

An algorithm of foot end trajectory tracking control for quadruped robot based on model predictive control

Jiaxin Guo, Yukun Zheng, Daoxiao Qu, Rui Song, Yibin Li

School of Control Science and Engineering
Shandong University
Jinan, China

jaxin1222@foxmail.com, zhengyk163@163.com,
1306561488@qq.com, rsong@sdu.edu.cn, liyb@sdu.edu.cn

Abstract—The control of the leg mechanism of the quadruped robot has always been one of the research hotspots in the field of robotics. This paper proposes to use the model predictive control method (MPC) to control the foot track tracking of the quadruped robot. Using the Newton-Euler formula to model the dynamics of the leg mechanical structure and obtaining its discrete linear state space model. According to the performance indicators and constraints given by the system, the actual output is feedback corrected to achieve real-time tracking of the required joint rotation angle trajectory.

Index Terms—Joint torque control, model predictive control, force control, quadruped robot.

I. INTRODUCTION

Mobile robots have broad application prospects in the fields of industry, agriculture, military, etc. Among them, legged robots have stronger obstacle-proof ability than wheeled robots or tracked robots, and can travel more flexibly and stably in the rugged terrain of mountains, jungles and swamps. Considering that the mechanical structure of the multi-footed robot is complicated and the floor space is large; the biped robot is difficult to adjust the balance when walking on uneven terrain, and the load capacity is limited; the quadruped robot has strong adaptability to the environment and has great advantages in terms of walking flexibility, load-bearing stability and operational efficiency. Therefore, the quadruped robot has been widely used in the complicated and harsh working environment, and it is currently one of the research hotspots in the field of robotics.

Among the many excellent quadruped robots that have been reported at home and abroad, the various versions of BigDog released by Boston Dynamics have shown excellent levels in complex terrain adaptability, mobility flexibility and anti-interference ability [1,2]. The quadruped robots Star1ETH and ANYmal developed by the Swiss Federal Institute of Technology can achieve diagonal trotting, adapt to uneven terrain and resist external impact [3,4]. Each version of Cheetah, developed by MIT's Biomimetic Robotics Lab, continues to revolutionize motor drives and leg mechanisms for continuous flexibility and stability [5-7].

There have been many studies on trajectory planning tracking control. Reference [8] proposes a continuous free gait planning method. During the walking process, the quadruped robot can independently select the step sequence based on the position of the optional landing point known in advance in the terrain, and select the optimal landing point for the swinging

foot. Through the swing of the center of gravity, it is ensured that the stability margin during the walking process is not less than the minimum stability margin. Based on dynamics and PD force control method, the force control method based on foot force distribution and the force control method based on the impedance control are given to realize the force control of the quadruped robot. Reference [9] estimates the walking stability of the robot from the minimum distance from the supporting edge of the quadruped robot to the ground pressure center. This general criterion considers the interference force, the height of the center of gravity, the inclination of the supporting plane during the motion of the quadruped robot and the quality of the robot, to ensure the stability of the robot movement. For the nonlinearity of joint load, the fuzzy adjustment algorithm is used to achieve good joint speed tracking. The integration of the control system lays the foundation for the offline control of the robot, realizes the intermittent crawling gait of the robot in different environments, and coordinates the walking gait. Reference [10] used the sine function to establish the CPG model, analyzed the motion control mechanism of the quadruped robot, and set the parameters of the robot CPG control network. According to different complex terrains, the gait switching mode based on CPG is described. Then the robot reflection model is established based on the adaptive characteristics of CPG network, which proves that CPG gait control method has excellent environmental adaptability. Reference [11] proposed a simple and robust intuitive controller based on VMC for quadruped trotting. Apply virtual force to control trunk movement during the support phase to adjust posture, height and speed. During the flight phase, virtual model control is applied to the flying toes to track the planned trajectory based on the lateral speed of the torso and the leg contact signal design. This virtual model control method uses the state machine to view the leg state and generate parameters through the advanced controller to control the robot trot state.

The above several control methods are basically based on the condition that the object is completely known. However, in the actual operation process, the terrain is unknown and rugged, which requires that the foot of the quadruped robot can track the desired trajectory in real time and ensure its tracking accuracy. MIT has developed a novel Policy Specification Model Predictive Control (PR-MPC) framework that enables MIT Cheetah quadruped robots to follow constraints while finding better solutions, tracking required

trajectories in real time, and performing general robust legged locomotion [12]. However, its control algorithm has not been reported yet, so it is still challenging to realize the real-time tracking of the foot trajectory of the quadruped robot based on the model predictive control.

This paper studies the kinematics and dynamics of the leg mechanism of a quadruped robot, and establishes the state space model to predict the angle, position and attitude of each joint in motion, which provides the basis for the design of the trajectory tracking controller. Use the model predictive control algorithm to predict the tracking trajectory of the robot's foot, which further improves the tracking accuracy during the motion, and reduces or even eliminates the impact of the time lag on the control system.

II. QUADRUPED ROBOT MODEL

The research object of this paper is Scalf-II, a hydraulic quadruped platform of Shandong University [13]. The physical prototype is shown in Fig. 1. The leg of the imitation mammal quadruped robot Scalf-II is configured as a front elbow and a back knee, and a robot model is constructed according to it.



Fig. 1 The quadruped robot Scalf-II

A. Simulation Model

Establish a simplified structural model of the quadruped bionic robot in the simulation software Webots, as shown in Fig. 2. It can be seen that the quadruped robot is mainly composed of a trunk and four legs, each leg consisting of two connecting rods, one rolling hip joint, one pitching hip joint and one pitching knee joint.

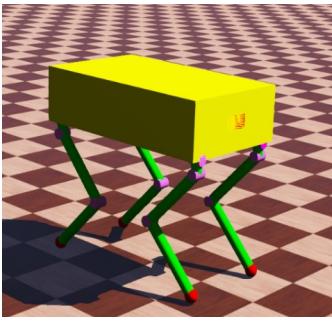


Fig. 2 Quadruped model built in Webots.

Each leg is connected to the torso through the rolling hip joint, so each leg has 3 degrees of freedom, including two pitch degrees of freedom and one roll degrees of freedom. The pitch joint is responsible for the inner and outer swing of the

leg, and the roll joint is responsible for the front and rear swing of the leg, and finally realizes the overall movement of the robot.

B. Dynamics model

Fig.3 shows that the structural parameters of the four legs of the quadruped robot are exactly the same, so one of the legs is taken as the research object to establish a dynamic model.

In this paper, the right front leg (Right Front, RF) of the robot is taken as an example to establish a coordinate system and study it. Define the torso centroid coordinate system as $\{O\}$, and establish the coordinate system for each joint of the RF leg according to the D-H method.

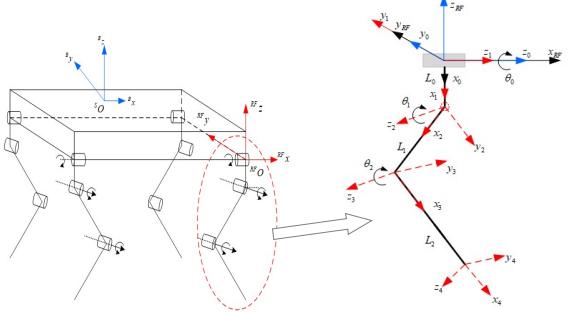


Fig.3 The schematic diagram of the quadruped robot

The D-H parameters of the RF leg can be obtained according to Fig. 3, as shown in Table.1.

TABLE I

Coordinate system number i	α_{i-1}	a_{i-1}	θ_i	d_i
1	0	0	θ_1	0
2	L_1	$\pi/2$	θ_2	0
3	L_2	0	θ_3	0
4	L_3	0	0	0

α_{i-1} is the angle at which z_{i-1} to z_i rotates around x_{i-1} , a_{i-1} is the distance measured by z_{i-1} to z_i along x_{i-1} , which is the length of the link $i-1$; θ_i represents the angle of rotation from x_{i-1} to x_i around z_i ; d_i represents the distance measured from x_{i-1} to x_i along z_i .

The dynamic model of the RF leg can be derived using the Newton-Eulerian iterative dynamics equation. The derivation process is as follows:

- 1) From link 1 to link 3, calculate the speed and acceleration of link outward iteratively

$$\begin{aligned} {}^i w_i &= {}^{i-1} R {}^{i-1} w_{i-1} + \dot{\theta}_i {}^i \hat{Z}_i \\ {}^i \dot{w}_i &= {}^{i-1} R {}^{i-1} \dot{w}_{i-1} + {}^{i-1} R {}^{i-1} w_{i-1} \times \dot{\theta}_i {}^i \hat{Z}_i + \ddot{\theta}_i {}^i \hat{Z}_i \\ {}^i \dot{v}_i &= {}^{i-1} R \left({}^{i-1} \dot{v}_{i-1} \times {}^{i-1} P_i + {}^{i-1} w_{i-1} \times \left({}^{i-1} w_{i-1} \times {}^{i-1} P_i \right) + {}^{i-1} \dot{v}_{i-1} \right) \quad (1) \\ {}^i \dot{v}_{C_i} &= {}^i \dot{w}_i \times {}^i P_{C_i} + {}^i w_i \times \left({}^i w_i \times {}^i P_{C_i} \right) + {}^i \dot{v}_i \end{aligned}$$

Where, ${}^i w_i$ is the angular velocity of rod i with respect to the coordinate system $\{i\}$, ${}^i \dot{w}_i$ is the angular acceleration of the rod i relative to the

coordinate system $\{i\}$, $\dot{\mathbf{v}}_i$ is the linear acceleration of rod i with respect to the coordinate system $\{i\}$, $\ddot{\mathbf{v}}_{Ci}$ is the linear acceleration of the centroid of rod i relative to the coordinate system $\{i\}$, $\mathbf{i}^{-1}\mathbf{R}$ is the position vector of rod i with respect to the coordinate system $\{i-1\}$, and $\mathbf{i}^{-1}\mathbf{P}_C$ is the position vector of the centroid of rod i with respect to the coordinate system $\{i\}$. $\mathbf{i}^{-1}\mathbf{R}$ is the rotation matrix of the link coordinate system $\{i\}$ with respect to $\{i-1\}$, $\mathbf{i}^{-1}\mathbf{R} = \mathbf{i}^{-1}\mathbf{R}^T$.

$$\begin{aligned}\mathbf{i}^{-1}\mathbf{R} &= \begin{bmatrix} 0 & \sin \theta_1 & -\cos \theta_1 \\ 0 & \cos \theta_1 & \sin \theta_1 \\ 1 & 0 & 1 \end{bmatrix} \\ \mathbf{i}^{-1}\mathbf{R} &= \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 \\ -\sin \theta_1 & 0 & \cos \theta_2 \\ 0 & -1 & 0 \end{bmatrix} \\ \mathbf{i}^{-2}\mathbf{R} &= \begin{bmatrix} \cos \theta_3 & \sin \theta_3 & 0 \\ -\sin \theta_3 & \cos \theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{i}^{-3}\mathbf{R} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\end{aligned}\quad (2)$$

- 2) The inertial force $\mathbf{i}^i\mathbf{F}$ and the torque $\mathbf{i}^i\mathbf{N}$ acting on the centroid of the connecting rod are respectively determined by Newton's formula and Euler's formula.

$$\begin{aligned}\mathbf{i}^i\mathbf{F} &= M_i \mathbf{i}^i\mathbf{v}_{C_i} \\ \mathbf{i}^i\mathbf{N} &= I_{C_i} \mathbf{i}^i\dot{\mathbf{w}}_i + \mathbf{i}^i\mathbf{w}_i \times I_{C_i} \mathbf{w}_i\end{aligned}\quad (3)$$

Where M_i is the mass of the connecting rod i , and I_{C_i} is the inertia tensor of the centroid of the connecting rod i .

- 3) From the connecting rod 3 to the connecting rod 1, iteratively and inwardly calculates the interaction force and moment between the connecting rods and the joint torque.

$$\begin{aligned}\mathbf{i}^{i-1}\mathbf{f}_{i-1} &= \mathbf{i}^{-1}\mathbf{R}^i \mathbf{f}_i + \mathbf{i}^{-1}\mathbf{F}_{i-1} \\ \mathbf{i}^{i-1}\mathbf{n}_{i-1} &= \mathbf{i}^{-1}\mathbf{N}_{i-1} + \mathbf{i}^{-1}\mathbf{R}^i \mathbf{n}_i + \mathbf{i}^{-1}\mathbf{P}_{C_{i-1}} \times \mathbf{i}^{-1}\mathbf{F}_{i-1} + \mathbf{i}^{-1}\mathbf{P}_i \times \mathbf{i}^{-1}\mathbf{R}^i \mathbf{f}_i \\ \tau_{i-1} &= \mathbf{i}^{i-1}\mathbf{n}_{i-1}^T \hat{\mathbf{Z}}_{i-1}\end{aligned}\quad (4)$$

$\mathbf{i}^i\mathbf{f}$ is the force acting on the rod i by the link $i+1$, and $\mathbf{i}^i\mathbf{n}$ is the torque acting on the rod i by the link $i+1$, τ is the driving torque vector of joint i .

The results of τ can be organized into the general structural formula of the closed form of the dynamic equation

$$\boldsymbol{\tau} = M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + V(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + G(\boldsymbol{\theta}) \quad (5)$$

$\boldsymbol{\theta}$ is joint position vector, $\dot{\boldsymbol{\theta}}$ is joint speed vector, $\ddot{\boldsymbol{\theta}}$ is joint acceleration vector. $M(\boldsymbol{\theta})$ is a function of the joint angle θ , called the inertia matrix of the robot. $V(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is the centrifugal force and the Coriolis force vector of the robot;

$G(\boldsymbol{\theta})$ is the component of the gravity of the robot, which is related to the shape and position of the robot. $M(\boldsymbol{\theta})$ is a 3×3 positive definite symmetric matrix, both $V(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ and $G(\boldsymbol{\theta})$ are 3×1 order matrices.

The angular acceleration can be obtained by (5):

$$\ddot{\boldsymbol{\theta}} = -M(\boldsymbol{\theta})^{-1}V(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + M(\boldsymbol{\theta})^{-1}\boldsymbol{\tau} - M(\boldsymbol{\theta})^{-1}G(\boldsymbol{\theta}) \quad (6)$$

When the legs of the quadruped robot are in the swing phase, the joint forces of each joint can be obtained according to (5). When the robot's leg is in the supporting phase, the foot of the supporting leg is also supported by the ground (denoted as F). According to the Jacobian matrix [14], we can get the joint force τ_{ex} corresponding to the external force F at the sole of the foot as

$$\tau_{ex} = J^T(\boldsymbol{\theta}) \cdot F \quad (7)$$

We can get the driving force of each joint of the quadruped robot leg during the movement, such as:

$$\boldsymbol{\tau}_T = \boldsymbol{\tau} + \tau_{ex} = M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + V(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + G(\boldsymbol{\theta}) + \delta J^T(\boldsymbol{\theta}) \cdot F \quad (8)$$

When the leg is in the swing phase, $\delta = 0$, and when the leg is in the support phase, $\delta = 1$.

C. State Space Model

The dynamic equation is a strongly coupled, nonlinear time-varying equation. If the model predictive control algorithm based on the linear superposition principle is directly controlled, a large number of computational processes will be generated, which will make the optimization process more complicated. Therefore, intercell linearization is adopted and then model predictive control is applied. In order to ensure the correctness of the whole control process, the control model needs to be optimized in real time to ensure that the algorithm is based on an effective model. The system status vector is

$$\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T = [\theta_1 \ \theta_2 \ \theta_3 \ \dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3]^T \quad (9)$$

Then can express the state equation of the system as

$$\left\{ \begin{array}{l} [\dot{x}_1 \ \dot{x}_2 \ \dot{x}_3 \ \dot{x}_4 \ \dot{x}_5 \ \dot{x}_6] = [\dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3 \ \ddot{\theta}_1 \ \ddot{\theta}_2 \ \ddot{\theta}_3]^T \\ y = H\mathbf{x} \quad , \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array} \right. \quad (10)$$

The sampling period is T , and the components of the state variable are discretized by Taylor expansion. The discretized state variables are

$$x(k) = \begin{bmatrix} x_1 + T\dot{x}_1 + \frac{T^2}{2}\ddot{x}_1 \\ x_2 + T\dot{x}_2 + \frac{T^2}{2}\ddot{x}_2 \\ x_3 + T\dot{x}_3 + \frac{T^2}{2}\ddot{x}_3 \\ x_4 + T\dot{x}_4 \\ x_5 + T\dot{x}_5 \\ x_6 + T\dot{x}_6 \end{bmatrix} \quad (11)$$

The discretized state space model of the system can be written:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + O \\ y(k) = Hx(k) \end{cases}, \quad k=1, \dots, N \quad (12)$$

From (11) we can get:

$$A = \begin{bmatrix} I_3 & TI_3 \\ O & I_3 \end{bmatrix}, B = \begin{bmatrix} \frac{T^2}{2} M(\theta)^{-1} \\ TM(\theta)^{-1} \end{bmatrix},$$

$$O = \begin{bmatrix} -\frac{T^2}{2} M(\theta)^{-1} (V(\theta, \dot{\theta}) + G(\theta)) \\ -TM(\theta)^{-1} (V(\theta, \dot{\theta}) + G(\theta)) \end{bmatrix}$$

The state variable of the model at the k th moment is $x(k)$, that is, the angle and angular velocity of the three joints of the RF leg, the control variable is the joint driving force $u(k)$, and the output variable $y(k)$ is the angle of the three joints. A , B , and O are constant values in the tracking process for each desired point, that is, the system is a time-invariant system.

III. MODEL PREDICTIVE CONTROL

A. Predictive Model

The state space model does not yet have the predictive function. The predictive model should predict the future value of the process output based on the current and future control inputs of the system and historical past values in the control process. The prediction model can utilize a nonparametric model such as an impulse response model or a step response model of the system, or a parameter model such as a transfer function or a state space model. In the single-leg control system, this paper will use the state space model of single-leg motion to establish a predictive model. Equation (13) give a typical linear discrete state space equation:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + O, \quad x(0) = x \\ y(k) = Hx(k) \\ u(k) = u(k-1) + \Delta u(k), \quad u(0) = u \end{cases}, \quad k=1, \dots, N \quad (13)$$

$y(k)$ is the predicted output at the k th sampling moment in the prediction time domain. Based on this, we can establish a prediction model of system incremental control as

$$\begin{aligned} y_p(k) &= y(k+1) = Hx(k+1) \\ &= H(A_p X(k) + B_p U(k) + O_p) = y_{p0}(k) + L \Delta u(k) \end{aligned} \quad (14)$$

Definition N is the prediction time domain and M is the control time domain,

$$y_p(k) = \begin{bmatrix} y(k+1|k) \\ \vdots \\ y(k+N|k) \end{bmatrix}, A_p = \begin{bmatrix} HA \\ HA^2 \\ \vdots \\ HA^N \end{bmatrix}, X(k) = \begin{bmatrix} x(k|k) \\ \vdots \\ x(k+N-1|k) \end{bmatrix},$$

$$B_p = \begin{bmatrix} HB \\ HAB \\ \vdots \\ HA^{N-1}B & HA^{N-2}B & \cdots & HB \end{bmatrix}, U(k) = \begin{bmatrix} \tau(k|k) \\ \vdots \\ \tau(k+N-1|k) \end{bmatrix},$$

$$O_p = \begin{bmatrix} HO \\ H(AO+O) \\ \vdots \\ H(A^{N-1}O+A^{N-2}O+O) \end{bmatrix}, y_{p0}(k) = \begin{bmatrix} y_0(k+1|k) \\ \vdots \\ y_0(k+N|k) \end{bmatrix},$$

$$L = \begin{bmatrix} I_{3 \times 3} & 0 & \cdots & 0 \\ I_{3 \times 3} & I_{3 \times 3} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_{3 \times 3} & I_{3 \times 3} & \cdots & I_{3 \times 3} \end{bmatrix}_{3N \times 3M}, \Delta u(k) = \begin{bmatrix} \Delta u(k|k) \\ \vdots \\ \Delta u(k+M-1|k) \end{bmatrix}$$

B. Rolling optimization

In order to adapt the system to the latest changes and to suppress the disturbance to ensure the accuracy of the trajectory tracking, it is necessary to reduce the deviation (15) between the controlled output value and the set output value as much as possible.

$$e_y(k) = y_p(k) - y_{ref}(k) \quad (15)$$

If B_p is a square matrix and is reversible, the optimal control input value can be solved by solving the equation. In fact, since the prediction time domain and the control time domain are selected, there is no guarantee that the B_p is a square matrix. Then the minimum value of the performance indicator (16) is required to make the controlled output as close as possible to the set value.

$$J_p = \|y_p(t) - y_{ref}(t)\|_Q^2 + \|\Delta u(t)\|_R^2 \quad (16)$$

Where, $Q = \text{diag}[Q_1, Q_2, \dots, Q_N]$, $R = \text{diag}[R_1, R_2, \dots, R_M]$ are the set error weight matrix and the control weight matrix respectively. y_{ref} is the set value of the desired output trajectory. We can get the optimal input (18) by the extreme value necessary condition (17).

$$\frac{dJ(k)}{d\Delta u(k)} = 0 \quad (17)$$

$$\Delta u(k) = (L^T Q L + R)^{-1} L^T Q (y_{ref} - y_{p0}) \quad (18)$$

When the next sampling moment begins, the optimization period will move forward at the same time, and the performance index J is recalculated, which is the essence of rolling optimization. In the prediction model, M future control increments predicted at time k are defined, but in fact, at this moment, the M control quantities only implement the first control value at a time, and the next time will be reseek. A new M control amount is generated, and only the first control value is implemented, and so on.

Therefore, the amount of control actually input into the system each time is

$$\Delta U(k) = C \Delta u(k), C = [1 \ 0 \ \cdots \ 0]^T \quad (19)$$

C. Feedback correction

Due to many uncertain factors such as environmental disturbance and model mismatch, the predicted input value in the system control process has a certain deviation from the actual situation. Therefore, the feedback correction method is adopted to control the actual output $y(k+1)$ in the controlled process. Compare with the model prediction output $y_p(k+1|k)$ to get the prediction error

$$e(k+1) = y(k+1) - y_p(k+1|k) \quad (20)$$

Corrected output position prediction value:

$$y_c(k+1) = y_p(k) + he(k+1) \quad (21)$$

$$y_c(k+1) = \begin{bmatrix} y_c(k+1|k+1) \\ \vdots \\ y_c(k+N|k+1) \end{bmatrix}, h = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix},$$

h is an N-dimensional correction vector.

IV. EXPERIMENTAL TEST

Built a three-degree-of-freedom one-legged model in three-dimensional space using the Robotics Toolbox robotic toolbox based on the MATLAB platform [15]. As shown in Fig. 4, simulate the single leg trajectory tracking based on MPC.

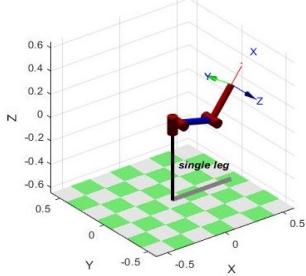


Fig. 4 One-legged model built by Robotics Toolbox.

Assuming that there is no uncertainty in the ideal situation, the single-leg model moves in the xyz three-dimensional space, and the motion trajectories are set as (22).

$$x = 0.5 \sin(t \cdot 2\pi / 1000)$$

$$y = 1 \quad (22)$$

$$z = -1$$

Set the single leg to track the desired trajectory from the coordinate origin $(0, 0, 0)$, set the simulation point to 1000, and select 50 for both the control time domain and the predicted time domain of the model prediction. The experimental results of model predictive control are shown in Fig.5. It can be seen from Fig. 5 that the single leg based on the model predictive control can quickly track the desired target trajectory, and the variation range is relatively stable.

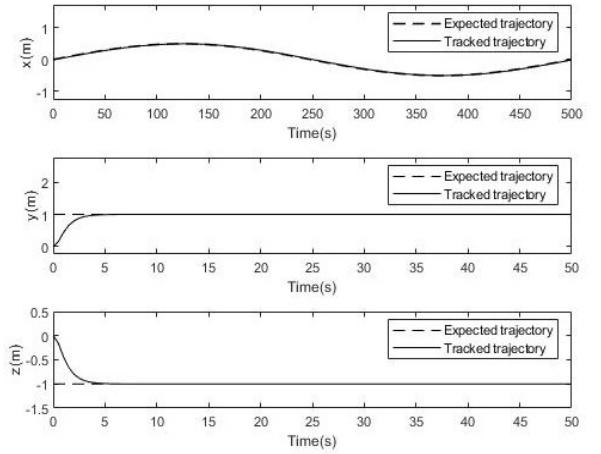


Fig. 5 Expected trajectory and MPC tracking trajectory

Fig. 6 shows the error caused by the MPC tracking the desired trajectory. It can be seen that the algorithm can quickly track the desired trajectory. A slight microwave motion occurs with the change of the slope of the curve in the x direction, but the floating error is within ± 0.02 mm. It can be seen that the accuracy of the MPC tracking desired trajectory is better.

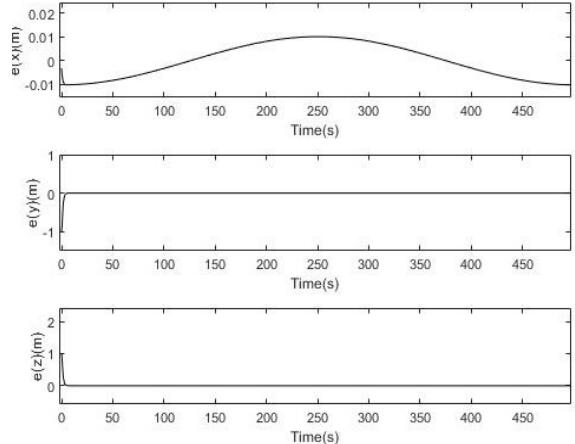


Fig. 6 MPC tracking trajectory error

Set the desired trajectory to a curve that changes continuously with time and set 1000 simulation points to compare the accuracy and speed of the tracking track between the MPC control algorithm and the traditional PID control algorithm. Fig. 7 is a comparison of the PID tracking trajectory and the MPC tracking trajectory. By comparison, the model predictive control algorithm can achieve the tracking effect on the desired trajectory more quickly and more stably than the traditional PID control algorithm.

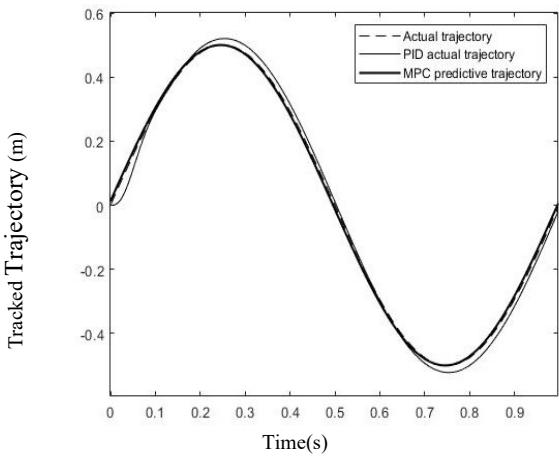


Fig. 7 Compare trajectory tracking ability between MPC and PID

Fig. 8 shows the tracking error of the MPC and PID for the desired trajectory. The change in the slope of the trajectory curve causes the tracking error to exhibit a small fluctuation, and both errors are kept within ± 0.06 mm. However, it is obvious that the error fluctuation of the MPC tracking trajectory is small, and the error is reduced faster.

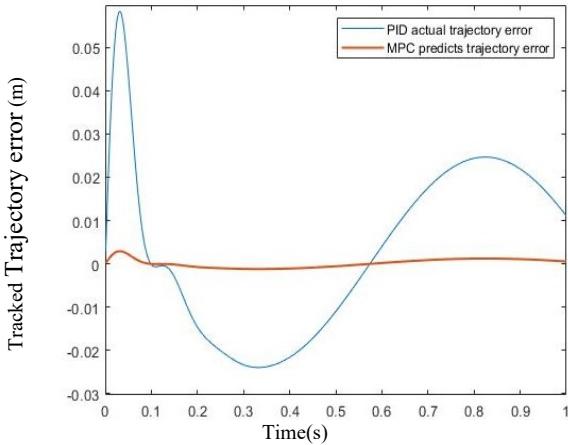


Fig. 8 Comparison of MPC and PID tracking trajectory errors

V. CONCLUSIONS AND FUTURE WORK

For the problem of the quadruped robot tracks the desired target trajectory with high precision and real-time in complex and harsh environment, this paper proposes a method based on model predictive control to trajectory planning for the quadruped robot. Through the dynamic analysis of the single leg of the robot, the prediction model is established and the optimal input value that satisfies the expectation is solved. The final simulation results show that the leg control method based on model predictive control can make the foot end of the robot have a higher level of real-time and accuracy of trajectory tracking. Therefore, the quadruped robot trajectory tracking algorithm based on model predictive control can realize the real-time effective tracking of the quadruped robot trajectory and has good operational accuracy. In the future work, we will introduce interference due to uncertainty, apply it to the quadruped robot, and conduct corresponding physical experiments.

ACKNOWLEDGMENT

The work reported in this paper is supported by Key Research and Development Program of Shandong Province, China (No. 2016ZDJS02A07), Key Research and Development Program of Shandong Province, China (2017CXGC0915) and Key Research and Development project of Shandong province, China, under Award No.2017CXGC0903-2.

REFERENCES

- [1] Playter, R., M. Buehler, and M. Raibert. "BigDog." Society of Photo-optical Instrumentation Engineers Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 2006.
- [2] Wooden, David, et al. "Autonomous Navigation for BigDog." *Robotics and Automation (ICRA), 2010 IEEE International Conference on IEEE*, 2010.
- [3] Hutter, M., et al. "High Compliant Series Elastic Actuation for the Robotic Leg ScarfETH." *International Conference on Climbing & Walking Robots* 2011.
- [4] Hutter, M., et al. "ANYmal - toward legged robots for harsh environments." *Advanced Robotics* (2017):1-14.
- [5] Seok, Sangok, et al. "Design Principles for Energy-Efficient Legged Locomotion and Implementation on the MIT Cheetah Robot." *IEEE/ASME Transactions on Mechatronics* 20.3(2015):1117-1129.
- [6] Park, Hae Won, and S. Kim. "Quadrupedal galloping control for a wide range of speed via vertical impulse scaling." *Bioinspiration & Biomimetics* 10.2(2015):025003.
- [7] Wensing, Patrick M., S. Kim, and J. J. E. Slotine. "Linear Matrix Inequalities for Physically-Consistent Inertial Parameter Identification: A Statistical Perspective on the Mass Distribution." *IEEE Robotics and Automation Letters* (2017):1-1.
- [8] Zhang, Shuaishuai, et al. "Generation of a continuous free gait for quadruped robot over rough terrains." *Advanced Robotics* (2019):1-16.
- [9] Wang, Pengfei, and L. Sun. "The Stability Analysis for Quadruped Bionic Robot." *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2006, October 9-15, 2006, Beijing, China IEEE*, 2006.
- [10] Zeng Xiangyu. Research on motion control of complex terrain under quadrilateral bionic robot based on CPG. Diss. Beijing Jiaotong University, 2011.
- [11] Zhang, Guoteng, et al. "Torso motion control and toe trajectory generation of a trotting quadruped robot based on virtual model control." *Advanced Robotics* (2015):1-14.
- [12] Bledt, Gerardo, P. M. Wensing, and S. Kim. "[IEEE 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) - Vancouver, BC (2017.9.24-2017.9.28)] 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) - Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the MIT cheetah." (2017):4102-4109.
- [13] Rong, Xuewen, et al. "Design and simulation for a hydraulic actuated quadruped robot." *Journal of Mechanical Science and Technology* 26.4(2012):1171-1177.
- [14] Yaxian, Xin, et al. "A comparative study of four Jacobian matrix derivation methods for quadruped robot." *Control Conference IEEE*, 2015.
- [15] Corke, Peter. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Publishing Company, Incorporated, 2011.