutional neural network, target detection, UAV

Edited by:
Shuai Li,
Swansea University, United Kingdom

Reviewed by:
Tugrul Oktay,
Erciyes University, Turkey
Ameer Tamoor Khan,
Hong Kong Polytechnic University,
Hong Kong SAR, China

****Correspondence:***
Yimin Zhou
ym.zhou@siat.ac.cn

Received: *06 April 2022*
Accepted: *25 May 2022*
Published: *08 July 2022*

Citation:
Zhou Y, Yu Z and Ma Z (2022) UAV Based Indoor Localization and Objection Detection.
Front. Neurorobot. 16:914353.
doi: 10.3389/fnbot.2022.914353

1. INTRODUCTION

Unmanned aerial vehicles (UAVs) have been developed rapidly in recent years, with diversified applications from military to civil fields, i.e., police patrolling, urban management, agriculture spraying, geology exploration, electric power patrolling, rescue and disaster relief, video shooting, and other industries due to their small size, low cost, high maneuverability, and fast speed (Chebrolu et al., 2018). As a common positioning system, GPS would lose function and fail to provide accurate position information in indoor or GPS-denied environments, hence visual SLAM (Simultaneous Localization And Mapping) technology can be adopted in such positioning scenes owing to its abundant positioning information and wide applicability (Kasyanov et al., 2017). It should be aware that the collected image information from UAV are 2-dimensional (2D) without stereoscopic information, so the "what" of the target can be obtained while the "where" of the target in the space is unknown. In order to complete such post-disaster rescuing tasks successfully under complex, unknown environments, two fundamental problems should be solved simultaneously, UAV self-localization and target detection (Cavaliere et al., 2017).

Visual localization is generally realized *via* visual SLAM technologies, where the visual odometry (VO) (Jiang et al., 2018) can be used to estimate the pose variation of the camera *via* captured consecutive frames, usually divided into two categories, i.e., indirect/feature-based methods and direct methods (Su and Cai, 2018). The most representative of the feature methods is the ORB-SLAM (Mur-Artal et al., 2015), with the aid of ORB (Oriented FAST and Rotated Brief) feature possessing rotation invariance and scale invariance *via* pyramid construction so as to assist SLAM algorithm to have endogenous consistency in the feature extraction and tracking, keyframe selection, 3D reconstruction, and closed-loop detection. Bundle Adjustment (BA) is performed to minimize the feature reprojection error in a local set of keyframes. Furthermore, the localization precision can be improved in ORB-SLAM3 (Campos et al., 2021), where the pose and map points are optimized *via* cumulative errors avoidance at the back-end. However, ORB-SLAM would fail to track in a less texture environment due to insufficient feature points extraction. Unlike feature-based methods, the direct method establishes the relationship between frames through the gray information of pixels to construct the camera motion estimation with a faster processing speed. The direct sparse odometry (DSO) system is further proposed to optimize the photometric parameters for robustness improvement (Wang et al., 2017). However, the cumulative error is inevitable due to a lack of global back-end optimization, resulting in overall poor system accuracy.

Combining the advantages of the direct methods and feature-based methods, a fusion scheme SVO (semi-direct visual odometry) is proposed (Forster et al., 2014), where the minimized photometric error is used to optimize the pose estimate. Moreover, only feature points are extracted from the keyframes for tracking without descriptors calculation for considerable speed enhancement. However, the initialization of the pose could fail in the head-up view if all the points are not on the same plane and SVO lacks global optimization. Then the 2nd version of SVO2.0 has been proposed by adding multi cameras to improve the tracking of edges and corners and IMU (internal measurement unit) pre-integration (Forster et al., 2016), which can further increase the processing speed. Although vision-fused IMU can enhance the robustness of the SLAM system, it would also bring higher algorithm complexity and degrade the real-time performance.

Object detection is dependent on the image understanding, mainly including two parts, i.e., type decision and size and position estimate of the object in the image. Since AlexNet (Beeharry and Bassoo, 2020) won the championship in ILSVRC2012, a deep convolutional neural network (CNN) has widely been applied with autonomously learning features. RCNN (region CNN) (Girshick et al., 2014) is a pioneering work of applying deep CNN to target detection. FastRCNN (Girshick, 2015) and FasterRCNN (Ren et al., 2016) are further proposed to effectively avoid the image scaling problem. On the other hand, YOLO (You Only Look Once), has been proposed in Redmon et al. (2016), which can directly regress multiple positions of the image to acquire the target box and category, thus simplifying the detection process. Combined with YOLO and FasterRCNN, a new algorithm SSD (single shot multibox detector) (Liu et al., 2016) is proposed to obtain the frame coordinates *via* different convolutional layers. To date, YOLOv5 is released in Oct 2020, possessing higher object identification accuracy (Kuznetsova et al., 2020).

Different from 2D target detection, 3D target detection should mark the spatial position of the target. In Chen et al. (2017), a set of 3D object proposals with stereo images for 3D object detection are generated by minimizing an energy function that encodes several depth-informed features, i.e., prior object size, object placement on the ground plane as an extension of FastRCNN in 3D field. Although the proposed method can outperform in object detection and orientation estimation tasks for all three KITTI (database issued by Karlsruhe Institute of Technology and Toyota Technological Institute) object classes, it has a low processing speed per image up to 4s with worse real-time performance. A stereo 3D object detection method is proposed with Instance-DepthAware module and disparity adaptation and matching cost reweighting in Peng et al. (2020), where solely RGB (Red-Green-Blue) images are used as the training data to predict the depth of the 3D bounding boxes centering in the images. Although image-based methods have achieved great success in object detection, the performance of 3D object detection falls behind the LiDAR-based (Light Detection And Ranging-based) approaches due to the inaccurate depth information. While the image-based depth maps can be converted to pseudo-LiDAR representation *via* transformation from the dense pixel depths of stereo imagery and back-projecting pixels into a 3D point cloud (Chen et al., 2020), the main challenge is the heavy computation load of the LiDAR-based detection.

The combination of the visual SLAM and object detection can increase the environment perception capability. For example, VSO (visual semantic odometry) (Liu H. et al., 2019) can optimize the reprojection error between images through semantic information. A semantics SLAM system (Lee et al., 2019) is proposed with the combination of sensor status and semantic landmarks which can transform the semantic map into a probability problem to optimize the reprojection error. The system robustness can also be improved *via* the intensity-SLAM (Wang et al., 2021) and loop detection (Liu Y. et al., 2019) optimization. Moreover, SOF-SLAM (Cui and Ma, 2019) can identify the dynamic features through semantic optical flow and remove these features, achieving stable tracking in dynamic scenes.

Visual SLAM can assist in target detection as well (Vincent et al., 2020). A semantic fusion method (Li et al., 2018) is proposed to combine CNN with a dense vision SLAM scheme for semantic segmentation, where the category probability distributions of the images are fused into a SLAM map to construct a 3D dense semantic map. Further, the semantic information acquirement efficiency can be greatly increased with the adoption of the SSD or YOLO framework (Bavle et al., 2020). For instance, Quadric SLAM (Nicholson et al., 2018) is proposed to identify the position, size and direction of the object simultaneously. However, this kind of method usually requires semantic segmentation of the target based on dense pixel or point cloud information.

The current target detection algorithms can only estimate the target position in the images, while the spatial position of the target still remains unknown. To tackle the mentioned problems during UAV indoor localization, this article proposes an object localization framework for object spatial location estimation (refer to **Figure 1**). To be specific, the positioning method is designed to achieve rapid and accurate location estimation of the UAV. Then the target spatial position estimation based on CNN

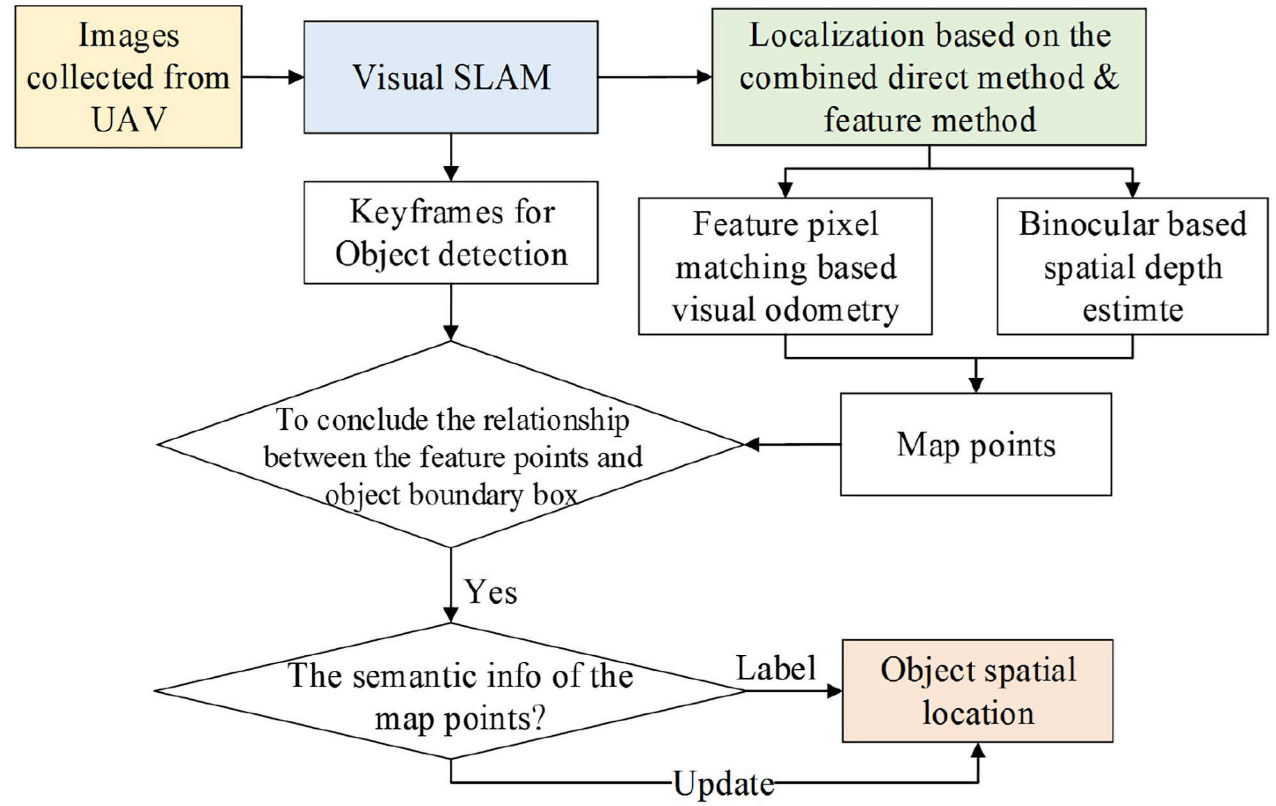

Images collected from UAV
Visual SLAM
Localization based on the combined direct method & feature method
Keyframes for Object detection
Feature pixel matching based visual odometry
Binocular based spatial depth estimte
Map points
To conclude the relationship between the feature points and object boundary box
Yes
The semantic info of the map points?
Label
Object spatial location
Update

FIGURE 1 | The procedure of the proposed method.

is applied to detect the searched target in an unknown space. The contributions of the article are summarized as follows.

1) A method is proposed for UAV fast self-positioning. Applying the direct method with a visual odometer delicately, the optical error is used for inter-frame matching to estimate the camera posture. The ORB feature extracted from the keyframes can improve the map consistency while a binocular depth filter is introduced to increase the positioning accuracy.
2) A method is proposed for target spatial position estimation based on YOLOv3. The spatial position of the target can be constructed by correlating the relationship among the feature points for reliable depth information providence and the target detection frame in the keyframes. Furthermore, the statistical outlier filter is used to eliminate the noise to acquire more accurate target position.
3) Unmanned aerial vehicle platform has been setup for indoor rapid positioning and object detection. A series of experiments on the public dataset and in the actual scene have been performed to verify the effectiveness of the proposed method with real-time target spatial localization performance.

The remainder of the article is organized as follows. Section II explains the method of indoor localization in detail. Section III explains the objection detection and spatial position estimation. In Section IV, the experiment platform of the indoor UAV fast localization and objection detection is setup for the proposed method verification with public data and field scenarios. Section V concludes the article and future directions are provided.

2. METHOD OF INDOOR LOCALIZATION

Combined with the direct method and feature-based method, a localization algorithm is proposed in this Section. Here, ORB-SLAM2 rather than ORB-SLAM3 algorithm is adopted with only an embedded stereo camera for the localization and the whole procedure is depicted in **Figure 2**. The localization algorithm includes four threads, i.e., tracking thread, feature extraction and depth filter (FEDF) thread, local mapping thread, and loop-closing thread.

The VO based on the direct method for localization in the tracking thread is the first proposed algorithm while the second algorithm for spatial point depth estimation is designed on the basis of stereo depth filter embedded in the FEDF thread. The other two local mapping and loop closing threads are kept the same as in ORB-SLAM2.

2.1. Direct Method Based Visual Odometer

Since the VO in ORB-SLAM2 has to extract feature points and calculate descriptors per frame, SVO is adopted for direct frame matching *via* feature points. The procedure of the direct

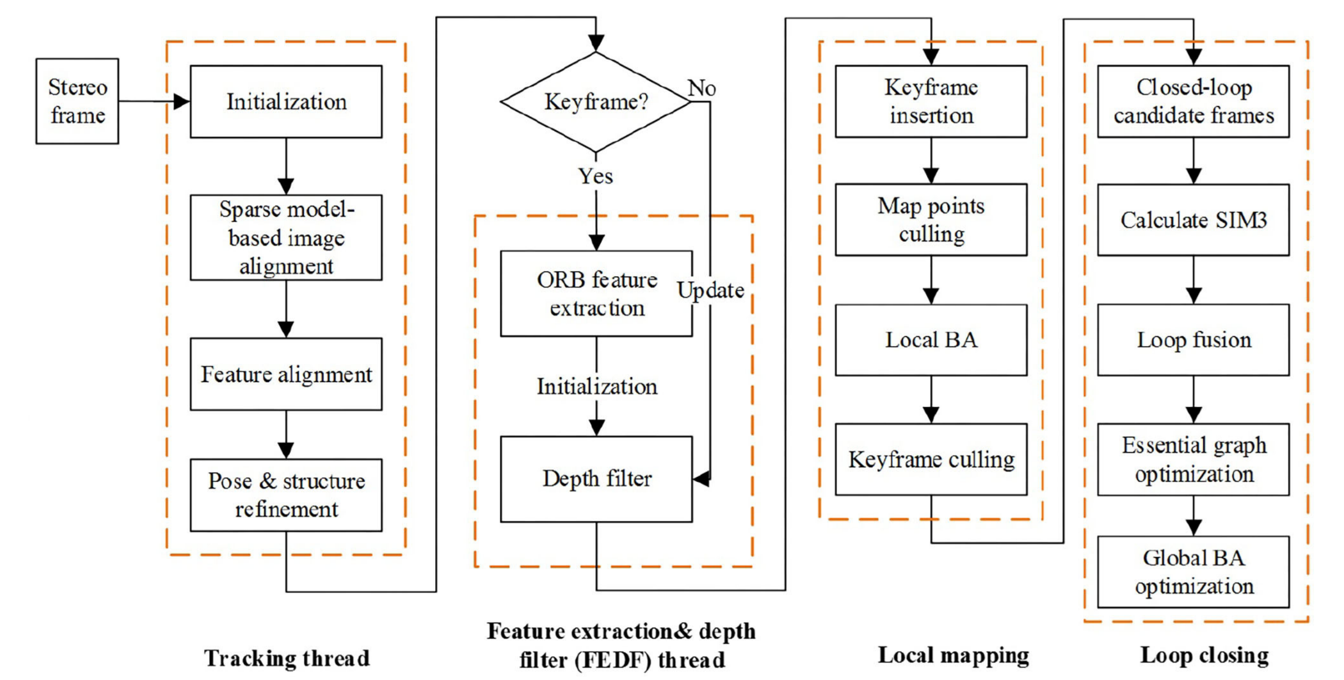


FIGURE 2 | The framework of the proposed localization algorithm.

method based VO contains three steps: pose estimate, feature points alignment and pose optimization, where the coordinate origin of the coordinate axes is the left lens optical center of the stereo camera.

2.2. Step I. Direct Method Based Pose Estimation

First, the feature points of the binocular image are extracted till the number exceeds the threshold, then the image is designated as the keyframe. Next, the parallax is obtained by binocular image feature matching and the map point depth can be obtained with triangulation so as to acquire the initial pose and map points position. As depicted in **Figure 3A**, $\{I_k, I_{k-1}\}$ are the image intensities at the k^{th} moment and the previous $(k-1)^{th}$ moment, while $T_{k,k-1}$ describes the pose variant from the $(k-1)^{th}$ moment to the k^{th} moment. π is used to describe the projection process from 3D space to the image space, while π^{-1} is the back projection process.

The initial pose of the $T_{k,k-1}$ can be obtained *via* the uniform velocity model or the identity matrix, while the feature point coordinate and the related spatial depth of the $(k-1)^{th}$ frame can be estimated from the previous multiple frames, denoted as (p_i, d_i). Then the specific feature point can be projected at the spatial point p_i with the coordinate $P_{i,k-1}$ in the $(k-1)^{th}$ frame reference. Through the $T_{k,k-1}$ transformation, it can be converted with the k^{th} frame reference, denoted as $P_{i,k}$. Afterwards, it can be projected on the k^{th} image *via* camera model with the coordinate p_i'.

The brightness of the same point in the two consecutive frames is assumed unchanged due to the transient time interval (Jiang et al., 2018). Thus, the residual function can be formed based on the gray value difference of the image patches adjacent the $(k-1)^{th}$ feature point and the reprojected point of the k^{th} frame,

$$\delta I\left(T_{\mathrm{k,k-1}}, p_i\right) = I_k\left(\pi \cdot \left(T_{\mathrm{k,k-1}} \cdot \pi^{-1}\left(p_i, d_{\mathrm{i}}\right)\right)\right) - I_{k-1}\left(p_i\right) \quad (1)$$

where $\delta I\left(T_{\mathrm{k,k-1}}, p_i\right)$ is the intensity variation due to the pose movement from the $(k-1)^{th}$ moment to the k^{th} moment. Then the pose variation in $T_{k,k-1}$ transformation can be optimized *via* the maximum likelihood estimation by minimizing the photometric residual,

$$T_{\mathrm{k,k-1}} = \arg\min_{T_{\mathrm{k,k-1}}} \frac{1}{2} \sum_{i \in R} \left\| \delta I\left(T_{\mathrm{k,k-1}}, p_i\right) \right\|^2 \quad (2)$$

where R is the visible image points set in the k^{th} frame back-projected from the points with the known depth d_i in the image at the $(k-1)^{th}$ moment. Gauss-Newton (G-N) or Levenberg-Marquadt (L-M) iterative methods (Balabanova et al., 2020) can be used to solve Equation (2) for the update of $T_{k,k-1}$ estimation.

2.3. Step II. Alignment of the Feature Pixels

With the obtained pose $T_{k,k-1}$ between two consecutive frames *via* direct method, the feature points from the previous frame $(k-1)$ can be reprojected on the current frame k but with coordinate inconsistency due to the noise. Since more accurate pixel information exists in the common view keyframe adjacent to the current frame k, the position of the feature pixels in the k frame can further be optimized through the established map points in the nearest common view keyframes. The pose

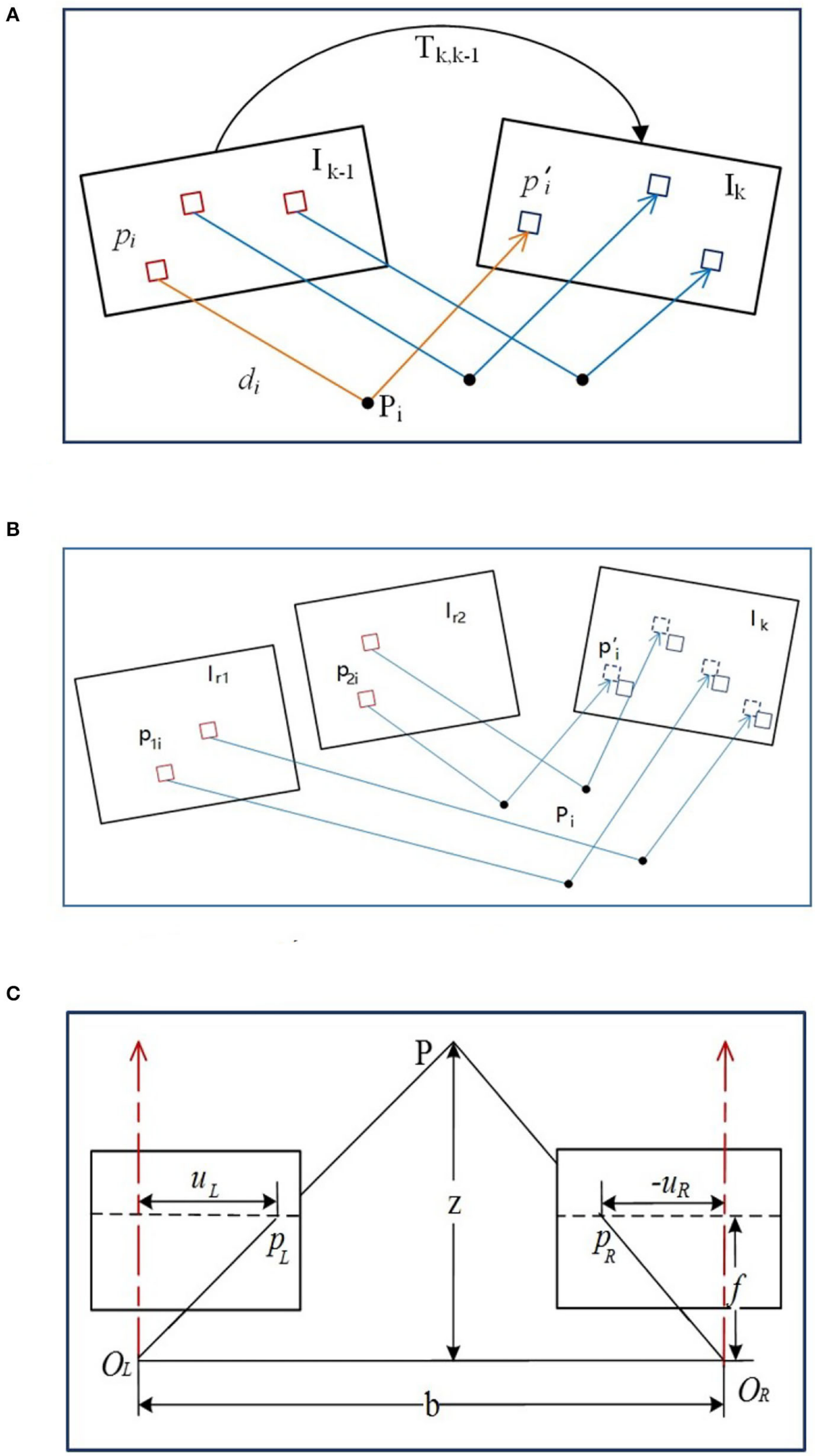


FIGURE 3 | The illustration of image measurement. **(A)** The direct method based pose estimation. **(B)** The alignment of feature pixels. **(C)** Triangulation measurement with stereo camera.

relationship among the k frame and the common view keyframes I_{r1} and I_{r2} can be acquired from $T_{k,k-1}$, so the feature points p_{1i} and p_{2i} in the keyframes can be reprojected on the current frame k, illustrated in **Figure 3B**. Assuming luminosity invariance, the residual function can be reconstructed from the image gray values difference to optimize the coordinates of feature points *via* minimized luminosity error,

$$p_i' = \arg\min_{p_i'} \frac{1}{2} \left\| I_k\left(p_i'\right) - A_i \cdot I_r\left(p_i\right) \right\|^2 \tag{3}$$

where $I_r(\cdot)$ is the previously observed frame, A_i is the rotating and stretching affine operation. If the common view keyframes are far away from the current frame, the feature patch in the keyframe should be first transformed *via* A_i operation for further comparison. Again, Equation (3) can be solved *via* G-N or L-M methods for more accurate coordinates estimate p_i' obtainment.

2.4. Step III. Pose Optimization

With a more precise match between the features of the keyframe and the previous frame, the feature points can be reprojected on the current I_k frame. Then the position residual function is formed by the pixel coordinate difference from the reprojection point and the related p_i', written as,

$$\left\| \delta p_i \right\| = \left\| p_i' - \pi \cdot \left(T_{\mathrm{k,w}} P_i\right) \right\| \tag{4}$$

and the pose of the current frame is optimized as,

$$T_{\mathrm{k,w}} = \arg\min_{T_{\mathrm{k,w}}} \frac{1}{2} \sum_i \left\| p_i - \pi \cdot \left(T_{\mathrm{k,w}} P_i\right) \right\|^2 \tag{5}$$

while the position P_i on the map can also be optimized *via* the same maximum likelihood function simultaneously with G-N or L-M methods for solution.

2.5. Binocular Based Spatial Point Depth Estimation

Building an accurate and reliable map is necessary for the camera pose calculation from the spatial map points *via* feature matching and triangulation. Since the map point depth from the triangulation is highly affected by the parallax from the two frames, a depth filter is introduced for depth optimization with the adoption of the calibrated stereo camera for the initial seed point depth determination.

2.5.1. Triangulation Ranging Model

The stereo camera model should be rectified first, usually with epipolar line adjustment (Bradski and Kaehler, 2008) to make the optical axes parallel with O_L and O_R centers. As shown in **Figure 3C**, b is the baseline of the stereo camera, f is the focal length, p_L and p_R are the positions of the spatial point P in the left and right cameras, and $\{|u_L|, |u_R|\}$ are the distances from each axis of $\{p_L, p_R\}$. Based on the similar triangles, it has,

$$\frac{z-f}{z} = \frac{b - u_L + u_R}{b},\ x = u_L - u_R \Rightarrow\ z = \frac{fb}{x} \tag{6}$$

where x is the parallax, i.e., the difference of binocular abscissa, and z is the spatial depth. The depth usually follows the normal distribution (Ammann and Mayo, 2018), i.e., the depth z_p of the spatial point P follows $N\left(\mu, \sigma^2\right)$ distribution. With the depth z_k (followed by $N\left(\mu_o, \sigma_o^2\right)$ normal distribution) calculated *via* epipolar line matching from the new k^{th} frame, new depth data are fused with the existing ones, still followed by normal distribution $N(\mu_f, \sigma_f^2)$,

$$\mu_f = \frac{\sigma_o^2 \mu + \sigma^2 \mu_o}{\sigma^2 + \sigma_o^2},\ \ \sigma_f^2 = \frac{\sigma_o^2 \sigma^2}{\sigma^2 + \sigma_o^2} \tag{7}$$

The depth can be assumed as converged if σ_f is less than the threshold, whereas the uncertainty of the depth should also be considered. Given a spatial point P, it is projected on p_1 and p_2 of any two frames with the optical centers O_1 and O_2, respectively, as depicted in **Figure 4A**.

The pixel error of the spatial point P is considered as the distance variation and the related angles changed from $\{\|d\|, \{p_2, \beta, \gamma\}\}$ to $\{\|d'\|, \{p_2', \beta', \gamma'\}\}$. Based on the geometric relationship, it has,

$$\begin{gathered} \vec{\alpha} = \vec{d} - \vec{t} \\ \beta = arccos < \vec{d},\ \vec{t} > \\ \alpha = arccos < \vec{\alpha},\ \overrightarrow{-t} > \end{gathered} \tag{8}$$

It can be rearranged as,

$$\left\| d' \right\| = \|t\| \frac{\sin \beta'}{\sin \gamma'} \tag{9}$$

where β is the angler between O_2P and O_1O_2, and γ is the angler between O_1P and O_1O_2. Thus, the depth uncertainty caused by one pixel bias is,

$$\sigma_o = \|d\| - \|d'\| \tag{10}$$

Since the inverse depth also obeys the normal distribution (Younes et al., 2019), the σ_o variance can be transformed as

$$\sigma_o = \left\| \frac{1}{d} \right\| - \left\| \frac{1}{d'} \right\| \tag{11}$$

Finally, with the pose calculation of the new frame, the depth filter is used to merge the depth and the depth uncertainty into the previous ones until the uncertainty is less than the threshold so as to generate the map points accordingly.

2.5.2. Depth Estimation

The depth filter is initialized with the calculated depth from feature matching of the extracted ORB features from the keyframe. Given a point P, the left and right image frames can be triangled *via* Equation (6) to obtain their initial depth values (seen in **Figure 4B**). The point P is assumed as the joint view with a non-keyframe and keyframe, while the new estimated depth can be calculated by matching through the feature block mean

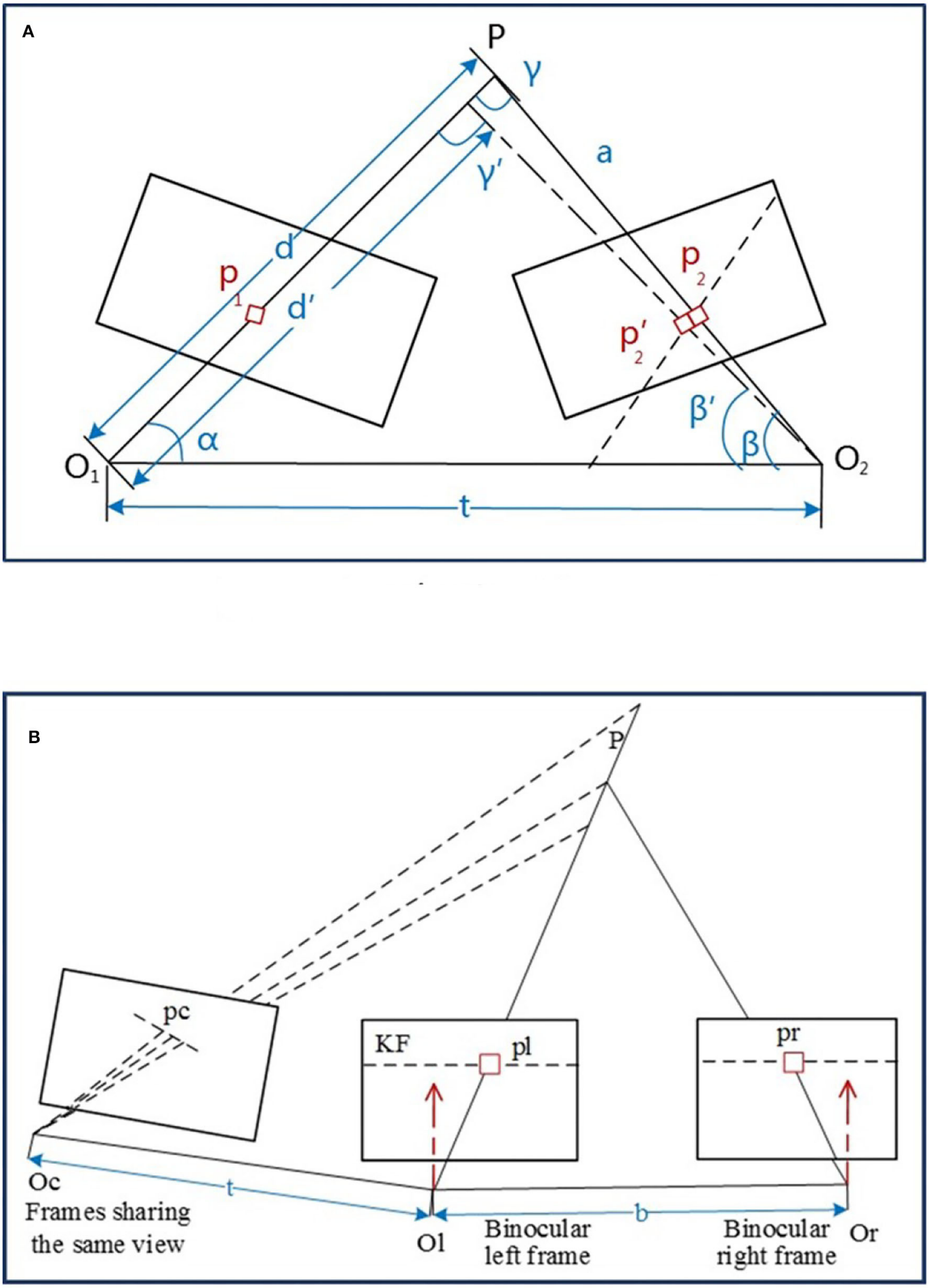


FIGURE 4 | The illustration of depth estimation. **(A)** Depth estimate with uncertainty. **(B)** Spatial point depth estimate.

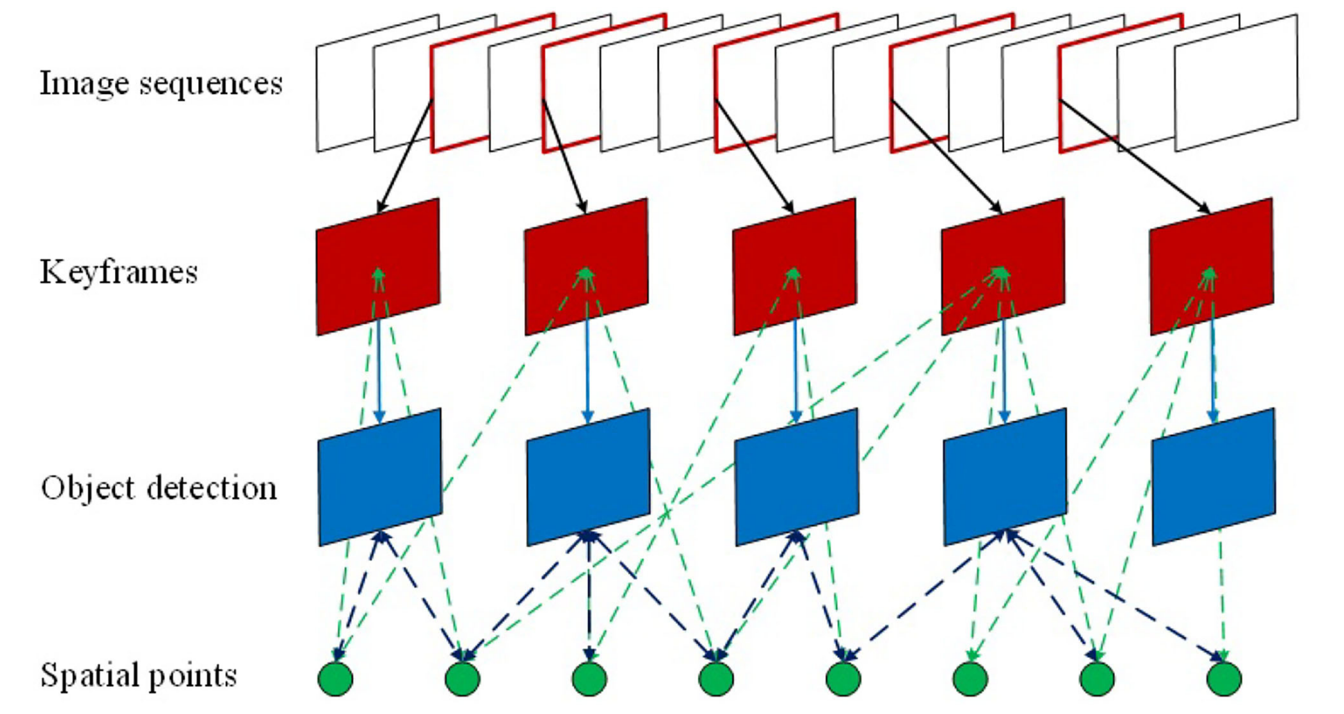


FIGURE 5 | The relationship between the object and spatial map points.

gradient and epipolar line searching so that the depth filter can be updated.

Furthermore, the newly added spatial points are fused with the existing map points. The minimized reprojection error function is constructed to optimize the pose of the keyframe and the coordinates of the spatial points. Finally, the redundant keyframes are filtered out and sent to the loop-closing thread.

Here, the loop-closing thread is the same as ORB-SLAM2 for similarity comparison of the bag of words of the new keyframes and existing ones. If the similarity level exceeds a dynamic threshold, a closed-loop occurs and the pose of the new keyframe and the closed-loop keyframe are adjusted so that the poses of all the keyframes are optimized. With the relocation and loopback ability of ORB-SLAM2, the lost camera's pose tracking can be recovered *via* the position comparison of the feature points among the previous keyframes and the current frame. At the same time, a more accurate trajectory and map can be obtained *via* closed-loop fusion with the data from different tracking periods.

3. OBJECT DETECTION AND POSITION ESTIMATION

3.1. Algorithm Framework

As general object detection algorithms, the target type and position in the images can be estimated *via* salient feature detection. However, especially in complex scenarios, only the 2D target position in the image is insufficient without the 3D target spatial position. Although the constructed semantic map combined with the objection detection and visual SLAM can be used to estimate the object category and position in a 3D environment, it is mostly achieved based on dense or semi-dense point clouds for multiple targets estimation, which has a higher requirement on the collection and calculation. As for the mission of UAV in indoor environments, there is no need to rebuild the 3D mapping environment or estimate the detailed posture of the target but only require to label the location of the target in the searching paths. Hence, object detection and spatial location estimate to the specific targets are proposed based on the sparse space points.

The YOLOv3 object detection framework based on CNN is adopted in this article, which is integrated into the visual SLAM system. The keyframes are sent to the YOLOv3 object detection framework to detect the specific target, i.e., people, and the spatial map points stored in the keyframes are used to construct the relationship between the detected target and the spatial point position, so as to estimate the spatial position of a specific target. The details of the YOLOv3 are omitted here, readers who are interested in the algorithm can refer to Zhang et al. (2020).

3.2. The Estimate of the Target Spatial Position

As seen in **Figure 5**, since the map points information of SLAM is kept in the keyframes, the feature points in the keyframes can thus be connected with the objects *via* the object detection so as to estimate the spatial location of the target.

First, the keyframe images are sent to the YOLOv3 network for object detection, which is used to determine the relationship

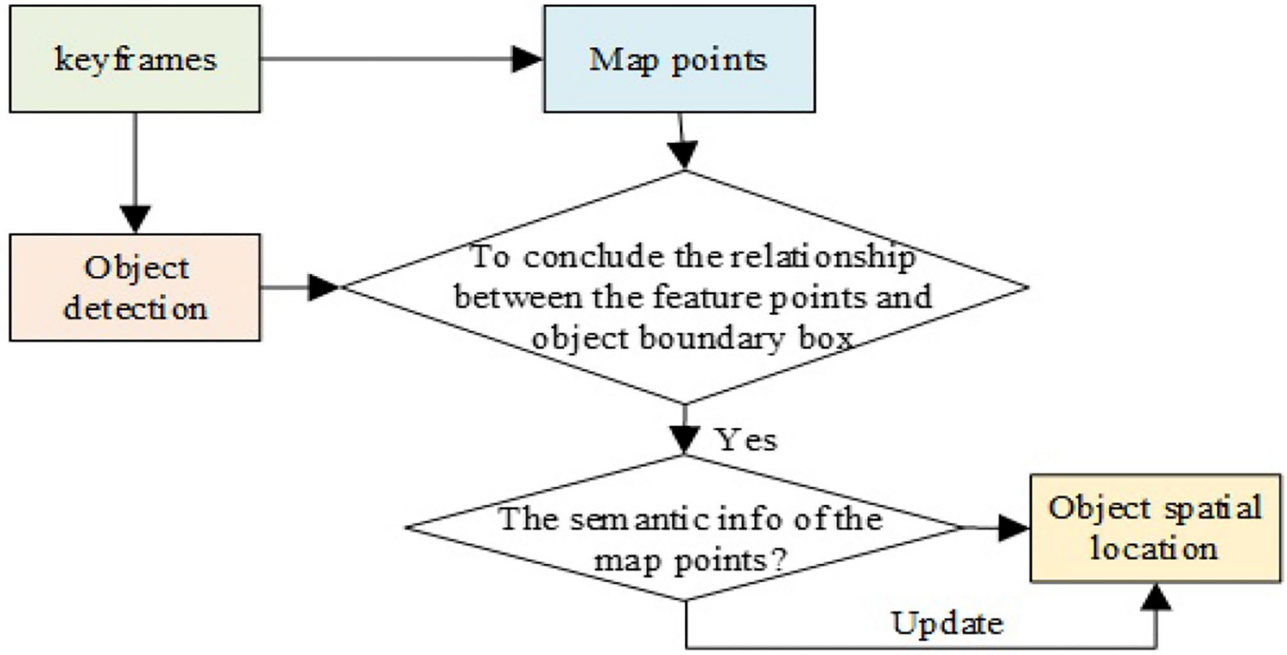

keyframes
Map points
Object detection
To conclude the relationship between the feature points and object boundary box
Yes
The semantic info of the map points?
Object spatial location
Update

FIGURE 6 | The procedure of the spatial target position estimation.

between the feature points in the keyframes and the object detection frame. If the feature point is inside the detection boxes, the map points related to the feature points are searched to identify whether the map points have already been marked with the target semantic information. If not, it will be marked as a new pending target and if it is marked as the same pending target in 3 consecutive frames, the point is thus marked as a new spatial target point. The world coordinate is applied here as the reference system to calculate the maximum distance of the spatial target point distribution in the x, y, and z directions, where these distances are used as the widths of the target bounding box to mark the spatial target position in the object detection box. Subsequently, the ratio of the map points is calculated carrying semantic information to all map points in the bounding box. When the ratio exceeds the threshold, the semantic information of the map points related to other feature points in the box is updated as the new semantic information, together with the spatial target position and the detection box. The procedure of the spatial object location estimation is illustrated in **Figure 6**.

The object detection box in the detection algorithm is usually a rectangle, which cannot accurately represent the boundary between the target area and the background area as background points are often included inside the target box resulting negative effect on the accuracy of the target spatial position estimation. For instance, the target rectangle box in the left graph of **Figure 7** not only contains the target "teddy bear" but also the feature points of the background wall.

Generally speaking, the distances between the background points and the target points are large, and the number of background feature points in the detection box is less than that of the target. Therefore, the depth information of the feature points in the object detection rectangle can be filtered by the spatial point distribution. Here, SOF (Statistical Outlier Filter) is adopted to remove the outliers (Bokovoy and Yakovlev, 2018).

As for any spatial point, the mean distance to its nearest k points can be calculated. Assuming the mean distance of each point follows the normal distribution with the expected average value μ and variance σ^2, the threshold of the mean distance is written as,

$$d_{max} = \mu + \alpha \times \sigma \tag{12}$$

where α is the scale factor of neighboring points. Therefore, the points whose distances between the neighboring points are larger than the defined threshold is removed (refer to the right graph of **Figure 7**) so as to estimate the position of the target in space more accurately.

4. EXPERIMENTS FOR INDOOR UAV FAST LOCALIZATION AND OBJECT DETECTION

4.1. Experiments of Localization on Public Data

The public UAV dataset EuRoC (ASL, 2012) is adopted here containing indoor sequences collected from an AscTecFirefly micro UAV where the resolution of the stereo camera is 752 × 480 and 20*fps* (frame per second) processing speed. EuRoC data includes 11 sequences, while the sequences {{MH01, MH02, MH03}, V201} are randomly selected from the large industrial workshop and general office room such as two

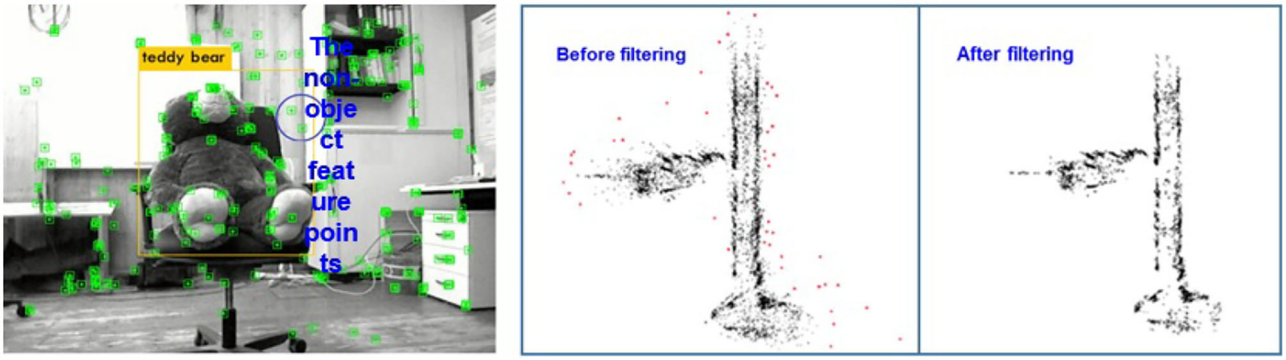


FIGURE 7 | (Left): The position illustration of the target box and the feature points; (Right): The result of the statistical outlier filter.

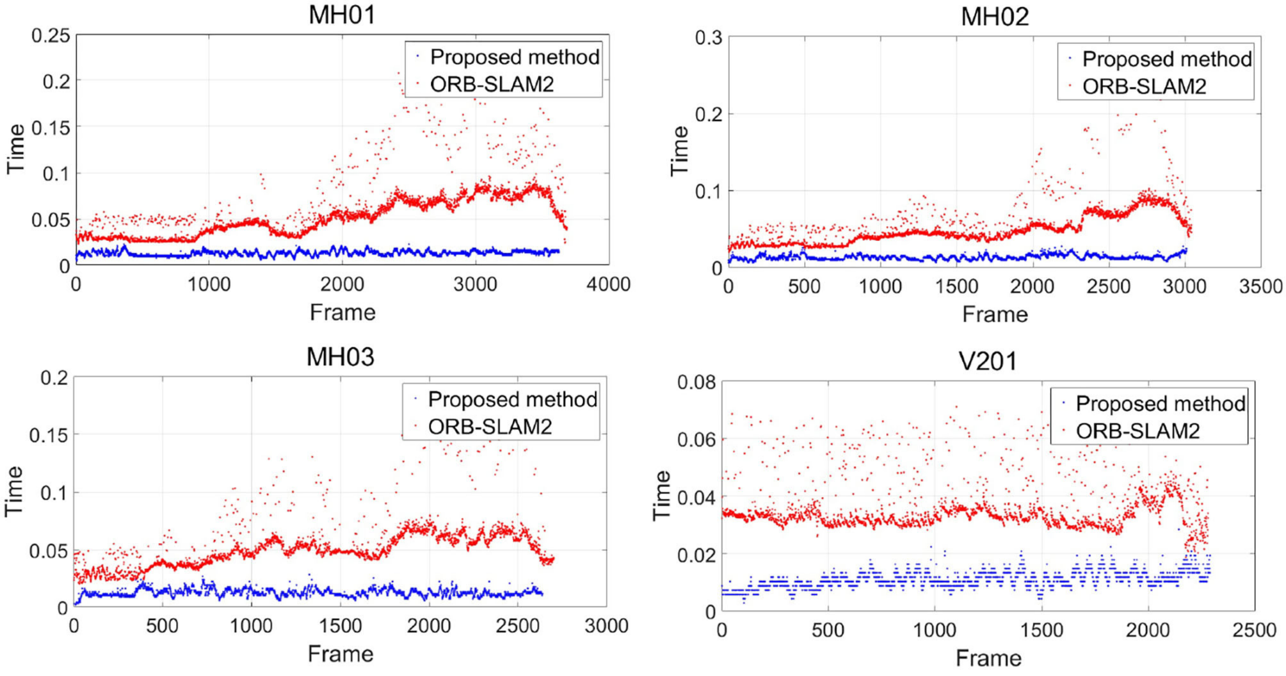


FIGURE 8 | Image processing time distribution per frame.

typical scenes. Besides, another real trajectory collected by the Leica MS50 lasers canner is used for performance comparison. The processing environment of the experiments is Ubuntu 16.04 system, GPU is Nvidia GeForce GTX 1080, and the processor is Inteli7-8750 with 16GB RAM. Then the proposed method is evaluated from the processing time per frame and pose localization accuracy in two criteria indexes.

4.1.1. Processing Time Evaluation

Figure 8 shows the distribution of processing time spent for each frame, while the horizontal coordinate is the frame number, and the vertical coordinate is the processing time in seconds. The red points and blue points delegate the processing rate of ORB-SLAM2 and the proposed method, respectively. It is shown in **Table 1** that the frame processing speed of our method is much faster than that of the ORB-SLAM2 with four random selected tracks, which is primarily due to the direct matching with the

TABLE 1 | The processing time per frame.

Data	Number of the frames	Proposed method(s)	ORB-SLAM2(s)
MH01	3,682	0.0128	0.0539
MH02	3,040	0.0129	0.0502
MH03	2,700	0.0127	0.0511
V201	2,280	0.0111	0.0357

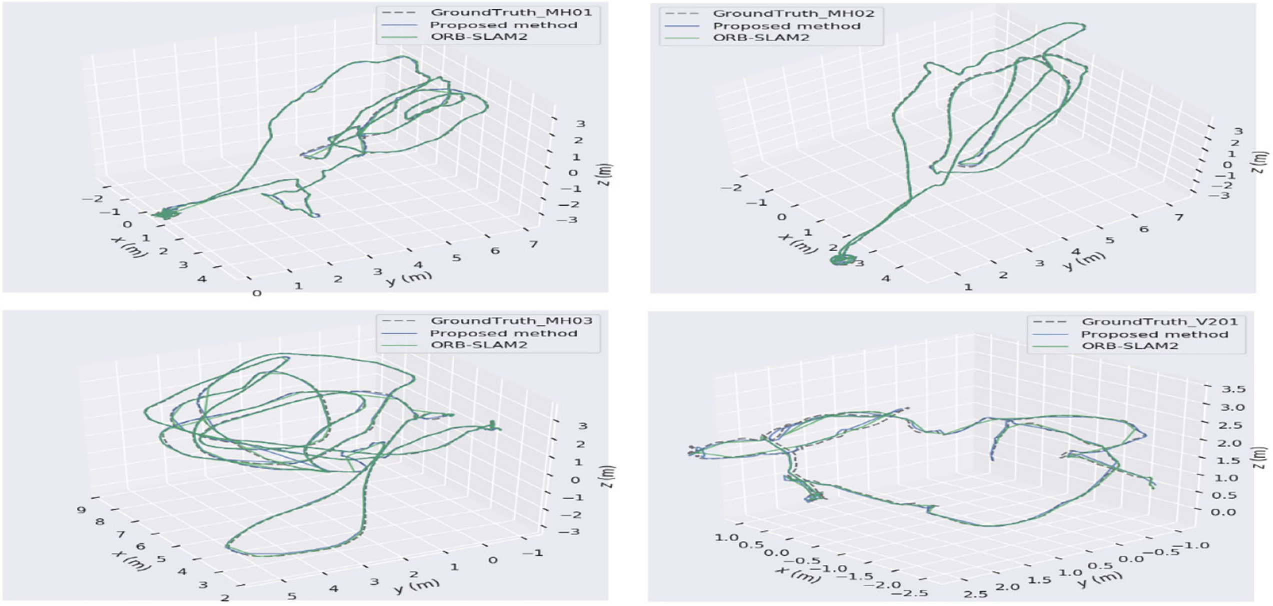


FIGURE 9 | Keyframe trajectory comparison on EuRoc datasets.

TABLE 2 | Keyframe trajectory accuracy comparison(m).

	Proposed method			**ORB-SLAM2**		
	Max	**Mean**	**RMSE**	**Max**	**Mean**	**RMSE**
MH01	0.0902	0.0274	0.0347	0.0810	0.0307	0.0363
MH02	0.0900	0.0276	0.0331	0.1012	0.0312	0.0375
MH03	0.1112	0.0338	0.0378	0.1071	0.0391	0.0421
V201	0.1182	0.0588	0.0613	0.0964	0.0392	0.0458

photometric error during tracking with no feature extraction and descriptor calculation.

4.1.2. Position Accuracy Evaluation

Due to the difficulty in measuring actual 3D map points, the track error of the camera motion is generally used for VO or visual SLAM algorithm performance evaluation. **Figure 9** illustrates the trajectories comparison among the four datasets and the actual one from the proposed algorithm and ORB-SLAM2, denoted as the blue curve, green curve, and dashed line, respectively. To be specific, the Root Mean Square Error (RMSE) is used for location evaluation between the estimate and the actual one,

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}^{n}\left(\hat{X}_i - X_i\right)^2} \tag{13}$$

where X_i is the real data and $\hat{X}_i$ is the estimate, n is the group number of the data. "Max" and "Mean" represent the maximum and mean discrepancy between the estimates and the real data. Then the numerical comparison results are listed in **Table 2**. As demonstrated in **Table 2**, the pose estimate accuracy is nearly the same as the proposed method and the ORB-SLAM2, which is due to the introduction of depth filter to optimize the depth of the spatial points during map point location estimation, allowing the position of the map points with higher accuracy. On the other hand, it can improve the pose estimation accuracy while the back-end optimization based on keyframes can enhance the localization precision as well. Therefore, the whole location precision is higher or the same compared to ORB-SLAM2.

4.2. Experiments of Target Detection on Public Data

4.2.1. Image Object Detection

The original YOLOv3 network is used to detect various object types, but only people are required to be detected under this circumstance. In order to mitigate the impact of other objects, the original network is retrained only for the human category. The images selected from the public data set VOC2012 (PASCAL, 2012) are used for comparison between the original YOLOv3 and the modified network, where the involved parameters setting in YOLOv3 are listed in **Table 3**. **Figure 10** displays the detection results where the modified network can well detect the expected objects.

4.2.2. Object Spatial Position Estimation

After the object is detected in the image, it is necessary to associate the object semantic information with the map points to estimate the spatial position of the object and mark it on the map. In this article, the public data set TUM (TUM, 2020) is used to test the object spatial position estimation. The Tum data set is provided by the Technical University of Munich, which includes RGBD, monocular, 3D reconstruction, and other various experimental scenes. In order to meet the requirement of indoor scenes containing the object (people) to be detected, $freiburg2 - desk - with - person$ data set is selected for verification (refer to **Figure 11**) with a resolution of 640 × 480 and average *fps* 28.6fps.

TABLE 3 | The parameters involved in YOLOv3.

Paramter	Value	Parameter	Value
Batch	64	Exposure	1.5
Subdivisions	16	hue	0.1
Width	416	Learning-rate	0.001
Height	416	Burn-in	1,000
Channels	3	Max-batches	50,200
Momentum	0.9	Policy	steps
Decay	0.0005	Steps	40,000, 45,000
Angle	0	Scales	0.1, 0.1
Saturation	1.5		

It is known that the relationship between the object detection frame and the image feature point can be constructed by detecting the specific object (people) in the keyframes by YOLOv3, so as to establish the relationship between the object and the spatial map point and estimate the object spatial position on the map. As shown in the left graph of **Figure 11**, the human object can be detected *via* the YOLOv3 model with the extracted feature points in the keyframe images. The result of the spatial object position estimation is demonstrated in the right graph of **Figure 11**, including the spatial point cloud, camera running trajectory, and the cubic mark of the detected human. Since the direction of the spatial object frame in this article is based

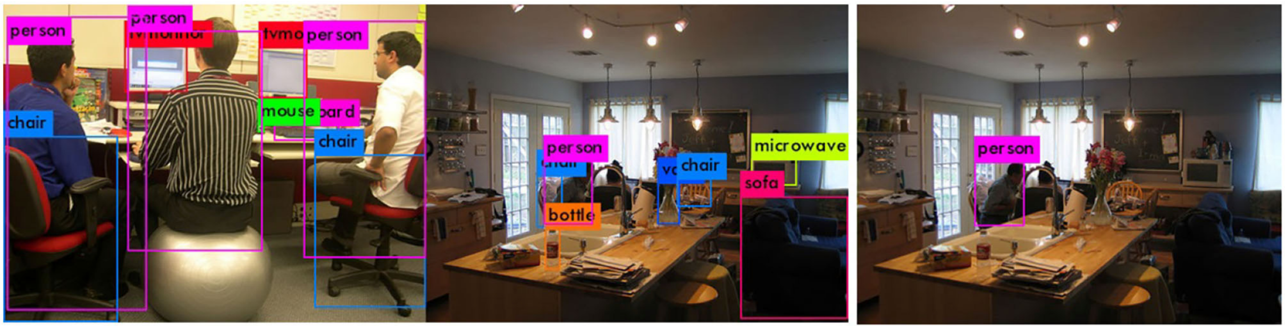


FIGURE 10 | The result of category detection. **(A)** Multi-category detection. **(B)** Single-category detection.

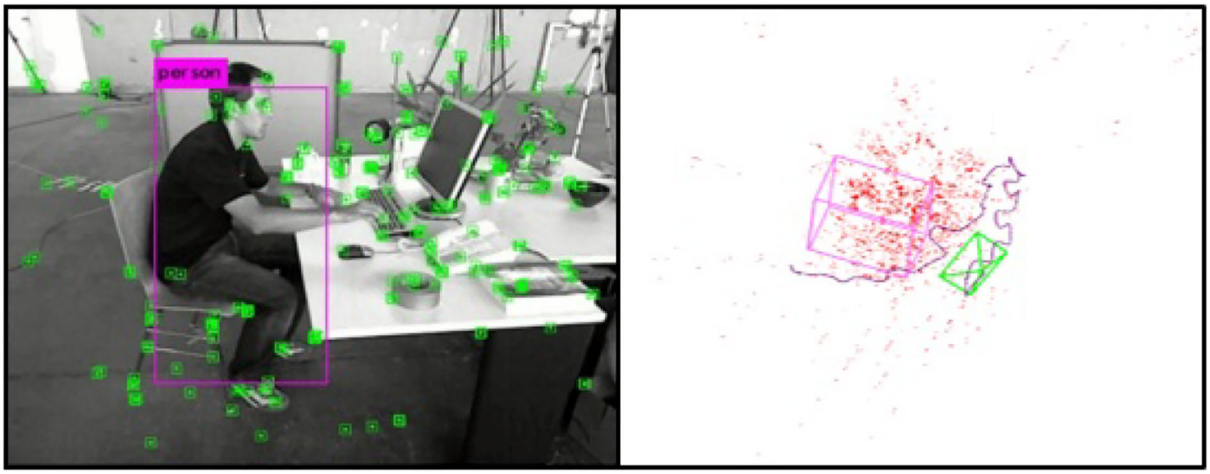


FIGURE 11 | Estimation results of the object spatial position.

TABLE 4 | The performance comparison of the tracking algorithms.

(A) AVERAGE PROCESSING TIME (S) PER IMAGE FRAME			
	Proposed algorithm	**ORB-SLAM2**	
Track 1	**0.0158**	0.0457	
Track 2	**0.0186**	0.0560	
(B) TRACKING ACCURACY COMPARISON(M)			
		Proposed algorithm	
	Max	**Mean**	**RMSE**
Track 1	0.0484	0.0309	0.0326
Track 2	0.0730	0.0433	0.0467

The bold values indicate the indexes of the proposed algorithm.

on the world coordinate system when the camera is initialized, the spatial object frame is not completely consistent with the direction of the image detection frame, but the relative spatial position of the object remains unchanged.

4.3. Field Experiments

The UAV platform in the real world has been well set up for fast localization and objection detection, which is configured with Pixhawk as the controller, together with main processor 32bit STM32F427 Cortex M4, and other embedded sensors. The experimental environment is an indoor laboratory scene, with a length of about 7m and a width of about 6m, mainly including tables, chairs, and several office supplies. The manually set labels are equally distributed on the ground with a 0.5m interval as the positioning reference, shown in **Figure 12A**. In the experiments, the quadrotor UAV equipped with a binocular camera is used as the experimental platform, where the resolution rate is $2,560 \times 720$, the frame processing rate is 30 fps, and the length of the binocular baseline is 60 mm. In order to facilitate the comparison tests, the collected binocular video is saved and converted into a data set, which is analyzed offline.

4.3.1. Indoor Positioning Test

The indoor positioning test is mainly evaluated with the processing time per frame and the positioning accuracy through the collected dataset. The dataset consists of two tracks, and the actual running screenshot is shown in the left graph of **Figure 12B**, while the middle track is the linear motion and the right track is the linear rotary motion.

With these two tracks, the processing rate is compared between the proposed algorithm and ORB-SLAM2. **Table 4A** lists the average processing time per frame under the two track segments. It can be seen that the proposed algorithm has a faster processing speed than that of the ORB-SLAM2 in actual scenes.

In order to acquire the positioning reference of the UAV flight position, ArUco tags with different IDs are equally distributed on the ground at 50 cm distance in the experimental environment (refer to the left graph in **Figure 12C**). The actual position of each ArUco tag in the world coordinate can be obtained by the ID of each ArUco tag, and the captured ArUco image is used to estimate the position of the ArUco tag, so as to estimate the camera position in the world coordinate accordingly.

The image of ArUco tag is captured by a vertically downward high-definition camera installed on the UAV, which has a resolution of $1,920 \times 1,080$ and frame processing rate 60 fps, fixed relative to the binocular camera. It is difficult to obtain the complete trajectory pose due to certain unrecognized phenomena in the moving process. Then the posture of ArUco on the path is used as the baseline for evaluation, while the results listed in **Table 4B** demonstrate that the proposed algorithm can realize real-time localization with high precision.

4.3.2. Object Detection Test

As for the object detection test, a dataset containing the detection object (people) is used for the experiments. The people to be detected are in a still sitting state. Through the UAV moving in the scene, the spatial position of the people can be estimated and marked with a box on the map (refer to **Figure 12C** of the actual running screenshot). Since the object detection is based on the keyframes, only the object detection results and the object position estimation from the keyframes are evaluated. In this experiment, the number of keyframes including the object is 112, and the number of keyframes with confirmed object detection is 88, while the recall rate is 78.6%.

As for the estimation of the object spatial position, the object's handheld ArUco tag is used as the comparison benchmark. With the confirmed keyframes inclusion, the relative position of the tag in the keyframes can be estimated *via* the identification of the handheld ArUco tag, shown as the tag held by the people in the left graph of **Figure 12C**. Whereas in the same keyframe, the relative position between the center of the spatial object frame and the current keyframe can also be obtained, compared with the relative position estimated by the ArUco tag, as shown in the right graph of **Figure 12C**. Then three keyframes are randomly selected where the distance difference between the two estimated positions is used for comparison, while the results of the estimated distance difference (m) of the three keyframes are $\{0.96, 1.04, 0.88\}$, respectively.

Through the evaluation of the detection recall rate and the object spatial position estimation, it can be concluded that the object can be detected effectively with more accurate spatial position estimation in actual scenes, which is suitable for real-time task implementation.

5. CONCLUSION

In this article, a real-time rapid positioning and object detection method based on UAV has been explored with the combination of visual SLAM and CNN techniques. Considering the advantages of feature-based methods with VO, a fast positioning algorithm is proposed where the camera pose can be tracked *via* the front-end VO with only ORB features extracted from the keyframes for the purpose of map consistency improvement *via* bundle adjustment. The feature-based method is also applied at the back-end with the depth filter to assist the depth

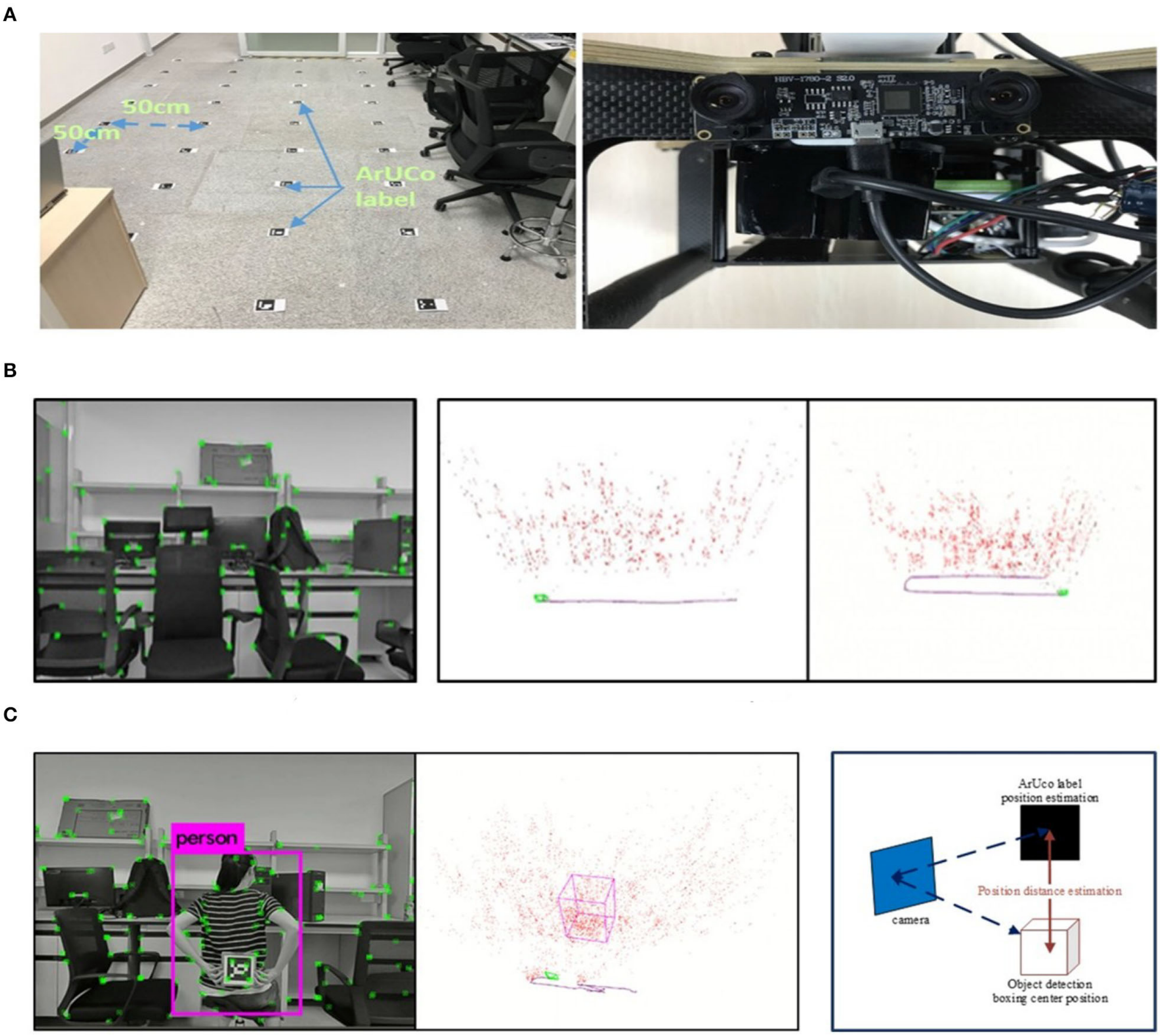


FIGURE 12 | The illustration of the indoor test procedure and results. **(A)** The experimental scene and binocular camera. **(B)**The tracking screenshot in the actual scene. **(C)** (Left): The object detection in the actual scene; (Right): Comparison of object position estimation.

convergence of the map points so as to optimize the framework for positioning accuracy improvement. Furthermore, a spatial target position estimation algorithm has been proposed with the CNN in an unknown space, while the YOLOv3 network is also applied for the target semantic info obtainment in the images so as to construct the relationship between the spatial points and the target. Moreover, the spatial noise can be removed from a statistical outlier filter so as to acquire a clearer target boundary. A series of experiments with public datasets and field tests have been performed to verify the accuracy and portability of the still object localization method with only embedded UAV hardware processor for surveillance or rescuing such task execution, especially in GPS-denied environments.

Future work will continue to study the UAV posture estimation with appropriate semantic segmentation and IMU modules to improve the robustness and accuracy of the UAV fast localization. More efficient signal filtering algorithms could be developed to remove the spatial noise in the key features. Besides, the target attitude estimate should be investigated to increase the localization accuracy since the actual orientation of the target has not been considered. The proposed localization method will be testified in more harsh field experiments.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

ZY and YZ contributed to conception and design of the study. ZY organized the database and performed the main experiments and wrote the first draft of the manuscript. ZY, YZ, and ZM corrected the sections of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This work was supported under National Natural Science Foundation of China (NSFC) (61973296, U1913201), STS project of Chinese Academy of Sciences (KFJ-STS-QYZX-107), Shenzhen Science and Technology Innovation Commission Project Grant (JCYJ20170818153635759), and Shenzhen Fundamental Research Program (JCYJ20200109114839874).

REFERENCES

Ammann, N., and Mayo, L. G. (2018). "Undelayed initialization of inverse depth parameterized landmarks in UKF-SLAM with error state formulation," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (Auckland: IEEE), 918–923.

ASL (2012). *ASL Datasets-The EuRoC MAV Dataset[EB/OL]*. Projects.asl.ethz.ch. Available online at: https://projects.asl.ethz.ch/datasets/

Balabanova, I., Kostadinova, S., and Markova, V., and Georgiev, G. (2020). "Synthesis of prediction models for levels of noises in electrical signals," in *International Conference on Biomedical Innovations and Applications* (Varna: IEEE), 149–152.

Bavle, H., De La Puente, P., and How, J. P. (2020). VPS-SLAM: visual planar semantic SLAM for aerial robotic systems. *IEEE Access* 8, 60704–60718. doi: 10.1109/ACCESS.2020.2983121

Beeharry, Y., and Bassoo, V. (2020). "Performance of ANN and AlexNet for weed detection using UAV-based images," in *International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering* (Balaclava), 163–167.

Bokovoy, A., and Yakovlev, K. (2018). "Sparse 3D point-cloud map upsampling and noise removal as a vSLAM post-processing step: experimental evaluation," in *International Conference on Interactive Collaborative Robotics*, (Leipzig), 23–33.

Bradski, G., and Kaehler, A. (2008). *Learning OpenCV: Computer Vision With the OpenCV library*. Los Angeles, CA: O'Reilly Media, Inc.

Campos, C., Elvira, R., and Rodriguez, J. (2021). ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Trans. Rob.* 37, 1874–1890. doi: 10.1109/TRO.2021.3075644

Cavaliere, D., Loia, V., and Saggese, A. (2017). Semantically enhanced UAVs to increase the aerial scene understanding. *IEEE Trans. Syst. Man Cybern. Syst.* 49, 555–567. doi: 10.1109/TSMC.2017.2757462

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

Chen, J., Lin, G., and Yelamandala, C. (2020). "High-accuracy mapping design based on multi-view images and 3D LiDAR point clouds," in *IEEE International Conference on Consumer Electronics (ICCE)* (Las Vegas, NV: IEEE), 1–2.

Chen, X., Kundu, K., and Zhu, Y. (2017). 3D object proposals using stereo imagery for accurate object class detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 1259–1272. doi: 10.1109/TPAMI.2017.2706685

Cui, L., and Ma, C. (2019). SOF-SLAM: A semantic visual SLAM for dynamic environments. *IEEE Access* 7, 166528–166539. doi: 10.1109/ACCESS.2019.2952161

Forster, C., Pizzoli, M., and Scaramuzza, D. (2014). "SVO: fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong: IEEE), 15–22.

Forster, C., Zhang, Z., and Gassner, M. (2016). SVO: semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Rob.* 33, 249–265. doi: 10.1109/TRO.2016.2623335

Girshick, R. (2015). "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision* (Santiago: IEEE), 1440–1448.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Columbus, OH: IEEE), 580–587.

Jiang, Y., Kang, M., and Fan, M. A. (2018). "Novel relative camera motion estimation algorithm with applications to visual odometry," in *2018 IEEE International Symposium on Multimedia (ISM)* (Taichung: IEEE), 215–216.

Kasyanov, A., Engelmann, F., Stückler, J., and Leibe, B. (2017). "Keyframe-based visual-inertial online SLAM with relocalization," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vancouver, BC: IEEE), 6662–6669.

Kuznetsova, A., and Maleva, T., and Soloviev, V. (2020). "Detecting apples in orchards using YOLOv3 and YOLOv5 in general and close-up images," in *International Symposium on Neural Networks* (Cairo: Springer), 233–243.

Lee, H., and Lee, K., and Kuc, T. (2019). "A new semantic descriptor for data association in semantic SLAM," in *The 19th International Conference on Control, Automation and Systems* (Jeju), 1178–1181.

Li, L., and Liu, Z., Özgner, Ü. (2018). "Dense 3D Semantic SLAM of traffic environment based on stereo vision," in *IEEE Intelligent Vehicles Symposium (IV)* (Changshu: IEEE), 965–970. doi: 10.1109/IVS.2018.8500714

Liu, H., and Ma, H., and Zhang, L. (2019). "Visual odometry based on semantic supervision," in *IEEE International Conference on Image Processing (ICIP)* (Taipei: IEEE), 2566–2570.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., et al. (2016). "Single shot multibox detector," in *European Conference on Computer Vision* (Amsterdam), 21–37.

Liu, Y., Petillot, Y., and Lane, D. (2019). "Global localization with object-level semantics and topology," in *IEEE international Conference on Robotics and Automation* (Montreal, QC: IEEE), 4909–4915.

Mur-Artal, R., Montiel, J., and Tardos, J. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans. Rob*. 31, 1147–1163. doi: 10.1109/TRO.2015.2463671

Nicholson, L., Milford, M., and Snderhauf, N. (2018). Quadricslam: dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Rob. Autom. Lett.* 4, 1–8. doi: 10.1109/LRA.2018.2866205

PASCAL (2012). *The PASCAL Visual Object Classes Challenge 2012[EB/OL]*. Host.robots.ox.ac.uk[2019-12-21]. Available online at: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

Peng, W., Pan, H., and Liu, H. (2020). "IDA-3D: instance-depth-aware 3D object detection from stereo vision for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition* (New York, NY: IEEE), 13015–13024.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 779–788.

Ren, S., He, K., and Girshick, R. (2016). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern. Anal. Mach. Intell.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Su, Y., and Cai, Y. (2018). "Image feature extraction for vision-based uav navigation," in *Chinese Automation Congress* (Xian), 1130–1134.

TUM (2020). *TUM Computer Vision Group - Datasets[EB/OL]*. Vision.in.tum.de[2020-1-28]. Available online at: https://vision.in.tum.de/data/datasets

Vincent, J., Labbé, M., and Lauzon, J. S. (2020). "Dynamic object tracking and masking for visual SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Las Vegas, NV: IEEE), 4974–4979.

Wang, H., Wang, C., and Xie, L. (2021). Intensity-SLAM: intensity assisted localization and mapping for large scale environment. *IEEE Rob. Autom. Lett.* 6, 1715–1721. doi: 10.1109/LRA.2021.3059567

Wang, R., Schworer, M., and Cremers, D. (2017). "Stereo DSO: large-scale direct sparse visual odometry with stereo cameras," in *IEEE International Conference on Computer Vision (ICCV)* (Venice: IEEE), 3903–3911.

Younes, G., Asmar, D., and Zelek, J. (2019). "A unified formulation for visual odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Macau: IEEE), 6237–6244.

Zhang, M., and Li, Z., and Song, Y. (2020). "Optimization and comparative analysis of YOLOV3 target detection method based on lightweight network structure," in *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)* (Dalian: IEEE), 20–24.

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.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

ORIGINAL RESEARCH
published: 18 August 2022
doi: 10.3389/fnbot.2022.945346

Refined Self-Motion Scheme With Zero Initial Velocities and Time-Varying Physical Limits *via* Zhang Neurodynamics Equivalency

Zanyu Tang[1,2,3] *and Yunong Zhang*[1,2*]

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China, [2] Research Institute of Sun Yat-sen University in Shenzhen, Shenzhen, China, [3] College of Information Science and Engineering, Jishou University, Jishou, China

By considering the different-level time-varying physical limits in joint space, a refined self-motion control scheme *via* Zhang neurodynamics equivalency (SMCSvZ) of redundant robot manipulators is proposed, analyzed, and investigated in this manuscript. The SMCSvZ is reformulated as a quadratic program with an equation constraint and a unified bound inequation constraint, which meets the self-motion requirements including the end effector keeping immobile and the initial joint-angle velocities being zero. Simulative verifications based on a six-degrees-of-freedom planar redundant manipulator substantiate the efficacy, accuracy, and superiority of the proposed control scheme, additionally by comparing it with two previous self-motion control schemes. Besides, simulative verifications based on a PUMA560 manipulator are carried out to further verify the availability and correctness of the SMCSvZ.

Keywords: self-motion control scheme, zero initial joint-angle velocities, time-varying physical limits, Zhang neurodynamics equivalency, redundant robot manipulators, quadratic program

Edited by:
Shuai Li,
Swansea University, United Kingdom

Reviewed by:
Vasilios N. Katsikis,
National and Kapodistrian University of Athens, Greece
Dechao Chen,
Sun Yat-sen University, China
Long Jin,
Lanzhou University, China

****Correspondence:***
Yunong Zhang
zhynong@mail.sysu.edu.cn

Received: *16 May 2022*
Accepted: *20 June 2022*
Published: *18 August 2022*

Citation:
Tang Z and Zhang Y (2022) Refined Self-Motion Scheme With Zero Initial Velocities and Time-Varying Physical Limits via Zhang Neurodynamics Equivalency.
Front. Neurorobot. 16:945346.
doi: 10.3389/fnbot.2022.945346

1. INTRODUCTION

Redundant robot manipulators refer to such kind of manipulators whose degrees of freedom (DoF) are more than the minimum number of DoF needed to perform specific end-effector tasks (Zhang et al., 2018; Liao et al., 2019; Zhou et al., 2019; Chen et al., 2020; Xiao et al., 2020; Zhao et al., 2020; Jin et al., 2021). Therefore, they have the capability to meet additional requirements, e.g., satisfying physical limits, avoiding obstacles, and avoiding singularity configurations. In the practical application, the redundant robot manipulator needs to adjust its configuration in some peculiar situations. For instance, the repetitive motion of the redundant robot manipulator is planned but joint-angle drift may happen. Similarly, the end-effector task may not be completed because of operating space limitations or manipulator physical limitations. Adjusting the manipulator configuration from one state to another state is essential and important for redundant robot manipulators (Jin et al., 2020). Thereinto, the self-motion of redundant robot manipulators is to adjust the manipulator configuration from the initial state to final state keeping the end effector immobile at its current position or orientation (Li and Zhang, 2012; Zhang et al., 2021a).

The self-motion could result in better manipulator performance such as manipulability improvement (Jin et al., 2021), end-effector task completeness, and singularity configuration avoidance (Pardi et al., 2020).

In recent years, many self-motion control schemes (SMCSs) or self-motion control approaches have been developed (Zhang et al., 2009, 2016, 2021a; Li and Zhang, 2012; Zhang and Xiao, 2012; Gong et al., 2019). For instance, a self-motion scheme in form of a quadratic program (QP) was presented in Zhang et al. (2009) and Liao et al. (2021), which was verified on the functionally redundant robot manipulator PUMA560 considering joint-angle limits and joint-angle-velocity limits. With singularities discussed, Zhang and Xiao (2012) proposed a QP-based self-motion scheme for manipulators compared with the pseudoinverse method and substantiated that the proposed scheme was effective on three kinds of manipulators. To eliminate the abrupt increase in joint velocity at the beginning of the self-motion task execution, Li and Zhang (2012) put forward a zero-initial-velocity self-motion scheme and verified its feasibility on a 6-DoF planar manipulator. Besides, in order to achieve high efficient self-motion tasks, Zhang et al. (2021a) put forth a varying-gain neural self-motion approach.

In many previous studies, researchers developed QP-based SMCSs considering time-invariant physical limits. However, some redundant manipulators are inherently subject to varying physical limits (Li and Zhang, 2012). Besides, in the engineering field, with the passage of time and mechanical wear, the physical limits may change with time, i.e., they are time-varying. Considering this case, we establish a refined SMCS for the redundant manipulators *via* the equivalency method in this article.

Zhang neurodynamics equivalency (ZNDE) inspired by Ma equivalency (Ma, 1996; Ma et al., 1996), is actually a class of practice-accepted approximation, which is derived from Zhang neurodynamics (Chen and Zhang, 2018; Qin et al., 2021). Some schemes of complex systems *via* the ZNDE approach were efficiently simplified (Guo et al., 2013; Qiu et al., 2016, 2018). Minimum-velocity-norm schemes of redundant robot manipulators at two different layers were established by Guo et al. (2013), and the equivalent relationship between two manipulator control schemes was also developed *via* the Zhang neurodynamics method. Zhang et al. (2020b) substantiated that the schemes of redundant robot manipulators formulated by the ZNDE approach were more robust. Zhang et al. (2020a) tried to solve complex inequality-related problems through the ZNDE approach. Besides, Zhang et al. (2021b) proposed a cyclic motion control scheme at the acceleration layer for manipulator systems *via* the ZNDE approach.

In this article, a refined self-motion control scheme *via* the ZNDE approach (named SMCSvZ) in form of standard QP is proposed to solve self-motion problems. With the time-varying physical limits considered and zero initial joint-angle velocities ensured, the proposed SMCSvZ is developed, proved, and obtained by the ZNDE theorem and the corresponding corollary. The simulation experiments based on two different robot manipulators are designed and carried out to substantiate the correctness and superiority of the proposed scheme by comparing the previous SMCSs. The remainder of this article consists of five sections. In Section 2, the requirements of self-motion problems are presented first. Then, by analyzing and comparing the two previous SMCSs, the refined SMCSvZ is proposed *via* the ZNDE approach. In Section 3, the derivation process of the SMCSvZ is provided, and the feasibility and availability of the SMCSvZ are analyzed theoretically. In Section 4, the SMCSvZ composed of performance index, equivalent equation constraint, and bounded inequality constraint is presented in form of standard QP formulation, and its corresponding neural network solver is also shown. In Section 5, the simulation experiments based on a 6-DoF planar manipulator are carried out, and the simulation results substantiate the efficacy and superiority of the SMCSvZ. Moreover, the simulation experiments based on a PUMA560 manipulator are also carried out to further verify the availability and correctness of the SMCSvZ. Finally, we conclude the paper in Section 6. The main contributions of the current study are presented as follows.

1. To better meet self-motion requirements, a refined self-motion control scheme of redundant robot manipulators is proposed with time-varying physical limits and zero initial joint-angle velocities considered.
2. The theorem and corollary of the ZNDE approach are proposed and theoretically proved, through which the refined SMCSvZ is obtained. Then, the SMCSvZ is applied to redundant manipulators to effectively realize the self-motion task.
3. By comparing the SMCSvZ with the two previous SMCSs, the simulation experiments based on a 6-DoF planar redundant manipulator and a PUMA560 manipulator are carried out with physical limits fully satisfied, which verifies the availability, effectiveness, and superiority of the proposed SMCSvZ.

2. PRELIMINARY, PROBLEM, AND SCHEMES

The forward-kinematics equation of redundant robot manipulators is written as $\mathbf{r} = F(\Theta)$, where $\mathbf{r} \in \mathbb{R}^m$ denotes the end-effector actual position with $\Theta \in \mathbb{R}^n$ denoting the joint-angle vector and $F(\cdot)$ being a differentiable nonlinear function. Furthermore, the inverse-kinematics equation about relationship between the derivative of end-effector position vector $\dot{\mathbf{r}} \in \mathbb{R}^m$ and the derivative of joint-angle-velocity vector $\dot{\Theta} \in \mathbb{R}^n$ is written as

$$J(\Theta)\dot{\Theta} = \dot{\mathbf{r}},$$

where $J(\Theta) = \partial F(\Theta)/\partial\Theta \in \mathbb{R}^{m\times n}$ is the Jacobian matrix.

In essence, the self-motion task of the redundant robot manipulator is to utilize the redundant DoF of the manipulator to adjust its configuration in joint space with the end effector being immobile. For guaranteeing the end effector is immobile, the self-motion task can be completed with the given joint angles being in the motion region. If the given joint angles are out of the motion region, the redundant robot manipulators also try to adjust the configuration to a suitable state. As a result, the manipulators become more flexible after the self-motion task. To execute the self-motion task, the SMCS of redundant robot manipulators needs to meet the following requirements, which constitute the problem formulation. ① Robot manipulators adjust joint angles $\Theta(0)$ to given joint angles Θ_{g} that are suitable joint angles within the workspace of redundant robots (Akli, 2021). ② Robot manipulators try to keep the end effector immobile during the process of self motion. ③ The initial velocities of the joint angles equal zero. ④ The final velocities of the joint angles equal zero. ⑤ Time-varying physical limits (including joint-angle layer and joint-angle-velocity layer limits) of redundant robot manipulators are all satisfied. Accordingly, we depict those requirements as

$$\Theta(t) \to \Theta_{\text{g}}, \tag{1}$$

$$F(\Theta(t)) \to F(\Theta(0)), \tag{2}$$

$$\dot{\Theta}(0) = \mathbf{0}, \tag{3}$$

$$\dot{\Theta}(t_{\text{end}}) = \mathbf{0}, \quad \text{with} \quad t_{\text{end}} \in [0, t_{\text{f}}] \tag{4}$$

$$\mathbf{l}_0^-(t) \le \Theta(t) \le \mathbf{l}_0^+(t), \tag{5}$$

$$\mathbf{l}_1^-(t) \le \dot{\Theta}(t) \le \mathbf{l}_1^+(t), \tag{6}$$

where $\Theta(t) \in \mathbb{R}^n$ and $\dot{\Theta}(t) \in \mathbb{R}^n$ denote the joint-angle vector and joint-angle-velocity vector, respectively; Θ_{g} is the given joint-angle vector; $\Theta(0)$ is the initial joint-angle vector; $\dot{\Theta}(t)$ denotes the derivative of $\Theta(t)$ with time instant $t \in [0, t_{\text{f}}]$ and t_{f} denoting last instant time of the self-motion duration; $\dot{\Theta}(0)$ denotes $\dot{\Theta}(t)$ with $t = 0$; $\dot{\Theta}(t_{\text{end}})$ denotes $\dot{\Theta}(t)$ with t_{end} being the end time of the self-motion task. In addition, $\mathbf{l}_0^-(t)$ and $\mathbf{l}_0^+(t)$ represent the time-varying joint-angle lower limit and upper limit, respectively; $\mathbf{l}_1^-(t)$ and $\mathbf{l}_1^+(t)$ represent the joint-angle-velocity lower limit and upper limit, respectively. The traditional SMCS at the joint-angle-velocity layer is formulated in Zhang et al. (2009) as

$$\begin{aligned}
\text{minimize} \quad & \|\dot{\Theta}(t) + \mathbf{q}(t)\|_2^2/2, && (7)\\
\text{subject to} \quad & J(\Theta(t))\dot{\Theta}(t) = \mathbf{0}, && (8)\\
& {}^1\mathbf{l}^- \le \dot{\Theta}(t) \le {}^1\mathbf{l}^+, && (9)\\
\text{with} \quad & \mathbf{q}(t) = \mu(\Theta(t) - \Theta\text{g}), && (10)\\
& {}^1\mathbf{l}^- = \max\{\kappa(\mathbf{l}_0^- - \Theta(t)), \mathbf{l}_1^-\}, && (11)\\
& {}^1\mathbf{l}^+ = \min\{\kappa(\mathbf{l}_0^+ - \Theta(t)), \mathbf{l}_1^+\}, && (12)
\end{aligned}$$

where symbol $\|\cdot\|_2$ denotes the two-norm of the vector, and the time-varying vector $\mathbf{q}(t) \in \mathbb{R}^n$ is defined according to the self-motion task. The design parameters $\mu > 0 \in \mathbb{R}$ and $\kappa > 0 \in \mathbb{R}$ are used to scale the magnitude of the manipulators. The "max" and "min" functions are used to obtain the maximum and minimum values of elements in the vector, respectively. The other parameters are the same as those of the requirements (1)–(6). We name the scheme (7)–(12) as SMCS-1 in this article.

As a further research of Zhang et al. (2009), a zero-initial-velocity self-motion scheme for redundant robot manipulators is proposed in Li and Zhang (2012) as shown below:

$$\begin{aligned}
\text{minimize} \quad & \|\dot{\Theta}(t) + \mathbf{q}(t)\|_2^2/2, && (13)\\
\text{subject to} \quad & J(\Theta(t))\dot{\Theta}(t) = \mathbf{0}, && (14)\\
& {}^2\mathbf{l}^- \le \dot{\Theta}(t) \le {}^2\mathbf{l}^+, && (15)\\
\text{with} \quad & \mathbf{q}(t) = \mu(\Theta(t) - \Theta\text{g}), && (16)\\
& {}^2\mathbf{l}^- = \max\{\kappa(\mathbf{l}_0^- - \Theta(t)), \sin(\pi t/(2\text{t}_{\text{f}}))\mathbf{l}_1^-\}, && (17)\\
& {}^2\mathbf{l}^+ = \min\{\kappa(\mathbf{l}_0^+ - \Theta(t)), \sin(\pi t/(2\text{t}_{\text{f}}))\mathbf{l}_1^+\}, && (18)
\end{aligned}$$

where the physical limits (${}^2\mathbf{l}^+$ and ${}^2\mathbf{l}^-$) are partly different from ${}^1\mathbf{l}^{\pm}$ presented in (17) and (18), which ensure $\dot{\Theta}(0) = \mathbf{0}$. The other parameters are the same as those of SMCS-1. The scheme (13)–(18) is named SMCS-2 in this article.

However, equation limits (8) in SMCS-1 and (14) in SMCS-2 are difficult to realize in practice. Different from SMCS-1 and SMCS-2, with comprehensive consideration of continuously and differentially time-varying physical limits, zeroing initial joint-angle velocities, and dynamically keeping the end-effector position immobile, we propose a refined SMCSvZ for redundant robot manipulators in this article, which is formulated as

$$\begin{aligned}
\text{minimize} \quad & \|\dot{\Theta}(t) + \mathbf{q}(t)\|_2^2/2, && (19)\\
\text{subject to} \quad & J(\Theta(t))\dot{\Theta}(t) = -\mu_1(F(\Theta(t)) - F(\Theta(0))), && (20)\\
& {}^3\mathbf{l}^- \le \dot{\Theta} \le {}^3\mathbf{l}^+, && (21)\\
\text{with} \quad & \mathbf{q}(t) = \mu_2 t(\Theta(t) - \Theta\text{g}), && (22)\\
& {}^3\mathbf{l}^-(t) = \max\{\dot{\mathbf{l}}_0^-(t) + \kappa(\mathbf{l}_0^-(t) - \Theta(t)), \mathbf{l}_1^-(t)\}, && (23)\\
& {}^3\mathbf{l}^+(t) = \max\{\dot{\mathbf{l}}_0^+(t) + \kappa(\mathbf{l}_0^+(t) - \Theta(t)), \mathbf{l}_1^+(t)\}, && (24)
\end{aligned}$$

where the positive design parameters μ_1, μ_2, and κ are used to scale the magnitude of the manipulators. In addition, $\mathbf{l}_0^-(t)$, $\mathbf{l}_0^+(t)$, $\mathbf{l}_1^-(t)$, and $\mathbf{l}_1^+(t)$ are the same as those of SMCS-1; $\dot{\mathbf{l}}_0^-(t)$ and $\dot{\mathbf{l}}_0^+(t)$ represent the derivatives of $\mathbf{l}_0^-(t)$ and $\mathbf{l}_0^+(t)$, respectively.

3. SMCSvZ DERIVATION AND ANALYSIS

In this section, the performance index, the equation constraint, and the unified bound inequation constraint in the QP-based SMCSvZ are deduced *via* the ZNDE approach. The theorem and corollary are given and proved for the analysis of the SMCSvZ.

3.1. Equation Constraint *via* ZNDE

In this subsection, the equivalence analyses of equations in SMCSvZ are carried out. To be specific, (19), (20), and (22) in SMCSvZ are derived and analyzed theoretically.

To ensure that initial joint-angle velocities are zero, physical limits (${}^2\mathbf{l}^{\pm}$) in SMCS-2 are different from ${}^1\mathbf{l}^{\pm}$ in SMCS-1, and one part of ${}^2\mathbf{l}^{\pm}$ is obtained through multiplying $\sin(\pi t/(2\text{t}_{\text{f}}))$ by ${}^1\mathbf{l}^{\pm}$. These changes realize $\dot{\Theta}(0) = \mathbf{0}$ but reduce the feasible region of $\dot{\Theta}(t)$. That is when the physical limits verge, $\dot{\Theta}(t)$ change to avoid exceeding the physical limits, but they can only change slowly and thus make the task spend more time. In SMCSvZ, we define (22) instead of (10) because the Equation (22) can better meet the requirements of self-motion tasks and it is practically and mathematically equivalent to Equation (10), which is proved *via* the following ZNDE-EEV theorem.

Theorem 1. *(ZNDE-EEV theorem) With differentiable $\epsilon(t) \in \mathbb{R}^n$, the zero vector $\mathbf{0} \in \mathbb{R}^n$, sufficiently large positive design parameter $\mu \gg 0$ and time instant $t \gg 0$,*

$$\dot{\epsilon}(t) = -\mu t \epsilon(t) \tag{25}$$

is practically mathematically equivalent (i.e., ZNDE equivalent) to

$$\epsilon(t) = \mathbf{0}. \tag{26}$$

Proof: The Equation (25) is a differential equation with $\dot{\epsilon}(t)$ denoting $\mathrm{d}(\epsilon(t))/\mathrm{d}t$. The analytical solution of (25) is $\epsilon(t) = \epsilon(0)\exp(-\mu t^2/2)$ with $\epsilon(0)$ denoting the initial value of $\epsilon(t)$ and μ denoting a large positive design parameter.

With $t \to \infty$, $\epsilon(t) = \epsilon(0)\exp(-\mu t^2/2) \to \mathbf{0}$ instantaneously. That is, each element of $\epsilon(t)$ quickly decreases to a tiny value that is considered to be zero in practical application, or $\epsilon(t) = \mathbf{0}$ with t large enough. Therefore, (25) is ZNDE equivalent to (26) with $\mu \gg 0$ and $t \gg 0$. The proof, thus, ends. ■

To settle the self-motion problem of redundant robot manipulators, we combine the self-motion requirement (1) and define an error function as $\epsilon(t) = \Theta(t) - \Theta_{\mathrm{g}}$. Then, according to Theorem 1, one gets that

$$\dot{\epsilon}(t) = \dot{\Theta}(t) = -\mu t(\Theta(t) - \Theta_{\mathrm{g}}) \tag{27}$$

is ZNDE equivalent to $\Theta(t) - \Theta_{\mathrm{g}} = \mathbf{0}$. From (27), one can get the following. First, the performance index at the velocity layer can be expressed as $\|\dot{\Theta}(t) + \mu t(\Theta(t) - \Theta_{\mathrm{g}})\|_2^2/2$, i.e., $\mathbf{q}(t) = \mu t(\Theta(t) - \Theta_{\mathrm{g}})$, which corresponds to (19) and (22) in SMCSvZ. Second, the self-motion task ends when the value of $\Theta(t_{\mathrm{end}}) - \Theta_{\mathrm{g}}$ equals $\mathbf{0}$ with $t_{\mathrm{end}} \in [0, t_{\mathrm{f}}]$. That is when the final velocities of joint angles $\dot{\Theta}(t_{\mathrm{end}})$ verge or equal $\mathbf{0}$, the self-motion task is considered to be over. Finally, when $t = 0$, the initial joint-angles velocity vector $\dot{\Theta}(0)$ equals $\mathbf{0}$. The above analyses show that (22) in SMCSvZ obtained by the ZNDE better meets the requirements (1), (3), and (4) of the self-motion task.

In addition, the self-motion requirement (2) is formulated as $J(\Theta(t))\dot{\Theta}(t) = \mathbf{0}$ in SMCS-1 and SMCS-2, which is difficult to guarantee in practice. Hence, we handle the problem by transforming this constraint to (20) in SMCSvZ *via* the ZNDE approach, which is illustrated clearly by the following lemma (Zhang et al., 2021b).

Lemma 1. *With differentiable $\epsilon(t) \in \mathbb{R}^n$, the zero vector $\mathbf{0} \in \mathbb{R}^n$, sufficiently large positive design parameter $\mu \gg 0$ and time instant $t \gg 0$, $\dot{\epsilon}(t) = -\mu\epsilon(t)$ is practically mathematically equivalent (i.e., ZNDE equivalent) to $\epsilon(t) = \mathbf{0}$.*

To meet the self-motion requirement (2), the error function is defined as $\epsilon(t) = \mathcal{F}(\Theta(t)) - \mathcal{F}(\Theta(0))$. By Lemma 1, we get that the equation $J(\Theta)\dot{\Theta}(t) = -\mu(\mathcal{F}(\Theta(t)) - \mathcal{F}(\Theta(0)))$ is ZNDE equivalent to (8) in SMCS-1 and (14) in SMCS-2, which is just equation constraint (20) in SMCSvZ. Meanwhile, equivalent equation constraint (20) in SMCSvZ dynamically keeps the end effector nearest to its initial position.

3.2. Inequation Constraint *via* ZNDE

In this subsection, we are to unify two-layer inequation constraints into one equivalent bound inequation constraint through the inequation type of the ZNDE that is described in the following lemma (Zhang et al., 2021b).

Lemma 2. *With differentiable $\epsilon(t) \in \mathbb{R}^n$, sufficiently large positive design parameter $\rho \gg 0$ and time instant $t \gg 0$, in a ZND manner, $\dot{\epsilon}(t) \leq -\rho\epsilon(t)$ is practically mathematically equivalent (i.e., ZNDE equivalent) to $\epsilon(t) \leq \mathbf{0}$.*

According to Lemma 2, the following corollary at the velocity layer is acquired.

Corollary 1. *Assume that vector $\Theta(t)$ and its time-varying physical limits $\mathbf{l}_0^{\pm}(t)$ are continuously differentiable. $\dot{\mathbf{l}}_0^-(t)$ and $\dot{\mathbf{l}}_0^+(t)$ represent the derivatives of $\mathbf{l}_0^-(t)$ and $\mathbf{l}_0^+(t)$, respectively. With design parameter $\rho \gg 0$ and time $t \gg 0$, in a ZND manner,*

$$\dot{\mathbf{l}}_0^-(t) - \rho(\Theta(t) - \mathbf{l}_0^-(t)) \leq \dot{\Theta}(t) \leq \dot{\mathbf{l}}_0^+(t) - \rho(\Theta(t) - \mathbf{l}_0^+(t)) \tag{28}$$

is practically mathematically equivalent (i.e., ZNDE equivalent) to

$$\mathbf{l}_0^-(t) \leq \Theta(t) \leq \mathbf{l}_0^+(t). \tag{29}$$

Proof: By defining the function $\epsilon(t) = \Theta(t) - \mathbf{l}_0^+(t) \leq \mathbf{0}$ according the left part of (29), one gets $\dot{\epsilon}(t) = \dot{\Theta}(t) - \dot{\mathbf{l}}_0^+(t) \leq -\rho(\Theta(t) - \mathbf{l}_0^+(t))$, which is ZNDE equivalent to $\Theta(t) \leq \mathbf{l}_0^+(t)$ *via* Lemma 2. Then, $\dot{\Theta}(t) \leq \dot{\mathbf{l}}_0^+(t) - \rho(\Theta(t) - \mathbf{l}_0^+(t))$ is obtained.

Similarly, by defining the function $\epsilon(t) = \mathbf{l}_0^-(t) - \Theta(t) \leq \mathbf{0}$ according the right part of (29), one gets $\dot{\epsilon}(t) = \dot{\mathbf{l}}_0^-(t) - \dot{\Theta}(t) \leq -\rho(\mathbf{l}_0^-(t) - \Theta(t))$, which is ZNDE equivalent to $\mathbf{l}_0^-(t) \leq \Theta(t)$ *via* Lemma 2. Then, $\dot{\mathbf{l}}_0^-(t) + \rho(\mathbf{l}_0^-(t) - \Theta(t)) \leq \dot{\Theta}(t)$ is obtained. Combined with the above results, the corollary is proved. ■

From the above corollary, (28) is ZNDE equivalent to the self-motion requirement (5). By combining (28) and the self-motion requirement (6), the unified equivalent bound inequation constraint (21) in SMCSvZ is obtained.

4. QP FORMULATION AND PROJECTION NEURAL NETWORK (PNN) SOLVER

By using the ZNDE approach, the QP-based SMCSvZ is obtained to control the redundant robot manipulators for realizing the self-motion task, which is handled by a projection neural network (PNN) solver.

4.1. Standard QP Formulation

On the basis of Theorem 1, Lemma 1, and Corollary 1, the SMCSvZ with time-varying physical limits satisfied is reformulated as a standard QP at the velocity layer as follows.

$$\text{minimize} \quad \frac{1}{2}\Upsilon^{\mathrm{T}}(t)A(t)\Upsilon(t) + \mathbf{p}^{\mathrm{T}}(t)\Upsilon(t), \tag{30}$$

$$\text{subject to} \quad B(t)\Upsilon(t) = \mathbf{b}(t), \tag{31}$$

$$\mathbf{l}^-(t) \leq \Upsilon(t) \leq \mathbf{l}^+(t), \tag{32}$$

$$\text{with} \quad \mathbf{l}^-(t) = \max\{\dot{\mathbf{l}}_0^-(t) + \kappa(\mathbf{l}_0^-(t) - \Theta(t)), \mathbf{l}_1^-(t)\}, \tag{33}$$

$$\mathbf{l}^+(t) = \min\{\dot{\mathbf{l}}_0^+(t) + \kappa(\mathbf{l}_0^+(t) - \Theta(t)), \mathbf{l}_1^+(t)\}, \tag{34}$$

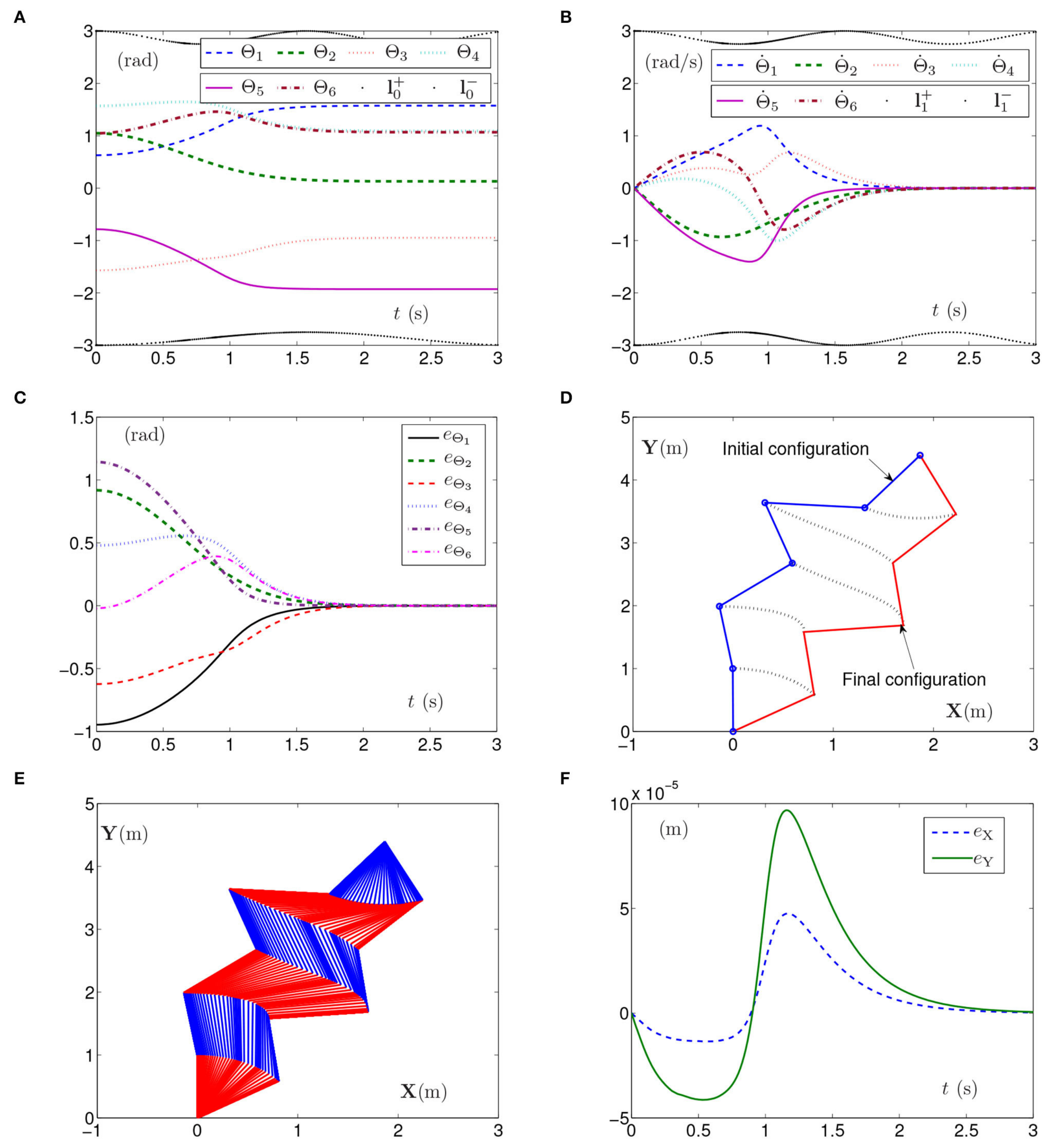


FIGURE 1 | Synthesized results of the planar manipulator using SMCSvZ with time-varying physical limits satisfied in case A. **(A)** Profiles of joint angles. **(B)** Profiles of joint-angle velocities. **(C)** Profiles of joint-angle errors. **(D)** Profiles of initial and final manipulator positions. **(E)** Profiles of the planar manipulator. **(F)** Profiles of end-effector position errors.

where $\Upsilon(t) = \dot{\Theta}(t)$; $A(t) = I_n$ denotes an $n \times n$ identity matrix; $B(t) = J(\Theta(t))$; $\mathbf{p}(t) = \mu_1 t(\Theta(t) - \Theta(0))$; $\mathbf{b}(t) = -\mu_2(F(\Theta(t)) - F(\Theta(0)))$ with μ_1 and μ_2 presenting the design parameters. Moreover, $\mathbf{l}^+(t)$ and $\mathbf{l}^-(t)$ are the physical upper and lower limits of synthesized time-varying unified layer, respectively. $\mathbf{l}_0^-(t)$,

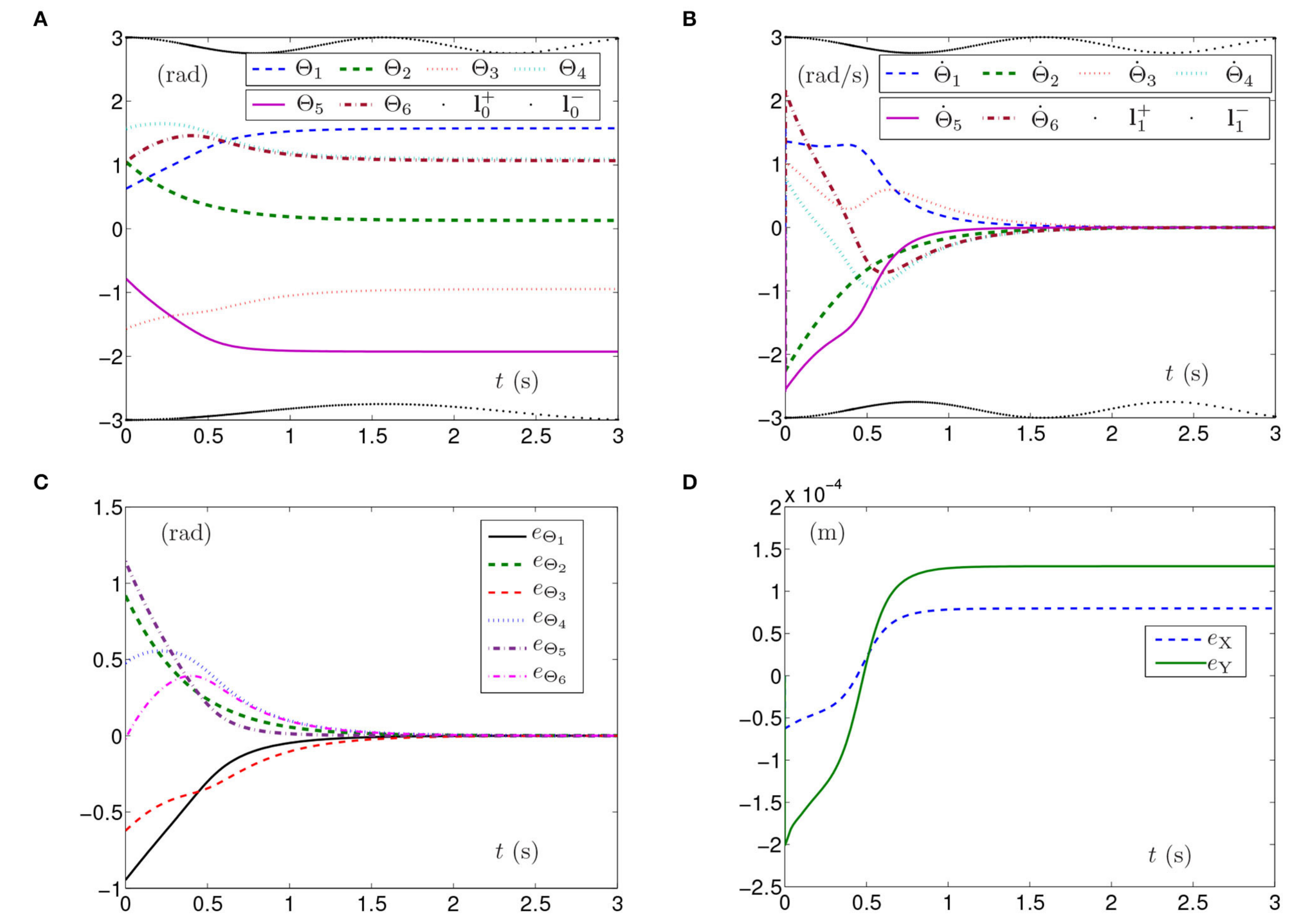


FIGURE 2 | Synthesized results of the planar manipulator using SMCS-1 with time-varying physical limits satisfied in case A. **(A)** Profiles of joint angles. **(B)** Profiles of joint-angle velocities. **(C)** Profiles of joint-angle errors. **(D)** Profiles of end-effector position errors.

$\mathbf{l}_0^+(t)$, $\mathbf{l}_1^-(t)$, $\mathbf{l}_0^+(t)$, $\dot{\mathbf{l}}_0^-(t)$, and $\dot{\mathbf{l}}_0^+(t)$ are the same as above SMCSvZ (19)–(24).

4.2. PNN Solver

To solve the QP-based SMCSvZ (30)–(34) in real time, we use a PNN solver to obtain the solution $\Upsilon(t)$ in this subsection, which is developed in the following lemma (Xia et al., 2020).

Lemma 3. *With $\gamma \in \mathbb{R}^+$ adjusting the convergence rate and large enough ς, the PNN solver for SMCSvZ is developed as*

$$\dot{\mathbf{u}}(t) = \gamma\left(I_{n+m} + M^{\mathrm{T}}(t)\right)\left(P_\Omega(\mathbf{u}(t) - (M(t)\mathbf{u}(t) + \mathbf{h}(t))) - \mathbf{u}(t)\right), \tag{35}$$

where $\mathbf{u}(t) = [\Upsilon(t); \varpi] \in \mathbb{R}^{n+m}$ and $\mathbf{h}(t) = [\mathbf{p}(t); -\mathbf{b}(t)] \in \mathbb{R}^{n+m}$ in MATLAB manner (Mathews and Fink, 2004). Meanwhile, I_{n+m} denotes a $(n+m)\times(n+m)$ identity matrix, and $\varpi \in \mathbb{R}^m$ is the dual decision vector defined corresponding to (31). Besides,

$$M(t) = \begin{bmatrix} A(t) & -B^{\mathrm{T}}(t) \\ B(t) & O_m \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)},$$

$$\mathbf{u}^-(t) = \begin{bmatrix} \mathbf{l}^-(t) \\ -\varsigma\mathbf{1}_{\mathrm{v}} \end{bmatrix} \in \mathbb{R}^{n+m}, \text{and } \mathbf{u}^+(t) = \begin{bmatrix} \mathbf{l}^+(t) \\ \varsigma\mathbf{1}_{\mathrm{v}} \end{bmatrix} \in \mathbb{R}^{n+m},$$

in which O_m denotes an $m \times m$ zero matrix and $\mathbf{1}_{\mathrm{v}} = [1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^m$.

5. SIMULATIONS AND COMPARISONS

In this section, the simulation experiments are conducted based on two different redundant robot manipulators, which include a 6-DoF planar manipulator and a PUMA560 manipulator. Thereinto, the PUMA560 manipulator works in three-dimensional space.

5.1. Simulations Based on 6-DoF Planar Manipulator

In the ensuing simulations, the initial joint states of the planar manipulator are set as $[0.628, 1.047, -1.570, 1.570, -0.785, 1.047]^{\mathrm{T}}$ rad with superscript $^{\mathrm{T}}$ denoting the transposition of the vector, and the given joint

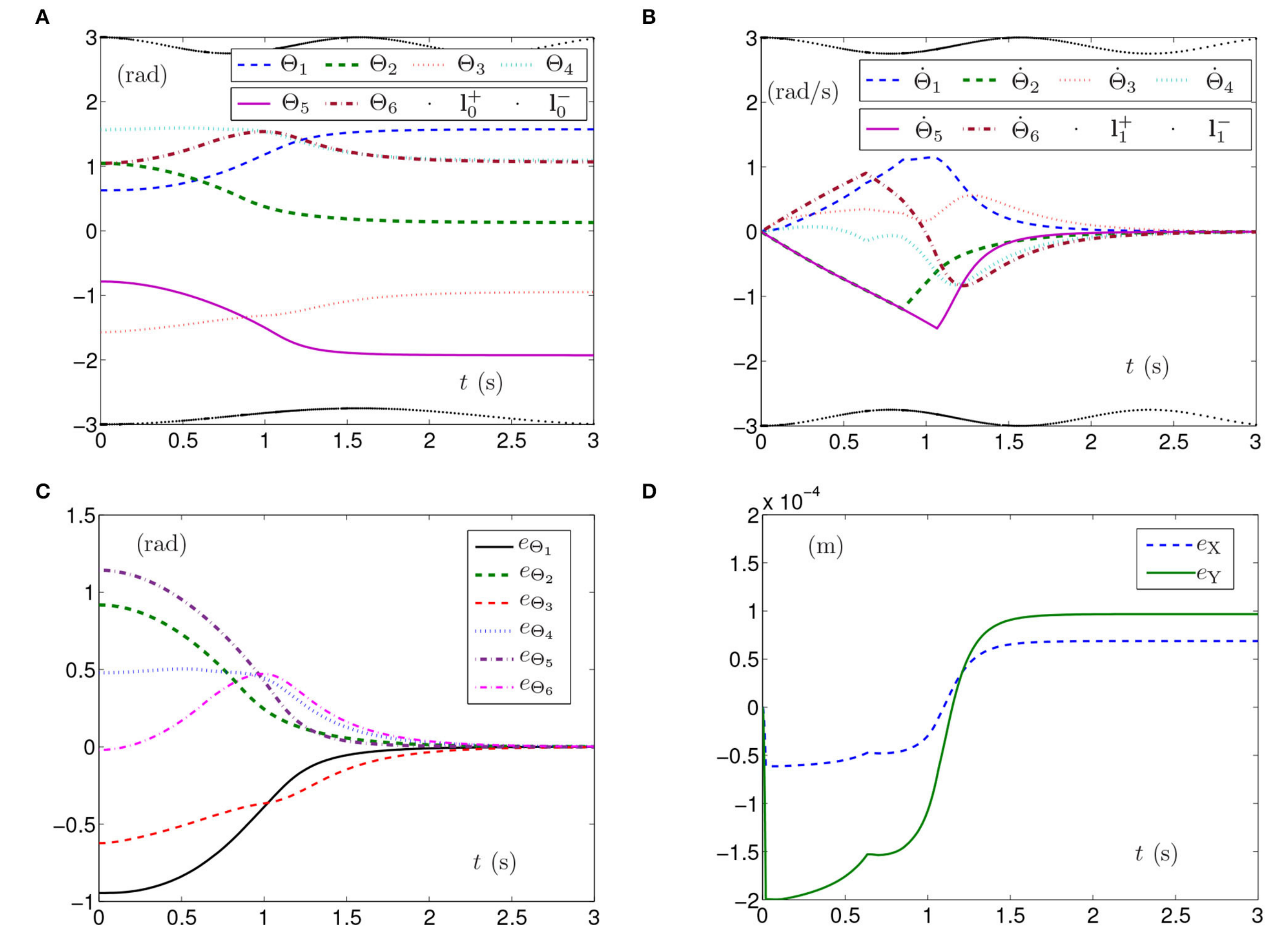


FIGURE 3 | Synthesized results of the planar manipulator using SMCS-2 with time-varying physical limits satisfied in case A. **(A)** Profiles of joint angles. **(B)** Profiles of joint-angle velocities. **(C)** Profiles of joint-angle errors. **(D)** Profiles of end-effector position errors.

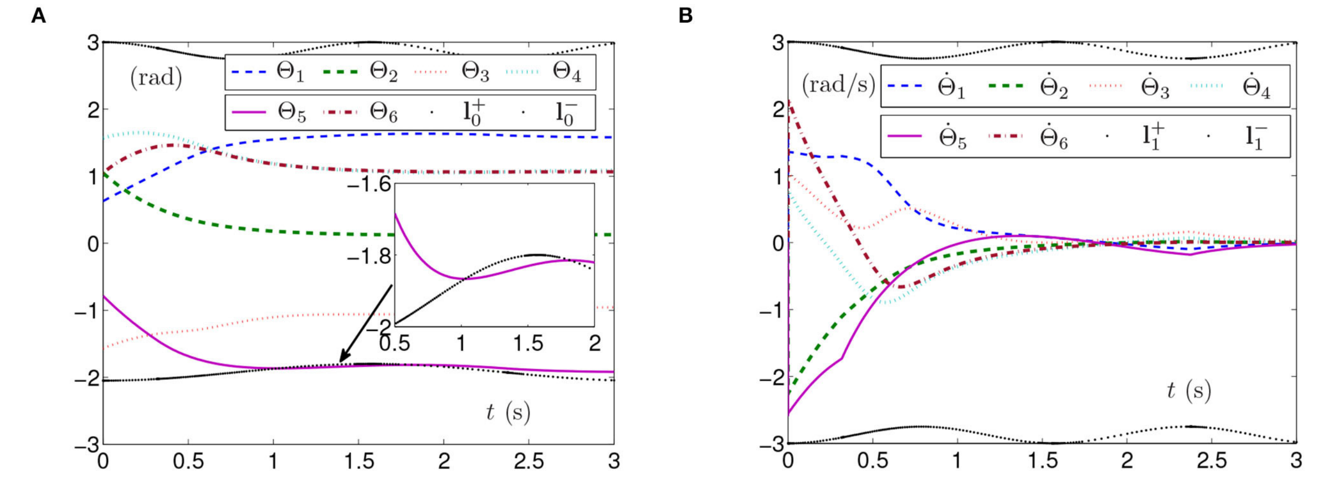


FIGURE 4 | Synthesized results of the planar manipulator using SMCS-1 with time-varying physical limits unsatisfied in case B. **(A)** Profiles of joint angles. **(B)** Profiles of joint-angle velocities.

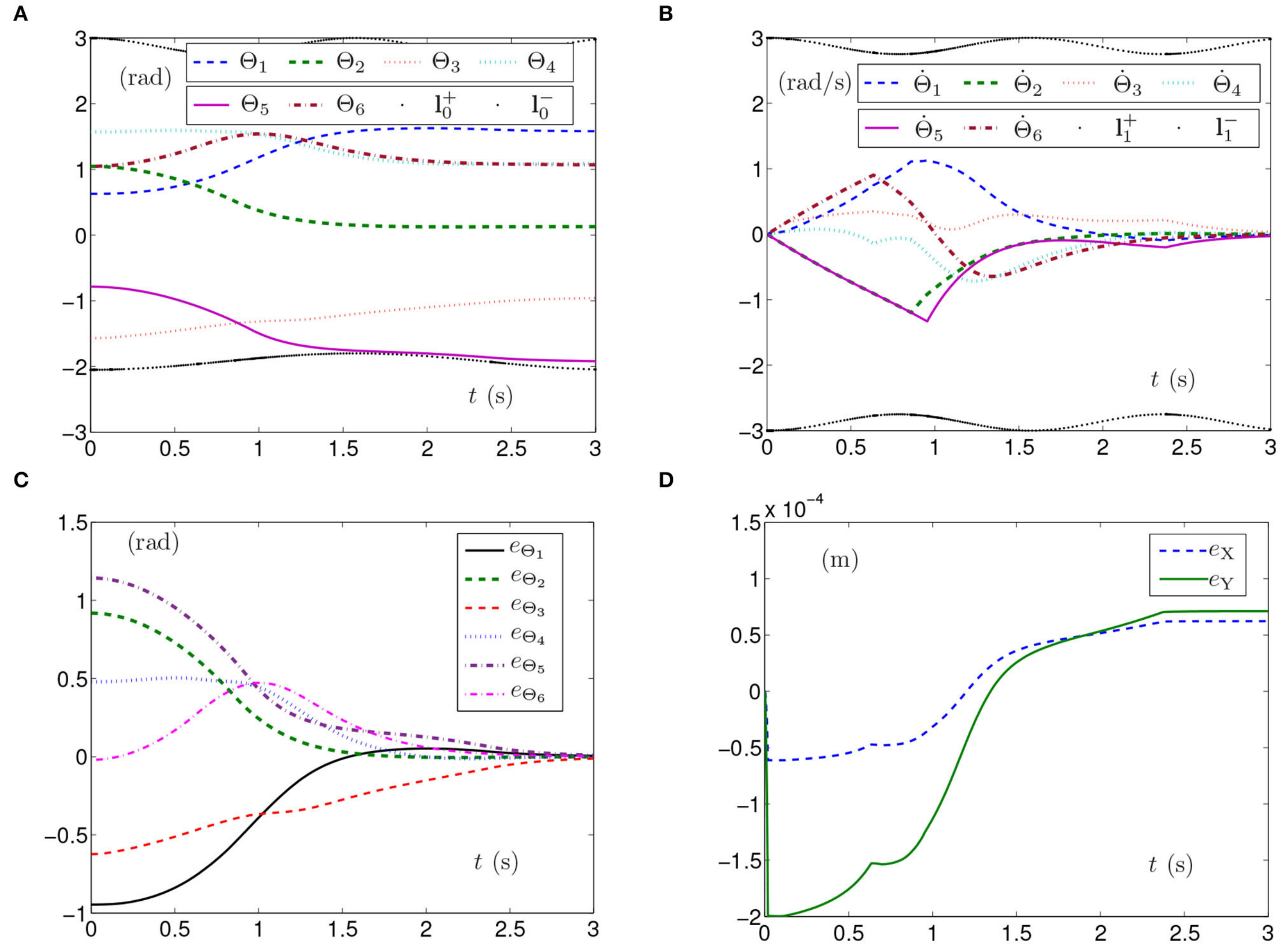


FIGURE 5 | Synthesized results of the planar manipulator using SMCS-2 with time-varying physical limits verged in case B. **(A)** Profiles of joint angles. **(B)** Profiles of joint-angle velocities. **(C)** Profiles of joint-angle errors. **(D)** Profiles of end-effector position errors.

states are set as $[1.574, 0.129, -0.947, 1.091, -1.928, 1.067]^{\mathrm{T}}$ rad. The task time-interval is set as [0, 3] s.

5.1.1. Case A: Loose Physical Limits

First, the simulation experiments are conducted with all physical limits satisfied. The joint-angle limits and joint-angle-velocity limits are time-varying, and the limits region are set loose. Specifically, each element in $\mathbf{l}_0^-(t)$ is set as $-3 + 0.25\sin^2(t)$ rad, and each element in $\mathbf{l}_0^+(t)$ is set as $3 - 0.25\sin^2(2t)$ rad. Each element in $\mathbf{l}_1^-(t)$ is set as $-3 + 0.25\sin^2(2t)$ rad/s, and each element in $\mathbf{l}_1^+(t)$ is set as $3-0.25\sin^2(2t)$ rad/s. The corresponding parameters are set as $\gamma = 10^4$, $\varsigma = 10^6$, and $\mu_1 = \mu_2 = 3$.

By PNN solver, the simulation results synthesized by the planar manipulator using the SMCSvZ are generated and presented in **Figure 1**. The curves of joint angles with the time-varying physical limits satisfied are presented in **Figure 1A**. As seen from **Figure 1B**, joint-angle velocities also satisfy their time-varying physical limits, the initial joint-angle velocities vector $\dot{\Theta}(0)$ equals **0**, and the time for $\dot{\Theta}(t)$ converging to zero is near to 2 s. That is, all physical limits are satisfied in the process of the self-motion task. Each joint angle as well as joint-angle velocity is not out of physical limits and does not need to be adjusted. Besides, **Figure 1C** depicts the value of $\mathbf{e}_\Theta$ (joint-angle error vector $\mathbf{e}_\Theta = \Theta(t) - \Theta_{\mathrm{g}}$ with e_{Θ_i} being the elements of $\mathbf{e}_\Theta$ $(i = 1, 2, \cdots, 6)$ and $t \in [0, t_{\mathrm{f}}]$), and the curves show that the joint angles gradually approach the given joint angles from initial ones over time. In addition, **Figure 1F** depicts end-effector position errors, which shows that the end effector keeps immobile in the practice. In specific, the maximal position error of the end effector is 1×10^{-4} m, and the position errors (i.e., e_{X} and e_{Y}) are near zero after 2 s. As seen in **Figures 1D,E**, the joint angles are adjusted then gradually approach the given ones from initial joint angles, and the planar manipulator completes the task successfully.

The simulation experiments based on the planar manipulator using SMCS-1 are carried out and the simulation results are depicted in **Figure 2**. The curves of joint angles with time-varying physical limits satisfied are shown in **Figure 2A**. However, the

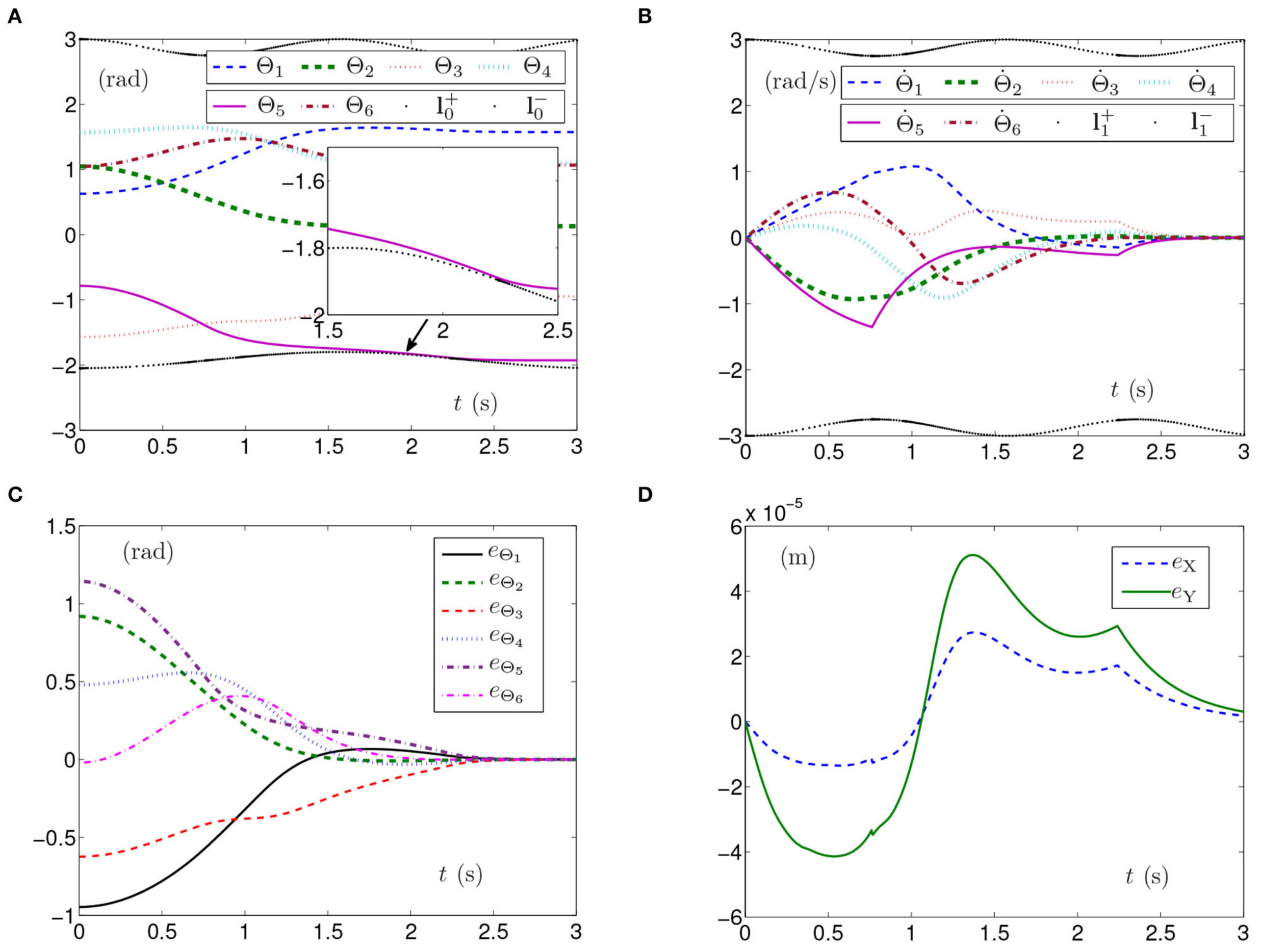


FIGURE 6 | Synthesized results of the planar manipulator using SMCSvZ with time-varying physical limits verged in case B. **(A)** Profiles of joint angles. **(B)** Profiles of joint-angle velocities. **(C)** Profiles of joint-angle errors. **(D)** Profiles of end-effector position errors.

initial joint-angle velocities vector $\Theta(0)$ does not equal zero as presented in **Figure 2B**, one of which is close to time-varying physical limits of joint-angle velocities. In this case, the time for $\dot{\Theta}(t)$ converging to zero is before 2 s. The curves shown in **Figure 2C** indicate that each element of $\mathbf{e}_\Theta$ converges to zero within 2 s, which also means the joint angles approach given joint angles within 2 s. As seen in **Figure 2D**, the maximal position error of the end effector is 1.5×10^{-4} m. Nevertheless, the position errors converge to some stable values but are not near zero. The joint angles also reach the given ones from initial joint angles, and the planar manipulator completes the task successfully. Due to similarity and space limitations, the corresponding pictures are omitted in the article, and the same is done in the following part.

The simulation results synthesized by SMCS-2 are shown in **Figure 3**. Thereinto, **Figure 3A** presents the curves of joint angles with time-varying physical limits satisfied. From **Figure 3B**, one obtains the initial velocities of the joint angles equal to zero with the time-varying physical limits satisfied, and the time for $\dot{\Theta}(t)$ converging to zero is after 2 s, which is fractionally longer than those shown in **Figures 1B**, **2B**. The curves shown in **Figure 3C** indicate that each element of $\mathbf{e}_\Theta$ converges to zero after 2 s, which also means the joint angles approach the given joint angles after 2 s. Besides, **Figure 3D** depicts that the maximal position error of the end effector is 2×10^{-4} m with the position errors stabilized after 1.5 s.

The above three experiment results in **Figures 1–3** show that when physical limits are all satisfied, the planar manipulator using SMCS-1 completes the task fastest but it does not have the zero initial velocities. On the premise that the values of initial velocities equal zero, compared with the planar manipulator using SMCS-2, the planar manipulator using SMCSvZ has a faster convergence speed and higher accuracy.

5.1.2. Case B: Stringent Physical Limits

Another simulation experiment based on the planar manipulator using SMCSvZ, SMCS-1, and SMCS-2 is carried out when the region of joint-angle physical limits is not large enough. The joint-angle lower limit $\mathbf{l}_0^-$ is set as$[\xi,\xi,\xi,\xi,\xi,\xi]^\mathrm{T}$ rad, where

TABLE 1 | Values of $\mathbf{e}$ synthesized by SMCS-1, SMCS-2, and SMCSvZ in case A and case B with $\mathbf{e} = \Theta(t_f) - \Theta_g$.

Joint-angle	1	2	3	4	5	6
	[a]$\mathbf{e}_1 \times 10^{-3}$	[a]$\mathbf{e}_1^* \times 10^{-2}$	[a]$\mathbf{e}_2 \times 10^{-2}$	[a]$\mathbf{e}_2^* \times 10^{-1}$	[a]$\mathbf{e}_z \times 10^{-3}$	[a]$\mathbf{e}_z^* \times 10^{-3}$
Θ_1	0.4035277546	0.5782355172	0.0015447915	0.0565260232	0.2921514678	0.5103526402
Θ_2	0.3129608608	−0.0545327541	0.0818730936	−0.0038875074	0.3244167469	0.1505453049
Θ_3	−0.5012693542	−0.8850811496	−0.1953758157	−0.1167433302	0.3180588608	−0.2854495128
Θ_4	0.1551459332	−0.3796040427	0.1295147141	−0.0181627340	0.2767581164	−0.1041636237
Θ_5	−0.5392726233	0.8367928486	−0.0282762806	0.0922359849	0.1932659195	−0.3721375789
Θ_6	0.2316018914	−0.0482526950	0.1809797644	0.0281766251	0.1004131585	−0.0027968302

[a] $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_z$ are obtained in case A, and $\mathbf{e}_1^*$, $\mathbf{e}_2^*$, and $\mathbf{e}_z^*$ are obtained in case B with subscripts $_1$, $_2$, and $_z$ representing SMCS-1, SMCS-2, and SMCSvZ, respectively.

TABLE 2 | Relation between position-error order and parameter γ in simulations based on planar manipulator using different control schemes in case A and case B.

		Order 10^{-6}	Order 10^{-5}	Order 10^{-4}	Order 10^{-3}
SMCSvZ	Case A	$\gamma = 10^8, 10^7, 10^6$	$\gamma = 10^5$	$\gamma = 10^4$	$\gamma = 10^3$
	Case B	$\gamma = 10^8, 10^7, 10^6, 10^5$	$\gamma = 10^4$	$\gamma = 10^3$	$\gamma = 10^2$
SMCS-1	Case A	$\gamma = 10^7$	$\gamma = 10^8, 10^6, 10^5$	$\gamma = 10^4$	$\gamma = 10^3$
	Case B	$\gamma = 10^7$	$\gamma = 10^8, 10^6, 10^5$	$\gamma = 10^4$	$\gamma = 10^3$
SMCS-2	Case A	–	$\gamma = 10^8, 10^7, 10^6, 10^5$	$\gamma = 10^4$	$\gamma = 10^3$
	Case B	$\gamma = 10^6$	$\gamma = 10^8, 10^7, 10^5$	$\gamma = 10^4$	$\gamma = 10^3$

$\xi = -2.1 + 0.25\sin^2(t)$. The other parameters are set the same as the above situation.

The simulation experiments based on the planar manipulator using SMCS-1 are carried out and the results are shown in **Figure 4**. As seen from **Figure 4A**, the curve of Θ_5 exceeds the curve of time-varying lower physical limit, which means that physical limits are not satisfied and the manipulator may be damaged. The curves of the joint-angle velocities in **Figure 4B** present that the manipulator has adjusted, but it still cannot avoid the joint angle exceeding the physical limits.

In comparison, **Figure 5** depicts the simulation results synthesized by SMCS-2. To be specific, the manipulator effectively adjusts Θ_5 to avoid exceeding the curve of the joint-angle physical limits, which means that the planar manipulator continues the self-motion task without mechanical damage in **Figure 5A**. As the allowable ranges of joint-angle velocities may be narrowed, the joint angles are adjusted slowly. The time for $\dot{\Theta}(t)$ converging to zero is near 3 s as depicted in **Figure 5B**. It takes nearly 3 s for the planar manipulator to adjust the joint angles to the given ones as depicted in **Figure 5C**. The maximal position error of the end effector depicted in **Figure 5D** is 1.5×10^{-4} m with the position errors converging to some stable values.

In addition to that, **Figure 6** depicts the simulation results synthesized by SMCSvZ. As seen in **Figure 6A**, each element of Θ satisfies the joint-angle physical limits in the process of the self-motion task. When Θ_5 is verging on its lower physical limit, the manipulator adjusts the joint-angle velocities, and thus the joint angles are correspondingly adjusted to avoid exceeding the physical limits. The time for $\dot{\Theta}(t)$ converging to zero is near 2.5 s as depicted in **Figure 6B**, which is less than one spent by the planar manipulator using SMCS-2. In addition, the curves shown in **Figure 6C** indicate that each element of $\mathbf{e}_\Theta$ converges to zero within 2.5 s. The maximal position error of the end effector depicted in **Figure 6D** is 6×10^{-4} m.

5.1.3. More Simulation Results

In this subsection, some other simulation results synthesized by the planar manipulator in case A and case B are presented. In **Table 1**, the data in columns 1, 3, and 5 are the values of $\mathbf{e}_\Theta$ obtained by the planar manipulator using SMCS-1, SMCS-2, and SMCSvZ in case A, respectively. Thereinto, the maximal errors of joint angles produced by SMCSvZ and SMCS-1 are of the order of 10^{-3} m, while it is of the order of 10^{-2} m that produced by SMCS-2. In addition, the data in columns 2, 4, and 6 in **Table 1** are the error values of $\mathbf{e}_\Theta$ obtained in case B. The maximal error of joint angles produced by SMCSvZ is of the order of 10^{-3} m, while it is of the order of 10^{-2} m when produced by SMCS-1, and it is of the order of 10^{-1} m when produced by SMCS-2. In this respect, the SMCSvZ is better than the other two schemes.

Generally, the parameters in the simulations influence the simulation results. For example, the position errors of the end effector reflect whether the end effector keeps motionless or not. As shown in **Figures 1F**, **2D**, **3D**, **5D**, **6D**, the maximal position errors are mostly of the order of 10^{-4} m when the parameter γ is set as 10^4 in case A and case B, which meet the practical requirements. If one desires to change the precision of position error, the value of γ can be changed as **Table 2** shown. For example, if γ is set as 10^5, the 6-DoF planar manipulator using SMCSvZ completes the self-motion, and the maximal position errors are of the order of 10^{-6} m. Furthermore, to improve the precision of position error, we can also change the value of μ_2. The simulation experiments are conducted based on the planar manipulator using SMCSvZ with different values of μ_2 in case B, and the different results of

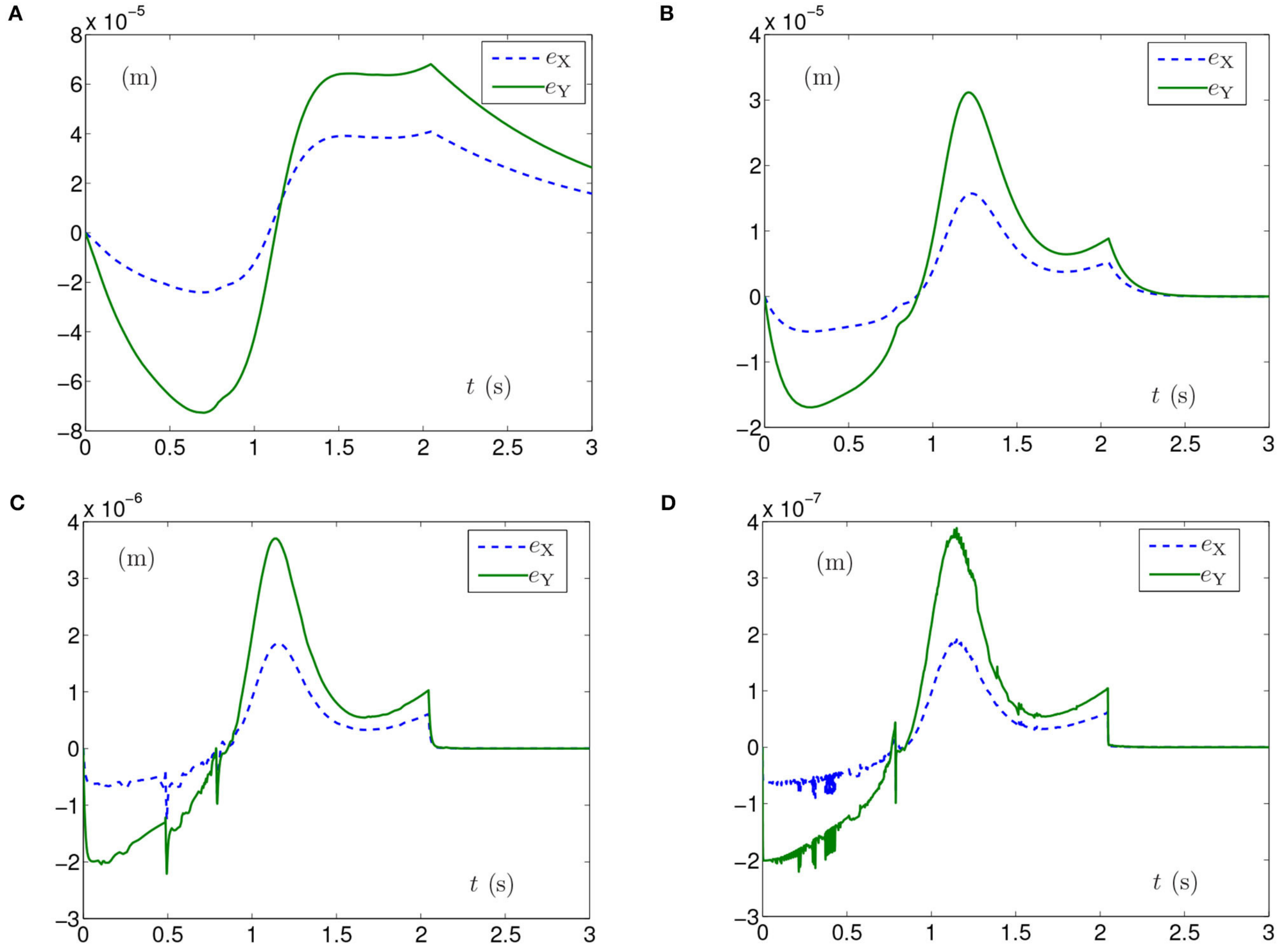


FIGURE 7 | End-effector position errors of planar manipulator using SMCSvZ with time-varying physical limits verged in Case B. **(A)** With $\mu_2 = 1$. **(B)** With $\mu_2 = 10$. **(C)** With $\mu_2 = 100$. **(D)** With $\mu_2 = 1,000$.

position errors are displayed in **Figure 7**. As seen in **Figure 7A**, when μ_2 is set as 1, the position error of the end effector does not converge within the duration of the task. When μ_2 is set as 10, the maximal position error of the end effector depicted in **Figure 7B** is 4×10^{-4} m and its convergence time is shortened. The position errors depicted in **Figures 7C,D** are of the orders of 10^{-5} m and 10^{-6} m, respectively. The design parameters can be set as appropriate values according to actual requirements.

In summary, with the desired precision and the physical limits satisfied, the planar manipulator using SMCSvZ completes the self-motion task more effectively and efficiently compared with the ones using SMCS-1 and SMCS-2.

5.2. Simulations Based on PUMA560 Manipulator

To further verify the efficiency of the proposed SMCSvZ, we conduct more simulation experiments based on the PUMA560 manipulator using SMCSvZ, SMCS-1, and SMCS-2.

The task time-interval of all simulation experiments is also set as [0, 3] s. The initial joint angles are set as $[0, -\pi/4, 0, \pi/2, -\pi/4, 0]^{\mathrm{T}}$ rad, and the given joint angles are set as $[0.1723, -0.9099, 0.122, 0, 0.0067, 0]^{\mathrm{T}}$ rad. Specifically, each element in $\mathbf{l}_0^-(t)$ is set as $-3 + 0.25\sin^2(t)$ rad, and each element in $\mathbf{l}_0^+(t)$ is set as $3 - 0.25\sin^2(2t)$ rad. Each element in $\mathbf{l}_1^+(t)$ is set as $3 - 0.25\sin^2(2t)$ rad/s, and each element in $\mathbf{l}_1^-(t)$ is set as $-3 + 0.25\sin^2(2t)$ rad/s. The parameter $\gamma = 1 \times 10^5$, and other parameters are set the same as above situation.

The simulation experiment based on the PUMA560 manipulator using SMCSvZ with the time-varying physical limits satisfied is done, and the results are displayed in **Figure 8**. The curves of joint angles are presented in **Figure 8A**. As seen in **Figure 8B**, the initial velocities of the joint angles equal zero, and the values of $\dot{\Theta}(t)$ converge to zero before 2 s. Besides, **Figure 8C** shows that the values of $\mathbf{e}_\Theta$ also converge to zero before 2 s. In addition, **Figure 8F** depicts that the maximal position error of the end effector is 6×10^{-6} m, and the values of position errors (i.e., e_X, e_Y, and e_Z) are close to zero over time. As seen in **Figures 8D,E**, the joint angles reach the given ones from

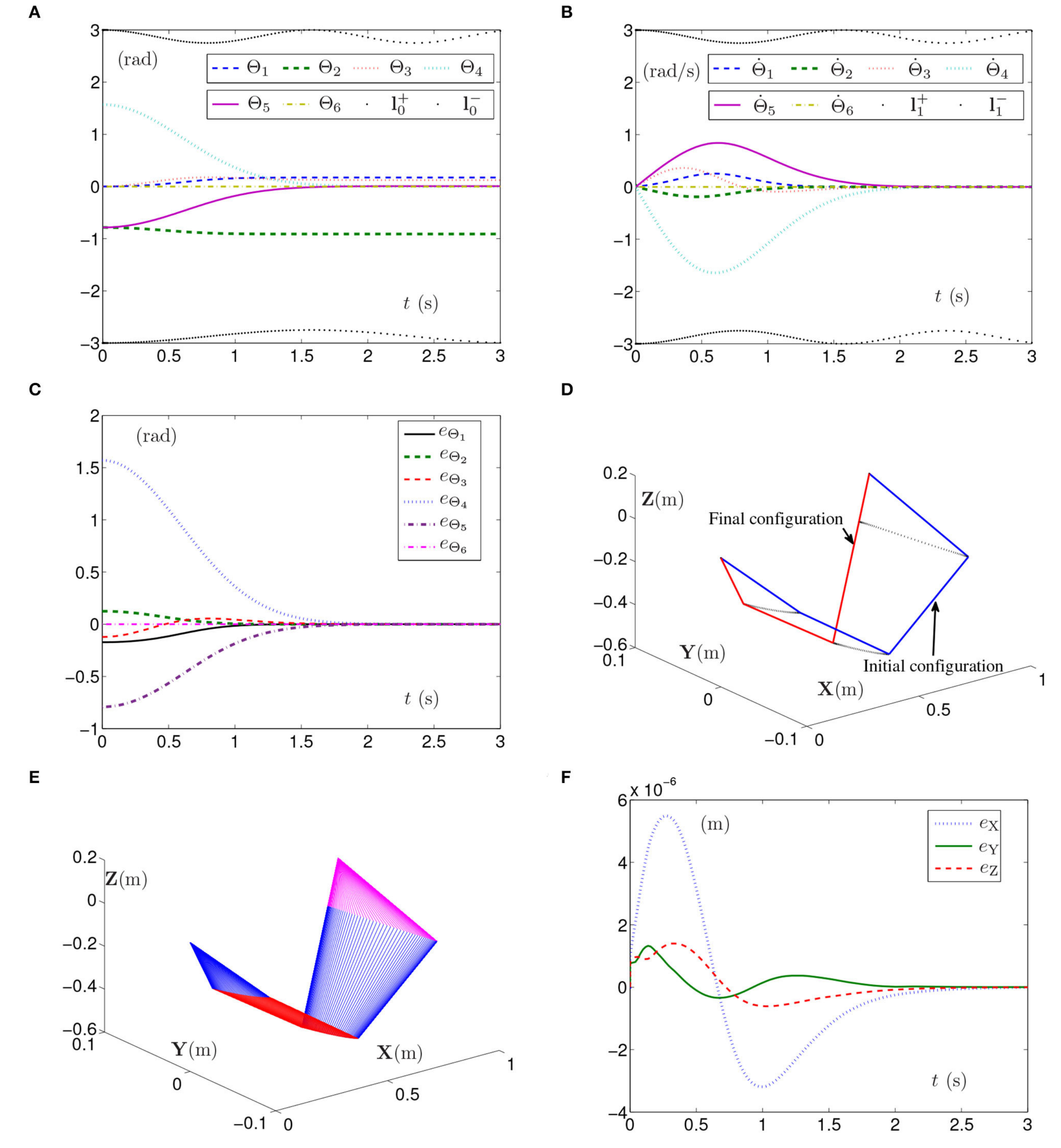


FIGURE 8 | Synthesized results of the PUMA560 manipulator using SMCSvZ with time-varying physical limits satisfied. **(A)** Profiles of joint angles. **(B)** Profiles of joint-angle velocities. **(C)** Profiles of joint-angle errors. **(D)** Profiles of initial and final manipulator positions. **(E)** Profiles of the planar manipulator. **(F)** Profiles of end-effector position errors.

initial joint angles, and the PUMA560 manipulator completes the task successfully.

When the joint-angle lower limit $\mathbf{l}_0^-$ is set as $[\xi,\xi,\xi,\xi,\xi,\xi]^{\mathrm{T}}$ rad with $\xi = -1.15 + 0.25\sin^2(t)$, the simulation results synthesized by the PUMA560 manipulator using the SMCSvZ are shown in **Figure 9**. As seen in **Figure 9A**, the self-motion task is completed in 3 s. Apparently, the curve of Θ_2 verges on the curve of the limit, and all physical limits are satisfied

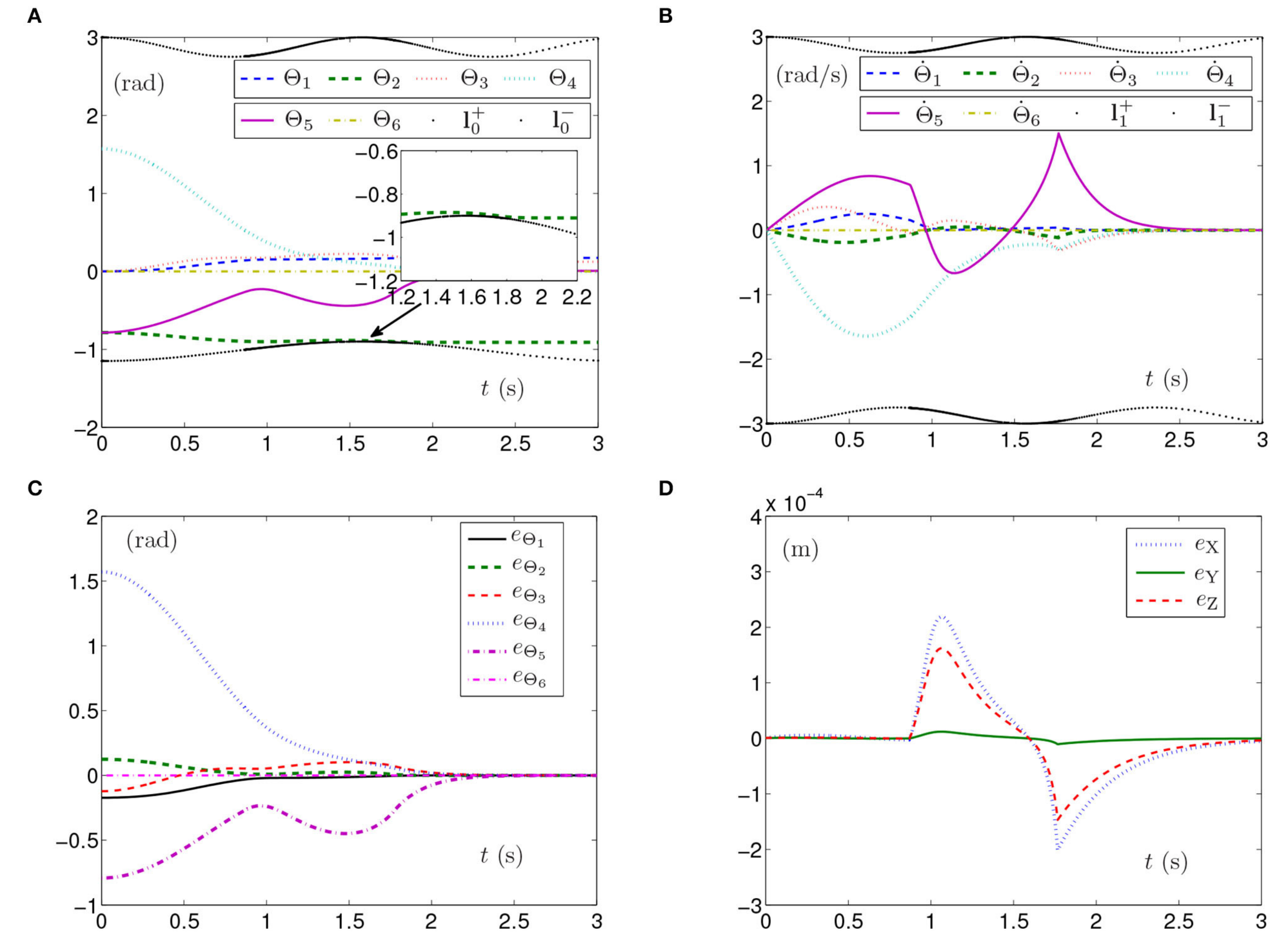


FIGURE 9 | Synthesized results of PUMA560 manipulator using SMCSvZ with time-varying physical limits verged. **(A)** Profiles of joint angles. **(B)** Profiles of joint-angle velocities. **(C)** Profiles of joint-angle errors. **(D)** Profiles of end-effector position errors.

in task durations. In **Figure 9B**, the self-motion requirement $\dot\Theta(0) = \mathbf{0}$ is satisfied, and $\dot\Theta(t)$ converges to $\mathbf{0}$ over time. In **Figure 9C**, the values of $\mathbf{e}_\Theta$ increasingly verge on $\mathbf{0}$. The maximal position error of the end effector is 3×10^{-4} m, and the values of position errors change slightly but converge to zero over time in **Figure 9D**, which indicates that the end effector also dynamically keeps immobile.

The simulation results synthesized by the PUMA560 manipulator using the SMCS-1 or SMCS-2 are similar to the results of the planar manipulator, which are omitted. To sum up, the PUMA560 manipulator using SMCSvZ can better meet the self-motion requirements, satisfy the time-varying physical limits, and complete the self-motion task efficiently.

6. CONCLUSION

We have proposed a refined QP-based self-motion control scheme of redundant robot manipulators with time-varying joint limits and zero initial joint-angle velocities satisfied *via* the ZNDE approach in the paper. The proposed scheme has been composed of a ZNDE equation constraint and a bound ZNDE inequation constraint. Compared with two previous SMCSs, we have theoretically analyzed the proposed SMCSvZ that well meets the self-motion requirements, then applied it to control the redundant robot manipulators for the self-motion task. The simulation experiments have been conducted based on the 6-DoF planar manipulator in two different cases. By comparing with the simulation results produced by the redundant robot manipulators using SMCS-1, SMCS-2, and SMCSvZ, the proposed SMCSvZ has shown its effectiveness, superiority, and practicability. Besides, the simulation results produced by the PUMA560 manipulator using SMCSvZ in two different cases have been obtained, and they have also verified the feasibility and correctness of the SMCSvZ. Based on ZNDE, more kinds of time-varying problems would be simplified and solved in future studies. Besides, the scheme established in the article is continuous-time and is not convenient for hardware implementation, and thus, the design and development of a discrete-time scheme could be one future research direction.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

ZT and YZ proposed the scheme and wrote the manuscript. ZT designed and carried out experiments. Both authors contributed to the article and approved the submitted version.

FUNDING

This study was aided by the National Natural Science Foundation of China (61976230), the project supported by Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2018), the Key-Area Research and Development Program of Guangzhou (202007030004), the Research Fund Program of Guangdong Key Laboratory of Modern Control Technology (2017B030314165), the Shenzhen Science and Technology Plan Project (JCYJ20170818154936083), and also the project supported by Hunan Education Department (19C1529).

REFERENCES

Akli, I. (2021). Trajectory planning for mobile manipulators including manipulability percentage index. *J. Intell. Robot*. 5, 543–557. doi: 10.1007/s41315-021-00190-3

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

Chen, D. and Zhang, Y. (2018). Robust zeroing neural-dynamics and its time-varying disturbances suppression model applied to mobile robot manipulators. *IEEE Trans. Neural Netw. Learn. Syst*. 29, 4385–4397. doi: 10.1109/TNNLS.2017.2764529

Gong, M., Li, X., and Lei, Z. (2019). Analytical inverse kinematics and self-motion application for 7-DOF redundant manipulator. *IEEE Access* 7, 18662–18674. doi: 10.1109/ACCESS.2019.2895741

Guo, D., Zhai, K., Yu, X., Cai, B., and Zhang, Y. (2013). "Zhang equivalence of different level robotic schemes: an MVN case study based on PA10 robot manipulator," in *International Conference on Robotics and Biomimetics* (Shenzhen), 1592–1597. doi: 10.1109/ROBIO.2013.6739694

Jin, L., He, L., Huang, Z., and Wang, J. (2020). Recurrent neural network for state adjustment of redundant manipulators. *IEEE Access* 8, 109783–109790. doi: 10.1109/ACCESS.2020.3002608

Jin, L., Zhang, J., Luo, X., Liu, M., Li, S., Xiao, L., et al. (2021). Perturbed manipulability optimization in a distributed network of redundant robots. *IEEE Trans. Ind. Electron*. 68, 7209–7220. doi: 10.1109/TIE.2020.3007099

Li, K., and Zhang, Y. (2012). Design and implementation of a zero-initial-velocity self-motion scheme on a six-DOF planar robot manipulator. *Ind. Robot*. 39, 401–411. doi: 10.1108/01439911211227980

Liao, B., Wang, Y., Li, W., Peng, C., and Xiang, Q. (2021). Prescribed-time convergent and noise-tolerant Z-type neural dynamics for calculating time-dependent quadratic programming. *Neural Comput. Appl*. 33, 5327–5337. doi: 10.1007/s00521-020-05356-x

Liao, B., Xiang, Q., and Li, S. (2019). Bounded Z-type neurodynamics with limited-time convergence and noise tolerance for calculating time-dependent lyapunov equation. *Neurocomputing* 325, 234–241. doi: 10.1016/j.neucom.2018.10.031

Ma, S. (1996). A balancing technique to stabilize local torque optimization solution of redundant manipulators. *J. Robot. Syst*. 13, 177–185. doi: 10.1002/(SICI)1097-4563(199603)13:3<177::AID-ROB5>3.0.CO;2-P

Ma, S., Hirose, S., and Yoshinada, H. (1996). Dynamic redundancy resolution of redundant manipulators with local optimization of a kinematic criterion. *Adv. Robot*. 10, 177–185.

Mathews, J. H., and Fink, K. D. (2004). *Numerical Methods Using MATLAB*. Upper Saddle River: Prentice Hall.

Pardi, T., Ortenzi, V., Fairbairn, C., Pipe, T., Esfahani, A. M. G., and Stolkin, R. (2020). Planning maximum-manipulability cutting paths. *IEEE Robot. Autom. Lett*. 5, 1999–2006. doi: 10.1109/LRA.2020.2970949

Qin, S., Gu, S., and Zhang, N. (2021). Recent developments in dynamic modeling, control and applications of neural networks. *Cybern. Syst*. 52, 1–2. doi: 10.1080/01969722.2020.1827791

Qiu, B., Zhang, Y., and Yang, Z. (2016). Revisit and compare Ma equivalence and Zhang equivalence of minimum velocity norm (MVN). *Adv. Robot*. 30, 416–430. doi: 10.1080/01691864.2015.1120243

Qiu, B., Zhang, Y., and Yang, Z. (2018). Analysis, verification and comparison on feedback-aided Ma equivalence and Zhang equivalency of minimum-kinetic-energy type for kinematic control of redundant robot manipulators. *Asian J. Control* 20, 2154–2170. doi: 10.1002/asjc.1706

Xia, Y., Wang, J., and Guo, W. (2020). Two projection neural networks with reduced model complexity for nonlinear programming. *IEEE Trans. Neural. Netw. Learn. Syst*. 31, 2020–2029. doi: 10.1109/TNNLS.2019.2927639

Xiao, L., Jia, L., Dai, J., and Tan, Z. (2020). Design and application of a robust zeroing neural network to kinematical resolution of redundant manipulators under various external disturbances. *Neurocomputing* 415, 174–183. doi: 10.1016/j.neucom.2020.07.040

Zhang, P., Ren, X., and Zhang, Z. (2021a). An efficient self-motion scheme for redundant robot manipulators: a varying-gain neural self-motion approach. *Robotica* 39, 1897–1908. doi: 10.1017/S0263574721000047

Zhang, Y., He, L., He, L., Yan, X., and Yu, X. (2016). "QP-based smoother self-motion planning and control of redundant manipulators using ZD variant with effective verifications," in *International Conference on Computer Science and Network Technology* (Harbin), 1545–1548. doi: 10.1109/ICCSNT.2015.7491024

Zhang, Y., Huang, Y., and Guo, D. (2009). "Self-motion planning of functionally redundant PUMA 560 manipulator via quadratic-program formulation and solution," in *International Conference on Mechatronics and Automation* (Changchun), 2518–2523. doi: 10.1109/ICMA.2009.5246498

Zhang, Y., Li, Z., Yang, M., Ming, L., and Guo, J. (2021b). Jerk-level Zhang neurodynamics equivalency of bound constraints, equation constraints, and objective indices for cyclic motion of robot-arm systems. *IEEE Trans. Neural Netw. Learn. Syst*. doi: 10.1109/TNNLS.2021.3110777

Zhang, Y., Ming, L., Yang, M., Qiu, B., and Li, Z. (2020a). "Inequality-type Zhang equivalency originating from neural dynamics for time-varying problems solving," in *International Conference on Mechatronics and Automation* (Beijing), 1516–1522. doi: 10.1109/ICMA49215.2020.9233778

Zhang, Y., and Xiao, L. (2012). "QP-based SMP scheme for robots with pseudoinverse method compared and singularities discussed," in *World Congress on Intelligent Control and Automation* (Beijing), 3583–3588. doi: 10.1109/WCICA.2012.6359068

Zhang, Y., Yang, M., Qiu, B., Li, J., and Zhu, M. (2020b). From mathematical equivalence such as Ma equivalence to generalized Zhang equivalency including gradient equivalency. *Theor. Comput. Sci*. 817, 44–54. doi: 10.1016/j.tcs.2019.07.027

Zhang, Z., Zhou, Q., and Fan, W. (2018). Neural-dynamic based synchronous-optimization scheme of dual redundant robot manipulators. *Front. Neurorobot*. 12:73. doi: 10.3389/fnbot.2018.00073

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

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

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 DC declared a shared affiliation with the authors ZT, YZ to the handling editor at the time of review.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

TYPE Original Research
PUBLISHED 24 August 2022
DOI 10.3389/fnbot.2022.955179

OPEN ACCESS

EDITED BY
Zhan Li,
Swansea University, United Kingdom

REVIEWED BY
Zhongda Tian,
Shenyang University of Technology, China
Juan Zhao,
Jingchu University of Technology, China

*CORRESPONDENCE
Binrui Wang
wangbinrui@163.com

RECEIVED 28 May 2022
ACCEPTED 27 July 2022
PUBLISHED 24 August 2022

CITATION
Wang L, Wang H, Yang X, Gao Y, Cui X and Wang B (2022) Research on smooth path planning method based on improved ant colony algorithm optimized by Floyd algorithm. *Front. Neurorobot.* 16:955179. doi: 10.3389/fnbot.2022.955179

Research on smooth path planning method based on improved ant colony algorithm optimized by Floyd algorithm

Lina Wang[1,2], Hejing Wang[1], Xin Yang[1], Yanfeng Gao[1], Xiaohong Cui[1,2] and Binrui Wang[1]*

[1]College of Mechanical and Electrical Engineering, China Jiliang University, Hangzhou, China, [2]Key Laboratory of Intelligent Manufacturing Quality Big Data Tracing and Analysis of Zhejiang Province, China Jiliang University, Hangzhou, China

Aiming at the problems of slow convergence and easy fall into local optimal solution of the classic ant colony algorithm in path planning, an improved ant colony algorithm is proposed. Firstly, the Floyd algorithm is introduced to generate the guiding path, and increase the pheromone content on the guiding path. Through the difference in initial pheromone, the ant colony is guided to quickly find the target node. Secondly, the fallback strategy is applied to reduce the number of ants who die due to falling into the trap to increase the probability of ants finding the target node. Thirdly, the gravity concept in the artificial potential field method and the concept of distance from the optional node to the target node are introduced to improve the heuristic function to make up for the fallback strategy on the convergence speed of the algorithm. Fourthly, a multi-objective optimization function is proposed, which comprehensively considers the three indexes of path length, security, and energy consumption and combines the dynamic optimization idea to optimize the pheromone update method, to avoid the algorithm falling into the local optimal solution and improve the comprehensive quality of the path. Finally, according to the connectivity principle and quadratic B-spline curve optimization method, the path nodes are optimized to shorten the path length effectively.

KEYWORDS
Floyd algorithm, ant colony optimization, fallback strategy, multi-objective optimization, quadratic B-spline curve

Introduction

The path planning of mobile robot is to plan the optimal path from the starting point to the target point in the specified area (Chen et al., 2020). At present, path planning algorithms is mainly presented in the form of traditional algorithms and intelligent algorithms. The traditional algorithms include the A* Algorithm (Xiong et al., 2020), Tabu Search (TS) (Khaksar et al., 2012), and D* Algorithm (Yao et al., 2021), etc. The intelligent algorithms include Ant Colony Optimization (ACO) (Wang, 2020), Particle Swarm Optimization (PSO) (Wang et al., 2020a), Genetic Algorithm (GA) (Chen and Gao, 2020), etc.

Intelligent algorithms can also be subdivided. Among them, the ant colony algorithm and particle swarm optimization algorithm belong to the swarm intelligent algorithm. Swarm intelligent algorithm has been a hot spot in path planning. There are two modes of swarm intelligence, namely, ant colony algorithm and particle swarm optimization algorithm. Swarm intelligence mainly refers to the intelligent behavior of many non-intelligent individuals in a group through simple cooperation. Swarm intelligence is applied to path planning, taking the ant colony algorithm as an example. It shows that a single ant in the ant colony has no intelligence, but through cooperation to form a complete system, it evolves into an intelligent whole that can explore the optimal path in a complex environment. Therefore, it is widely studied and applied in path planning.

Swarm intelligence is mainly manifested in five principles: (1) Proximity principle; (2) Quality principle; (3) The principle of diverse response; (4) Stability principle; (5) Adaptability principle.

Swarm intelligence also has four features.

(1) The control of swarm intelligence is decentralized, and there is no unified control center, so it can adapt to various environments and has strong robustness. For example, the ant colony algorithm can carry out path planning in various complex environments and obtain the optimal path.
(2) Each individual in the swarm can communicate by changing the environment, which has good scalability. For example, the ants change the pheromone content in the environment by leaving pheromones on the path, to realize communication with other individuals.
(3) The behavior of individuals in the swarm or the rules they follow are very concise, so it is very convenient to realize swarm intelligence. For example, individuals in the ant colony only need to follow the state transition rules to find the path and leave pheromones to inform the latecomers.
(4) The complex behavior of a swarm is the result of individual communication and cooperation. Under the guidance of appropriate rules, swarm intelligence can play a role in some form of emergence through communication and cooperation. For example, individuals in the ant colony interact through pheromones and then complete path exploration. Then pheromone update mechanism plays a role in guiding the ant colony to optimize the path further and finally get the optimal path.

Ant colony algorithm in swarm intelligence fully reflects the characteristics of swarm intelligence. It is simple to set parameters, suitable for various complex environments, and has strong robustness. Therefore, it is widely used in robot path planning. In this paper, the ant colony algorithm will be deeply studied and optimized.

Italian scholar Marco Dorigo proposed the ant colony algorithm in 1992. The algorithm was derived from the path finding behavior of ants looking for food sources in nature (Mac et al., 2016). The most prominent feature of the ant colony algorithm is the positive feedback mechanism (Zhang et al., 2021) which is conducive to obtaining the optimal solution quickly. Then, the ant colony can change the environment by releasing pheromone, so as to communicate indirectly (Yi et al., 2019). At last, the ant colony adopts the distributed computing method to search the path (Zheng et al., 2020), and the parallel computing is carried out by multiple individuals at the same time. Nevertheless, the defects of slow convergence speed and easy to fall into the local optimal solution cannot be ignored (Yang et al., 2019).

For the defects of the ant colony algorithm, many researchers have proposed optimization schemes that can be divided into three categories. (1) In consideration of the slow convergence speed of the ant colony algorithm, improve the initial pheromone allocation method, or improve the state transition probability matrix, such as Luo et al. (2020) and Li et al. (2021); etc. (2) In order to optimize the defect of the ant colony algorithm that it is easy to fall into local optimal solution, the pheromone matrix updating method is optimized or pheromone concentration is limited, such as Akka and Khaber (2018) and Wang et al. (2020), etc. (3) Many schemes to improve the path smoothness of ant colony algorithm have been proposed. There are mainly two ways: improving the heuristic function and optimizing the path nodes, such as Dai et al. (2019) and Yang et al. (2019), etc. Some optimization schemes will be introduced in detail below.

To improve the ant colony algorithm, there are a lot of optimization schemes (Akka and Khaber, 2018; Luo et al., 2020; Li et al., 2021). In Luo et al. (2020), an improved ant colony algorithm was proposed. The algorithm constructs unequally distributed initial pheromone in the early stage of path planning. At the same time, the pseudo-random state transition rule is used to select the trail. The deficiency is that the algorithm only sets the initial pheromone according to the position information of the node, which is not conducive to avoiding obstacles in the process of the ant search path, and the guidance of the ant colony is not direct enough. In Li et al. (2021), an improved algorithm based on turning angle constraint was proposed. Firstly, the initial pheromone concentration between the starting node and the target node is increased. Then, the evaluation function and rotation constraint factor of the A * algorithm is added to the heuristic function. The nodes with the optimal path length and rotation number can be selected in the next step. Finally, in the pheromone updating part, the distribution principle of the wolf swarm algorithm is introduced to strengthen the influence of a high-quality population. The algorithm proposed by Li effectively avoids falling into optimal local solutions, but the convergence speed in a complex environment cannot meet the requirements. In Akka and Khaber (2018), an improved ant colony optimization algorithm was proposed. The algorithm uses stimulus probability to help ants select the following grid, and uses new heuristic information to improve visibility accuracy. In addition, the improved algorithm adopts new pheromone updating rules and dynamically adjusts the evaporation rate, which accelerates the convergence speed and expands the search space. This algorithm does not consider the requirements of path smoothness when effectively accelerating the convergence speed, which is not conducive to reducing the energy consumption and mechanical loss of the robot.

In summary, to solve the problems of slow convergence rate and easily fall into the local optimal solution of ant colony algorithm, this paper proposes an improved algorithm.

(1) For the difficulties in Luo et al. (2020), the Floyd algorithm is introduced to generate the guidance path. The path is a feasible path without collision with obstacles. Setting the initial pheromone based on the track can help the ant colony avoid blind search and take into account the obstacle avoidance needs.
(2) Considering that the ants easily fall into the deadlock and self-locking state, the fallback strategy is proposed to reduce the number of dead ants and help improve the success rate of the algorithm to solve the way.
(3) For the problems that have not been solved in Li et al. (2021), the APF method and the concept of the distance between the optional node and the target node are introduced to optimize the structure of the heuristic function, which improves the state transition probability and accelerates the convergence rate.
(4) Given the shortcomings of Akka and Khaber (2018), the connectivity principle and quadratic B-spline curve optimization method are proposed to optimize the corner nodes, further shortening the path length and reducing the mechanical loss of the robot in the working process.
(5) Moreover, this paper proposes a multi-objective optimization method, taking into account the path length, path safety, and path energy consumption, to solve the bearing with the highest comprehensive quality. The pheromone updating method is improved based on the multi-objective optimization method and dynamic principle, which prevents the algorithm from falling into the local optimal solution to the greatest extent.

The rest of this paper is as follows. The second part briefly describes the two-dimensional grid environment modeling method, which is a crucial environment for algorithm operation. The third part introduces the core part of the classic ant colony algorithm. The fourth part gives the progress measures of the algorithm in detail. In the fifth part, the classic ant colony algorithm and the improved algorithm are compared and analyzed. The sixth part summarizes the contributions and shortcomings of the improved algorithm, and briefly looks forward to future work.

Environment modeling

Environment modeling is the basic part of a path planning algorithm (Mac et al., 2016). The grid method is used in mobile robot path planning algorithms because of its simple modeling method, easy programming, and ability to express irregular obstacles. It is a commonly used environmental modeling method (Ouyang and Yang, 2014). The grid method converts environmental information into grid form (Zhang et al., 2019), and distinctive blocks are regularly processed and properly expanded, as shown in Figure 1, which greatly reduces the difficulty of path planning.

In the grid map, the white grid is the free space and optional node, represented by "0." The black grid is the obstacle space and belongs to the tabu node, represented by "1."

In addition, the selection of grid size is also a key factor of the algorithm. If the grid is too small, the map resolution is high, which is not conducive to fast decision-making. If the grid is too large, the map resolution will be low, which is conducive to quick decision-making. Still, it cannot guarantee a viable path in the dense obstacle environment.

Although the grid sequence number method saves more memory, it is not conducive to the rapid iteration of the ant colony algorithm (Xiao et al., 2021). To ensure the convergence speed, the grid sequence number will be converted to coordinate (x, y), and the conversion formula is as follows.

$$\begin{cases} x = \mathrm{mod}(i, M) - 0.5 \\ y = M - ceil(i/M) + 0.5 \end{cases} \tag{1}$$

In the formula, $M$is the map size, mod is the solution function that returns the abscissa of the grid, and the*ceil*process returns the grid ordinate (Ali et al., 2020).

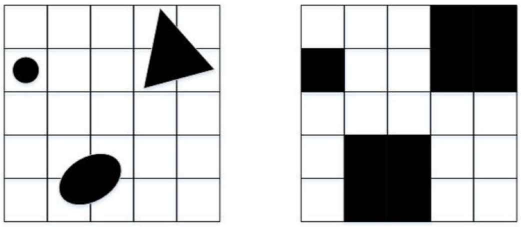


FIGURE 1
Obstacle rule processing.

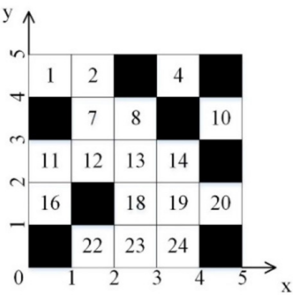


FIGURE 2
Grid coordinate diagram.

The coordinate form of the grid map is shown in Figure 2.

Ant colony optimization algorithm

Ant colony algorithm is derived from the path finding behavior of ants looking for food sources in nature, which has strong robustness in complex environment. The classic ant colony algorithm is easy to implement, the parameter setting is convenient, and the requirement for the computing environment is low (Zhang et al., 2020). The two important mechanisms of the ant colony algorithm are positive feedback and pheromone communication. The positive feedback mechanism guarantees the convergence of the ant colony algorithm. The higher the path quality, the more pheromones will accumulate when the pheromone is updated, which will encourage more ants to choose and realize the fast convergence of the algorithm. The pheromone communication mechanism is indirect communication for ant colony individuals. Ants leave pheromones on the path they traversed. Other individuals combine known environmental information and pheromones on the course to become new prior knowledge, which will help ant colonies reduce blind search and find target nodes faster.

These two mechanisms evolve into two key links in algorithm implementation: state transition probability (Chen et al., 2021) and pheromone update (Li and Wang, 2020).

State transition probability

Ants need to go through many intermediate nodes in the process of finding the path. For the selection of each intermediate node, the state transition probability matrix of the optional node should be established first, and then select from the probability matrix by roulette operation (Wang et al., 2020b).

The state transition probability is shown in equation (2).

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_{j \in allowed_k} \tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)} & s \in allowed_k \\ 0 & s \notin allowed_k \end{cases} \tag{2}$$

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{3}$$

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4}$$

Where, τ_{ij} is the pheromone content from node i to node j, η_{ij} is the heuristic function, d_{ij} is the Euclidean distance from node i to node j, α is the pheromone heuristic factor, β is the expected heuristic factor, and $allowed_k$ is the set of optional nodes in the next step (Xiong et al., 2021).

Pheromone update mode

Individuals in the ant colony will leave pheromones when passing through each path. As a prior knowledge of subsequent individuals, ants communicate indirectly through the pheromones. After several iterations, the ants traverse the map, and the pheromone content of the path indicates the quality of the trajectory. The higher the quality of the path pheromone concentration is higher. In the algorithm implementation process, to facilitate calculation, a pheromone update is placed after each iteration of the ant complete path search.

The pheromone update method is shown in equation (5).

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{5}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{6}$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L_k} & tour(i,j) \in tour_k \\ 0 & tour(i,j) \notin tour_k \end{cases} \tag{7}$$

Where, ρ is the pheromone volatilization rate, $\Delta\tau_{ij}(t)$ is the total pheromone increment of the path in this iteration, $\Delta\tau_{ij}^{k}(t)$ is the pheromone increment brought by the k-th ant, m is the number of ants, Q is the pheromone increase intensity, and L_k is the path length traveled by the k-th ant (Tao et al., 2021).

Improvement of ant colony optimization algorithm

Initial pheromone matrix

The Floyd algorithm is named after Robert Floyd (Hao and He, 2008), one of the founders. Floyd algorithm is a dynamic programming algorithm, suitable for dense maps, simple and effective, and easy to implement. Its efficiency is higher than the Dijkstra algorithm (Shi and Wang, 2009). Taking the optimal path obtained by the Floyd algorithm as the guiding path of the ant colony algorithm can help set the initial pheromone matrix with a guiding effect (Tian, 2021a).

Floyd algorithm can calculate the shortest path between each node in the map environment, and the core idea is to solve the shortest path matrix (Lyu et al., 2021). There are only two possible shortest paths from node*i*to node*j*. One is the Euclidean distance of two nodes. That is, two nodes are connected, and the other is from node *i* to node *j* through several intermediate nodes (Yang, 2020). Therefore, $Dis(i,j)$ is set as the Euclidean distance from node*i*to node*j*, and then all nodes*k*except these two nodes are judged. If$Dis(i,k)+Dis(k,j) < Dis(i,j)$holds, it is proved that the path from node*i*to node*k*and then to node *j*is shorter than the path from node*i*to node*j*, then let $Dis(i,j) = Dis(i,k) + Dis(k,j)$. After traversing node *k*, the shortest distance from node *i* to node *j* is recorded in$Dis(i,j)$.

The implementation of the Floyd algorithm is as follows.

(1) Initialize the shortest distance matrix*Dist*as the adjacency matrix of the map, and the path node matrix *path*is empty. The elements in the adjacency matrix are initialized to infinity. If two nodes have edges, the corresponding elements in the matrix are set as weight values.

(2) For node *i* to node *j*, traversing the remaining nodes to determine whether there is node*k*makes the distance from node *i* to node *k* and then to node *j* shorter than the known path. If it exists, update matrix*Dist*and matrix*path*.

The state transition equation is shown in equation (8).

$$Dist(i,j) = \begin{cases} Dist(i,j) & Dist(i,k) + Dist(k,j) \geq Dist(i,j) \\ Dist(i,k) + Dist(k,j) & Dist(i,k) + Dist(k,j) < Dist(i,j) \end{cases} \tag{8}$$

After determining the starting node and the target node, the Floyd algorithm can quickly obtain the optimal path. Then take the generated path as the guidance to change the pheromone content on the path so that it is different from other paths. Because the ant will be affected by pheromone when choosing the path, it is easier to choose the guidance path. The pheromone difference between the guide path and other paths will make the ant tend to the former to quickly find the target node. The initial pheromone matrix is set as follows.

$$\tau_{ij}(0) = \begin{cases} k * C & tour(i,j) \in tour_F \\ C & tour(i,j) \notin tour_F \end{cases} \tag{9}$$

Where, $\tau_{ij}(0)$is the initial pheromone matrix. $tour_F$is the guiding path generated by the Floyd algorithm, and the pheromone concentration of the guiding path is set to*k*times of other paths.

The APF has also been used to generate the guidance path of the ant colony algorithm. Therefore, under the same conditions, the path planning results of APF method and Floyd algorithm are compared. The results are shown below.

According to Figure 3 and Table 1, the path of APF method will pass through obstacles, which is not allowed, while the path of Floyd algorithm fully realizes the requirements of obstacle avoidance. In addition, the Floyd algorithm has few redundant nodes, and the length is only 51.40 % of the APF method. The Floyd algorithm is much better than the APF method. Therefore, introducing the optimal path of the Floyd algorithm as the guiding path will help the ant colony algorithm quickly find the target node and accelerate the convergence speed of the algorithm.

Ant fallback strategy

The ants often encounter deadlock problems when exploring paths (Dai et al., 2019), including self-locking and deadlock caused by obstacles. The deadlock problem will cause excessive death of ants, weaken the ability of the ant colony to explore the path, and slow down the convergence speed of the algorithm (Tian, 2020b; Wang, 2020).

Obstacles that will form ant deadlocks are usually concave. Because the ant follows the rule of putting the passed nodes in the tabu list when exploring the path to reduce the generation of redundant nodes, when ants encounter concave obstacles, this rule will make ants unable to stay away from the obstacles and thus trapped near the obstacles. Self-locking is due to that ants have no clear direction of the target node at the beginning of the iteration, only blind search, and ultimately face the plight of no optional nodes. The above two deadlock problems are shown in Figure 4.

The particles in Figure 4 are the ants searching path. On the left side of Figure 4, the ant at node P1 chooses the left node P2. It cannot retreat away from the obstacles because of the tabu list rules. The ant can only continue to select the left node P3, and finally, it is trapped in the barrier. On the right side of Figure 4, the ant at node P1 does not get a clear direction of the target node and can only choose the next node based on roulette. The ant follows the series of nodes like$P1 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow P5 \rightarrow P6$and finally, the ant is trapped in a self-locking dilemma. In the classic ant colony algorithm, ants are usually discarded after they fall into the deadlock dilemma so that subsequent ants continue to search the path. The situation when ants fall into deadlock can be described by the following formula.

$$allowed_i \cap Obs = allowed_i \tag{10}$$

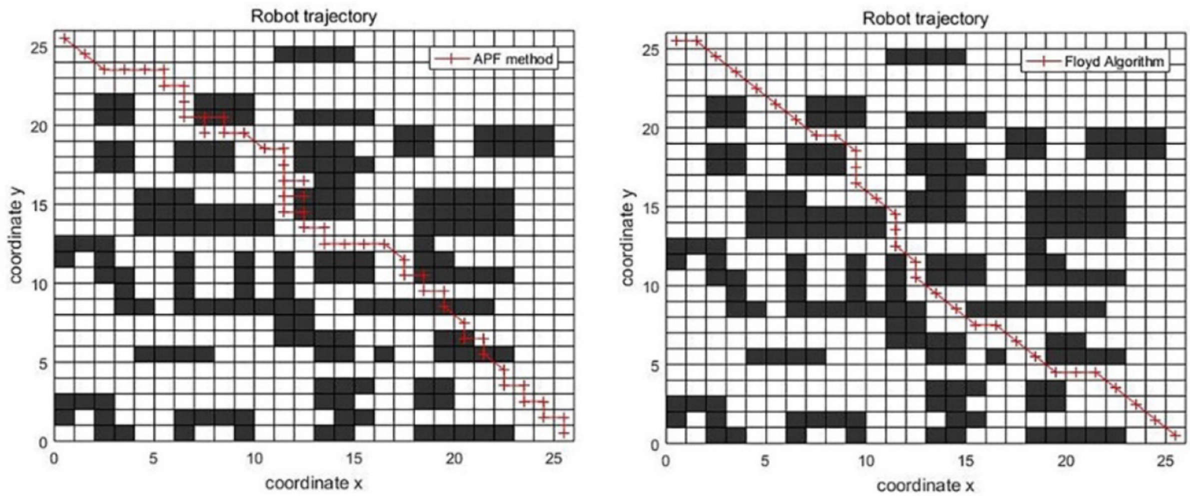


FIGURE 3
Comparison of boot paths.

TABLE 1 Comparison of boot paths length.

	APF method	Floyd algorithm
Optimal path length	74.4853	38.2843

Where, $allowed_i$is the list of optional nodes, $Obs$is the tabu list.

To solve the deadlock problem, the ant fallback strategy is proposed. When the ant has no optional node and has not reached the target point, the fallback strategy is implemented. That is, the current node is added to the tabu list and returned to the previous node, and the pheromone concentration at the current node is reduced. If there are new optional nodes at this time, the fallback strategy will end. If not, continue to execute the fallback strategy until there are optional nodes for ants to select. The fallback strategy is shown in Figure 5.

In Figure 5, the ant at node P3 falls into the deadlock and starts to perform the fallback strategy. The node P3 is added to the tabu list, and the ant returns to node P2. There is no optional node for the ant to choose, so the ant continues to implement the strategy. When returning to node P1, the ants find new optional nodes. At this time, it ends the execution of the fallback strategy. The ant selects the node P4 by the roulette rule and continues to explore new paths. Since the trap has been added to the tabu list, subsequent ants will no longer fall into the deadlock dilemma here. The treatment of the self-locking dilemma is similar (Tian and Chen, 2021a).

The pheromone update method when executing the fallback strategy is as follows.

$$\tau_{ij}(t+1) = (1-\lambda) * \tau_{ij}(t) \tag{11}$$

Where, λ is the pheromone penalty evaporation coefficient, which reduces the pheromone concentration of the trap nodes and helps the ant avoid the trap.

Heuristic function optimization strategy

To better solve the slow convergence problem of the ant colony algorithm, some optimization schemes are proposed for the heuristic function.

The heuristic function of the ant colony algorithm is the reciprocal of the Euclidean distance between the current node and the optional node, as follows.

$$\eta_{ij} = \frac{1}{d_{ij}}, d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{12}$$

The heuristic function does not contain the information of the target node, and the ant lacks guidance in finding the path, which is easy to search blindly, resulting in the slow convergence of the algorithm. This paper proposes the concept of distance between the optional node j and target node E, replacing the original heuristic function, as follows.

$$\eta_{ij} = \frac{1}{d_{jE}}, d_{jE} = \sqrt{(x_E - x_j)^2 + (y_E - y_j)^2} \tag{13}$$

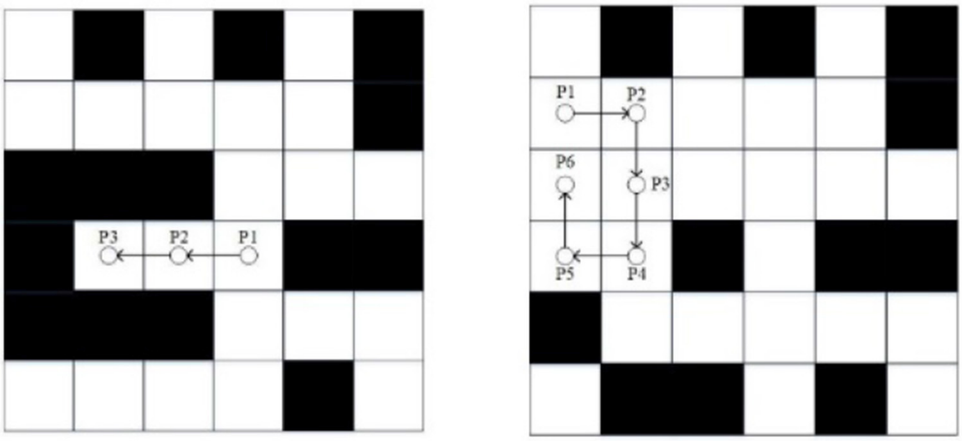


FIGURE 4
Deadlock and self-locking.

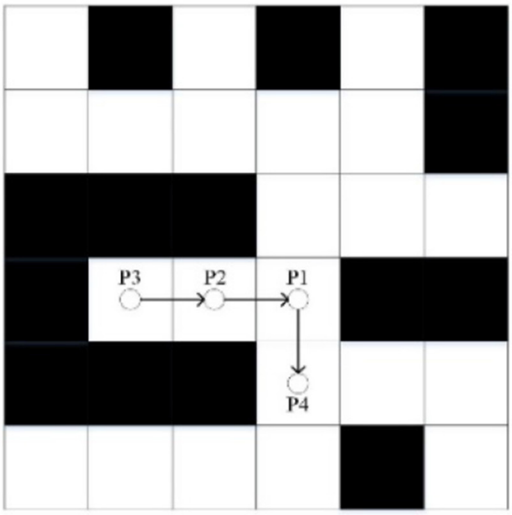


FIGURE 5
Fallback strategy.

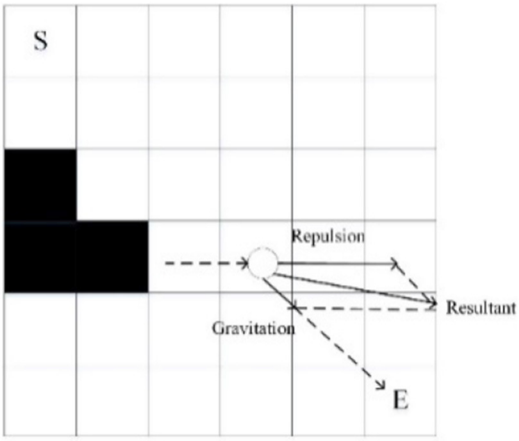


FIGURE 6
Artificial potential field method.

The heuristic value of the optional node closest to the target node is the largest, and the probability of being selected is also the largest. With the information of the target node, the ant colony has a clear direction in exploring the path, and the convergence speed will be accelerated.

Based on the new heuristic function, considering further improving the convergence speed of the algorithm, the APF method is an option. The APF method has the advantages of low calculation and fast convergence speed, so it is considered to optimize the heuristic function (Wang et al., 2020c).

As one of the widely used path planning algorithms, the APF method was first proposed by Khatib. O in 1985 (Pan et al., 2019). The main idea of the APF method is to regard the motion environment of the robot as a virtual force field (Li and Wang, 2022). The target node and obstacles generate gravitational and repulsive forces, respectively, in the robot, and the motion of the robot is controlled by the resultant force. The effect of the APF method is shown in Figure 6. The particle is a mobile robot (Wang and Wang, 2020a).

After simulation experiments, the gravity concept of the target node to the robot in the APF method is used to optimize the heuristic function, which is as follows.

$$Gra = sigma * \sqrt{(x_E - x_j)^2 + (y_E - y_j)^2} \tag{14}$$

$$d_{jE} = 1/\sqrt{(x_E - x_j)^2 + (y_E - y_j)^2} \tag{15}$$

$$\eta_{ij} = \frac{power(0.5, Gra)}{d_{jE}} \tag{16}$$

Where, *Gra*is the gravitational effect of the target node on the optional node, *sigma*is the gravitational constant, *power*is the power function that returns the value of the exponential power of the given bottom number. *j* is the optional node and *E* is the target node.

In the early iterations, the pheromone is accumulating, and the state transition probability is dominated by the heuristic function. In the middle and late iterations, the ants complete the path exploration, and the pheromone accumulates a lot. The pheromone dominates the state transition probability. Therefore, the heuristic function needs to be adjusted adaptively to match the pheromone matrix, so as to improve the update mode of the state transition probability matrix.

The adaptive adjustment of the heuristic function is related to the number of iterations. Therefore, the normal distribution function is introduced and combined with the heuristic function, as shown below.

$$Nd\_function = e^{(-((k/K)\hat{}2)/2)} \tag{17}$$

$$P_{ij}^{k}(t) = \begin{cases} \frac{\tau_{ij}^{\alpha}(t)*(Nd\_function*\eta_{ij}(t))^{\beta}}{\sum\limits_{j \in allowed_k} \tau_{ij}^{\alpha}(t)*(Nd\_function*\eta_{ij}(t))^{\beta}} & s \in allowed_k \\ 0 & s \notin allowed_k \end{cases} \tag{18}$$

Where, *Nd_function*is the deformation of the standard normal distribution function, omitting the coefficient, *k*is the current number of iterations, *K*is the maximum number of iterations, $P_{ij}^{k}(t)$is the optimized state transition probability.

Optimization strategy of pheromone updating method

In the classic ant colony algorithm, the pheromone updating method is only related to the path length. The shorter the path, the higher the pheromone increment. The updating method ignores other requirements, such as path security and energy consumption. In addition, the pheromone volatilization coefficient is constant and does not dynamically update with iterations. In the late iterations, the optimal path has been fixed. The behavior of finding a better path has stopped, which causes the local optimal solution (Wang and Wang, 2020b).

To solve the above problems, this paper proposes a pheromone updating method based on the multi-objective optimization method (Guo et al., 2020) and dynamic principle (Tian et al., 2020; Tian, 2021b).

Multi-objective optimization has been used in other path planning algorithms, mainly to improve the quality of the algorithm. Based on the idea of multi-objective optimization, this section puts forward three optimization objectives: path length, path security, and path energy consumption, which are used as the standard to update the pheromone matrix. Where, the path length is the sum of the distances of the path nodes, denoted as *Length*; path safety is the number of dangerous nodes on the path, denoted as *Risk*; path energy consumption depends on the number of turns and turning angles of the path, denoted as*Consumption*. The multi-objective optimization function is shown below.

$$Length = \sum_{i=S}^{j=E} d_{ij} \tag{19}$$

$$Risk = \sum D\_nodes \tag{20}$$

$$Consumption = \sum 0.5 * N\_corner + 0.5 * T\_angle \tag{21}$$

$$J\_quality = k_1 * Length + k_2 * Risk + k_3 * Consumption \tag{22}$$

$$k_1 + k_2 + k_3 = 1 \tag{23}$$

Where, *Length* is the sum of distances of all nodes in the path. *Risk*is the sum of dangerous nodes that the path passes. Dangerous nodes refer to nodes whose ratio of optional nodes to obstacle nodes is <1. *Consumption*is the sum of the number of corners and the turning angles of the path. *J_quality*is the comprehensive quality, a comprehensive index composed of path length, path safety, and path energy consumption with different proportions. The smaller the value is, the higher the comprehensive quality of the path is.

Replacing the path length with the comprehensive quality is the way to realize the multi-objective optimization idea. The improved pheromone update method is as follows.

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{J_{best}} + \frac{b*Q}{BEST} & tour(i,j) \in tour_{best} \\ \frac{Q}{J_{worst}} - \frac{w*Q}{WORST} & tour(i,j) \in tour_{worst} \\ \frac{Q}{J} & tour(i,j) \in tour_{other} \end{cases} \tag{24}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{25}$$

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{26}$$

Where, J_{best} is the comprehensive quality of the local optimal path, *b*is the number of ants on the local optimal path, *BEST*is the comprehensive quality of the global optimal path. $J_{worst}$is the comprehensive quality of the local worst path, *w*is the number of ants on the local worst path, *WORST*is the comprehensive quality of the global worst path. The optimal path, the worst path and other paths update the pheromone according to the three forms in equation (24), respectively.

The optimized pheromone updating method is based on the comprehensive quality of the path. The reward and punishment system is implemented for the optimal path and the worst path, and the pheromone gap between them gradually increases. The subsequent ants will be more inclined to the optimal path, which helps to accelerate the convergence speed of the algorithm (Dai et al., 2022).

In the late iteration of classic ant colony algorithm, the pheromone accumulation is completed, and ants are affected by pheromone, so it is difficult to continue to explore other paths, and the local optimal solution needs to be solved (Tian and Chen, 2021b).

In this paper, the concept of dynamic updating volatility coefficient is proposed. In the late iterations, if the quality of the optimal path of five consecutive iterations does not change, the volatility coefficient is dynamically updated, which can increase the volatilization of pheromone and weaken the attraction of pheromone. This helps ants explore other paths. The dynamic volatility coefficient is shown below.

$$\rho = \begin{cases} 1.2 * \rho & J(k) = J(k-5), k \geq 35 \\ 0.8 & \rho \geq 0.8 \end{cases} \tag{27}$$

Where, $J(k)$and$J(k-5)$are the comprehensive quality of the optimal path of this iteration and five iterations ago, respectively. In the late iterations, if the J_{best} does not change in the five consecutive iterations, the volatilization coefficientρ increases, and the volatilization is enhanced, which makes the ants explore better solutions. The upper limit of the volatilization coefficient is 0.8.

Path smoothing

The path obtained by classic ant colony algorithm has many redundant nodes and corners, which not only affects the path length, but also is not conducive to reducing the energy consumption of the robot. Therefore, the improved ant colony algorithm needs to optimize the path nodes. In this paper, the connectivity processing and quadratic B-spline curve optimization method are proposed to optimize the nodes, which further shortens the path length and reduces the energy consumption of the robot (Tian, 2020a; Tian et al., 2021).

Aiming at the path feature of the ant colony algorithm, which is composed of a series of nodes, the connectivity principle is proposed. Due to the limitation of step size, there are many redundant nodes. Connectivity processing is an effective method for eliminating redundant nodes, and its principle is shown in Figure 7.

As can be seen from the Figure 7, the line between P1 and P5 does not cross obstacles, so the two nodes are connected. P2, P3, and P4 are redundant nodes. After connectivity processing, the corner is reduced, and the path length is shortened, which is beneficial to the robot.

The connectivity processing of the complete path is shown in Figure 7.

As can be seen from the Figure 7, the left side is the path obtained by the classic ant colony algorithm. There are redundant nodes, and the path length is 43.6985. The right side is the path processed by connectivity. The corner is reduced, and the path length is 39.1901. It can be seen that the effect of connectivity processing is significant.

In addition to redundant nodes, the path smoothing also includes the smoothness operation of the corner. Therefore, this paper introduces the quadratic B-spline curve optimization method to optimize the corner.

In 1946, Schoenberg proposed a spline-based approach to approximate curves. In 1972, based on Schoenberg's work, Gordon and Riesenfeld proposed B-spline curves and a series of corresponding geometric algorithms. The B-spline curve is the generalization of the Bezier curve, which solves the problem that the Bezier curve is difficult to smooth transition at the endpoint. Besides, the B-spline curve has higher accuracy (Zeng et al., 2019). The definition of the B-spline curve is as follows.

$$P(t) = \sum_{i=0}^{n} P_i N_{i,k}(t) \tag{28}$$

$$N_{i,k}(t) = \frac{1}{k!} * \sum_{j=0}^{k-i} (-1)^j * C_{k+1}^{j} * (t+k-i-j)^k \tag{29}$$

$$0 \leq t \leq 1, i = 0, 1, \ldots, k-1, C_{k+1}^{j} = \frac{(n+1)!}{j! * (n+1-j)!} \tag{30}$$

Where, $P_i$is the original endpoint, $N_{i,k}(t)$is the basic function, and$P(t)$is the set of points on the curve.

The quadratic B-spline method needs only three endpoints to construct a smooth curve. And it can meet the requirements of curve smoothness. Therefore, this paper selects the quadratic B-spline method to deal with the corner problem.

When$n = 2$in equation (28), the quadratic B-spline curve of the following form can be obtained through the spline basis function.

$$P(t) = \frac{1}{2} * (1-t)^2 * P_0 + \frac{1}{2} * (-2 * t^2 + 2 * t + 1) * P_1 + \frac{1}{2} * t^2 * P_2 \tag{31}$$

The quadratic B-spline method is shown below.

As can be seen from the Figure 8, the black path is the original path, and the red path at the corner is the new path generated by the quadratic B-spline method.

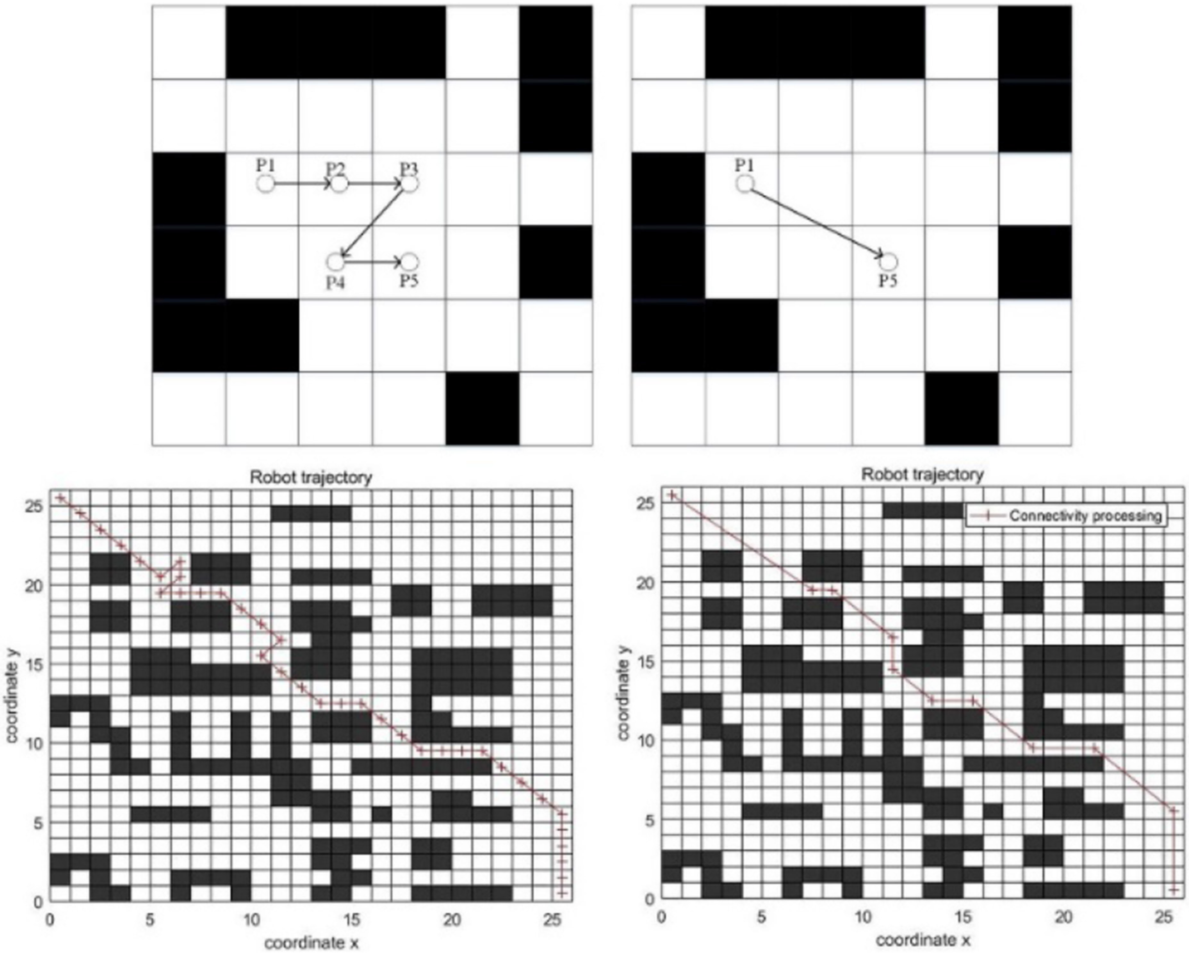


FIGURE 7
Connectivity processing.

The comparison results between the path after connectivity processing and the path generated by the quadratic B-spline method based on connectivity processing is shown in Figure 8.

As can be seen from the Figure 8, the corner processed by the quadratic B-spline method is smoother. The path length before processing is 39.1901, and the path length after processing is 38.9281.

There is almost no redundant node of the path processed by connectivity and the quadratic B-spline method, which is shorter than the path generated by the classic ant colony algorithm and is more suitable for mobile robots.

Algorithm flow

To sum up, the flow of the improved ant colony algorithm is shown in Figure 9.

The execution steps of the algorithm are as follows.

Step 1: Initialize the parameters of the improved ant colony algorithm and Floyd algorithm.

Step 2: Set up the grid map, initialize the pheromone matrix and tabu list according to the guidance path generated by the Floyd algorithm.

Step 3: Build candidate solutions according to the tabu list and state transition rules and select the next node by roulette principle.

Step 4: Determine whether the ants fall into the deadlock. If so, execute the fallback strategy until the ants get out of the trap. Otherwise, continue step 5.

Step 5: Update the tabu list and record the path nodes and length.

Step 6: Determine whether the ant reaches the target node. If so, continue step 7. Otherwise, return to step 3.

Step 7: Determine whether the number of ants reaches the upper limit. If so, continue step 8. Otherwise, return to step 3.

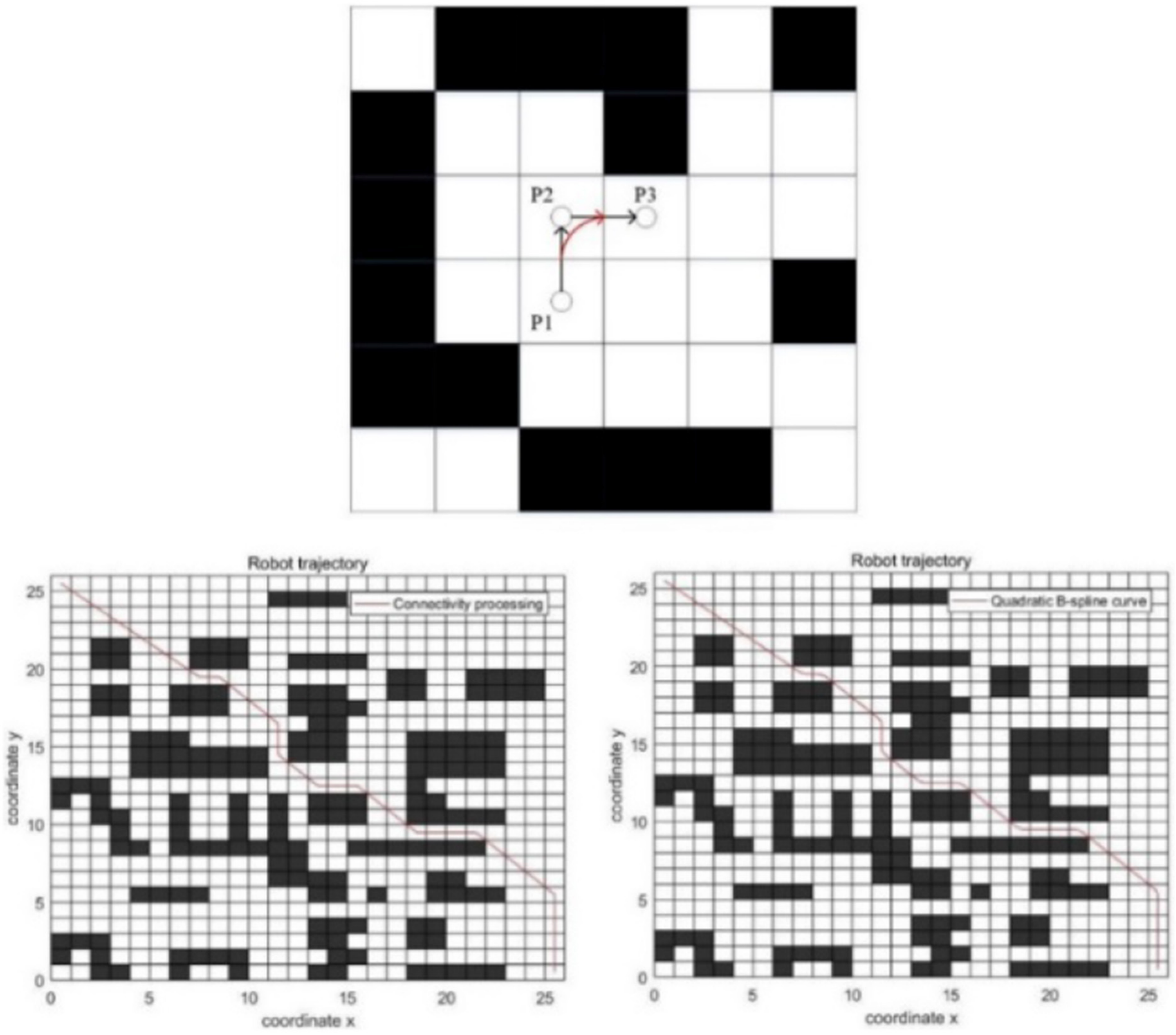


FIGURE 8
Quadratic B-spline curve.

Step 8: Smooth the path and update the global pheromone.

Step 9: Determine whether the maximum number of iterations is reached. If so, outputs the optimal solution and ends. Otherwise, return to step 3.

Step 10: Draw the algorithm iteration diagram and the optimal path curve.

Experimental results and discussions

In this section, the effectiveness of the improved algorithm in path planning is verified through different scenarios. All experiments were performed using the same PC. The MATLAB (R2016b) programming platform was used to encode and implement all algorithms. In order to obtain real experimental results and avoid accidental situations, all experiments were carried out independently under the same experimental conditions.

The 26 × 26 scale grid map is adopted in this paper. There are three different environments, namely, the concentrated obstacle environment, the partially dispersed obstacle environment, and the decentralized obstacle environment. The algorithm in this paper, the classic ant colony algorithm, and the algorithm of Li et al. (2021), Luo et al. (2020), and Akka and Khaber (2018) are compared experimentally. The algorithm parameters are set as shown in Table 2.

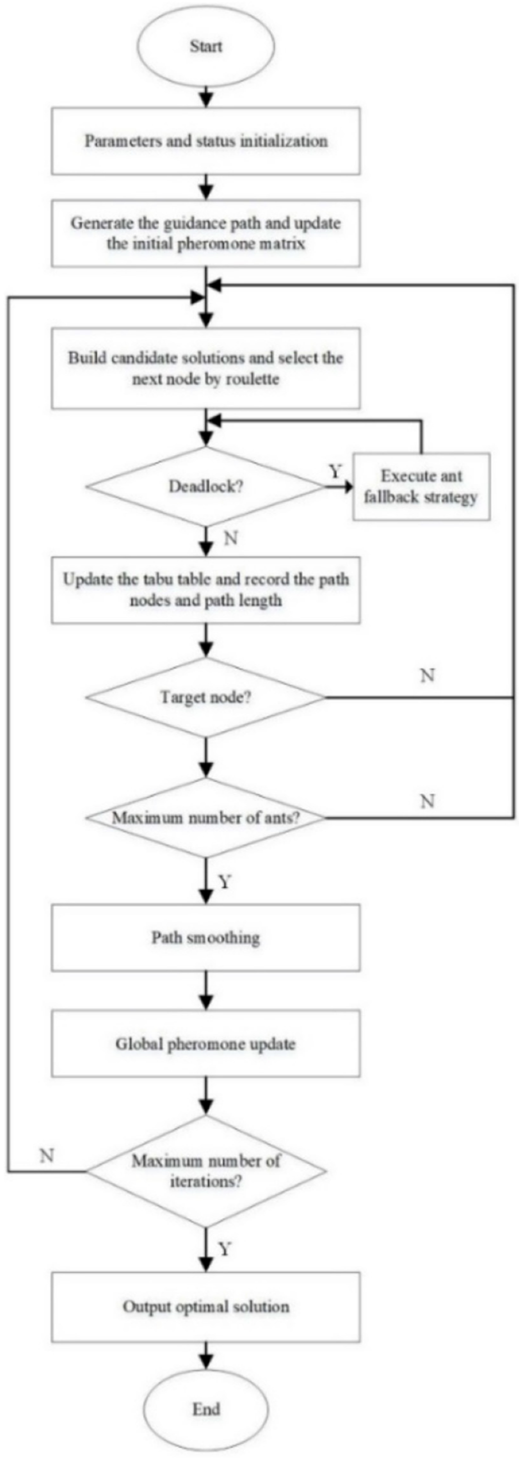

Start
Parameters and status initialization
Generate the guidance path and update the initial pheromone matrix
Build candidate solutions and select the next node by roulette
Deadlock?
Y
Execute ant fallback strategy
N
Update the tabu table and record the path nodes and path length
Target node?
N
Maximum number of ants?
N
Y
Path smoothing
Global pheromone update
N
Maximum number of iterations?
Y
Output optimal solution
End

FIGURE 9
Flow chart of improved ant colony optimization algorithm.

TABLE 2 Parameter setting.

Parameter	
Starting point S	1
Target point E	676
Maximum number of iterations K	100
The number of ants M	50
Pheromone heuristic factor α	1
Expected heuristic factor β	6
Pheromone volatilization factor ρ	0.6
Pheromone intensity factor Q	1
Pheromone penalty evaporation coefficient λ	15

Concentrated obstacle environment

In the concentrated obstacle environment with the 26 × 26 scale grid, the experimental results of five algorithms are shown in Figure 10.

The specific results of the experiment are shown in Table 3. Index 1 is the average path length, index 2 is the optimal path length, index 3 is the average number of iterations, and index 4 is the average number of corners.

It can be seen from Figure 10 and Table 3 that the comprehensive performance of the improved algorithm in this paper is the best in the concentrated obstacle environment. In terms of the optimal path length, the improved algorithm is 4.29% less than the classic ant colony algorithm, 2.82% less than the algorithm in Li et al. (2021), 8.10% less than the algorithm in Luo et al. (2020), and 2.04% less than the algorithm in Akka and Khaber (2018). In terms of the average path length, the improved algorithm is 4.43, 2.68, 9.70, and 1.99% less than other algorithms, respectively. In terms of the average number of iterations, the improved algorithm is 63 times less, 2 times more, 5 times less and 3 times less than other algorithms, respectively. In terms of the average number of corners, the improved algorithm is 41.67, 22.22, 53.33, and 30% less than other algorithms respectively. To sum up, in the concentrated obstacle environment, the performance of the improved algorithm in this paper is better than the other four algorithms, including the classic algorithm.

Partially decentralized obstacle environment

In the partially decentralized obstacle environment with the 26 × 26 scale grid, the experimental results of five algorithms are shown in Figure 11.

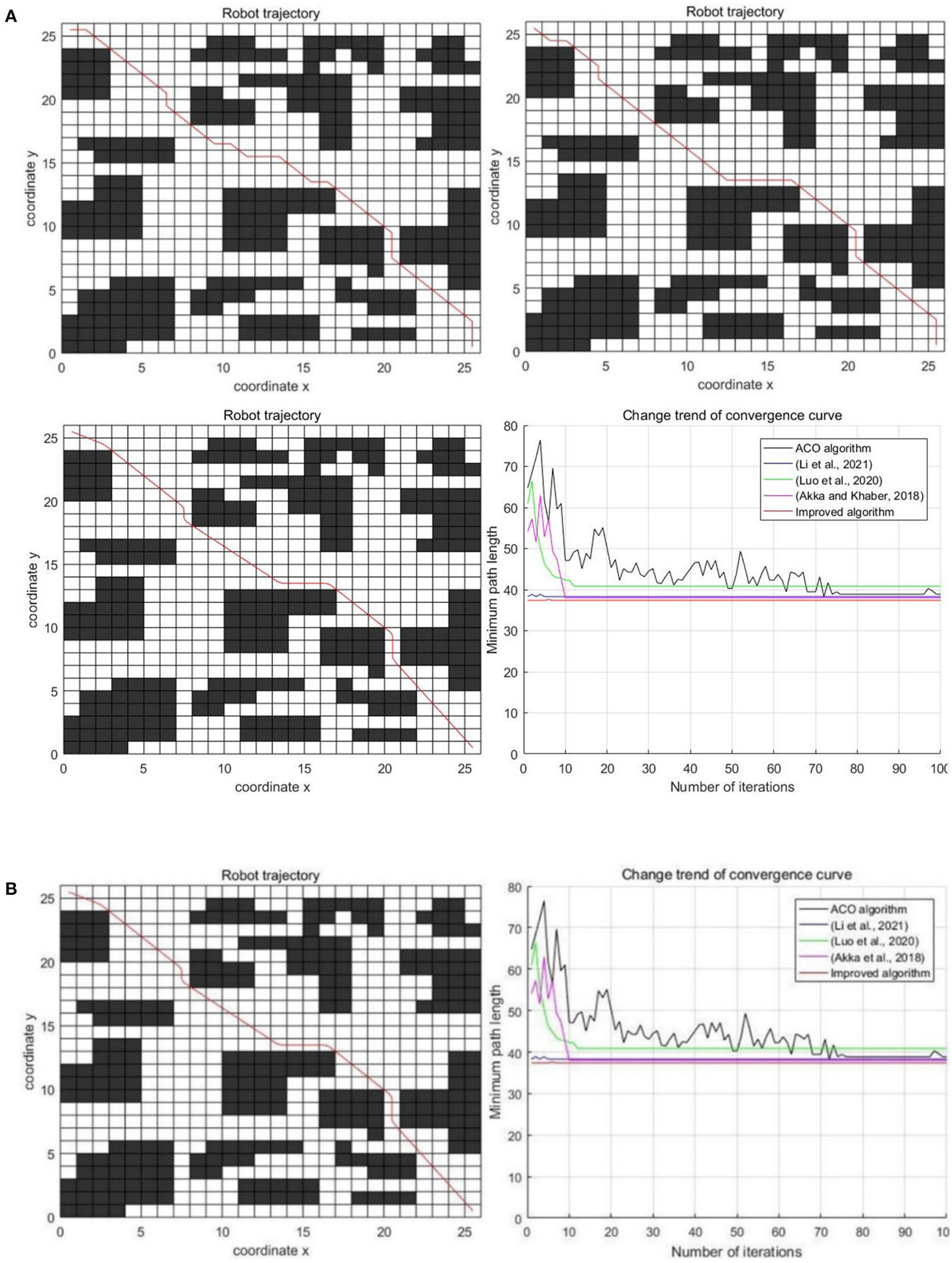


FIGURE 10
Experimental results of three algorithms in the concentrated obstacle environment. **(A)** Classic ant colony algorithm, Luo et al. (2020), Li et al. (2021), and Akka and Khaber (2018) **(B)** Improved algorithm and comparison of five algorithms.

TABLE 3 Comparison of five algorithms.

	Index	Concentrated obstacle environment	Partially decentralized obstacle environment	Decentralized obstacle environment
Classic ACO	1	39.2843	41.4578	43.2763
	2	38.8701	40.0416	41.4558
	3	70	65	75
	4	12	14	18
Li et al. (2021)	1	38.5772	39.9771	48.6639
	2	38.2843	39.1127	46.2543
	3	5	10	18
	4	9	12	15
Luo et al. (2020)	1	41.5772	40.4056	43.2132
	2	40.4807	39.6985	41.7990
	3	12	9	8
	4	15	15	16
Akka and Khaber (2018)	1	38.3045	40.8078	41.3356
	2	37.9793	39.6853	40.9214
	3	10	10	12
	4	10	13	16
Improved algorithm	1	37.5438	39.0204	39.6872
	2	37.2033	38.9281	39.1280
	3	7	11	10
	4	7	9	11

The specific results of the experiment are shown in Table 3.

As can be seen from Figure 11 and Table 3, the performance of the improved algorithm in this paper is still better than that of other algorithms in the partially decentralized obstacle environment. In terms of the optimal path length, the improved algorithm is 2.87% less than the classic ant colony algorithm, 0.47% less than the algorithm in Li et al. (2021), 1.94% less than the algorithm in Luo et al. (2020), and 1.91% less than the algorithm in Akka and Khaber (2018). In terms of the average path length, the improved algorithm is 5.88, 2.39, 3.43, and 4.38% less than other algorithms respectively. In terms of the average number of iterations, the improved algorithm is 54 times less, 1 time more, 2 times more and 1 time more than other algorithms respectively. In terms of the average number of corners, the improved algorithm is 35.71, 25, 40, and 30.77% less than other algorithms respectively. It can be seen from the above that the performance of the improved algorithm in this paper still has certain advantages in the partially decentralized obstacle environment.

Decentralized obstacle environment

In the decentralized obstacle environment with the 26×26 scale grid, the experimental results of five algorithms are shown in Figure 12. The specific results of the experiment are shown in Table 3.

It can be seen from Figure 12 and Table 3 that the improved algorithm in this paper has more obvious advantages than other algorithms in the decentralized obstacle environment. In terms of the optimal path length, the improved algorithm is 5.62% less than the classic ant colony algorithm, 15.41% less than the algorithm in Li et al. (2021), 6.39% less than the algorithm in Luo et al. (2020), and 4.38% less than the algorithm in Akka and Khaber (2018). In terms of the average path length, the improved algorithm is 8.29, 18.45, 8.16, and 3.99% less than other algorithms, respectively. In terms of the average number of iterations, the improved algorithm is 65 times less, 8 times less, 2 times more and 2 times less than other algorithms, respectively. In terms of the average number of corners, the improved algorithm is 38.89, 26.67, 31.25, and 31.25% less than other algorithms, respectively. From the above comparisons, as the complexity of the environment increases, the improved algorithm in this paper always has significant advantages.

From the above experiments, it can be seen that in the simple environment, except for the classic ant colony algorithm, the other three algorithms are close to the improved algorithm. As the complexity of the environment increases, the indicators of the five algorithms have changed, and the performance of the improved algorithm has always remained stable, which has been better than the other four algorithms, including the classic

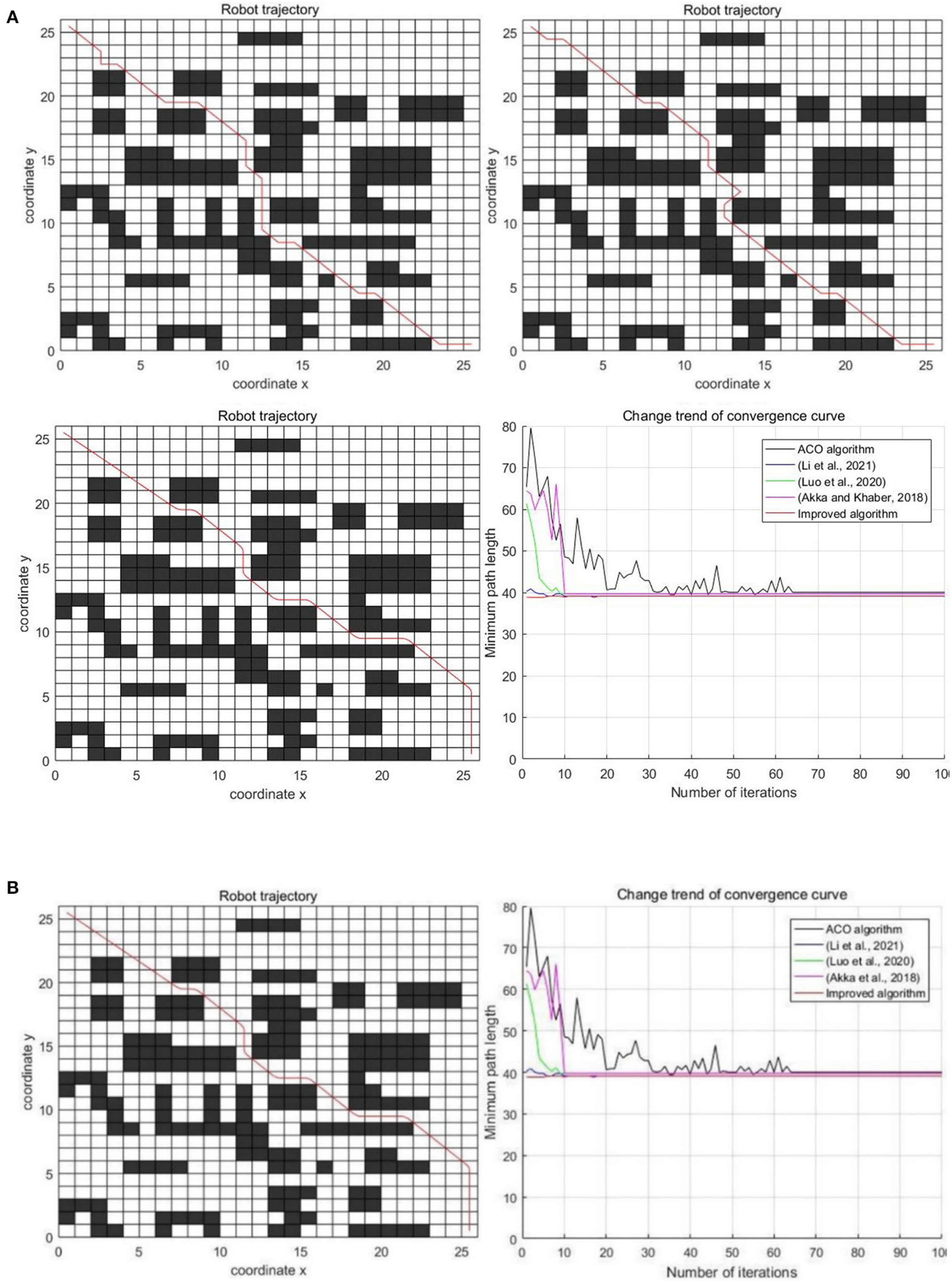


FIGURE 11
Experimental results of three algorithms in the partially decentralized obstacle environment. **(A)** Classic ant colony algorithm, Luo et al. (2020), Li et al. (2021), and Akka and Khaber (2018). **(B)** Improved algorithm and comparison of five algorithms.

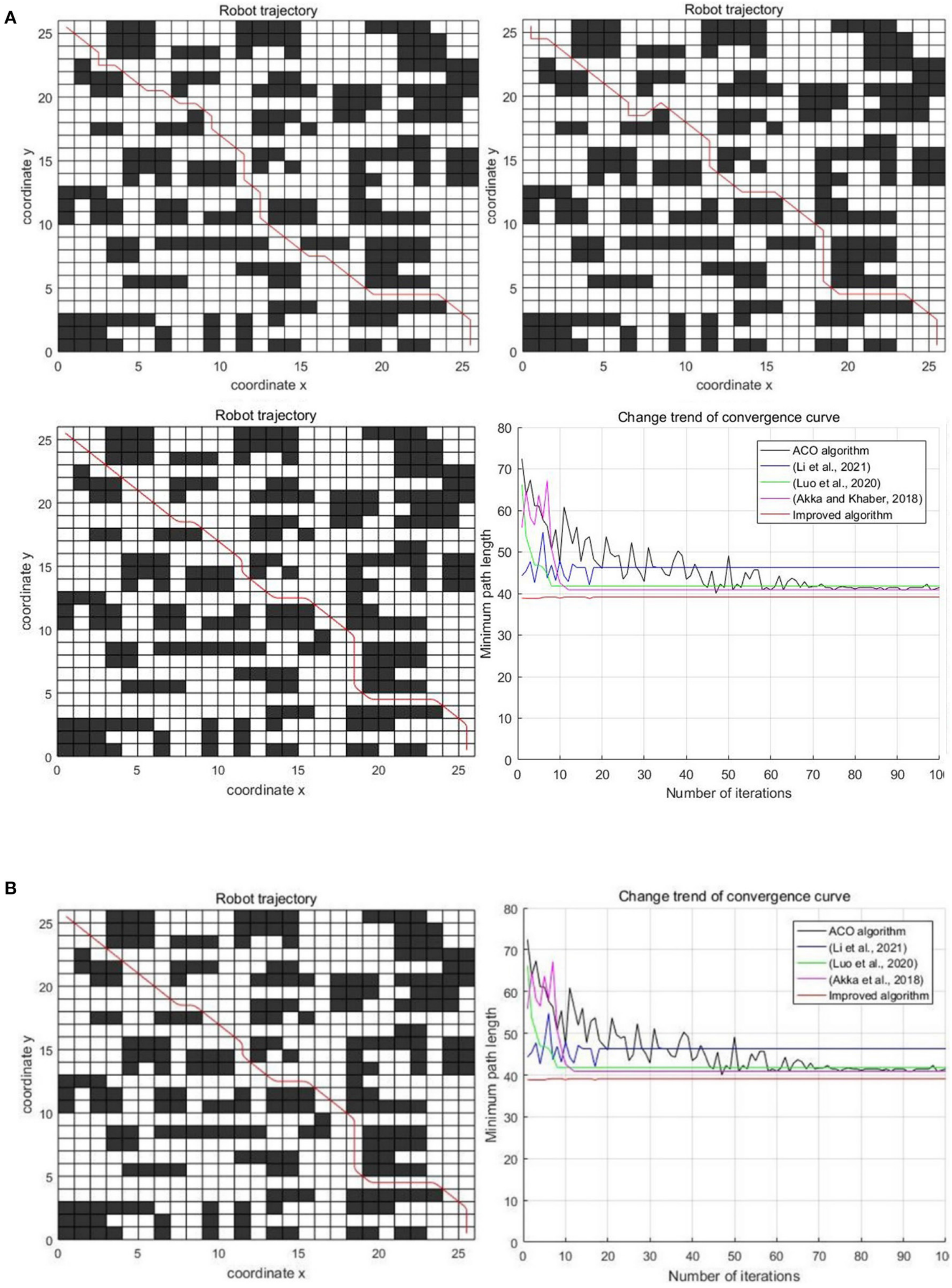


FIGURE 12
Experimental results of three algorithms in the decentralized obstacle environment. **(A)** Classic ant colony algorithm, Luo et al. (2020), Li et al. (2021), and Akka and Khaber (2018). **(B)** Improved algorithm and comparison of five algorithms.

algorithm. Among the five algorithms, the improved algorithm is the best, which is most conducive to the energy-saving and stable operation of the robot.

Conclusion

The ant colony algorithm is widely used in robot path planning. However, the classic ant colony algorithm still has the problems of slow convergence speed and easily fall into the local optimal solution. Therefore, this paper proposes an improved ant colony algorithm. Firstly, the Floyd algorithm is introduced to generate the guidance path to optimize the initial pheromone matrix and effectively accelerate the initial convergence speed of the ant colony algorithm. Ant fallback strategy can help avoid ants dying due to the deadlock dilemma and improve the global search ability of the algorithm. The improved heuristic function proposed by referring to the gravity concept in the APF method accelerates the convergence speed of the ant colony algorithm. It makes up for the influence of the fallback strategy on the convergence rate. The pheromone updating method based on a multi-objective optimization idea and dynamic principle considers the path length, path security, and path energy consumption. It helps the ant colony algorithm avoid the local optimal solution and improves the comprehensive performance of the algorithm, which is more suitable for mobile robots. Connectivity processing and the quadratic B-spline method effectively reduce the redundant nodes of the path, improve the smoothness of the path and further shorten the path length.

Through experimental comparisons, as can be seen, the improved algorithm has strong stability. From the simple obstacle environment to the complex obstacle environment, it can always maintain the optimal comprehensive performance, the shortest path, and the least corner. The problems of the classic ant colony algorithm has been solved. In addition, the multi-objective optimization idea and the node optimization method introduced in the improved algorithm can effectively help the mobile robot to save energy and improve the work efficiency.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://pan.baidu.com/s/1QQiLWLKRe3_SM1QdnwwaqA?pwd=100d.

Author contributions

LW and HW proposed this contribution, verified, and concluded simulation results. XC and BW gave suggestions for manuscript writing. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the National Natural Science Foundation of China under Grant (31901400 and 61903351), Natural Science Foundation of Zhejiang under Grant LY22F030009, Special Project for Cultivating Young Scientific and Technological Talents (Class A) (2022YW20), National Key Technologies Research and Development of China (2018YFB2101004).

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Akka, K., and Khaber, F. (2018). Mobile robot path planning using an improved ant colony optimization. *Int. J. Adv. Robot. Syst.* 15:1729881418774673. doi: 10.1177/1729881418774673

Ali, H., Gong, D. W., Wang, M., and Dai, X. L. (2020). Path planning of mobile robot with improved ant colony algorithm and MDP to produce smooth trajectory in grid-based environment. *Front. Neurorobotics* 14:44. doi: 10.3389/fnbot.2020.00044

Chen, H., Niu, L., and Ji, Y. (2020). Research on path planning of robot based on adaptive ACS fused with SHAA neural network. *Meas. Control* 53, 1911–1919. doi: 10.1177/0020294020959751

Chen, X., and Gao, P. (2020). Path planning and control of soccer robot based on genetic algorithm. *J. Ambient Intell. Humaniz. Comput.* 11, 6177–6186. doi: 10.1007/s12652-019-01635-1

Chen, Y., Bai, G., Zhan, Y., Hu, X., and Liu, J. (2021). Path planning and obstacle avoiding of the USV based on improved ACO-APF hybrid algorithm with adaptive early-warning. *IEEE Access* 9, 40728–40742. doi: 10.1109/ACCESS.2021.3062375

Dai, W., Wang, L., Wang, B., Cui, X., and Li, X. (2022). Research on WNN greenhouse temperature prediction method based on GA. *Phyton Int. J. Exp. Bot.* 91, 2283–2296. doi: 10.32604/phyton.2022.021096

Dai, X., Long, S., Zhang, Z., and Gong, D. (2019). Mobile robot path planning based on ant colony algorithm with A* heuristic method. *Front. Neurorobotics* 13:15. doi: 10.3389/fnbot.2019.00015

Guo, X., Ji, M., Zhao, Z., Wen, D., and Zhang, W. (2020). Global path planning and multi-objective path control for crewless surface vehicle based on modified particle swarm optimization (PSO) algorithm. *Ocean Eng.* 216, 107693–107708. doi: 10.1016/j.oceaneng.2020.107693

Hao, Z., and He, S. (2008). Some discussions on the Floyd algorithm. *J. Chongqing Univ. Technol.* 22, 156–159. doi: 10.3969/j.issn.1674-8425-B.2008.05.039

Khaksar, W., Hong, T., Khaksar, M., and Motlagh, O. (2012). Sampling-based tabu search approach for online path planning. *Adv. Robot.* 26, 1013–1034. doi: 10.1163/156855312X632166

Li, E., and Wang, Y. (2022). Research on obstacle avoidance trajectory of mobile robot based on improved artificial potential field. *Comput. Eng. Appl.* 58, 296–304. doi: 10.3778/j.issn.1002-8331.2108-0122

Li, K., Liu, S., Hu, Q., and Tang, Y. (2021). Path planning for improved ant colony optimization algorithm based on corner constraints. *J. Comput. Appl.* 41, 2560–2568. doi: 10.11772/j.issn.1001-9081.2020111713

Li, X., and Wang, L. (2020). Application of improved ant colony optimization in mobile robot trajectory planning. *Math. Biosci. Eng.* 17, 6756–6774. doi: 10.3934/mbe.2020352

Luo, Q., Wang, H., Zheng, Y., and He, J. (2020). Research on path planning of mobile robot based on improved ant colony algorithm. *Neural Comput. Appl.* 32, 1555–1566. doi: 10.1007/s00521-019-04172-2

Lyu, D., Chen, Z., Cai, Z., and Piao, S. (2021). Robot path planning by leveraging the graph-encoded Floyd algorithm. *Futur. Gener. Comp. Syst.* 122, 204–208. doi: 10.1016/j.future.2021.03.007

Mac, T., Copot, C., Tran, D., and DeKeyser, R. (2016). Heuristic approaches in robot path planning: a survey. *Robot. Auton. Syst.* 86, 13–28. doi: 10.1016/j.robot.2016.08.001

Ouyang, X., and Yang, S. (2014). Obstacle avoidance path planning of mobile robot based on potential field grid method. *Control Eng. China* 21, 134–137. doi: 10.3969/j.issn.1671-7848.2014.01.031

Pan, Z., Li, D., Yang, K., and Deng, H. (2019). Multi-robot obstacle avoidance based on the improved artificial potential field and PID adaptive tracking control algorithm. *Robotica* 37, 1883–1903. doi: 10.1017/S026357471900033X

Shi, W., and Wang, K. (2009). Floyd algorithm for the shortest path planning of mobile robot. *Chin. J. Sci. Instrum.* 30, 2088–2092. doi: 10.19650/j.cnki.cjsi.2009.10.014

Tao, Y., Gao, H., Ren, F., Chen, C., Wang, T., Xiong, H., et al. (2021). A mobile service robot global path planning method based on ant colony optimization and fuzzy control. *Appl. Sci.* 11, 3605–3619. doi: 10.3390/app11083605

Tian, Z. (2020a). Preliminary research of chaotic characteristics and prediction of short-term wind speed time series. *Int. J. Bifurcation Chaos* 30: 2050176. doi: 10.1142/S021812742050176X

Tian, Z. (2020b). Short-term wind speed prediction based on LMD and improved FA optimized combined kernel function LSSVM. *Eng. Appl. Artif. Intell.* 91:103573. doi: 10.1016/j.engappai.2020.103573

Tian, Z. (2021a). Approach for short-term traffic flow prediction based on empirical mode decomposition and combination model fusion. *IEEE Trans. Intell. Transp. Syst.* 22, 5566–5576. doi: 10.1109/TITS.2020.2987909

Tian, Z. (2021b). Modes decomposition forecasting approach for ultra-short-term wind speed. *Appl. Soft. Comput.* 105:107303. doi: 10.1016/j.asoc.2021.107303

Tian, Z., and Chen, H. (2021a). A novel decomposition-ensemble prediction model for ultra-short-term wind speed. *Energy Conv. Manag.* 248:114775. doi: 10.1016/j.enconman.2021.114775

Tian, Z., and Chen, H. (2021b). Multi-step short-term wind speed prediction based on integrated multi-model fusion. *Appl. Energy* 298:117248. doi: 10.1016/j.apenergy.2021.117248

Tian, Z., Li, H., and Li, F. (2021). A combination forecasting model of wind speed based on decomposition. *Energy Rep.* 7, 1217–1233. doi: 10.1016/j.egyr.2021.02.002

Tian, Z., Li, S., and Wang, Y. (2020). A prediction approach using ensemble empirical mode decomposition-permutation entropy and regularized extreme learning machine for short-term wind speed. *Wind Energy* 23, 177–206. doi: 10.1002/we.2422

Wang, L. (2020). Path planning for unmanned wheeled robot based on improved ant colony optimization. *Meas. Control* 53, 1014–1021. doi: 10.1177/0020294020909129

Wang, L., Liu, L., Qi, J., and Peng, W. (2020a). Improved quantum particle swarm optimization algorithm for offline path planning in AUVs. *IEEE Access* 8, 143397–143411. doi: 10.1109/ACCESS.2020.3013953

Wang, L., and Wang, B. (2020a). Construction of greenhouse environment temperature adaptive model based on parameter identification. *Comput. Electron. Agric.* 174:105477. doi: 10.1016/j.compag.2020.105477

Wang, L., and Wang, B. (2020b). Greenhouse microclimate environment adaptive control based on a wireless sensor network. *Int. J. Agric. Biol. Eng.* 13, 64–69. doi: 10.25165/j.ijabe.20201303.5027

Wang, L., Wang, B., and Zhu, S. (2020b). Multi-model adaptive fuzzy control system based on switch mechanism in a greenhouse. *Appl. Eng. Agric.* 36, 549–556. doi: 10.13031/aea.13837

Wang, X., Shi, H., and Zhang, C. (2020c). Path planning for intelligent parking system based on improved ant"' colony optimization. *IEEE Access* 8, 65267–65273. doi: 10.1109/ACCESS.2020.2984802

Xiao, S., Tan, X., and Wang, J. (2021). A simulated annealing algorithm and grid map-based UAV coverage path planning method for 3D reconstruction. *Electronics* 10, 853–868. doi: 10.3390/electronics10070853

Xiong, N., Zhou, X. Z., Yang, X. Q., Xiang, Y., and Ma, J. Y. (2021). Mobile robot path planning based on time taboo ant colony optimization in dynamic environment. *Front. Neurorobotics* 15:642733. doi: 10.3389/fnbot.2021.642733

Xiong, X. Y., Min, H. T., Yu, Y. B., and Wang, P. Y. (2020). Application improvement of A* algorithm in intelligent vehicle trajectory planning. *Math. Biosci. Eng.* 18:1. doi: 10.3934/mbe.2021001

Yang, H., Qi, J., Miao, Y., Sun, H., and Li, J. (2019). A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization. *IEEE Trans. Ind. Electron.* 66, 8557–8566. doi: 10.1109/TIE.2018.2886798

Yang, J. (2020). Application of Floyd algorithm in designing a coastal tourism route optimization system. *J. Coast. Res.* 106, 668–671. doi: 10.2112/SI106-151.1

Yao, Y., Liang, X., Li, M., Yu, K., Chen, Z., Ni, C., et al. (2021). Path planning method based on D* lite algorithm for unmanned surface vehicles in complex environments. *China Ocean Eng.* 35, 372–383. doi: 10.1007/s13344-021-0034-z

Yi, G., Feng, Z., Mei, T., Li, P., Jin, W., and Chen, S. (2019). Multi-AGVs path planning based on improved ant colony algorithm. *J. Supercomput.* 75, 5898–5913. doi: 10.1007/s11227-019-02884-9

Zeng, D., Zhu, P., Zhang, P., and Chen, H. (2019). Cubic B-spline curve for obstacle avoidance trajectory planning of unmanned vehicle. *J. Tongji Univ.* 47, 159–163.

Zhang, C., Pu, J., and Si, Y. (2021). An adaptive improved ant colony system based on population information entropy for path planning of mobile robot. *IEEE Access* 9, 24933–24945. doi: 10.1109/ACCESS.2021.3056651

Zhang, C., Pu, J., Si, Y., and Sun, L. (2020). Survey on application of ant colony algorithm in path planning of mobile robot. *Comput. Eng. Appl.* 56, 10–19. doi: 10.3778/j.issn.1002-8331.1912-0160

Zhang, W., Ma, Y., Zhao, H., Zhang, L., Li, Y., and Li, X. (2019). Obstacle avoidance path planning of intelligent mobile based on improved fireworks-ant colony hybrid algorithm. *Control Decis.* 34, 335–343. doi: 10.13195/j.kzyjc.2017.0870

Zheng, Y., Luo, Q., Wang, H., Wang, C., and Chen, X. (2020). Path planning of mobile robot based on adaptive ant colony algorithm. *J. Intell. Fuzzy Syst.* 39, 5329–5338. doi: 10.3233/JIFS-189018

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 02 September 2022
DOI 10.3389/fnbot.2022.990453

OPEN ACCESS

EDITED BY
Yan Wu,
Institute for Infocomm Research
(A*STAR), Singapore

REVIEWED BY
Gongfa Li,
Wuhan University of Science and Technology, China
Ameer Tamoor Khan,
Hong Kong Polytechnic University,
Hong Kong SAR, China

*CORRESPONDENCE
Dapeng Chen
dpchen@nuist.edu.cn

RECEIVED 10 July 2022
ACCEPTED 17 August 2022
PUBLISHED 02 September 2022

CITATION
Liu J, Gu Q, Chen D and Yan D (2022) VSLAM method based on object detection in dynamic environments. *Front. Neurorobot.* 16:990453. doi: 10.3389/fnbot.2022.990453

VSLAM method based on object detection in dynamic environments

Jia Liu, Qiyao Gu, Dapeng Chen* and Dong Yan

School of Automation, C-IMER, B-DAT, CICAEET, Nanjing University of Information Science & Technology, Nanjing, China

Augmented Reality Registration field now requires improved SLAM systems to adapt to more complex and highly dynamic environments. The commonly used VSLAM algorithm has problems such as excessive pose estimation errors and easy loss of camera tracking in dynamic scenes. To solve these problems, we propose a real-time tracking and mapping method based on GMM combined with YOLOv3. The method utilizes the ORB-SLAM2 system framework and improves its tracking thread. It combines the affine transformation matrix to correct the front and back frames, and employs GMM to model the background image and segment the foreground dynamic region. Then, the obtained dynamic region is sent to the YOLO detector to find the possible dynamic target. It uses the improved Kalman filter algorithm to predict and track the detected dynamic objects in the tracking stage. Before building a map, the method filters the feature points detected in the current frame and eliminates dynamic feature points. Finally, we validate the proposed method using the TUM dataset and conduct real-time Augmented Reality Registration experiments in a dynamic environment. The results show that the method proposed in this paper is more robust under dynamic datasets and can register virtual objects stably and in real time.

KEYWORDS

dynamic target detection, VSLAM, YOLOv3, GMM, Kalman filter

1. Introduction

Initially, SLAM (Simultaneous Localization and Mapping) was proposed to solve the problem of robot movement in an unknown environment. After the robot observes the environment, it immediately feeds back its posture and movement trajectory, and constructs a map of the environment simultaneously. The early SLAM system mainly used single-line lidar, sonar and other sensors to realize its own positioning. With the rapid development of computer vision, the VSLAM (Visual SLAM) system with the help of cameras has begun to become the mainstream of research by various teams due to its convenient use and low cost. The VSLAM system has been well applied in the fields of augmented reality (Calloway and Megherbi, 2020), driverless driving (Nguyen et al., 2018), and robotics (Liu, 2021). Virtual objects registered with VSLAM technology have better stability and accuracy in today's popular augmented reality applications. To achieve a more immersive visual experience in the dynamic environment of mobile devices, the VSLAM system with AR (Augmented Reality) technology needs a more excellent background update mode.

Many Augmented Reality Registration methods are based on the front-end visual odometry of SLAM systems, while many VSLAM systems are usually built on static environments. However, the real environment is much more complicated than the ideal environment. Dynamic objects such as people and cars are often unavoidable in scenarios such as classrooms, hospitals, and outdoor shopping places. Those VSLAM systems built on a static environment have poor adaptability to dynamic and complex scenes, leading to substantial errors in the obtained map points and pose matrix (Cheng et al., 2019). Indirectly, it will cause problems such as drift of virtual objects registered in the world coordinate system. Aiming at the problems of excessive pose estimation error and easy loss of camera tracking in the commonly used VSLAM algorithm in high dynamic scenes, we propose a real-time tracking and mapping method based on GMM combined with YOLOv3. This method can guarantee the robust registration of virtual objects in dynamic environments.

To ensure that the camera produces robust results when moving, we combine the affine transformation matrix to correct the continuous frame image (Sun et al., 2022). In the non-key frame stage, we employ GMM (Gaussian Mixture Model; Stauffer and Grimson, 1999) to model the background image, effectively utilizes the global discontinuity of the keyframe, and increases the GMM training time to improve the training effect of the background model. When creating the keyframe, we combine the image frame of the previous time series to segment the foreground dynamic area, and provide prior knowledge for the YOLO detector. To improve the detection accuracy of the dynamic target of VSLAM, we employ the observation value provided by YOLO (You Only Look Once) v3 (Redmon and Farhadi, 2018) in the tracking thread to predict the area of the dynamic target in real time. Our method combines the dynamic area detected by YOLO with the dynamic area obtained after GMM training, and uses the IOU (Intersection Over Union) result as the probability information to obtain the largest possible dynamic target. We choose YOLOv3 because it is a single-stage detector that can achieve good accuracy while meeting the real-time nature of Augmented Reality Registration. Moreover, compared with traditional methods such as frame difference method, optical flow method, and background removal method, YOLOv3 has better real-time performance and robustness. But the disadvantage is that YOLOv3 does not provide prior knowledge that can identify dynamic regions. Our method is complementary to both. It uses a GMM model to train background images, estimates motion regions when creating new keyframes, and provides priors for YOLOv3. At the same time, YOLOv3 meets the real-time and robustness requirements, and can achieve dynamic target detection between consecutive frames.

The most traditional tracking algorithm is the filtering algorithm based on the Bayesian framework (Goan and Fookes, 2020). It utilizes prior information to make an optimal estimation of the state of the target in the current frame to track the target, such as the Kalman filter (Xu Y. et al., 2018) and the Particle filter (Chakravarty et al., 2017). In actual operation, the observed value is easily affected by factors such as the camera itself and lighting. The traditional Kalman filter will affect the next predicted value when an error occurs in the observed value, leading to the accumulation of errors. We provide an improved Kalman filter method that uses the first N groups of observations to establish a nonlinear fitting curve to predict the next set of observations. Then, we employ an evaluation metric to determine whether to choose the predicted "observed value" or the value observed by the system. After the update, it can obtain a more accurate and practical background. We employ this improved Kalman filter algorithm to predict and track YOLO objects to ensure the continuity of the regional frame. The real-time accuracy of background map construction determines the reliability of VSLAM applications in many directions. This method can accurately eliminate the dynamic noise during the mapping thread, and obtain a good mapping effect, providing a good mapping environment for Augmented Reality Registration. The system diagram is shown in Figure 1.

The rest of this paper is structured as follows. The second section describes the related work of VSLAM implementation in a dynamic environment. The third section describes the method of this paper in detail. The fourth section gives the experimental results. The fifth section gives some conclusions and analysis of the experiment.

2. Related work

2.1. Classic VSLAM

The classic VSLAM system has gone through a series of explorations and improvements and has formed an effective execution framework. Davison et al. (2007) first proposed MonoSLAM, a SLAM scheme based on a monocular camera. Klein and Murray (2007) proposed a keyframe mechanism in the PTAM scheme, which realized the parallelization of tracking and mapping, distinguished the front and back ends for the first time, and used nonlinear optimization as the back-end optimization scheme. The two earliest proposed VSLAM solutions have problems such as small application scenarios and easy tracking loss. However, these innovative framework ideas have been used to this day.

Subsequently, scholars began to improve the front-end visual odometer. At present, the feature point method composed of key points and descriptors is the most mainstream front-end algorithm, such as SIFT (Lowe, 2004), SURF (Bay et al., 2006), and ORB (Rublee et al., 2011). This kind of method

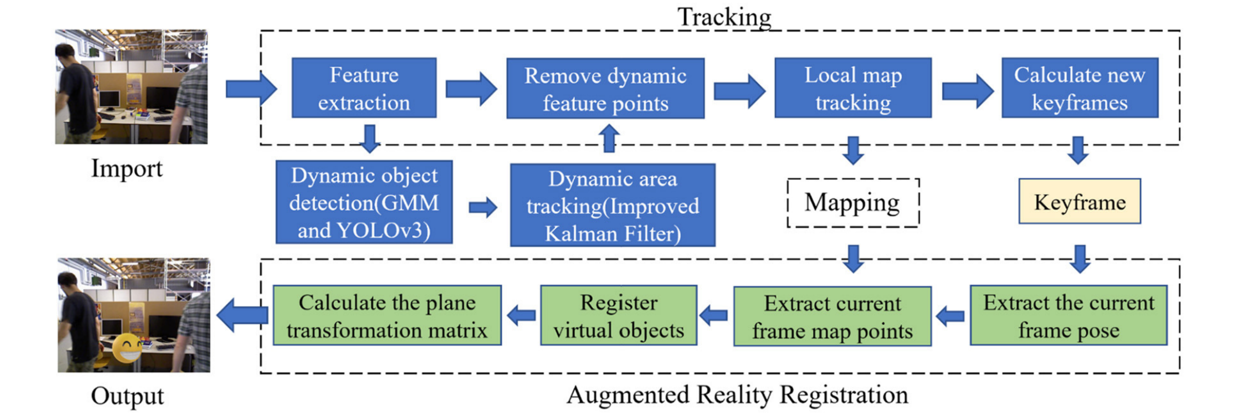

Tracking
Feature extraction
Remove dynamic feature points
Local map tracking
Calculate new keyframes
Import
Dynamic object detection(GMM and YOLOv3)
Dynamic area tracking(Improved Kalman Filter)
Mapping
Keyframe
Output
Calculate the plane transformation matrix
Register virtual objects
Extract current frame map points
Extract the current frame pose
Augmented Reality Registration

FIGURE 1
System diagram. This system mainly consists of three parts: tracking thread, mapping thread, and Augmented Reality Registration thread. The tracking thread uses GMM combined with the YOLOv3 method for dynamic object detection and uses the improved Kalman filter method for dynamic object tracking. Next, it removes the feature points of dynamic objects in the keyframes and transfers the keyframes that retain the static feature points to the augmented reality registration thread. In the Augmented Reality Registration thread, it registers virtual objects through the map points and camera poses of the current frame and tracks virtual objects through matrix calculation.

is stable and relatively mature. Another front-end algorithm is the direct method based on pixel brightness information. Engel et al. (2014, 2015) proposed LSD-SLAM, which uses the direct method to build maps. It has the advantages of fast speed and good real-time performance, but it is very sensitive to the camera's internal parameters and exposure, and it is easy to lose when the camera moves quickly. Forster et al. (2014) proposed high-speed real-time mapping using the sparse direct method SVO. It is extremely fast, but due to abandoning the calculation of the descriptor, its pose estimation is prone to cumulative errors. When the camera moves quickly, the location information is easy to lose, and it is difficult to relocate after being lost. In the case of much noise in the dynamic environment, the result of this method is still not satisfactory.

Mur-Artal et al. (2015) proposed a monocular ORB-SLAM system. ORB-SLAM utilizes unified ORB features in each link of tracking, mapping, relocation and loop detection. It has high computational efficiency, good rotation and scaling invariance (Mur-Artal and Tardós, 2017; Campos et al., 2021), and its performance in a dynamic environment can be further improved. Many SLAM systems improved through dynamic target detection and deep learning are also implemented under the ORB-SLAM's framework.

2.2. Dynamic VSLAM scheme of deep learning and geometric view

In terms of dynamic target detection, traditional methods are greatly affected by scene brightness changes, noise, etc., and there will be false detections and missed detections in the target detection process. This also leads to drift during target tracking, which in turn affects the accuracy of target tracking (Huang et al., 2022).

In recent years, dynamic target detection has put forward higher tracking accuracy and target number requirements, and many excellent SLAM frameworks have emerged continuously (Gehrmann et al., 2019). In the past, semantic segmentation was used to train static objects to generate semantic maps that increase the amount of information. For example, the semantic map construction proposed by Goerke and Braun (2009). When building a map in a dynamic environment, it is necessary to segment and remove dynamic characters. The method proposed by Wang et al. (2016) is a new method for classifying human motion regions. It divides human activities into categories and predicts the travel of the human body through general movement patterns. But this method is only suitable for fixed cameras. Riazuelo et al. (2017) proposed a semantic SLAM method in dense portrait scenes. This method solves the limitation of camera fixation. It completes a complete SLAM system based on the visual odometer (Wang et al., 2007). It can detect which are dynamic objects, but it cannot detect changes caused by static objects. Bescos et al. (2018) proposed DynaSLAM, which employs the Mask RCNN (Ammirato and Berg, 2019) to arrange the scene prior knowledge and estimate the possible moving targets through the geometric view method. This method removes the feature points of the moving target through a mask to maintain the algorithm's accuracy. After removing the dynamic target, the previously observed static information is used to repair the area. However, when repairing the occluded background of the current frame, using the pixel area corresponding to the last frame will cause the accumulation of errors. Zhong et al. (2018) proposed Detect-SLAM. It combines the single shot multibox detector (Liu et al., 2016) on the basis of ORB-SLAM, and uses semantic information to eliminate the influence of dynamic targets in SLAM. In addition, it also contains a method to propagate the dynamic possibilities of each feature point in real time, which solves the problem of delay in the transmission of semantic information. Yu et al. (2018) proposed DS-SLAM, which utilizes the optical flow method to track feature points and employs RANSAC (Raguram et al., 2012) to eliminate outliers and calculate the basic matrix. The dynamic and static points are judged based on the distance from the feature point to the epipolar line. Then, SegNet (Badrinarayanan et al., 2017) is used to divide the dynamic area and eliminate the feature points of the dynamic area. However, because the semantic information is not comprehensive enough and the semantics are untargeted, there are problems in dynamic filtering in some aspects, such as gesture occlusion. Xiao et al. (2019) proposed Dynamic-SLAM. Based on the same work as DynaSLAM (Bescos et al., 2018), it reduces the dynamic error and builds a better map (Fan et al., 2020).

These SLAM systems for dynamic scenes generally use semantic information, either using geometric information or a simple combination of methods for dynamic object detection. Cui and Ma (2019) proposed SOF-SLAM, which combines semantics and optical flow methods. It fully utilizes the dynamic characteristics of features hidden in semantic and geometric information. Cui and Ma (2020) proposed SDF-SLAM, utilizes a depth filter to describe each map point's inverse depth, updates the inverse depth of the 3D map points in the Bayesian framework, and divides the 3D map points into active or inactive points. However, the problem of using a semantic combination of VSLAM is still undeniable. They all rely heavily on the training effect of the network model. The prestage workload is enormous, but it can only be divided and cannot be tracked well. For the classic network model of target detection, the candidate area method proposed by RCNN is very time-consuming and cannot be run in real time. YOLO innovatively proposed merging the candidate area and recognition process in RCNN to increase computing speed significantly (Redmon and Farhadi, 2018).

Inspired by deep learning, improved view geometry methods are also constantly advancing, and new system models appear. Sun et al. (2017) proposed a motion removal method based on RGB-D cameras. Since this method relies on the maximum posterior scheme to determine the foreground, the segmentation results are limited (Sun et al., 2019). Xu X. et al. (2018) proposed a multi-view spectral clustering framework that combines multiple models together, integrating the affine, tomography, and basic matrix. Sun et al. (2018) proposed MR-SLAM, which improved their previous method (Sun et al., 2017) to model prospects in different classes, so the number of moving objects was not limited during segmentation. This method adds online learning capabilities, allowing it to update the foreground model incrementally. Although, MR-SLAM can effectively deal with dynamic factors, it consumes too much time in the process of precise detection and segmentation of moving targets, and it is not outstanding in real-time performance. Cheng et al. (2019) inspired by deep neural networks, proposed SMR-SLAM, which employs the Bayesian formula to solve the probability distribution of the feature point area of the geometric view. Small-probability events are eliminated to help SLAM distinguish dynamic regions as much as possible. It can learn and perform well in scenes with low dynamics, but the error is more evident in highly dynamic scenes or excessive complexity. Liu et al. (2022) optimized the sparse point cloud map through the YOLOv4 framework to enhance the interactivity of the robot. Gao et al. (2020) proposed a feature map fusion one-shot multi-box detector, which has higher detection accuracy and real-time performance compared to SSD and DSSD methods. The occlusion of the hand is also one of the reasons for the failure of virtual object registration, and the detection of the hand is also very necessary (Gao et al., 2019). YOLOv3 has better real-time and accuracy in hand detection, and can detect hand occlusion in real time, helping to complete better virtual object registration.

We provide the method of GMM combined with YOLOv3. Our method uses the ORB-SLAM2 framework and improves its tracking thread to analyze dynamic targets. The method provided in this paper detects and tracks dynamic targets, eliminates dynamic points in real time, and optimizes the update mode of the background to ensure the accuracy of pose solving and map creation.

3. Method

3.1. Dynamic target detection

3.1.1. Moving target detection algorithm based on GMM

GMM is a method to accurately quantify things with a Gaussian probability density function and decompose them into several models based on a Gaussian probability density function. The GMM application in background elimination establishes a Gaussian mixture model for each pixel in the video frame. If the pixel model has a significant weight, it is indicated as a background pixel; otherwise, it is a foreground image. Since background pixels often occupy high weights, the generated data is more trustworthy on the background pixels so that GMM can distinguish the foreground and the background in the long-term observation sequence generated by the video.

Our method employs the ORB feature extraction and matching to extract feature points, calculates the affine matrix M by matching the front and back frames, corrects the current frame image through the affine matrix M. Then, the method employs GMM to learn background pixels and segment the foreground and background images by finding the pixel group

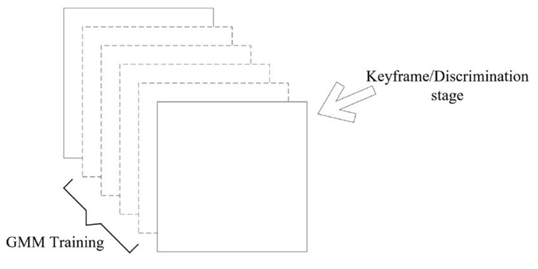


FIGURE 2
GMM dynamic solution.

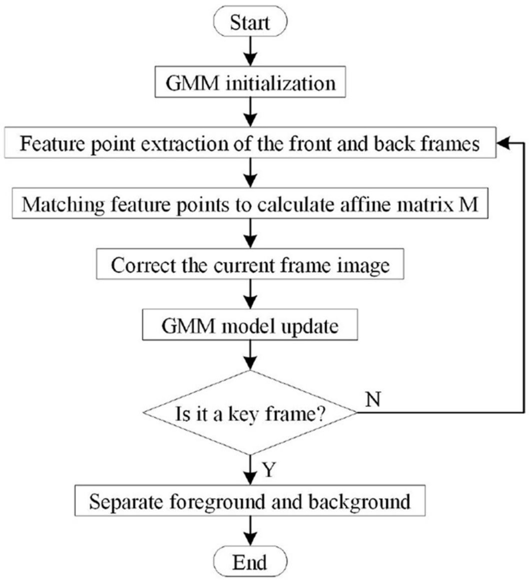

Start
GMM initialization
Feature point extraction of the front and back frames
Matching feature points to calculate affine matrix M
Correct the current frame image
GMM model update
Is it a key frame?
N
Y
Separate foreground and background
End

FIGURE 3
Flow chart of dynamic target detection combining GMM and affine matrix.

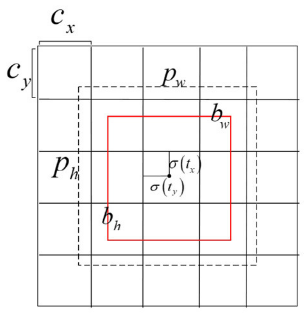


FIGURE 4
Schematic diagram of bounding box regression.

closest to the background. The dynamic target detection process is as follows:

a. In the initialization phase, the method completes the initial setting of the parameters of the GMM model.
b. In the non-key frame stage, the feature points are extracted from the VSLAM to match the continuous frame images. Calculate the affine transformation matrix M by this method, utilize the M matrix to make affine changes, and set the threshold (the threshold is 20 in this paper) to correct the current frame.
c. At the same time, to reduce the image shift caused by the affine matrix error, the method uses the mean filter to process the images before and after the transformation.
d. Our method trains the corrected image on the GMM Gaussian mixture model. It combines the image frame of the previous time series to determine the foreground dynamic area when the keyframe is created.

The principle diagram of dynamic target detection combining VSLAM and GMM is shown in Figure 2, and the specific flow chart is shown in Figure 3.

3.1.2. Target detection algorithm based on YOLOv3

The moving target detection algorithm based on the GMM extracts the keyframes and then performs the difference. Between keyframes, we employ the YOLOv3 algorithm to detect objects that may need to be tracked. YOLOv3 is a single-stage detector that can meet real-time performance for Augmented Reality Registration while maintaining accuracy compared to methods *via* R-CNN. The method divides the input image into a 13×13 table and then lets each cell detect the target. The bounding box and the discrimination probability value through each grid are obtained to judge whether the target object and the position information and probability information of the target area in the grid. The dimensional clustering method on the bounding box is chosen to select 3 scales and nine types of bounding boxes, the bounding box detection problem is converted into a regression problem, and the 4 coordinates t_x, t_y, t_w, t_h (as shown in formulas 1–4) of each bounding box are predicted. For the problem of bounding box regression, for the 13×13 feature scale map, we utilize three bounding boxes of 10×13, 16×30, and 33×23 pixels; for the 26×26 feature scale map, we utilize three bounding boxes of 30×61, 62×45, 59×119 pixels; for the 52×52 feature scale map, we utilize three bounding boxes of 116×90, 156×198, 373×326 pixels. The regression diagram of the bounding box is shown in Figure 4.

The definition formula for the bounding box is as follows:

$$b_x = \sigma\left(t_x\right) + c_x \tag{1}$$

$$b_y = \sigma\left(t_y\right) + c_y \tag{2}$$

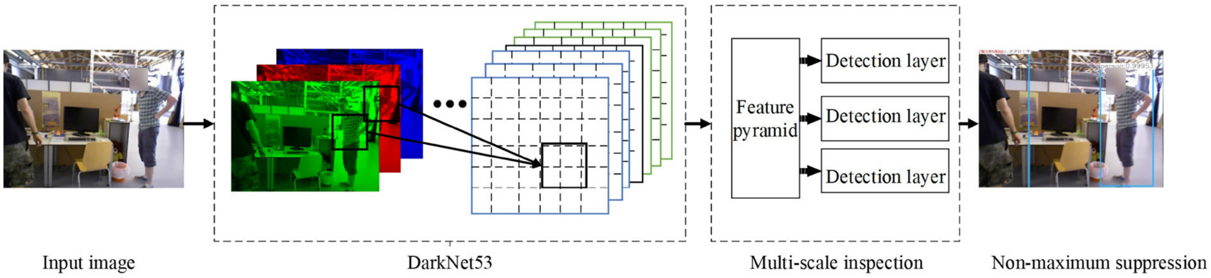


FIGURE 5
YOLOv3 detection process.

$$b_w = p_w e^{t_w} \tag{3}$$

$$b_h = p_h e^{t_h} \tag{4}$$

where t_x, t_y, t_w, and t_h represent the offset of x coordinate, y coordinate, width, and height offset, respectively. b_x, b_y, b_w, and b_h represent the result of the final goal box. $\sigma\ (x)$ represents the Sigmoid function. The result of x is normalized to speed up network convergence, where p_w and p_h are the width and height of the bounding box, respectively. The overall YOLOv3 detection process is shown in Figure 5.

In VSLAM, to ensure the accuracy of map point construction, it is necessary to eliminate all possible dynamic targets in the keyframe by comparing the information of the last frame when generating the keyframe. Due to the instability of the feature points, the calculated affine matrix has errors, so when the dynamic target is moving, the real-time calculation result using the frame difference method is often not satisfactory. Our method in this paper establishes dynamic candidate areas through keyframes. At the same time, it employs the YOLOv3 algorithm to receive each candidate area and discard candidate areas that cannot be identified. The method employs the GMM model to train the background image, estimates the motion area when creating new keyframes, provides prior knowledge for YOLOv3, and exploits the fast and robust advantages of YOLOv3 to achieve dynamic target detection between consecutive frames. With the advantage of discontinuous VSLAM keyframes in time series, each time a keyframe is established, this method analyzes the dynamic area to increase or decrease the dynamic tracking frame. This method can ensure the real-time performance of VSLAM and avoid the problem of local map tracking failure caused by too few map points due to multiple additions and reductions of candidate areas. The schematic diagram is shown in Figure 6.

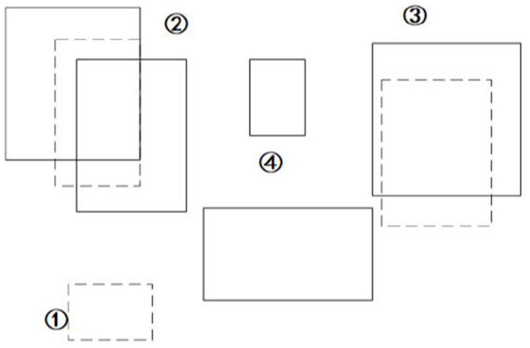


FIGURE 6
Schematic diagram of dynamic area detection.

The dotted box represents the dynamic candidate area provided by GMM, and the solid box represents all targets detected by YOLOv3. Then we use the IOU result as the probability information to get the largest possible dynamic target [such as at ②], discard the area where GMM dynamic detection fails [such as ①], discard other static targets obtained by YOLOv3 [such as ④], and detect the area (solid box) such as ②③.

3.2. Dynamic target tracking based on an improved Kalman filter

Multi-target detection algorithms are easily affected by factors such as illumination, occlusion, and pixel blur when moving (Li and Shi, 2019). The dynamic area will not disappear irregularly, so we build a tracking model to achieve multi-target tracking between two keyframes to ensure the continuity of the bounding box detected by YOLO. The Kalman filter algorithm itself is a linear system. Since the value observed in this paper is the state value, it is easy to estimate the value from the previous state to the next state by using the state transition matrix of the Kalman filter algorithm. The Kalman filter algorithm only considers the relationship between the upper and lower frames to a certain extent, so the Kalman filter algorithm needs a very accurate observation effect. However, in the actual operation process, the observed values are not

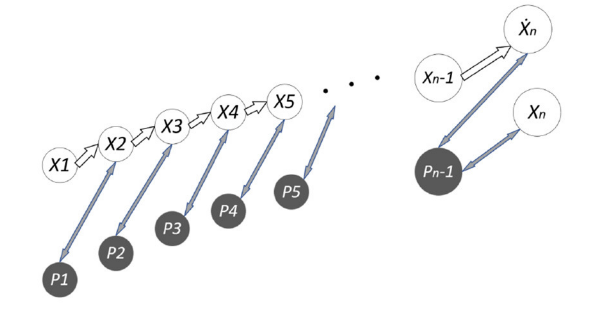


FIGURE 7
Improved observation value selection principle diagram.

necessarily accurate, whether due to the influence of the camera or lighting effects. To solve such problems, we propose an improved Kalman filter method. We exploit the improved Kalman filter to predict the maximum probability position and length and width information of the next frame. It uses its error covariance to calculate the predicted value of the state variable, find the observed value by combining the detection algorithm, correct the predicted value with Kalman gain, and finally obtain the optimal value of the variable.

The improved Kalman filtering algorithm exploits the first N groups of observations to establish a nonlinear fitting curve to predict the next group of observations. The algorithm uses an evaluation index to determine the selected predicted "observed value" or the value observed by the system. Since the feature points are affected by environmental factors or camera shake factors, linear fitting is performed according to the absolute values of the errors of the previous N-1 groups of predictions and observations. While ensuring the real-time performance of the algorithm, it can distinguish whether the target is moving fast or instantaneously due to observation errors. The improvement principle is shown in Figure 7 ($\dot{X}_n$ are fitted observations, and X_n is an actual observation. P_1, P_2... P_{n-1} represent the error covariance).

The observation values selected in this paper are the central pixels $\left(\frac{1}{2}\sum_{2i}^{2} x_{2i}, \frac{1}{2}\sum_{2i+1}^{3} y_{2i+1}\right)$ of the four boundary corners of the target image to input to the Kalman filter system to obtain the predicted value. Then, take the predicted point as the center, use the value of max $\left\{\left|x_j - x_i\right|\right\}$ obtained in the last frame as the width of the rectangle, and the value of max $\left\{\left|y_j - y_i\right|\right\}$ as the length of the rectangle, and then crop a new area. The camera pose is detected and calculated in this area to obtain a new set of measured values, and the rectangular area size and area of the next frame are updated from the measured values and the new boundary corner points.

First, we establish an 8-dimensional state vector and a 4-dimensional observation vector according to the linear condition satisfied by the Kalman filter. The 8-dimensional state vector values represent the center pixel position x and y, the aspect ratio and height of the bounding box, and their corresponding velocity values. Expressed by the equation of motion $x_k = A_k x_{k-1} + B_k \mu_k + w_k$, due to the lack of a control vector, B_k is set to a 0 vector, which satisfies the state transition matrix:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

This formula expresses the displacement of the previous state plus the unit velocity to represent the displacement of the current state, and considers the system error and the observation error, $w_k \sim N\left(0, Q_k\right), v_k \sim N\left(0, R_k\right)$. The observation equation is expressed as $z_k = H_k x_k + v_k$ according to the Kalman filter. Because of the special relationship between the observation equation and the state equation in this paper, H_k is a 4×8 matrix, where the observation equation is only related to the first four dimensions of the current state vector, that is, the displacement point, so take:

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6}$$

To satisfy the system's optimal estimation of the state equation, we modify the state value during the observation phase and introduce the covariance matrix to update:

$$P_{k|k-1} = A \cdot P_{k-1|k-1} \cdot A^T + Q \tag{7}$$

where $P_{k|k-1}$ represents the covariance matrix of the predicted state value and obtains the optimal estimation of the current state through the prediction result of the current system and the measurement of the current state:

$$x_{k|k} = x_{k|k-1} + K_k \left(z_k - H \cdot x_{k|k-1}\right) \tag{8}$$

where K_k represents the current Kalman gain coefficient, which is represented by the covariance matrix P and the measurement matrix H:

$$K_k = P_{k|k-1} \cdot \frac{H^T}{\left(H \cdot P_{k|k-1} \cdot H^T + R\right)} \tag{9}$$

We bring the Kalman gain at this time into the optimal estimation solution and exploit this gain to calculate the required covariance matrix value at the next moment:

$$P_{k|k} = \left(1 - K_k \cdot H\right) P_{k|k-1} \tag{10}$$

We can completely predict the center point's position at the next moment through observations. However, the selection of observations affects the stability of the entire system. In the case of minimal noise, if there is a significant error in the observed value, the predicted value will also be inaccurate. We improve the performance of the entire system by improving the selection of observations, and the method is as follows:

(1) Initialization phase

First, we establish a non-linear loss function model. Our method sets the model as:

$$f(x) - exp\left(a \cdot x^2 + b \cdot x + c\right) \qquad (11)$$

Secondly, we define *N* groups of observation data (*N* is set to 20 in this chapter), and establish the least square function through the observation data:

$$\text{again} \sum_{x} ||f(x) - \exp\left(a \cdot x^2 + b \cdot x + c\right)||_2 \qquad (12)$$

At the same time, we assign values to the initial values of the first N groups. If all the first N groups are assigned a value of 0, the finally obtained parameters are easy to fall into the local optimal solution, and the parameters to be sought are solved incorrectly in the initialization stage. Therefore, we add Gaussian disturbance to the value of f(x) and x to make them in a fluctuating state.

(2) Solving stage

To ensure that the data of a given fitting does not increase over time, the problem of incorrect fitting parameters and a significant increase in the number of calculations does not occur. Our method accepts new data while removing the old data to maintain it at the value of *N*. Within the parameter range. Our method uses the L-M method to iterate, and finally finds the solution of the unit at the next time through the known parameters, which is the "observation point" for solving the prediction.

(3) Judgment stage

Our method has obtained two sets of observation points: the observed points and the predicted "observation points." Of course, it is hoped that the actual observation points are accurate, but regardless of the presence of noise or the influence of light factors, the observed data may always be wrong. This paper introduces third-party evaluation indicators to determine which value is more accurate.

We assume that the previous observation data are accurate (or the observation data has been corrected), and there are also errors between the predicted value of the Kalman filter and the observation of the next frame, and the error may be small. Our method builds a set of fitting data by the absolute value of the error between the observation value of the next frame and the predicted value of Kalman filter. At the same time, we fit the linear equations with the previous N-1 sets of data, predict the "observed value" of the Nth set of data, and calculate the absolute value of the error between it and the Kalman predicted value. Finally, we judge which observation value is more reliable according to the error growth rate. The calculation function of the judgment is as follows:

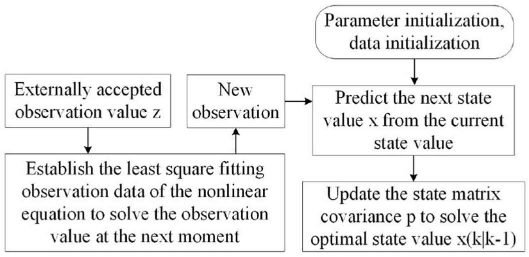

Parameter initialization, data initialization
Externally accepted observation value z
New observation
Predict the next state value x from the current state value
Establish the least square fitting observation data of the nonlinear equation to solve the observation value at the next moment
Update the state matrix covariance p to solve the optimal state value x(k|k-1)

FIGURE 8
Improved Kalman forecast update flow chart.

$$z_k = \min_{z_k} \left\{3 \times \left|\left|\widehat{z_k} - p_{k-1}\right| - g_k\right|\right\}, \left\{\left|\left|z_k - p_{k-1}\right| - g_k\right|\right\} \qquad (13)$$

where g_k is the predicted value of the error, $\widehat{z_k}$ is the predicted "observed value," z_k is the observation value of the system, and p_{k-1} is the predicted value of the last frame. In order to ensure the reliability of system prediction, we assign weight to both of them to avoid local optimization. Finally, the closer "observation point" is selected as the new observation point. The flow chart of the algorithm is shown in Figure 8.

3.3. Incremental model

The method provided in this paper performs the above dynamic target detection and tracking on the image sequence between every two keyframes. When VSLAM constructs a keyframe, it rejudges whether a new target area needs to be constructed. Therefore, the following incremental model is added during the keyframe construction to ensure that the dynamic increment can be tracked stably in the tracking thread or use the incremental model to determine whether to cancel tracking the lost target information. The incremental model is shown in Figure 9, where F_{Last} represents the last frame of the keyframe, F_{Cur} represents the current frame that can also be understood as a keyframe, and *Tracker* is the tracker designed in this paper.

4. Experiments

This experiment utilizes the dynamic objects dataset in the TUM dataset for dynamic target detection and the verification

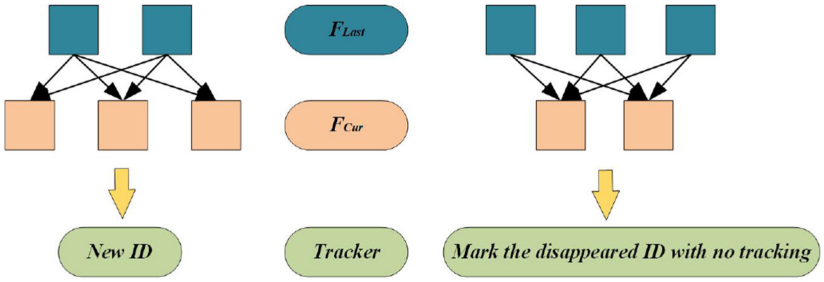


FIGURE 9
Incremental model.

TABLE 1 Dynamic target detection results.

Data sets	Temporal difference method	Optical flow method	Ours
fr2/desk_with_person	0.2514	0.4613	**0.5756**
fr3/sitting_static	0.1011	0.2167	**0.4783**
fr3/sitting_xyz	0.2331	0.5098	**0.5933**
fr3/sitting_halfsphere	0.4060	0.4200	**0.5749**
fr3/sitting_rpy	0.1991	0.2340	**0.6764**
fr3/warking_static	0.4788	0.6993	**0.6032**
fr3/warking_xyz	0.5423	0.5745	**0.7220**
fr3/warking_halfsphere	0.4421	0.6421	**0.6854**
fr3/warking_rpy	0.5322	0.3210	**0.6010**

Bold values mean the best result among the methods.

of the tracking algorithm based on the improved Kalman filter. Finally, the algorithm is integrated into the VSLAM to eliminate the dynamic target. We verify the effectiveness of the algorithm proposed in this paper by two metrics: ATE (absolute trajectory error) and RPE (relative pose error). The test platform for this experiment is Ubuntu 16.04, the primary language for building the platform is C++, and the Python environment is applied for ATE and RPE analysis.

4.1. Analysis of target detection results based on the dynamic environment

At present, there is no clear data set for dynamic target detection in a dynamic environment. In order to verify the robustness of the dynamic target detection algorithm proposed in this paper, we search for dynamic targets in the Dynamic Object dataset in the TUM dataset. First, we employ YOLOv3 to set prior knowledge to label dynamic targets artificially. Next, we find the IOU value of the target detected by YOLOv3 and the result of dynamic target detection. The larger the experimental result, the more concentrated the detection distribution and the higher the detection accuracy. In order to reflect the superiority of the detection algorithm proposed in this paper, this experiment employs the traditional frame difference method, optical flow method and other algorithms that are often used in dynamic target detection to compare. Table 1 shows the results (calculate the average IOU value for each frame detected under each data set). For multiple dynamic targets in an image frame, calculate the average value of IOU in the current frame and then map it to the global data set. The algorithm's performance in this paper on the data set fr2/desk_with_person is shown in Figure 10.

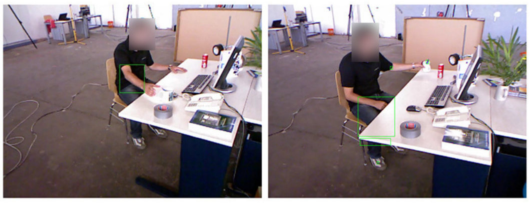


FIGURE 10
Dynamic detection performance effect of the data set.

It can be seen from the data in Table 1 that the target detection algorithm used in this paper effectively improves the detection accuracy of the dynamic region.

4.2. Analysis of long-term tracking results based on YOLOv3 and improved Kalman filter

This experiment utilizes the Dynamic Object dataset to verify the effectiveness of target tracking, and utilizes the MOT16 dataset to verify the robustness of the multi-target tracking algorithm used in this paper. This experiment utilizes YOLOv3 to detect pedestrians, and utilizes an improved Kalman filter algorithm to track the observation results provided by YOLOv3.

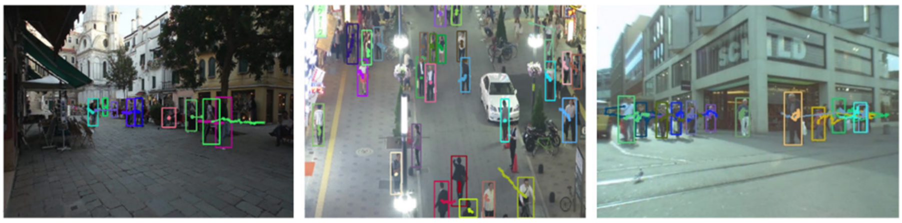


FIGURE 11
Operation effect of the tracking algorithm MOT16 in this paper.

We employ the Hungarian algorithm to find the match between the previous and next frames in terms of data association. The effect of running on MOT16 is shown in Figure 11.

The experimental results show that in a highly dynamic environment, the detection and tracking algorithm can better assign weights and find the best prediction results. It assigns the maximum possible motion trajectory to the target through cascade matching, avoiding the problem of target loss caused by occlusion.

4.3. Analysis of experimental results based on the VSLAM dynamic environment

The segmentation idea we adopt is that under the target area frame, the proportion of target pixels is always the larger one, so we perform a sliding window search according to the depth value of the depth image to search for the pixel area with the largest proportion (we divide the depth image pixels into 16. There are 16-pixel areas per copy to ensure that each pixel value from 0 to 255 can be searched). In the augmented reality technology, the reason for the deviation of the virtual object in the map is often the calculation error of the posture point. Therefore, we use two indicators, ATE and RPE, to verify the algorithm in this paper. At the same time, in order to ensure that the method can be effectively applied to the augmented reality environment, we exploit the TUM data set fr3/w_xyz combined with the Augmented Reality Registration algorithm for verification. The feature collection effect of our method under the TUM data set fr3/w_xyz is shown in Figure 12. The binary image on the left is the result of dynamic target segmentation, and the image on the right is the feature points detected by VSLAM.

We analyze the results of multiple dynamic data sets in the TUM data set, and employ the absolute trajectory error graph ATE to verify the algorithm in this paper. It directly measures the point difference between the real trajectory and the estimated trajectory. The longer the red segment, the larger the estimation error and the lower the positioning accuracy. The ground truth, the estimated camera motion, and the localization error for each camera pose are represented as the black, blue, and red segments, respectively. The algorithm proposed in this paper is compared and analyzed with ORB-SLAM2 (Mur-Artal and Tardós, 2017) and SMR-SLAM (Cheng et al., 2019). Figure 13 shows the analysis results of the performance comparison between the proposed algorithm and ORB-SLAM2 under the conditions of three dynamic data sets fr3/w_half, fr3/w_rpy, and fr3/w_xyz. Figure 14 shows the analysis results of the performance comparison between the proposed algorithm and SMR-SLAM under the conditions of three dynamic data sets fr3/w_half, fr3/w_xyz, and fr2/desk_with_person.

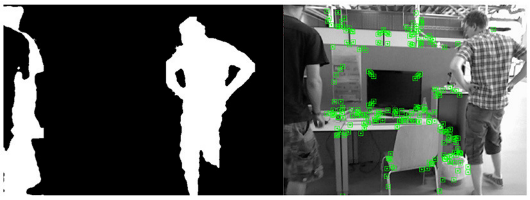


FIGURE 12
The feature collection effect of our method under the TUM data set fr3/w_xyz.

Through the analysis of Figures 13, 14, it can be seen from the results of absolute trajectory error analysis that the algorithm proposed in this paper has more significant advantages in dynamic scenes and still maintains good results in low-dynamic scenes.

To reflect that the algorithm in this paper can maintain stable and superior performance under different data sets, we employ the official ATE and RPE test files provided by TUM to test the fr2 and fr3 series of data sets and obtain the data results shown in Tables 2–4. RMSE is the mean root mean square error, and SD is the standard deviation, using ORB-SLAM2 (RGB-D) (Mur-Artal and Tardós, 2017), MR-SLAM (Sun et al., 2018), and SMR-SLAM (Cheng et al., 2019) as comparisons.

It can be seen from Tables 2–4 that the performance results of the VSLAM method proposed in this paper on the dynamic

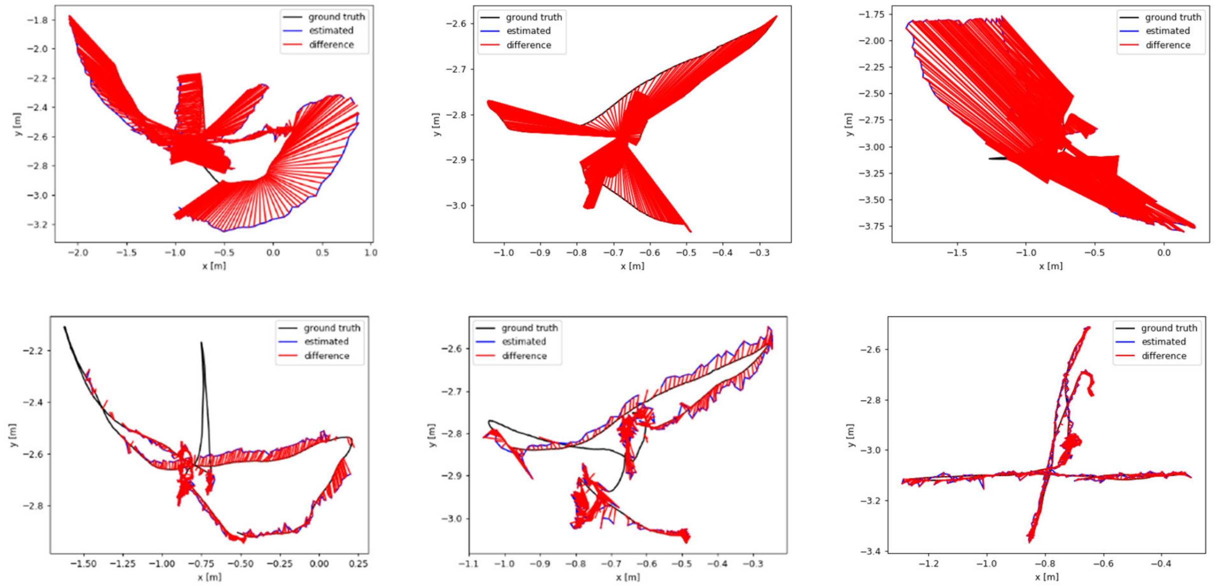


FIGURE 13
The **(top)** row is the trajectory error graph of ORB-SLAM2, and the **(bottom)** row is the trajectory error graph of ours. The ground truth, the estimated camera motion, and the localization error for each camera pose are represented as the black, blue, and red segments, respectively.

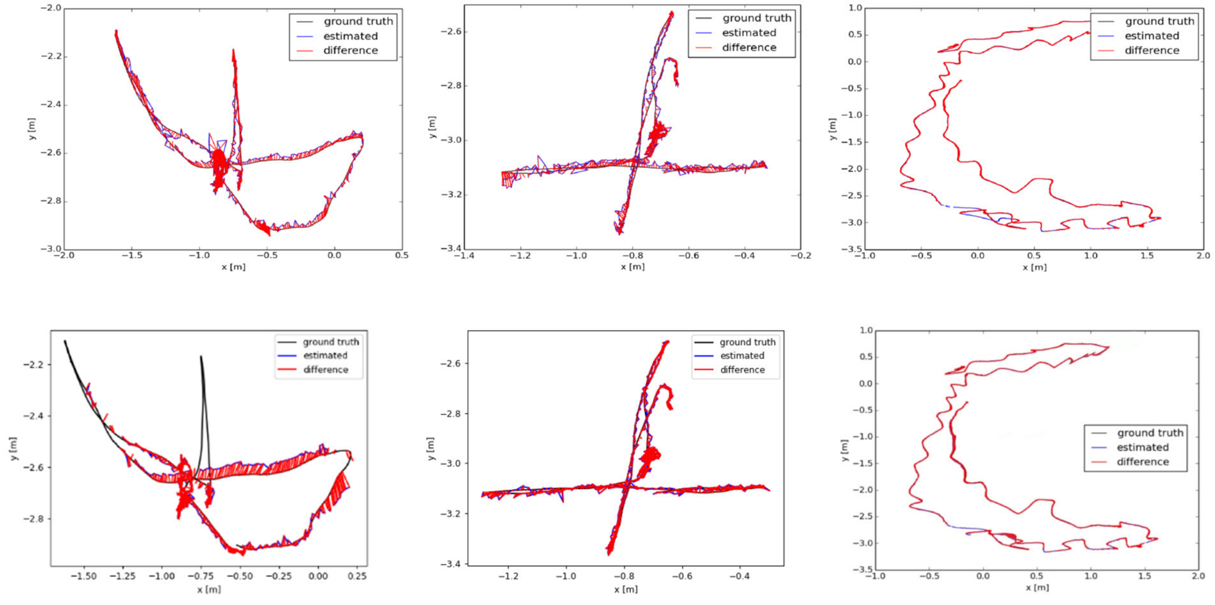


FIGURE 14
The **(top)** row is the trajectory error graph of SMR, and the **(bottom)** row is the trajectory error graph of ours. The ground truth, the estimated camera motion, and the localization error for each camera pose are represented as the black, blue, and red segments, respectively.

data set are much better than ORB-SLAM2. Compared with the more advanced VSLAM systems, MR-SLAM and SMR-SLAM, currently proposed, it also has an advantage. Although the performance on the low-dynamic dataset is slightly inferior to that of the SMR-SLAM algorithm, it still maintains a better advantage than ORB-SLAM2. This result is consistent with the results in Figures 13, 14. Our method occupies an absolute advantage in evaluating rotation drift, which can effectively avoid errors caused by dynamic target interference in many applications such as AR.

TABLE 2 Ate in meters for the experiments using ORB-SLAM2, MR-SLAM, SMR-SLAM, and Ours.

Data sets	ORB-SLAM2		MR-SLAM		SMR-SLAM		Ours	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
w_halfsphere	0.2668	0.1429	0.0668	0.0266	0.0352	0.0207	**0.0342**	**0.0206**
w_xyz	0.2774	0.1230	01230	0.0657	**0.0186**	**0.0098**	0.0331	0.0176
w_rpy	0.1677	0.0958	0.0729	0.0335	0.0436	0.0253	**0.0347**	**0.0160**
w_static	0.0250	0.0147	0.0334	0.0207	0.0238	0.0113	**0.0142**	**0.0071**
s_halfsphere	0.0219	0.0133	0.0664	0.0386	**0.0210**	**0.0127**	0.0438	0.0305
s_xyz	**0.0089**	**0.0046**	0.0514	0.0280	0.0138	0.0076	0.0255	0.0113
desk_person	**0.0056**	**0.0030**	0.0759	0.0313	0.0068	0.0031	0.0728	0.0207

Bold values mean the best result among the methods.

TABLE 3 Translational drift (RPE) in m/s for the experiments using ORB-SLAM2, MR-SLAM, SMR-SLAM, and Ours.

Data sets	ORB-SLAM2		MR-SLAM		SMR-SLAM		Ours	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
w_halfsphere	0.8078	0.4958	0.0611	0.0268	0.0816	0.0419	**0.0539**	**0.0301**
w_xyz	0.6181	0.3778	0.0668	0.0369	**0.0337**	**0.0162**	0.0470	0.0227
w_rpy	1.5083	0.9031	0.0968	0.0510	0.0337	0.0162	**0.0214**	**0.0134**
w_static	0.5436	0.3783	0.0307	0.0205	0.0829	0.0479	**0.0276**	**0.0165**
s_halfsphere	0.0326	0.0198	0.0547	0.0318	**0.0307**	**0.0183**	0.0654	0.0429
s_xyz	**0.0132**	**0.0063**	0.0357	0.0225	0.0242	0.0106	0.0363	0.0167
desk_person	0.0383	0.0228	0.0213	0.0151	0.0369	**0.0213**	**0.0121**	0.0646

Bold values mean the best result among the methods.

TABLE 4 Rotational drift (RPE) in m/s for the experiments using ORB-SLAM2, MR-SLAM, SMR-SLAM, and Ours.

Data sets	ORB-SLAM2		MR-SLAM		SMR-SLAM		Ours	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
w_halfsphere	17.7267	10.2391	1.9004	0.7629	1.1556	0.5359	**1.0076**	**0.4283**
w_xyz	10.9428	7.1977	1.5950	0.8236	0.7473	0.4333	**0.7427**	**0.4266**
w_rpy	28.0287	17.3043	2.5936	1.3210	1.6024	0.9284	**1.0777**	**0.5112**
w_static	9.9384	6.9106	0.8998	0.6470	1.1366	0.6269	**0.4823**	**0.2975**
s_halfsphere	0.8217	0.3594	2.2677	1.3861	**0.8038**	**0.3495**	1.0254	0.4454
s_xyz	**0.5775**	0.3016	1.0362	0.5304	0.6905	0.3474	0.6601	**0.2998**
desk_person	1.4668	0.6857	**0.7744**	**0.4767**	1.3784	0.6742	1.4410	0.6932

Bold values mean the best result among the methods.

4.4. Experiments with augmented reality registration

4.4.1. Robustness experiments

The above experiment is the result analysis of the VSLAM algorithm we proposed under the dynamic data set. At the same time, in order to verify the robustness of our proposed method in the Augmented Reality system, we utilize the fr3/w_xyz data set to test, select the appropriate Kth frame, insert a virtual object, and observe the dynamic performance of the virtual object during operation. The experimental results of the ORB-SLAM2 method are shown in Figure 15. The experimental results of our proposed method are shown in Figure 16.

We choose to insert a virtual square in the 20th frame of the data set fr3/w_xyz. From frame 100 to frame 500, we sampled the result six times. In these six images, there are objects entering, a single object moving slowly, a single object moving quickly, multiple objects moving, the lens moving up and down, the lens moving left and right, and the lens rotating. Figures 15, 16 show the AR implementation effects of the ORB-SLAM2 method and the method in this paper, respectively. It can be seen from Figure 15 that under the influence of camera motion and video portrait motion, ORB-SLAM2 cannot accurately analyze the plane, and the error situation shown in Figure 15 often occurs. In terms of long-term attitude tracking, the ORB-SLAM2 method has attitude offset, which will also cause the inserted virtual object to not be in the original position. It can be seen from the results in Figure 16 that the VSLAM method proposed in this paper can accurately fit and create a virtual object, which greatly improves the registration of augmented reality and the tracking of virtual objects.

4.4.2. Real-time experiment

We conduct real-time comparison experiments of Augmented Reality Registration in a dynamic laboratory environment. We register virtual objects at the 50th, 100th, 200th, 350th, and 500th frames after initialization, calculate the response time, and compare with our method through several classical algorithms such as SURF+KLT, ORB-SLAM2,

TABLE 5 Real-time analysis (ms).

Frame	SURF+KLT	VINS-Mono	ORB-SLAM2	Ours
50	42.5	26.3	22.3	**23.9**
100	44.5	29.5	22.5	**23.1**
200	49.0	44.9	23.5	**24.5**
350	50.2	69.9	23.2	**24.9**
500	50.5	108.6	23.4	**24.8**

Bold values mean the best result among the methods.

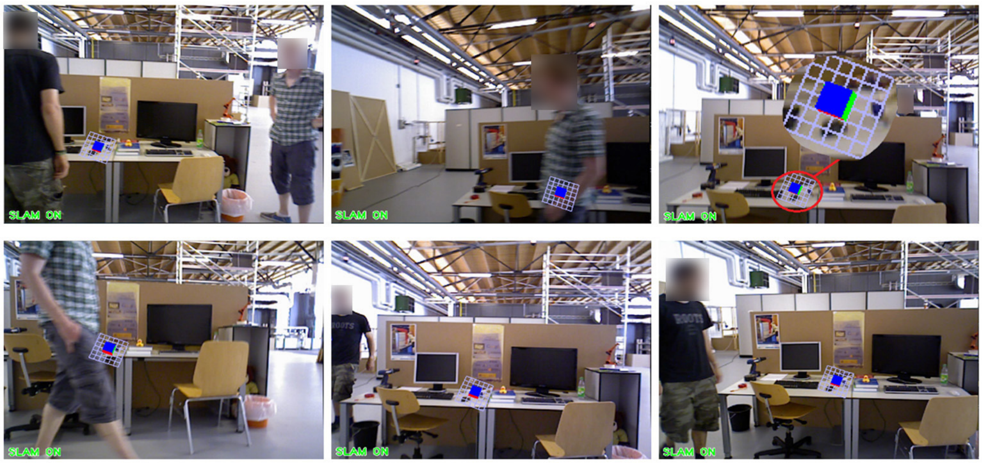


FIGURE 15
The application effect of ORB-SLAM2 method in augmented reality experiment.

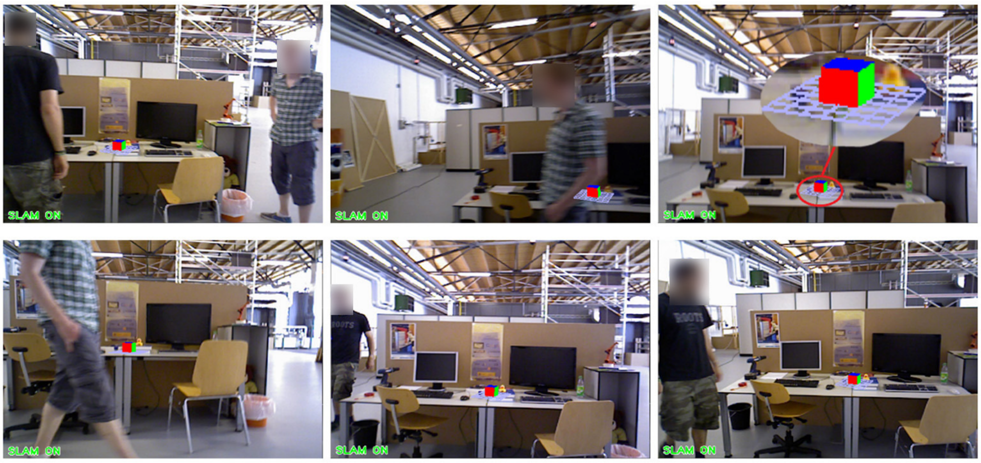


FIGURE 16
The application effect of our method in the augmented reality experiment.

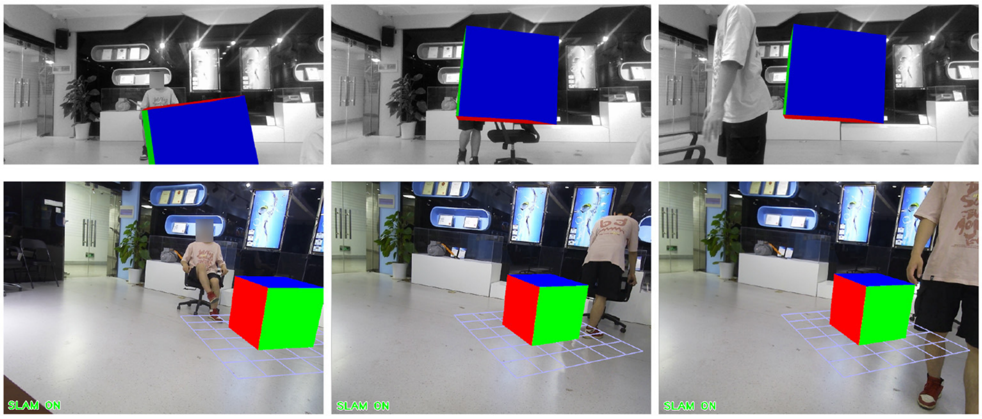


FIGURE 17
Application of VINS and our method in augmented reality experiment.

and VINS-Mono (Mur-Artal and Tardós, 2017). The data are shown in Table 5. The experimental results show that the registration real-time performance of our method is better than the traditional SURF+KLT method at different time stages. Although, the computational cost of detection causes our method to consume slightly more time than the ORB-SLAM2 method for Augmented Reality Registration, this method provides better robustness while the registration latency remains stable below 25 ms.

4.4.3. Comparison experiment with VISLAM

VISLAM is the most commonly used registration method for AR today. Although, the use of IMU provides good assistance for camera pose, it does not perform so well when tracking for long periods of time in dynamic environments. As shown in the VINS-Mono data in Table 5, after 350 frames, it shows a great drift and the registration time is also much longer. In the laboratory dynamic environment, we carried out many experiment of dragging the chair to move. After initialization, insert a virtual object, and verify by dragging the chair to move together. We select one of the experimental results for comparison, as shown in Figure 17. It can be seen from the results that the virtual objects registered by the VINS-Mono method are not very robust in dynamic environments. However, the virtual objects registered by our method remain stable in long-term dynamic environments.

5. Conclusion

In recent years, augmented reality technology is prevalent, and it is often applied in small map scenarios. Therefore, a small number of dynamic points in the map will significantly affect the registration effect. The dynamic target detection and tracking algorithm proposed in this paper can effectively help the stable operation of the Augmented Reality Registration technology in a dynamic environment. The stable operation of YOLOv3 can effectively help eliminate the feature points of small dynamic targets. Considering that the offset of augmented reality in the map is always the calculation error of the pose point is too large, this paper uses the ATE and RPE indicators to verify the algorithm of this paper. The final result analysis shows that the algorithm proposed in this paper has an excellent performance in each target detection stage and long-term tracking. The results of the ATE and RPE indicators indicate that the algorithm proposed in this paper performs well in both small and large dynamic scenarios and can be well applied in augmented reality technology. When we integrated the object detection method into the SLAM system, we did not choose the more efficient YOLOv4 due to the problem of computing power. Therefore, we use the prior data provided by GMM to compensate for the accuracy problem, which can use less computing power while maintaining the accuracy and real-time required for Augmented Reality Registration. There are better solutions now, like YOLOv5 and the recently released YOLOv7, and we're working on it. And, we need to optimize the computational cost in the next work so that Augmented Reality Registration requires less computational power and has better real-time performance.

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

JL and DC provided research ideas and plans. JL and QG improved the algorithm. QG and DY wrote the programs and conducted the experiments. DC and QG were responsible for collecting data. QG wrote the manuscript with the help of JL and DC. DC revised the manuscript and approved the final submission. All authors contributed to the article and approved the submitted version.

Funding

This work was partially supported by the Key R&D Program of Jiangsu Province (Industry Prospects and Key Core Technologies) under Grant BE2020006-2, the National Natural Science Foundation of China under Grants 61773219 and 62003169, the Natural Science Foundation of Jiangsu Province under Grant BK20200823, the Jiangsu Innovation and Entrepreneurship Talent Program Project under Grant JSSCBS202030576, and the Natural Science Research Project of Jiangsu Higher Education Institutions under Grant 20KJB520029.

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Ammirato, P., and Berg, A. C. (2019). A mask-rcnn baseline for probabilistic object detection. *arXiv preprint arXiv:1908.03621, abs/1908.03621*. doi: 10.48550/arXiv.1908.03621

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). SEGNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 2481–2495. doi: 10.1109/TPAMI.2016.2644615

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). "SURF: speeded up robust features," in *Proceedings of European Conference on Computer Vision* (Graz: Springer), 404–417. doi: 10.1007/11744023_32

Bescos, B., Fácil, J. M., Civera, J., and Neira, J. (2018). Dynaslam: tracking, mapping, and inpainting in dynamic scenes. *IEEE Robot. Autom. Lett.* 3, 4076–4083. doi: 10.1109/LRA.2018.2860039

Calloway, T., and Megherbi, D. B. (2020). "Three tiered visual-inertial tracking and mapping for augmented reality in urban settings," in *2020 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)* (Tunis). doi: 10.1109/CIVEMSA48639.2020.9132969

Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M., and Tardós, J. D. (2021). ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE Trans. Robot.* 37, 1874–1890. doi: 10.1109/TRO.2021.3075644

Chakravarty, S., Banerjee, M., and Hung, C.-C. (2017). "Kalman particle filtering algorithm and its comparison to Kalman based linear unmixing," in *Proceedings of IEEE International Geoscience Remote Sensing Symposium* (Fort Worth, TX), 221–224. doi: 10.1109/IGARSS.2017.8126934

Cheng, J., Wang, C., and Meng, M. Q.-H. (2019). Robust visual localization in dynamic environments based on sparse motion removal. *IEEE Trans. Autom. Sci. Eng.* 17, 658–669. doi: 10.1109/TASE.2019.2940543

Cui, L., and Ma, C. (2019). SOF-SLAM: a semantic visual slam for dynamic environments. *IEEE Access* 7, 166528–166539. doi: 10.1109/ACCESS.2019.2952161

Cui, L., and Ma, C. (2020). SDF-SLAM: semantic depth filter slam for dynamic environments. *IEEE Access* 8, 95301–95311. doi: 10.1109/ACCESS.2020.2994348

Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O. (2007). MONOSLAM: real-time single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 1052–1067. doi: 10.1109/TPAMI.2007.1049

Engel, J., Schöps, T., and Cremers, D. (2014). "LSD-SLAM: large-scale direct monocular slam," in *Proceedings of European Conference on Computer Vision* (Zurich: Springer), 834–849. doi: 10.1007/978-3-319-10605-2_54

Engel, J., Stückler, J., and Cremers, D. (2015). "Large-scale direct slam with stereo cameras," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Hamburg), 1935–1942. doi: 10.1109/IROS.2015.7353631

Fan, Y., Zhang, Q., Liu, S., Tang, Y., Jing, X., Yao, J., and Han, H. (2020). Semantic slam with more accurate point cloud map in dynamic environments. *IEEE Access* 8, 112237–112252. doi: 10.1109/ACCESS.2020.3003160

Forster, C., Pizzoli, M., and Scaramuzza, D. (2014). "SVO: fast semi-direct monocular visual odometry," in *Proceedings of International Conference on Robotics and Automation* (Hong Kong), 15–22. doi: 10.1109/ICRA.2014.6906584

Gao, Q., Liu, J., and Ju, Z. (2020). Robust real-time hand detection and localization for space human-robot interaction based on deep learning. *Neurocomputing* 390, 198–206. doi: 10.1016/j.neucom.2019.02.066

Gao, Q., Liu, J., Ju, Z., and Zhang, X. (2019). Dual-hand detection for human-robot interaction by a parallel network based on hand detection and body pose estimation. *IEEE Trans. Indus. Electron.* 66, 9663–9672. doi: 10.1109/TIE.2019.2898624

Gehrmann, S., Strobelt, H., Krüger, R., Pfister, H., and Rush, A. M. (2019). Visual interaction with deep learning models through collaborative semantic inference. *IEEE Trans. Vis. Comput. Graph.* 26, 884–894. doi: 10.1109/TVCG.2019.2934595

Goan, E., and Fookes, C. (2020). "Bayesian neural networks: an introduction and survey," in *Case Studies in Applied Bayesian Data Science*, eds K. L. Mengersen, P. Pudlo, and C. P. Robert (Cham: Springer), 45–87. doi: 10.1007/978-3-030-42553-1_3

Goerke, N., and Braun, S. (2009). "Building semantic annotated maps by mobile robots," in *Proceedings of Towards Autonomous Robotic System* (Londonderry), 149–156.

Huang, L., Chen, C., Yun, J., Sun, Y., Tian, J., Hao, Z., et al. (2022). Multi-scale feature fusion convolutional neural network for indoor small target detection. *Front. Neurorobot.* 16:1021. doi: 10.3389/fnbot.2022.881021

Klein, G., and Murray, D. (2007). "Parallel tracking and mapping for small AR workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality* (Nara), 225–234. doi: 10.1109/ISMAR.2007.4538852

Li, H., and Shi, L. (2019). Robust event-based object tracking combining correlation filter and CNN representation. *Front. Neurorobot.* 13:82. doi: 10.3389/fnbot.2019.00082

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). "SSD: single shot multibox detector," in *Proceedings of*

European Conference on Computer Vision (Amsterdam: Springer), 21–37. doi: 10.1007/978-3-319-46448-0_2

Liu, Y., Xu, M., Jiang, G., Tong, X., Yun, J., Liu, Y., et al. (2022). Target localization in local dense mapping using RGBD slam and object detection. *Concurr. Comput*. 34:e6655. doi: 10.1002/cpe.6655

Liu, Z. (2021). "Implementation of slam and path planning for mobile robots under ROS framework," in *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)* (Xi'an) doi: 10.1109/ICSP51882.2021.9408882

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis*. 60, 91–110. doi: 10.1023/B:VISI.0000029664.99615.94

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE Trans. Robot*. 31, 1147–1163. doi: 10.1109/TRO.2015.2463671

Mur-Artal, R. and Tardós, J. D. (2017). ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robot*. 33, 1255–1262. doi: 10.1109/TRO.2017.2705103

Nguyen, D.-D., Elouardi, A., Florez, S. A. R., and Bouaziz, S. (2018). HOOFR SLAM system: an embedded vision slam algorithm and its hardware-software mapping-based intelligent vehicles applications. *IEEE Trans. Intell. Transp. Syst*. 20, 4103–4118. doi: 10.1109/TITS.2018.2881556

Raguram, R., Chum, O., Pollefeys, M., Matas, J., and Frahm, J.-M. (2012). USAC: a universal framework for random sample consensus. *IEEE Trans. Pattern Anal. Mach. Intell*. 35, 2022–2038. doi: 10.1109/TPAMI.2012.257

Redmon, J., and Farhadi, A. (2018). YOLOv3: an incremental improvement. *arXiv preprint arXiv:1804.02767*. doi: 10.48550/arXiv.1804.02767

Riazuelo, L., Montano, L., and Montiel, J. (2017). "Semantic visual slam in populated environments," in *Proceedings of European Conference on Mobile Robots* (Paris), 1–7. doi: 10.1109/ECMR.2017.8098697

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). "ORB: an efficient alternative to sift or surf," in *Proceedings of the IEEE International Conference on Computer Vision* (Barcelona), 2564–2571. doi: 10.1109/ICCV.2011.6126544

Stauffer, C., and Grimson, W. E. L. (1999). "Adaptive background mixture models for real-time tracking," in *Proceedings of 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149) Vol. 2* (Fort Collins, CO), 246–252. doi: 10.1109/CVPR.1999.784637

Sun, Y., Liu, M., and Meng, M. Q.-H. (2017). Improving RGB-D slam in dynamic environments: a motion removal approach. *Robot. Auton. Syst*. 89, 110–122. doi: 10.1016/j.robot.2016.11.012

Sun, Y., Liu, M., and Meng, M. Q.-H. (2018). Motion removal for reliable rgb-d slam in dynamic environments. *Robot. Auton. Syst*. 108, 115–128. doi: 10.1016/j.robot.2018.07.002

Sun, Y., Liu, M., and Meng, M. Q.-H. (2019). Active perception for foreground segmentation: an RGB-D data-based background modeling method. *IEEE Trans. Autom. Sci. Eng*. 16, 1596–1609. doi: 10.1109/TASE.2019.2893414

Sun, Y., Zhao, Z., Jiang, D., Tong, X., Tao, B., Jiang, G., et al. (2022). Low-illumination image enhancement algorithm based on improved multi-scale retinex and ABC algorithm optimization. *Front. Bioeng. Biotechnol*. 10:865820. doi: 10.3389/fbioe.2022.865820

Wang, C.-C., Thorpe, C., Thrun, S., Hebert, M., and Durrant-Whyte, H. (2007). Simultaneous localization, mapping and moving object tracking. *Int. J. Robot. Res*. 26, 889–916. doi: 10.1177/0278364907081229

Wang, Z., Jensfelt, P., and Folkesson, J. (2016). "Building a human behavior map from local observations," in *Proceedings of the 25th IEEE International Symposium on Robot and Human Interactive Communication* (New York, NY), 64–70. doi: 10.1109/ROMAN.2016.7745092

Xiao, L., Wang, J., Qiu, X., Rong, Z., and Zou, X. (2019). Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robot. Auton. Syst*. 117, 1–16. doi: 10.1016/j.robot.2019.03.012

Xu, X., Cheong, L. F., and Li, Z. (2018). "Motion segmentation by exploiting complementary geometric models," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 2859–2867. IEEE. doi: 10.1109/CVPR.2018.00302

Xu, Y., Xu, K., Wan, J., Xiong, Z., and Li, Y. (2018). "Research on particle filter tracking method based on Kalman filter," in *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)* (Xi'an), 1564–1568. doi: 10.1109/IMCEC.2018.8469578

Yu, C., Liu, Z., Liu, X.-J., Xie, F., Yang, Y., Wei, Q., et al. (2018). "DS-SLAM: a semantic visual slam towards dynamic environments," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots System* (Madrid), 1168–1174. doi: 10.1109/IROS.2018.8593691

Zhong, F., Wang, S., Zhang, Z., and Wang, Y. (2018). "Detect-SLAM: making object detection and slam mutually beneficial," in *Proceedings of IEEE Winter Conference on Applications of Computer Vision* (Lake Tahoe, NV), 1001–1010. doi: 10.1109/WACV.2018.00115

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 13 October 2022
DOI 10.3389/fnbot.2022.934260

OPEN ACCESS

EDITED BY
Yan W. U.,
Institute for Infocomm Research (A*STAR), Singapore

REVIEWED BY
Qianli Xu,
Institute for Infocomm Research (A*STAR), Singapore
Michel Paindavoine,
Université de Bourgogne, France

*CORRESPONDENCE
Xiangyong Liu
1510280@tongji.edu.cn
Zhi-Xin Yang
zxyang@um.edu.mo

RECEIVED 02 May 2022
ACCEPTED 29 August 2022
PUBLISHED 13 October 2022

CITATION
Liu X, Yang Z-X, Xu Z and Yan X (2022) NeuroVI-based new datasets and space attention network for the recognition and falling detection of delivery packages.
Front. Neurorobot. 16:934260.
doi: 10.3389/fnbot.2022.934260

NeuroVI-based new datasets and space attention network for the recognition and falling detection of delivery packages

Xiangyong Liu[1,2,3]*, Zhi-Xin Yang[1,2]*, Zhiqiang Xu[4] and Xiaoan Yan[1,2]

[1]The State Key Laboratory of the Internet of Things for Smart City (IOTSC), University of Macau, Macau, Macao SAR, China, [2]Department of Electromechanical Engineering, University of Macau, Macau, Macao SAR, China, [3]College of Automotive Engineering, Tongji University, Shanghai, China, [4]School of Mechanical Engineering, Tongji University, Shanghai, China

With the popularity of online-shopping, more and more delivery packages have led to stacking at sorting centers. Robotic detection can improve sorting efficiency. Standard datasets in computer vision are crucial for visual detection. A neuromorphic vision (NeuroVI) camera is a bio-inspired camera that can capture dynamic changes of pixels in the environment and filter out redundant background information with low latency. NeuroVI records pixel changes in the environment with the output of event-points, which are very suitable for the detection of delivery packages. However, there is currently no logistics dataset with the sensor, which limits its application prospects. This paper encodes the events stream of delivery packages, and converts the event-points into frame image datasets for recognition. Considering the falling risk during the packages' transportation on the sorting belt, another falling dataset is made for the first time. Finally, we combine different encoding images to enhance the feature-extraction on the YOLO network. The comparative results show that the new datasets and image-confusing network can improve the detection accuracy with the new NeuroVI.

KEYWORDS

neuromorphic vision, delivery packages, recognition and falling datasets, space attention network, detection

Introduction

As the internet grows in popularity, more and more people would like to shop online. The increased amount of packages amount has led to the stacking of packages. Vision-based robotic detection and grasp will become the trend at the packages' sorting centers. Large datasets are critical for the development of computer vision algorithms. At present, many sensors, such as RGB cameras, radars, and depth cameras, have been adopted to annotate object datasets (Ouaknine et al., 2020). The COCO, VOC, and KITTI datasets are the most representative image datasets in the field of computer vision (Cheng et al., 2020). The Cornell dataset is a representative dataset in the field of object grasping, which uses rectangular annotation boxes (Liu et al., 2022a). With the use of the Kinect camera, it has become popular to obtain RGB and depth information in the environment

(Liu et al., 2017). However, the above datasets are mainly based on RGB cameras, and are mainly used for indoor and outdoor object recognition, industrial parts, etc. (Zhao et al., 2020). The traditional sensors can simultaneously capture the background and objects' information, which increases the computational complexity of the detection network. At the same time, when the delivery packages move quickly, the RGB images will appear blurred, which will also increase the detection difficulty. Therefore, previous RGB-D based detection is only suitable for slowly moving objects. As the packages increase in amount, there is an urgent need for faster testing.

Different from traditional cameras, NeuroVI can capture pixels' changes in the image and generate event points at a certain pixel point. The outputs of the NeuroVI are a series of digital "events" and "spikes" (Sun et al., 2021; Gallego et al., 2022). In a static environment, only the moving packages can lead to the pixel changes, especially breeding on the packages' edges (Liu et al., 2022b). When the NeuroVI camera is fixed, it can only capture the contours of delivery packages. So, the NeuroVI has great advantages in capturing the moving packages particularly. Therefore, the NeuroVI camera with the ability of capturing color changes will be very suitable for the movement detection of delivery packages, which will promote the sorting speed of packages and the development of the logistics industry in the future. However, there are currently no delivery packages' detection datasets associated with the NeuroVI cameras.

Accurate sorting of delivery packages includes the recognition and grasping operations (Xu et al., 2017). In addition, packages may fall due to the rapid movement on the sorting belt. In fact, the recognition and falling occur pre-detection before the following robotic grasping operation. Our research provides the only package detection dataset with the NeuroVI camera (Mueggler et al., 2017a). Some other object detection can also be achieved with NeuroVI, such as moving cars, bicycles, pedestrians, and flying objects (Liu et al., 2022a). They do not appear in the packages' sorting scenario, so it is not necessary for the package dataset to contain other unrelated objects. Our work is the first to apply a NeuroVI camera to the field of logistics in a sorting center. In summary, our contributions include the following three aspects:

- We provide the dynamic recognition and falling datasets of delivery packages for the first time.
- Three encoding methods are provided to achieve different feature-extraction for network detection. And the TAE instant encoding method can provide a space attention branch layer to improve the position-detection accuracy.
- The comparative detection experiments demonstrate that our dataset and attention-based network can improve the detection accuracy.

Related Work

The NeuroVI camera is an event-spired vision sensor. The events stream produced by this sensor is recorded in the form of a tuple unit $[t, x, y, p]$, where t denotes the time of the event, (x, y) denote the pixel coordinates of the event, and p denotes the polarity of the event (Gallego et al., 2022). Based on the principles of NeuroVI camera, the pixels along the contours of the objects usually change harshly, and the features along the edges can be more prominent and further enhanced. The depth camera has similar properties to a certain extent, which can show sudden depth changes along objects' contours (Mueggler et al., 2017b). Ni et al. (2011) utilized a depth camera to identify objects. But depth cameras are susceptible to the depth changes along slope surfaces. Besides, there may be no depth feedback when encountering weak-reflection material.

As a new sensor from the last decade, one of the main challenges faced by the NeuroVI camera is the lack of datasets, which limits the further maturity of event cameras. Previously, several datasets by NeuroVI camera were provided. Orchard *et al* recorded a paragraph of pedestrian behavior with a fixed NeuroVI camera, and the recordings could be played automatically without image extraction (Serrano-Gotarredona and Linares-Barranco, 2015). Krishnan and Koushik (2022) provided a pedestrian-falling detection dataset, which was mainly used for human safety warnings. Barranco et al. (2016) recorded an image dataset of QR codes with a NeuroVI camera for automatic navigation. A dataset of highway vehicles was recorded, and the segmentation of the event points was achieved by a clustering method (Chen et al., 2018). Li (2020) proposed the first NeuroVI dataset for grasping dataset, but it is still a statically grasping dataset with the external requirement of light-compensation, not a dynamic grasping dataset for delivery packages. In addition, there are no recognition and falling datasets in the previous research. Although several datasets are currently available, there is still a lack of datasets for sorting scenarios, which limits its application prospects in the logistics sorting field.

Compared with the traditional RGB-represented images, NeuroVI cameras have the following advantages: simple pixel generation, low latency, and high resolution. But the difficulty faced by NeuroVI cameras is that they cannot directly generate images like traditional cameras (Gallego et al., 2022). At present, FRE and LIF spike methods are the most effective clustering methods for NeuroVI images' extraction (Cheng et al., 2020; Zhang et al., 2021). The image of the FRE algorithm is the accumulation of all event-points in a fixed time-interval, and the image of the LIF algorithm is the accumulation of the spike potential energy in the time-interval. For another time-interval changing method, an events segmentation was introduced to cluster the event-points with a fixed number of points (Song et al., 2020). For the objects with different moving speeds, the encoding time-interval is adjusted instantly to display different types of moving objects (Li and Shi, 2019). To filter the noise events, an OTSU method is introduced to calculate the event-points' thickness threshold, which can distinguish the event points and noise points (Liu et al., 2020). However, all the encoding methods can lead to the profile-extension fluctuation of moving objects, especially viewing the objects within a closer distance. The extended or bolded profiles will reduce the predicted position accuracy.

As for the visual detection, the traditional methods mainly include SIFT method (Yi et al., 2015), optical flow method, and frame difference method (Li et al., 2020). However, traditional methods have poor feature-extraction capabilities, so some researchers use techniques such as deep learning and convolutional neural networks to extract features (Mahler et al., 2018). Currently, detection algorithms based on convolutional neural networks are usually two-stage classes, including R-CNN and FastR-CNN (Ren et al., 2017; Zhang et al., 2019). Although the accuracy of two-stage detection is high, the process is complex and slow. In order to reduce the complexity of the two-stage algorithm, some scholars have proposed single-stage detection, such as SSD and YOLO (Lu et al., 2015; Zhou et al., 2020). In addition, some scholars achieve better detection results by adding channel or spatial attentions on the detection network (Hori et al., 2017). However, the above research is all based on single-frame image detection, which is limited by the delay of image-sampling intervals, and the real-time position accuracy is not satisfactory.

Materials and methods

In this section, we first introduce the system construction for the two types of datasets. Then, the event stream's encoding methods are further elaborated. Finally, the YOLO-attention detection network is designed to combine different encoding methods.

The system's construction and delivery packages' types

The datasets were recorded by a fixed bracket and a DVS346 camera. All the datasets were recorded at a logistics sorting center. The DVS346 camera has a resolution of 346*260 pixels. Each event point is recorded as a tuple of $[t, x, y, p]$ (Figure 1A). The unit of t is *us*. The x parameter distributes in the range of [0, 345]. The y parameter distributes in the range of [0, 259]. And the polarity p is a binary variable that takes the values 0 or 1. All the information is recorded by the JAER software.

We conduct the detection experiment with the package's different viewing scales. The detection results show that our method can reach the minimum resolution of 14*13 pixels on the NeuroVI image. In fact, the delivery packages are usually distributed and sorted within several meters from the robot. And the package's viewing scale is far beyond the 85*65 pixels. So, the 346*260 pixel resolution is enough to achieve detection.

The DVS camera is fixed and owns a suitable viewing angle to record the moving scope of the delivery packages. The viewing zones of the camera are distributed at different distances, and the sorting belts carrying the packages are set with different moving speeds. As shown in Figure 2A, the delivery packages come in three different shapes, namely cube type with a hard surface, round type with a hard surface, and flat type with a soft surface. The length-width-height ratio can be the definitive criteria to categorize the package types for dataset annotation with subjective judgment. Then, the packages can be annotated in the datasets, which can be learned and predicted with a state-of-the-art network. The definition can be found in Table 1. Different packages (cube, round, flat) use different grasping claws (Figure 3A), and the recognition dataset can be used to select and replace the claws of the robots in advance. Falling dataset can be used to warn of drop hazards.

Encoding methods

As the scattered event-points cannot be trained, the traditional computer vision methods cannot be directly applied to event-points from the NeuroVI camera. To deal with this problem, this paper introduces three methods, namely Frequency, TAE, and LIF. They can achieve the event-points' accumulation for the delivery packages, and the encoding effects are shown in Figure 2B. The encoding processes are drawn in Figure 2C.

Frequency

Considering that more event points occur along the edges of the object, we use the event frequency as the pixel value to enhance the contour display of the packages (Chen et al., 2019). The main challenge is the noise event-points with small event point's number. Compared with a large number of event-points on the packages' contours, the frequency method can restrict the pixels' gray values on the noise-events pixels. Therefore, the encoding method can weaken the pixels appearance caused by the discrete noise from the environment. We accumulate all event points on each pixel, and the corresponding pixel value of the event points is calculated in formula (1). The exponent value (e^{-n}) is greater than 0, which ensures that the calculated value of the pixel is distributed in [0, 255]. In addition, with the increase of event-points, the pixel value increases, which is consistent with the principle of NeuroVI camera.

$$\delta(n) = 255 * 2 * (\frac{1}{1 + e^{-n}} - 0.5) \tag{1}$$

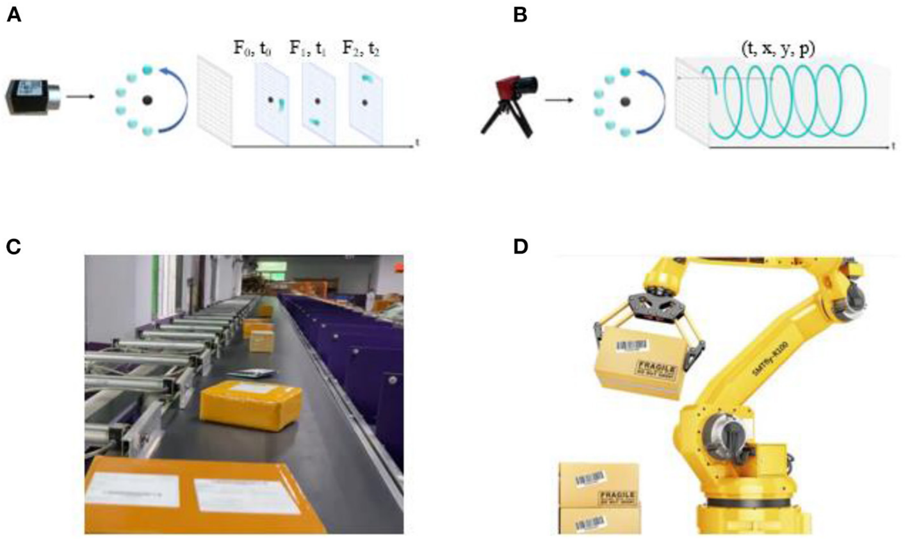


FIGURE 1
(A) The frame-based camera captured the RGB pixels with a fixed frequency. **(B)** The NeuroVI camera captured the pixel events with low latency (Chen et al., 2020). **(C)** The delivery packages are needed to be grasped and sorted in the logistics center. **(D)** A robotic arm grasps the packages automatically.

Among them, n denotes the number of positive/negative events generated on the pixel (x, y), $\delta(n)$ denotes the pixel value of the event points on the NeuroVI image, and its value is distributed in [0, 255].

Time of active events (TAE)

In order to take advantage of the event timestamp recorded by the NeuroVI camera, the TAE method is designed to enhance the contours of the delivery packages. Specifically, regardless of the polarity of the event points, the pixel value of each event point will be calculated according to the maximum occurrence time t_{p-max} within the fixed time interval.

$$[TAE:\ t \Rightarrow t_{p-\max}(x,y)] \tag{2}$$

In order to obtain the frame image of the event points, the pixel value is optimized by calculating the time-interval between the last time and the initial time in one accumulated period, and the relevant calculation is shown in equation (3). The pixel value calculated by the TAE method can capture the most recent time features of the delivery packages. And the TAE method is able to avoid bolding the profiles of the packages.

$$g(x,y) = 255^{*}\frac{t_{p\text{-}\max} - t_0}{T} \tag{3}$$

Leaky integrate-and-fire (LIF)

According to the Leaky Integrate-and-Fire model, each pixel can be viewed as a neuron associated with the potential energy and the number of spikes. The potential energy is influenced by both the number of event points and the elapsed time. When an event occurs, the potential energy increases. When there is no event point, the potential energy decreases gradually. Specially, when the potential energy exceeds the threshold, a spike is generated, and the associated potential energy is set to zero. In a fixed time-interval, we count the number of spikes, which is encoded as the pixel value of the frame image (Lansky et al., 2016).

As shown in Figure 2B, the frequency or LIF encoding methods will extend or bold the packages' profiles. The TAE algorithm can strengthen the appearance of the nearest event-points and weaken the event-points in the time-interval's initial moment. Although the bolded profiles are beneficial to reduce the class and object loss, they are harmful when improving the box-position loss.

Space attention-based network model

Object detection is an important task in computer vision, and is defined as finding target objects in an image. Object detection not only requires identification of these objects, but also requires marking the locations of these objects. There are five information parameters on each object, and they are the object's center position (x, y), width-height (h and w), and category.

The YOLO detection network has the advantages of a fast detection speed, simple pipeline, and strong versatility.

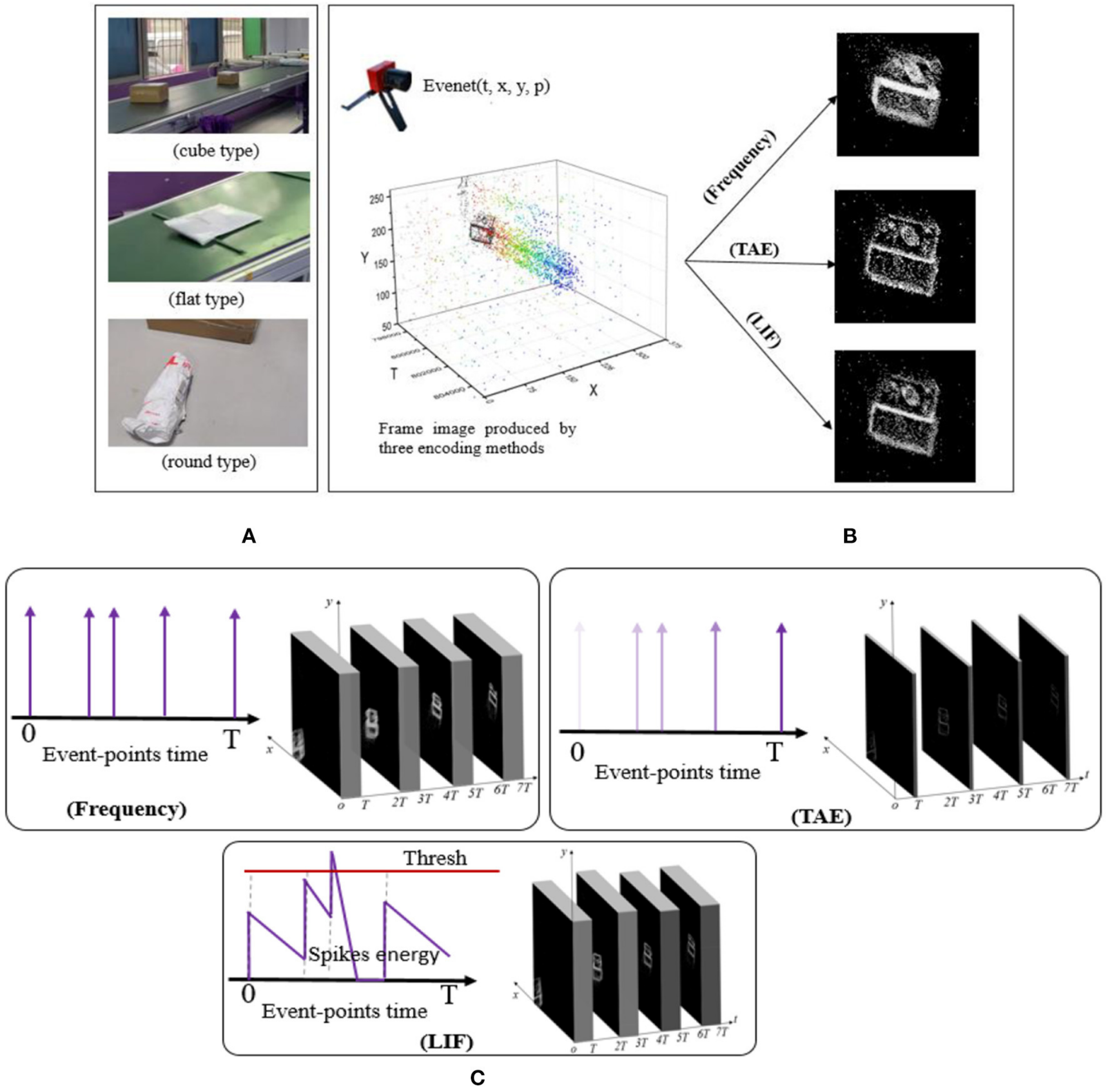


FIGURE 2
(A) The different delivery packages, namely cube, round, and flat types. **(B)** The frame images encoded by the Frequency, TAE, and LIF methods. **(C)** The link and outputs among the three encoding methods in a time-interval. The TAE algorithm can strengthen the display of the nearest event-points and weaken the event-points in the time-interval's initial moment, which can promote the packages' actual appearance instantly.

TABLE 1 The definitive criteria of different delivery packages.

Criteria	Cube package	Round package	Flat package
Length-width ratio	$R_{l-w} \leq 3$	$8 < R_{l-w}$	$3 < R_{l-w} \leq 8$
Length-hight ratio	$R_{l-h} \leq 5$	$5 < Rl\text{-}h \leq 10$	$10 < R_{l-h}$
Width-hight ratio	$R_{w-h} \leq 4$	$4 < R_{w-h} \leq 8$	$8 < R_{w-h}$

Compared with other detection networks (FastR-CNN network SDD, etc.), it can be adapted to the detection requirements of different object's sizes and categories. But YOLO also has the disadvantage of lower position accuracy of objects. Therefore, we combine the different encoding methods of NeuroVI to design a spatial attention network to improve the detection accuracy.

YOLO is a single-stage object detector that consists of backbone and head networks. Backbone network adopts the Resnet structure to realize down-sampling and features-extraction. The head network combines the features of backbone to achieve up-sampling. The detection head contains nine anchors, and the Non-maxima suppression is adopted to predict the best prediction box.

The up-sampling extraction process can be facilitated by a spatial attention model that can focus on feature attention

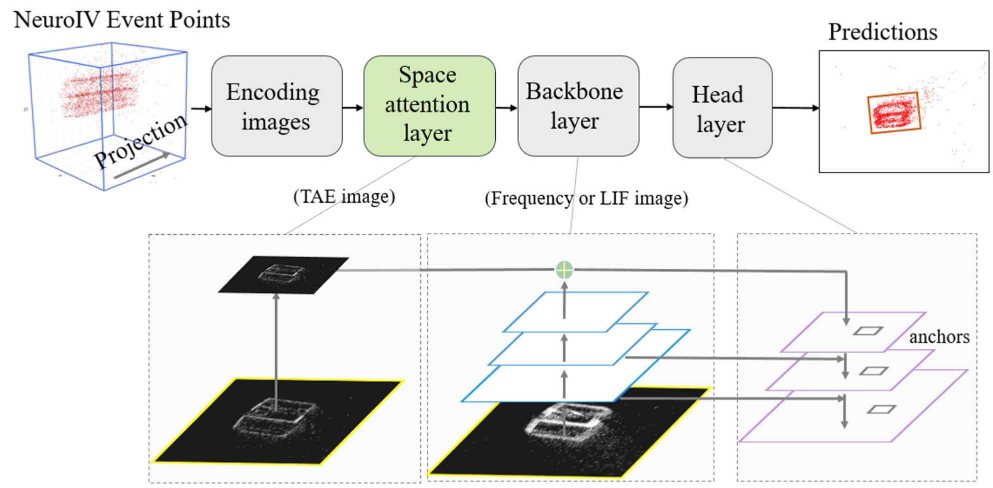


FIGURE 3
NeuroVI-based YOLO detection model is fused with space attention. The TAE images are used to construct the space attention layer, which avoids the time delay and improves the box-position accuracy. The backbone layer extracts features from the frequency or LIF image.

for location information. TAE images have the advantage of recent event-points appearance, so the feature information can be optimized by using the spatial attention mechanism with the accessibility of TAE images. Different from the backbone layer, the attention layer adopt a scaled mechanism on the TAE images. Spatial attention is able to generate regions with different weight distributions. The more obvious part on the TAE image will be exerted with higher calculation weight. Since this method only adds additional cross-layer connections on the basis of the original network, it adds hardly any extra time and computation in practical applications.

Through such connections, the features of different resolutions and different semantic strengths are fused. The feature maps with different resolutions are fused for object's detection. This ensures that each layer has the appropriate resolution and strong semantic features.

Results

We built two NeuroVI-based datasets, including the packages' recognition and falling datasets (Specian et al., 2018). All these datasets can be downloaded from the public website in this paper.

Different motion directions, distances, and viewing angles will lead to different counter recordings, which will influence the detection accuracy of NeuroVI images. Therefore, the recording process of the dataset should include all scenarios as much as possible, including straight driving, turning, and different viewing distances. The ring sorting belt can cover all of the above scenarios. At the same time, the recordings of the dataset should be kept for a period of time, to avoid missing any viewing details. A summary of the three datasets is shown in Table 2.

TABLE 2 The experimental settings of two datasets for delivery packages.

	Recognition dataset	Falling dataset
Number of the packages	15	15
Shapes of package	Cube, round and flat	Cube, round and flat
Number of videos	9	6
Average video length	30 s	20 s
Scenarios	Ring sorting belt	Ring sorting belt
Sensor	DAVIS346 Color	DAVIS346 Color
Resolution	346 * 320	346 * 320
Movement	Going straight, turning left, turning right	Falling down, falling down with inclination, throwing up, throwing up with inclination
Number of annotated frame images	3,920	3,400

Delivery packages' datasets for recognition and falling

The recognition dataset of delivery packages are mainly recorded from linear movement, turning-left movement, and turning-right movements. Each recording lasts for 30s,

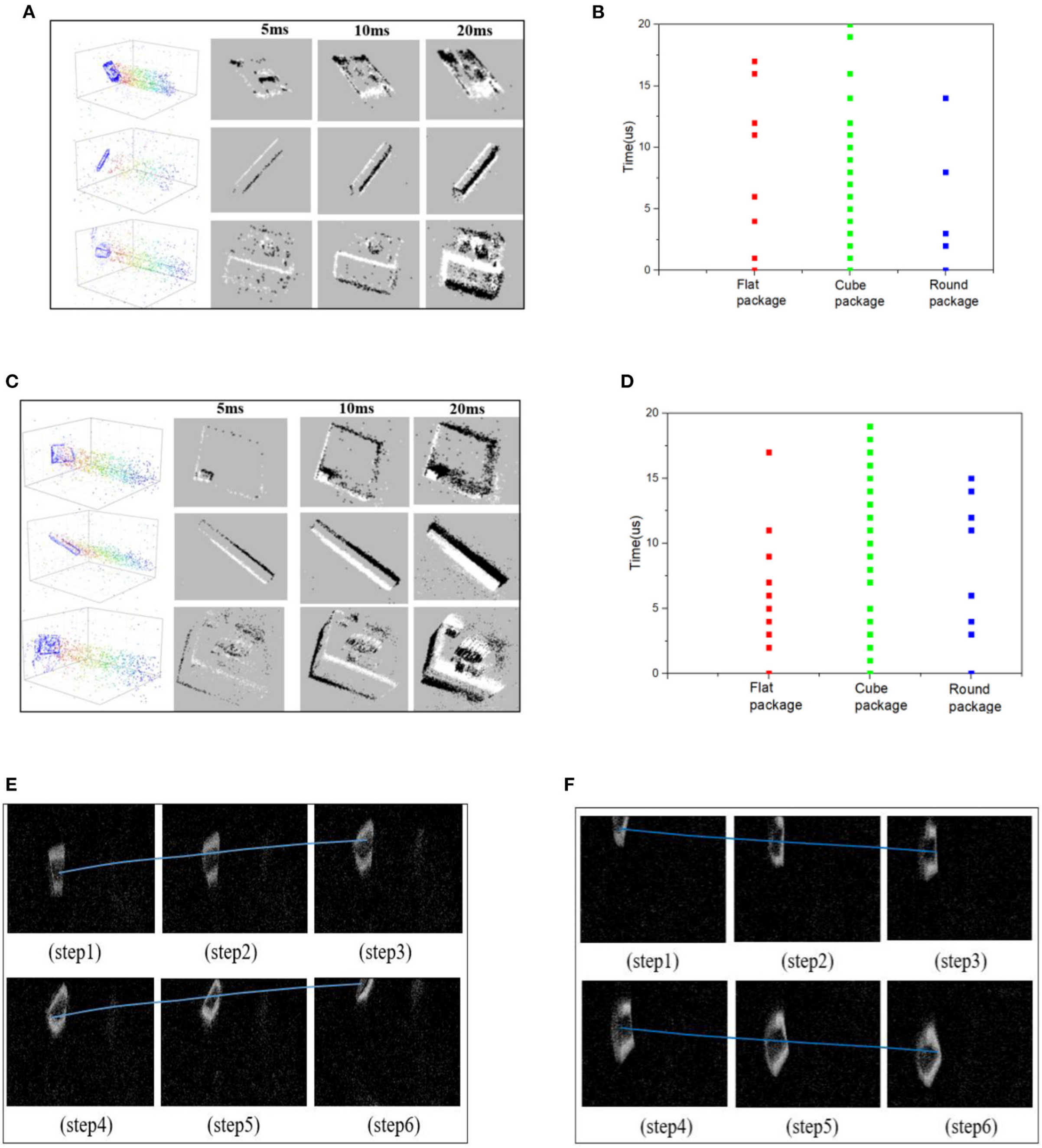


FIGURE 4
(A) The event points and encoded frames of recognition dataset. **(B)** The event thickness of recognition dataset. **(C)** The event points and encoded frames of falling dataset. **(D)** The event thickness of falling dataset. **(E)** The up-thrown experiment process with a falling angle of (0°∼90°). **(F)** The down-thrown experiment process with a falling angle of (−90°∼0°).

including different viewing distances and moving speeds. And these scenarios can be found in the packages' sorting center. Figure 4A shows the recognition datasets for three types of delivery packages. By setting the time-interval as 20ms, the ideal frame image can be extracted through the Frequency encoding method. This is equivalent to ordinary RGB images captured with a frequency of 50 fps. All the frame images were annotated by LabelImg software.

The falling dataset includes 15 packages with different sizes. The falling phenomenon is defined as dropping down off the belt, not as the rolling movement. Each package includes up-movement, down-movement, and incline-movement, which are

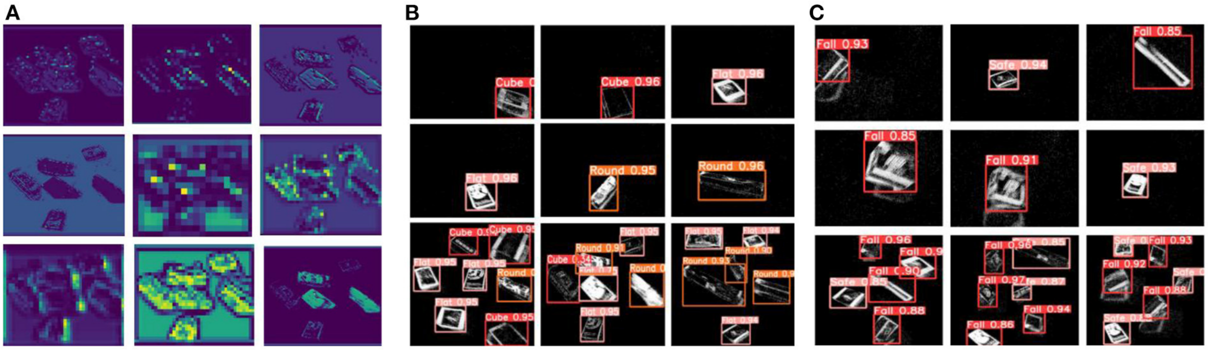


FIGURE 5
(A) The process visualization of extraction features. (B) Different packages' prediction results with the recognition dataset. (C) Different packages' falling detection with the falling dataset.

TABLE 3 The encoding time-intervals' comparisons for different delivery packages.

	Time period	Box loss	Class loss	Object loss
Cube type	10 ms	1e-2	0	4e-3
	20 ms	5e-3	0	2e-3
	30 ms	4e-2	0	3e-3
Flat type	10 ms	1e-2	0	4e-3
	20 ms	5e-3	0	2e-3
	30 ms	5e-2	0	3e-3
Round type	10 ms	1e-2	0	5e-3
	20 ms	4e-3	0	2e-3
	30 ms	3e-2	0	4e-3
Three mixed types	10 ms	1e-2	1e-3	4e-3
	20 ms	6e-3	0	2e-3
	30 ms	3e-2	2e-3	3e-3

used to simulate the possible falling phenomenon caused by high speed or collision with surroundings. Among them, the packages' images with a falling angle of (-90°∼0°) are marked as falling samples in the down-thrown experiment. And the packages' images with a falling angle of (0°∼90°) are marked as other falling samples in the up-thrown experiment. The process of each falling experiment lasted for 20 s. The down-thrown and up-thrown experiments can be utilized to make the falling datasets with subjective annotation. If the package on the sorting belt has a stable transportation, the falling angle is 0°. Figure 4C shows the frame images obtained by the encoding method. Compared with the recognition dataset, the falling dataset appears with obvious inclinations on the packages' profiles, which can be learned by the intelligent network to achieve the falling dangers' warning. Figures 4E,F shows an illustration to explain the observed falling (up or down) movements, including the successive moving sequences.

The event thickness comparisons for different packages

Event thickness is the event-points number on a pixel within a fixed time-interval. If the event thickness is larger, the corresponding pixel value will be more obvious, and the packages' profiles on the NeuroVI image will be more obvious. So the higher event thickness will promote the package appearance and detection work.

The time-interval is an important factor to influence the packages' appearance on the NeuroVI image. In Figures 4A,C, the different time-intervals are set to compare the image appearance effects. The larger time-interval, the more obvious the packages' appearance. As the packages move during the interval, the profiles of the packages will be bolded or extended, which will weaken the position prediction. So, the time-interval should not be very large or small.

The moving speed is another important factor to influence the NeuroVI image appearance. In Figures 4B,D, we compare the event thickness in two datasets. In each dataset, the cube, round, and flat packages were set to the same moving speed and distance. It is easy to find that the cube package has a larger event thickness than the other packages, which is mainly contributed to by package's longer profiles. At the same time, the vertical velocity is overlaid when the package falls, which makes the packages in the falling dataset have a higher moving speed and event thickness than the other two datasets.

The comparative detection experiments with the recognition and falling dataset

Based on the recognition dataset and YOLO-attention network, we predict the recognition results of different packages

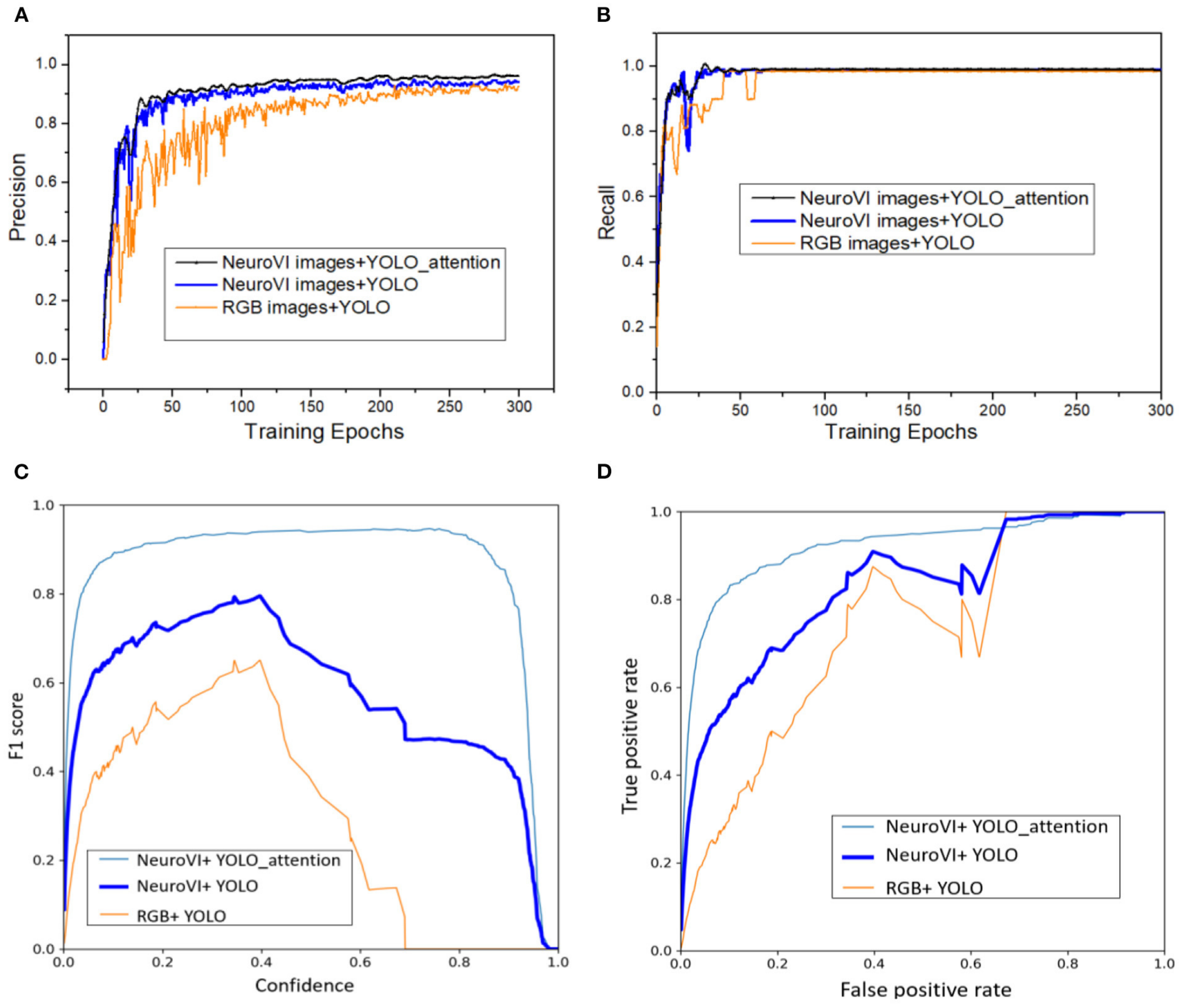


FIGURE 6
The training comparisons among different datasets and networks. **(A)** The precision curves in training process. **(B)** The recall curves in training process. **(C)** The F1 scores curves with the different confidence. **(D)** The ROC curves with the validation dataset.

(Figure 5A). The tested part in the datasets contains both the individual and mixed packages. As shown in Figure 5B, three types of packages are all accurately predicted with the marked detection accuracy. Based on our dataset, every recognition accuracy of the packages is beyond 90%.

In Figure 5C, the falling dataset is utilized to warn of drop hazards. There are two statuses: the "safe" label means that the package is moving stably on the sorting belt, and the "fall" label means that the package is dropping down. The distinguishing criteria may be that the packages appear to be in inclined or non-inclined states, which can be learned by the YOLO-attention model. All the falling predictions have accuracies beyond 85%.

Different encoding time-intervals can lead to different image appearances, which will bring different training loss. Therefore, we set different time-intervals to get the most feasible time-interval. At the same time, the Cube, Flat, and Round packages all appear with the same forms of event points, which raises concerns on its discriminatory ability with the new datasets. So, we also compare the training loss results of single and mixed packages.

The training loss includes Box loss, Class loss, and Object loss (Liang et al., 2018). The box loss is defined as the distribution deviation between the actual and predicted boxes. The class loss is defined as the labels' deviation (cube, flat, or round labels; safe or fall labels). The object loss is defined as whether there are true objects on the image. The loss comparisons are recorded in Table 3. When the interval time reaches 30ms, the box loss reaches the maximum. This is because the larger time-interval enlarges the package profiles and reduces the instant position display. When the interval time reaches 10ms, the object loss reaches the maximum. This is because the lower time-interval makes the package profiles less obvious. Experiments verify that the encoding time interval of 20ms is most reasonable. As there is only one class in the cube, flat, and round packages respectively,the class losses of them in Table 3 are 0. The results also demonstrate that the single and mixed NeuroVI datasets own the ability to distinguish different types of packages.

In order to compare the superiority of NeuroVI images and the YOLO-attention model, we also collected the package dataset with the RGB images, and then conducted the comparative detection experiments. There are 300 epochs in the training process. Precision, recall, F1 score, and ROC are the test metrics.

As shown in Figure 6, NeuroVI images have higher detection accuracy and lower fluctuation than RGB images. Although they both have high recall metrics, NeuroVI owns a faster and better convergence rate. The main reason is that NeuroVI images cannot be disturbed by complex backgrounds, and it can capture the key morphological information of moving parcels.

The YOLO-attention network with the TAE spatial attention achieves the best precision. For the recall metric, the YOLO-attention model has smaller fluctuations than the YOLO model. This is mainly because the TAE image can optimize the spatial features' weights, which is used to obtain more accurate spatial location information for convolution operation.

F1 Score is another indicator that is used to measure the accuracy of the detection model. It can be defined as a weighted calculation of model precision and recall. By setting different confidences during the training process, we calculated the F1 Score. The ROC curve is related to the true positive rate (TPR) and false positive rate (FPR). The larger the area under the ROC curve, the better the detection performance of the model. By comparing the F1 Scores and ROC curves, the detection based on NeuroVI and YOLO-attention achieved the best detection results.

Conclusion

In order to improve the efficiency of logistics sorting, we are the first to provide the datasets of delivery packages with the DAVIS346Color NeuroVI camera. According to the application requirements, the packages' datasets include recognition dataset and falling dataset. Video files are recorded in the format of (.aedat4) type. In order to facilitate neural network training, three encoding methods of event streams are utilized to extract the packages' frame images. All the codes have been opened now. In addition, three encoding methods are provided to achieve different feature extraction for network detection with a space attention layer.

The NeuroVI camera has the advantage of capturing pixel changes in the environment, which can accumulate the profiles of the objects. Therefore, the simplified frame image can simplify the design of the network, improve the learning efficiency, and output stable detection results. At the same time, the high-speed processing capability of the NeuroVI camera can improve the detection speed, especially for the delivery packages' dynamic grasping. In the next work, we will design a lightweight detection network with the above datasets, and achieve a faster grasping operation.

In summary, our recognition and grasping datasets can improve the detection speed and dynamic grasping accuracy, which can sort more delivery packages within a limited time. Although a high transmission speed may lead to dropping risk, our falling dataset can provide online feedback and alarms. We hope that the established dataset can promote the application of the NeuroVI camera in the field of logistics sorting, and improve the sorting speed of delivery packages.

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 authors.

Author contributions

The discussion and assistance are from the University of Macau and Tongji University. Z-XY and XY are greatly appreciated for their work. All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Funding

This work was funded in part by Science and Technology Development Fund, Macau SAR (Grant Nos. 0018/2019/AKP, 0008/2019/AGJ, and SKL-IOTSC-2021-2023), the State Key Laboratory of Process Automation in Mining and Metallurgy (No. BGRIMM-KZSKL-2021-02), the Chinese Postdoctoral Fund (No. 2020T130474), the Guangdong Science and Technology Department (Grant No. 2020B1515130001), and UMMTP-MYSP-2021 (AM2021003).

Acknowledgments

The authors would like to thank University of Macau and Tongji University for the discussion and assistance.

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Barranco, F., Fermuller, C., Aloimonos, Y., and Delbruck, T. A. (2016). dataset for visual navigation with neuromorphic methods. *Front. Neurosci.* 10, 1–9. doi: 10.3389/fnins.2016.00049

Chen, G., Cao, H., Aafaque, M., Chen, J., Ye, C., Röhrbein, F., et al. (2018). Neuromorphic vision based multivehicle detection and tracking for intelligent transportation system. *J. Adv. Transp.* 28, 1–13. doi: 10.1155/2018/4815383

Chen, G., Cao, H., Ye, C., Zhang, Z., Liu, X., Mo, X., et al. (2019). Multi-cue event information fusion for pedestrian detection with neuromorphic vision sensors. *Front. Neurorobot.* doi: 10.3389./fnbot.2019.00010

Chen, G., Chen, W., Yang, Q., Xu, Z., Knoll, A. (2020). A novel visible light positioning system with event-based neuromorphic vision sensor. *IEEE Sensors J.* 20, 10211–9. doi: 10.1109/JSEN.2020.2990752

Cheng, G., Yang, J., Gao, D., Guo, L., and Han, J. (2020). High-quality proposals for weakly supervised object detection. *IEEE Transact. Image Proc.* 29, 5794–804. doi: 10.1109/TIP.2020.2987161

Gallego, G., Delbruck, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., et al. (2022). "Event-based vision: a survey," in *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 44* (IEEE), 154–180. doi: 10.1109/TPAMI.2020.3008413

Hori, C., Hori, T., Lee, T. Y., Sumi, K., Hershey, J., Marks, T., et al. (2017). Attention-based multimodal fusion for video description. *IEEE Int. Conf. Comput Vis.* 4203–12. doi: 10.1109./ICCV.2017.450

Krishnan, K. S., and Koushik, K. S. (2022). Benchmarking conventional vision models on neuromorphic fall detection and action recognition dataset. *arXiv.* doi: 10.1109./CCWC54503.2022.9720737

Lansky, P., Sacerdote, L., and Zucca, C. (2016). The Gamma renewal process as an output of the diffusion leaky integrate-and-fire neuronal model. *Biol Cybern.* 110, 193–200. doi: 10.1007/s00422-016-0690-x

Li, B., Cao, H., Qu, Z., Hu, Y., Wang, K., Liang, Z., et al. (2020). Event-based robotic grasping detection with neuromorphic vision sensor and event-grasping dataset. *Front Neurorobot.* 14, 1–14. doi: 10.3389/fnbot.2020.00051

Li, C. (2020). Dangerous posture monitoring for undersea diver based on frame difference method. *J. Coastal Res.* 103, 93–104. doi: 10.2112/SI103-195.1

Li, H., and Shi, L. (2019). Robust event-based object tracking combining correlation filter and CNN representation. *Front. Neurorobot.* 13, 1–11. doi: 10.3389/fnbot.2019.00082

Liang, Y., Tang, Z., Yan, M., and Liu, J. (2018). Object detection based on deep learning for urine sediment examination. *Biocybern Biomed Eng. (2018)* 38:661–70. doi: 10.1016/j.bbe.05.004

Liu, D., Tao, X., Yuan, L., Du, Y., and Cong, M. (2022a). Robotic objects detection and grasping in clutter based on cascaded deep convolutional neural network. *IEEE Trans. Instrum Meas.* 71, 1–10. doi: 10.1109/TIM.2021.3129875

Liu, X. Y., Chen, G., Sun, X. S., and Alois, K. (2020). Ground moving vehicle detection and movement tracking based on the neuromorphic vision sensor. *IEEE Int. Things J.* 7, 9026–39. doi: 10.1109/JIOT.2020.3001167

Liu, X. Y., Yang, Z., Hou, J., and Huang, W. (2022b). Dynamic scene's laser localization by neeuroVI-based moving objects detection and LIDAR points evaluation. *IEEE Trans. Geosci. Remote Sens.* 60, 1–14. doi: 10.1109/TGRS.2022.3184962

Liu, Z., Huang, J., Han, J., Bu, S., and Lv, J. (2017). Human motion tracking by multiple RGBD cameras. *IEEE Trans. Circuits Syst. Video Technol.* 27, 2014–27. doi: 10.1109/TCSVT.2016.2564878

Lu, Y., Shu, J., Guo, J., Li, S., and Mutlu, O. (2015). High-performance lightweight transaction support in flash-based SSD. *IEEE Transacti Comp.* 64, 2819–32. doi: 10.1109/TC.2015.2389828

Mahler, J., Matl, M., Liu, X., Li, A., Gealy, D., Goldberg, K., et al. (2018). Dex-Net 3, 0. *computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. IEEE Int. Conf. Robot.* 5620–5627. doi: 10.1109/ICRA.2018.8460887

Mueggler, E., Bartolozzi, C., and Scaramuzza, D. (2017b). Fast event-based corner detection. *In: British Machine Vision Conference (BMVC).* doi: 10.5244./C.31.33

Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., and Scaramuzza, D. (2017a). The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *Int. J. Rob. Res.* 36, 142–9. doi: 10.1177/0278364917691115

Ni, B., Wang, G., and Moulin, P. (2011). RGBD-HuDaAct: A Color-Depth Video Database for Human Daily Activity Recognition. In: *International Conference on Computer Vision Workshops.* Barcelona, Spain: ICCV Workshops. p. 6–13. doi: 10.1109./ICCVW.2011.6130379

Ouaknine, A., Newson, A., and Tupin, F., Pérez l. (2020). *CARRADA* dataset: camera and automotive radar with range-angle-doppler annotations. *Comput Soc Conf Comput. Vis.*

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1137–49. doi: 10.1109/TPAMI.2016.2577031

Serrano-Gotarredona, T., and Linares-Barranco, B. (2015). Poker-DVS and MNIST-DVS their history, how they were made, other details. *Front Neurosci.* 9. doi: 10.3389./fnins.2015.00481

Song, G., Wang, K., Kang, H., Liu, T., Gao, Y., Li, T., et al. (2020). Bin loss for hard exudates segmentation in fundus images. *Neurocomputing.* 392:314–24. doi: 10.1016/j.neucom.10103

Specian, A., Mucchiani, C., Yim, M., and Seo, J. (2018). Robotic edge-rolling manipulation: a grasp planning approach. *IEEE Robot. Autom. Lett.* 3, 3137–44. doi: 10.1109/LRA.2018.2849828

Sun, R., Shi, D., Zhang, Y., Li, R., and Li, R. (2021). Data-driven technology in event-based vision. *Complexity.* doi: 10.1155./2021/6689337

Xu, J., Alt, N., Zhang, Z., and Steinbach, E. (2017). Grasping posture estimation for a two-finger parallel gripper with soft material jaws using a curved contact area friction model. *IEEE Int. Conf. Robot.* doi: 10.1109./ICRA.2017.7989258

Yi, C., Cui, L., and Luo, C. (2015). Moving target tracking algorithm based on improved optical flow technology. *Open. Autom. Control. Syst J.* 7, 1387–92. doi: 10.2174/1874444301507011387

Zhang, D., Li, J., Lin, L., and Yang, S. (2019). Cycle-consistent domain adaptive faster RCNN. *IEEE Access.* 7, 123903–11. doi: 10.1109/ACCESS.2019.2938837

Zhang, S., Xu, H., Li, Z., Liu, S., Song, B., Li, Q. A., et al. (2021). Compact model of ovonic threshold switch combining thermal dissipation effect. *Front Neurosci.* 15, 1–9. doi: 10.3389/fnins.2021.635264

Zhao, J., Jiang, X., Wang, X., Wang, S., and Liu, Y. (2020). Assembly of randomly placed parts realized by using only one robot arm with a general parallel-jaw gripper. *IEEE Int. Conf. Robot.* doi: 10.1109./ICRA40945.2020.9197396

Zhou, L., Min, W., Lin, D., Han, Q., and Liu, R. (2020). Detecting motion blurred vehicle logo in IoV using filter-DeblurGAN and VL-YOLO. *IEEE Trans. Control Syst. Technol.* 69, 3604–14. doi: 10.1109/TVT.2020.2969427

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 25 October 2022
DOI 10.3389/fnbot.2022.978014

Estimation of knee joint movement using single-channel sEMG signals with a feature-guided convolutional neural network

Song Zhang[1,2†], Jiewei Lu[1,2†], Weiguang Huo[1,2*], Ningbo Yu[1,2,3*] and Jianda Han[1,2,3]

[1]College of Artificial Intelligence, Nankai University, Tianjin, China, [2]Tianjin Key Laboratory of Intelligent Robotics, Nankai University, Tianjin, China, [3]Institute of Intelligence Technology and Robotic Systems, Shenzhen Research Institute of Nankai University, Shenzhen, China

Check for updates

OPEN ACCESS

EDITED BY
Zhan Li,
Swansea University, United Kingdom

REVIEWED BY
Hui Zhou,
Nanjing University of Science and Technology, China
Rui Li,
Xi'an University of Technology, China

*CORRESPONDENCE
Ningbo Yu
nyu@nankai.edu.cn
Weiguang Huo
weiguang.huo@nankai.edu.cn

[†]These authors have contributed equally to this work and share first authorship

RECEIVED 25 June 2022
ACCEPTED 28 September 2022
PUBLISHED 25 October 2022

CITATION
Zhang S, Lu J, Huo W, Yu N and Han J (2022) Estimation of knee joint movement using single-channel sEMG signals with a feature-guided convolutional neural network. *Front. Neurorobot.* 16:978014. doi: 10.3389/fnbot.2022.978014

Estimating human motion intention, such as intent joint torque and movement, plays a crucial role in assistive robotics for ensuring efficient and safe human-robot interaction. For coupled human-robot systems, surface electromyography (sEMG) signal has been proven as an effective means for estimating human's intended movements. Usually, joint movement estimation uses sEMG signals measured from multiple muscles and needs many sEMG sensors placed on the human body, which may cause discomfort or result in mechanical/signal interference from wearable robots/environment during long-term routine use. Although the muscle synergy principle implies that it is possible to estimate human motion using sEMG signals from even one signal muscle, few studies investigated the feasibility of continuous motion estimation based on single-channel sEMG. In this study, a feature-guided convolutional neural network (FG-CNN) has been proposed to estimate human knee joint movement using single-channel sEMG. In the proposed FG-CNN, several handcrafted features have been fused into a CNN model to guide CNN feature extraction, and both handcrafted and CNN-extracted features were applied to a regression model, i.e., random forest regression, to estimate knee joint movements. Experiments with 8 healthy subjects were carried out, and sEMG signals measured from 6 muscles, i.e., vastus lateralis, vastus medialis, biceps femoris, semitendinosus, lateral or medial gastrocnemius (LG or MG), were separately evaluated for knee joint estimation using the proposed method. The experimental results demonstrated that the proposed FG-CNN method with single-channel sEMG signals from LG or MG can effectively estimate human knee joint movements. The average correlation coefficient between the measured and the estimated knee joint movements is 0.858 ± 0.085 for LG and 0.856 ± 0.057 for MG. Meanwhile, comparative studies showed that the combined handcrafted-CNN features outperform either the handcrafted features or the CNN features; the performance of the proposed signal-channel sEMG-based FG-CNN method is comparable to those of the traditional multi-channel sEMG-based methods. The outcomes of this study enable the possibility of developing a single-channel sEMG-based human-robot interface for knee joint movement estimation, which can facilitate the routine use of assistive robots.

KEYWORDS

single-channel sEMG signals, human-robot interaction, joint movement estimation, level walking, feature-guided convolutional neural network (FG-CNN)

1. Introduction

Surface electromyography (sEMG) has been extensively used to ensure accurate and safe human-robot interaction (HRI) in robotic devices for rehabilitation or performance enhancement (Nam et al., 2014; Spanias et al., 2016; Caulcrick et al., 2021). Regarding the sEMG-based HRI, one crucial issue is to estimate human motion intention (e.g., intended joint movements) from the sEMG signals (Ding et al., 2017; Bi et al., 2019; Lu et al., 2019). Due to the sEMG signal with the characteristics of preceding the corresponding motion by 20–100 ms and containing neuromuscular control information, the sEMG-based motion estimation benefits in achieving a more natural and fluent HRI and can differentiate how much of the motion is caused by muscles: a unique advantage compared with the inertial measurement unit (IMU)/optical-based method (Xiong et al., 2021). Recently, many approaches have been proposed to estimate the human joint movements based on multi-channel sEMG signals, such as adaptive hybrid classifier for hand gesture recognition (Ding et al., 2019) and Hill-based method or deep learning method for joint movement prediction (Fleischer and Hommel, 2008; Wang et al., 2021; Zhong et al., 2022).

Although multi-channel sEMG signals can provide rich information and contribute to estimate the corresponding joint movement accurately, using multi-channel sEMG has some practical limitations: first, collecting multi-channel sEMG subjects is subject to some limitations, such as weakness or spasticity of one or more specific muscles and mechanical/signal interference between sEMG sensors and wearable robots/environment (e.g., sitting on a chair); second, increasing the number of physical channels would increase the system complexity, making it difficult to deploy, as well as increase the power consumption (He et al., 2019). The above drawbacks limit the routine use of sEMG-based assistive robots. Therefore, it is important to investigate the estimation of human motion using sEMG signals from fewer muscles or even a single muscle.

Recently, some related studies on hand gesture identification (Kumar et al., 2013), upper limb movement recognition (Shao et al., 2020), terrain identification (Gupta and Agarwal, 2019), and lower limb movement recognition (Wei et al., 2022) used single-channel sEMG signals. However, the existing studies mainly focused on recognizing discrete motion modes rather than estimating continuous joint movements. Compared with discrete modes, continuous joint movements can enable simultaneous and proportional control (Bao et al., 2021), realizing more effective and safer HRI for rehabilitation and assistive robots and orthoses. To the best of our knowledge, few studies demonstrated an accurate joint movement estimation method based on single-channel sEMG. According to the muscle synergy principle, which is widely accepted as a constitutional function unit of the central neural systems that control muscles in groups (d'Avella et al., 2003; Jiang et al., 2014; Dwivedi et al., 2020; Kubota et al., 2021), a group of related muscles' activities have certain common components or patterns, which enables the possibility of estimating joint movements using sEMG signals from one or fewer muscles. Therefore, developing a single-channel sEMG-based continuous joint movement estimation method has great potential for facilitating the routine use of assistive robots.

Compared to recognizing motion modes, it is more challenging to accurately estimate continuous joint movements using single-channel sEMG signals due to the limited muscular information. To guarantee an accurate and robust estimation of human joint movement, it is crucial to extract muscular information from single-channel sEMG signals adequately. There are two main ways of extracting muscular information: One is directly computing handcrafted features using mathematical equations (Phinyomark et al., 2012; Thongpanja et al., 2016) and another one is to extract learning features by deep learning, e.g., convolutional neural network (CNN). The learning features may complement the handcrafted features (Atzori et al., 2016; Phinyomark and Scheme, 2018; Côté-Allard et al., 2020). Therefore, it is possible to obtain relatively adequate muscular information from single-channel sEMG by fusing the handcrafted and learning features.

In this study, a new feature extraction method, namely feature-guide convolutional neural network (FG-CNN), was proposed to estimate knee joint movements using single-channel sEMG signals. In the proposed FG-CNN, 14 handcrafted features (Wei et al., 2019) were first fed into a fusion layer to guide a traditional CNN in extracting 14 implicit features (i.e., CNN extracted features). The 28 FG-CNN features containing 14 handcrafted features and 14 CNN features were applied to a regression model, e.g., random forest regression, to estimate continuous knee joint movements. To verify the effectiveness of the FG-CNN, the proposed method was respectively evaluated on six kinds of single-channel sEMG signals measured from vastus lateralis (VL), vastus medialis (VM), biceps femoris (BF), semitendinosus (ST), lateral and medial gastrocnemius (LG and MG) for estimating the movements. Meanwhile, the 28 FG-CNN features were respectively compared to 14 handcrafted features and 28 CNN features (extracted by the traditional CNN) on the same regression model. The experimental results show that the proposed FG-CNN method with single-channel sEMG signals from LG or MG can effectively estimate the movement. The FG-CNN features outperform the handcrafted features and CNN features on single-channel sEMG-based movement estimation, suggesting that the FG-CNN features contain more muscular information.

The main contributions of this study are as follows:

1) This is the first study, to our knowledge, to investigate the feasibility of using single-channel sEMG signals to estimate the human knee angles.
2) A new feature extraction algorithm has been developed to extract muscular information from single-channel sEMG adequately and a new scheme is proposed to estimate knee joint angles based on single-channel sEMG signals using FG-CNN and regression models.
3) The effectiveness of the proposed method has been evaluated *via* experiments with eight subjects during walking. sEMG signals from a single muscle, LG or MG, can be used to improve the estimation performance with the proposed method.

2. Methods

2.1. Feature-guided convolutional neural network

The proposed FG-CNN was depicted in Figure 1. In the FG-CCN, 14 typical sEMG features, i.e., the handcrafted features, were extracted from the raw sEMG data and fed into a fusion layer of a CNN to guide the CNN in extracting implicit muscular information. Furthermore, both the handcrafted features and the CNN extracted features were connected and used to estimate knee joint movements using a regression model, e.g., random forest regression (RF) and light gradient boosting machine (LGBM).

2.1.1. Handcrafted feature extraction

A method called overlapping analysis windows with a window length of 50 ms and an increment of 20 ms was used to segment the sEMG signals. The vector $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ represents sEMG signal in a window, where n is the length of $\mathbf{x}$. 14 handcrafted features, including integrated EMG (IEMG), mean absolute value (MAV), mean, root mean square (RMS), variance (VAR), Kurtosis, skewness, zero crossing (ZC), slop sign change (SSC), waveform length (WL), and four auto-regressive (AR) model coefficients are calculated using the overlapping analysis windows (Wei et al., 2019). The above 14 handcrafted features are concatenated as a vector ($\mathbf{p} = \{\mathrm{IEMG}, \mathrm{MAV}, \ldots, \mathrm{AR}_4\}$) fused into a CNN to extract CNN features.

2.1.2. FG-CNN feature extraction

The CNN feature extraction (see Figure 1) is used to extract implicit muscular information from the input sEMG signal vector $\mathbf{x}$. In the CNN structure, five convolution layers had 2, 4, 8, 16, and 32 filters, respectively, where the filters were 5×1, 4×1, 3×1, 2×1, and 1×1. Max pooling was conducted on 2×1 area with a stride of 1. The two fully-connected layers contain 192 and 14 neurons. For each convolutional layer, $\mathbf{x} \in \mathbb{R}^{L' \times D'}$ is defined as the input, where the L' and D' denote the length and the number of channels. Assuming the D convolutional kernels $\mathbf{k}$, the output of the convolutional layer $\mathbf{y} \in \mathbb{R}^{L \times D}$ is described as:

$$\mathbf{y} = f\big(\mathbf{k} * \mathbf{x} + \mathbf{w}\big), \tag{1}$$

where f represents an activation function, $\mathbf{w}$ is a bias parameter vector, and $*$ denotes convolution.

The LeakyReLU nonlinearity (Maas et al., 2013) is applied as the activation function of convolutional layers mentioned in (1), which is defined as:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ ax, & \text{otherwise,} \end{cases} \tag{2}$$

where a is a learnable parameter.

To avoid overfitting the model, batch normalization (BN) (Ioffe and Szegedy, 2015) is applied after each convolutional layer. The convolutional layer is followed by a max-pooling layer with a length of 2, which transforms the outputs of multiple neurons in one layer into a single neuron in the next layer. The input of the pooling layer is the output of the convolutional layer before it, i.e., $\mathbf{y} \in \mathbb{R}^{L \times D}$. The output of each pooling layer $\mathbf{y}' \in \mathbb{R}^{\frac{L}{2} \times D}$ is described as:

$$\mathbf{y}'(i,j) = \max(\mathbf{y}(2i-1, j), \mathbf{y}(2i, j)), \tag{3}$$

where $i = 1, 2, \ldots \frac{L}{2}, j = 1, 2, \ldots D$.

The output of the last pooling layer is then flattened into a vector, $\mathbf{l} \in \mathbb{R}^{14 \times 1}$, which is called CNN features.

A fusion layer is introduced to combine the extracted handcrafted features ($\mathbf{p}$) and the CNN extracted features ($\mathbf{l}$). In

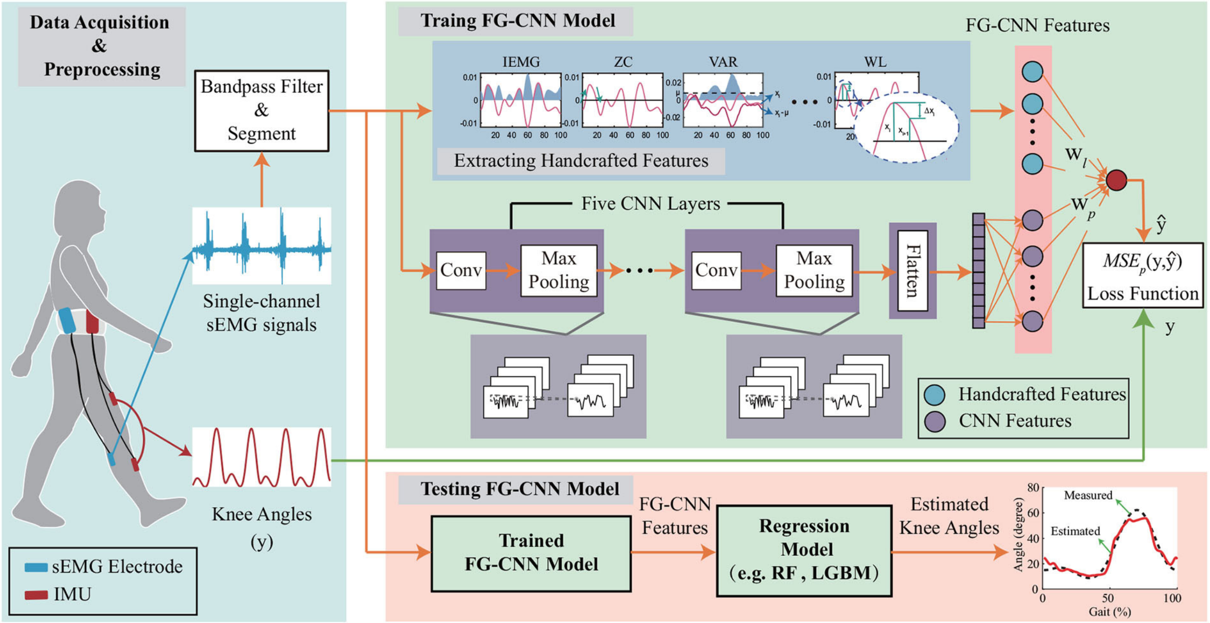


FIGURE 1
The overall framework of the FG-CNN-based motion estimation. The FG-CNN consists of the operation of extracting handcrafted features, five CNN layers, one flatten layer, and one fusion layer for fusing CNN features and handcrafted features. IMU represents the inertial measurement unit; Conv denotes the convolutional layers; IEMG, VAR, ZC, and WL represent the handcrafted features. The symbols w_l and w_p, respectively, represent the weights of the handcrafted and CNN features in the fusion layer. RF and LGBM represent the random forest model and light gradient boosting machine, respectively.

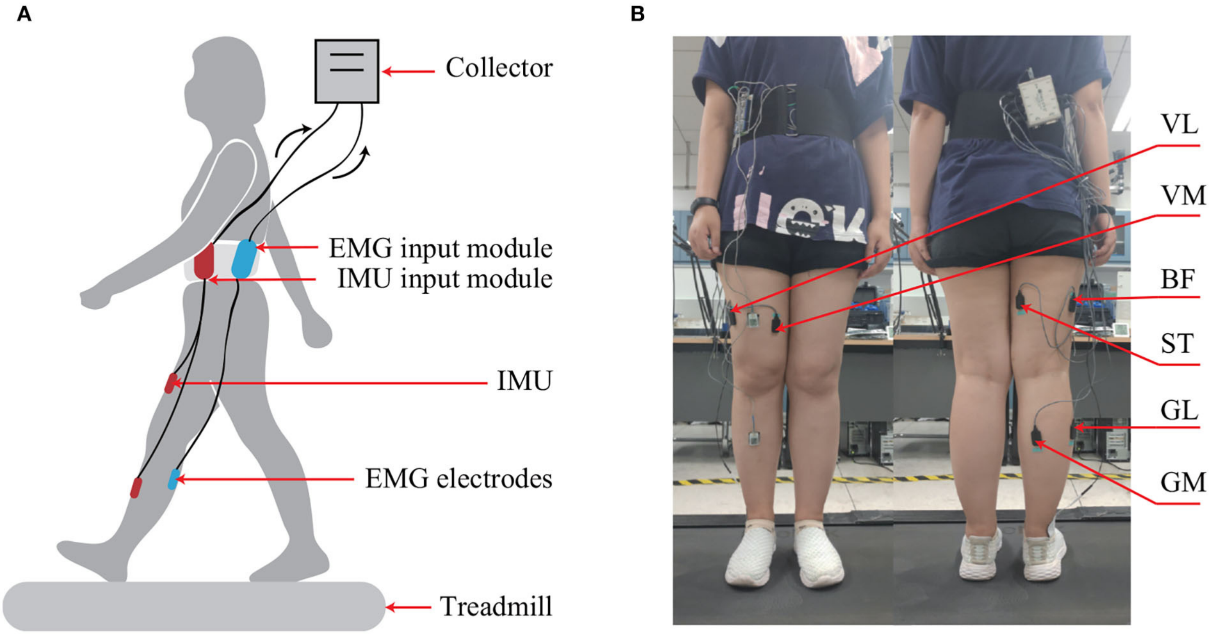


FIGURE 2
The experimental setup. **(A)** Schematic diagram of the experimental setup. **(B)** The locations of sEMG electrodes.

the fusion scheme, both the handcrafted and CNN extracted features are fed into the last fully connected layer to estimate the joint motion ($\hat{\mathbf{y}}$) (see Figure 1). Instead of using the LeakyReLU, the tanh function, which normalizes fused features to [-1,1], is chosen to avoid the blow-up phenomenon. The fusion scheme is defined as follows:

$$\hat{\mathbf{y}} = \tanh(\mathbf{w}_p\mathbf{p} + \mathbf{w}_l\mathbf{l}), \tag{4}$$

where $\mathbf{w_p}$ and $\mathbf{w_l}$ refer to the connection weights.

The built FG-CNN model is trained using a mean squared error function (MSE_p) as follows:

$$\begin{aligned} MSE_p &= \frac{1}{N}\sum_{i=1}^{N}(\mathbf{y}(i) - \hat{\mathbf{y}}(i))^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}(\mathbf{y}(i) - \tanh(\mathbf{w}_p\mathbf{p}(i) + \mathbf{w}_l\mathbf{l}(i)))^2, \end{aligned} \tag{5}$$

where $\mathbf{y}$ and $\hat{\mathbf{y}}$ denote the measured and estimated knee joint movements, respectively. N is the total number of samples.

During the training, Adam algorithm (Diederik and Jimmy, 2015) is utilized to update the weights. Unlike the traditional CNN, the proposed FG-CNN includes the handcrafted features in the weight updating process to guide the CNN to extract features. Correspondingly, the weights of the fusion layer of the FG-CNN are updated in the following steps.

Step 1: Compute the gradients:

$$\begin{cases} \dfrac{\partial MSE_p}{\partial \mathbf{w}_p} = \dfrac{\partial MSE_p}{\partial \hat{\mathbf{y}}(i)}\dfrac{\partial \hat{\mathbf{y}}(i)}{\partial \mathbf{w}_p} = -\dfrac{2\mathbf{p}}{N}\displaystyle\sum_{i=1}^{N}(\mathbf{y}(i) - \hat{\mathbf{y}}(i))(1 - \hat{\mathbf{y}}(i)^2) \\ \dfrac{\partial MSE_p}{\partial \mathbf{w}_l} = \dfrac{\partial MSE_p}{\partial \hat{\mathbf{y}}(i)}\dfrac{\partial \hat{\mathbf{y}}(i)}{\partial \mathbf{w}_l} = -\dfrac{2\mathbf{l}}{N}\displaystyle\sum_{i=1}^{N}(\mathbf{y}(i) - \hat{\mathbf{y}}(i))(1 - \hat{\mathbf{y}}(i)^2) \end{cases} \tag{6}$$

Step 2: Update the weights:

$$\begin{cases} \mathbf{w}_p \leftarrow \mathbf{w}_p - \dfrac{\mathbf{a_p} + \mathbf{b}\frac{\partial MSE_p}{\partial \mathbf{w}_p}}{\sqrt{\mathbf{c_p} - \mathbf{d}(\frac{\partial MSE_p}{\partial \mathbf{w}_p})^2} + \epsilon} \\ \mathbf{w}_l \leftarrow \mathbf{w}_l - \dfrac{\mathbf{a_l} + \mathbf{b}\frac{\partial MSE_p}{\partial \mathbf{w}_l}}{\sqrt{\mathbf{c_l} - \mathbf{d}(\frac{\partial MSE_p}{\partial \mathbf{w}_l})^2} + \epsilon}, \end{cases} \tag{7}$$

where ϵ denotes the constant, $\epsilon = 10^{-8}$. $\mathbf{a_p}$, $\mathbf{a_l}$, $\mathbf{b}$, $\mathbf{c_p}$, $\mathbf{c_l}$, and $\mathbf{d}$ are given as follows:

$$\begin{aligned} &\mathbf{a_p} = \frac{\alpha\beta_1\mathbf{s_p}}{1-\beta_1^t} \quad \mathbf{a_l} = \frac{\alpha\beta_1\mathbf{s_l}}{1-\beta_1^t} \quad \mathbf{c_p} = \frac{\alpha\beta_2\mathbf{r_p}}{1-\beta_2^t} \\ &\mathbf{c_l} = \frac{\alpha\beta_2\mathbf{r_l}}{1-\beta_2^t} \quad \mathbf{b} = \frac{\alpha(1-\beta_1)}{(1-\beta_1^t)} \quad \mathbf{d} = \frac{\alpha(1-\beta_2)}{(1-\beta_2^t)}, \end{aligned} \tag{8}$$

where $\mathbf{s_p}$ and $\mathbf{s_l}$ denote the first and second order moment vector of handcrafted features, $\mathbf{r_p}$ and $\mathbf{r_l}$ denote the first and second order moment vector of FG-CNN features, and β_1, β_1^t, β_2, β_2^t are the exponential decay rates for the moment estimations. α denotes the step-size.

As seen in Equation (7), the updated weights of the fusion layer are updated using both handcrafted and CNN extracted features. With a learning rate of 0.001, the FG-CNN was trained on NVIDIA Quadro P5000 GPU by using the Adam algorithm for 50 epochs in our experiments.

2.2. Experimental protocol and data acquisition

Eight healthy subjects (six men and two women, aged 25.13 ± 3.27 years old) participated in the experiments. All experiments were conducted in accordance with the ethical standards encoded in the latest Declaration of Helsinki. Before the experiments, each participant was fully informed of the experimental purpose and procedures and provided their written consent to participate in this study. The experiments were proved by the local ethics committee of Nankai University.

The experiment scheme is shown in Figure 2. Six channels of sEMG electrodes were respectively placed on six muscles, namely, vastus lateralis (VL), vastus medialis (VM), biceps femoris (BF), semitendinosus (ST), lateral gastrocnemius (LG), and medial gastrocnemius (MG) (Lu et al., 2021), which are relative to the knee joint motion. The data of sEMG were obtained by an acquisition system (Bagnoli, Delsys, MA, USA) under the sampling rate of 5 kHz. At the same time, the data of knee joint angles were also measured using two IMUs with the sampling rate of 100 Hz. Each subject was asked to walk for 1 min with a velocity of 1.25 m/s on the treadmill per trial and perform 11 trials in total with a 3-min rest in between to avoid muscle fatigue. The raw sEMG signals were pre-processed using a Butterworth bandpass filter with cutoff frequencies of 10 Hz and 500 Hz. The sEMG signals measured from the six muscles were tested separately to estimate the knee joint movements. Five cross-validations were used to split the dataset into training data and testing data that are independent of each other. For each time, the proposed FG-CNN model was trained on 80% of data and evaluated on 20% of data.

2.3. Evaluation indicators

The estimation performance of the trained FG-CNN was evaluated by using two indicators: normal root-mean-squared error (NRMSE) and correlation coefficients (CC), respectively (Kwon and Kim, 2011; Qing et al., 2022). The NRMSE is used to reflect the deviation between the measured and estimated knee joint angles, in percentage (%) (Zhu et al., 2022). The CC value can reflect the strength of the correlation between the measured and estimated knee joint angles, which is close to 1, which indicates meaning a good match between the measurement and the estimation.

The NRMSE is defined as

$$\mathrm{NRMSE} = \frac{\sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}}{y_{\max} - y_{\min}}, \tag{9}$$

where y_i and $\hat{y}_i$ are the measured and estimated knee joint angles, respectively, n denotes the total number of samples, and $y_{\max}$ and $y_{\min}$ are the maximum and minimum values of the measured angles, respectively.

CC is defined as

$$\mathrm{CC} = \frac{C_{y\hat{y}}}{\sigma_y \cdot \sigma_{\hat{y}}}, \tag{10}$$

where $C_{y\hat{y}}$ denotes the covariance between the measured and estimated angles and σ_y and $\sigma_{\hat{y}}$ represent the standard deviation of measured and estimated angles, respectively.

2.4. Statistical analysis

Statistical analyses were performed to compare the estimation performances between the proposed FG-CNN and the other compared methods. As the evaluation indicators were not normally distributed, the Kruskal-Wallis test was conducted to compare the different estimation methods with the FG-CNN to identify differences in NRMSE and CC. For all tests, the significance level was set at a $p < 0.05$. Statistical analyses were conducted with MATLAB (MathWorks, Natick, MA, USA).

3. Experimental results

3.1. Performance of knee joint estimation

To verify the effectiveness of the proposed method in extracting implicit features from single-channel sEMG signals, a comparison study was carried out, in which the handcrafted features, the CNN features, and the FG-CNN features were separately used to estimate the knee joint movements *via* a random forest (RF) regression model (see Figure 1).

The details are given as follows:

1) HF-RF: Fourteen handcrafted features were fed into the RF regression.
2) CNN-RF: Twenty-eight CNN features were used as the inputs of the RF regression.
3) FG-CNN-RF: Twenty-eight FG-CNN features were fed into the RF regression.

Figure 3 shows the knee joint angles measured using the IMUs (i.e., the reference) and the ones estimated using the HF-RF, CNN-RF, and FG-CNN-RF. For all approaches, the estimated angles from FG-CNN-RF are closer to the reference than those from HF-RF and CNN-RF. For all muscles, it can also be seen that the estimated knee joint angles using single-channel sEMG signals measured from LG or MG are more accurate than those estimated using sEMG signals from VL, VM, BF, or ST.

Meanwhile, to quantitatively evaluate the estimation results, the indicators of NRMSE and CC were used. The NRMSE values using the LG and MG from the FG-CNN-RF were, respectively, 15.2±3.5% (LG) and 15.7±3.1% (MG), and the CC values were, respectively, 0.858 ± 0.085 (LG) and 0.856 ± 0.057 (MG). Although the estimation performance obtained from single-channel sEMG signals (LG or MG) is slightly lower than that from six-channel signals (NRMSE: 10.2±1.4%; CC: 0.948±0.013, shown in Figure 4), the estimation accuracy using the proposed single-channel based method is comparable to the that using six-channel sEMG. The results can be explained by the muscle synergy analysis, in which the muscles are controlled in groups to generate a desired joint movement, and all relevant muscles share some common components.

For each muscle, the mean NRMSE values of the FG-CNN-RF were lowest and decreased by 15% (VL), 19.2% (VM), 8.1% (BF), 13.3% (ST), 37.6% (LG), and 39.4% (MG) compared to those of the HF-RF and by 11.3% (VL), 9% (VM), 8.6% (BF), 12.1% (ST), 18.8% (LG), and 19.2% (MG) compared to those of the CNN-RF (shown in Figure 5A). The average CC values of FG-CNN-RF were highest, which were increased by 33.4% (VL), 59.3% (VM), 42.4% (BF), 51.7% (ST), 27.3% (LG), and 33.8% (MG) compared to those of the HF-RF, and increased by 32.2% (VL), 24.8% (VM), 34.1% (BF), 42.8% (ST), 7.9% (LG), and 9.8% (MG) compared to those of the CNN-RF (see Figure 5B).

The Kruskal-Wallis test was conducted to identify differences in NRMSE and CC between the case using FG-CNN features and the cases using the other two kinds of features (handcrafted features and CNN features). Both the NRMSE and CC values of FG-CNN had statistically significant differences with respect to handcrafted features and CNN features (see Figure 5). The above results implies that the FG-CNN features contain more muscular information than the handcrafted features or CNN features for ensuring a more accurate estimation of knee joint movement.

3.2. Comparison of various regression models

To further test the effectiveness of the FG-CNN features, five difference regression models were used, including RF regression, light gradient boosting machine (LightGBM), multilayer perceptron (MLP), support vector regression (SVR),

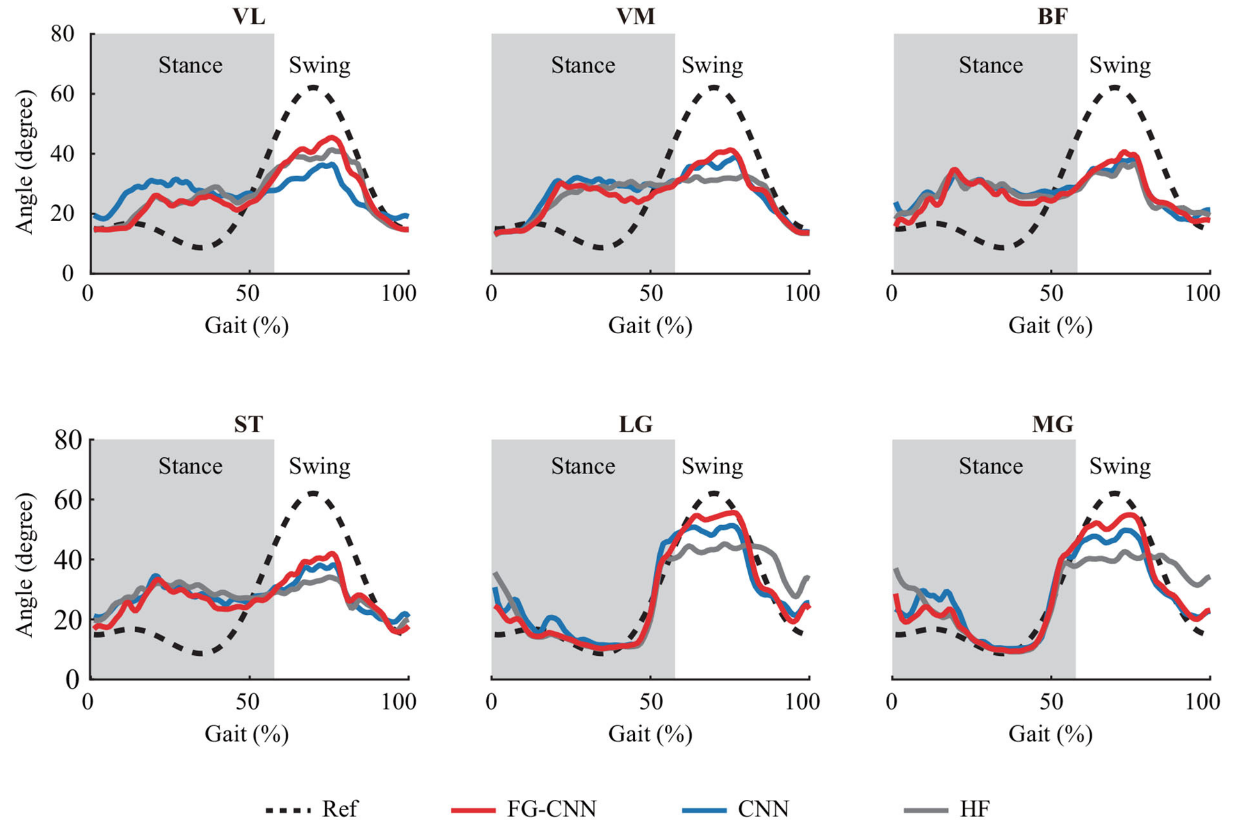


FIGURE 3
Knee joint angles profiles averaged across 57 strides with subject 2. Dashed and solid lines are reference angles measured by IMUs and estimated angles by FG-CNN-RF, CNN-RF, and HF-RF, using single-channel sEMG signals. The gray shaded area indicates the stance phase.

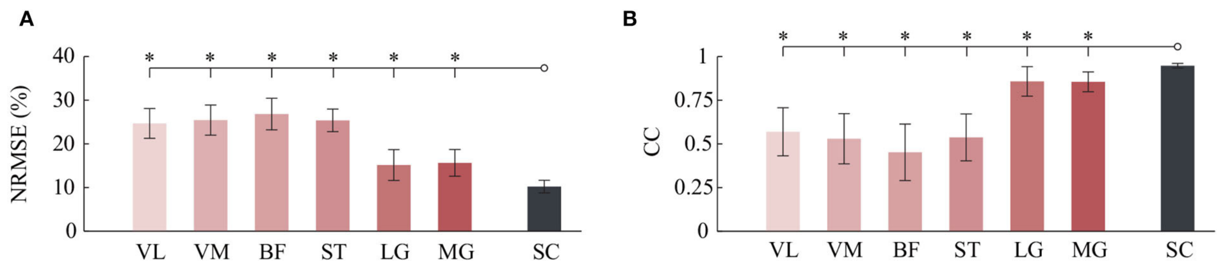


FIGURE 4
Comparison of angle estimation accuracy between six kinds of single-channel sEMG and six-channel sEMG (SC). **(A)** Comparison of NRMSE. **(B)** Comparison of CC. Bars are means, error bars are standard error of the mean (SEM), and asterisks denote statistically significant differences with respect to the six-channel sEMG ($P < 0.05$).

and k-nearest neighbors (KNN) regression. The inputs of the five models were FG-CNN features. Figure 6 shows the NRMSE and CC values for knee joint movement estimation using the five regression models. The regression models of RF, LightGBM, and MLP have similar estimation performance with lower NRMSE and greater CC values than the other two models.

The Kruskal-Wallis test was also used to determine if there were differences between the RF and the other four regression models. Table 1 showed the statistical analysis results on CC and NRMSE. For each single-channel sEMG, the NRMSE values of RF significantly decreased compared to these of SVR and KNN. Meanwhile, the CC values of RF significantly increased

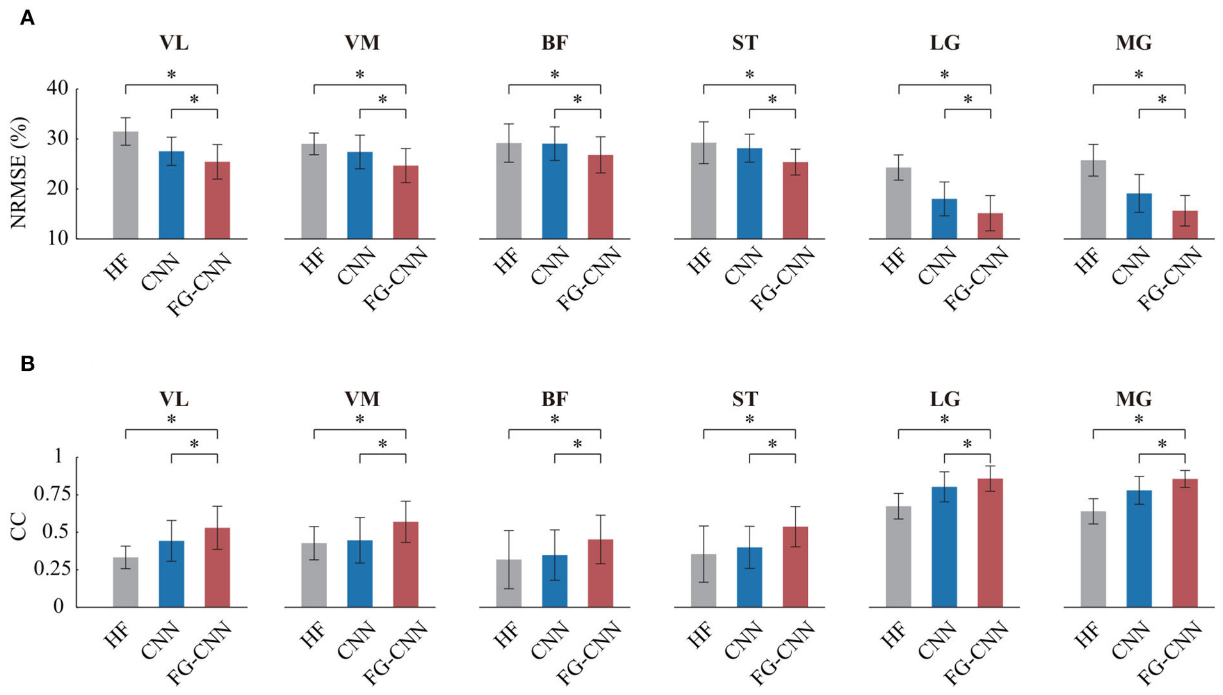


FIGURE 5
Comparison of angle estimation accuracy by HF, CNN, and FG-CNN. **(A)** Comparison of NRMSE. **(B)** Comparison of CC. Bars are means, error bars are standard error of the mean (SEM), and asterisks denote statistical significance ($P < 0.05$).

compared to those of SVR and KNN. The CC values of MLP from LG and MG were different compared to that from the RF (LG: $P = 0.015$; MG: $P = 0.0015$), while no significant differences were found between RF and the other two models (LightGBM and MLP) on NRMSE and CC.

4. Discussion

Human joint movement estimation using multi-channel sEMG signals has been widely used to enable HRI systems' intuitive and voluntary control. However, some issues have inhibited the collection of high-quality sEMG signals from all relevant muscles, such as weakness or spasticity of one or more specific muscles, mechanical/signal interference between EMG sensors and wearable robots/environment, discomfort for long-term use, etc. Therefore, using fewer channels or single-channel sEMG signals is of practical importance. It remains unknown whether the continuous knee joint movement can be estimated using single-channel sEMG signals. In addition, it is a challenge to ensure high estimation accuracy using only single-channel sEMG signals. This study verified the feasibility of continuous joint movement estimation only using single-channel sEMG signals and proposed a new feature extraction scheme, namely FG-CNN, to improve the estimation performance effectively.

The main advantage of FG-CNN is that it contains both handcrafted features and CNN features, which can improve the motion estimation performance just by using single-channel sEMG signals (as shown in Figure 5). When multi-channel sEMG signals were used, the handcrafted features or CNN features contained enough muscular information and could be successfully adopted in motion estimation. However, with the number of channels decreasing, the muscular information in the handcrafted features or CNN features will not be sufficient. To further extract implicit features, this study fed the handcrafted features into a fusion layer to guide the extraction of CNN features. Compared with both the handcrafted features and CNN features, the FG-CNN features can ensure a more accurate joint movement estimation, which implies that the FG-CNN features contained more muscular information.

Based on the single-channel sEMG signals of LG or MG, the estimated angle profiles were similar to the reference during the gait cycle, as shown in Figure 3. On the contrary, for single-channel sEMG of VL, VM, BF, or ST, the trends of estimated angles profiles were reversed, especially at the stance phase. In this study, the used estimation methods, such as random forest regression and light gradient boosting machine, directly mapped the sEMG features to knee angles without any biomechanics. The estimation performance was influenced by the correlation between inputs (sEMG features) and outputs (knee angles). In

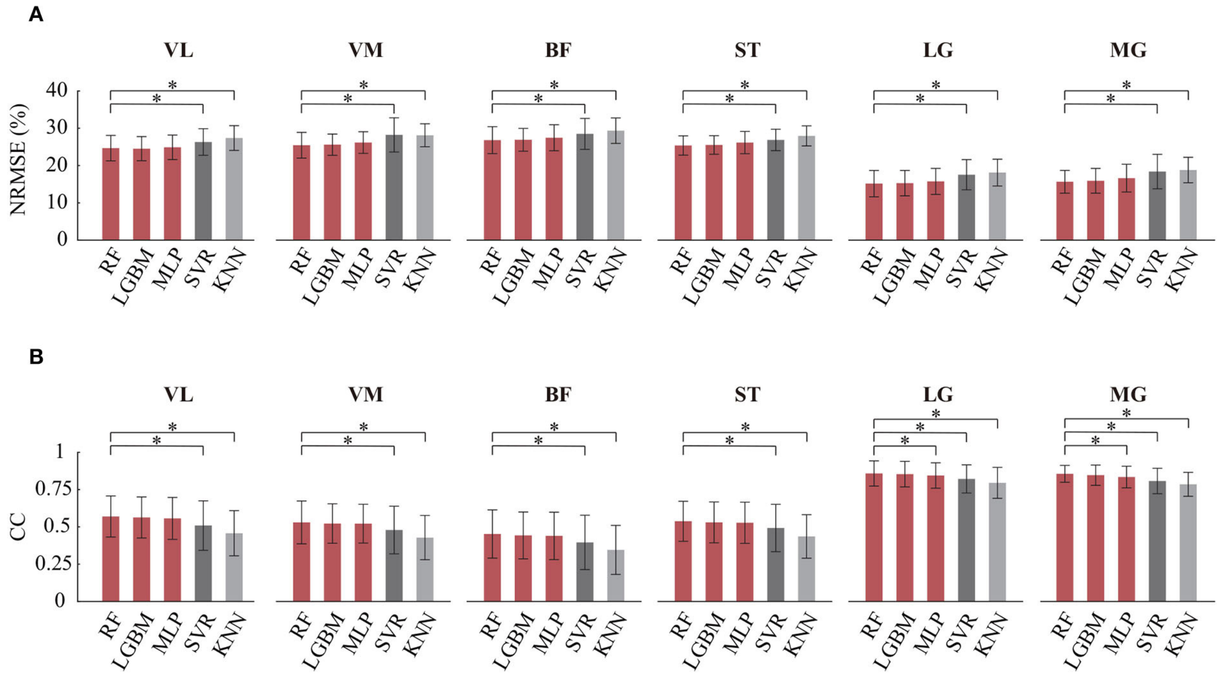


FIGURE 6
Comparison of angle estimation accuracy on different regression models. RF, random forest regression; LGBM, gradient boosting machine; MLP, multilayer perceptron (MLP); SVR, support vector regression; KNN, k-nearest neighbors regression. **(A)** Comparison of NRMSE. **(B)** Comparison of CC. Bars are means, error bars are standard error of the mean (SEM), and asterisks denote statistically significant differences with respect to the RF ($P < 0.05$).

a gait cycle, compared to the other four muscle activity profiles, the correlation between the gastrocnemius activity profiles and the knee angle profiles was stronger (shown in Figure 7). This can explain why the estimation accuracy from LG or MG is highest.

sEMG will be changed on different days and subjects due to the changing skin impedance, which is affected by physiological factors such as subcutaneous tissue, the physiological cross-sectional area of the muscle, or dynamic factors such as sweat. It is a complex and significant issue need to be addressed in practical application. To date, some studies made efforts to address this issue. Bao et al. established a two-stream CNN with shared weights to enhance inter-subject performances in the wrist kinematics estimation (Bao et al., 2021). The results showed that the NRMSE and CC values were 22% and 0.67, respectively, which outperformed a state-of-the-art transfer learning method. Dantas et al. demonstrated that the CNN decoder performed significantly better than polynomial Kalman filters in most analyzed cases of temporal separations (0–150 days) between the acquisition of the training and testing datasets (Dantas et al., 2019). The above studies demonstrated the potential for the utilization of CNN to address the limitations of using sEMG on different days and subjects. Therefore, although the proposed FG-CNN degrades its estimation performance using sEMG from different subjects or days, it is possible not to degrade too much. Future studies should advance in this direction.

In comparison with the previous studies, the main advantage of the proposed method is to achieve good estimation performance using single-channel sEMG signals rather than the multi-channels, which can be used to improve the usability of low limb wearable robotics in weakness or spasticity of one or more specific muscles (Wei et al., 2022). Although the mean CC values of the proposed FG-CNN with single-channel EMG signals were slightly lower than the CC values of the state-of-the-art studies with multi-channel sEMG (shown in Table 2), the mean CC values of the proposed FG-CNN is around 0.85, which suggests the strong correlation strength between the estimated and the measured.

It is worth noting that the proposed methods have limitations. One method is that the sEMG data from each subject were recorded on the same day and the proposed FG-CNN was trained and tested on the same subject. This study did not consider the influence of sEMG changes on different days and subjects. Future studies should advance in this direction. Another method is that the estimation performance still has room for improvement. Future studies would be required to recruit more subjects and further to improve the accuracy of knee joint estimation by advanced single-channel sEMG-based methods.

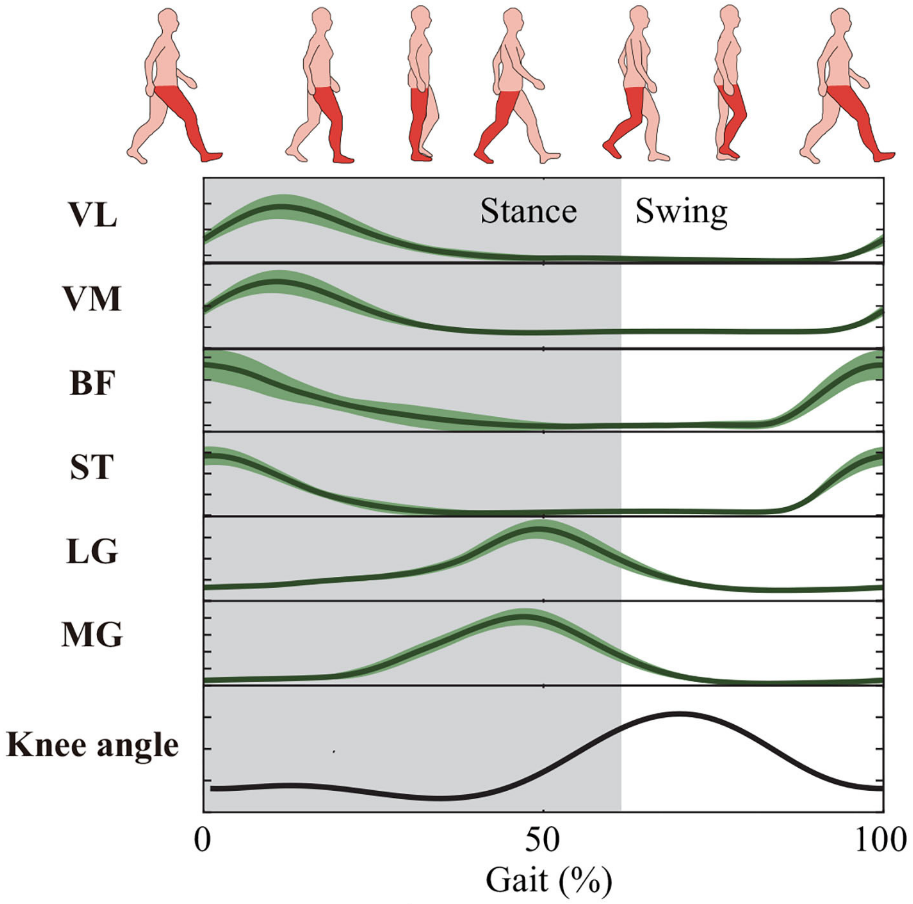


FIGURE 7
Muscle activity profiles and knee angle profiles during the gait cycle.

TABLE 1 Statistical analysis results between RF and the other regression models (including LGBM, MLP, SVR, and KNN) on NRMSE and CC.

Muscle	NRMSE				CC			
	LGBM	MLP	SVR	KNN	LGBM	MLP	SVR	KNN
VL	0.7847	0.6135	**0.0000**	**0.0000**	0.5101	0.2079	**0.0019**	**0.0000**
VM	0.4528	0.0867	**0.0000**	**0.0000**	0.3551	0.2391	**0.0039**	**0.0000**
BF	0.6847	0.1712	**0.0014**	**0.0000**	0.2615	0.1744	**0.0016**	**0.0000**
ST	0.3291	0.1341	**0.0000**	**0.0000**	0.3622	0.3445	**0.0120**	**0.0000**
LG	0.4885	0.0993	**0.0000**	**0.0000**	0.3879	**0.0150**	**0.0000**	**0.0000**
MG	0.5874	0.0806	**0.0000**	**0.0000**	0.3993	**0.0015**	**0.0000**	**0.0000**

The bold values indicate the statistical significance values of $p < 0.05$ with respect to RF.

TABLE 2 Comparison with related research.

References	Number of sensors	Method	Performance (knee angle)
Yi et al. (2022)	9	LSTM	CC = 0.88 ± 0.04
Zhong et al. (2022)	8	Muscle synergy-driven ANFIS model	CC = 0.92 ± 0.02
Wang et al. (2021)	8	Multi-branch neural network	CC = 0.96 ± 0.03

This work	**1**	FG-CNN	CC = 0.858 ± 0.085 (LG)
			CC = 0.856 ± 0.057 (MG)

The bold values indicate the results of this work.

5. Conclusion

In this study, a new feature extraction method, namely FG-CNN, was proposed to estimate human knee joint movement using single-channel sEMG signals. To verify the effectiveness of this method, sEMG signals measured from six muscles, including the vastus lateralis, the vastus medialis, the biceps femoris, the semitendinosus, the lateral or medial gastrocnemius (LG or MG), were separately evaluated for estimating knee joint movements using the proposed FG-CNN. The experimental results showed that combined handcrafted-CNN features outperform either the handcrafted features or the CNN features. In addition, the results demonstrated that the proposed FG-CNN method with sEMG signals from LG or MG can effectively estimate the movements with average NRMSE values of 15.2 ± 3.5% (LG) and 15.7 ± 3.1% (MG) and average CC values of 0.858 ± 0.085 (LG) and 0.856 ± 0.057 (MG). The performance of the proposed signal-channel sEMG-based FG-CNN method was comparable to those of traditional multi-channel sEMG-based methods. The proposed FG-CNN have the potential to provide an alternative means for knee joint movement estimation to overcome the aforementioned limitations faced by the traditional multi-channel sEMG-based methods.

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 the local Ethics Committee of Nankai University. The patients/participants provided their written informed consent to participate in this study.

Author contributions

WH, NY, and JH initiated and supervised the research project. SZ and JL carried out the research. SZ analyzed the data. SZ, JL, WH, NY, and JH interpreted the data and drafted and revised the manuscript. All authors read and approved the final manuscript.

Funding

This work was supported by the National Natural Science Foundation of China (U1913208, 61873135, and 61720106012).

Acknowledgments

The authors are grateful to all participants in experiments.

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Atzori, M., Cognolato, M., and Mller, H. (2016). Deep learning with convolutional neural networks applied to electromyography data: a resource for the classification of movements for prosthetic hands. *Front. Neurorobot.* 10, 9. doi: 10.3389/fnbot.2016.00009

Bao, T., Zaidi, S. A. R., Xie, S., Yang, P., and Zhang, Z. Q. (2021). Inter-subject domain adaptation for cnn-based wrist kinematics estimation using semg. *IEEE Trans. Neural. Syst. Rehabil. Eng.* 29, 1068–1078. doi: 10.1109/TNSRE.2021.3086401

Bi, L., Feleke, A. G., and Guan, C. (2019). A review on emg-based motor intention prediction of continuous human upper limb motion for human-robot collaboration. *Biomed. Signal Process. Control* 51, 113–127. doi: 10.1016/j.bspc.2019.02.011

Caulcrick, C., Huo, W., Hoult, W., and Vaidyanathan, R. (2021). Human joint torque modelling with mmg and emg during lower limb human-exoskeleton interaction. *IEEE Robot. Automat. Lett.* 6, 7185–7192. doi: 10.1109/LRA.2021.3097832

Côté-Allard, U., Campbell, E., Phinyomark, A., Laviolette, F., Gosselin, B., and Scheme, E. (2020). Interpreting deep learning features for myoelectric control: a comparison with handcrafted features. *Front. Bioeng. Biotechnol.* 8, 158. doi: 10.3389/fbioe.2020.00158

Dantas, H., Warren, D. J., Wendelken, S. M., Davis, T. S., Clark, G. A., and Mathews, V. J. (2019). Deep learning movement intent decoders trained with dataset aggregation for prosthetic limb control. *IEEE Trans. Biomed. Eng.* 66, 3192–3203. doi: 10.1109/TBME.2019.2901882

d'Avella, A., Saltiel, P., and Bizzi, E. (2003). Combinations of muscle synergies in the construction of a natural motor behavior. *Nat. Neurosci.* 6, 300–308. doi: 10.1038/nn1010

Diederik, P. K., and Jimmy, L. B. (2015). "Adam: a method for stochastic optimization," in *International Conference on Learning Representations (ICLR)* (San Diego, CA).

Ding, Q., Han, J., and Zhao, X. (2017). Continuous estimation of human multi-joint angles from semg using a state-space model. *IEEE Trans. Neural. Syst. Rehabil. Eng.* 25, 1518–1528. doi: 10.1109/TNSRE.2016.2639527

Ding, Q., Zhao, X., Han, J., Bu, C., and Wu, C. (2019). Adaptive hybrid classifier for myoelectric pattern recognition against the interferences of uutlier motion, muscle fatigue, and electrode doffing. *IEEE Trans. Neural. Syst. Rehabil. Eng.* 27, 1071–1080. doi: 10.1109/TNSRE.2019.2911316

Dwivedi, S. K., Ngeo, J., and Shibata, T. (2020). Extraction of nonlinear synergies for proportional and simultaneous estimation of finger kinematics. *IEEE Trans. Biomed. Eng.* 67, 2646–2658. doi: 10.1109/TBME.2020.2967154

Fleischer, C., and Hommel, G. (2008). A human-exoskeleton interface utilizing electromyography. *IEEE Trans. Robot*. 24, 872–882. doi: 10.1109/TRO.2008.926860

Gupta, R., and Agarwal, R. (2019). Single channel emg-based continuous terrain identification with simple classifier for lower limb prosthesis. *Biocybern. Biomed. Eng*. 39, 775–788. doi: 10.1016/j.bbe.2019.07.002

He, J., Sheng, J., Zhu, X., Jiang, C., and Jiang, N. (2019). Spatial information enhances myoelectric control performance with only two channels. *IEEE Trans. Ind. Inf*. 15, 1226–1233. doi: 10.1109/TII.2018.2869394

Ioffe, S., and Szegedy, C. (2015). "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning* (Lille), 448–456.

Jiang, N., Rehbaum, H., Vujaklija, I., Graimann, B., and Farina, D. (2014). Intuitive, online, simultaneous, and proportional myoelectric control over two degrees-of-freedom in upper limb amputees. *IEEE Trans. Neural. Syst. Rehabil. Eng*. 22, 501–510. doi: 10.1109/TNSRE.2013.2278411

Kubota, K., Hanawa, H., Yokoyama, M., Kita, S., Hirata, K., Fujino, T., et al. (2021). Usefulness of muscle synergy analysis in individuals with knee osteoarthritis during gait. *IEEE Trans. Neural. Syst. Rehabil. Eng*. 29, 239–248. doi: 10.1109/TNSRE.2020.3043831

Kumar, D. K., Arjunan, S. P., and Singh, V. P. (2013). Towards identification of finger flexions using single channel surface electromyography able bodied and amputee subjects. *J. Neuroeng. Rehabil*. 10, 50. doi: 10.1186/1743-0003-10-50

Kwon, S., and Kim, J. (2011). Real-time upper limb motion estimation from surface electromyography and joint angular velocities using an artificial neural network for human-machine cooperation. *IEEE Trans. Inform. Tech. Biomed*. 15, 522–530. doi: 10.1109/TITB.2011.2151869

Lu, Y., Wang, H., Hu, F., Zhou, B., and Xi, H. (2021). Effective recognition of human lower limb jump locomotion phases based on multi-sensor information fusion and machine learning. *Med. Biol. Eng. Comput*. 59, 883–899. doi: 10.1007/s11517-021-02335-9

Lu, Z., Stampas, A., Francisco, G. E., and Zhou, P. (2019). Offline and online myoelectric pattern recognition analysis and real-time control of a robotic hand after spinal cord injury. *J. Neural Eng*.,16, 036018. doi: 10.1088/1741-2552/ab0cf0

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). "Rectifier nonlinearities improve neural network acoustic models," in *International Conference on Machine Learning*.

Nam, Y., Koo, B., Cichocki, C., and Choi, S. (2014). Gom-face: Gkp, eog, and emg-based multi-modal interface with application to humanoid robot control. *IEEE Trans. Biomed. Eng*. 61, 453–462. doi: 10.1109/TBME.2013.2280900

Phinyomark, A., Phukpattaranont, P., and Limsakul, C. (2012). Feature reduction and selection for emg signal classification. *Expert Syst. Appl*. 39, 7420–7431. doi: 10.1016/j.eswa.2012.01.102

Phinyomark, A., and Scheme, E. (2018). Emg pattern recognition in the era of big data and deep learning. *Big Data Cogn. Comput*. 2, 21. doi: 10.3390/bdcc2030021

Qing, Z., Lu, Z., Liu, Z., Cai, Y., Cai, S., He, B., et al. (2022). A simultaneous gesture classification and force estimation strategy based on wearable a mode ultrasound and cascade model. *IEEE Trans. Neural Syst. Rehabil. Eng*. 30, 2301–2311. doi: 10.1109/TNSRE.2022.3196926

Shao, J., Niu, Y., Xue, C., Wu, Q., Zhou, X., Xie, Y., et al. (2020). Single-channel semg using wavelet deep belief networks for upper limb motion recognition. *Int. J. Ind. Ergon*. 76, 102905. doi: 10.1016/j.ergon.2019.102905

Spanias, J. A., Perreault, E. J., and Perreault, L. J. (2016). Detection of and compensation for emg disturbances for powered lower limb prosthesis control. *IEEE Trans. Neural Syst. Rehabil. Eng*. 24, 226–234. doi: 10.1109/TNSRE.2015.2413393

Thongpanja, S., Phinyomark, A., Quaine, F., Laurillau, Y., Limsakul, C., and Phukpattaranont, P. (2016). Probability density functions of stationary surface emg signals in noisy environments. *IEEE Trans. Instr. Meas*. 65, 1547–1557. doi: 10.1109/TIM.2016.2534378

Wang, X., Dong, D., Chi, X., Wang, S., Miao, Y., An, M., et al. (2021). semg-based consecutive estimation of human lower limb movement by using multi-branch neural network. *Biomed. Signal Process. Control* 68, 102781. doi: 10.1016/j.bspc.2021.102781

Wei, C., Wang, H., Hu, F., Zhou, B., Feng, N., Lu, Y., et al. (2022). Single-channel surface electromyography signal classification with variational mode decomposition and entropy feature for lower limb movements recognition. *Biomed. Signal Process. Control* 74, 103487. doi: 10.1016/j.bspc.2022.103487

Wei, W., Dai, Q., Wong, Y., Hu, Y., Kankanhalli, M., and Geng, W. (2019). Surface-electromyography-based gesture recognition by multi-view deep learning. *IEEE Trans. Biomed. Eng*. 66, 2964–2973. doi: 10.1109/TBME.2019.2899222

Xiong, D., Zhang, D., Zhao, X., and Zhao, Y. (2021). Deep learning for emg-based human-machine interaction: a review. *IEEE/CAA J. Autom. Sin*. 8, 512–533. doi: 10.1109/JAS.2021.1003865

Yi, C., Jing, F., Zhang, S., Guo, H., Yang, C., Ding, Z., et al. (2022). Continuous prediction of lower-limb kinematics from multi-modal biomedical signals. *IEEE Trans. Circuits Syst*. 32, 2592–2605. doi: 10.1109/TCSVT.2021.3071461

Zhong, W., Fu, X., and Zhang, M. (2022). A muscle synergy-driven anfis approach to predict continuous knee joint movement. *IEEE Trans. Fuzzy Syst*. 30, 1553–1563. doi: 10.1109/TFUZZ.2022.3158727

Zhu, Y., Wu, Q., Chen, B., and Zhao, Z. (2022). Design and voluntary control of variable stiffness exoskeleton based on semg driven model. *IEEE Robot. Automat. Lett*. 7, 5787–5794. doi: 10.1109/LRA.2022.3160668

TYPE Original Research
PUBLISHED 10 November 2022
DOI 10.3389/fnbot.2022.979949

Check for updates

OPEN ACCESS

EDITED BY
Zhan Li,
Swansea University, United Kingdom

REVIEWED BY
Chen Chen,
Shanghai Jiao Tong University, China
Farong Gao,
Hangzhou Dianzi University, China
Angelo Davalli,
National Institute for Insurance Against Accidents at Work (INAIL), Italy

*CORRESPONDENCE
Zhufeng Lu
zhufeng.luk@qq.com
Chen Fan
641154688@qq.com

RECEIVED 28 June 2022
ACCEPTED 03 October 2022
PUBLISHED 10 November 2022

CITATION
Zhang X, Lu Z, Fan C, Wang Y, Zhang T, Li H and Tao Q (2022) Compound motion decoding based on sEMG consisting of gestures, wrist angles, and strength.
Front. Neurorobot. 16:979949.
doi: 10.3389/fnbot.2022.979949

Compound motion decoding based on sEMG consisting of gestures, wrist angles, and strength

Xiaodong Zhang[1,2], Zhufeng Lu[1,2]*, Chen Fan[3]*, Yachun Wang[1,2], Teng Zhang[1,2], Hanzhe Li[1,2] and Qing Tao[4]

[1]School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China, [2]Shaanxi Key Laboratory of Intelligent Robot, Xi'an Jiaotong University, Xi'an, China, [3]Department of Mechanical Engineering, Xi'an Jiaotong University City College, Xi'an, China, [4]School of Mechanical Engineering, Xinjiang University, Wulumuqi, China

This study aimed to highlight the demand for upper limb compound motion decoding to provide a more diversified and flexible operation for the electromyographic hand. In total, 60 compound motions were selected, which were combined with four gestures, five wrist angles, and three strength levels. Both deep learning methods and machine learning classifiers were compared to analyze the decoding performance. For deep learning, three structures and two ways of label encoding were assessed for their training processes and accuracies; for machine learning, 24 classifiers, seven features, and a combination of classifier chains were analyzed. Results show that for this relatively small sample multi-target surface electromyography (sEMG) classification, feature combination (mean absolute value, root mean square, variance, 4th-autoregressive coefficient, wavelength, zero crossings, and slope signal change) with Support Vector Machine (quadric kernel) outstood because of its high accuracy, short training process, less computation cost, and stability ($p < 0.05$). The decoding result achieved an average test accuracy of 98.42 ± 1.71% with 150 ms sEMG. The average accuracy for separate gestures, wrist angles, and strength levels were 99.35 ± 0.67%, 99.34 ± 0.88%, and 99.04 ± 1.16%. Among all 60 motions, 58 showed a test accuracy greater than 95%, and one part was equal to 100%.

KEYWORDS

surface EMG, compound motion decoding, deep learning, machine learning, myoelectric prosthesis

Introduction

Surface electromyography (sEMG) is a bioelectric signal naturally produced during the neural activation of muscles (Vredenbregt and Rau, 1973). Through the mapping relationship between the activation degree and the position of muscles, sEMG contains the movement intention of the human body. It has been considered to be one of the modalities of human-machine interface (HMI) in the context of human-centered robotics (Zhang et al., 2015). Compared with other bioelectric signals, such as electroencephalograms, sEMG shows stronger controllability, more decoding patterns, and higher stability. As one popular representation of human intention, sEMG gets its widest application in controlling a myoelectric hand (De Luca, 1997), an exoskeleton (Kiguchi and Hayashi, 2012), and so on.

To realize the sEMG-based control, a number of research studies focused on gesture decoding were carried out first. As early as the 1970s, Taylor D began using the sEMG collected by multi-electrode arrays to control upper limb prostheses (Wirta et al., 1978). In 2007, Chu et al. (2007) achieved an average accuracy of 97.4% in nine kinds of hand motion decoding with four surface electrodes (Chu et al., 2007). In 2016, Adenike realized the decoding of 19 classes, including hand grasps and individual finger motions, and achieved an accuracy of 96% for non-amputees (Adewuyi et al., 2016). To reduce the individual differences, Xue proposed a novel user-independent framework on 13 gesture decoding with an accuracy of 78.15% in 2021, which combined the canonical correlation analysis and optimal transport (Boschmann et al., 2013).

Gesture decoding above took the lead in ensuring the basic grasping function. For a human hand, the complete movement relies not only on the fingers and the palm but also on the cooperation of the wrist and elbow. Beyond basic EMG decoding of gestures, researchers have paid attention to the wrist, the elbow, and the compound motions. In 2010, Zeeshan used the forearm sEMG to classify 19 wrist torques, which showed an 88% accuracy (Khokhar et al., 2010). In 2019, Zhang et al. (2019) proposed a novel preprocessing method for joint force estimation with high-density sEMG. In 2021, Xiang Chen reported a convolutional neural network with a transfer learning strategy in decoding 30 hand gestures involving various states of the finger, elbow, and wrist, which achieved an accuracy of 92.13% with high-density sEMG (Chen et al., 2021). Other studies, such as Zhang et al. (2011), Lu et al. (2014), McIntosh et al. (2016), and so on, have made gesture classification with wrist coupling by combining sEMG with additional sensors.

The sEMG-based decoding of joint movement allows for a more versatile application. Meanwhile, considering the spatio-temporal difference, the decoding of compound actions also guarantees the stability of gesture decoding in multiple poses of the upper limbs. A more practical way to apply the decoded targets to the control of the myoelectric hand is by combining machine intelligence with human intention. By fully using the closed-loop control and the sensory feedback, the decoded target can be viewed as merely enabling a flag, relying upon the prosthesis to complete the blind grasp. In 2011, Hao Dang proposed a stable robotic grasping method based on tactile feedback and hand kinematics, which can further be applied to the blind grasping of the myoelectric hand (Dang et al., 2011). In 2016, Xiong reported the implementation of an anthropomorphic hand for replicating human grasping functions, which realized the blind grasp automatically and was further endued with myoelectric control (Xiong et al., 2016). In 2020, Mayer et al. (2020) reported a closed-loop control method based on tactile feedback to ensure the grasping of the myoelectric hand. Meanwhile, leading commercial prostheses such as the Michelangelo prosthetic hand by Ottobock© (Hashim et al., 2017) and the i-Limb by Össur© (van der Niet et al., 2013) provide customers with EMG-based solutions combined with intelligence control to ensure better practice for daily usage.

By properly combining human intention and machine intelligence, grasping the myoelectric can be more stable and realistic than relying on real-time sEMG-decoding alone. In addition to the grip, through daily observation, we have noticed that different control purposes can exist within the same gesture, such as "grip an egg" vs. "crush an egg." The expression of these detailed purposes has been mostly neglected in the design of the myoelectric hand. Most of the research has focused on one purpose, possibly firmly grasping, to carry out the closed-loop control.

For a fixed gesture, different control purposes (such as griping vs. crushing) mainly correspond to different strength levels. Considering the controllability and the measurement in research, we mapped the strength level to different load levels. In addition to the distinction of control purpose, to ensure the stability of gesture decoding at different wrist angles, there was also a demand for composite motion decoding. In this paper, focusing on flexible myoelectric control and the control purposes of switching, the decoding of compound motions was proposed. These compound motions consisted of the product of gestures, wrist angles, and strength levels, allowing for simultaneous control of the gesture and wrist, as well as switching control purposes. The Materials and Methods section describes the selection basis for compound actions, the experimental setup and data segment, and the various methods adopted for compound motion decoding and performance comparison. The Result section reports the detailed result. The Discussion and the Conclusion sections state the discussion and conclusion separately.

Materials and methods

Demands of compound motion decoding

Control strategy of the myoelectric hand

In reality, the myoelectric hand control merely relies on sEMG-decoding, which is obviously unstable due to the inevitable online misrecognition. Even a tiny misrecognition can result in the failure of a whole task. Moreover, the misoperation caused by the misrecognition may decrease the user's faith in the myoelectric system, resulting in a worse operating state. Therefore, a good combination of human intention and machine intelligence is more realistic to ensure practical controllability.

When reaching into a bag, it is intuitive and straightforward for a natural human hand to grasp objects without any

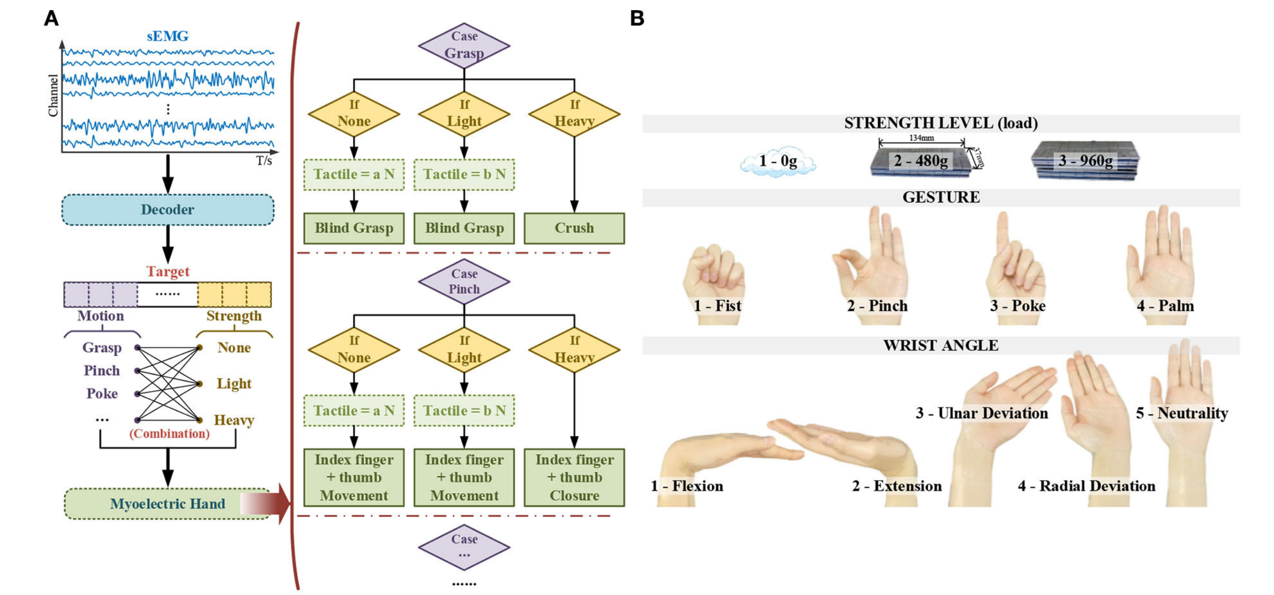


FIGURE 1
The selection basis of compound motions **(A)** the sEMG-based control strategy for the myoelectric hand **(B)** the illustration of compound motions.

pre-existing geometric or visual information (Dang et al., 2011). To implement the same function in the anthropomorphic hand, blind grasping, also known as the hot spot technique, has been studied by many research groups worldwide. As human beings, the grasping gesture was gradually formed according to the tactile perception of the object's surface. Depending on the sensory-feedback closed-loop control, the myoelectric hand can share the same grasping strategy as human hands. By adopting blind grasping to grip firmly, the gesture of the myoelectric hand can automatically be detected instead of being defined by the sEMG-decoding. Compared with sEMG-dependent gesture decoding, such a scheme has higher stability and practicability. Thus, with this strategy, the sEMG target can be viewed as an enabling flag rather than a real-time control command for firmly grasping. It significantly reduced the grasping gestures that needed to be decoded *via* sEMG. According to tactile-based blind grasping, the robust control law tends to use all known fingers to perceive unknown objects, thus completing the power grasp (Shaw-Cortez et al., 2019). By selecting one sEMG enabling flag corresponding to the blind power grasping with all fingers (Shaw-Cortez et al., 2019), another sEMG-based detailed decoding can be left to precision grasp gesture (such as pinch), gestures with specific usage (such as poke) and other sign languages.

During grasping, since there can be multiple purposes under one same gesture (such as griping vs. crushing), the specification of control purpose *via* sEMG is necessary. Under one fixed gesture, different purposes mainly correspond to different levels of muscle strength and can be mapped to multiple control logics of the anthropomorphic hand. In the classic, such myoelectric control directly associated with the proportion of strength is regarded as the direct control (DC) approach (Mereu et al., 2021). By going a step further to distinguish the strength level under different postures, a more diverse purpose of control can be provided to the myoelectric hand based on the existing posture control logic.

Following the statement above, Figure 1A illustrates the sEMG-based control strategy for the myoelectric hand in this work.

Selection of compound motions

According to the control strategy, under different combinations of multi-targets decoded *via* sEMG, the myoelectric hand would adopt different logic to execute the desired movements.

As stated in Feix's report, there are 33 different grasp types in daily usage, including power and precision (Feix et al., 2016). By adopting this sEMG-based strategy above, for power blind grasping with all myoelectric fingers participating, the fist gesture (five-finger fist) was set as the enabling flag, which represents the grasp intention intuitively. Once enabled, according to the control strategy, the myoelectric hand would execute the blind grasp or the crush according to different strength levels. Besides the power grasp, the most representative pinch gesture (with index finger and thumb) was selected in the instruction set for precision grasp. Meanwhile, to fulfill the prior functionality, the poke (as one commonly used specific gesture with index finger stretch) and the palm (corresponding to the reset of myoelectric hand gesture) was also chosen as the sEMG-decoded gestures.

Since the gestures needed to be significantly decoded were reduced to only a few gestures, the core comes down to the decoding stability under various postures. Completing the hand task is inseparable from the flexible movement of the wrist. The angle of the wrist was taken into consideration to ensure the sEMG decoding stability. To ensure robust gesture decoding and further provide potential wrist control ability for the myoelectric prosthesis, discrete wrist angles were selected to form complex wrist-hand compound motions. The wrist has two degrees of freedom flexion/extension and radial/ulnar deviation. Thus, five discrete wrist angles were set, including flexion maximum, extension maximum, radial deviation maximum, ulnar deviation maximum, and neutrality position. As for the angle values, considering the individual differences and the feasibility when being applied to the myoelectric control, each maximum corresponds to the user's own limit.

Three strength levels were determined to increase the difference sufficient to switch the control purpose while providing adequate possibilities for subsequent development. Since the measurement of the strength level lacked calibration, as one initial work, different load levels were adopted to activate corresponding strength levels. Considering the experimental repeatability, the dimension of the adult's hand (Standardization, 1988) and counterweight, and the strength difference across genders, with the *Fe* adhesive weight, 0 g, 480 g, and 960 g, were selected. The underside of the weight was 134 mm in length and 37 mm in width, which fit the size of most adults' hands. Among three levels, 0 g represented the stably grip in blind grasping, 480 g (the approximated weight of a bottled drink) represented the grip with deformation, and 960 g represented the crush.

According to the basis above, by multiplying these four gestures, five wrist angles, and three strength levels, 60 modes decoded *via* sEMG were formed, as illustrated in Figure 1B.

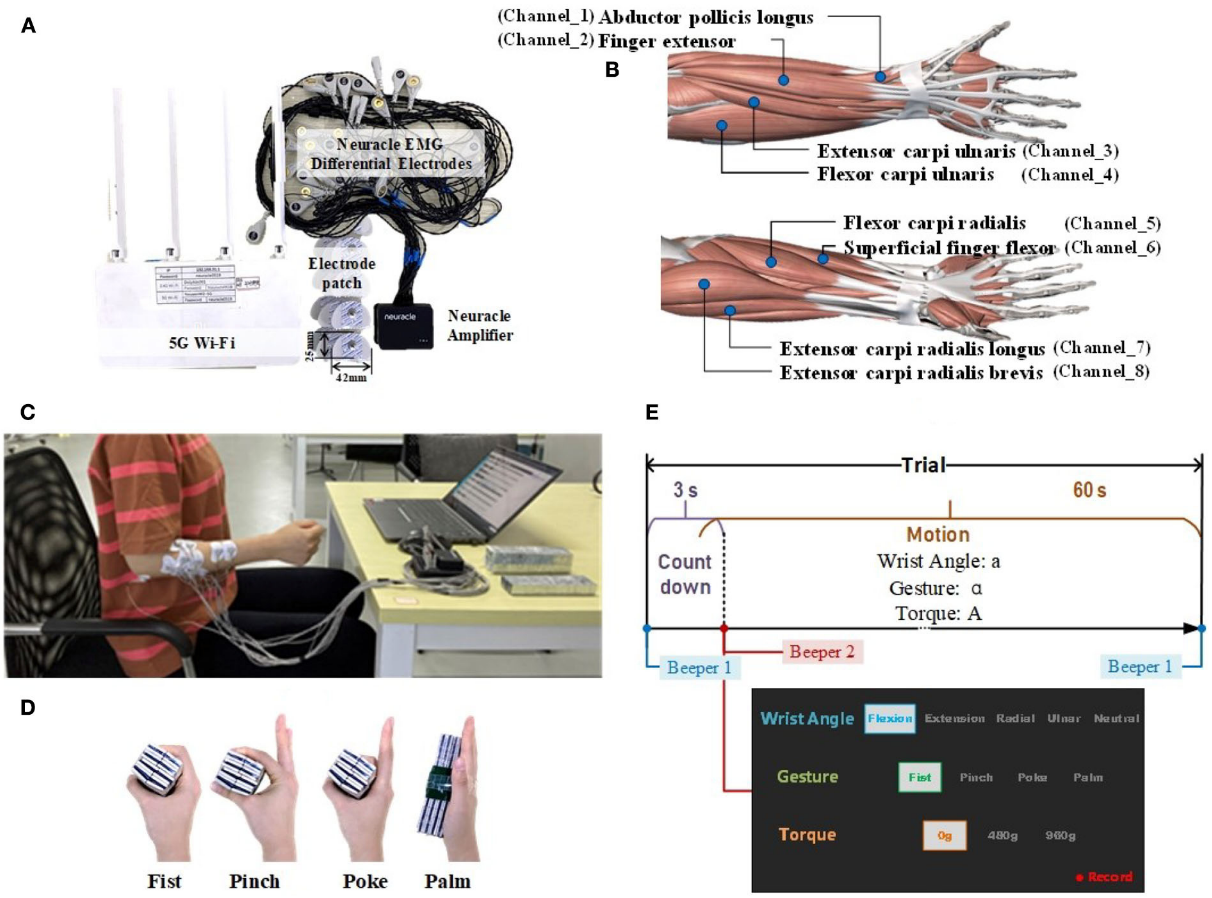

A
Neuracle EMG Differential Electrodes
Electrode patch
Neuracle Amplifier
5G Wi-Fi
25mm
42mm
B
(Channel_1) Abductor pollicis longus
(Channel_2) Finger extensor
Extensor carpi ulnaris (Channel_3)
Flexor carpi ulnaris (Channel_4)
Flexor carpi radialis (Channel_5)
Superficial finger flexor (Channel_6)
Extensor carpi radialis longus (Channel_7)
Extensor carpi radialis brevis (Channel_8)
C
D
Fist
Pinch
Poke
Palm
E
Trial
3 s
60 s
Count down
Motion
Wrist Angle: a
Gesture: α
Torque: A
Beeper 1
Beeper 2
Beeper 1
Wrist Angle
Flexion
Extension
Radial
Ulnar
Neutral
Gesture
Fist
Pinch
Poke
Palm
Torque
0g
480g
960g
● Record

FIGURE 2
Experiment. **(A)** The Neuracle EMG acquisition system. **(B)** The electrode's placement. **(C)** The illustration of upper limb posture. **(D)** The way to load counterweight with different gestures. **(E)** The timing diagram of one session, along with the on-screen prompt example.

Materials

sEMG recording

The commercial wireless portable EMG acquisition system (Neuracle Technology Co., Ltd., Changzhou, PRC) supporting up to 16 channels (each channel consisted of two surface differential electrodes) with a 1000 Hz sampling rate was adopted, as shown in Figure 2A. To decode the composite motion of fingers and wrist, sEMG electrodes were placed on the forearm. Eight large forearm muscles, which play a major role in grasping gestures and wrist movements, were selected, as illustrated in Figure 2B. Eight channels were targeted at these muscles with electrode patches (sized 42 mm in length and 25 mm in width), and the reference electrode was placed at the elbow. Before sticking the sEMG patches, alcohol was used to clean the skin.

Subjects

In total, 12 healthy subjects (aged 22–30 years, ten males and two females) participated in this study (Association, 2013). None of the subjects has a history of the upper extremity or other musculoskeletal complaints. Before starting, each subject was informed of the content, the purpose, and the detailed process of this experiment.

Experimental protocol

The experiment was conducted on the right arm. During the experiment, subjects sat with their elbows naturally hung down, and their forearms raised nearly horizontally. All the motions were completed with the palm kept vertically. The upper body posture is shown in Figure 2C.

In the experiment, each motion was recorded for one trial. Each trial began with a countdown for 3 s, followed by a motion mode hold for 60 s. Between every two trials, a 1-min break was arranged to avoid fatigue. A total of 60 trials were collected, corresponding to 60 modes (multiplied by four gestures, five wrist angles, and three strength levels). During the experiment, the loads were added to the hand. To stably add loads, tapes were attached to ensure the loads could be directly stuck to the palm. Figure 2D demonstrates how to load the counterweight with different gestures.

During the collection, subjects followed the on-screen prompt and the beeper to complete the specified motions. The timing diagram of each trial and the on-screen prompt example are illustrated in Figure 2E.

Dataset

For each subject, a total of 60 trials were collected. The first 20 trials were sEMG under-strength with 0 loads. The 21^{st}-40^{th} trials were with a 480 g load, and the 41^{st}-60^{th} trials

TABLE 1 The specific number ID of each compound motion.

Strength	0 load					480 g load					960 g load				
Wrist	Flexion	Extension	Ulnar	Radial	Neutrality	Flexion	Extension	Ulnar	Radial	Neutrality	Flexion	Extension	Ulnar	Radial	Neutrality
Fist	1	2	3	4	5	21	22	23	24	25	41	42	43	44	45
Pinch	6	7	8	9	10	26	27	28	29	30	46	47	48	49	50
Poke	11	12	13	14	15	31	32	33	34	35	51	52	53	54	55
Palm	16	17	18	19	20	36	37	38	39	40	56	57	58	59	60

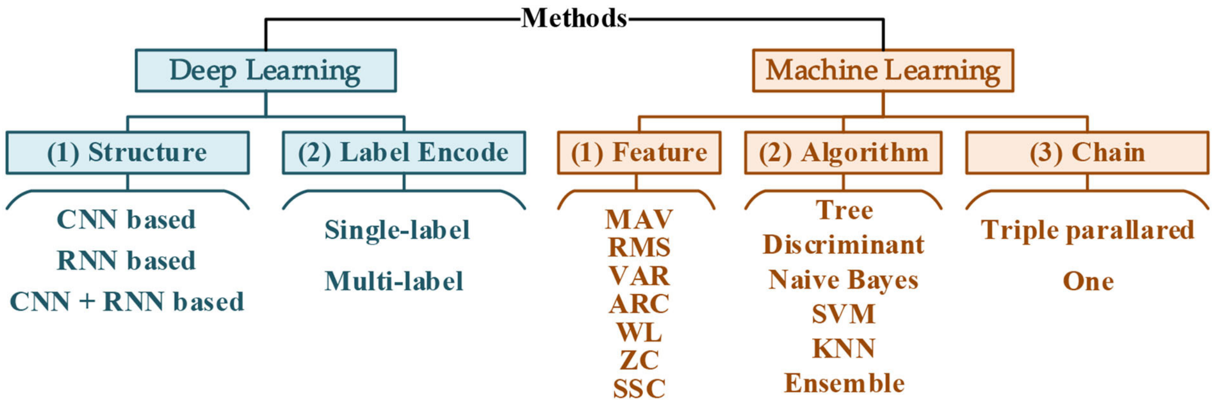


FIGURE 3
Summary of methods.

were with a 960 g load. Each gesture was held for five trials in order at one strength level, with the wrist angle shifted in turn, according to the order in Figure 1B. The specific number ID of each compound motion is listed in Table 1. All data were preprocessed through detrending, the 2nd-order infinite impulse response notch filter at 50 Hz, and the 4th-order Butterworth bandpass filter at 20–250 Hz (Zhao et al., 2020).

Considering the real-time performance of sEMG decoding, Lauer et al. (2000) stated that any delay greater than 200 ms would degrade the performance of one neuro-based task accomplishment. Taking the data acquisition and signal processing processes together, to ensure the system delay was less than 200 ms, data were sliced with a window length of 150 ms and 0 overlap. Thus, for each mode, there were 400 samples.

Decoding methods comparison

To study the decoding performance with these 60 compound motions, both deep learning and machine learning combined with different designs were evaluated, as summarized in Figure 3. The dataset from S1-S6 was adopted in the methods comparison. According to the chronological order, the first 90% of samples of each mode were selected as the training set (i.e., 1s—54s), and the rest were the validation set (i.e., 55s ∼ the 60s). Within each dataset, the samples were shuffled.

Deep learning

Structure

The most well-known typical computations in deep learning were convolutional neural network (CNN), which originated from image decoding (Bengio and Lecun, 1997), and the recurrent neural network (RNN) from the natural language processing (Rumelhart et al., 1986). Developed from the RNN, the long short-term memory (LSTM) layers (Hochreiter and Schmidhuber, 1997) gained broader attention as its variant. Based on the CNN and the LSTM, lots of works achieved impressive results in sEMG decoding (Zhai et al., 2017; Hu et al., 2018; Rehman et al., 2018; Ameri et al., 2019). This work studied three structures (CNN-based, LSTM-based, and CNN+LSTM based) for their performance in 60 compound motion decoding.

CNN-based: In the CNN-based structure (Figure 4A), each convolution block consisted of a 2-dimensional convolution layer (Conv2D), a batch normalization layer (BN), a max pooling layer, and a dropout layer. For multi-convolution blocks, the number of filters descended, as the first block was 32 filters, the second was 16 filters, and the third was 8. For both 2 convolution blocks and 3 convolution blocks, the dense layer with 4096 nodes were deleted.

LSTM-based: Based on the RNN concept, the LSTM layer as a variant was adopted instead for better performance. Both the time positive and the time reverse order were considered the bidirectional LSTM, as illustrated in Figure 4B.

CNN+LSTM-based: This study adopted their combination to take advantage of CNN and LSTM. For the temporal sequence, the 1-dimensional convolution layer (Conv1D) along the temporal domain was taken first, followed by the LSTM, as in Figure 4C. For the convolution block, the same as the CNN-based structure, the first block comprised 32 filters.

Label encoding

Since these 60 compound motions consisted of gestures, wrist angles, strength levels, and two different ways of label encoding were adopted to compare their performance.

One-hot label: One-hot label treated compound motion decoding as a single-labeled multiclass problem. The corresponding output was 60 dimensions (Figure 4D). In accordance, the activation for the output layer was SoftMax.

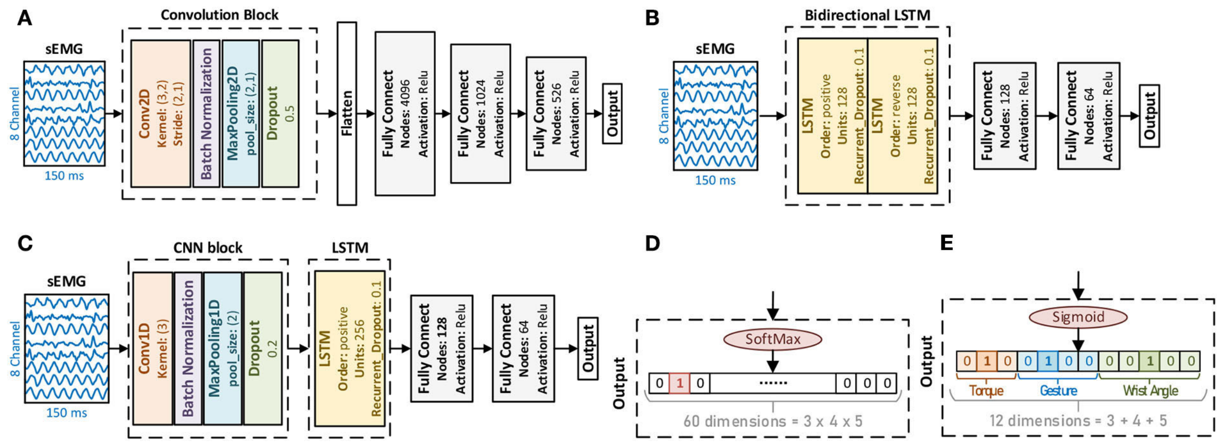


FIGURE 4
Deep learning methods have different structures and label encoding ways. **(A)** CNN-based structure. **(B)** LSTM-based structure. **(C)** CNN+LSTM-based structure. **(D)** One-hot label. **(E)** Multilabel.

TABLE 2 The specific number ID of each compound motion.

Feature	Mathematical definition
MAV	$\mathrm{T}_{MAV} = \frac{1}{N}\sum_{i=1}^{N} \lvert x_i \rvert$
RMS	$\mathrm{T}_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} {x_i}^2}$
VAR	$\mathrm{T}_{VAR} = \frac{1}{N-1}\sum_{i=1}^{N} (x_i - \overline{x})^2$
4th-ARC	$x_i = -\sum_{k=1}^{4} a_k x_{i-k} + \omega_i$
WL	$\mathrm{T}_{wl} = \sum_{i=1}^{N-1} \lvert x_{i+1} - x_i \rvert$
ZC	$\mathrm{T}_{zc} = \sum_{i=1}^{N-1} \mathrm{sgn}(-x_i x_{i+1})$
SSC	$(x_i - x_{i-1}) \times (x_i - x_{i+1}) \geq \omega$, where $\omega = 0.05std$

Multilabel: By decomposing these compound motions into their corresponding modes in gesture, wrist angle, and strength, the problem could be transferred to the multilabel, multiclass classification. Thus, the output dimension was 12 (Figure 4E). For multilabel, the activation adopted was Sigmoid.

Machine learning

As mentioned in the dataset segment ahead, there were only 400 samples for each motion. The scale of the dataset was far from large. Thus, the traditional machine learning methods were also adopted in this work.

Feature

Considering the short window length (150 ms), several commonly used features in the time domain were selected, such as mean absolute value (MAV), root means square (RMS), variance (VAR), 4th-autoregressive coefficient (ARC), wavelength (WL), zero crossings (ZC), and slope signal change (SSC) (Englehart and Hudgins, 2003; Zhao et al., 2020). Their mathematical definitions are listed in Table 2, where $x_i(i = 1, 2, ..., N)$ is the EMG time series, N equals 150 according to the window length, a_k is the autoregressive coefficient, and is the white noise. Different feature vectors can be formed through the permutations of these three features from eight sEMG channels. Combining these three features, a 56-dimension feature vector can be extracted in maximum, as shown in Figure 5A.

Algorithm

Various types of classifiers were adopted to evaluate the decoding performance, including tree, discriminant, support vector machine (SVM), K-nearest neighbor (KNN), and some ensemble methods. The details of these 24 algorithms are listed in Table 3, where $G(x_i, x_j)$.denotes elements in the gram matrix, x_i, x_j denote the observations, and γ is the width of the Gaussian kernel.

Chain

Similar to the label encoding way in deep learning, the structure of machine learning classifiers can also be designed in one integrated or three separated ways (Figures 5B,C). There was a single classifier decoding 60 categories for the integrated structure. The separate, triple-parallel structure was set, and each took charge of strength, wrist angle, or gesture separately.

Compound motion decoding

Throughout the methods above, the best was selected under the comparison with average accuracy, prediction speed, training speed, and so on. Then, with the best decoding method, datasets from S1-S12 were all adopted to provide a detailed

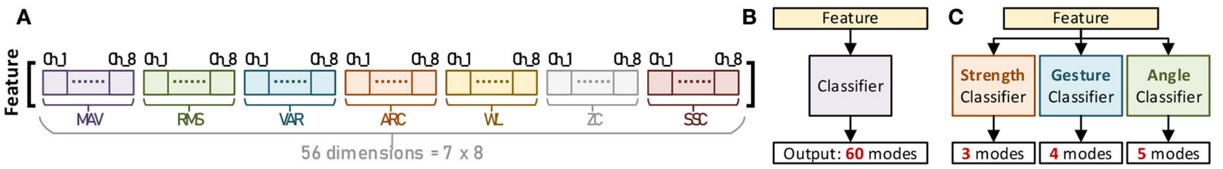


FIGURE 5
The feature vector and the classifier chain of machine learning methods. **(A)** Feature vector. **(B)** Integrated classifier. **(C)** Separated classifier.

TABLE 3 The specific number ID of each compound motion.

Algorithm	Subdivided	Attribute		Notes	
Decision trees	Fine	Maximum leaf	100	Split criterion	Gini Index
	Medium	Maximum leaf	20		
	Coarse	Maximum leaf	4		
Discriminant	Linear	\		\	
	Quadratic	\		\	
Naïve bayes	Gaussian	Distribution	Normal		
	Kernel	Type	Gaussian	Width	Automatic
SVM	Linear	Kernel	$G(x_i, x_j) = {x_i}' x_j$		
	Quadratic		$G(x_i, x_j) = (1 + {x_i}' x_j)^2$		
	Cubic		$G(x_i, x_j) = (1 + {x_i}' x_j)^3$		
	Fine gaussian		$G(x_i, x_j) = \exp(-\lVert x_i - x_j \rVert^2 / \gamma\ )$	Scale γ	0.56
	Medium gaussian		$G(x_i, x_j) = \exp(-\lVert x_i - x_j \rVert^2 / \gamma\ )$	Scaleγ	2.2
	Coarse gaussian		$G(x_i, x_j) = \exp(-\lVert x_i - x_j \rVert^2 / \gamma\ )$	Scaleγ	8.9
KNN	Fine	Distance	$d_{st}^2 = (x_i - x_j)(x_i - x_j)'$	Neighbors	1
	Medium		$d_{st}^2 = (x_i - x_j)(x_i - x_j)'$	Neighbors	10
	Coarse		$d_{st}^2 = (x_i - x_j)(x_i - x_j)'$	Neighbors	100
	Cosine		$d_{st} = (1 - x_i {x_j}' / \sqrt{(x_i x'_i)(x_j x'_j)}\ )$	Neighbors	10
	Cubic		$d_{st} = \sqrt[3]{\sum_{u=1}^{n} \lvert x_{iu} - x_{ju} \rvert^3}$	Neighbors	10
	Weighted		$d_{st}^2 = (x_i - x_j) V^{-1} (x_i - x_j)'$	Neighbors	10
Ensemble	Boosted trees	\		\	
	Bagged trees	\		\	
	Subspace discriminant	\		\	
	Subspace KNN	\		\	
	RUSBoosted trees	\		\	

decoding performance analysis among all these 60 compound motions. Similarly, according to the chronological order, the first samples of each mode were selected as a training set, and the rest were test-set. Within each dataset, the samples were shuffled.

To focus more on the decoding performance among all 60 compound motions, the analysis included (1) the change of test accuracy with the decrease of the training set; (2) the test accuracies of each motion and their confusion matrix; (3) and the test accuracy for separate motions of strength levels, gestures, and wrist angles.

Results

Decoding methods comparison

Deep learning

Through comparisons, considering the accuracy, stability, overfitting, and training time, the CNN+LSTM-based structure with one-hot label achieved the best performance at 94.61 ± 3.20% for training and 94.20 ± 4.06% for validation. The details are given below.

Structure

The TensorFlow (Abadi et al., 2016) 2.0 framework was adopted as the supporting backend for deep learning realization. The optimizer and batch size remained the same for CNN-based, LSTM-based, and CNN+LSTM-based structures (Table 4). At first, a one-hot label connected with SoftMax activation was adopted. The training and validation curves are illustrated in Figure 6.

The subject-average accuracy curve in Figure 6 indicates that the CNN+LSTM-based structure outstands these three structures. The final subject-averaged training and validation accuracies for CNN+LSTM-based are 94.61 ± 3.20% and 94.20 ± 4.06%. Especially for subject S4, the validation accuracy approaches 99.77% for 60 modes with CNN+LSTM. Structure (a) with only one convolution block showed an ideal training curve but poor validation. An extensive validation decay in the post-training period emerged from S3, S5, and S6 (9.36, 2.19, and 40.36%); for S1, S2, and S4, the validation accuracy ended at 78.72, 87.19, and 99.47%. Such huge individual differences indicate the instability of structure (a) in this compound motion decoding. For bidirectional LSTM-based structure (b), the highest validation accuracy was 61.38% by S4, while the rest of the subjects remained below 10%. Under the same configuration environment, with Win 10, i5-6500 (3.20 GHz), and GTX 960, the average time cost for training 70 epochs was 278.91s for CNN-based, 1,491.74s for LSTM-based, and 528.94s for CNN+LSTM based.

For the CNN-based network, with the increase of convolution blocks and the deletion of the dense layer (4,096 nodes), the final decay in Figure 6 has been greatly relieved, as shown in Figure 7. Compared with only one convolution block in Figure 6, the validation performances are largely improved, ending at nearly 90%. However, more blocks led to higher overfitting, with the gaps between training and validation being 8% for two blocks and 10% for three blocks. The early decline in the validation curve indicated that great overfitting occurred in the early stage due to insufficient training. The validation accuracy is improved with the adjustment of convolution blocks but is within the CNN+LSTM. The average training time cost for two blocks was 76.90 s and 56.54 s for three blocks. The training was accelerated by reducing trained parameters as the block increased.

TABLE 4 Compile setting in structure comparison.

Type	Method	Parameter	Value
Optimizer	RMSprop	Learning rate	0.0008
		Clip value	1.0
		Decay	1e-8
Training	Batch	Batch size	1,024
	Epoch		70

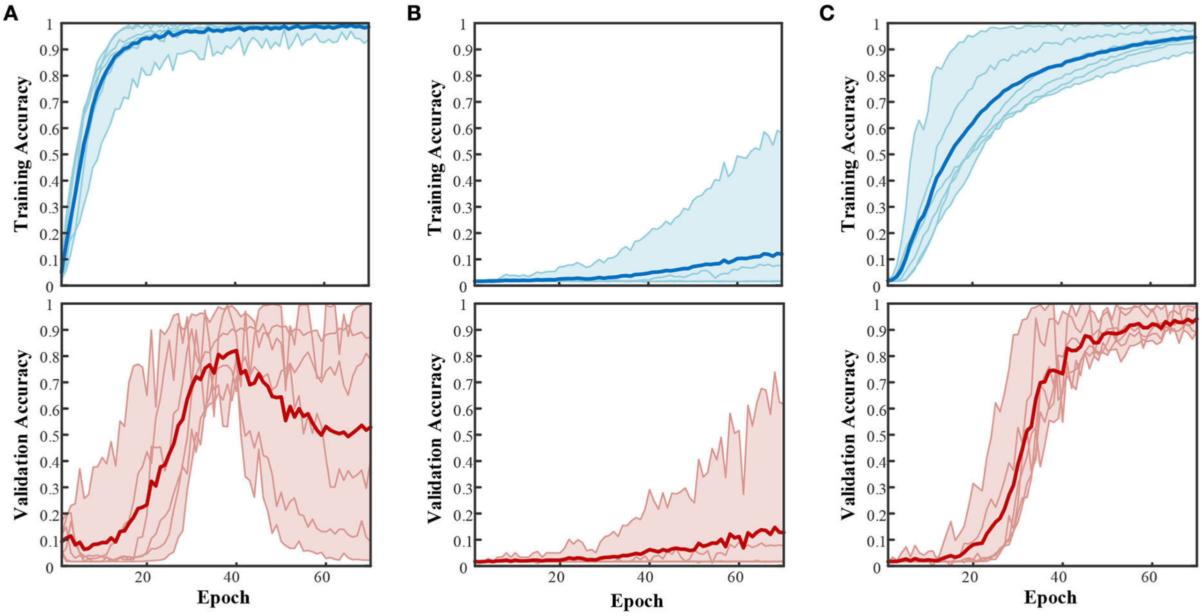


FIGURE 6
Accuracy curves during training and validation with three structures. **(A)** CNN-based structure with a one-hot label. **(B)** LSTM-based structure with a one-hot label. **(C)** CNN+LSTM-based structure with a one-hot label.

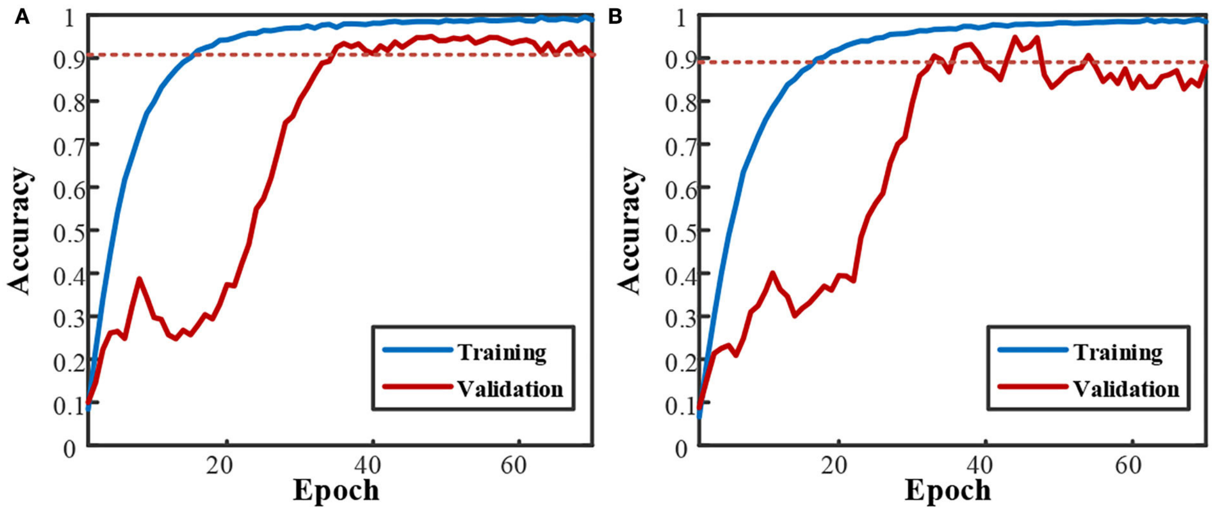


FIGURE 7
Subject-averaged training and validation performance of CNN based networks with increased convolution block. **(A)** 2 CNN blocks. **(B)** 3 CNN blocks.

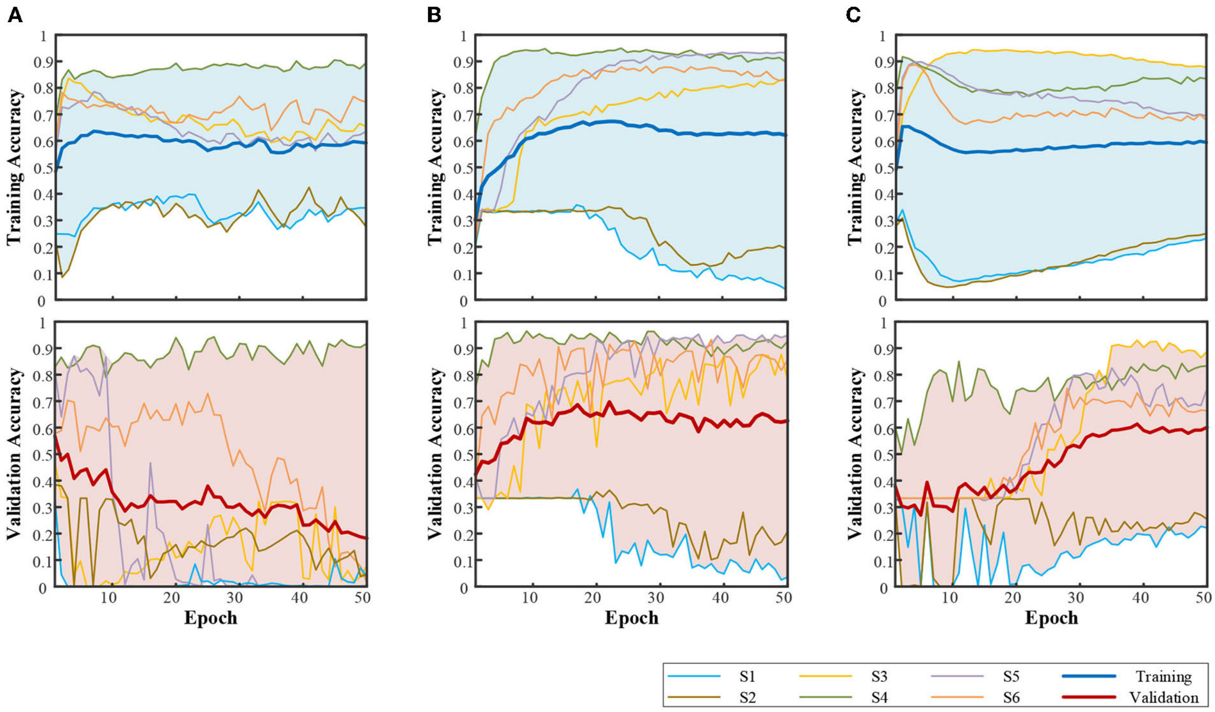


FIGURE 8
Performance with multilabel. **(A)** CNN-based structure. **(B)** LSTM-based structure. **(C)** CNN+LSTM-based structure.

Label encoding

The influence of multilabel is shown in Figure 8. The performance of the one-hot label is already shown in Figure 6. The compilation was kept the same as in Table 4. Compared with the one-hot label, the filling part in Figure 8 shows a larger individual difference, led by the multilabel. For bidirectional LSTM, the performance of S3–S6 was significantly improved with multilabel, while there was no help for S1 and S2. It illustrates that multilabel classification of compound motion resulted in greater instability.

TABLE 5 Comparison of machine learning classifiers.

Model type		Accuracy* (%)								Training time* (sec)	Prediction Speed* (obs/sec)	Accuracy* (%)		
		S1	S2	S3	S4	S5	S6	AVE	Std Dev			Strength	Gesture	Angle
Tree	Fine	63.9	63.9	63.4	99.9	63.7	99.0	75.63	16.84	\	\	96.50	93.05	85.38
	Medium	27.4	22.9	30.0	35.0	29.7	35.0	30.00	4.23	\	\	94.20	78.30	66.87
	Coarse	6.6	6.5	8.2	8.3	7.0	8.3	7.48	0.80	\	\	88.72	57.68	47.73
Linear discriminant	90.7	90.3	91.9	99.5	96.7	99.7	94.80	3.98	11.3	228.3	92.98	77.95	72.02	
Quadratic discriminant	94.2	94.5	95.2	\	99.3	99.9	\	\	\	\	92.57	89.75	87.73	
Naive bayes	Gaussian	90.2	91.0	87.1	99.9	96.9	99.6	94.12	4.93	\	\	83.45	56.32	47.50
	Kernel	92.1	93.8	92.3	99.6	97.9	99.6	95.88	3.25	619.3	429.3	91.65	79.93	\
SVM	Linear	96.0	96.1	96.4	100.0	98.8	99.8	97.85	1.73	467.0	696.7	94.28	87.72	81.70
	Quadratic	96.5	96.5	97.3	100.0	99.2	99.9	98.23	1.51	443.8	586.7	98.35	98.57	97.35
	Cubic	96.2	96.2	97.1	100.0	99.1	99.9	98.08	1.64	452.0	485.0	\	\	\
	Fine Gaussian	\	\	\	95.6	\	\	\	\	\	\	98.62	98.13	97.27
	Medium Gaussian	95.4	95.8	96.0	99.9	98.9	99.9	97.65	1.95	538.8	421.7	98.48	98.17	96.00
	Coarse Gaussian	93.5	93.9	94.0	100.0	97.9	99.7	96.50	2.78	539.2	418.3	94.18	84.52	74.28
KNN	Fine	91.0	90.2	94.5	100.0	97.4	99.7	95.47	3.89	153.4	730.0	98.63	98.33	97.30
	Medium	91.9	91.0	94.5	99.9	97.2	99.7	95.70	3.51	158.8	768.3	98.68	98.38	97.38
	Coarse	88.0	87.2	90.5	99.4	94.1	99.3	93.08	4.94	\	\	97.70	97.05	95.60
	Cosine	91.0	89.6	93.2	100.0	96.8	99.6	95.03	4.04	\	\	98.52	98.02	97.03
	Cubic	91.0	89.9	\	99.9	\	\	\	\	\	\	\	\	\
	Weighted	92.4	91.7	94.8	100.0	97.5	99.7	96.02	3.29	196.7	935.0	98.80	98.52	97.65
Ensemble	Boosted Trees	65.2	62.4	59.0	99.9	73.8	97.3	76.27	16.43	\	\	73.52	89.52	77.18
	Bagged Trees	94.7	94.8	95.6	100.0	98.3	99.7	97.18	2.23	225.7	661.7	98.98	98.55	97.57
	Subspace Discriminant	88.0	87.8	90.2	99.4	95.1	99.6	93.35	4.97	\	\	91.62	71.85	65.47
	Subspace KNN	89.3	88.3	93.4	99.9	97.4	99.5	94.63	4.64	\	\	\	97.70	\
	RUSBoosted Trees	26.2	21.3	30.0	35.0	29.7	35.0	29.53	4.81	\	\	71.97	78.32	66.87

*Evaluated on the Win10, AMD R7 (integrated graphics). The ‘ín accuracy indicated that the training time for this classifier exceeded 1,000 s. The ‘ín training time and prediction speed indicated that the performance of this classifier did not meet expectations, so no analysis was done.

Machine learning

For this relatively small sample multiclass sEMG classification, combined with feature engineering, several machine learning classifiers show great applicability and excellent performance. Through comparison, the quadratic SVM for 60 modes once achieved the best ($p = 0.04 < 0.05$ with T-Test). The details are as given below.

Algorithm

The averaged validation accuracy of 60 modes from various machine learning classifiers is listed in Table 5, along with their training times and prediction speeds. In Table 5, among all 24 classifiers, the average accuracies of 16 classifiers exceed 90%, and ten classifiers exceed 95%. Such generally high performance demonstrates the applicability of traditional feature extraction and machine learning methods in the small sample multiclass sEMG decoding. Among all classifiers, the SVM with a quadratic kernel achieved the highest subject-averaged accuracy at 98.23 ± 1.51%. Its 1.5% standard deviation indicates a small individual difference among subjects and stable overall performance. In terms of time spent, the average time cost for quadratic SVM training is 443.8 s without acceleration.

Chain

The subject-averaged validation accuracies using the triple-parallel classifier chain for decoding the gestures, the wrist angles, and the strength levels are presented in Table 6. The highest accuracies were achieved by subspace discriminant for

TABLE 6 Averaged validation accuracies for triple-paralleled classifier chain.

Model type		Accuracy (%)		
		Strength level	Gesture	Wrist angle
Tree	Fine	96.50	93.05	85.38
	Medium	94.20	78.30	66.87
	Coarse	88.72	57.68	47.73
Linear discriminant		92.98	77.95	72.02
Quadratic discriminant		92.57	89.75	87.73
Naive bayes	Gaussian	83.45	56.32	47.50
	Kernel	91.65	79.93	\
SVM	Linear	94.28	87.72	81.70
	Quadratic	98.35	98.57	97.35
	Cubic	\	\	\
	Fine Gaussian	98.62	98.13	97.27
	Medium Gaussian	98.48	98.17	96.00
	Coarse Gaussian	94.18	84.52	74.28
KNN	Fine	98.63	98.33	97.30
	Medium	98.68	98.38	97.38
	Coarse	97.70	97.05	95.60
	Cosine	98.52	98.02	97.03
	Cubic	\	\	\
	Weighted	98.80	98.52	97.65
Ensemble	Boosted Trees	73.52	89.52	77.18
	Bagged Trees	98.98	98.55	97.57
	Subspace Discriminant	91.62	71.85	65.47
	Subspace KNN	\	97.70	\
	RUSBoosted Trees	71.97	78.32	66.87

three strength levels (98.98 ± 1.51%); quadratic SVM for four gestures (98.57 ± 1.15%); and weighted KNN for five wrist angles (97.65 ± 1.72%). By combining these three classifiers to form the triple-paralleled chain, we can see that the theoretical decoding accuracy was the product of three accuracies, which equals 92.93, 90.79, 92.47, 100.00, 96.05, and 99.60% for S1∼S6. The average subject accuracy for the classifier chain was 95.30 ± 3.54%. The classifier chain did not show superiority compared with the one integrated classifier.

Feature

In total, seven kinds of temporal features were adopted. Figure 9 illustrates the change in validation accuracy as the number of features increases. As the number of features increases, the average validation accuracy and the standard deviation decrease from 1.68 to 0.63%. The performance remained above 99.00% in the range of 5–7 features. Among them, the combination of all seven features was the best.

Compound motion decoding

In the overall comparison, the SVM with a quadratic kernel performed the best after 443.8 s of training. Feature engineering was the combination of all seven features. Figure 10 depicts the variation in test accuracy resulting from altering the proportion of the training set to the testing set for such a decoding method. In each mode, the latter data were used for testing.

As the training proportion decreased from 90 to 50%, the average test accuracy remained higher than 95%, which showed great generalization. However, the standard deviation increased from 1.71 to 2.96%, and the individual differences became prominent. Six of the twelve subjects kept their test accuracy higher than 95% in all the processes, while S3 and S8 gradually dropped to approach 90%. S3 performed the worst (from 94.29 to 89.98%). S8 decreased the most (from 99.21 to 90.07%). S6 achieved the most stable performance (with an average accuracy of 99.97 ± 0.01% from 90% training to 50%). Figure 11A shows the test accuracy of each mode with a 90% training set.

The average test accuracy with a 90% training set was 98.42 ± 1.71%. Among all subjects, S4 and S6 achieved the best at 99.96 ± 0.32%. Four of these twelve subjects maintained their accuracies for 60 modes to be all greater than 95%, and most were equal to 100%. Figure 11B is the confusion matrix.

Within 60 modes in Figure 11B, eleven modes achieved 100% test accuracy for all twelve subjects. Thirty-one modes are higher than 99%. Fifty-eight modes are higher than 95%. Motion 38 [palm, ulnar deviation, 480g], and motion 57 [palm, extension, 960g] performed lower than 95%, with motion 38 the worst at 85.2%. Table 7 lists the test accuracy for separate motions, which achieved 99.35 ± 0.67% for three classes of strength-level decoding, 99.34 ± 0.88% for four classes of gesture decoding, and 99.04 ± 1.16% for five classes of wrist-angle decoding. Meanwhile, since the addition of wrist angles ensures decoding stability in various postures, the average test accuracy of "strength + gesture" is 98.95 ± 1.11%.

Discussion

This study stated the need for compound motion decoding in myoelectric control and further investigated and realized the classification of 60 compound motions with 150 ms sEMG collected from eight forearm muscles. Different methods of deep learning and machine learning were adopted to assess their capability. In deep learning, three structures and two ways of label encoding were analyzed. Among them, the CNN+LSTM with a one-hot label performed the best. In machine learning, 24 classifiers, different combination of features, and classifier chain were tested. The quadratic SVM combined with seven features showed the highest validation accuracy and the smallest variance. Compared with deep learning, classifiers from machine learning showed more stability and robustness. Overall, the

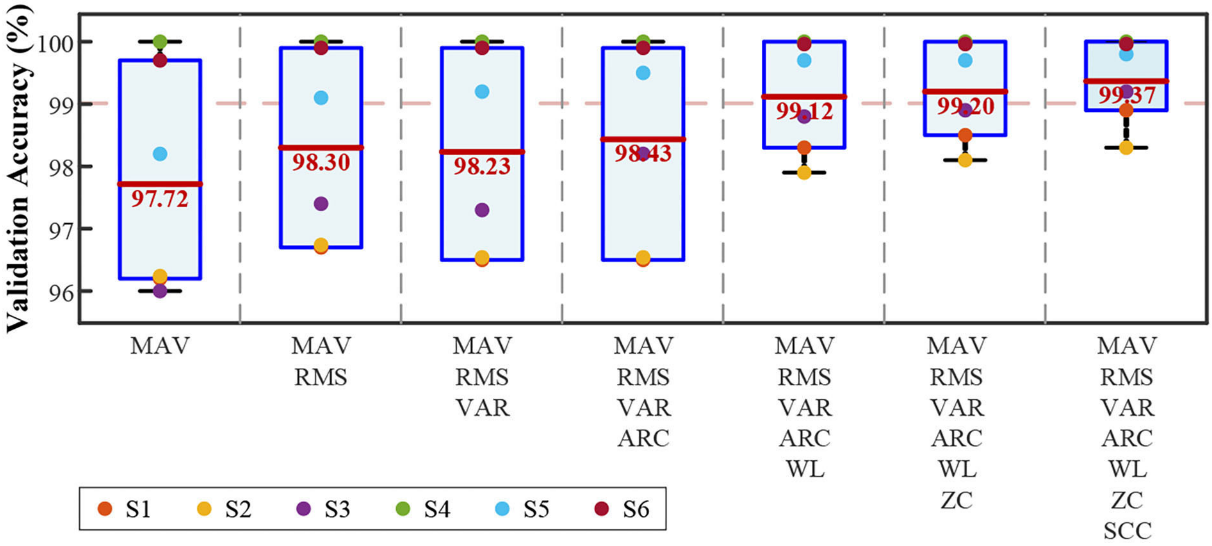


FIGURE 9
Validation accuracies of quadratic SVM with an increased number of features.

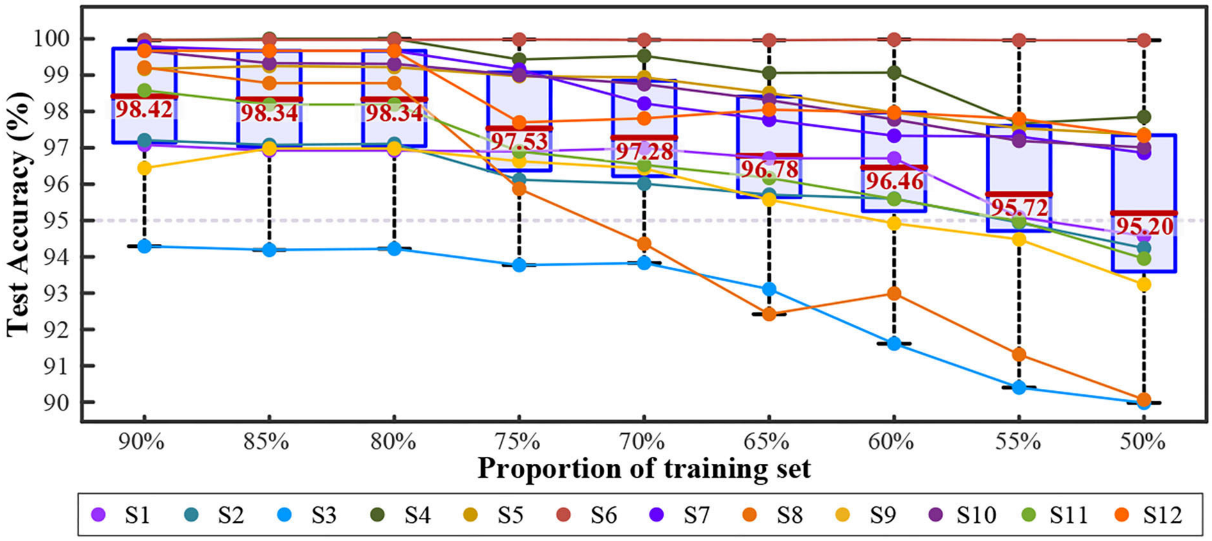


FIGURE 10
The change of validation accuracies with quadratic SVM combining seven features.

quadratic SVM exceeded the CNN + LSTM with higher validation accuracy, lower training time, and less variance. This result demonstrated the ability of traditional machine learning on relatively small sample sEMG multi-classification problems.

The significance and the performance

Considering the ultimate goal to be a more flexible control of the myoelectric hand by incorporating the blind grasp, this study proposes the need for sEMG-based compound motion decoding, paying particular attention to the need for control purpose switching (such as grip vs. crush), the differentiation between power grasp and precision grasp, and the manipulation of the wrist joint. With the 60 compound motions in this work (as the product of four gestures, five wrist angles, and three strength levels), we shall not only guarantee the flexibility of control ability but also guarantee the stability of sEMG-based decoding under a variant upper limb posture.

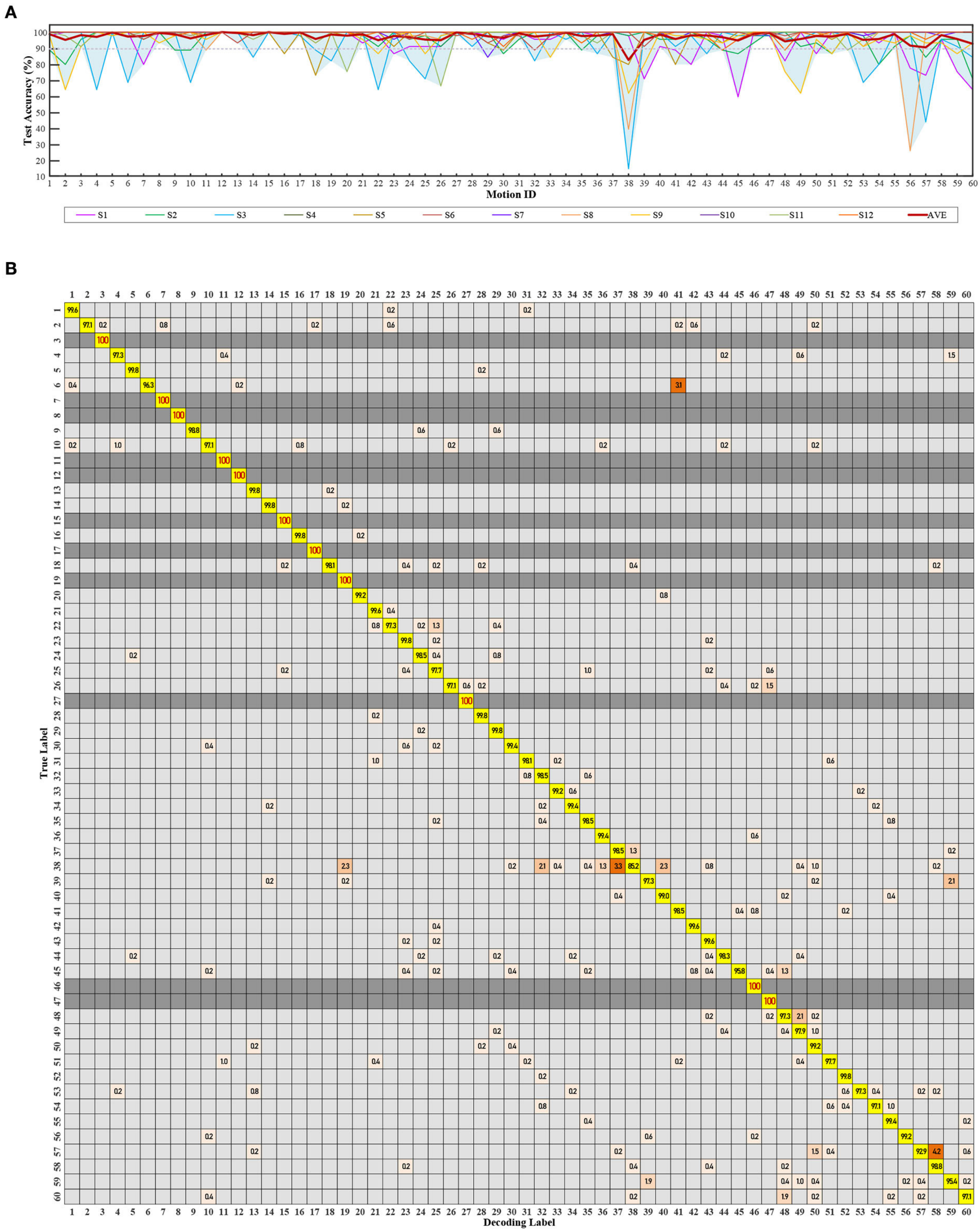


FIGURE 11
Testing performance of 60 compound motions with quadratic SVM combining seven features. The label 1~60 match Table 1. **(A)** The validation and test accuracy. **(B)** Subject average confusion matrix of 60 compound motions.

TABLE 7 The test accuracy for separate motions of strength levels, gestures, and wrist angles with quadratic SVM combining seven features.

Subject	Test accuracy (%)			
	Strength	Gesture	Angle	Strength+Gesture
S1	98.54	98.71	98.08	97.63
S2	98.25	99.04	99.13	97.54
S3	98.71	97.00	95.83	96.92
S4	100.00	100.00	99.96	100.00
S5	100.00	99.92	99.17	99.92
S6	100.00	100.00	99.96	100.00
S7	99.96	99.79	99.96	99.79
S8	99.54	99.96	99.25	99.50
S9	98.58	98.38	98.13	97.79
S10	99.96	99.75	99.96	99.71
S11	98.79	99.58	99.33	98.75
S12	99.83	99.96	99.79	99.83
AVE	99.347	99.340	99.045	98.948
Std	0.674	0.880	1.159	1.109

In comparing deep learning methods and machine learning classifiers, facing the same ultimate goal as flexible control, besides the accuracy, the training time, the prediction speed, and the dependence on computing hardware, all matter. Combined with manual feature engineering, most classifiers in Table 5 showed appreciating results with small variance among subjects. While among the three structures in the deep learning method, only CNN + LSTM steadily converged (Figure 6). This result indicates that the information contained in the short-windowed original sEMG is sparse and chaotic. The capability of simply designed CNN or LSTM in auto feature extraction is limited in this multi-classification with only small samples. This affirms the value of traditional manual feature engineering in small-sample multiclass sEMG decoding.

With the help of this manual feature engineering, the training process of machine learning classifiers was speedy. For most classifiers, without the acceleration of GPU, the training process can still be kept for approximately 5 mins. Conversely, the shorter time for deep learning is consistent with the longer time in machine learning.

As for stability, since the SVM has a high generalization, although the accuracy decreases with the training proportion, the overall performance was still acceptable. For the balance among 60 modes, considering the individual differences, the standard deviation varied from 0.32% (S4, S6) to 14.74% (S3). Two subjects showed excellent decoding performance, with the standard variance close to 0.00%.

In conclusion, with the addition of feature extraction, the machine learning approaches in this very small sample multiclass sEMG compound motion decoding stand out for their excellent accuracy, fast training procedure, low computation cost, and stability.

The limitation and future work

Experiment protocol

In the materials, limited by the size of the electrode patches adopted in this study, targeted placement was adopted instead of equally spaced. However, several studies have reported that the equally spaced placement achieved better performance for the machine learning method. In future research, tinier patches will be used to compare the performance under different electrode placements. Meanwhile, it has been noticed that the motions executed in sequence might increase the inter-class difference and decrease the intra-class difference. This may lead to a seemingly appropriate decoding performance. In future work, we are considering further reducing the data collection work of each motion and improving the data collection scheme to be decentralized and disordered.

Decoding performance

In the subsequent research work, feature engineering with the quadratic SVM resulted in regretful test accuracies for the contralateral decoding and cross-subjects. This demonstrated that manual feature engineering has distinct personal characteristics and that transferring the trained network to other people or extra objects is difficult. However, for deep learning, several papers reported the transfer learning ability in cross-subject sEMG decoding. In 2021, Chen constructed a CNN-based general gesture EMG feature extraction network of 30 hand gestures, then transferred it into the decoding of extra gestures, which improved the recognition accuracy by 10 and 38% (Chen et al., 2021). Jiang proposed a correlation-based data weighting method that achieved a low root mean square error in cross-subject evaluation with significant performance improvement (Hautier et al., 2000). Based on CNN, Yu proposed a transfer learning strategy for instantaneous gesture recognition that improved the average accuracy of new subjects and new gesture recognition by 18.7 and 8.74% (Yu et al., 2021). In 2017, Cote-Allard used the CNN-based transfer learning techniques to leverage inter-user data from the first dataset and alleviate the data generation burden imposed on a single individual (Cote-Allard et al., 2017). The above research makes us believe that, with the help of transfer learning, deep learning is more suitable for cross-subject and cross-object research. However, manual feature engineering and machine learning still have a place in subject-specific decoding with small samples and large categories. Further, the paper lacks online validation on amputees as an initial work. When verifying the feasibility of compound motion decoding under a 150 ms window length, some degradation of decoding accuracy may occur when applied to online decoding. Meanwhile, the study on decoding methods, the study of blind-grasp, and the research on improving stability and reducing noise interference are equally important for amputees' successful online task operation.

Improvements in compounded motions

As for the selection of compound motions, considering the repeatability and controllability during the experiment, loads with counterweights are used to activate the strength levels. However, such specified strength levels can hardly be reproduced in the realistic online control of the myoelectric hand for the disabled. Since the separability of three levels of strengths in 60 compound motions has been demonstrated, future research will emphasize the practicality by replacing strength levels with three different loads (stuck to the hand) with one's maximum strength, medium strength, and weak strength, thereby realizing the switch of control purposes during the online control of the myoelectric hand. Meanwhile, with the more complex design of the myoelectric hand's control purpose, the number of strength levels would be increased according to the demand of the control purposes. Next, on the premise of stability, more gestures would be included to enlarge the instruction sets. The four gestures now selected in this work were all functional gestures, and none of them was sign language. In the following research, according to the proposed control logic, besides the functional gestures, more sign languages are planned to be added to enrich the communicational usage of the myoelectric hand. Moreover, based on this work, transfer learning is planned to be studied next for the adaptation of more users and more complex and personalized decoding sets.

Conclusion

Considering the control purpose switching (such as grip vs. crush), the distinction between power grasp and precision grasp, and the manipulation of the wrist joint in the control of the myoelectric hand, this work puts forward the need for compound motion decoding. With 150 ms sEMG from eight muscles, decoding 60 upper limb compound motions achieved an average accuracy of 98.42 ± 1.71%. These 60 motions were the product of four gestures, five wrist angles, and three strength levels. Among all 60 motions, 48 showed a test accuracy greater than 95%, and one part was equal to 100%. In comparing decoding performance, several deep learning methods and machine learning classifiers were adopted, with the contrast among structures, label encoding ways, and algorithms. The feature engineering (MAV+RMS+VAR+ARC+WL+ZC+SSC) combined with the SVM (quadratic kernel) stood out for its high accuracy, short training process, less computation cost, and well stability ($p < 0.05$). The comparison results highlighted the value of manual feature engineering and machine learning classifiers in relatively small sample multiclass sEMG decoding. As a prerequisite work for myoelectric control, this study provides a flexible solution for the subsequent involvement of blind grasping and control purposes, aiming to provide a more stable, diversified, and convenient operation for the myoelectric hand.

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 experiment in this study was approved by the Ethics Committee (Institutional Review Board) of Xi'an Jiaotong University (No. 20211452). The patients/participants provided their written informed consent to participate in this study.

Author contributions

XZ supervised this work and revised the manuscript. ZL proposed, did the research, and wrote the manuscript. CF edited the manuscript and financially supported the publication. YW, TZ, and HL organized and carried out the experiments. QT revised the manuscript. All authors contributed to the article and approved the submitted version.

Funding

This work was supported in part by the Key Research and Development Program of Shaanxi (No. 2020SF-148), the Science and Technology Supporting Xinjiang Project (No. 2020E0259), and partially supported by grants from the National Key Research and Development Program of China (No. 2017YFB1300303).

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abadi, M., Barham, P., Chen, J. M., Chen, Z. F., Davis, A., Dean, J., et al. (2016). *TensorFlow: A System for Large-Scale Machine Learning*. Berkeley: Usenix Assoc.

Adewuyi, A. A., Hargrove, L. J., and Kuiken, T. A. (2016). An analysis of intrinsic and extrinsic hand muscle EMG for improved pattern recognition control. *IEEE Transac. Neural Syst. Rehab. Eng.* 24, 485–494. doi: 10.1109/TNSRE.2015.2424371

Ameri, A., Akhaee, M. A., Scheme, E., and Englehart, K. (2019). Regression convolutional neural network for improved simultaneous EMG control. *J. Neural Eng.* 16, 11. doi: 10.1088/1741-2552/ab0e2e

Association, W. M. (2013). World medical association declaration of helsinki: ethical principles for medical research involving human subjects. *JAMA* 310, 2191–2194. doi: 10.1001/jama.2013.281053

Bengio, Y., and Lecun, Y. (1997). Convolutional networks for images, speech, and time-series. *Handb. Brain Theory Neural Netw.* 3361, 1995.

Boschmann, A., Nofen, B., and Platzner, M. (2013). "Improving transient state myoelectric signal recognition in hand movement classification using gyroscopes," in *35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. p. 6035–6038. doi: 10.1109/EMBC.2013.6610928

Chen, X., Li, Y., Hu, R., Zhang, X., and Chen, X. (2021). Hand gesture recognition based on surface electromyography using convolutional neural network with transfer learning method. *IEEE J. Biomed. Health Inform.* 25, 1292–1304. doi: 10.1109/JBHI.2020.3009383

Chu, J., Moon, I., Lee, Y., Kim, S., and Mun, M. (2007). A supervised feature-projection-based real-time EMG pattern recognition for multifunction myoelectric hand control. *IEEE/ASME Trans. Mechatr.* 12, 282–290. doi: 10.1109/TMECH.2007.897262

Cote-Allard, U., Fall, C. L., Campeau-Lecours, A., Gosselin, C., Laviolette, F., and Gosselin, B. (2017). "Transfer learning for sEMG hand gestures recognition using convolutional neural networks," in *Ieee International Conference on Systems, Man, and Cybernetics*, (New York, NY: IEEE). p. 1663–1668. doi: 10.1109/SMC.2017.8122854

Dang, H., Weisz, J., and Allen, P. K. (2011). "Blind Grasping: Stable Robotic Grasping Using Tactile Feedback and Hand Kinematics," in *2011 IEEE International Conference on Robotics and Automation*. doi: 10.1109/ICRA.2011.5979679

De Luca, C. J. (1997). The use of surface electromyography in biomechanics. *J. Appl. Biomech.* 13, 135–163. doi: 10.1123/jab.13.2.135

Englehart, K., and Hudgins, B. (2003). A robust, real-time control scheme for multifunction myoelectric control. *IEEE Trans. Biomed. Eng.* 50, 848–854. doi: 10.1109/TBME.2003.813539

Feix, T., Romero, J., Schmiedmayer, H. B., Dollar, A. M., and Kragic, D. (2016). The GRASP taxonomy of human grasp types. *IEEE Trans. Human-Mach. Syst.* 46, 66–77. doi: 10.1109/THMS.2015.2470657

Hashim, N. A., Bin Abd Razak, N. A., Gholizadeh, H., and Abu Osman, N. A. (2017). Analysis of voluntary opening Ottobock Hook and Hosmer Hook for upper limb prosthetics: a preliminary study. *Biomed. Eng. Biomed. Technik* 62, 447–454. doi: 10.1515/bmt-2016-0130

Hautier, C. A., Arsac, L. M., Deghdegh, K., Souquet, J., Belli, A., and Lacour, J. R. (2000). Influence of fatigue on EMG/force ratio and cocontraction in cycling. *Med. Sci. Sports Exer.* 32, 839–843. doi: 10.1097/00005768-200004000-00017

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hu, Y., Wong, Y. K., Wei, W. T., Du, Y., Kankanhalli, M., and Geng, W. D. (2018). A novel attention-based hybrid CNN-RN N architecture for sEMG-based gesture recognition. *PLoS ONE* 13, 18. doi: 10.1371/journal.pone.0206049

Khokhar, Z. O., Xiao, Z. G., and Menon, C. (2010). Surface EMG pattern recognition for real-time control of a wrist exoskeleton. *Biomed. Eng. Online* 9, 17. doi: 10.1186/1475-925X-9-41

Kiguchi, K., and Hayashi, Y. (2012). An EMG-based control for an upper-limb power-assist exoskeleton robot. *IEEE Trans. Syst. Man Cybern. Part B-Cybern.* 42, 1064–1071. doi: 10.1109/TSMCB.2012.2185843

Lauer, R. T., Peckham, P. H., Kilgore, K. L., and Heetderks, W. J. (2000). Applications of cortical signals to neuroprosthetic control: A critical review. *IEEE Trans. Rehab. Eng.* 8, 205–208. doi: 10.1109/86.847817

Lu, Z. Y., Chen, X., Li, Q., Zhang, X., and Zhou, P. (2014). A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices. *IEEE Trans. Human-Mach. Syst.* 44, 293–299. doi: 10.1109/THMS.2014.2302794

Mayer, R. M., Garcia-Rosas, R., Mohammadi, A., Tan, Y., Alici, G., Choong, P., et al. (2020). Tactile feedback in closed-loop control of myoelectric hand grasping: conveying information of multiple sensors simultaneously via a single feedback channel. *Front. Neurosci.* 14, 12. doi: 10.3389/fnins.2020.00348

McIntosh, J., Mcneill, C., Fraser, M., Kerber, F., Loechtefeld, M., and Krueger, A. (2016). "EMPress: practical hand gesture classification with wrist-mounted EMG and pressure sensing," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems,* (ACM). p. 2332–2342. doi: 10.1145/2858036.2858093

Mereu, F., Leone, F., Gentile, C., Cordella, F., Gruppioni, E., and Zollo, L. (2021). Control strategies and performance assessment of upper-limb TMR prostheses: a review. *Sensors.* 21, 1953. doi: 10.3390/s21061953

Rehman, M. Z. U., Waris, A., Gilani, S. O., Jochumsen, M., Niazi, I. K., Jamil, M., et al. (2018). Multiday EMG-based classification of hand motions with deep learning techniques. *Sensors* 18, 16. doi: 10.3390/s18082497

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature* 323, 533. doi: 10.1038/323533a0

Shaw-Cortez, W., Oetomo, D., Manzie, C., and Choong, P. (2019). Robust object manipulation for tactile-based blind grasping. *Control Eng. Prac.* 92, 104136. doi: 10.1016/j.conengprac.2019.104136

Standardization, C. N. I. O. (1988). "Human dimensions of Chinese adults," in *China National Technical Supervision Bureau.*

van der Niet, O., Bongers, R. M., and Van Der Sluis, C. K. (2013). Functionality of i-LIMB and i-LIMB Pulse hands: Case report. *J. Rehab. Res. Develop.* 50, 1123–1128. doi: 10.1682/JRRD.2012.08.0140

Vredenbregt, J., and Rau, G. (1973). Surface electromyography in relation to force, muscle length and endurance. *Clin. Biomech.* 24, 607–622. doi: 10.1159/000394062

Wirta, R. W., Taylor, D. R., and Finley, F. R. (1978). Pattern-recognition arm prosthesis: a historical perspective-a final report. *Bull. Prost. Res.* 8–35.

Xiong, C. H., Chen, W. R., Sun, B. Y., Liu, M. J., Yue, S. G., and Chen, W. B. (2016). Design and implementation of an anthropomorphic hand for replicating human grasping functions. *IEEE Trans. Robot.* 32, 652–671. doi: 10.1109/TRO.2016.2558193

Yu, Z., Zhao, J., Wang, Y., He, L., and Wang, S. (2021). Surface EMG-based instantaneous hand gesture recognition using convolutional neural network with the transfer learning method. *Sensors* 21, 2540. doi: 10.3390/s21072540

Zhai, X. L., Jelfs, B., Chan, R. H. M., and Tin, C. (2017). Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network. *Front. Neurosci.* 11, 11. doi: 10.3389/fnins.2017.00379

Zhang, C., Chen, X., Cao, S., Zhang, X., and Chen, X. (2019). A Novel HD-sEMG preprocessing method integrating muscle activation heterogeneity analysis and kurtosis-guided filtering for high-accuracy joint force estimation. *IEEE Transac. Neural Syst. Rehab. Eng.* 27, 1920–1930. doi: 10.1109/TNSRE.2019.2933811

Zhang, Q., Zheng, C. F., and Xiong, C. H. (2015). "EMG-based estimation of shoulder and elbow joint angles for intuitive myoelectric control," in *2015 Ieee International Conference on Cyber Technology in Automation, Control, and Intelligent Systems,* (New York, NY: IEEE). p. 1912–1916. doi: 10.1109/CYBER.2015.7288239

Zhang, X., Chen, X., Li, Y., Lantz, V., Wang, K. Q., and Yang, J. H. (2011). A framework for hand gesture recognition based on accelerometer and EMG sensors. *IEEE Trans. Syst. Man Cybern. Part a-Syst. Hum.* 41, 1064–1076. doi: 10.1109/TSMCA.2011.2116004

Zhao, S., Wu, X., Zhang, X., Li, B., Mao, J., and Xu, J. (2020). Automatic gesture recognition with surface electromyography signal. *J. Xi'an Jiaotong Univ.* 54, 149–156. doi: 10.7652/xjtuxb202009017

TYPE Original Research
PUBLISHED 18 November 2022
DOI 10.3389/fnbot.2022.1007324

Check for updates

OPEN ACCESS

EDITED BY
Zhan Li,
Swansea University, United Kingdom

REVIEWED BY
Lale Dulger,
İzmir University of Economics, Turkey
Manolo Garabini,
University of Pisa, Italy

*CORRESPONDENCE
Bingshan Hu
hubingshan@usst.edu.cn

RECEIVED 30 July 2022
ACCEPTED 28 October 2022
PUBLISHED 18 November 2022

CITATION
Hu B, Mao B, Lu S and Yu H (2022) Design and torque control base on neural network PID of a variable stiffness joint for rehabilitation robot. *Front. Neurorobot.* 16:1007324.
doi: 10.3389/fnbot.2022.1007324

Design and torque control base on neural network PID of a variable stiffness joint for rehabilitation robot

Bingshan Hu[1,2]*, Binghao Mao[1], Sheng Lu[1] and Hongliu Yu[1,2]

[1]Institute of Rehabilitation Engineering and Technology, University of Shanghai for Science and Technology, Shanghai, China, [2]Shanghai Engineering Research Center of Assistive Devices, Shanghai, China

Variable stiffness joints have been gradually applied in rehabilitation robots because of their intrinsic compliance and greater ability to adjust mechanical stiffness. This paper designs a variable stiffness joint for upper limb rehabilitation training. The joint adopts the variable stiffness principle based special curved surface. The trapezoidal lead screw in the variable stiffness module has a self-locking function, and the stiffness can be maintained without the continuous output torque of the motor. In the aspect of control, back propagation (BP) neural network PID control strategy is used to control the torque of variable stiffness joint. Experiments show that this control method can effectively improve the torque control performance of variable stiffness joints in the case of low stiffness, and the isotonic centripetal resistance training can be realized by using the joints and control methods designed in this paper.

KEYWORDS

mechatronics, variable stiffness actuators, rehabilitation robotics, neural network PID, torque control

Introduction

Based on medical theory, the upper limb rehabilitation robot drives the affected limb to carry out scientific and effective training (Xu et al., 2011), so that the patient's motor function can be recovered better (Lin et al., 2003). According to the process of patient rehabilitation, the rehabilitation training stage of upper limb rehabilitation robot includes passive training (Lizheng et al., 2015), robot-assisted training (Chang and Kim, 2013) and resistance training (Song et al., 2014). During passive rehabilitation training, the joints of rehabilitation robot are required to have high stiffness to ensure the stability and high bandwidth of closed-loop position control, so as to drive the limbs of patients to reach the specified position accurately (Gopura et al., 2016). In the stage of assisted and resistance training, the rehabilitation robot must have good flexibility and exert different forces on the patient's limbs to ensure the safety and comfort of the rehabilitation process (Marchal-Crespo and Reinkensmeyer, 2009). It can be seen that at different stages of rehabilitation training, the driving joints of the rehabilitation robot need to have different stiffness (Ma et al., 2016). The traditional robot usually collects a large number of position, torque, speed and other data by increasing the type and number of sensors on the basis of rigid driving joints (Palazzolo et al., 2019), and then designs a controller that can process these data effectively, so as to achieve the effect of controlling the impedance of the rehabilitation robot (Yuan et al., 2013). This requires that the sensors, driving and control circuits of the rehabilitation robot run fast enough, and the system needs to establish an accurate dynamic model (Cestari et al., 2014). For example, the biped robot designed by Hubicki et al. (2018) uses inertial measurement units, encoders and other sensors to collect large amounts of data, and then processes these data through algorithms to achieve stable control effects. The driving joint of the traditional rehabilitation robot is not inherently compliant and cannot store energy, which leads to its inability to absorb the energy of the instantaneous impact (Caldwell et al., 2015). In recent years, some scholars have developed a variety of rehabilitation robots with variable stiffness joints instead of rigid joints. Vanderborgh et al. summarized the variable impedance actuators, and classified it into active impedance by control, inherent compliance and damping actuators, inertial actuators, etc. based on the principle of variable stiffness and impedance (Vanderborght et al., 2013). Yi et al. designed a variable stiffness joint of exoskeleton (Yi et al., 2018). Yang et al. used a variable stiffness rehabilitation robot to perform elbow rehabilitation training for stroke patients (Yang et al., 2021). Baser and Kizilhan designed a wearable ankle exoskeleton with variable stiffness (Baser and Kizilhan, 2018).

According to the working principle, the working principle of variable stiffness actuator (VSA) can be divided into variable lever arm, special curved surface, changing the number of elastic elements and so on. The way of changing lever arm is to change the transmission ratio between load and spring according to the proportion of lever arm. It is mainly composed of load point, pivot and spring contact point. Changing any position can adjust the stiffness of the mechanism. For example, Chaichaowarat designed a variable stiffness spring mechanism, which is composed of a slider, a roller and an adjustable unsupported length leaf spring. The adjustment of the slider position changes the spring contact point, thus obtaining the variable stiffness characteristics (Chaichaowarat et al., 2021). Similar variable stiffness actuators include AWAS-I (Jafari et al., 2010), AWAS-II (Jafari et al., 2014), COMPACT-VSA (Tsagarakis et al., 2011), etc. In the variable stiffness principle of special curved surface, the variable stiffness mechanism is connected in series between the reducer and the output shaft of the joint. The stiffness control motor changes the relative position of the cam disc to control the pretension force of the spring to adjust the stiffness of the joint. For example, the FSJ joint developed by Wolf, when the joint is subjected to passive torque load or the spring preload is changed, the cam disc rotates relatively, the spring is compressed, and the stiffness of the joint changes (Wolf et al., 2016). Similar special surface variable stiffness actuators include MESTRAN (Hung Vu et al., 2011), VSM (Sun et al., 2018) and SJM-II (Park and Song, 2010). Some variable stiffness joints adjust their stiffness by changing the number of elastic elements. In the discrete variable stiffness actuator designed by Hussain, the spring set is connected in series between the drive motor and the load end. The springs are controlled by the clutch, respectively. The on-off of the clutch can be realized by changing the number of spring connections (Hussain et al., 2021). In addition, there are other methods. For example, Garabini et al. designed a soft robots that mimic the neuromusculoskeletal system, which reproduces many of the characteristics of an agonistic-antagonistic muscular pair acting on a joint (Garabini et al., 2017). For the variable lever arm type variable stiffness joint, the torque curve depends on the length of the lever. If the length of the lever is increased, the volume of the mechanism will increase accordingly. For the variable number of elastic elements type, the volume of the mechanism will also increase due to the number of elastic elements. The ideal asymptote torque curve can be obtained only by changing the contour of the special surface, which is easier to achieve miniaturization in terms of volume and weight. Therefore, the principle of variable stiffness of the special curved surface will be used in this paper.

In terms of control, variable stiffness joints operate in a large stiffness range. In the case of low stiffness, vibration is often accompanied, and its torque response performance is also affected (Albu-Schaffer et al., 2010), which reduces the safety of human-machine interaction in rehabilitation training. To solve such problems, Liu investigated a closed-loop torque controlled variable stiffness actuator (VSA) combined with a disturbance observer, and a better dynamic response with high and low stiffness was achieved (Liu et al., 2021). Albu-Schaffer proposed a general variable stiffness joint model for nonlinear control design, and then designed a simple gain scheduling state feedback controller for active vibration reduction of weak damping joints (Albu-Schaffer et al., 2010). Misgeld designed a gain scheduling torque controller to improve the human-machine interaction characteristics of the joint. The eigenvalue of the gain scheduling is determined by the zeros and poles of the multi-channel $H\infty$-control strategy, which can perform gain-scheduled control on multiple stiffness values of the joint in discrete time (Misgeld et al., 2017). In the interactive control of the manipulator, the gradient of the gain scheduled variable hyperplane is adjusted according to the real-time identified environmental stiffness to achieve stable position and torque control in different environments (Iwasaki et al., 2002). Mengacci et al. designed an iterative learning control scheme based on torque by decoupling the motion/stiffness of the articulated soft robot and learning the expected action of the robot, and accurately controlled the position trajectory of the articulated soft robot without changing the flexibility of the articulated soft robot. In this paper, the PID control scheme based on BP neural network is adopted to improve the torque response speed of variable stiffness actuator under different stiffness conditions (Mengacci et al., 2020).

In this paper, based on the existing research, a variable stiffness joint used in upper limb rehabilitation training is designed to improve the safety and comfort of the rehabilitation process. Compared with the previous work, the contributions of this paper can be summarized as follows.

(1) From the mechanical design point of view, the joint adopts the variable stiffness principle based special curved surface. The trapezoidal lead screw in the variable stiffness module has a self-locking function, and the stiffness can be maintained without the continuous output torque of the motor.

(2) From the control algorithm point of view, the PID control based on BP neural network is adopted to improve the torque control response performance of the elbow joint rehabilitation robot driven by variable stiffness joints under low stiffness by adjusting the gain parameters under different stiffness.

(3) Based on the principle of variable stiffness, this paper designs a variable stiffness joint for the elbow joint, and combines the PID control method based on BP neural network to build the experimental platform of the variable stiffness elbow joint rehabilitation robot, and the isotonic centripetal resistance training experiment of elbow joint is carried out to verify the effect of BP neural network PID torque control.

The rest of this paper is organized as follows: the second section introduces the mechanical design of the variable stiffness joint based on principle of special curved surface. The third section establishes the dynamic model of the joint, studies the PID control method based on BP neural network, and conducts simulation verification. The fourth section builds an experimental platform for a variable stiffness elbow joint rehabilitation robot, and takes the elbow joint isotonic centripetal resistance training as an example to verify the effect of BP neural network PID torque control. The fifth section is conclusions and discussions.

Mechanical design of variable stiffness joint

Mechanical structure design

As shown in Figure 1, the overall size of the variable stiffness joint designed in this paper is 500*110*137.5mm, and the total mass is 1.5 kg. According to the structure and function, the mechanism can be divided into main drive module and variable stiffness module. The main drive module provides the output torque, and the variable stiffness module adjusts the joint output stiffness.

The main drive module includes a main motor and a harmonic reducer. The main motor is a brushless DC motor with a rated torque of 269 mNm, and the reduction ratio of the harmonic reducer is 100:1. Figure 2 is the schematic diagram of the variable stiffness module mechanism. The stiffness adjustment motor drives a pair of gears to rotate, the gears drive the trapezoidal lead screw to rotate, the lead screw nut moves forward, the spring is compressed, and then the cam rotates to adjust the joint stiffness.

The specific variable stiffness module is shown in Figure 3A. The left is the input flange connecting the main drive module, and the right is the output flange connecting the load. The input flange and output flange are supported by cross roller bearings. The driving source of the variable stiffness module is a brushless DC motor with a rated torque of 10.8 mNm. The output of the motor is connected with a planetary reducer. The stiffness adjustment motor is installed on the input flange. The output of the reducer drives the gear 1 to rotate, and one end of the gear 2 is meshed with the gear 1, the other end of the gear 2 is connected with the trapezoidal lead screw. The trapezoidal screw rotates to push the screw nut forward. Due to the introduction of trapezoidal lead screw, the variable stiffness module has self-locking function. So, the stiffness adjustment motor does not need to output torque while maintaining the compression of spring to reduce energy consumption. The slide block 1 installed on the right side of the lead screw nut moves forward and compresses the upper end of the spring. The lower end of the spring is connected with the sliding block 2, and a cam follower is installed on the sliding block 2. The cam follower is tangent to the cam contour in the output flange and transmits the spring force to the output flange.

In addition, as shown in Figure 3B, a rotary encoder is installed on the input flange, and the magnetic steel of the encoder is installed in the gear 3. An internal gear groove is designed in the output flange to mesh with gear 3. When there is relative rotation between the input flange and the output flange, the gear 3 rotates, and the encoder collects the rotation degree of the magnetic steel to obtain the relative rotation angle between the two flanges, that is, the deformation angle.

Analysis of variable stiffness characteristics of special curved surface

In the variable stiffness joint with special curved surface, the design of cam contour determines the stiffness characteristics of the compliant joint, which needs to be analyzed and designed according to the contour. The variable stiffness joint designed in this paper is intended to be applied to the elbow rehabilitation robot. According to the existing literature research, the stiffness of the human elbow joint varies from 0 to 20 Nm/rad, and the output torque is 10 Nm. Therefore, the

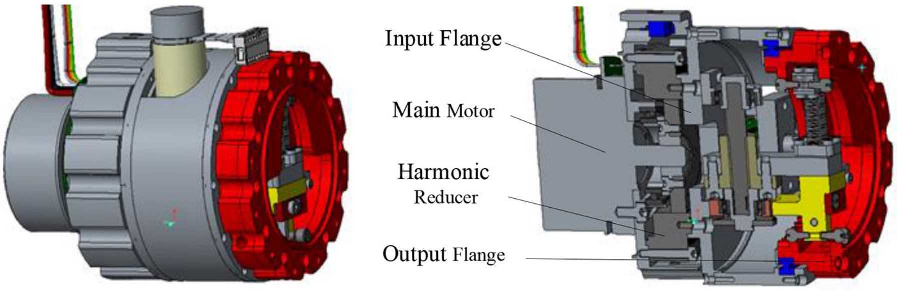


FIGURE 1
Variable stiffness joint model.

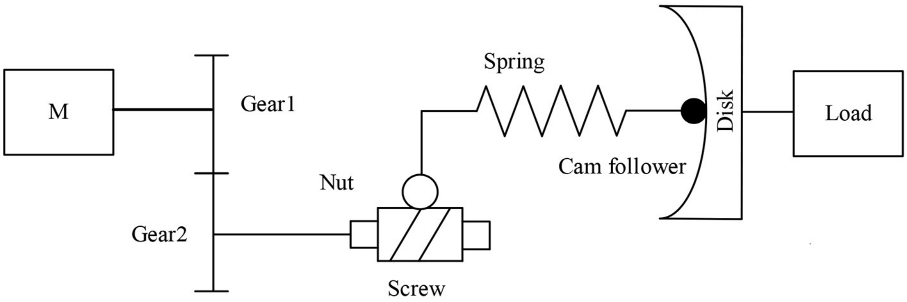


FIGURE 2
Schematic diagram of variable stiffness module mechanism.

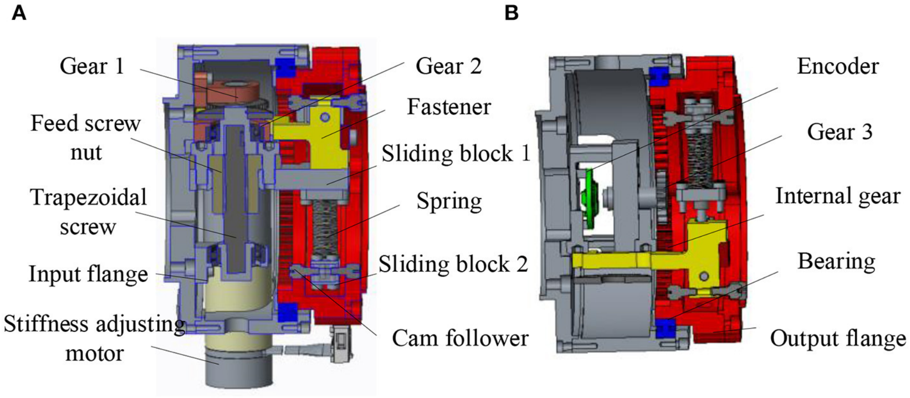


FIGURE 3
Three-dimensional model of variable stiffness joint **(A)** Sectional view 1 of variable stiffness module **(B)** Sectional view 2 of variable stiffness module.

variable stiffness joint designed in this paper should meet the above index requirements.

Here, for the convenience of analysis, the output flange in Figure 3A is fixed, and the cam follower is idealized as a point. As shown in Figure 4, the cam contour is designed as an ellipse, the center of the cam disc is the o origin, the long axis of the cam contour is in the positive direction of the y axis, and the short axis is in the positive direction of the x axis. The xoy coordinate system is established.

Assuming that the length of the short axis of the ellipse is a and the length of the long axis of the ellipse is b, combined with Figure 4, the general equation of the elliptic

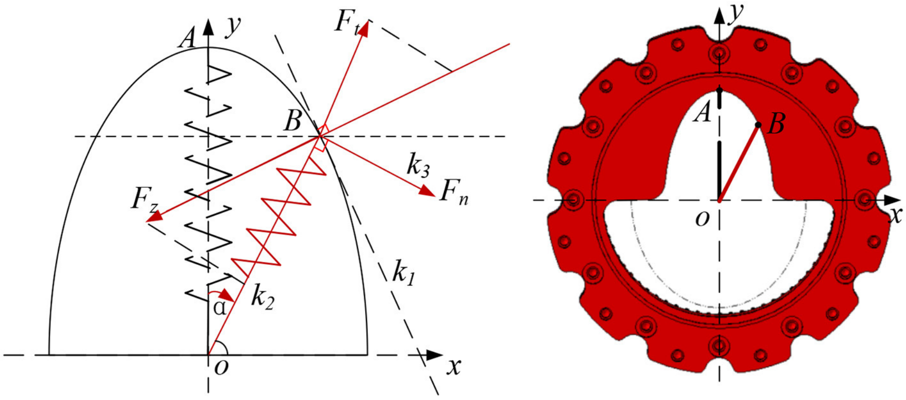


FIGURE 4
Theoretical analysis of special surface.

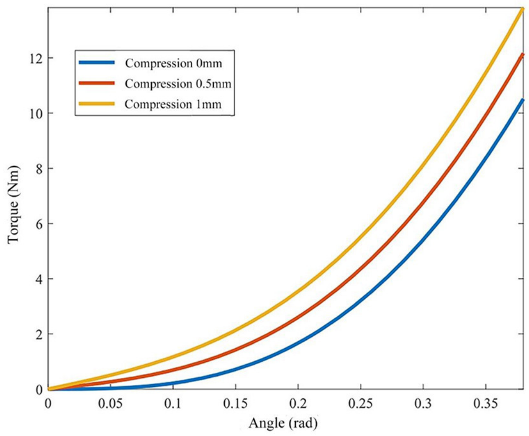


FIGURE 5
Relationship between joint torque and deformation angle.

curve is parameterized:

$$\begin{cases} x_0 = a \cdot \sin(\alpha) \\ y_0 = b \cdot \sin(\alpha) \end{cases} \tag{1}$$

According to the previous section, the input flange rotates α rad, spring OA also rotates α rad and reach to OB position. At this time, the tangent slope k_1 of the ellipse at point B and the length c of OB are:

$$k_1 = \frac{y_0}{x_0} \tag{2}$$

$$c = \overrightarrow{OB} = \sqrt{{x_0}^2 + {y_0}^2} \tag{3}$$

At point B, due to the compression of the spring, the elastic force F_t is generated, then F_t is:

$$F_t = 2^* kk^* \left(b - c + l_0\right) \tag{4}$$

In equation (4), l0 is the initial compression of the spring, and kk is the spring stiffness coefficient.

The slope k2 of Ft can be obtained:

$$k_2 = -\frac{b^{2*}x_0}{a^{2*}y_0} \tag{5}$$

According to equations (2) and (5), the slope k_3 of the tangential force F_n of the output flange can be obtained as:

$$k_3 = \left|\frac{k_1 - k_2}{1 + k_1^* k_2}\right| \tag{6}$$

According to equations (2)–(6), the functional relationship between the output torque T_n, deformation angle of the variable stiffness joint α and the initial compression of the spring l_0 can be obtained.

$$T_n = f\left(\alpha, l_0\right) = \frac{2{*}kk{*}\left(b - \sqrt{{x_0}^2 + {y_0}^2} + l_0\right)}{\left|\frac{k_1 - k_2}{1 + k_1 * k_2}\right|} * \sqrt{{x_0}^2 + {y_0}^2}{*}10^{-3} \tag{7}$$

In equation (7), x_0 and y_0 are functions of α, and it can be measured by the encoder.

The expression of static stiffness of variable stiffness joint is:

$$K_s = \frac{dT_n}{d\alpha} \tag{8}$$

According to equation (7), the cam profile on the output flange determines the nonlinear relationship between the output torque T_n, the deformation angle of the variable stiffness module α and the initial spring compression l_0, which determines the stiffness characteristics of the joint. The initial compression l_0 of the spring can be changed by the screw actuated by the stiffness adjustment motor to adjust the joint stiffness. Take the spring

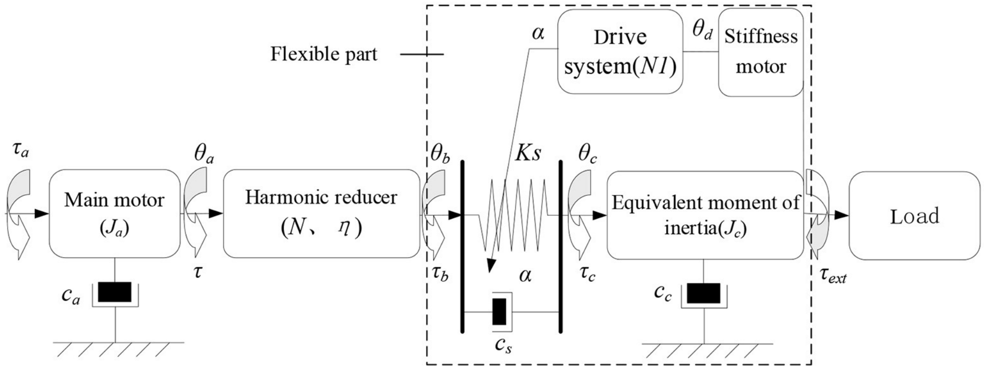


FIGURE 6
Dynamic model of elbow joint with variable stiffness.

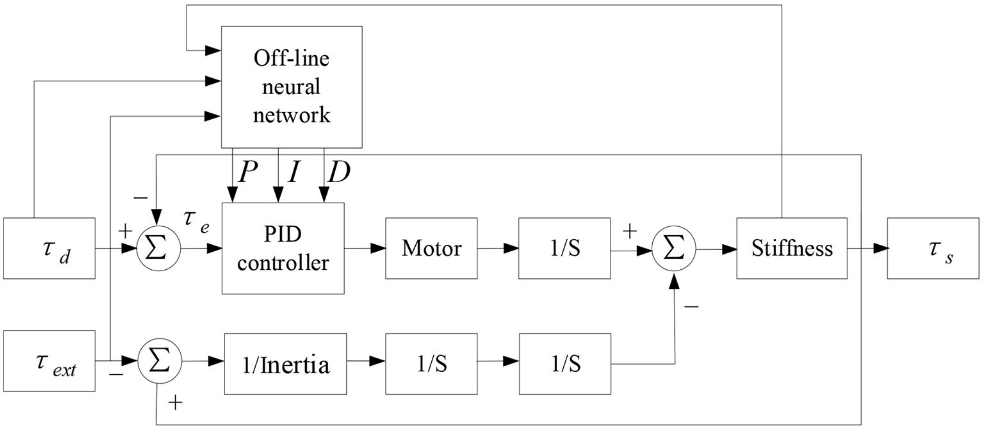


FIGURE 7
PID control model based on BP neural network.

stiffness coefficient $kk = 88$ N/mm, the ellipse long axis a as 60 mm, and the short axis b as 30 mm. According to the above theoretical equation, the relationship curve between the output torque T_n, the deformation angle α and the spring compression l_0(0, 0.5, and 1 mm) is shown in Figure 5.

It can be seen from Figure 5 that the joint output torque curve shows an upward trend as a whole. With the increase of deformation angle, the joint output torque curve shows an upward trend. The rising slope of the first half is small, the joint stiffness is low and the range of stiffness change is small, so it can be used for resistance training. The second half of the output torque curve has a large rising slope, high joint stiffness and a wide range of stiffness changes, which is suitable for passive training. The greater the initial compression of the spring l_0, the greater the joint stiffness at the same joint deformation angle α. Finally, it is found that the range of joint deformation angle is 0–0.4 rad, the corresponding output torque range is 0–14 Nm, and the stiffness variation range is 0–35 Nm/rad, which meets the requirements of human elbow joint stiffness and torque output, ant it can used in elbow rehabilitation.

Dynamic modeling and control

Dynamic model

In this section, a variable stiffness joint dynamic model is established to derive the joint transfer function. As shown in Figure 6, the variable stiffness joint is composed of motor, reducer, elastic element, transmission system and load. The main motor provides the power source, which is output to the flexible link through the harmonic reducer. The torque is output from the end of the flexible link and connected with the external load.

The torque of main motor τ_a satisfy the following relation:

$$\tau_a = J_a\ddot{\theta}_a + C_a\dot{\theta}_a + \tau \tag{9}$$

In equation (9), J_a and C_a are the inertia and damping coefficient of the main motor, τ is the output torque of the main motor, θ_a is the output angle of the main motor. Torque and position at the output end of harmonic reducer are τ_b and θ_b. τ_a and θ_a satisfy the following relation,

$$\begin{cases} \theta_a = \frac{\theta_b}{N} \\ \tau_b = \frac{\tau_a}{N_\eta} \end{cases} \tag{10}$$

In equation (10), N is the harmonic reduction ratio, η is the transmission efficiency. In this joint mechanism, the flexible link only plays the role of transmitting torque, so the torque at both ends of the flexible link has the following relationship:

$$\begin{aligned} \tau_a \; &= \tau_c = J_c\ddot{\theta}_c + C_c\dot{\theta}_c + \tau_{ext} = K_s\left(\theta_b - \theta_c\right) \\ &+ C_s\left(\theta_b - \theta_c\right) \end{aligned} \tag{11}$$

In equation (11), $\tau_c$is the output torque of the flexible link, J_c is the equivalent inertia of the transmission system, C_c is the damping coefficient of the transmission system, C_s is the damping coefficient of the flexible element, θ_c is the rotation angle of the output end, τ_{ext} is the external load torque. Finally, the dynamic model of variable stiffness joint can be obtained as:

$$\begin{cases} T_a = J_a\ddot{\theta}_a + C_a\dot{\theta}_a + \frac{\tau_b}{N_\eta} \\ \tau_b = \tau_c = K_s\left(\frac{\theta_a}{N} - \theta_c\right) + C_s\left(\frac{\dot{\theta}_a}{N} - \dot{\theta}_c\right) \\ \tau_c = J_c\ddot{\theta}_c + C_c\dot{\theta}_c + \tau_{ext} \end{cases} \tag{12}$$

Then the variable stiffness joint transfer function is:

$$G\left(s\right) = \frac{\tau_c\left(s\right)}{\tau_a\left(s\right)} = \frac{J_c s^2\theta_c\left(s\right) + C_c s\theta_c\left(s\right) + \tau_{ext}\left(s\right)}{J_c s^2\theta_c\left(s\right) + C_a s\theta_a\left(s\right) + \frac{\tau_c(s)}{N_\eta}} \tag{13}$$

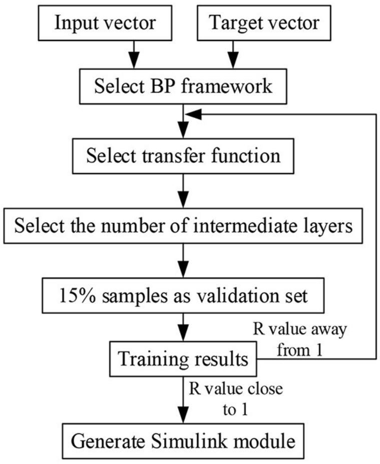


FIGURE 8
BP neural network construction process.

PID torque control based on bp neural network

Generally, the traditional PID control method is adopted for the joint torque control, and the control command is τ_d. τ_d, τ_a and τ_c has the relationship as following,

$$\tau_a = k_p\left(\tau_d - \tau_c\right) + k_i\int\left(\tau_d - \tau_c\right)dt + k_d\frac{d\left(\tau_d - \tau_c\right)}{dt} \tag{14}$$

In equation (14), k_p, k_i and k_d are proportional, integral and differential control parameters, respectively. At this time, the joint torque transfer function is:

$$G_1\left(s\right) = \frac{\left(k_d s^2 + k_p s + k_i\right)G\left(s\right)}{\left(k_d s^2 + k_p s + k_i\right)G\left(s\right) + s} \tag{15}$$

TABLE 1 Partial input vector and target vector parameters.

Input vector			**Target vector**		
K_s**(Nm/rad)**	τ_d**(Nm)**	τ_{ext}**(Nm)**	k_p	ki	k_d
3	10	5	2,862.667	42,101.83	6.594912
6	10	5	2,425.731	54,381.63	3.2941
9	10	5	3,103.1	133,517.8	2.352503
12	10	5	2,092.687	75,721.37	0.289952
15	10	5	1,648.609	52,694.09	1.162397
18	10	5	1,712.596	65,955.46	0.892257
21	10	5	1,489.212	54,583.03	0.701764
24	10	5	1,235.035	40,014.22	0.580256
27	10	5	1,134.435	36,022.09	0.477105
30	10	5	1,005.529	34,946.03	0.734407
35	10	5	980.4292	39,073.43	0.66269

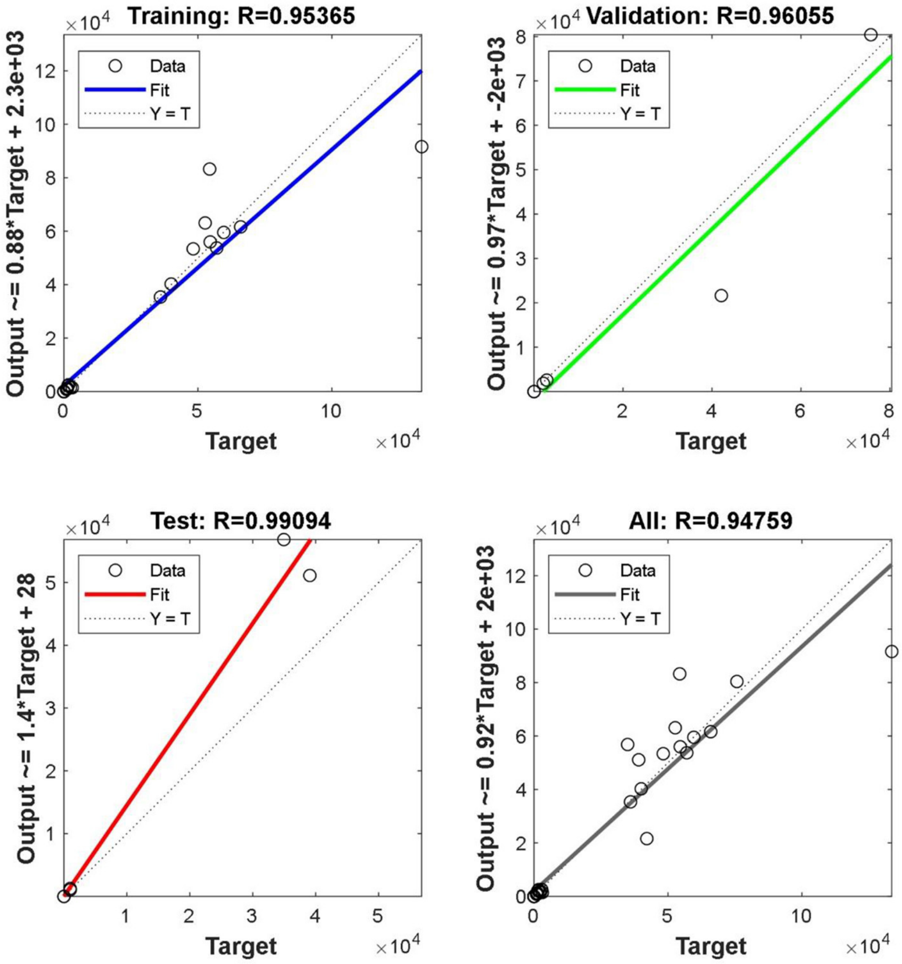


FIGURE 9
Correlation coefficient of each sample set.

From equations (12) and (13), it can be seen that the torque response of variable stiffness joint is related to joint stiffness K_s, external load τ_{ext}, torque command τ_d and PID gain parameters. K_s and τ_d are determined by the control command, and the external load τ_{ext} is determined by the human-machine interaction torque. When the traditional PID control method is adopted, the fixed PID gain parameters are difficult to meet the torque control performance under different stiffness conditions, especially in the case of low stiffness, improper parameter adjustment will produce oscillation, slow response, large overshoot, long system stability time and other phenomena, which will reduce the safety of in rehabilitation training. Therefore, a PID control strategy based on BP neural network is designed as shown in Figure 7.

BP off-line neural network can optimize the gain parameters of PID controller for three groups of variables: different joint stiffness K_s, external load τ_{ext} and driving torque τ_d, so as to improve the real-time, accuracy and stability of torque response of joints under low stiffness furtherly. The specific implementation process is shown in Figure 8. Select a limited sample set (K_s, τ_d, τ_{ext}) as the input vector, the K_s parameter covers the range of 3–40 Nm/rad, the τ_d parameter covers the range of 0–20 Nm, and the τ_{ext} parameter covers the range of 0–20 Nm/rad. In the case of finite set, the optimal k_p, k_i, k_d parameters are used as the objective vector. Total 600 groups of samples (Combination 1: τ_d and τ_{ext} remain unchanged, K_s changes, and the optimized k_p, k_i and k_d parameters are obtained by simulation. Combination 2: $K_s$and τ_d remain unchanged, τ_{ext}changes, and the optimized k_p, k_i and k_d parameters are obtained by simulation. Combination 3: K_s and τ_{ext} remain unchanged, τ_d changes, and the optimized k_p, k_i and k_d parameters are obtained by simulation). Optimized parameters are shown in Table 1. Due to space limitations, only some parameters are shown here. Because BP neural network has the characteristic of back propagation, which can improve the accuracy of target vector, this paper selects

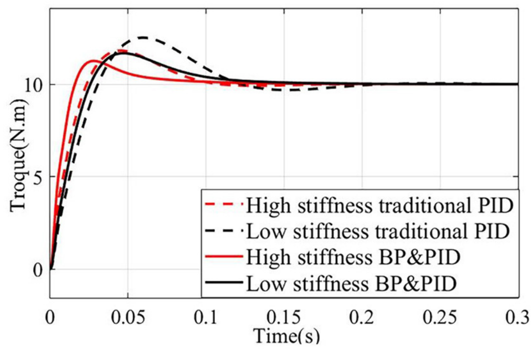


FIGURE 10
Joint torque response based on PID controller and Gain-scheduled control.

BP neural network framework, its hidden layer adopts tansig transfer function, and its output layer adopts purelin transfer function. The transfer equation between different layers has the following relationship:

$$\begin{cases} a_1 = tansig\left(IW^{1,1} \cdot p_1 + b_1\right) \\ a_2 = purelin\left(LW^{2,1} \cdot a_1 + b_2\right) \end{cases} \tag{16}$$

In equation (16), a_1 is the output of the first hidden layer, a_2 is the output of the first output layer, p_1 is the input vector of the first layer, $IW^{1,1}$ is the weight of the first hidden layer, b_1 is the threshold of the first hidden layer, $LW^{2,1}$ is the weight of the first output layer, b_2 is the threshold of the first output layer, 80% of the random samples are used as the training set, 10% of the samples are used as the verification set, and 10% of the samples are used as the test set for training. As shown in Figure 9, the abscissa represents the target output, and the ordinate represents the fitting function between the prediction output and the target output. In Figure 9, the regression r value of the training set is 0.90598, the regression r value of the verification set is 0.99356, the regression r value of the test set is 0.87803, and the overall R value is 0.91155. When the regression r value after training is close to 1, it indicates that the error is very small.

Finally, the PID torque control model based on BP neural network is built in Simulink for simulation analysis. Under the conditions of low stiffness (spring compression of 0 mm) and high stiffness (spring compression of 1 mm), the simulation results are compared with the traditional PID torque control, and the results are shown in Figure 10. It can be seen that the response of the joint based on BP neural network PID torque control is better than that of the traditional PID torque control in the case of low stiffness and high stiffness, especially in the case of low stiffness, which is more suitable for the resistance training of the joint in the case of low stiffness. Under the condition of high stiffness, the joint system based on BP neural network PID torque control reaches the maximum overshoot of 11.6 Nm after 0.03 s, which is 0.02 s faster than the traditional PID torque control and the overshoot is 0.6 Nm smaller. In the case of low stiffness, the joint system based on BP neural network PID torque control reaches the maximum overshoot of 11.8 Nm after 0.045 s, which is 0.02 s faster than the traditional PID torque control, and the overshoot is about 1 Nm smaller.

Experimental verification

Experimental platform

In the experiment, the simulation platform developed by Links company is used, as shown in Figure 11. On this platform, the static stiffness experiment is carried out firstly, and then the BP neural network PID torque control experiment is carried out to verify the joint torque control performance under the condition of low stiffness. Finally, the isotonic centripetal resistance rehabilitation training experiment is carried out.

The host-computer in the experimental platform runs RT-sim and MATLAB/Simulink software. RT-sim software compiles the Simulink model into C code that can be executed under VxWorks system, and the host-computer is equipped with operation and monitoring modules. The slave-computer is composed of VxWorks, servo motor driver. VxWorks runs the program compiled by RT-sim, and sends the command to the motor servo driver to control the variable stiffness joint. At the same time, the sensor signals in the variable stiffness joint are collected to monitor the joint state.

Static stiffness test

Firstly, the static torque of the variable stiffness joint is measured. When the joint is fixed, pull the rotating arm through the tension meter to rotate. The length of the rotating arm is 30 cm, and the joint deformation angle is measured by the rotary encoder. Under different initial spring compression, the joint rotates evenly between 0 and 0.35 rad by the same interval, and the measured tension value is recorded to obtain the variable stiffness joint torque. As the deformation angle increases, the joint torque shows an upward trend, and under the same torsion angle, the greater the spring compression, the greater the torque, and the greater the joint stiffness. Compared with the theoretical value in Figure 5, the experimental value in Figure 12 has a large error, which is mainly caused by the machining and assembly errors of the mechanism.

Torque control experiments

The torque control experiment was carried out under the conditions of low stiffness (spring compression of 0 mm) and high stiffness (spring compression of 1 mm). PID control and BP neural network based PID controller were used to carry out the step response experiment, and the step command was 3 Nm.

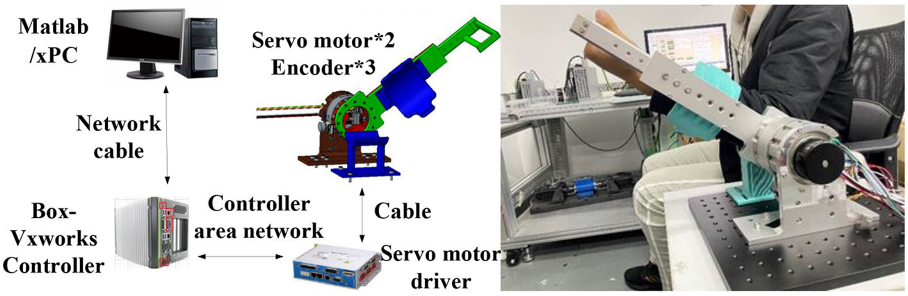


FIGURE 11
Principle of variable stiffness joint experimental platform.

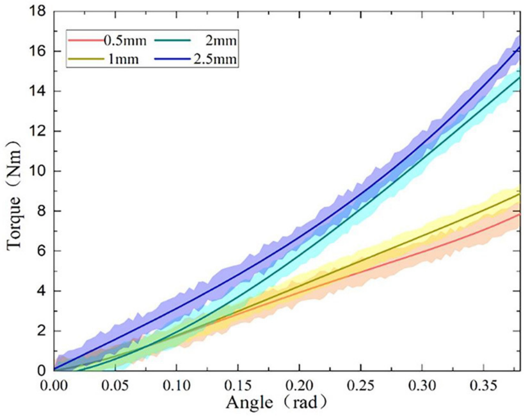


FIGURE 12
Torque-deformation angle curve.

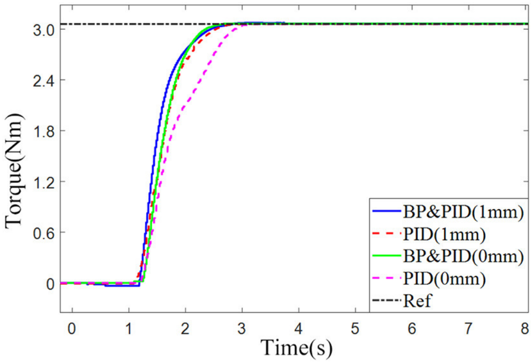


FIGURE 13
Comparison of joint torque response using different control methods.

As shown in Figure 13, in the low stiffness experimental group, the time for BP neural network PID torque control to reach the steady state is about 1.0 s, and the response time of traditional PID controller is about 1.5 s, an increase of 0.5 s. In the high stiffness experimental group, the time for the BP neural network PID torque control response to reach the steady state is about 1.2 s, the time for the traditional PID response to reach the steady state is about 1.4 s, and the response rate is increased by 0.2 S. It is found that the lower the stiffness is, the more significant the effect of BP neural network PID torque control on improving torque response is, which is more conducive to resistance training.

Isotonic centripetal resistance training experiment

Finally, the isometric resistance training experiment is carried out. The tester needs to overcome the resistance moment of the variable stiffness joint to complete the elbow flexion movement, and the extension movement is brought back to the initial position by the active torque provided by the variable stiffness mechanism, so as to complete the isometric centripetal resistance training by reciprocating motion. The experiment was divided into two groups. One group set the variable stiffness joint to provide a constant resistance torque of 8 Nm under the condition of high stiffness (spring compression 1 mm) and the other group under the condition of low stiffness (spring compression 0 mm).

Figure 14A shows the torque-time curve of isometric concentric resistance training. 0–4s is the preparation stage for the tester, and 4–20s is the active resistance training stage. The human-machine interaction torque gradually rises from 0 to 8 Nm within about 1s and maintains for a period of time. 20–30s is the relaxation stage. The human elbow joint reaches the limit flexion position, and the elbow joint starts to extend driven by the variable stiffness joint. The human-machine interaction torque gradually decreases from

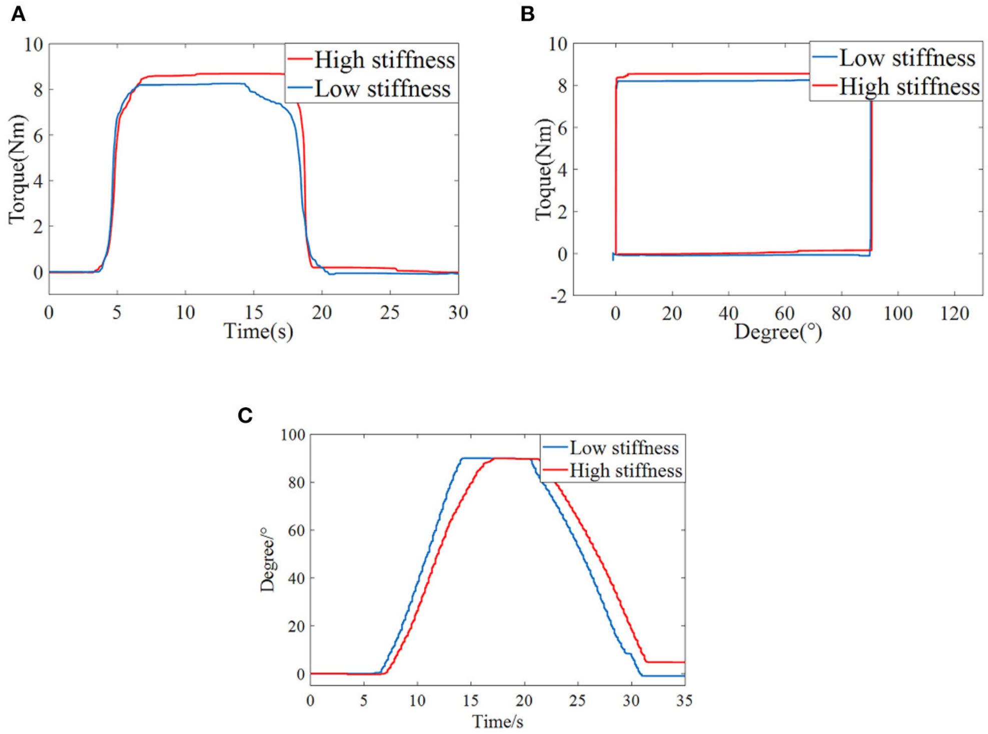


FIGURE 14
Elbow isotonic centripetal resistance training **(A)** Torque-Time curve **(B)** Torque-Angle curve **(C)** Angle-Time curve.

8 to 0 Nm. The torque-angle curve is shown in Figure 14B. Due to the active torque of the user's elbow joint, the human-machine interaction torque rises sharply at 0° until the mechanism starts to rotate when it overcomes the set isotonic resistance torque of 8 Nm, and the rotation angle moves from 0 to 90°. At this time, the elbow joint flexes to the limit position and is about to enter the relaxation stage. After that, the human elbow joint release active torque, and the human-machine interaction torque drops to 0 Nm at 90° rapidly. Then the variable stiffness joint drives the elbow to extend, and the joint angle returns to 0 from90. The angle-time curve in the training process is shown in Figure 14C, corresponding to the preparation stage, the mechanism is in the zero position and does not rotate. In the resistance training stage, the mechanism keeps moving at a constant speed, from 0 to 90°. If the user maintains the torque, the rotation angle of the variable stiffness joint does not change. In the relaxation stage, the mechanism drives the patient's elbow joint to return to its original position. It can be seen from the Figure that under high and low stiffness, the flexible elbow rehabilitation robot can complete isometric resistance training, and the torque is stable, which verifies that the PID torque control method based on BP neural network designed in this paper is suitable for torque control ability under different stiffness again.

Conclusions

The driving joint of the traditional upper limb rehabilitation robot is rigid, which is difficult to ensure the safety and comfort of the rehabilitation process. To solve this problem, this paper proposes a variable stiffness joint applied to the elbow rehabilitation robot. The joint adopts the variable stiffness principle of special curved surface, and the trapezoidal screw in the joint has a self-locking function, and the stiffness can be maintained without the continuous output torque of the stiffness adjustment motor. In the control aspect, the BP neural network based PID controller is used to improve the torque control response performance by adjusting the gain parameters under different stiffness. The experimental platform of elbow rehabilitation robot with variable stiffness joint is built, and the isotonic centripetal resistance training experiment of elbow joint is carried out to verify the effect of BP neural network PID torque control. The future work will be combined with online neural network to match the elbow stiffness of patients in real time for resistance rehabilitation training.

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

BH: conceptualization, supervision, and project administration. BM: writing-original draft and visualization. SL: methodology and formal analysis. HY: supervision and project administration. All authors contributed to the article and approved the submitted version.

Funding

This work was supported in part by National Key R&D Program of China (2020YFC2005800 and 2020YFC2005804), Natural Science Foundation of Shanghai (20ZR1437800), and Biomedical Science and Technology support project of Shanghai (22S31901400).

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Albu-Schaffer, A., Wolf, S., Eiberger, O., Haddadin, S., Petit, F., Chalon, M., et al. (2010). "Dynamic modeling and control of variable stiffness actuators," in *IEEE International Conference on Robotics and Automation (ICRA), May 03-08 2010 Anchorage, AK*, 2155–2162. doi: 10.1109/ROBOT.2010.5509850

Baser, O., and Kizilhan, H. (2018). Mechanical design and preliminary tests of VS-AnkleExo. *J. Braz. Soc. Mech. Sci. Eng.* 40. 1–6. doi: 10.1007/s40430-018-1365-4

Caldwell, D., Tsagarakis, N., Jafari., and Amir. (2015). *Energy efficient actuators with adjustable stiffness: a review on* AwAS, AwAS-II and CompACT VSA changing stiffness based on lever mechanism. *Indus. Robot* 4638–4643. doi: 10.1109/ICRA.2011.5979994

Cestari, M., Sanz-Merodio, D., Arevalo, J. C., and Garcia, E. (2014). An adjustable compliant joint for lower-limb exoskeletons. *IEEE/ASME Transac. Mechatron.* 20, 889–898. doi: 10.1109/TMECH.2014.2324036

Chaichaowarat, R., Nishimura, S., and Krebs, H. I. (2021). *Design and Modeling of a Variable-Stiffness Spring Mechanism for Impedance Modulation in Physical Human-Robot Interaction*, 7052–7057. doi: 10.1109/ICRA48506.2021.9560966

Chang, W. H., and Kim, Y. H. (2013). Robot-assisted therapy in stroke rehabilitation. *J Stroke,* 15, 174–81. doi: 10.5853/jos.2013.15.3.174

Garabini, M., Santina, C. D., Bianchi, M., Catalano, M., Grioli, G., Bicchi, A., et al. (2017). "*Soft* robots that mimic the neuromusculoskeletal system," in *Converging Clinical and Engineering Research on Neurorehabilitation II.* Berlin, Germany, Springer. doi: 10.1007/978-3-319-46669-9_45

Gopura, R. A. R. C., Bandara, D. S. V., Kiguchi, K., and Man, G. K. I. (2016). Developments in hardware systems of active upper-limb exoskeleton robots: A review. *Robot. Auton. Syst.* 75, 203–220. doi: 10.1016/j.robot.2015.10.001

Hubicki, C., Abate, A., Clary, P., Rezazadeh, S., Jones, M., Peekema, A., et al. (2018). Walking and running with passive compliance. *IEEE Robot. Autom. Mag.* 25, 23–39. doi: 10.1109/MRA.2017.2783922

Hung Vu, Q., Aryananda, L., Sheikh, F. I., Casanova, F., and Pfeifer, R. (2011). "A novel mechanism for varying stiffness via changing transmission angle," in *2011 IEEE International Conference on Robotics and Automation (ICRA 2011)*, 5076–5081. doi: 10.1109/ICRA.2011.5980097

Hussain, I., Albalasie, A., Awad, M. I., Tamizi, K., and Gan, D. (2021). Design and control of a discrete variable stiffness actuator with instant stiffness switch for safe human-robot interaction. *IEEE Access* 9, 118215–118231. doi: 10.1109/ACCESS.2021.3105587

Iwasaki, M., Tsujiuchi, N., Koizumi, T. (2002). "Force control for unknown environment using sliding mode controller with gain-scheduled variable hyperplane," in *28th Annual Conference of the IEEE Industrial-Electronics-Society, Nov 05-08 2002 Seville, Spain*, 1812–1817. doi: 10.1109/IECON.2002.1185246

Jafari, A., Tsagarakis, N. G., Sardellitti, I., and Caldwell, D. G. (2014). A new actuator with adjustable stiffness based on a variable ratio lever mechanism. *IEEE-Asme Transac. Mechatron.* 19, 55–63. doi: 10.1109/TMECH.2012.2218615

Jafari, A., Tsagarakis, N. G., Vanderborght, B., Caldwell, D. G., and IEEE. (2010). "A Novel Actuator with Adjustable Stiffness (AwAS)," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 18-22 2010 Taipei, TAIWAN*, 4201–4206. doi: 10.1109/IROS.2010.5648902

Lin, C. C., Ju, M. S., and Lin, C. W. (2003). The pendulum test for evaluating spasticity of the elbow joint. *Arch. Phys. Med. Rehabil.* 84, 69–74. doi: 10.1053/apmr.2003.50066

Liu, L., Hong, Z. J., Penzlin, B., Misgeld, B. J. E., Ngo, C., Bergmann, L., et al. (2021). Low impedance-guaranteed gain-scheduled GESO for torque-controlled vsa with application of exoskeleton-assisted sit-to-stand. *IEEE-Asme Transac. Mechatron.* 26, 2080–2091. doi: 10.1109/TMECH.2020.3032372

Lizheng, P., Aiguo, S., Suolin, D., and Baoguo, X. (2015). *Robot-assisted humanized passive rehabilitation training based on online assessment and regulation. Bio-med. Mat. Eng.* 26, S655–S664. doi: 10.3233/BME-151357

Ma, X., Yang, Q., Jing, C., Sun, M., and Song, J. (2016). "*Design and research of 7 - DOF upper-limb rehabilitation robot flexible joint*," in *International Conference on Advanced Robotics, and Mechatronics*, 614–619. doi: 10.1109/ICARM.2016.7606991

Marchal-Crespo, L., and Reinkensmeyer, D. J. (2009). Review of control strategies for robotic movement training after neurologic injury. *J. NeuroEng. Rehabil.* 6, 20. doi: 10.1186/1743-0003-6-20

Mengacci, R., Angelini, F., Catalano, M. G., Grioli, G., Bicchi, A., Garabini, M., et al. (2020). On the motion/stiffness decoupling property of articulated soft robots with application to model-free torque iterative learning control. *Int. J. Robot. Res.* 40, 348–374. doi: 10.1177/0278364920943275

Misgeld, B. J. E., Hewing, L., Liu, L., and Leonhardt, S. (2017). "Robust gain-scheduled control of variable stiffness actuators," in *20th World Congress of the International-Federation-of-Automatic-Control (IFAC), Jul 09-14 2017 Toulouse, FRANCE*, 8804–8809. doi: 10.1016/j.ifacol.2017.08.1535

Palazzolo, J. J., Ferraro, M., Krebs, H. I., Lynch, D., Volpe, B. T., Hogan, N., et al. (2019). Stochastic estimation of arm mechanical impedance during robotic stroke rehabilitation. *IEEE Trans. Neural. Syst. Rehabil. Eng.* 15, 94–103. doi: 10.1109/TNSRE.2007.891392

Park, J., and Song, J. (2010). "Safe joint mechanism using inclined link with springs for collision safety and positioning accuracy of a robot *arm*," in *2010 IEEE International Conference on Robotics and Automation* (Piscataway, NJ: IEEE), 818. doi: 10.1109/ROBOT.2010.5509492

Song, Z., Guo, S., Pang, M., Zhang, S., Nan, X., Gao, B., et al. (2014). Implementation of resistance training using an upper-limb exoskeleton rehabilitation device for elbow joint. *J. Med. Biol. Eng.* 34, 188–196. doi: 10.5405/jmbe.1337

Sun, J. T., Guo, Z., Zhang, Y. B., Xiao, X. H., and Tan, J. R. (2018). A novel design of serial variable stiffness actuator based on an archimedean spiral relocation mechanism. *IEEE-Asme Transac. Mechatron.* 23, 2121–2131. doi: 10.1109/TMECH.2018.2854742

Tsagarakis, N. G., Sardellitti, I., Caldwell, D. G., and IEEE. (2011). "A new variable stiffness actuator (CompAct-VSA): design and modelling," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Sep 25-30 2011 San Francisco, CA*, 378–383. doi: 10.1109/IROS.2011.6095006

Vanderborght, B., Albu-Schaeffer, A., Bicchi, A., Burdet, E., Caldwell, D. G., Carloni, R., et al. (2013). Variable impedance actuators: a review. *Roboti. Auton. Syst.* 61, 1601–1614. doi: 10.1016/j.robot.2013.06.009

Wolf, S., Feenders, J. E., and IEEE. (2016). "Modeling and benchmarking energy efficiency of Variable Stiffness Actuators on the example of the DLR FSJ," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct 09-14 2016 Daejeon, SOUTH KOREA*, 529–536. doi: 10.1109/IROS.2016.7759104

Xu, G. Z., Song, A. G., and Li, H. J. (2011). Adaptive impedance control for upper-limb rehabilitation robot using evolutionary dynamic recurrent fuzzy neural network. *J. Intell. Robot. Syst.* 62, 501–525. doi: 10.1007/s10846-010-9462-3

Yang, Z., Guo, S., Liu, Y. (2021). "Preliminary evaluation of a performance-based stiffness control for upper limb elbow joints rehabilitation," in *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*, 1280–1285. doi: 10.1109/ICMA52036.2021.9512573

Yi, L., Shuxiang, G., Songyuan, Z., and Boulardot, L. (2018). "Modeling and analysis of a variable stiffness actuator for a safe home-based exoskeleton," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)* (Piscataway, NJ: IEEE), 2243–2248. doi: 10.1109/ICMA.2018.8484298

Yuan, B., Sekine, M., Gonzalez, J., Tames, J. G., and Yu, W. (2013). Variable impedance control based on impedance estimation model with emg signals during extension and flexion tasks for a lower limbrehabilitation robotic system. *J. Novel Physiotherap.* 03, 2. doi: 10.4172/2165-7025.1000178

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 24 November 2022
DOI 10.3389/fnbot.2022.1053360

OPEN ACCESS

EDITED BY
Zhan Li,
Swansea University, United Kingdom

REVIEWED BY
Hui Zhou,
Nanjing University of Science and Technology, China
Mingming Zhang,
Southern University of Science and Technology, China

*CORRESPONDENCE
Rong Song
songrong@mail.sysu.edu.cn

†These authors have contributed equally to this work and share first authorship

RECEIVED 25 September 2022
ACCEPTED 04 November 2022
PUBLISHED 24 November 2022

CITATION
Peng H, Zhou J and Song R (2022) A triple-step controller with linear active disturbance rejection control for a lower limb rehabilitation robot.
Front. Neurorobot. 16:1053360.
doi: 10.3389/fnbot.2022.1053360

A triple-step controller with linear active disturbance rejection control for a lower limb rehabilitation robot

Huanfeng Peng[1,2†], Jie Zhou[1,2†] and Rong Song[1,2*]

[1]The Key Laboratory of Sensing Technology and Biomedical Instrument of Guangdong Province, School of Biomedical Engineering, Sun Yat-sen University, Guangzhou, China, [2]School of Biomedical Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen, China

Lower limb rehabilitation robots (LLRRs) have shown promising potential in assisting hemiplegic patients to recover their motor function. During LLRR-aided rehabilitation, the dynamic uncertainties due to human-robot coupling, model uncertainties, and external disturbances, make it challenging to achieve high accuracy and robustness in trajectory tracking. In this study, we design a triple-step controller with linear active disturbance rejection control (TSC-LADRC) for a LLRR, including the steady-state control, feedforward control, and feedback control. The steady-state control and feedforward control are developed to compensate for the gravity and incorporate the reference dynamics information, respectively. Based on the linear active disturbance rejection control, the feedback control is designed to enhance the control performance under dynamic uncertainties. Numerical simulations and experiments are conducted to validate the effectiveness of TSC-LADRC. The results of simulations illustrate that the tracking errors under TSC-LADRC are obviously smaller than those under the triple-step controller without LADRC (TSC), especially with the change of external loads. Moreover, the experiment results of six healthy subjects reveal that the proposed method achieves higher accuracy and lower energy consumption than TSC. Therefore, TSC-LADRC has the potential to assist hemiplegic patients in rehabilitation training.

KEYWORDS

lower limb rehabilitation robot, triple-step method, linear active disturbance rejection control, dynamic uncertainties, trajectory tracking

Introduction

Globally, stroke is a major threat to human health, and post-stroke care has brought a substantial economic burden to society (Johnson et al., 2019). Due to brain injury, stroke often leads to lower limb dysfunction, which greatly reduces patients' quality of life (Hobbs and Artemiadis, 2020).

Therefore, rehabilitation training is needed to help these patients recover their motor function or reduce the risk of several medical consequences secondary to paralysis, such as muscle atrophy and obesity (Chen et al., 2016). In traditional rehabilitation training, the physiotherapist manually guides the patients with impaired limbs to perform repetitive movement training, which is labor-intensive and difficult to quantitatively assess the level of recovery (Akdogan and Adli, 2011). In order to reduce the workload of physiotherapists and enhance the rehabilitation effect, many studies have been conducted on lower limb rehabilitation robots (LLRRs), such as LOPES (Veneman et al., 2007), HAL (Sankai, 2007), and Lokomat (Riener et al., 2005).

Controllers are the critical factor determining the effectiveness of LLRR-aided rehabilitation (Hussain et al., 2013). Among the present research works, most controllers are designed to assist dysfunctional lower limbs in tracking a predefined trajectory (Li et al., 2021). As a model-free controller with a simple and generic control structure, the proportional-integral-derivative (PID) controller has been widely applied to LLRRs (Wu et al., 2015; Zhang et al., 2016; Al-Waeli et al., 2021). However, due to the underutilization of model information, the robotic systems based on the PID controller show poor robustness to external disturbances. Therefore, model-based controllers are proposed to strengthen the anti-disturbance ability of LLRRs. Shen et al. (2020) combined the kinematics and friction models with adaptive robust position control to improve the tracking performance of LLRR under a complex interaction environment. Hernández et al. (2020) designed a non-singular fast terminal sliding mode control for a powered four-degree-of-freedom LLRR, showing strong robustness to external disturbances. Based on a unilateral human-robot dynamical model, a robust controller was designed to drive a LLRR to follow a pre-specified trajectory (Qin et al., 2020). In fact, the LLRR system is characterized by non-linearity, hence the calculation and deduced process of the designed controllers is complicated. Inspired by the triple-step method (Gao et al., 2014; Zhou et al., 2019) proposed a triple-step non-linear controller for LLRR to guarantee control accuracy under different levels of interaction torque. The triple-step method simplified the complicated design of a non-linear controller as a triple-step design process, including the design of steady-state control, feedforward control, and feedback control. On this basis, the structure of the deduced controller was concise.

Dynamic uncertainties of the LLRR system are the main issue that should be considered in controller design (Li et al., 2021). In the LLRR-aided rehabilitation training, the dynamic uncertainties such as human-robot coupling, model uncertainties, and external disturbances, significantly affect the tracking performance. Owing to the non-linear mapping capability, Zhang et al. (2020) combined a radial basis functions neural network (RBFNN) with a sliding mode controller to approach and compensate for the model uncertainties and external disturbances. Besides, Huang et al. (2022) integrated a disturbance observer (DO) into the controller design to compensate for dynamic uncertainties. Khamar et al. (2021) used a non-linear DO in the backstepping sliding controller to assess the wearer's muscle effort and the uncertainties in modeling. Although the control performance of LLRR can be improved by the RBFNN and DOs, the parameters they introduce are difficult to adjust. Long et al. (2017) presented a controller for trajectory tracking under dynamic uncertainties based on active disturbance rejection control (ADRC), which facilitated the parameter tuning. First proposed by Han (2009), the core idea of ADRC is to view the system's external disturbances and internal uncertainties as "total disturbance", estimate the real-time value of the total disturbance by an extended state observer, and finally compensate for it through feedback to achieve satisfying control performance. Moreover, Gao (2003) proposed a linear version of ADRC (LADRC), i.e., a combination of linear extended state observer (LESO) and linear state feedback, which simplified the control structure and reduced the number of tuning parameters.

Although the LADRC technique is a powerful tool to cope with dynamic uncertainties, extra model information is necessary to further improve the control performance (Li et al., 2022; Long and Peng, 2022). In this paper, a triple-step controller with LADRC (TSC-LADRC) is designed for a LLRR to accurately assist the user in tracking a predefined gait trajectory. On the one hand, the triple-step method establishes the main framework of a model-based controller. On the other hand, the feedback control is modified based on the control conception of LADRC using a second-order error auxiliary system. Accordingly, the total disturbance will be estimated in real time by the LESO and compensated with the feedback control input. To validate the effectiveness of TSC-LADRC, simulations considering the dynamic uncertainties are carried out, and experiments with the LLRR are performed on six healthy subjects. All results show that the trajectory tracking performance under TSC-LADRC is more accurate and robust than that under TSC, especially with different external loads.

System description

Mechanical structure

Based on the physiological characteristics of the human's lower limb, we have developed a LLRR with three degrees of freedom, as shown in Figure 1A. The LLRR includes the hip, knee, and ankle joints, where the hip and knee joints of this LLRR are active joints driven by brushless motors (EC 90flat, Maxon, Switzerland) to assist the movement of the wearer's lower limb in the sagittal plane. The linkage is mainly made of lightweight aluminum and nylon materials through machining and three-dimensional printing. Besides, both the thigh and the

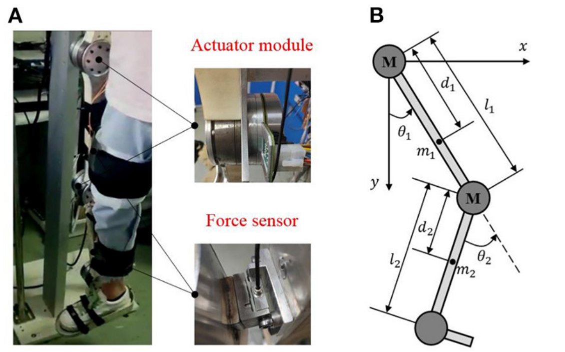


FIGURE 1
The structure of the LLRR. **(A)** is the actual prototype of the LLRR. **(B)** is the simplified two-linkage model of the LLRR.

shank parts are designed as a two-segment mosaic structure that can adapt to subjects of different heights. The wearer's lower limb is fixed to the exoskeleton by Velcro. And the fixed points in the limb are equipped with force sensors (FSSM-500 N, Forsentek, China), which can measure the human-robot interaction forces.

Since the exoskeleton and the wearer perform motions in a shared workspace, the designed exoskeleton must be safe. According to the ranges of motion for the lower limb exoskeleton (Veneman et al., 2007), once the program detects that the joint angle or speed is out of the normal range, the control system will immediately stop driving the motor. In addition, an emergency shutdown button is set to allow the operator to turn off the motor in time. Mechanical limit plays the ultimate role in protection. Please refer to our previous work for more details (Zhou et al., 2021).

Dynamics model

As shown in Figure 1B, the exoskeleton can be simplified to a two-link model in the sagittal plane. Considering the external disturbances, joint friction torques as well as the uncertain model parameters, the dynamics of the LLRR can be modeled by the Euler-Lagrange method as follows:

$$\hat{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}+\hat{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}}+\hat{G}(\boldsymbol{\theta}) =\boldsymbol{\tau}-\boldsymbol{T} \tag{1}$$

$$\boldsymbol{T}=\boldsymbol{\tau}_{HR}+\boldsymbol{f}(\dot{\boldsymbol{\theta}})+\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}+\boldsymbol{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}}+\boldsymbol{G}(\boldsymbol{\theta}) \tag{2}$$

Where $\boldsymbol{\theta}=[\boldsymbol{\theta}_1;\boldsymbol{\theta}_2]\in\mathbb{R}^{2\times 1}$, $\dot{\boldsymbol{\theta}}\in\mathbb{R}^{2\times 1}$ and $\ddot{\boldsymbol{\theta}}\in\mathbb{R}^{2\times 1}$ are joint angle, velocity and acceleration vectors, respectively; $\boldsymbol{\tau}=[\tau_1;\tau_2]\in\mathbb{R}^{2\times 1}$ are the control torques; $\mathbf{f}\left(\dot{\boldsymbol{\theta}}\right)\in\mathbb{R}^{2\times 1}$ and $\tau_{HR}\in\mathbb{R}^{2\times 1}$ are joint friction torques and human-robot interaction torques; $\hat{\mathbf{M}}(\boldsymbol{\theta})\in\mathbb{R}^{2\times 2}$, $\hat{\mathbf{C}}\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right)\in\mathbb{R}^{2\times 2}$ and $\hat{\mathbf{G}}(\boldsymbol{\theta})\in\mathbb{R}^{2\times 1}$ are the nominal inertia matrix, the nominal centripetal and Coriolis matrix, and the nominal gravitational vector, respectively; $\mathbf{M}(\boldsymbol{\theta})\in\mathbb{R}^{2\times 2}$, $\mathbf{C}\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right)\in\mathbb{R}^{2\times 2}$ and $\mathbf{G}(\boldsymbol{\theta})\in\mathbb{R}^{2\times 1}$ are the corresponding model uncertainties between nominal dynamics and actual dynamics; $\mathbf{T}\in\mathbb{R}^{2\times 1}$ is defined as the total disturbances including the structural and non-structural uncertainties.

The nominal dynamics matrixes are expressed in detail as:

$$\begin{cases} \hat{M}(\boldsymbol{\theta})=\begin{bmatrix}\hat{M}_{11}(\boldsymbol{\theta}) & \hat{M}_{12}(\boldsymbol{\theta})\\ \hat{M}_{21}(\boldsymbol{\theta}) & \hat{M}_{22}(\boldsymbol{\theta})\end{bmatrix}\\ \hat{M}_{11}(\boldsymbol{\theta})=\hat{m}_1 d_1^2+\hat{m}_2 l_1^2+\hat{m}_2 d_2^2+2\hat{m}_2 l_1 d_2\cos(\boldsymbol{\theta}_2)+\hat{I}_1+\hat{I}_2\\ \hat{M}_{12}(\boldsymbol{\theta})=\hat{m}_2 d_2^2+\hat{m}_2 l_1 d_2\cos(\boldsymbol{\theta}_2)+\hat{I}_2\\ \hat{M}_{21}(\boldsymbol{\theta})=\hat{M}_{12}(\boldsymbol{\theta})\\ \hat{M}_{22}(\boldsymbol{\theta})=\hat{m}_2 d_2^2+\hat{I}_2 \end{cases} \tag{3}$$

$$\begin{cases} \hat{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})=\begin{bmatrix}\hat{C}_{11}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) & \hat{C}_{12}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\\ \hat{C}_{21}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) & \hat{C}_{22}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\end{bmatrix}\\ \hat{C}_{11}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})=-2\hat{m}_2 l_1 d_2\sin(\boldsymbol{\theta}_2)\dot{\boldsymbol{\theta}}_2\\ \hat{C}_{12}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})=-\hat{m}_2 l_1 d_2\sin(\boldsymbol{\theta}_2)\dot{\boldsymbol{\theta}}_2\\ \hat{C}_{21}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})=\hat{m}_2 l_1 d_2\sin(\boldsymbol{\theta}_2)\dot{\boldsymbol{\theta}}_1\\ \hat{C}_{22}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})=0 \end{cases} \tag{4}$$

$$\hat{\mathbf{G}}(\boldsymbol{\theta})=\begin{bmatrix}(\hat{m}_1 g d_1+\hat{m}_2 g l_1)\sin\boldsymbol{\theta}_1+\hat{m}_2 g d_2\sin(\boldsymbol{\theta}_1+\boldsymbol{\theta}_2)\\ \hat{m}_2 g d_2\sin(\boldsymbol{\theta}_1+\boldsymbol{\theta}_2)\end{bmatrix} \tag{5}$$

Where $\hat{m}_1$ and $\hat{m}_2$ are the mass of the thigh part and the calf part; l_1 and l_2 are the length of the thigh part and the calf part; d_1 and d_2 are the distances from the center of mass to the center of rotation; $\hat{I}_1$ and $\hat{I}_2$ are the moments of inertia of the thigh part and the calf part.

Triple-step controller with LADRC

In this section, a position controller for LLRR is proposed by combining the triple-step method with LADRC, in order to ensure that the robot follows the reference gait trajectory with high accuracy. Designed by the triple-step method, the control framework of the position controller is shown in Figure 2. The core concept is to divide the design process of a non-linear controller into three steps: steady-state control is to compensate for the effect of gravity, which can improve the steady-state performance of the system; feedforward control takes the change of reference dynamics into account, so as to improve the system response speed; feedback control is designed through LADRC using a second-order error auxiliary system, to reduce the influence of dynamic uncertainties during trajectory tracking.

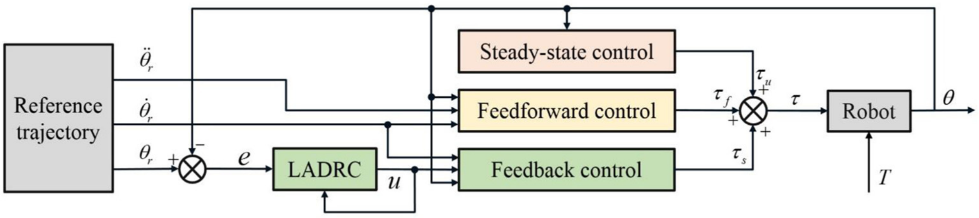


FIGURE 2
Control framework of the triple-step controller with LADRC.

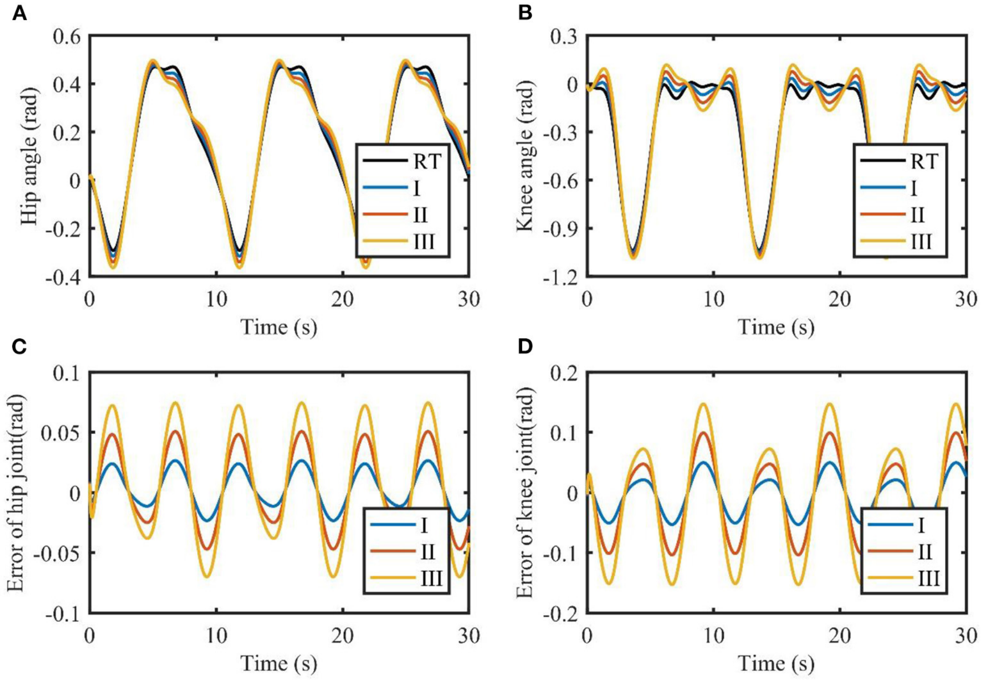


FIGURE 3
Tracking performance of TSC with different loads. **(A,B)** are the joint angles of the hip and knee. **(C,D)** are the tracking errors of the hip and knee. I, II, and III represent three cases of different loads. RT, reference trajectory.

Steady-state control

By assigning zero to $\dot{\boldsymbol{\theta}}$, $\ddot{\boldsymbol{\theta}}$ and $\mathbf{T}$, and replacing τ with $\tau_\mathbf{s}$ in (1), we can obtain the control input defined by steady-state control:

$$\boldsymbol{\tau}_s = \hat{\mathbf{G}}\left(\boldsymbol{\theta}\right) \tag{6}$$

Feedforward control

For a complex system with non-linear and time-varying characteristics, steady-state control alone cannot achieve good control performance. Hence, a feedforward control input is designed to improve the system response speed by considering the variation of the reference dynamics.

By defining $\ddot{\boldsymbol{\theta}} = \ddot{\boldsymbol{\theta}}_r, \dot{\boldsymbol{\theta}} = \dot{\boldsymbol{\theta}}_r, \boldsymbol{\tau} = \boldsymbol{\tau}_s + \boldsymbol{\tau}_f$ and assigning zero to $\mathbf{T}$ in (1), the control input defined by the reference-dynamics-based feedforward control can be obtained:

$$\boldsymbol{\tau}_f = \hat{\mathbf{M}}\left(\boldsymbol{\theta}\right)\ddot{\boldsymbol{\theta}}_r + \hat{\mathbf{C}}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}_r\right)\dot{\boldsymbol{\theta}}_r \tag{7}$$

where $\dot{\boldsymbol{\theta}}_r$, $\ddot{\boldsymbol{\theta}}_r \in \mathbb{R}^{2 \times 1}$ are the derivative and the second derivative of the reference joint angle $\boldsymbol{\theta}_r$.

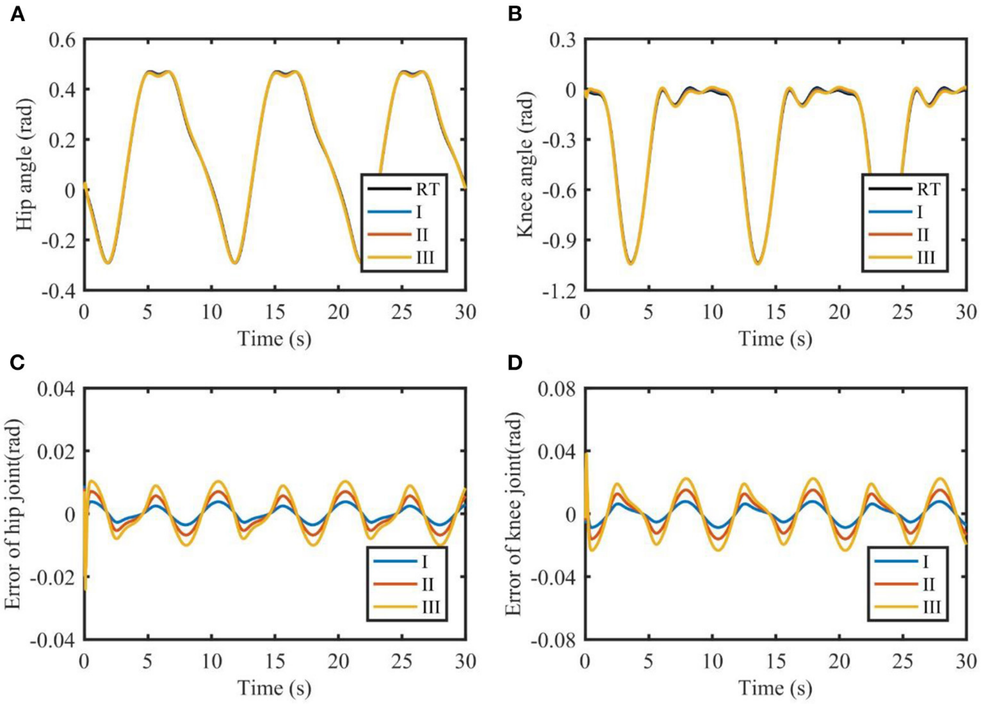


FIGURE 4
Tracking performance of TSC-LADRC with different loads. **(A,B)** are the joint angles of the hip and knee. **(C,D)** are the tracking errors of the hip and knee. I, II, and III represent three cases of different loads. RT, reference trajectory.

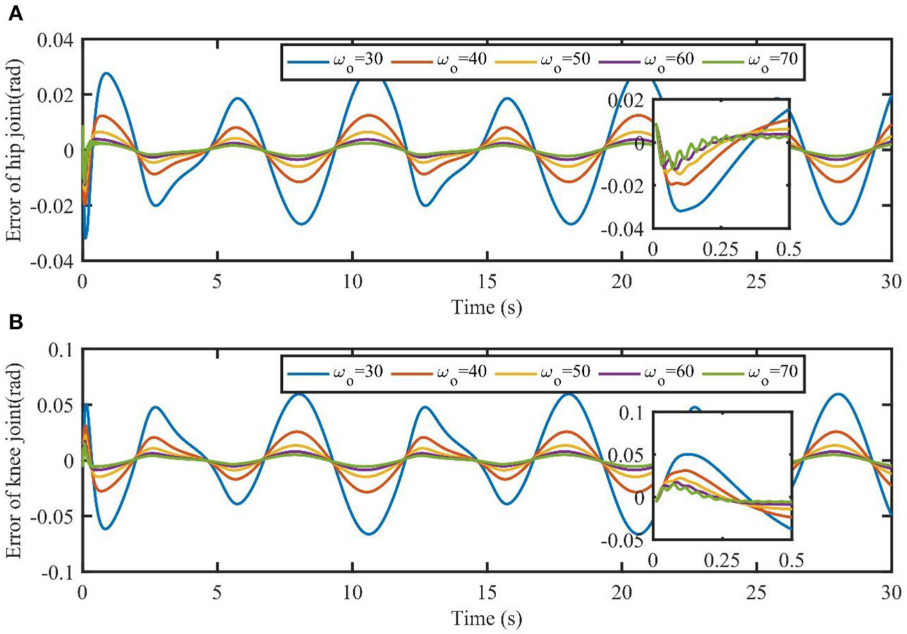


FIGURE 5
Tracking errors of TSC-LADRC under different values of ω_o. The part plots are the initial response of joints' tracking errors in the time interval [0, 0.5s].

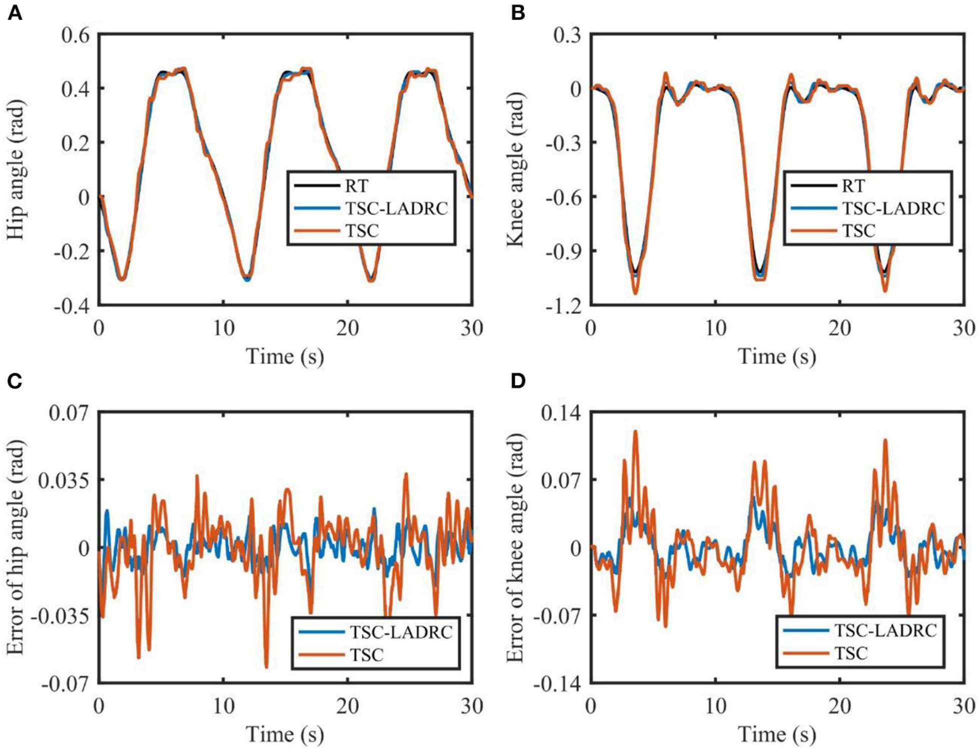


FIGURE 6
Tracking performance of the exoskeleton robot under TSC-LADRC and TSC. **(A,B)** are the joint angles of the hip and knee. **(C,D)** are the tracking errors of the hip and knee. RT, reference trajectory.

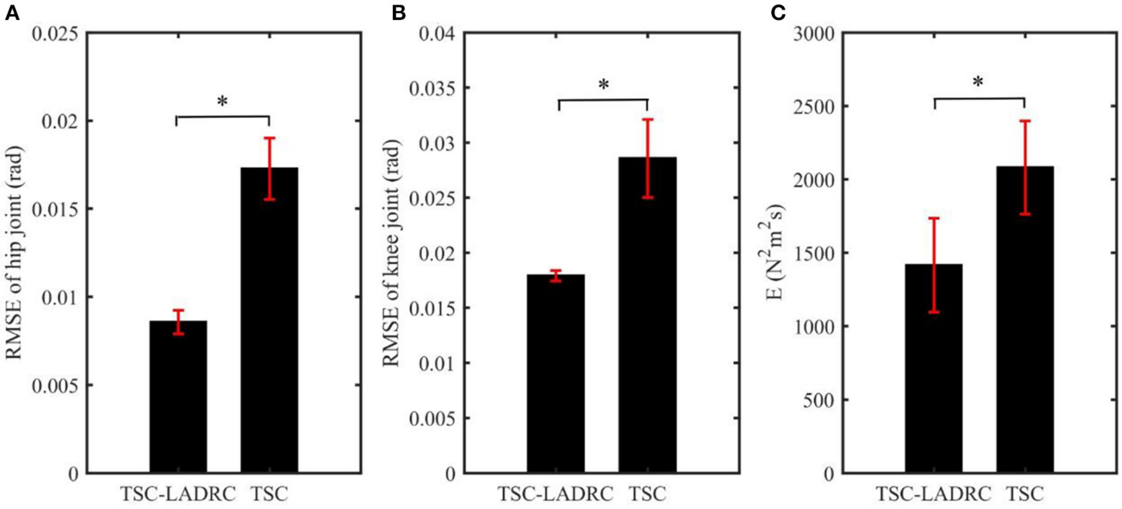


FIGURE 7
The mean values of each evaluation index in all experiments: **(A)** RMSE results of hip joint; **(B)** RMSE results of knee joint; **(C)** Energy index. The error bars indicate the standard errors. The asterisk reveals significant difference (p<0.05).

Feedback control

As for the dynamics equation of the non-linear exoskeleton in (1), the uncertainties such as external disturbances and the change of structural parameters will degrade the performance of the controller. Therefore, it is significant to design a feedback control input to improve the accuracy and robustness of the non-linear system.

Since $\hat{\mathbf{M}}(\boldsymbol{\theta})$ is a positive definite symmetric matrix, the dynamics equation (1) can be rewritten as:

$$\ddot{\boldsymbol{\theta}} = \hat{\boldsymbol{M}}(\boldsymbol{\theta})^{-1}(\boldsymbol{\tau} - \boldsymbol{T}) - \hat{\boldsymbol{M}}(\boldsymbol{\theta})^{-1}\hat{\boldsymbol{C}}\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right)\dot{\boldsymbol{\theta}} - \hat{\boldsymbol{M}}(\boldsymbol{\theta})^{-1}\hat{\boldsymbol{G}}(\boldsymbol{\theta}) \quad (8)$$

By letting $\boldsymbol{\tau} = \boldsymbol{\tau}_s + \boldsymbol{\tau}_f + \boldsymbol{\tau}_u$ and defining the tracking error $\mathbf{e}=\boldsymbol{\theta}_r - \boldsymbol{\theta}$, we can obtain:

$$\begin{aligned}\ddot{\boldsymbol{e}} = {} & \hat{\boldsymbol{M}}(\boldsymbol{\theta})^{-1}\hat{\boldsymbol{C}}\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right)\dot{\boldsymbol{\theta}} - \hat{\boldsymbol{M}}(\boldsymbol{\theta})^{-1}\hat{\boldsymbol{C}}\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}_r\right)\dot{\boldsymbol{\theta}}_r \\ & -\hat{\boldsymbol{M}}(\boldsymbol{\theta})^{-1}\boldsymbol{\tau}_u + \hat{\boldsymbol{M}}(\boldsymbol{\theta})^{-1}\boldsymbol{T}\end{aligned} \quad (9)$$

Based on the LADRC, a second-order error auxiliary system is defined:

$$\ddot{\boldsymbol{e}} = \hat{\boldsymbol{M}}(\boldsymbol{\theta})^{-1}\boldsymbol{T} + (\boldsymbol{b} - \boldsymbol{b}_0)\boldsymbol{u} + \boldsymbol{b}_0\boldsymbol{u} \quad (10)$$

Where $\mathbf{e} = [e_1; e_2]$, $\mathbf{u} = [u_1; u_2]$, and $\mathbf{b} = diag\left(b_1, b_2\right)$ are the output, the input, the dynamic characteristics of the auxiliary system respectively, and $\mathbf{b}_0 = diag\left(b_{01}, b_{02}\right)$ are the presetting values of the dynamic characteristics.

$$\boldsymbol{\tau}_u = -\hat{\boldsymbol{M}}(\boldsymbol{\theta})\boldsymbol{b}\boldsymbol{u} + \hat{\boldsymbol{C}}\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}\right)\dot{\boldsymbol{\theta}} - \hat{\boldsymbol{C}}\left(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}_r\right)\dot{\boldsymbol{\theta}}_r \quad (11)$$

The second-order auxiliary system constructed in (10) is studied as follows.

Defining the total disturbance as $\mathbf{d} = \left[d_1; d_2\right] = \hat{\mathbf{M}}(\boldsymbol{\theta})^{-1}\mathbf{T}+(\mathbf{b} - \mathbf{b}_0)\mathbf{u}$, the auxiliary system can be rewritten as:

$$\ddot{e}_i = d_i + b_{0i}u_i, \quad i = 1,\ 2 \quad (12)$$

The core concept of LADRC is to estimate the real-time action value of the external disturbances and internal uncertainties, and compensate for it in the feedback to eliminate the influence of the total disturbance and thus enhance the performance of disturbance rejection. Specifically, the value of d_i can be estimated as $\hat{d}_i$ by LESO.

The extended state space model of (12) can be expressed as:

$$\begin{cases}\dot{\boldsymbol{x}}_i = \boldsymbol{A}_i\boldsymbol{x}_i + \boldsymbol{B}_i u_i + \boldsymbol{E}_i d_i \\ e_i = \boldsymbol{C}_i\boldsymbol{x}_i\end{cases} \quad (13)$$

Where $\mathbf{x}_i = \left[e_i; \dot{e}_i; d_i\right]$ is the extended state vector, $\mathbf{A}_i = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}$, $\mathbf{B}_i = \left[0; b_{0i}; 0\right]$, $\mathbf{E}_i = [0; 0; 1]$, $\mathbf{C}_i = [1,\ 0,\ 0\ ]$.

The corresponding continuous LESO is:

$$\begin{cases}\dot{\boldsymbol{o}}_i = \boldsymbol{A}_i\boldsymbol{o}_i + \boldsymbol{B}_i u_i + \boldsymbol{L}_i\left(e_i - \hat{e}_i\right) \\ \hat{e}_i = \boldsymbol{C}_i\boldsymbol{o}_i\end{cases} \quad (14)$$

Where $\mathbf{o}_i = [z_{i1}; z_{i2}; z_{i3}] = \left[\hat{e}_i; \hat{\dot{e}}_i; \hat{d}_i\right]$ is the state vector of the observer, and $\mathbf{L}_i = [\beta_{i1}; \beta_{i2}; \beta_{i3}]$ is defined as $\left[3\omega_{oi}; 3\omega_{oi}^2; \omega_{oi}^3\right]$, so that the gain vector of the observer is uniquely related to the bandwidth of the observer, i.e., ω_{oi}. The explanation will be given in Section Stability of the LESO.

Replacing $e_i - \dot{e}_i$ with $\mathbf{C}_i\left(\mathbf{x}_i - \mathbf{o}_i\right)$, the observer equation can be rewritten as:

$$\begin{cases}\dot{\boldsymbol{o}}_i = \boldsymbol{A}_{ei}\boldsymbol{o}_i + \left[\boldsymbol{B}_i\ \ \boldsymbol{L}_i\right]\boldsymbol{u}_{ci} \\ \boldsymbol{y}_i = \boldsymbol{o}_i\end{cases} \quad (15)$$

Where $\mathbf{u}_{ci} = [u_i; e_i]$ is the combined input of the observer, $\mathbf{y}_i = \mathbf{o}_i$ is the observer output, and $\mathbf{A}_{ei} = \mathbf{A}_i - \mathbf{L}_i\mathbf{C}_i$.

For the second-order error auxiliary system, LESO can estimate the external and internal disturbances in real time. Therefore, the integrator in classical PID for eliminating static error under constant disturbance is no longer needed. The linear state feedback control law is further simplified to a proportional–derivative controller:

$$u_{0i} = -k_{Lpi}z_{i1} - k_{Ldi}z_{i2} \quad (16)$$

Where z_{i1} and z_{i2} are states obtained from LESO; k_{Lpi} and k_{Ldi} are the gain coefficients, defined as $k_{Lpi} = \omega_{ci}^2$ and $k_{Ldi} = 2\omega_{ci}$, based on the closed-loop transfer function of the auxiliary system (12):

$$G_i = \frac{k_{Lpi}}{s^2 + k_{Ldi}s + k_{Lpi}} = \frac{\omega_{ci}^2}{(s + \omega_{ci})^2} \quad (17)$$

In this way, the auxiliary system becomes a pure second-order system without zeros and the controller parameters are uniquely related to the controller bandwidth, i.e., ω_{ci}.

Remark 1: The ways to determine the gains of the LESO and the linear error feedback control are termed as ω_o-Optimization and ω_c-Optimization respectively (Gao, 2003). The empirical value of the controller bandwidth ω_{ci} is one-third to one-fifth of the observer bandwidth ω_{oi}. Generally as the value of bandwidth increases, the estimated error decreases and the controller performs better. When the bandwidth increases to a certain extent, the observer will introduce high-frequency noise and reduce the robustness of the system (Han et al., 2021). Since the bandwidths are significantly related to the control performance of the system, the tuning process of the controller is greatly simplified.

The control input of the error system is designed as follows to reject the estimated disturbance:

$$u_i = \frac{u_{0i} - z_{i3}}{b_{0i}} = \frac{-k_{Lpi}z_{i1} - k_{Ldi}z_{i2} - z_{i3}}{b_{0i}} \tag{18}$$

Therefore, the final control input of feedback control is:

$$\begin{aligned}\boldsymbol{\tau}_u &= \hat{\boldsymbol{M}}(\boldsymbol{\theta})\,\boldsymbol{b}\boldsymbol{b}_0^{-1}\left(\boldsymbol{K}_{Lp}\boldsymbol{Z}_1 + \boldsymbol{K}_{Ld}\boldsymbol{Z}_2 + \boldsymbol{Z}_3\right) \\ &\quad + \hat{\boldsymbol{C}}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right)\dot{\boldsymbol{\theta}} - \hat{\boldsymbol{C}}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}_r\right)\dot{\boldsymbol{\theta}}_r\end{aligned} \tag{19}$$

where$\mathbf{b} = diag\left(b_1, b_2\right)$, $\mathbf{b}_0 = diag\left(b_{01}, b_{02}\right)$, $\mathbf{K}_{Lp} = diag\left(k_{Lp1}, k_{Lp2}\right)$, $\mathbf{K}_{Ld} = diag\left(k_{Ld1}, k_{Ld2}\right)$, $\mathbf{Z}_1 = [z_{11}; z_{21}]$ $\mathbf{Z}_2 = [z_{12}; z_{22}]$, $\mathbf{Z}_3 = [z_{13}; z_{23}\ ]$.

Sum up (6), (7), and (19), and the final control law of the triple-step controller with LADRC is obtained as:

$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{\tau}_s + \boldsymbol{\tau}_f + \boldsymbol{\tau}_u = \hat{\boldsymbol{M}}(\boldsymbol{\theta})\,\boldsymbol{b}\boldsymbol{b}_0^{-1}\left(\boldsymbol{K}_{Lp}\boldsymbol{Z}_1 + \boldsymbol{K}_{Ld}\boldsymbol{Z}_2 + \boldsymbol{Z}_3\right) \\ &\quad + \hat{\boldsymbol{M}}(\boldsymbol{\theta})\,\ddot{\boldsymbol{\theta}}_r + \hat{\boldsymbol{C}}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right)\dot{\boldsymbol{\theta}} + \hat{\boldsymbol{G}}(\boldsymbol{\theta})\end{aligned} \tag{20}$$

Stability analysis

Assuming the derivative of the total disturbance $\dot{d}_i$ is bounded, the closed-loop system (1) can be bounded-input-bounded-output stable under the triple-step controller with LADRC.

Stability of the LESO

Defining the estimated error of the observer as

$$\boldsymbol{e}_i^* = \boldsymbol{x}_i - \boldsymbol{o}_i = \left[e_{i1}^*; e_{i2}^*; e_{i3}^*\right] \tag{21}$$

and subtracting (13) and (14), we can obtain the error equation of LESO:

$$\dot{\boldsymbol{e}}_i^* = \boldsymbol{A}_{ei}\boldsymbol{e}_i^* + \boldsymbol{E}_i\dot{d}_i \tag{22}$$

where

$$\boldsymbol{A}_{ei} = \begin{bmatrix} -\beta_{i1} & 1 & 0 \\ -\beta_{i2} & 0 & 1 \\ -\beta_{i3} & 0 & 0 \end{bmatrix} \tag{23}$$

The characteristic polynomial of $\mathbf{A}_{ei}$ is

$$\lambda(s) = s^3 + \beta_{i1}s^2 + \beta_{i2}s + \beta_{i3} \tag{24}$$

By the way of ω_o-Optimization, $\beta_{i1} = 3\omega_{oi}$, $\beta_{i2} = 3\omega_{oi}^2$, $\beta_{i3} = \omega_{oi}^3$, and thus all the roots of $\lambda(s)$ are in the left part of the complex plane. Based on this, the LESO is obviously bounded-input-bounded-output stable because $\dot{d}_i$ is bounded (Qing et al., 2007).

Stability analysis of the triple-step controller with LADRC

According to the second-order error auxiliary system defined by (10) and the control input defined by (18), we can obtain:

$$\begin{cases} \ddot{\boldsymbol{e}} = \boldsymbol{d} + \boldsymbol{b}_0\boldsymbol{u} \\ \boldsymbol{u} = -\boldsymbol{b}_0^{-1}\left(\boldsymbol{K}_{Lp}\boldsymbol{Z}_1 + \boldsymbol{K}_{Ld}\boldsymbol{Z}_2 + \boldsymbol{Z}_3\right) \end{cases} \tag{25}$$

which can be simplified as:

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_{Lp}\boldsymbol{Z}_1 + \boldsymbol{K}_{Ld}\boldsymbol{Z}_2 + \boldsymbol{Z}_3 = \boldsymbol{d} \tag{26}$$

Combining (21) with (26), the dynamics equation of the tracking error can be obtained:

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_{Lp}\boldsymbol{e} + \boldsymbol{K}_{Ld}\dot{\boldsymbol{e}} = \boldsymbol{K}_{Lp}\boldsymbol{E}_1^* + \boldsymbol{K}_{Ld}\boldsymbol{E}_2^* + \boldsymbol{E}_3^* \tag{27}$$

where $\mathbf{E}_1^* = \left[e_{11}^*; e_{21}^*\right]$, $\mathbf{E}_2^* = \left[e_{12}^*; e_{22}^*\right]$, $\mathbf{E}_3^* = \left[e_{13}^*; e_{23}^*\right]$.

As stated in Section Stability of the LESO, as long as $\mathbf{A}_{ei}$ is a Hurwitz matrix and $\dot{\mathbf{d}}$ is bounded, the boundness of $\mathbf{E}_1^*$, $\mathbf{E}_2^*$ and $\mathbf{E}_3^*$ can be guaranteed. Besides, $\mathbf{K}_{Lp}$ and $\mathbf{K}_{Ld}$ are positive-definite by the way of ω_c-Optimization. Therefore, according to the Routh criterion, the tracking error $\mathbf{e}$ is bounded and the system is bounded-input-bounded-output stable.

Simulation

Simulation setup

The uncertainties of the dynamics model, such as the uncertainty of model parameters, sensor measurement noises, external disturbances, load changes and so on, have a significant impact on the performance of a model-based control method. In this section, numerical simulations with uncertainties are carried out in Matlab (R2020b, MathWorks), to verify the excellent performance of the triple-step controller with LADRC

(TSC-LADRC) compared with the original triple-step controller (TSC). The control law of TSC is expressed as (Zhou et al., 2019):

$$\begin{aligned}\boldsymbol{\tau} = {} & \hat{\boldsymbol{M}}(\boldsymbol{\theta})\left(\boldsymbol{K}_p\boldsymbol{e} + \boldsymbol{K}_i\int \boldsymbol{e}dt + \boldsymbol{K}_d\dot{\boldsymbol{e}}\right) + \hat{\boldsymbol{M}}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}_r \\ & + \hat{\boldsymbol{C}}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \hat{\boldsymbol{G}}(\boldsymbol{\theta}) - \boldsymbol{\tau}_{HR}\end{aligned} \tag{28}$$

Where $\mathbf{K}_p$, $\mathbf{K}_i$ and $\mathbf{K}_d$ denote the proportional, the integral and the derivative gain vectors.

The nominal physical parameters are set as $\hat{m}_1 = 2.582\ kg$, $\hat{m}_2 = 3.192\ kg$, $l_1 = 0.390\ m$, $l_2 = 0.464\ m$, $d_1 = 0.328\ m$, $d_2 = 0.355\ m$, and the actual mass parameters are set as $m_1 = \hat{m}_1 \times 120\%$, $m_2 = \hat{m}_2 \times 120\%$. Referring to Yang et al. (2019), the interaction torques are assumed to be periodic, i.e., $\tau_{HR} = [2\cos(0.2\pi t)\ ;\ 2\sin(0.2\pi t)]$. The reference angle of each joint is fitted from a healthy subject's gait data (Zhou et al., 2021). The trajectory's period is set to 10s and a simulation includes three cycles. All the simulations are conducted with a sample time 0.01s. For simplicity, the observer bandwidths of hip and knee joints are set to be the same value $\omega_o = 60$, and the controller bandwidth is one-third of the observer bandwidth. Moreover, $\mathbf{b} = diag\,(10, 10)$,$\mathbf{b}_0 = diag\,(5, 5)$,$\mathbf{K}_p = diag\,(950, 1, 020)$,$\mathbf{K}_i = diag\,(80, 90)$,$\mathbf{K}_d = diag\,(110, 130\ )$.

The interaction between the wearer and the exoskeleton can be measured by force sensors. However, it is difficult to accurately measure the disturbances exerted by the time-varying load through the force sensors. Hence, in order to demonstrate the robust control performance of TSC-LADRC against external loads, three sets of time-varying external torques with different magnitudes are applied to the system to simulate different external loads, i.e., (I) $\tau_{L1} = 5\cos(0.4\pi t)$, $\tau_{L2} = 5\sin(0.4\pi t)$; (II) $\tau_{L1} = 10\cos(0.4\pi t)$, $\tau_{L2} = 10\sin(0.4\pi t)$; and (III) $\tau_{L1} = 15\cos(0.4\pi t)$, $\tau_{L2} = 15\sin(0.4\pi t\ )$.

Simulation results

The results of the simulations are shown in Figures 3, 4. It can be seen from Figure 3 that TSC is able to follow the reference trajectories. However, the presence of model uncertainties makes it difficult to reduce the tracking errors. Furthermore, TSC is susceptible to the changes of the load. As the load increases, the tracking errors of TSC increase significantly, with the maximum errors of the two joints exceeding 0.05 and 0.1 rad, respectively. By contrast, the absolute value of the tracking errors shown in Figures 4C,D are almost less than 0.02 and 0.04 rad, respectively, demonstrating that the proposed TSC- LADRC is robust against different external loads and can achieve higher control accuracy.

To further verify the effect of the observer bandwidth described in Remark 1, the simulation results of TSC-LADRC under different bandwidths are shown in Figure 5. External loads are fixed to $\tau_{L1} = 5\cos(0.4\pi t)$ and $\tau_{L2} = 5\sin(0.4\pi t)$. For simplicity, the observer bandwidths of hip and knee joints keep the same value ω_o. From Figure 5, the larger the value of ω_o, the smaller the tracking errors, which means larger observer bandwidths can enhance the control performance of TSC-LADRC. Besides, when the bandwidth increases to 70, the observer will introduce high-frequency noise and reduce the smoothness of the trajectories. These results are consistent with what we described in **Remark 1**, hence we can intuitively set the parameters by control performance.

TABLE 1 Control parameters.

Method	Parameter	Value
TSC-LADRC	$\mathbf{b}$	$diag\,(0.3,\ 0.5)$
	$\mathbf{b}_0$	$diag\,(1,\ 1)$
	ω_o	$diag\,(36,\ 36)$
	ω_c	$diag\,(12,\ 12)$
TSC	$\mathbf{K}_p$	$diag\,(76.01,\ 126.78)$
	$\mathbf{K}_i$	$diag\,(91.28,\ 253.57)$
	$\mathbf{K}_d$	$diag\,(2.11,\ 5.87)$

Experiment

Experiment protocol

To further validate the superiority of the proposed method to the original triple-step controller, experiments were conducted on the actual exoskeleton robot, based on LADRC-TSC and TSC, respectively. Six healthy subjects were recruited to perform passive trajectory tracking experiments on the LLRR (four males and two females; age, 24.33 ± 2.56 years; height, 1.69 ± 0.07 m; weight, 59.33 ± 7.76 kg). The reference angle of each joint and the sampling time were set the same as those in the simulations. Each subject was asked to perform five experiments for each controller. The experimental operator assisted the subjects in getting familiar with the LLRR before the experiments. Besides, the ethical approval of our study was authorized by the Ethics Committee of Guangdong Work Injury Rehabilitation Center and written informed consents were signed by all subjects. The control parameters of TSC and TSC-LADRC shown in Table 1 are chosen according to the control performance by a trial-and-error method.

During the experiments, the workflow of the robotic system can be described as follows. First, the actual positions of the joints are measured using the angle encoders, which are assembled with the joint motors. Second, a data acquisition board (NI USB-6341, National Instruments, USA) transfers the angle data to a laptop computer with an Intel i5 12500H CPU (2.5 GHz) and 16 GB of RAM. Next, the computer processes the signals in LabVIEW 2018 software based on the position controller. The software generates the control input and translates it into pulse width modulation (PWN) signals. Finally, the PWN signals are transferred to the data acquisition board through a USB interface, and the board sends the signals to the motor drivers, which can supply specific voltages for the joint motors. Meanwhile, the actual positions of the joints are measured again. Therefore, the robot can be driven to follow the reference angles based on this workflow.

Evaluation method

To evaluate the control performance of TSC-LADRC, we calculate the root mean square error (RMSE) based on the trajectory tracking errors of the hip and knee joints, respectively:

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{k=1}^{n} e^{2}\left(k\right)} \tag{29}$$

Where $e\left(k\right)$ is the tracking error at the kth sampling time point, and n is the sample number.

Besides, we calculate the amount of energy that motors consume during trajectory tracking, based on an energy index E (Jiang et al., 2017):

$$E = \sum\nolimits_{i=1}^{2} \int_{0}^{l} \left|\tau_i\left(t\right)\right| dt \tag{30}$$

Where τ_i (i =1, 2) is the control torque and l is the length of the torque signal.

The two evaluation indicators were calculated across each experiment. All the indicators are expressed in the form of mean ± standard error. And the paired t-test with a significance level of 0.05 was utilized to test the effect of the control algorithms statistically.

Experiment results

The tracking results in one experiment are shown in Figure 6. TSC can assist the LLRR in following the reference angles. However, due to the factors such as friction, sensor noise, model uncertainties, and so on, the actual trajectories based on TSC have undesirable chattering phenomena and deviate from the reference trajectories at some point. Compared with TSC, the proposed TSC-LADRC is able to realize more accurate trajectory tracking, and the problem of chattering can be reduced. From Figures 6C,D, the maximum tracking errors of TSC-LADRC are almost half smaller than those of TSC.

The calculation results of RMSE are shown in Figures 7A,B. The RMSE results of the TSC- LADRC are 0.0086 ± 0.0007 and 0.0179 ± 0.0005 rad, while the RMSE results of TSC are 0.0173 ± 0.0017 and 0.0286 ± 0.0035 rad, respectively. It can be seen from Figures 7A,B that the RMSE results of the TSC-LADRC are reduced significantly by 50.29 and 37.41% respectively compared with those of TSC. Besides, the standard error of the RMSE results under TSC-LADRC is less than that under TSC, which means that the proposed controller is more robust than the TSC when coping with different loads. Hence, the TSC-LADRC can improve the accuracy and robustness of trajectory tracking for the lower limb rehabilitation robot.

The energy consumption of each controller is given in Figure 7C. We can see that the energy consumption of LADRC-TSC is less than that of TSC in all experiments. The average energy consumption of LADRC-TSC is 1,414.62 N^2m^2s and that of TSC is 2,081.00 N^2m^2s, showing a significant difference. From the above results, it can be concluded that TSC-LADRC not only realizes more accurate and robust trajectory tracking but also achieves less energy consumption than TSC.

Discussion

Control accuracy and robustness are critical during robot-aided rehabilitation training, while LLRR is vulnerable to dynamic uncertainties due to the unexpected behavior of stroke patients, model uncertainties, and external disturbances (Yang et al., 2019, 2020). In this study, TSC-LADRC was a robust position controller that addressed the dynamic uncertainties in trajectory tracking through a simple and easy-to-apply control structure.

Our previous work has verified that TSC can guarantee control accuracy under different interaction torque levels (Zhou et al., 2019). However, the simulation results in this study revealed that the model errors and external loads would degrade the tracking performance of TSC. Compared with TSC (Zhou et al., 2019), the key feature of TSC-LADRC was to define a second-order error auxiliary system, which could estimate and reject the total uncertainties based on the LADRC (Gao, 2003). On the other hand, the parameter tuning of TSC-LADRC was more straightforward than that of TSC. Based on LADRC (Gao, 2003), TSC-LADRC had only two main parameters to be tuned, the controller bandwidth ω_{ci} and the observer bandwidth ω_{oi}. Moreover, an empirical setting of ω_{ci} was $\frac{1}{5} \sim \frac{1}{3}\omega_{oi}$, meaning that the tuning of TSC-LADRC was further simplified. From the simulation results shown in Figure 5, the tracking errors decreased with the observer bandwidth increasing, which was consistent with the results of Long et al. (2017). Therefore, the main control parameter ω_{oi} of TSC-LADRC was directly related to the control performance and easy to be tuned.

Trajectory error directly reflects the tracking ability of the position controller. Compared with TSC, the decrease in RMSE values of TSC-LADRC demonstrated that the tracking ability of TSC-LADRC improved significantly, which could be explained by the reason that the total disturbances of the robotic system were estimated and compensated by the LADRC-based feedback control. It is worth mentioning that, the RMSE values of TSC-LADRC were smaller than the RMSE results of Huang et al. (2022), and the tracking errors shown in Figures 6C,D were under the average errors of Zhang et al. (2020). This means that, compared with the DO-based controller (Huang et al., 2022) and RBFNN-based controller (Zhang et al., 2020), TSC-LADRC not only facilitates the parameter tuning, but can also address the dynamic uncertainties and improve the tracking accuracy. Moreover, compared with TSC, the decrease in *E* value of TSC-LADRC demonstrated that the energy consumption efficiency of the controller improved significantly (Jiang et al., 2017). We attributed this phenomenon to the fact that, by combining LADRC with TSC, small and bounded tracking errors were guaranteed, which could also lead to a smaller feedback gain in the control law. For the rehabilitation robot system, the energy reduction is beneficial to improving the portability of the exoskeleton design (Ferris et al., 2007).

In future work, experiments will be carried out on patients with motor dysfunction to further verify the clinical effectiveness of TSC-LADRC. Moreover, the LLRR will be combined with treadmill and the motor performance of the wearer's non-paretic limb will be assessed in real time. Based on this real-time assessment, we will focus on the adaptation law of the observer bandwidth to improve patients' gait symmetry and promote their active effort (Wolbrecht et al., 2008; Zhong et al., 2022).

Conclusion

In this study, a triple-step controller with LADRC was proposed for a LLRR to improve gait tracking performance. Under the design framework of the triple-step method, LADRC was incorporated into the feedback control to improve the accuracy and robustness against dynamic uncertainties. Results of numerical simulations and experiments showed that TSC-LADRC could achieve better control performance than TSC. Moreover, our proposed controller facilitated the tuning of control parameters. Therefore, it has the potential to be an easy-to-implement position controller for LLRRs to achieve promising performance, and can be extended to other rehabilitation robots.

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 Ethics Committee of Guangdong Work Injury Rehabilitation Center. The patients/participants provided their written informed consent to participate in this study.

Author contributions

JZ and RS contributed conception of the control algorithm. HP designed and performed the simulations, experiments, and wrote the first draft of the manuscript. All authors contributed to the manuscript revision and approved the submitted version.

Funding

This research was supported by the National Key Research and Development Program of China (Grant No. 2022YFE0201900), the Shenzhen Science and Technology Research Program (Grant No. SGDX20210823103405040), the Guangdong Science and Technology Plan Project (Grant No. 2020B1212060077) and the Natural Science Foundation of Guangdong Province (Grant No. 2020A1515010735).

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Akdogan, E., and Adli, M. A. (2011). The design and control of a therapeutic exercise robot for lower limb rehabilitation: physiotherabot. *Mechatronics* 21, 509–522. doi: 10.1016/j.mechatronics.2011.01.005

Al-Waeli, K. H., Ramli, R., Haris, S. M., Zulkoffli, Z. B., and Amiri, M. S. (2021). Offline ANN-PID controller tuning on a multi-joints lower limb exoskeleton for gait rehabilitation. *IEEE Access* 9, 107360–107374. doi: 10.1109/ACCESS.2021.3101576

Chen, B., Ma, H., Qin, L.-Y., Gao, F., Chan, K.-M., Law, S.-W., et al. (2016). Recent developments and challenges of lower extremity exoskeletons. *J. Orthop. Transl.* 5, 26–37. doi: 10.1016/j.jot.2015.09.007

Ferris, D. P., Sawicki, G. S., and Daley, M. A. (2007). A physiologist's perspective on robotic exoskeletons for human locomotion. *Int. J. Hum. Robot.* 4, 507–528. doi: 10.1142/S0219843607001138

Gao, B., Chen, H., Liu, Q., and Chu, H. (2014). Position control of electric clutch actuator using a triple-step nonlinear method. *IEEE Trans. Ind. Electron.* 61, 6995–7003. doi: 10.1109/TIE.2014.2317131

Gao, Z. (2003). "Scaling and bandwidth-parameterization based controller tuning," in *Annual American Control Conference (ACC 2003)* (Denver, CO: *IEEE*), p. 4989–4996.

Han, J. (2009). From PID to active disturbance rejection control. *IEEE Trans. Ind. Electron.* 56, 900–906. doi: 10.1109/TIE.2008.2011621

Han, X., Liu, G., Chen, B., and Zheng, S. (2021). Surge disturbance suppression of AMB-rotor systems in magnetically suspension centrifugal compressors. *IEEE Trans. Control Syst. Technol.* 30, 1550–1560. doi: 10.1109/TCST.2021.3112765

Hernández, J. H., Cruz, S. S., López-Gutiérrez, R., González-Mendoza, A., and Lozano, R. (2020). Robust nonsingular fast terminal sliding-mode control for sit-to-stand task using a mobile lower limb exoskeleton. *Control Eng. Pract.* 101, 104496. doi: 10.1016/j.conengprac.2020.104496

Hobbs, B., and Artemiadis, P. (2020). A review of robot-assisted lower-limb stroke therapy: unexplored paths and future directions in gait rehabilitation. *Front. Neurorobot.* 14, 19. doi: 10.3389/fnbot.2020.00019

Huang, P., Li, Z., Zhou, M., Li, X., and Cheng, M. (2022). Fuzzy enhanced adaptive admittance control of a wearable walking exoskeleton with step trajectory shaping. *IEEE Trans. Fuzzy Syst.* 30, 1541–1552. doi: 10.1109/TFUZZ.2022.3162700

Hussain, S., Xie, S. Q., and Jamwal, P. K. (2013). Control of a robotic orthosis for gait rehabilitation. *Robot. Auton. Syst.* 61, 911–919. doi: 10.1016/j.robot.2013.01.007

Jiang, B., Li, C., and Ma, G. (2017). Finite-time output feedback attitude control for spacecraft using "Adding a power integrator" technique. *Aerosp. Sci. Technol.* 66, 342–354. doi: 10.1016/j.ast.2017.03.026

Johnson, C. O., Minh, N., Roth, G. A., Nichols, E., Alam, T., Abate, D. et al. (2019). Global, regional, and national burden of stroke, 1990-2016: a systematic analysis for the global burden of disease study 2016. *Lancet Neurol.* 18, 439–458. doi: 10.1016/S1474-442230034-1

Khamar, M., Edrisi, M., and Forghany, S. (2021). Designing a robust controller for a lower limb exoskeleton to treat an individual with crouch gait pattern in the presence of actuator saturation. *ISA Trans.* doi: 10.1016/j.isatra.2021.08.027

Li, P., Wang, L., Zhong, B., and Zhang, M. (2022). Linear active disturbance rejection control for two-mass systems via singular perturbation approach. *IEEE Trans. Ind. Inform.* 18, 3022–3032. doi: 10.1109/TII.2021.3108950

Li, Y., Ke, J., and Zeng, J. (2021). Tracking control for lower limb rehabilitation robots based on polynomial nonlinear uncertain models. *Int. J. Robust Nonlinear Control* 31, 2186–2204. doi: 10.1002/rnc.5381

Long, Y., Du, Z., Cong, L., Wang, W., Zhang, Z., and Dong, W. (2017). Active disturbance rejection control based human gait tracking for lower extremity rehabilitation exoskeleton. *ISA Trans.* 67, 389–397. doi: 10.1016/j.isatra.2017.01.006

Long, Y., and Peng, Y. (2022). Extended state observer-based nonlinear terminal sliding mode control with feedforward compensation for lower extremity exoskeleton. *IEEE Access* 10, 8643–8652. doi: 10.1109/ACCESS.2021.3049879

Qin, F., Zhao, H., Zhen, S., Sun, H., and Zhang, Y. (2020). Lyapunov based robust control for tracking control of lower limb rehabilitation robot with uncertainty. *Int. J. Control Autom. Syst.* 18, 76–84. doi: 10.1007/s12555-019-0175-5

Qing, Z., Gao, L. Q., and Zhiqiang, G. (2007). "On stability analysis of active disturbance rejection control for nonlinear time-varying plants with unknown dynamics," in *2007 46th IEEE Conference on Decision and Control* (New Orleans, LA: *IEEE)*, p. 3501–3506. doi: 10.1109/CDC.2007.4434676

Riener, R., Lunenburger, L., Jezernik, S., Anderschitz, M., Colombo, G., and Dietz, V. (2005). Patient-cooperative strategies for robot-aided treadmill training: First experimental results. *IEEE Trans. Neural Syst. Rehabil. En.* 13, 380–394. doi: 10.1109/TNSRE.2005.848628

Sankai, Y. (2007). "HAL: hybrid assistive limb based on cybernics," in *13th International Symposium on Robotics Research (ISSR)* (Hiroshima: *Springer)*, p. 25–34. doi: 10.1007/978-3-642-14743-2_3

Shen, Z., Zhuang, Y., Zhou, J., Gao, J., and Song, R. (2020). Design and test of admittance control with inner adaptive robust position control for a lower limb rehabilitation robot. *Int. J. Control Autom. Syst.* 18, 134–142. doi: 10.1007/s12555-018-0477-z

Veneman, J. F., Kruidhof, R., Hekman, E. E. G., Ekkelenkamp, R., Van Asseldonk, E. H. F., and van der Kooij, H. (2007). Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation. *IEEE Trans. Neural Syst. Rehabil. Eng.* 15, 379–386. doi: 10.1109/TNSRE.2007.903919

Wolbrecht, E. T., Chan, V., Reinkensmeyer, D. J., and Bobrow, J. E. (2008). Optimizing compliant, model-based robotic assistance to promote neurorehabilitation. *IEEE Trans. Neural Syst. Rehabil. Eng.* 16, 286–297. doi: 10.1109/TNSRE.2008.918389

Wu, Q., Wang, X., Du, F., and Zhang, X. (2015). Design and control of a powered hip exoskeleton for walking assistance. *Int. J. Adv. Robot. Syst.* 12, 1–11. doi: 10.5772/59757

Yang, S., Han, J., Xia, L., and Chen, Y.-H. (2020). An optimal fuzzy-theoretic setting of adaptive robust control design for a lower limb exoskeleton robot system. *Mech. Syst. Signal Process* 141, 1–25. doi: 10.1016/j.ymssp.2020.106706

Yang, Y., Huang, D., and Dong, X. (2019). Enhanced neural network control of lower limb rehabilitation exoskeleton by add-on repetitive learning. *Neurocomputing* 323, 256–264. doi: 10.1016/j.neucom.2018.09.085

Zhang, F., Hou, Z.-G., Cheng, L., Wang, W., Chen, Y., Hu, J. et al. (2016). iLeg- a lower limb rehabilitation robot: a proof of concept. *IEEE Trans. Hum. Mach. Syst.* 46, 761–768. doi: 10.1109/THMS.2016.2562510

Zhang, P., Zhang, J., and Zhang, Z. (2020). Design of RBFNN-based adaptive sliding mode control strategy for active rehabilitation robot. *IEEE Access* 8, 155538–155547. doi: 10.1109/ACCESS.2020.3018737

Zhong, B., Guo, K., Yu, H., and Zhang, M. (2022). Toward gait symmetry enhancement *via* a cable-driven exoskeleton powered by series elastic actuators. *IEEE Robot. Autom. Lett.* 7, 786–793. doi: 10.1109/LRA.2021.3130639

Zhou, J., Li, Z., Li, X., Wang, X., and Song, R. (2021). Human-robot cooperation control based on trajectory deformation algorithm for a lower limb rehabilitation robot. *IEEE/ASME Trans. Mechatron.* 26, 3128–3138. doi: 10.1109/TMECH.2021.3053562

Zhou, J., Yang, R., Zhao, L., Gao, J., and Song, R. (2019). "Gait tracking based triple-step nonlinear control for a lower limb rehabilitation robot," in *2019 IEEE 9th Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, (Suzhou: *IEEE*), 1006–1009. doi: 10.1109/CYBER46603.2019.9066769

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 28 November 2022
DOI 10.3389/fnbot.2022.982505

UVMS task-priority planning framework for underwater task goal classification optimization

Yu-er Gao[1,2], Xiaohui Zhang[1,2]*, Yan Su[1,2], Jinbao Wang[1,2], Qihang Yang[1,2], Wenqi Bai[1,2] and Songnan Yang[1,2]

[1]Department of Information and Control Engineering, Xi'an University of Technology, Xi'an, China,
[2]Hanjiang-Weihe River Valley Water Diversion Project Construction Co., LTD., Xi'an, China

Check for updates

OPEN ACCESS

EDITED BY
Shuai Li,
Swansea University, United Kingdom

REVIEWED BY
Ameer Tamoor Khan,
Hong Kong Polytechnic University,
Hong Kong SAR, China
Bin Zhi Li,
Chongqing Institute of Green and
Intelligent Technology (CAS), China

*CORRESPONDENCE
Xiaohui Zhang
xhzhang@xaut.edu.cn

RECEIVED 30 June 2022
ACCEPTED 31 October 2022
PUBLISHED 28 November 2022

CITATION
Gao Y-e, Zhang X, Su Y, Wang J, Yang Q, Bai W and Yang S (2022) UVMS task-priority planning framework for underwater task goal classification optimization.
Front. Neurorobot. 16:982505.
doi: 10.3389/fnbot.2022.982505

This paper presents a task prioritization strategy based on a generic underwater task goal classification transformation for multitasking underwater operational tasks: attitude control, floating manipulation, collision-free motion, especially optimizing trajectory of the end-effector of an underwater vehicle manipulator system (UVMS) in a complex marine environment. The design framework aims to divide the complex underwater operational tasks into UVMS executable generic task combinations and optimize the resource consumption during the whole task. In order to achieve the corresponding underwater task settings, the system needs to satisfy different task scheduling structures. We consider the actual application scenarios of the operational goals and prioritize and define each category of task hierarchy accordingly. Multiple tasks simultaneously enable fast adaptation to UVMS movements and planning to complete UVMS autonomous movements. Finally, an underwater vehicle manipulator system implements the task prioritization planning framework for a practical scenario with different constraints on different goals. We quickly and precisely realize the interconversion of different tasks under goal constraints. The autonomous motion planning and real-time performance of UVMS are improved to cope with the increasing operational task requirements and the complex and changing practical engineering application environments.

KEYWORDS

underwater vehicle manipulator system, task prioritization strategy, motion planning, trajectory optimization, nonlinear optimization

Introduction

As the largest ecosystem on Earth, the ocean regulates not only global climate change but also supports global economic development by providing humans with productive resources such as protein, water, and energy. In the past decades, understanding and developing the oceans require various high technologies and equipment, including underwater robots, which have been the focus of attention worldwide. In particular, the underwater vehicle manipulator system (UVMS) plays a pivotal role in national projects [RAUVI (Sanz et al., 2011) or ARCHROV (Casalino et al., 2012)] and European projects [FP7 STREP TRIDENT (Sanz et al., 2010), PANDORA (Heshmati-Alamdari et al., 2018), MORPH, Eurofelts2 (Olguin-Diaz et al., 2013), etc.] about underwater robots, which indicates that the emergence as one of the most powerful tools for human research and exploitation activities of marine resources.

Due to the redundancy of degrees of freedom, nonlinearity, strong coupling, time variation, high dimensionality, low bandwidth of sensor data acquisition, and interference by hydrodynamic forces, the system design, autonomous control, and operation planning of UVMS has become a challenging topic in the field of underwater robotics at home and abroad. Since the underwater environment is different from the surface, the realization of the control task of the robot system becomes more difficult. The control problem of UVMS as a redundant system is challenging to be solved completely. However, the redundancy characteristics of the system can be explored reasonably to perform multiple tasks simultaneously and ensure that each task is eventually completed. Therefore, it is necessary to coordinate the motions of the body and the manipulator when meeting the operational task requirements (Fujie et al., 2020). The most commonly used methods for motion planning are the minimum parametric solution method, the weighted pseudo-inverse method, the gradient projection method, and the task prioritization method to achieve UVMS motion planning (Whitney, 1969). On top of this method, some scholars have improved and extended it by introducing the weight matrix into the robot body and manipulator's joints to achieve weighted parametric optimality in robot configuration. Gianluca Antonelli in Italy proposed a task priority-based planning method for UVMS motion, which sets the primary and secondary tasks, prioritizes to ensure the completion of the primary task, and completes the secondary tasks as much as possible under the premise of completing the primary task (Han et al., 2011). Task prioritization methods are used to decide the order of task execution according to the task priority level when multiple tasks are in conflict and are often used to solve redundancy problems. For example, Antonelli and Chiaverini (1998), Cieslak et al. (2015), Changmi (2022), and Gancet et al. (2016) used this approach. Tang et al. (2017) proposed an acceleration level task priority redundancy decomposition method. Simetti et al. (2018) proposed a task priority approach that can be applied to different scenarios in UVMS. The multitask weight gradient method has also been used for secondary task weight assignment (Wang et al., 2017). Sotiropoulos et al. (2015) proposed a fast motion planning algorithm for UVMS in semi-structured environments. Youakim et al. (2017) used different motion planning methods to simulate and analyze the motion of an underwater manipulator, solving the problem of "which planner to choose". Depending on the specific situation, different strategies are needed for the dual-arm problem (Moe et al., 2014; Bae et al., 2018) and the cooperative operation problem (Xuefeng and Xinqian, 2000; Chang, 2004; Conti et al., 2015; Simetti and Casalino, 2016). Some researchers have developed analysis software packages, such as UWSim (Prats et al., 2012) and MoveIt! For example, it is guaranteed that the end position pose of UVMS reaches the desired value, and then constraints such as system energy consumption or manipulator limit are implemented. Subsequently, based on this algorithm, researchers introduced the fuzzy theory to adjust the parameters online for the problem of optimal allocation of primary and multiple secondary tasks (Antonelli and Chiaverini, 2000) to improve task self-adaptability and achieve multitask planning. The literature (Podder and Sarkar, 2000) decomposed the overall motion into the body and the manipulator motion based on the response differences between the body and the manipulator systems in the dynamics model. Based on the UVMS kinetic model, Huang Hai et al. of Harbin Engineering University proposed pairwise optimization combined with a genetic algorithm for trajectory planning of UVMS motion (Huang et al., 2016), which obtained a set of hull positions and manipulator joint angles by genetic, crossover, and variation operations. If the set of sequences satisfied the error range, then the adaptive function was compared and continuously iterated to obtain the optimal global solution.

However, most of the above studies consider a single task during planning. Few have come to deal with task coordination and planning of transitions between tasks in underwater vehicle manipulator system. Tasks can be kinematic (position) or kinetic (force) goals for robot motion control. The ability of a robot to accomplish a goal depends on its physical limitations and surrounding environmental obstacles. Nakamura et al. (1987) introduced the concept of task prioritization associated with the inverse problem of redundant degree manipulator kinematics to determine joint motions with sequential tasks. A general framework for managing multiple tasks of highly redundant robotic systems was proposed in Siciliano and Slotine (1991), but only equality tasks were considered. Researchers tended to transform inequality tasks into equality tasks with the highest priority (Sentis and Khatib, 2005; Mansard and Chaumette, 2007), which could lead to discontinuities. Mansard et al. (2009) used a weighted solution to overcome this drawback with a limited number of inequalities. Inspired by the sequential least-squares formulation of the classical task framework (Jin, 1996), Kanoun et al. (2011) extended the task prioritization framework to inequality tasks. They applied the algorithm to the humanoid robot HRP-2. While much work has been done at the UVMS control level, many scholars have proposed many approaches to the trajectory tracking control of UVMS. Han et al. considered the effect of external disturbances and proposed nonlinear H-optimal control with disturbance observer for implementing tracking control of UVMS in Ref. (Han and Chung, 2008). Mohan et al. proposed an indirect adaptive control method based on the Kalman filter for autonomous operation of UVMS, which overcomes the drawbacks of existing disturbance observer and direct adaptive control. The method can target the consideration of load compensation, underwater currents, or external disturbance compensation. Xu et al. proposed a neuro-fuzzy-based intelligent control algorithm for operational control of UVMS. The proposed decentralized neural network compensator was used to estimate UVMS dynamics, which can better cope with load variations and hydrodynamic disturbances (Xu et al., 2012). Olguin-Diaz et al. (2013) proposed a force-motion-based control framework for operational control of UVMS, which does not require prior modeling of UVMS dynamics and applies a second-order model-free sliding mode control method to calculate the control rate in the force-motion framework. Lynch and Ellery (2014) combined feedback and a feedforward approach to achieve robot attitude and position stabilization for effective manipulator control. However, the problem of task planning as the highest level of the control system, task goal assignment, and transformation during underwater tasks performed by UVMS remains a critical issue in the field of robotics. Although studies consider the uncertainty of its operation in the real world, this is an open research area that needs to produce robust planning algorithms.

Optimization is a high-level task in trajectory planning to seek safety when per-forming tasks in a cluttered, dense, and complex underwater environment. Since resources are limited in the marine environment, mission planning needs to be done under the given constraints, solving its underwater constraints, optimization objectives, physical limitations, and resource invocation problems. Therefore, this paper focuses on the UVMS-based task prioritization strategy. Here, we define the hierarchy of underwater control tasks and their priority relationships. The lowest priority optimization objective is a linear constrained quadratic programming problem. Multiple objective functions with priority equation constraints on the tasks define this optimization problem in this study. The main contributions of this paper are twofold: (1) We develop a UVMS-based task prioritization framework to select the priority of the control tasks in terms of the hierarchy of tasks used and the difference in the priority of the objectives. We give the final planning results in task sequences and resource allocation schemes for each phase, which has been achieved successfully. In addition, we deal with the difficulties of multiple task interconversion activation and multi-objective classification definition at one time. Furthermore, the proposed task prioritization framework can extend widely to underwater operations in real scenarios. (2) Based on the highest objective optimization as the lowest priority objective setting, we propose a soft constraint optimization method to avoid possible collisions and expect the optimization trajectory to reach the ideal trajectory to ensure the underwater task execution of high quality.

The rest of this paper is organized as follows. In Section UVMS system modeling, we complete the modeling of the UVMS. Section Task Planning presents the task prioritization framework composition, defines the division of control objectives and control tasks, and provides detailed definitions of the five categories of tasks. Section Trajectory planning explains the trajectory optimization algorithm and proposes an improved collision avoidance method based on soft constraints. In Section Simulation results, we develop code for the task manager and part of the kinematic control layer, do the most important related simulation experiments, and verify the method's effectiveness with two case studies. Finally, Section Conclusions gives conclusions and future work.

UVMS system modeling

As shown in Figure 1, the UVMS consists of a vehicle and a seven-function manipulator, which is very flexible and well suited for task scenarios with continuous underwater operations.

As shown in Figure 2, where the fixed inertial frame of the world < w >, the vehicle frame of the UVMS < v >, the sensor frame < s > and the operational target frame < o >, the tool frame < t >.

Kinematics of UVMS

The kinematics of the end-effector needs to be represented by the whole system, and the system's structure c is described by a vector of the parameters of the degrees of freedom of each part of the structure.

$$c = \begin{bmatrix} q \\ \eta \end{bmatrix} \tag{1}$$

where q is the description vector of the underwater manipulator

$$q = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix} \tag{2}$$

where η is the description vector of the vehicle

$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \in R^6 \tag{3}$$

The relationship between the Euler angular derivative (RPY) and the angular velocity ν_2 of the vehicle chassis is shown below.

$$\eta_1 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \eta_2 = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{4}$$

$$\dot{y} = \begin{bmatrix} \dot{q} \\ \nu \end{bmatrix} \quad \nu = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} \quad \begin{matrix} \nu_1 =^{\nu} v \\ \nu_2 =^{\nu} w \end{matrix} \tag{5}$$

where the relationship between the Euler angular rate $(\dot{\phi},\ \dot{\theta},\ \dot{\psi})$ and the angular velocity $(p,\ q,\ r)$ of the object in the ontological coordinate system is given by the following equation.

$$\begin{cases} \dot{\phi} = p + q(\sin\phi\tan\theta) + r(\cos\phi\tan\theta) \\ \dot{\theta} = 0 + q(\cos\phi) + r(-\sin\phi) \\ \dot{\psi} = 0 + q(\sin\phi/\cos\theta) + r(\cos\phi/\cos\theta) \end{cases} \tag{6}$$

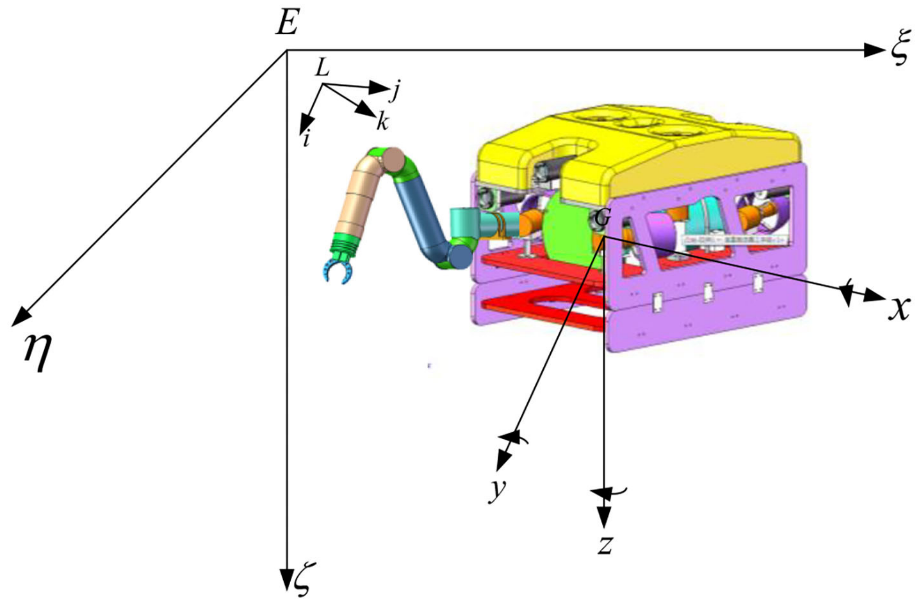


FIGURE 1
UVMS system structure diagram.

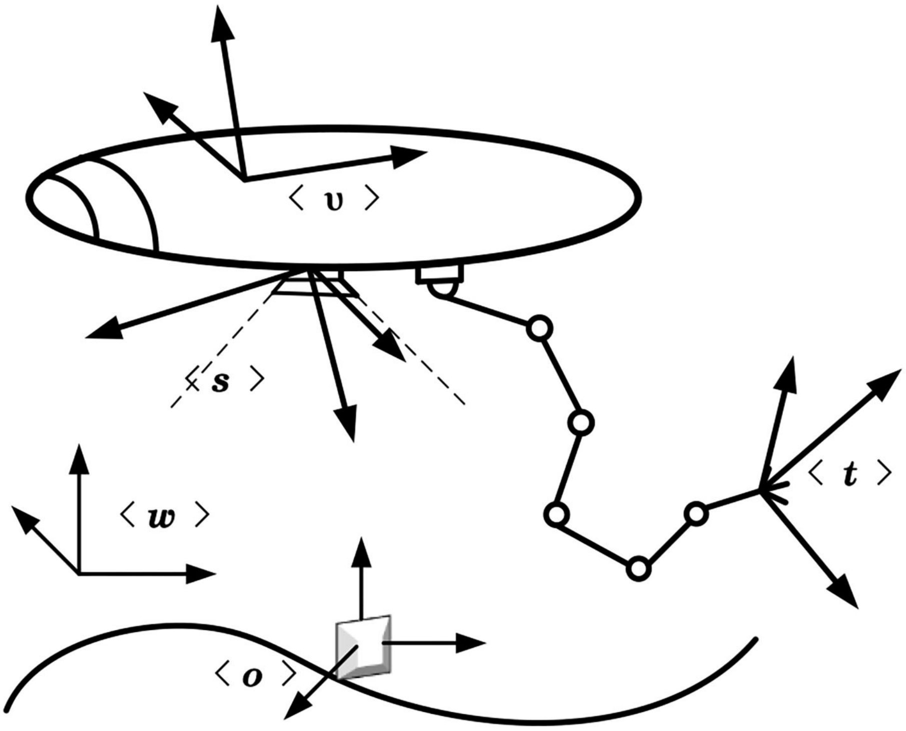


FIGURE 2
The UVMS and its relevant frames.

Putting it into matrix form yields

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{7}$$

For the angular velocity, the rotation matrix $J(\eta)$ is

$$J(\eta) = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{pmatrix} \tag{8}$$

i.e.,

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{pmatrix} \upsilon_2 \tag{9}$$

Therefore, the following relationship exists between the world frame < w > of the Earth and the vehicle frame < v > of UVMS.

$${}^{w}\dot{x} = J^{v}\dot{y} \tag{10}$$

Dynamics of UVMS

This part is modeled by the Lagrangian method. The kinetic energy of UVMS consists of two parts: the translational kinetic energy and the rotational kinetic energy. The mathematical expression is

$$T_i = \frac{1}{2}m_i\dot{r}_i^2 + \frac{1}{2}J_i\omega_i^2 \quad (i = 0, 1, 2) \tag{11}$$

where

$$\begin{aligned} \omega_0 &= \dot{\alpha}Z_0; \\ \omega_1 &= (\dot{\alpha} + \dot{\theta}_1)Z_1; \\ \omega_2 &= (\dot{\alpha} + \dot{\theta}_1 + \dot{\theta}_2)Z_2 \end{aligned} \tag{12}$$

The kinetic energy of the entire single-arm system is

$$\begin{aligned} T &= \sum_{i=0}^{2} T_i \\ &= \frac{1}{2}m_0\left(\dot{x}_0^2 + \dot{y}_0^2\right) + \frac{1}{2}J_0\omega_0^2 + \frac{1}{2}m_1\left(\dot{x}_1^2 + \dot{y}_1^2\right) \\ &\quad + \frac{1}{2}J_1\omega_1^2 + \frac{1}{2}m_2\left(\dot{x}_2^2 + \dot{y}_2^2\right) + \frac{1}{2}J_2\omega_2^2 \\ &= f_1\left(\dot{x}_0^2 + \dot{y}_0^2\right) + f_2\dot{\alpha}^2 + f_3\left(\dot{\alpha} + \dot{\theta}_1\right)^2 + f_4\left(\dot{\alpha} + \dot{\theta}_1 + \dot{\theta}_2\right)^2 \\ &\quad + f_5\dot{\alpha}\left(\dot{x}_0\cos\alpha - \dot{y}_0\sin\alpha\right) + f_6\left(\dot{\alpha} + \dot{\theta}_1\right) \\ &\qquad \left[\dot{x}_0\cos\left(\alpha + \theta_1\right) - \dot{y}_0\sin\left(\alpha + \theta_1\right)\right] \\ &\quad + f_7\left(\dot{\alpha} + \dot{\theta}_1 + \dot{\theta}_2\right)\Big[\dot{x}_0\cos\left(\alpha + \theta_1 + \theta_2\right) \\ &\quad - \dot{y}_0\sin\left(\alpha + \theta_1 + \theta_2\right)\Big] \\ &\quad + f_8\dot{\alpha}\left(\dot{\alpha} + \dot{\theta}_1\right)\cos\theta_1 + f_9\dot{\alpha}\left(\dot{\alpha} + \dot{\theta}_1 + \dot{\theta}_2\right)\cos\left(\theta_1 + \theta_2\right) \\ &\quad + f_{10}\left(\dot{\alpha} + \dot{\theta}_1\right)\left(\dot{\alpha} + \dot{\theta}_1 + \dot{\theta}_2\right)\cos\theta_2 \end{aligned} \tag{13}$$

Here, assuming the gravitational potential energy of the system $V = 0$, the Lagrangian function of the system: $L = T - V$, and using the Lagrangian equation:

$$Q = \frac{d}{d_t}\left(\frac{\partial L}{\partial \dot{q}_0}\right) - \frac{\partial L}{\partial q_0} \tag{14}$$

where $\dot{q}_0 = \begin{bmatrix} \dot{x}_0 & \dot{y}_0 & \dot{\alpha} & \dot{\theta}_1 & \dot{\theta}_2 \end{bmatrix}^T$ is the state vector of the system; $Q = \begin{bmatrix} 0 & 0 & \tau_0 & \tau_1 & \tau_2 \end{bmatrix}^T$ is the control torque matrix of the system.

Substituting into Equation (14) yields the following kinetic equation.

$$D\left(q_0\right)\ddot{q}_0 + H(q_0, \dot{q}_0)\dot{q}_0 = Q \tag{15}$$

where $D\left(q_0\right)$ is the 5*5-dimensional symmetric, positive definite mass matrix. $H(q_0, \dot{q}_0)$ is the 5*1 dimensional matrix with Koch forces and centripetal forces.

At this point, the dynamics model of the UVMS is established using the Lagrangian modeling method. This dynamical model provides the basis for the task priority control of the underwater manipulator and the underwater robot. Due to the dynamical coupling effect between the manipulator and the base, the motion of the base, and the end effector in free-floating mode is highly dependent on the joint trajectory. Therefore, rational design of task planning solves the problem of multitasking underwater operation tasks such as precise positioning, floating manipulation, or collision-free motion.

Task planning

A two-tier framework for task planning

The UVMS control system in this research consists of task planning, trajectory planning, and motion control. As the top layer of the manipulator control system, task planning is responsible for receiving, analyzing, and disassembling task targets. The purpose is to divide complex task targets into action sequences that the manipulator can directly plan and execute. Due to the diversity of ways for the manipulator to complete tasks, task planning also involves scheduling various types of resources for the manipulator system to optimize resource consumption during the entire task. We give the final planning result through task combination sequences and activation methods. Figure 3 shows the two-layer framework of task planning designed in this paper.

The implementation architecture of the task planning approach proposed in this research is shown in Figure 4.

1. Task Manager: Notifies the Kinematic Control Layer about the actions that must be executed based on the current mission.
2. Kinematic Control Layer: Implements the task priority control framework and generates the system reference

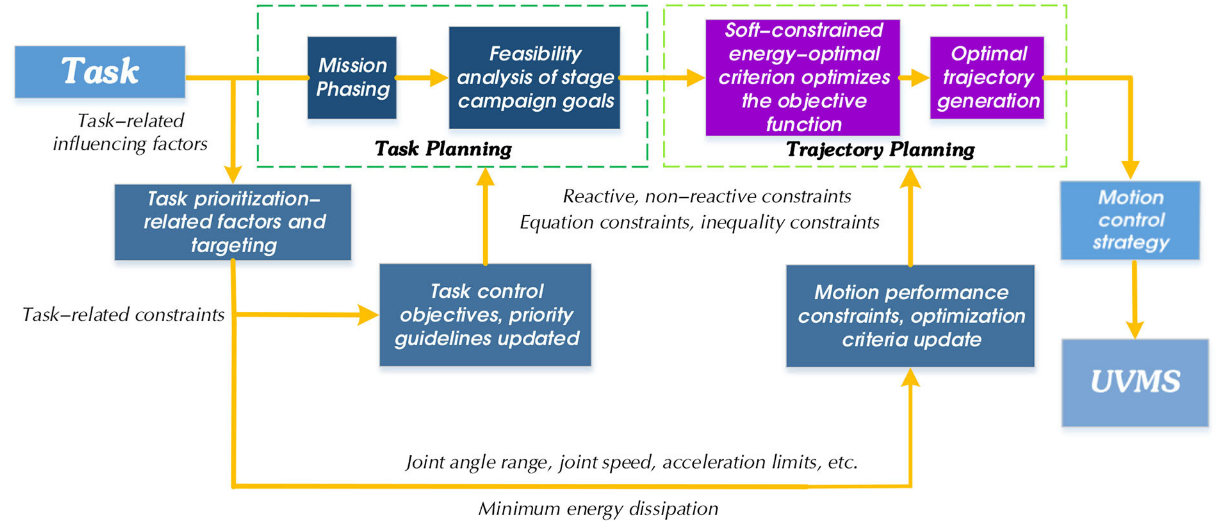

Task
Task–related influencing factors
Mission Phasing
Feasibility analysis of stage campaign goals
Task Planning
Soft–constrained energy–optimal criterion optimizes the objective function
Optimal trajectory generation
Trajectory Planning
Task prioritization–related factors and targeting
Reactive, non–reactive constraints
Equation constraints, inequality constraints
Motion control strategy
Task–related constraints
Task control objectives, priority guidelines updated
Motion performance constraints, optimization criteria update
UVMS
Joint angle range, joint speed, acceleration limits, etc.
Minimum energy dissipation

FIGURE 3
Two-tier framework diagram for task planning.

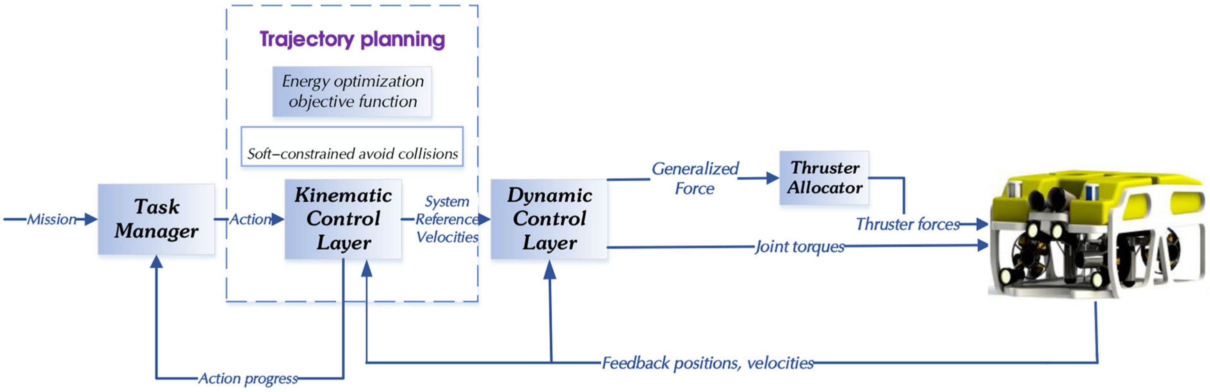


FIGURE 4
General architecture of the task prioritization approach.

velocities. The kinematics control layer mainly manages the end position, joint angle, and speed of UVMS in real-time. For the end-effector, it moves according to the motion trajectory generated by the optimization algorithm designed in this paper and according to the specified motion parameters.

3. Dynamic Control Layer: Tracks the system reference velocities by generating appropriate force/torque references for the vehicle and manipulator.

In the simulation, we develop the code for the Mission Manager and parts of the Kinematic Control Layer.

Task-priority handling strategy

Considering the task requirements for autonomous UVMS underwater operations, we designed two categories of control tasks.

1. Reactive control task (R): Capable of tracking feedback-generated reference rate $\dot{x}$.
2. Non-Reactive control task (NR): Defined directly in specific task velocity space. Thus, the reference velocity tracked is not generated by feedback.

TABLE 1 Control tasks and prioritization.

Priority	Category	Description	Objective
1	Safety	[R,I,S]	Minimum Altitude Control (MAC)
2	Prerequisite	[R,I,P]	Horizontal Attitude (HA)
3	Action-defining	[R,E,AD]	Landing Altitude (LA)
4	Prerequisite	[R,E,C]	Alignment Target (AT)
5	Action-defining	[R,E,AD]	Position Control (PC)

[R/NR, I/E, C/S/P/AD] Name of the task/objective.

Also, depending on the type of task, we have designed two different types of objectives.

1. Equality control objective (E): Given by $x(c) = x_0$, which is given as a constant equation.
2. Inequality control objective (I): Given by $x(c) \geq x_{min}$ or $x(c) \leq x_{\max}$, which is given as an interval range.

Among them, the objectives can be classified into the following five major categories according to their categories, and this feature assigns priority to each task. The constraint tasks have the highest priority, and the optimization tasks have the lowest priority:

1. Constraints (C): Objective related to the physical constraints of the system.
2. Safety (S): Objectives related to the safety of the robot.
3. Operational Prerequisite (P): Objective that is a prerequisite for the given action.
4. Action Defining (AD): Action-oriented objectives.
5. Optimization (O): Trajectory optimization objective.

Finally, the UVMS task priorities designed in Table 1 are described in conjunction with the definitions of goal types, goal categories, and control tasks elaborated above.

In Table 1, for task-related factors, the typical underwater tasks of UVMS are first decomposed into the following five basic task categories according to their requirements for autonomous underwater operations.

1. MAC—Assure safe minimum altitude control task [R, I, P]: used to keep the UVMS altitude above a certain threshold.
2. HA—Horizontal attitude task controls the horizontal attitude [R, I, S]: it is critical to maintaining the vehicle-level relative to the whole world frame < w >.
3. LA—Altitude control task, also known as the landing task [R, E, AD]: This action-defining task has the same priority as the vehicle position. The minimum task altitude is not enabled there because we need to land; therefore, UVMS needs to be below a fixed minimum altitude threshold.
4. AT—Target Control Alignment Target Task to Task Alignment [R, I, P]: this is a prerequisite task with a higher priority than the action-defining task. The error range of the inequality is equal to 0.07 m.
5. PC—The vehicle position control task [R, E, AD]: This action-defining task has a lower priority.

Next, we will explain these five types of tasks in detail regarding their task priority selection relationships and reference relationships.

Highest priority task "MAC"

The minimum altitude control task "MAC" set in this paper is the security control target. Therefore, its priority must be higher than the actions that define the objective, such as task "PC". Task "MAC" is the highest priority because avoiding collisions with the seafloor is more important than maintaining the vehicle's horizontal altitude objective during UVMS underwater tasks. This task improves the ability of the UVMS to avoid collisions with the seafloor.

The following Jacobian relationship characterizes the task "MAC":

$$^{w}\dot{x}_{mac} = J_{mac}{}^{v}\dot{y} \tag{16}$$

where $^{w}\dot{x}_{mac} \in R^6$ represents the task description, $J_{mac} \in R^{6\times 13}$ represents the Jacobian matrix of this task; J_{mac} has three rows corresponding to the dimensions of the reference velocity $(\dot{x}_{mac})$, where $\dot{x}_{mac}$ has only linear components.

$$^{w}\dot{x}_{mac} = ^{w} v_{3\times 1} = \begin{bmatrix} 0_{3\times 7} & ^{w}R_{v3\times 3} & 0_{3\times 3} \end{bmatrix}^{v} \begin{bmatrix} \dot{q}_{7\times 1} \\ v_{3\times 1} \\ ^{v}\omega_{3\times 1} \end{bmatrix} \tag{17}$$

The task "MAC" is based on an inequality objective, the main goal of which is to ensure that the vehicle maintains its altitude above a certain minimum threshold.

$$h_{\text{actual}} \geq h_{\text{min\_thresh}} \tag{18}$$

Because the control variable uses the convention [X Y Z], the task references we compute have the following structure:

$$w\dot{x}_{max} = \begin{bmatrix} 0 \\ 0 \\ ^{w}v_z \end{bmatrix} \tag{19}$$

$$w\dot{x}_{max} = k[(d_{limit} + \Delta) -^{w} d_{sensor}] \tag{20}$$

where k is the control gain, d_{limit} is the desired minimum distance from the seafloor, Δ is the safety distance at which the activation of the task starts to trigger, ${}^{w}d_{sensor}$ is the third component of the distance vector, measured by the sensor and projected on the world frame < w >.

Here, we provide a direction of the velocity we need to control the movement along the z-axis. We use the same vehicle position Jacobi matrix but only select the components associated with the z-axis. To ensure that UVMS achieves the goal of "MAC", the activation variable A_{mac} for this task have the following structure:

$$A_{mac} = \begin{bmatrix} a_{1,1} & 0 & 0 \\ 0 & a_{2,2} & 0 \\ 0 & 0 & a_{3,3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & a_{3,3} \end{bmatrix} \tag{21}$$

where $a_{1,1}$ and $a_{2,2}$ are equal to zero because the intent is to constrain the system for only the velocity in the z direction of the world frame < w >, given by Equation (21). Thus, for x and y components, the activation should always remain zero.

The desired behavior for reactive control of our inequality objective is:

1. The task should be fully active only when the inequality is false.
2. The transition for activation should be smooth.

Therefore, we use *DecreasingBellshaped* function in order to calculate $a_{3,3}$:

$$a_{3,3} \triangleq \begin{cases} 1 \quad h_{actual} < h_{\min\_thresh} \\ decreasingbell \; h_{\min\_thresh} \leq h_{actual} \\ \leq h_{\min\_thresh} + \Delta \\ 0 \quad h_{actual} > h_{\min\_thresh} + \Delta \end{cases} \tag{22}$$

We compute variable ${}^{w}v_z$ using the below equation:

$${}^{w}v_z = \lambda \left({}^{w}\bar{h} - {}^{w}h_{actual} \right) \tag{23}$$

where ${}^{w}h_{actual}$ is the distance given by the ${}^{v}d_{sensor}$ on the z-axis of the world frame < w >. This quantity represents the vehicle's distance from the seafloor seen from the vehicle itself.

Note: ${}^{y}x$, where y represents the name of the frame and x represents the vector.

The division of the minimum altitude threshold defines the interval.

- $h_{actual} < h_{\min\_thresh}$: The reference velocity (Equation 23) will be positive and will drive the robot toward $h_{\min\_thresh} + \Delta$ with activation $a_{3,3} = 1$.
- $h_{\min\_thresh} < h_{actual} < h_{\min\_thresh} + \Delta$: The reference velocity (Equation 23) will be positive and will drive the robot toward $h_{\min\_thresh} + \Delta$ with activation $a_{3,3} < 1$ and $a_{3,3} > 0$ (transition region).
- $h_{actual} > h_{\min\_thresh} + \Delta$: The reference velocity (Equation 23) will be negative, but the activation $a_{3,3} = 0$ and therefore does not have any effect on the UVMS.

TABLE 2 Comparison between behavior for different thresholds for different types of seafloor.

Type of seafloor	$h_{\min\_thresh} = 1$	$h_{\min\_thresh} = 5$	$h_{\min\_thresh} = 10$
Almost flat	Safe	Safe	Safe
Small protuberances	Not completely safe	Safe	Safe
Large protuberances	Not safe	Not always safe	Safe

As seen from the above intervals, we choose to implement a minimum altitude threshold $w\bar{h} = h_{\min\_thresh} + \Delta$, which helps us avoid over-constraining the system.

Therefore, different thresholds apply to different types of seafloor. We have simulated different values of the minimum altitude in Table 2. These values all have the same k-gain; we summarize the possible scenarios.

The *sensorDistance* we used is the distance measured by the sensor on the UVMS along the z-axis of the sensor frame < s >.

$${}^{v}d_{sensor} = \begin{bmatrix} 0 \\ 0 \\ {}^{s}sensorDistance \end{bmatrix} \tag{24}$$

However, ${}^{v}d_{sensor}$ is the distance vector measured by the sensor and projected on the vehicle frame < v >. Since we need to project it onto the world frame < w >, we apply the following rotation matrix:

$${}^{w}d_{sensor} = {}^{w}R_{v} {}^{v}d_{sensor} \tag{25}$$

To obtain the distance between seafloor and robot in the world frame < w >, we use the below equation:

$${}^{w}d_{sensor} = \begin{bmatrix} {}^{w}X - component \\ {}^{w} - component \\ {}^{w}h_{actural} \\ 0 \end{bmatrix} = {}^{w}T_{v} {}^{v}d_{sensor} \tag{26}$$

where with h_{actual} extracted by the z component of the sensorDistance projected in the world frame < w >. We assume that sensor frame < s > and vehicle frame < v > coincide.

The next highest priority "HA" and its mutually binding "PC" and "LA" tasks

The "HA" is the horizontal attitude task set in this paper and is the next highest priority task. This task ensures that UVMS does not flip for a given reference speed. Suppose we try to swap the priority of "PC" and "HA". In that case, the vehicle will try to achieve the direction of the target if its horizontal roll or vertical sway is different from zero. Therefore, when we change the priority, the behavior observed during the simulation is almost the same as when the horizontal attitude task is disabled. The horizontal attitude is not enabled, and it will appear that the UVMS is not parallel to the bottom. In addition, swapping priorities is wrong because a horizontal attitude is a safety task and should have a higher priority than the task that defines the action. As can be seen above, the multiple solutions of the higher priority task ("HA") already constrained task "PC". We give that the vehicle position control task "PC" and the horizontal attitude task "HA" are mutually constrained as follows:

$$\dot{y} = \rho_1 + Q_1 \dot{z}_1 \quad \forall \dot{z}_1 \tag{27}$$

The vehicle position control task "PC" is set in this paper. We initialize the UVMS at a place far from the seabed, i.e., at sea level, and give the target position far enough, considering the actual situation. This task aims to perform the vehicle position control task to ensure that the vehicle achieves the required position and orientation.

The Jacobian of the task "PC" is:

$$^{w}\dot{x}_2 {=}^{w}\dot{x}_{posc} = J_{posc}\ ^{v}\dot{y} \tag{28}$$

$$\dot{x}_{posc} \in R^6, J_{posc} \in R^{6\times 13},\ \dot{y} \in R^{13}$$

where the Jacobi matrix has 13 columns corresponding to the dimensions of the control variables at the kinematic level, the control variables are the seven joints of the manipulator and the six D.O.F of the vehicle base. Six rows correspond to the dimensions of the reference velocity. The time volume is the difference between the initial and target positions or λ-value.

$$^{w}\dot{x}_{posc6\times 1} = \begin{bmatrix} ^{w}v_{3\times 1} \\ ^{w}\omega_{3\times 1} \end{bmatrix} = \begin{bmatrix} 0_{3\times 7} & ^{w}R_{v3\times 3} & 0_{3\times 3} \\ 0_{3\times 7} & 0_{3\times 3} & ^{w}R_v 3\times 3 \end{bmatrix} \begin{bmatrix} \dot{q}_{7\times 1} \\ ^{v}v_{3\times 1} \\ ^{v}\omega_{3\times 1} \end{bmatrix} \tag{29}$$

Since the task "PC" is a reactive control task, the task reference is computed using the formula of the closed-loop feedback reference rate such that:

- Required position:

$$^{w}\overline{v} = \lambda_l \left(^{w}\overline{x}_{position\_goal} - ^{w}x_{actual\_position} \right) \tag{30}$$

- Required orientation:

$$^{w}\bar{\omega} = \lambda_a\ VersorLemma\ \left(^{w}\overline{x}_{orientation\_goal}, ^{w}x_{actual\_orientation} \right) \tag{31}$$

and in a compact form:

$$w\bar{x} = \begin{bmatrix} \lambda_1 & \lambda_r \end{bmatrix} \begin{bmatrix} w & r \\ ^{w}\delta & \theta \end{bmatrix} \tag{32}$$

with:

$$\begin{bmatrix} r \\ \theta \end{bmatrix} = CartError \left(^{w}\overline{x}_{position\_goal}, ^{w}x_{actual\_position} \right) \tag{33}$$

We use "*CartError*" function to calculate r and θ , and we set the two gains equal to $\lambda_l = 0.2$ and $\lambda_a = 0.5$.

The altitude control task "LA" is set in this paper, and since "LA" is an action definition (AD) target, it is placed after the security task. It is important to note that the objectives of "LA" and "PC" are so different that it is unlikely that they will be activated simultaneously. Therefore, their relative priority does not affect the solution. There are two main differences between this task and the minimum altitude control task in this paper, the first being that landing is not a safety task but rather an action that defines a safety task. While using the minimum altitude task to avoid collisions with the seafloor, the landing task "LA" defines an action as a vehicle position, therefore, has a lower priority; the second difference is that the minimum altitude is unequal to the landing is an equal task.

The Jacobian for task "LA" is:

$$^{w}\dot{x}_{la} {=}^{w}\ v_{3\times 1} = \begin{bmatrix} 0_{3\times 7} & ^{w}R_{v3\times 3} & 0_{3\times 3} \end{bmatrix} \begin{bmatrix} \dot{q}_{7\times 1} \\ v_{3\times 1} \\ ^{v}\omega_{3\times 1} \end{bmatrix} \tag{34}$$

As with the minimum altitude control objective, only the Jacobi matrix is needed to control the components along the z-axis. We calculate the task reference as:

$$^{w}\dot{\bar{x}}_{\text{land}} = k\left[\left(d_{\text{landing}} + \Delta_{\text{safeguard}} \right) - ^{w}d_{\text{sensor}} \right] \tag{35}$$

where k is the control gain, d_{landing} is the distance from the seafloor, in this case, $\Delta_{\text{safeguard}}$ is set to 0.17 m to avoid interpenetration between UVMS and the seafloor. $^{w}d_{\text{sensor}}$ is the component along the z axis of the distance vector measured by the sensor and projected on the world frame < w >.

Goal control alignment target task "AT"

If we only use the position control task "PC", we can only guarantee that we reach the target position, but not that we are aligned with the job target to complete the job task. We must add additional constraints to make the vehicle face the target task. The approach we take is to add an alignment task between the vehicle x-axis and the target. In particular, the vehicle x-axis should be aligned with the projection of the unit vector on the inertial level that connects the vehicle frame < v > to the target frame < o >.

So we add the target control alignment task "AT" to the task hierarchy. We decide to place "AT" under "PC" to take advantage of the remaining arbitrariness to align the robot in the direction we want during the target task activation phase. Before landing, the robot tries to align itself with the operational target. When it reaches sufficient alignment (θ < 0.5), the "PC" phase is activated, during which the robot reaches the target position and aligns as much as possible.

The main goal of this task is to align the vehicle, especially the direction to the target. For this purpose, we have to calculate the rotation vector ρ between the two necessary vectors.

The two vectors are:

- a: x-axis of the vehicle frame

$$^{v}a = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{36}$$

- b: A vector between the vehicle frame and target projected onto the inertial horizontal plane and expressed in the vehicle frame < v >.

$$^{v}b{=}^{v}R_{w}\left(I - kk^{\top}\right)^{w} Distance\_target \tag{37}$$

Now, we can proceed to compute ρ vector in the following way:

$$a \wedge b = n\sin(\theta) \tag{38}$$

$$\rho = n\theta \tag{39}$$

where that returns the direction n and the magnitude θ that vector a must perform to be aligned with b. Since our final goal is to have the x-axis of the vehicle aligned with the target, we have to study the behavior of this resulting vector during the time. Thus, we want:

$$\dot{x}_{ref} = \dot{\rho} = \gamma n\theta \tag{40}$$

Considering a generic observer, we have:

$$D_{\alpha}\rho = \dot{\theta}n + \theta D_{a}(n) = nn\omega_{b/\alpha} + N_{\alpha}(\theta)(\omega_{b/\alpha}, \omega_{a/\alpha}) \tag{41}$$

Considering an observer inside the rigid space of the vehicle frame < v > on which there is vector a:

$$D_{a}\rho = nn\omega_{b/a} + N_{\alpha}(\theta)\omega_{b/a} = nn\omega_{b/a} \tag{42}$$

where the second term is exactly equal to zero due to the fact that we have ρ and $\dot{\rho}$ aligned. Finally, we want:

$$\gamma n\theta = \omega_{b/a} \tag{43}$$

Since the quantity $\omega_{\mathrm{b/a}}$ is not easy to compute we can compute it using the law of addition of angular velocity vectors:

$$\omega_{\mathrm{b/a}} = \omega_{\mathrm{b/w}}\text{-}\omega_{\mathrm{a/w}} \tag{44}$$

where $\omega_{\mathrm{a/w}}$ is the angular velocity of the vehicle with respect to the world. $\omega_{\mathrm{b/w}}$ is the angular velocity given by the movement of the vehicle with respect to the target that produces a change of direction of unit vector joining the vehicle frame < v > to the target frame < o >. It is important to compute this quantities using the same observer as for ρ , computed in Equation 38, thus in our case:

It is essential to calculate this quantity using the same observer as for ρ in Equation (38), therefore, in the vehicle frame < v >.

$$^{v}\omega_{\mathrm{b/a}}{=}^{v}\omega_{\mathrm{b/w}}{-}^{v}\omega_{\mathrm{a/w}} \tag{45}$$

in particular:

$$^{v}\omega_{\mathrm{b/a}} = \left(\frac{^{v}b}{\left\|^{v}b\right\|} \wedge \frac{-^{v}vp}{\left\|^{v}vp\right\|}\right)\frac{\left\|^{v}vp\right\|}{\left\|^{v}b\right\|} \tag{46}$$

where ^{v}vp is the projection of the linear velocity of the vehicle on the inertial horizontal plane expressed in vehicle frame < v > thus:

$$^{v}vp =^{v} R_{w}\left(I - kk^{T}\right)^{w} R_{v}{}^{v}v \tag{47}$$

We can now proceed to compute the desired Jacobian matrix that, according to Equations 5 and 6, must be the following one:

$$\dot{x} = \dot{\rho} = \begin{bmatrix} 0_{3\times7} & -\frac{\|^{v}vp\|}{\|^{v}b\|}\left(\frac{^{v}b}{\|^{v}b\|} \wedge \frac{^{v}R_{w}(I-kk^{T})^{w}R_{v}}{\|^{v}v\mathbf{p}\|}\right) & 1 \end{bmatrix}\begin{bmatrix} \dot{q} \\ ^{v}v \\ ^{v}w \end{bmatrix} \tag{48}$$

Moreover, the Jacobian relationship for the alignment target control task "AT" is derived from the following formula.

$$D_{w}(p) = J_{at}\dot{y} \tag{49}$$

where p is the misalignment vector. $D_{w}(p)$ is the derivative of the misalignment vector. J_{at} is the Jacobian we want to compute. We know that

$$D_{w}(p) =^{w} \mathbf{n}_{\rho}\dot{\theta} + \underbrace{\theta D_{w}{}^{w}\mathbf{n}_{\rho}}_{\text{orthogonal}} \tag{50}$$

Since we are not interested in the orthogonal components, we can ignore them. By looking at the first term of the sum, we know that ${}^{w}\mathbf{n}_{\rho}$ is the misalignment vector projected on the world and θ can be written as

$$\dot{\theta} = {}^{w}\omega_{v} - {}^{w}\omega_{\text{goal}} \tag{51}$$

where ${}^{w}\omega_{v}$ is the angular velocity of the vehicle, referred to the world. ${}^{w}\omega_{\text{goal}}$ is the angular velocity of the projected distance between the target and the vehicle, referred to the world.

By describing ${}^{w}\omega_{v}$ in terms of $\dot{y}$, we can deduce the following Jacobian

$$\mathbf{J}_{vehicle} = \left[0_{3\times 10} \quad {}^{w}R_{v3\times 3}\right] \tag{52}$$

In order to describe ${}^{w}\omega_{\text{goal}}$ in terms of $\dot{y}$, we use the following relationship

$${}^{w}\upsilon_{v} = {}^{w}\omega_{\text{goal}} \wedge {}^{w}d \tag{53}$$

where ${}^{w}\upsilon_{v}$ is the linear velocity of the vehicle projected on the world frame $<$ w $>$.${}^{w}d$ is the projected distance on the horizontal inertial frame, between the target and the z-axis of the vehicle.

$$J_{goal} = \frac{1}{\left\|{}^{w}d^{2}\right\|}\left[{}^{w}d\wedge\right]{}^{w}\upsilon_{v} \tag{54}$$

Since we are interested only in the x and y component of ${}^{w}\upsilon_{v}$, we select such components by premultiplying as given below:

$$J_{goal} = \frac{1}{\left\|{}^{w}d^{2}\right\|}\left[{}^{w}d\wedge\right]\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}{}^{w}\upsilon_{v} \tag{55}$$

From the last equation, we deduce the following Jacobian

$$J_{goal} = \frac{1}{\left\|{}^{w}d^{2}\right\|}\left[{}^{w}d\wedge\right]\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\left[0_{3\times 7} \quad {}^{w}R_{v3\times 3} \quad 0_{3\times 3}\right] \tag{56}$$

The resulting Jacobian is obtained by substitution and it is equal to

$$J_{at} = {}^{w}n_{r}^{\top}\left[J_{vehicle} - J_{goal}\right] \tag{57}$$

We compute the task reference as:

$${}^{w}\dot{\bar{x}}_{\text{at}} = k\left(0 - \left\|{}^{w}p\right\|\right) \tag{58}$$

where k is the control gain. $\left\|{}^{w}p\right\|$ is the norm of the misalignment vector, in this case, we want it to be 0.

Trajectory planning

Trajectory optimization goal

Once the task is divided into subtasks, the placement of the job manipulator is critical because it affects the subsequent manipulation tasks. Poor essential placement may even fail to reach the final target state. A significant problem with this approach is the suboptimality of the generated solution trajectories. Even though optimal solutions can be generated for each subtask, the set of these solutions does not necessarily produce a globally optimal solution. The goal state of the previous subtask will significantly affect the planning of the next task. It may even prevent the generation of feasible solutions, resulting in the need to replan the previous task. As a result, this approach will lead to local optima, global suboptimal paths, or many unsuccessful motion planning queries. Combining the system with high-degree-of-freedom maneuver planning for the entire task can alleviate the suboptimal global problem, but this requires extensive computation. Motion coordination between the vehicle and the manipulator, collision checking and self-collision checking with the environment, and motion constraints are some added complexities in this approach.

In this research, the trajectory optimization objective is the lowest priority to address this issue. After the vehicle has completed all priority tasks, we focus on considering the desired trajectory of the end-effector on the UVMS. We convert the trajectory planning problem into finding feasible joint trajectories considering the priority tasks first while optimizing the cost function given by the expectation. In velocity-resolved inverse kinematics, the task is the expectation of the robot configuration function, represented in the task description by an equation or inequality constraint. Finally, the trajectory planning problem for the end-effector can be formulated as the following optimization problem.

The optimized smooth trajectory needs to consider its boundary conditions, including the start and end states, the relay node as the waypoint through which the robot passes, and the smoothing criterion to evaluate whether the generated trajectory is smooth. Knowing the angles to be reached by M joint, a polynomial fit will result in segment M-1 trajectories, each represented by a polynomial, and the set of trajectories needs to satisfy the following constraints:

- Desired angle constraint:

$$\begin{cases} f_{j}^{(k)}\left(T_{j-1}\right) & = x_{0,j}^{(k)} \\ f_{j}^{(k)}\left(T_{j}\right) & = x_{T,j}^{(k)} \end{cases} \tag{59}$$

- Continuity constraint:

The velocity and acceleration of adjacent trajectories are continuous:

$$f_j^{(k)}\left(T_j\right)=f_{j+1}^{(k)}\left(T_j\right) \tag{60}$$

The cost function is chosen to minimize the Snap value for all trajectories. Snap is the fourth-order derivative of position, and minimizing Snap allows the end-effector to meet the autonomous operational movement suitable for UVMS. At the same time, its kinetic states, such as velocity and acceleration, cannot change abruptly. Reducing the range of acceleration and deceleration enables UVMS to work longer in energy-limited underwater environments.

The cost function determined to minimize snap is expressed as follows:

$$\begin{aligned} J(T) &= \int_{T_{j-1}}^{T_j} (f^4(t))^2\, dt \\ &= \sum_{i\geq 4, l\geq 4} \frac{i(i-1)(i-2)(i-3)j(l-1)(l-2)(l-3)}{i+l-7} \\ &\quad \times (T_j^{i+l-7} - T_{j-1}^{i+l-7}) p_i p_j \end{aligned} \tag{61}$$

The coefficients of each order are extracted separately, and the cost function can be written in the quadratic form:

$$J_j(T) = \mathbf{p}_j^T \mathbf{Q}_j \mathbf{p}_j \tag{62}$$

where $\mathbf{Q}_j$ is the Hessian matrix that transforms the trajectory optimization problem into a quadratic programming problem, for the final trajectory of the manipulator, each trajectory point should satisfy the following constraints.

Track point constraint, each trajectory should pass through the track point obtained by the path search, and the displacement, speed, acceleration, jerk, and snap at the track point should all exist. To satisfy $f_\mu^{(k)}(T_i) = d_{ik}$, where $\mu \in \{x, y, z\}, k \in \{0, 1, 2, 3\}, i \in \{1, 2, \cdots, M\}$ continuity constraints, the displacement, velocity, acceleration, jerk, and snap at the track point should also be continuous. That is satisfied $f_\mu^{(k)}(T_i) = d_{ik}$, where $\mu \in \{x, y, z\}, k \in \{0, 1, 2, 3\}, i \in \{1, 2, \cdots, M\}$. It can be written as an equality constraint.

We need to fix each trajectory time $T_j = 0.1s$ for all joints. Ensure that all robot joints reach the desired angle and end position at the same moment is always stable.

Summarizing the above constraints and cost functions, they are written in matrix form.

- Desired angle constraint:

$$f_j^{(k)}\left(T_j\right) = x_j^{(k)} \tag{63}$$

$$\Rightarrow \sum_{i\geq k} \frac{i!}{(i-k)!} T_j^{i-k} p_{j,i} = x_{T,j}^{(k)} \tag{64}$$

$$\Rightarrow \begin{bmatrix} \ldots \cdots & \frac{i!}{(i-k)!} T_j^{i-k} \cdots & \ldots \end{bmatrix} \begin{bmatrix} \vdots \\ p_{j,i} \\ \vdots \end{bmatrix} = x_{T,j}^{(k)} \tag{65}$$

$$\Rightarrow \begin{bmatrix} \frac{i!}{(i-k)!} T_{j-1}^{i-k} \\ \vdots \\ \vdots \\ \frac{i!}{(i-k)!} T_j^{i-k} \end{bmatrix} \begin{bmatrix} \vdots \\ p_{j,i} \\ \vdots \end{bmatrix} = \begin{bmatrix} x_{0,j}^{(k)} \\ x_{T,j}^{(k)} \end{bmatrix} \tag{66}$$

$$\Rightarrow \mathbf{A}_j \mathbf{p}_j = \mathbf{d}_j \tag{67}$$

- Continuity constraint: smoothness constraint ensures continuity between trajectory segments without giving a specific derivative.

$$f_j^{(k)}\left(T_j\right) = f_{j+1}^{(k)}\left(T_j\right) \tag{68}$$

$$\Rightarrow \sum_{i\geq k} \frac{i!}{(i-k)!} T_j^{i-k} p_{j,i} - \sum_{l\geq k} \frac{l!}{(l-k)!} T_j^{l-k} p_{j+1,l} = 0 \tag{69}$$

$$\Rightarrow \begin{bmatrix} \cdots & \frac{i!}{(i-k)!} T_j^{i-k} & \ldots & \ldots & -\frac{l!}{(l-k)!} T_j^{l-k} & \cdots \end{bmatrix} \begin{bmatrix} p_{j,i} \\ \vdots \\ \vdots \\ p_{j+1,l} \end{bmatrix} = 0 \tag{70}$$

$$\Rightarrow \begin{bmatrix} \mathbf{A}_j & -\mathbf{A}_{j+1} \end{bmatrix} \begin{bmatrix} \mathbf{p}_j \\ \mathbf{p}_{j+1} \end{bmatrix} = 0 \tag{71}$$

Substitution function:

$$\begin{aligned} J(T) &= \textstyle\int_{T_{j-1}}^{T_j} \left(f^4(t)\right)^2 dt \\ &= \begin{bmatrix} \vdots \\ p_i \\ \vdots \end{bmatrix}^T \begin{bmatrix} \cdots \frac{i(i-1)(i-2)(i-3)l(l-1)(l-2)(l-3)}{i+l-7} T^{i+l-7} \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ p_l \\ \vdots \end{bmatrix} \end{aligned} \tag{72}$$

$$J_j(T) = \mathbf{p}_j^T \mathbf{Q}_j \mathbf{p}_j \tag{73}$$

Writing the above problem as an equation constraint in standard form, then the quadratic programming problem can be expressed as:

$$\min \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_M \end{bmatrix}^T \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \vdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_M \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_M \end{bmatrix} \tag{74}$$

$$s.t. \quad \mathbf{A}_{\text{eq}} \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_M \end{bmatrix} = \mathbf{d}_{eq} \tag{75}$$

The above equation is a linear constraint quadratic programming problem (QP).

Collision avoidance

UVMS requires only six degrees of freedom to reach an arbitrary position in underwater motion. Adding a manipulator gives the entire system more than six degrees of freedom, resulting in the redundancy of degrees of freedom. Kinematic redundancy allows the planner to satisfy additional constraints, such as collision avoidance. Researchers have mainly focused on approximating the robot or the obstacle with strictly convex targets and considering only the closest points in the detection algorithm to avoid collisions to reduce computational effort.

The optimization process has no environmental constraints after the trajectory planning solution based on the minimum snap principle. When a new trajectory is encountered after optimization, the obstacles force the trajectory to be modified again, wasting computational resources and reducing the planning frequency. For example, it is necessary to add constraints on the environment during optimization, generally based on hard constraint solving. The hard constraint solution is to generate a safe region in the environment by extending the algorithm and using it as a hard constraint. Adding hard constraints in the optimization process forms a convex polygon, which transforms the QP problem into a convex optimization problem that can be solved by convex optimization algorithms such as the interior point method.

While the process of underwater obstacle avoidance, most of the surrounding objects are non-strictly convex polyhedra, and these approximation methods are not accurate enough when operating in close range. The problems in the practical application process are ignored. Because the remaining safe regions are treated equally during optimization, there is no good way to handle the extreme cases with underwater sensor noise. The optimized trajectory may go past the edge of the safety zone. Once the controller makes an error, it leads to a severe failure of the manipulator body by colliding with the internal and external environment. Inspired by the penalty function, we propose a more intuitive collision-free motion planning method oriented to UVMS.

An improved collision avoidance method based on soft constraint

By design, we use the principle of soft constraint to improve the collision avoidance method. The essence of the soft constraint method is to apply a "pushing force" to push the trajectory away from the direction of the obstacle. The core problem is the designed objective function. When the objective function is not set correctly, the path may hit an obstacle, which is the shortcoming of soft constraint. Therefore, a gradient-based optimization algorithm sets the objective function to impose a soft constraint on the underwater manipulator to push the underwater manipulator body away from the obstacle.

For Equation 16, the objective function becomes:

$$J = J_s + J_c + J_d = \lambda_1 J_1 + \lambda_2 J_2 + \lambda_3 J_3 \tag{76}$$

$$J_S = \sum_{\mu \in \{x,y,z\}} \int_0^T \left(\frac{d^k f_\mu(t)}{dt^k} \right) dt \tag{77}$$

$$\begin{aligned} &\begin{bmatrix} d_F \\ d_P \end{bmatrix}^T C^T M^{-T} Q M^{-T} C \begin{bmatrix} d_F \\ d_P \end{bmatrix} \\ &= \begin{bmatrix} d_F \\ d_P \end{bmatrix}^T \begin{bmatrix} R_{FF} & R_{FP} \\ R_{PF} & R_{PP} \end{bmatrix} \begin{bmatrix} d_F \\ d_P \end{bmatrix} \end{aligned} \tag{78}$$

where the smoothness cost function J_s is the cost of smoothness generated using minimum-snap.

$$\begin{aligned} J_c &= \int_{T_0}^{T_M} c(p(t)) ds \\ &= \sum_{k=0}^{T|\delta t} c(p(T_k)) \| \left\| v(t) \right\| \| \delta t, T_k = T_0 + k\delta t \end{aligned} \tag{79}$$

where the collision cost function J_c, i.e., the collision cost, penalizes obstacles that are too close.

where the kinetic cost function J_d penalizes exceeding the kinetic constraints. Since the objective function of penalizing the velocity and acceleration is not a convex function, it needs to be solved by step-by-step derivation. The smooth term solution is shown in the previous derivation, and the relationship between the collision term and the free variables $d_{p\mu}$ is as follows.

$$\begin{aligned} J_c &= \int \{ \sum_{k=0}^{T|\delta t} \{ \forall_\mu c(p(T_k)) \| \|v\| \| F + c(p(T_k)) \frac{v_\mu}{\| \left\| v \right\| \|} G \} \delta t d_{p\mu} \}, \\ &\mu \in \{x, y, z\} \end{aligned} \tag{80}$$

where the F and G are, respectively:

$$F = TL_{dp}, \quad G = TV_m L_{dp} \tag{81}$$

where L_{dp} is the right half of the matrix $M^{-1}C$, V_m is the mapping matrix of joint position variables to joint velocity variables, $T = \left[T_k^0, T_k^1, \ldots, T_k^n \right]$.

The second-order derivative results in:

$$\mathbf{H}_o = \left[\frac{\partial^2 f_o}{\partial \mathbf{d}_{P_x}^2}, \frac{\partial^2 f_o}{\partial \mathbf{d}_{P_y^2}}, \frac{\partial^2 f_o}{\partial \mathbf{d}_{P_z^2}} \right]$$

$$\begin{aligned} \frac{\partial^2 f_o}{\partial \mathbf{d}_{P\mu}^2} &= \sum_{k=0}^{\tau/\delta t} \{ \mathbf{F}^T \nabla_\mu c(p(T_k)) \frac{v_\mu}{\|v\|} \mathbf{G} + \mathbf{F}^T \nabla_\mu^2 c(p(T_k)) \|v\| \mathbf{F} \\ &+ \mathbf{G}^T \nabla_\mu c(p(T_k)) \frac{v_\mu}{\|v\|} \mathbf{F} + \mathbf{G}^T c(p(T_k)) \frac{v_\mu^2}{\|v\|^3} \mathbf{G} \} \delta t \end{aligned} \tag{82}$$

TABLE 3 Control tasks and priorities for fixed base manipulation operations.

Priority	Category	Description	Objective
1	Constraint	[NR,E,C]	NR
2	Safety	[R,I,S]	MAC
3	Prerequisite	[R,I,P]	HA
4	Action-defining	[R,E,AD]	LA
5	Prerequisite	[R,E,P]	AT
6	Action-defining	[R,E,AD]	PC

[R/NR, I/E, C/S/P/AD] Name of the task/objective.

TABLE 4 Examples of external activation states for different tasks.

Priority	Tasks	Way point	Alignment	Landing	Tool frame
1	NR	0	0	0	1
2	MAC	1	1	0	0
3	HA	1	1	1	1
4	LA	0	0	1	0
5	AT	0	1	1	0
6	PC	1	0	0	0

0/1 External inactive/active.

Simulation results

In this section, the task priority processing strategy and soft-constrained trajectory optimization objective designed in this paper are verified on a kinematically redundant underwater vehicle manipulator system. The system consists of a free-floating underwater vehicle and a seven-function manipulator. The simulation is performed in a MATLAB/Simulink environment.

Case 1: Test the given tasks

The simulation phase first initializes the UVMS for safe waypoint navigation. Afterward, we move the vehicle to a position close to the currently defined end-effector target position, slightly above the target position. Finally, the action change is triggered, and the UVMS executes the ocean float operation. We do not consider any disturbances and assume that the robot can provide the desired speed without delays. In addition, once the robot reaches the desired position, the rest of the tasks are responsible for the end-effector reaching the desired target position and orientation, so the "PC" task will also be closed. For the end-effector to operate as a stationary-based robot, we need to constrain the vehicle not to move. Because, as we noticed, the vehicle will "help" the arm to reach the desired position by moving itself (in line with the expected behavior of the tool task). To avoid this problem, we need to perform a non-reactive task to constrain the vehicle to move or reach the operating position where the task will make the vehicle move. In this case, we test UVMS landing on the seafloor, try the vehicle to its target coordinate system, and then use the end-effector to reach the operational target position. Observe whether the vehicle does not move and perform the ocean float operation of the UVMS.

The uniform hierarchy of tasks we use and their priorities, with the addition of constrained tasks at the priority level, is described in Table 3, with non-reactive tasks ("NR") added at the top of the hierarchy to constrain the vehicle not to move.

We have the following tasks in an active/inactive state for each of the different phases, is described in Table 4.

We conducted a multitask prioritization strategy experiment to explain the task prioritization strategy better. First, multitasking is divided into multiple action phases.

1. Action A, safe waypoint navigation with all the safety tasks enabled. This action finishes when the position error is below a fixed threshold (in this case 0.1 m), as Figure 5 shows.
2. Action B, alignment to the nodule with all the safety tasks enabled. This action finishes when the misalignment error is below a fixed threshold (in this case 0.07 m).
3. Action C, landing, and smooth rotation align with the target. This action finishes when the vehicle touches the seafloor (in the simulation, this happens at approximately 0.17 m).
4. Action D, manipulator actuation after landing. In this action, Vehicle Null Velocity task is enabled, preventing vehicle movements. The only movement will be the extension of the manipulator to reach the desired target position.

From Figure 6, we observe that the position and orientation errors of the vehicle base remain almost constant after 30 s of simulation when the UVMS has completed the landing phase and started the tool holder phase. Therefore, the tool frame phase's active task "NR" helps us achieve a fixed datum operation. Because after a reasonable time, the position error r and orientation error θ converge to near zero. When the error of the carriage position remains constant, the tool frame error of the manipulator operation is almost zero. Action A's fixed threshold of position error is within 0.1 m. Action B sets the fixed maximum of unaligned error by 1.3 mm, well below the set fixed threshold of 0.07 m, which provides the basis for the subsequent accurate completion of the operational target. Figure 7 shows the corresponding simulation results. Figure 8 shows that the maximum range of vehicle error aligned with the target is 0.17 m

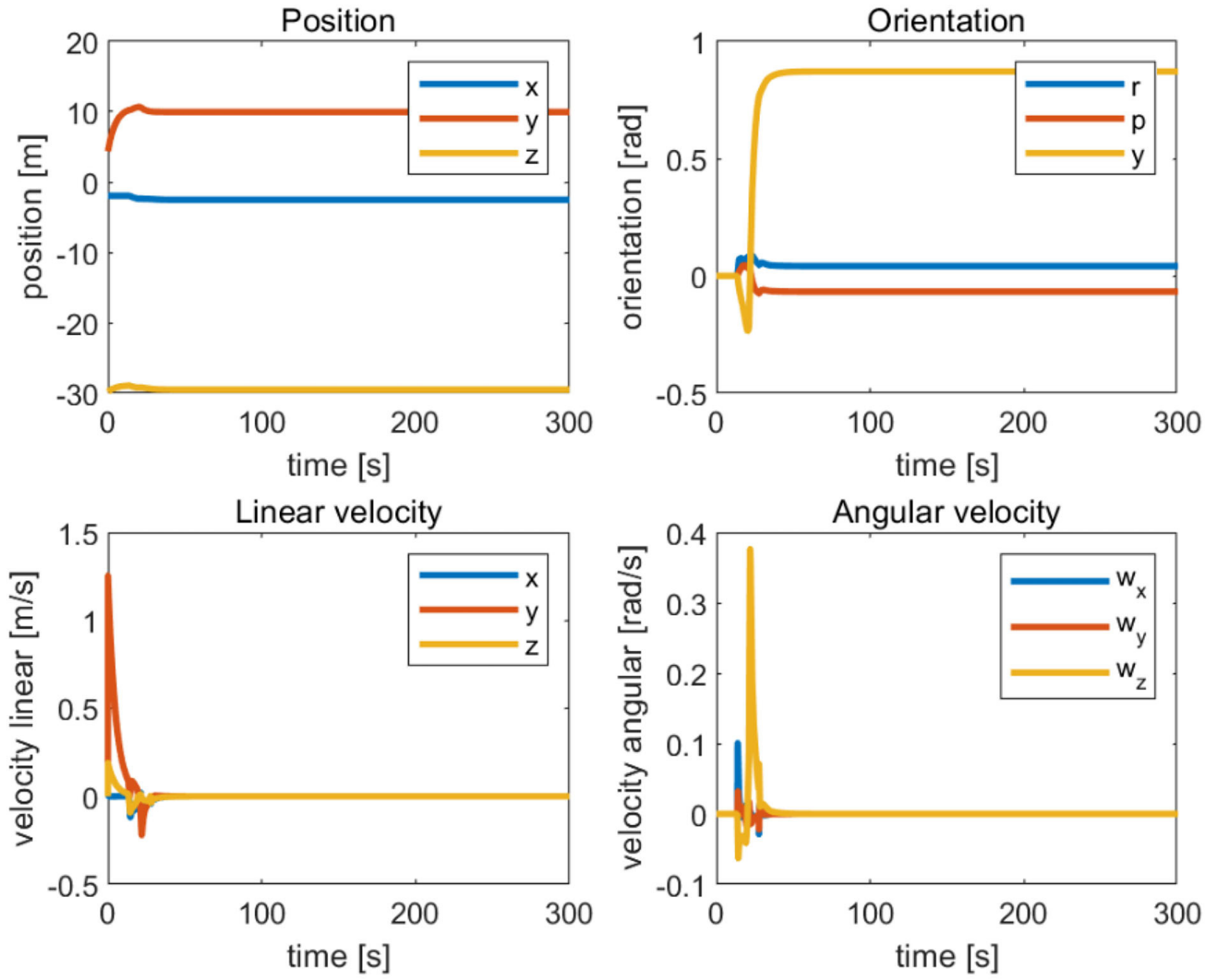


FIGURE 5
Position-directional linear velocity angular velocity tracking curve of the completed vehicle initialization.

in Action C. The Action ends when the vehicle touches the seabed, i.e., when the height is 0.

Since the Vehicle Null only puts vehicle velocities to zero, it cannot compensate for the disturbance. In a similar scenario, the currents will influence the UVMS: the vehicle can arrive at the desired position and land if the disturbance is not too big. However, when the vehicle Null task is triggered in Action D, the UVMS will drift, eventually losing its target position. The manipulator will keep trying to reach the goal by stretching as much as possible. With a check on the position error during the mission phase update, it is possible to return to Action A and achieve the desired position again. Then the task phase update transition requires a new command relationship to achieve it. Therefore, the transition from one activity to another when completing a given job task using the "task" "phase" variable in the Matlab structure. The "phase" variable is updated when the previous Action completes the desired precision. The task update phase starts when the UVMS approaches the desired M path point navigation target position. The "UpdateMissionPhase" of each loop performed the phase update condition check.

As shown in Figure 9, by taking into account the buffer time, achieving a seamless transition (from one activity to another) is by using a bell curve (increasing or decreasing) activation. The transition triggers Actions A and B by realizing the vehicle's target position. We calculate the Cartesian error between the target frame and the vehicle frame, and the task phase changes when the error is below a given threshold (0.1 m in this case). When we want to disable the running task, use a decreasing bell function to perform a smooth transition. In this case, the minimum height and vehicle position tasks are disabled at the beginning of the second phase.

Similarly, when we want to activate the task, a smooth transition is performed using an increasing bell function, as

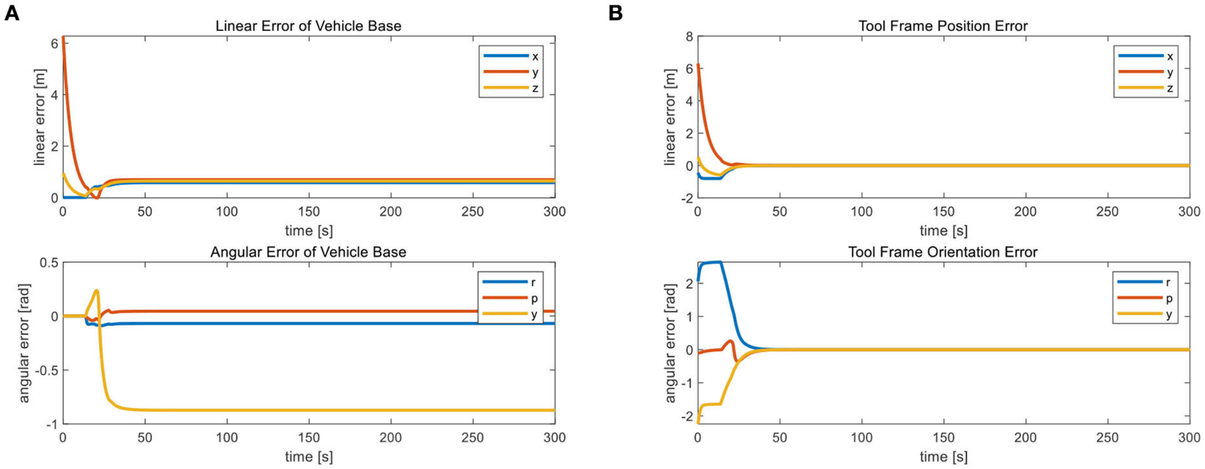


FIGURE 6
(A) The change curve of vehicle base error during the test and **(B)** the tool frame error.

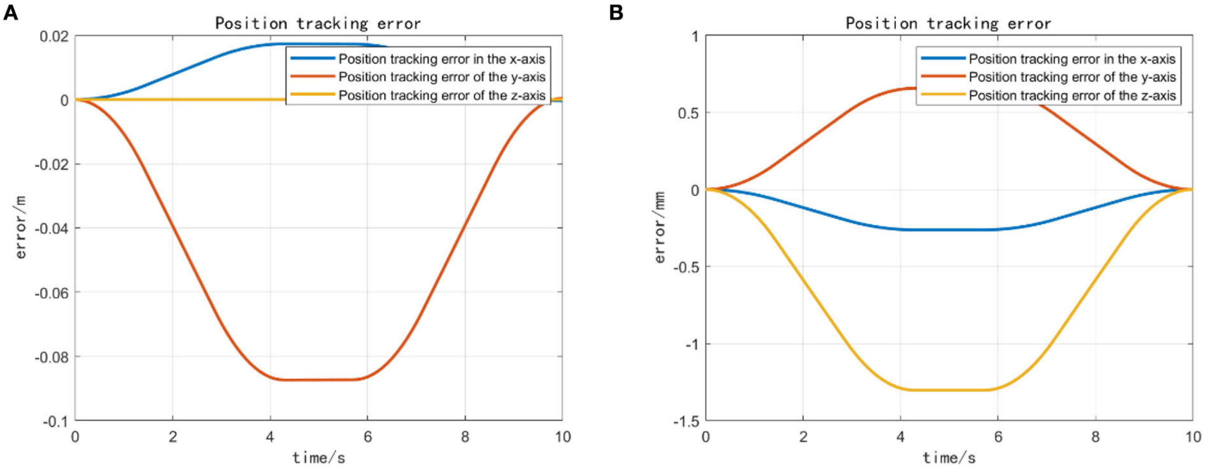


FIGURE 7
(A) The vehicle error completed for action A and **(B)** the unaligned error vehicle error completed for action B.

we did for the landing task. We calculate the slope of such a function based on the task phase time, which helps us determine the time interval between vehicles switching from one activity to another (0.2 s in our case). At the same time, we obtain a smoother shape and avoid discontinuities in the motor drive. Due to the active hold state of the landing task "LA" and the horizontal attitude task "HA", the vehicle's altitude and velocity in the Z-axis direction remain constant during this process and continue to be 0. Figure 9B indicates that the vehicle does not float with the external during the task transition. The task prioritization strategy has a strong constraint, proving its stability.

Case 2: Add an optimal control target

After the task transition is complete, begin completing a joint limits avoidance task. Attempt to reach the specified operating position using the end-effector. Moreover, we observe that the vehicle does not move and that all joints are within their soft restraint limits. This task is a safety task, so it has a higher priority than other tasks that define movements. It can control the operation of the joints without exceeding their fixed thresholds. The action is the same as before; the only difference is that the joint restraint is always active, as this is a safety task. It is essential to ensure that the end-effector performs the final

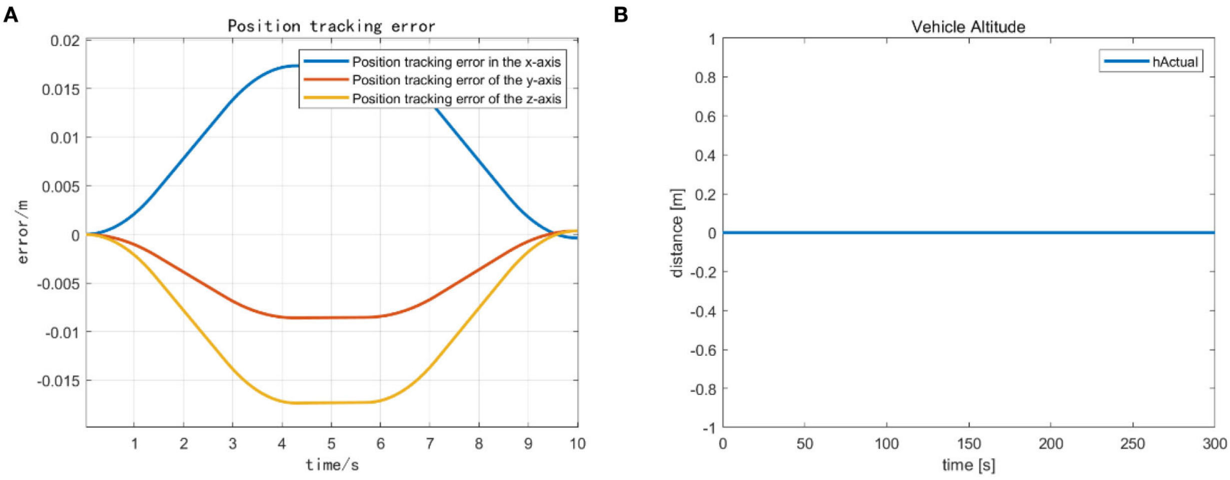


FIGURE 8
(A) The vehicle error for the completion of action C. **(B)** The smooth landing state of the vehicle after action C.

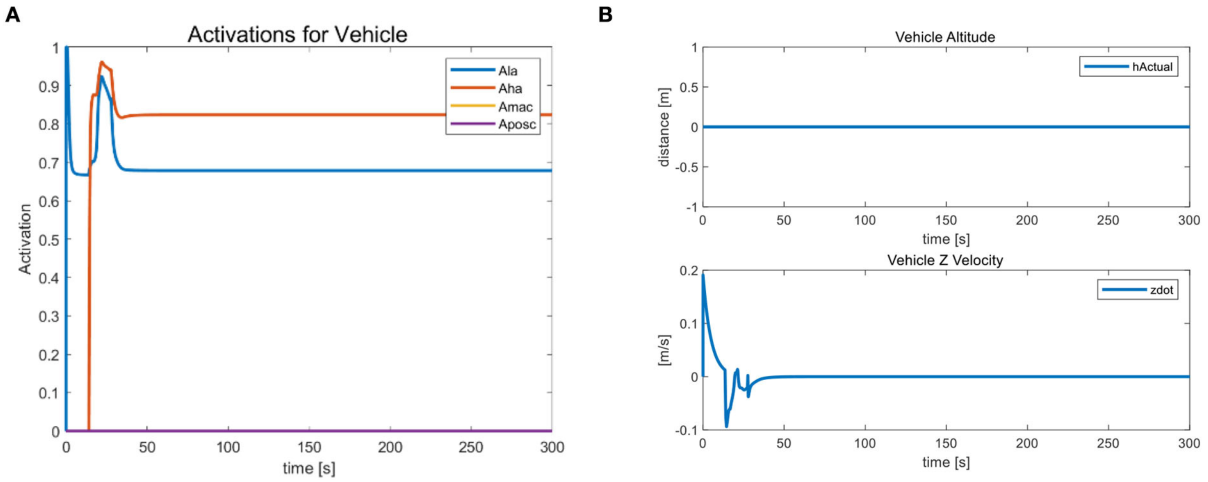


FIGURE 9
(A) The task activation status used for action transitions and **(B)** the behavior of the landing task.

operation. This case adds the optimization goal of keeping the four joints of the manipulator's end-effector to complete the trajectory optimized for the target behavior. For the rest of the tasks, we kept the same hierarchy as in the previous case and added only the optimization task "MP". This task has the lowest priority because we can only perform trajectory optimization of the end-effector after the UVMS completes all actions.

We mainly activate the state of the four joints near the end of the end-effector, as shown in Figure 10A, and limit the motion of the remaining joints. The designation of the joint limit task is to test whether we can effectively control the activation state of each joint and motion-tracked it in real-time during the vehicle manipulation task. Figure 11 shows that while the end-effector optimization task is active, the horizontal attitude task "HA" is kept highly active to maintain the stability of the vehicle position. Figure 12 shows the optimal end-effector trajectory based on the trajectory optimization objective. The trajectory optimization is performed based on the satisfaction of the proposed constraints, and the desired optimal trajectory of the end-effector coincides with the trajectory tracking as much as possible. We ensure the accuracy and idealization of the task execution. In addition, Figure 12A shows the joint motion and Figure 12B is smoother compared to Figure 10B. The optimal solution here satisfies the primary collision-free motion task, meaning that the solution found here satisfies the primary collision-free task but is optimal compared to the suboptimal pose task. Therefore, the pure QP

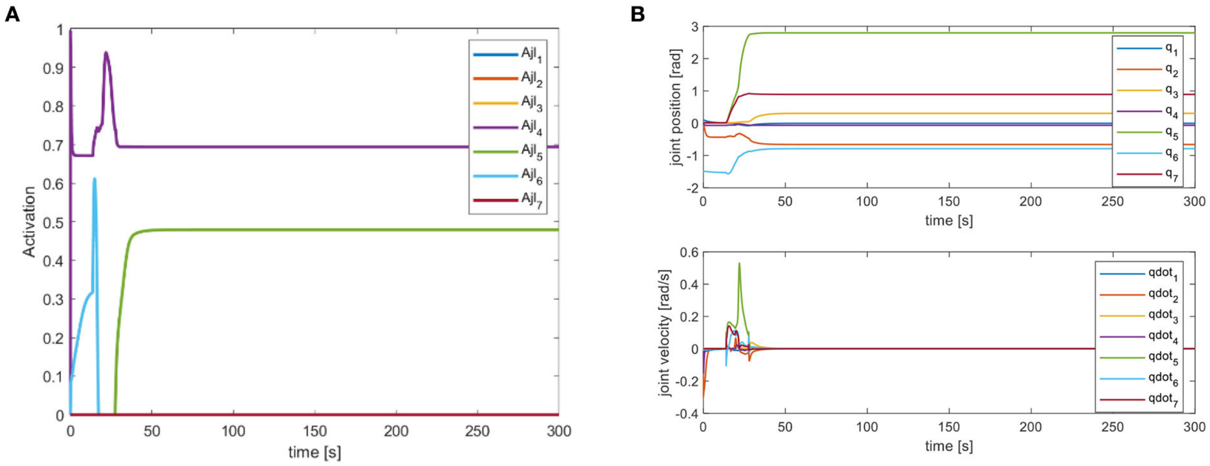


FIGURE 10
(A) Joint limitation task activation of action D. **(B)** Robot arm joint position and velocity variation curve.

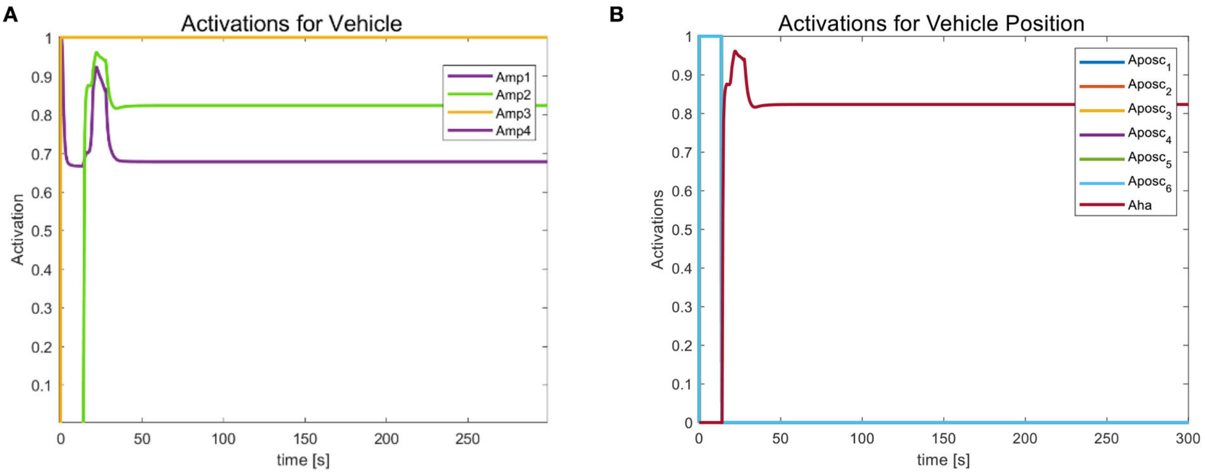


FIGURE 11
(A) Optimized task activation of the end-effector. **(B)** Task activation of the vehicle position.

process cannot handle more than one task simultaneously, so we have placed the optimization task at the lowest priority.

Conclusions

This paper studied the problem of multiple-task planning from the motion planning level for the underwater vehicle manipulator system. The task prioritization strategy to perform various tasks at once is considered in the mission planning to derive an optimal and feasible planning scheme; secondly, the optimization algorithm is adopted during the execution of the tasks, considering the system's limitations and the interference of the environment. We proposed soft constraints as an improved collision avoidance method to add more conditions to smooth the joint trajectory. The combination of the above two aspects can achieve the continuous planning of the phased execution of the task, ensure the stability of the end-effector work, and improve the reliability of UVMS autonomous underwater operations. We perform a series of simulations in a simulation environment established by kinematic and dynamic analysis of the underwater vehicle manipulator system. The simulation results verify the effectiveness and feasibility of this paper's task prioritization processing strategy. In this sense, we believe that the approach of using a simulation environment instead of a natural underwater application environment proves to be cost-saving in planning and effective in improvement. And in this

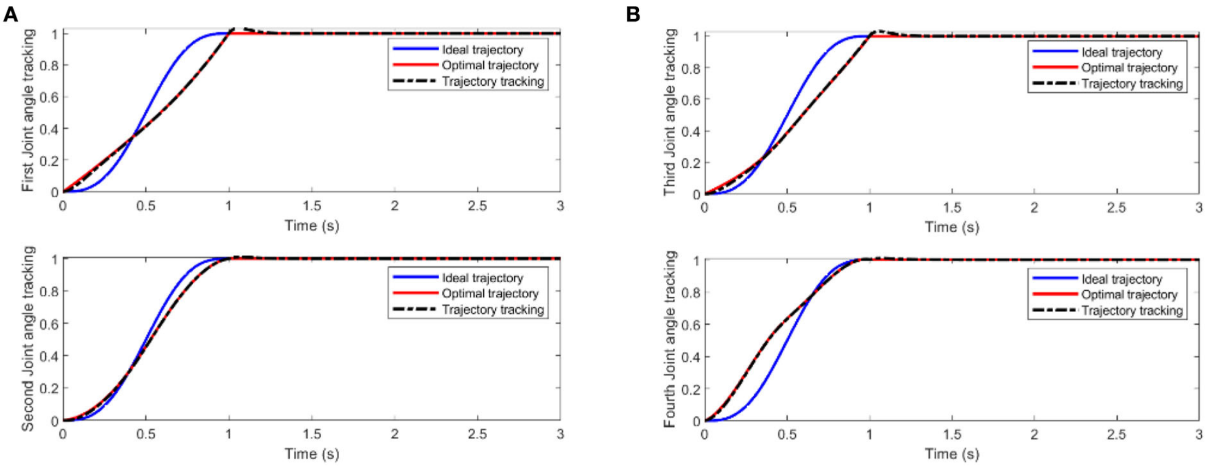


FIGURE 12
(A) Two-joint angular tracking of end-effector. **(B)** Two-joint angular tracking of end-effector.

sense, we believe that our work has achieved some progress in extending the scope of applying task prioritization planning methods based on the motion planning level in control studies of UVMS. In addition, our proposed method provides an optional way of thinking for controlling underwater robots and other types of robots. In the future, we plan to extend the research for different kinds of robots for real-time planning.

Indeed, the current research has its limitations, along with some results. In the algorithm proposed in this research, we run the simulations carried out under ideal conditions, so it needs to complete realistic experiments to verify the correctness and feasibility of the proposed method. In addition, more influencing factors should be considered, such as the reliability of the sensor, actuator, and controller execution methods. Moreover, adding and improving the controller's performance and stability to accomplish the smooth execution of the task is an essential topic for further research. Ultimately, UVMS-related research has broad application background and important theoretical and engineering significance. Our proposed method will be applied to UVMS for autonomous motion planning in unknown sea environments to enhance its subsea operation capability and meet the application requirements of keeping up with the times.

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.

Author contributions

Conceptualization: Y-eG and XZ. Methodology, writing—original draft preparation, writing—review and editing, software, formal analysis, and data curation: Y-eG. Validation: WB, JW, and QY. Supervision, project administration, and funding acquisition: WB, SY, and YS. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by the Basic Research in Natural Science and Enterprise Joint Fund of Shaanxi (No. 2021JLM-58).

Conflict of interest

All authors were employed by company Hanjiang-Weihe River Valley Water Diversion Project Construction Co., LTD.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Antonelli, G., and Chiaverini, S. (1998). Task-priority redundancy resolution for underwater vehicle-manipulator systems. *IEEE International Conference on Robotics and Automation*. p. 768–773.

Antonelli, G., and Chiaverini, S. (2000). Fuzzy approach to redundancy resolution for underwater vehicle-manipulator systems. *Control Eng. Pract.* 11, 445–452. doi: 10.1016/S0967-0661(02)00319-2

Bae, J., Bak, J., Jin, S., Seo, T., and Kim, J. (2018). Optimal configuration and parametric design of an underwater vehicle manipulator system for a valve task. *Me chanism and Machine Theory* 123, 76–88. doi: 10.1016/j.mechmachtheory.2018.01.014

Casalino, G., Zereik, E., Simetti, E., Torelli, S., Sperinde, A., Turetta, A. (2012). A task and subsystem priority based control strategy for underwater floating manipulators. *IFAC Proceedings Volumes* 45, 170–177. doi: 10.3182/20120410-3-PT-4028.00029

Chang, W. (2004). *Research on Control and Coordination of Multiple AUVs Based on Neural Network.* Harbin: Harbin Engineering University.

Changmi, X. (2022). *Dynamics Modeling and Motion Control of Underwater Robot-Manipulator System*. Ocean University of China. doi: 10.7666/d.y1827957

Cieslak, P., Ridao, P., and Giergiel, M. (2015). Autonomous underwater panel operation by GIRONA500 UVMS: A practical approach to autonomous underwater manipulation. *IEEE International Conference on Robotics and Automation (ICRA).* doi: 10.1109/ICRA.2015.7139230

Conti, R., Meli, A., Ridolfi, A., and Allotta, B. (2015). An innovative decentralized strategy for I-AUVs cooperative manipulation tasks. *Rob. Auton. Syst.* .

Fujie, Y., Yuan,C., and Qingzhong, L. (2020). Coordinated motion planning based on virtual workspace constraints for AUVMS. *J. Mechanical Eng.* 56, 249–64. doi: 10.3901/JME.2020.12.249

Gancet, J., Weiss, P., Antonelli, G., Pfingsthorn, M. F., Calinon, S., Turetta, A., et al. (2016). Dexterous undersea interventions with far distance onshore supervision: the DexROV project. *IFAC Papersonline* 49, 414–419. doi: 10.1016/j.ifacol.2016.10.439

Han, J., and Chung, W. K. (2008). "Coordinated motion control of Underwater Vehicle-Manipulator System with minimizing restoring moments," *IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE.* p. 3158–3163.

Han, J., Park, J., Chung, W. K. (2011). Robust coordinated motion control of an underwater vehicle-manipulator system with minimizing restoring moments. *Ocean Eng.* 38, 1197–1206. doi: 10.1016/j.oceaneng.2011.05.014

Heshmati-Alamdari, S., Bechlioulis, C. P., Karras, G. C., et al. (2018). A robust interaction control approach for underwater vehicle manipulator systems. *Annu. Rev. Control.* 46, 315–325. doi: 10.1016/j.arcontrol.2018.10.003

Huang, H., Zhang, G. C., and Yang, Y. I. (2016). Dynamics modeling and coordinated motion trajectory optimization of underwater unmanned vehicle manipulator system. *J. Shanghai Jiaotong Univ.* 50, 1437–1443.+1451. doi: 10.16183/j.cnki.jsjtu.2016.09.015

Jin, Y. Y. (1996). Numerical methods for generalized least squares problems - ScienceDirect. *J. Comput. Appl. Math.* 66, 571–584. doi: 10.1016/0377-0427(95)00167-0

Kanoun, O., Lamiraux, F., and Wieber, P. B. (2011). Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks. *IEEE Transactions on Robotics* 27, 785–792. doi: 10.1109/TRO.2011.2142450

Lynch, B., and Ellery, A. (2014). Efficient control of an AUV-manipulator system: an application for the exploration of Europa. *IEEE J. Oceanic Eng.* 39, 552–570. doi: 10.1109/JOE.2013.2271390

Mansard, N., and Chaumette, F. (2007). Task sequencing for high-level sensor-based control. *IEEE Trans. Robot.* 23, 60–72. doi: 10.1109/TRO.2006.889487

Mansard, N., Khatib, O., and Kheddar, A. (2009). *Integrating Unilateral Constraints Inside the Stack of Tasks*. doi: 10.1109/TRO.2009.2020345

Moe, S., Antonelli, G., and Pettersen, K. Y. (2014). Null-space-based behavior guidance of planar dual-arm UVMS. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. doi: 10.1109/ROBIO.2014.7090419

Nakamura, Y., Hanafusa, H., Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *Int. J. Robot. Res.* 6. doi: 10.1177/027836498700600201

Olguin-Diaz, E., Arechavaleta, G., Jarquin, G., et al. (2013). A passivity-based model-free force–motion control of underwater vehicle-manipulator systems. *IEEE Transact. Robot.* 29, 1469–1484. doi: 10.1109/TRO.2013.2277535

Podder, T., and Sarkar, N. (2000). "Dynamic trajectory planning for autonomous underwater vehicle-manipulator systems", *international conference on robotics and automation* 2000. p. 3461–3466. doi: 10.1109/ROBOT.2000.845264

Prats, M., Perez, J., Fernandez, J. J., Sanz, P. (2012). "An open source tool for simulation and supervision of underwater intervention missions", *IEEE/RSJ International Conference on Intelligent Robots and Systems.* p. 2577–2582. doi: 10.1109/IROS.2012.6385788

Sanz, P. J., Prats, M., Ridao, P., Ribas, D., Oliver, G., Ortiz, A. (2011). *Recent progress in the RAUVI project: A Reconfigurable autonomous underwater vehicle for intervention. Elmar*. Zadar, Croatia: IEEE.

Sanz, P. J., Ridao, P., Oliver, G., Melchiorri, C., Casalino, G., Silvestre, C., et al. (2010). TRIDENT: A framework for autonomous underwater intervention missions with dexterous manipulation capabilities. *IFAC Proceedings Volumes* 43, 187–192. doi: 10.3182/20100906-3-IT-2019.00034

Sentis, L., and Khatib, O. (2005). "Control of free-floating humanoid robots through task prioritization," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (IEEE)*, 1718–1723. doi: 10.1109/ROBOT.2005.1570361

Siciliano, B., and Slotine, J. A. (1991). "General framework for managing multiple tasks in highly redundant robotic systems," in *International Conference on Advanced Robotics. IEEE.* doi: 10.1109/ICAR.1991.240390

Simetti, E., and Casalino, G. (2016). Manipulation and transportation with cooperative underwater vehicle manipulator systems. *IEEE J. Oceanic Eng.* 99, 1–18. doi: 10.1109/JOE.2016.2618182

Simetti, E., Casalino, G., Wanderlingh, F., Aicardi, M. (2018). Task priority control of underwater intervention systems: theory and applications. *Ocean Eng.* 164, 40–54. doi: 10.1016/j.oceaneng.2018.06.026

Sotiropoulos, P., Kolonias, V., Aspragathos, N., and Housus, E. (2015). Rapid motion planning algorithm for optimal UVMS interventions in semi-structured environments using GPUs. *Rob. Auton. Syst.* 74, 15–29. doi: 10.1016/j.robot.2015.06.005

Tang, Q., Liang, L., Xie, J., Li, Y., Deng, Z. (2017). Task-priority redundancy resolution on acceleration level for underwater vehicle-manipulator system. *Int. J. Adv. Robotic Systems* 14, 172988141771982. doi: 10.1177/1729881417719825

Wang, Y., Jiang, S., Yan, F., Gu, L., Chen, B. (2017). A new redundancy resolution for underwater vehicle–manipulator system considering payload. *Int. J. Adv. Robotic Systems* 14, 172988141773393. doi: 10.1177/1729881417733934

Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transact. Man Machine Syst.* 10, 47–53. doi: 10.1109/TMMS.1969.299896

Xu, B., Pandian, S. R., Sakagami, N., et al. (2012). Neuro-fuzzy control of underwater vehicle-manipulator systems. *J. Franklin Institute* 349, 1125–1138. doi: 10.1016/j.jfranklin.2012.01.003

Xuefeng, D., and Xinqian, B. (2000). On coordinated control of multi-arm underwater vehicle. *J. Harbin Eng. Univ*. 21, 13–16. doi: 10.1007/s11804-006-6024-y

Youakim, D., Ridao Rodríguez, P., Rovira, N. P., Spadafora, F., Ribas, D., Muzzup, M., et al. (2017). MoveIt!: autonomous underwater free-floating manipulation. *IEEE Robot. Autom. Mag.* 24, 41–51. doi: 10.1109/MRA.2016.2636369

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 02 December 2022
DOI 10.3389/fnbot.2022.1066714

Check for updates

OPEN ACCESS

EDITED BY
Yan Wu,
Institute for Infocomm Research (A*STAR), Singapore

REVIEWED BY
Yanjie Wang,
Hohai University, China
Hongqian Lu,
Qilu University of Technology, China

*CORRESPONDENCE
Tiezheng Guo
guotiezheng@njit.edu.cn

RECEIVED 11 October 2022
ACCEPTED 16 November 2022
PUBLISHED 02 December 2022

CITATION
Guo T, Liu J, Liang H, Zhang Y, Chen W, Xia X, Wang M and Wang Z (2022) Design and dynamic analysis of jumping wheel-legged robot in complex terrain environment. *Front. Neurorobot.* 16:1066714. doi: 10.3389/fnbot.2022.1066714

Design and dynamic analysis of jumping wheel-legged robot in complex terrain environment

Tiezheng Guo[1]*, Jinhui Liu[1], Haonan Liang[1], Yitong Zhang[2], Wei Chen[1], Ximing Xia[1], Meiqing Wang[1] and Zhiming Wang[3]

[1]Industrial Center, Nanjing Institute of Technology, Nanjing, Jiangsu, China, [2]Direct Drive Technology Ltd., Dongguan, Guangdong, China, [3]Key Laboratory of Crop Harvesting Equipment Technology of Zhejiang Province, Jinhua, Zhejiang, China

Wheel-legged robots have fast and stable motion characteristics on flat roads, but there are the problems of poor balance ability and low movement level in special terrains such as rough roads. In this paper, a new type of wheel-legged robot with parallel four-bar mechanism is proposed, and the linear quadratic regulator (LQR) controller and fuzzy proportion differentiation (PD) jumping controller are designed and developed to achieve stable motion so that the robot has the ability to jump over obstacles and adapt to rough terrain. The amount of energy released by the parallel four-bar linkage mechanism changes with the change of the link angle, and the height of the jump trajectory changes accordingly, which improves the robot's ability to overcome obstacles facing vertical obstacles. Simulations and real scene tests are performed in different terrain environments to verify obstacle crossing capabilities. The simulation results show that, in the pothole terrain, the maximum height error of the two hip joint motors is 2 mm for the obstacle surmounting method of the adaptive retractable wheel-legs; in the process of single leg obstacle surmounting, the maximum height error of the hip joint motors is only 6.6 mm. The comparison of simulation data and real scene experimental results shows that the robot has better robustness in moving under complex terrains.

KEYWORDS

complex terrain environment, wheel-legged robot, dynamic analysis, adaptive obstacle crossing, motion analysis

Introduction

With the continuous development of robot technology, the application scope of mobile robots is constantly expanding, and the diversification of application scenarios leads to the increase of robots facing complex terrain environments (Zhang et al., 2014; Gao et al., 2020). Wheeled robots have the advantages of high efficiency and high energy utilization on flat roads (Wu, 2020; Xin et al., 2020), but they have poor adaptability to complex terrain. When the height of obstacles is greater than the radius of the wheels, they cannot effectively cross obstacles (Ding and Zhang, 2022). Legged robots have excellent adaptability when moving on the uneven and rough roads, but slow moving speed and low movement energy efficiency have always been technical problems that are difficult to break through (Xin et al., 2019; Xin and Vijayakumar, 2020). To solve this problem, the researchers turned their attention to wheel-legged robots. Wheel-legged robots combine the advantages of wheeled robots and legged robots. Double wheels can maximize energy utilization efficiency and maneuverability, and the leg structure makes the robot more adaptable to complex terrain environments.

At present, scholars have carried out a lot of research on the stability and high obstacle crossing ability of wheel-legged robots in unstructured terrain and have achieved a series of results: Liu et al. (2019, 2022) and Zhang et al. (2019) proposed a bipedal wheeled robot SR600 for logistics in scenarios such as distribution and home services, it can change height while maintaining dynamic balance. The size design of the human body can make it better interact with people. Kim et al. (2014) developed the Wheel Transformer, a variable-diameter wheel-legged robot. When encountering an obstacle, the wheels are transformed into two three-legged wheels to complete the action of crossing over the obstacle. It can overcome obstacles that 3.25 times higher than the wheel radius, but there is also the problem of low efficiency of crossing obstacles. In nature, animals jump over obstacles and avoid enemies attack by jumping (Fei et al., 2012; Cheng, 2021). Inspired by this, the bionic jumping theory was applied to the wheel-legged robot, and the jumping obstacle was realized by the wheel-legged robot (Zhuang et al., 2021; Hao et al., 2022). Chen et al. (2021) studied the jumping of a bipedal wheel-legged robot, proposed a W-SLIP model to characterize the jumping process dynamics, and verified the robot's jumping performance through V-REP simulation. Bipedal wheel-legged robot Ascento produced by ETH Zurich that adopted a compact design structure and can jump over obstacles while keeping the robot flexible and compact (Klemm et al., 2019; Klemm et al., 2020). The quadruped wheel-legged robot ANYmal (Bjelonic et al., 2019) of ETH Zurich also reflected the advantages of the combination of legged robots and wheeled robots to a large extent. The typical wheel-legged robot Handle developed by Boston Dynamics Ltd (2017). Achieved self-balancing through a dynamic control center, and used a hydraulic drive to jump to a height of 1.2 m (Zhang et al., 2018). At present, wheel-legged robots are still mainly used in simple application scenarios such as logistics handling, while wheel-legged robots suitable for complex terrain environments have been rarely reported.

Aiming at the problems that the wheel-legged robot is not stable in the complex terrain environment, a jumping wheel-legged robot is proposed in this paper, and an adaptive retractable wheel-leg mechanism is designed to keep the body parallel to the ground to achieve smooth obstacle-surmounting. The main motion forms of the robot include wheel and wheel-leg movement modes, and the modes are switched by the rotation of the hip motor. The wheel movement mode is used on flat terrain, and the wheel-leg movement mode is used on obstacles and rough terrain to improve energy utilization efficiency. In the wheel-leg movement mode, the robot cannot only use the wheels to move quickly and stably, but also jump over obstacles through the expansion and contraction of the wheel-leg linkage mechanism. The paper is structured as follows: In Section "Kinematics and dynamics analysis," the robot is introduced and the kinematics and dynamics modeling, including the self-balancing dynamics model and the jumping dynamics model. In Section "Analysis of obstacle crossing conditions," the wheeled overcoming obstacles and jumping over obstacles are analyzed respectively, and the conditions for overcoming the obstacles are obtained. The control system is presented in Section "Robot Control System," divided into self-balancing and jump control. Section "Simulation and experiment" conducts simulation tests on robot crossing obstacles in different terrains, and conducts actual test verifications. The paper is concluded in the last section.

Kinematics and dynamics analysis

The overall structure design of the robot is shown in **Figure 1**, which consists of the body, six motors, two linkage mechanisms, two pairs of universal wheels and two wheels. The two waist motors (including No. 4 in **Figure 1**) are hidden inside the body, and two motors are installed in the left and right wheels to drive the wheels to move. The waist motors are used to adjust the pitch angle of the body, and the hip motors are used to control the robot to change the height of the body and realize the jumping function. The hip motor rotates, and under the action of gravity, the connection between the Connecting rod 2 and the hip motor rotates around the rotation axis of the hip joint to realize the expansion and contraction of the leg linkage mechanism. The height of the body is raised and lowered through the expansion and contraction of the linkage mechanisms, and the wheel-leg linkage mechanisms contract and stretch in a short time, completing the accumulation and release of the energy required for jumping, and the robot realizes the jumping action. The wheel movement mode and the wheel-leg movement mode are shown in **Figures 2A,B** below, respectively.

Kinematic modeling

As shown in **Figure 3**, the kinematic model of the standing posture of the wheel-legged robot is established. $\{W\}$ is the

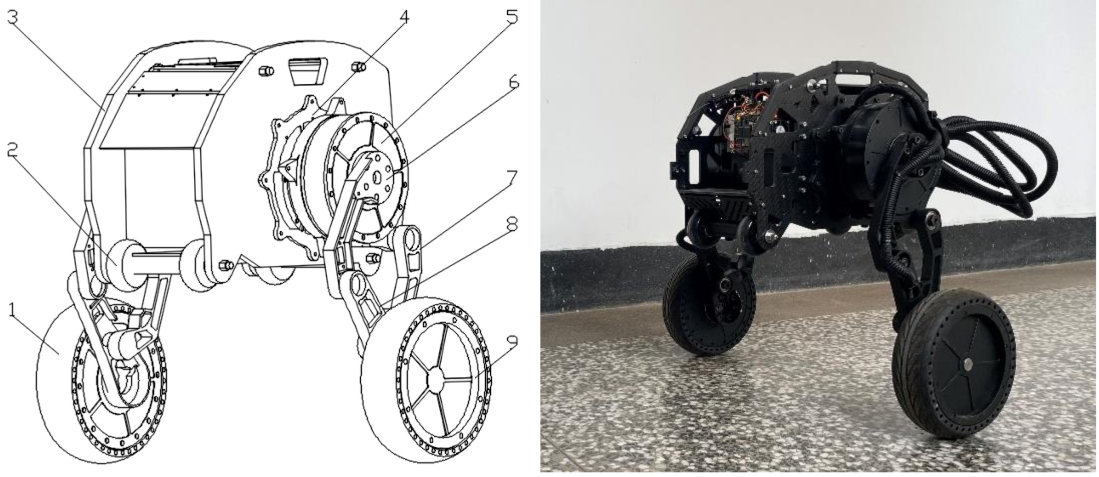


FIGURE 1
Schematic diagram of the robot structure. 1. Right wheel (driving wheel); 2. Universal wheel; 3. Body; 4. Waist motor; 5. Hip motor; 6. Connecting rod 1; 7. Connecting rod 2; 8. Connecting rod 3; 9. Left wheel (driving wheel).

world coordinate system, the Z axis is vertically upward, the X axis is perpendicular to the Z axis and points to the right end, and the Y axis direction is determined according to the right-hand rule. In order to simplify the kinematics problem, the base coordinate system $\{B\}$ is established at the contact point between the wheel and the ground, the direction is parallel to the world coordinate system, and the coordinate system $\{0\}$ is established at the center of the wheel. The coordinate system $\{1\}$ is established at the connection between the Connecting rod 3 and the wheel, the coordinate system $\{2\}$ is established at the connection between the Connecting rod 1 and the Connecting rod 3, and the coordinate system is established at the position shown in the **Figure 3** in turn. The $Z_i(i = 1 \sim 5)$ axis of the link is along the joint. The positive direction of the axis is placed perpendicular to the surface of the paper, the positive direction of the X_i axis points to the common perpendicular of the i axis and the i+1 axis, and the direction of the Y_i axis is determined by the right-hand rule.

In the kinematic model of **Figure 3**, a_{i-1} is the length of the connecting rod, α_{i-1} is the rotation angle of the connecting rod, d_i is the offset distance of the connecting rod, θ_i is the joint angle, l_i is the distance between the origin of the coordinate system $\{i-1\}$ and the origin of the coordinate system $\{i\}$ in the Z-X plane. where $i = 1 \sim 5$ is the rotational joint of the robot, $d_1 = 0.026\ m$, $d_2 = 0.021\ m$, $d_3 = 0.016\ m$, $d_5 = 0.016\ m$. $l_2 = 0.14\ m$, $l_3 = 0.14\ m$, $l_4 = 0.09\ m$, $l_5 = 0.14\ m$, $l_6 = 0.09\ m$, the robot wheel radius $r = 0.095\ m$.

According to the established kinematics model, the forward kinematics is solved, and the connecting rod transformation matrix is obtained as:

$$ {}^0_1T = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^0_2T = \begin{bmatrix} C_{12} & -S_{12} & 0 & C_1 l_2 \\ S_{12} & C_{12} & 0 & S_1 l_2 \\ 0 & 0 & 1 & -d_1 - d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

$$ {}^0_3T = \begin{bmatrix} C_{123} & -S_{123} & 0 & C_1 l_2 + C_{12} l_3 \\ S_{123} & C_{123} & 0 & S_1 l_2 + S_{12} l_3 \\ 0 & 0 & 1 & -d_1 - d_2 - d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

$$ {}^0_4T = \begin{bmatrix} C_{1234} & -S_{1234} & 0 & C_1 l_2 + C_{12} l_3 + C_{123} l_4 \\ S_{1234} & C_{1234} & 0 & S_1 l_2 + S_{12} l_3 + S_{123} l_4 \\ 0 & 0 & 1 & -d_1 - d_2 - d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

$$ {}^0_5T = \begin{bmatrix} C_{1234} & -S_{1234} & 0 & C_1 l_2 + C_{12} l_3 + C_{123} l_4 + C_{1234} l_5 \\ S_{1234} & C_{1234} & 0 & S_1 l_2 + S_{12} l_3 + S_{123} l_4 + S_{1234} l_5 \\ 0 & 0 & 1 & d_5 - d_1 - d_2 - d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3} $$

Where, C_{1234} is the meaning of $\cos(\theta_1 + \theta_2 + \theta_3 + \theta_4)$, S_{1234} is the meaning of $\sin(\theta_1 + \theta_2 + \theta_3 + \theta\ _4)$.

The robot is a parallelogram linkage mechanism, and the robot body is expected to be parallel to the ground while maintaining balance, so the constraint equation is attached:

$$ \theta_2 = \theta_4 = \pi - \theta_3,\ \ \theta_1 = \frac{1}{2}\theta_3 \tag{4} $$

Use UG software to analyze the position of the center of mass of each rod and the body of the robot, and obtain the coordinates of the center of mass $c1$ of the Connecting rod 3, the center of mass $c2$ of the Connecting rod 1, the center of mass $c3$ of the

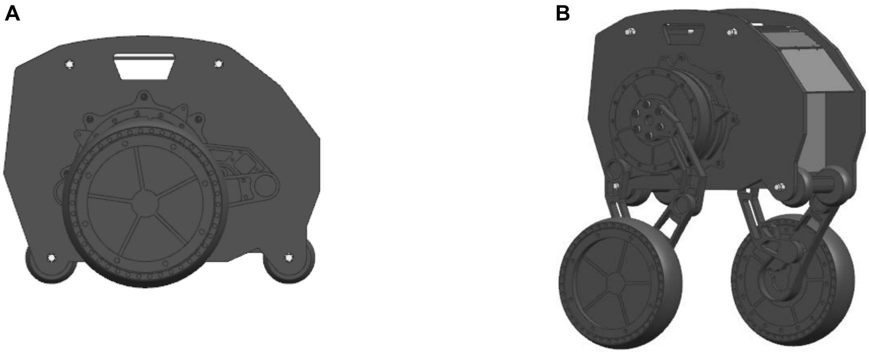


FIGURE 2
Schematic diagram of robot motion posture: **(A)** Wheel movement mode; **(B)** Wheel-leg movement mode.

body, and the center of mass $c4$ of the Connecting rod 2 relative to the X-Y plane of the {1}, {2}, {3}, {4} coordinate system:

$$\begin{cases} {}^{1}X_{c1} = 0.0545 \\ {}^{1}Y_{c1} = 0 \end{cases}, \begin{cases} {}^{2}X_{c2} = 0.0802 \\ {}^{2}Y_{c2} = -0.0128 \end{cases}, \\ \begin{cases} {}^{3}X_{c3} = -0.0219 \\ {}^{3}Y_{c3} = -0.0235 \end{cases}, \begin{cases} {}^{4}X_{c4} = 0.0714 \\ {}^{4}Y_{c4} = -0.0152 \end{cases} \tag{5}$$

The angle between the line connecting the center of mass $c2$ and the origin of the coordinate system {2} and the positive direction of the X_2 axis is $\varphi_2 = atan2({}^{2}Y_{c2}, {}^{2}X_{c2})$, the angle between the line of the center of mass $c3$ and the origin of the coordinate system of {3} and the positive direction of the X_3 axis is $\varphi_3 = atan2({}^{3}Y_{c3}, {}^{3}X_{c3})$, the angle between the line connecting the center of mass c4 and the origin of the coordinate system of {4} and the positive direction of the X_4 axis is $\varphi_4 = atan2({}^{4}Y_{c4}, {}^{4}X_{c4})$, the center of mass of the hip motor $c5$ is at the center of rotation. Let the lengths of the center of mass $c1$, $c2$, $c3$, and $c4$ from the origin of the {1}, {2}, {3}, and {4} coordinate systems be l_{c1}, l_{c2}, l_{c3}, and l_{c4}, respectively. Then the position of the center of mass of each rod ci relative to the world coordinate system is $(X_{ci}, \ Z_{ci})$, and the velocity is $\sqrt{\dot{X}_{ci}^2 + \dot{Z}_{ci}^2}$, $i = 1 \sim 5$.

Dynamic modeling

Self-balancing dynamic modeling

When the robot maintains a standing posture, the leg linkage mechanism is kept fixed by locking the hip joint motor. At this time, the robot can be equivalent to a two-wheeled self-balancing robot, as shown in Figure 4. The center of mass is located above the wheel axis of the robot, and the pose of the robot in the world coordinate system is $[x_b, y_b, z_b, \ \alpha]^T$, where the position coordinate of the midpoint of the axis of the driving wheels of the robot is $[x_b, \ y_b, z_b + r]$, α is the heading angle of the robot, the distance between the centers of the two wheels is D, and the radius of the wheel is r, the angles that the left and right wheels have turned are θ_L, θ_R, and the displacements of the left and right wheels are x_L, x_R, respectively. Assume that the body mass of the simplified robot is M, the length of the body is L, the moment of inertia of the body around the Y axis is I_b, and the position coordinate of the center of mass of the body is $[x, \ 0, z]$. The moment of inertia of each connecting rod at the center of mass is I_{ci}, which is obtained with the assistance of Adams simulation software. The tilt angle and body length of the equivalent model are:

$$\phi = \arctan(\frac{x}{z-r}), L = \sqrt{x^2 + (z-r)^2} \tag{6}$$

Taking the left wheel of the robot as an example to analyze the force on the wheel and the body, the balance formula of the force and moment of the body and the wheel can be obtained:

$$\begin{cases} F_{sL} - F_{xL} = m\ddot{x}_L \\ F_{nL} - F_{zL} = m_b g \end{cases}, \begin{cases} T_L - rF_{sL} - T_{fL} = I_W \ddot{\theta}_L \\ T_{fL} = b\left(\dot{\theta}_L - \dot{\phi}\right) \end{cases} \tag{7}$$

According to the relationship between the displacement of the midpoint of the line connecting the centers of the two wheels and the wheel rotation, the equation can be obtained:

$$(T_L + T_R) - r\left(F_{sR} + F_{sL}\right) - b\left(\dot{\theta}_L + \dot{\theta}_R\right) + \\ 2b\dot{\phi} - I_w\left(\ddot{\theta}_L + \ddot{\theta}_R\right) = 0 \tag{8}$$

According to the balance relationship between force and moment, follows is got:

$$(T_L + T_R) + ML\cos\phi\left(\ddot{x}_b + \ddot{\phi}L\cos\phi - \dot{\phi}^2 L\sin\phi\right) + \\ ML\sin\phi(-g + \ddot{\phi}L\sin\phi + \\ \dot{\phi}^2 L\cos\phi) - b\left(\dot{\theta}_L + \dot{\theta}_R\right) + 2b\dot{\phi} = I_b\ddot{\phi} \tag{9}$$

Among them, assuming that there is no slippage between the wheels and the ground, then $x_L = r\theta_L$, the relationship between the displacement of the midpoint of the line connecting the

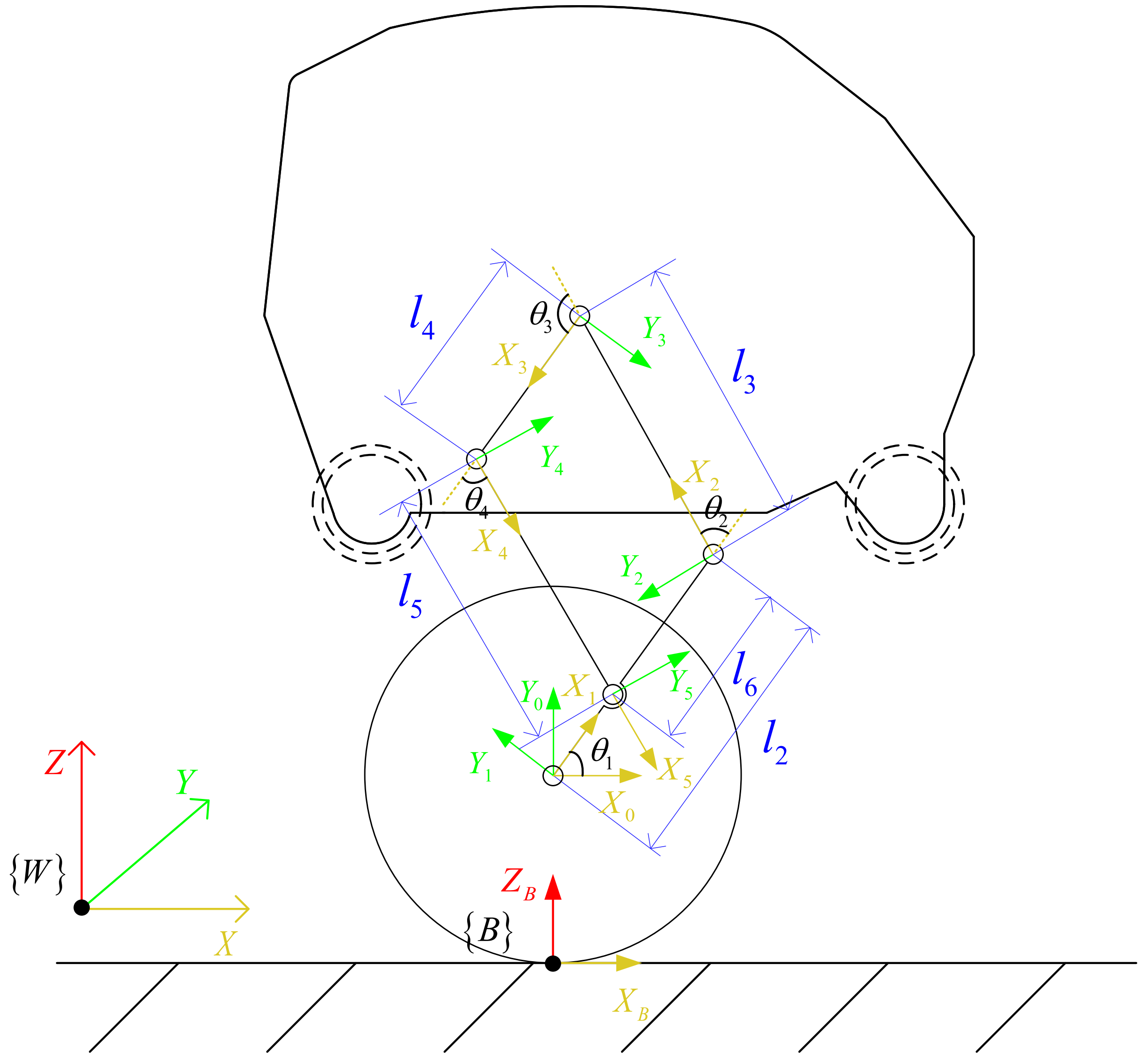


FIGURE 3
Kinematics model of robot.

centers of the two wheels and the left and right wheel angles is: $x_b = 1/2(\theta_L + \theta_R)$. I_w is the moment of inertia of the wheel, F_x is the interaction force between the wheel and the body in the X-axis direction, F_z is the interaction force between the wheel and the body in the Y-axis direction, T_L and T_R are the output torques of the left and right wheel motors, respectively, T_{fL} and T_{fR} are the friction torque, b is the friction coefficient between the wheel and the rod, the support force between the wheel and the ground is F_n, and the friction force is F_s.

The top view of the robot is shown in Figure 5, the $\dot{s}_L$ and $\dot{s}_R$ are the speeds of the left and right wheels in the X direction, respectively. Assuming that the moment of inertia of the robot around the vertical direction is I_t, the robot realizes differential turning when there is a differential speed between the two wheels. The turning dynamics equation is:

$$\left(\frac{DI_w}{r} + mDr + \frac{2I_t r}{D}\right)\ddot{\alpha} = T_L - T_R \tag{10}$$

The dynamic equations of the robot in the self-balancing mode can be sorted out as:

$$\begin{cases} T_L + T_R - \ddot{x}_b \left(2m_b r + Mr + I_w \frac{2}{r}\right) - \\ Mr\left(\ddot{\phi} L\cos\phi - \dot{\phi}^2 \sin\phi\right) - 2b\left(\frac{\dot{x}_b}{r} + \dot{\phi}\right) = 0 \\ \ddot{x}_b ML\cos\phi - MgL\sin\phi + \ddot{\phi}\left(ML^2 - I_b\right) - \\ \frac{2b}{r}\dot{x}_b + 2b\dot{\phi} + T_L + T_R = 0 \\ \left(\frac{DI_w}{r} + m_b Dr + \frac{2I_t r}{D}\right)\ddot{\alpha} = T_L - T_R \end{cases} \tag{11}$$

Organize the dynamic equations into non-linear dynamic equations in state-space form:

$$\begin{cases} \ddot{\phi} = \dfrac{\begin{gathered}(a_{21}\cos\phi - a_{11})(T_L + T_R) + \\ (a_{11}a_{24} - a_{15}a_{21}\cos\phi)\dot{x}_b - (a_{14}a_{21} + a_{11}a_{25})\dot{\phi} \\ + a_{13}a_{21}\dot{\phi}^2\sin\phi\cos\phi + a_{11}a_{22}\sin\phi\end{gathered}}{a_{11}a_{23} + a_{12}a_{21}\cos^2\phi} \\ \ddot{x}_b = \dfrac{\begin{gathered}-(a_{23} + a_{12}\cos\phi)(T_L + T_R) - a_{13}a_{23}\dot{\phi}^2\sin\phi + \\ (a_{14}a_{23} - a_{12}a_{25}\cos\phi)\dot{\phi} \\ + (a_{15}a_{23} + a_{12}a_{24}\cos\phi)\dot{x}_b + a_{12}a_{22}\sin\phi\cos\phi\end{gathered}}{a_{11}a_{23} + a_{12}a_{21}\cos^2\phi} \\ \ddot{\alpha} = \dfrac{T_L - T_R}{a_{31}} \end{cases} \tag{12}$$

Where, $a_{11} = 2m_b r + Mr + \frac{2I_w}{r}$, $a_{12} = MLr$, $a_{13} = Mr$, $a_{14} = 2b, a_{15} = \frac{2b}{r}, a_{21} = ML, a_{22} = Mgl, a_{23} = ML^2 - I_b$, $a_{24} = \frac{2b}{r}, a_{25} = 2b, a_{31} = \frac{DI_w}{r} + m_b Dr + \frac{2I_t r}{D}$.

For the dynamic equation of the robot in the self-balancing mode, take $X = [x_b, \dot{x}_b, \phi, \dot{\phi}, \alpha, \dot{\alpha}]^T$ as the system state variable and $u = [T_L, T_R]^T$ as the system input variable for linearization. Assuming that the robot is near the equilibrium position, there is $\phi \approx 0$, $\dot{\phi} \approx 0$, which is brought into the non-linear dynamic equation to obtain the linearization of the system. Equation of state:

$$\begin{bmatrix} \dot{x}_b \\ \ddot{x}_b \\ \dot{\phi} \\ \ddot{\phi} \\ \dot{\alpha} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & A_{42} & A_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_b \\ \dot{x}_b \\ \phi \\ \dot{\phi} \\ \alpha \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ B_2 & B_2 \\ 0 & 0 \\ B_4 & B_4 \\ 0 & 0 \\ B_6 & -B_6 \end{bmatrix} \begin{bmatrix} T_L \\ T_R \end{bmatrix} \tag{13}$$

Among them,

$$\Delta = r^2(2M^2L^2 + 2m_b ML^2 - 2m_b I_b - MI_b) + 2ML^2 I_w - 2I_w I_b$$
$$A_{22} = 2(bML^2 - bI_b + MLrb)/\Delta$$
$$A_{23} = M^2 gL^2 r^2/\Delta$$
$$A_{42} = (4brm_b + 2Mbr + 4bI_w/r - 2MbL)/\Delta$$
$$A_{43} = (2Mm_b gLr^2 + M^2 gLr^2 + 2I_w MgL)/\Delta$$
$$B_2 = (-ML^2 r + I_b r - MLr^2)/\Delta$$
$$B_4 = (-2m_b r^2 - Mr^2 - 2I_w + MLr)/\Delta$$
$$B_6 = Dr/(D^2 I_w + m_b D^2 r^2 + 2I_t r^2)$$

The linear system is decoupled into a separate balance subsystem and steering subsystem, and the straight-running torque T_ϕ and the steering torque T_ω of the robot are, respectively, input. The relationship between the left and right wheel torques and T_ϕ and T_ω is expressed as a matrix:

$$\begin{bmatrix} T_L \\ T_R \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} T_\phi \\ T_\omega \end{bmatrix} \tag{14}$$

Then the state space equations of the equilibrium system and the steering system are:

$$\dot{X}_1 = A_1 X_1 + B_1 T_\phi, \dot{X}_2 = A_2 X_2 + B_2 T_\omega \tag{15}$$

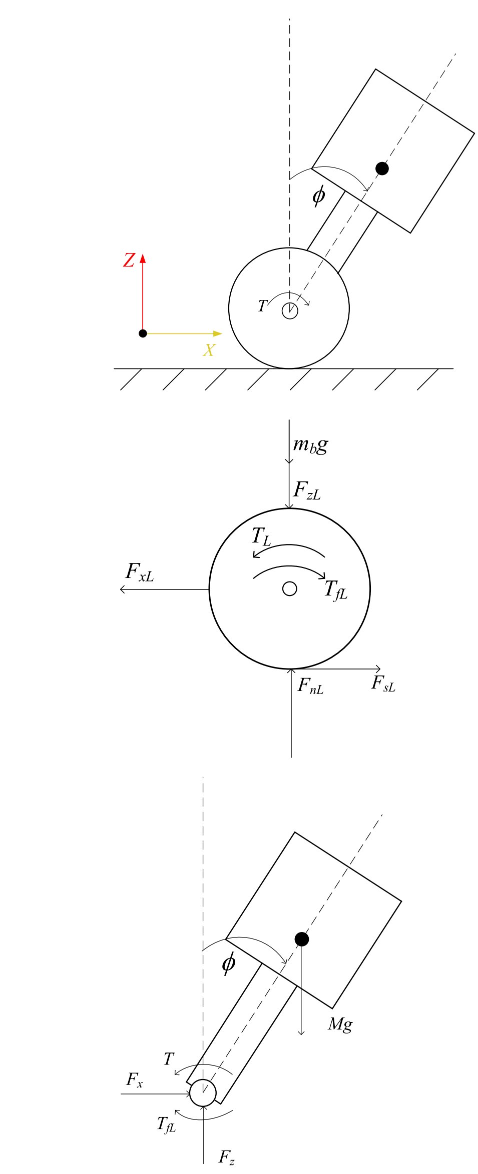


FIGURE 4
Force diagram of equivalent model.

Where,

$$X_1 = \begin{bmatrix} x_b \\ \dot{x}_b \\ \phi \\ \dot{\phi} \end{bmatrix}, A_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & A_{22} & A_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & A_{42} & A_{43} & 0 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ B_2 \\ 0 \\ B_4 \end{bmatrix},$$
$$X_2 = \begin{bmatrix} \alpha \\ \dot{\alpha} \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ B_6 \end{bmatrix}. \tag{16}$$

Jump phase dynamic modeling

The inertia tensor of the robot is calculated by Adams software. The coordinate origin of the inertia tensor is the

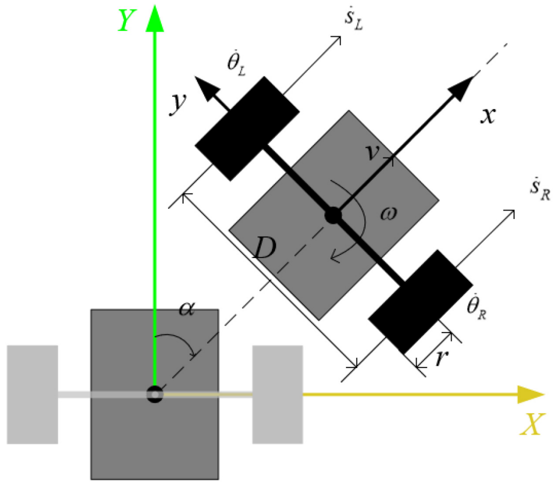


FIGURE 5
Top view of the robot.

center of mass of each rod. The jumping action of the robot is analyzed on the *XOZ* plane of the base coordinate system. The rotation axis of the rod is the *Z* axis of the corresponding joint coordinate system. In order to simplify the calculation of the inertia tensor, let the coordinate system with the center of mass of each rod as the origin of the inertia tensor coincide with the three inertia axes of the rod, and according to the parallel shift axis theorem, the change of the rotation axis of the connecting rod only changes the size of the moment of inertia in the inertia tensor, so the inertia tensor can be easily calculated using Adams software.

In this paper, the Lagrangian equation is used to establish the dynamic model of the wheel-legged robot, and θ_3 is used as the generalized coordinate, and the Lagrangian equation is applied to the process of the robot's take-off phase. The kinetic energy *K* and potential energy *V* of the system are:

$$\begin{cases} K_0 = \frac{1}{2} m_b \dot{X}_B^2 \\ K_1 = \frac{1}{2} m_{c1} (\dot{X}_{c1}^2 + \dot{Z}_{c1}^2) + \frac{1}{2} I_{c1} \dot{\theta}_1^2 \\ K_2 = \frac{1}{2} m_{c2} (\dot{X}_{c2}^2 + \dot{Z}_{c2}^2) + \frac{1}{2} I_{c2} (\dot{\theta}_1 + \dot{\theta}_2)^2 \\ K_3 = \frac{1}{2} m_{c3} (\dot{X}_{c3}^2 + \dot{Z}_{c3}^2) \\ K_4 = \frac{1}{2} m_{c4} (\dot{X}_{c4}^2 + \dot{Z}_{c4}^2) + \frac{1}{2} I_{c4} (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4)^2 \\ K_5 = \frac{1}{2} m_{c5} (\dot{X}_{c5}^2 + \dot{Z}_{c5}^2) + \frac{1}{2} I_{c5} \dot{\theta}_3^2 \end{cases} \tag{17}$$

$$V = 2m_{c1}gZ_{c1} + 2m_{c2}gZ_{c2} + m_{c3}gZ_{c3} + 2m_{c4}gZ_{c4} + \\ 2m_{c5}gZ_{c5} + 2m_b gZ_b \tag{18}$$

$$L = 2K_0 + 2K_1 + 2K_2 + K_3 + 2K_4 + 2K_5 - V \tag{19}$$

Where, *L* is the Lagrangian, which represents the difference in value between the kinetic energy *K* and potential energy *V* of the robot.

The hip joint motor torque $\tau = \left[\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{\theta}_3} \right) - \frac{\partial K}{\partial \theta_3} + \frac{\partial V}{\partial \theta_3} \right] / 2$ is obtained, and the hip joint space dynamics state space equation of the robot is sorted out as:

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \tag{20}$$

Where, $M(q)$ is the inertia matrix of the robot, $V(q, \dot{q})$ is the centrifugal and Coriolis matrix, and $G(q)$ is the gravity compensation vector.

Analysis of obstacle crossing conditions

The robot's ability to overcome obstacles is mainly affected by its own structure and road conditions. The wheel movement mode of the wheel-legged robot proposed in this paper is mainly used for smooth road movement, and the wheel-leg mode is used to pass the rough road with obstacles. In the following, the obstacle crossing analysis is carried out, respectively, for the cases that the obstacle height is lower than the wheel radius and the obstacle height is higher than the wheel radius in the wheel-leg mode.

Wheeled obstacle crossing analysis

When the height of the obstacle is less than the radius of the wheel, the analysis is performed at the initial stage of the robot crossing the obstacle. At the beginning of obstacle crossing, keep the center of mass of the robot and the center of rotation of the wheel on the same vertical line, and perform force analysis on it.

Figure 6 shows the robot cross over the obstacle in wheel-leg mode, at the beginning of obstacle crossing, the robot is balanced by the forces and moments between the ground and the wheels and between the obstacles and the wheels. The balance formula of the robot's two legs over obstacle is shown in Equation (21):

$$\begin{cases} M - Gr\cos\alpha + F_{N1} r \cos\alpha = 0 \\ F_{N2}\cos\alpha - f_1 - f_2 \sin\alpha = 0 \\ G - F_{N1} - F_{N2}\sin\alpha = 0 \end{cases} \tag{21}$$

In Equation (21), M_w is the torque of the wheel motor, *G* is its own gravity, F_{N1} is the support force of the ground facing the wheel, F_{N2} is the support force of the obstacle contact point to the wheel, f_1 is the friction force of the ground facing the wheel, and f_2 is the frictional force on the wheel at the contact point of the obstacle, α is the angle between the supporting force of the contact point between the wheel and the obstacle and the ground, *h* is the height between the contact point of the obstacle and the ground, and *r* is the radius of the wheel.

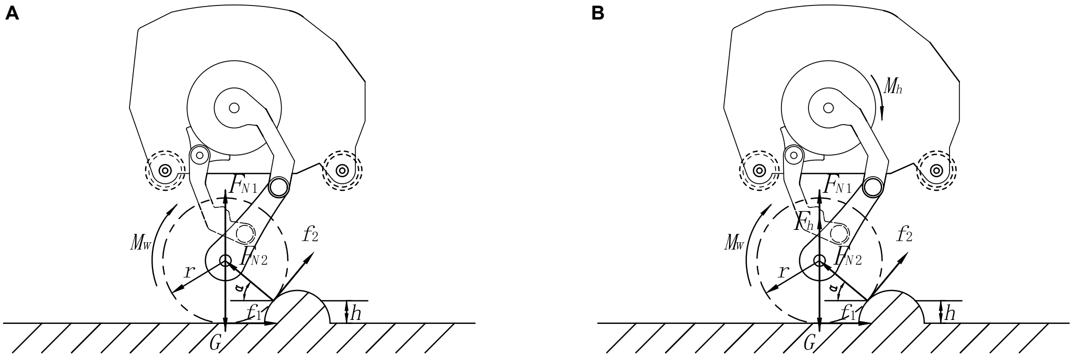


FIGURE 6
Wheel over obstacles in wheel-leg mode: **(A)** Cross obstacle on two legs; **(B)** Cross obstacle on one leg.

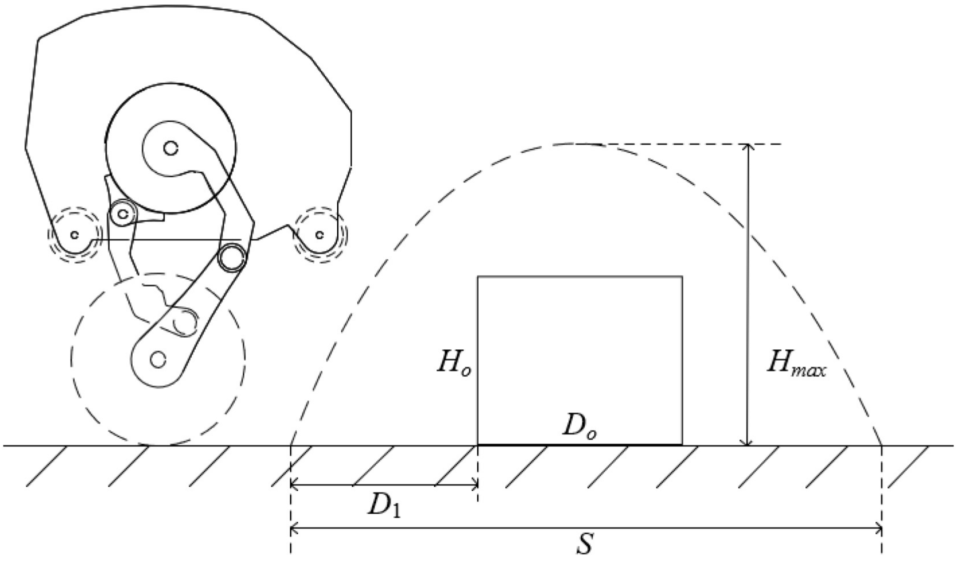


FIGURE 7
Schematic diagram of the robot over the obstacle.

The balance formula of the robot's one leg over obstacle is shown in Equation (22):

$$\begin{cases} M_w - Gr\cos\alpha + F_{N1}r\cos\alpha + F_h r\cos\alpha = 0 \\ F_{N2}\cos\alpha - f_1 - f_2\sin\alpha = 0 \\ G - F_{N1} - F_{N2}\sin\alpha - F_h = 0 \end{cases} \tag{22}$$

In Equation (22), M_h and F_h are, respectively, the torque of the hip motor in the adaptive contraction state of the robot when crossing the obstacle with one leg, and the force transmitted by the torque of the hip motor to the wheel through the connecting rod of the wheel leg. The meanings of other parameters are consistent with Equation (21).

In Equations (21, 22), $f_1 = \mu F_{N1}$, $f_2 = \mu F_{N2}$, $\alpha = arcsin\frac{r-h}{r}$. When $\alpha > 0$, the wheel can advance over the obstacle by its own rotational motion driven by the motor.

Analysis of jumping over obstacles

In the analysis of the robot jumping over obstacles, due to the symmetric design of the robot, the center of mass in the process of movement can be kept only in the X and Z direction displacement in the world coordinate system, and the jumping process is analyzed according to the plane robot analysis method. Assume that the take-off phase time of the robot is $[t_0, t_1)$ and the flight phase time is $[t_1, t_4)$. If the robot wants to jump off the ground after energy storage, it needs to meet the following conditions:

$$\begin{cases} \dot{Z}_c > 0 \\ \ddot{Z}_c = -g \end{cases} \tag{23}$$

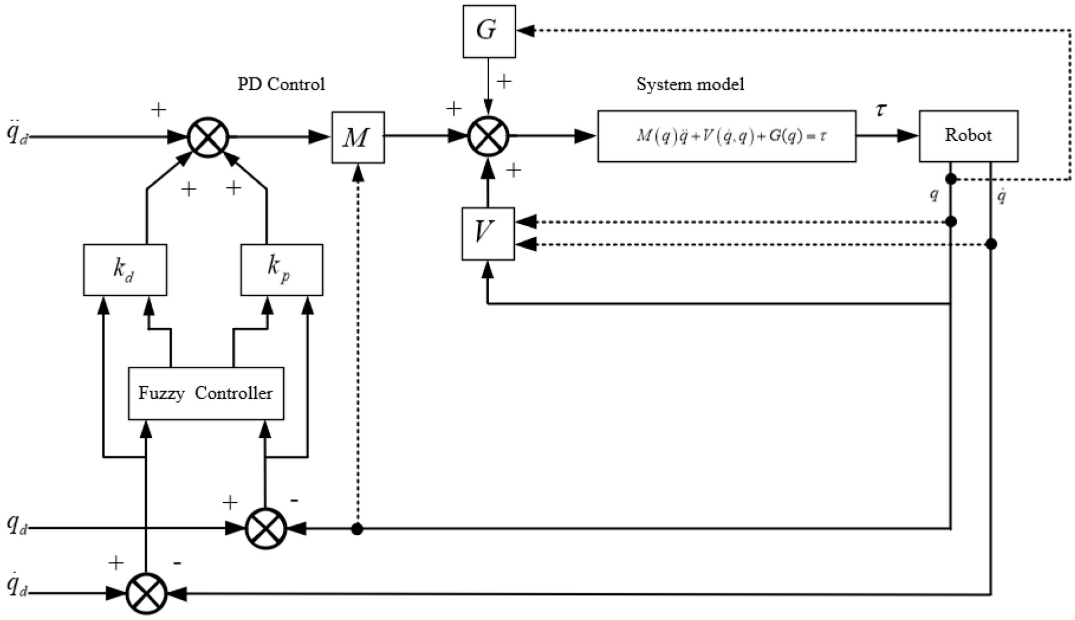


FIGURE 8
Fuzzy Proportion Differentiation (PD) control block diagram.

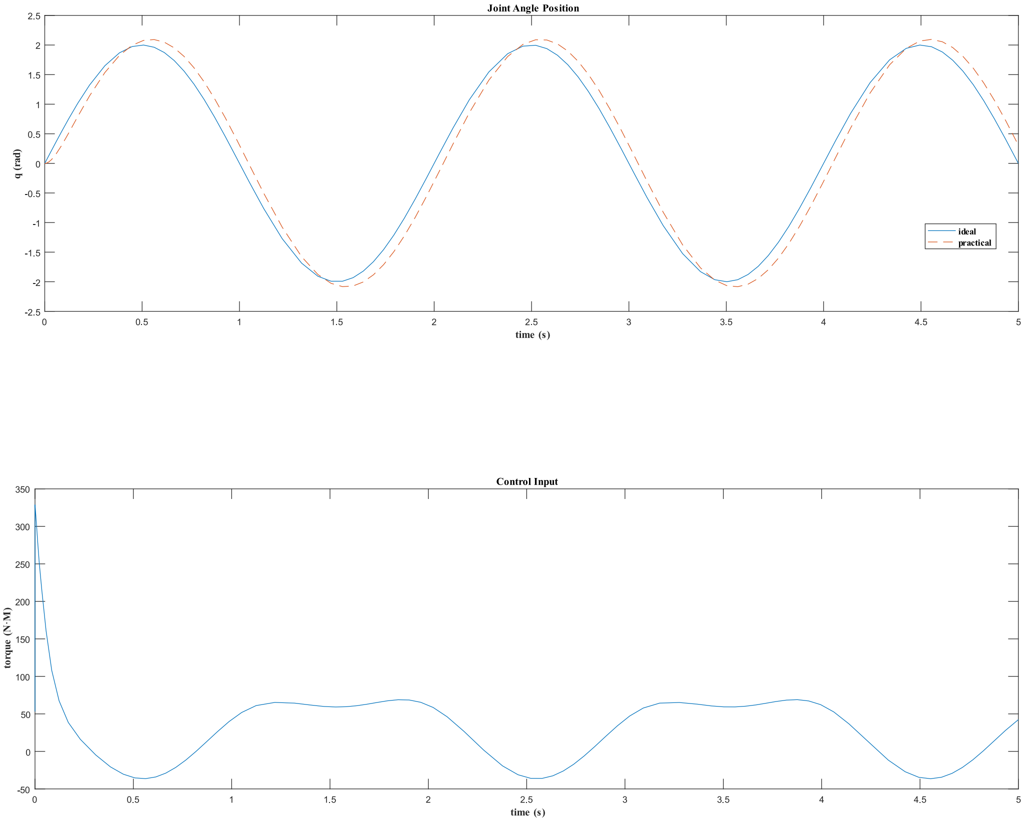


FIGURE 9
Control effect of fuzzy PD controller.

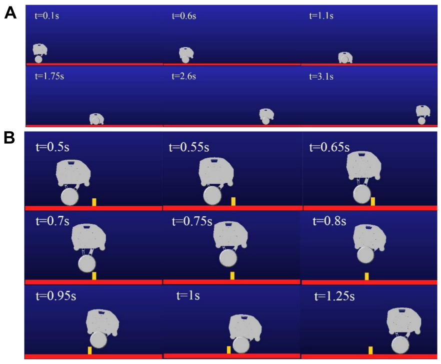


FIGURE 10
Screenshot of movement process: **(A)** Mode switching process diagram; **(B)** Jump process diagram.

Therefore, the following conditions should be met in the take-off phase time $[t_0, t_1)$:

$$\begin{cases} \dot{X}_c(t_1) = \dot{X}_{c0} \geq 0 \\ \dot{Z}_c(t_1) = \dot{Z}_{c0} \geq 0 \\ \ddot{Z}_c(t_1) \geq -g \end{cases} \tag{24}$$

When the speed of the robot in the horizontal and vertical directions is greater than zero, it will make an oblique throw motion with only the initial speed in the flight phase. It is assumed that the robot makes uniform linear motion in the horizontal direction before jumping, and the velocity of the center of mass in the horizontal and vertical directions is v_x and v_y, respectively. Then, the robot's flight time is $t = \frac{2v_y}{g}$, the highest jumping height is $H_{max} = \frac{v_y^2}{2g}$, and the horizontal displacement distance is $S = \frac{2v_x v_y}{g}$.

The obstacle crossing process is shown in Figure 7, it's required that when the horizontal displacement is $S_{t2} = D_1$, the jump height $H_{t2} > H_o$. When the horizontal displacement is $S_{t3} = D_1 + D_o$, the jump height $H_{t3} > H_o$. Then, the conditions under which the robot can overcome obstacles are:

$$\begin{cases} H_{\max} = \frac{V_y^2}{2g} > H_o \\ H_{t2} > H_o \\ H_{t3} > H_o \\ S = \frac{2V_x V_y}{g} > D_1 + D_o \end{cases} \tag{25}$$

When considering the robot perform jumps has higher request to the speed and acceleration, vertical direction of the robot center of mass in accordance with the planned five times polynomial interpolation method, the initial and the end of a robot position, velocity and acceleration, respectively, y_{init}, v_{init}, a_{init} and y_{end}, v_{end}, a_{end}, using the method of undetermined coefficients can get desired trajectory:

$$y(t) = a_0 + a_1 t + a_2 + t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{26}$$

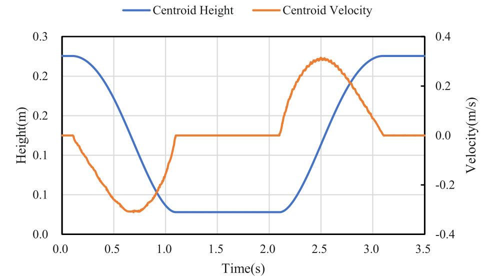


FIGURE 11
Position and velocity curve of state switching process.

Among them,

$$\begin{aligned} a_0 &= y_{init} \\ a_1 &= v_{init} \\ a_2 &= a_{init}/2 \\ a_3 &= [20y_{end} - 20y_{init} - (8v_{end} + 12v_{init})t_4 - \\ &\quad (3a_{init} - a_{end})t_4^2]/2t_4^3 \\ a_4 &= [30y_{init} - 30y_{end} - (14v_{end} + 16v_{init})t_4 - \\ &\quad (3a_{init} - 2a_{end})t_4^2]/2t_4^4 \\ a_5 &= [12y_{end} - 12y_{init} - (6v_{end} + 6v_{init})t_4 - \\ &\quad (a_{init} - a_{end})t_4^2]/2t_4^5 \end{aligned}$$

Robot control system

Balance and speed control

In general, the wheel-leg movement mode of the robot proposed in this paper can be divided into two forms: wheel movement and jumping over obstacles, which are self-balancing mode and jumping mode, respectively. The robot can switch between the two modes to complete the task, and it is necessary to design controllers for the two motion modes, respectively. The balance and speed control of the robot are realized by the Linear Quadratic Regulator (LQR) method. The state equation of the linear time-invariant system is:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{27}$$

For the balance system of the robot, the speed, tilt angle and tilt angle velocity are taken as the state variables, and the state variable is $x = [\dot{x}_b, \phi, \dot{\phi}]$. Then the equilibrium system can be expressed as:

$$\begin{bmatrix} \ddot{x}_b \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} A_{22} & A_{23} & 0 \\ 0 & 0 & 1 \\ A_{42} & A_{43} & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_b \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \\ B_2 \end{bmatrix} T_\phi \tag{28}$$

If the robot takes $\dot{x}_{ref}$ as the reference speed and keeps balance, then $\dot{\phi} = 0$, $\dot{x} = \dot{x}_{ref}$, the system stable state variable

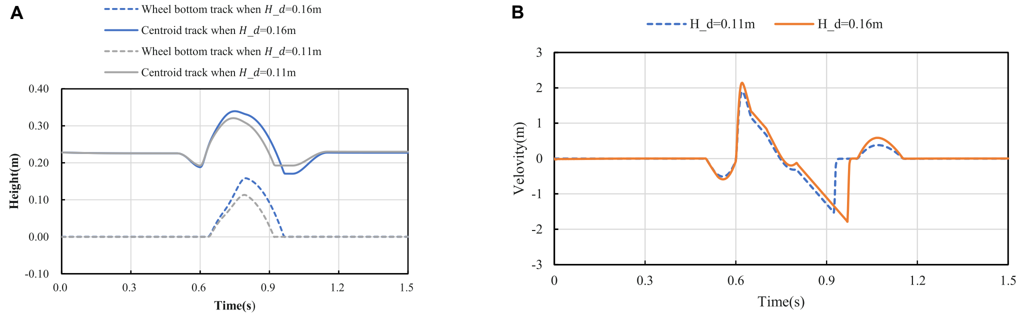


FIGURE 12
Jump process curve: **(A)** Position curve; **(B)** Vertical velocity curve.

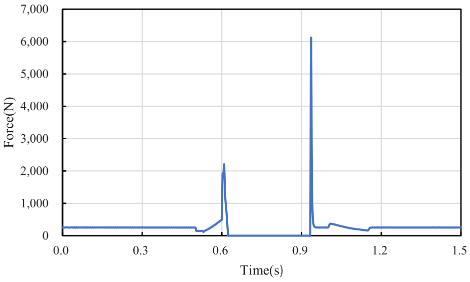


FIGURE 13
Ground support reaction force variation curve.

$x_s = f(\dot{x}_{ref})$, the stability control input $u_s = g(\dot{x}_{ref})$, take the new state variable $\Delta x = x - x_s$, the new system input $\Delta u = T_\phi - u_s$, then the new system state equation is:

$$\Delta x = \begin{bmatrix} A_{22} & A_{23} & 0 \\ 0 & 0 & 1 \\ A_{42} & A_{43} & 0 \end{bmatrix} \Delta x + \begin{bmatrix} B_1 \\ 0 \\ B_2 \end{bmatrix} \Delta u \qquad (29)$$

The optimal control input is obtained as follows:

$$u = u_s - k\Delta x \qquad (30)$$

Where, $f(\dot{x}_{ref}) = \begin{bmatrix} \dot{x}_{ref} \\ \frac{A_{22}B_2 - A_{42}B_1}{A_{43}B_1 - A_{23}B_2} \\ 0 \end{bmatrix}, g(\dot{x}_{ref}) = \frac{A_{42}A_{23} - A_{22}A_{43}}{A_{43}B_1 - A_{23}B_2}$.

Jumping control

In this paper, the jump control of the robot adopts fuzzy Proportion Differentiation (PD) control. According to the dynamic equation of the jump stage obtained in the dynamic analysis:

$$M(q)\ddot{q} + \underbrace{V(\dot{q}, q) + G(q)}_{H(\dot{q}, q)} = \tau \qquad (31)$$

The control scheme for calculating torque is set as:

$$M(q)(\ddot{q}_d + K_p e + K_d \dot{e}) + V(\dot{q}, q)\dot{q} + G(q) = \tau \qquad (32)$$

Where, $e = q_d - q$, $e = \dot{q}_d - \dot{q}$, q_d and q are the ideal angle and the actual angle, respectively.

The fuzzy PD controller is designed to calculate the torque control, e and $\dot{e}$ are taken as the input of the fuzzy controller. Mamdan rule is used for fuzzy inference, and min-max-center of gravity method is used for fuzzy resolution. The output is ΔK_p and ΔK_d, let the self-tuning parameters $K_p = K_{p0} + \Delta K_p$, $K_d = K_{d0} + \Delta K_d$. K_{p0} and K_{d0} are initial values, KP and KD are final values. The block diagram of the control scheme is shown in **Figure 8**.

For the dynamic model of the robot in the take-off phase, assuming that the angle tracking command of the robot hip joint is $q_d = 2\sin(\pi t) rad$, the dynamic equation is written as s-function by the Simulink module in Matlab to test the simulation effect. Fuzzy PD control is used to design the control law, and the initial values of PD parameters are set as $K_{p0} = 20$ and $K_{d0} = 20$. The simulation results are shown in **Figure 9**.

It can be seen that the joint angle tracking curve of the fuzzy PD controller is close to the expected tracking angle curve in **Figure 9**, and has a fast response speed, no obvious overshoot, and has a good control effect.

Simulation and experiment

Simulation

The 3D model of the robot was imported into Adams simulation software, and the co-simulation of the self-balance and obstacle crossing of the robot is carried out by using Simulink module of Matlab. The hip motor rotates at different angles, which can adjust the height of the robot when standing. First, the simulation of the robot shifting from the wheel mode to the wheel-leg mode is carried out. When the hip motors rotate in wheel-leg mode, the robot switches to wheel movement

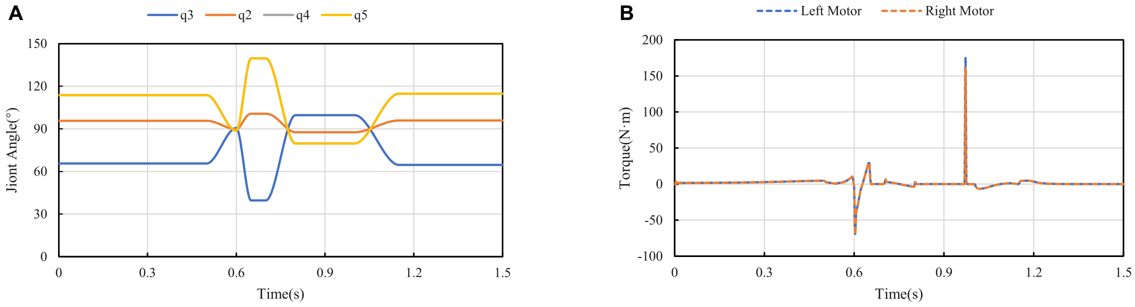


FIGURE 14
Curve change of jump process: **(A)** Curve of joint angle change; **(B)** Output torque diagram of motors.

mode, and then reverts to the original posture through motors reversals. The simulation process is shown in Figure 10A. In order to realize the function of jumping over obstacles, the hip motors need to rotate with different steering to complete the contraction and extension of the wheel leg connecting rod, so as to storage and release the energy needed for jumping. Taking the single jump process of the robot as an example, its screenshot is shown in Figure 10B. When the robot moves at a constant speed of 1 m/s, the robot reaches the highest jump height of 0.16 m at 0.8 s. The take-off phase of the robot is 0.5–0.6 s, the flight phase is 0.6–0.95 s, and the landing phase is 0.95–1.2 s.

In the process of mode switching, the position and velocity change curve of the robot's center of mass in the vertical direction are shown in Figure 11. It can be seen that the velocity change curve is smooth and continuous, which can ensure the stable mode switching. According to different expected jumping heights, the jumping trajectories of the robot are different. Jump simulation experiments are conducted for the expected heights $H_d = 0.16\ m$ and $H_d = 0.11\ m$, and the longitudinal trajectories of the bottom of the robot wheel and the robot center of mass are obtained as shown in Figure 12A. It can be seen from Figure 12B, the motion of the robot after take-off is oblique throwing motion, and the planned motion wants to achieve a higher jumping height, so a larger longitudinal velocity is needed at the take-off point to prolong the time of flight. In the take-off phase, the wheel-leg linkage mechanisms successively contract and extend to realize the energy storage and release of the machine mechanisms. When the condition of leaving the ground is reached, the end of the robot's wheels leave the ground and enter the phase of flight. In the flight phase, the robot system is in the state of momentum conservation, so the method of contracting the wheel-legs to increase the jumping height causes the overall centroid velocity to fluctuate. In order to reduce the impact of landing, a method is used to keep the mechanism retraction until the wheels touch the ground. After

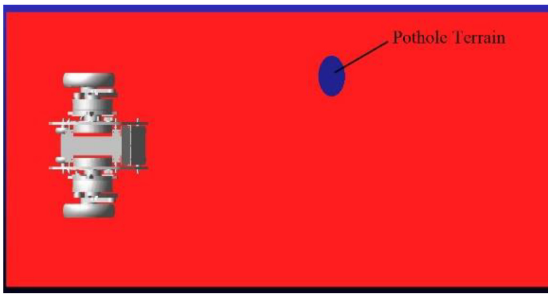


FIGURE 15
Schematic diagram of pits.

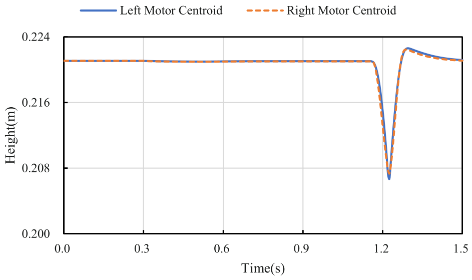


FIGURE 16
Left and right motor location diagram.

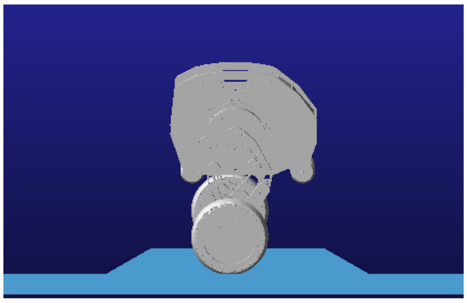


FIGURE 17
Obstacle crossing of wheel-leg without adaptive contraction.

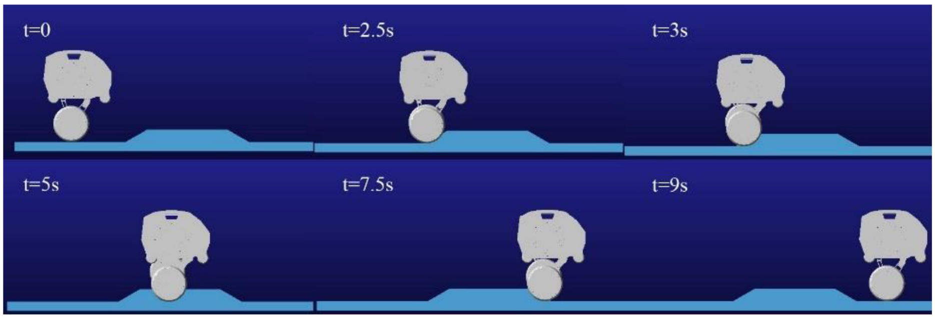


FIGURE 18
Process diagram of single-leg obstacle crossing.

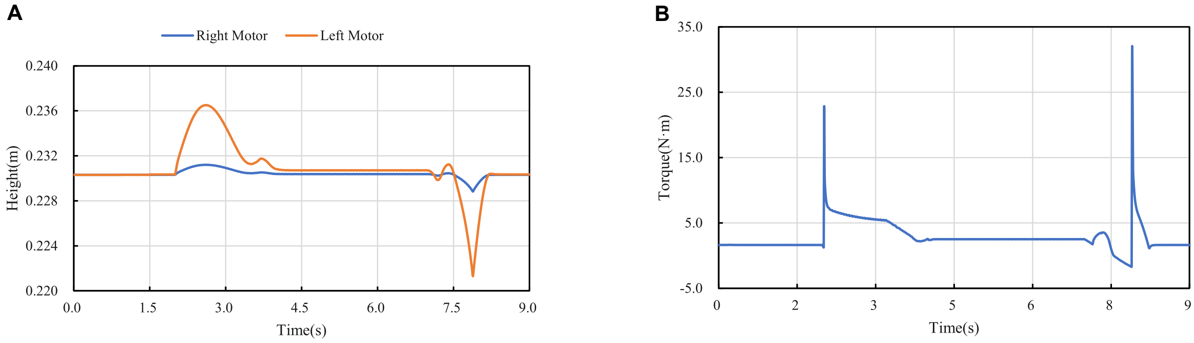


FIGURE 19
Crossing the obstacle with one leg: **(A)** Diagram of motors' locations; **(B)** Diagram of motors' torques.

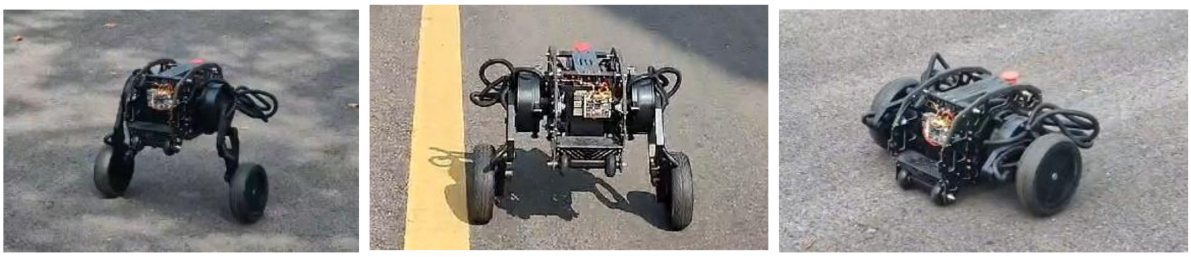


FIGURE 20
Schematic diagram of different motion states.

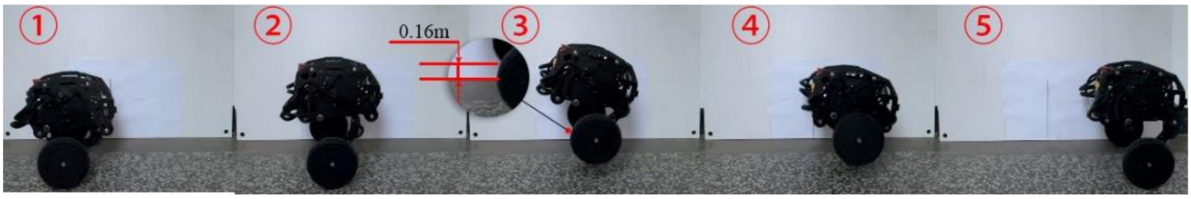


FIGURE 21
Diagram of jump process.

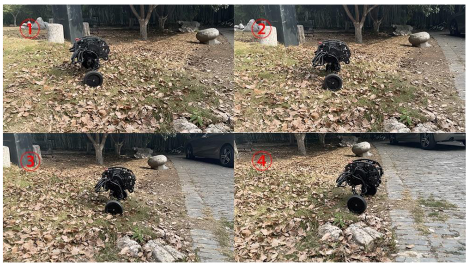


FIGURE 22
Moving over potholed ground.

the robot lands, the wheel-legs extend back to the position before the jump.

In Figure 13, the curve of ground support reaction force on the wheel in the process of jumping is measured, and it's found that the support reaction force gradually increases in the take-off phase, reaches the maximum value at 0.6 s at the end of wheel-leg contraction, and decreases rapidly at the moment of take-off. When the wheels leave the ground, the support reaction force is 0. At the moment of landing, the support reaction force will change due to impact. The change of the support reaction force corresponds to the height above the ground in each stage of the robot jumping process, which verifies the feasibility of the jumping.

Figure 14A shows the angle variation of each joint when the jump height is 0.16 m. Since the wheel-leg of the robot is a parallelogram linkage mechanism, $q_4 = q_5$. Among them, the angles q_2, q_3, q_4, and q_5 represent are the angle between Connecting rod 3 and Connecting rod 1, the angle between Connecting rod 3 and Connecting rod 2 at the hinge point, The angle between Connecting rod 2 and the motor rotation center and the Connecting rod 2 hinge, and the angle between Connecting rod 2 and Connecting rod 3. Robot hip motor torque curves is shown in Figure 14B, it can be seen two hip motors' output torques are basically consistent, the motors' torques of the wheel-legs contraction in the take-off phase are shown in the curves from 0.5 to 0.6 s, which are the torques curves of the wheel-legs extension from 0.6 to 0.65 s, and the torque generated near 0.97 s is generated when the robot lands and touches the ground.

When the robot passes over the pavement with pothole (major diameter: 15 cm, minor diameter: 10 cm, depth: 5 cm) as shown in Figure 15, the robot is required to keep moving smoothly. When the left leg of the robot passes over the pothole, the linkage mechanism of the right leg contracts to keep the axes of the motor rotation of the hip joint of the two legs coincide as much as possible. Figure 16 shows the height of the rotation axes of the left and right wheel-leg hip motors in this process. It can be seen that the axes of the two motors basically coincide. At time of 1.2 s, the deviation of the axis of the two legs is relatively large, but as a whole, the deviation is small, only about 2 mm. The robot can smoothly pass through the pothole terrain.

The following simulation experiments are done for the robot to cross the slope obstacle with one leg. The obstacle set by the simulation is about 6.5 cm in height and 65 cm in length. When the robot crosses the obstacle without adaptive contraction wheel-legs, the side view of the obstacle crossing process of the robot is shown in Figure 17. In this case, crossing the obstacle may cause damage to the connecting rod structure of the robot wheel-leg or the motors of the hip joints. Therefore, when the vertical height of the contact between the left and right wheels and the ground is different, the wheel-leg linkage mechanism with relatively high vertical position of the contact point needs to be properly contracted to ensure the stability of the robot body. For this purpose, the one-leg obstacle crossing simulation experiment was carried out, and the whole process of obstacle crossing was captured, as shown in Figure 18.

Figure 19A shows the height variation curve of the rotation axis of the left and right hip joint motors in the vertical direction. It can be seen from the figure that the left leg of the robot starts to cross the obstacle when it contacts the obstacle at about 2 s, and the robot starts to leave the obstacle at about 7.15 s. In the process of obstacle crossing, the highest position deviation of the motor axis of the hip joint of the two legs is generated at 7.9 s, and the maximum deviation between the motor position of the left leg and the motor position of the right leg is about 6.6 mm. It can be seen that in the whole process of obstacle crossing, the position height deviation of the two hip motors is low, and the performance of obstacle crossing is better. And because of gravity, the time for going up and down is different, and the time for going down is less than that for going up. Figure 19B is the robot's left leg hip motor torque figure, can be seen from the figure when the robot starts uphill because access to the obstacles, will jump a torque value, and in the process of uphill, with the contraction of the wheel-leg, it gradually attenuates to the torque value of maintaining the motor position locking, and also produces a sudden torque when downhill.

Experimental verification

On the basis of simulation, in order to verify the feasibility of jumping over obstacles designed in this paper and the correctness of the simulation results, real robot tests are carried out. The experiments are based on the DDT robot platform of Direct Drive Technology Ltd. The total mass of the robot is about 30 kg, the height of the robot in the wheeled mode is about 21 cm, the length of the robot is about 35 cm, and the width is about 53 cm. The height of the robot can be adjusted under the wheel-leg mode. The robot moves in wheel mode and wheel-leg mode at different height of center of mass on flat ground, the motion posture is shown in Figure 20.

The real tests are carried out in different terrain, first of all, the jump height and jump feasibility will be analyzed and

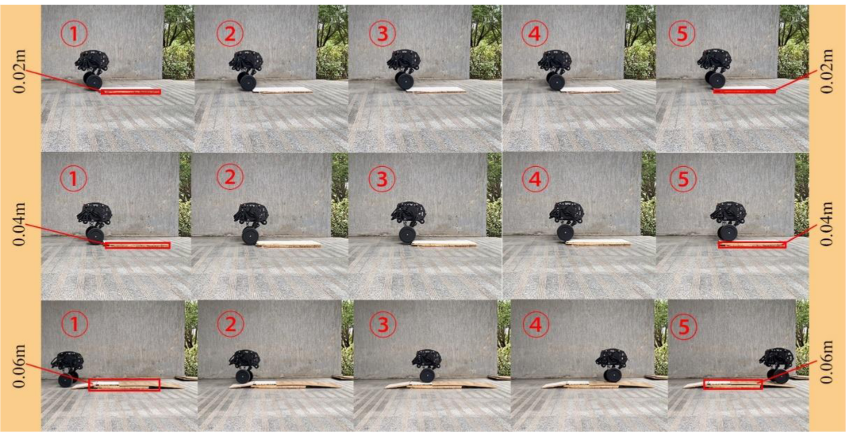


FIGURE 23
Comparison of different height of obstacle crossing.

verified under flat ground. Control the robot to jump and compare the jump height with the scale of the preset whiteboard (see Figure 21). The energy needed for the robot to jump is stored by the contraction of the leg mechanisms, the energy release is completed by the extension of the leg mechanism and the jump is carried out, the wheel-legs are contracted in the flight phase to improve robot's height off the ground, and the purpose of jumping over obstacles is realized. After comparing with the scale of the white board, it is found that when the robot jumps at a low speed, the jump height can be 0.16 m. The experimental results are basically consistent with the simulation results, which verifies the feasibility of the jump action of the wheel-legged robot designed in this paper.

The adaptive contraction of the wheel leg is actually tested on the pothole ground. In Figure 16, it can be seen that in the simulation experiment of pothole terrain, the robot passes through the pothole in about 1.2 s, corresponding to the robot state shown in serial number ① and serial number ② of Figure 22. At this time, the robot walks on the road with potholes. The axes of the two hip motors are always in the same vertical position by contracting the wheel-leg which is at a higher position, it ensures the stability of the robot when walking on the pothole ground.

For the test of single-leg obstacle crossing, a single block, two blocks and three blocks of wood are set in front of the right leg as obstacles. The height of a single plank is 2 cm, the height of the corresponding obstacles is 2, 4, and 6 cm, respectively, and the radius of the wheel is 9.5 cm. The obstacle crossing process is shown in Figure 23. The process from contacting the obstacle to adaptively adjusting the expansion of the wheel-leg is shown. It can be seen that the degree of contraction of the wheel-leg is different when the height of the obstacle is different. The robot can easily cross the obstacles with height lower than the radius of its own wheels, and keep moving smoothly through obstacles by adjusting the wheel-leg adaptively.

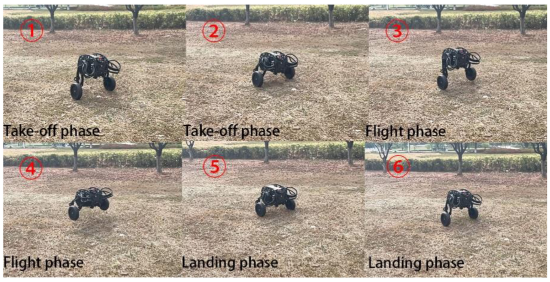


FIGURE 24
Jumping process on uneven ground.

In addition to jumping on the flat and open ground, jumping over obstacles on the uneven ground is an important embodiment of the robot's ability to jump over obstacles. The jumping ability was further tested on the uneven grass outside, and the test results in Figure 24 shows that the robot still had good jumping ability on the uneven ground.

Conclusion

In this paper, a bipedal wheel-legged robot with parallel four-bar linkage wheel-leg structure is proposed. The kinematics and dynamics of the robot are analyzed, and LQR controller and fuzzy PD controller are designed for balance and jump, respectively. According to the output torque curve and hip joint angle tracking curve obtained by Simulink simulation experiment, it can be seen that it has a good control effect. In view of the different ground conditions that the robot may encounter in the complex terrain environment, Adams and Simulink are used to simulate the robot's obstacle crossing, respectively, for the pothole road surface, the obstacle height is higher than the wheel radius and the obstacle crossing with one leg. Under the gait strategy of adaptive wheel-leg contraction, the error of each simulation data of the robot is small and the output torque is within the effective output range of the motor, which can ensure the smooth obstacle crossing. In the jump simulation, when the expected jump height is 0.11 and 0.16 m, the vertical velocity of the former is lower than that of the latter. To increase the height of obstacle crossing, the vertical velocity of the take-off should be increased. The gait strategies used in each simulation experiment are verified in real scene test, and the robot can smoothly cross the obstacles, which verifies the feasibility of the jumping and control method.

Data availability statement

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

Author contributions

TG: methodology, project administration, and writing—review and editing. JL: simulation, analysis, experiments, and writing—original draft preparation. HL: control system design. YZ: conceptualization, design, and mechanical modeling. WC, XX, and MW: supervision and review. ZW: funding acquisition. All authors read and agreed to the published version of the manuscript.

Funding

This research was funded by the Jiangsu Provincial Agricultural Science and Technology Independent Innovation Fund Project [number: CX(21)1007], the Open Project of the Zhejiang Provincial Key Laboratory of Crop Harvesting Equipment and Technology (number: 2021KY03), the Jiangsu Province Postgraduate Research and Practice Innovation Program (number: SJCX22_1063), and Direct Drive Technology Ltd.

Acknowledgments

We thank to the Direct Drive Technology Ltd. for their technical support.

Conflict of interest

YZ was employed by Direct Drive Technology 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.

The authors declare that this study received funding from Direct Drive Technology Ltd. The funder had the following involvement with the study: providing DDT robot platform.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2022.1066714/full#supplementary-material

References

Bjelonic, M., Bellicoso, C. D., De Viragh, Y., Sako, D., Tresoldi, F. D., Jenelten, F., et al. (2019). Keep Rollin' - whole-body motion control and planning for wheeled Quadrupedal robots. *IEEE Robot. Autom. Lett.* 4, 2116–2123. doi: 10.1109/LRA.2019.2899750

Boston Dynamics Ltd (2017). *Handle*. Available online at: https://www.bostondynamics.com/handle (accessed September 22, 2022).

Chen, H., Wang, B., Hong, Z., Shen, C., Wensing, P. M., and Zhang, W. (2021). Underactuated motion planning and control for jumping with wheeled-bipedal robots. *IEEE Robot. Autom. Lett.* 6, 747–754. doi: 10.1109/LRA.2020.3047787

Cheng, W. (2021). *Leg structure optimization and landing stability analysis of grasshopper bionic robot*. Zhuzhou: Hunan University of Technology. doi: 10.27730/d.cnki.ghngy.2021.000181

Ding, D. D., and Zhang, S. J. (2022). Design and research of a variable-diameter wheel-legged obstacle overcoming robot. *Mech. Sci. Technol. Aerosp. Eng.* 1–8. doi: 10.13433/j.cnki.1003-8728.20220014 [Epub ahead of print].

Fei, L., Liu, W., Xin, F., Bonsignori, G., Scarfogliero, U., Stefanini, C., et al. (2012). Jumping like an insect: Design and dynamic optimization of a jumping mini robot based on bio-mimetic inspiration - sciencedirect. *Mechatronics* 22, 167–176. doi: 10.1016/j.mechatronics.2012.01.001

Gao, Z. W., Dai, X. W., and Zheng, Z. D. (2020). Optimal energy consumption trajectory planning for mobile robot based on motion control and frequency domain analysis. *Acta Autom. Sinica* 46, 934–945. doi: 10.16383/j.aas.c180399

Hao, Y., Lu, B., Cao, H., Dong, J., and Liu, R. (2022). "Run-and-jump Planning and control of a compact two-wheeled legged robot," in *Proceedings of the 2022*

7th Asia-Pacific conference on intelligent robot systems (ACIRS), Tianjin, 1–6. doi: 10.1109/ACIRS55390.2022.9845589

Kim, Y.-S., Jung, G.-P., Kim, H., Cho, K.-J., and Chu, C.-N. (2014). Wheel transformer: A wheel-leg hybrid robot with passive transformable wheels. *IEEE Trans. Robot.* 30, 1487–1498. doi: 10.1109/TRO.2014.2365651

Klemm, V., Morra, A., Gulich, L., Mannhart, D., Rohr, D., Kamel, M., et al. (2020). LQR-assisted whole-body control of a wheeled bipedal robot with kinematic loops. *IEEE Robot. Autom. Lett.* 5, 3745–3752. doi: 10.1109/LRA.2020.2979625

Klemm, V., Morra, A., Salzmann, C., Tschopp, F., Bodie, K., Gulich, L., et al. (2019). "Ascento: A two-wheeled jumping robot," in *Proceedings of the 2019 international conference on robotics and automation (ICRA)*, Montreal, QC, 7515–7521. doi: 10.1109/ICRA.2019.8793792

Liu, T., Zhang, C., Song, S., and Meng, M. Q.-H. (2019). "Dynamic height balance control for bipedal wheeled robot based on ROS-Gazebo," in *Proceedings of the 2019 IEEE international conference on robotics and biomimetics (ROBIO)*, Dali, 1875–1880. doi: 10.1109/ROBIO49542.2019.8961739

Liu, T., Zhang, C., Wang, J., Song, S., and Meng, M. Q.-H. (2022). Towards terrain adaptablity: In situ transformation of wheel-biped robots. *IEEE Robot. Autom. Lett.* 7, 3819–3826. doi: 10.0031109/LRA.2022.3148486

Wu, K. (2020). *Research on motion control method of biped-wheeled roboton complex terrain environment*. Jinan, Shandong University. doi: 10.27272/d.cnki.gshdu.2020.005315

Xin, S., and Vijayakumar, S. (2020). "Online dynamic motion planning and control for wheeled biped robots," in *Proceedings of the 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Las Vegas, NV, 3892–3899. doi: 10.1109/IROS45743.2020.9340967

Xin, Y. X., Chai, H., Li, Y. B., Rong, X. W., Li, B., and Li, Y. (2019). Speed and acceleration control for a two wheel-leg robot based on distributed dynamic model and whole-body control. *IEEE Access* 7, 180630–180639. doi: 10.1109/ACCESS.2019.2959333

Xin, Y. X., Li, Y. B., Chai, H., Rong, X. W., and Li, B. (2020). Research on jumping method of two wheeled-leg robot based on whole-body torque control. *Acta Autom. Sinica* 46, 1–9. doi: 10.16383/j.aas.c200486

Zhang, C., Liu, T., Song, S., and Meng, M. Q.-H. (2019). "System design and balance control of a bipedal leg-wheeled robot," in *Proceedings of the 2019 IEEE international conference on robotics and biomimetics (ROBIO)*, Dali, 1869–1874. doi: 10.1109/ROBIO49542.2019.8961814

Zhang, X. F., Zhang, Y. H., Sun, H. X., Jia, Q. X., and Xiao, H. (2014). Design and dynamics analysis of a two-wheel robot with hopping ability. *Robotics* 36, 355–361.

Zhang, Y., Zhang, L., Wang, W., Li, Y., and Zhang, Q. (2018). Design and implementation of a two-wheel and hopping robot with a linkage mechanism. *IEEE Access* 6, 42422–42430. doi: 10.1109/ACCESS.2018.2859840

Zhuang, Y., Xu, Y., Huang, B., Chao, M., Shi, G., Yang, X., et al. (2021). "Height control and optimal torque planning for jumping with wheeled-bipedal robots," in *Proceedings of the 2021 6th IEEE international conference on advanced robotics and mechatronics (ICARM)*, Chongqing, 477–482. doi: 10.1109/ICARM52023.2021.9536196

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 20 December 2022
DOI 10.3389/fnbot.2022.1086578

OPEN ACCESS

EDITED BY
Yan WU,
Institute for Infocomm Research
(A*STAR), Singapore

REVIEWED BY
Xinxing Chen,
Southern University of Science and Technology, China
Ningbo Yu,
Nankai University, China

*CORRESPONDENCE
Wei Dong
dongwei@hit.edu.cn

RECEIVED 01 November 2022
ACCEPTED 23 November 2022
PUBLISHED 20 December 2022

CITATION
Wang J, Wu D, Gao Y and Dong W (2022) Interaction learning control with movement primitives for lower limb exoskeleton.
Front. Neurorobot. 16:1086578.
doi: 10.3389/fnbot.2022.1086578

Interaction learning control with movement primitives for lower limb exoskeleton

Jiaqi Wang, Dongmei Wu, Yongzhuo Gao and Wei Dong*

State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin, China

Research on robotic exoskeletons both in the military and medical fields has rapidly expanded over the previous decade. As a human–robot interaction system, it is a challenge to develop an assistive strategy that makes the exoskeleton supply efficient and natural assistance following the user's intention. This paper proposed a novel interaction learning control strategy for the lower extremity exoskeleton. A powerful representative tool probabilistic movement primitives (ProMPs) is adopted to model the motion and generate the desired trajectory in real-time. To adjust the trajectory by the user's real-time intention, a compensation term based on human–robot interaction force is designed and merged into the ProMPs model. Then, compliant impedance control is adopted as a low-level control where the desired trajectory is put into. Moreover, the model will be dynamically adapted online by penalizing both the interaction force and trajectory mismatch, with all the parameters that can be further learned by learning algorithm $\mathrm{PI}^{\mathrm{BB}}$. The experimental results verified the effectiveness of the proposed control framework.

KEYWORDS

human–robot interaction, hierarchical control, lower limb exoskeleton, reinforcement learning, movement primitives

Introduction

Technological improvements have led to the prosperous development of lower extremity exoskeletons for the physical assistance and recovery of human locomotion since the 1960s (Mosher, 1967). Major gains in robotic hardware, electronics, actuators, sensors and energy supplies have propelled the use and acceptance of viable prototypes further. A significant issue that still remains is how to effectively control the exoskeletons to maximize the benefits of these robotic devices. Unlike with other technologies, there is not a general convergence of solutions for exoskeleton control as a very wide variety of controls are used (Young and Ferris, 2016). The intended use and the target individuals vary, as well as the development of a single control strategy for each particular design. Therefore, the control strategies should be considered from the actual application requirements.

For the exoskeletons that are used for walking assistance in daily living for able-bodied and elderly individuals, which is the focus of this paper, the aim of control is not only to provide appropriate assistance but also to make the robots actively cooperate with the human user. In this case, it is important for the exoskeleton to possess the cognition of human movement and encode it intelligently in order to achieve flexible and coordinated human-robot cooperation.

Human movement modeling has been extensively investigated by researchers in the field of bipedal walking. The most classic strategy that is based on dynamic model and stability criteria, like the link model (Hirai et al., 1998; Fu and Chen, 2008), inverted pendulum model (Yokoi et al., 2001; Komura et al., 2005; Kazemi and Ozgoli, 2019), zero-moment point (Kagami et al., 2002; Vukobratović and Borovac, 2004; Huang and Nakamura, 2005; Al-Shuka et al., 2016; He et al., 2017) have been widely used. This kind of method has over-reliance on the model, mostly expensive computation, and poor adaptability to the environment (Kazemi and Ozgoli, 2019). Besides, the exoskeletons are wearable and literally work in parallel with humans which leads to higher requirements for flexibility. In order to encode and reproduce human motion rather than just copy, the approaches learning from the demonstration have gained considerable interest in robot systems (Deng et al., 2018; Yang et al., 2018).

In our case, lower limb exoskeletons can possess a better understanding of human behavior and reproduce it by learning human movements. Movement primitives (MPs) is a well-established approach to modular robot movement (Schaal et al., 2003; Schaal, 2006; Kulić et al., 2012). Dynamic movement primitives (DMPs) presented by Ijspeert et al. (2002, 2013) has been widely used in exoskeleton systems (Huang et al., 2018; Yang et al., 2018). In Huang et al. (2018), motion trajectories are modeled with DMPs and learned with locally weighted regression method. Except for a powerful representative model, it is also necessary that the model should be adjustable online, for the benefit of the different users (Tran et al., 2014), and to reduce the effect of uncertainties and disturbances. The exoskeleton is required to continuously improve the trajectory generation by optimizing the objective function. Reinforcement learning (RL) (Schmidhuber, 2015) is one of the most general frameworks of learning control to provide truly self-autonomous learning systems. PI^2 (Theodorou et al., 2010) is a reinforcement learning policy improvement algorithm that combines optimal control and dynamic programming. Lots of works have illustrated the functionality of PI^2 in a complex robot learning scenario (Tran et al., 2014; Huang et al., 2018), it offers currently one of the most efficient, numerically robust, and easy to implement algorithms for RL. Yuan et al. (2019) proposed a trajectory-learning scheme based on PI^2 combined with DMP for motion generation.

The combination of DMPs and PI^2 performs well, but there's still room for improvement. DMPs has some limitations, like the generalization to new situations (Paraschos et al., 2018). To this effect, a novel ProMPs is proposed by Paraschos et al. (2013, 2018), which incorporates a variety of advantages from other well-known previous MP representations (d'Avella and Bizzi, 2005; Schaal et al., 2005; Kober et al., 2010). As for PI^2, the exploration and parameter update methods are slightly complicated for an online system. Stulp and Sigaud (2012) proposed a new algorithm PI^{BB}, which is a simplified version of PI^2 but has better performance. PI^{BB} belongs to black-box optimization in essence, and it has been proven that PI^{BB} outperform PI^2 in terms of convergence speed and final cost. Therefore, in our previous work, ProMPs combined with PI^{BB} are innovatively adopted to model the motion in lower limb exoskeletons, and the effectiveness has been verified under zero-mode control that the motion generation process is more quickly and accurately.

In this paper, we present an interaction learning control strategy for the lower limb exoskeleton, which is based on previous work motion generation research. The motion learning part is still based on the powerful representative tool ProMPs to generate desired trajectories. For considering the current intention of the user, the human–robot interaction (HRI) shouldn't be ignored in assistive mode control, which is an important indicator of the naturalness and comfort of the exoskeleton HRI system. Therefore, we integrate the real-time HRI force into motion online generation. In specific, a compensation term modeled by HRI force, which can reflect the user's current intention, is designed and incorporated in ProMPs. Also, the learning algorithm PI^{BB} is adopted to tune the parameters of the whole new model for different gait patterns. In this way, the exoskeleton will not only have a better understanding and reproduction of human motion, but also can quickly respond to the new situation. And based on that, to complete the entire hierarchical control for assistive mode, then the very efficient and often adopted method, impedance control, is used in low-level control, endowing compliance between the exoskeleton actuators and the user (Hogan, 1984). Experimental results reveal that our method can model the present motion more precisely and quicken the convergence of HRI force. The performance of the method meets the practical requirements in the application of the lower limb assistant exoskeleton.

The structure of this paper is organized as follows. Section Methods introduces the process and the details of the proposed interactive motion learning strategy. In Section Experiments and analyses, the experiments are carried out, and the results are presented and analyzed. Finally, the paper is concluded in Section Conclusion and future work.

Methods

This section presents the methodology details of the proposed interaction learning control strategy. Figure 1 illustrates the whole framework, which can be seen as a

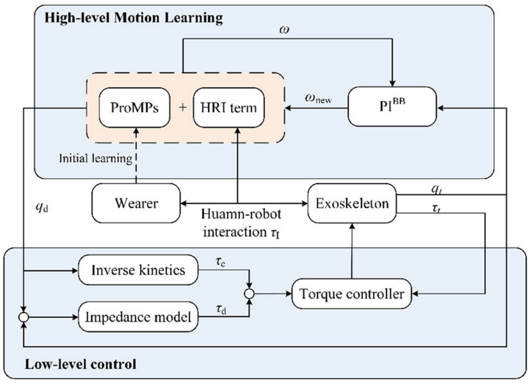


FIGURE 1
The interaction learning control framework.

hierarchical control. In high-level motion learning, the trajectory generation ProMPs model is firstly built in offline initial modeling. Besides, the model is incorporated with a compensation term, which is a transformation of HRI force that can reflect the user's current intention. In the actual application, the real-time position will be put into PI$^{\text{BB}}$ and optimize the model further. Afterward, the low-level control works on a new generated trajectory. The following Motion generation model, Motion model compensation, and Motion model adaptive learning section introduces motion learning sequential, and the Section Low-level control is the low-level control part. The meaning of the letters in the figure will be introduced in the corresponding part.

Motion generation model

The fundamental of the motion learning part is ProMPs model. The concept of ProMPs introduced in d'Avella and Bizzi (2005) is a probability distribution over robot trajectories. To establish ProMPs motion model, the first is the representation of the probabilistic trajectory. A joint of the limb of the exoskeleton corresponds to a degree of freedom. To facilitate the description of the robot's motion trajectory distribution, q_t and$\dot{q}_t$are used to respectively represent the joint angular position and joint angular velocity of each degree of freedom at time t. A movement trajectory of time length T is modeled by $y_t = \{q_t\}_{t=1}^{T}$. For encoding the time-varying variance of movements, ω is used to compactly represent a trajectory as an underlying weight vector. The trajectory is given as a linear basis function model,

$$y_t = \begin{bmatrix} q_t \\ \dot{q}_t \end{bmatrix} = \Phi_t^T \omega + \varepsilon_y \tag{1}$$

where $\Phi_t = [\phi_t, \dot{\phi}_t]$ is the $N \times 2$ dimensional time-dependent basis matrix, N defines the number of basis functions of each degree of freedom. $\varepsilon_y \sim N(0, \sum_y)$ Gaussian noise with 0 mean.

Temporal modulation is needed for modeling the human walking motion because the speed of walking is not fixed. A phase variable z is introduced to separate the movement from the time signal. The phase can be any function that monotonically increases with time z_t, and the speed of the movement can be modulated by modifying the rate of the phase variable. In this paper, z_t is adopted as equation (5),

$$z_t = \alpha t. \tag{2}$$

At the beginning of the gait movement, phase z_0 is defined as 0 and $z_{\text{E}} = 1$ at the end. The basis function φ_t now directly depends on the phase instead of time,

$$\phi_t = \phi(z_t). \tag{3}$$

The choice of the basis functions depends on the type of movement. For human walking motion, the movement of the joint is more like a rhythmic movement rather than a stroke-based. Hence, Von-Mises basis functions b_i (Spiegelhalter et al., 2002) is used to model periodicity in the phase variable z,

$$b_i(z_t) = \exp\left(\frac{\cos(2\pi(z_t - c_i))}{h}\right), \tag{4}$$

where h denotes the width of the basis and c_i is the center of the ith basis function. Then it is normalized by

$$\phi_i(z_t) = \frac{b_i(z_t)}{\sum_{j=1}^{N} b_j(z_t)}. \tag{5}$$

Then based on Paraschos et al. (2013) the probability of observing a trajectory y_t is introduced as,

$$p(y_t|\omega) = \prod_{t=1}^{T} \mathcal{N}\left(y_t \middle| \Phi_t\omega, \sum{}_y\right). \tag{6}$$

The probability distribution equation (6) depends on the parameter vector ω. Therefore, vector ω is the essential parameter for describing the trajectory, and what we mainly working on in this paper. In specific, according to Paraschos et al. (2013), a Gaussian distribution is assumed $p(\omega\,;\theta) = \mathcal{N}(\omega|\mu_\omega, \sum\omega)$ with parameters θ to capture the variance of the trajectories by learning it. $\theta = \{\mu_\omega, \sum\omega\}$ is a set of parameters that specifies the mean and the variance of ω, which capture the similarities and differences of different realizations of the MPs.

To generate more reasonable motion, $p(\omega\,;\theta)$ need to be learned from multiple demonstrations. Assuming there are M demonstration trajectories, M sets of weight vectors can be obtained by linear fitting of the basis function. In this case, the weights for each trajectory are estimated using linear ridge regression,

$$\omega_m = \left(\Phi^T\Phi + \lambda I\right)^{-1}\Phi^T Y_m. \tag{7}$$

where Y_m represents the position of all steps for the mth demonstration trajectory, and $\lambda = 0.1$. Then the parameters $\theta = \{\mu_\omega, \sum_\omega\}$ are obtained by the maximum likelihood estimation algorithm. The mean μ_ω and covariance $\sum_\omega$ are computed from samples ω_m,

$$\begin{cases} \mu_\omega = \frac{1}{M}\sum_{m=1}^{M}\omega_m \\ \sum_\omega = \frac{1}{M-1}\sum_{m=1}^{M}\left(\omega_m - \mu_\omega\right)\left(\omega_m - \mu_\omega\right)^{\mathrm{T}} \end{cases} \tag{8}$$

Now the trajectory distribution $p\left(y_t\,;\theta\right)$ can be defined by the hierarchical Bayesian model whose parameters are given by the parameters θ and the observation noise variance $\sum_y$,

$$\begin{aligned} p\left(y_t\,;\theta\right) &= \int p\left(\tau\,|\omega\right)p\left(\omega\,;\theta\right)d\omega \\ &= \int \mathcal{N}\left(y_t\,|\Phi_t\omega, \Sigma_y\right)\mathcal{N}\left(\omega|\mu_\omega,\, \Sigma_\omega\right)d\omega \\ &= \mathcal{N}\left(y_t|\Phi_t\mu_\omega,\, \Phi_t\sum\nolimits_\omega \Phi_t^T + \Sigma_y\right). \end{aligned} \tag{9}$$

Motion model compensation

MPs is a well-established approach for representing modular robot movement generators, due to their compact representation of the inherently continuous and high-dimensional robot movements. However, the wearable lower limb exoskeleton is a typical human-in-loop human-robot coupled system, so we should adapt ProMPs model to our local conditions by means of combining HRI with it. This combination can cooperate the human and exoskeleton together. Besides, HRI force is the most intuitive and practical way to estimate the user's intention, so the user's intentions are considered when generating trajectories. In this paper, the interaction between the user and the exoskeleton has been modeled, and then innovatively incorporated into the ProMPs as a compensation term. Then the trajectory generation for the current step can be affected by the HRI from the last step. Therefore, the linear basis function model of the trajectory becomes,

$$y_t^{\mathrm{I}} = \begin{bmatrix} q_t \\ \dot{q}_t \end{bmatrix} = \Phi_t^T\omega + \Psi_t^T\omega_{\mathrm{I}} + \varepsilon_y \tag{10}$$

Where $\Psi_t = [\varphi_t, \dot{\varphi}_t]$ is an $N \times 2$ dimensional time-dependent basis function matrix. Gaussian function is adopted in here.

The weight vector ω_{I} is associated with the interaction force τ_{I} on each joint, then the trajectory generation will be affected by the real-time HRI force. In order to compensate for the track position more reasonably, discretize the interaction force into a form corresponding to ω_{I} by zero-order holder (ZOH)

$$\tau_{\mathrm{I}}^{\mathrm{D}} = \tau_{\mathrm{I}}\left(1\left(t\right) - 1\left(t - {}^{T_{\mathrm{L}}}\!/_{N}\right)\right) \tag{11}$$

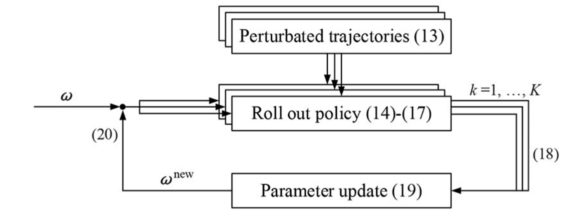


FIGURE 2
PI$^{\mathrm{BB}}$ policy improvement process. (13)-(20) Are corresponding formulas.

Where $1\left(\cdot\right)$ is a unit step function, and T_{L} is the period of the last gait step. There are total N force values of $F_{\mathrm{I}}^{\mathrm{D}}$ that are arranged in order. These values are denoted as a vector $F_{\mathrm{I}}^{\mathrm{D}}$. Now, we can obtain the weight vector ω_{I} as follows,

$$\omega_{\mathrm{I}} = \alpha_{\mathrm{I}}\tau_{\mathrm{I}}^{\mathrm{D}} \tag{12}$$

Where α_{I} is the scaling factor.

Motion model adaptive learning

Even though the trajectory can be compensated in real-time with HRI force, it still lacks agility when the system faces very different gait patterns. The exoskeleton is required to continuously improve the trajectory generation by optimizing the objective function. Hence, the decisive parameters ω in the motion generation model need to be updated constantly.

Figure 2 is the PI$^{\mathrm{BB}}$ motion learning policy improvement process. One execution of a policy is called a 'roll-out'. In each iteration, the policy is perturbed and executed K times. A total of K alternative trajectories with slightly different is randomly generated around the last optimal trajectory. Based on these trajectories, policy improvement methods then update the parameter vector $\omega \rightarrow \omega^{\mathrm{new}}$ such that the policy is expected to incur lower costs.

First, the policy parameter perturbation during a roll-out is generated from the model of the trajectory with noise

$$y_t^{'} = \Phi_t^T\left(\omega + \varepsilon\right) + \Psi_t^T\omega_{\mathrm{I}} \tag{13}$$

Where ε is interpreted as a constant exploration noise.

The immediate cost function is the mismatch of the time point positions of trajectories,

$$r_t = \left(q_t^{'} - q_t^d\right)^2 \tag{14}$$

where q_t^d represent the joint angle and the actual position of the last gait step. The trajectory cost R is,

$$R = \sqrt{\frac{1}{E}\sum\nolimits_{t=1}^{E} r}. \tag{15}$$

The cost function formula of each of the *k*th roll-out trajectory is computed with noise,

$$M_{t,k} = \frac{R^{-1}\Phi_{t,k}\Phi_{t,k}^{\mathrm{T}}}{\Phi_{t,k}^{\mathrm{T}}R^{-1}\Phi_{t,k}}, \tag{16}$$

$$S_k = \phi_{E,k} + \sum\nolimits_{t=0}^{E-1} r_{t,k} + \frac{1}{2}\sum\nolimits_{t=1}^{E-1}\left(\omega + M_{t,k}\varepsilon_k\right)^{\mathrm{T}}, \tag{17}$$

where $M_{t,k}$ is a projection matrix onto the range space of Φ_t under the metric R^{-1}. $\phi_{E,k}$ is the terminal cost of *k*th trajectory, and $r_{t,k}$ is the immediate cost of *k*th trajectory at time *t*.

The probability of the *k*th roll-out trajectory is obtained by mapping the cost of each trajectory to [0,1] through the Softmax function, as shown in equation (18),

$$P_k = \frac{e^{-\frac{1}{\gamma}S_k}}{\sum_{k=1}^{K}\left[e^{-\frac{1}{\gamma}S_k}\right]}, \tag{18}$$

where the parameter γ is a constant coefficient within (0, 1]. It can be seen from equation (18) that the higher the cost, the lower the probability, thus ensuring $\mathrm{PI}^{\mathrm{BB}}$ converges to the value with low cost.

For $\mathrm{PI}^{\mathrm{BB}}$, the parameters are updated based on the scalar aggregated cost. Therefore, the parameter updating through reward-weighted averaging is,

$$\delta\omega = \sum\nolimits_{k=1}^{K}\left[P_k\varepsilon_k\right]. \tag{19}$$

The final parameter updates with,

$$\omega^{\mathrm{new}} = \omega + \delta\omega. \tag{20}$$

There are many index notations in this paper, so for the convenience of the readers, they are concluded: *i*th represents *N* basis functions; The *t*th of *E* number of time steps; The *k*th of *K* roll-out trajectories; The *m*th of *M* demonstration trajectories.

Low-level control

The hierarchical control scheme needs a compliance control to work with motion learning results as a low-level control layer. The impedance control strategy emphasizes the active compliance of the exoskeleton system by establishing the dynamic relationship between the interaction force and the position. It can provide the exoskeleton with certain compliance while following the generated trajectory, and also allow the user to actively deviate from the desired trajectory to his comfortable way. In the application, the desired force τ_{d} is generated according to the position difference and the desired impedance model, then the desired force added to the compensation force τ_{c} calculated by the robot dynamics model is the joint driving force τ_{r}, as shown in Figure 1 lower-level control. In this way, the robot system exhibits the desired characteristics of the impedance model. As for the mathematical description, a typical formulation of the impedance model is

$$\tau_{\mathrm{d}} = M\left(\ddot{q}_{\mathrm{d}} - \ddot{q}\right) + B\left(\dot{q}_{\mathrm{d}} - \dot{q}\right) + K\left(q_{\mathrm{d}} - q\right) \tag{21}$$

Where *M* is the target impedance inertia parameter matrix; *B* is the damping, and *K* is the stiffness. $\ddot{q}_d, \dot{q}_d, q_d$ are the desired acceleration, velocity, and position of the exoskeleton, andare the corresponding actual values.

It can be seen from the formula that the choice of parameters directly determines the quality of the system control effect. The target impedance inertia parameter matrix *M* reflects the smoothness of the system response. *B* can reflect the energy consumed by the system. *K* measures the contact stiffness of the robot with the external environment. In our case, *M* is 1, *B* is 5 and *K* is 10.

As mentioned before, the dynamic model of the exoskeleton is required for this kind of impedance control. There are many ways to analyze robot dynamics. In this paper, the Lagrange equation is adapted which is standardized,

$$\tau_{\mathrm{c}} = H\left(q\right)\ddot{q} + C\left(q,\ddot{q}\right)\dot{q} + G\left(q\right) + \tau_f \tag{22}$$

Where $H(q)$ is the Inertia matrix, $C\left(q,\ddot{q}\right)$ is the Centrifugal force and the Coriolis force matrix, *G(q)* is about gravity. τ_f is the friction.

The human gait dynamic model is complex, and the dynamic model varies with different gait phases. For the swing phase and standing phase, the models are simplified as two connecting rods and three connecting rods respectively. The process of the calculation and identification are not exhibited here in detail.

Experiments and analyses

Hardware

To verify the control scheme, real-time implementations were performed on an exoskeleton system named HEXO. Figure 3 shows the main components of the HEXO (Wang et al., 2022). The backpack is equipped with the ARM control board (ARM-Cortex-A9, ARM, UK), the power supply, and the data acquisition card. The lower limb exoskeleton is designed as an anthropomorphic device, so it has the same DOFs as the human lower limb. There are seven DOFs of the exoskeleton in total, four of which are active DOF (hip and knee flexion/extension DOFs). The actuation system is powered by a brushless DC motor (EC 60 flat, Maxon Motor, Switzerland), which is efficient and reliable. The incremental encoder (MILE Encoder 1024 CPT, Maxon Motor, Switzerland) is integrated into the motor. The servo drivers of these motors are arranged on the

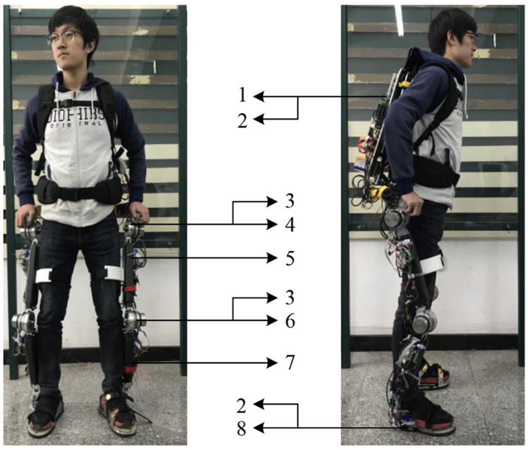


FIGURE 3
The hardware system of HEXO exoskeleton. 1. Backpack; 2. SFS; 3. Encode and torque sensor; 4. Hip joint; 5. IMU; 6. Thigh limb; 7. Knee joint; 8. Calf limb.

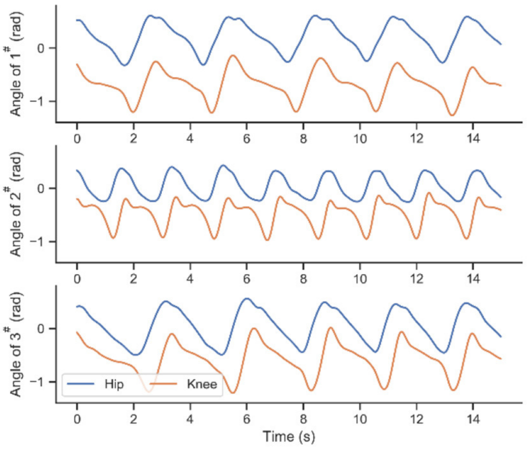


FIGURE 4
The walking trajectory data of subjects 1#, 2#, 3#.

thigh segment and the shank segment respectively (G-SOLTWI 10/100SE, ELMO, Israel). The lower limb motion is measured by the inertial measurement unit (IMU) (LPMS-CU2, ALUBI, China) mounted on the thighs and shank carbon-fiber limbs to avoid the signal burrs caused by the angular difference. The torque sensors (type: 2210C, SRI, China) are placed at hip and knee joints, which are used to measure the torque of the joint. Besides, three six-axis force sensors (SFS) (M3715D, SRI, China) are installed at the back and sensing shoes between the user and the exoskeleton to perceive the human–robot interaction force. The sensing shoes also have four load cells (AT8106, AUTODA, China) for each shoe. All sensor data is transmitted to ARM (type: ARM-Cortex-A9, ARM, UK) through the CAN (Controller Area Network) bus, whose transmission rate is up to 1Mbits.

Experimental protocol

Six healthy subjects (average height: 1.77 ± 0.07 m; averaged weight: 67.7 ± 10.1 kg) volunteered to participate in the experimental activities. As shown in Figure 1, in the process of adaptative motion learning, there is an initial modeling procedure need to be done before online application. A fundamental parameter set of ProMPs will be acquired in this offline procedure, and then the parameters will be updated in real-time. The walking data for the network modeling are collected from subjects $1^{\#}$, $2^{\#}$, and $3^{\#}$ by performing the defined track under the zero-torque mode of the exoskeleton at a self-selected pace. Figure 4 exhibited part of the walking data profiles. As can be seen, the differences between each step are inevitable in walking even though participants are asked to keep their cadences constant during the execution of their track.

For online experiments, subjects $4^{\#}$, $5^{\#}$, and $6^{\#}$ are recrewed. In order to testify the effectiveness of the proposed method, comparative experiments with different strategies are set up. Subjects $4^{\#}$, $5^{\#}$, and $6^{\#}$ are asked to perform and repeat the track under different assistive-mode utilization at a self-selected pace. In both offline and online procedures, the system works at 100Hz. The four joints of the HEXO are processed simultaneously. All angle profile figures in this paper are from the right leg.

Initial modeling

After trajectory data are obtained, the first step is to represent and model the trajectory by ProMPs. The process of ProMPs modeling the reference trajectory is described in section Motion generation model. The regression parameter λ in equation (8) is generally set to 10^{-12}, and the basic function width h in equation (4) adopts 0.05. The number of basis functions N is a crucial factor in the representative ability of primitives. Thus, one step of the walking trajectory is learned firstly for choosing an appropriate N. Figure 5A shows the learning result under different N, and Figure 5B is the RMSE (Root Mean Square Error) between the learned trajectory and target trajectory. It can be seen that the representation ability is weak when the N is small which is normal, but it grows extremely faster with the increase of N compare to other primitives. The shape of the trajectory can be approximately described by 8 basis functions with 0.017 RMSE, and converged after 15 basis functions within 0.0046 RMSE. The trajectory learned by 10 basis functions which is the purple line in Figure 5A completely coincides with the

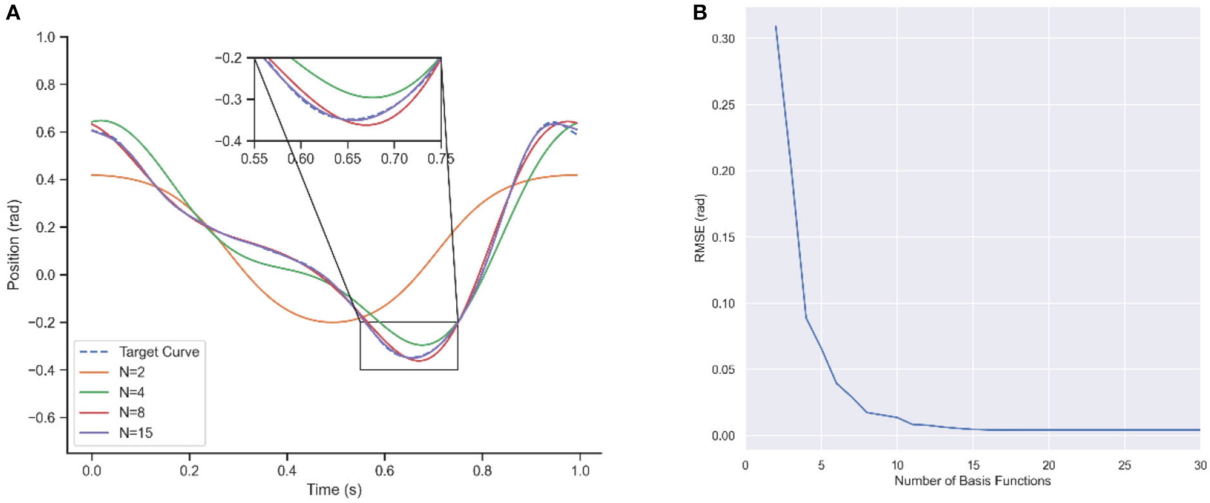


FIGURE 5
Trajectory learning results for different number of basis functions. **(A)** Curve after Learning. **(B)** RMSE of learning.

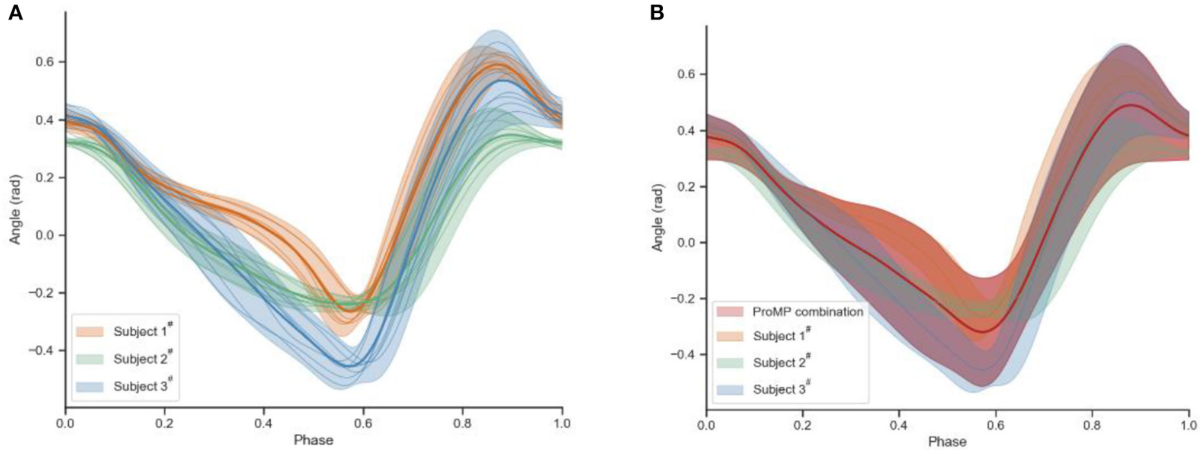


FIGURE 6
(A) The cut and normalized walking data of subjects 1#, 2#, 3#. **(B)** ProMPs learning result.

target dotted blue one. Therefore, the basis function of ProMPs is adopted 15.

Then the walking data was fed to the ProMPs that were collected from three subjects 1#, 2#, and 3# in the formal initial modeling. Figure 6A shows the mean and covariance of the walking data of three subjects' right hip, which is cut and normalized based on the gait cycle. Only the profiles of the right hip joint are shown in this paper in order to be brief. The red area of Figure 6B is the result learned by ProMPs from all three of them, which contains all their possibilities. The red line can be regarded as the average of all acquired trajectories, so it is more representative of the characteristics of human gait behavior than any other ones. Besides, the more trajectories are learned, the more general the reference trajectory is.

Experiments

For online experiments, the hardware platform is HEXO as mentioned before, embedding the corresponding control frame. The trajectory learned in section Initial modeling is regarded as the reference trajectory at the beginning. In order to test and verify the performance of the proposed method, we also set up other two baselines as comparisons. The first strategy

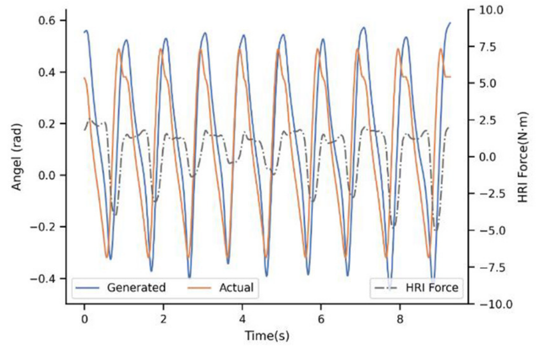


FIGURE 7
Experimental results of subject 4# with strategy P-I.

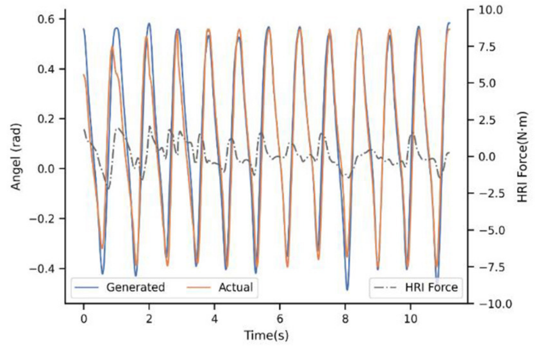


FIGURE 9
Experimental result of subject 4# with strategy PIP-I.

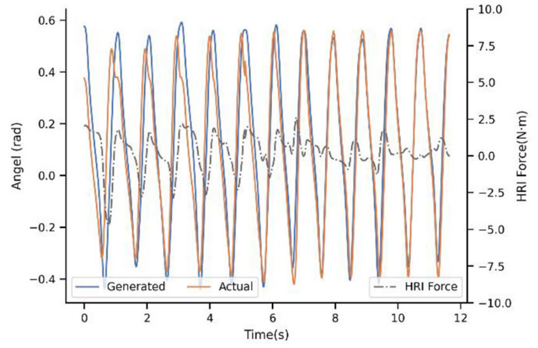


FIGURE 8
Experimental result of subject 4# with strategy PP-I.

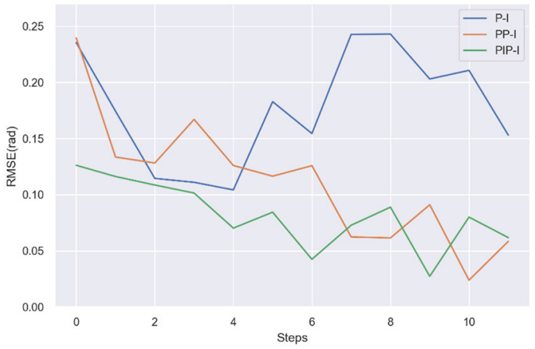


FIGURE 10
Comparison of the RMSE of the mismatch error of three strategies, P-I, PP-I, and PIP-I.

denoted as “P-I” is the exoskeleton works only with the ProMPs model as motion learning. The model is unchangeable so the desired trajectory remains the same. The impedance control provides adequate compliance which makes sure the subjects can walk naturally in the way that he wants. In the second comparison “PP-I”, the motion learning model ProMPs can be constantly updated by the $\mathrm{PI}^{\mathrm{BB}}$. The data from the actual joint trajectories are imported into the $\mathrm{PI}^{\mathrm{BB}}$ algorithm to calculate the corresponding cost value, and the $\mathrm{PI}^{\mathrm{BB}}$ adjusts the decisive parameters ω in motion learning according to the cost. Then a new trajectory with new characteristics is able to be generated. The last is the proposed interaction learning control strategy, entitle “PIP-I”. There is a dynamical interaction compensation term added into the motion learning part compared to the second “PP-I”, which is promising to reduce human–robot interaction force timely. Except that, the updating process is the same.

Figure 7 shows the result of the first baseline P-I. It compares the generated trajectory and the actual one at the beginning of the walking of subject $4^{\#}$. Note that the actual trajectory represents the intention of the users’ in this situation. In practice, the reference trajectories learned by ProMPs still differ across the users, maybe owing to the learning samples of the ProMPs model is not enough based on our limited experimental conditions. However, even though there are much more samples to learn, the difference cannot be eliminated due to different physical characteristics and the uniqueness of the human gait. In addition, it can be seen that the generated trajectory is the same for every step if there is no adaptation of the model, and when the step time between the actual and generated one are unsynchronized then the generated angel keeps the last value. Usually, the impedance control will drive the actual angle close to the ideal to a certain extent, but we ask subjects to insist on their own walking way because the ultimate goal is exoskeleton actively consistent with the humans instead of the opposite. Therefore, there are mismatches between the desired trajectory and the actual one. In this way, it will cost a lot to maintain the gait of the subjects, and the HRI is not ideal without a doubt. The HRI force is an intuitive benchmark to measure the performance of synchronization.

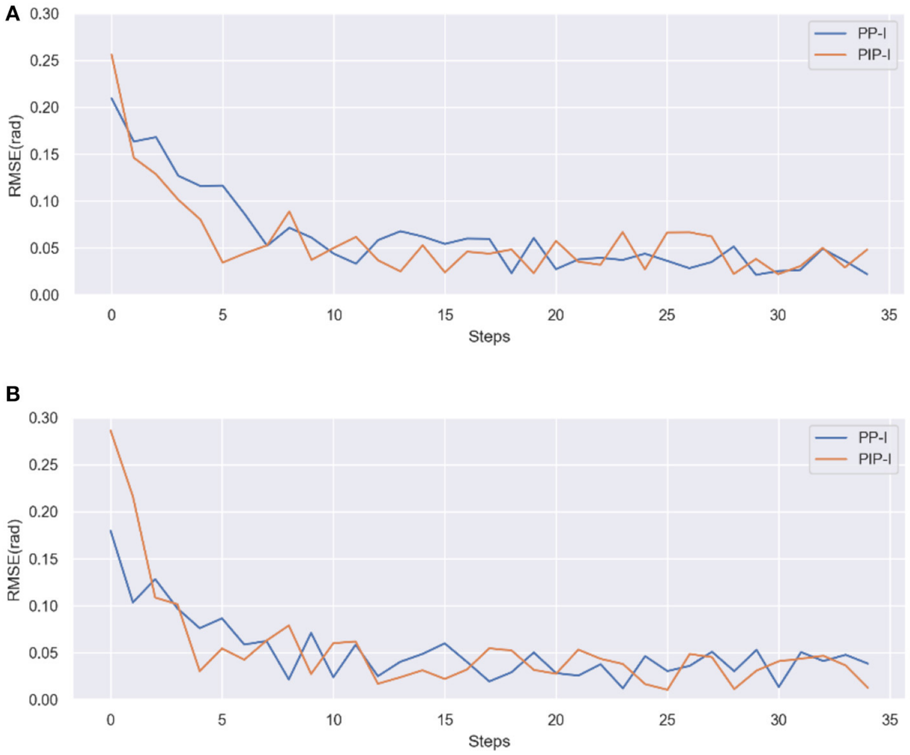


FIGURE 11
The performances of PP-I and PIP-I in RMSE of error profiles. **(A)** Subject 5#. **(B)** Subject 6#.

With regard to the second strategy PP-I, the experimental result is demonstrated in the Figure 8. The updating rate is one step, which is also the minimal unit because of the motion trajectory that ProMPs modeled is for every step. For the first step, there still is a big difference. Nevertheless, with the help of the adaptation, the generated trajectory is adjusting constantly by the $\mathrm{PI}^{\mathrm{BB}}$, and closely follows the actual after 8-9 steps. Moreover, the HRI force decreased a lot after convergence, as a result of trajectory adaptation which decreased the conflict between human and exoskeleton. That is to say, PP-I is able to provide a more coordinated HRI for the system and a more comfort assistant experience.

Up to now, the superiority of dynamic adaptation is quite apparent. Adjusting the generated trajectory constantly adds flexibility to the system, by enhancing the ability to adapt to the new situation. However, the adaptation rule is only extracted from trajectory mismatch seems inadequate, which brought the next experiment. Figure 9 illustrates the adaption performance of PIP-I, which is the proposed method in this paper. It indicates that the convergence trend is similar to PP-I, but the convergence speed is improved a lot. For subject $4^{\#}$, the convergence only takes 3-5 steps. The underlying reason is that our adaptation is extracted from a cost function that penalizes both interaction and trajectory mismatch, by adding the HRI compensation term into the motion generation model.

Figure 10 compares the average mismatch errors of three strategies. Note that, comparing the errors of the first step is of no necessity because the value is random. After the first step, the errors gradually reduced both in PP-I and PIP-I, while there is no obvious convergence in P-I. There is not much difference in steady state error of PP-I and PIP-I, but the PIP-I approaches the stable error faster. The decrease in mismatch errors indicates that adapted trajectories are more consistent with the human-exoskeleton dynamics. It is a very promising result for the proposed method, in terms of the exoskeleton system can consistent with the different users more quickly.

Figure 11 illustrates the experimental result of error profiles on subjects $5^{\#}$, $6^{\#}$. The results of baseline P-I are the same as the subject $4^{\#}$, so it will not be pointed out in further detail. For the PP-I and PIP-I, PIP-I shows the absolute advantage of convergence speed in all cases. It is hard to tell which step the errors converges to, but it is about 7-10 steps for PP-I while only 3-6 steps are needed by PIP-I for all three subjects.

The interaction force was almost proportional to the errors and followed the same trend as the errors. In Table 1, we show the average HRI force of the 4–6th step and 9–11th step respectively. We can see from the Table 1, that the HRI force

TABLE 1 The comparison of HRI force performance for strategies P-I, PP-I, and PIP-I.

Average HRI force(N m)	4–6th step			9–11th step		
	P-I	PP-I	PIP-I	P-I	PP-I	PIP-I
Subject 4#	1.21	1.20	0.74	2.09	0.48	0.56
Subject 5#	2.65	1.10	0.62	2.12	0.49	0.47
Subject 6#	1.86	0.78	0.55	1.93	0.28	0.44

with P-I is not changed as the number of steps increases. PP-I reduces the force to stable until the 9–11th step, but PIP-I can reach the steady states just with only the 4–6th steps needed. Although there is the effect of the gait randomicity, it is still can be drawn that PIP-I is able to reduce the adaptation time nearly by half. As a matter of fact, adjustment time halved has a great impact on the actual wearing experience. Moreover, according to the subjects, it is stable and comfortable when the exoskeleton works on PP-I and PIP-I, and the adaption process of PIP-I is rapid and hardly conscious. Consequently, the experiments exhibited that the proposed method can model the human trajectory and perceive the human intention online with not only low error but also, most importantly, high efficiency. With this dynamical interaction learning control, the lower limb exoskeleton can provide natural and comfortable assistance to the users in the way that he wants, and the system can also adapt to different users and situations due to the adaptive updating procedure. With the HRI compensation term, the flexibility and coordination of the human-robot system are further improved.

Conclusion and future work

We presented a novel interaction learning control framework for the lower limb exoskeleton to naturally assist people. The motion learning model generates desired trajectory online which is a combination of movement primitives and human–robot interaction force, and it is adjustable to converge to human intention and adapt to different users. We firstly adopted ProMPs to model the human motion trajectory in the lower limb exoskeleton, and in this paper, it is further integrated with HRI working on assistive mode. Furthermore, the motion learning model is constantly updated online by $\mathrm{PI}^{\mathrm{BB}}$, which can ensure the adaptability of the method to different gait patterns of various users. The experiments reveal that the proposed strategy can timely provide a smooth and natural trajectory online which is in line with the user's pattern so that the exoskeleton system could cooperate with the human user with smaller HRI. Most importantly, the convergence time is further reduced by adding the HRI compensation term, which improved the efficiency of the system and its comfort. Our analysis and experiment results show the applicability and effectiveness of the proposed method and its feasibility to be used in lower limb exoskeletons.

For this paper, the locomotion mode involved in testing is only level walking. In the future, all basic rhythmic locomotion modes in daily living will be included, such as stair ascent, stair descent, ramp ascent, and ramp descent. The performance of the learning should be similar since there is no essential difference between these motions. Besides, the situation of speed changing in walking should be taken into consideration, so the adaptability of the method for that need to be verified further.

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

The studies involving human participants were reviewed and approved by Heilongjiang Provincial Hospital. 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

JW proposed the methods and conducted the experiments and wrote the manuscript. DW, YG, and WD supervised the whole process. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the National Natural Science Foundation of China (U21A20120).

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Al-Shuka, H. F., Corves, B., Zhu, W-. H., and Vanderborght, B. (2016). Multilevel control of zero-moment point-based humanoid biped robots: a review. *Robotica*, 34, 2440. doi: 10.1017/S0263574715000107

d'Avella, A., and Bizzi, E. (2005). Shared and specific muscle synergies in natural motor behaviors. *Proc. Natl. Acad. Sci.* 102, 3076–3081. doi: 10.1073/pnas.0500199102

Deng, M., Li, Z., Kang, Y., Chen, C. P., and Chu, X. (2018). A learning-based hierarchical control scheme for an exoskeleton robot in human–robot cooperative manipulation. *IEEE Transac. Cybern.* 50, 112–125. doi: 10.1109/TCYB.2018.2864784

Fu, C., and Chen, K. (2008). Gait synthesis and sensory control of stair climbing for a humanoid robot. *IEEE Transac. Indus. Electron.* 55, 2111–2120. doi: 10.1109/TIE.2008.921205

He, W., Li, Z., and Chen, C. P. (2017). A survey of human-centered intelligent robots: issues and challenges. *IEEE/CAA J. Autom. Sinica* 4, 602–609. doi: 10.1109/JAS.2017.7510604

Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. (1998). "The development of honda humanoid robot," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146), vol. 2*. New York, NY: IEEE. 1321–1326.

Hogan, N. (1984). "Impedance control: An approach to manipulation," in *1984 IEEE American Control Conference*, 304–313.

Huang, Q., and Nakamura, Y. (2005). Sensory reflex control for humanoid walking. *IEEE Transac. Robot.* 21, 977–984. doi: 10.1109/TRO.2005.851381

Huang, R., Cheng, H., Guo, H., Lin, X., and Zhang, J. (2018). Hierarchical learning control with physical human-exoskeleton interaction. *Inf. Sci.* 432, 584–595. doi: 10.1016/j.ins.2017.09.068

Ijspeert, A. J., Nakanishi, J., and Hoffmann, H. (2013). Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Comput.* 25, 328–373. doi: 10.1162/NECO_a_00393

Ijspeert, A. J., Nakanishi, J., and Schaal, S. (2002). "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), vol. 2*. New York, NY: IEEE. 1398–1403.

Kagami, S., Kitagawa, T., Nishiwaki, K., Sugihara, T., Inaba, M., Inoue, H., et al. (2002). A fast dynamically equilibrated walking trajectory generation method of humanoid robot. *Autonom. Robot.* 12, 71–82. doi: 10.1023/A:1013210909840

Kazemi, J., and Ozgoli, S. (2019). Real-time walking pattern generation for a lower limb exoskeleton, implemented on the exoped robot. *Robot. Autonom. Syst.* 116, 1–23. doi: 10.1016/j.robot.2019.02.012

Kober, J., Mülling, K., Krömer, O., Lampert, C. H., Schölkopf, B., and Peters, J. (2010). Movement templates for learning of hitting and batting," in *2010 IEEE International Conference on Robotics and Automation*. New York, NY: IEEE. 853–858. doi: 10.1109/ROBOT.2010.5509672

Komura, T., Nagano, A., Leung, H., and Shinagawa, Y. (2005). Simulating pathological gait using the enhanced linear inverted pendulum model. *IEEE Transac. Biomed. Eng.* 52, 1502–1513. doi: 10.1109/TBME.2005.851530

Kulić, D., Ott, C., Lee, D., Ishikawa, J., and Nakamura, Y. (2012). Incremental learning of full body motion primitives and their sequencing through human motion observation. *Int. J. Robot. Res.* 31, 330–345. doi: 10.1177/0278364911426178

Mosher, R. S. (1967). *Handyman to Hardiman, Technical Report, SAE Technical Paper*. Available online at: https://doi.org/10.4271/670088.

Paraschos, A., Daniel, C., Peters, J., and Neumann, G. (2018). Using probabilistic movement primitives in robotics. *Auton. Robots* 42, 529–551. doi: 10.1007/s10514-017-9648-7

Paraschos, A., Daniel, C., Peters, J., Neumann, G., et al. (2013). Probabilistic movement primitives. *Adv. Neural Inf. Process. Syst.* 26.

Schaal, S. (2006). "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," in *Adaptive motion of animals and machines*. Berlin, Germany: Springer, 261–280.

Schaal, S., Ijspeert, A., and Billard, A. (2003). Computational approaches to motor learning by imitation. Philosophical Transactions of the Royal Society of London. *Ser. B: Biol. Sci.* 358, 537–547. doi: 10.1098/rstb.2002.1258

Schaal, S., Peters, J., Nakanishi, J., and Ijspeert, A. (2005). Learning movement primitives," in Robotics research. the eleventh international symposium. Berlin, Germany: Springer. 561–572.

Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Netw.* 61, 85–117. doi: 10.1016/j.neunet.2014.09.003

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. *J. Royal Stat. Soc. Ser. B (Stat. Methodol.)* 64, 583–639. doi: 10.1111/1467-9868.00353

Stulp, F., and Sigaud, O. (2012). *Policy Improvement Methods: Between Blackbox Optimization and Episodic Reinforcement Learning,* 34

Theodorou, E., Buchli, J., and Schaal, S. (2010). A generalized path integral control approach to reinforcement learning. *J. Mach. Learn. Res.* 11, 3137–3181. doi: 10.1007/s10846-010-9417-8

Tran, H-. T., Cheng, H., Lin, X., Duong, M-. K., and Huang, R. (2014). The relationship between physical human-exoskeleton interaction and dynamic factors: using a learning approach for control applications. *Sci. China Inf. Sci.* 57, 1–13. doi: 10.1007/s11432-014-5203-8

Vukobratović, M., and Borovac, B. (2004). Zero-moment point—thirtyfive years of its life. *Int. J. Humanoid Robot.* 1, 157–173. doi: 10.1142/S0219843604000083

Wang, J., Wu, D., Gao, Y., Wang, X., Li, X., Xu, G., et al. (2022). Integral real-time locomotion mode recognition based on GA-CNN for lower limb exoskeleton. *J. Bionic Eng.* 19, 1359–1373. doi: 10.1007/s42235-022-00230-z

Yang, C., Chen, C., Wang, N., Ju, Z., Fu, J., Wang, M., et al. (2018). Biologically inspired motion modeling and neural control for robot learning from demonstrations. *IEEE Transac. Cogn. Dev. Syst.* 11, 281–291. doi: 10.1109/TCDS.2018.2866477

Yokoi, K., Kanehiro, F., Kaneko, K., Fujiwara, K., Kajita, S., Hirukawa, H., et al. (2001). "A Honda humanoid robot controlled by aist software," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 259–264.

Young, A. J., and Ferris, D. P. (2016). State of the art and future directions for lower limb robotic exoskeletons. *IEEE Transac. Neural Syst. Rehabil. Eng.* 25, 171–182. doi: 10.1109/TNSRE.2016.2521160

Yuan, Y., Li, Z., Zhao, T., and Gan, D. (2019). Dmp-based motion generation for a walking exoskeleton robot using reinforcement learning. *IEEE Transac. Industrial Electronics*, 67, 3830–3839. doi: 10.1109/TIE.2019.2916396

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 23 March 2023
DOI 10.3389/fnbot.2023.1050167

OPEN ACCESS

EDITED BY
Yan Wu,
Institute for Infocomm Research (A*STAR), Singapore

REVIEWED BY
Ravi Shekhar Tiwari,
Chadura Tech, India
Yiannis Aloimonos,
University of Maryland, College Park, United States
Peter Sutor,
University of Maryland, College Park, United States, in collaboration with reviewer YA

*CORRESPONDENCE
Zhengping Hu
✉ hzp@ysu.edu.cn

†These authors share first authorship

RECEIVED 21 September 2022
ACCEPTED 28 February 2023
PUBLISHED 23 March 2023

CITATION
Hu Z, Mao J, Yao J and Bi S (2023) 3D network with channel excitation and knowledge distillation for action recognition.
Front. Neurorobot. 17:1050167.
doi: 10.3389/fnbot.2023.1050167

3D network with channel excitation and knowledge distillation for action recognition

Zhengping Hu[1,2*†], Jianzeng Mao[1†], Jianxin Yao[1] and Shuai Bi[1]

[1]School of Information Science and Engineering, Yanshan University, Qinhuangdao, China, [2]Hebei Key Laboratory of Information Transmission and Signal Processing, Qinhuangdao, China

Modern action recognition techniques frequently employ two networks: the spatial stream, which accepts input from RGB frames, and the temporal stream, which accepts input from optical flow. Recent researches use 3D convolutional neural networks that employ spatiotemporal filters on both streams. Although mixing flow with RGB enhances performance, correct optical flow computation is expensive and adds delay to action recognition. In this study, we present a method for training a 3D CNN using RGB frames that replicates the motion stream and, as a result, does not require flow calculation during testing. To begin, in contrast to the SE block, we suggest a channel excitation module (CE module). Experiments have shown that the CE module can improve the feature extraction capabilities of a 3D network and that the effect is superior to the SE block. Second, for action recognition training, we adopt a linear mix of loss based on knowledge distillation and standard cross-entropy loss to effectively leverage appearance and motion information. The Intensified Motion RGB Stream is the stream trained with this combined loss (IMRS). IMRS surpasses RGB or Flow as a single stream; for example, HMDB51 achieves 73.5% accuracy, while RGB and Flow streams score 65.6% and 69.1% accuracy, respectively. Extensive experiments confirm the effectiveness of our proposed method. The comparison with other models proves that our model has good competitiveness in behavior recognition.

KEYWORDS

action recognition, channel excitation, knowledge distillation, 3D convolution, deep learning

1. Introduction

With the advent of new, sophisticated deep learning architectures that are based on 3D Convolutional Neural Network variations, video processing has advanced dramatically in recent years (Diba et al., 2018; Feichtenhofer et al., 2019; Feichtenhofer, 2020; Zhu et al., 2020; Fayyaz et al., 2021). They have excelled at both the upstream and downstream tasks of video action recognition (Jiang et al., 2018; Xu et al., 2020; Zhao et al., 2020). However, it can be difficult and expensive to install these networks for inference tasks. Recent work treats recognition from motion as its objective, in which a "temporal stream" observes just a hand-designed motion representation as input, while another network, the "spatial stream," observes the raw RGB video frames (Simonyan and Zisserman, 2014). When the spatial stream is a 3D Convolutional Neural Network, however, it has Spatio-temporal filters that respond to motion in the video. This, in theory, should allow the spatial stream to learn motion properties, a notion supported by the research (Tran et al., 2015; Lee et al., 2018). However, integrating a "temporal" 3D CNN that takes an explicit motion representation, often optical flow, as input yields significant accuracy gains (Carreira and Zisserman, 2017). The method of mixing 3D CNN-based RGB and Flow streams delivers better results, but it has considerable downsides. For starters, two-stream techniques necessitate explicit and precise optical flow extraction from RGB frames, which is computationally demanding. Second, the optical flow must be evaluated before the network's forward pass can be computed. Thus, two-stream techniques not only necessitate a large number of CPU resources but also result in excessive latency when identifying actions in an online context.

In this study, we present a unique learning strategy based on channel excitation and the distillation idea that avoids flow computation at test time while maintaining the performance of two-stream approaches. To begin, we train a 3D CNN with channel excitation on RGB input that hallucinates features from the Flow stream. More specifically, we minimize the difference between high-level features from the layer preceding the network's last fully-connected layer and motion stream features at the same level (see Figure 1). In other words, our stream is architecturally and input-wise comparable to the RGB stream, but it is trained using a different loss function called the linear combination loss function. We demonstrate that by utilizing this method, flow features can be extracted from RGB frames without the need for explicit optical flow computation during inference. This network is referred to as the Intensified Motion RGB Stream for convenience (IMRS). IMRS demonstrates that by precisely simulating the Flow stream, knowledge gathered from optical flow can be efficiently transferred to a stream with RGB inputs based on 3D Convolutions. More crucially, it indicates that at test time, flow computation can be avoided. Experiments show that a network trained using our innovative approach outperforms individual RGB and Flow streams. This demonstrates how IMRS effectively uses both appearance and motion information. Specifically, IMRS achieves 74.1% accuracy on HMDB51 (split-1), whereas RGB and Flow achieve 67.6 and 70.2% accuracy, respectively. The main contributions of this paper are summarized as follows:

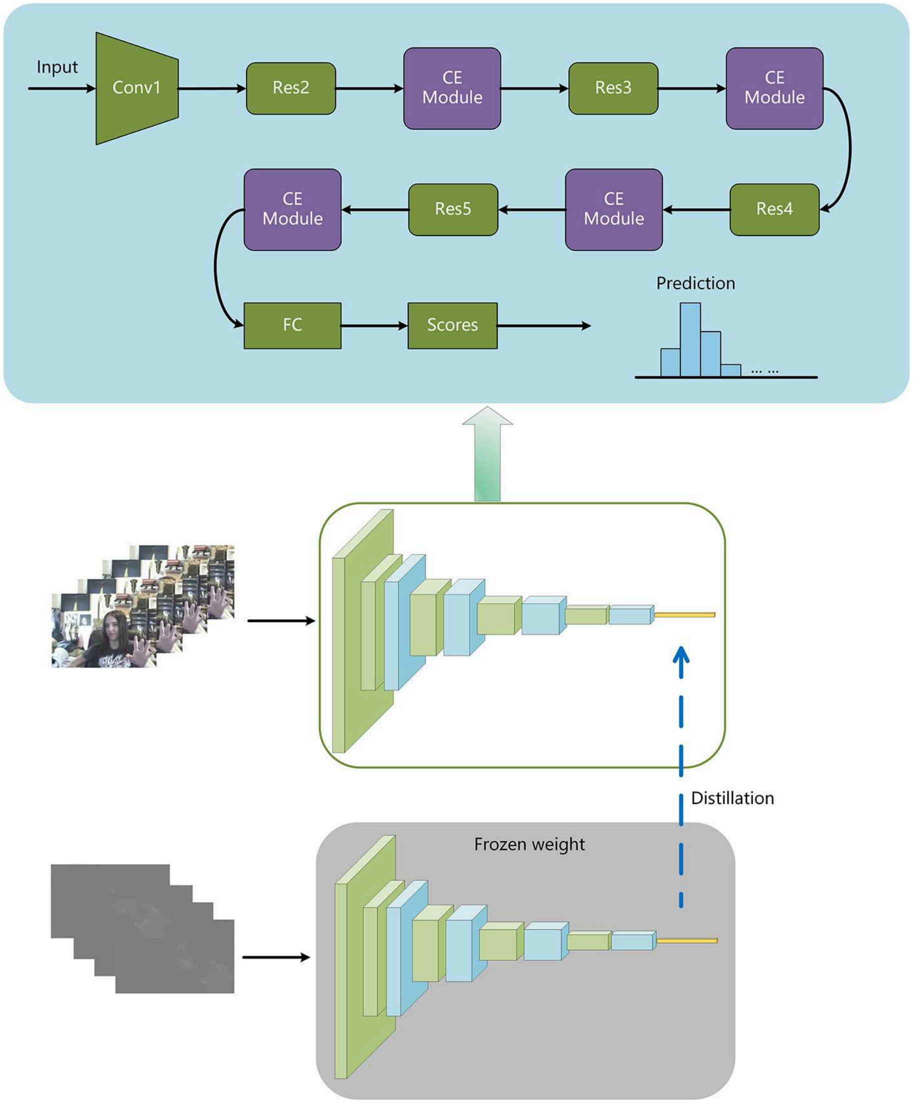


FIGURE 1
Training to make use of motion and appearance data. Initially, we use optical flow clips with cross-entropy loss to train the flow stream to identify activities and then freeze its weights. IMRS exploits both motion and appearance information by backpropagating the linear combination loss between features' overall network levels.

1. A channel excitation module CE is presented to improve the effectiveness of video frame feature representation by strengthening the channel information interaction in a 3D network.
2. To improve the knowledge distillation effect between the teacher and student models, a linear combination loss function is developed.
3. With channel excitation and knowledge distillation, we design a network. Experiments suggest that it is more successful for action recognition issues, with higher accuracy on UCF101 and HMDB51.

2. Related work

Methods for recognizing video actions can be divided into two categories. To begin, there are 2D CNN techniques that use single-frame models to process each frame independently. Second, there are 3D CNN techniques, in which a model learns video-level information through the use of 3D filters. As we will see, both types of approaches frequently employ a two-stream approach, with one stream capturing features from appearance and the other capturing data from motion. Our research focuses on Two-Stream 3D CNNs.

2.1. 2D CNNs

Many ways take advantage of the power of single-image (2D) CNNs by applying a CNN to each video frame and pooling the predictions over time (Simonyan and Zisserman, 2014; Donahue et al., 2015). However, naive average pooling overlooks the video's temporal dynamism. Two-Stream Networks incorporate a second network termed the temporal stream to capture temporal information, which accepts a sequence of successive optical flow frames as input (Simonyan and Zisserman, 2014). The outputs of these networks are then integrated using late fusion or, in certain cases, early fusion, which involves allowing the early layers of the spatial and temporal streams to interact (Feichtenhofer et al., 2016). Other methods have used other approaches to include motion, such as modifying how characteristics are pooled across time with an LSTM or CRF (Donahue et al., 2015; Sigurdsson et al., 2017). These approaches have proven to be quite effective, especially when video data is scarce and training a 3D CNN is difficult. However, recent large-scale video dataset releases have accelerated progress in 3D CNNs (Zisserman et al., 2017).

2.2. 3D CNNs

By increasing the filters to three dimensions and applying them temporally, single-frame CNNs can be generalized to video (Ji et al., 2013). Because 3D CNNs have more parameters, they require more data to train. The earliest 3D CNNs were enabled by large-scale video datasets such as Sports-1M, but they were typically not significantly more accurate than 2D CNNs applied frame-by-frame, raising the question of whether 3D CNNs model motion (Karpathy et al., 2014). To compensate, many 3D CNN systems employ additional motion-incorporation algorithms. Motion is included in C3D utilizing Improved Dense Trajectory (IDT) features, resulting in a 5.2% gain in absolute accuracy on UCF-101(Wang and Schmid, 2013; Tran et al., 2015). Using a two-stream strategy in I3D, S3D-G, and R(2+1)D results in absolute improvements of 3.1, 2.5, and 1.1% on Kinetics, respectively (Carreira and Zisserman, 2017; Tran et al., 2018; Xie et al., 2018). These studies also show that optical flow input can significantly increase 3D CNN recognition ability.

2.3. Attention mechanism

The attention mechanism can direct the model's attention to key regions and enable the enhancement of critical features, hence enhancing recognition performance. SENet (Hu et al., 2018) uses the Squeeze and Excitation (SE) module to explicitly characterize channel dependency and improve channel characteristics. Woo et al. built an attention module (Convolutional Block Attention Module, CBAM) based on the channel excitation module to perform adaptive feature refinement on the input feature map (Woo et al., 2018). Based on a self-attention mechanism, Fu et al. suggested a dual-attention network to capture feature interdependence in the spatial and channel dimensions separately (Fu et al., 2019). Chi et al. offer a Cross-Modality Attention (CMA) algorithm that allows a two-stream network to acquire information from other modalities in a hierarchical fashion (Chi et al., 2019). In this paper, we propose a channel excitation module (CE) starting from the structure of SE Block, and verify the effectiveness of the module through experiments.

2.4. Distillation

The concept of distillation is central to our proposed learning strategy. Distillation was first proposed for knowledge transfer from a complicated to a simple model by using the complex model's class probabilities as a "soft goal" for the smaller one (Hinton et al., 2015). In a similar vein, our goal is to transmit knowledge from the motion stream to a network that simply accepts RGB input and does not do explicit flow computation. In our scenario, optical flow, along with RGB, is available for training, but only RGB is available during test time.

3. Methodology

3.1. Network architecture

Figure 1 depicts the network structure described in this article. The instructor model is one of the optical flow inputs, while the RGB input is the student model. Freeze the parameters of the teacher model after training it with the cross-entropy loss, and then train the student model. During the student model's training, the optical flow is fed into the teacher model with frozen parameters, and the RGB stream is fed into the student model. The combined loss function introduced in this research is used to calculate the loss of the output of the student model's fully connected layer and the output of the teacher model's fully connected layer. Backpropagation is used in the student model to

TABLE 1 Network size information.

Layer	*N*	*F*	Output size
Input	–	3 or 2	16 × 112 × 112
Conv1	–	64	16 × 56 × 56
Max pool	–	64	8 × 28 × 28
Res2	3	256	8 × 28 × 28
CE module	–	256	8 × 28 × 28
Res3	4	512	4 × 14 × 14
CE module	–	512	4 × 14 × 14
Res4	23	1,024	2 × 7 × 7
CE module	–	1,024	2 × 7 × 7
Res5	3	2,048	1 × 4 × 4
CE module	–	2,048	1 × 4 × 4
Avgpool	–	2,048	1 × 1 × 1
FC	–	–	1 × classes

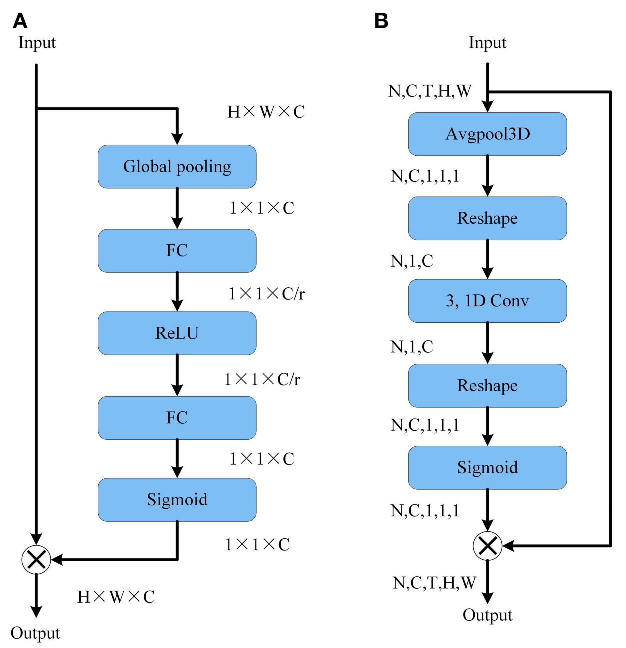


FIGURE 2
(A) SE block structure. **(B)** CE module structure.

update the parameters so that they can take advantage of motion and appearance information.

We employ 3DResNext101 (Hara et al., 2018) as the network backbone in this paper to extract deep features from input consecutive video frames. 3DResNext101 utilizes ResNet's repeated layer technique, which minimizes network complexity while boosting network width and depth to increase classification accuracy. Table 1 shows the network size after we introduced the CE module. *F* denotes the number of feature map channels, *N* is the number of residual blocks in each convolutional layer, and classes the number of action categories. The convolutional layer Conv1 is a 3D convolutional layer with a convolution kernel size of 7 × 7 × 7, 64 output channels, stride (1, 2, 2), and padding (3, 3, 3). The subsequent residual layers Res2-Res5 are stacked by residual blocks.

3.2. Channel excitation module

We insert a CE module after each residual layer of the network, hoping to increase channel excitation, to improve the network's ability to pay attention to relevant information. SE Block, which adaptively calibrates channel feature responses by explicitly modeling channel interdependencies, won first place in the ILSVRC2017 classification (Hu et al., 2018). We propose a channel excitation module (CE Module) based on the structure of the SE Block and test its usefulness through experiments.

3.2.1. Local cross-channel interaction

SE Block reduces the dimension of the channel information using the squeeze operation, then utilizes the ReLU function to execute non-linear interaction on the squeezed channel information. As illustrated in Figure 2A, an expansion operation is added to the channel information after extrusion to restore the channel number before the extrusion operation. For a given feature $y \in R^C$ without dimensionality reduction, the channel excitation can be learned by the following formula:

$$\eta = \sigma\left(W_k y\right), \tag{1}$$

where η is the channel excitation coefficient, σ is the Sigmoid function, W_k is a parameter matrix of $k \times C$ dimension and its form is as follows:

$$W_k = \begin{bmatrix} w^{1,1} & \cdots & w^{1,k} & 0 & 0 \\ 0 & w^{2,2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & w^{C,C} \end{bmatrix}. \tag{2}$$

Its meaning is: to calculate the weight w^i of the channel y_i, only considering the interaction with its k neighbors, where i is the channel number, j is the number of the neighbors. That is

$$\eta_i = \sigma\left(\sum\nolimits_{j=1}^{k} w_i^j y_i^j\right), \; y_i^j \in \Omega_i^k, \tag{3}$$

where η_i is the channel excitation coefficient numbered i and Ω_i^k represents the set of k adjacent channels of the channel y_i. One way is to have all channels share the same weights, i.e.,

$$\eta_i = \sigma\left(\sum\nolimits_{j=1}^{k} w^j y_i^j\right), \; y_i^j \in \Omega_i^k. \tag{4}$$

This method of parameter sharing can be implemented by a 1D convolution with a convolution kernel size of, i.e.,

$$\eta = \sigma\left(C1D_k(y)\right), \tag{5}$$

where $C1D$ represents 1D convolution. The method in formula (5) is implemented by the channel excitation module CE and only involves k parameters.

3.2.2. Local cross-channel interaction scope

The extent of the interactions (i.e., kernel size of 1D convolutions) must be established for the CE module to correctly capture local cross-channel interactions. Because 3 × 3 is the most commonly used convolution kernel size in 2DCNNs and the number of parameters is limited, we select $k = 3$ as the default choice. Figure 2B depicts the construction of the CE module that we employed.

3.3. Knowledge distillation and loss function

Distillation was first proposed as a method for transferring knowledge from complex to simple models by employing complex model class probabilities as "soft targets" for smaller models (Hinton et al., 2015). Crasto et al. (2019) translates knowledge from motion flow to RGB input-only networks without explicitly computing optical flow. We train an RGB-enhanced flow model with high-level optical flow characteristics (Intensified Motion RGB Stream, IMRS) using knowledge distillation with a linear combination loss function, which requires just RGB inputs during testing.

In the field of image recognition, the cross-entropy loss is frequently employed as the classification model's loss function, and it takes the following form:

$$Loss = CrossEntropy\left(s, \hat{y}\right), \tag{6}$$

where s is the class score predicted by the model and $\hat{y}$ is the true class. The proposed teacher model in this paper only uses optical flow as input, and adopts cross-entropy loss for model training, i.e.,

$$L_{Flow} = CrossEntropy\left(s_{Flow}, \hat{y}\right), \tag{7}$$

where L_{Flow} is the teacher model classification loss and s_{Flow} is the class score predicted by the teacher model. A linear combination loss function of the following form is used to train the student model using only RGB input:

$$\begin{aligned} L = {} & CrossEntropy\left(s_{RGB}, \hat{y}\right) + \lambda_1 \cdot \left\| fc_{RGB} - fc_{Flow} \right\| \\ & + \lambda_2 \cdot KL(P(fc_{RGB})||P(fc_{Flow})), \end{aligned} \tag{8}$$

where s_{RGB} is the class score predicted by the student model. CNN's first layer output represents low-level local information, while later layer outputs represent high-level global features (Zeiler and Fergus, 2014). fc_{RGB} and fc_{Flow} in formula (8) represent the high-level global features of the student model and the teacher model, λ_1 is the scalar weight that adjusts the influence of the motion feature and λ_2 is the scalar weight that adjusts the influence of the probability distribution of the motion feature. The values of λ_1 and λ_2 will be introduced in the experimental part. $KL(P(fc_{RGB})||P(fc_{Flow}))$ denotes the relative entropy (Kullback-Leibler divergence) of the RGB model's high-level feature probability distribution and the FLOW model's high-level feature probability distribution. The formula for calculating relative entropy is as follows:

$$KL(P||Q) = \sum P(x) \frac{\log(P(x))}{\log(Q(x))}, \tag{9}$$

where $P(x)$ and $Q(x)$ are the probability distributions of two discrete variables. In this paper $P(x)$ and $Q(x)$ are replaced by $P(fc_{RGB})$ and $P(fc_{Flow})$.

4. Experiments and evaluations

4.1. Datasets

We concentrate on two popular action recognition benchmarks: HMDB51 (Kuehne et al., 2011) and UCF101 (Soomro et al., 2012). HMDB51 comprises 6,849 video clips divided into 51 activity categories. Human actions, face actions, and interactive activities are all featured in the videos. UCF101 comprises 13,320 video clips with an average duration of around 7 s and 101 action categories. Human-object interaction, human-human contact, human movement, sports, and musical instrument performance are some of the activity categories covered in videos. Figure 3 shows some of the data from the above two datasets. The two datasets mentioned here were trained and evaluated using the three officially provided splits, which means that there is no intersection between the training and evaluation data. To the best of our knowledge, the same works as the dataset used in our work are (Wang et al., 2016b; Crasto et al., 2019; Li et al., 2021). The first splits of HMDB51 and UCF101 are denoted as HMDB51-1 and UCF101-1, respectively.

4.2. Implementation details

The experiments in this paper generate models and conduct research in the Python3.6 environment using the Pytorch deep learning framework. All programs are executed on a server that has a V100 GPU.

4.2.1. Data preprocessing

In this paper, the OpenCV toolkit in the Python environment is used to frame the original video dataset and extract the optical flow. In the process of video framing, all frames of the video data are extracted, and the image size of the video frame is adjusted to 256 × 256 pixels and saved in jpeg format. After the video frame is divided, the optical flow extraction operation is performed. In this paper, the TV-L1 (Zach et al., 2007) method is used to extract the optical flow, and the default parameter settings of OpenCV are used. We truncate the value of the optical flow file between −20 and 20 and map it to the (0,255) pixel range, saving it in jpeg format.

4.2.2. Training

We use consecutive 16 frames as input during training after setting (Hara et al., 2018). A random cut to 112 × 112 size is

FIGURE 3
(A) HMDB51 dataset. **(B)** UCF101 dataset.

performed on the input image, and a horizontal flip is randomly applied, which includes a random horizontal flip of the x-direction component for the optical flow input. Subtract the mean of the ActivityNet distribution for RGB input and 127.5 for FLOW input, assuming that the FLOW distribution is centered at 0. Following the settings of Hara et al. (2018), we adopt the SGD optimization method with a weight decay of 0.0005, a momentum of 0.9, and an initial learning rate of 0.1 for *ab initio* training. During the fine-

TABLE 2 The influence of CE module position and number.

Location	Number	HMDB51-1(%)
Res_2	1	65.9
Res_3	1	65.8
Res_4	1	67.0
Res_5	1	66.2
Res_{2-5}	4	**67.6**

The bold values represent the highest accuracy.

TABLE 3 The effect of the CE module.

Model	Params (M)	UCF101-1 (%)	HMDB51-1 (%)
3DResNeXt101	48.34	90.7	63.8
3DResNeXt101+SE	49.04	91.0	65.4
3DResNeXt101+CE	48.34	**91.8**	**67.6**

The bold values represent the highest accuracy.

tuning phase, we use a pre-trained model on the Kinetics400 dataset with a learning rate of 0.001, and when the performance no longer improves for 10 consecutive epochs, the learning rate becomes 0.1 times the previous one. The number of training cycles in this paper is 80 epochs, the batch size in the training phase is 32, and the batch size in the testing phase is 1.

4.3. Ablation experiments

4.3.1. Impact of CE module

In this study, we investigate the effect of CE module position and number on model performance using RGB input on HMDB51-1. Table 2 shows that the ideal method for embedding the CE module into the network is to add a CE module to each Res2-Res5 residual layer, for a total of 4 CE modules added to the network.

In this research, we investigate the effects of the SE block and the CE module on model performance using RGB input on UCF101-1 and HMDB51-1, respectively. It can be seen from Table 3 that the amount of model parameters hardly increases after the CE module is embedded in the network. Compared with adding the SE block to the network, the number of parameters is less and higher recognition accuracy is achieved.

4.3.2. The effect of λ_1 and λ_2

Regarding the linear combination loss function scalar weights λ_1 and λ_2 in formula (8), we conduct an experimental manual search on HMDB51-1. The experimental results are shown in Table 4. The optimal parameters obtained from the experiment are $\lambda_1 = 50, \lambda_2 = 200$.

4.4. Results and discussion

Figure 4 depicts the accuracies of the single-stream 3DResNext101+CE model on the UCF101 and HMDB51 validation sets, with average accuracy over *Top-1* and *Top-5*. The overall performance of the dataset is calculated by averaging the dataset's three parts.

TABLE 4 The effect of λ_1 and λ_2.

λ_1	λ_2	HMDB51-1(%)
1	1	73.1
10	10	73.3
50	50	72.1
50	100	73.3
50	200	**74.1**
50	300	73.0

The bold values represent the highest accuracy.

In terms of single-stream performance, the FLOW stream outperforms the RGB stream, demonstrating that motion information is more effective than appearance information for action recognition. The models all outperformed HMDB51 on UCF101, indicating that HMDB51 is more difficult to recognize actions than UCF101.

On both datasets, IMRS outperforms RGB and FLOW single-stream, indicating that the strategy of transferring motion flow information to appearance flow in the form of knowledge distillation is effective. On HMDB51-1, IMRS *Top-1* accuracy is 6.5 and 3.9% greater than pure RGB and FLOW streams, respectively. On UCF101-1, IMRS *Top-1* accuracy is 3.6 and 2.8% greater than pure RGB and FLOW streams, respectively.

4.5. Comparison with mainstream methods

Table 5 compares the model suggested in this paper to the industry standard approaches for behavior recognition. This paper's recognition effect on UCF101 and HMDB51 is the average of three splits. Our method outperforms the two-stream method in terms of recognition accuracy, but the number of parameters and computation rises due to the usage of 3D convolution.

Compared with 3D CNN methods, our method outperforms some models, such as T3D+TSN (Diba et al., 2017), and ECO.En (Zolfaghari et al., 2018), 3DCNN Ensemble+iDT (Huang et al., 2020), STDA-ResNeXt-101 (Li J. et al., 2020). Compared with MARS (Crasto et al., 2019), the channel excitation module is introduced in the network structure, which brings a small number of parameters and a small amount of calculation. MARS only reported the accuracy on split1 for 16 frames input in the original text, and the average effect of this paper on the three splits of the dataset was higher than that of MARS. Among them, the performance of S3D-G (Xie et al., 2018) on the two datasets is higher than IMRS proposed in this paper. The main reason is that S3D-G is pre-trained on ImageNet+Kinetics and needs 64-frame input. In this paper, we just use 16 frames of input and solely employ Kinetics' pretrained parameters. On UCF101 and HMDB51, S3D-G is 1.1 and 2.4% higher than ours, respectively, but the computational cost of ours is much lower than S3D-G (18.04 vs. 71.38). On HMDB51,

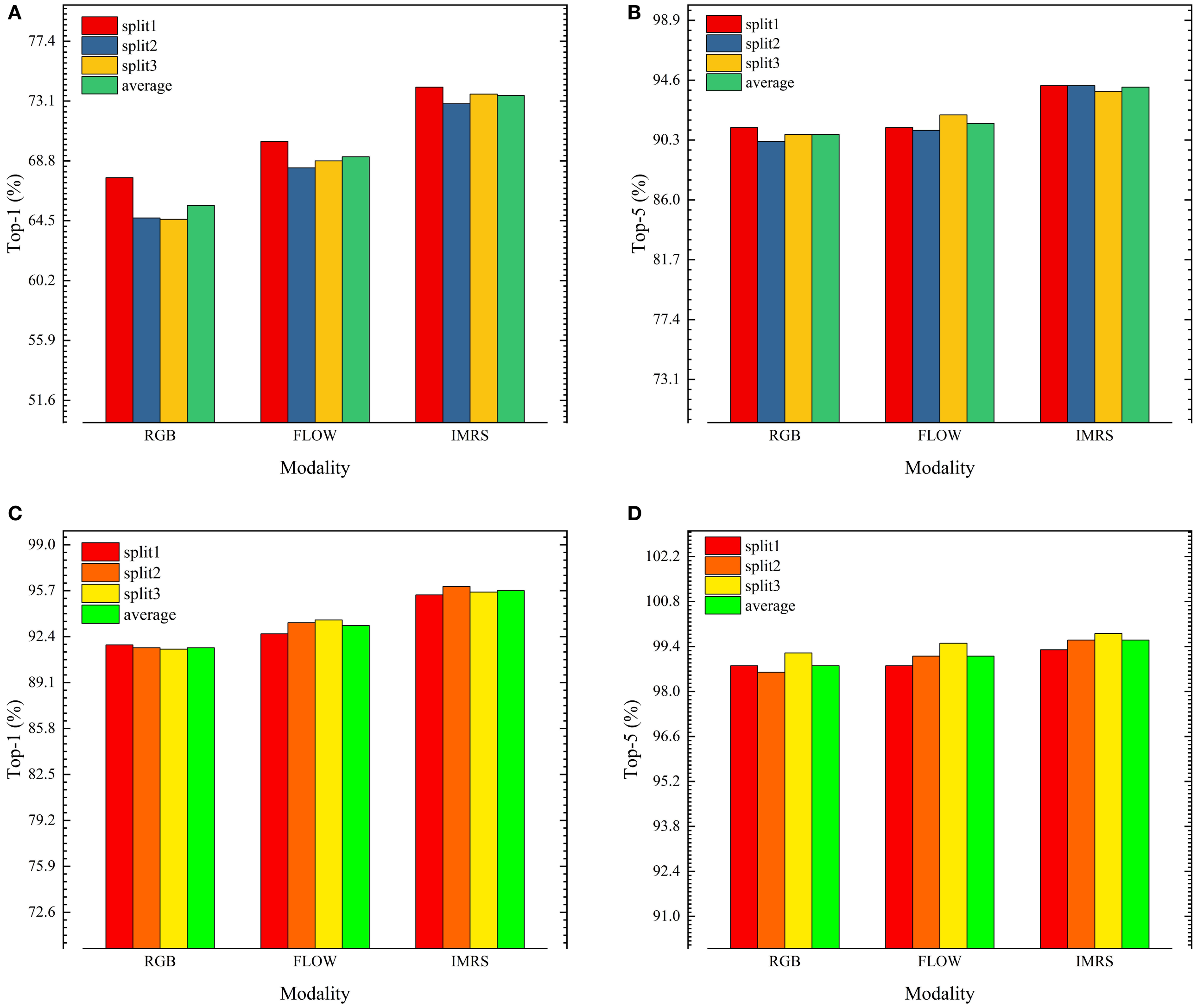


FIGURE 4
(A) HMDB51 accuracy top-1. **(B)** HMDB51 accuracy top-5. **(C)** UCF101 accuracy top-1. **(D)** UCF101 accuracy top-5.

RGB-I3D (Carreira and Zisserman, 2017) is 1.3% higher than ours, but it is higher in terms of the number of input frames and computation.

Compared with the 2D CNN method, the method in this paper has an advantage in terms of computational complexity, and the recognition accuracy is less different from TSM (Lin et al., 2019). On HMDB51, our technique has a 1.3% greater recognition accuracy than STM (Jiang et al., 2019). On UCF101, our method's recognition accuracy is lower than that of 2D CNN, but it has an absolute advantage in terms of computational complexity. Based on these findings, we can conclude that our method has some advantages over conventional methods.

5. Conclusions and discussions

We present an action recognition network based on channel excitation and knowledge distillation in this work. A channel excitation module is employed by the network to improve channel information interaction and the network's capacity to extract relevant features. At the same time, the constitutes the linear combination loss function to train the teacher-student model to improve the student model's knowledge-learning ability in comparison to the teacher model. Furthermore, during student model inference, the network only takes RGB as input, which decreases model inference delay, and demonstrates good performance on behavior recognition tasks, which has reference relevance for the research of video behavior recognition algorithms. Future studies will concentrate on reducing the number of model parameters to boost recognition accuracy yet further. At the same time, we will focus on improving the training speed of the model and reducing the reasoning time.

In this work, we apply attention mechanism and knowledge distillation to the field of behavior recognition, and the experimental results show that these methods are simple and

TABLE 5 Compared with the mainstream methods.

Method	Frame	Params (M)	FLOPs (G)	Pre train	UCF101 (%)	HMDB51 (%)
Two-stream (Simonyan and Zisserman, 2014)	1+1	12	–	I	88.0	59.4
TSN(3 modalities) (Wang et al., 2016a)	3+3	10.4	16.4	I	94.2	69.4
CMA_$iter_1$_R (Chi et al., 2019)	64	43.74	–	K	95.3	–
AMFNet-C (Liu and Ma, 2022)	6+5	–	–	I	95.9	71.2
RGB-I3D (Carreira and Zisserman, 2017)	64	12	108	I+K	95.6	74.8
T3D+TSN (Diba et al., 2017)	16	–	–	K	93.2	63.5
S3D-G (Xie et al., 2018)	64	11.56	71.38	I+K	96.8	75.9
3DResNeXt101 (Hara et al., 2018)	16	48.34	9.57	K	90.7	63.8
ECO.En (Zolfaghari et al., 2018)	{16,20,24,32}	150	267	K	94.8	72.4
MARS (Crasto et al., 2019)	16	95.23	18.03	K	94.6 (s1)	72.3 (s1)
3DCNN Ensemble+iDT (Huang et al., 2020)	36	1324.7	–	I	92.7	69.1
STDA-ResNeXt-101 (Li J. et al., 2020)	64	382	–	K	95.5	72.7
TSM (Lin et al., 2019)	8	24.3	33	I+K	95.9	73.5
STM (Jiang et al., 2019)	16	23.88	32.93	I+K	96.2	72.2
TEA (Li Y. et al., 2020)	16	–	70	I+K	96.9	73.3
CT-Net (Li et al., 2021)	16	–	145.5	I+K	96.2	73.2
IMRS (ours)	16	95.23	18.04	K	**95.7**	**73.5**

"S" stands for Sports-1M, "I" stands for ImageNet, "K" stands for Kinetics, "s1" stands for the first split and "–" means that the original text does not report this item. The bold values highlight the effect of our proposed method.

effective. We believe that this approach is not only suitable for solving action recognition problems, subsequent work could consider extending this approach to other fields, such as action prediction and anticipation (Dessalene et al., 2021), and robot self-learning (Yang et al., 2015). We will explore the application of this method to action prediction in the future.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://www.crcv.ucf.edu/data/UCF101.php and https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/.

Author contributions

ZH: conceptualization and methodology. JM: writing and original draft preparation. JY: formal analysis. SB: resources and data curation. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the National Natural Science Foundation of China General Program 61771420 and the National Natural Science Foundation of China 62001413.

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Carreira, J., and Zisserman, A. (2017). "Quo vadis, action recognition? A new model and the kinetics dataset," in *2017 IEEE Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 4724–4733. doi: 10.1109/CVPR.2017.502

Chi, L., Tian, G., Mu, Y., and Tian, Q. (2019). "Two-stream video classification with cross-modality attention," in *IEEE International Conference on Computer Vision Workshop* (Seoul), 4511–4520. doi: 10.1109/ICCVW.2019.00552

Crasto, N., Weinzaepfel, P., Alahari, K., and Schmid, C. (2019). "MARS: motion-augmented RGB stream for action recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 7874–7883. doi: 10.1109/CVPR.2019.00807

Dessalene, E., Devaraj, C., Maynord, M., Fermuller, C., and Aloimonos, Y. (2021). Forecasting action through contact representations from first person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence PP*. doi: 10.1109/TPAMI.2021.3055233. [Epub ahead of print].

Diba, A., Fayyaz, M., Sharma, V., Arzani, M. M., Yousefzadeh, R., Gall, J., et al. (2018). "Spatio-temporal channel correlation networks for action classification," in *European Conference on Computer Vision* (Munich), 299–315. doi: 10.1007/978-3-030-01225-0_18

Diba, A., Fayyaz, M., Sharma, V., Karami, A. H., Arzani, M. M., Yousefzadeh, R., et al. (2017). Temporal 3D ConvNets: new architecture and transfer learning for video classification. *arXiv Preprint.* arXiv:1711.08200. Available online at: https://arxiv.org/pdf/1711.08200.pdf

Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., et al. (2015). "Long-term recurrent convolutional networks for visual recognition and description," in *IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 2625–2634. doi: 10.1109/CVPR.2015.7298878

Fayyaz, M., Rad, E. B., Diba, A., Noroozi, M., Adeli, E., Gool, L. V., et al. (2021). "3D CNNs with adaptive temporal feature resolutions," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Montreal, QC), 4729–4738. doi: 10.1109/CVPR46437.2021.00470

Feichtenhofer, C. (2020). "X3D: expanding architectures for efficient video recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA), 200–210. doi: 10.1109/CVPR42600.2020.00028

Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). "SlowFast networks for video recognition," in *IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 6201–6210. doi: 10.1109/ICCV.2019.00630

Feichtenhofer, C., Pinz, A., and Wildes, R. P. (2016). "Spatiotemporal residual networks for video action recognition," in *Neural Information Processing Systems* (Spain: NIPS Proceedings), 3468–3476. doi: 10.1109/CVPR.2017.787

Fu, J., Liu, J., Tian, H., Fang, Z., and Lu, H. (2019). "Dual attention network for scene segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA), 3141–3149. doi: 10.1109/CVPR.2019.00326

Hara, K., Kataoka, H., and Satoh, Y. (2018). "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 6546–6555. doi: 10.1109/CVPR.2018.00685

Hinton, G. E., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *Comput. Sci.* 14, 38—39. Available online at: https://arxiv.org/pdf/1503.02531.pdf.

Hu, J., Shen, L., Albanie, S., Sun, G., and Wu, E. (2018). "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 2011–2023. doi: 10.1109/CVPR.2018.00745

Huang, Y., Guo, Y., and Gao, C. (2020). Efficient parallel inflated 3D convolution architecture for action recognition. *IEEE Access* 8, 45753–45765. doi: 10.1109/ACCESS.2020.2978223

Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3D Convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 221–231. doi: 10.1109/TPAMI.2012.59

Jiang, B., Wang, M., Gan, W., Wu, W., and Yan, J. (2019). "STM: spatiotemporal and motion encoding for action recognition," in *IEEE/CVF International Conference on Computer Vision* (Seoul), 2000–2009. doi: 10.1109/ICCV.2019.00209

Jiang, J., Cao, Y., Song, L., Zhang, S., Li, Y., Xu, Z., et al. (2018). Human centric spatio-temporal action localization. in *IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE).

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). "Large-scale video classification with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH), 1725–1732. doi: 10.1109/CVPR.2014.223

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011). "HMDB: a large video database for human motion recognition," in *International Conference on Computer Vision* (Barcelona), 2556–2563. doi: 10.1109/ICCV.2011.6126543

Lee, M., Lee, S., Son, S., Park, G., and Kwak, N. (2018). "Motion feature network: fixed motion filter for action recognition," in *European Conference on Computer Vision* (Cham: Springer International Publishing), 392–408. doi: 10.1007/978-3-030-01249-6_24

Li, J., Liu, X., Zhang, M., and Wang, D. (2020). Spatio-temporal deformable 3D ConvNets with attention for action recognition. *Pattern Recogn.* 98, 1–9. doi: 10.1016/j.patcog.2019.107037

Li, K., Li, X., Wang, Y., Wang, J., and Qiao, Y. (2021). CT-Net: channel tensorization network for video classification. *arXiv Preprint.* arXiv:2106.01603.

Li, Y., Ji, B., Xintian, S., Zhang, J., Kang, B., and Wang, L. (2020). "TEA: temporal excitation and aggregation for action recognition," in *IEEE Conference of Computer Vision and Pattern Recognition* (Seattle, WA). doi: 10.1109/CVPR42600.2020.00099

Lin, J., Gan, C., and Han, S. (2019). "TSM: temporal shift module for efficient video understanding," in *2019 IEEE/CVF International Conference on Computer Vision* (Seoul), 7082–7092. doi: 10.1109/ICCV.2019.00718

Liu, S., and Ma, X. (2022). Attention-driven appearance-motion fusion network for action recognition. *IEEE Trans. Multimedia*. 1–12. doi: 10.1109/TMM.2022.3148588

Sigurdsson, G. A., Divvala, S. K., Farhadi, A., and Gupta, A. K. (2017). "Asynchronous temporal fields for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 5650–5659. doi: 10.1109/CVPR.2017.599

Simonyan, K., and Zisserman, A. (2014). "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems* (Montreal, QC: Massachusetts Institute of Technology Press), 568–576.

Soomro, K., Zamir, A. R., and Shah, M. (2012). UCF101: a dataset of 101 human actions classes from videos in the wild. *arXiv Preprint.* arXiv:1212.0402. Available online at: https://arxiv.org/pdf/1212.0402.pdf

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). "Learning spatiotemporal features with 3D convolutional networks," in *IEEE International Conference on Computer Vision* (Santiago: IEEE), 4489–4497. doi: 10.1109/ICCV.2015.510

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). "A closer look at spatiotemporal convolutions for action recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 6450–6459. doi: 10.1109/CVPR.2018.00675

Wang, H., and Schmid, C. (2013). "Action recognition with improved trajectories," in *IEEE International Conference on Computer Vision* (Sydney, NSW: IEEE), 3551–3558. doi: 10.1109/ICCV.2013.441

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., et al. (2016a). Temporal segment networks: towards good practices for deep action recognition. *arXiv Preprint.* arXiv:1608.00859. doi: 10.1007/978-3-319-46484-8_2

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., et al. (2016b). "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition", in *European Conference on Computer Vision* (Cham: Springer International Publishing), 20–36.

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. (2018). "CBAM: convolutional block attention module," in *European Conference on Computer Vision*. (Munich).

Xie, S., Sun, C., Huang, J., Tu, Z., and Murphy, K. (2018). "Rethinking spatiotemporal feature learning: speed-accuracy trade-offs in video classification," in *European Conference on Computer Vision* (Cham: Springer International Publishing), 305–321. doi: 10.1007/978-3-030-01267-0_19

Xu, M., Zhao, C., Rojas, D. S., Thabet, A. K., and Ghanem, B. (2020). "G-TAD: sub-graph localization for temporal action detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA), 10153–10162. doi: 10.1109/CVPR42600.2020.01017

Yang, Y., Li, Y., Fermüller, C., and Aloimonos, Y. (2015). "Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web," in *AAAI Conference on Artificial Intelligence*. (Austin, TX).

Zach, C., Pock, T., and Bischof, H. (2007). "A duality based approach for realtime TV-L1 optical flow," in *DAGM Conference on Pattern Recognition*. (Heidelberg, Berlin).

Zeiler, M. D., and Fergus, R. (2014). "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*. (Zürich).

Zhao, P., Xie, L., Ju, C., Zhang, Y., Wang, Y., and Tian, Q. (2020). "Bottom-up temporal action localization with mutual regularization," in *European Conference on Computer Vision*. (Glasgow).

Zhu, S., Yang, T., Mendieta, M., and Chen, C. (2020). A3D: adaptive 3D networks for video action recognition. *arXiv Preprint.* arXiv:2011.12384. Available online at: https://arxiv.org/pdf/2011.12384.pdf

Zisserman, A., Carreira, J., Simonyan, K., Kay, W., Zhang, B. H., Hillier, C., et al. (2017). The kinetics human action video dataset. *arXiv Preprint.* arXiv:1705.06950. Available online at: https://arxiv.org/pdf/1705.06950.pdf

Zolfaghari, M., Singh, K., and Brox, T. (2018). "ECO: efficient convolutional network for online video understanding," in *European Conference on Computer Vision* (Cham: Springer International Publishing), 695–712. doi: 10.1007/978-3-030-01216-8_43

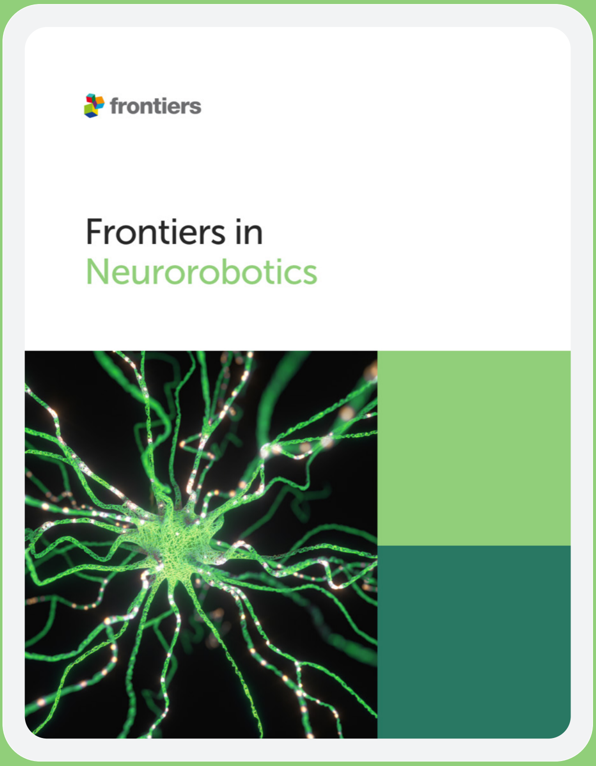
Frontiers in Neurorobotics

Investigates embodied autonomous neural systems and their impact on our lives

Part of the most cited neuroscience series, this journal advances understanding of neurorobotics - from prosthetic devices to brain machine interfaces, and wearable systems to home appliances.

Discover the latest Research Topics

See more →



frontiers

Frontiers in Neurorobotics

Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact