Research Article

Universal Walking Control Framework of Biped Robot Based on Dynamic Model and Quadratic Programming

Xiaokun Leng,^{1,2} Songhao Piao¹,¹ Lin Chang,^{1,2} Zhicheng He,^{1,2} and Zheng Zhu²

¹*School of Computer Science, Harbin Institute of Technology, Harbin 150001, China*

²*Leju (Shenzhen) Robotics, Shenzhen 518057, China*

Correspondence should be addressed to Songhao Piao; piaoshhit@163.com

Received 16 June 2020; Revised 29 August 2020; Accepted 1 October 2020; Published 31 October 2020

Academic Editor: Hang Su

Copyright © 2020 Xiaokun Leng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Biped robot research has always been a research focus in the field of robot research. Among them, the motion control system, as the core content of the biped robot research, directly determines the stability of the robot walking. Traditional biped robot control methods suffer from low model accuracy, poor dynamic characteristics of motion controllers, and poor motion robustness. In order to improve the walking robustness of the biped robot, this paper solves the problem from three aspects: planning method, mathematical model, and control method, forming a robot motion control framework based on the whole-body dynamics model and quadratic planning. The robot uses divergent component of motion for trajectory planning and introduces the friction cone contact model into the control frame to improve the accuracy of the model. A complete constraint equation system can ensure that the solution of the controller meets the dynamic characteristics of the biped robot. An optimal controller is designed based on the control framework, and starting from the Lyapunov function, the convergence of the optimal controller is proved. Finally, the experimental results show that the method is robust and has certain anti-interference ability.

1. Introduction

Humanoid robots are expected to perform various tasks in the near future. Humanoid robots have the characteristics of multiple joints and multiple degrees of freedom, which makes them have the potential to satisfy multitasking. The humanoid robot's motion control often faces some difficult situations, such as the number of degrees of freedom required by the robot to perform a variety of expected tasks is higher than the number of degrees of freedom the robot has. For a legged robot with six underactuated degrees of freedom as a whole, the most important tasks of motion control are maintaining balance and completing the expected trajectory. Generally speaking, the robot's feet are used to complete these motion trajectories, and then the corresponding grasping tasks are performed by the arms. However, with the development of motion control technology, arm movements can also be used to enhance the overall balance performance of the robot. Arm movement was originally used for dynamic balance during walking,

but it can also be used to assist in tasks such as kicking [1]. Therefore, a full-body motion control framework needs to be established to find the best control output, so that the robot can complete various expected tasks under the constraint conditions such as maintaining balance [2, 3].

The selection of the control variables of the robot's whole-body motion control frame has a great influence on the control performance that the robot can achieve. Examples of well-known whole-body motion control include virtual model control [4, 5], passivity-based whole-body controllers [6–8], and inverse dynamics-based approaches such as those presented in [9, 10]. Most methods solve local minima, i.e., optimize only the state of the current control frame, and then use the centroid dynamics model for predictive control. The method of using the whole-body dynamics model for predictive control has a higher model complexity and a larger amount of calculation [11]. Thanks to the improvement of the computing power of the controller, it can be applied to the gait control of biped robots [12].

There are too many control algorithms for biped robots, and it is difficult to describe all control systems with one structure. The controller is usually divided into walking pattern generator and stabilizer. The former is used to generate a series of joint trajectories to act on the actual robot. When the robot is about to fall, the latter will modify the previously generated trajectory based on various sensor readings (contact force sensor, gyroscope, accelerometer, etc.). The ASIMO robot is the most outstanding representative of this method, usually based on the zero moment point (ZMP), the divergent motion component (DCM), or their deformation [13]. Although it is impossible to unify all methods, the structure of the controller can be divided into upper-level motion plan (advanced motion plan) and lower-level tracking (low-level tracking) [14]. The upper-level motion plan usually uses a simplified dynamic model to generate the center of gravity and foot trajectory, while the lower level below considers using a more complex model to generate the position or torque of each joint. The generation of high-level motion trajectories usually requires model-based prediction of the future. This is the basic idea of model predictive control (MPC). The mainstream method is still to form a trajectory optimization problem and then use MPC to solve the optimization problem and implement a real-time solution [15]. The simplified model is usually used because the calculation amount of the complex model is too large, and there is currently no method for performing real-time calculation. The specific degree of simplification is related to the movement to be achieved, the form of trajectory optimization, and the computing power of the existing controller. It should be noted that even in the case where the upper-layer trajectory is based on a simplified model, since the lower layer considers a more complex model, this makes the resulting joint trajectory conform to the dynamic characteristics of the actual robot and can follow the upper-layer trajectory very well. Usually, the lower level needs to solve inverse kinematics (IK) or inverse dynamics (ID) [16]. The mainstream method is to integrate IK or ID into one or more quadratic programming (QP) and then solve these QP problems in real time to generate the final joint trajectory [17, 18].

This paper proposes a whole-body motion control method based on inverse dynamics, integrating motion control into an optimization framework based on quadratic programming and using joint torque and linear contact force as optimization variables for quadratic programming problems. Considering the complexity of the algorithm and the real-time nature of control, this paper selects the existing technology and finally forms the most complex control framework. The second section introduces the DCM trajectory planning; the third section introduces the control framework used for trajectory tracking; and the fourth section introduces the construction of the secondary planning controller. Section 5 verifies the control framework through experiments.

2. Biped Gait Trajectory Planning Based on DCM

DCM points in three-dimensional space can be defined as

$$\xi = \mathbf{x} + b\dot{\mathbf{x}}. \quad (1)$$

$\xi = [\xi_x, \xi_y, \xi_z]^T$ is DCM point; $\mathbf{x} = [x, y, z]^T$ and $\dot{\mathbf{x}} = [\dot{x}, \dot{y}, \dot{z}]^T$ are the position and velocity of the center of mass (CoM) point, respectively. The formula above shows that the CoM point can naturally and steadily follow the movement of the DCM point [19, 20]. After differentiation, we obtain

$$\dot{\xi} = -\frac{1}{b}\mathbf{x} + \frac{1}{b}\xi + \frac{b}{m}\mathbf{F}. \quad (2)$$

It shows that the external force $\mathbf{F}$ can directly affect the dynamics of DCM. Through analysis, the combined force $\mathbf{F}$ received by the system is

$$\mathbf{F} = \frac{mg}{\Delta z_{\text{vrp}}}(\mathbf{x} - \mathbf{r}_{\text{vrp}}). \quad (3)$$

In formula (3), m is the system mass, g is the gravity coefficient, $\mathbf{x}$ is the robot coordinate, $\mathbf{r}$ is the VRP, which is the space vector, and Δz_{vrp} is the scalar, which represents the height of the robot's torso.

In order to make the actual DCM point run on the planned trajectory, the closed-loop control method is used to track the expected DCM trajectory:

$$\dot{\xi} - \dot{\xi}_d = -k_{\xi} \underbrace{(\xi - \xi_d)}_{e_{\xi}}. \quad (4)$$

Adopt proportional (P) control rate to achieve stable tracking of DCM points. Among them, when $k_{\xi} > 0$, the DCM error can converge gradually and steadily. For a given virtual repellant point (VRP) $\mathbf{r}_{\text{vrp}}$ and the corresponding centroidal moment pivot point (CMP), $\mathbf{r}_{\text{ecm}}$ position, it can converge to the expected value. Expected DCM tracking control rate is

$$\mathbf{r}_{\text{vrp}} = \xi + k_{\xi} \sqrt{\frac{\Delta z_{\text{vrp}}}{g}}(\xi - \xi_d) - \sqrt{\frac{\Delta z_{\text{vrp}}}{g}}\dot{\xi}_d. \quad (5)$$

Among them, formula (5) only stabilizes the unstable part of DCM and does not affect the natural and stable CoM dynamic changes. The closed-loop dynamic situation is

$$\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix} = \underbrace{\begin{bmatrix} -\sqrt{\frac{g}{\Delta z_{\text{vrp}}}}I & \sqrt{\frac{g}{\Delta z_{\text{vrp}}}}I \\ 0 & -k_{\xi}I \end{bmatrix}}_{\text{feedback}} \begin{bmatrix} x \\ \xi \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ k_{\xi}I & I \end{bmatrix}}_{\text{feedforward}} \begin{bmatrix} \xi_d \\ \dot{\xi}_d \end{bmatrix}. \quad (6)$$

Among them, I and 0 represent the unit matrix and zero matrix of 3×3 . For $k_{\xi} > 0$, the eigenvectors of the system matrix are stable, and the feed-forward terms ξ_d and $\dot{\xi}_d$ are

both bounded and then can meet bounded input bounded output stable (BIBO) standards.

3. Dynamic Control Framework for Biped Robot

A universal control method for biped robots is to control rigid joint tracking position through IK. This method only requires kinematic models and is widely used in actual robots; the other is based on ID. The method can provide a compliant motion trajectory and is robust to external disturbances, but its performance depends largely on the quality of the dynamic model, which is difficult to obtain by measuring actual robots.

At present, the better method is to use the inverse kinematics IK as a complement to the controller based on the inverse dynamics ID to compensate the modeling error to improve the anti-interference ability. Many biped robot control methods can be decoupled into two layers: a planner that outputs the position of the center of mass and a low-level controller responsible for joint layer control. However, in order to expand the application space of the biped robot and make the robot more robust to external disturbances, the underlying controller also needs to consider full-body kinematics and dynamics. The controller architecture needs to solve for whole-body ID and IK in each control cycle to track higher-level targets. The control model is divided into two parts: IK and ID. Both are formulated as two independent QP problems. Problems have their own goals and constraints.

The servo control command of the joint layer is calculated by the following formula:

$$i = K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) + K_f(\tau_d - \tau). \quad (7)$$

In formula (7), q_d , $\dot{q}_d$, and τ_d are the expected joint position, speed, and torque, and q , $\dot{q}$, and τ are measured quantities. Use ID to calculate τ_d for force control and IK to calculate q_d and $\dot{q}_d$.

For robot gait, the robot trajectory is generated at the user level, such as the expected trajectory of feet, hands, and CoM in the Cartesian coordinate system. The full-body robot controller takes it as input and calculates the control amount of each individual joint, such as the joint position, speed, and torque, and then uses these control variables as reference inputs for the joint layer servo controller. The robot's joint position, speed, joint acceleration, and torque are calculated separately. IK and ID are described by QP problems:

$$\begin{aligned} \min_{\chi} \quad & 0.5\chi^T G\chi + g^T \chi \\ \text{s.t.} \quad & C_E\chi + c_E = 0, \\ & C_I\chi + c_I \geq 0. \end{aligned} \quad (8)$$

In the formula, unknown state variables χ and constraints C_E , c_E , C_I , and c_I are all related to specific control problems. Two QP problems are solved once in each control cycle. The cost function can be optimized as

$$0.5\|A\chi - b\|^2. \quad (9)$$

So, $G = A^T A$ and $g = -AT^b$. A and b can be broken down into

$$\begin{aligned} A &= [w_0 A_1 \ w_1 A_1 \ \cdots \ w_n A_1]^T, \\ b &= [w_0 b_1 \ w_1 b_1 \ \cdots \ w_n b_1]^T, \end{aligned} \quad (10)$$

w_i is the weight needed for the cost function.

3.1. Joint Force Control Based on Inverse Dynamics. The motion equation and constraint equation of a biped robot can be described as

$$\begin{aligned} M(q)\ddot{q} + h(q, \dot{q}) &= S\tau + J^T(q)F, \\ J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} &= \ddot{x}. \end{aligned} \quad (11)$$

In formula (11), $(q, \dot{q})$ is the complete state of the system, including 6 degrees of freedom on the underactuated torso, $M(q)$ is the inertia matrix, and $h(q, \dot{q})$ is the sum of gravity, centrifugal force, and Coriolis force, S is a selection matrix, where the first 6 rows of the 6 degree of freedom (DOF) corresponding to the underactuated torso are zero, and the rest is a unit matrix, τ is the joint moment vector, $J^T(q)$ is the Jacobian matrix of all contact points, F is the vector of all contact forces in the world coordinate system, and x is the vector of contact position and direction in the Cartesian space. The number of rows and columns of F and J^T depends on the number of contact points. In order to describe the contact force of the biped robot with the ground more clearly, the equation of motion (11) was rewritten as follows:

$$M(q)\ddot{q} + h(q, \dot{q}) = S_\tau + \sum_{k=0}^{N_c} J_k^T(q) \cdot f_k. \quad (12)$$

In formula (12), N_c is the number of foot contact friction cones, $J_k^T(q)$ is the Jacobian matrix of each friction cone, and f_k is the contact force.

3.1.1. Cartesian Space Acceleration. Using the acceleration deviation in the Cartesian space as the evaluation function, we obtain

$$\begin{aligned} \ddot{x} &= J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q}, \\ A_{\text{cart}} &= [J(q) \ 0 \ 0], \\ b_{\text{cart}} &= \ddot{x}^* - \dot{J}(q, \dot{q})\dot{q}, \\ \ddot{x}^* &= K_p(x_d^* - x) + K_d(\dot{x}_d^* - \dot{x}) + \ddot{x}_d^*. \end{aligned} \quad (13)$$

Calculate the input $\ddot{x}^*$ by formula (13), where x_d^* , $\dot{x}_d^*$, and $\ddot{x}_d^*$ are specified by the higher-level controller. Calculate the sum based on the current robot state through forward kinematics, and track the CoM, hand, foot, and torso directions. Acceleration can be calculated by equation (13). For different calculation targets, the Jacobian rows without constraints need to be omitted accordingly. In order to enhance the stability of calculations during contact, contact is not regarded as a hard constraint, but a penalty function

with a higher weight is used to represent the contact to speed up the solution of the quadratic programming problem. The cost function at the time of contact x_d^* , $\dot{x}_d^*$ can be ignored and set $\ddot{x}^* = 0$.

At the same time, considering the influence of CoM trajectory, torso posture, foot center trajectory of swinging leg, and foot center trajectory of standing leg on robot control, the cost function formula (10) is extended to (14), where w is the weighting coefficient and A_{com} is the centroid tracking cost in the end acceleration tracking cost:

$$A = \begin{bmatrix} w_{\text{CoM}} A_{\text{CoM}} \\ w_{\text{Torsor}} A_{\text{Torsor}} \\ w_{\text{Swing}} A_{\text{Swing}} \\ w_{\text{Stance}} A_{\text{Stance}} \\ w_x A_x \end{bmatrix}, \quad (14)$$

$$b = \begin{bmatrix} w_0 b_0 \\ w_1 b_1 \\ \vdots \\ w_n b_n \\ w_x b_x \end{bmatrix}.$$

Constraint equation (15) corresponds to the CoM trajectory:

$$\begin{aligned} \ddot{x} &= J \cdot \ddot{q} + \dot{J} \cdot \dot{q}, \\ A_{\text{CoM}} &= [J_{\text{CoM}}(q)], \\ b_{\text{CoM}} &= \ddot{x}_{\text{CoM}}^* - \dot{J}_{\text{CoM}}(q) \cdot \dot{q}. \end{aligned} \quad (15)$$

Similarly, the swing leg and standing leg trajectory constraint equations are

$$\begin{aligned} A_{\text{Swing}} &= [J_{\text{Swing}}(q)], \\ b_{\text{Swing}} &= \ddot{x}_{\text{Swing}}^* - \dot{J}_{\text{Swing}}(q) \cdot \dot{q}, \\ A_{\text{Stance}} &= [J_{\text{Stance}}(q)], \\ b_{\text{Stance}} &= \ddot{x}_{\text{Stance}}^* - \dot{J}_{\text{Stance}}(q) \cdot \dot{q}. \end{aligned} \quad (16)$$

The body posture constraint equation is

$$\begin{aligned} A_{\text{Torsor}} &= [J_{\text{Torsor}} \omega], \\ b_{\text{Torsor}} &= \dot{\omega}_{\text{Torsor}}^* - J_{\text{Torsor}}(q) \cdot \dot{q}, \\ \dot{\omega}_{\text{Torsor}}^* &= K_p (\xi_d^* - \xi) + K_d (\omega_d^* - \omega) + \dot{\omega}_d. \end{aligned} \quad (17)$$

3.1.2. Center of Pressure (CoP). Given the contact force and contact moment ${}^b M$, ${}^b F$ in the foot coordinate system, the position of the pressure center in the foot coordinate system is

$$p = \begin{bmatrix} \left(\frac{{}^b M_y}{{}^b F_z} \right) \\ \left(\frac{{}^b M_x}{{}^b F_z} \right) \end{bmatrix}. \quad (18)$$

Cost function of pressure center deviation is

$$A_{\text{CoP}} = \begin{bmatrix} 0 & 0 & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix}, \quad (19)$$

$$b_{\text{CoP}} = 0.$$

In formula (19), p is the contact force amplitude vector, which is consistent with the general β_{ij} meaning of the constraint equation, $*$ represents the reference value, and p_x^* , p_y^* is the expected pressure center position in the foot coordinate system given by the high-level controller. In formula (20), R is the transformation matrix from the world coordinate system to the foot coordinate system.

3.1.3. Weight Distribution. In the two-leg support, specify the desired weight distribution $w^* = F_{zl} / (F_{zl} + F_{zr})$, and add this to the cost function:

$$\begin{aligned} A_{\text{weight}} &= \begin{bmatrix} 0 & 0 & S_{\text{weight}} \end{bmatrix}, \\ b_{\text{weight}} &= 0. \end{aligned} \quad (20)$$

In formula (20), S_{weight} is a row vector with zeros, except for $S_{\text{weight}}(3) = 1 - w^*$ and $S_{\text{weight}}(9) = -w^*$, the rest are zero.

3.1.4. Track State Variables and Regularization Directly. Use the expected value directly to evaluate χ ; then formula (21) is obtained, where A is the selection matrix, b is target vector, q is joint position, τ is torque, F is end contact force, and $*$ represents the reference value:

$$\begin{aligned} A_{\text{state}} &= I, \\ b_{\text{state}} &= [\ddot{q}^* \quad \tau^* \quad F^*]^T. \end{aligned} \quad (21)$$

If no target value is specified in the formula, 0 is used. This item can be used to directly control specific joints or forces, which can directly adjust the value of the state variable χ , making QP problems easier to solve.

3.1.5. Torque Change. In order to avoid joint torque oscillation, the speed of change of τ needs to be considered:

$$\begin{aligned} A_{d\tau} &= \begin{bmatrix} 0 & I & 0 \end{bmatrix}, \\ b_{d\tau} &= \tau_{\text{prev}}. \end{aligned} \quad (22)$$

In formula (22), τ_{prev} is the output of the previous period.

3.2. Friction Cone Contact Model. In the control of a biped robot, it is difficult to solve the contact force under different contact conditions. In order to simplify the solution process, the contact surface of the foot is approximated by multiple contact points, and the contact force of these contact points is solved. For the feet of a robot, a friction cone constraint can be used to determine whether the point will slide. The point will not slide only if the contact force vector is inside the friction cone. As shown in Figure 1, the apex of the friction cone is located at the contact point, the normal line is perpendicular to the ground, and the slope is determined by the friction coefficient of the contact between the sole and the point.

However, this constraint method represented by a friction cone is nonlinear, although the solver can handle conic constraints. However, the calculation of the cone constraint is heavy. In order to speed up the solution, this paper uses a convex polygon as shown in Figure 1 to approximate the cone constraint of the contact point. In the contact force model represented by the convex polygon model, the feasible contact force and contact moment of the foot are a linear combination of the contact force f_i of each point. f_i is the product of a unit vector ${}^0u f_i$ representing the direction of the contact force in space and a scalar ρ_i representing the magnitude of the contact force, as shown in formula (25):

$$f_i = {}^0u f_i \rho_i. \quad (23)$$

Therefore, at a given contact point, the contact forces in all friction cones can be represented by the contact forces at a total of $n\rho$ contact points. The unit vector ${}^0u f_i$ of the feasible contact force direction is determined by the contact point. The amplitude scalar ρ_i of the contact force is limited to a positive value. All $n\rho$ contact force amplitude scalars are combined into a contact force amplitude vector ρ ; then,

$$\rho = [\rho_1 \ \cdots \ \rho_{n_p}]^T. \quad (24)$$

Contact force vector ρ can be mapped to six-dimensional space contact force and moment:

$$\begin{bmatrix} F_x & F_y & F_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T = {}^0A_{p,k} \rho. \quad (25)$$

In formula (25), k represents the k th part of the robot that is in contact with the external environment. Each ${}^0A_{p,k}$ maps the contact force amplitude vector ρ on the limb to the contact force and contact torque to the k limb.

Each foot of a biped robot is approximated by four contact points located on the edge of the foot, and the friction cone of each contact point is approximated by 4 contact force unit vectors, so the total number of contact force variables per leg $n_p = 16$. In formula (25), the force ρ is mapped to the foot contact torque through the matrix $A_{p,k}$.

It should be noted that the friction force obtained is reasonable only when the measured friction coefficient, the contact state of the foot, and the ground normal vector of the

FIGURE 1: Friction cones and polyhedral cones.

contact point are approximately accurate; otherwise, the system will be seriously unstable.

3.3. Constraints. Considering the constraints such as the output torque of each joint of the actual robot, the motion equations need to be used as equation constraints, and various inequality constraints must be added to ensure that the solution results do not violate relevant physical conditions. Inequality constraints include positive contact forces, joint moments within a certain range, etc.; the optimization problem is formulated as

$$\begin{aligned} \min_{\ddot{q}, \beta, f} & \omega_1 \omega_1 A_{\text{CoP}} f_i - b^2 + \omega_2 A_{\ddot{q}} \ddot{q} - b^2 + \varepsilon_1 \tau - \tau_{\text{prev}}^2 + \varepsilon_2 \sum_{ij} \beta_{ij}^2, \\ M(q) \ddot{q} + h(q, \dot{q}) &= S_\tau + \sum_{j=1}^{N_c} J_j^T(q) \cdot f_j, \\ f_j &= \sum_{i=1}^{N_c} \beta_{ij} v_{ij}, \quad \beta_{ij} \geq 0, \quad \forall i = \{1, \dots, 4\}, \\ \forall j &= \{1, 2, \dots, N_c\}, \\ \tau &< \tau_{\text{max}}. \end{aligned} \quad (26)$$

In formula (26), they are dynamic model constraint, friction cone constraint, contact force amplitude vector constraint, and joint torque constraint. The cost items are CoP tracking cost, end acceleration tracking cost, and torque change cost, τ is the joint torque, contact force vector cost, β_{ij} is the contact force amplitude vector, f_i represents the friction between the sole of the foot and the ground, and τ_{max} is the limit value of the output torque of each joint. The coefficients of each cost term are ω_1 , ω_2 , ε_1 , and ε_2 .

At present, the friction coefficient f is obtained through preliminary experiments, and the friction coefficient of different roads is different. The function of the friction cone is to limit the force of the robot on the ground when the friction coefficient is small, to prevent sliding friction during

walking, which will cause the robot to lose stability. In the future, under unknown walking conditions, sliding friction during walking will be detected and corrected.

Finally, the robot control framework is shown in Figure 2. There are many parameters in the control framework that need to be adjusted. The parameters that need to be adjusted include the PD controller feedback coefficient when calculating the end acceleration cost, the weight coefficient of each end point cost in the end acceleration cost, and the weight coefficient of each cost in the secondary programming problem, such as ω_1 , ω_2 , ε_1 , and ε_2 . Parameter debugging process is as follows:

- (1) The weight coefficients w_1 and w_2 in the end acceleration cost are set to a larger value, about 1–100. Debug ω_1 and ω_2 until better CoP tracking and end acceleration tracking are achieved.
- (2) Debug the PD controller feedback coefficient until the robot achieves a better position tracking effect.
- (3) Given small values of ε_1 and ε_2 , about 10^{-6} – 10^{-2} , adjust the parameters until the robot joint torque and contact force fluctuations are minimal.

Although there are many total parameters, it is easier to determine the appropriate control parameters when debugging in stages.

4. Optimal Controller Based on PD Control and Quadratic Programming

The trajectory planning algorithm and the IK and ID algorithms can be used to obtain the desired angle of each joint of the robot, but the actual output torque of each joint is limited. Therefore, it is necessary to design an optimal controller to limit the joint torque of the biped robot. Through optimized force control, the robot's motion posture can better track the expected posture. The biped robot dynamics model is

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Bu + J(q)F_1. \quad (27)$$

Output dynamics are

$$\dot{y} = T_y \dot{q} - \frac{\partial y_d}{\partial t}. \quad (28)$$

Differentiating formula (28), we obtain

$$\ddot{y} = \dot{T}_y \dot{q} + J_y \ddot{q} - \frac{\partial^2 y_d}{\partial t^2} - \frac{\partial}{\partial q} \left(\frac{\partial y_d}{\partial t} \right) \dot{q}. \quad (29)$$

Substitute $\ddot{q}$ into equation (29) to obtain equation (31), where

$$\ddot{q} = -D^{-1}C\dot{q} - D^{-1}G + D^{-1}Bu + D^{-1}J(q)F, \quad (30)$$

$$\begin{aligned} \ddot{y} = & \dot{J}_y \dot{q} + J_y D^{-1}BU - J_y D^{-1}(G - JF_1 + c\dot{q}) \\ & - \frac{\partial^2 y_d}{\partial t^2} - \frac{\partial}{\partial q} \left(\frac{\partial y_d}{\partial t} \right) \dot{q}. \end{aligned} \quad (31)$$

Sorting formula (31), we obtain

$$\begin{aligned} \ddot{y} - \frac{1}{J_y \dot{q}} + \frac{\partial^2 y_d}{\partial t^2} + \frac{\partial}{\partial q} \left(\frac{\partial y_d}{\partial t} \right) \dot{q} + J_y D^{-1}(G - JF_1 + c\dot{q}) \\ = J_y D^{-1}Bu. \end{aligned} \quad (32)$$

Multiplying formula (32) left by $D_y = (J_y D^{-1} J_y^T)^{-1}$, we obtain

$$F = - \left(K_p y + K_d \dot{y} + \dot{D}_y \dot{y} - (C_y + J_{y_d}) \dot{q} - G_y - D_y \frac{\partial^2 y_d}{\partial t^2} \right). \quad (33)$$

Among them, $D_y J_y D^{-1} BU = F$, $D_y T_y D^{-1}(G - JF_1) = G_y$, $J_{y_d} = D_y (\partial/\partial q) (\partial y_d/\partial t)$, and $C_y = D_y (J_y D^{-1} C - \dot{J}_y)$. Corresponding form is

$$Bu = (\mathbf{1} - N[(\mathbf{1} - BB^T)N]^+) J_y^T F. \quad (34)$$

Among them,

$$N = (\mathbf{1} - J_y^T (J_y D^{-1} J_y^T)^{-1} J_y D^{-1}). \quad (35)$$

Formula (34) describes the relationship between the auxiliary variable, and when the controller F is used, the corresponding control law can be substituted into the equation to find. Consider the form of F :

$$F = - \left(K_p y + K_d \dot{y} + \dot{D}_y \dot{y} - (C_y + J_{y_d}) \dot{q} - G_y - D_y \frac{\partial^2 y_d}{\partial t^2} \right). \quad (36)$$

Put it into formula (27); then,

$$\begin{aligned} D_y \ddot{y} = & -k_p y - k_d \dot{y} - \dot{D}_y \dot{y}, \\ \ddot{y} = & -D_y^{-1} k_p y - D_y^{-1} k_d \dot{y} - D_y^{-1} \dot{D}_y \dot{y}. \end{aligned} \quad (37)$$

For the above system, consider the following Lyapunov function:

$$V = \frac{1}{2} y^T k_p y + \frac{1}{2} \dot{y}^T D_y \dot{y} + \alpha y^T D_y \dot{y}. \quad (38)$$

Deriving from the above formula, we obtain

$$\begin{aligned} \dot{V} = & \frac{1}{2} \dot{y}^T k_p y + \frac{1}{2} y^T k_p \dot{y} + \frac{1}{2} \dot{y}^T D_y \dot{y} + \frac{1}{2} \dot{y}^T \dot{D}_y \dot{y} \\ & + \frac{1}{2} \dot{y}^T D_y \dot{y} + \alpha y^T D_y \dot{y} + \alpha \dot{y}^T D_y \dot{y} + \alpha y^T \dot{D}_y \dot{y} + \alpha \dot{y}^T \dot{D}_y \dot{y} \\ = & y^T (-\alpha k_p) y + y^T \left(k_p - \frac{k_p D_y^T D_y}{2} - \alpha k_d + \alpha \dot{D}_y \right) \dot{y} \\ & + \dot{y}^T \left(-\frac{k_p}{2} \right) y + \dot{y}^T \left(\frac{k_d D_y^T D_y + \dot{D}_y^T D_y + k_d - 2\alpha D_y}{2} \right) \dot{y}. \end{aligned} \quad (39)$$

FIGURE 2: Robot control framework.

Converting the above formula into a matrix, we obtain

$$\dot{V} = - \begin{bmatrix} y^T & \dot{y}^T \end{bmatrix} \begin{bmatrix} \alpha k_p & -k_p + \frac{k_d D_y^{-T} D_y}{2} + \alpha k_d - \dot{\alpha} D_y \\ \frac{k_p}{2} & \frac{k_d D_y^{-T} D_y + \dot{D}_y^T D_y^{-T} D_y + k_d - 2\alpha D_y}{2} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \end{bmatrix}. \quad (40)$$

In order to ensure the positive definiteness of the matrix in formula (40), $k_p > ((k_d D_y^T D_y)/2)$, a smaller value α is often selected, but it is necessary to ensure that formula (41) is a positive definite matrix:

$$\begin{bmatrix} \alpha k_p & -k_p + \frac{k_d D_y^{-T} D_y}{2} + \alpha k_d - \dot{\alpha} D_y \\ \frac{k_p}{2} & \frac{k_d D_y^{-T} D_y + \dot{D}_y^T D_y^{-T} D_y + k_d - 2\alpha D_y}{2} \end{bmatrix}. \quad (41)$$

For an actual biped robot system, the physical meaning of D in the actual system is an inertia matrix, which is positively definite and symmetrical. Therefore, D_y can be obtained as a symmetric matrix. The selection conditions for k_p can be modified to $k_p > ((k_d D_y^T D_y)/2)$; the corresponding matrix to be verified can also be simplified to

$$\begin{bmatrix} \alpha k_p & -k_p + \frac{k_d}{2} + \alpha k_d - \dot{\alpha} D_y \\ \frac{k_p}{2} & \frac{k_d + \dot{D}_y^T + k_d - 2\alpha D_y}{2} \end{bmatrix}. \quad (42)$$

The construction space of the biped robot system includes the joint angles $q = (q_1, q_2, \dots, q_{12})$ and the corresponding underactuated degrees of freedom. The mass data and the moment of inertia data of each part of the robot body are obtained from the CAD model. Considering trajectory following, its control problem is of the form:

$$\begin{aligned} \min_{F, u, \delta} \quad & (F - F_{\text{ref}}(t, q, \dot{q}))^T (F - F_{\text{ref}}(t, q, \dot{q})) + 1000 * \delta^2 \\ \text{s.t.} \quad & (\alpha(y)y^T + y^T) \left(K_p y + K_d \dot{y} + \frac{1}{2} \dot{D}_y \dot{y} - C_y \dot{q} - G_y - D_y \frac{\partial^2 y_d}{\partial t^2} + u \right) \leq \delta, \\ & Bu = (1 - N[(1 - BB^T)N]^+) J_y^T F, \\ & |u_i| \leq U_{\text{lim}} Nm. \end{aligned} \quad (43)$$

Among them, u is the actual output torque value of each joint, and U_{lim} is the maximum allowable torque of each joint, according to the actual torque limit of each joint. By reasonable selection of the coefficients K_p and K_d in the controller, the influence of system

nonlinearity on system stability can be avoided to the greatest extent. The matrix B represents the selection matrix in the underdrive system. The first 6 rows of the corresponding underdrive 6 DOFs are 0, and the rest are the unit matrix.

FIGURE 3: Biped robot for experiment. (a) Front view. (b) Side view. (c) Oblique view.

FIGURE 4: Angle of joints of biped robot. (a) Position control. (b) Force control.

FIGURE 5: CoP trajectory of biped robot. (a) Position control. (b) Force control.

FIGURE 6: SDF data of biped robot. (a) Position control. (b) Force control.

In order to quickly solve the optimal control problem, the open-source OSQP solver is selected to solve the abovementioned quadratic programming problem. The solver uses an improved first-order alternating direction multiplier method (ADMM) to solve the quadratic programming problem [21]. By decomposing the matrix at the initial stage, the calculation amount in the subsequent process is greatly reduced. The optimal control algorithm is robust; its optimization results are not sensitive to the initial

value, and good solution results can be obtained by using different initial values.

The control framework integrates methods that can improve the robustness of the system. The framework uses DCM for trajectory planning, and DCM itself has anti-interference ability; uses friction cone and whole-body dynamics model to improve the accuracy of the model and reduce the control error caused by the rough model; considers responsibility for the actual situation to obtain a variety

FIGURE 7: ZMP trajectory of biped robot. (a) Position control. (b) Force control.

FIGURE 8: Biped robot increase weight experiment. (a) Angle of joints. (b) CoP trajectory.

of constraints and use QP; realizes the optimal controller; and further improves the robustness of robot walking.

5. Experimental Verification

Under the robot control framework, force control and position control are applied to the biped robot, respectively, and the comparison effect is examined. The experimental conditions are that the biped robot moves forward 5 steps,

the robot stride is 30 cm, and the period is 1.2 seconds. The weight of the physical robot is about 60 kg, as shown in Figure 3. The robot's inverted pendulum time constant ω is selected according to the interest rate difference.

Figure 4(a) is the joint angle plan and actual trajectory curve under position control, and Figure 4(b) is the joint angle plan and actual trajectory curve under force control. Figure 5(a) shows the CP and CoP trajectories under position control, and Figure 5(b) shows the CP and CoP

FIGURE 9: The centroid of the biped robot moves forward 4 cm experiment. (a) Angles of joints. (b) CoP trajectory.

FIGURE 10: Biped robot antithrust experiment. (a) Angles of joints. (b) CoP trajectory.

trajectories under force control. It can be seen that based on the position control, each joint performs position control according to the trajectory plan calculated in advance, so the joint angle planning curve and the actual trajectory curve are basically coincident. In the case of force control, the joint output force is tracked and controlled, and the joint angle planning curve and actual curve are no longer consistent. Although the position control ensures the accuracy of position output, the control effect of the CoP trajectory is not as good as that of the force control method. Under the force control method, the CoP trajectory curve of the robot walking will be smoother, and the fluctuation is smaller than that of the position control method.

Under the position control and force control methods, the six-dimensional force (SDF) data of the ankle joint is shown in Figures 6(a) and 6(b). The force control method can effectively reduce the impact force of the robot's foot and the ground. Under position control, the robot's feet are in hard contact with the ground, and the ground reaction force is greater than the force control method, which affects the robot's walking stability. The robot goes through several walking cycles to balance the impact. Compared with the position control method, the force control method can better absorb the external impact and ground reaction force and restore the balance in a shorter time.

Figures 7(a) and 7(b) are ZMP trajectories under two control modes.

It can be seen that based on position control, ZMP trajectory control is not as good as force control. During walking, ZMP can only be proved to be stable if it is within the stable area of the foot, but from the perspective of ZMP trajectory, the actual ZMP is calculated and fluctuates greatly, so ZMP trajectory only shows the stability of walking to a certain extent.

Figures 8(a) and 8(b) show the trajectory of the joint position after the robot's mass increases by 2 kg and the trajectory effect of the CP point and the CoP. It can be seen that after the mass of the robot increases, its joint trajectory remains almost unchanged, and there is a certain deviation between the actual CP trajectory and the reference trajectory, but due to the robustness of the optimal controller, the tracking error of the robot is small. The CP trajectory will remain within a certain range and will not cause the robot to fall, and the tracking of the CoP trajectory will be better.

Figures 9(a) and 9(b) show the robot joint position trajectory after the robot's center of mass is moved forward by 4 cm and the CP point and CoP trajectory effects. There is a certain inherent deviation between the actual CP trajectory and the reference trajectory of the robot, but due to the robustness of the optimal controller, the deviation value of the CP trajectory will eventually tend to a more stable value, which will not cause the robot to be unstable. In other words, CoP trajectory tracking is better.

Figures 10(a) and 10(b) show that the trunk of the robot is subjected to a continuous thrust of 30 N at 1.5–1.6 s, and the thrust is along the positive direction of the x -axis. It can be seen that when the robot is subjected to thrust, the CP trajectory along the x -axis direction has a significant deviation, but it will not cause the CP tracking divergence. After

the thrust stops, due to the role of the optimal controller, the CP tracking error gradually decreases until converged to approximately 0; relatively speaking, the CoP tracking is good, and the corresponding joint running trajectory is less affected by thrust, which is almost the same as the effect when it is not subjected to thrust.

6. Conclusion

This paper presents the technique of the robot motion control based on the dynamic model and quadratic programming. In order to improve the walking robustness of the biped robot, this article starts with the construction of the robot motion control framework from three aspects: planning method, mathematical model, and control method. Firstly, the biped gait trajectory planning is introduced based on the divergent component of motion. Then, the dynamic control framework is discussed for biped robot, including the joint force control, friction cone contact model, and various constraints. This paper integrates these methods into the control framework and verifies the effectiveness of the control framework on physical robots. Finally, the experimental results show that the method is robust and has certain anti-interference ability.

Further research work will focus on improving the accuracy of the dynamics model, the robustness of the control method, and the verification experiments in more complex scenarios, for example, complex walking scenes (such as sand or outdoors) and greater walking disturbances, etc. The ultimate goal is for the robot to walk stably in a human environment.

Nomenclature

- CoM: Center of mass, the trajectory curve of the center of mass during the robot walking
- CoP: Center of pressure, the trajectory curve of the pressure center of the foot of the robot during the robot walking
- CP: Capture point, robot's walking trajectory based on DCM planning
- DCM: Divergent motion component, a walking strategy of biped robot
- ID: Inverse dynamics, calculate the torque required for the joints given the motion of the robot
- IK: Inverse kinematics, calculate the required joint angle for a given foot position
- QP: Quadratic programming, convex quadratic programming problem
- ZMP: Zero moment point, a criterion of walking stability of biped robot.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Leju (Shenzhen) Robotics Co., Ltd.

Supplementary Materials

The supplementary attachment is the description of the experimental platform and experimental data. (*Supplementary Materials*)

References

- [1] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Motion planning for quadrupedal locomotion: coupled planning, terrain mapping and whole-body control," 2020, <http://arxiv.org/abs/2003.05481>.
- [2] C. Yang, K. Huang, H. Cheng, Y. Li, and C.-Y. Su, "Haptic identification by ELM-controlled uncertain manipulator," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2398–2409, 2017.
- [3] C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng, and C.-Y. Su, "Finite-time convergence adaptive fuzzy control for dual-arm robot with unknown kinematics and dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 574–588, 2018.
- [4] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: an intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.
- [5] M. Hutter, C. D. Remy, M. A. Hoepflinger, and R. Siegwart, "Scarleth: design and control of a planar running robot," in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 562–567, IEEE, San Francisco, CA, USA, September 2011.
- [6] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Reactive whole-body control: dynamic mobile manipulation using a large number of actuated degrees of freedom," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 20–33, 2012.
- [7] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Integration of reactive, torque-based self-collision avoidance into a task hierarchy," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1278–1293, 2012.
- [8] 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 *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3253–3258, IEEE, Chicago, IL, USA, September 2014.
- [9] T. Koolen, J. Smith, G. Thomas et al., "Summary of team IHMC's virtual robotics challenge entry," in *Proceedings of the 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 307–314, IEEE, Atlanta, GA, USA, October 2013.
- [10] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal, "Optimal distribution of contact forces with inverse-dynamics control," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 280–298, 2013.
- [11] M. Van, X. P. Do, and M. Mavrovouniotis, "Self-tuning fuzzy pid-nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators," *ISA Transactions*, vol. 96, pp. 60–68, 2020.
- [12] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2589–2594, IEEE, Hong Kong, China, May 2014.
- [13] S. Shigemori, A. Goswami, and P. Vadakkepat, "ASIMO and humanoid robot research at honda," in *Humanoid Robotics: A Reference*, pp. 55–90, Springer, New York City, NY, USA, 2019.
- [14] J. Park, Y. Lee, M. Kim, S. Hwang, J. Jung, and J. Kim, "Human-assisted humanoid robot control," in *Robotics Research*, pp. 85–90, Springer, New York City, NY, USA, 2020.
- [15] J. Nubert, J. Kohler, V. Berenz, F. Allgower, and S. Trimpe, "Safe and fast tracking on a robot manipulator: robust mpc and neural network control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3050–3057, 2020.
- [16] S. Lyu and C. C. Cheah, "Human-robot interaction control based on a general energy shaping method," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2445–2460, 2019.
- [17] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, "Adaptive parameter estimation and control design for robot manipulators with finite-time convergence," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8112–8123, 2018.
- [18] C. Yang, J. Luo, Y. Pan, Z. Liu, and C.-Y. Su, "Personalized variable gain control with tremor attenuation for robot teleoperation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 10, pp. 1759–1770, 2018.
- [19] J. Engelsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4420–4427, IEEE, San Francisco, CA, USA, September 2011.
- [20] J. Engelsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.
- [21] T. V. Dang, K. V. Ling, and J. M. Maciejowski, "Embedded ADMM-based QP solver for MPC with polytopic constraints," in *Proceedings of the 2015 European Control Conference (ECC)*, pp. 3446–3451, IEEE, Linz, Austria, July 2015.