

Humanoid Online Locomotion Generation based on Markov Chain with Microsoft Robotics Studio

Lingyun Hu, Changjiu Zhou, Bi Wu, Tianwu Yang

Abstract— The aim of this work is to generate and synchronize biped walking online with Microsoft Robotics Studio simulation platform. By defining three action primitives as Swing, Lift and Move, pose nodes with different geometric configuration and dynamical stability profile would be created for Markov chain. Using ZMP displacement as cost function and Maximum Likelihood Estimation Algorithm, Markov chain cut off redundant similar pose nodes. After training, only ε Independent Pose Group will be used in online gait generation system. The full-scale humanoid robot Robo-Erectus Senior can achieve flexible and dynamically stable walking online using the proposed algorithm. Only successful gaits qualified by Microsoft Robotics Studio will be applied on Robo-Erectus for practical locomotion. Action primitives make it easy to stabilize walking and changing step length and velocity smoothly.

I. INTRODUCTION

Humanoid robot is expected to walk and move like human as its name indicates. Biped gait can be generated by following different approaches, depending on whether on-line or off-line. Artificial intelligence is firstly introduced into off-line gait generation [1]. To satisfy the real-time requirement for on-line generation, two prevailing idea were proposed as passive walking model [2] and Zero Moment Point [3] based dynamically balancing techniques. For more general robot structure, ZMP provides a more universal dynamic balance criterion for generating and controlling.

Mostly, the dynamic walking control strategy of biped robots is based on walking pattern modification with preparative gaits. Online balancing algorithm will modify these gaits according to sensory feedback. Preview control of ZMP and future reference of desired ZMP trajectory are used in these kinds of online algorithms for gait tracking and modification with sensor information [4]. Also modification mainly takes on upper body motion for simplicity and efficiency.

Unlike these approaches, gaits are not generated by organizing the joint movement or with the distribution of actuated torques in the proposed method, they are constructed by candidate poses using Markov chain [5]. Similarly idea has been used in cartoon movie design and upper gesture generation but without consideration of the dynamical specification of biped robot itself.

Online gait generation system with predefined action primitives is proposed in this paper. Not gaits but three action

primitives are stored online. Different poses are created by combining these three primitives with different geometric configuration and dynamical stability profile. Sequence relationship in term of conditional probability between poses of conjoint time slices is formulated by Markov chain. Markov chain has been employed in gesture and gait recognition from image sequences [6]. It is used here to extract gait features from candidate poses for online storage and application.

By setting ZMP displacement as the cost function, conditional probability can be obtained with Maximum Likelihood Estimation (MLE) Algorithm. Similar poses will be cut off from Markov chain to form the ε Independent Pose Group. It decreases Markov chain size and improves efficiency of the online gait generation system. It can be concluded from the experimental result that action primitives is useful for classification, understanding and learning of novel movement tasks [7].

Moreover, gaits are tested in Microsoft Robotics Studio before physical experiments. Mechanical structure of Robo-Erectus is set in simulation platform. Successful gaits will be applied on robot then. It provides the connection between simulation and hardware part.

The rest of the paper is organized as follows. In the following section, biped robot model is presented with gait pattern and dynamical stability definition. Three action primitives and their operation are defined in Section IV. Section V describes the four components as candidate pose, probability function, cost function and Markov chain for online gait generation system. Microsoft Robotics Studio simulation platform is also introduced in this section. Based on it, the proposed algorithm with analysis will then be defined in Section V. Simulation and experimental results are shown in Section VI to verify the online gait generation system. This is followed by some concluding remarks in Section V.

II. BIPED ROBOT AND BIPED GAIT MODEL

A. Biped Gait Pattern

Biped locomotion, walking on level surface in sagittal plane, is assumed to have two phases as swing and double support phase. As shown in Fig. 1, for the k th gait cycle, the swing phase starts from the swing foot toeing off the ground at kT and ends when the heel touches the ground at $kT + T_a + T_b$; the swing foot reaches the highest point at $kT + T_a$ and after T_b lands on the ground. The double support phase begins at the moment that the front heel touching the ground and ends when the rear toe sways off after T_c time. End configuration of this phase initiates the swing phase at the next gait cycle.

This work was jointly supported by the Singapore Tote Fund and the Singapore Polytechnic R&D Fund. It is also partially supported by Singapore TB Model Project (11-27801-36-M096 and 11-30012-36-M096).

L. Hu, C. Zhou, B. Wu and Tianwu Yang are with the Advanced Robotic Intelligent Control Center in School of Electrical and Electronic Engineering at Singapore Polytechnic.

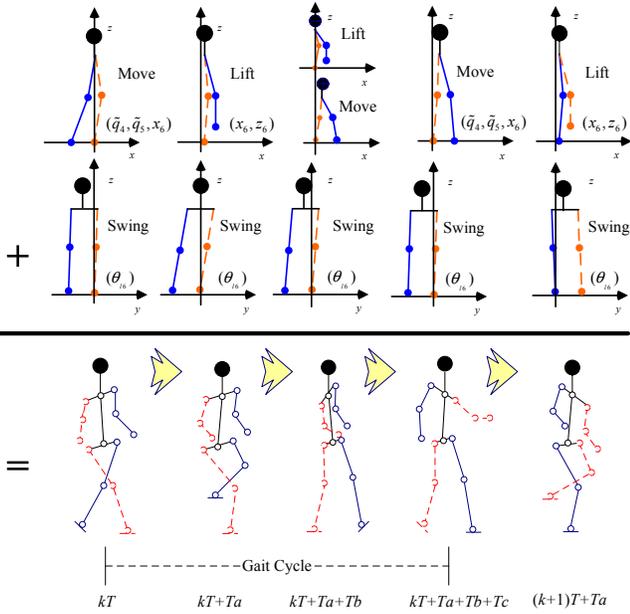


Fig. 1. Combination of primitive actions to form the candidate poses in one gait cycle

Landing Impact happens at the beginning of double support phase in infinitesimal length of time. Ratio between swing and double support phase is a function of walking velocity. To be mentioned in simulation as an example, swing phase is set to be twenty percent of one walking cycle according to [8].

Poses at this selected moments (kT , $kT + T_a$, $kT + T_a + T_b$ and $(k + 1)T$) are defined as key poses, which will be formed by the three action primitives, Swing, Lift and Move, as defined in Section III.

III. ACTION PRIMITIVES

Let $[q_1, \dots, q_7]$ be the coordinate sets describing the link locomotion in joint space with respect to a world reference frame. It combines left ankle, left knee, trunk, right knee, right ankle, left swing and right swing angles in sequence. The supporting ankle, noted as l_0 is adopted as the basic link of the mechanism. l_1 to l_5 represent left crus, left thigh, trunk, right thigh and right crus respectively.

A. Definition of Action Primitives

Let left leg be the standing leg (the blue one in Fig. 1). The three basic action primitives, Swing, Life and Move, are given in Fig. 2.

1) *Swing*: Action Swing brings q_6 and q_7 from θ to the designated angular value $\tilde{\theta}$ with zero ending velocity of frontal view joints. Let $\tilde{\mathbf{q}} = S(\mathbf{q}, \tilde{\theta})$, then

$$\tilde{\mathbf{q}} = \{q_1, \dots, q_5, \tilde{\theta}, \tilde{\theta}\}$$

2) *Lift*: Standing leg keeps motionless during Lift action. Feet maintain parallel to the ground in frontal plane. Swing foot is Supposed to move to $\tilde{\mathbf{p}} = (\tilde{x}_6, \tilde{y}_6, \tilde{z}_6)$. Let $\tilde{\mathbf{q}} = L(\mathbf{q}, \tilde{\mathbf{p}})$,

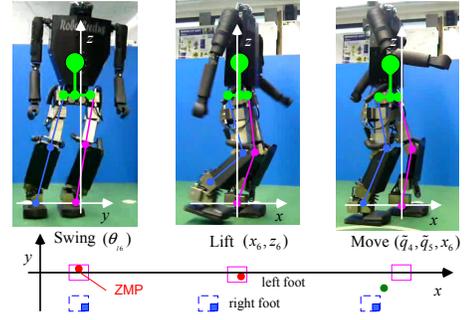


Fig. 2. The three action primitives.

$q_6 = q_7 = \theta$, then

$$\begin{aligned} \sum_{i=1,2,4,5} l_i \cos(\tilde{q}_i) \sin(\theta) &= \tilde{z}_6, \\ \sum_{i=1,2,4,5} l_i \sin(\tilde{q}_i) &= \tilde{x}_6 - x_1, \end{aligned}$$

$\tilde{q}_1 = q_1$, $\tilde{q}_2 = q_2$, $\tilde{q}_3 = 0$ and $\tilde{q}_6 = \tilde{q}_7 = \theta$.

Action Lift lifts up leg to avoid impact during walking. ZMP moves in correspondence to the lifted foot direction in the area of supporting foot, i.e. Ψ_{ls} and Ψ_{rs} . ZMP is confined in supporting foot to guarantee the dynamical stability.

3) *Move*: Supporting leg moves forward accompanied by the swing foot touching ground in the action of Move. Let $\tilde{\mathbf{q}} = M(\mathbf{q}, \tilde{q}_4, \tilde{q}_5, \tilde{x}_6)$, by setting \tilde{q}_4 , \tilde{q}_5 and pace length \tilde{x}_6 , it gets

$$\begin{aligned} \sum_{i=1,2,4,5} l_i \cos(\tilde{q}_i) &= 0, \\ \sum_{i=1,2,4,5} l_i \sin(\tilde{q}_i) &= \tilde{x}_6 - x_1, \end{aligned}$$

$\tilde{q}_3 = 0$, $\tilde{q}_6 = q_6$, $\tilde{q}_7 = q_7$.

Swing foot touches down after the action of Move. Different from action Lift, Standing leg sways forward in Move action to shift the ZMP from standing foot towards swing foot in the stable polygon Ψ_{rl} and Ψ_{lr} .

B. Primitive Combination

$\tilde{\mathbf{q}} = S(M(L(\mathbf{q}, \tilde{\mathbf{p}}), \tilde{x}_6), \tilde{\theta}) = S(M(L(\mathbf{q}, \Theta)))$, where $\Theta = \{\tilde{\mathbf{p}}, \tilde{x}_6, \tilde{\theta}\}$ is parameter set to be determined by pose configuration, then

$$\begin{aligned} \sum_{i=1,2,4,5} l_i \cos(\tilde{q}_i) &= 0, \\ \sum_{i=1,2,4,5} l_i \sin(\tilde{q}_i) &= \tilde{x}_6 - x_1, \\ \sum_{i=1,2} l_i \cos(q_i) \sin(\tilde{\theta}) + \sum_{i=4,5} l_i \cos(\tilde{q}_i) \sin(\tilde{\theta}) &= \tilde{z}_6, \\ \sum_{i=1,2} l_i \sin(q_i) + \sum_{i=4,5} l_i \sin(\tilde{q}_i) &= \tilde{x}_6 - x_1, \end{aligned}$$

and $\tilde{q}_3 = 0$.

Since action Swing effects on frontal plane, its order in the combination with Lift and Move is flexible.

IV. DEFINITION OF ONLINE GAIT GENERATION SYSTEM COMPONENT

Let $q(t) = [q_1(t), \dots, q_{N_q}(t)]^T$ be the angle vector of biped robot at time t . Where $q_i(t)$ is the angle of the i th joint at time t in global coordinate, $i = 1, \dots, N_q$. N_q denotes the number of actuated joint in biped mechanism. Correspondingly, velocity and acceleration vector of the biped robot can be described by $\dot{q}(t) = [\dot{q}_1(t), \dots, \dot{q}_{N_q}(t)]^T$ and $\ddot{q}(t) = [\ddot{q}_1(t), \dots, \ddot{q}_{N_q}(t)]^T$. $\dot{q}_i(t)$ and $\ddot{q}_i(t)$ are the velocity and acceleration of joint i , $i = 1, \dots, N_q$. Thus biped pose in angular space at time t can be represented by the vector $\phi(t) = [q(t), \dot{q}(t), \ddot{q}(t)]$.

We define the online gait generation system O with the three components as a finite set of biped candidate poses Φ , their corresponding conditional and united probability set $pro(\Phi)$ and the cost function $f(\Phi)$, that is, $O = \{\Phi, pro(\Phi), f(\Phi)\}$. It will be presented in Markov chain structure. Details of the three components and Markov chain are explained as follows.

A. Candidate Pose Φ

Supposing that $q(t_i, n_i)$ is the joint angle vector of the n_i th candidate pose at the t_i th key moment. $t_i = 1, \dots, N_{km}$ and $n_i = 1, \dots, N_{kp_i}$. N_{kp_i} is the total candidate pose at the t_i th key moment and N_{km} represents the sampling frequency. Hereby, the j th candidate pose at the t_i th sampling time can be described by $\phi(t_i, j) = [q(t_i, j), \dot{q}(t_i, j), \ddot{q}(t_i, j)]^T$. Let $A(t_i, n_i, \dots, n_j) = [n_i, \dots, n_j]$ records the index of candidate poses in time sequence between $[t_i, t_j]$. Hereby, phase constructed by $[\phi(t_i, n_i), \dots, \phi(t_j, n_j)]$ can be expressed as $\Phi(A(t_i, n_i, \dots, n_j))$. Specially, the gait formed by $[\phi(t_1, n_1), \dots, \phi(t_{km}, n_{km})]$ is $\Phi(A(t_1, n_1, \dots, n_{km}))$.

The candidate poses used to form the biped gaits are created off-line by the three action primitives. That is

$$\Phi(A(t_1, n_1, \dots, n_{km})) = S(L(M(\mathbf{q}, \dot{\mathbf{q}}), \Theta)). \quad (1)$$

Specification of parameter Θ is determined by pose specifications. With N_r prescribed gaits, where N_r is large enough to contain various kinds of poses for formulation, their pose sequence $\Phi(A(t_1, t_{km}))$ can be factorized into the three predefined action primitives, S , L and M by Primitive Component Analysis algorithm.

B. Probability Function $pro(\phi)$ and Related Concept On Distance

Let $pro(\Phi(A(t_i, n_i, n_j)))$ be the united probability of phases $\phi(t_i, n_i)$ and $\phi(t_j, n_j)$, $proc(\phi(t_{i+1}, n_{i+1})|\phi(t_i, n_i))$ be the conditional probability of pose $\phi(t_{i+1}, n_{i+1})$ with given pose $\phi(t_i, n_i)$. Then phase probability of $\Phi(A(t_i, n_i, \dots, n_j))$ can be computed by the conditional probability as $pro(\Phi(A)) = pro(\phi(t_i, n_i))proc(\phi(t_{i+1}, n_{i+1})|\phi(t_i, n_i)) \dots proc(\phi(t_j, n_j)|\phi(t_{j-1}, n_{j-1}))$. These probability concept will be used to define the pose and gait distances as follows.

Definition 2.1: (Pose Inner Distance) Let $A(t_i, n_i, n_{i+1})$ be the index set of phase sequence constructed by $[\phi(t_i, n_i), \phi(t_{i+1}, n_{i+1})]$, $n_i = 1, \dots, N_{kp_i}(t_i)$, $t_i =$

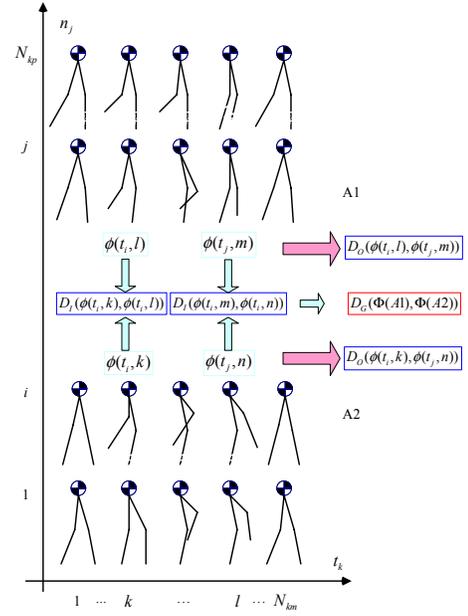


Fig. 3. Gait distance definition

$1, \dots, N_{km}$. Distance between two poses $\phi(t_{i+1}, k)$ and $\phi(t_{i+1}, l)$ is defined by

$$D_I(t_{i+1}, k, l) = \frac{1}{N_{kp_i}(t_i)} \sum_{n_i=1}^{N_{kp_i}(t_i)} \left| \lg \frac{pro(\Phi(A(t_i, n_i, k)))}{pro(\Phi(A(t_i, n_i, l)))} \right|. \quad (2)$$

The pose inner distance $D_I \geq 0$ and it equals to zero only when the two compared poses are the same. It can be regarded as a divergence function, weighting the discrepancy between the two poses logarithmically.

Definition 2.2: (ε Independent Pose Group) Let $\omega(t_i)$ be the group of candidate poses at time t_i as $\omega(t_i) = \{\phi(t_i, 1), \dots, \phi(t_i, N_{kp_i}(t_i))\}$. For a fixed nonnegative value ε , $\omega_\varepsilon(t_i)$ is called ε independent pose group if the difference of pose inner distance between any two poses $\phi(t_i, k)$ and $\phi(t_i, l)$ in $\omega_\varepsilon(t_i)$ is equal or greater than ε , i.e.

$$|D_I(t_i, j, k)| \geq \varepsilon$$

stands for any $1 \leq j, k \leq N_{kp_i}(t_i)$

By calculating the density probability $pro(\phi(t_i, n_i))$ and the united probability $pro(\Phi(A(t_i, n_i, n_j)))$ of poses, candidate poses $\phi(t_i, n_i)$ will then be sorted out to form the ε independent pose group $\Omega_\varepsilon(t_i)$. $n_i = 1, 2, \dots, N_{kp_i}(t_i)$.

The concept of ε independent pose group ensures the distinctness between poses in the group in term of probability density. It will be used as the selection standard for candidate poses. Similarly, we define the ζ independent gait group for online gait generation system.

C. Cost Function

Let cost function $f(\Phi) \rightarrow [0, 1]$

$$f(\Phi(A(t_i, n_i, n_j))) = \aleph \left(\sum_{t=t_k} |p_{zmp}(t) - p_{dzmp}(t)|^2 \right). \quad (3)$$

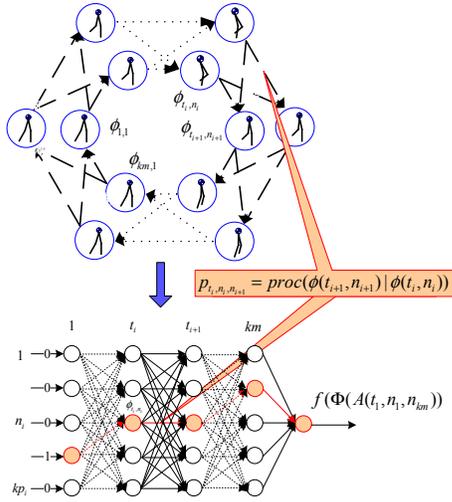


Fig. 4. Markov chain modelling on biped walking.

be formulated as the summation of ZMP displacements. $p_{zmp}(t) = [x_{zmp}(t), y_{zmp}(t)]$ and $p_{dzmp}(t) = [x_{dzmp}(t), y_{dzmp}(t)]$ are the real and desired ZMP positions at time t , t_k are key moments during one gait cycle.

If $\phi(A(t_{i-2}, n_{i-2}, n_{i+2}))$ is the pose constructed by third order spline function I on poses $\phi(t_{i-2}, n_{i-2})$, $\phi(t_{i-1}, n_{i-1})$, $\phi(t_{i+1}, n_{i+1})$ and $\phi(t_{i+2}, n_{i+2})$. Then the corresponding ZMP position $p_{zmp}(t_i, n_i)$ of $\phi(t_i, n_i)$ also satisfy the spline function with the four ZMPs' position $p_{zmp}(t_j, n_j)$, $j = i - 2, i - 1, i + 1, i + 2$. i.e. $p_{zmp}(t_i, n_i) = I(p_{zmp}(t_j, n_j))$

Furthermore, to manage the candidate poses and their related probability for online gait construction, Markov chain is applied as follows.

D. Markov Chain Model

With given pose candidate at t_i , Markov chain will select out consecutive ε independent pose group $\Omega_\varepsilon(t_{i+1})$ with the minimum cost function value by MLE Algorithm. The creation of candidate poses that needs large amounts of computation is finished off-line.

1) *Element of Markov Chain*: Supposing that the sequence of pose candidates $\phi(t_1, n_1), \dots, \phi(t_i, n_i), \dots, \phi(t_{km}, n_{km})$ is satisfied with the Markov property, namely that, given the present state, the future and past states are independent. Fig. 4 shows the designed Markov Chain used to represent the pose primitives in biped walking.

It contains km layers and kp_i nodes in the i th layer. States are connected by transition matrix $\mathbf{P} = [p_{t_i, n_i, n_{i+1}}]$, where

$$p_{t_i, n_i, n_{i+1}} = \text{proc}(\phi(t_{i+1}, n_{i+1}) | \phi(t_i, n_i)). \quad (4)$$

They are updated by MLE algorithm. For randomly selected pose sequence $A(t_1, \bar{n}_1, \bar{n}_{km}) = \{\phi(t_1, \bar{n}_1), \dots, \phi(t_i, \bar{n}_i), \dots, \phi(t_{km}, \bar{n}_{km})\}$, its cost function $f(\Phi(A))$ is used in updating as

$$p_{t_i, \bar{n}_i, \bar{n}_{i+1}} = p_{t_i, \bar{n}_i, \bar{n}_{i+1}} + 1 - f(\Phi(A)). \quad (5)$$

All wedges will be normalized after one step updating as

$$p_{t_i, n_i, n_{i+1}} = \frac{p_{t_i, n_i, n_{i+1}}}{\sum_{n_i} p_{t_i, n_i, n_{i+1}}}. \quad (6)$$

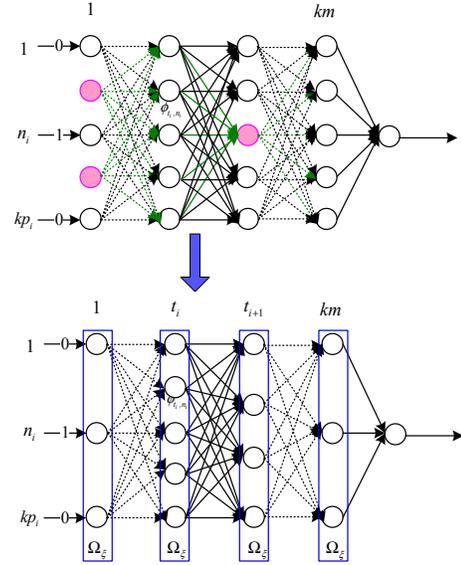


Fig. 5. Markov chain modification

2) *Modification of Markov Chain*: Using the trained Markov chain, conditional probability and thereby inner distance between candidate poses are achievable. Consequently, ε independent pose group at each key moment can be built up with pose inner distance.

Corresponding nodes of the redundant poses, that has close distance with ε independent pose group at each key moment will then be kicked out from the Markov chain. Accordingly, their related wedges are eliminated as exhibited in Fig. 5.

E. Microsoft Robotics Studio

Microsoft Robotics Studio is a Windows-based environment for hobbyist, academic and commercial developers to create robotics applications for a variety of hardware platforms. It includes a lightweight REST-style, service-oriented runtime, a set of visual authoring and simulation tools. We choose Microsoft Robotics Studio because

- Concurrency and coordination runtime makes writing and managing the asynchronous processes, like multi-motor control, easy.
- Decentralized System Services reuse model controllers inside and provide failure tolerance.
- Visualization tool enables fast prototyping and extensible code.

Gaits are tested firstly on Microsoft Robotics Studio platform. Successful gaits will then be applied on the physical robot.

V. CONSTRUCTION OF ONLINE GAIT GENERATION SYSTEM

The online gait generation system works as follows.

- **Formulation of candidate Poses** Candidate poses are initialized by the three action primitives according to Equation (1), where parameters are determined by the geometric features at each key moment as presented

TABLE I

KEY POSE GENERATION BASED ON ACTION PRIMITIVE COMBINATION

key moments	kp_1	kp_2	kp_3	kp_4
primitives	S+M	S+L	S+L+M	S+M
x_6 (m)	-0.15	0.02, 0.04	0.12, 0.14	0.15
θ_{l6} ($^\circ$)	3, 8	12	3, 8	± 1
z_6 (m)	-	0.04, 0.08	0.01, 0.02	-
q_4 ($^\circ$)	10 \sim 15	-	-20 \sim -15	-25 \sim -20
q_5 ($^\circ$)	15 \sim 20	-	-15 \sim -10	-20 \sim -15

in Fig. 1. For example, swing foot reaches the highest point $p_h = (x_h, y_h, z_h)$ at the middle of one gait cycle. Candidate pose at this moment will be constructed by combining Life and Swing with the parameter p_h . Number of candidate poses for each key moment is equal to kp . Totally $kp \times km$ key poses will be constructed for one gait cycle.

- **Calculation of cost function value** Randomly select N_r groups of pose sequence. For each candidate pose sequence $A(t_1, n_1, n_{km})$, cost function $f(\Phi(A))$ will be calculated for online gait generation as shown in Equation (3).
- **Performance of Markov chain** As demonstrated in Fig. 4, Markov chain is set up with the candidate poses. It is trained by the N_r groups of pose sequence and their cost function value as given in equations (5) and (6).
- **Calculation of pose distance** After training, conditional probability between candidate poses can be obtained as given in Equation (4). They will be used to acquire pose distance by Equation (2) and form the ε independent pose group.
- **Modification of Markov chain** Using the ε independent pose group, Markov chain is updated structurally with kp_i and $w_{t_i, n_i, n_{i+1}}$ as shown in Fig. 5.
- **Test on Microsoft Robotics Studio** Check Markov chain on Microsoft Robotics Studio platform.
- **Generation of gait online** With the finalized Markov chain, for given candidate pose $\phi(t_i, n_i)$, its next pose $\phi(t_{i+1}, n_{i+1})$ with the minimal cost function value will be chosen. i.e.

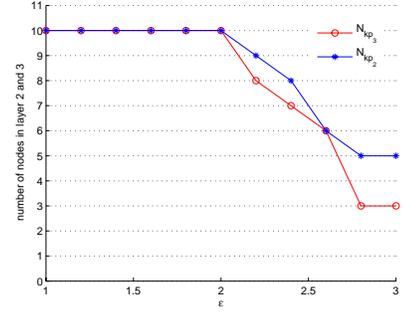
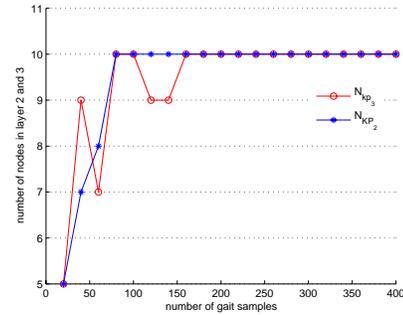
$$\phi(t_{i+1}, n_{i+1}^*) = \operatorname{argmin} f(\phi(t_{i+1}, n_{i+1}))$$

VI. EXPERIMENTAL RESULTS

A. Robot Experimental Platform

The proposed online generation system is applied on the biped experimental platform Robo-Erectus Senior. It got 4th place in the Humanoid TeenSize League at RoboCup2007. It was also featured in CNN Live (<http://www.youtube.com/watch?v=12Hsw1piR4c>). Mass and length of robot crus, femur and trunk are [4, 2.5, 16] kg and [0.246, 0.346, 0.5] m. Pace length is 20cm. Gait height is set to be 4cm and 8cm two kinds of value. Let $N_q = 7$, $N_{km} = 4$, $N_{kp_i} = 10$ for $i = 2, 3$ and $N_{kp_i} = 5$ for $i = 1, 4$. Then parameters for the key pose generation with the three action primitives are given in Table I.

100 gait samples are used to train the Markov Chain. Let ε vary from 1 to 3 with step length of 0.2, the updated number

Fig. 6. N_{kp_2} and N_{kp_3} with different ε .Fig. 7. N_{kp_2} and N_{kp_3} with different number of gait samples.

of nodes in the second and the third layer of Markov chain is summarized in Fig 6. As expected, node number decreases when distance boundary factor ε increase from 2 to 3. Too big distance boundary reduces number of nodes sharply. For example, both layer 2 and 3 cut down 70% nodes when $\varepsilon = 3$. They form $\zeta = 12$ Independent Gait Group then.

Let $\varepsilon = 2.5$. As number of gait samples N_r changes from 20 to 400 with step length of 50, N_{kp_2} and N_{kp_3} varies as shown in Fig 7. They increase as number of gait sample increases and keep to 10 when $N_r > 160$. That is because Markov chain is updated by MLE Algorithm. Inner distance between nodes is magnified as more gait samples are evolved in updating. In other words, large number of gait samples can differentiate similar nodes in Markov chain with the cost of calculation. Since modification of Markov chain can simplify structure and thereby alleviate calculation load, to screen out closer nodes, ζ would have a relationship of increasing function with N_r .

B. Online Experiment

In online application, for given pose n_{t_1} at the first key moment, Markov chain can select out the most stable pose at the next key moment. Then primitive components to form these poses are sent to online control panel. They are decomposed to joint angles and used to drive motors in Robo-Erectus Senior. Action primitives classify joint angles into groups of frontal and sagittal planes respectively. It synchronizes joint angles in locomotion and makes walking subhuman.

Let $T = 3s$, $T_a = 0.6s$, $T_b = 1.8s$, feedback joint angles for biped walking from stand still, followed by two gait cycles and last stepping are presented in Fig 8. To keep foot parallel

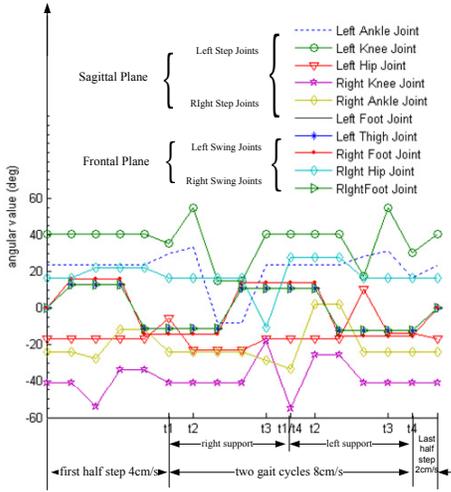


Fig. 8. Joint angles of biped walking.

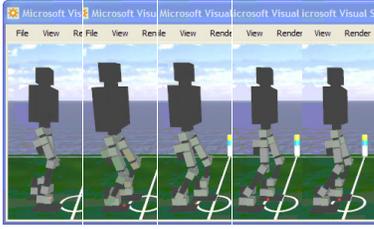


Fig. 9. Snapshot of walking in Microsoft Robotics Studio simulation.

to ground in frontal plane, left and right foot and thigh angles vary in phase. All joints works during $[-80^\circ, 80^\circ]$, which is only half of the maximum angular velocity of each joint motor ($150^\circ/s$) when walking with the velocity of $8cm/s$. The corresponding simulation result in Microsoft Robotics Studio simulation is presented in Figure 9. The red point is the calculated ZMP under simulated environment. The successful gait is applied on experimental platform then.

Fig. 10 is the snapshots of online walking with four key poses in sequence. Robot walks from stand still with the velocity of $0.04m/s$ and accelerates to $0.08m/s$ after first half step. Then it performs two full gait cycles and stops with the last half step ($0.02m/s$). It changes walking velocity smoothly by setting different input pose in Markov chain. In key pose 1, robot swings to stand foot direction (see the most left figure in Fig. 10). It swings more and lifts up swing foot (left one) in key pose 2 (see the second left figure in Fig. 10). As shown in the third figure of Fig. 10, the swing foot touches ground in key pose 3 by Move action. Robot swings back in key pose 4 finally and changes to key pose 1 as given in the first right figure in Fig. 10.

ZMP trajectory of two gait cycles is given in Fig. 11. ZMP shifts from one supporting foot to the other one and keeps in stable region within the sequence of $\{\Psi_{rs}, \Psi_{rl}, \Psi_{ls}, \Psi_{lr}\}$ during locomotion.

VII. CONCLUSION

This paper presents the online gait generation algorithm with Markov chain and action primitives. For given pose,



Fig. 10. Snapshot of walking in sagittal plane.

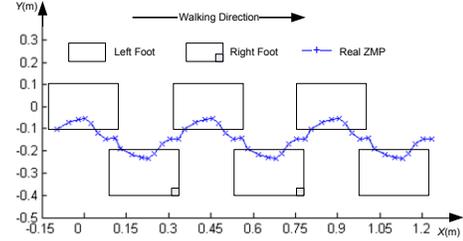


Fig. 11. ZMP trajectory of biped walking.

the trained Markov chain can recognize the most stable pose at the next key moment. Online control algorithm will then decompose the acquisition and perform the corresponding action primitives. Action primitives help to differentiate poses in Markov chain and harmonize the realization of joint permutations.

The proposed algorithm has been applied on the Microsoft Robotics Studio and the full-scale humanoid robot Robo-Erectus Senior separately. The simulation platform is used as a reliable test tool with physical model and 3D environment simulation. Using the interface between hardware and the application scenery provided by Microsoft Robotics Studio, future work is hope to control robot with browser.

REFERENCES

- [1] D. Kati and M. Vukobratovic, "Survey of intelligent control techniques for humanoid robots," *Journal of Intelligent and Robotic Systems*, vol. 37, no. 2, pp. 117–141, 2003.
- [2] T. McGeer, "Passive dynamic walking," *Int. Jour. of Rob. Res.*, vol. 9, pp. 62–82, 1990.
- [3] M. Vukobratovic and B. Borovac, "Zero moment point—thirty five years of its life," *Int. Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.
- [4] Y. Ogura, S. Andoa, H. ok Lim, and A. Takanishi, "Sensory-based walking motion instruction for biped humanoid robot," *Robotics and Autonomous Systems*, vol. 48, no. 4, pp. 223–230, 2003.
- [5] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol. 77, no. 2, Feb. 1989, p. 257286.
- [6] R. Zhang, C. Vogler, and D. Metaxas, "Human gait recognition at sagittal plane," *Image and Vision Computing*, vol. 25, no. 3, pp. 321–330, 2007.
- [7] E. Drumwright and V. Ng-Thow-Hing, "The task matrix: an extensible framework for creating versatile humanoid robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Orlando, 2006, pp. 448–455.
- [8] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie, "Planning walking patterns for a biped robot," *IEEE Trans. on Rob. and Aut.*, vol. 17, no. 3, pp. 280–290, 2001.