



OPEN ACCESS

EDITED BY

Tomohiro Shibata,
Kyushu Institute of Technology, Japan

REVIEWED BY

Bin Zhi Li,
Chinese Academy of Sciences (CAS), China
Wentai Zhang,
Peking Union Medical College Hospital
(CAMS), China

*CORRESPONDENCE

Hailong Yu
✉ 1955416359@qq.com

[†]These authors have contributed equally to
this work

RECEIVED 17 January 2025

ACCEPTED 28 March 2025

PUBLISHED 09 May 2025

CITATION

Xie Y, Wang A, Zhao X, Jiang Y, Wu Y and Yu H
(2025) Motion control and singular
perturbation algorithms for lower limb
rehabilitation robots.
Front. Neurobot. 19:1562519.
doi: 10.3389/fnbot.2025.1562519

COPYRIGHT

© 2025 Xie, Wang, Zhao, Jiang, Wu and Yu.
This is an open-access article distributed
under the terms of the [Creative Commons
Attribution License \(CC BY\)](#). The use,
distribution or reproduction in other forums is
permitted, provided the original author(s) and
the copyright owner(s) are credited and that
the original publication in this journal is cited,
in accordance with accepted academic
practice. No use, distribution or reproduction
is permitted which does not comply with
these terms.

Motion control and singular perturbation algorithms for lower limb rehabilitation robots

Yanchun Xie^{1†}, Anna Wang^{2†}, Xue Zhao³, Yang Jiang⁴, Yao Wu⁴
and Hailong Yu^{1*}

¹Department of Orthopaedics, General Hospital of Northern Theater Command, Shenyang, China,

²Department of Burns and Plastic Surgery, General Hospital of Northern Theater Command, Shenyang, China, ³Daniel L. Goodwin College of Business, Benedictine University, Chicago, IL, United States, ⁴Faculty of Robot Science and Engineering, Northeastern University, Shenyang, China

To better assist patients with lower limb injuries in their rehabilitation training, this paper focuses on motion control and singular perturbation algorithms and their practical applications. First, the paper conducts an in-depth analysis of the mechanical structure of such robots and establishes detailed kinematics and dynamics models. An optimal S-type planning algorithm is proposed, transforming the S-type planning into an iterative solution problem for efficient and accelerated trajectory planning using dynamic equations. This algorithm comprehensively considers joint range of motion, speed constraints, and dynamic conditions, ensuring the smoothness and continuity of motion trajectories. Second, a zero-force control method is introduced, incorporating friction terms into the traditional dynamic equations and utilizing the LuGre friction model for friction analysis to achieve zero-force control. Furthermore, to address the multi-scale dynamic system characteristics present in rehabilitation training, a control method based on singular perturbation theory is proposed. This method enhances the system's robustness and adaptability by simplifying the system model and optimizing controller design, enabling it to better accommodate complex motion requirements during rehabilitation. Finally, experiments verify the correctness of the kinematics and optimal S-type trajectory planning. In lower limb rehabilitation robots, zero-force control can better assist patients in rehabilitation training for lower limb injuries, while the singular perturbation method improves the accuracy, response speed, and robustness of the control system, allowing it to adapt to individual rehabilitation needs and complex motion patterns. The novelty of this paper lies in the integration of the singular perturbation method with the LuGre friction model, significantly enhancing the precision of joint dynamic control, and improving controller design through the introduction of a torque deviation feedback mechanism, thereby increasing system stability and response speed. Experimental results demonstrate significant improvements in tracking error and system response compared to traditional methods, providing patients with a more comfortable and safer rehabilitation experience.

KEYWORDS

rehabilitation robots, trajectory planning, singular perturbation, flexible control, controller design

1 Introduction

With the global aging trend and evolving lifestyles, the demand for lower limb rehabilitation following injuries has surged, driving the widespread adoption of rehabilitation robots in medical applications. Notably, these robots exhibit significant potential in lower limb rehabilitation training (Qassim and Wan Hasan, 2020). Traditional rehabilitation relies heavily on specialized therapists to manually guide patients through passive exercises, a process that struggles to ensure consistency and continuity due to the limited availability of qualified therapists and variations in their skill levels stemming from individual differences. These factors represent key bottlenecks in current rehabilitation technology. Flexible lower limb rehabilitation robots address these challenges by simulating natural human movement patterns, offering personalized training programs, and facilitating the gradual restoration of motor functions, thereby alleviating the shortage of skilled trainers. However, technical challenges persist, particularly in achieving real-time responsiveness and precise trajectory planning in practical applications. The system models of lower limb rehabilitation robots typically exhibit complex characteristics, such as high order, non-linearity, and strong coupling. Additionally, the accuracy of these models is often challenging to ensure ultimately, which complicates the control structure. However, the singular perturbation method can simplify system models (Kevorkian and Cole, 2012) by partitioning the system into fast and slow subsystems based on time scales (Yu and Chen, 2015). Subsequently, controller design is conducted separately for these two subsystems. This approach reduces the order of the robot model and significantly decreases the computational burden. Li et al. (2021) developed an image-driven control strategy for the slow dynamic part involving rigid body motion to reduce visual errors, while compensating for errors in the approximated Jacobian matrix. For the fast dynamic part related to elastic vibrations, they designed an observer to predict the fast dynamic state to avoid relying on direct vibration state measurements. Based on these predicted states, a control feedback mechanism was developed to mitigate vibrations in the flexible robot. However, this system requires extra visual sensors, which introduces additional failure points. These visual sensors may experience malfunctions or degraded performance due to lighting conditions and dust.

The trajectory planning methods for rehabilitation robots have evolved from traditional robotic arms. Additionally, literature 0 introduced a smooth algorithm for trajectory planning of spray guns to address trajectory connection issues (From et al., 2010). Literature 0 explored the trajectory planning for the robot end effector (Dhanaraj et al., 2022). While literature 0 investigated the robot kinematic model using neural network approaches (Gao, 2020). However, due to the model complexity and the lengthy computation time, its engineering applications are limited. Motion control and singular perturbation control are critical technologies in rehabilitation robot control systems, directly impacting rehabilitation training effectiveness and safety. Further optimization of these control strategies significantly enhances the performance and clinical rehabilitation robot value.

This paper explores the development of kinematic and dynamic models for lower limb rehabilitation robots, providing

a theoretical basis for trajectory planning. By applying maximum acceleration constraints to S-type trajectory planning, the study generates optimal acceleration trajectories that consider joint motion ranges, speed limits, and dynamic conditions, ensuring smooth and efficient robot operations. The integration of the LuGre friction model into the dynamic equations enables zero-force control, enhancing patient comfort during rehabilitation. Additionally, the combination of singular perturbation control and motor current-based joint torque control improves system performance, offering precise regulation and rapid response to dynamic changes. These advancements enhance the robustness and stability of the control system under external disturbances, supporting personalized rehabilitation treatments and expanding the clinical applicability of lower limb rehabilitation robots.

2 Motion control of lower limb rehabilitation robot

As shown in Figure 1, this system employs a hierarchical open architecture design Based on the human lower limb structure and its regular movement mechanisms to enhance patients' efficiency and enjoyment during rehabilitation training while allowing for personalized adjustments based on individual differences. Regarding safety, the robot legs can be equipped with flexible tactile sensor modules, which enhance the robot's interactivity and the precision of rehabilitation training. The robot also incorporates multiple safety mechanisms, including force feedback, position limits, and emergency stop buttons, ensuring rapid response in abnormal situations to protect patient safety.

2.1 Kinematic model

In Figure 1, represent the lengths of the three links, while β denotes the angle between the extension line of link 3 and link 2. Let the height of the rise at joint 1 be h , and the rotational angles at the three joints be $\theta_1, \theta_2, \theta_3$. These three rotational angles collectively determine the spatial position and orientation. By adjusting the angles, the robot can navigate to different points in three-dimensional space, providing versatile functionality for patient interaction and treatment.

The DH parameters (as shown in Table 1) are used to describe the geometric relationship between adjacent joints in robotic manipulators, consisting of four key parameters: θ (joint angle), d (link offset), a (link length), and α (link twist angle). Specifically, θ represents the rotation angle about the Z-axis, describing the rotational displacement of the current joint and is typically a variable; d denotes the translation distance along the Z-axis, indicating the linear displacement of the current joint along the Z-axis and is usually a constant; a signifies the translation distance along the X-axis, representing the linear displacement of the current joint along the X-axis and is generally a constant; and α represents the rotation angle about the X-axis, describing the twist angle between adjacent joint axes and is typically a constant. Together, these parameters define the relative position

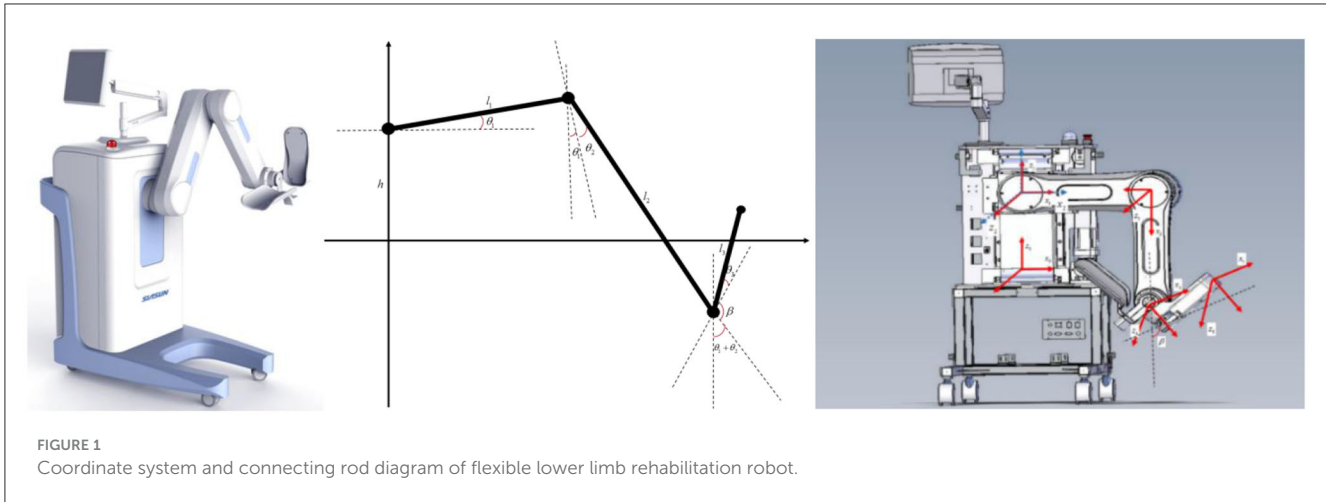


FIGURE 1
Coordinate system and connecting rod diagram of flexible lower limb rehabilitation robot.

TABLE 1 Robot D-H parameters.

rods	α	a	d	θ
1	0	0	h	0
2	$\frac{\pi}{2}$	0	0	θ_1
3	0	l_1	0	$\theta_2 + \frac{\pi}{2}$
4	0	l_2	0	$\theta_3 + \beta$
H	0	l_3	0	0

and orientation between the joints of a robotic arm, providing a foundational framework for kinematic analysis.

$${}^0_H T = {}^0_1 T \cdot {}^1_2 T \cdot {}^2_3 T \cdot {}^3_4 T \cdot {}^4_H T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$\begin{cases} n_x = -C1S2C3 + S1C2C3 + C1C2S3 + S1S2S3 \\ n_y = 0 \\ n_z = S1S2C3 - C1C2C3 - S1C2S3 - C1S2S3 \\ o_x = C1S2S3 - S1C2S3 + C1C2C3 + S1S2C3 \\ o_y = 0 \\ o_z = 0 \\ a_x = 0 \\ a_y = -1 \\ a_z = -S1S2S3 + C1C2S3 - S1C2C3 - C1S2C3 \\ p_x = l_1C1 + l_2(-C1S2 + S1C2) + l_3(-C1S2C3 + S1C2C3 + C1C2S3 + S1S2S3) \\ p_y = 0 \\ p_z = h - l_1S1 + l_2(S1S2 - C1C2) + l_3(S1S2C3 - C1C2C3 - S1C2S3 - C1S2S3) \end{cases}$$

The known variables are:

$$\begin{cases} C1 = \cos \theta_1 \\ S1 = \sin \theta_1 \\ C2 = \cos \theta_2 \end{cases} \begin{cases} S2 = \sin \theta_2 \\ C3 = \cos (\theta_3 + \beta) \\ S3 = \sin (\theta_3 + \beta) \end{cases} \quad (3)$$

The pose transformation matrix (Equation 1) can be obtained according to DH table, and the forward kinematics equation can be derived. In this matrix, the first three columns correspond to the projections of the new coordinate system's X -axis, Y -axis, and Z -axis direction vectors within the original coordinate system, represented as $\mathbf{n} = [n_x, n_y, n_z]^T$, $\mathbf{o} = [o_x, o_y, o_z]^T$, respectively. These vectors collectively define the rotational (orientational) transformation of the new coordinate system relative to the original one. The fourth column $\mathbf{p} = [p_x, p_y, p_z]^T$ denotes the position vector of the new coordinate system's origin with respect to the original coordinate system, encapsulating the translational transformation. The final row $[0, 0, 0, 1]$ serves as the normalization component of the homogeneous coordinates. Such transformation matrices are extensively utilized in robotic kinematics, where sequential multiplication facilitates the computation of the end-effector's pose relative to the base coordinate system.

2.2 Zero-force control

Set on the joint coordinates $q = [h \ q_1 \ q_2 \ q_3]^T = [h \ \theta_1 \ \theta_1 + \theta_2 \ \theta_1 + \theta_2 + \theta_3]^T$. m_1 , m_2 and m_3 respectively, represent the mass of the three members, l_1 , l_2 and l_3 represent the length of the three members, d_1 , d_2 , d_3 indicate the distance from the mass center to the axes of the three members, respectively.

(1) Rehabilitation of robot potential energy

(2) Potential energy P_1 of rehabilitation robot rod 1

$$P_1 = m_1 g (h + d_1 \cos q_1) \quad (4)$$

The potential energy P_2 of rehabilitation robot rod 2

$$P_2 = m_2 g (h + l_1 \cos q_1 - d_2 \cos q_2) \quad (5)$$

The potential energy P_3 of rehabilitation robot rod 3

$$P = m_3 g (h + l_1 \sin q_1 - l_2 \sin q_2 + d_3 \cos (q_3 + \beta)) \quad (6)$$

The rehabilitation robot total potential energy as Equation 7.

$$P = P_1 + P_2 + P_3 \quad (7)$$

(2) The rehabilitation robot total kinetic energy of can be similarly obtained as follows

$$K = K_1 + K_2 + K_3 \quad (8)$$

The Lagrange function L is:

$$L = K - P = (K_1 + K_2 + K_3) - (P_1 + P_2 + P_3) \quad (9)$$

The known variables are:

$$\begin{cases} C1 = \cos q_1 \\ S1 = \sin q_1 \\ C2 = \cos q_2 \end{cases} \begin{cases} S2 = \sin q_2 \\ C3 = \cos(q_3 + \beta) \\ S3 = \sin(q_3 + \beta) \end{cases} \quad (10)$$

The driving torque for each joint of the rehabilitation robot is then given by:

$$\begin{aligned} \tau_1 = & \frac{d}{dt} \frac{\partial L}{\partial \dot{h}} - \frac{\partial L}{\partial h} = (m_1 + m_2 + m_3) \ddot{h} + (m_1 d_1 + m_2 l_1 \\ & + m_3 l_1) C1 \ddot{q}_1 + \\ & (-m_2 d_2 + m_3 l_2) S2 \ddot{q}_2 - m_3 d_3 S3 \ddot{q}_3 - (m_1 d_1 + m_2 l_1 \\ & + m_3 l_1) S1 \dot{q}_1^2 + \\ & (-m_2 d_2 + m_3 l_2) C2 \dot{q}_2^2 - m_3 d_3 C3 \dot{q}_3^2 - (m_1 + m_2 + m_3) g \end{aligned} \quad (11)$$

Similarly, we can get other joint driving torques. Take the friction term into the traditional dynamic equation and use the LuGre friction model for friction analysis and realize zero-force control. This study utilized a flexible lower-limb rehabilitation robotic experimental platform to approach enabled zero-force and high-precision control of the end effector force in the robot, achieving effective force feedback management.

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + T_f(\dot{q}) = \tau \quad (12)$$

T_f is torque of LuGre friction, \dot{q} is angular velocity vector, \ddot{q} is rotational angular acceleration vector.

This paper uses the LuGre friction model to compensate for joint dynamics. LuGre friction method not only enhances the fit of friction characteristics simulation but also improves the overall accuracy of the collision detection model. The LuGre friction model is

$$T_f(\dot{q}) = \left\{ \left[f_c + (f_s - f_c) \exp\left(-\left(\frac{\dot{q}}{q_s}\right)^2\right) \right] \right\} \text{sgn}(\dot{q}) + \sigma_2 \dot{q} \quad (13)$$

In order to obtain the real friction parameters, experiments are carried out on the vacuum manipulator platform to obtain the following data: among others, f_c is Coulomb friction, f_s is static friction; $\dot{q}(t)$ provides real-time speed for the joints; \dot{q}_s for Stribeck Speed δ is a parameter related to the shape of the contact surface; σ_2 is the coefficient of viscous friction; $\tau(t)$ is the joint drive torque measured by a sensor. The

sampling frequency is high enough to capture changes in friction over time.

In rehabilitation robotics, the LuGre friction model provides critical support for joint friction compensation. When a patient applies a minimal external force, the robot, based on real-time joint velocity measurements and identified friction parameters (such as static friction), generates an equivalent compensating torque through the controller to counteract the inherent friction resistance of the mechanical system. This process enables the end-effector to exhibit a “weightless” zero-force control effect for the patient, meaning that only a minimal force is required to drive the robot’s motion. This significantly enhances the compliance of rehabilitation training and the safety of human-robot interaction. By experimentally calibrating parameters (e.g., low-speed sweep tests for static friction and high-speed constant velocity tests for viscous friction separation), the LuGre model further optimizes friction compensation accuracy, ensuring the robustness of zero-force control in complex motion scenarios.

2.3 Optimal S-type planning

Under optimal trajectory control, the lower limb rehabilitation robot significantly improves its trajectory planning ability and operational flexibility and maximizes the advantages of the lower limb rehabilitation robot. The trajectory interpolated core considers both kinematic constraints (such as joint angles and end-effector positions) and dynamic constraints (including forces, torques, and inertia). Optimal trajectory parameters that satisfy these constraints are computed using optimization algorithms.

A complete S-curve trajectory planning can be constructed based on Equations 14, 15.

$J(t)$ is Jerk, $A(t)$ is Acceleration and $V(t)$ is Velocity:

$$\begin{aligned} J(t) = & \begin{cases} J0 \leq t < t_1 \\ 0 t_1 \leq t < t_2 \\ -J t_2 \leq t < t_3 \\ 0 t_3 \leq t < t_4 \\ -J t_4 \leq t < t_5 \\ 0 t_5 \leq t < t_6 \\ J t_6 \leq t < t_7 \end{cases} \\ A(t) = & \begin{cases} J\tau_1 0 \leq t < t_1 \\ J T_1 t_1 \leq t < t_2 \\ J T_1 - J\tau_3 t_2 \leq t < t_3 \\ 0 t_3 \leq t < t_4 \\ -J\tau_5 t_4 \leq t < t_5 \\ -J T_5 t_5 \leq t < t_6 \\ -J T_5 + J\tau_7 t_6 \leq t < t_7 \end{cases} \\ V(t) = & \begin{cases} V_s + \frac{1}{2} J\tau_1^2 0 \leq t < t_1 \\ V_{01} + J T_1 \tau_2 t_1 \leq t < t_2 \\ V_{02} + J T_1 \tau_3 - \frac{1}{2} J\tau_3^2 t_2 \leq t < t_3 \\ V_{03} t_3 \leq t < t_4 \\ V_{04} - \frac{1}{2} J\tau_5^2 t_4 \leq t < t_5 \\ V_{05} - J T_5 \tau_6 t_5 \leq t < t_6 \\ V_{06} - J T_5 \tau_7 - \frac{1}{2} J\tau_7^2 t_6 \leq t < t_7 \end{cases} \end{aligned} \quad (14)$$

Position is $S(t)$:

$$S(t) = \begin{cases} V_s \tau_1 + \frac{1}{6} J \tau_1^3 & 0 \leq t < t_1 \\ S_{01} + V_{01} \tau_2 + \frac{1}{2} J T_1 \tau_2^2 & t_1 \leq t < t_2 \\ S_{02} + V_{02} \tau_3 + \frac{1}{2} J T_1 \tau_3^2 - \frac{1}{6} J \tau_3^3 & t_2 \leq t < t_3 \\ S_{03} + V_{03} \tau_4 & t_3 \leq t < t_4 \\ S_{04} + V_{04} \tau_5 - \frac{1}{6} J \tau_5^3 & t_4 \leq t < t_5 \\ S_{05} + V_{05} \tau_6 - \frac{1}{2} J T_5 \tau_6^2 & t_5 \leq t < t_6 \\ S_{06} + V_{06} \tau_7 - \frac{1}{2} J T_5 \tau_7^2 + \frac{1}{6} J \tau_7^3 & t_6 \leq t < t_7 \end{cases} \quad (15)$$

An example of a planning segment from time node t_0 to t_7 is considered, where t_0 represents the initial moment and t_7 indicates the final moment. At both the start and end points, the robot velocity and acceleration are set to zero to ensure a smooth initiation and cessation of motion. Furthermore, it is essential to account for the limits of robot speed, acceleration, and jerk (the accelerated change rate) during the movement process. These include the max velocity v_m , the max acceleration a_{mu} , the max deceleration a_{md} , as well as the max jerk during the acceleration and deceleration phases, denoted as j_{mu}, j_{md} . These parameters are directly related to robot safety and efficiency.

The states of trajectory planning at each discrete point $(q_i, \dot{q}_i, \ddot{q}_i)$ are obtained using the robot S-type programming model. Based on the established dynamic model (Equation 16), the inverse dynamic IDy is used to solve the theoretical torque $\tau_{i,cal}$ of each joint demand.

$$\tau_{i,cal} = IDy(q_i, \dot{q}_i, \ddot{q}_i) \quad (16)$$

The calculated moment value is compared with each manipulator joint's maximum output moment $\tau_{i,max}$. If the calculated moment value exceeds the theoretical moment value, it indicates that the joint cannot provide the moment value and must be corrected to make the planning result conform to the moment limit. The corrected joint calculated moment is as follows Equation 17.

$$\tau_{i,calmax} = \text{sign}(\tau_{i,cal}) \cdot \min(\tau_{i,max}, |\tau_{i,cal}|) \quad (17)$$

The optimal trajectory control can be achieved only when the robot dynamics constraint is applied to the trajectory planning of the manipulator's end. According to the dynamic equation of the manipulator, the corresponding relationship between the joint torque and acceleration \ddot{q}_{newmax} is obtained as follows Equation 18.

$$\ddot{q}_{newmax} = M(q) \left(\tau_{i,calmax} - \dot{q}^T C(q) \dot{q} - g(q) \right) \quad (18)$$

On this basis, the torque constraint is mapped to the acceleration constraint, the acceleration subject to the torque constraint is obtained, and the final acceleration constraint is obtained by comparing it with the joint acceleration in the initial state. The process is as follows Equation 19. $\ddot{q}_{i,max}$ is the maximum acceleration constraint in the initial state, $\ddot{q}_{i,modified}$

is the maximum acceleration of the i joint calculated by torque constraint.

$$\ddot{q}_{i,newmax} = \min(\ddot{q}_{i,max}, |\ddot{q}_{i,modified}|) \quad (19)$$

$\ddot{q}_{i,newmax}$ is the maximum acceleration constraint adjusted by the dynamic constraint, and \ddot{q} is the vector formed by the maximum joint acceleration calculated by the torque constraint. If the new acceleration constraint is substituted into the dynamic trajectory planning, the complete trajectory planning satisfies the dynamic constraint.

3 Singular perturbation method of lower limb rehabilitation robot

In the widespread robot application, a particular type of system is often encountered, characterized by significant differences in the rates of change between various states, exhibiting unique singularities and separated phenomena in motion (Weingartshofer et al., 2023; Amersdorfer and Meurer, 2022; Jiang et al., 2023; Wang et al., 2021; Pana et al., 2023; Cao et al., 2023; Zhang et al., 2024). Such systems, including electric motors, generators, precision robotic structures, and complex biological systems, are collectively called singularly perturbed systems or systems with dual time scales. For these dual time-scale dynamic systems, also known as singular perturbation systems, the design of control strategies must consider their multi-time-scale characteristics. Traditional control methods based on a single time scale are often inadequate, as the dynamic behaviors of such systems differ significantly across various time scales. To achieve precise control of these systems, it is common to utilize singular perturbation theory to decompose the original system into multiple subsystems that are temporally separated yet mutually coupled, with each subsystem corresponding to a specific time scale.

3.1 Singular perturbation model of robotics

This chapter focuses on transforming dynamic models into singular perturbation forms. Based on the inherent dynamic characteristics, mathematical transformation methods are applied to derive robot system models that comply with the requirements of singular perturbation theory analysis. During this derivation process, friction factors are temporarily excluded from consideration to simplify the analysis and highlight the core dynamic characteristics. The aim is to capture the essential dynamic behavior; it provides a theoretical framework for the design of subsequent controllers based on singular perturbation theory. The singular perturbation model is categorized into separate fast and slow subsystems. Typically, the fast subsystem reflects the high-frequency dynamic characteristics, while the slow subsystem characterizes the long-term behavior and primary motion trends.

By temporarily neglecting the influence of friction, a simplified dynamic model of the robotic joint torque system can be

expressed as follows Equation 20, N is the accelerator transmission ratio, J_{em} is motor rotor inertia, is the electromotor torque formula, which from the AC permanent magnet synchronous motor torque model. τ is joint torque, τ_{ext} is the external reaction torque.

$$\begin{cases} M(q)\ddot{q} + C(\dot{q}, \dot{q})\dot{q} + G(q) = \tau + \tau_{ext} \\ NJ_{em}\ddot{q} + \tau = N\tau_{em} \\ \tau_{em} = \frac{3}{2}n_p\psi_f i_{sq} \end{cases} \quad (20)$$

It can be observed as follows Equation 21.

$$\tau = \frac{3}{2}Nn_p\psi_f i_{sq} - NJ_{em}\frac{d\omega}{dt} \quad (21)$$

In the analysis of robotic dynamics, the electromotor torque τ_{em} serves as the driving source, and its output is transmitted to the robot joints through the transmission system, resulting in the joint torque τ . When the robot design includes joint torque sensors, this critical parameter can be directly measured. The expression for joint torque is given by Equation 22, where K represents the inherent stiffness of the joint, $N^{-1}\theta$ is the equivalent joint angular displacement obtained by inversely transforming the motor angle θ through the gear ratio N , and q is the actual joint angle. This Equation illustrates the dynamic relationship between joint torque, motor angle, and joint angle, as well as the impact of stiffness K on system response. The motor angular acceleration is obtained from Equation 23.

$$\tau = K(N^{-1}\theta - q) \quad (22)$$

$$\ddot{\theta} = NK^{-1}\ddot{\tau} + N\ddot{q} \quad (23)$$

Substituting Equations 22, 23 into the second equation in Equation 20, we get Equation 24.

$$N^2J_mK^{-1}\ddot{\tau} + \tau = N\tau_m - N^2J_m\ddot{q} \quad (24)$$

Since the inertia matrix is a symmetric positive definite matrix, the following equation can be obtained.

$$\ddot{q} = M(q)^{-1}(\tau + \tau_{ext} - C(q, \dot{q})\dot{q} - G(q)) \quad (25)$$

Substituting Equation 25 into Equation 24 gives Equation 26.

$$K^{-1}\ddot{\tau} + \left(J_m^{-1}N^{-2} + M(q)^{-1}\right)\tau = J_m^{-1}N^{-1}\tau_m + M(q)^{-1}(C(q, \dot{q})\dot{q} + G(q) - \tau_{ext}) \quad (26)$$

By introducing a small parameter ε , the system equations can be reformulated as a standard singular perturbation model. The joint stiffness K is often very high, so $\frac{K_\varepsilon}{\varepsilon^2}$ can be used instead of K . K_ε is positive definite diagonal matrix. Substituting the expression of K and sorting

it out, we can get the singular perturbation model of the system:

$$\begin{cases} M(q)\dot{q} + C(\dot{q}, \dot{q})\dot{q} + G(q) = \tau + \tau_{ext} \\ \varepsilon^2\ddot{\tau} + K_\varepsilon\left(J_m^{-1}N^{-2} + M(q)^{-1}\right)\tau = K_\varepsilon J_m^{-1}N^{-1}\tau_{em} \\ + K_\varepsilon M(q)^{-1}(C(q, \dot{q})\dot{q} + G(q) - \tau_{ext}) \end{cases} \quad (27)$$

The system boundary layer model is derived by introducing a new coordinate variable as follows Equation 28.

$$\begin{cases} h(q, \dot{q}, t) = (J_m^{-1}N^{-2} + M^{-1})^{-1}(J_m^{-1}N^{-1}\bar{\tau}_{em} \\ + M^{-1}(C\dot{q} + G - \bar{\tau}_{ext})) \\ \gamma = \tau - h(q, \dot{q}, t), \gamma \in \mathbb{R}^n \end{cases} \quad (28)$$

Introduce a new time scale as follows Equation 29.

$$\begin{cases} v = (t - t_0)/\varepsilon \\ \text{if } \varepsilon = 0 \end{cases} \quad (29)$$

Equation 30 is derived from Equations 28, 29.

$$\begin{aligned} \frac{d^2\gamma}{dv^2} + K_s(J_m^{-1}N^{-2} + M^{-1})(\gamma + h) \\ = K_sJ_m^{-1}N^{-1}\tau_{em} + K_sM^{-1}(C\dot{q} + G - \tau_{ext}) \end{aligned} \quad (30)$$

At the time scale v , the state variables and the external inputs τ_{ext} are regarded as static values. Equation 31 can be derived.

$$\begin{aligned} (J_m^{-1}N^{-2} + M^{-1})h = J_m^{-1}N^{-1}\bar{\tau}_{em} \\ + M^{-1}(C\dot{q} + G - \bar{\tau}_{ext}) \end{aligned} \quad (31)$$

By substituting it into Equation 30, the final system boundary layer model is as follows Equation 32.

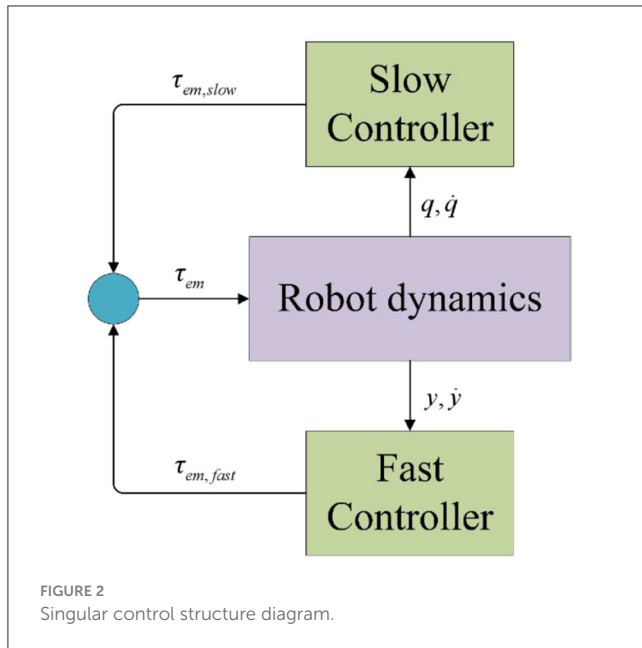
$$\frac{d^2\gamma}{dv^2} + K_s(J_m^{-1}N^{-2} + M^{-1})\gamma = K_sJ_m^{-1}N^{-1}(\tau_{em} - \bar{\tau}_{em}) \quad (32)$$

3.2 Controller design based on singular perturbation

In robotic controller design, singular perturbation methods are often integrated into composite feedback strategies to optimize system performance, as shown in Figure 2. This decomposition strategy independently addresses the different time scale issues in the system dynamics, thereby enhancing system stability, improving response speed, and simplifying the controller complexity (Bhardwaj et al., 2021; Khan et al., 2022; Zheng et al., 2022; Iskandar et al., 2022).

This design decomposes the control input into slow and fast system components, as shown in Equation 33.

$$\tau_{em} = \tau_{em,slow} + \tau_{em,fast} \quad (33)$$



The slow subsystem represents the long-term behavior and primary motion trends of the robot. It is characterized by slower dynamics, such as the robot's overall trajectory and position control. The slow subsystem employs classical control strategies, such as computed torque control and torque feedforward control, which are commonly used in rigid robot models. These strategies ensure that the robot's motion is smooth and accurate over time.

For the selection of the control input $\tau_{em,slow}$ for the slow subsystem, it must be based on the assumption that the system has a standard form, meaning that the average time scale $\bar{\tau}$ in the quasi-steady-state model exists and has a unique solution. The key point is that the determination of $\tau_{em,slow}$ should depend solely on the state variables of the quasi-steady-state model (q, \dot{q}), avoiding the direct introduction of instantaneous feedback from the joint torque τ , to ensure the independence and stability of the control input.

The fast subsystem captures high-frequency dynamics, such as vibrations and rapid adjustments in the robot's joints. These dynamics are critical for ensuring stability during rapid movements or when external disturbances are present.

In designing the control input $\tau_{em,fast}$ for the fast subsystem, it is essential to strictly adhere to the principle that it has no impact on the quasi-steady-state model when boundary layer effects are significant ($\varepsilon = 0$). This requires that the norm tends to zero under the limit condition ($\tau_{em,fast}|_{\varepsilon \rightarrow 0} = 0$). This design criterion aims to isolate potential disturbances from the fast subsystem on the dynamic behavior of the slow subsystem, ensuring the overall stability and accuracy of the system during rapid dynamic adjustments while optimizing performance in the boundary layer region. And we get the Equation 34.

$$\begin{cases} \tau_{em,slow} = \tau_{em}|_{\varepsilon \rightarrow 0} = \bar{\tau} \\ \tau_{em,fast} = \tau_{em} - \bar{\tau} = -\varepsilon D_{\tau} \dot{\tau} \end{cases} \quad (34)$$

It becomes clear that this structure aligns with that of a rigid robot model. This alignment allows for the direct application of

classical control strategies used for rigid manipulators, such as the computed torque method and torque feedforward control, for design slow subsystem.

The damping matrix D_{τ} is designed as a positive definite matrix to ensure the stability and convergence of the system. A positive definite damping matrix introduces an energy dissipation mechanism into the system, which helps to suppress oscillations and stabilize the system. This is particularly crucial for rehabilitation robots, as smooth and stable motion is vital for patient safety and comfort.

For the slow subsystem, the controller design follows the fundamental principles (Equation 35) of the rigid manipulator model. The variables q, \dot{q} represent the joint position and velocity. Considering the specificity of the boundary layer system, where time t and state variables can be treated as constants within a specific analytical framework, the original term $-\varepsilon D_{\tau} \dot{\tau}$ can be reasonably transformed into $-\varepsilon D_{\tau} \dot{\tau}$. This transformation deepens the understanding of the system dynamic behavior under boundary conditions.

$$\tau_{em,slow} = \tau_{rigid}(q, \dot{q}) \quad (35)$$

The design of the overall composite feedback controller for the robotic system integrates the control strategies of the aforementioned fast and slow subsystems. By adjusting the positive definite damping matrix, the control laws of the rigid manipulator model, and the dynamic adjustments under boundary layer effects, a robust and efficient control system framework is ultimately formed. This framework is capable of addressing complex and variable operational environments and task requirements.

$$\tau_{em} = \tau_{rigid}(q, \dot{q}) - \varepsilon D_{\tau} \dot{\tau} \quad (36)$$

It utilizes an increased damping term to act on the system. However, it does not address or optimize the dynamic characteristics of the boundary layer, which limits the system's overall performance improvement. To overcome this shortcoming, we introduce a torque deviation feedback mechanism. By monitoring and feeding back torque deviations in real-time, we can dynamically adjust the control strategy, enhancing system stability and significantly improving the boundary layer's dynamic response and adjustment capability, leading to more efficient system control.

The torque deviation feedback is Equation 37.

$$K_{\tau} \gamma = K_{\tau}(\tau - h(q, \dot{q}, t)) \quad (37)$$

K_{τ} is gain matrix, which used to improve the boundary layer dynamic performance. Therefore, the entire controller can be expressed as follows:

$$\begin{aligned} \tau_{em} = \tau_{rigid}(q, \dot{q}) - K_{\tau} \gamma - \varepsilon D_{\tau} \dot{\tau} = \tau_{rigid}(q, \dot{q}) \\ - K_{\tau}(\tau - h(q, \dot{q}, t)) - \varepsilon D_{\tau} \dot{\tau} \end{aligned} \quad (38)$$

3.3 Improved controller design

To implement an effective feedback mechanism, it is often necessary to accurately solve the steady-state function $h(q, \dot{q}, t)$. However, this can be quite challenging in practical applications, such as the difficulties in accurately measuring external torque τ_{ext} and the complex computation of the inverse inertia matrix $M(q)$. The Tychonov theorem states that for trajectory tracking errors, it ensures that the deviation from the quasi-steady state is bounded by a high-order infinitesimal quantity ε . This characteristic becomes significant only as ε approaches zero, highlighting the remarkable advantage of the boundary layer model over the quasi-steady state model in terms of dynamic response speed. By finely tuning the parameter ε , rapid convergence of the error to a range close to the quasi-steady state can be achieved. Its introduction enables us to decompose complex multi-timescale systems into two independent subsystems—the fast subsystem and the slow subsystem—thereby simplifying controller design. The fast subsystem is typically employed to address high-frequency dynamics (e.g., vibrations, rapid adjustments), while the slow subsystem is utilized to manage low-frequency dynamics (e.g., overall trajectory control, position control).

To lessen the complexity of the state function, the controller is broken down into the following structure.

$$\tau_{em} = \tau_{ss}(q, \dot{q}) - K_\tau \tau - \varepsilon D_\tau \dot{\tau} \quad (39)$$

The positive definite control gain matrix K_τ, D_τ are crucial, as they ensure the system stability and responsiveness. The real-time control input $\tau_{ss}(q, \dot{q})$ in the quasi-steady state model is dynamically adjusted based on the system state to achieve precise control. By replacing $K_\tau \tau$ in the original Equation 39 with $K_\tau y$, it directly affects the dual time scale control strategy. On one hand, it directly acts on the fast system component, optimizing the computation of $\tau_{m, \text{fast}}$ to enhance the system rapid response capability and dynamic performance. On the other hand, this strategy also indirectly affects the control input of the slow system component $\tau_{m, \text{slow}}$ by optimizing the long-term control strategy, thereby improving the system robustness and steady-state accuracy. During the design process, careful adjustments to K_τ, D_τ are necessary, taking into account the interactions between the fast and slow systems to ensure that the overall control system can respond quickly to external changes while maintaining long-term stable operation.

$$\tau_{ss}(\bar{q}, \bar{\dot{q}}) = N^{-1}(I + K_\tau)\tau_d, \tau_d \quad (40)$$

$$\begin{aligned} (M(\bar{q}) + (I + K_\tau)^{-1}N^2J_{em})\bar{\ddot{q}} + C(\bar{q}, \bar{\dot{q}})\bar{\dot{q}} + G(\bar{q}) \\ = (I + K_\tau)^{-1}N\tau_{ss}(\bar{q}, \bar{\dot{q}}) \end{aligned} \quad (41)$$

τ_d as a new control input, it can be designed based on rigid robots to derive the steady-state system as follows Equation 41. Finally, we get the Equation 42 and Closed-loop control system Equation 43.

$$\tau_{em} = \tau_d - K_\tau (\tau - \tau_d) - \varepsilon D_\tau \dot{\tau} \quad (42)$$

$$\begin{aligned} \frac{d^2\gamma}{dv^2} + K_\varepsilon J_{em}^{-1}N^{-1}D_\tau \frac{d\gamma}{dv} + K_\varepsilon J_{em}^{-1} \\ (N^{-2} + J_{em}M(q)^{-1} + N^{-1}K_\tau)\gamma = 0 \end{aligned} \quad (43)$$

For the newly constructed quasi-steady state system, the design of the inertia matrix K_τ is closely related to the control gain matrix. We can dynamically configure the inertia distribution coefficient of the motor rotor by adjusting this parameter. This not only enhances the system resistance to external disturbances but also optimizes the response speed and stability of the control loop.

For the modified boundary layer system, the control input for the fast subsystem part is in the form of Equation 44.

$$\tau_{em, \text{fast}} = -K_\tau(\bar{\tau} - \bar{\tau}) - \varepsilon D_\tau \dot{\tau} = -K_\tau \gamma - \varepsilon D_\tau \dot{\gamma} \quad (44)$$

$$\begin{aligned} \frac{d^2\gamma}{dv^2} + K_\varepsilon J_{em}^{-1}N^{-1}D_\tau \frac{d\gamma}{dv} + K_\varepsilon J_{em}^{-1} \\ (N^{-2} + J_{em}M(q)^{-1} + N^{-1}K_\tau)\gamma = 0 \end{aligned} \quad (45)$$

4 Experiment

4.1 Kinematics and trajectory planning

The kinematic experiment uses cylindrical helices and conical helices to verify the kinematic trajectories. The parametric equation for a cylindrical helix can be expressed as Equation 46.

$$\begin{cases} x(t) = R \cdot \cos(t) \\ y(t) = R \cdot \sin(t) \\ z(t) = c \cdot t \end{cases} \quad (46)$$

Here, t is the parameter, R is the radius of the helix, and c is the pitch constant, which represents the distance the helix moves in the z direction for each complete turn.

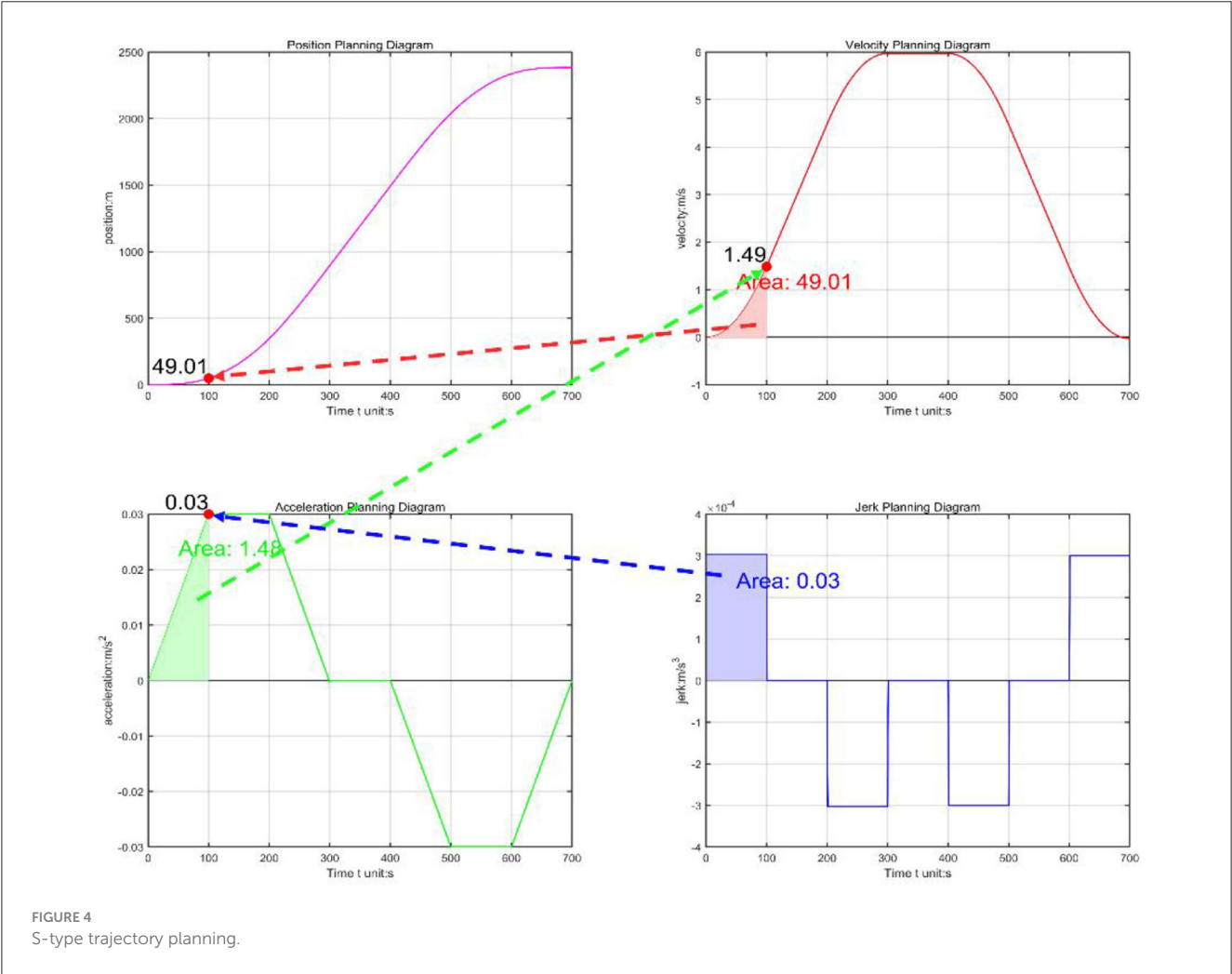
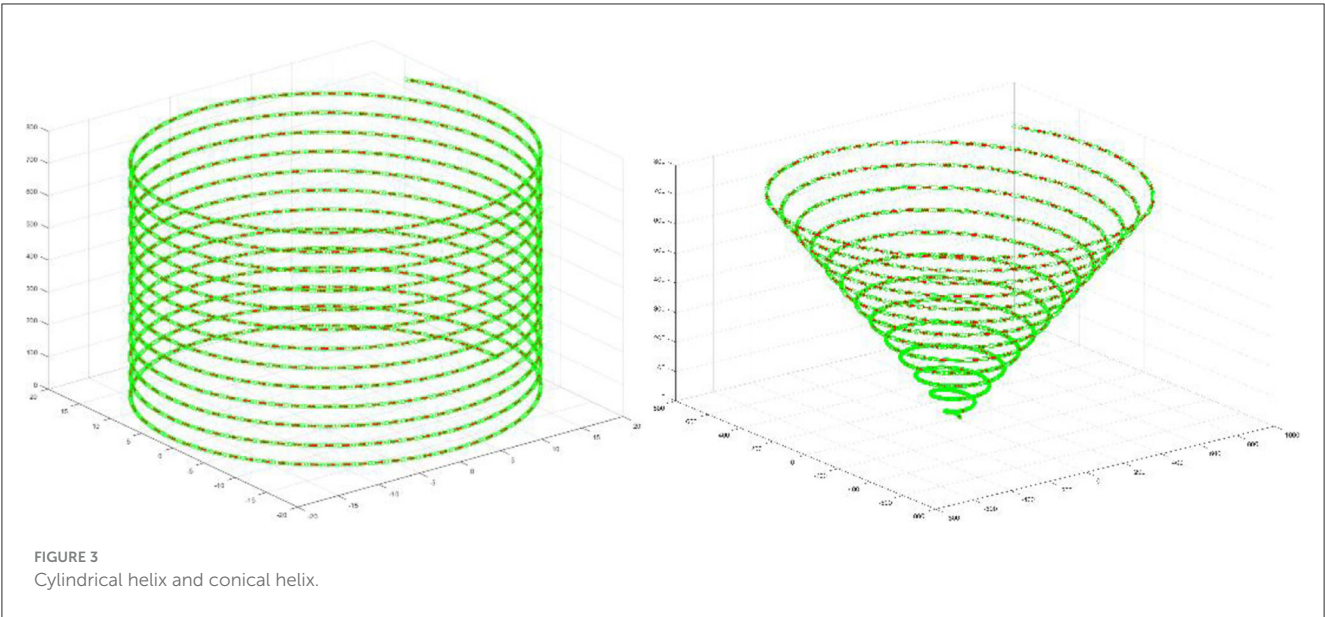
The parametric equation for a conical helix can be expressed as Equation 47.

$$\begin{cases} x(t) = (R_0 + kt) \cdot \cos(t) \\ y(t) = (R_0 + kt) \cdot \sin(t) \\ z(t) = c \cdot t \end{cases} \quad (47)$$

Here, R_0 is the initial radius of the helix, k is the phasent rate of change and c is the pitch constant.

As shown in Figure 3, by comparing the original end effector trajectory with the trajectory obtained from the forward kinematics calculations, the results indicate a high degree of consistency between the two. This not only verifies the correctness of the inverse and forward solution algorithms but also further demonstrates the accuracy and reliability of the robot kinematic model in design and practical applications.

Figure 4 clearly illustrates the trajectory constraints of S-curve trajectory planning within a specified time frame and how to accurately generate trajectories that comply with preset acceleration limits, thereby validating the correctness



of this planning method. The main advantage of trajectory planning in joint space is the direct control of the robotic joint movements. Compared to trajectory planning in operational space, this approach significantly enhances operational flexibility and accuracy. Furthermore, this method effectively avoids potential issues arising from motion singularities and

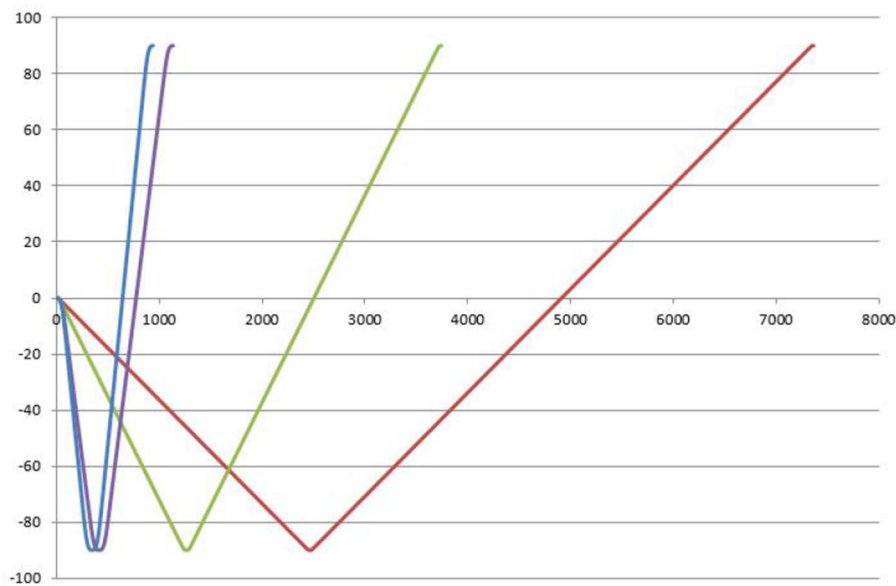


FIGURE 5
Optimal trajectory control end position time.

operational complexities, thereby enhancing the robustness of the system.

In this paper, the torque value obtained by the complete dynamic model is converted into the acceleration value, and the acceleration value is further applied as a limiting factor in trajectory planning to obtain the optimal trajectory control of the lower limb rehabilitation robot. As shown in Figure 5, the horizontal coordinate is the time, and the vertical coordinate is the TCP end position of the vacuum manipulator. Red is the general planning curve, and blue is the optimal trajectory curve. When the end position is from 0 to -80 and then to 90 , the robot's time is obtained from the acceleration calculated by the dynamics, and the planning time becomes significantly faster.

The acceleration calculated by the dynamics is applied to the S-type trajectory planning, and the optimal trajectory control of the acceleration of the S-type trajectory is performed indirectly according to the acceleration solved by the dynamics to achieve the corresponding optimal. Comparing the running time with the ordinary planning time, optimal S-type planning performs quickly. Optimal S-type planning typically requires that servo drives use position control, which enhances the speed and current response of the motor, thereby improving the effectiveness of position control.

4.2 Zero-force control based on friction model

Zero force control is a compliant control strategy that aims to ensure that the force between the system and the environment remains at zero level. The LuGre friction parameters are identified and optimized, and the theoretical driving torque is solved against the collected actual torque, as shown in Figure 6. The blue curve represents the real friction moment curve, while the red curve is the moment curve of the LuGre friction model proposed in this paper.

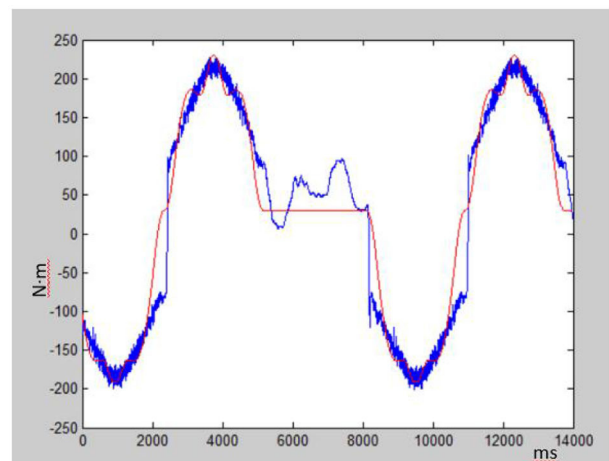


FIGURE 6
Comparison of actual torque and theoretical torque on zero-load control.

The deviation of actual friction and LuGre friction is calculated to be about 5 N.m . The results verify that the whole dynamic algorithm with The LuGre friction can effectively solve zero-force control.

To intuitively and scientifically validate the effectiveness of the dynamic algorithm, this study utilized a flexible lower-limb rehabilitation robotic experimental platform to demonstrate a zero-load control algorithm. As shown in Figure 7, this approach enabled high-precision control of the end effector force in the robot, achieving effective force feedback management.

Implementing the zero-load control algorithm on the experimental platform signifies that the robot can more accurately simulate the natural movement patterns of the human body while performing rehabilitation training tasks. This reduces

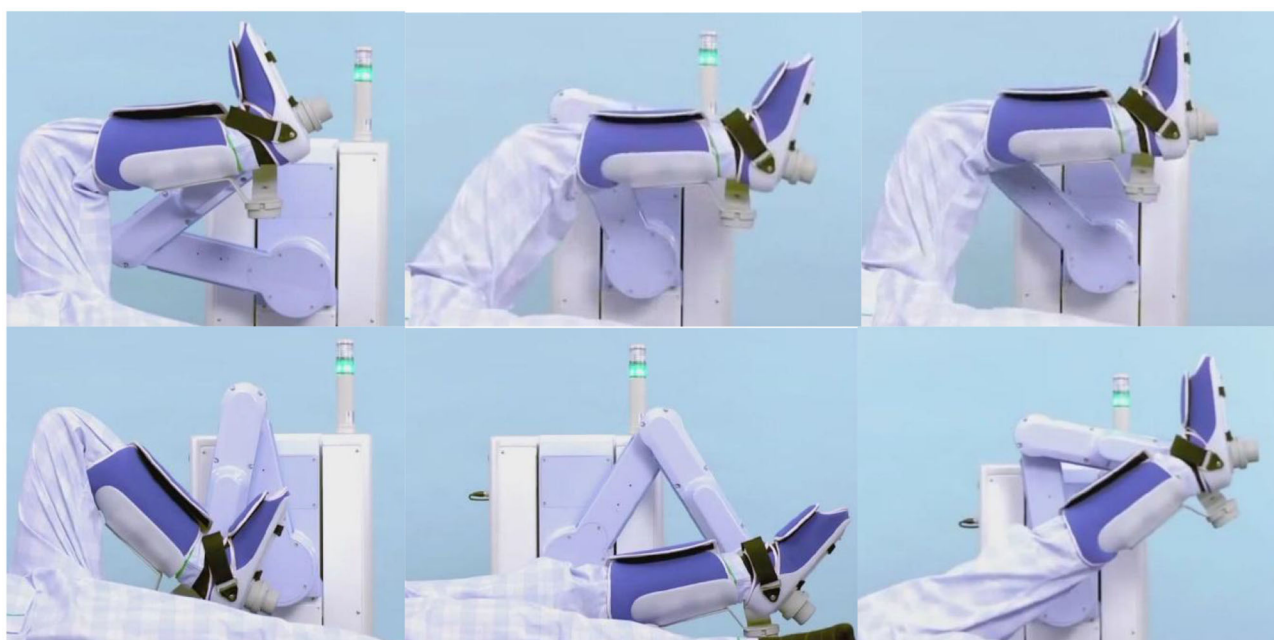


FIGURE 7
Zero-load control of flexible lower limb rehabilitation robot.

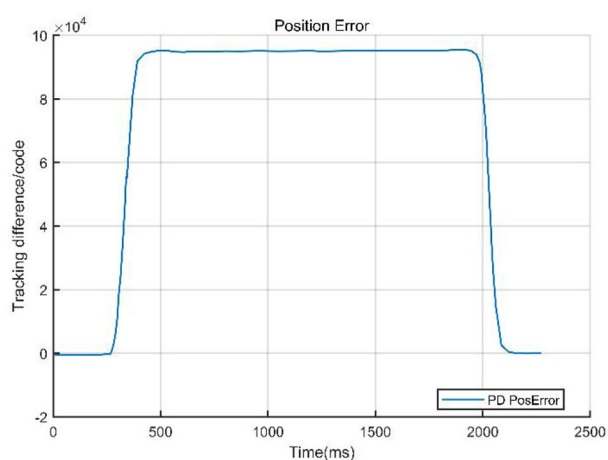


FIGURE 8
Tracking error of PD control.

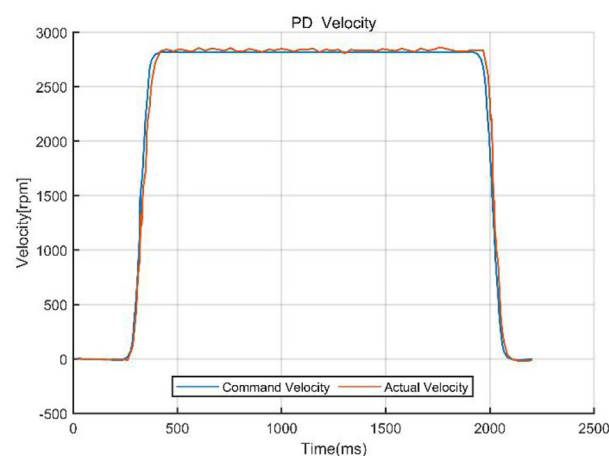


FIGURE 9
Command velocity vs. actual velocity with PD control.

unnecessary mechanical resistance and discomfort, enhancing patient rehabilitation experience and therapeutic outcomes. Such advancements have profound implications for improving rehabilitation efficiency, shortening recovery periods, and reducing overall rehabilitation costs.

4.3 Singular perturbation

To validate the effectiveness of the designed control method, we compared the PD control strategy with a PD-based composite

control strategy that integrates singular perturbation theory. The core of the experiment involved quantitatively analyzing the system performance under both control architectures, focusing on key metrics such as system tracking error, velocity response curves, and velocity error. Figures 8, 9 visually demonstrate the system dynamic behavior using classical PD control, providing a benchmark for subsequent comparison with PD control combined with singular perturbation theory. In figures, the Tracking difference on the Y-axis represents the actual code drive value minus the planned code drive value, and its units can be expressed in PPR (Pulse Per Revolution).

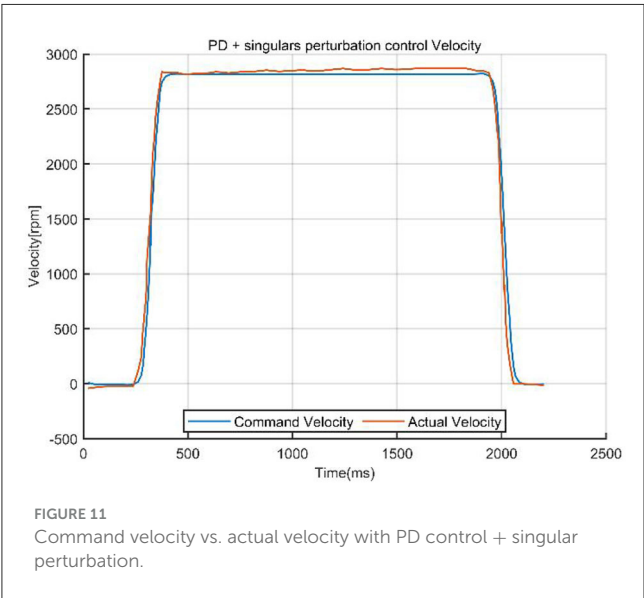
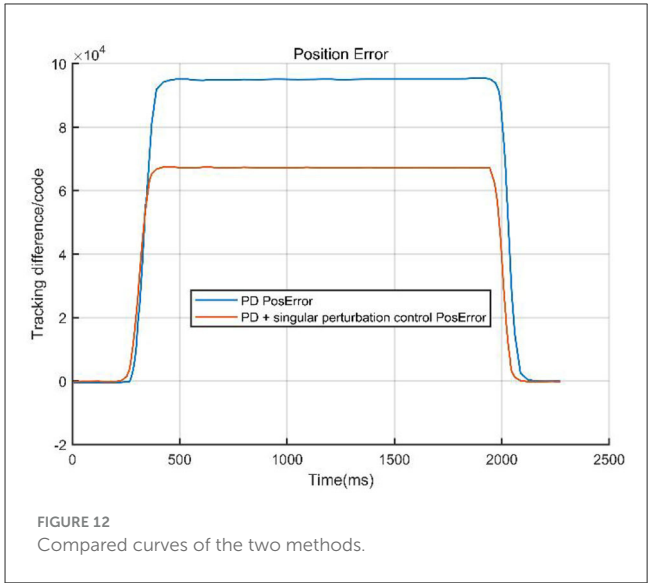
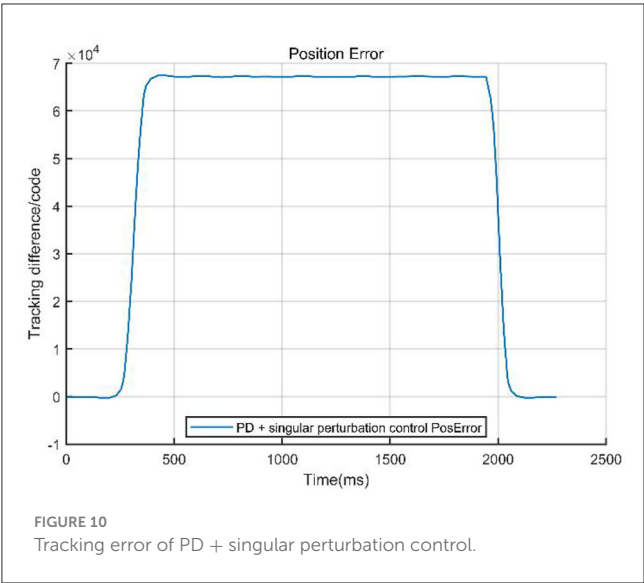


TABLE 2 Compared methods.

Method	Advantages	Limitations
PD	1. Simple structure	1. Limited effectiveness in multi-time-scale systems
	2. Low computational cost	2. Difficulty in handling high-frequency dynamics
	3. Performs well in rigid systems	3. Larger tracking errors and slower response speed
PD+SC	1. Improves response	1. Complex controller design
	2. Effectively handles multi-time-scale dynamics	2. Higher computational cost
	3. Enhances system robustness	

The system data curves for the PD combined with singular perturbation control are presented in [Figures 10, 11](#).

The comparison curves ([Figure 12](#)) indicate that a singular perturbation controller significantly enhances system tracking performance, reducing the tracking error by about one-third compared to using only a PD controller. As a critical step in evaluating the controller performance, experimental validation, through comparative data, demonstrates that the PD with singular perturbation control effectively reduces system tracking errors, ensuring more accurate trajectory tracking.

From [Table 2](#), we can intuitively observe the advantages and disadvantages of the two control methods. The PD control excels in its simplicity and low computational cost, making it suitable for real-time applications in rigid systems. However, it struggles with multi-time-scale dynamics and external disturbances, leading to larger tracking errors and slower response times. In contrast, the PD + Singular Control (PD+SC) significantly improves response

speed, tracking accuracy, and system robustness by effectively handling multi-time-scale dynamics. Nevertheless, it comes with increased complexity in controller design, higher computational demands, and potential errors due to model simplification. This comparison highlights the trade-offs between simplicity and performance in control system design.

The improvement in tracking error has significant practical implications for patient rehabilitation. By reducing tracking errors, the robot can more precisely follow the patient's movement intentions, minimizing unnecessary mechanical resistance or sudden adjustments. This enhanced precision makes the rehabilitation training feel more natural and comfortable for patients, reducing discomfort caused by inaccurate robot movements. Additionally, smaller tracking errors enable the robot to respond more accurately to the patient's actions during training, enhancing safety by preventing accidents due to control delays or errors. For example, if a patient suddenly stops or changes direction, the robot can adjust quickly and accurately, lowering the risk of injury. Furthermore, precise motion control allows patients to engage in more effective rehabilitation training. By providing consistent and accurate assistance, the robot helps patients better restore their motor functions, potentially

accelerating the recovery process and shortening rehabilitation time. The experimental results allow for further adjustments and optimization of the control parameters to achieve the best control outcomes. This validates the theoretical design's correctness and provides valuable experience and data support for future engineering applications.

5 Conclusion

This paper focuses on developing a flexible lower limb rehabilitation robot, with an in-depth exploration of the construction process for motion control and singular perturbation algorithms. The classical DH method was employed to ensure the accuracy and practicality of the models. The study demonstrates a dynamic zero-load control algorithm to utilize a flexible lower-limb rehabilitation robotic. Take the friction term into the traditional dynamic equation and use the LuGre friction model for friction analysis to realize zero-force control. This approach enabled high-precision control of the end effector force in the robot, achieving effective force feedback management. In this paper, an optimal S-type planning with limited acceleration is proposed, and a complete algorithm for all possible trajectory shapes under various constraints is given. The trajectory planning algorithm proposed in this paper is more consistent with the actual motion performance of the robot. The limitations of traditional singular perturbation control methods in flexible joint robots were anal in control strategize. The rehabilitation robot employing the improved singular perturbation control strategy can rapidly respond to and adjust the applied external forces when the patient actively participates in training, significantly reducing the discomfort and stress risks perceived by the patient. This technology demonstrates significant control accuracy, system response speed, and operational safety advantages.

Looking ahead, we plan to expand further and optimize the findings of this research, including but not limited to improving the accuracy of the dynamic models, optimizing the efficiency of the control algorithms, and enhancing the robustness and adaptability of the system. Additionally, we will actively explore the application potential of this technology in other rehabilitation robot fields, such as upper limb and spinal rehabilitation, to benefit more patients in need of rehabilitation support.

References

- Amersdorfer, M., and Meurer, T. (2022). Equidistant tool path and cartesian trajectory planning for robotic machining of curved freeform surfaces. *IEEE Trans. Automat. Sci. Eng.* 19, 3311–3323. doi: 10.1109/TASE.2021.3117691
- Bhardwaj, S., Khan, A. A., and Muzammil, M. (2021). Lower limb rehabilitation robotics: the current understanding and technology. *Work* 69, 775–793. doi: 10.3233/WOR-205012
- Cao, Y., Chen, X., Zhang, M., and Huang, J. (2023). Adaptive position constrained assist-as-needed control for rehabilitation robots. *IEEE Trans. Indus. Electron.* 71, 4059–4068. doi: 10.1109/TIE.2023.3273270
- Dhanaraj, N., Yoon, Y. J., Malhan, R., Bhatt, P. M., Thakar, S., and Gupta, S. K. (2022). A mobile manipulator system for accurate and efficient spraying on large surfaces. *Proc. Comput. Sci.* 200, 1528–1539. doi: 10.1016/j.procs.2022.01.354
- From, P. J., Gunnar, J., and Gravidahl, J. T. (2010). Optimal paint gun orientation in spray paint applications—experimental results. *IEEE Trans. Automat. Sci. Eng.* 8, 438–442. doi: 10.1109/TASE.2010.2089450
- Gao, R. (2020). Inverse kinematics solution of Robotics based on neural network algorithms. *J. Ambient Intell. Human. Comput.* 11, 6199–6209. doi: 10.1007/s12652-020-01815-4

Data availability statement

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

Author contributions

YX: Writing – original draft. AW: Writing – original draft. XZ: Writing – review & editing. YJ: Writing – original draft. YW: Writing – original draft. HY: Writing – original draft.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This research was funded by Basic Research on Stress induced Bone Growth of Porous Titanium and Its Potential Application in Spinal Fusion Cages (Project No.: 2023JH2/101700136), Basic and Applied Research Project of Liaoning Provincial Science and Technology Planning.

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.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

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.

- Iskandar, M., van Ommeren, C., Wu, X., Albu-Schäffer, A., and Dietrich, A. (2022). Model predictive control applied to different time-scale dynamics of flexible joint robots. *IEEE Robot. Automat. Lett.* 8, 672–679. doi: 10.1109/LRA.2022.3229226
- Jiang, Y., Yu, Y., Zhang, H. Y., Sun, D. K., and Li, Z. H. (2023). Research on application of compliant control of smart meter box handling robot. *Energy Rep.* 9, 1494–1501 doi: 10.1016/j.egy.2023.06.050
- Kevorkian, J., and Cole, J. D. (2012). *Multiple Scale and Singular Perturbation Methods*, Vol. 114. New York, NY: Springer Science & Business Media.
- Khan, R. F. A., Rsetam, K., Cao, Z., and Man, Z. (2022). Singular perturbation-based adaptive integral sliding mode control for flexible joint robots. *IEEE Trans. Indus. Electron.* 70, 10516–10525. doi: 10.1109/TIE.2022.3222684
- Li, K., Wang, H., Liang, X., and Miao, Y. (2021). Visual servoing of flexible-link manipulators by considering vibration suppression without deformation measurements. *IEEE Trans. Cybernet.* 52, 12454–12463. doi: 10.1109/TCYB.2021.3072779
- Pana, C. F., Popescu, D., and Radulescu, V. M. (2023). Patent review of lower limb rehabilitation robotic systems by sensors and actuation systems used. *Sensors* 23:6237. doi: 10.3390/s23136237
- Qassim, H. M., and Wan Hasan, W. Z. (2020). A review on upper limb rehabilitation robots. *Appl. Sci.* 10:6976. doi: 10.3390/app10196976
- Wang, Z., Li, Y., Sun, P., Luo, Y., Chen, B., and Zhu, W. (2021). A multi-objective approach for the trajectory planning of a 7-DOF serial-parallel hybrid humanoid arm. *Mechanism Machine Theory* 165:104423. doi: 10.1016/j.mechmachtheory.2021.104423
- Weingartshofer, T., Hartl-Nesic, C., and Kugi, A. (2023). Automatic and flexible robotic drawing on complex surfaces with an industrial robot. *IEEE Trans. Control Syst. Technol.* 32, 1602–1615. doi: 10.1109/TCST.2023.3345209
- Yu, X. Y., and Chen, L. (2015). Modeling and observer-based augmented adaptive control of flexible-joint free-floating space manipulators. *Acta Astronaut.* 108, 146–155. doi: 10.1016/j.actaastro.2014.12.002
- Zhang, X., Rong, X., and Luo, H. (2024). Optimizing lower limb rehabilitation: the intersection of machine learning and rehabilitative robotics. *Front. Rehabil. Sci.* 5:1246773. doi: 10.3389/fresc.2024.1246773
- Zheng, K., Zhang, Q., and Zeng, S. (2022). Trajectory control and vibration suppression of rigid-flexible parallel robot based on singular perturbation method. *Asian J. Control* 24, 3006–3021. doi: 10.1002/asjc.2729