Real Time Biped Walking Gait Pattern Generator for a Real Robot

Feng Xue[1], Xiaoping Chen[1], Jinsu Liu[1], and Daniele Nardi[2]

[1] Department of Computer Science and Technology,
University of Science and Technology of China,
Hefei, 230026, China
[2] Department of Compute and System Science,
Sapienza University of Roma,
Via Ariosto 25, Roma, Italy

Abstract. The design of a real time and dynamic balanced biped walking gait pattern generator is not trivial due to high control space and inherently unstable motion. Moreover, in the Robocup domain, robots that are able to achieve the goal footstep in a short duration have a great advantage when playing soccer. In this paper, we present a new technique to realize a real time biped walking gait pattern generator on a real robot named Nao. A Zero Moment Point (ZMP) trajectory represented by a cubic polynomial is introduced to connect the goal state (the position and the velocity of the CoG) to the previous one in only one step. To apply the generator on the real robot Nao, we calculate the compensation for two HipRoll joints in a theoretical way by modeling them as elastic joints. The nao of version 3.3 is used in the experiments. The walk is intrinsically omnidirectional. When walking with step duration 180ms, the robot can respond to the high level command in 180ms. The maximum forward speed is around 0.33m/s. The maximum backward speed is around 0.2m/s. The maximum sideways speed is around 0.11m/s. The maximum rotational speed is around 90°/s.

Keywords: Biped Walking, ZMP, 3D Inverted Pendulum, Elastic Joints.

1 Introduction

The design of a real time and dynamic balanced biped walking gait pattern generator is not trivial due to high control space and inherently unstable motion. Moreover, in the Robocup domain, robots that are able to respond to higher level commands in a small delay and achieve the goal footstep in a short duration have a great advantage when playing soccer.

Zero Moment Point (ZMP) [15] has been applied widely to ensure dynamic balance in many biped walking gait pattern generators. ZMP is a point about which the sum of the horizontal ground reaction moment due to ground reaction force is zero. It is an effective criterion for measuring instantaneous balance. The state-of-art biped walking gait pattern generators usually plan the Center of

T. Röfer et al. (Eds.): RoboCup 2011, LNCS 7416, pp. 210–221, 2012.

Gravity (CoG) trajectory according to the given ZMP trajectory which satisfies this criterion.

An approach based on control theory has been proposed by Kajita et al. [9] in which the CoG is designed to converge at the end of previewing period. Stefan et al. [3] also used the preview controller and handled the sensor feedback in a different, more direct and intuitive way. Due to the inner tracking error of the preview controller, the magnitude of the preview gain needs around 1.5 seconds to converge. Thus, it is not suitable for a motion that changes its direction or speed in high frequency, such as in RoboCup domain.

Colin et al. [6] proposed a CoG-based gait in which CoG and foot trajectory are represented as functions with lots of parameters, instead of following the dynamic equation of the robot. The notion of assuring a dynamic balanced gait is not discussed. In the following work [5], they used basic 3D-LIPM theory and the ZMP is set at the origin of the foot for each step resulting in discontinuous ZMP trajectory.

Kajita et al. [8] proposed an analytical technique by modeling the robot as a 3D Linear Inverted Pendulum (3D-LIP). Jinsu Liu et al. [10] also uses this technique to achieve an online sampling search to switch between different walking commands. It also proposed a *Simplified Walking* in which a ZMP point namely *ZMP Decision* is used for each step. The previous state and the goal one are connected using two *ZMP Decisions* (two steps). We extend this work and use a ZMP trajectory represented by a cubic polynomial. Thus we do not need two steps, but only one to connect the goal state to the previous one.

In the work [14] and in the walking engine developed by Aldebaran company, the two HipRoll joints are compensated with a sinusoidal offset in an experimental way. In our generator, we consider the two joints as elastic joints which have been widely studied over decades in trajectory tracking tasks and regulation tasks for manipulators [12][17]. To our knowledge, we are the first to propose elastic joints on a humanoid robot. The main contribution of this part is that, we model the robot as a 3D-LIP, so that the computation cost is reduced a lot to attain a real time computation.

The remainder of the paper is organized as follows. Section 2 will present our walking engine focusing on the simultaneously planning of ZMP and CoG. Section 3 gives the theoretical way of compensating the two HipRoll joints. Experiments are shown in section 4. And we conclude in section 5.

2 Walking Engine

A gait pattern is usually a set of trajectories of the desired ZMP, the feet, the arms and the upper body.

Before the gait pattern generator, see Figure 1, a footstep planning algorithm is employed. In the case of a complex environment having obstacles, stairs and slopes, some heuristic or stochastic search algorithms are applied to generate footstep trajectory leading the robot from the start position to the goal position[2][11]. In the case of Robocup domain, it is relatively simple. The input command is a vector, denoted as $[\dot{x}, \dot{y}, \dot{\theta}]$, where $\dot{x}$ is the linear speed along

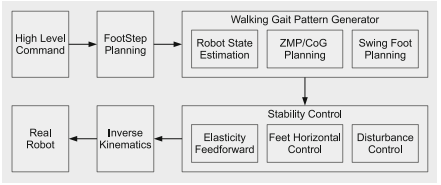

Fig. 1. The Architecture of Our Walking Engine

X axis, $\dot{y}$ is the linear speed along Y axis and $\dot{\theta}$ is the rotation speed around Z axis. Thus if the i^{th} footstep is $\bar{fs}^i = [\bar{x}^i, \bar{y}^i, \bar{\theta}^i]$, then $(i+1)^{th}$ footstep is $\bar{fs}^{i+1} = [\bar{x}^{i+1} + \dot{x}T, \bar{y}^{i+1} + \dot{y}T, \bar{\theta}^{i+1} + \dot{\theta}T]$, where T is the step duration. The walking engine usually handles input of a vector, in which each value specifies the related portion of the maximum speed, ranging between [-1.0, 1.0].

Now we have the footsteps, in another words, a set of support polygons, so the ZMP trajectory for reference is determined. In our previous work[10], a single ZMP namely *ZMP Decision* is placed at the center of each footstep.

We can see that the ZMP based approach is intrinsically omnidirectional. The method using Central Pattern Generator (CPG) suffers from this problem and additional techniques are studied [1].

Following the gait pattern generator is a stability controller which is in charge of rejecting the external disturbance applied from outside, and also the internal one due to the model error.

Then, an inverse kinematics solver is applied to generate the joint angles which are used to command the joint actuators. We solve it analytically in a standard way which is also reported in [6].

Given the cycle duration T as known constraint, we have N control cycles. Taking the Nao robot as an example, it can be controlled every 10ms, so, N is 50 if T equals to 0.5s. In our walking engine, if the robot reaches the last control cycle, it will read the cached user command, call footstep generation algorithm, set ZMP points for reference and call walking gait pattern generator.

2.1 Overview of Biped Walking Gait Pattern Generator

As mentioned above, the state-of-art biped walking gait pattern generator deals with a given ZMP trajectory which is in the support polygon set to guarantee a dynamic balanced gait. Then, the upper body trajectory which satisfies the desired ZMP trajectory is calculated using an approximate dynamics model.

In our walking gait pattern generator, as shown in Figure 2, the i_{th} step begins with a single support phase named $ss1^i$, following a double support phase named ds^i, and ends up with a single support phase named $ss2^i$. In $ss1^i$, the foot reaches the goal footstep of $(i-1)^{th}$ cycle. Compared to our previous work [10], the *Simplified Step* is divided into two swing phases: $ss2^{i-1}$ and $ss1^i$. As mentioned above, for the reason of highly response, we read the cached command

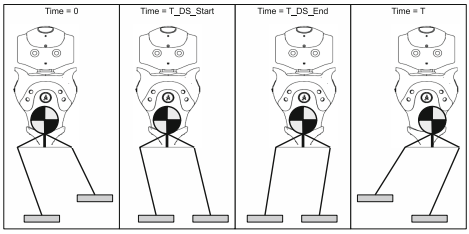

Fig. 2. A Step Is Formed by Two Swing Phases and One Double Support Phase

at the last control cycle of $ss2^i$, so the time duration for the next step is not known and the duration for $ss1^{i+1}$ could be fast or slow. Consequently, we set $ss2^{i-1}$ for lifting and $ss1^i$ for landing.

In our generator, we connected the goal CoG state (the position and the velocity of CoG) to the previous one in one step, so the goal user command can be reached in a small time which is determined by the duration of a step. A ZMP trajectory represented by a cubic polynomial is introduced to do this, which is talked about in Subsection 2.3 after the dynamic model in the next subsection.

2.2 Dynamics

In our point of view, no matter how accurate the dynamic model is, there will still be difference compared to the real robot, not to mention the accuracy of joint actuators which are usually not modeled. And, the model error can be treated as inner disturbance and handled together with the external disturbance. That's what the stability controller does. Extension of this controller is reported in detail in [16].

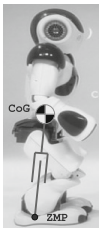

Fig. 3. Physics Model: The 3D Inverted Pendulum

The 3D Inverted Pendulum (3D-IP) proposed by Kajita et al. [8] is used to describe the approximate movement of a biped walking when the robot is supporting its body on one leg. A 3D-IP is an inverted pendulum which moves in 3D space. Assuming the current ZMP of the pendulum is at (p_x, p_y, p_z), the physics model shown in Figure 3 can be described as follows according to the definition of ZMP:

$$\ddot{x}(z - p_z) = (x - p_x)(g + \ddot{z}) \tag{1}$$

$$\ddot{y}(z - p_z) = (y - p_y)(g + \ddot{z}) \tag{2}$$

where g is the acceleration due to gravity, (x, y, z) and $(\ddot{x}, \ddot{y}, \ddot{z})$ are the position and the acceleration of the CoG.

In order to get a linear equation, a horizontal plane with intersection with Z axis z_p is applied. And the 3D-IP becomes a 3D Linear Inverted Pendulum (3D-LIP) [8]. The following equations hold:

$$\ddot{x} = \frac{g}{z_p}(x - p_x) \tag{3}$$

$$\ddot{y} = \frac{g}{z_p}(y - p_y) \tag{4}$$

In the case that the ZMP of X axis p_x is considered as a constant value p_x^*, the analytical solution of Equation 3 is (solution of Y axis is similar):

$$\begin{bmatrix} x_f(t) \\ \dot{x}_f(t) \end{bmatrix} = A(t) \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix} + [I - A(t)] \begin{bmatrix} p_{x^*} \\ 0 \end{bmatrix} \tag{5}$$

where $[x_i, \dot{x}_i]^T$ is the initial state, $[x_f(t), \dot{x}_f(t)]^T$ is the final state at time t, I is a 2×2 identity matrix and $A(t)$ is a state transition matrix which only depends on the duration t:

$$A(t) = \begin{bmatrix} cosh(qt) & \frac{1}{q}sinh(qt) \\ qsinh(qt) & cosh(qt) \end{bmatrix}, \quad q = \sqrt{\frac{g}{z_p}} \tag{6}$$

We now give the analytical solution when ZMP of X axis P_x is not constant, but represented by a cubic polynomial: $P_x = \sum_{i=0}^{3} a_i t^i$. The analytical solution of Equation 3 is:

$$\begin{bmatrix} x_f(t) \\ \dot{x}_f(t) \end{bmatrix} = A(t) \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix} + [I - A(t)] \begin{bmatrix} a_0 + 2a_2/q^2 \\ a_1 + 6a_3/q^2 \end{bmatrix} + \begin{bmatrix} \sum_{i=0}^{3} a_i t^i + 6a_3 t/q^2 - a_0 \\ 3a_3 t^2 + 2a_2 t \end{bmatrix} \tag{7}$$

2.3 Simultaneously Planning of ZMP and CoG

The length of the foot is around 14cm which is large compared to the short leg. Usually, the distance between each footstep in sagittal direction is lower than 8cm due to kinematics constraint caused by the relatively short leg. Thus, the heel of one foot will never exceed the toe of the other one, which is making a large support polygon. This inspires us of not using two *ZMP Decisions* to connect the desired CoG state to the previous one.

In consequence, we need to plan ZMP and CoG trajectory simultaneously. We first introduce the sagittal direction. The initial ZMP denoted as PY_i and the initial state of CoG denoted as $(y_i, \dot{y}_i)$ are decided by the last control cycle of i^{th}

footstep. The goal ZMP denoted as PY_{i+1} is determined by the $(i+1)^{th}$ footstep. In order to make a quick stop where the two foot positions along sagittal direction are the same, we put the goal CoG position denoted as y_f at the center of the $(i+1)^{th}$ footstep. In the work proposed by Harada et al. [7], they use this similar technique to connect the new trajectories to the current ones. However, in our case, the robot could change speed rapidly, so without control of the target CoG velocity may lead the robot to fall down. Thus, we introduce another parameter denoted as α which is the percentage of the user required speed Y_{user}. And let $\dot{y}_f$ is equal to αY_{user}. Then, according to Equation 7, the four coefficients $a_{0..3}$ in the cubic polynomial are determined using the following equations:

$$PY_i^* = \sum_{i=0}^{3} a_i (iT)^i$$

$$PY_{i+1}^* = \sum_{i=0}^{3} a_i (i(T+1))^i$$

$$\begin{bmatrix} y_f \\ \dot{y}_f \end{bmatrix} = A(T) \begin{bmatrix} y_i \\ \dot{y}_i \end{bmatrix} + [I - A(T)] \begin{bmatrix} a_0 + 2a_2/q^2 \\ a_1 + 6a_3/q^2 \end{bmatrix} + \begin{bmatrix} \sum_{i=0}^{3} a_i t^i + 6a_3 t/q^2 - a_0 \\ 3a_3 t^2 + 2a_2 t \end{bmatrix}$$

In the case of lateral direction, the initial state of CoG and the initial and goal ZMP are already known like the case of sagittal direction. The goal CoG velocity is known too, which is equal to zero, see Figure 2. We now explain how the goal CoG position can be determined using *Simplified Walking*. Note that the goal CoG state will become the initial CoG state of the next step. And we know that the initial CoG velocity ($\dot{x_i}$) of the next step is zero and the CoG position (x_f) of the next step after half step is determined by offset of Hip Joint. The two unknown variables are the initial CoG position (x_i) of the next step and the CoG velocity ($\dot{x_f}$) of the next step at the time of half step duration, and they are determined using Equation 5. Thus, the goal CoG position is known. So, like the case of sagittal direction, the coefficients in the cubic polynomial are determined.

By applying the cubic ZMP trajectory, a CoG state (both in X and Y axis) can be achieved in one step as long as the computed ZMP trajectory are in the support polygons. The preview control based walking needs around 1.5s

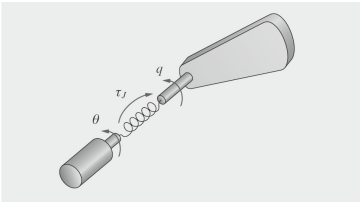

Fig. 4. Schematic Representation of An Elastic Joint

to converge. Our previous work *Simplified Walking* needs two steps. Thus, this technique has a great advantage for soccer players that change speed or direction rapidly.

3 Elasticity Modeling Control

The compensation to the HipRoll Joint has beed discussed in the previous works. However, a lack of theory or model leads the compensation to an empirical way. Here we model the joint as elastic joint, see Figure 4. Under the assumption proposed in [13] by Spong, we have the following dynamic equations:

$$M(q)\ddot{q} + S(q,\dot{q})\dot{q} + g(q) + K(q-\theta) = 0 \tag{8}$$

$$I\ddot{\theta} + K(\theta - q) = \tau \tag{9}$$

Where q is the (n × 1) vector denoting the link positions. θ is the (n × 1) vector denoting the motor positions. Note that q_i is not equal to θ_i because of elasticity. M(q) is the (n × n) robot link inertia matrix. $S(q,\dot{q})\dot{q}$ is the (n × 1) vector of centrifugal and Coriolis torques. K is the (n × n) diagonal matrix of joint stiffness coefficients. g(q) is the (n × 1) vector of gravitational torque. I is an (n × n) constant diagonal matrix including rotor inertia and the gear ratios. τ is the (n × 1) vector of the torque acting on the elastic joints.

Vector q is given by the Inverse Kinematics. Vector θ is the final joint commands that will be sent to the robot. However, the computation power of Nao does not allow for the above heavy computation. Thus, we again take the 3D-LIP as a simplified model.

Let ${}^{i-1}T_i$ denote the transformation matrix from link frame i to link frame i-1, for the Nao robot, ${}^1T_2 = Rot_y(q_2)$, ${}^2T_3 = Rot_x(q_3)$, ${}^3T_4 = Trans_z(-R)Rot_x(q_4)$, ${}^4T_5 = Trans_z(-S)Rot_x(q_5)$, ${}^5T_6 = Rot_y(q_6)$. Note that, the coordinate is rotated - 90 degrees compared to the coordinate used in the document provided by Aldebaran [4]. The transformation matrix from CoG to the first link of left leg is: ${}^GT_1 = Trans_x(-H)Trans_z(-N)$, where H is the Hip offset along X axis and N is the Hip offset along Z axis. The first joint is not considered, so ${}^1T_2$ is an identity matrix. Following the standard Denavit-Hartenberg method, we have:

$${}^GT_6 = {}^GT_1 \prod_2^6 {}^{i-1}T_i \tag{10}$$

Thus, ${}^6T_G$ is obtained by computing the inverse matrix. The 3rd column of this matrix is the CoG position relative to the foot coordinate, denoted as $P(q_2, q_3, q_4, q_5, q_6)$, where q_i is the i-th link position. So, the Lagrange function is:

$$L = \frac{1}{2}M\dot{P}^T\dot{P} - MgP_z - \frac{1}{2}(q-\theta)^T K(q-\theta) \tag{11}$$

Following the Lagrange approach, we have:

$$\frac{d}{dt}\frac{\partial\,(\Gamma - V_{grav})}{\partial\,\dot{q}_i} - \frac{\partial\,(\Gamma - V_{grav})}{\partial\,q_i} + k_i(q_i - \theta_i) = 0 \tag{12}$$

Compared to Equation 8 and 9, Equation 12 does not contain the term of centrifugal and Coriolis torques. Thus, the computation for elasticity is simplified quite a lot. By using the symbol differential tool developed by us, the final equation for one joint is formed by, approximately, one thousands of terms (in average 10 multiplications per term). That's nearly one third of the computation required by Equation 8 and 9 induced from full dynamic model. However, this is still relatively heavy for Nao.

In the Robocup domain, the robot is walking on a planar ground, and the foot is always horizontal to the ground. Thus, we have:

$$\begin{aligned} q_2 + q_6 &= 0 \\ q_3 + q_4 + q_5 &= 0 \end{aligned} \tag{13}$$

By, substituting q_6 with q_2 and q_4 with $-q_3 - q_5$ in Equation 12, we have 167 terms for computing θ_3, 471 terms for computing θ_4, 517 terms for computing θ_5. The equations for computing θ_2 and θ_6 are simpler and can be written down:

$$\begin{aligned} k_2\theta_2 = {} & \ddot{L_z}MN\sin(q_2) + 2\dot{L}zMN\dot{q_2}\cos(q_2) - L_zMN\dot{q_2}^2\sin(q_2) + \\ & L_zMN\ddot{q2}\cos(q_2) + H^2M\dot{q_2}^2\cos(q_2)\sin^3(q_2) + HMN\dot{q_2}^2 - \\ & 2HMN\dot{q_2}^2\sin^2(q_2) - HMN\dot{q_2}^2\cos^4(q_2) + HMN\dot{q_2}^2\sin^4(q_2) + \\ & HMg - MN^2\dot{q_2}^2\cos(q_2)\sin^3(q_2) + k_2q_2 \end{aligned}$$

$$\begin{aligned} k_6\theta_6 = {} & -\ddot{L_z}L_zM\sin(q_2)\cos(q_2) - L_zMg\sin(q_2) + \ddot{L_z}MN\sin(q_2) + \\ & 2\dot{L_z}MN\dot{q_2}\cos(q_2) + L_z^2M\dot{q_2}^2\cos(q_2)\sin(q_2) - L_z^2M\ddot{q_2}\cos^2(q_2) + \\ & L_zMN\ddot{q_2}\cos(q_2) - L_zMN\dot{q_2}^2\sin(q_2) - 2H^2M\dot{q_2}^2\sin(q_2)\cos(q_2) + \\ & H^2M\dot{q_2}^2\sin^3(q_2)\cos(q_2) + 2H^2M\dot{q_2}^2\cos^3(q_2)\sin(q_2) - \\ & 2\dot{L_z}L_zM\dot{q_2}\cos^2(q_2) + HMg - k_6q_2 \end{aligned} \tag{14}$$

where $\dot{L_z}$, $\ddot{L_z}$, $\dot{q_2}$ and $\ddot{q_2}$ are determined by:

$$\begin{bmatrix} \dot{X_g} \\ \dot{Z_g} \end{bmatrix} = \begin{bmatrix} -\sin(q_2) & -L_z\cos(q_2) \\ -\cos(q_2) & L_z\sin(q_2) \end{bmatrix} \begin{bmatrix} \dot{L_z} \\ \dot{q_2} \end{bmatrix} \tag{15}$$

$$\begin{bmatrix} \ddot{X_g} \\ \ddot{Z_g} \end{bmatrix} = \begin{bmatrix} -\sin(q_2) & -L_z\cos(q_2) \\ -\cos(q_2) & L_z\sin(q_2) \end{bmatrix} \begin{bmatrix} \ddot{L_z} \\ \ddot{q_2} \end{bmatrix} + \begin{bmatrix} -2\dot{L_z}\dot{q_2}\cos q_2 & L_z\dot{q_2}^2\sin q_2 \\ 2\dot{L_z}\dot{q_2}\sin q_2 & L_z\dot{q_2}^2\cos q_2 \end{bmatrix} \tag{16}$$

Thus, the robot is able to do the compensation online. We now explain how to determine the stiffness coefficients. When the robot is standing with one of its leg, we have $\dot{X_g} = \ddot{X_g} = 0$, Equation 14 are:

$$\theta_2 = HMg/k_2 + q_2 \tag{17}$$

$$\theta_6 = -L_zMg\sin(q_2)/k_6 + HMg/k_6 - q_2 \tag{18}$$

The deflection of HipRoll $(\theta_2 - q_2)$ is constant, which is equal to HMg/k_2. So we conducted an experiment in which the robot is standing with one of its leg, and k_2 is adjusted online according to the readings from inertial sensor. The rest joint stiffness coefficients are simply set to the same value.

4 Experiments

All the experiments are conducted on a real robot named Nao with the version of 3.3.

4.1 Simultaneously Planning CoG and ZMP Trajectory

The width of the foot is around 8cm and the length of the foot is around 12cm (actually longer). ZMP trajectory should be kept in the support polygon decided by the foot dimension.

We first let the robot walk left with step duration equaling 480ms (100ms for double support phase), step length equaling 4cm. The left part of Figure 5 shows the results, where the black line is the ZMP trajectory and the green line is the CoG trajectory. The red and blue lines show the allowed support polygon. The shadow indicates the double support phase. We can see that with a double support of 17%, the ZMP trajectory is kept in the support polygon, thus making a dynamic balanced gait.

Next, we let the robot walk forward with step duration equaling 400ms (100ms for double support phase), maximum step length equaling 8cm and the portion of the user required speed equaling 50%. At first, the robot is walking with step length 0.8cm, that is 10% of the maximum speed 0.2m/s. Then it switches to full speed 0.2m/s at 10.5s. The right paft of Figure 5 shows the result. The CoG state is achieved in one step without breaking the dynamic balance law. Note that at the time of 10.5s, the ZMP trajectory first moves ahead of the CoG to decelerate and then moves back in a short time to accelerate. This effect is due to the cubic polynomial.

4.2 Elasticity Control

The compensation to the elastic joints can obey Equation 14. However, in our dynamic balance control module, the AnkleRoll joint will be modified, indicating

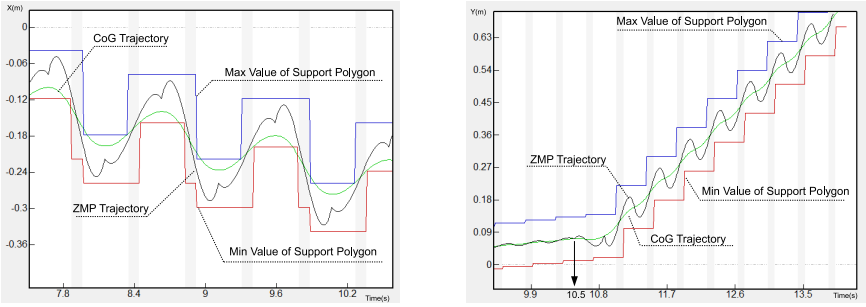

Fig. 5. Simultaneously Planning of ZMP/CoG Trajectory. The Left One Is along Lateral Axis and the Right One Is along Sagittal Axis.

it does not need a feed forward compensation. Besides, the compensations to the HipPitch, KneePitch and AnklePitch are relatively small and computation for these three joints are relatively heavy, so they are not compensated too.

However, the compensations to the two HipRoll joints can not just obey Equation 14, because in the double support phase it is a closed-loop chain, not an open-loop chain as in the single support phase. The key part of this compensation is to guarantee that the landing foot will not hit the ground hard, or else the robot will be in a unbalanced state. So, for the stance leg, the compensation only obeys Equation 14 at the second single support phase of a step. The rest of it is compensated using interpolation to keep the curve continuous, as shown in Figure 6. The step duration is 400ms (100ms for double support phase), and the stiffness coefficient for each HipRoll joint is 41.

Figure 7 shows the trunk angle around sagittal direction. Before the time of 2335s, elasticity is not enabled and the total amplitude is around 0.15 rad. After elasticity control is enabled at the time 2335s, the total amplitude is deduced to 1 rad. When walking in high frequency, if the elasticity is not enabled, the robot will not be able to walk. Our elasticity control algorithm by modeling the joints as elastic ones is effective.

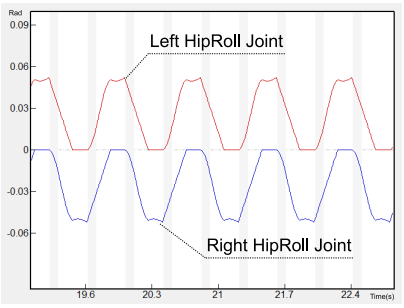

Fig. 6. Trajectories of HipRoll Joints1

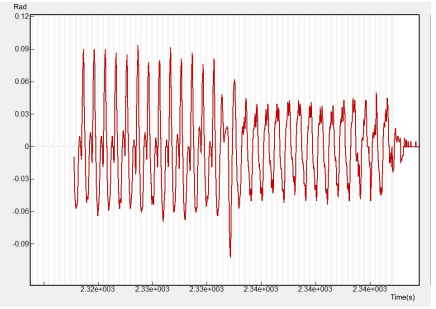

Fig. 7. Trunk Angle Around Sagittal Axis

4.3 Fast Walking on Nao

Together with the technique in terms of dynamic balance control proposed in [16], we achieved a fast walk with step duration 180ms (20ms for double support phase). The parameter that specifies the percentage of the user required speed is 0.8. The maximum forward speed is around 0.33m/s. The maximum backward speed is around 0.2m/s. The maximum sideways speed is around 0.11m/s. The maximum rotational speed is around 90°/s. The video is at `http://ai.ustc.edu.cn/en/demo/index.php`.

Team HTWK achieved a forward speed of 0.32m/s based on machine learning approach and its sideways motion is not good. Team Dortmund achieved a very fast walk with forward speed of 0.44m/s. However it is not stable. Team

B-Human achieved a walk with maximum forward speed 0.28m/s, maximum backward speed 0.17m/s, maximum sideways speed 0.07m/s and maximum rotational speed 90°/s.

5 Conclusion

we presented a new technique to achieve a real time biped walking gait pattern generator. By simultaneously planning the ZMP trajectory represented by a cubic polynomial and the CoG trajectory, only one step is needed to connect the goal state (the position and velocity of the CoG) to the previous one.

To apply the generator on Nao used in Standard Platform League (SPL), we calculate the compensation for two HipRoll joints in a theoretical way by modeling the leg joints as elastic ones. Note that, the elasticity control is not just for walk, but also for other motions such as kick when the robot is standing on one foot.

Demonstrations are performed on Nao of version 3.3, showing that the proposed method is effective on a soccer player robot.

Acknowledgements. This work is supported by the National Hi-Tech Project of China under grant 2008AA01Z150, the Natural Science Foundations of China under grant 60745002 and the USTC 985 Project.

References

1. Behnke, S.: Online trajectory generation for omnidirectional biped walking. In: IEEE International Conference on Robotics and Automation (2006)
2. Chestnutt, J., Kuffner, J., Nishiwaki, K., Kagami, S.: Planning biped navigation strategies in complex environments. In: IEEE/RSJ Int. Conf. on Humanoid Robotics, Humanoids 2003, Karlsruhe & Munich, Germany (2003)
3. Czarnetzki, S., Kerner, S., Urbann, O.: Observer-based dynamic walking control for biped robots. Robotics and Autonomous Systems 57(8) (2009)
4. Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., Maisonnier, B.: Mechatronic design of NAO humanoid. In: IEEE International Conference on Robotics and Automation (2009)
5. Graf, C., Röfer, T.: A closed-loop 3D-LIPM gait for the RoboCup Standard Platform League Humanoid. In: Zhou, C., Pagello, E., Behnke, S., Menegatti, E., Röfer, T., Stone, P. (eds.) Proceedings of the Fourth Workshop on Humanoid Soccer Robots in Conjunction with the 2010 IEEE-RAS International Conference on Humanoid Robots (2010)
6. Graf, C., Härtl, E., Röfer, T., Laue, T.: A robust closed-loop gait for the Standard Platform League Humanoid. In: Proceedings of the 4th Workshop on Humanoid Soccer Robots. A Workshop of the 2009 IEEE-RAS Intl. Conf. on Humanoid Robots (2009)
7. Harada, K., Kajita, S., Kaneko, K., Hirukawa, H.: An analytical method on real-time gait planning for a humanoid robot. Journal of Humanoid Robotics 3(1) (2006)

8. Kajita, S., Kanehiro, F., Kanako, K., Yokoi, K., Hirukawa, H.: The 3D Linear Inverted Pendulum Model: A simple modeling for a biped walking pattern generation. In: IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS 2001, Hawaii, USA (2001)
9. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generation by using preview control of Zero-Moment Point. In: IEEE Int. Conf. on Robotics and Automation, ICRA 2003 (2003)
10. Liu, J., Manuela, V.: Online ZMP sampling search for biped walking planning. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS (2008)
11. Liu, J., Xue, F., Chen, X.: A universal biped walking generator for rough terrain with pattern feasibility checking. International Journal of Humanoid Robotics (to be appeared in 2011)
12. Luca, A.D., Siciliano, B., Zollo, L.: PD control with on-line gravity compensation for robots with elastic joints: theory and experiments. Automatica 41(10) (2005)
13. Spong, M.W.: Modeling and control of elastic joint robots. ASME Journal of Dynamic Systems, Measurement, and Control 109 (1987)
14. Strom, J., Slavov, G., Chown, E.: Omnidirectional Walking Using ZMP and Preview Control for the NAO Humanoid Robot. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009. LNCS (LNAI), vol. 5949, pp. 378–389. Springer, Heidelberg (2010)
15. Vukobratovic, M., Borovac, B.: Zero-Moment Point - thirty five years of its life. International Journal of Humanoid Robotics 1(1), 157–173 (2004)
16. Xue, F., Chen, X., Liu, J.: Integrated balance control on uneven terrain based on inertial sensors. Submitted to IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS 2011 (2011)
17. Zollo, L., Siciliano, B., Luca, A.D., Guglielmelli, E., Dari, P.: Compliance control for an anthropomorphic robot with elastic joints: theory and experiment. ASME Transactions: Journal of Dynamic Systems, Measurements, and Control 127(3) (2005)