IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 14, 2020, accepted July 28, 2020, date of publication July 31, 2020, date of current version August 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013434

Virtual Model Control for Quadruped Robots

GUANGRONG CHEN[1], SHENG GUO[1], BOWEN HOU[2,3], AND JUNZHENG WANG[4]

[1]Robotics Research Center, Beijing Jiaotong University, Beijing 100044, China
[2]Beijing Engineering and Technology Research Center of Rail Transit Line Safety and Disaster Prevention, Beijing 100044, China
[3]School of Civil Engineering, Beijing Jiaotong University, Beijing 100044, China
[4]State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Guangrong Chen (grchen@bjtu.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2019JBM051, in part by the Beijing Engineering and Technology Research Center of Rail Transit Line Safety and Disaster Prevention Open Foundation for Research under Grant RRC201701, in part by the Beijing Natural Science Foundation under Grant 3204051, and in part by the Science and Technology Research Project of China Railway Corporation under Grant P2018G047.

ABSTRACT Virtual model control is a motion control framework that uses virtual components to create virtual forces/torques, which are actually generated by joint actuators when the virtual components interact with robot systems. Firstly, this paper employs virtual model control to do a dynamic balance control of whole body of quadruped robots' trot gait in a bottom controller. In each leg, there exists a designed swing phase virtual model control and a stance phase counterparts. In the whole body, virtual model control is utilized to achieve a attitude control containing roll, pitch and yaw. In the attitude control, a forces/torques distribution method between two stance legs is pre-investigated. In a high-level implemented controller, an intuitive velocity control approach proposed by Raibert is applied for the locomotion of quadruped robots. Secondly, an anti-disturbance control, which contains compensating gravity, adjusting step length, adjusting swing trajectory, adjusting attitude, and adjusting virtual forces/torques, is investigated to improve the robustness, terrain adaptability, and dynamic balance performance of quadrupedal locomotion. Thirdly, a trajectory tracking control method based on an intuitive velocity control is addressed through considering four factors: terrain complexity index, curvature radius of given trajectory, distance to terminal, and maximum velocity of quadruped robots. Finally, simulations validate the effectiveness of proposed controllers.

INDEX TERMS Virtual model control, dynamic balance control, anti-disturbance control, trajectory tracking control, quadruped robots.

I. INTRODUCTION

Nowadays, legged robots have been a research hotspot and gained various attention [1]–[3]. Quadruped robots are hard to control because they are nonlinear, unstable, and multi-input multi-output (MIMO); interact with environment; and behave time variant and switch dynamics (during support exchange). Many control methods have been investigated to handle the dynamic balance problem of legged robots [4], [5]. Zero moment point (**ZMP**) has been a popular method to analyze the stability of legged robots [6], [7]. Research group in Italian Institute of Technology (IIT) proposed a feedback/feedforward control structure by using a **inverse dynamics** (ID) approach, which was utilized in a quadruped robot HyQ [8]. Boston Dynamics employed the idea of spring loaded inverted pendulum (**SLIP**) model to guide the locomotion of Bigdog [9]. Researchers in University of Michigan proposed a hybrid zero dynamics (**HZD**) method for planar biped walkers [10] and it was employed to prove the local stability of PD controlled bipedal walking robots [11], [12]. Similarly, a method of virtual constraints (**VC**) was combined with model predictive control (MPC) and quadratic programming (QP) to develop a hierarchical nonlinear control algorithm for stable quadrupedal locomotion patterns [13]. Kitani, Makoto et al proposed a **CPG**-based (central pattern generator, CPG) method to do the suppression of roll oscillation in turning of quadruped robot by asymmetric amplification of CPG output waveform [14]. Besides, a hierarchical controller [15], [16] was utilized to do quadrupedal locomotion in Anymal [17] and MIT Cheetah [18], [19]. Neural Network-based method was also investigated to control robots and manipulators [20].

Robots typically have an individual actuator at each joint which can result in a nonintuitive and difficult control problem. In this paper we present a control method in which the real joint actuators are used to mimic virtual actuators

The associate editor coordinating the review of this manuscript and approving it for publication was Yangmin Li.

which can be more intuitive and hence make the control problem more straightforward [21]. This intuitive control scheme is called virtual model control (**VMC**) [22]. VMC does not need a robot dynamics model [23], it only uses the Jacobian matrix to calculate joint/control torques. VMC also does not need a hierarchical controller [24], [25], robot locomotion (velocity & attitude control) can be achieved directly by joint torques control. Actually, VMC had been proposed by J. Pratt as an intuitive approach for bipedal locomotion [26]. Researchers in IIT extended the applications of VMC to quadrupedal locomotion [24], [27]. But they just mentioned VMC in dynamic balance control and did not give out the concrete realizations. In literature [28], Xie *et al.* deduced the detailed control laws of an intuitive approach for quadruped robot trotting based on VMC. Desai *et al.* gave out the VMC for dynamic lateral balance of quadruped robots as well [25]. Even though normal quadrupedal trot walking could be implemented and push recovery could be guaranteed under a little bit external disturbance in [25], [28], extra anti-disturbance control actions, such as gravity compensation, step length adjustment, swing trajectory adjustment, attitude adjustment, and virtual forces/torques adjustment, were still not studied to improve the robustness and terrain adaptability of quadrupedal locomotion. Zhang *et al.* proposed a quadruped robot adaptive control in trotting gait walking on slopes by using the estimation and feedback of ground slope degree [29]. However, quadruped robots should have the ability to walk on more rough terrains in real world. How to achieve this kind of dynamic balance based on VMC is still an essential issue to be addressed.

Virtual model control is a motion control framework that uses simulations of virtual components to generate desired joint torques. These joint torques create the same effect that the virtual components would have created, had they existed, thereby creating the illusion that the simulated components are connected to the real robot. Such components can include simple springs, dampers, dashpots, masses, latches, bearings, nonlinear potential and dissipative fields, or any other imaginable component [26]. Virtual model control borrows ideas from virtual reality, hybrid position-force control [30], stiffness control [31], impedance control [32]–[34], and the operational space formulation [15].

The advantages of virtual model control contain: 1) intuitive and straightforward; 2) low computational requirement without inverse dynamics; 3) easy implement and high efficiency; 4) compliance characteristic; 5) easy to combine with a high-level controller. The disadvantages of virtual model control contain: 1) low control performance in force/position control compared with high tracking performance control [35], [36]; 2) response lag in force/position control; 3) forces/torques distribution problem in stance phase of multi legs; 4) disturbance caused by switch control between swing and stance phase.

As for the disadvantages mentioned above, the tracking error and response lag are allowed to some extent since quadruped robots in dynamic gait have to keep switching trot gait to ensure their dynamic balance without any stop. The forces/torques distribution problem is addressed by the proposed control state machine and forces/torques distribution method. And, the disturbance problem is issued by the proposed anti-disturbance control. Besides, a trajectory tracking control [37] based on an intuitive velocity control of robot base origin (RBO) of quadruped robots, is proposed as a high-level implemented controller. Also, the trajectory tracking control considers four factors: terrain complexity index, curvature radius of given trajectory, distance to terminal, and maximum velocity of quadruped robots.

In this paper, the virtual model control for quadruped robots is proposed. In the control framework, virtual components that have physical counterparts such as mechanical spring and damper, are placed at strategic locations within the robot or between the robot and the environment. Physical intuition is needed to place these virtual components. Once the placement is done, the interactions between these components and the robots automatically generate the desired torques or forces at the actuators. No dynamic model of robots is necessary in the control algorithm. This approach is applied to quadruped robots. The main contributions can be concluded as follows:

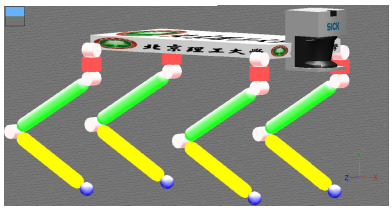
- A dynamic balance control based on virtual model control, from each leg to the whole body, is proposed for quadruped robots in the trot gait.
- An anti-disturbance control is proposed for enhancing the robustness, terrain adaptability, and dynamic balance performance of quadrupedal locomotion.
- A trajectory tracking control method, which is based on the velocity planning and control and added on the implemented VMC, is proposed with four important factors.
- The proposed controllers based on VMC are validated by co-simulations and comparative simulations show the proposed controllers costs less computing time.

II. MODEL

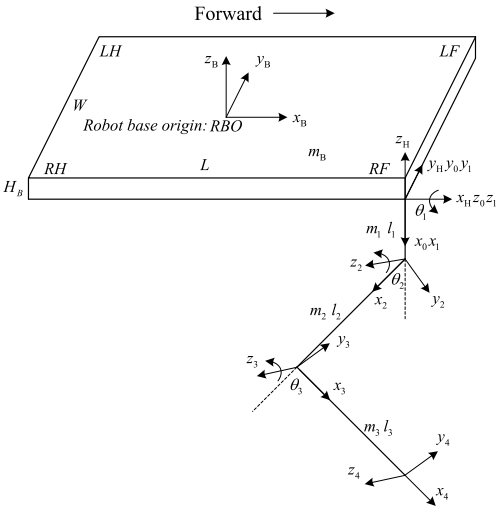
The simulation model set up by WEBOTS software and the related coordinate systems of quadruped robots are shown in FIGURE 1(a) and 1(b), respectively. There are three degrees of freedom (DoF) in each leg: 2 DoFs in hip joint (roll & pitch) and 1 DoF in knee joint (pitch). The front and hind legs are all elbow style so that there are not singular and redundancy phenomenon. Meanwhile, all the legs share the same Jacobian matrix.

In FIGURE 1(b), the coordinates of foot of RF leg under body coordinate system ${}^{B}p_{FOOT} = \begin{bmatrix} x & y & z \end{bmatrix}^{T}$ can be written as

$$ {}^{B}p_{FOOT} = FK\,(q) = \begin{bmatrix} \frac{L}{2} + l_2 s_2 + l_3 s_{23} \\ -\frac{W}{2} + l_1 s_1 + l_2 s_1 c_2 + l_3 s_1 c_{23} \\ -\frac{H_B}{2} - l_1 c_1 - l_2 c_1 c_2 - l_3 c_1 c_{23} \end{bmatrix} \tag{1} $$



(a) Simulation model



(b) Coordinate system of RF leg

FIGURE 1. System model.

where s_i, c_i are $\sin\theta_i, \cos\theta_i$, respectively. s_{ij}, c_{ij} are $\sin(\theta_i+\theta_j), \cos(\theta_i+\theta_j)$, respectively. $q = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3\end{bmatrix}^T$ is the joint angle vector of RF leg. FK is the forward kinematics.

According to the geometric relationship of RF leg, the inverse kinematics IK could be written as (2), shown at the bottom of the page.

Differentiate (1), the Jacobian matrix for each leg could be obtained as (3), shown at the bottom of the page.

Lagrange method can be applied to derive the robot dynamics as follows

$$\begin{aligned} Q &= \begin{bmatrix}\ddot{q} & \dot{q} & q\end{bmatrix}^T = FD(\tau) \\ \tau &= \begin{bmatrix}\tau_1 & \tau_2 & \tau_3\end{bmatrix}^T = ID(Q) \end{aligned} \tag{4}$$

where FD and ID are forward dynamics and inverse dynamics, respectively.

Jacobian matrix, FK and IK are easy to obtain, while FD and ID are difficult. Exactly, VMC does not consider dynamics (FD or ID). This is one reason why VMC is an intuitive approach.

III. DYNAMIC BALANCE CONTROL BASED ON VIRTUAL MODEL CONTROL

A. VIRTUAL MODEL CONTROL

In general, there are two classical dynamic model controls for robots: one is in joint space and the other is in cartesian space [38], as shown in FIGURE 2(a) and FIGURE 2(b), respectively.

In FIGURE 2(a), it yields

$$\begin{aligned} \tau_f &= M\ddot{q}_r + C\dot{q} + G \\ \tau_r &= K_q\left((q_r - q) + D_q\left(\dot{q}_r - \dot{q}\right)\right) \\ \tau &= \tau_f + \tau_r \end{aligned} \tag{5}$$

In FIGURE 2(b), it yields

$$\begin{aligned} \tau_f &= M\ddot{q}_r + C\dot{q} + G \\ \tau_r &= J^T F_r = J^T K_x\left((x_r - x) + D_x\left(\dot{x}_r - \dot{x}\right)\right) \\ \tau &= \tau_f + \tau_r \end{aligned} \tag{6}$$

$$q = IK\left({}^B p_{FOOT}\right) = \begin{bmatrix} \tan^{-1}\left(\dfrac{y+\frac{W}{2}}{z+\frac{H_B}{2}}\right) \\ \tan^{-1}\dfrac{x-\frac{L}{2}}{\sqrt{\left(y+\frac{W}{2}\right)^2+\left(z+\frac{H_B}{2}\right)^2-l_1}} - \xi \\ \dfrac{l_2+l_3}{l_3}\xi \end{bmatrix}$$

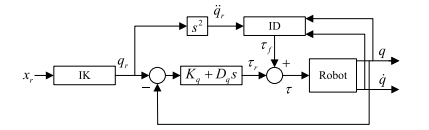
$$\xi = \cos^{-1}\frac{l_2^2 + \left(x-\frac{L}{2}\right)^2 + \left(y+\frac{W}{2}-l_1\sin\theta_1\right)^2 + \left(z+\frac{H_B}{2}+l_1\sin\theta_1\right)^2 - l_3^2}{2l_2\sqrt{\left(x-\frac{L}{2}\right)^2 + \left(y+\frac{W}{2}-l_1\sin\theta_1\right)^2 + \left(z+\frac{H_B}{2}+l_1\sin\theta_1\right)^2}} \tag{2}$$

$${}^B\dot{p}_{FOOT} = J\dot{q}$$

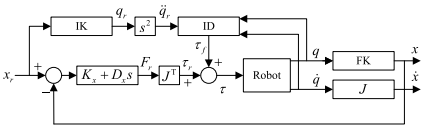
$$J = \begin{bmatrix} 0 & l_2c_2 + l_3c_{23} & l_3c_{23} \\ l_1c_1 + l_2c_1c_2 + l_3c_1c_{23} & -l_2s_1s_2 - l_3s_1s_{23} & -l_3s_1s_{23} \\ l_1s_1 + l_2s_1c_2 + l_3s_1c_{23} & l_2c_1s_2 + l_3c_1s_{23} & l_3c_1s_{23} \end{bmatrix} \tag{3}$$

where K_q, D_q are the stiffness and damping matrixes in joint space, respectively. K_x, D_x are the stiffness and damping matrixes in cartesian space, respectively. τ_f, τ_r are the forward and feedback torques, respectively.

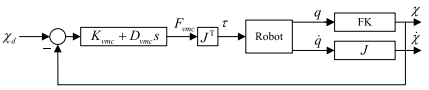
Easy to know that both two classical dynamic model controls need robot dynamics model. And the more accurate dynamics, the better control performance. Compared with VMC, VMC is simpler, as shown in FIGURE 2(c), where K_χ, D_χ are the virtual stiffness and damping matrixes. Only FK and Jacobian matrix are used without considering dynamics model of robot. Thus, VMC is employed in this paper.



(a) Dynamic model control in joint space



(b) Dynamic model control in cartesian space



(c) Virtual model control

FIGURE 2. **Three control methods of robot.**

Actually, the virtual forces/torques produced by virtual components do not exist in real world. They are generated by joint actuators. This paper employed two simple virtual components: spring and damper. Then, the virtual forces/torques can be written as

$$F_{vmc} = K_{vmc}\left(\chi_d - \chi\right) + D_{vmc}\left(\dot{\chi}_d - \dot{\chi}\right) \tag{7}$$

where K_{vmc}, D_{vmc} are the virtual coefficients of spring and damper, respectively. That are stiffness and damping coefficients. χ could be a joint angle or a position of end-effector. $\chi, \dot{\chi}$ are the actual displacement and velocity of χ, respectively. $\chi_d, \dot{\chi}_d$ are the desired displacement and velocity of χ, respectively.

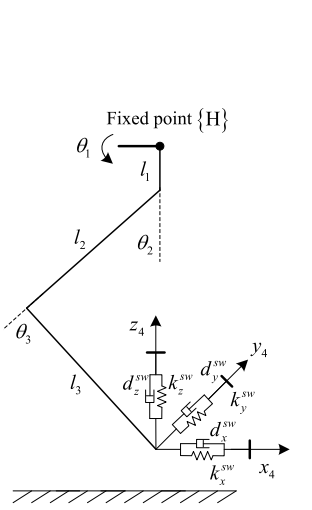
The virtual components are set to the RBO or foot end-effector to drive it move along the desired gait trajectory as far as possible. If the objective is to control the robot reach the given movement velocity, especially in the horizontal direction, only a damper is needed in (7) ($K_{vmc} = 0$). If the objective is to control the robot go along the given trajectory, such as a given foot trajectory or RBO height, both spring and damper are needed. From (7), it can be found that the values of virtual forces/torques are related to the high-level controller, such as controlling a trajectory of foot end-effector, a desired RBO trajectory or a desired horizontal movement velocity of robot. Also, adding some compensation items to the right side of (7) to improve the performance of quadruped locomotion is possible. Thus, many complex motions of quadruped robots from a upper controller can be easily implemented. This is another reason why virtual model control is called as an intuitive control.

Take one leg of quadruped robots into consideration, the virtual model control should be analyzed in two phases: swing and stance. Note that virtual components (spring-damper) are added to foot in swing phase and hip joint in stance phase, respectively, as shown in FIGURE 3. As for each leg, in stance phase, the foot can be seen as a fixed point on the ground compared with the origin of hip joint coordinate system {H}. And then we can assume that there exist virtual forces driving the origin of hip joint coordinate system {H} to make the robot move forward. Similarly, in swing phase, the origin of hip joint coordinate system {H} can be seen as a fixed point on the body trunk compared with foot. And then we can assume that there exist virtual forces driving the foot to accomplish the planned gait trajectory.

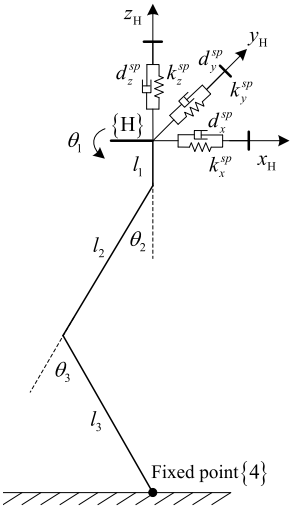
1) SWING PHASE

In swing phase, as shown in FIGURE 3(a), the origin of hip joint coordinate system {H} is set as a fixed point. And three virtual spring-dampers are connected between the origin of foot coordinate system {4} and environment. Combining (7) and FIGURE 3(a), the virtual forces produced by the three virtual spring-dampers in swing phase can be written as

$$F_{vmc}^{sw} = \begin{bmatrix} F_x^{sw} \\ F_y^{sw} \\ F_z^{sw} \end{bmatrix}$$



(a) Virtual model control in swing phase: the joint coordinate system {H} is fixed; the foot end effector is connected to the environment with spring-dampers



(b) Virtual model control in stance phase: the foot end effector {4} is fixed; the joint coordinate system {H} is connected to the environment with spring-dampers

FIGURE 3. **Virtual model control for legs.**

$$= \begin{bmatrix} k_x^{sw} & 0 & 0 \\ 0 & k_y^{sw} & 0 \\ 0 & 0 & k_z^{sw} \end{bmatrix} \begin{bmatrix} x_d^{sw} - x^{sw} \\ y_d^{sw} - y^{sw} \\ z_d^{sw} - z^{sw} \end{bmatrix} + \begin{bmatrix} d_x^{sw} & 0 & 0 \\ 0 & d_y^{sw} & 0 \\ 0 & 0 & d_z^{sw} \end{bmatrix} \begin{bmatrix} \dot{x}_d^{sw} - \dot{x}^{sw} \\ \dot{y}_d^{sw} - \dot{y}^{sw} \\ \dot{z}_d^{sw} - \dot{z}^{sw} \end{bmatrix} \quad (8)$$

where x^{sw}, y^{sw}, z^{sw} and $\dot{x}^{sw}, \dot{y}^{sw}, \dot{z}^{sw}$ the actual displacements and velocities of foot under the hip joint coordinate system {H}, respectively. $x_d^{sw}, y_d^{sw}, z_d^{sw}$ and $\dot{x}_d^{sw}, \dot{y}_d^{sw}, \dot{z}_d^{sw}$ are the desired counterparts, respectively. $k_x^{sw}, k_y^{sw}, k_z^{sw}$ and $d_x^{sw}, d_y^{sw}, d_z^{sw}$ are the stiffness and damping coefficients of three virtual spring-dampers, respectively. Then, the joint control torques in swing phase can be obtained as

$$\tau^{sw} = J^T F_{vmc}^{sw} \quad (9)$$

2) STANCE PHASE

Similarly, in stance phase, as shown in FIGURE 3(b), the origin of foot coordinate system {4} is set as a fixed point. And three virtual spring-dampers are connected between the origin of hip joint coordinate system {H} and environment. Combining (7) and FIGURE 3(b), the virtual forces produced by the three virtual spring-dampers in stance phase can be written as

$$F_{vmc}^{sp} = \begin{bmatrix} F_x^{sp} \\ F_y^{sp} \\ F_z^{sp} \end{bmatrix} = \begin{bmatrix} k_x^{sp} & 0 & 0 \\ 0 & k_y^{sp} & 0 \\ 0 & 0 & k_z^{sp} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ z_d^{sp} - z^{sp} \end{bmatrix} + \begin{bmatrix} d_x^{sp} & 0 & 0 \\ 0 & d_y^{sp} & 0 \\ 0 & 0 & d_z^{sp} \end{bmatrix} \begin{bmatrix} \dot{x}_d^{sp} - \dot{x}^{sp} \\ \dot{y}_d^{sp} - \dot{y}^{sp} \\ \dot{z}_d^{sp} - \dot{z}^{sp} \end{bmatrix} \quad (10)$$

where x^{sp}, y^{sp}, z^{sp} and $\dot{x}^{sp}, \dot{y}^{sp}, \dot{z}^{sp}$ are the actual displacements and velocities of origin of hip joint coordinate system {H} under foot coordinate system {4}, respectively. $x_d^{sp}, y_d^{sp}, z_d^{sp}$ and $\dot{x}_d^{sp}, \dot{y}_d^{sp}, \dot{z}_d^{sp}$ are the desired counterparts, respectively. $k_x^{sp}, k_y^{sp}, k_z^{sp}$ and $d_x^{sp}, d_y^{sp}, d_z^{sp}$ are the stiffness and damping coefficients of three virtual spring-dampers, respectively. Note that if the robot trunk moves stable without too much oscillations, all these displacements and velocities in (10) above equal to the RBO counterparts. Therefore, it is necessary to control the robot trunk attitude well. And the stable trunk movement is guaranteed by the attitude control of robot, which is discussed in the dynamic balance control of quadruped robots later.

In stance phase, the virtual forces F_{vmc}^{sp} are added on the origin of hip joint coordinate system {H}. It is equivalent to that there exist virtual forces $-F_{vmc}^{sp}$ adding on the foot since force is mutual. Or assume that the virtual forces $-F_{vmc}^{sp}$ are added on the foot, then their counterforces F_{vmc}^{sp} are added on the origin of hip joint coordinate system {H}. Since Jacobian matrix is described from the origin of hip joint coordinate system {H} to foot, the joint control torques in stance phase can be obtained as follow

$$\tau^{sp} = -J^T F_{vmc}^{sp} \quad (11)$$

Equations (8)(9) and (10)(11) are the virtual model control approach in swing and stance phase, respectively. Intuitively, virtual model control is a guidance mechanism, which does not require the robot to complete a certain movement or reach a certain position accurately. Compared with the control method based on dynamics model, virtual model control is simpler, easier to achieve, more efficient and stable as well. The virtual forces are calculated directly based on the instructions (displacement, velocity and attitude) of upper controller. Then joint control torques are generated through the corresponding Jacobian matrix (3). In sum, virtual model control is designed without considering complex dynamic models, inverse kinematics solutions and high precisions of position tracking.

B. DYNAMIC BALANCE CONTROL

To make the trunks of quadruped robots move stable in trot gait, dynamic balance control for the robot attitude is addressed. Note that dynamic balance control is mainly achieved by two stance legs while two swing legs movement are taken as an internal disturbance.

1) CONTROL STATE MACHINE

In order to solve the disturbance problem caused by the switch control between swing and stance phase and the complexity problem of forces/torques distribution of quadruped robots with multi stance legs, a control state machine for quadruped robots is proposed as shown in FIGURE 4. When quadruped robots are in state S_1, S_2 or S_3, legs RF, LH are in stance phase control and legs LF, RH are in swing phase control. When quadruped robots are in state S_1', S_2' or S_3', legs LF, RH are in stance phase control and legs RF, LH are in swing phase control. S_2, S_2' are two main control states. In state S_2 or S_2', even though one (S_1/S_1') or two (S_0/S_0') legs in stance phase takes/take off, or one leg (S_3/S_3') in swing phase touches down, quadruped robots are still controlled in state S_2 or S_2' without any change. Only when quadruped robots are in state S_4, two legs switch their stance phase controls to swing phase controls, and the other two legs switch their swing phase control to stance phase controls. As thus, the virtual forces/torques are only distributed between two diagonal legs in stance phase. As for the other two legs in swing phase, attitude control and proper control parameters can be employed to reject their swing disturbance to the body trunk of robot.

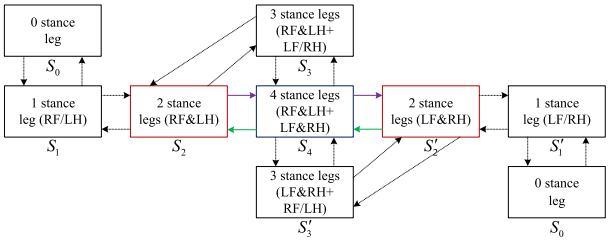


FIGURE 4. Control state machine for quadruped robots.

2) FORCES/TORQUES DISTRIBUTION METHOD

In trot gait of quadruped robots, there are always two diagonal legs in swing phase and the other two legs in stance phase at one moment in general cases. Control state machine is utilized to do the switch control between stance and swing phase. The VMC in FIGURE 3 is used to control one swing/stance leg of quadruped robots. In order to apply VMC to quadruped robots with two swing/stance legs, the forces/torques distribution problem should be investigated. For quadruped robots with two swing/stance legs, virtual components are set to RBO to drive it along the desired trajectory generally. To get the virtual forces on the origin of hip joint coordinate system of each leg in stance phase, a average distribution principle is used to do forces/torques distribution control for simple. As shown in FIGURE 5, it yields

$$F_{vmc}^{LH} = F_{vmc}^{RF} = \frac{1}{2}F_{vmc}^{\mathrm{B}} \tag{12}$$

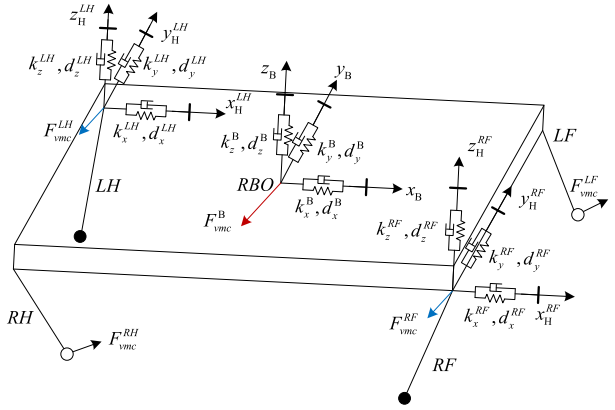


FIGURE 5. Forces/torques distribution method: RF and LH are stance legs; RH and LF are swing legs; the virtual forces/torques on the RBO can be distributed to the joint coordinate system {H} of two stance legs (RF and LH).

Combining (7) and (10) yields

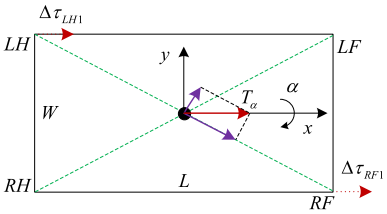
$$\begin{cases} k_x^{LH} = k_x^{RF} = \frac{1}{2}k_x^{\mathrm{B}} \\ k_y^{LH} = k_y^{RF} = \frac{1}{2}k_y^{\mathrm{B}} \\ k_z^{LH} = k_z^{RF} = \frac{1}{2}k_z^{\mathrm{B}}, \end{cases} \quad \begin{cases} d_x^{LH} = d_x^{RF} = \frac{1}{2}d_x^{\mathrm{B}} \\ d_y^{LH} = d_y^{RF} = \frac{1}{2}d_y^{\mathrm{B}} \\ d_z^{LH} = d_z^{RF} = \frac{1}{2}d_z^{\mathrm{B}} \end{cases} \tag{13}$$

Noting that (13) is for two stance legs. The virtual forces F_{vmc}^{RH}, F_{vmc}^{LF} of two legs in swing phase control are internal forces and will be taken as disturbances to deal with.

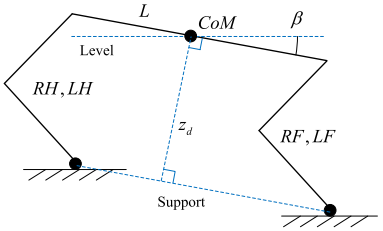
3) ATTITUDE CONTROL

In trot gait walking, quadruped robots will lean forward or backward easily. Why? This phenomenon will cause that the swing legs touch down early or mop the ground, which results in external disturbances that make robots walk unstable. From FIGURE 5, assume that RBO and two hip joints of stance legs are collinear, the gravity and virtual forces of stance legs will not generate a torque to make the robot trunk turn along the diagonal line. Thus, one reason why quadruped robots lean forward or backward easily is due to the reaction torques of forward joint θ_2, θ_3 [28]. Because the three virtual forces on RBO in FIGURE 5 are controllable via (10)(11)(13), while the three virtual torques on RBO are not. And the lateral joint θ_1 can stay still in trot walking. The flipping of robots will lead to a direct change in trunk roll and pitch angle. Thus, the trunk roll and pitch control would reject the flipping phenomenon efficiency. Moreover, yaw control is also essential to keep the forward direction of robots.

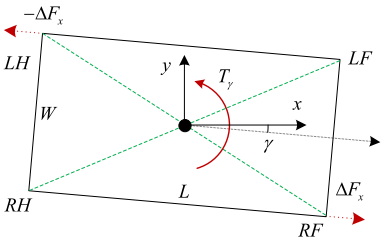
The attitude control of quadruped robots includes roll, pitch and yaw control and the control principle is shown in FIGURE 6.



(a) Roll control principle based on the virtual torque T_α along the trunk roll α



(b) Pitch control principle based on the hip joint heights control



(c) Yaw control principle based on the virtual torque T_γ along the trunk yaw γ

FIGURE 6. Attitude control principle.

a: ROLL CONTROL

The trunk roll is caused by the reaction torques in the forward joints θ_2, θ_3 and lateral joint θ_1. Since the forward joints will produce a main power to drive robots to move forward, strengthening or weakening the driving force provided by the forward joints θ_2, θ_3 intentionally will cause that robots fail to move along the desired trajectory. Fortunately, the lateral joint θ_1 not only can not provide the action torque in the forward movement, but also decouples with the forward joints θ_2, θ_3

orthogonally. Thus, lateral joint θ_1 is employed to provide the extra torque to control the trunk roll. Then, a rotate virtual spring-damper is added on RBO to control trunk roll α and the virtual torque is written as

$$T_\alpha = K_\alpha (\alpha_d - \alpha) + D_\alpha (\dot{\alpha}_d - \dot{\alpha}) \tag{14}$$

where $\alpha, \dot{\alpha}$ are the actual roll angle and angular velocity, respectively. $\alpha_d, \dot{\alpha}_d$ are the desired roll angle and angular velocity, respectively. K_α, D_α are stiffness and damping coefficients of virtual spring-damper, respectively.

Similarly, the distributive torque on each stance leg can be written as follow based on the average distribution principle

$$\Delta\tau_{i1} = \frac{T_\alpha}{2} = k_\alpha (\alpha_d - \alpha) + d_\alpha (\dot{\alpha}_d - \dot{\alpha}) \tag{15}$$

where $k_\alpha = \frac{K_\alpha}{2}, d_\alpha = \frac{D_\alpha}{2}$. Since force is mutual, the torque generated by the lateral joint should be reversed as $-\Delta\tau_{i1}$.

Adding $-\Delta\tau_{i1}$ on the lateral joint θ_1 would produce a similar trunk roll control result as the virtual torque T_α. However, it will also cause a lateral force on the foot of stance leg, which will result in a unnecessary lateral movement. Therefore, the actual trunk roll α should be controlled around its given value α_d so that $-\Delta\tau_{i1}$ changes between positive and negative values according to (15). Then, the lateral movement will reach a dynamic balance. From FIGURE 6(a), it is easy to find that a large amount of T_α ($T_\alpha \frac{L}{\sqrt{L^2+W^2}}$) goes along the diagonal two stance legs and controls the trunk roll directly. While the rest amount of T_α ($T_\alpha \frac{W}{\sqrt{L^2+W^2}}$) is vertical to the diagonal counterpart and affects the heights of stance legs, and then changes the pitch angle. Thus, a trunk pitch control is also needed.

b: PITCH CONTROL

Neglecting the influence of trunk roll, the simplest way to control the trunk pitch is increasing or decreasing hip joint heights of stance legs. In order to keep the height of RBO z_d unchanged, a virtual spring-damper is added on the origin of hip joint coordinate system and the control strategy is as follow

$$z_{id} = \begin{cases} z_d - k_\beta \sin(\beta_d - \beta) - d_\beta \left(\dot{\beta}_d - \dot{\beta}\right), & i = RF, LF \\ z_d + k_\beta \sin(\beta_d - \beta) + d_\beta \left(\dot{\beta}_d - \dot{\beta}\right), & i = RH, LH \end{cases} \tag{16}$$

where $\beta, \dot{\beta}$ are the actual pitch angle and angular velocity, respectively. $\beta_d, \dot{\beta}_d$ are the desired pitch angle and angular velocity, respectively. k_β, d_β are spring-like and damping-like coefficients, respectively. Based on the geometric relation in FIGURE 6(b), the initial $k_\beta = L/2$, and k_β, d_β can be tuned by trials. Specially, keeping the hip joint heights of stance legs consistent with the desired height of RBO z_d would achieve $\beta = \beta_d$. Besides, hip joint heights of stance legs could be self-determined to adjust complex terrains.

c: YAW CONTROL

A rotate virtual spring-damper is added on RBO along z axis to control the trunk yaw and the virtual torque is written as

$$T_\gamma = K_\gamma (\gamma_d - \gamma) + D_\gamma (\dot{\gamma}_d - \dot{\gamma}) \tag{17}$$

where $\gamma, \dot{\gamma}$ are the actual yaw angle and angular velocity, respectively. $\gamma_d, \dot{\gamma}_d$ are the desired yaw angle and angular velocity, respectively. K_γ, D_γ are stiffness and damping coefficients of virtual spring-damper, respectively. Obviously, the virtual torque should be produced by two stance legs. Since the lateral joint θ_1 is used to control the trunk roll, a forward force caused by forward joints θ_2, θ_3 are used to produce the virtual torque T_γ. The forward force will drive the robot to move forward and affect the robot forward velocity $\dot{x}$ [28]. To guarantee the unchanged forward resultant force and combining with the average distribution principle, the forward force in each stance leg should be equal in size and opposite in direction. That is

$$\Delta F_x = \mu \frac{T_\gamma}{W} \tag{18}$$

where $\mu = 1$ for RF leg and $\mu = -1$ for LH leg. Combining (17) yields

$$\Delta F_x = \mu \left(k_\gamma (\gamma_d - \gamma) + d_\gamma (\dot{\gamma}_d - \dot{\gamma})\right) \tag{19}$$

where $k_\gamma = \frac{K_\gamma}{W}, d_\gamma = \frac{D_\gamma}{W}$. As thus, the extra forward force difference between two stance legs $2\Delta F_x$ can control the trunk yaw follow the given γ_d.

4) VIRTUAL MODEL CONTROL IN STANCE PHASE

Combining equations (10),(11),(14),(15),(16),(17),(18),(19), the virtual model control of leg in stance phase can be concluded as: 1) the virtual forces along three axes are utilized to control the body RBO height, adjust the lateral velocity and forward direction; 2) extra torques $-\Delta\tau_{i1}$ are added on the lateral joint θ_1 to do the trunk roll control; 3) the pitch control is transferred to a translational control in z axis; 4) two inverse forward forces ΔF_x are added on two stance legs to do the yaw control. In sum, the whole control law expression in stance phase can be written as

$$\begin{aligned} F_i^{sp} &= \begin{bmatrix} F_{ix}^{sp} \\ F_{iy}^{sp} \\ F_{iz}^{sp} \end{bmatrix} \\ &= \begin{bmatrix} k_x^{sp} & 0 & 0 \\ 0 & k_y^{sp} & 0 \\ 0 & 0 & k_z^{sp} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ z_d - z \end{bmatrix} \\ &\quad + \begin{bmatrix} d_x^{sp} & 0 & 0 \\ 0 & d_y^{sp} & 0 \\ 0 & 0 & d_z^{sp} \end{bmatrix} \begin{bmatrix} \dot{x}_d - \dot{x} \\ \dot{y}_d - \dot{y} \\ \dot{z}_d - \dot{z} \end{bmatrix} \\ &\quad + \mu \begin{bmatrix} k_\gamma (\gamma_d - \gamma) + d_\gamma (\dot{\gamma}_d - \dot{\gamma}) \\ 0 \\ 0 \end{bmatrix} \\ \tau_i^{sp} &= -J_i^T F_i^{sp} - \begin{bmatrix} k_\alpha (\alpha_d - \alpha) + d_\alpha (\dot{\alpha}_d - \dot{\alpha}) \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{20}$$

and its normal format can be written as

$$\begin{aligned} F_i^{sp} &= K_\chi\left(\chi_{id}-\chi\right)+D_\chi\left(\dot{\chi}_{id}-\dot{\chi}\right)+\mu_i T_\psi \\ T_\psi &= K_\psi\left(\psi_d-\psi\right)+D_\psi\left(\dot{\psi}_d-\dot{\psi}\right) \\ \tau_i^{sp} &= -J_i^T F_i^{sp}-\nu T_\psi \end{aligned} \tag{21}$$

where $\chi = \left[x\ y\ z\right]^T$, $\psi = \left[\alpha\ \beta\ \gamma\right]^T$ are respectively robot position and attitude, which can be obtained by a back-calculation of foot position and measured via an on-body IMU, respectively. The control parameters matrixes are respectively $K_\chi = \begin{bmatrix} k_x^{sp} & 0 & 0 \\ 0 & k_y^{sp} & 0 \\ 0 & 0 & k_z^{sp} \end{bmatrix}$, $D_\chi = \begin{bmatrix} d_x^{sp} & 0 & 0 \\ 0 & d_y^{sp} & 0 \\ 0 & 0 & d_z^{sp} \end{bmatrix}$,

$K_q = \begin{bmatrix} k_\alpha & 0 & 0 \\ 0 & k_\beta & 0 \\ 0 & 0 & k_\gamma \end{bmatrix}$, $D_q = \begin{bmatrix} d_\alpha & 0 & 0 \\ 0 & d_\beta & 0 \\ 0 & 0 & d_\gamma \end{bmatrix}$,

$$\mu_i = \begin{cases} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, i = RF, RH \\ \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, i = LF, LH, \end{cases} \qquad \nu = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Note that virtual model control in swing phase does not joint in the attitude control and is taken as a disturbance.

5) VELOCITY CONTROL

Walking velocity is mainly determined by step length and frequency. Frequency is limited in robots. Thus, adjusting step length is an efficient way to control the walking velocity of robot. Referring to the three separate parts control method proposed by Raibert in [39], this paper adjusts the step length or foothold based on the desired robot velocity

$$\begin{cases} x_f = \dfrac{1}{2}\dot{x}T_{sp} - k_{vx}\left(\dot{x}_d - \dot{x}\right) \\ y_f = \dfrac{1}{2}\dot{y}T_{sp} - k_{vy}\left(\dot{y}_d - \dot{y}\right) \end{cases} \tag{22}$$

where $\dot{x}, \dot{y}$ are forward and lateral actual velocity, respectively. $\dot{x}_d, \dot{y}_d$ are forward and lateral desired velocity, respectively. k_{vx}, k_{vy} are the gains of difference of velocity; T_{sp} is the stance time, which can be set as a constant or the same as the stance time of last step [39]. When T_{sp} is constant, the faster speed, the larger step length, and vice versa. When the speed is zero, the robot comes into a marking time. The planning of z_f refers to the pitch control or is suggested to be its initial value z_0 specially. With the initial point in swing phase (which is the final point in stance phase) and the final foothold/point of swing phase known, a cycloid curve is utilized to plan the swing gait trajectory as follow

$$\begin{cases} x_{sw} = x_0 + (x_f - x_0)\dfrac{\varphi - \sin\varphi}{2\pi} \\ y_{sw} = y_0 + (y_f - y_0)\dfrac{\varphi - \sin\varphi}{2\pi} \\ z_{sw} = z_0 + H\dfrac{1-\cos\varphi}{2} \\ \varphi = \dfrac{2\pi t}{T_{sw}} \end{cases} \tag{23}$$

As thus, the velocity control is achieved through the foothold control.

IV. ANTI-DISTURBANCE CONTROL

Even though virtual model control could be a simple controller for quadruped robots to walk stable, it is still hard to handle complex terrains. Thus, an anti-disturbance control associated with virtual model control is essential.

A. GRAVITY COMPENSATION

Due to the effect of gravity on the vertical direction of z axis in the stance phase control, the actual height of hip joint is always below the desired height z_d. As thus, there exist unnecessary control errors in two stance legs. The errors can be addressed by introducing extra force ΔF_z to compensate the gravity. Average distribution principle is still used to do the forces/torques distribution control between two stance legs and it yields

$$\Delta F_z = \frac{mg}{2} \tag{24}$$

If a force sensor is added on the foot, then the compensation method can be

$$\Delta F_z = f_z \tag{25}$$

where f_z is the vertical force after filtering.

B. STEP LENGTH ADJUSTMENT

Legged animals will reduce their step lengthes automatically when they go up or down. Visual analysis shows that walking with a small step length can overcome less gravity each time to facilitate climbing, and control the downhill speed easily to prevent sliding. Based on (22), the step length is determined by changing the movement velocity and the period of stance phase. In this paper, the movement velocity is controlled by a upper controller. Then the method of changing the period of stance phase T_{sp} is used to adjust the step length. In the moving process, the pitch angle β can be used to estimate the slope of ground under the condition that the robot trunk is parallel to the ground. The larger ground slope, the smaller step, and it yields

$$T_{sp} = T_{sp0}\cos\left(k_{sp}\beta\right) \tag{26}$$

where T_{sp0} is the period of stance phase on the flat ground and k_{sp} is the control gain.

C. SWING TRAJECTORY ADJUSTMENT

To adapt complex terrains, an exploratory gait is proposed. Exploratory gait, referring to (23), is with a touch sensing mechanism (force sensor). When the foot touches down the ground at $t \leq T_{sw}$, the swing gait ends without executing the remaining planned trajectory. Once the foot does not touch down the ground after $t > T_{sw}$, the following planned gait is employed to continue dropping down until the foot touches down

$$\begin{cases} x_{sw} = x_f \\ y_{sw} = y_f \\ z_{sw} = z_f - k_{sw}\left(t - T_{sw}\right) \end{cases} \tag{27}$$

where k_{sw} is the dropping down speed leading the foot to touch down.

D. ATTITUDE ADJUSTMENT

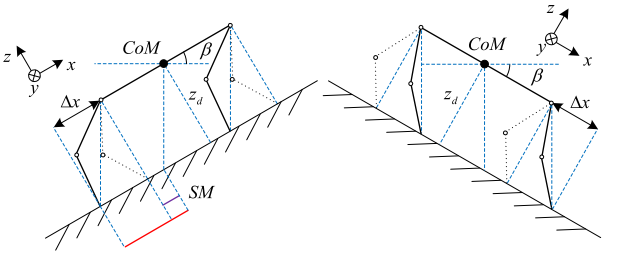
As shown in FIGURE 7, the ability of quadruped robots walking up and down slopes and on inclined terrains is mainly determined by the trunk inclined angle, the RBO height and the related positions of stance legs. And it reflects in the gravity distribution and stability margin (SM), which is the minimum distance between the projection point of RBO on the supporting surface and footholds of stance legs. Taking walking up a slope as an example, 1) if trunk pitch β is maintained horizontally, then the hind legs are too long for climbing while the front legs are too short, which results in the limited workspace for four legs; 2) if the RBO height z_d is too high or too low, it will also reduce the effective workspace, and if the body height is too high, it will reduce the stability margin as well, which will lead to slip down easily; 3) if the related positions of stance legs are the same as that on the flat ground (dotted line leg), and then the stability margin is small relatively, and the gravity distribution between the front and hind legs is uneven, and the front legs are easy to slip because of the small support reaction force from the ground. So do walking a down slope and on inclined terrains. To solve above problems, inclined attitude (solid line leg) is utilized to make the projection of RBO at the midpoint of two stance legs as far as possible, so as to obtain the maximum stability margin and optimal gravity balance at the same time. According to the geometric relationship, the offsets of x, y axis before and after the foot adjustment can be respectively given as

$$\begin{aligned} \Delta x &= z_d \tan\beta \\ \Delta y &= -z_d \tan\alpha \end{aligned} \tag{28}$$

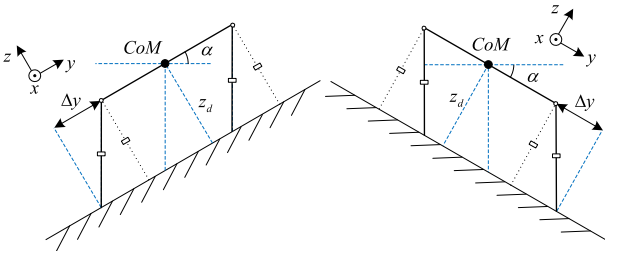
Adding it into the foothold planning/selecting in the velocity planning (22) yields

$$\begin{cases} x_f = \dfrac{1}{2}\dot{x}T_{sp} - k_{vx}\left(\dot{x}_d - \dot{x}\right) + \Delta x \\ y_f = \dfrac{1}{2}\dot{y}T_{sp} - k_{vy}\left(\dot{y}_d - \dot{y}\right) + \Delta y \end{cases} \tag{29}$$

And then the attitude adjustment is achieved.



(a) Attitude adjustment in walking up and down slope



(b) Attitude adjustment in walking on inclined terrains

FIGURE 7. Attitude adjustment strategy: SM is the stability margin; Δx, Δx are the foot adjustment to achieve the attitude adjustment.

E. VIRTUAL FORCES/TORQUES ADJUSTMENT

In the virtual model control in stance phase (20), F_{ix}^{sp}, F_{iy}^{sp} are mainly determined by d_x^{sp}, d_y^{sp}. As shown in FIGURE 7, using the same small d_x^{sp}, d_y^{sp} in both an even terrain and a slope will produce related small forces F_{ix}^{sp}, F_{iy}^{sp} in a slope walking so that the robot is unable to overcome the gravity component ($mg\sin\alpha$, $mg\sin\beta$) in the slope direction. If d_x^{sp}, d_y^{sp} were too large, the feet will be more sensitive to speed and produce large forces F_{ix}^{sp}, F_{iy}^{sp}, which result in forces overshoots and feet slippage. Therefore, the virtual forces F_{ix}^{sp}, F_{iy}^{sp} should be adjusted to adapt the change of terrains. The gravity component along the slope can be simply added as a feedforward compensation as follow

$$\begin{aligned} F_{ix}^{sp} &= d_x^{sp}\left(\dot{x}_d - \dot{x}\right) + k_\gamma\left(\gamma_d - \gamma\right) + d_\gamma\left(\dot{\gamma}_d - \dot{\gamma}\right) \\ &\quad - k_{F_x} mg\sin\beta \\ F_{iy}^{sp} &= d_y^{sp}\left(\dot{y}_d - \dot{y}\right) + k_{F_y} mg\sin\alpha \end{aligned} \tag{30}$$

where $k_{F_x}, k_{F_y} \in [0, 1]$ are the compensation proportions and have the function of speed regulation. They can be adjusted by trials to adapt the slope to ensure that the forward velocity of robot is constant.

V. TRAJECTORY TRACKING CONTROL

A. VELOCITY PLANNING & TRAJECTORY TRACKING CONTROL

For a robot with a velocity control to follow a given trajectory, both velocity and yaw should be planned indirectly. FIGURE 8 shows the scheme of trajectory tracking control, where $X_0\left(x_0, y_0\right) \rightarrow X_f\left(x_f, y_f\right)$ is the given trajectory from a starting point to an ending point in the world coordinate system; R is the curvature radius of desired position in the given trajectory. $X_t\left(x_t, y_t\right)$, $X_{td}\left(x_{td}, y_{td}\right)$ are the actual and

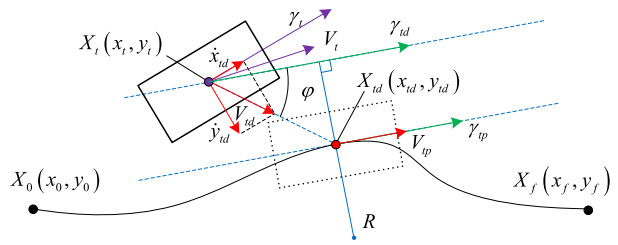


FIGURE 8. Trajectory tracking control: the solid and dotted square are the actual and desired position and attitude of quadruped robots, respectively; the solid curve is the planned trajectory.

desired position at time t, respectively. $\gamma_t, \gamma_{td}, \gamma_{tp}$ are the actual, desired, and planned robot trunk yaw at time t, respectively. V_t, V_{td}, V_{tp} are the actual, desired, and planned velocity, respectively.

Considering four factors: the terrain complexity index S_{TCI}, the curvature radius of given trajectory R, the distance to terminal $|X_t X_f|$, and the maximum velocity of quadruped robots velocity planning, yields the planned velocity as

$$V_{tp} = \min\left\{\frac{k_{TCI}}{S_{TCI}}, k_R\sqrt{R}, k_X\left|X_t X_f\right|, V_{\max}\right\} \tag{31}$$

where $k_{TCI}, k_R, k_X, d_X > 0$ are the related coefficients of influence factors, which can be tuned by trials.

Terrain is more complex, S_{TCI} is larger. They are in a positive correlation, while the planned velocity should be smaller when S_{TCI} is larger. Referring to the significance of probability and statistics of standard deviation, which represents the degree of discretization between individuals in a data set, S_{TCI} can be written as

$$S_{TCI} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(h_i - \bar{h}\right)^2}, \bar{h} = \frac{\sum_{i=1}^{N} h_i}{N} \tag{32}$$

Because the terrain complexity index is related to the robot size and step length, a chosen ground area, which is twice as the trunk area and outside the RBO, is meshed by 1/2 step length (minimum grid). Then the height standard deviation of all grid points h_i $(i = 1, \ldots, N)$ is calculated as S_{TCI}.

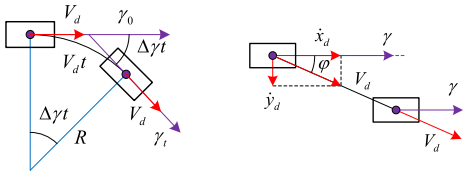
The planned velocity V_{tp}, detailed velocity (22) and yaw control (17) should be all done to make the robot follow a given trajectory. Yaw control (17) makes sure that γ follows γ_d well. As shown in FIGURE 8, to get a simple velocity control, the desired velocity of robot $\dot{x}_{td}, \dot{y}_{td}$ are set to be equal in size and has an angle φ in the direction with the planned velocity V_{tp}. φ is related to the curvature and the relative position between actual and desired robot position. Then, the desired velocity containing the forward and lateral velocity for robot can be set as

$$\begin{aligned}\dot{x}_{td} &= \left|V_{tp}\right|\cos\left(\varphi + \gamma_t - \gamma_{td}\right)\\ \dot{y}_{td} &= -\left|V_{tp}\right|\sin\left(\varphi + \gamma_t - \gamma_{td}\right)\\ \gamma_{td} &= \gamma_{tp}\end{aligned} \tag{33}$$

This is the designed trajectory tracking control. Note that desired velocity V_{td} is in the body coordinate system and the planned velocity V_{tp} is in the world coordinate system.

B. TURNING PLANNING & SLANTING PLANNING

Assume the desired velocity as V_d, the turning and slanting planning can be shown in FIGURE 9.



(a) Turning planning (b) Slanting planning

FIGURE 9. Turning planning & slanting planning based velocity.

In turning planning, the turning is achieved by changing the body trunk yaw. If the change rate of body trunk yaw is $\Delta\gamma$, then the passing arc length of robot is $V_d t$ after t and there exists

$$\Delta\gamma t = \gamma_t - \gamma_0 = \frac{V_d t}{R} \tag{34}$$

Eliminating t yields

$$\Delta\gamma = \frac{V_d}{R} \tag{35}$$

That implies, to achieve a turning with a curvature radius R under the desired velocity V_d, the change rate of body trunk yaw $\Delta\gamma$ should be (35), the turning angle is determined by (34).

It is simpler in slanting planning, and only the forward and lateral velocity $(\dot{x}_d, \dot{y}_d)$ of quadruped robots should be controlled. The desired velocity V_d and the slant angle φ are implemented as follows

$$\begin{aligned}\dot{x}_{td} &= V_d\cos\varphi\\ \dot{y}_{td} &= -V_d\sin\varphi\end{aligned} \tag{36}$$

The given/planned trajectory and path can be obtained by combining the turning and slanting planning.

VI. CONTROLLER SCHEME

As shown in FIGURE 10, the whole controller scheme is composed of a upper controller and a bottom controller. The upper controller contains the trajectory tracking control (33), anti-disturbance control (24)∼(30) and velocity control (22)(23). The trajectory tracking control consists of velocity planning (31), turning planning (34)(35) and slanting planning (36). The velocity control incorporates the upper controller with the bottom controller. The bottom controller is the whole body VMC as well. The detailed whole body VMC for quadruped robots is shown in FIGURE 11. Control state machine in FIGURE 4 is utilized to do the switch control. There are only two main

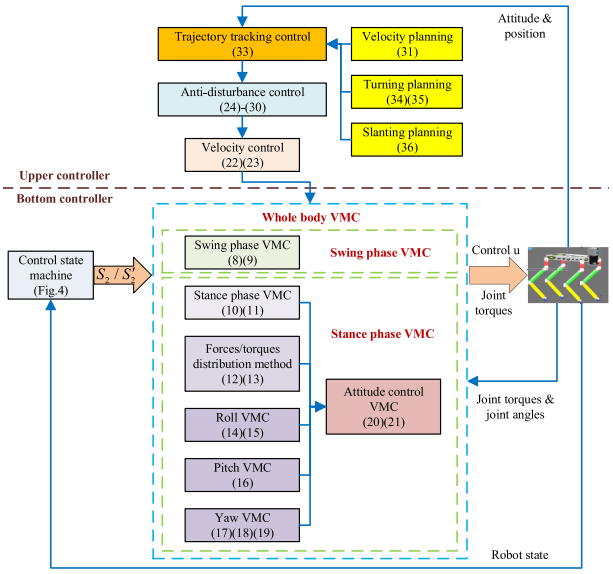


FIGURE 10. Controller scheme.

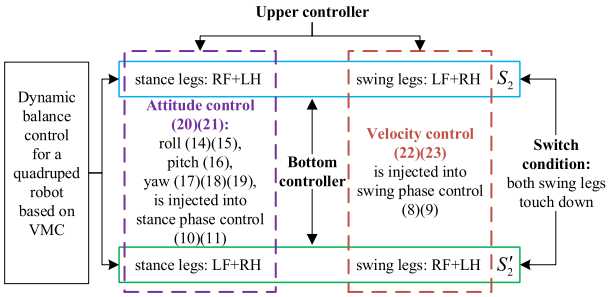


FIGURE 11. Whole body VMC.

states: S_2 and S_2', should be controlled. The switch condition between S_2 and S_2' is both swing legs touch down. Attitude control (20)(21): roll (14)(15), pitch (16), yaw (17)(18)(19), is injected into stance phase control (10)(11). Velocity control (22)(23) is injected into swing phase control (8)(9). Then the calculated joint torques are taken as the control inputs for quadruped robots. As thus, the dynamic balance control for quadruped robots locomotion based on VMC is implemented.

VII. SIMULATIONS

Simulations are implemented by co-simulating between WEBOTS and MATLAB. WEBOTS is utilized to setup the dynamic model of quadruped robots. MATLAB is applied to design the proposed controller. In MATLAB, ode4 (Runge-Kutta) solver is chosen and the fixed-step size is set as $0.001s$. In WEBOTS, the impact model is default. Co-simulations are employed to validate the proposed VMC for quadruped robots. The system and simulation parameters are shown in TABLE 1. Many simulation videos can be seen through web links in TABLE 2.

TABLE 1. System and simulation parameters.

Parameters	Value
l_1, l_2, l_3	$0.08m, 0.41m, 0.43m$
$\theta_1, \theta_2, \theta_3$	$-33^\circ \sim 25^\circ, 7^\circ \sim 88^\circ, 46^\circ \sim 133^\circ$
L, W, H_B(Body height)	$1.1m, 0.57m, 1.08m$
m_1, m_2, m_3, m_B	$1kg, 2kg, 2kg, 50kg$
Walking height, Mass, Load	$0.45m \sim 0.83m, 60kg, 300kg$
$k_x^{sp}, k_y^{sp}, k_z^{sp}$	$0, 0, 3 \times 10^4 N/m$
$d_x^{sp}, d_y^{sp}, d_z^{sp}$	$1800, 1800, 1080N/(m/s)$
$k_x^{sw}, k_y^{sw}, k_z^{sw}$	$3300, 3300, 1.3 \times 10^4 N/m$
$d_x^{sw}, d_y^{sw}, d_z^{sw}$	$90, 90, 260N/(m/s)$
$k_\alpha, k_\beta, k_\gamma$	$1800, 2900, 1200Nm/rad$
$d_\alpha, d_\beta, d_\gamma$	$200, 120, 80N/(rad/s)$
k_{vx}, k_{vy}	$0.2, 0.2s$
$\dot{x}_d, \dot{y}_d, z_d$	$0.1m/s, 0, 0.59m$
H, T_{sp0}, k_{TCI}	$0.08m, 0.3s, 0.003m^2/s$
k_R, k_X, V_{max}	$0.1s^{-1}, 0.1s^{-1}, 0.6m/s$

A. PERFORMANCE INDEX

Walking speed, cost of transportation (CoT) [40], and stability are three important indexes of performance for quadruped robots. Walking speed is determined by velocity control (22). As for cost of transportation, since the energy cost of walking robot is not easy to acquire, the work cost by all actuators per walking distance unit is taken as CoT to measure the energy consumption of walking robot and it yields

$$\eta = \frac{W}{mgs} = \frac{\sum_{i=1}^{12} |T_i \theta_i|}{mgs} \tag{37}$$

where s is the walking distance, θ_i and T_i are the joint angles and joint torques of robot, respectively.

Stability margin could be utilized to measure the static walking stability [41]. There does not exist a proper evaluating indicator for dynamic walking stability. Similarly, referring to the calculation of S_{TCI}, the reciprocal of square root of standard deviation of attitude angle α_i, β_i, γ_i $(i = 1, \ldots, N)$ in a gait cycle is taken as a reference of dynamic walking stability and it yields

$$\begin{aligned}
S_d &= \frac{1}{\sqrt{\sigma_\alpha^2 + \sigma_\beta^2 + \sigma_\gamma^2}} \\
\sigma_\alpha &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\alpha_i - \bar{\alpha})^2}, \bar{\alpha} = \frac{\sum_{i=1}^{N}\alpha_i}{N} \\
\sigma_\beta &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\beta_i - \bar{\beta})^2}, \bar{\beta} = \frac{\sum_{i=1}^{N}\beta_i}{N} \\
\sigma_\gamma &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\gamma_i - \bar{\gamma})^2}, \bar{\gamma} = \frac{\sum_{i=1}^{N}\gamma_i}{N}
\end{aligned} \tag{38}$$

B. ANTI-DISTURBANCE CONTROL

FIGURE 12 shows the simulation of quadruped robots walking up and down $0.2rad$ slope and its simulation results are

TABLE 2. Simulation video links.

Simulations	Links
Slopes: $k_{F_x}=0, k_{F_y}=0$, $v_x=0.1m/s, v_y=0$	https://youtu.be/NWADUpj7uGE
Slopes: $k_{F_x}=1, k_{F_y}=1$, $v_x=0.1m/s, v_y=0$	https://youtu.be/_sHQBO9LBRk
Slopes: $k_{F_x}=2, k_{F_y}=2$, $v_x=0.1m/s, v_y=0$	https://youtu.be/YPhpfLRMFzc
Hitting, $v_x=0.1m/s, v_y=0$	https://youtu.be/564TcR2-AEM
$y=0.5\sin\left(\frac{\pi x}{2}\right)m, x=0.1\,(t-6)\,m, yaw=dy/dx$	https://youtu.be/OMeS-yg45_0
$y=0.5\sin\left(\frac{\pi x}{2}\right)m, x=0.1\,(t-6)\,m, yaw=0$	https://youtu.be/1jFwK9TynOc
Trot walking on even terrain with constant speed $v_x=0.3m/s, v_y=0$	https://youtu.be/7c_fa8Ejw9E
Trot walking through uneven terrain with constant speed $v_x=0.3m/s, v_y=0$	https://youtu.be/4KnO2LFBwGk
Trot marking time on uneven terrain	https://youtu.be/RMOuH-pH_Ns
Trot walking with different speed $v_x\in[0,0.5]\,m/s, v_y=0$	https://youtu.be/KPK4xJtaOms
Trot walking with different speed $v_x=0, v_y\in[0,0.2]\,m/s$	https://youtu.be/HYaA_G5DaTk
Trot walking with varying $v_x\in[-0.1,0.3]\,m/s$ and $v_y\in[-0.1,0.1]\,m/s$	https://youtu.be/n8uSjmU9_QM

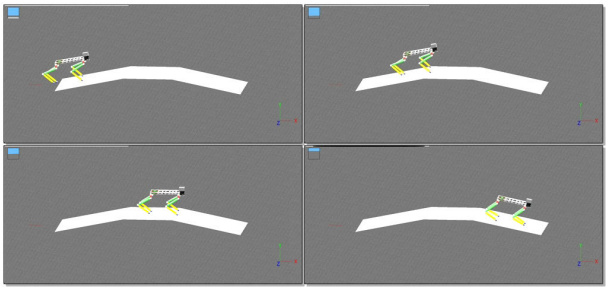


FIGURE 12. Simulation of quadruped robots walking up and down 0.2*rad* slope.

shown in FIGURE 13. RBO displacement, RBO velocity, RBO acceleration, attitude adjustment, energy cost (before divided by the walking distance in (37)), and dynamic stability (38) are shown in FIGURE 13(a)∼13(f), respectively. In FIGURE 13(a), at the beginning of motion, the robot squats down (bends its knees) from a upright state. Thus, the robot does not move but its height (in z axis) decreases firstly. Easy to see that RBO has a stable forward velocity. In the lateral movement, the coupling between forward and lateral velocity results in a left deviation on uphill and a right deviation on downhill. Thus, trajectory tracking control is needed to eliminate the lateral deviation. Because of anti-disturbance control, the RBO velocity and attitude angle fluctuate around the expected values and are not affected by the walking up and down slopes. But the fluctuation on slopes is larger than that on the flat ground. The pitch angle keeps parallel to the ground. By calculation, the CoT is about 0.79 in the whole progress. And CoT is large on uphill and small on downhill. The dynamic stability is smaller than that on the flat ground as well. The detailed quantitative indexes of simulation results are shown in TABLE 3.

Specially, the simulation results of quadruped robots walking up and down 0.2*rad* slope with/without k_{F_x}, k_{F_y} in (30) are shown in Fig 14. Easy to see that the actual velocity can track the desired velocity $0.1m/s$ well with virtual forces/torques adjustment (30). Without the virtual forces/torques adjustment, the actual velocity can not track the desired velocity on up and down slopes. The actual velocity decreases on uphill and increases on downhill.

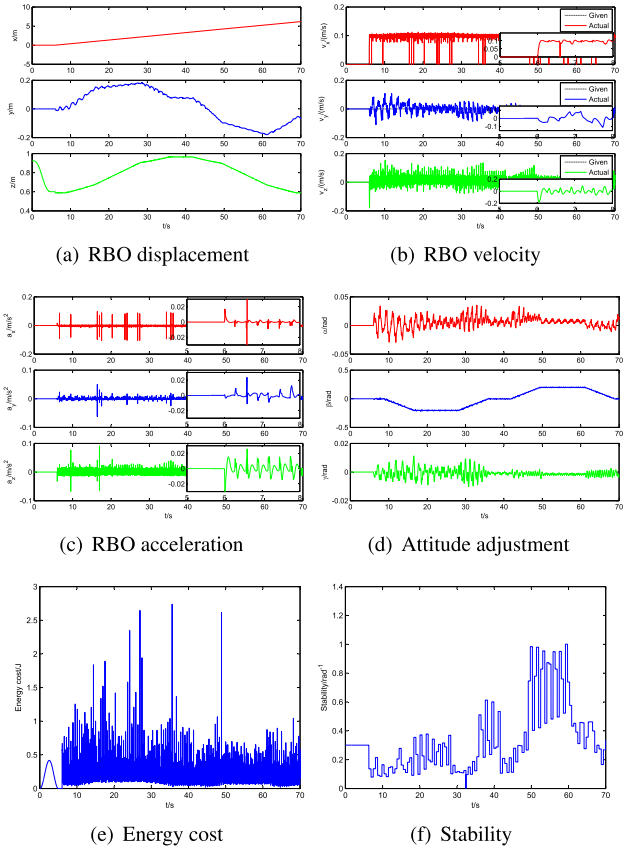


FIGURE 13. Simulation results of quadruped robots walking up and down 0.2*rad* slope.

To do further research on the anti-disturbance control, simulation of trot walking with hitting is done, as shown in FIGURE 15. External disturbances are generated by rotating a bar at times $13.2s$, $16.4s$, $19s$ with torques $400Nm$, $500Nm$, $600Nm$ respectively to strike the front, middle and hind ends of robot trunk. The action time of torques is $0.03s$. RBO displacement, RBO velocity, attitude adjustment, energy cost, and dynamic stability are shown in FIGURE 16(a)∼16(e), respectively. It can be seen that the forward movement of RBO is smooth, while the lateral movement has a left deviation (about $0.45m$) due to the external disturbance forces

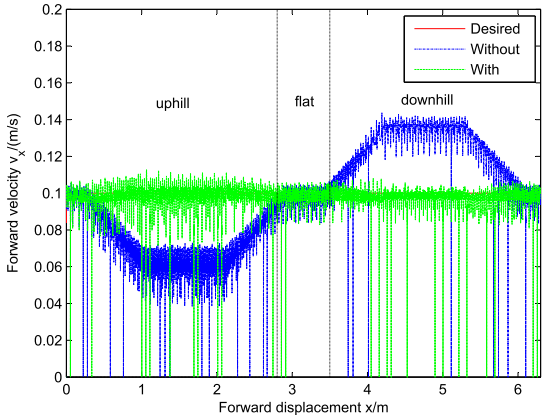


FIGURE 14. Simulation results of quadruped robots walking up and down 0.2*rad* slope with/without k_{F_x}, k_{F_y}.

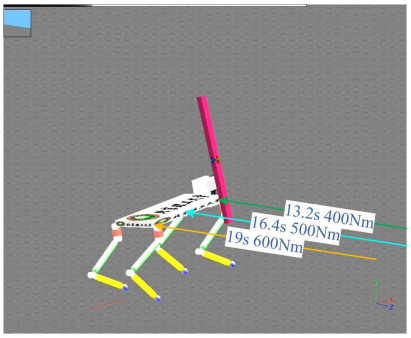


FIGURE 15. Simulation of walking with hitting.

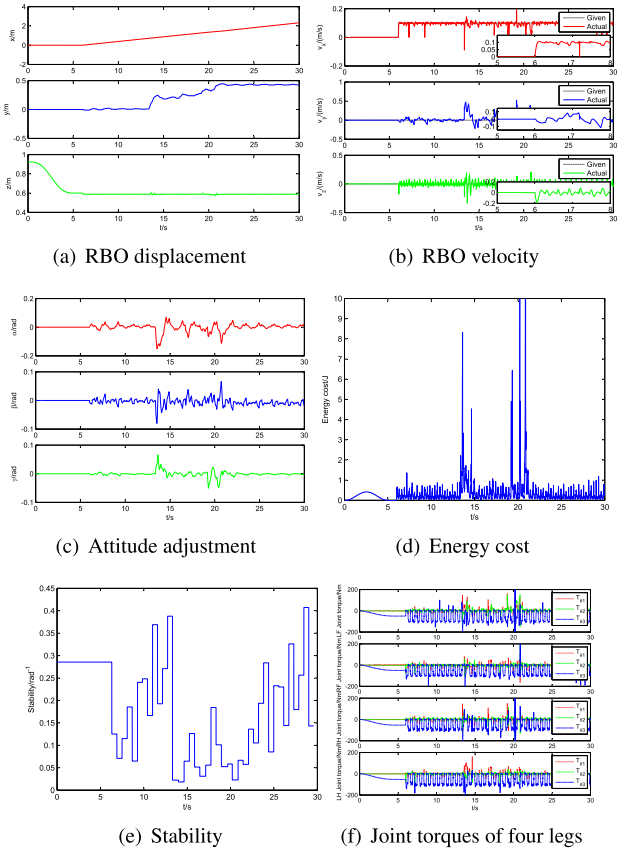


FIGURE 16. Simulation results of walking with hitting.

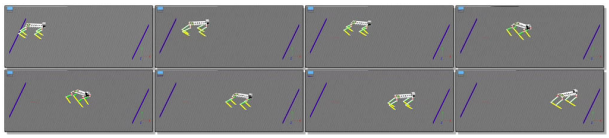


FIGURE 17. Simulation of walking with hitting.

(three hits). The external disturbances make the fluctuations of RBO velocity and attitude angle of robot increase about three times, and the stability of robot decreases sharply after hitting. Fortunately, the robot can be adjusted to stable again quickly by the proposed dynamic balance control, which consumes a lot of energy at the same time. FIGURE 16(f) is the joint torques of four legs. During the period of disturbances, the joints exert additional torques to restrain the disturbances and maintain a balance state. Thus, CoT increases to 1.44. More detailed quantitative indexes of simulation results are shown in TABLE 3.

C. TRAJECTORY TRACKING CONTROL

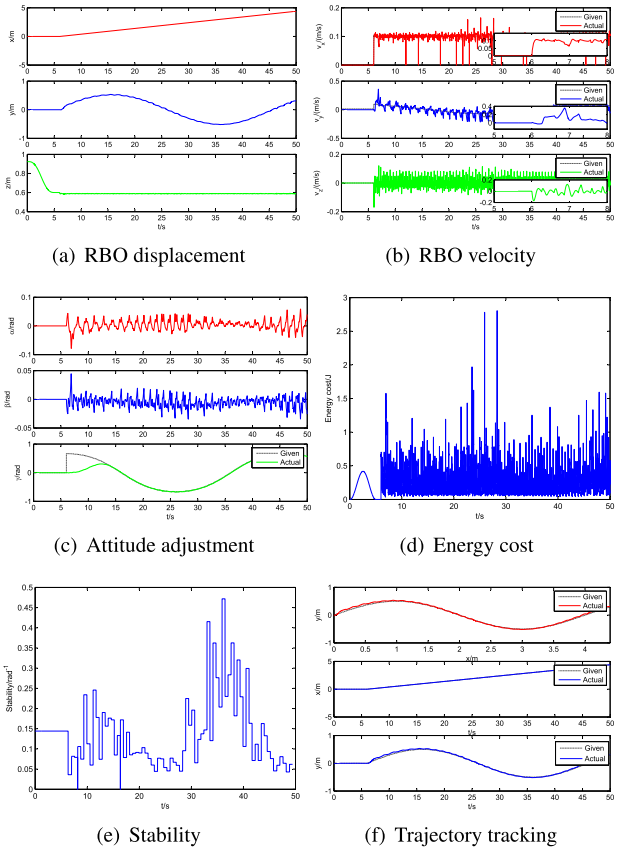
The forward and lateral velocity controls are coupled and will cause displacement offset to each other. This kind of coupling is disadvantageous to use the odometry method for trajectory tracking of quadruped robots. Then, a trajectory tracking control is proposed as shown in FIGURE 8 and its simulation is shown in FIGURE 17. FIGURE 18 shows the control performance of trajectory tracking with a given trajectory: $y = 0.5 \sin\left(\frac{\pi x}{2}\right)$, $x = 0.1\,(t-6)$, after $t = 6s$. RBO displacement, RBO velocity, attitude adjustment, energy cost, and dynamic stability are shown in FIGURE 18(a)∼18(f), respectively. Easy to see that the actual trunk displacement (FIGURE 18(a)), actual velocity (FIGURE 18(b)) and actual yaw (FIGURE 18(c)) can follow the given ones well. The control error of roll is: $-0.1rad \sim 0.1rad$, larger than the pitch and yaw errors, which are $-0.05rad \sim 0.05rad$ and $-0.04rad \sim 0.04rad$, respectively. The trajectory tracking performance is shown in FIGURE 18(f). The main trajectory tracking error occurs in the lateral direction, and its steady error is about $0.02m$, larger than that in the forward direction: $0.003m$. More detailed quantitative indexes of simulation results are shown in TABLE 3.

D. COMPARATIVE SIMULATIONS

To highlight the significance of the proposed VMC for quadruped robots, comparative simulations are implemented with the inverse dynamics (ID) approach [8]. Comparative simulations are about the tracking of planned gait trajectory of one leg of quadruped robot. The gait planning is based on the cycloid curve method in (23). The parameters of gait planning are: step length $S = 0.1m$, step height $H = 0.1m$, swing time $T_{sw} = 1s$, stance time $T_{sp} = 1s$. The

TABLE 3. Quantitative indexes of simulation results.

Type of variable	$0.2rad$ slope	hitting	trajectory tracking
error of x	$-0.014m \sim 0.015m$	$-0.02m \sim 0.02m$	$-0.01m \sim 0.01m$
error of y	$-0.2m \sim 0.2m$	$-0.02m \sim 0.45m$	$-0.16m \sim 0.16m$
error of z	$-0.012m \sim 0.01m$	$-0.013m \sim 0.016m$	$-0.09m \sim 0.07m$
error of v_x	$-0.02m/s \sim 0.015m/s$	$-0.03m/s \sim 0.03m/s$	$-0.018m/s \sim 0.015m/s$
error of v_y	$-0.1m/s \sim 0.1m/s$	$-0.025m/s \sim 0.05m/s$	$-0.02m/s \sim 0.01m/s$
error of v_z	$-0.02m/s \sim 0.02m/s$	$-0.025m/s \sim 0.02m/s$	$-0.04m/s \sim 0.05m/s$
error of α	$-0.03rad \sim 0.03rad$	$-0.17rad \sim 0.06rad$	$-0.03rad \sim 0.03rad$
error of β	$-0.015rad \sim 0.01rad$	$-0.08rad \sim 0.07rad$	$-0.03rad \sim 0.02rad$
error of γ	$-0.01rad \sim 0.01rad$	$-0.05rad \sim 0.07rad$	$-0.01rad \sim 0.01rad$
CoT	0.79	1.44	0.71
Stability	$0.08rad^{-1} \sim 1rad^{-1}$	$0.02rad^{-1} \sim 0.42rad^{-1}$	$0.03rad^{-1} \sim 0.47rad^{-1}$



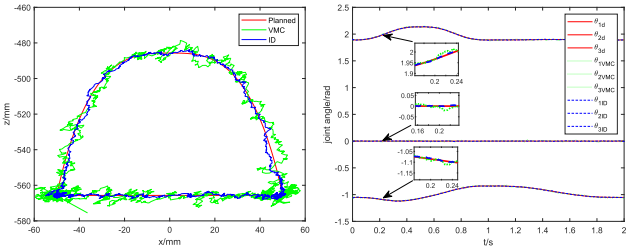
(a) RBO displacement (b) RBO velocity

(c) Attitude adjustment (d) Energy cost

(e) Stability (f) Trajectory tracking

FIGURE 18. Simulation results of trajectory tracking control.

tracking performance of gait trajectory and joint angles are shown in Fig. 19(a) and 19(b), respectively. On one hand, it is easy to find that the tracking performance of proposed VMC is not better than the ID's. But it is acceptable for the quadruped locomotion. On the other hand, the comparative simulations are implemented on a PC with CPU: Intel(R) Core(TM) i7-8750H 2.2GHz. The simulation time of one control cycle of proposed VMC is about $0.51ms$, which is smaller than that of ID (about $0.75ms$). Therefore, the proposed controllers based on VMC has an acceptable tracking performance and lower computational cost, which provide more freedom to design the intelligent algorithm in the upper controller.



(a) The tracking performance of gait trajectory (b) The tracking performance of joint angles

FIGURE 19. Comparative simulations between VMC and ID in the tracking of planned gait trajectory of one leg of quadruped robot.

VIII. CONCLUSION

In this paper, an intuitive control scheme called virtual model control is proposed for quadruped robots locomotion. Firstly, a dynamic balance control based on virtual model control, from each leg to the whole body, is proposed for quadruped robots in the trot gait. And, an anti-disturbance control is proposed to improve the robustness, terrain adaptability, and dynamic balance performance of quadrupedal locomotion. Then, a trajectory tracking control based on an intuitive velocity control is proposed to incorporate the upper controller (trajectory planning & tracking) with the bottom controller (the whole body VMC of quadruped robot). Finally, by selecting a proper set of gait parameters, simulation results demonstrate that a stable dynamic walking can be achieved for the quadruped. Compared with the inverse dynamics approach, the proposed controllers based on VMC has lower computational cost even though the tracking performance of proposed controllers is not better than that of inverse dynamics approach. The low computational cost is meaningful, which makes room for the artificial intelligence algorithm design of quadruped robots in our future work, which includes the experimental validations as well.

REFERENCES

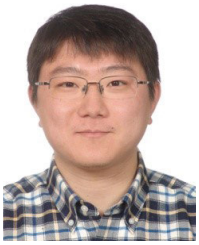
[1] A. Iqbal, Y. Gao, and Y. Gu, "Provably stabilizing controllers for quadrupedal robot locomotion on dynamic rigid platforms," *IEEE/ASME Trans. Mechatronics*, early access, Jun. 4, 2020, doi: 10.1109/TMECH.2020.2999900.

[2] C. Yang, L. Ding, K. Wang, R. Song, D. Tang, H. Gao, and Z. Deng, "The effects of walking speed and hardness of Terrain on the foot-Terrain interaction and driving torque for planar human walking," *IEEE Access*, vol. 7, pp. 56174–56189, 2019.

[3] G. Chen, J. Wang, L. Wang, and Y. He, "Design and simulation of a hydraulic biped robot," in *Proc. 32nd Chin. Control Conf. (CCC)*, 2013, pp. 4244–4249.
[4] M. Chignoli and P. M. Wensing, "Variational-based optimal control of underactuated balancing for dynamic quadrupeds," *IEEE Access*, vol. 8, pp. 49785–49797, 2020.
[5] Z. Li, Y. Li, X. Rong, and H. Zhang, "Grid map construction and Terrain prediction for quadruped robot based on C-Terrain path," *IEEE Access*, vol. 8, pp. 56572–56580, 2020.
[6] G. Chen, J. Wang, and L. Wang, "Gait planning and compliance control of a biped robot on stairs with desired ZMP," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 2165–2170, 2014.
[7] Y. de Viragh, M. Bjelonic, C. D. Bellicoso, F. Jenelten, and M. Hutter, "Trajectory optimization for wheeled-legged quadrupedal robots using linearized ZMP constraints," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1633–1640, Apr. 2019.
[8] C. Semini, "Hyq-design and development of a hydraulically actuated quadruped robot," Ph.D. dissertation, Dyn. Legged Syst. Lab, Istituto Italiano di Tecnologia, Univ. Genoa, Genoa, Italy, 2010.
[9] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter Bigdog, "The rough-Terrain quadruped robot.," *IFAC Proc.*, vol. 41, no. 2, pp. 822–825, 2008.
[10] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Trans. Autom. Control*, vol. 48, no. 1, pp. 42–56, Jan. 2003.
[11] S. Kolathaya, "Local stability of PD controlled bipedal walking robots," *Automatica*, vol. 114, Apr. 2020, Art. no. 108841.
[12] J. P. Reher, A. Hereid, S. Kolathaya, C. M. Hubicki, and A. D. Ames, "Algorithmic foundations of realizing multi-contact locomotion on the humanoid robot Durus," in *Proc. 12th Adv. Robot. Algorithmic Found. Robot.*, vol. 13. Cham, Switzerland: Springer, 2020, pp. 400–415.
[13] H. K. Akbari, J. Kim, and A. Pandala, "Quadrupedal locomotion via event-based predictive control and qp-based virtual constraints," 2020, *arXiv:2004.06858*. [Online]. Available: https://arxiv.org/abs/2004.06858
[14] M. Kitani, R. Asami, N. Sato, Y. Morita, T. Fujiwara, T. Endo, and F. Matsuno, "Suppression of roll oscillation in turning of quadruped robot by asymmetric amplification of central pattern generator output waveform," *J. Robot., Netw. Artif. Life*, vol. 6, no. 2, pp. 79–83, 2019.
[15] M. Hutter, H. Sommer, C. Gehring, M. Hoepflinger, M. Bloesch, and R. Siegwart, "Quadrupedal locomotion using hierarchical operational space control," *Int. J. Robot. Res.*, vol. 33, no. 8, pp. 1047–1062, Jul. 2014.
[16] D. J. Hyun, S. Seok, J. Lee, and S. Kim, "High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT cheetah," *Int. J. Robot. Res.*, vol. 33, no. 11, pp. 1417–1445, Sep. 2014.
[17] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal–a highly mobile and dynamic quadrupedal robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 38–44.
[18] A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert, "Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot," *Int. J. Robot. Res.*, vol. 32, no. 8, pp. 932–950, Jul. 2013.
[19] H.-W. Park, P. M. Wensing, and S. Kim, "High-speed bounding with the MIT cheetah 2: Control design and experiments," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 167–192, Feb. 2017.
[20] H. Jun-Pei, H. Qi, L. Yan-Hui, W. Kai, Z. Ming-Chao, and X. Zhen-Bang, "Neural network control of space manipulator based on dynamic model and disturbance observer," *IEEE Access*, vol. 7, pp. 130101–130112, 2019.
[21] M. Massari, F. Cavenago, and M. Canafoglia, "Virtual model control for planetary hexapod robot walking on rough terrain," in *Proc. IEEE Aerosp. Conf.*, Mar. 2019, pp. 1–10.
[22] J. Tian, C. Ma, C. Wei, and Y. Zhao, "A smooth gait planning framework for quadruped robot based on virtual model control.," in *Proc. Int. Conf. Intell. Robot. Appl.*, 2019, pp. 398–410.
[23] X. Wang, H. Liu, B. Liang, X. Wang, and J. Yang, "Locomotion control for quadruped robot combining central pattern generators with virtual model control," in *Proc. IEEE 15th Int. Conf. Control Autom. (ICCA)*, Jul. 2019, pp. 399–404.
[24] A. Winkler, I. Havoutis, S. Bazeille, J. Ortiz, M. Focchi, D. G. Caldwell, and C. Semini, "Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, Hong Kong, 2014, pp. 6476–6482, doi: 10.1109/ICRA.2014.6907815.
[25] R. Desai, H. Geyer, and J. K. Hodgins, "Virtual model control for dynamic lateral balance," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Nov. 2014, pp. 856–861.
[26] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *Int. J. Robot. Res.*, vol. 20, no. 2, pp. 129–143, Feb. 2001.
[27] A. Winkler, I. Havoutis, S. Bazeille, J. Ortiz, M. Focchi, R. Dillmann, D. Caldwell, and C. Semini, "Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 6476–6482.
[28] H. Xie, M. Ahmadi, J. Shang, and Z. Luo, "An intuitive approach for quadruped robot trotting based on virtual model control," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 229, no. 4, pp. 342–355, Apr. 2015.
[29] S. Zhang, H. Ma, Y. Yang, and J. Wang, "The quadruped robot adaptive control in trotting gait walking on slopes," *AIP Conf. Proc.*, vol. 1890, Feb. 2017, Art. no. 020004.
[30] M. C. Yip and D. B. Camarillo, "Model-less hybrid Position/Force control: A minimalist approach for continuum manipulators in unknown, constrained environments," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 844–851, Jul. 2016.
[31] M. Koehler, A. M. Okamura, and C. Duriez, "Stiffness control of deformable robots using finite element modeling," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 469–476, Apr. 2019.
[32] G. Chen, S. Guo, B. Hou, and J. Wang, "Fractional order impedance control," *IEEE Access*, vol. 8, pp. 48904–48916, 2020.
[33] G. Chen, J. Wang, S. Wang, J. Zhao, and W. Shen, "Compliance control for a hydraulic bouncing system," *ISA Trans.*, vol. 79, pp. 232–238, Aug. 2018.
[34] C. Guangrong, W. Junzheng, Z. Jiangbo, M. Liling, and S. Wei, "Stable impedance control of a single leg of hydraulic legged robot based on virtual decomposition control.," *Robot*, vol. 39, no. 5, pp. 704–714, 2017.
[35] F. Weng, Y. Ding, and M. Tang, "LPV model-based robust controller design of electro-hydraulic servo systems," *Procedia Eng.*, vol. 15, pp. 421–425, 2011.
[36] G. Chen, J. Wang, S. Wang, J. Zhao, and W. Shen, "Indirect adaptive robust dynamic surface control in separate meter-in and separate meter-out control system," *Nonlinear Dyn.*, vol. 90, no. 2, pp. 951–970, Oct. 2017.
[37] J. Guo, Y. Zheng, D. Qu, R. Song, and Y. Li, "An algorithm of foot end trajectory tracking control for quadruped robot based on model predictive control," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Oct. 2019, pp. 828–833.
[38] M. Focchi, "Strategies to improve the impedance control performance of a quadruped robot," Ph.D. dissertation, Dyn. Legged Syst. Lab., Istituto Italiano di Tecnologia, Univ. Genoa, Genoa, Italy, 2013.
[39] M. H. Raibert, *Legged Robots That Balance*. Cambridge, MA, USA: MIT Press, 1986.
[40] C. B. Cunningham, N. Schilling, C. Anders, and D. R. Carrier, "The influence of foot posture on the cost of transport in humans," *J. Experim. Biol.*, vol. 213, no. 5, pp. 790–797, Mar. 2010.
[41] L. Zhang and S. Gao, "Normalized energy stability margin based analysis of omni-directional static walking of a quadruped robot on rough terrain," *Adv. Mater. Res.*, vols. 383–390, pp. 7401–7405, Nov. 2011.

GUANGRONG CHEN received the B.Eng. degree in automation and the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2012 and 2018, respectively. He joined the Robotics Research Center, Beijing Jiaotong University, as a Lecturer, in 2018. He is a reviewer for a number of international journals, including the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS: SYSTEMS, the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, and the IEEE TRANSACTIONS ON MECHATRONICS. His research interests include robotic control, nonlinear systems, and adaptive control.

SHENG GUO received the Ph.D. degree from Beijing Jiaotong University, in 2005. He was then a Postdoctoral Researcher at National Cheng Kung University, from 2005 to 2006. He was a Visiting Scholar with the University of California, Irvine, USA, from 2010 to 2011. He is currently a Full Professor, the Vice Director of the Robotics Institute, and the Vice Dean of the School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University. His research interests include robotics mechanism and mechatronics.



BOWEN HOU received the Ph.D. degree from Beijing Jiaotong University, in 2016. He was a Postdoctoral Researcher with Beijing Jiaotong University, from 2016 to 2018. He was a Visiting Scholar with Heriot-Watt University, Edinburgh, U.K., from 2017 to 2018. He is currently a Lecturer with the School of Civil Engineering, Beijing Jiaotong University. His research interests include vehicle-track dynamics, environmental vibration and noise, and track dynamics in-situ measurement.

JUNZHENG WANG received the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 1994. He is currently the Deputy Director of the Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, where he is also a Professor and a Ph.D. Supervisor. His current research interests include motion control, static and dynamic performance testing of electric and electric hydraulic servo systems, and dynamic target detection and tracking based on image technology. He is a Senior Member of the Chinese Mechanical Engineering Society and the Chinese Society for Measurement. He received the Second Award from the National Scientific and Technological Progress (No. 1), in 2011.