Control strategy of optimal distribution of feet forces for quadruped robots based on virtual model

Xinjie Chen¹, Yuegang Tan²*, Ruiya Li³, Yecheng Mei⁴

¹,²,³,⁴School of Mechanical and Electrical Engineering, Wuhan University of Technology, Wuhan, 430000, China
*Corresponding author’s e-mail: ygtan@whut.edu.cn

Abstract: In order to improve the dynamic walking performance of quadruped robot, a control method of optimizing feet forces distribution based on virtual model is proposed. In the supporting phase, the virtual model control method is applied to solve the virtual force of the torso. Combined with the gravity of the center of mass (CoM), the distribution problem between the virtual force of the CoM and the feet forces of the supporting legs is transformed into a quadratic programming (QP) problem, which is solved by Gurobi to realize the optimal distribution of the feet forces. Similarly, the virtual force of the swinging leg is solved by using the virtual model, and the joints torques of the robot's swinging legs are obtained by combining the inverse dynamics feedforward of the swinging legs. Through the simulation of quadruped robot's trot gait walking by webots and vs2019, it is verified that this method can stabilize the robot's attitude angles and body speeds near the target values. Compared with the feet forces distribution method that abandons the lateral force control, the application of this method makes the fluctuation range of the attitude angles of the robot and the ground reaction forces (GRFs) of the supporting legs smaller. It is proved that this control method can effectively improve the walking stability and robustness of the quadruped robot.

1. Introduction
Quadruped mammals in nature have a strong locomotion ability. The robot of bionic quadruped has good adaptability and movement flexibility in complex terrain because of its discrete contact with the environment and its legs have multiple degrees of freedom [1]. The locomotion control algorithm of robot is the core module that can realize flexible and stable movement. The most famous control algorithm of quadruped robot was proposed by Raibert which was based on spring loaded inverted pendulum (SLIP) [2]. In the robot’s movement process, the leg mass is ignored, and the diagonal leg and torso are regarded as a SLIP model. Ignoring the leg mass, the diagonal legs and the torso are regarded as a SLIP model, and the locomotion control of the robot is realized by controlling the forward speed, bounce height and attitude of it. Zero moment point method (ZMP) was first proposed by Vukobratovic, a Yugoslav scholar. It is the most commonly used static stability control method to ensure that the robot is under static stability conditions at every time in the process of motion according to ZMP static stability criterion [3]. The neural network control method based on the response and reflection mechanism of central pattern generator (CPG) is also widely used control method of quadruped robots [4]. The control method based on virtual model (VMC) firstly assumes the existence of virtual components on the control object, such as spring and damper, then analyzes the virtual force on the control object, and establishes the mapping relationship between virtual force and joint force through Jacobian matrix [5-7].
Handling the contact forces between the robot and the ground and reasonably distributing the joint forces is the basis for realizing the flexible and efficient movement of quadruped robot. HyQ [8-9] quadruped robot of IIT and StarETH [10] quadruped robot of Zurich University of technology used VMC to achieve better plantar interaction and compliance characteristics. In the process of robot locomotion control, how to realize the reasonable force distribution is one of the most important problems. Zhang [11] proposed that abandoned the lateral force artificially, and the plantar force distribution was obtained by applying constraint \( f_{p_y} = f_{h_y} \). E Ming Cheng et al. [12] proposed using weighted average vertical force to realize force distribution which was not reasonable in unstructured terrain. In order to realize the locomotion stably and optimal plantar force distribution of quadruped robot during dynamic locomotion, this paper transforms the distribution problem of CoM virtual force and feet forces into a QP problem, the optimal distribution of feet forces is realized by finding the optimal values.

2. Quadruped robot model

The model of quadruped robot is shown in Figure 1. The robot has 12 degrees of freedom, FL represents the front left leg of the quadruped robot (FR: the front right leg, HL: the hind left leg, HR: the hind right leg). Each leg has three active joints: adduction joint \( \theta_1 \), hip joint \( \theta_2 \) and knee joint \( \theta_3 \). The whole length of the body is \( 2(L + d_1) \), the width is \( 2W \), and the structural parameters of the legs are \( l_1, l_2, d_2 \). The specific values are shown in Table 1.

| Table 1 robot structural parameters |
|-------------------------------------|
| Parameter          | Value of parameter |
| length of half body | L / mm            |
| Width of half body  | W / mm            |
| Height of body     | H / mm            |
| body offset        | d_1 / mm          |
| hip offset         | d_2 / mm          |
| thigh link length  | l_1 / mm          |
| shin link length   | l_2 / mm          |
| whole body mass    | m / kg            |

2.1 Forward kinematics model

For the sake of description, coordinate system \( O_B \) is established at the CoM of the body. The coordinate systems \( O_i \) (\( i = 0,1,2,3,4 \)) in FR leg are established as shown in Figure 2, and coordinate system \( O_0 \) is established at the hip and coordinate system \( O_4 \) is established at the foot end. The transformation matrix from foot coordinate system to body coordinate system is calculated according to D-H method as:

Figure 1. Quadruped simulation model

Figure 2. Coordinate system of FR leg
The forward kinematics (FK) equation can be obtained from the transformation matrix as follows:

\[ \begin{bmatrix} s_{23} & -c_{23} & 0 & l_2s_{23} + l_1s_2 + d_1 + L \\ s_{c23} & -s_{c23} & -c_1 & l_2s_{c23} + l_1c_2 - d_1c_1 - W \\ -c_{c23} & c_{c23} & -s_1 & -l_2c_{c23} - l_1c_1c_2 - d_1s_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

(1)

\[ s_i = \sin(\theta_i), c_i = \cos(\theta_i), s_{23} = \sin(\theta_2 + \theta_3), c_{23} = \cos(\theta_2 + \theta_3) (i = 1, 2, 3) \]

The forward kinematics (FK) equation can be obtained from the transformation matrix as follows:

\[ P = \begin{cases} p_s \\ p_{p_s} \\ p_p \\ p_p \end{cases} = \begin{bmatrix} l_2s_{23} + l_1s_2 + d_1 + L \\ l_2s_{c23} + l_1c_2 - d_1c_1 - W \\ -l_2c_{c23} - l_1c_1c_2 - d_1s_1 \end{bmatrix} \]

(2)

According to Equation (2), the Jacobi matrix can be obtained by differentiating the joint angles with time as follows:

\[ J = \begin{bmatrix} 0 & l_1c_2 + l_2c_{23} & l_2c_{23} \\ l_2c_{c23} + l_1c_2 + d_1s_1 & -l_1s_2 - l_2s_{23} & -l_2s_{23} \\ l_2s_{c23} + l_1s_2 + d_2c_1 & l_2c_{s23} + l_1c_1s_2 & l_2c_{s23} \end{bmatrix} \]

(3)

2.2 Inverse dynamic model

The inverse dynamics model is established by Lagrange equation, and its general calculation model is as follows:

\[ \tau = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} \right) \]

(4)

In Equation (4), \( L = E_k - E_p \), \( E_k \) is the sum of the kinetic energy of the robot leg link, \( E_p \) is the sum of the potential energy of the robot leg link; \( q = (\theta_1, \theta_2, \theta_3) \) is the joint angles of the robot. The kinetic equation as shown in Equation (5) can be obtained through differential derivation calculation and arrangement, where \( M \) represents inertial parameters, \( V \) represents Coriolis and centrifugal forces, and \( G \) represents gravity.

\[ \tau_{in} = M(\ddot{\theta}) + V(\dot{\theta}, \dot{\theta}) + G \]

(5)

3. Virtual model controller

3.1 Stance phase virtual model

Figure 3. Static forces acting on the robot. The supporting forces and torques of the supporting legs are balanced with the virtual forces and gravity of the CoM of the torso.
Compared to the torso, the mass of the robot leg is very small so that it can be ignored in the virtual model, and the mass of the robot lies in its CoM. Statics analysis is performed on the supporting legs in each gait cycle of the robot in the stable motion of diagonal gait, as shown in Figure 3.

\[
\begin{bmatrix}
  {^B F} \\
  {^B T}
\end{bmatrix} = \begin{bmatrix} I & I \\ {^B P}_x & {^B P}_x \times \end{bmatrix} \begin{bmatrix} F_f \\ F_h \end{bmatrix}
\]

(6)

Where \( \begin{bmatrix} {^B F}, {^B T}\end{bmatrix}^T \) is the virtual force of the CoM, \( F_f = \begin{bmatrix} f_{Fx}, f_{Ty}, f_{Fz} \end{bmatrix}^T \) and \( F_h = \begin{bmatrix} f_{Hx}, f_{Hy}, f_{Hz} \end{bmatrix}^T \) are the force of front and rear the support legs, and \( {^B P}_x \) is the position of the \( i^{th} \) supporting leg in the body coordinate system. The relationship between them is shown in Equation (6).

Based on the principle of VMC, the external force on the centroid of the robot's torso is equivalent to a virtual system composed of spring-damping. The virtual spring-damping system is used to follow the desired three-dimensional locomotion and attitude of the robot. As shown in FIG. 4, a virtual force model with six degrees of freedom is established in the CoM.

\[
\begin{bmatrix}
  {^B F}_x = K_x (x_d - x) + B_x (\dot{x}_d - \dot{x}) \\
  {^B F}_y = K_y (y_d - y) + B_y (\dot{y}_d - \dot{y}) \\
  {^B F}_z = K_z (z_d - z) + B_z (\dot{z}_d - \dot{z}) \\
  {^B T}_x = K_\alpha (\alpha_d - \alpha) + B_\alpha (\dot{\alpha}_d - \dot{\alpha}) \\
  {^B T}_y = K_\beta (\beta_d - \beta) + B_\beta (\dot{\beta}_d - \dot{\beta}) \\
  {^B T}_z = K_\gamma (\gamma_d - \gamma) + B_\gamma (\dot{\gamma}_d - \dot{\gamma})
\end{bmatrix}
\]

(7)

In Equation (7), \( \begin{bmatrix} P, \psi \end{bmatrix}^T = \begin{bmatrix} x_d, y_d, z_d, \alpha_d, \beta_d, \gamma_d \end{bmatrix}^T \) are the expected position and attitude angles of the CoM where \( \begin{bmatrix} P, \psi \end{bmatrix}^T = \begin{bmatrix} x, y, z, \alpha, \beta, \gamma \end{bmatrix}^T \) are the actual position and attitude angles. In addition, the robot is also constrained by its own gravity, combining Equation (6) ~ (7) and considering the gravity component, it can be obtained:

\[
\begin{bmatrix}
  {^B F}_x - mg \sin \theta \\
  {^B F}_y - mg \cos \theta \\
  T_x \\
  T_y \\
  T_z \\
  B
\end{bmatrix} = \begin{bmatrix}
  1 & 0 & 0 & 1 & 0 & 0 \\
  0 & 1 & 0 & 0 & 1 & 0 \\
  0 & 0 & 1 & 0 & 0 & 1 \\
  0 & -z_f & y_f & 0 & -z_H & y_H \\
  z_f & 0 & -x_f & z_H & 0 & -x_H \\
  -y_f & x_f & 0 & -y_H & x_H & 0
\end{bmatrix} \begin{bmatrix} f_{Fx} \\ f_{Fy} \\ f_{Fz} \\ f_{Hx} \\ f_{Hy} \\ f_{Hz} \end{bmatrix}
\]

(8)
Since \( \text{rank}(A)=5<6 \), there is no unique solution for the distribution problem between the virtual force of the CoM and the robot’s support feet. To solve this problem, this paper transformed the problem of feet forces distribution into a QP problem, so as to realize the aim of optimal distribution of feet forces.

Under the condition that the plantar force distribution satisfies Equation 8, the output forces should be as smaller as possible. In order to prevent the foot from slipping, friction cone constraint is added to make the feet forces not exceed the scope of static friction constraint. At the same time, add a constraint to ensure that the vertical forces are greater than 0.

\[
\min \quad f = (Ax - B)^T S (Ax - B) + x^T W x
\]
\[
\text{s.t.} \quad |f_{ix}| \leq \mu f_{iz}, \quad |f_{iy}| \leq \mu f_{iz}, \quad f_{iz} \geq 0
\]

Where \( S \in \mathbb{R}^{6 \times 6} \) is a weight diagonal matrix, and \( W \in \mathbb{R}^{6 \times 6} \) denotes the penalty diagonal matrix. They are used to make the output follow the target virtual force and prevent excessive virtual force. \( \mu \) represents the friction coefficient. Since the forces are interactive, the output force of the robot support foot is equal to the support force of the ground to the support foot, and the direction is opposite. Combined the Jacobian matrix calculated in Equation (3), the joint control torques in stance phase as:

\[
\tau_{st} = -J^T f
\]

### 3.2 Support phase virtual model

Similarly, virtual model control is still used to control the swinging legs. As shown in FIG. 5, the forces on the swinging leg is equivalent to a virtual system composed of spring damping in three directions.

![Figure 5. Swing phase virtual model](image)

**Figure 5. Swing phase virtual model**

**Figure 6. The foot trajectory generated**

The virtual force of swinging legs:

\[
\begin{bmatrix}
    f_{x}^{sw} \\
    f_{y}^{sw} \\
    f_{z}^{sw}
\end{bmatrix} =
\begin{bmatrix}
    k_x (x'^w - x^w) + k_{xv} (\ddot{x}^w - \dot{x}^w) \\
    k_y (y'^w - y^w) + k_{yv} (\ddot{y}^w - \dot{y}^w) \\
    k_z (z'^w - z^w) + k_{zv} (\ddot{z}^w - \dot{z}^w)
\end{bmatrix}
\]

The virtual forces of swinging legs are mapped to each joint by Jacobi matrix, and the torques can be obtained as follows:

\[
\tau_{sw} = J^T F^{sw}
\]

Combined with inverse dynamics feedforward torques given in Equation (5), the torques of swinging leg joints can be obtained as follows:

\[
\tau_{sw} = \tau_{ID} + \tau_{sw}^{mc}
\]

### 3.3 Swing trajectory generator

In this paper, the trajectory of the swinging leg is planned in the foot end coordinate system along the x, y and z directions respectively. The trajectory planning in the x direction realizes the forward motion of
the robot, while in the y direction realizes the lateral swing motion and in the z direction realizes the vertical step of the swinging leg.

$$x_{sw}(t) = \begin{cases} 
\frac{6L_x}{T_x^5}(t-a_xT)^5 - \frac{15L_x}{T_x^4}(t-a_xT)^4 + \frac{10L_x}{T_x^3}(t-a_xT)^3, & 0 \leq t < a_xT \\
L_x & a_xT \leq t < a_xT \\
L_x & t \geq a_xT 
\end{cases} \quad (14)$$

$$y_{sw}(t) = \begin{cases} 
\frac{6L_y}{T_y^5}(t-a_yT)^5 - \frac{15L_y}{T_y^4}(t-a_yT)^4 + \frac{10L_y}{T_y^3}(t-a_yT)^3, & 0 \leq t < a_yT \\
L_y & a_yT \leq t < a_yT \\
L_y & t \geq a_yT 
\end{cases} \quad (15)$$

$$z_{sw}(t) = H \left( \frac{16t^4}{T^4} \frac{32t^3}{T^3} + \frac{16t^2}{T^2} \right) + z_0 \quad (16)$$

Where $T$ is the time period of the swinging legs, $T_x = T_y = (a_x - a_y)T$, and $(x_0, y_0, z_0)$ is the initial position of the swinging leg, $L_x, L_y$ respectively is the stride length of the swinging leg in the x direction and y direction, and $H$ is the stride height of the swinging leg. The foot trajectory of swinging leg in the X-Z plane is shown in FIG. 6.

3.4 Overall control architecture of robot

As shown in FIG. 7, the overall movement phase of the robot is divided into stance phase and swing phase, and the two phases are switched alternately by the state machine. In the stance phase, the virtual force distribution between the body centroid and plantar force is transformed into a QP problem, and the real-time optimal force distribution is realized through Gurobi library to find the optimal values. In the swing phase, inverse dynamic feedforward and VMC are combined to achieve smooth trajectory following. FK represents forward kinematics, ID represents inverse dynamics, and force controller is a PD controller.
Figure 8. Trot gait sequence diagram. The black squares represent landing state, while the white squares represent leaving ground state.

Gait time sequence refers to the proportion of time in the swing phase and the standing phase for each leg in a complete gait cycle. In nature, the diagonal gait is the most common gait of quadrupeds. When a quadruped robot moves in a trot gait, one diagonal leg touch the ground and the other diagonal leg swing and the stance phase time is equal to the swing phase time. The phase switching is carried out by the state machine to realize the alternating forward movement of the diagonal legs of the robot.

4. Simulation and results

4.1 Walking simulation of quadruped

Walking is an effective method to verify the diagonal gait control algorithm of quadruped robot. Make the target speed of quadruped robot walking from 0.2m/s forward speed, lateral speed and rotation speed to 0.

Table 2  Robot simulation parameters

| Parameter                  | Value of parameter |
|----------------------------|--------------------|
| $K_s, K_y, K_z$ (Nm)       | 0, 0, 8000         |
| $K_n, K_p, K_r$ (Nm / rad) | 2000, 10000, 200   |
| $B_s, B_y, B_z$ (Ns / m)   | 200, 0.01, 800     |
| $B_n, B_p, B_r$ (Ns / m)   | 120, 50, 0         |
| $k_s, k_y, k_z$ (N / m)    | 100, 8000, 1000    |
| $k_n, k_p, k_r$ (Ns / m)   | 100, 100, 1000     |
| weight matrix $S$          | diag (2.5, 2.5, 50, 5, 5, 5) |
| penalty matrix $W$         | diag (0.1, 0.1, 0.1, 0.1, 0.1, 0.1) |

The robot is controlled by using VMC and optimal feet forces distribution. In Table 2, there are simulation parameters in webots software. After the simulation starts, record the changes of body’s speed as well as pitch angle.

Figure 9. Screenshot of robot walking simulation process in Webots

Figure 10. Curve of forward speed and pitch value during walking

The position and posture of the quadruped robot within 0-6 seconds were captured as shown in FIG. 9, which reveals that the robot could move smoothly. Figure 10 shows the target velocity tracking curve
of the robot and the change curve of the body’s pitch angle. It can be seen from the figure that the forward velocity of the robot fluctuates slightly within the range of 0.2m/s after starting time, which can track the desired velocity well, and the pitch fluctuation is also very small.

4.2 Quadruped status contrast before and after optimization

The above simulation results verify the stability of the proposed method and the feasibility of the optimal foot force distribution algorithm based on virtual model. In order to test the optimization effect of the force distribution method in the locomotion process of quadruped robot, a group of comparative experiments are carried out. Under the same step frequency and step height, compared with the foot force distribution method without lateral force, the advantages of this force distribution method are explained by the foot forces as well as joint torques feedback of the FR leg, and attitude angles of the body.

In FIG. 11 and 12, the curves are torques feedback of the FR leg joints and attitude angles, and the blue line is the distribution curve of force control method with optimization while the red line without (the same below). It’s obvious that the torque of the robot leg joints as well as fluctuation ranges are significantly reduced after using the optimal feet forces distribution control algorithm. Compared with the control algorithm without optimization, the roll, pitch and yaw and the fluctuation ranges are significantly smaller, which significantly improves the stability of the robot.
FIG. 13 shows the GRFs change curve of the FR leg. Compared with the control algorithm without optimization, the fluctuation of forces becomes smaller in the X and Y directions, which means that the robot walks more stably. And in Z direction, the GRFs reduced by about 50%, which means that the impact of the ground on the vertical direction is greatly reduced and the robot walks more smoothly.

5. Conclusion
In this paper, an optimal force distribution control method based on virtual model is proposed for the quadruped robot in trot gait. The force distribution problem that cannot be solved directly is transformed into a constrained QP problem, which can be solved through Gurobi C++ library in real time. The optimized force distribution method is applied to the locomotion control of the robot, and the robot can walk stably forward. From the perspective of plantar force impact, joint torques and robot posture change during the movement, the proposed method plays a good role in stabilizing the motion of the quadruped robot and reducing the impact force compared with the force distribution method without lateral force.

References
[1] Yang, J.J., Sun, H., Wang, C.H., Chen, X.D. (2019) Review of quadruped robot research. [J]. Navigation Positioning and Timing., 6 (05): 61-73.
[2] Raibert, M., Blanke, K., Nelson, G., et al. (2008) BigDog, the rough terrain quadruped robot. [J]. IFAC Proceedings Volumes., 41(2): 10823-10825.
[3] VUKOBRATOVIĆ M, STEPANENKO J. (1972) On the stability of anthropomorphic systems. [J]. International journal of humanoid robotics., 15(1-2):1-37.
[4] Zhang, Y., Guo, Z.W., Chen, D.J. Wang, B.R. (2018) Gait control of pneumatic muscle quadruped robot based on Kimura oscillator and virtual model. [J]. Journal of ordnance industry., 39 (07):1411-1418.
[5] Pratt, J., Chew, C., Torres, A., Dilworth, P., and Pratt, G. (2001) Virtual model control: An intuitive approach for bipedal locomotion. [J]. International Journal of Robotics Research., 20: 129-143.
[6] Desai, R. Geyer, H. and Hodgins, J.K. (2014) Virtual model control for dynamic lateral balance. [C]. IEEE-RAS Int. Conf. Humanoid Robots., Nov. 2014, pp. 856–861
[7] Zhang G T, Rong X W, Hui C, et al. (2016) Torso motion control and toe trajectory generation of a trotting quadruped robot based on virtual model control. [J]. Advanced Robotics., 30(4): 284-297.
[8] Boaventura, T., Buchli, J., Semini, C., et al. (2015) Model-based hydraulic impedance control for dynamic robots. [J]. IEEE Transactions on Robotics, 31(6): 1324-1336.
[9] Winkler, A. W., Mastalli, C., Havoutis, I., et al. (2015) Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain. In: IEEE International Conference on Robotics and Automation. Washington. 5148-5154.
[10] Gehring, C., Coros, S., Hutter, M., et al. (2013) Control of dynamic gaits for a quadrupedal robot. In: IEEE International Conference on Robotics and Automation. Karlsruhe. 3287-3292.
[11] Zhang, G.T., Rong X.W, Li, Y.B., Chai, H., Li, B. (2016) Gait control method of quadruped robot diagonal trot based on virtual model. [J]. Robot., 38 (01): 64-74
[12] E, M.C., Liu, H., Zhang, X.L., et al. (2014) Compliant gait generation for a quadruped bionic robot walking on rough terrains. [J]. Robot., 36(5): 584-591.