

# Online ZMP Sampling Search for Biped Walking Planning

Jinsu Liu<sup>1,2</sup> and Manuela Veloso<sup>2</sup>

<sup>1</sup> Computer Science Department, University of Science and Technology of China

jslau@mail.ustc.edu.cn

<sup>2</sup> Computer Science Department, Carnegie Mellon University

{jinsuliu,veloso}@cmu.edu

**Abstract**—In this paper, we present a new method that uses random search for online planning of biped walking, given a feasible footstep plan. The Linear Inverted Pendulum dynamic model and the Zero Moment Point concept are employed to solve the walking problem. We consider walk planning as the choice of a sequence of ZMPs leading to a stable walk that satisfies all the dynamic and mechanical constraints of the robot. We contribute a novel online sampling algorithm to efficiently search for such ZMP sequence. We demonstrate the effectiveness of the algorithm by successful combined walking tasks in a faithful simulation of a full-body humanoid robot.

**Index Terms**—Humanoid robot, walking planning, sampling search, online planning, linear inverted pendulum (LIP), zero moment point (ZMP)

## I. INTRODUCTION

Biped motion planning is a challenging problem in humanoid robotics research, because of nonlinear dynamics and the high dimensionality of the state space. Sampling-Based Planning[1] was introduced to solve footstep planning and has enjoyed much success in robot motion planning. However, it is still difficult to apply random search to online, low-level planning because of solution efficiency.

This paper presents a new method, which we name ZMP Sampling Search, as a universal solution for the planning of biped motions with constant torso height. Our approach employs a random search algorithm to choose the ZMP sequence for our simplified walking. Our algorithm is able to achieve online planning for any given feasible footstep plan.

The rest of the paper is organized as follows: Section II gives an overview of related work. Section III introduces a simplified model of biped walking and the dynamic models which are used for our walk planning. Section IV describes the ZMP sampling search algorithm that solves the walking problem. Section V presents the performance of our algorithm and experimental results. We conclude in Section VI.

## II. RELATED WORK

In recent biped motion research, adaptability, online, universality and intelligence are addressed. RRT and other search techniques have been introduced to solve global footstep planning problems [7][8][9]; this demonstrates the possibility of using online search in biped motion planning to improve walking flexibility.

However, in lower level planning, such as walking pattern generation, the dynamics of the robot need to be taken into account. Offline planning is widely applied to these problems[6]. The Linear Inverted Pendulum (LIP) model [2][3] approximates the dynamic process of biped motion; Zero Moment Point (ZMP) [4] is a powerful constraint that ensures a stable walk. In practical applications, different kinds of motions are predefined by different trajectory planners; the robot is able to achieve the motions and keep its balance using the feedback from its sensors. Although genetic algorithms [11] and Monte Carlo methods [12] have already been used for offline optimization, it is difficult to employ an adaptive solver based on random search in the planning, because biped walking is a complex nonlinear dynamic process.

RRT search has been successfully applied in footstep planning and other motions of humanoid robots [10]. RRT employs a random method to find a solution effectively; it also reduces the dimension of the planning problem by using a universal "AutoBalancer" [13] as a dynamic filter. In this work, we employ an online random search algorithm to find a motion plan that satisfies the mechanical and dynamic constraints of biped walking.

## III. DYNAMIC MODELS

### A. 3D-LIP and ZMP

We use the Linear Inverted Pendulum (LIP) to describe the approximate movement of a biped walking when the robot is supporting its body on one leg. A 3D linear inverted pendulum is an inverted pendulum which moves in a specific plane. If the constraint plane is a horizontal plane, the physics model shown in Figure 1. can be described as follows [5]:

$$\ddot{x} = \frac{g}{h}x + \frac{1}{mh}\tau_y \quad (1)$$

$$\ddot{y} = \frac{g}{h}y - \frac{1}{mh}\tau_x \quad (2)$$

where  $g$  is the acceleration due to gravity,  $h$  is the given height,  $m$  is the mass of the pendulum,  $(x, y)$ ,  $(\ddot{x}, \ddot{y})$  is the position and the acceleration of the pendulum in the constraint plane, and  $\tau_x, \tau_y$  are the torques around the  $x$ -axis and  $y$ -axis, respectively.

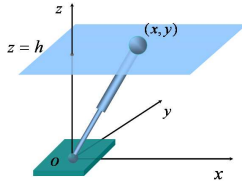


Fig. 1. 3D Linear Inverted Pendulum

The Zero Moment Point (ZMP) is another important concept in biped walking. The ZMP is defined as a point on the ground where the sum of all the moments (due to gravity and inertial forces) equals zero [6]. When the ZMP is within the convex hull of the contact points between the feet and the ground, the robot can walk in a stable manner; the feet of the robot will fully contact the ground. For bipedal robots with touch sensors attached under their feet, it is especially important to adjust the balance using the feedback provided by these sensors.

In a 3D LIP, as shown in Figure 1, the ZMP can be calculated by

$$p_x = -\frac{1}{mg}\tau_y \quad (3)$$

$$p_y = \frac{1}{mg}\tau_x \quad (4)$$

where  $(p_x, p_y)$  is the ZMP. We can rewrite (1) and (2) as

$$\ddot{x} = \frac{g}{h}(x - p_x) \quad (5)$$

$$\ddot{y} = \frac{g}{h}(y - p_y) \quad (6)$$

Solving (5) as a differential equation, we get the relationship between a final state and an initial state in x-axis of a LIP including the position and the velocity:

$$\begin{bmatrix} x_f(t) \\ v_f(t) \end{bmatrix} = A(t) \begin{bmatrix} x_i \\ v_i \end{bmatrix} + [I - A(t)] \begin{bmatrix} p_x \\ 0 \end{bmatrix} \quad (7)$$

where  $(x_i, v_i)$  is the initial state,  $(x_f(t), v_f(t))$  is the final state at time  $t$ ,  $I$  is a  $2 \times 2$  identity matrix and  $A(t)$  is a state transition matrix which only depends on the duration  $t$ :

$$A(t) = \begin{bmatrix} \cosh(qt) & \frac{1}{q}\sinh(qt) \\ q\sinh(qt) & \cosh(qt) \end{bmatrix} \quad (8)$$

$q$  is the constant:

$$q = \sqrt{\frac{g}{h}} \quad (9)$$

By defining the state of 3D LIP at time  $t$  as:

$$X(t) = \begin{bmatrix} x(t) & y(t) \\ \dot{x}(t) & \dot{y}(t) \end{bmatrix} \quad (10)$$

and the control input as:

$$U = \begin{bmatrix} p_x & p_y \\ 0 & 0 \end{bmatrix} \quad (11)$$

we obtain:

$$X(t) = A(t)X(0) + [I - A(t)]U \quad (12)$$

## B. Biped Robot

The humanoid robot Nao (Figure 2) is used for our research and simulation experiments. Nao is a new type of full-body humanoid robot produced by Aldebaran Robotics [14]. The height of the robot is about 60 cm and the weight is about 4.5 kg. The robot has 21 degrees of freedom (DOF). There are two DOFs in the head, four DOFs in each arm, five DOFs in each leg, and one in the pelvis (shared between the legs). This robot model is able to provide enough flexibility for our research.

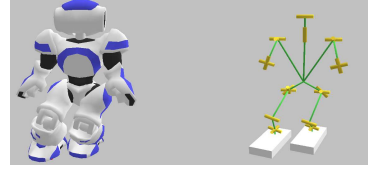


Fig. 2. Model of the humanoid robot Nao and its joints.

## C. Simplified Walking

Biped walking is a complex nonlinear dynamic process. It can be considered as a periodic phenomenon. A whole walking cycle consists of two phases: the *double-support phase* and the *single-support phase*. Although the double-support phase is only about 20% of the duration of a walk cycle [6], it is important for making the walk more stable and reducing the impact between the swing foot and the ground. However, the single-support phase is the key part of biped walking. During the single-support phase, the robot must place its swing foot in position for the next step while keeping its balance.

In this paper, we simplify biped walking by considering it as a series of single-support phases for the sampling search algorithm, and focus on choosing the ZMPs in these phases. The double-support phase can be planned according to the ZMP decisions in a lower-level planning to achieve a complete walking. Our simplified model of biped walking is defined as follows:

1) The robot is considered as a linear inverted pendulum which is supported by one of its feet. The mass of the robot is distributed on the Center of Mass (CoM) which moves in a horizontal plane with a constant height. The effects of inertia are ignored.

2) The walking process only consists of a series of single-support phases. The robot uses each of its legs as the supporting leg in turn. Theoretically, one single-support phase is able to switch to the next one immediately.

3) While in the single-support phase, we only use a single ZMP decision instead of a ZMP trajectory.

Figure 3 shows our biped walking planner approach based on ZMP sampling search. The inputs of the planner consist of a footstep plan, the initial state, and the final state of the CoM. A footstep plan, including the pose and the duration of each footstep, comes from higher-level planning, such as an RRT path search [7], A-star search [8], a static footstep library, or as created by hand by a human. The output of this

walking planner is a sequence of ZMP decisions. A detailed description of the algorithm will be given in the next section.

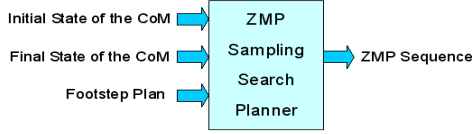


Fig. 3. Our biped walking planner approach.

The ZMP sampling search planner is supported by lower-level algorithms which provide CoM trajectory generation, trajectory modification for the double-support phase, and calculation of the desired joint angles by using inverse kinematics. Due to the limitations on paper length, the details of these lower-level functions are not described in this paper.

#### IV. ZMP SAMPLING SEARCH ALGORITHM

We introduce two ZMP sampling search algorithms to find the ZMP sequence as the solution of a biped walk. One is the unidirectional search which works when the robot does not have a specific final state, the other is the bidirectional search which can achieve a given final state.

##### A. Definitions

The following definitions are used in the algorithm description:

1) *Foot Pose*  $F_k$ : A tuple with a 2D position on the ground and a direction to describe the gesture of the supporting foot in the global reference frame. In this paper, we assume that the shape of the underside of a foot is a rectangle.

2) *State*  $s$ : The position and velocity of the CoM in the constraint plane in the global reference frame.

3) *Kinematic Range*  $R_k$ : A convex polygon which can be any shape inside the reachable area of the CoM as large as possible.

The CoM of the robot has a 2D reachable area in the constraint plane because of the leg shapes and the joint angle limits of the robot. Generally, the reachable area has an irregular shape. For the humanoid robot Nao, the shape only depends on the poses of the robot's feet because of the decrease of DOF by the shared joint in pelvis. For other bipedal robots with 6 DOFs in each leg, the orientation of the torso should be specified or related with the poses of feet for us to obtain the reachable area.

For the purpose of algorithm efficiency, rectangles are used to represent the kinematic range, as shown in Figure 4. We build a lookup table to store the kinematic ranges offline; this table can be used online for our search.

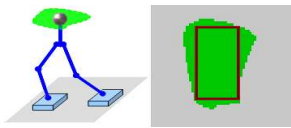


Fig. 4. Reachable area and kinematic range.

4) *K-Steps Sample*  $\pi^k$ : A tuple  $\pi^k = (p_1, p_2, \dots, p_k, s)$ . It includes  $k$  2D points and a final state. The  $k$ th point is a ZMP decision to control the walking in the  $k$ th step. The final state is the ending state of the CoM of the  $k$ th step. A zero-step sample has only a final state.

5) *K-Steps Sample Set*  $S^k$ : A set of a certain number of  $k$ -steps samples. It has the following form:  $S^k = \{\pi_1^k, \pi_2^k, \dots, \pi_n^k\}$ .

##### B. Algorithm of Unidirectional Search

Given a footstep plan and an initial state of the CoM, we use a unidirectional search algorithm to search for a sequence of ZMP points that make the robot follow the footstep plan from the initial state. A valid solution satisfies two kinds of constraints: 1) *Dynamic Constraints*: the  $k$ th ZMP decision should be inside the  $k$ th foot pose. 2) *Mechanical Constraints*: the position of the ending state of the  $k$ th step should be inside the  $k$ th kinematic range. Table I shows the pseudocode for our algorithm. We explain each procedure in detail.

TABLE I

PSEUDOCODE FOR OUR UNIDIRECTIONAL SEARCH ALGORITHM.

SEARCH ( $F_{1..k}, t_{1..k}, R_{1..k}, s_{init}$ )	
1	$S^0 \leftarrow \{(s_{init})\}$
2	<b>for</b> $i = 1$ <b>to</b> $k$ <b>do</b>
3	$S^i \leftarrow \text{GENERATE}(S^{i-1}, F_i, t_i, R_i)$
4	<b>if</b> ( $S^i = \emptyset$ ) <b>then</b> Return <i>Failure</i>
5	$S^i \leftarrow S^i \cup \text{CONVEX\_EXPAND}(S^i)$
6	Return $S^k$
GENERATE ( $S^{k-1}, F_k, t_k, R_k$ )	
1	$S^k \leftarrow \emptyset$
2	<b>for</b> $i = 1$ <b>to</b> $M$ <b>do</b>
3	$\pi^{k-1} \leftarrow \text{RANDOM\_ELEMENT}(S^{k-1})$ // $\pi^{k-1} = (p_1, p_2, \dots, p_{k-1}, s)$
4	$p_k \leftarrow \text{RANDOM\_POSITION}(F_k)$
5	$s_{new} \leftarrow \text{LIP}(s, p_k, t_k)$
6	<b>if</b> ( $\text{INSIDE}(s_{new}, R_k)$ ) <b>then</b>
7	$S^k \leftarrow S^k \cup \{(p_1, p_2, \dots, p_{k-1}, p_k, s_{new})\}$
8	<b>if</b> ( $ S^k  = \text{Size}_{gen}$ ) <b>then</b> Return $S^k$
9	Return $S^k$
CONVEX_EXPAND ( $S^k$ )	
1	$S_{new}^k \leftarrow \emptyset$
2	<b>for</b> $i = 1$ <b>to</b> $\text{Size}_{exp}$ <b>do</b>
3	$\pi_1^k \leftarrow \text{RANDOM\_ELEMENT}(S^k)$
4	$\pi_2^k \leftarrow \text{RANDOM\_ELEMENT}(S^k)$
5	$\alpha \leftarrow \text{RANDOM\_VALUE}(0, 1)$
6	$S_{new}^k \leftarrow S_{new}^k \cup \text{LINEAR\_COMBINE}(\pi_1^k, \pi_2^k, \alpha)$
7	Return $S_{new}^k$

1) *Main Procedure of Unidirectional Search*: The main loop of the unidirectional search algorithm is an iterative process. The inputs are the Foot Poses  $F_{1..k}$  of  $k$  single-support phases, Supporting Durations  $t_{1..k}$ , Kinematics Ranges  $R_{1..k}$  and the initial state of the CoM  $s_{init}$ . The procedure generates an  $i$ -steps sample set with an  $(i-1)$ -steps sample set in the  $i$ th loop. The loop terminates whenever there is a failure to generate any sample set. The

CONVEX\_EXPAND function is used to ensure that each sample set is of a certain size.

Assume a  $k$ -steps sample set is  $S^k = \{\pi_1^k, \pi_2^k, \dots, \pi_n^k\} = \{(p_{1,1}, p_{1,2}, \dots, p_{1,k}, s_1), (p_{2,1}, p_{2,2}, \dots, p_{2,k}, s_2), \dots, (p_{n,1}, p_{n,2}, \dots, p_{n,k}, s_2)\}$ . Then, we obtain a ZMP solution from the  $k$ -steps sample set with the function

$$\Gamma(S^k) = \left( \frac{1}{n} \sum_{i=1}^n p_{i,1}, \frac{1}{n} \sum_{i=1}^n p_{i,2}, \dots, \frac{1}{n} \sum_{i=1}^n p_{i,k} \right) \quad (13)$$

The solution is the average center of the feasible solutions, given the sample set. Due to noise and other disturbances, the average solution is considered much safer than the border solutions. The solution satisfies all the constraints. It can be proved by mathematical induction, similar as the proof in the Convex Expansion section below.

2) *Generation*: The GENERATE function is the core of our sampling-based search process. GENERATE returns a set of samples of possible ZMP values by sampling the space of the robot foot and validating the samples according to the LIP model.

In each attempt to generate a  $k$ -steps sample, the algorithm selects an element from the  $(k-1)$ -steps sample set randomly, and uses a random ZMP decision inside the supporting foot to obtain a new state of the CoM. The new  $k$ -steps sample is created and added to the  $k$ -steps sample set when the new state satisfies the kinematic range constraint. The size of the sample set will not exceed the *generated size*  $Size_{gen}$ .

The function  $LIP(s, p_k, t_k)$  calculates the final state, given an initial state  $s$ , a ZMP decision  $p_k$  and a duration time  $t_k$  according to (12).

Our GENERATE function is an incomplete search process. The loop terminates in  $M$  iterations whenever it gets  $Size_{gen}$  samples, where  $M$  is a given constant. Thus, when the generation function returns an empty sample set, we cannot determine whether a sample exists. However, we do know that it is very difficult to find a sample, so the solution (if it exists) must be in a tiny area of the solution space. In a complex nonlinear system full of noise, it is unwise to use a solution that demands high-precision robot control; in this case, the higher-level planner should notice the problem and adjust its footstep plan accordingly.

3) *Convex Expansion*: Convex expansion is an important technique to reduce the computation time of ZMP sampling search. The convex expansion function creates a  $k$ -steps sample by repeatedly combining two randomly selected  $k$ -steps samples using a linear combination. Assuming two  $k$ -steps samples are selected:  $\pi_1^k = (p_{1,1}, p_{1,2}, \dots, p_{1,k}, s_1)$   $\pi_2^k = (p_{2,1}, p_{2,2}, \dots, p_{2,k}, s_2)$ , a linear-combined  $k$ -steps sample can be calculated as linear combinations by a random value  $\alpha$  which is between 0 and 1.

$$\begin{aligned} \pi_{com}^k = & (\alpha p_{1,1} + (1-\alpha)p_{2,1}, \\ & \alpha p_{1,2} + (1-\alpha)p_{2,2}, \\ & \dots, \\ & \alpha p_{1,k} + (1-\alpha)p_{2,k}, \\ & \alpha s_1 + (1-\alpha)s_2) \end{aligned} \quad (14)$$

*Claim*: A linear-combined sample satisfies all the dynamic and mechanical constraints.

*Proof*: Let  $\mathcal{F}_k$  denote the area on the ground occupied by the  $k$ th supporting foot with the pose of  $F_k$ .  $\forall i, p_{1,i} \in \mathcal{F}_i, p_{2,i} \in \mathcal{F}_i$ ; the shape of  $\mathcal{F}_i$  is a rectangle, which is a convex polygon. Therefore, when it is given a value  $\alpha \in (0, 1)$ , the point  $p_{com,i} = \alpha p_{1,i} + (1-\alpha)p_{2,i}$  satisfies  $p_{com,i} \in \mathcal{F}_k$ .

We define the initial state as  $X_0$ , the state of the  $k$ th step as  $X_k$ , the  $k$ th state transition matrix as  $A_k$ , the input control (which depends on the  $k$ th ZMP decision) as  $U_k$  and the area expressed by the kinematic range  $R_k$  as  $\mathcal{R}_k$ . Let  $e$  denote a vector  $[1, 0]^T$  so that the position of a state can be write as  $X^T e$ .

For the initial state,  $X_{1,0} = X_{2,0} = X_0$ . We obtain  $X_{com,0} = \alpha X_{1,0} + (1-\alpha)X_{2,0} = X_0$ .

Assume that we are able to get the  $k$ th state  $X_{com,k}$  by using the former  $k$  combined ZMP decisions; also assume that it satisfies that  $X_{com,k}^T e \in \mathcal{R}_k$ . According to (12), we get:

$$X_{1,k+1} = A_{k+1}X_{1,k} + (I - A_{k+1})U_{1,k+1}$$

$$X_{2,k+1} = A_{k+1}X_{2,k} + (I - A_{k+1})U_{2,k+1}$$

$$\begin{aligned} \text{Then, } X_{com,k+1} &= \alpha X_{1,k+1} + (1-\alpha)X_{2,k+1} = \\ & \alpha[A_{k+1}X_{1,k} + (I - A_{k+1})U_{1,k+1}] + (1-\alpha)[A_{k+1}X_{2,k} + \\ & (I - A_{k+1})U_{2,k+1}] = A_{k+1}[\alpha X_{1,k} + (1-\alpha)X_{2,k}] + \\ & (I - A_{k+1})[\alpha U_{1,k} + (1-\alpha)U_{2,k}] = A_{k+1}X_{com,k} + (I - \\ & A_{k+1})U_{com,k+1} \end{aligned}$$

Therefore,  $X_{com,k+1}$  can be achieved by applying the  $k$ th combined ZMP decision to the  $k$ th combined state, and it satisfies that  $X_{com,k+1}^T e \in \mathcal{R}_{k+1}$ , because the shape of  $\mathcal{R}_k$  is a rectangle which is also a convex polygon.

Then, we can conclude that a linear-combined sample satisfies all the mechanical and dynamics constraints.  $\square$

The number of expanded samples is the *expanded size*  $Size_{exp}$ . The performance of the algorithm is directly affected by the generated size and the expanded size. This analysis is shown below, in Section V.

### C. Algorithm of Bidirectional Search

In regular biped walking, it is also important for the robot to arrive at a specific state—for example, to stop moving. The bidirectional search algorithm (shown in Table II) can be used to find the ZMP solution, when the walking planner is given an initial state, a final state, and a footstep plan.

1) *Main Procedure of Bidirectional Search*: Our bidirectional search algorithm can be considered as a combination of two unidirectional searches. One direction is from the first step to the last step and the other is in the opposite direction. The two unidirectional searches are connected by two steps in the middle of the footstep plan. We name the returned sample sets as *head sample set* and *tail sample set*, respectively.

The function  $REV(s)$ , which appears in the algorithm, returns a *reverse state*, defined as follows:

$$REV\left(\begin{bmatrix} x(t) & y(t) \\ \dot{x}(t) & \dot{y}(t) \end{bmatrix}\right) = \begin{bmatrix} x(t) & y(t) \\ -\dot{x}(t) & -\dot{y}(t) \end{bmatrix} \quad (15)$$

A reverse state is a virtual state which is used for connecting the searches in the two different directions.

TABLE II  
PSEUDOCODE FOR OUR BIDIRECTIONAL SEARCH ALGORITHM.

---

```

BI-SEARCH ( $F_{1..k}, t_{1..k}, R_{1..k-1}, s_{init}, s_{final}$ )
1  $d \leftarrow \lfloor k/2 \rfloor - 1$ 
2  $e \leftarrow \lfloor k/2 \rfloor + 1$ 
3  $S_{head}^d \leftarrow \text{SEARCH}(F_{1..d}, t_{1..d}, R_{1..d}, s_{init})$ 
4  $S_{tail}^{k-e} \leftarrow \text{SEARCH}(F_{k..e+1}, t_{k..e+1},$ 
    $R_{k-1..e}, \text{REV}(s_{final}))$ 
5 if ( $S_{head}^d = \text{Failure}$  or  $S_{tail}^{k-e} = \text{Failure}$ ) then
6   Return Failure
7  $S^k \leftarrow \text{CONNECT}(S_{head}^d, F_{d+1}, t_{d+1},$ 
    $S_{tail}^{k-e}, F_e, t_e, R_{d+1}, s_{final})$ 
8 if ( $S^k = \emptyset$ ) then Return Failure
9 Return  $S^k$ 

```

---

```

CONNECT ( $S_1^m, F_1, t_1, S_2^n, F_2, t_2, R, s$ )
1  $k \leftarrow m + n + 2$ 
2  $S^k \leftarrow \emptyset$ 
3 for  $i = 1$  to  $W$  do
4    $\pi_1^m \leftarrow \text{RANDOM\_ELEMENT}(S_1^m)$ 
   //  $\pi_1^m = (p_{1,1}, p_{1,2}, \dots, p_{1,m}, s_1)$ 
5    $\pi_2^n \leftarrow \text{RANDOM\_ELEMENT}(S_2^n)$ 
   //  $\pi_2^n = (p_{2,1}, p_{2,2}, \dots, p_{2,n}, s_2)$ 
6    $z_1, z_2 \leftarrow \text{CONNECT\_ZMP}(s_1, t_1, s_2, t_2)$ 
7    $s_{new} \leftarrow \text{LIP}(s_1, z_1, t_1)$ 
8   if (INSIDE( $z_1, F_1$ ) and
   INSIDE( $z_2, F_2$ ) and
   INSIDE( $s_{new}, R$ )) then
9      $S^k \leftarrow S^k \cup \{(p_{1,1}, p_{1,2}, \dots, p_{1,m}, z_1, z_2,$ 
    $p_{2,n}, p_{2,n-1}, \dots, p_{2,1}, s)\}$ 
10  if ( $|S^k| = \text{Size}_{con}$ ) then Return  $S^k$ 
11 Return  $S^k$ 

```

---

*Claim:* Given a movement of the CoM from  $X_1$  to  $X_2$  with transition  $A$  and control input  $U$ , there necessarily exists a reverse moment as shown in Figure 5. from  $\text{REV}(X_2)$  to  $\text{REV}(X_1)$  with the same control input.

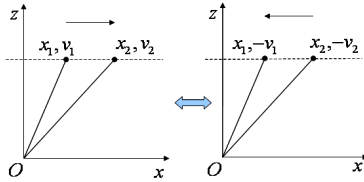


Fig. 5. A Pair of Inverse Movements

*Proof:* Let  $E$  denote  $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ . We get  $\text{REV}(X) = EX$  and  $EU = U$ . According to the definition of  $A$  as (8), we obtain  $EA = A^{-1}E$ , and according to (12), it satisfies

$$\begin{aligned}
X_2 &= AX_1 + (I - A)U \\
\Rightarrow EX_2 &= EAX_1 + E(I - A)U \\
\Rightarrow EX_2 &= A^{-1}EX_1 + (E - A^{-1}E)U \\
\Rightarrow AEX_2 &= EX_1 + (AE - E)U \\
\Rightarrow EX_1 &= AEX_2 + (I - A)EU
\end{aligned}$$

Therefore,  $\text{REV}(X_1) = A \cdot \text{REV}(X_2) + (I - A)U$ . We obtain a reverse movement from  $\text{REV}(X_2)$  to  $\text{REV}(X_1)$  with the same control input  $U$ .  $\square$

2) *Connection:* Connection is also a random search process. The search repeats  $W$  times to collect the con-

nected samples. In each attempt, the algorithm selects one random element  $head$ , from the head sample set, and one random element  $tail$ , from the tail sample set.  $\text{CONNECT\_ZMP}(s_1, t_1, s_2, t_2)$  is a function that calculates and returns a pair of ZMPs for the connection. We rewrite the final state of the head sample  $s_1$  as  $X_{head}$  and the final state of the tail sample  $s_2$  as  $X_{tail}$ . Let  $A_1$  denote the transition matrix for duration  $t_1$  and  $A_2$  for duration  $t_2$ . A pair of control inputs can be solved by the following equation:

$$A_1 X_{head} + (I - A_1)U_1 = \text{REV}[A_2 X_{tail} + (I - A_2)U_2] \quad (16)$$

Then the middle state is calculated by  $A_1 X_{head} + (I - A_1)U_1$ . When the state satisfies the kinematics range constraint and the two ZMP decisions generated from the control inputs are inside their areas of supporting foot, we obtain a connected sequence  $(p_{1,1}, p_{1,2}, \dots, p_{1,m}, z_1, z_2, p_{2,n}, p_{2,n-1}, \dots, p_{2,1})$ . The sequence is able to control the CoM to arrive at the specific final state, and the movement satisfies all the constraints which can be proved by the character of reverse moment.

## V. EXPERIMENTS

Figure 6 shows an example of ZMP sampling search for 10 steps of bipedal motion. In the figure, (a) and (b), the ZMP samples are shown as points, and the ZMP decisions are shown as crosses. The figure (c) presents the trajectory of the CoM which can be used in the lower level to control the joints with inverse kinematics.

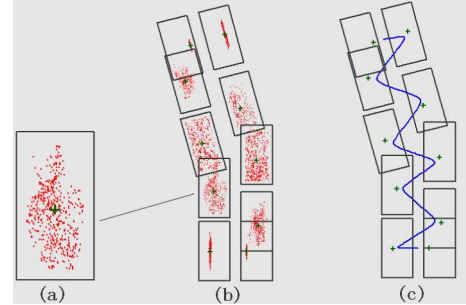


Fig. 6. ZMP samples (a) and (b), and CoM trajectory (c).

To study the performance of the algorithm, we performed thousands of searches for the same motion on a 2.0GHz computer. Each measurement for a specific generated size and expanded size is based on repeating the search 30 times. Figure 7 shows our results. The computation time reported is the average computation duration of the 30 trials; the average error presents the average deviation of the the ZMP solutions. We found that the computation time mainly depends on the generated size, though the number of expanded samples also contributes a fraction of the computation time. A similar phenomenon is seen with the average error.

Searching is more stable when the average error is small; however, in practical applications, we only need the average error less than a specific value, rather than as low as possible. To demonstrate the improvement of performance by convex expansion, we collected the smallest expanded size

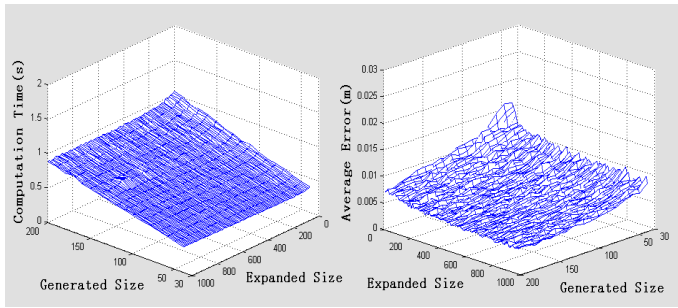


Fig. 7. Computation time and average error of different sample sizes.

and associated computation time when the average error is less than 8mm. As shown in Figure 8, with an appropriate expansion, the algorithm is clearly able to achieve the same average error without increasing the computation time.

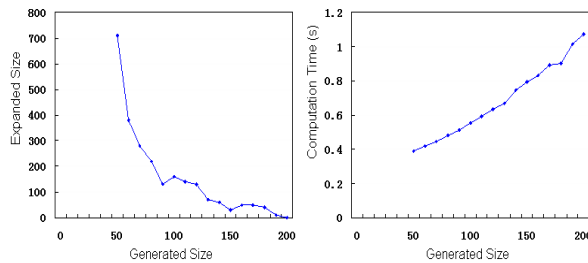


Fig. 8. Expanded size and computation time for the same average error.

We conducted a simulation of combined motion that demonstrates the use of ZMP sampling search for online biped walk planning. The demonstration continuously combined a series of motions, including walking, turning while walking, walking backward, sidle, turning, and stopping, as shown in the snapshots in Figure 9. Every three steps, the robot searched for the ZMP decisions of the next 8 steps online. Using this approach, the motion planning computation only utilized 20% of the available CPU time. Therefore, the algorithm has the capability to be executed online even in systems with lower CPU performance.

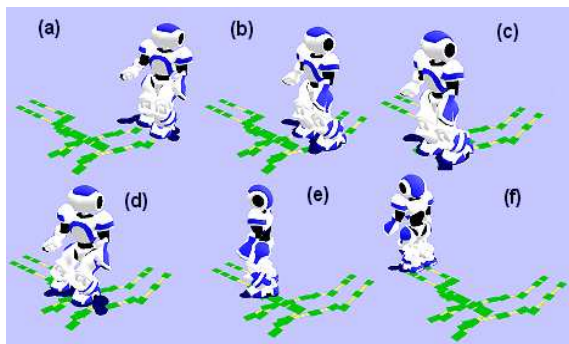


Fig. 9. A detailed walking demonstration.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a new method for biped walking planning, based on random search. We introduced the LIP model, the ZMP concept and a simplified walking model. We introduced an algorithm for ZMP sampling search, including unidirectional search and bidirectional search. In experiments, we studied how the search was affected by the generated size and expanded size. We have shown that the algorithm is able to accomplish walk planning online. We have also demonstrated a combined walking task in a simulated environment.

In the future, we are interested in converting the ZMP decision into a ZMP trajectory in order to make the walking of a real humanoid robot more stable. To achieve fast motions, the inertia and multi-body physical properties of the robot will be considered. Our next step is to include the mathematical analysis of these considerations in the ZMP sampling search.

## Acknowledgements

This research is supported in part by National Nature Science Foundation of China (Grant No. 60745002) and National 863 Programme (Grant No. 2008AA01Z150).

## REFERENCES

- [1] Branicky, M. and Curtiss, M., Sampling-Based Planning and Control, in *Proc. 2004 International Conference on Intelligent Robots and Systems*, 2004
- [2] Kajita, S. and Kanehiro, F. et al., The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation, in *Proce. of 2001 International Conference on Intelligent Robots and Systems*, vol. 1, 2001, pp. 239-246.
- [3] Sugihara, T. and Nakamura, Y. et al., Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control, in *Proce. of 2002 ICRA*, vol. 2, 2002, pp. 1404-1409
- [4] Sardain, P. and Bessonnet, G., Force Acting on a Biped Robot. Center of pressure-zero moment point, *IEEE Trans. Systems, Man and Cybernetics*, A, vol. 34, 2004, pp. 630-637
- [5] Kajita, S. and Kanehiro, F. et al., Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point, in *Proc. of 2003 Robotics and Automation*, vol. 2, 2003, pp. 1620-1626
- [6] Huang, Q. and Yokoi, K. et al., Planning Walking Patterns for a Biped Robot, *IEEE Trans. Robotics and Automation*, vol. 17, 2001, pp. 280-289
- [7] Kuffner, J. and Kagami, S. et al., Dynamically-stable Motion Planning for Humanoid Robots, *Autonomous Robots*, vol. 12, 2002, pp. 105-118
- [8] Kuffner, J. and Nishiwaki, K. et al., Motion Planning for Humanoid Robots, in *Proc. 11th Int'l Symp. of Robotics Research*, 2003
- [9] Kuffner, J. and Kagami, S. et al., Online Footstep Planning for Humanoid Robots, in *Proc. of 2003 ICRA*, vol.1, 2003, pp. 932-937
- [10] Kuffner, J. Nishiwaki, K. et al., Motion Planning for Humanoid Robots Under Obstacle and Dynamic Balance Constraints, in *Proc. 2001 IEEE International Conference on Robotics & Automation*, vol. 1, 2001, pp. 692-698
- [11] Lee, B. and Kong, J., et al., Optimal Trajectory Generation for a Humanoid Robot Based on Fuzzy and Genetic Algorithm, in *Proc. 2006 IEEE Congress on Evolutionary Computation*, 2006, pp. 1968-1974
- [12] Guan, Y. and Yokoi, K. et al., Reachable Space Generation of A Humanoid Robot Using The Monte Carlo Method, in *Proc. of 2006 International Conference on Intelligent Robotis and System*, 2006, pp. 1984-1989
- [13] Kagami, S. and Kanehiro, F. et al., AutoBalancer: An Online Dynamic Balance Compensation Scheme for Humanoid Robots, in *Proc. Int. Workshop Alg. Found. Robot.*, 2000
- [14] <http://www.aldebaran-robotics.com/>