International Journal of Advanced Robotic Systems

OPEN ACCESS ARTICLE

A Dynamic Balancing Approach for a Quadruped Robot Supported by Diagonal Legs

Regular Paper

Jian Meng[1], Yibin Li[1]* and Bin Li[1]

1 Shandong University, Jinan, Shandong, China
*Corresponding author(s) E-mail: liyb@sdu.edu.cn

Received 21 October 2014; Accepted 03 September 2015

DOI: 10.5772/61542

© 2015 Author(s). Licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

For legged robots, the most important task is to keep balance. This paper proposes a new balance control approach. To simplify the control complexity, first, LQR (linear quadratic regulator) control was used to obtain stable state feedback for the model. Then, the 6-DOF model was stabilized by dividing the whole robot into three separate parts. After that, VMC (virtual model control) was used to change the configuration of the joints. The simulation results showed that the proposed method allowed the quadruped robot to walk stably, even when certain types of disturbance were exerted on the models. In the simulation model, to mimic real conditions, noise was added to the sensors; the algorithm was then verified as still suitable for the quadruped robot.

Keywords Quadruped Robot, LQR, Balance Control, VMC

1. Introduction

For legged robots, a robust balancing algorithm is needed to ensure stability. There are many methods for evaluating the stability margin required to keep a legged robot stable. The most commonly used criterion is the static stability criterion [1], which was proposed by McGhee and Frank in 1968. This method states that the vertical projection of the COG (centre of gravity) on the supporting plane must be within the support polygon enclosed by the supporting legs, and that the shortest distance between the projection and the edge of the support polygon can be defined as the SSM (static stability margin). In addition, Papadopoulos [2] has proposed force-angle stability, which included the effect of outer forces exerted on the robot, while Vukobratovic [3] proposed the concept of the ZMP (zero-moment point): a method that considers the effect of inertia force, giving a criterion that is widely used in biped robots. Much work has been done on the control of quadruped robots [4–13]. However, the control approaches and criteria mentioned above can only be used when the robot possesses a support polygon, formed either by several supporting legs or by large feet; thus, larger distances between legs and larger feet are often preferred so as to obtain a larger stability margin. When a robot's feet are too small or when the robot jumps into the air, these criteria can no longer be used.

In the 1980s, Raibert [14] proposed the foot placement algorithm (FPA) for the control of legged robots, which divided the balancing control of the robot into three elements: speed control, posture control, and hopping height control. This approach differed greatly from the static stable methods mentioned above. With this FPA method, a series of robots with one, two and four legs were developed [15, 16]. They were well balanced with small feet and few supporting legs, but they had to keep jumping all the time to prevent from falling, which can lead to an increase in the energy consumption of quadruped robots. Emura and Arakawa [17] built a reaction-wheel-type inverted pendulum to keep their quadruped robot stable. As a result, the quadruped robot could stand stably on two diagonal legs, but its balance was only held for about 10 seconds.

In this paper, a robust and effective method for balancing a quadruped robot using two diagonal supporting legs is proposed and the simulation results show the effectiveness of the proposed method.

The remainder of the paper is organized as follows: Section 2 describes the biped model approach, and the LQR control used to stabilize this system; in Section 3, the model is modified to include six DOFs; in Section 4, the direction of the coordinate system is changed and the model extended to match the most commonly used joint configuration; white noise is added to the ideal sensors in Section 5; and conclusions are given in Section 6.

2. The Balancing Control Strategy of the Simplest Biped Supporting Model

There are various quadruped robot prototypes, many of them using prismatic legs, such as Raibert's quadruped robot and the Scout II robot.

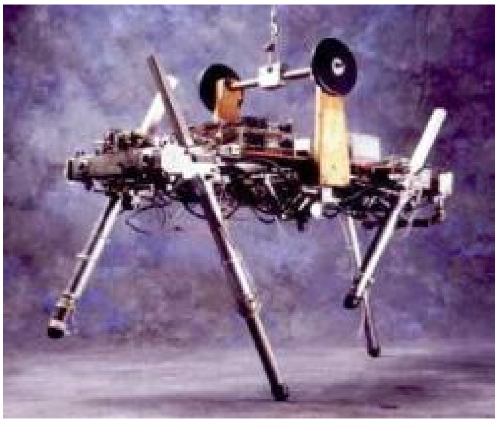


Figure 1. Raibert's quadruped robot [16]

The legs of the robots are designed based on spring-loaded inverted pendulums and can run very fast. They can keep balance easily when running or standing with four legs, but they can hardly remain stable on two legs. This problem is discussed in this paper.

When a quadruped robot stands on its diagonal legs, the other two legs are drawn back to the body, and can be seen as part of the body, meaning the quadruped robot can be simplified in terms of a body and two diagonal legs. Therefore, to analyse the stability of a quadruped with two diagonal legs, the simplest biped supporting model should firstly be built.

Figure 2. Scout II robot

In this model, two cylindrical legs are connected to a cuboid-shaped trunk by two rotary joints; the two legs stand on the ground. First, we will build the coordinate system of the model and deduce the Lagrange dynamics equation.

2.1 Coordinate Systems

All the bodies in the system are assumed to be rigid and the frame Σ_b is fixed in the trunk as shown in Figure 3. For convenience, the origin of Σ_b is set to the COG of the trunk. The front leg coordinate Σ_F is fixed in the front leg: the origin is fixed in the hip of this leg, the X axis points to the toe, and the Z axis points to the hind leg. Meanwhile, the hind leg coordinate Σ_H is fixed in the hind leg: the origin is also fixed in the leg hip, the X axis points to the toe of the hind leg, the Z axis shares the same direction as that of Σ_F, and the legs rotate around their Z axes.

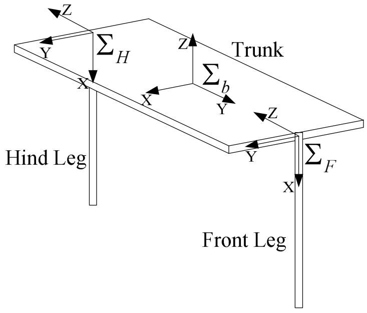


Figure 3. Coordinate systems of the biped robot model

2.2 Dynamics of the Biped Supporting Model

It is always important to analyse the dynamics of a particular robot, and there are many methods that can achieve this effectively [18–21]. In this paper, the Lagrange method is used to analyse the dynamics of the models.

Suppose the pitch angle θ_{body}, and the leg positions θ_{FS} and θ_{HS}, can be obtained by means of the attitude sensor and angle sensors. In this case, the two angles are defined as follows:

$$\theta_F = \theta_{FS} - \theta_{body} \tag{1}$$

$$\theta_H = \theta_{HS} - \theta_{body} \tag{2}$$

where θ_F is the angle between the front leg and the direction of gravity, and θ_H is the angle between the hind leg and the direction of gravity. The relationship between these angles is shown in Figure 4.

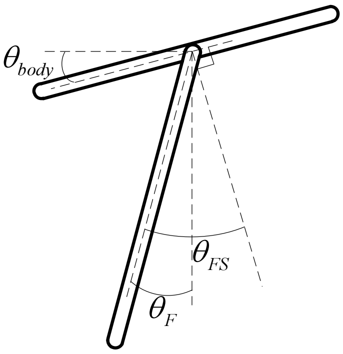


Figure 4. Angles of the biped model (front view)

In order to deduce the Lagrange equation of the model, the parameters of the symbols used are shown in Table 1 and these parameters are illustrated in Figure 5.

Symbol	Parameter	Value	Unit
m_{body}	trunk mass	100	kg
m_{leg}	mass of a single leg	5	kg
W	half of the trunk width	0.25	m
D	distance between the hips of the legs	1	m
L_{leg}	length of each leg	0.5	m
g	gravity acceleration	9.8	m/s^2
τ_F	torque exerted by front hip actuator	Variable	Nm
τ_H	torque exerted by hind hip actuator	Variable	Nm

Table 1. Parameters and values of the symbols

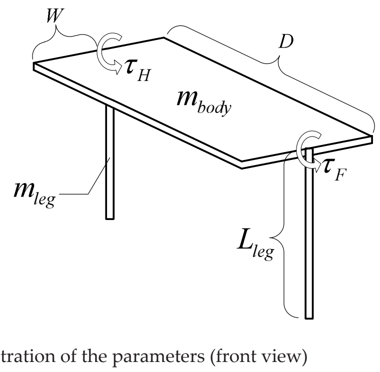


Figure 5. Illustration of the parameters (front view)

Based on the symbol definition above, the dynamic model of the biped supporting model can be derived from a Lagrange equation:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = S\tau \tag{3}$$

$$L = \Sigma E_k - \Sigma E_p \tag{4}$$

where E_k is the total kinetic energy, E_p is the total potential energy, $q = [\theta_F \ \theta_H \ \ \theta_{body}]^T$, $S = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}^T$, and $\tau = \begin{bmatrix} \tau_F \\ \tau_H \end{bmatrix}$.

Additionally,

$$\begin{aligned}
\tau_F &= \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_F} - \frac{\partial L}{\partial \theta_F} \\
&= -\frac{m_{body}}{2} g L_{leg} \sin\theta_F - \frac{m_{leg}}{2} g L_{leg} \sin\theta_F \\
&\quad - \frac{m_{body}}{12} D^2 L_{leg}^2 \dot{\theta}_F^2 \cos\theta_F \sin\theta_F \\
&\quad + \frac{m_{body}}{4} L_{leg}^2 \dot{\theta}_H^2 \sin(\theta_F - \theta_H) + \frac{m_{body}}{4} L_{leg}^2 \ddot{\theta}_F \\
&\quad + \frac{m_{body}}{12} D^2 L_{leg}^2 \dot{\theta}_H^2 \cos\theta_F \sin\theta_H \\
&\quad + \frac{m_{body}}{12} D^2 L_{leg}^2 \ddot{\theta}_F \cos\theta_F^2 + \frac{m_{leg}}{3} L_{leg}^2 \ddot{\theta}_F \\
&\quad + \frac{m_{body}}{4} L_{leg}^2 \ddot{\theta}_H \cos(\theta_F - \theta_H) \\
&\quad - \frac{m_{body}}{12} D^2 L_{leg}^2 \ddot{\theta}_H \cos\theta_F \cos\theta_H
\end{aligned} \tag{5}$$

$$\begin{aligned}
\tau_H &= \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_H} - \frac{\partial L}{\partial \theta_H} \\
&= -\frac{m_{body}}{2} g L_{leg} \sin\theta_H - \frac{m_{leg}}{2} g L_{leg} \sin\theta_H \\
&\quad + \frac{m_{body}}{12} D^2 L_{leg}^2 \dot{\theta}_F^2 \cos\theta_H \sin\theta_F \\
&\quad - \frac{m_{body}}{4} L_{leg}^2 \dot{\theta}_F^2 \sin(\theta_F - \theta_H) + \frac{m_{body}}{4} L_{leg}^2 \ddot{\theta}_H \\
&\quad - \frac{m_{body}}{12} D^2 L_{leg}^2 \dot{\theta}_H^2 \cos\theta_H \sin\theta_H \\
&\quad + \frac{m_{body}}{4} L_{leg}^2 \ddot{\theta}_F \cos(\theta_F - \theta_H) \\
&\quad - \frac{m_{body}}{12} D^2 L_{leg}^2 \ddot{\theta}_F \cos\theta_F \cos\theta_H \\
&\quad + \frac{m_{body}}{12} D^2 L_{leg}^2 \ddot{\theta}_H \cos^2\theta_H + \frac{m_{leg}}{3} L_{leg}^2 \ddot{\theta}_H
\end{aligned} \tag{6}$$

$$\begin{aligned}
\tau_{body} &= \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_{body}} - \frac{\partial L}{\partial \theta_{body}} \\
&= W^2 m_{body} \ddot{\theta}_{body} \\
&= -\tau_F - \tau_H
\end{aligned} \tag{7}$$

To control the balance of the biped supporting model, the state space equation of the model is constructed based on the following definition of the state variable: $X = [\theta_F \;\; \dot{\theta}_F \;\; \theta_H \;\; \dot{\theta}_H \;\; \theta_{body} \;\; \dot{\theta}_{body}]^T$. Then, the dynamic equation is linearized and the state space equation can be defined as

$$\dot{X} = AX + B\tau \tag{8}$$

and a state feedback based on LQR control can be obtained to stabilize the system:

$$\tau = -KX \tag{9}$$

The control structure of the system is shown in Figure 6.

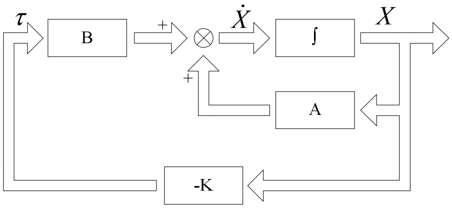


Figure 6. Control structure of the system with state feedback

Since a too large hip torque may cause the slippage of the legs, the torque must be constrained. The maximum torque is determined by the maximum static friction force between the toe and the ground, and is shown as follows:

$$\tau_{\max} = \frac{\mu_{static} F}{L_{leg}} \tag{10}$$

where μ_{static} is the coefficient of static friction, and F is the normal force that the leg exerts on the ground. If the expected hip torque exceeds $\tau_{\max}$, it is limited to $\tau_{\max}$.

3. Balance of a 6-DOF Diagonal Leg Model

In the biped model, each leg is constrained in a single plane. Hence, to allow the legs to work in 3D space, a 6-DOF model must be built. In this model, each leg has three active DOFs: the joint nearest to the hip rotates around the trunk diagonal line, the next joint rotates around the axis that is perpendicular to the diagonal line in the trunk plane, and the third joint is a prismatic joint that moves along the leg and can change the length of the leg. Based on this type of joint configuration, the motion can be decoupled into three parts, with each pair of legs resisting one type of disturbance. The decoupling mechanism is shown in Figure 7.

Just as the motion of the 6-DOF model is divided into three parts, so the control of the whole robot is also divided into three parts. As is shown in Figure 7, Part A controls the torques of the two joints around the diagonal line against postural disturbance, Part B controls the torques of the other two hip joints to resist disturbance along the diagonal line, and Part C controls the forces of the two prismatic joints to maintain the height of the robot. Of the three decoupled models, Part B and Part C are easily controlled by a PID controller, since they do not contain any dynamically balancing problems.

Therefore, the only problem is how to balance Part A to maintain the walking stability of the robot. The model of Part A is similar to the biped supporting model, so a similar control method can be applied to this part. However, before LQR control can be used, some modifications should be made. The coordinate systems of this model are different, as shown in Figure 8.

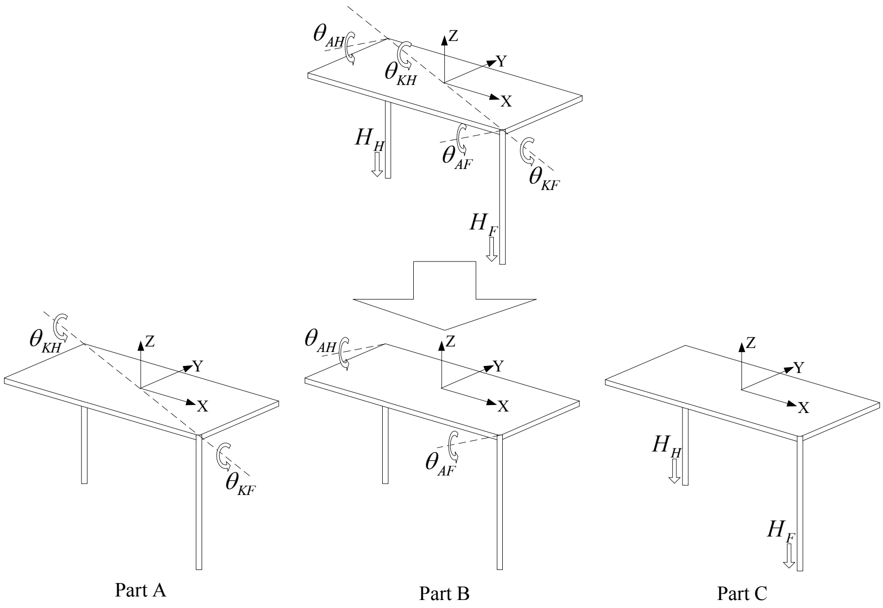


Figure 7. Joint configuration of the 6-DOF model and the decoupled models

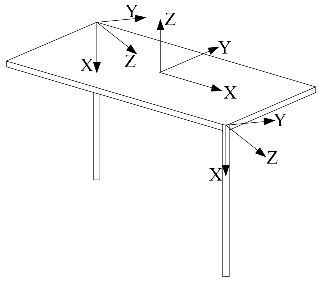
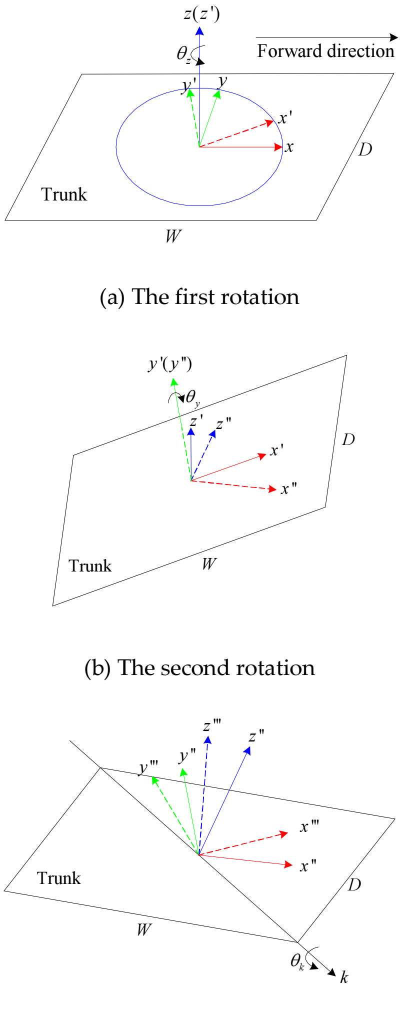


Figure 8. Coordinate systems of Part A

In Part A, the body rotation is around the diagonal line of the trunk, meaning the traditional Z-Y-X Euler angles cannot be used for body rotation angle feedback directly, and a transformation is therefore needed. Here, Z-Y-K angles are built, as shown in Figure 9; these are similar to Z-Y-X Euler angles, but the third rotation is not around the X axis of the moving coordinate system. Instead it rotates around the K axis, which is in the same direction as the diagonal line determined by the hips of the two supporting legs.



(a) The first rotation

(b) The second rotation

(c) The third rotation

Figure 9. Rotation representation of Z-Y-K angles for the 6-DOF model

The transformation matrix can be defined as follows:

$$
\begin{aligned}
R_{zyk} &= R_z(\theta_z) R_y(\theta_y) R_k(\theta_k) \\
&= \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}
\end{aligned}
\tag{11}
$$

where $R_z(\theta_z)$ represents the rotation matrix around the Z axis of the rotating coordinate system, and the rotation angle is θ_z.

So

$$
\begin{cases}
\sin\theta_y = \dfrac{BR_{32} - AR_{31}}{A} \\
\sin\theta_z = \dfrac{A^2 R_{21}\cos\theta_y - ABR_{22}\cos\theta_y - B^2 R_{12} + ABR_{11}}{A^2\cos^2\theta_y + B^2} \\
\cos\theta_z = \dfrac{A^2 R_{11}\cos\theta_y - ABR_{12}\cos\theta_y + B^2 R_{22} - ABR_{21}}{A^2\cos^2\theta_y + B^2} \\
\sin\theta_k = \dfrac{BR_{31}\cos\theta_y + BR_{33}\sin\theta_y + AR_{32}\cos\theta_y}{B^2\sin^2 y + \cos^2 y} \\
\cos\theta_k = \dfrac{R_{33}\cos\theta_y - B^2 R_{31}\sin\theta_y - ABR_{32}\sin\theta_y}{B^2\sin^2 y + \cos^2 y}
\end{cases}
\tag{12}
$$

where

$$A = \frac{D}{\sqrt{D^2 + W^2}} \tag{13}$$

$$B = \frac{W}{\sqrt{D^2 + W^2}} \tag{14}$$

In this model, the state variable θ_{body} is set to θ_k, which is still the rotation angle of the trunk. Since the only changes are the positions of the hips of the legs, the behaviour of the closed-loop system is the same as in the previous biped model.

4. Balance of the Diagonal Legs Model

In the previous section, the joints rotated around or were perpendicular to the diagonal line, but this structure is seldom used in the real world. For most quadruped robots, the legs are composed of roll joints and pitch joints, so further modifications are needed. Here, virtual model control (VMC) [22] is used as a bridge between the diagonal joints model and the common joint configuration model of quadruped robots, as shown in Figure 10.

The procedure of using the virtual model can be divided into three steps: firstly, mapping the sensor information from the normal model onto the virtual model; then, using the controller of the virtual model to calculate the output virtual torque that should be exerted on the virtual model; and finally, mapping the control force and torque of the virtual model onto the normal model. The transformation

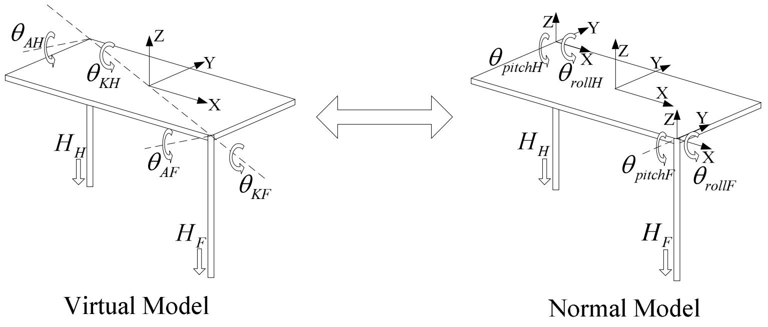


Figure 10. Change of the direction of the joints

between the virtual model controller and normal model controller is illustrated in Figure 11.

Based on the method of virtual model control, kinematic equations of the two models can be obtained. As the length of the leg is almost a constant, H is used as an approximate value to simplify the equations. Equation 15 shows the direct kinematic equation of the virtual model, and equation 16 shows that of the normal model.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = {}^{G}R_0 \, {}^{0}R_K \, {}^{K}R_A \begin{bmatrix} H \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \dfrac{D\sin\theta_a + W\cos\theta_a \sin\theta_k}{\sqrt{D^2+W^2}} H \\ \dfrac{D\cos\theta_a \sin\theta_k - W\sin\theta_a}{\sqrt{D^2+W^2}} H \\ -\cos\theta_k \cos\theta_a H \end{bmatrix} \tag{15}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sin\theta_{roll} \\ \sin\theta_{roll}\cos\theta_{pitch} \\ -\cos\theta_{roll}\cos\theta_{pitch} \end{bmatrix} H \tag{16}$$

Therefore, the joint angles of the normal model can be mapped to the virtual model, based on the above two equations:

$$\theta_k = \tan^{-1} \frac{W \dfrac{\tan\theta_{pitch}}{\cos\theta_{roll}} + D\tan\theta_{roll}}{\sqrt{D^2+W^2}} \tag{17}$$

$$\theta_a = \sin^{-1} \frac{D\sin\theta_{pitch} - W\cos\theta_{pitch}\sin\theta_{roll}}{\sqrt{D^2+W^2}} \tag{18}$$

The force Jacobians can be derived from kinematic equations. In the virtual model, the force Jacobians are defined as

$$\begin{bmatrix} \tau_k \\ \tau_a \\ F_h \end{bmatrix} = M_{\tau F1} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{19}$$

and in the normal model of quadruped robot, they are

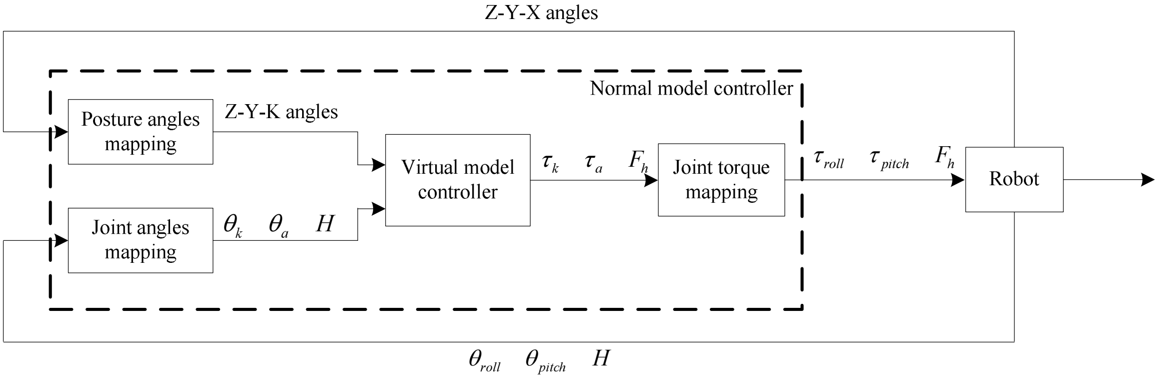


Figure 11. Transformation between the virtual model controller and the normal model controller

$$\begin{bmatrix} \tau_{roll} \\ \tau_{pitch} \\ F_h \end{bmatrix} = M_{\tau F2} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{20}$$

Hence, the relationship between the torques of the two models can be obtained from equation 19 and equation 20:

$$\begin{bmatrix} \tau_{roll} \\ \tau_{pitch} \\ F_h \end{bmatrix} = M_{\tau F1} M_{\tau F2}^{-1} \begin{bmatrix} \tau_k \\ \tau_a \\ F_h \end{bmatrix} \tag{21}$$

where

$$M_{\tau F1} = \begin{bmatrix} \frac{W\cos\theta_a \cos\theta_k}{\sqrt{D^2+W^2}}H & \frac{D\cos\theta_a \cos\theta_k}{\sqrt{D^2+W^2}}H & H\cos\theta_a \sin\theta_k \\ \frac{(D\cos\theta_a - W\sin\theta_a \sin\theta_k)}{\sqrt{D^2+W^2}}H & \frac{(-W\cos\theta_a - D\sin\theta_a \sin\theta_k)}{\sqrt{D^2+W^2}}H & H\cos\theta_k \sin\theta_a \\ \frac{D\sin\theta_a + W\cos\theta_a \sin\theta_k}{\sqrt{D^2+W^2}} & \frac{-W\sin\theta_a + D\cos\theta_a \sin\theta_k}{\sqrt{D^2+W^2}} & -\cos\theta_a \cos\theta_k \end{bmatrix} \tag{22}$$

$$M_{\tau F2} = \begin{bmatrix} 0 & H\cos\theta_{pitch}\cos\theta_{roll} & H\cos\theta_{pitch}\sin\theta_{roll} \\ H\cos\theta_{pitch} & -H\sin\theta_{pitch}\sin\theta_{roll} & H\cos\theta_{roll}\sin\theta_{pitch} \\ \sin\theta_{pitch} & \cos\theta_{pitch}\sin\theta_{roll} & -\cos\theta_{pitch}\cos\theta_{roll} \end{bmatrix} \tag{23}$$

5. Simulation and Analysis

5.1 Simulation of the Biped Robot

The control method was tested in RobotBuilder [23], a simulation software based on the DynaMechs [24] dynamics engine. The environment parameters were set according to Table 2.

Ground normal spring stiffness (N/m)	100000
Ground planar spring stiffness (N/m)	10000
Ground normal damper (Ns/m)	3000
Ground planar damper (Ns/m)	1000
Coefficient of static friction	0.4
Coefficient of kinetic friction	0.3

Table 2. Environment parameters

In the simulation, the closed-loop system was able to maintain balance for a long time; therefore, to test the stability of the biped model, various disturbances were exerted on the robot model. The pitch angular velocity was set to 1rad/s, 2rad/s, and 3rad/s, consecutively. The results of the simulation showed that the robot did not lose balance, and the three angles converged towards zero slowly, as shown in Figure 12.

Furthermore, yaw angular velocity was added to the system to test its torsional stability; the four disturbances were 1rad/s, 2rad/s, 3rad/s, and 4rad/s, separately. From Figure 13, we can see that the robot didn't fall down but that the legs moved on the ground. During the motion, the hip torque constraints were still effective; the slippage was instead caused by the fast rotating trunk, which pulled the legs up from the ground for a very short time. When the legs touched the ground again, the trunk began to slow down and finally all the state variables returned to zero.

The previous disturbances to the robot model were all symmetrical disturbances, which did not affect the position of its COM (centre of mass). Lateral disturbance was harder to resist, since the legs were not allowed to move on the ground; in this case, a little displacement can cause large gravitational moment, causing the robot to fall to the side. Here, however, the method showed great robustness. A lateral velocity disturbance of 0.1m/s was imposed on the trunk, but the biped robot did not fall to the side. Instead, the trunk suddenly changed its posture to nearly vertical, which caused the COM to move to the opposite position, and then slowly move back to the equilibrium position. The trajectories of these variables are shown in Figure 14.

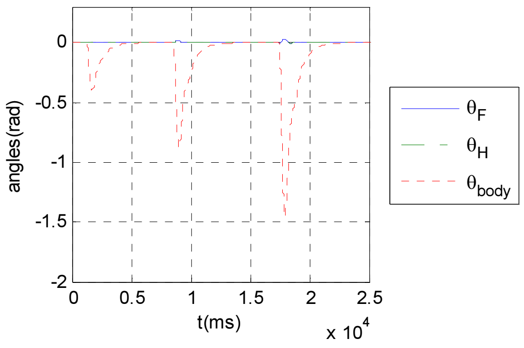


Figure 12. Pitch angle disturbance test of the biped robot

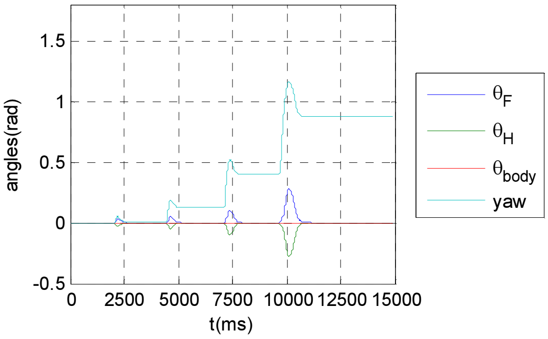


Figure 13. Yaw angle disturbance test of the biped robot

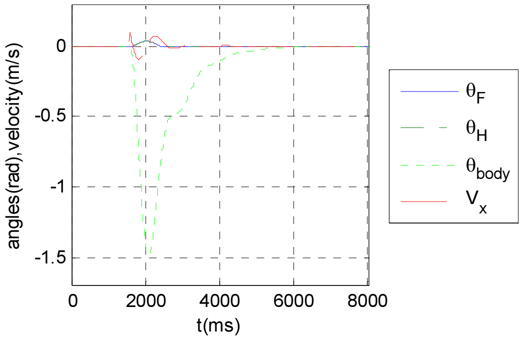


Figure 14. Lateral velocity disturbance test of the biped robot

5.2 Simulation of the 6-DOF Model

In this section, postural disturbance was tested. The angular velocity was set to 1rad/s and 2rad/s. Although the robot remained stable, its ability to resist disturbance seemed weaker than that of the 2-DOF biped model, and the final values of the angles did not return to exactly zero. This was because, in this case, the disturbance's angular velocity rotated around the X axis of the body rather than the diagonal line. The initial angular velocity was so large that the front leg lifted off the ground for a short time, and the touch-down point was a little further out, meaning that θ_{AF} and θ_{AH} drifted from zero. Curves of the logged data are shown in Figure 15.

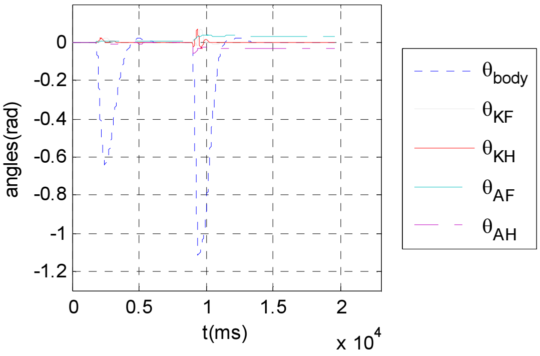


Figure 15. 6-DOF model with roll angular velocity disturbance

Yaw angular velocity disturbance was tested on the 6-DOF model (Figure 16). The angular velocities were set to 1rad/s, 2rad/s, 3rad/s, and 4rad/s, which were the same as for the 2-DOF biped model. Although it had more degrees of freedom, the robot was still able to remain stable when disturbed. Slippage also occurred, but the final yaw angle was not as large as it was for the 2-DOF biped model. This was because the lengths of the legs were variable; therefore, when the legs lifted off the ground, they extended a little, and this had the advantage of stopping the torsional motion.

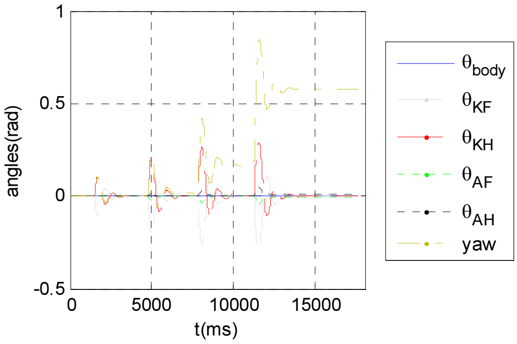


Figure 16. 6-DOF model with yaw angular velocity disturbance

The model's resistance to lateral velocity disturbances was also better than that of the 2-DOF biped robot (Figure 17), when lateral velocities of 0.1m/s and 0.2m/s were tested. This is because the lateral velocity disturbance could be seen as having two parts. One part was along the diagonal line, which was easily resisted by the active joints; the other part was perpendicular to the diagonal line, and this part could be stabilized only by adjusting the posture of the body, just like the lateral velocity disturbance in the 2-DOF biped model.

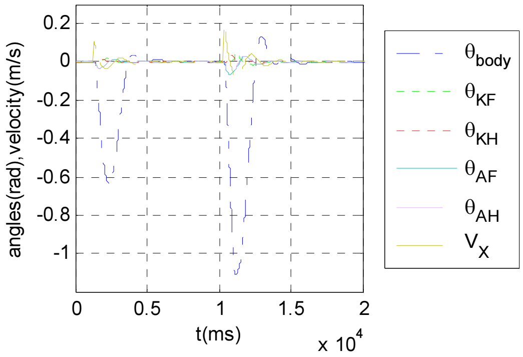


Figure 17. 6-DOF model with lateral velocity disturbance

5.3 Simulation of the Normal Model

The three kinds of disturbances were also tested on the normal model, and the angles of the joints were logged. The results are shown in the figures below. In Figure 18, the roll angular velocity was set to 1 rad/s and 2 rad/s respectively;

the maximum roll angles, the time taken to return to horizontal for the first time, and the duration of the overshoot were all similar to the virtual model. Figure 19 shows the data when yaw velocity disturbances of 1rad/s, 2rad/s, 3rad/s and 4rad/s were exerted on the model, and Figure 20 shows the model's behaviour after lateral velocities of 0.1m/s and 0.2m/s were applied. The data from the normal model were always similar to those of the virtual model.

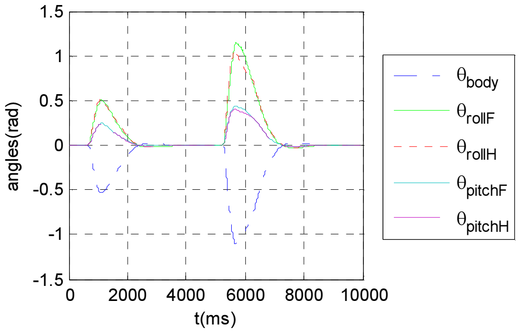


Figure 18. Trunk roll angular motion in the normal model

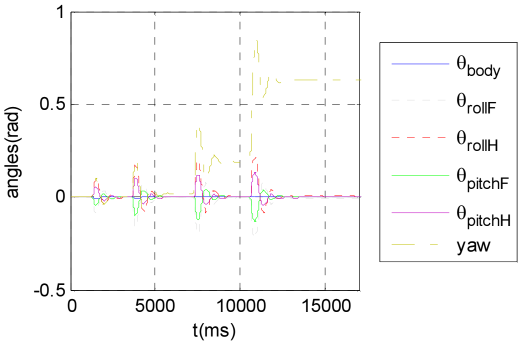


Figure 19. Trunk yaw angular motion in the normal model

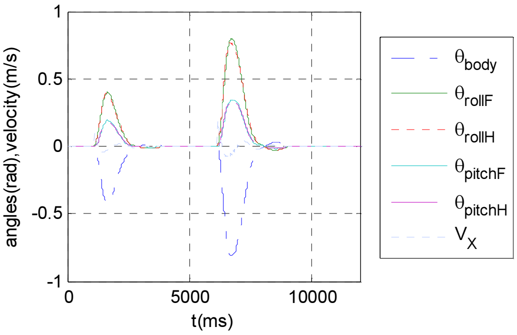


Figure 20. Trunk lateral velocity disturbance in the normal model

5.4 Simulation with Noise

In the real world, there is electromagnetic interference everywhere, so the signals from the sensors may be mixed with noise. To make the simulation more realistic, white noise was overlaid on the data from each ideal sensor, and the Box-Muller method was used to ensure the independence of the random numbers, which obeyed Gaussian distribution. The mean value of the noises was zero, the standard deviation of the joint angle sensors was set to 0.001rad, and the standard deviation of the posture angle was set to 0.005rad. This precision could be reached by sensors commonly used at present. A Kalman filter was used to smoothen the raw data mixed with noise. Figure 21 shows the process of the adjustment after roll angular disturbance, in simulation; the posture of the robot and the coordinate axes of each joint were clearly shown in Robot-Builder. Figures 22 to 24 give the simulation results. The roll angular motion test and the lateral velocity test both showed the effect of the noise; the posture of the trunk could not return to horizontal but fluctuated a little, although the robot did not lose balance. The robustness of the system against yaw angular motion was weakened; only 1rad/s and 2rad/s disturbances were tested, and the latter nearly toppled the robot.

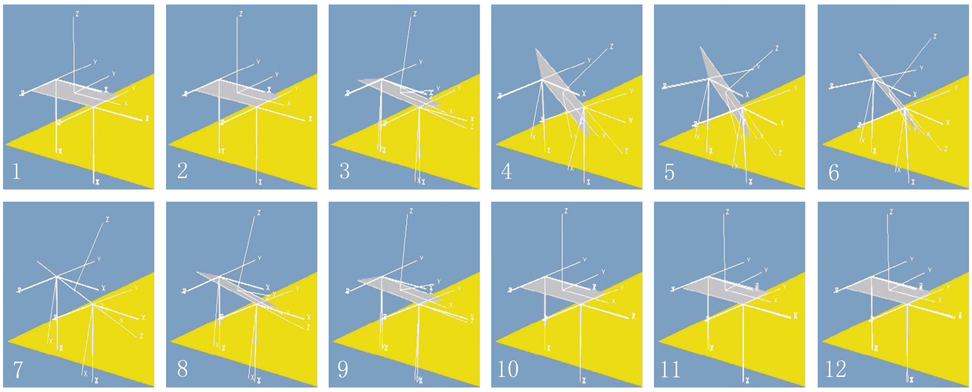


Figure 21. Snapshots of the simulation under roll angular motion

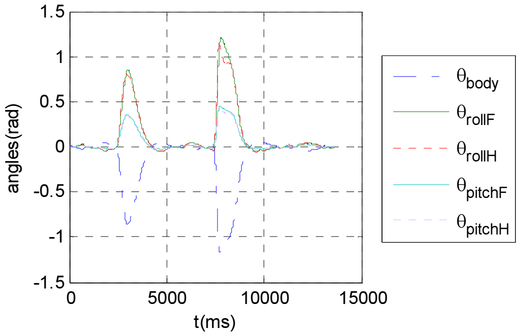


Figure 22. Trunk roll angular motion with noise

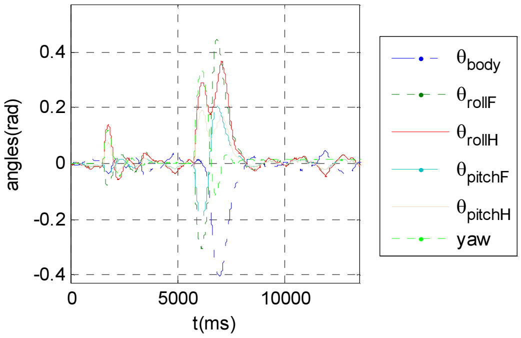


Figure 23. Trunk yaw angular motion with noise

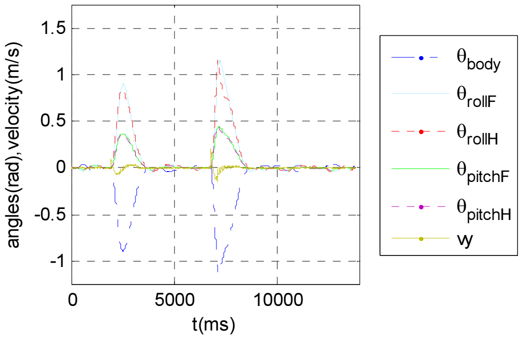


Figure 24. Trunk lateral velocity disturbance with noise

6. Conclusions

Balance control is a very important consideration for legged robots; most of them maintain balance with a support polygon formed by large feet or more than three legs, or they are hopping robots that need to jump all the time to keep balance. In this paper, a novel method of balance control was proposed to stabilize the motion of a quadruped robot. The model uses just two diagonal legs to support the body, and each leg makes contact with the ground at only a single point, so neither a support polygon nor jump motion is involved. Several models had been built to resolve the above problem; these evolved step by step, before the diagonal leg supporting model was finally built. In the proposed method, LQR control was used to stabilize the systems. In order to control the model with six DOFs, control of the robot's whole body was separated into three independent parts, with each part stabilized by its own controller. VMC was used in the last model. Each model was tested under several types of disturbances in RobotBuilder, and the simulation results showed that they all had good stability; they were not only able to maintain their balance, but they also resisted these disturbances. In particular, these models showed great robustness against lateral velocity disturbances; the trunk would suddenly adjust its posture automatically to reduce lateral speed, and then return to a stable state. This method could also be applied to legged robots with other numbers of legs: biped robots could use the first model directly, and hexapod or octopod robots could also keep their balance using only two supporting legs.

7. Acknowledgements

This paper is supported by the National Natural Science Foundation of China (61233014, 61305130) and the China Postdoctoral Science Foundation (2013M541912).

8. References

[1] McGhee R B, Frank A A (1968) On the stability properties of quadruped creeping gaits. Mathematical Biosciences. 3: 331–351.

[2] Papadopoulos E G, Rey D A (1996) A new measure of tipover stability margin for mobile manipulators. Proceedings of IEEE International Conference on Robotics and Automation (22–28 April 1996; Minneapolis, Minnesota). IEEE. pp. 3111–3116.

[3] Vukobratovic M, Juricic D (1969) Contribution to the synthesis of biped gait. IEEE Transactions on Biomedical Engineering. 1: 1–6.

[4] Hu N, Li S, Huang D, Gao F (2014) Crawling gait planning for a quadruped robot with high payload walking on irregular terrain. Proceedings of the 19th World Congress of the International Federation of Automatic Control (24–29 August 2014; Cape Town, South Africa). IEEE. pp. 2153–2158.

[5] Zhang S, Gao J, Duan X, et al. (2013) Trot pattern generation for quadruped robot based on the ZMP stability margin. International Conference on Complex Medical Engineering. (15–19 July 2013; San Jose, California). IEEE. pp. 608–613.

[6] Zhang W, Zhang L (2012) Quadruped robot omnidirectional walking on a slope based on static balance. Advanced Materials Research. 591: 1436–1440.

[7] Sudheer A, Vijayakumar R, Mohanda K (2012) Stable gait synthesis and analysis of a 12-degree of freedom biped robot in sagittal and frontal planes. Journal of Automation, Mobile Robotics & Intelligent Systems. 6: 36–44.

[8] Loc V, Koo I M, Tran D T, et al. (2012) Body workspace of quadruped walking robot and its applica-

bility in legged locomotion. Journal of Intelligent & Robotic Systems. 67: 271–284.

[9] Kamikawa K, Takubo T, Mae Y, et al. (2012) Omnidirectional gait of multi-legged robot on rough terrain by following the virtual plane. Journal of Robotics and Mechatronics. 24: 71–85.

[10] Geva Y, Shapiro A (2012) A combined potential function and graph search approach for free gait generation of quadruped robots. International Conference on Robotics and Automation. (14–18 May 2012; St. Paul, Minnesota). IEEE. pp. 5371–5376.

[11] Gagliardini L, Tian X, Gao F, et al. (2012) Modelling and trajectory planning for a four legged walking robot with high payload. Social Robotics. 1: 552–561.

[12] Chen S, Ko C, Li C, et al. (2012) Stair climbing in a quadruped robot. International Journal of Automation and Smart Technology. 2: 11–20.

[13] Loc V, Koo I M, Trong T D, et al. (2010) A study on traversability of quadruped robot in rough terrain. International Journal of Control, Robotics and Systems Conference. 1707–1711.

[14] Raibert M (1986) Legged Robots That Balance. Cambridge, Massachusetts/London, UK: The MIT Press.

[15] Raibert M H, Brown Jr H B, Chepponis M, et al. (1989) Dynamically stable legged locomotion [Internet]. Available from: http://dspace.mit.edu/handle/1721.1/6820#files-area. Accessed on 21 Oct 2014.

[16] Raibert M H, Chepponis M, Brown Jr H B (1986) Running on four legs as though they were one. IEEE Journal of Robotics and Automation. 2: 70–82.

[17] Emura T, Arakawa A (1991) Attitude control of a quadruped robot during two legs supporting. Robots in Unstructured Environments. Fifth International Conference on Advanced Robotics (19–22 June 1991; Pisa, Italy). IEEE. pp. 711–716.

[18] Erkaya S (2013) Trajectory optimization of a walking mechanism having revolute joints with clearance using ANFIS approach. Nonlinear Dynamics. 71: 75–91.

[19] Zhao Y, Bai Z F (2011) Dynamics analysis of space robot manipulator with joint clearance. Acta Astronautica. 68: 1147–1155.

[20] Erkaya S (2012) Investigation of joint clearance effects on welding robot manipulators. Robotics and Computer-Integrated Manufacturing. 28: 449–457.

[21] Zhu J, Ting K L (2000) Uncertainty analysis of planar and spatial robots with joint clearances. Mechanism and Machine Theory. 35: 1239–1256.

[22] Pratt J, Dilworth P, Pratt G (1995) Virtual model control of a bipedal walking robot [PhD thesis]. Cambridge, Massachusetts: Massachusetts Institute of Technology.

[23] Rodenbaugh S J (2003) RobotBuilder: a graphical software tool for the rapid development of robotic dynamic simulations [Master's thesis]. Columbus, Ohio: Ohio State University.

[24] McMillan S, Orin D E, McGhee R B (1995) DynaMechs—an object oriented software package for efficient dynamic simulation of underwater robotic vehicles. In Yuh J (ed.) Underwater Robotic Vehicles: Design and Control. TSI Press. pp. 73–98.