PAPER • OPEN ACCESS

Motion Planning for a Bounding Quadruped Robot Using iLQG Based MPC

To cite this article: Yezhong Huang *et al* 2021 *J. Phys.: Conf. Ser.* **1905** 012016

Motion Planning for a Bounding Quadruped Robot Using iLQG Based MPC

Yezhong Huang[1], Qing Wei[1], Hongxu Ma[1] and Honglei An[1].

[1] National University of Defense Technology, Changsha, China.

hyz1022@yeah.com

Abstract. When tackling motion control problem of the quadruped bionic robot with traditional control method, it is more and more difficult to improve the control effect due to the uncertainty caused by the factors such as model mismatch, distortion, and interference. Therefore, the demand for utilizing advanced control method to realize the motion control for legged bionic robot is growing. Among various advanced control methods, the model prediction control (MPC) has low requirements on model accuracy, and the system has good robustness and stability, which provides a good prospect for solving the balance problem of dynamic robots. In this paper model predictive control is used to realize the bounding gait motion control for a quadruped bionic robot. The MPC planner automatically generates the foot end position and contact force trajectory of the robot, so that the robot can track the desired motion state. However, as the model of the quadruped bionic robot is nonlinear, the generated MPC algorithm may be difficult to meet the real-time requirements. In order to solve this problem, this paper uses the iLQG optimization algorithm in the design of a model prediction controller and adds heuristic-based reference input. The comparison between the simulation and the classical SQP algorithm shows that the MPC based on the iLQG optimization algorithm has good convergence and tracking effect for the quadruped robot running, and has fast real-time performance.

1. Introduction

Terrestrial robot is a kind of flexible walking system that can perform tasks that is too dangerous for human beings. Terrestrial robots mainly include wheeled robots, crawler robots, and legged robots. Among them, legged robots have the strongest environmental adaptability and have many outstanding advantages compared with wheeled and crawler robots. It can span the rugged and complex terrain and have potential applications in an unstructured environment.

According to the number of legs, legged robots are usually divided into single-foot robots, biped robots, quadruped robots, and multi-legged robots. Among them, the four-legged walking robot has greater advantages in terms of balance between bearing capacity and stability. Raibert uses the Virtual Leg model and the "height, attitude, speed" three-point control strategy to achieve stable motion of various gaits such as the quadruped robots Trot, Pace and Bound [1], [2]. Italy's quadruped bionic robot HyQ and Swiss single-legged prototype Scarlett also use virtual model control methods to control body posture or single-leg jump movement [3-6]. The Buehler team at McGill University proposed a bound gait control method based on joint driving force control based on the analysis of the fixed-point of the planar three-bar simplified model based on the SLIP model and applied to the Scout and PAW quadruped robots. Motion control [7-10]. Duane W. Marhefka [11], Department of Mechanical Engineering, Stanford University, developed a four-legged jumping robot KOLT that mimics the biological characteristics of goat jumping. It uses the Levenberg-Marquardt online learning method and the Adaptive Adaptive Fuzzy Control strategy. The intelligent control method realizes the single-leg control of KOLT, and the control effect achieves the single-leg effect of the dynamic jumping of the four-legged robot.

When quadruped robot running with high-speed, strong nonlinearity, non-holonomic constraints, extremely short foot-ground action time and large instantaneous driving force appears in its dynamics. Traditional control methods often have problems such as overly complex models and difficult solutions. Model predictive control is a feedback control strategy widely used and developed in recent years. The advantage of MPC is that the model of the control object does not need to be too complicated, which simplifies the modeling process and the algorithm is also very robust.

The basic idea of MPC is to solve a finite-time open-loop optimization problem online based on the obtained measurement at each sampling time and apply the first element of the obtained control sequence to the controlled object. At the next sampling moment, the above process is repeated, and the new measured value is used as the initial condition for predicting the future dynamics of the system at this time, and the optimization problem is refreshed and resolved. The application of MPC in robot bouncing [12], biped robot [13-15] and quadruped robot walking [16] prove that the algorithm has a very good effect on the planning and control of hybrid dynamic system motion.

However, the model of the quadruped robot is a nonlinear dynamic equation. For the constrained nonlinear system, the exact analytical solution cannot be obtained by directly solving the HJB equation of the system. Therefore, a numerical solution method is needed. In [17], SQP was applied to NMPC to solve the control amount and applied to the distillation column process to obtain better control effects. Literature [18] combines the Continuation Method with the Generalized Minimum Residual Method (GMRES) to propose a numerical method for quickly solving nonlinear predictive control. These methods have the disadvantages of large computational complexity and low real-time performance. Intelligent algorithms such as genetic algorithms and particle swarm optimization algorithms have also achieved good results in solving such problems. However, due to the stochastic optimization solution, the convergence rate is unstable, which is not suitable for a robot system that strictly requires gait to travel.

The iLQG [19] optimization algorithm proposed by Todorov is an improvement of the real-time performance of the differential dynamic programming (DDP) algorithm. Unlike the DDP algorithm, the first derivative of the dynamic equation is used instead of the second derivative in the calculation process. Therefore, the iLQG algorithm achieves faster optimization while sacrificing partial accuracy. The iLQG-based MPC has been applied in the control of the pinball and has achieved good results [20].

The content of this paper is to plan the footstep position and foot forces of the robot according to its planned movement speed and gait on the basis of the planned path and movement speed of the robot. The main contribution of this paper is to apply the iLQG optimization algorithm with high real-time performance to the running movement of four-legged robots. A heuristic MPC controller is used to plan and control the landing point and foot force of the bound gait of a quadruped robot.

The rest of this paper is organized as follows. The second part details the overall structure of the MPC controller. Includes simplified predictive models that preserve the main dynamics of the system, objective functions and constraints, and heuristic design. The third section introduces the simple principle and algorithm flow of the iLQG algorithm. The results of this method are given in the fourth section. Its optimization results, convergence, trajectory tracking and real-time performance are introduced in detail and compared with the traditional SQP algorithm. Finally, Section V provides a brief summary discussion.

2. The MPC controller

Model predictive control is achieved by solving an optimization that considers the modeled dynamics of the system over a receding prediction horizon and returns the set of control inputs that minimize a cost function for the desired behavior at the predicted timesteps [21]. The MPC controller mainly includes a prediction model, cost function and constraint, and optimization algorithm (figure 1).

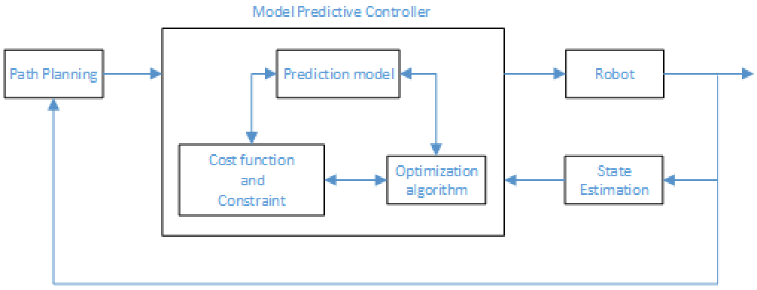


Figure1. System control block diagram

2.1. Prediction model

We can model the dynamics of the full system for the four-legged robot which of the author's laboratory (as shown in figure 2). However, because the robot is designed such that it has light legs with low inertia and only about 10% of the total mass, the weight of the legs is neglected so that its dynamics can be simplified to only include the base body while treating the legs as massless rigid bodies. This approach has been proven to be valid for this robot in [22], which uses the same single rigid body base assumption for bounding and autonomous jumping. This simplification helps to decrease the solve times of direct collocation optimization problems for MPC while retaining the main dynamic effects important to locomotion.

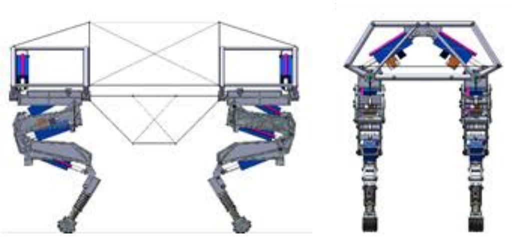


Figure2. Four-legged legged robotic platform

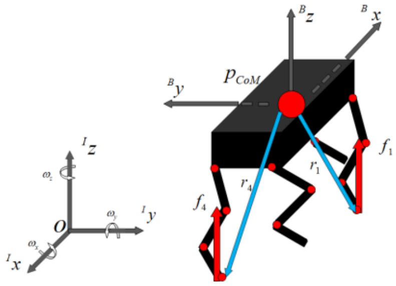


Figure3. Coordinate system and parameter definition of 3d model.

Figure 3 depicts the dynamic coordinate system definitions as well as the system inputs. We define an inertial coordinate system denoted by I as the leading superscript. A body-fixed coordinate system is attached to the robot's CoM and denoted by B as the leading superscript.

The forces $\boldsymbol{f_i}$ provide the force under each foot, while the vectors $\boldsymbol{r_i}$ specify the position of each foot relative to the body CoM.

The state of the robot is defined to be updated as

$$x = \begin{pmatrix} P_c \\ \Theta \\ \dot{P}c \\ {}^{B}\omega \end{pmatrix} \tag{1}$$

Where $\boldsymbol{P_c}$ provides the position of the body CoM, $\boldsymbol{\Theta} = [\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}]$ is the Euler angle, with an earth-fixed roll ($\boldsymbol{\theta}$), pitch ($\boldsymbol{\phi}$), yaw ($\boldsymbol{\psi}$), $\dot{\boldsymbol{P}}_{\boldsymbol{c}}$ is the velocity of the body to coordinate system $\boldsymbol{I}$, and ${}^{B}\omega$ is the angular velocity of the body.

Forces $\boldsymbol{f_i}$ acting at foot locations $\boldsymbol{r_i}$ proved an input to the system. Foot positions relative to the inertial frame origin are denoted as $\boldsymbol{P_j} = \boldsymbol{P_c} + \boldsymbol{r_j}$. $\boldsymbol{P_c}$ is the coordinates of the COM of the body in the inertial coordinate system. The system input vector $\boldsymbol{u} = [\boldsymbol{r_1} \quad \boldsymbol{f_1} \quad \dots \quad \boldsymbol{r_4} \quad \boldsymbol{f_4}]$ is the forces and relative foot position for each of the legs.

The forcing screw for the body can be expressed as the vector of net force and net moment:

$$\left\{ \begin{matrix} f \\ {}^{B}\tau \end{matrix} \right\} = h(u) = \sum_{j=1}^{4} \begin{bmatrix} I \\ {}^{I}R_B^T [r_j]_\times \end{bmatrix} f_j \tag{2}$$

where is a skew-symmetric matrix such that, for any vector. Thus, the dynamic of the body described by equations of motion:

$$\dot{x} = f\left(x,u \right) = \left\{ \begin{matrix} \dot{P}_c \\ {}^{I}R_B [{}^{B}\omega]_\times \\ \frac{1}{m} f - g \\ {}^{B}\bar{I}^{-1}\left({}^{B}\tau - {}^{B}\omega \times {}^{B}\bar{I}\,{}^{B}\omega \right) \end{matrix} \right\} \tag{3}$$

where $\mathbf{g} \in R^3$ is the gravitational acceleration and ${}^{B}\bar{I}$ is the inertial tensor in body coordinates.

Under well-controlled locomotion, the pitch and roll of the base body are bounded above and below by a small value, the equations of motion can be simplified, and the transformation matrix can be approximately noting that

$$\lim_{(\hat{\theta},\hat{\phi}) \to (0,0)} {}^{I}\hat{R}_B(\Theta) = R_z\left(\hat{\psi}\right) \tag{4}$$

where $R_z\left(\hat{\psi}\right)$ means a rotation of $\hat{\psi}$ radians about the ${}^{I}Z$ axis. And, ${}^{B}\omega \times {}^{B}\bar{I}\,{}^{B}\omega$ are neglected because they are small, then the approximate angular acceleration can noting as

$${}^{B}\hat{\omega} = {}^{B}\bar{I}^{-1B}\hat{\tau} \tag{5}$$

The Euler angular acceleration is approximated by neglecting roll Angle and Pitch Angle

$$\Theta \approx {}^{B}\omega \tag{6}$$

Through all of these approximations are applied to the dynamics of the system, the final simplified model is then described by

$$\dot{x} = \begin{pmatrix} 0 & 0 & E & 0 \\ 0 & 0 & 0 & E \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m}E & 0 \\ 0 & {}^{B}\bar{I}^{-1} \end{pmatrix} h(u) + \begin{pmatrix} \mathbf{g} \\ 0 \end{pmatrix} \tag{7}$$

Where $\boldsymbol{E} \in \boldsymbol{R}^3$ is the identity matrix and $\mathbf{g}$ is the gravitational acceleration.

Through discretization, the final discrete control model is：

$$x_{i+1} = A_i x_i + B_i h(u_i) + d_i \tag{8}$$

where $\boldsymbol{A_i}$ and $\boldsymbol{B_i}$ are both fixed matrices and $\boldsymbol{d_i}$ a fixed vector dependent on the discrete time step, Δt_i

$$A_i = \begin{bmatrix} E_6 & \Delta t_i E_6 \\ 0 & E_6 \end{bmatrix}, B_i = \begin{bmatrix} \frac{\Delta t_i^2}{2}\,{}^{B}I^{-1} \\ \Delta t_i\,{}^{B}I^{-1} \end{bmatrix}, d_i = \begin{bmatrix} \frac{\Delta t_i^2}{2} a_g \\ \Delta t_i a_g \end{bmatrix} \tag{9}$$

where, ${}^{B}I = \mathrm{diag}\left(mE_3, {}^{B}\bar{I}\right)$ and $\boldsymbol{a_g} = [\mathbf{g}^{\boldsymbol{T}}, 0_{3\times1}^T]^T$. This discrete dynamic equation has a simple form, this helps to decrease the solve times of direct collocation optimization problems for MPC.

2.2. The cost function and constraints

MPC is designed as a direct collocation formulation so that we can find suitable controls:

$$\begin{gathered}\min_x \sum_{i=0}^{T} l_i(\hat{x}_i,\hat{u}_i)+l_T(\hat{x}_T)\\ s.t\quad \hat{x}_{i+1}=f(\hat{x}_i,\hat{u}_i)\\ c_i(\hat{x}_i,\hat{u}_i)\leqslant 0\\ c_i(\hat{x}_i,\hat{x}_{i+1},\hat{u}_i,\hat{u}_{i+1})\leqslant 0\end{gathered}\tag{10}$$

2.2.1. Cost function: The MPC needs to plan the footstep positions and foot forces according to the planned trajectory of the COM so that the robot can follow the desired state while ensuring stability and specifying gait. In this paper, a heuristic [23] that satisfies the stability condition and the specified gait is used as a reference input. When there is a model mismatch, a rough road or an external impact, the result of the optimization is in a field of reference input, thereby ensuring its stability and basic gait.

At the same time, the planned footstep positions and foot forces of the robot can make the state of the robot follow the desired state.

Therefore, the control sequence $\{u_k\}$ that needs to be planned is the vector of the footstep positions and foot forces, and the trajectory sequence $\{x_k\}$ is the position of the CoM. Here, two items need to be optimized. First, the trajectory error $\tilde{x}=x_d-\hat{x}$ is as small as possible, where x_d is the expected trajectory. The other is to control the sequence error $\tilde{u}=u_{ref}-\hat{u}$ as small as possible, where u_{ref} is the programming sequence obtained by the previous heuristic.

So the cost function is:

$$\begin{gathered}l_i(\hat{x}_i,\hat{u}_i)=\tilde{x}_i^T Q_i\tilde{x}_i+\tilde{u}_i^T R_i\tilde{u}_i\\ l_T(\tilde{x}_T)=\tilde{x}_T^T Q_T\tilde{x}_T\end{gathered}\tag{11}$$

2.2.2. Constraints: For the duration of the prediction, constraints must be added to the optimization for the results to produce physically realizable foot placements and ground reaction forces.

In the process of optimization calculation, all state vectors must adhere to the simplified dynamic equation

$$\hat{x}_{i+1}=\hat{y}_i=A_i\hat{x}_i+B_i\hat{h}(\hat{x}_i,\hat{u}_i)+d_i\tag{12}$$

It is also required that this foothold on the ground should not be unreachable due to the limitation of leg length

$$\|r_{j,k}-p_{h,j}\|\leqslant l_{\max}\tag{13}$$

where, $l_{\max}$ is the maximum length of the leg.

In addition, the forces must also adhere to physical constraints during the movement process, and the force, normal to the ground plane must be positive

$$\hat{F}_f\cdot\nabla\hat{G}(\hat{p}_f)\geqslant 0\tag{14}$$

where $\nabla\hat{G}(\hat{p}_f)$ is the estimated ground surface equation at the foot location.In this paper, it is assumed that the ground is uniformly flat and the height, $\hat{z}_g(x,y)=0$ that means $\hat{G}(\hat{p}_f)=[0\quad 0\quad 1]^T$.

In the physical system, the point-to-surface contact model is constrained by the friction cone. When the ratio of the tangential friction force to the normal contact force of the contact surface is greater than a certain value, the foot will slip. To avoid this phenomenon, the tangential force on the contact surface should be limited within the scope of the friction cone. The force is limited to the range of the friction cone. When the foot end of the foot robot is in contact with the horizontal plane, the constraint on the horizontal foot end force is:

$$\begin{aligned}|f_x| &\leqslant \mu f_z \\ |f_y| &\leqslant \mu f_z\end{aligned} \tag{15}$$

where, μ is the coefficient of friction.

*2.2.3. Bound gait:*The bound gait is a gait in which the front legs or the back legs move simultaneously. The bound gait can generally be divided into four stages, namely the hind leg support period, the front leg support period and the two vacant periods(as shown in figure 4). In the process of MPC prediction, the range of prediction is too large, and the required data is large, and the amount of calculation is large, which causes the real-time control to be reduced. Conversely, too little data will affect the solution of the control input, and the effect of the control will be poor. In this paper, we choose the range of predictions to be a full gait.

Figure4. Bound gait

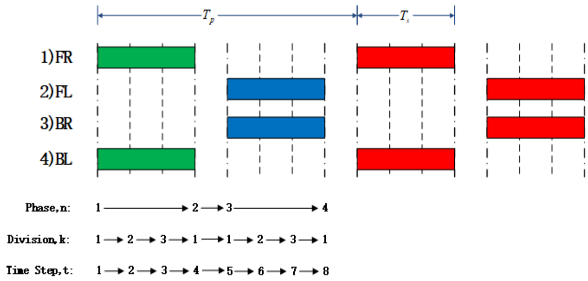


Figure5. The phase divisions and prediction ranges of the bound gait (green and blue).Only the first stage (green) is executed at a time.

The scope of each optimization prediction is $N=4$.Since the force exerted by the ground to the ontology is zero during the flight phases, the control input is also zero. Considering that the force in the stance phases is a constantly changing process, in order to better optimize the effect of the force, the stance phases are mainly divided into $K=3$ stages. Although the model dynamics predicted four stages, only the first prediction stage was optimally inputted in the simulation, as shown in the green part in the figure 5. When the optimal foot

position and ground reaction force of the first stage are determined, the controller will re-predict a set of new optimal controls for the next prediction layer.

*2.2.4. The heuristic:*The MPC optimization algorithm has no fixed form and can be arbitrarily selected. For many optimization algorithms, the initial value is very important and may affect the program's results, in order to make the results of calculation in the heuristic design target point within a certain field, and in order to enable the calculation to quickly reach the extreme points that meet the physical requirements, in this paper, the point of the heuristic design is selected as the initial value point of the MPC.

The determination of the footstep location must consider the stability of the motion and the speed of the centroid of the ontology. Here, referring to the acquisition of the capture point [24], the step size of each leg is designed as follows:

$$p_{ref,j} = P^T P\left(p_{c,0} + p_{h,j} + \frac{T_s}{2}\dot{p}_{c,d} + \sqrt{\frac{z_0}{g}}\left(\dot{p}_c - \dot{p}_{c,d}\right)\right) \tag{16}$$

where $P=\begin{bmatrix}1&0&0\\0&1&0\end{bmatrix}$ is a projection matrix to the XY plane, z_0 is the height of locomotion, $p_{h,j}$ provides the position of j relative to the CoM, $p_{c,0}$ is the initial position of the CoM at the start of the step, T_s is the associated stance time, and $\dot{p}_{c,d}$ is the expected velocity of the COM.

Thus, the footstep location is

$$r_{ref,j,i}=p_{ref,j}-p_{c,i} \tag{17}$$

Reference forces are calculated equivalently for each foot using the principle of vertical impulse scaling from [25], [26]

$$f_{ref}=\frac{mT_p}{4T_s}\begin{bmatrix}\cos(\psi_d)\dot{\psi}_d\|\dot{p}_{c,d}\|\\ \sin(\psi_d)\dot{\psi}_d\|\dot{p}_{c,d}\|\\ \|g\|\end{bmatrix} \tag{18}$$

where , T_p is the overall gait period. The first two provide the centripetal force of the deflection, and the last one counteracts the gravity impulse.

3. iLQG algorithm

The iLQG algorithm is an improvement based on differential dynamic programming. The main idea is to find a random control sequence $\{\bar{u}(k)\}$, and then do a calculation according to the control to get the trajectory $\{\bar{x}(k)\}$. In the vicinity of this trajectory, linearize the dynamic characteristics of the system, quadrate the loss function, and consider how to find perturbations in $\{\bar{u}(k)\}$ and $\{\bar{x}(k)\}$ around the system $\delta u=u-\bar{u}$ and $\delta x=x-\bar{x}$ make the new control better than before. The optimal control sequence is obtained by iteratively updating until convergence. The most important feature of this algorithm is its real-time performance. Because the first derivative of the motion equation is used instead of the second derivative, it has a higher prediction speed. Todorov's team has used this algorithm to achieve good results in pinball control.

4. Simulation

This section mainly evaluates the use of the iLQG algorithm in the bound gait. First, predict the planning results of the front and back legs in one step, and then give the process data of the algorithm to illustrate the convergence of the algorithm. The SQP algorithm used in the comparison simulation is the MATLAB optimization extremum function fmincon (). The parameters used in this process are show in Table 1.

Table 1. Weight parameters of Q and R.

Parameter	Weight Vector			Units
Q_p	0	0	500000	$\left(\frac{1}{m}\right)^2$
$Q_{\dot{\Theta}}$	5	10	25	$\left(\frac{1}{rad}\right)^2$
$Q_{\dot{p}}$	500000	500000	500000	$\left(\frac{s}{m}\right)^2$
$Q_{^B\dot{w}}$	2.5	2.5	300	$\left(\frac{s}{rad}\right)^2$
$R_{\dot{r}}$	35000	35000	35000	$\left(\frac{1}{m}\right)^2$
$R_{\dot{f}}$	250	250	1	$\left(\frac{1}{N}\right)^2$

4.1. Results of the one-step prediction

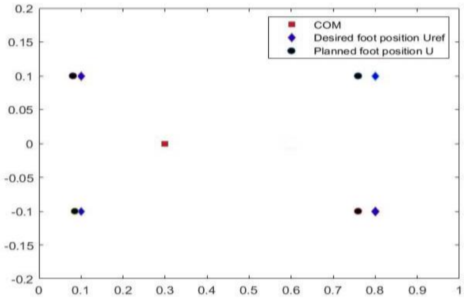


Figure6. The graph use iLQG to make a one-step prediction

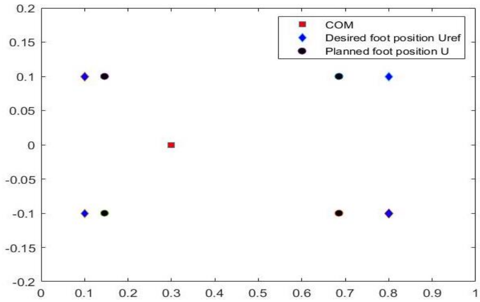


Figure7. The graph use the SQP to make a one-step prediction

The motion of the ontology assumed here is 0.3 seconds for the stance phases, 0.3 seconds for the flight phases, 1 m/s for the motion of the body, and the x-axis for the direction of motion. The initial point of the robot coincides with the initial point of the trajectory. The result is the location of the landing point of the first step after optimization by MPC. The first two landing points indicate the current position of the front leg support and the latter two points indicate the position of the rear leg support. It can be seen that the final optimization result is closer to the designed position.

Figure 6 is the prediction using iLQG, and Figure 7 is the result using SQP.By comparison, it can be found that the deviation between the position of the result of the iLQG algorithm and the SQP algorithm is small, and does not exceed 0.1 m.

The following is the tracking of the robot's centroid obtained by the iLQG algorithm in the first prediction range(Table 2 is the expected position and velocity of the COM and table 3 is the position and velocity of the COM within a predicted range).

Table 2. The expected position and velocity of the COM.

State variable	Current moment state	The first state of expectation	The second state of expectation	The third state of expectation	The fourth state of expectation
x	0.3	0.6	0.9	1.2	1.5
y	0	0	0	0	0
z	0.4	0.4	0.4	0.4	0.4
v_x	1	1	1	1	1
v_y	0	0	0	0	0
v_z	0	0	0	0	0

Table 3. The position and velocity of the COM within a predicted range.

State variable	Current moment state	The first state of predicted	The second state of predicted	The third state of predicted	The fourth state of predicted
x	0.3	0.6	0.8976	1.1952	1.4926
y	0	0	0	0	0
z	0.4	0.4	0.6956	0.1092	0.5858
v_x	1	0.9920	0.9920	0.9912	0.9912
v_y	0	0	0	0	0
v_z	0	0.09	-0.1	0.5	-0.9

Through comparison, it can be found that the position and speed of the robot body are well tracked within the predicted range, and the error will increase with the extension of the predicted range. Since we only input the first value in the end, the prediction input error after that is acceptable.

The MPC algorithm is used to continuously predict the input so that the robot can finally track the desired trajectory.

4.2. The convergence of the iLQG algorithm

The planning results of the front legs and the back legs of the first step are taken as an example:

The initial value of the algorithm is a preset heuristic. It can be found from the figure 8 that in the process of calculating the footstep position, the distribution result of each calculation is concentrated and gradually converges to the optimal value.

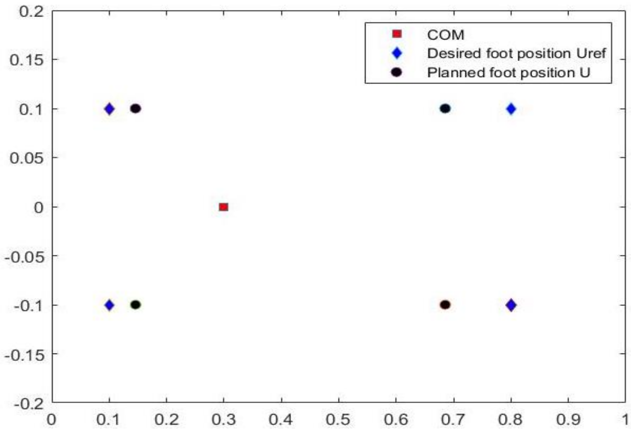


Figure8. One-step prediction of the convergence of the process

4.3. The trajectory tracking effect of the algorithm

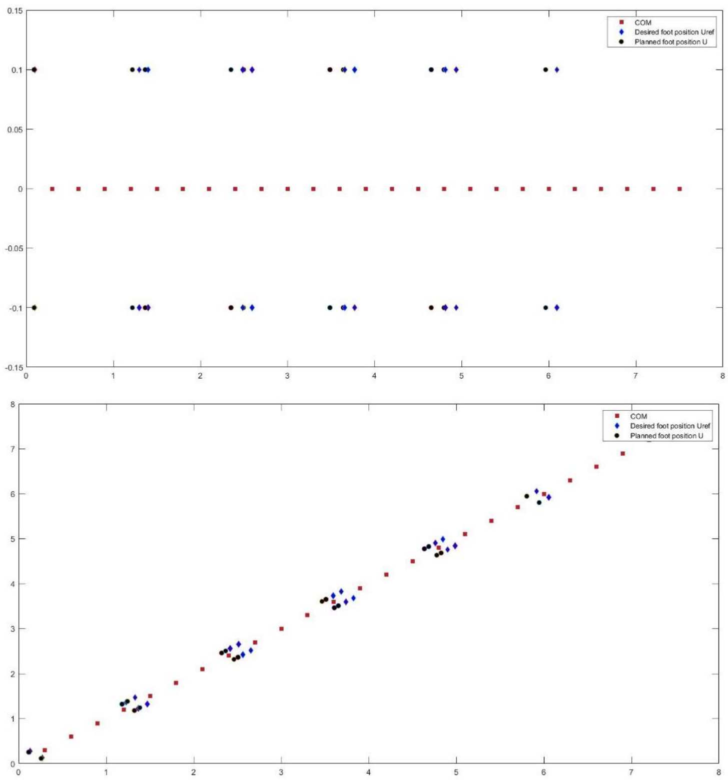


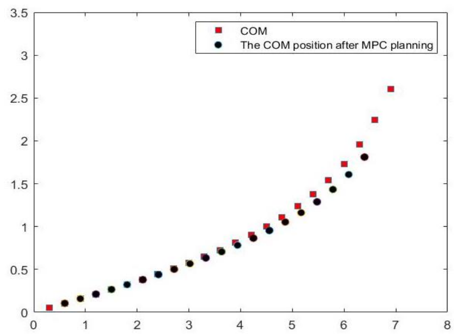
Figure9. Graph algorithm for tracking the trajectory

In the simulation, two kinds of motion trajectories are mainly set. The initial state of the robot is toward the x-axis forward direction. In the first case, the desired trajectory is a uniform linear motion along the x-axis forward direction of 1 m/s. The second is a uniform linear motion that is deflected by 45°.

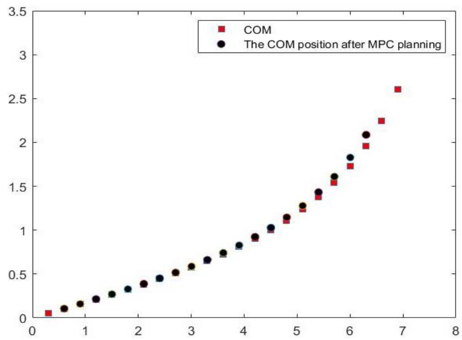
It can be seen from the figure 9 that in the above two cases, the planned footstep position can well follow the planned trajectory.

4.4. Follow the COM

In this section, the desired trajectory is a following of the motion trajectory when the velocity component in the x-axis direction is 1 m/s and the angular velocity is $\pi/6$.



(a)iLQG



(b)SQP

Figure10. Example of a ONE-COLUMN figure caption

Figure 10(a) shows the tracking of the curve trajectory by the MPC control based on the iLQG optimization algorithm, and figure 10(b) shows the tracking effect based on the SQP algorithm. It can be found that the tracking accuracy based on the SQP algorithm is better than the iLQG algorithm.

4.5. Real-time performance of the algorithm

All of the above control algorithms were performed on a Dell microcomputer using a 2.6 GHz Intel Core i7-8750H six-core processor. All code is programmed in the MATLAB 2018a environment.

In the simulation, a total of 10 footstep position and foot forces were optimized, and the iterative optimization time was 5.078s. The average time spent on each optimization was about 0.5s. When planning with the SQP function, the average planning time for each foot and foot force is 5s. It can be seen that the algorithm iLQG is much faster than the SQP algorithm.

5. Conclusion

A new predictive controller based on the iLQG optimization algorithm is designed. A heuristic reference input with stable information is added to the cost of the trajectory optimizer. In this way, the final result of the solution can be guided to a stable motion. iLQG solves the nonlinear MPC by local linearization and obtains a faster convergence rate while losing the quadratic convergence. By comparison with the SQP algorithm, the deviation is small.

References

[1] Raibert M, Chepponis M, Brown H. Running on four legs as though they were one[J]. IEEE Journal on Robotics & Automation, 1986, 2(2):70-82.
[2] Raibert M H. Trotting, pacing and bounding by a quadruped robot[J]. Journal of Biomechanics, 1990, 23 Suppl 1(1):79-81.
[3] Pratt J, Torres A, Dilworth P, et al. Virtual actuator control[C]//International Conference on Intelligent Robots and Systems, 1996, 3: 1219-1226.
[4] Faraji S, Pouya S, Möckel R, et al. A Model-Based Control Approach for Locomotion Control of Legged Robots[M]. Lausanne, Switzerland: Lausanne
[5] Pratt J, Chew CM, Torres A, et al. Virtual model control: An intuitive approach for bipedal locomotion[J]. The International Journal of Robotics Research, 2001, 20(2): 129-143.
[6] Havoutis I, Semini C, Caldwell D G. Virtual model control for quadrupedal trunk stabilization[J]. 2013.

[7] Taleb S, Poulakakis I, Papadopoulos E, et al. Quadruped robot running with a bounding gait[M]//Experimental Robotics VII. Springer Berlin Heidelberg,2001: 281-289.
[8] Hawker G, Buehler M. Quadruped trotting with passive knees: design, control, and experiments[C]//Proceedings of International Conference on Robotics and Automation, 2000, 3: 3046-3051.
[9] Papadopoulos D, Buehler M. Stable running in a quadruped robot with compliant legs[C]//Proceeding International Conference on Robotics and Automation, 2000, 1: 444-449.
[10] de Lasa M, Buehler M. Dynamic compliant quadruped walking[C]//Proceeding International Conference Robotics and Automation, 2001, 3: 3153-3158.
[11] Palmer L R, Orin D E, Marhefka D W, Intelligent control of an experimental articulated leg for a galloping machine[C]// IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION. 2003:3821 - 3827.
[12] M. Rutschmann, B. Satzinger, M. Byl, and K. Byl. Nonlinear model predictive control for rough-terrain robot hopping. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pages 1859–1864, Oct. 2012.
[13] M. J. Powell, E. A. Cousineau, and A. D. Ames. Model predictive control of underactuated bipedal robotic walking. In Proc. of the IEEE Int. Conf. on Robotics and Automation, pages 5121–5126, May 2015.
[14] Y. Tassa, N. Mansard, and Todorov E. Control-limited differential dynamic programming. In Proc. of the IEEE Int. Conf. on Robotics and Automation, pages 1168–1175, 2014.
[15] B. Henze, C. Ott, and M. A. Roa. Posture and balance control for humanoid robots in multi-contact scenarios based on model predictive control. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems page 3253–3258, Sept 2014.
[16] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli. Trajectory Optimization Through Contacts and Automatic Gait Discovery for Quadrupeds. ArXive-prints, July 2016.
[17] Diehl M, Bock HG, Schlöder JP, et al.Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations.Journal of Process Control,2002,12:577-585.
[18] Ohtsuka T.A continuation/GMRES method for fast computation of nonlinear receding horizon control.Automatica,2004,40(4):563-574.
[19] Paul Kulchenko, Emanuel Todorov. First-exit model predictive control of fast discontinuous dynamics: Application to the ball bouncing[C]// IEEE International Conference on Robotics & Automation.2011.
[20] Emanuel Todorov, Weiwei Li. A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems [C]. AACC.2005.
[21] Gerardo Bledt. Policy Regularized Model Predictive Control Framework for Robust Legged Locomotion.2018.2
[22] Hae-Won Park, Patrick Wensing, and Sangbae Kim. Online planning for autonomous running jumps over obstacles in high-speed quadrupeds. In Proceedings of Robotics: Science and Systems, Rome, Italy, July 2015.
[23] Gerardo Bledt, Patrick M. Wensing, et al. Policy-Regularized Model Predictive Control to Stabilize Diverse Quadrupedal Gaits for the MIT Cheetah. International Conference on Intelligent Robots and Systems (IROS), September 24–28, 2017, Vancouver, BC, Canada.
[24] J. Pratt, J. Cardiff, S. Drakunov, and A. Goswami. Capture point: A step toward humanoid push recovery. In IEEE-RAS Int. Conf. on Humanoid Robots, pages 200–207, Genova, Italy, Dec. 2006.
[25] H. W. Park, Sangin Park, and S. Kim. Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3d running of MIT cheetah 2. In 2015 IEEE International Conference on Robotics and Automation (ICRA),pages 5163–5170, May 2015.
[26] Darren Krasny and David Orin. Evolution of a 3D gallop in a quadrupedal model with biological characteristics. Journal of Intelligent & Robotic Systems,60:59–82, 2010.