2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

Falling Prediction and Recovery Control for a Humanoid Robot

Tianqi Yang[1], Weimin Zhang[1,3], Zhangguo Yu[1,2], Libo Meng[1], Chenglong Fu[4] and Qiang Huang[1,2].

Abstract— **It is very easy for biped robots to fall down. Some previous studies have been carried out to detect the fall state and protect the robot from damage. But it is not enough to detect a fall. It is very important for the biped robot to predict whether it will fall in the future based on the current state. In this paper, we consider a fall state predicted problem for bipedal robots. Based on the D 'Alembert principle, this method can predict the fall state at the moment the biped robot deviates from the normal state in every conditions such as standing and walking. It can give the robot more time to recover from the unstable state or protect itself from damage. And its stable control strategy matching the proposed method is also proposed to protect the robot from falling. The result is verified via simulations.**

I. INTRODUCTION

Over the past few decades, many studies have been concerned with the walking stability of biped robots [1-3]. Most methods are based on data from a three-dimensional linear inverted pendulum (LIP) model. Because a biped robot is a complex dynamical system, the trajectory generated by the LIP model can not always guarantee the walking stability of a biped robot. In addition, the high center of mass (COM) and small contact area with the ground make it easy for the biped robots to fall down when it is disturbed by the outside world [4-10]. In fact, many conditions can lead to a change in the COM state of a robot. This is why we need a method that can predict whether a robot will fall down depending on its current state.

Because the biped robots are very easy to fall, there are a lot of researches to study the fall detection and the fall protection. Javier Moya proposed two separate instability detection methodologies that can be applied to the robot while it is standing or walking. The first step is to measure the trunk attitude to achieve fast instability detection [11]. The second method uses Hotelling's T-square statistics to perform anomaly detection with a predetermined statistical confidence level. JUuiz-Del-Solar J et al. use the LIP model, with an offset, to predict the robot's COM motion. If successful, this routine can prevent a robot from falling during the demonstration phase of DARPAs Robotics Challenge (DRC) Finals [12]. Xinjilefu X et al. determine an offset of a robot's current capture point which is called the corrected capture point (CCP) to save the robot from falling down twice during the DRC Finals [13].

But most previous studies can only detect the fall state which means that when the robot begins to fall it can detect the state. When the robot lose its stability, it is a divergent progress. It is very fast from the state that the robot lose its stability to the state of the collision between the robot and the ground especially when the robots are disturbed by a great external force. It is very important for the robot to predict its fall state. There are also some methods to predict the fall state. A falling prediction for the robot standing is proposed in [14] based on energy state. But it can only predict the standing falling. Another method use the multi-sensor to predict the falling state. It can also predict the standing falling and the method is complex so it is hard to meet fast prediction[15]. In this paper, our main contribution is to propose a new method to predict the fall state no matter in the robot standing or walking based on the current state in real time. This can give more time for the robot to recover from the fall state. Another contribution is to propose a new recovery control strategy for the robot based on the optimal control theory when the proposed method predicts that the robot will fall in the future.

In this paper, we propose a method to predict a robot's falling state based on its current state, which can be used while the robot is standing or walking. When a robot is standing, its state can be modeled as a single rigid body. When a robot is walking, its state can be modeled as a multi-rigid body system. The input of the system is the trajectory of a robot's COM under normal condition. The proposed can predict the deviation angle of the robot to decide whether it will fall. By predicting the fall state of the robot, the robot can adjust its foothold position and landing time, which not only ensures the robot's stability but also minimizes a robot's acceleration. At the end of the double support period (DSP), the state of a robot's COM can satisfy the divergence conditions of the LIP.

The remainder of the paper is organized as follows. In Section II, we evaluate the stability of a robot's standing and walking state when there is an interference. In Section III, we propose a method that takes the ZMP, which is based on the state of a robot after it has been disturbed by an external force, as the input of the system. This step can return a robot to a stable state in the shortest time. In Section IV, we provide some simulation results to verify the effectiveness of the method.

*This work was supported in part by the National Natural Science Foundation of China under Grant 61533004 and 61673069, and in part by the 111 Project under Grant B08043.

[1]Intelligent Robotics Institute, School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China. Corresponding author is Weimin Zhang, Email: zhwm@bit.edu.cn.

[2]Key Laboratory of Biomimetic Robots and Systems (Beijing Institute of Technology), Ministry of Education, Beijing 100081, China.

[3]International Joint Research Laboratory of Biomimetic Robots and Systems (Beijing Institute of Technology), Ministry of Education, Beijing 100081, China.

[4]Department of Mechanical and Energy Engineering, Southern University of Science and Technology, GuangDong, 518055.

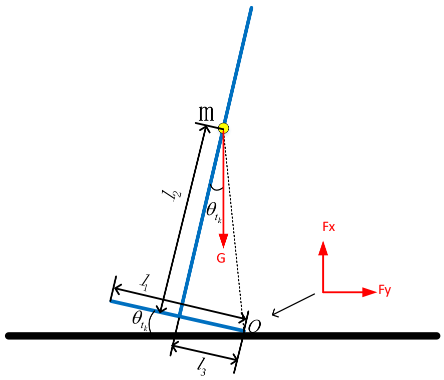

Fig. 1. fall detection model when standing

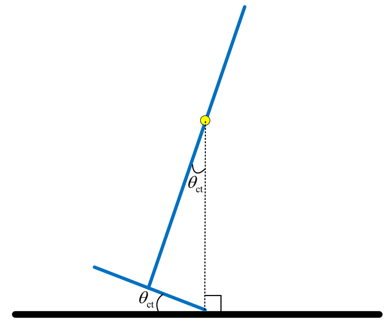

Fig. 2. Critical fall state when standing

II. The standing model of the fall detection algorithm

The classical mechanics [16] clearly explain the motion law of the object. The proposed fall predicted method is based on the classic mechanics to analyze the motion of the biped robot in the non-inertial reference. The biped robot is regarded as a rigid system.

A. standing model of fall detection algorithm

When a robot is standing, it can be modeled as a single rigid system as shown in Fig. 1. The angular position and velocity of a robot that deviate from the vertical direction θ_{t_k} at time t_k can be measured via angular sensors. The term m represents the robot's mass, which is concentrated on the COM. The terms l_1, l_2 and l_3 represent the robot's size parameters. We first generate a dynamic expression of a single rigid system where the initial angular position $\theta(t_{\text{in}})$ and the angular velocity $\dot{\theta}(t_{\text{in}})$ are known

$$mg(\frac{l_1}{2}cos\theta(t) - l_2 sin\theta(t)) = -J_o \frac{d\dot{\theta}(t)}{dt}, \tag{1}$$

Where

$$\theta(t_{\text{in}}) = \theta_{t_k} \quad \dot{\theta}(t_{\text{in}}) = \dot{\theta}_{t_k}$$

where J_o is the moment of inertia for the robot at the foot position o. The angular position and velocity can be obtained by integrating the two sides of Eq. (1).

$$\frac{1}{2}(\dot{\theta}^2 - \dot{\theta}_{t_k}^2) = \frac{mg}{J_o}[l_2(cos\theta - \cos\theta_{t_k}) - \frac{l_1}{2}(sin\theta - sin\theta_{t_k}) \tag{2}$$

From Eq. (2), we can establish the future relationship between θ and $\dot{\theta}$ is based on the robot's current state. A standing robot can easily predict the likelihood of that it will reach a critical state and fall. This critical state is shown in Fig. 2. When the COM is located at the top of the foot's support point, the angular position is characterized by θ_{ct} and its angular velocity is characterized by $\dot{\theta}_{\text{ct}}$. The resulting state vector, characterized by the angular position and velocity pair, is just zero. We define this state as the critical state a robot reaches before falling. If the robot's angular position θ equals to the critical angle θ_{ct}, we only need to compare the angular velocity with the critical angular velocity, which is always zero when a robot is standing still. There are three possible outcomes for each element of the state vector. These are shown in Eq. (3).

$$\theta{=}\theta_{\text{ct}}, \begin{cases} \dot{\theta} > 0 & \text{falling down} \\ \dot{\theta} = 0 & \text{critical state} \\ \dot{\theta} < 0 & \text{stablility} \end{cases} \tag{3}$$

B. Fall detection model for a walking robot

When a robot is walking, its components are moving relative to each other. Therefore, the robot can not be seen as a single rigid body. The standing model can be regarded as a special state. So we will propose a general falling prediction method in this section.

When modeling a robot walking, the robot can be characterized by a multi-rigid model, which can easily analyze a robot's state vector. As shown in Fig.3, when a robot walks on a flat floor, the position of its foot is planned ahead (at the end of the single support period). If the robot is disturbed, the foot will not be parallel to the ground, which means that the position state prediction will be incorrect since the foot drops onto the ground at the wrong position. In addition, any interference can also change the current position state of the robot, including the position and velocity of the COM. These incorrect state can very easily lead to a robot falling. Thus, a robot needs to adjust its gait efficiently when it has an interference.

As shown in Fig. 4, the foot of the swing leg should be adjusted to the planned height h. At any moment, the angles of the joints in the legs of the robot are known and can be classified into two sets $\theta_{\text{L}}(t) = (\ \theta_{1,\text{L}}(t) \ \ \ldots \ \ \theta_{6,\text{L}}(t)\ )$ and $\theta_{\text{R}}(t) = (\ \theta_{1,\text{R}}(t) \ \ \ldots \ \ \theta_{6,\text{R}}(t)\ )$. The position and posture of the COM in the world coordinate system only depends on the angle of the support leg and its deviating angle θ_{ct}. After we know the position of the trail leg, we can calculate the six angles of the trail leg according to inverse kinematics. To ensure that a foot is always parallel to the ground, we should define a compensation term $\theta_{\text{cp}} = -\theta_{\text{ct}}$. The angle θ_{ct} can be measured by the sensor so the trail leg angles can be updated in time to move the foot to a predetermined trajectory, no matter what kind of external disturbance the robot is subjected to.

As shown in Fig. 4, the robot is disturbed by an external force and it has deviated from its angular position and normal

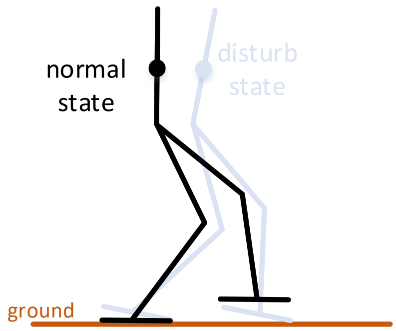

Fig. 3. the robot is disturbed when walking

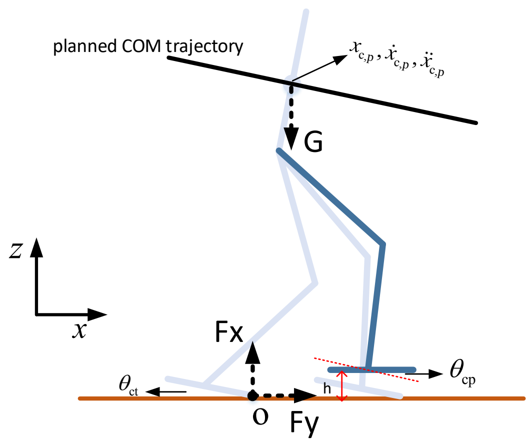

Fig. 4. Walking model analysis and swing adjustment strategy based on its current posture

posture θ_{ct}. The COM also moves as planned, which is characterized by the black line in Fig. 4. If the robot is dumped forward, the support point is always o. Then we can get the motion of the whole time based on the D 'Alembert principle.

$$J(t)\ddot{\theta}(t)-mg(cos\theta(t)\cdot(x_c-l_1)-sin\theta(t)\cdot z_c)=-mz_c\ddot{x}_c \quad (4)$$

where

$$\theta(t_{\text{in}})=\theta_{ct} \quad \dot{\theta}(t_{\text{in}})=\dot{\theta}_{ct}$$

where z_c is the distance between the COM and the planar surface of the floor. $J(t)$ is the rotational inertia about the supporting point of the robot's foot, which depends on the robot's current state. The state can be expressed by the joint angles $\theta_{\text{L}}(t)$ and $\theta_{\text{R}}(t)$. The terms $\theta(t_{\text{in}})$ and $\dot{\theta}(t_{\text{in}})$ are the initial conditions of the system, which are collected by the sensor in the time domain.

$$J(t)=J(\theta_{ij}(t),t) \qquad (i=R,L) \quad (j=1,2,...,6) \qquad (5)$$

$$J(\theta_{ij}(t),t)=\sum_{i=L}^{R}\sum_{j=1}^{6}(J_c(\theta_{ij})+m_{ij}s_{\text{c,o}}^2(\theta_{ij})) \qquad (6)$$

where $J_c(\theta_{ij})$ is the rotational inertia of the COM associated with ij. The parameter $s_{\text{c,o}}(\theta_{ij})$ is the distance between the COM associated with ij and the supporting point of the foot.

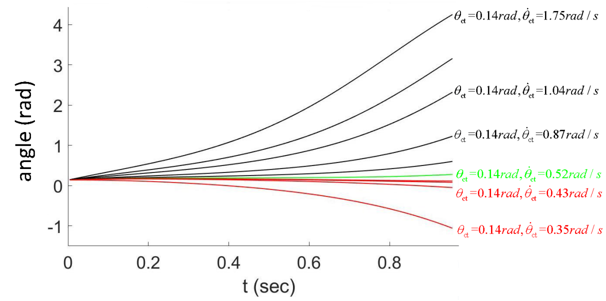

Fig. 5. Different interferences when the robot is walking

In control theory, when we consider $\ddot{x}_c$ as the input of the system, it is possible for the robot to predict its future state, i.e., the position and velocity of the COM and the angle and angular velocity of the robot. The system of equations is not a system of linear differential equations. As a result, an analytical solution can not be obtained. Therefore, Eq. (5) needs to be discretized. Because the initial conditions and the input $\ddot{x}_c$ are known, we can solve the numerical system of equations. The resulting control cycle is characterized by δ and ($t=k\delta \quad (k=1,2,....n)$).

We can predict future states of the robot while it is walking, and disturbed by an external force. But unlike the standing case, walking is a dynamic process; thus, there is no fixed critical falling attitude. Predicting a falling state is much more complex when walking than when standing. We only consider the dumped falling in the longitudinal direction as an example and it can also be applied in the lateral direction. Because we can only calculate the numerical solution of Eq. (7), we should analyze the numerical solution to find a decision parameter.

We give the robot different sizes of force from small to large for the same deviation angle. As is shown in Fig.5, the deviation angle is $0.14rad$ and for different forces, the initial angle velocity is also different. The green line is the critical state which also means the maximum interference that the robot can bear in this deviation angle. When the external interference is greater than this value, the robot will fall and when the external interference is smaller than this value, the robot will not fall and the angle will come back to the stable state in the future. In Fig.5, this method can predict the falling state about $0.2s-0.4s$ ahead of time.

But for different deviation angles and for different robots, the critical state is also different. So it is very important to judge whether the robot will fall down in the future when the critical state is not known. Through a lot of prediction calculations, we can get that the tendency is the same. The characteristic of derivative can be used to judge whether or not to fall down. The criterion is shown in formula (7).

$$\frac{d\theta(t_c)}{dt}=0,\begin{cases}\frac{d\theta(t)}{dt}\geq 0 & t\geq t_c \quad \text{falling down}\\ \frac{d\theta(t)}{dt}=0 & t\geq t_c \quad \text{critical state}\\ \frac{d\theta(t)}{dt}\leq 0 & t\geq t_c \quad \text{stablility}\end{cases} \qquad (7)$$

Where t_c can be seen as the critical falling state. According

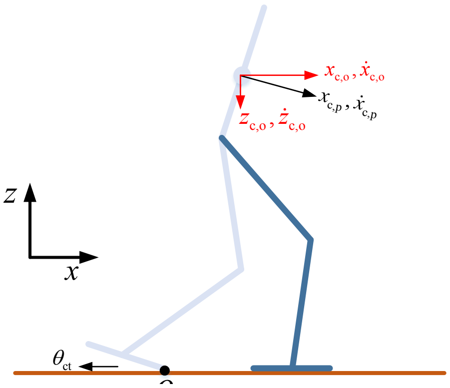

Fig. 6. The calculation of the state parameters of a robot

to the formula(7), it is easy to predict the falling state of the robot. This method can make sense in real time no matter how long the external force acts on the robot. It only depends on the current state of the robot including the deviation angle and the angle velocity.

III. The stability control strategy based on optimal control

A. The stable control strategy

In Section II, we propose a monitoring strategy that can predict when the robot will fall. If the algorithm predicts that the robot will not fall in the future, the robot will only adjust its foot height h and the compensation angle θ_{cp} as shown in Fig. 4. The compensation angle is used to ensure the foot of the swing leg moves in accordance to the predetermined trajectory in the world coordinate system. If θ_{cp} exists, then the height of the COM in the world coordinate will also change as shown in Fig. 6. The term $x_{\mathrm{c,o}}$ is the position of the COM in the forward direction in the world coordinate, while $z_{\mathrm{c,o}}$ is the position in the vertical direction.

$$\begin{cases} x_{\mathrm{c,o}}(t) = f_1(\theta_{ct}(t), \theta_S(t)) \\ \dot{x}_{\mathrm{c,o}}(t) = f_2(\theta_{ct}(t), \dot{\theta}_{ct}(t), \theta_S(t), \dot{\theta}_S(t)) \end{cases}$$
$$\begin{cases} z_{\mathrm{c,o}}(t) = g_1(\theta_{ct}(t), \theta_S(t)) \\ \dot{z}_{\mathrm{c,o}}(t) = g_2(\theta_{ct}(t), \dot{\theta}_{ct}(t), \theta_S(t), \dot{\theta}_S(t)) \end{cases} \tag{8}$$

$$m_{\mathrm{d}}\ddot{h}(t) + d_{\mathrm{d}}\dot{h}(t) + k_{\mathrm{d}}h(t) = f_{\mathrm{d}}(t) \tag{9}$$

where

$$h(t) = z_{\mathrm{c,o}}(t) - h_{COM}(t), f_{\mathrm{d}}(t) = f_z(t) - mg$$

where m_{d} is the inertial coefficient of the spring damping model, d_{d} is the damping coefficient and k_{d} is the Elastic coefficient. The terms f_z characterizes the contact force between the sole and the ground in the vertical direction.

Eq. (9) works at the end of the single foot support period, which is also the beginning of the double support period. Because the height of the COM is not constant, the COM will have a vertical speed. In order to avoid the collision between the robot and the ground, we use the spring damping model. In the double support period, the support area is larger than that of the single support period. We can also use the LIP model to plan the trajectory of the COM, where we can also assume that the height of the COM is constant.

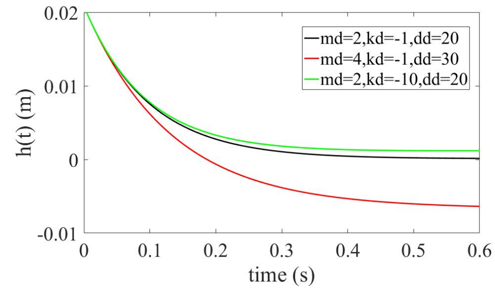

Fig. 7. Different results of COM error for different parameters

As is shown in Fig.7, for different parameters, the centroid height error will converge to different values. What we need is that the final error of the COM height is 0 so the black line is suitable for the result.

B. Optimization of COM trajectory

If the robot is disturbed by an external force, the actual COM state will also change. Therefore, it is essential for the robot to adjust its state to a normal state. Based on the falling detection method proposed in Section II, if the robot will not fall down in the future, the single support period is consistent with the planning time, but if not, the single support period will change based on the state of the COM. As is mentioned above, the falling is a divergent state so we need to swing the leg to the ground ahead of time to make it easier to restore to the stable state. Because in the single support period robots have only toe or heel contact with the ground so it is hard to recover the stabile state. Our control method plays a role in the double support period.

The dynamic of the LIP has been discussed in [3-4] and in this paper, our recovery control is also based on that. The step length and the step time are very important for the robot and this has been discussed in reference [3]. When the planned ZMP trajectory satisfies Eq. (10):

$$x_{\mathrm{z}}(t) = \sum \alpha_i x^i_{\mathrm{z,step}}(t) \tag{10}$$

where $x^i_{\mathrm{z,step}}$ is the step function of ZMP at time t_i, which is $x^i_{\mathrm{z,step}}(t) = x_{z,step}(t - t_i)$ and α_i is the weight coefficient and it also means length of every step. The whole number i is the ith step. By reviewing previous work, we can provide a single support period expression.

$$T^i_s = -\frac{1}{\omega}\ln\frac{1}{\alpha_i}[x_{\mathrm{c}}(t_s) + \frac{\dot{x}_{\mathrm{c}}(t_s)}{\omega}] \tag{11}$$

We can determine the time that the foot of the swing leg touches the ground via Eq. (11). When the step time is determined, the foot length can also be determined via Eq. (12)

$$\alpha_{\mathrm{i}} = e^{\omega t_{\mathrm{i}}}\left[x_{\mathrm{c}}(t_s) + \frac{\dot{x}_{\mathrm{c}}(t_s)}{\omega}\right] \tag{12}$$

When the robot is disturbed by an external force, the actual state of the COM is also different from the planned state. The dynamics of the LIP is analyzed in reference[4]. Once the trajectory of the ZMP has been determined, the initial state should satisfy initial constraint to guarantee the stability of the robot. In this paper, we only consider this stability criterion for a single support period. For the double support period, we should adjust the posture of the robot in an optimal way because the double support area is larger than that of the single support period.

The LIP model describes the relationship between the COM and ZMP. We use the acceleration of x_c as the system input to adjust the COM state, $\mathrm{u}(t)=\ddot{x}_c(t)$. The system of equations that characterized the LIP model is

$$\begin{bmatrix} \dot{x}_c(t) \\ \ddot{x}_c(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_c(t) \\ \dot{x}_c(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathrm{u}(t) \quad (13)$$

The intensity of the change in the state of the COM directly affects the stability of the robot when it is walking. When the robot is out of the original state, the initial condition will not satisfy the initial stability constraint. In this paper, we propose a minimum state change rate constraint method to control the robot while it returns to a normal state.

Based on the LIP model, the input $u(t)$ is not arbitrary and its constraint can be expressed by Eq. (14).

$$\ddot{x}_{\min} \leq u(t) \leq \ddot{x}_{\max} \quad (14)$$

where

$$\ddot{x}_{\min} = \omega^2(x - x_{zmp}^{\max})$$
$$\ddot{x}_{\max} = \omega^2(x - x_{zmp}^{\min})$$

The initial constraint condition

$$x_c(tin) = x_{c,o}(t_e)$$
$$\dot{x}_c(tin) = \dot{x}_{c,o}(t_e)$$

where $x_{zmp}^{\min}$ and $x_{zmp}^{\max}$ are the boundary of the support region in the double support period. The boundary is related to the current position of the COM. $x_{c,o}(t_e)$ and $\dot{x}_{c,o}(t_e)$ are the actual state of the COM at the end single support period which is calculated based on formula (8). What we need in the proposed method is to minimize the rate of change in the state to define the objective function

$$J = \int u^2(t)dt \quad (15)$$

As we know. if the robot can walk stably in the initial condition and The COM state must satisfy initial constraint. Therefore, the terminal cross-section condition is

$$x_c(t_e) + \frac{1}{\omega}\dot{x}_c(t_e) = \omega \int_{t_s}^{\infty} e^{-\omega(t-\tau)} x_z(\tau) d\tau$$

IV. SIMULATION AND DISCUSSION

To validate the illustrated approach, the falling detection and walking algorithm has been tested on the BHR-6 robot in the V-REP simulation environment. The BHR-6 robot is the latest humanoid robot of Beijing Institute of Technology which is mainly concentrated on the multi-motion. It can walk, craw, roll and recover form the fall detection.

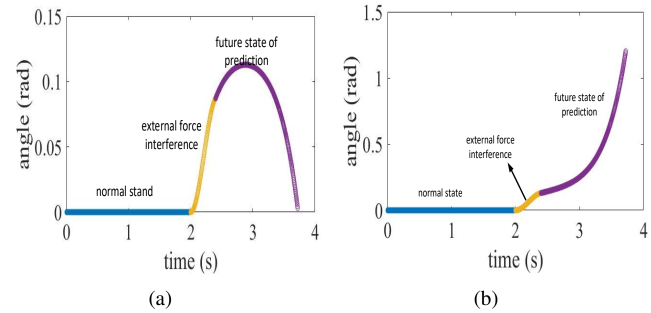

Fig. 8. stand fall detection result ((a): small extern force (b)large extern force

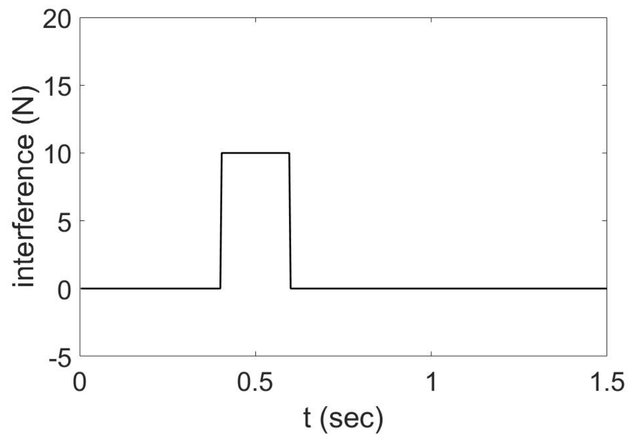

Fig. 9. interference acting on the walking robot

We first test the standing fall predicted method in our robot. We give two different forces acting on the robot when it is standing. The time of action is 0.3*s*. As is shown in Fig.9, the left figure is the result of the acting force 20*N*. At the end of the force action time, we predict the future state of the robot according to the current state. It is obvious that the robot will not fall down in the future so the robot will come back to the stable state eventually. But for the right figure in Fig.9, the acting force is 30*N* so the predicted result is that the robot will fall down at the time 2.8*s* and this gives the robot more about 0.3 to do the protective action.

Second, we test the walking falling predicted method. When the robot is walking, it is more easy to fall down than stand up. Besides, when the robot loses stability during walking, the fall when walking is more faster than taht when standing. Thus, the method that detects potential falls is more important to the robot when it is walking. While the robot is walking, its trunk state is always kept perpendicular to the ground, making the position of the COM agreement as planned. In Section II, We have provided the criterion to determine whether the robot will fall down or not. This criteria can determine all the cases that the robot may take place. And we also verify the correction of the detection method by giving an external force 10*N* whose action time is 0.2*s* as is shown in Fig.10. The prediction result is shown in Fig. 10.

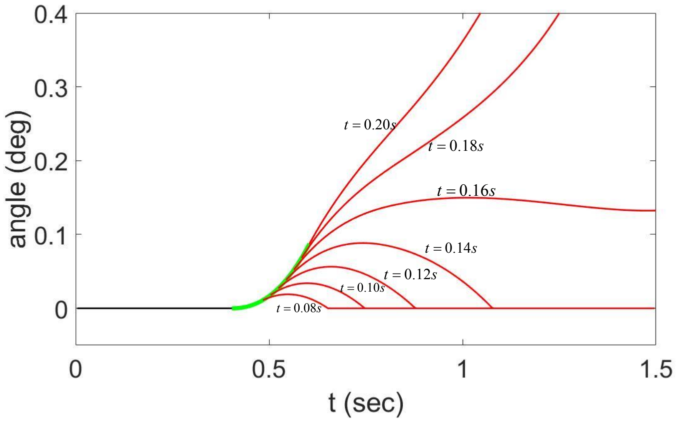

Fig. 10. fall detection result when walking

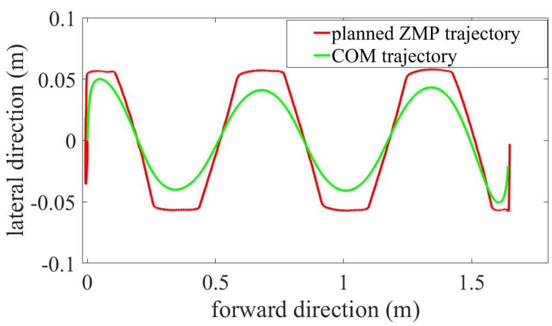

Fig. 11. walking cycle

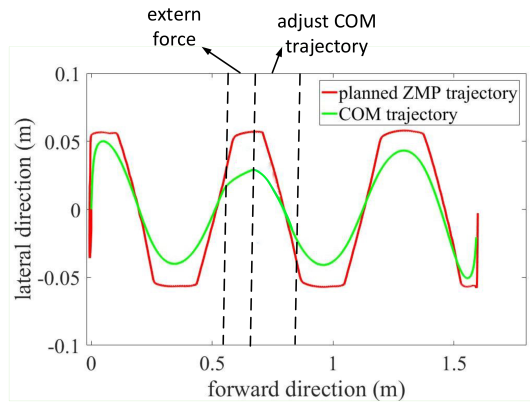

Fig. 12. COM and ZMP trajectories when disturbed by extern force

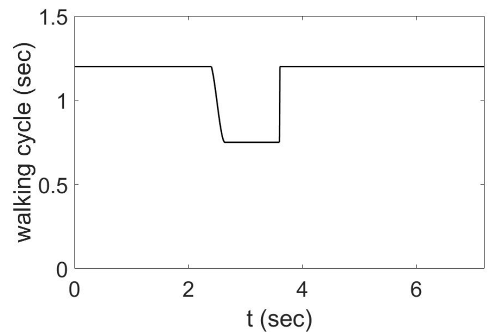

Fig. 13. COM and ZMP trajectories when disturbed by extern force

As we can see in Fig.11, when the force acts on the robot, our method will predict the future state in real time. In this figure, we predict the future state every $0.2s$. Because the extern force is not very large, when the acting time of the force is less than $0.16s$, the predicted result is that the robot will not fall in the future. When the acting time of the force is $0.16s$, the state of the robot is a critical fall state. And when the force continues to act on the robot, the predicted result is that the robot will fall down in the future. The proposed method can predict the fall sate in advance $0.2s - 0.3s$.

When the robot loses the stability, it will adjust its posture to prevent itself from falling. So we also do some simulations to verify the correction of the stable control strategy. First, we plan the ZMP trajectory and use the LIP model to generate the trajectory of the COM with a walking speed $1.0km$ per hour as is shown in Fig.12.

Then we give an extern force at the beginning of the third step when the robot is walking. The force is also $10N$ and the time is $0.2s$. As is shown in Fig.13, at the beginning of the third step, the robot is disturbed by extern force. The trajectories of the COM and ZMP are both changed. The robot's centroid trajectory will move forward quickly when the robot is dumped forward. For the normal walking, the walking step is $0.33m$ and the walking cycle is $1.2s$. But the change od the COM state will lead to the change of the walking cycle. Basd on the formula(11), the walking cyclce of the third step becomes $0.8s$ and the walking step becomes 0.26m. At the end of the third step, the COM state will return to the normal based on the optimization of the COM trajectory. And the robot can also walk in the normal state. The Fig.14 shows the walking cycle changes of the walking when disturbed by extern force.

V. Conclusion

In this paper, we propose a fall detection method and stable control method. Our method has the following major contributions:

1). The fall detection method can predicted the future state of the robot so the robot have more time to adjust its posture to protect itself from damage.

2). The stable control method can optimize the state of the COM when the robot will fall down.

References

[1] Huang Q, Yokoi K, Kajita S, et al. Planning walking patterns for a biped robot. Automation IEEE Transactions on Robotics, 2001, 17(3):280-289.

[2] Kajita S, Tani K, Kobayashi A. Dynamic walk control of a biped robot along the potential energy conserving orbit. Robotics, Automation IEEE Transactions on, 1987, 8(4):431-438.

[3] Lanari L, Hutchinson S, Marchionni L. Boundedness issues in planning of locomotion trajectories for biped robots. Ieee-Ras International Conference on Humanoid Robots. IEEE, 2014:951-958.

[4] Lanari L, Hutchinson S. Planning desired center of Mass and zero moment point trajectories for bipedal locomotion. Ieee-Ras, International Conference on Humanoid Robots. IEEE, 2015:637-642.

[5] Stephens B J, Atkeson C G. Push Recovery by stepping for humanoid robots with force controlled joints. Ieee-Ras International Conference on Humanoid Robots. IEEE, 2010:52-59.

[6] Stephens B. Push Recovery Control for Force-Controlled Humanoid Robots. Dissertations Theses - Gradworks, 2013.
[7] Herdt A, Perrin N, Wieber P B. Walking without thinking about it. IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2010:190-195.
[8] Kajita S, Kanehiro F, Kaneko K, et al. Biped walking pattern generation by using preview control of zero-moment point. IEEE International Conference on Robotics and Automation, 2003. Proceedings. 2003:1620–1626.
[9] Wieber P B. Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations. Ieee-Ras International Conference on Humanoid Robots. IEEE, 2006:137-142.
[10] Lanari L, Hutchinson S. Inversion-based gait generation for humanoid robots. Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2015:1592-1598.
[11] Javier Moya, Javier Ruiz-del-Solar, Marcos Orchard, et al. Fall Detection and Damage Reduction in Biped Humanoid Robots[J]. International Journal of Humanoid Robotics, 2015, 12(01):1550001-.
[12] JRuiz-Del-Solar J, Moya J, Parra-Tsunekawa I. Fall detection and management in biped humanoid robots. IEEE International Conference on Robotics and Automation. 2010:3323-3328.
[13] Xinjilefu X, Hayward V, Michalska H. Stabilization of the spatial double inverted pendulum using stochastic programming seen as a model of standing postural control. 2009.
[14] Li Z, Zhou C, Castano J, et al. Fall Prediction of legged robots based on energy state and its implication of balance augmentation: A study on the humanoid. IEEE International Conference on Robotics and Automation. 2015:5094-5100.
[15] Subburaman R, Lee J, Caldwell D G, et al. Multi-sensor based fall prediction method for humanoid robots. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. 2017:102-108.
[16] Athorne C. Fundamental principles of classical mechanics: a geometrical perspective. 2014, 57(2):1-4.
[17] Fogarty J, Hudson S E, Atkeson C G, et al. Predicting human interruptibility with sensors. ACM Trans. on Computer-Human Interaction, 2005, 12(1):119-146.
[18] Scianca N, Cognetti M, Simone D D, et al. Intrinsically stable MPC for humanoid gait generation. Ieee-Ras, International Conference on Humanoid Robots. IEEE, 2017:601-606.
[19] Harada K, Kajita S, Kanehiro F, et al. Real-Time Planning of Humanoid Robot's Gait for Force-Controlled Manipulation. IEEE/ASME Transactions on Mechatronics, 2007, 12(1):53-62.