IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

Direction and Trajectory Tracking Control for Nonholonomic Spherical Robot by Combining Sliding Mode Controller and Model Prediction Controller

Yifan Liu[1], Yixu Wang[1], Xiaoqing Guan[1], Tao Hu[1], Ziang Zhang[1], Song Jin[1], You Wang[1], Jie Hao[2] and Guang Li[1]

***Abstract*—A spherical robot is a nonlinear, nonholonomic, and unstable system which increases the difficulty of the direction and trajectory tracking problem. In this study, we propose a new direction controller Hierarchical Terminal Sliding Mode Controller (HTSMC), an instruction planning controller called Model Prediction Control-based Planner (MPCBP), and a trajectory tracking framework named MPCBP-HTSMC-HSMC (MHH). The HTSMC is designed by integrating a fast terminal algorithm, a hierarchical method, the motion features of a spherical robot, and its dynamics. In addition, the new direction controller has an excellent control effect with a quick response speed and strong stability. MPCBP can obtain optimal commands that are then transmitted to the velocity and direction controller. Since the two torque controllers in MHH are all Lyapunov-based sliding mode controllers, the MHH framework may achieve optimal control performance while assuring stability. Finally, the two controllers eliminate the requirement for MPCBP's stability and dynamic constraints. Finally, hardware experiments demonstrate the efficacy of the HTSMC, MPCBP, and MHH.**

***Index Terms*—Motion control, underactuated robots, integrated planning and control, spherical robots.**

I. INTRODUCTION

THE spherical robot is a novel kind of mobile robot which can be used in a range of fields such as environmental detection, search and rescue, security patrols and entertainment. Compared with other mobile robots like wheeled robots and legged robots, spherical robots have several advantages [1]–[3]. For example, because of its spherical rolling mechanism, the spherical robot can have flexible two-dimensional planar mobility and minimal frictional energy consumption.

Manuscript received: May 10, 2022; Revised July 25, 2022; Accepted August 19, 2022.

This paper was recommended for publication by Editor C. Gosselin upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Key Research and Development Program of China (No. SQ2019YFB130016) and the Open Research Project of the State Key Laboratory of Industrial Control Technology (No. ICT2021B10). (Corresponding author: You Wang.)

[1]Yifan Liu, Yixu Wang, Xiaoqing Guan, Tao Hu, Ziang Zhang, Song Jin, You Wang and Guang Li are with State Key Laboratory of Industrial Control Technology, Institute of Cyber Systems and Control, Zhejiang University, Hangzhou, 310027, China. [yifanliu, yixuwang, 3150103730, 3170104918, zhangziang, optimus, king_wy, guangli]@zju.edu.cn

[2]Jie Hao is with Luoteng Hangzhou Techonlogy Co.,Ltd., Hangzhou, 310027, China.mac@rotunbot.com

Digital Object Identifier (DOI): see top of this page.

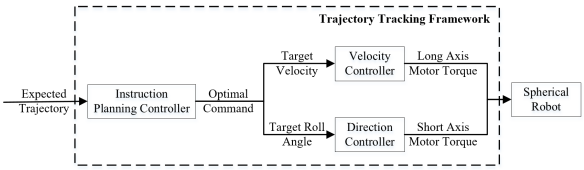


Fig. 1. Trajectory Tracking Framework of the spherical robot.

Furthermore, its closed spherical shell can aid in protecting the inside electrical system and mechanical framework from collision and damage [4]. However, despite its advantages, the spherical robot is difficult to control due to its non-holonomic, non-linear and under-actuated characteristics [5].

Trajectory tracking control plays an important role in the robot's motion and planning. For 2D mobile robots, the trajectory tracking motion can be decomposed into the forward velocity and its direction, as well as the x- and y-axis velocities. The former one is adopted in the spherical robot's trajectory tracking based on its kinematic model, and motion can be obtained if its velocity and direction control are properly executed. However, with such a non-holonomic and under-actuated robot, its velocity and direction cannot be directly implemented, and it is difficult to maintain its stability. Based on the above, we designed a spherical robot's trajectory tracking control framework which includes three portions, i.e., velocity controller and direction controller based on its dynamic models, and instruction planning controller which transmits instructions to torque controllers mentioned above. Fig. 1 illustrates the framework of the tracking mechanism. An efficient velocity controller have been presented in our previous work in [6], [7]. In this study, we focus primarily on constructing the direction controller and instruction planning controller to realize trajectory tracking control of the robot. In addition, the spherical robot's direction control system is a complex, delayed, non-holonomic, and nonlinear system that demands the design of an efficient controller. During trajectory tracking control, stability is extremely challenging, yet crucial.

Over the last few years, a series of works have been made on the control of the spherical robot. The majority of study concentrates on the spherical robot's velocity control [1], [8]–[10]. Several studies, for instance, develop a velocity controller based on feedback control, such as a PI-type fuzzy

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

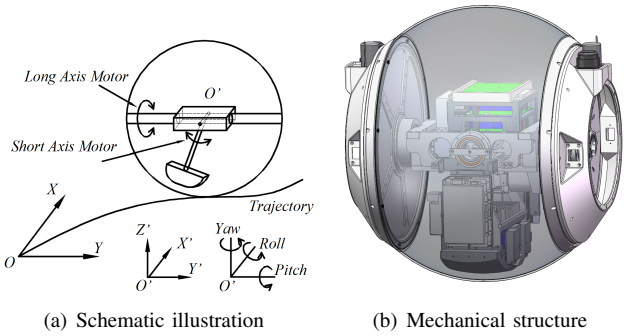
logic controller and sliding mode controller (SMC) [8], [10]. Adaptive velocity controllers for spherical robots are also available by predicting the uncertainty [1], [7]. The path tracking of spherical robots has also been the subject of some studies [4]. For instance, [11] presents a combination of fuzzy logic and SMC algorithm. There are also research focusing on motion planning of spherical robots with optimal planners that place a strong emphasis on obstacles [12], [13]. The majority of spherical robot direction controllers only have simulation results, such as the direction control component in [4], [11], [14]. A full-state feedback controller for roll motion is provided in [15]. Due to the lack of an experiment-verified dynamic model for spherical robots, PID controllers are widely used for these robots and an efficient Fuzzy-PID controller is proposed in [16] last year. However, the nonholonomic property of the spherical robot created feedback control problems [9]. Hierarchical control methods are advantageous for non-holonomic and under-actuated single input multiple output (SIMO) systems [17]. And fast terminal sliding mode controller is a type of SMC that may increase the response speed while maintaining stability [18], [19]. These may assist us in resolving the preceding issue.

There are other studies on trajectory tracking and instruction planning [14], [20], [21]. However, they only provide simulation results without on hardware experiments. Model prediction controller (MPC) is a kind of effective instruction planning controller for mobile robot's trajectory tracking [22]–[24]. As it is quite easy for spherical robots to become unstable, it is crucial to obtain optimal control performance while preserving stability [25].

We propose a trajectory tracking control framework MPCBP-HSMC-HTSMC (MHH) of spherical robot, as well as a direction controller Hierarchical Terminal Sliding Mode Controller (HTSMC) and instruction planning controller called Model Prediction Control-based Planner (MPCBP). Two Lyapnov-based SMCs in the framework guarantee the system's stability, hence eliminating the need to add stability constraints to the instruction planning controller MPCBP. Specifically, this study makes the following contributions:

- We designed the spherical robot's direction controller HTSMC by combining fast terminal sliding mode algorithm, hierarchical method and the spherical robot's motion features. As far as we know, this is the first model-based direction controller for spherical robot that is applied on hardware.
- An integral trajectory tracking control framework is proposed for spherical robot, a task that has never been solved on hardware for spherical robot before. Two Lyapnov-based SMCs assure system stability, which helps simplify the nonlinear program in MPCBP.
- An instruction planning controller MPCBP is presented for spherical robot which is the first control scheme applied on hardware for spherical robot.
- Practical experiments are conducted to demonstrate the controllers' effectiveness. And the whole-body dynamic model of the spherical robot has been given.

The paper is organized as follows: Section II establishes the spherical robot's kinematic and whole-body dynamic models. Section III designs a trajectory tracking control framework of the spherical robot. In section IV, practical experiments are carried out, results and comparisons are presented. Finally, conclusions are given in Section V.



(a) Schematic illustration (b) Mechanical structure

Fig. 2. Mechanical schematic of the spherical robot. The frame OXY is the world frame where O is the frame origin. And O'X'Y'Z' is the local frame centered on the robot O', with X' representing the forward direction.

II. MATHEMATICAL MODELS

The spherical robot is driven by a 2-DOF heavy pendulum and the mechanical schematic of the robot can be seen in Fig. 2. A 2-DOF heavy pendulum driven by two motors is employed to drive the robot. The long axis motor is employed to control the forward velocity while the short axis motor is employed to control the direction by adjusting its roll angle. As a result, the spherical robot may perform flexible motion on a two-dimensional plane by adjusting the output torque of the two motors [3].

To achieve trajectory tracking control, it is necessary to first derive the kinematic and dynamic models. With the kinematic model and instruction planning controller, the robot can obtain the optimal instructions for tracking the trajectory with minimal loss, including velocity and moving direction. In contrast, the dynamic model and the other two controllers determine the output torque to execute the instruction as quickly as possible.

A. Kinematic Model

Assuming the robot moves on a flat terrain without slipping, according to [12], [13], [16], the kinematic model of the spherical robot can be obtained as follows. The global frame OXY in Fig. 2 serves as the frame for the X, Y, ϕ, while the frame O'X'Y'Z' serves as the frame for the q_r, v, θ. By the way, the difference between our kinematic model and the others is that the coordinate's definition varies. And these kinematics will be the same after transforming the coordinates.

$$\begin{cases} \dot{X} = v\cos\phi - R\dot{q}_r\sin\phi \\ \dot{Y} = v\sin\phi + R\dot{q}_r\cos\phi \\ \dot{\phi} = v\tan q_r / R \end{cases} \tag{1}$$

where variables' definition can be seen in Table. I. Restricted by the the long shaft and shell's support bracket of the spherical robot, the heavy pendulum cannot keep rotating in the direction of the roll angle. Finally, the robot's roll angle

TABLE I
LIST OF NOMENCLATURE

Symbol	Description
ϕ	Yaw angle of the robot
α, β	Swing angle of the pendulum in pitch and roll direction
φ, q_r	Pitch angle and roll angle of the robot
x	Distance that the center of the spherical robot moves, and $x = \varphi R$
v	Forward velocity of the spherical robot, and $v = \dot{x}$
τ_v, τ_r	Output torque of the long axis motor and short axis motor
m	Mass of heavy pendulum
m_f	Mass of frame
m_s	Mass of shell
I_{mp}, I_{mr}	Moment of inertia of heavy pendulum in pitch and roll direction
I_{fp}, I_{fr}	Moment of inertia of frame in pitch and roll direction
I_sp, I_r	Moment of inertia of shell in pitch and roll direction
R	Radius of spherical robot
l	Distance between the center of the spherical robot and the center of the pendulum
F_{fp} F_{fr}	Friction between shell and ground in X' and Y' direction
ζ	Viscous damping coefficient

ranges from $-17°$ to $17°$. By integrating the (1), it can be seen that the distance is significantly more affected by the first term of the first two equations than by the second term since the integrated distance caused by the second term is bounded by q_r, which means, in this scenario, velocity v is primarily responsible for determining the moving distance X and Y. Therefore, to simplify the MPCBP' optimal problem, we simplify (1) as follows:

$$\begin{cases} \dot{X} = v\cos\phi \\ \dot{Y} = v\sin\phi \\ \dot{\phi} = v\tan q_r/R \end{cases} \tag{2}$$

B. Whole-body Dynamic Model

The schematic illustration of the spherical robot is given in Fig. 2. The spherical robot can be simplified into three components: a heavy pendulum, a spherical shell, and a central frame. Variables of the robot are represented in Table. I. The Euler-Lagrange method is utilized to construct the spherical robot's whole-body dynamic model. Let L and Ψ represent the lagrangian function and the Rayleigh's dissipation function of the spherical robot, the Euler-Lagrange equations can be written as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \frac{\partial L}{\partial q_k} + \frac{\partial \Psi}{\partial \dot{q}} = \tau_{q_k} \tag{3}$$

where $q_k \in \boldsymbol{q} = \begin{bmatrix}\alpha & x & \beta & q_r\end{bmatrix}^T$, and τ_{q_k} is the related external torque.

Transform (3) into matrix form, then we can get the whole-body dynamic model of the robot as follows:

$$\boldsymbol{M}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{E}\boldsymbol{\tau} \tag{4}$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{4\times4}$ is the inertia matrix, $\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^4$ is the nonlinear matrix and $\boldsymbol{q}$ is the input matrix. And

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} I_{mp}+I_{fp} & ml\cos\alpha & 0 & 0 \\ mRl\cos\alpha & MR+I_s/R & 0 & 0 \\ 0 & 0 & I_{mr} & mRl\cos\beta \\ 0 & 0 & 0 & I_{sr}+I_{fr}+MR^2 \end{bmatrix}$$

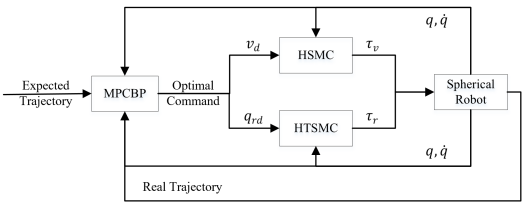


Fig. 3. The control scheme of MHH.

$$\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \begin{bmatrix} mgl\sin\alpha\cos\beta + \zeta\left(\dot{\alpha}+\dot{x}\cos\alpha/R\right) \\ -mRl\dot{\alpha}^2\sin\alpha + \zeta\left(\dot{\alpha}\cos\alpha+\dot{x}/R\right) + F_{fp}R \\ mgl\cos\alpha\sin\beta + \zeta\left(\dot{\beta}+\dot{q_r}\cos\beta\right) \\ -mRl\dot{\beta}^2\sin\beta + \zeta\left(\dot{q_r}+\dot{\beta}\cos\beta\right) + F_{fr}R \end{bmatrix}$$

$$\boldsymbol{E} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}^T, \boldsymbol{\tau} = \begin{bmatrix} \tau_v \\ \tau_r \end{bmatrix}.$$

The whole-body dynamic model can be decomposed into two 2D sub-models according to the motor torque, which are used for designing velocity controller and direction controller, respectively. Two sub-models are shown as follows:

$$\boldsymbol{M_p}\cdot\ddot{\boldsymbol{q}}_{\boldsymbol{p}} + \boldsymbol{N_p} = \begin{bmatrix}1 & 1\end{bmatrix}^T\cdot\tau_v \tag{5}$$

$$\boldsymbol{M_r}\cdot\ddot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{N_r} = \begin{bmatrix}1 & 1\end{bmatrix}^T\cdot\tau_r \tag{6}$$

where $\boldsymbol{M_p}, \boldsymbol{M_r} \in \mathbb{R}^{2\times2}$, and $\boldsymbol{M} = diag\left[\boldsymbol{M_p}, \quad \boldsymbol{M_r}\right]$. $\boldsymbol{N_p}, \boldsymbol{N_p} \in \mathbb{R}^2$, and $\boldsymbol{N} = \left[\boldsymbol{N_p}; \quad \boldsymbol{N_r}\right]$. $\boldsymbol{q_p} = \begin{bmatrix}\alpha & x\end{bmatrix}^T$, $\boldsymbol{q_r} = \begin{bmatrix}\beta & q_r\end{bmatrix}^T$. And in this paper, we build the HTSMC based on (6) to realise spherical robot's direction control. Velocity controller have been proposed in our previous work in [6].

III. CONTROL ALGORITHM

In this section, we first introduce a direction controller HTSMC for spherical robot in subsection A. This controller adjusts the robot's direction by controlling its roll angle. Then, as for the instruction planning controller, we designed an MPCBP controller that uses the reference trajectory to figure out the optimal control instructions. This is described in subsection B. With the velocity controller HSMC proposed in our previous work [6], [7], we finally built the trajectory tracking framework MHH for unstable non-holonomic spherical robot. The scheme of the control structure is shown in Fig. 3. The algorithm will then be talked about in detail.

A. Direction Controller: HTSMC

According to (2), spherical robot's direction can be adjusted by modifying its roll angle since velocity has been specified by the sum of the velocity vectors in the x and y directions. Therefore, the roll angle controller is considered the direction controller. Before designing the controller, we need to transform the (6) into state space form. The state space equations are shown as follows:

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = f_1(\boldsymbol{\chi}) + b_1(\boldsymbol{\chi})\,\tau_r \\ \dot{x_3} = x_4 \\ \dot{x_4} = f_2(\boldsymbol{\chi}) + b_2(\boldsymbol{\chi})\,\tau_r \end{cases} \tag{7}$$

where $\boldsymbol{\chi} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T = \begin{bmatrix} \beta & \dot{\beta} & q_r & \dot{q}_r \end{bmatrix}^T$ is the state variable of the system, and

$$\begin{cases} \begin{bmatrix} f_1(\boldsymbol{\chi}) & f_2(\boldsymbol{\chi}) \end{bmatrix}^T & = -\boldsymbol{M_r}^{-1} \cdot \boldsymbol{N_r} \\ \begin{bmatrix} b_1(\boldsymbol{\chi}) & b_2(\boldsymbol{\chi}) \end{bmatrix}^T & = \boldsymbol{M_r}^{-1} \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}^T . \end{cases}$$

Obviously it is a complex second-order control system, with the control purpose of making $q_r - q_{rd} = 0$ where q_{rd} is the target roll angle. Additionally, the system is an SIMO one, and we adjust q_r while influencing β. States associated with β may cause instability in the control process if not taken into consideration, thus we have to add constraint for β. Under these conditions, the HTSMC is proposed for this system. HTSMC has two layer sliding surfaces, the first of which is combined with two subsystem sliding surfaces, respectively are $S_1(t)$ and $S_2(t)$. S_1 is also called the control surface which aims to accomplish the control purpose. The equation of S_1 is shown in (8), and it is a kind of fast terminal SMC. In the sliding phase, the convergence speed will become $-ae - ce^{\frac{p}{q}}$. When error is far away from zero, the two sliding surface's speed are mainly determined by the term $-ae$ whose fast convergence when far away from zero is well understood. When error is close to zero, its convergence speed will mostly rely on its second term $-ce^{\frac{p}{q}}$ which is faster than $-ae$ in traditional surface which is the reason the terminal surface can shorten time in sliding phase.

$$S_1 = \dot{e}_r + a_1 e_r + c_1 e_r^{\frac{p_1}{q_1}} \tag{8}$$

where $e_r = q_r - q_{rd}$, and a_1, c_1, p_1, q_1 are control parameters. And $0 < p_1 < q_1$, $a_1 > c_1$.

S_2 is called the stability surface to keep the stability by taking β's control into consideration. When designing S_2, we found the heavy pendulum swing β about the roll axis to balance lateral friction according to spherical robot's physical characteristics. Then we can get target pendulum swing angle β_d as follows:

$$\beta_d = \arcsin \frac{F_{fr} R}{mgl \cos \alpha} \tag{9}$$

F_{fr} is the static friction since the robot moves without slipping and cannot be acquired. However, we can estimate the terminal $\hat{F}_{fr}$ according to q_{rd} and the kinematic model. Turning radius R_{turn} can be obtained when the roll angle has been adjusted to q_{rd}. And at this time $\hat{F}_{fr}$ is all applied to provide the centripetal force. Then we can get:

$$R_{turn} = \frac{R}{\tan q_{rd}} \tag{10}$$

$$\hat{F}_{fr} = \frac{Mv^2}{R_{turn}} \tag{11}$$

Replacing F_{fr} with $\hat{F}_{fr}$, combining equations in (9), (10) and (11), then we can get β_d as follows:

$$\beta_d = \arcsin \frac{Mv^2 \tan q_{rd}}{mgl \cos \alpha} \tag{12}$$

When q_r is adjusted to q_{rd}, β should equal to β_d. After finding this constraint, we can design the stability surface S_2:

$$S_2 = \dot{e}_\beta + a_2 e_\beta + c_2 e_\beta^{\frac{p_2}{q_2}} \tag{13}$$

where $e_\beta = \beta - \beta_d$, and a_2, c_2, p_2, q_2 are control parameters. And $0 < p_2 < q_2$. q_1 and q_2, p_1 and p_2 can be equal by default, these four parameters are all odd numbers. When fine-tuning, we can specify $q_1 = q_2 = p_1 = p_2$ first, and then gradually decrease p_1 and p_2. However, $p_1/q_1, p_2/q_2$ cannot be too small to keep the system's stability. And $c_1 = c_2 = 1$ can be set by default.

The second layer sliding surface $S(t)$ can be seen as the combination of the first layer terminal sliding surfaces. The expression is shown as follows:

$$S = A \cdot S_1 + B \cdot S_2 \tag{14}$$

where $A > B > 0$ are weight scale coefficients of the two subsystem surfaces. When fine-tuning, specify $A = B$ first, and then gradually decrease B before the system is about to overshoot. According to (7), (8) and (13), we can get the derivative of the second layer sliding surface as follows.

$$\begin{aligned} \dot{S} &= A \cdot \dot{S}_1 + B \cdot \dot{S}_2 \\ &= A(f_2 - \ddot{q}_{rd}) + B(f_1 - \ddot{\beta}_d) + (Ab_2 + Bb_1)\tau_r + Aa_1 \dot{e}_r \\ &\quad + Ba_2 \dot{e}_\beta + Ac_1 \frac{p_1}{q_1} e_r^{1-\frac{p_1}{q_1}} \dot{e}_r + Bc_2 \frac{p_2}{q_2} e_\beta^{1-\frac{p_2}{q_2}} \dot{e}_\beta \end{aligned} \tag{15}$$

To drive the system to reach the sliding surface $S(t) = 0$ as soon as possible, we choose the same reaching law as the velocity controller HSMC which is shown in (16). The reaching law depends the control process and reaching speed.

$$\dot{S} = -k \cdot S - \varepsilon \cdot \operatorname{sgn}(S), \tag{16}$$

where k determines the reaching speed in the reaching phase, and it cannot be too large to prevent overshoot. The system may oscillate when ε is too tiny, while being stable with chattering when it is too high. Based on the equations in (15) and (16), we can get the control law of the output torque τ_r as follows:

$$\begin{aligned} \tau_r = \frac{-1}{(Ab_2 + Bb_1)} & [A(f_2 - \ddot{q}_{rd}) + B(f_1 - \ddot{\beta}_d) + Aa_1 \dot{e}_r \\ & + Ba_2 \dot{e}_\beta + Ac_1 \frac{p_1}{q_1} e_r^{1-\frac{p_1}{q_1}} \dot{e}_r + Bc_2 \frac{p_2}{q_2} e_\beta^{1-\frac{p_2}{q_2}} \dot{e}_\beta \\ & + k \cdot S + \varepsilon \cdot \operatorname{sgn}(S)] \end{aligned} \tag{17}$$

Stability analysis of the HTSMC is shown as below. Take Lyapunov function as follows:

$$L = \frac{1}{2} S^2 \tag{18}$$

Apply (17) to (15), then the derivative of L will be:

$$\begin{aligned} \dot{L} &= S\dot{S} \\ &= S[-k \cdot S - \varepsilon \cdot \operatorname{sgn}(S)] \\ &= -k \cdot S^2 - \varepsilon \cdot |S| \le 0 \end{aligned} \tag{19}$$

According to (19), $\dot{L} \le 0$. And when $\dot{L} \equiv 0$, $S \equiv 0$. According to the Lasalle's invariance principle, this system is asymptotically stable [26]. Additionally, while fine-tuning the parameters, A, a_1, c_1, k are used to shorten the response time and reduce the steady state error, whereas B, a_2, c_2, ε can be raised when the system's stability is not being properly maintained or overshoot is occurred.

Finally, the direction controller HTSMC is obtained. And the velocity controller HSMC can be found in our earlier work [6], the key difference between the two is the design of sliding surface. As the velocity control system is a first-order system, it will be simpler to maintain stability and control. The sliding surface of HSMC S_v is depicted in (20), with the first term indicating the error between the robot's velocity and velocity command and the second term representing the heavy pendulum's angular velocity. Based on the surface, the reaching law, and the Lyapunov function, the control law for the long-axis motor can be derived. And we have now got the two torque controllers.

$$S_v = A_v \cdot e_v + B_v \cdot \dot{\alpha} \tag{20}$$

B. Instruction Planning Controller: MPCBP

Instruction planning controller is necessary to tell the robot how to track the reference trajectory with minimum cost. Then two torque controllers will work to execute the instructions. In this subsection, an MPCBP controller is designed to find the optimal control policy. The cost function $J(\cdot)$ that has to be minimized to control the trajectory of the robot is in general given by the equation below:

$$\min_{\overline{\boldsymbol{U}}(\cdot)} \quad J(\cdot) = \sum_{i=1}^{N} \left\| \overline{\boldsymbol{X}} - \boldsymbol{X}_{ref} \right\|_{\boldsymbol{Q}} + \sum_{i=1}^{N-1} \left\| \overline{\boldsymbol{U}} - \boldsymbol{U}_{ref} \right\|_{\boldsymbol{R}} \tag{21}$$

$$\text{s.t.} \quad \dot{\overline{\boldsymbol{X}}}(i|k) = F\left(\overline{\boldsymbol{X}}(i|k), \overline{\boldsymbol{U}}(i|k)\right), \tag{21a}$$

$$\overline{\boldsymbol{X}}(0|k) = \overline{\boldsymbol{X}}_k, \ \ \overline{\boldsymbol{U}}(0|k) = \overline{\boldsymbol{U}}_k, \tag{21b}$$

$$\overline{\boldsymbol{X}}(i|k) \in \left[\overline{\boldsymbol{X}}_{min}, \ \ \overline{\boldsymbol{X}}_{max}\right], \tag{21c}$$

$$\overline{\boldsymbol{U}}(i|k) \in \left[\overline{\boldsymbol{U}}_{min}, \ \ \overline{\boldsymbol{U}}_{max}\right], \tag{21d}$$

$$G\left(\overline{\boldsymbol{X}}(i|k), \ \ \overline{\boldsymbol{U}}(i|k)\right) \leq 0 \tag{21e}$$

where N is the prediction horizon, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are definite weighting coefficient matrix, the states and inputs are defined as $\overline{\boldsymbol{X}} = \begin{bmatrix} X & Y & \phi \end{bmatrix}^T$ and $\overline{\boldsymbol{U}} = \begin{bmatrix} v & q_r \end{bmatrix}^T$, respectively. $\boldsymbol{X}_{ref} = \begin{bmatrix} X_{ref} & Y_{ref} & \phi_{ref} \end{bmatrix}^T$ and $\boldsymbol{U}_{ref} = \begin{bmatrix} v_{ref} & q_{ref} \end{bmatrix}^T$ are reference trajectory and reference inputs, respectively. (21a) is the kinematic model which is also presented in (2). Initial states and inputs are defined as $\overline{\boldsymbol{X}}_k$ and $\overline{\boldsymbol{U}}_k$, respectively, which are obtained by sensors. Limitations for states and inputs are respectively presented in (21c) and (21d). Inequality constraint (21e) is employed to indicate the boundary value problem of direct multi-shooting method. Direct multi-shooting method can break down the system integration into short time intervals and break the trajectory into pieces. And (21e) can be transformed into the following equation, in which g_{min} and g_{max} are its boundaries.

$$G : \overline{\boldsymbol{X}}(i+1|k) - \dot{\overline{\boldsymbol{X}}}(i|k) \cdot \mathrm{d}t - \overline{\boldsymbol{X}}(i|k) \in [g_{min}, \ \ g_{max}] \tag{22}$$

After defining the optimization problem, we solve it by employing the Sequential Quadratic Programming (SQP) approach. The sequence of optimal control inputs $\overline{\boldsymbol{U}}(\cdot)$ is then obtained. Then we apply $\overline{\boldsymbol{U}}(1|k)$ to velocity controller HSMC and direction controller HTSMC. Following that, the two controllers execute the optimal commands in a predictable manner, and transfer torque command to the motors. Then we can get the optimal trajectory which is close to the reference trajectory. Finally the whole trajectory tracking framework MHH is established.

Furthermore, to make sure the system is stable when tracking trajectories, MPCBP may need stability constraints like limits on angle velocity and acceleration. And dynamic model may also be required. This may make the optimal problem hard to solve. However, two Lyapunov-based sliding mode controllers ensures the stability of the system. This can eliminate the requirement for stability constraints for MPCBP, which can simplify the nonlinear program in (21). Therefore, the trajectory tracking framework can be of stability.

IV. EXPERIMENTAL RESULTS

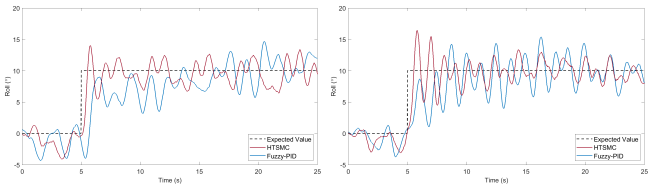
In this section, we aims to verify the effectiveness and robustness of the direction controller (or named roll angle controller) HTSMC and trajectory tracking framework MHH. In subsection A, experiments are carried out to test the roll angle controller HTSMC's tracking effect. For comparison, the same experiments were conducted using the fuzzy-PID controller proposed in our previous work which is verified to be better than traditional PID controller [16]. In subsection B, two trajectory tracking frameworks are compared by tracking three kinds of trajectories. One is the MHH and the other is MHF (MPCBP-HSMC-Fuzzy-PID) whose direction controller is fuzzy-PID rather than HTSMC.

Furthermore, all of the experiments are carried out on the flat tiled floor, where whole-body dynamic model's parameters were obtained through system identification. Information about angles, velocities, and accelerations is obtained using an IMU and an encoder. An extended Kalman filter is employed in the two IMUs we use to deal with the noise problem and drift issue. Algorithms run in concurrent threads on a mini PC (Intel i7-8559U, 2.70 GHz, Quad-core 64-bit). The calculated output torque will be transmitted to the lower computer (TI TMS320F28069) to control the motors. Moreover, the sampling time is 20ms, control frequency is 50Hz for velocity controller HSMC and direction controller HTSMC, while the prediction frequency is 10Hz for MPCBP.

A. Direction Control Experiments

1) Single Roll Angle Tracking Under Different Velocities: In this portion, our purpose is to turn the spherical robot left at velocities of 0.5m/s and 1.0m/s. As previously explained, turning motion can be achieved by controlling its roll angle. And in this experiments, we hope the robot can move straight for the first five seconds and then turn left immediately with the expected roll angle of 10°. The results are shown in Fig. 4.

In order to better compare the control effect of the two controllers, we selected five indicators, i.e. rising time t_r, setting time t_s, mean error e_m which is the average error in the turning process (from 5s to the end), root mean square error value of the roll angle e_{rmse} in steady-state process and the error boundary Δh. We assume the rising time is the time



(a) When velocity is 0.5m/s (b) When velocity is 1.0m/s

Fig. 4. Single roll angle tracking.

TABLE II
SINGLE ROLL ANGLE CONTROL EFFECTS.

Velocity	Controller	Indicators				
		$t_r(s)$	t_s (s)	e_m (°)	e_{rmse} (°)	Δh(°)
When 0.5m/s	HTSMC	0.54	1.61	-0.358	1.727	-3.59, +3.38
	Fuzzy-PID	8.51	6.86	-0.351	2.097	-6.28, +4.61
When 1.0m/s	HTSMC	0.54	2.68	-0.097	1.509	-3.88, +3.39
	Fuzzy-PID	3.41	3.66	-0.172	2.5373	-5.968, +5.32

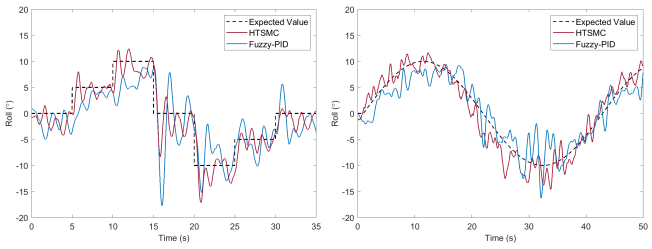
when the system first reaches the expected value. And steady-state is reached when envelope of the angle starts to remain basically horizontal. Results are shown in Table. II.

According to results in Fig. 4 and Table. II, it is obvious that HTSMC has much smaller rising time and setting time compared with Fuzzy-PID, which means it can reach the setting point very fast, able to return stable and steady state quickly. This is owing to the design of its hierarchical sliding surfaces. We also tried to increase the k_p or k_i of Fuzzy-PID, but the outcome was system oscillation and it was difficult to reestablish stability. On the other hand, results also showed that HTSMC has higher stability than Fuzzy-PID, since it has much smaller e_{rmse} and Δh. It is also proved that our new controller has similar good performance under different velocities, which means high robustness.

2) Multiple Discrete Roll Angle Tracking: In this test, we would like to operate the spherical robot to track a time-variant roll angle which is a discrete function. The expected roll angle changes every 5 seconds. HSMC controls the robot to keep a velocity of 0.5m/s. The results can be seen in Fig. 5(a).

In order to further compare the control performance of the two controllers, we choose their root mean square error value e_{rmse} and mean absolute error e_{mae} of the tracking roll angle in the whole procedure to indicate the tracking effect. Results of the two indicators are shown in Table. III.

According to results in Fig. 5(a) and Table. III, it is obvious that HTSMC offers superior tracking performance and fewer errors than Fuzzy-PID. And the HTSMC controller offers a quicker response time and a shorter rising time. HTSMC exhibits a similar control effect and robustness for all expected angles in this experiment, while the other one does not. Specifically, the data imply that the HTSMC has higher stability in discrete roll angle tracking.



(a) Multiple discrete roll angle. (b) Sine-wave tracking experiment.

Fig. 5. Multiple Roll Angle Tracking.

3) Sine-wave Tracking of Roll Angle Controller: The expected time-variant roll angle in this part is a sine-wave like function which is also shown in (23). We also apply velocity controller HSMC to keep the robot's velocity at 0.5m/s. Final results are shown in Fig. 5(b).

$$q_{rd} = \begin{cases} 0 & ,t < 0.1\text{s} \\ 10\sin(0.15t - 0.015) & ,t >= 0.1\text{s} \end{cases} \tag{23}$$

To further compare the control performance of the two controllers, we again select their root mean square error value e_{rmse} and mean absolute error e_{mae} of the tracking roll angle during the entire procedure to represent the tracking effect. The outcomes of the two indicators are likewise displayed in Table. III.

Based on the results in Fig. 5(b) and Table. III, HTSCM can track the sine-wave expected angles well with much smaller e_{mae} and has more stability with smaller e_{rmse}. It seems that there is a tracking delay for Fuzzy-PID, which may be owing to its slower response speed and need for time to accumulate. This results also prove that HTSMC can handle continuous roll angle tracking problem. And with such a direction controller, the system can be more stable in trajectory tracking.

B. Trajectory Tracking Experiments

In this part, three types of trajectories will be tracked to compare two trajectory tracking frameworks. A same instruction planing controller MPCBP proposed in III.B and a same velocity controller HSMC proposed in [6], [7] are employed for these two tracking frameworks. And the main difference is the direction controller, with MHH employing HTSMC and MHF Fuzzy-PID. Experiments in this subsection attempt to elucidate the effect of MPCBP and MHH on the control of the trajectory.

For all of the trajectories, the initial state $\overline{\boldsymbol{X}} = \begin{bmatrix} X & Y & \phi \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ and $\overline{\boldsymbol{U}} = \begin{bmatrix} v & q_r \end{bmatrix}^T = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$. Parameters of the MPCBP are set as below. $N = 10$, $\boldsymbol{Q} = diag\begin{bmatrix} 2 & 2 & 1 \end{bmatrix}$, $\boldsymbol{R} = diag\begin{bmatrix} 1 & 1 \end{bmatrix}$. Green arrays in figures below represent the initial state. Trajectories can all be transformed into math formulation as follows, and experiment results are shown as below as well. Furthermore, CasADi

TABLE III
TIME-VARIANT ANGLE TRACKING.

Velocity	Controller	Indicators	
		e_{mae} (°)	e_{rmse} (°)
Discrete Tracking	HTSMC	1.852	2.511
	Fuzzy-PID	3.191	4.128
Sine-wave Tracking	HTSMC	1.631	2.082
	Fuzzy-PID	2.115	2.600

framework, SQPmethod and qpOASEs are employed to solve the nonlinear program in (21) [27].

1) Tracking a Sine-wave:

$$\begin{cases} X_{ref} &= 0.5t \\ Y_{ref} &= 2\sin(0.25t) \\ \phi_{ref} &= \arctan\cos(0.25t) \end{cases} \tag{24}$$

where $t \in [0, 12\pi]$. Results are shown in Fig. 6.

2) Tracking a Lemniscate of Gerono:

$$\begin{cases} X_{ref} &= 8\sin(t/16) \\ Y_{ref} &= 8\sin(t/16)\cos(t/16) \\ \phi_{ref} &= \text{atan2}\left(\cos(t/8)/\cos(t/16)S_{ign},\ \ S_{ign}\right) \\ S_{ign} &= \text{sgn}(\cos(t/16)) \end{cases} \tag{25}$$

where $t \in [0, 32\pi]$, atan2$(\cdot)$ is a function in C++ which is a special arctan$(\cdot)$ function with returns in $[-\pi, \pi]$. Results are shown in Fig. 7.

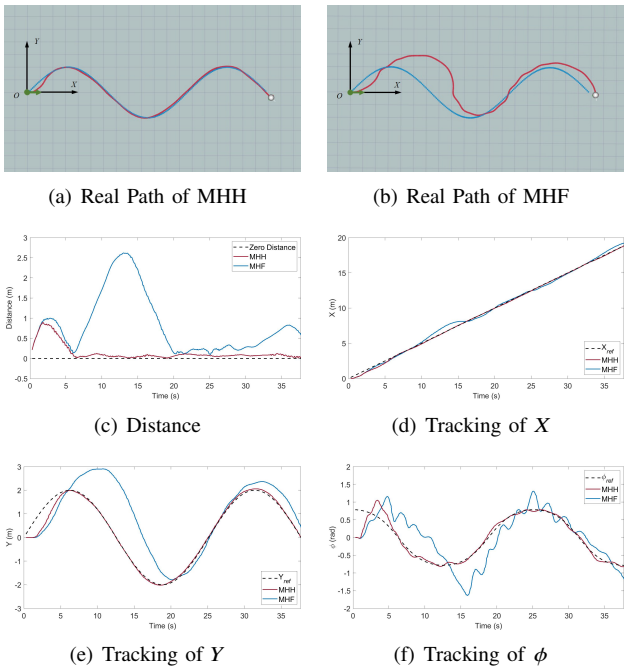
3) Tracking a Trajectory with Disturbance: when 0 <= t < 12 s:

$$\begin{cases} X_{ref} &= 0.5t \\ Y_{ref} &= 0 \\ \phi_{ref} &= 0 \end{cases} \tag{26}$$

When 12 <= t < 12π s:

$$\begin{cases} X_{ref} &= 4\cos((t-12)/8 - \pi/2) + 6 \\ Y_{ref} &= 4\sin((t-12)/8 - \pi/2) + 4 \\ \phi_{ref} &= (t-12)/8 \end{cases} \tag{27}$$

where $t \in [0, 12 + 12\pi]$. The trajectory, which is made up of a line and a circle trajectory, has two disturbances placed in its path. The two disturbances are acupressure mats with dimensions of $2.0m \times 0.6m$, the center points of which are located at (3, 0) and (10, 4), respectively. Details and results can be seen in Fig. 8.

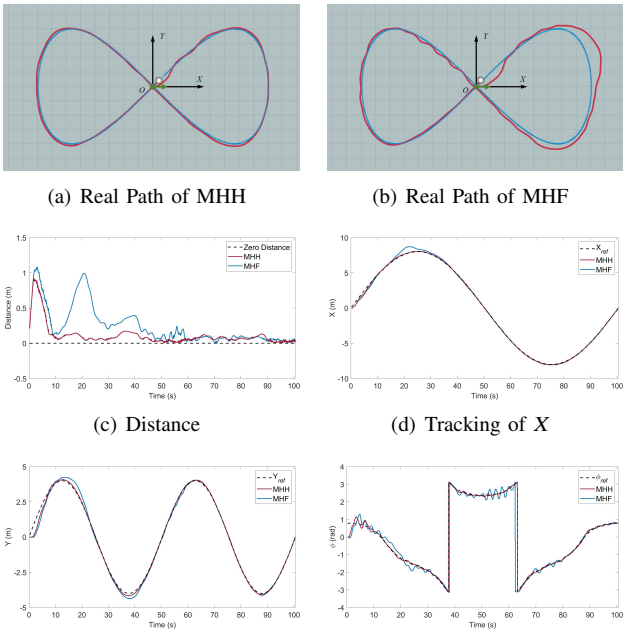


(a) Real Path of MHH (b) Real Path of MHF

(c) Distance (d) Tracking of X

(e) Tracking of Y (f) Tracking of ϕ

Fig. 6. Tracking the trajectory of a sine-wave.

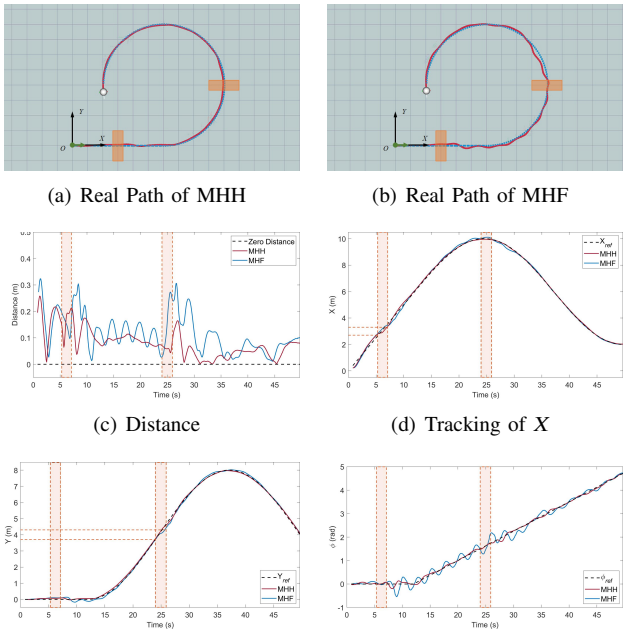


(a) Real Path of MHH (b) Real Path of MHF

(c) Distance (d) Tracking of X

(e) Tracking of Y (f) Tracking of ϕ

Fig. 7. Tracking the trajectory of a lemniscate of gerono.



(a) Real Path of MHH (b) Real Path of MHF

(c) Distance (d) Tracking of X

(e) Tracking of Y (f) Tracking of ϕ

Fig. 8. Tracking the trajectory with disturbance. The orange-red areas are disturbances, which are made up of acupressure mats.

4) Analysis: According to Fig. 6 and Fig. 7, it is obvious that MHH can track trajectories considerably better than MHF with less errors, especially when the robot is turning continuously. MHH's true curve is also smoother and has less fluctuation. According to distance figures, it can be proved that MHH can track the expected trajectory faster. Based on the figures of ϕ tracking, MHH appears to be more stable than MHF, with less oscillation and a smoother slope. Furthermore,

according to Fig. 8, when traveling through the disturbance region, the MHH is less impacted by the disturbances and soon regains stability, whereas the MHF is severely affected by the disturbances, recovers slowly, and oscillates. And this is more apparent in the distance figure. All in all, MHH has superior anti-interference capability, higher robustness, and better control effect.

More information on those on-hardware experiments can be found in the associated video. From the experiments, it also can be imply that HTSMC has better control effect in direction tracking compared with Fuzzy-PID. HTSMC is more stable, maybe because its design is based on the Lyapunov function. As a result, stability constraints are not required for MPCBP to remain stable. Overall, MHH is an effective trajectory tracking framework for spherical robots that are non-holonomic, nonlinear, and under-actuate.

V. CONCLUSIONS

A new direction controller, HTSMC, is devised in this work to enable spherical robots follow expected roll angles more stable and quickly. Hierarchical algorithm, fast terminal sliding mode controller, and spherical robot motion features make up this controller. Furthermore, we present an instruction planning controller MPCBP, and ultimately, we combine two sliding mode controllers and a model prediction controller to create a spherical robot's trajectory tracking framework MHH.

Then we carry out a series of experiments on hardware. The new direction controller provides greater control performance and stability, according to the results. The new trajectory tracking framework is more stable thanks to two Lyapunov-based sliding mode controllers, which eliminate the need for external stability restrictions for MPCBP. Finally, with the new controller, the spherical robot can have broader applications.

There is still work to be done in our future work. First we need a new rubber layer outside the shell with special texture that will prevent the robot from slipping on relatively smooth terrain, such as oiled ground. Second, to enable multi-terrain trajectory tracking with a spherical robot, we are attempting to design a new method, which can be integrated with the MHH as compensation, for obtaining uncertainties in our whole-body dynamic model and kinematic model.

REFERENCES

[1] E. Kayacan, E. Kayacan, H. Ramon, and W. Saeys, "Adaptive neuro-fuzzy control of a spherical rolling robot using sliding-mode-control-theory-based online learning algorithm," *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 170–179, 2013.

[2] S. Chen and J. T. Wen, "Adaptive neural trajectory tracking control for flexible-joint robots with online learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2358–2364.

[3] L. Ma, H. Sun, and J. Song, "Fractional-order adaptive integral hierarchical sliding mode control method for high-speed linear motion of spherical robot," *IEEE Access*, vol. 8, pp. 66 243–66 256, 2020.

[4] M. Taheri Andani, Z. Ramezani, S. Moazami, J. Cao, M. M. Arefi, and H. Zargarzadeh, "Observer-based sliding mode control for path tracking of a spherical robot," *Complexity*, vol. 2018, pp. 1–15, 10 2018.

[5] S.-B. Chen, A. Beigi, A. Yousefpour, F. Rajaee, H. Jahanshahi, S. Bekiros, R. A. Martínez, and Y. Chu, "Recurrent neural network-based robust nonsingular sliding mode control with input saturation for a non-holonomic spherical robot," *IEEE Access*, vol. 8, pp. 188 441–188 453, 2020.

[6] Y. Liu, Y. Wang, X. Guan, T. Hu, Y. Wang, and G. Li, "New hierarchical sliding mode control method for velocity tracking of the spherical robot," in *2021 China Automation Congress (CAC)*, 2021, pp. 1455–1460.

[7] Y. Liu, Y. Wang, X. Guan, Y. Wang, S. Jin, T. Hu, W. Ren, J. Hao, J. Zhang, and G. Li, "Multi-terrain velocity control of the spherical robot by online obtaining the uncertainties in the dynamics," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2732–2739, 2022.

[8] N. N. Kamis, A. H. Embong, and S. Ahmad, "Velocity control for spherical robot using pi-fuzzy logic," in *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, 2019, pp. 155–160.

[9] M. ZHENG, Q. ZHAN, J. LIU, and Y. CAI, "Control of a spherical robot: Path following based on nonholonomic kinematics and dynamics," *Chinese Journal of Aeronautics*, vol. 24, no. 3, pp. 337–345, 2011.

[10] M. Ayati and S. Zarei, "Fault detection algorithm based on sliding-mode method for spherical rolling robots," in *2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM)*, 2017, pp. 334–339.

[11] C. Yao, Z. Qiang, and X. Xi, "Path tracking control of a spherical mobile robot," *Mechanism and Machine Theory*, vol. 51, pp. 58–73, 2012.

[12] M. Roozegar, M. Mahjoob, and M. Jahromi, "Optimal motion planning and control of a nonholonomic spherical robot using dynamic programming approach: simulation and experimental results," *Mechatronics*, vol. 39, 05 2016.

[13] M. Roozegar, M. Mahjoob, M. Esfandyari, and M. Shariat Panahi, "Xcs-based reinforcement learning algorithm for motion planning of a spherical mobile robot," *Applied Intelligence*, vol. 45, 10 2016.

[14] S. Zihao, W. Bin, and Z. Ting, "Trajectory tracking control of a spherical robot based on adaptive pid algorithm," in *2019 Chinese Control And Decision Conference (CCDC)*, 2019, pp. 5171–5175.

[15] D. Liu, H. Sun, Q. Jia, and L. Wang, "Motion control of a spherical mobile robot by feedback linearization," in *2008 7th World Congress on Intelligent Control and Automation*, 2008, pp. 965–970.

[16] Y. Wang, X. Guan, T. Hu, Z. Zhang, Y. Wang, Z. Wang, Y. Liu, and G. Li, "Fuzzy pid controller based on yaw angle prediction of a spherical robot," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 3242–3247.

[17] D. Qian, J. Yi, and D. Zhao, "Hierarchical sliding mode control for a class of simo under-actuated systems," *Control and Cybernetics*, vol. 37, 01 2008.

[18] X. Yu and M. Zhihong, "Fast terminal sliding-mode control design for nonlinear dynamical systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 2, pp. 261–264, 2002.

[19] A. Singh and Q. Ha, "Fast terminal sliding control application for second-order underactuated systems," *International Journal of Control, Automation and Systems*, vol. 17, 05 2019.

[20] Y. Bai, M. Svinin, and M. Yamamoto, "Adaptive trajectory tracking control for the ball-pendulum system with time-varying uncertainties," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2083–2090.

[21] Y. Cai, Q. Zhan, and C. Yan, "Two-state trajectory tracking control of a spherical robot using neurodynamics," *Robotica*, vol. 30, no. 2, p. 195–203, 2012.

[22] T. P. Nascimento, C. E. T. Dórea, and L. M. G. Gonçalves, "Nonholonomic mobile robots' trajectory tracking model predictive control: a survey," *Robotica*, vol. 36, no. 5, p. 676–696, 2018.

[23] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952–964, 2017.

[24] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1398–1404.

[25] R. Grandia, A. J. Taylor, A. Singletary, M. Hutter, and A. Ames, "Nonlinear model predictive control of robotic systems with control lyapunov functions," in *Proceedings of Robotics: Science and Systems*, 2020.

[26] X. Liu, R. Han, and X. Dong, "Adaptive fuzzy wavelet network sliding mode control for dual-robot system with time delay and dynamic uncertainty," *IEEE Access*, vol. 7, pp. 73 564–73 572, 2019.

[27] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.