Proceedings of the 2002 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
EPFL, Lausanne, Switzerland • October 2002

Online Generation of Humanoid Walking Motion based on a Fast Generation Method of Motion Pattern that Follows Desired ZMP

Koichi Nishiwaki[1], Satoshi Kagami[2], Yasuo Kuniyoshi[1], Masayuki Inaba[1], Hirochika Inoue[1]

[1]*The University of Tokyo, Tokyo, Japan, {nishi,kuniyosh,inaba,inoue}@jsk.t.u-tokyo.ac.jp,*
[2]*National Institute of Advanced Industrial Science and Technology, Tokyo, Japan, s.kagami@aist.go.jp*

Abstract

This paper presents an efficient online method to generate humanoid walking motions that satisfy desired upper body trajectories while simultaneously carrying objects. A fast motion pattern genaration technique that follows the desired ZMP is adopted. In order to satisfy the control input given online, subsequent motion patterns are updated and connected stably to the old ones while executing. During the creation of motion trajectories online, the commanded motion parameters are checked and modified automatically considering the performance limitations of the hardware. As an example application, we have implemented a one step cycle control system on the Humanoid H7. Experiments controlling the upper body motion and walking direction using a joystick interface are explained to demonstrate the validity of the proposed method.

1 Introduction

Humanoid robots are considered to have the advantage of working in environments designed for real humans. In order to flexibly operate in environments where humans live and work, the capability of changing its motion online according to sensory information or control commands is important since the environment is not well modeled and always changing. Of course, it also applies to walking motion.

On the other hand, biped humanoids have more complicated dynamics model than biped walking robots that are developed primarily to verify walking theories. Because of the high cost of calculating the dynamics, walking on biped humanoids has been realized by constructing a dynamically-stable trajectory in advance, and executing it[1, 2, 3].

In recent years, several researches that realize online control of humanoid walking were reported. Setiawan, et al. realized online control of forward and backward walking by connecting motion patterns generated in advance[4]. Yokoi, et al. realized online generation of walking pattern[5] applying Three-Dimensional Linear Inverted Pendulum Mode based method[6] to humanoid type robot. Lim, et al. proposed "quasi-realtime" walking pattern generation using FFT based dynamically stable motion construction method online[7].

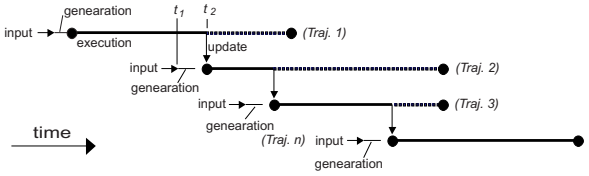

Figure 1: *Online Control of Walking by Trajectory Updates*

We also proposed a real-time walking pattern generation method that make a humanoid to follow specified footprint locations online by dynamically stable mixture and connection of pre-designed motion trajectories[8]. However upper body motion and step cycle are limited by prepared motion trajectories. In this paper, new online walking control method is presented. This method is based on a fast generation method of motion pattern that follows desired ZMP[9]. Since what can be specified online only depends on the limitation of the adopted generation method, in this case online generation allows to specify upper body motion, step cycle, and swinging leg trajectories as well as footprint locations. Change of dynamics model such as carrying an object can also be managed online.

2 Online Control of Humanoid Walking

2.1 Online Generation and Updates of Position based Trajectory

In order to apply position based trajectory generation method for online walk control, we propose repetitive generation and updates method. As shown in Figure 1, a first trajectory that satisfies the con-

trol value at that moment is generated and walking motion starts by executing it (*Traj. 1*). During the execution, at t_1 generaton of a trajectory (*Traj. 2*) that connects to the executing trajectory (*Traj. 1*) at t_2 and satisfies the control value at t_1 starts. Here t_2 is decided by adding estimated upper limit of generation time and some time margin to t_1. Then at t_2 *Traj. 1* is connected to *Traj. 2* smoothly. Repeating this procedure walking motion that follows control values given online is realized witout stopping. All the trajectories are designed to make a robot stably stop if executed to the last. Owing to this design the robot will stop safely executing the rest of the trajectory when updates of trajectory fails for some reason, unless the change of the environment affects the robot motion.

In order to apply a motion trajectory generation method to this online update system, following characteristics are required;

- Time for generation is quick enuough,
- It is possible to generate a trajectory that connects to a specified motion state smoothly,
- It can guarantee that the generated trajectories are executable on the real robot considering robot performance constraints as well as dynamical stability constraints.

2.2 Calculation Time

It is necesarry for generation time to be shorter than its motion time in order to update the motion continuously, and less generation time achieves less latency from the control input. The adopted generation method includes iterative calculation to gain higher accuracy. The calculation time is proportional to the motion time and the number of iteration. It takes 3 to 6 times iteration to obtain the motion the average ZMP error of which is less than 2.0[mm]. When 6 times iteration is required, it takes about 2.4% of its motion time to generate the dynamically stable trajectory on the computer inside the robot (Humanoid H7: dual Pentium III 1.1[GHz], calculation step: 50[ms]). We implemented the method in the realtime layer of RT-Linux[10] and made it possible to limit the number of iteration in order to guarantee the upper limit of calculation time.

2.3 Smooth Connection to the Previous Trajectory

The adopted generation method changes horizontal position of upper body trajectory from initially given motion in order to satisfy desired ZMP, but it does not change the position and the velocity at the ends of a trajectory from the initial ones. When a initial trajectory of connecting motion is designed, the position and velocity at the connecting time are set to the same as the connected ones. Then the position and velocity will be the same at a connecting point.

2.4 Validation of Realizability of Generated Motion Trajectories

Since this online generation method can generate various walking motion and real robots have many performance constraints, realizability on a real robot becomes a problem. Following performance constraints are taken into consideration in this paper;

- Existence of the solution for inverse-kinematics of legs,
- Limitation of joint angle range,
- Limitation of joint angle velocity,
- Collision between links.

Dynamically stable motion generation time takes most of online generation time. If it becomes evident that the generated trajectory can not be realized on a real robot after generation, it is impossible to regenerate a trajectory that connects to the same time. In order to raise the probability that the trajectory can be executed, validation of trajectory is carried out in the following two steps;

- Heuristic limitation of paremeters that are used for trajectory generation,
- Validation of generated trajectory.

The details of this process is described in section 4.2.

3 Generation of Walking Trajectory that Follows Desired ZMP

In this section the fast dynamically stable walking trjectory generation method we adopted is explained. In general, ZMP trajectory can be analytically led from the robot motion trajectory, however leading robot trajectory that satisfies given ZMP trajectory analitically is difficult, since it has to solve nonlinear, interfered 2^{nd} order differencial equation with joint constraints. In this section, a fast generation method of walking trajectory that follows given ZMP trajectory by modifying horizontal torso trajectory from given initial trajectory is explained.

3.1 Linearlize and Decouple the Dynamic Equation about ZMP

Let i^{th} robot link position, mass, inertia tensor, angular velocity vector be $\mathbf{r}_i = (x_i, y_i, z_i)^{\mathrm{T}}, m_i, \mathbf{I}_i, \omega_i,$

and gravity be g. Set the coordinates as x-y plane be the ground and as z-axis be negative gravity direction. Let ZMP be $\mathbf{P} = (x_p, y_p, 0)^{\mathrm{T}}$.

Then equation to lead ZMP from robot motion is described as follows (the same for y_p);

$$x_p = \frac{\sum m_i z_i \ddot{x}_i - \sum \{m_i(\ddot{z}_i + g)x_i + (0,1,0)^{\mathrm{T}}\mathbf{I}_i\dot{\boldsymbol{\omega}}_i\}}{-\sum m_i(\ddot{z}_i + g)}. \tag{1}$$

Let robot motion trajectory be described as follows;

$$\begin{aligned}\mathbf{A}(t) = \quad & (x_1(t), y_1(t), z_1(t), \theta_1(t), \phi_1(t), \psi_1(t), \\ & \cdots, x_n(t), y_n(t), z_n(t), \theta_n(t), \phi_n(t), \psi_n(t)).\end{aligned} \tag{2}$$

ZMP trajectory $\mathbf{P}_A(t) = (x_{p_a}(t), y_{p_a}(t), 0)^T$ can be solved by Eq. (1). Now consider to generate trajectory that follows desired ZMP $\mathbf{P}^*_A(t)$ by only modifying $x_i(t), y_i(t)$ in $\mathbf{A}(t)$ to $x_i^{'}(t), y_i^{'}(t)$.

$$x^*_p = \frac{\sum m_i z_i \ddot{x}_i^{'} - \sum \left\{m_i(\ddot{z}_i + g)x_i^{'} + (0,1,0)^{\mathrm{T}}\mathbf{I}_i\dot{\boldsymbol{\omega}}_i\right\}}{-\sum m_i(\ddot{z}_i + g)}. \tag{3}$$

This problem is solving $x_i^{'}$ that satisfies Eq. (3) (the same for $y_i^{'}$). Eq. (3) − Eq. (1) gives the following equation;

$$x^e_p = \frac{\sum m_i z_i \ddot{x}^e_i - \sum m_i(\ddot{z}_i + g)x^e_i}{-\sum m_i(\ddot{z}_i + g)}. \tag{4}$$

Here, $x^e_p = x^*_p - x_{p_a}, x^e_i = x_i^{'} - x_i$.
Since there is redundancy, consider $x^e_i = x^e$, that is, modifying the horizontal position of all the link in same distance. In reality, feet position can not be changed relative to the ground, however upper-body can satisfy this modification. When modifying upper body position in horizontal plane, the position changes of leg links, which are led by inverse-kinematics, are approximately propotional to that of the upper body in horizontal plane and little for rotational and vertical component.

3.2 Numerical Solution of Differencial Equation

Set $x^e_i = x^e$ to Eq. (4), then following equation is obtained;

$$-\frac{\sum m_i z_i}{\sum m_i(\ddot{z}_i + g)}\ddot{x}^e + x^e = x^e_p. \tag{5}$$

In order to solve numerically time is discretized to $0, \cdots, t_m$ with time step Δt. Accelaration at each time $\ddot{x}^e(t_i)$ can be represented as follows;

$$\ddot{x}^e(t_i) = \frac{x^e(t_{i+1}) - 2x^e(t_i) + x^e(t_{i-1})}{\Delta t^2}. \tag{6}$$

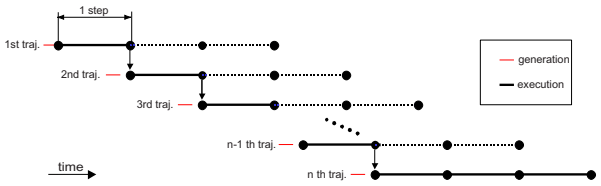

Figure 2: *One Step Cycle Controlling System.*

Then Eq. (4) can be expressed as trinomial equations. Setting boundary conditions as $x^e(0) = 0$ and $x^e(t_m) = 0$, $x^e(i)$ $(i = 1$ to $t_m - 1)$ are obtained.

Therefore, robot trajectory can be calculated that satisfies given ZMP trajectory $P^*_A(T)$.

In order to obtain more accurate trajectory, calculated trajectory is set to initial trajectory, and this procedure is repeated.

3.3 Walking Trajectory Generation System

Walking trajectory generation is carried out in following steps: 1) Initial posture, initial velocity, final posture, final velocity, arm trajectory, foot posture at each discrete step, time for single & dual leg support phase, torso posture at the middle of dual leg support phase, etc. are given, 2) Initial motion trajectory, and desired ZMP are generated, and 3)The motion trajectory that satisfies desired ZMP is solved as mentioned above.

4 Design and Implementation of One Step Cycle Controlling System

Every one step cycle controlling system is designed as an application of the online walking control system described in section 2. In this design desired motion vector (horizontal translation and rotation around vertical axis) of next step, next step cycle time, and upper body motion during next step are specified for every one step.

4.1 Length of Trajectory to Generate

Since the trajectory is updated every one step, the minimum length of the trajectory is two steps, a step that satisfies the control value and the following stopping step. Maximizing the connectivity when the same control value is given is considered. Then three steps, a step that satisfies the control value, next step that follows the same control value, the last step to stop the motion, are generated every one step. Table. 1 shows the velocity and the position at the end of first step (at the middle of dual leg support phase) of forward walking (200[mm], 1[s] per a step) according to the number of generated steps. This indicates

Table 1: *State of Motion at the End of First Step*

num. of steps	position		velocity	
	x[mm]	y[mm]	x[mm/s]	y[mm/s]
2	96.560654	-1.425083	343.660	304.362
3	98.659909	-2.316348	352.912	301.460
4	98.738370	-2.289558	353.323	301.537
5	98.741380	-2.290479	353.343	301.534
6	98.741504	-2.290453	353.345	301.534

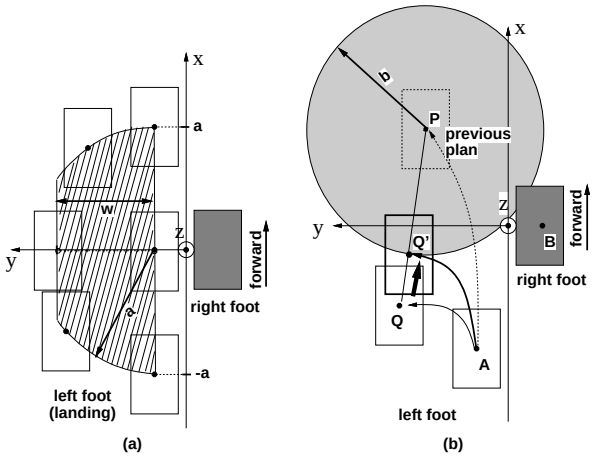

Figure 3: *Limitation of Landing Position. (a) Relative to the supporting leg foot position, (b) Relative to the previously planned position.*

there is no need to generate more than three steps in order to maximize the connectivity.

4.2 Limitation of Parameters and Validation of Trajectory

As mentioned in section 2.4, in order to construct realizable trajectory, limitation of parameters before trajectory generation and validation of trajectory after generation are carried out.

Limitation of parameters given to the trajectory generation system is performed in following steps;

1. Landing position is limited in the area shown in Figure 3(a) relative to the position of supporting leg foot, so that the foot can be reached. It is circle area whose center is neutral landing position and it is also limited by the sideward width. The narrowest is the neutral landing position.

2. Limit the change of the first step landing position of a new trajectory from the planned position in the previous (connected) trajectory. Figure 3(b) is the situation to generate a new trajectory that connects to old one when left and right feet are at A and B respectively. The next landing position was planned to be P in the previous (executing) trajectory, then if the first step landing position of new trajectory is set to Q, it is modified to Q′.

   It reduces the discontinuity of the acceleration at the connecting point. Reduction of the discontinuity contributes to the little change of ZMP at that point.

3. Limit minimum sideward distance of the landing position from the foot of supporting leg in order to prevent collision of links between two legs. The minimum distance is function of relative rotational angle between two feet.

4. Limit the maximum hip joint height from the ankle joint of the same leg (h) to $\sqrt{l_0^2 - a^2}$ (Figure 4). This helps to increase probability of existence of solution for inverse kinematics and to reduce the maximum velocity of knee joints.

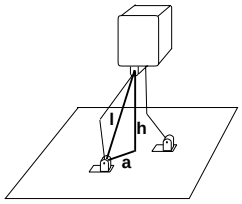

Figure 4: *Limitation of Torso Height.*

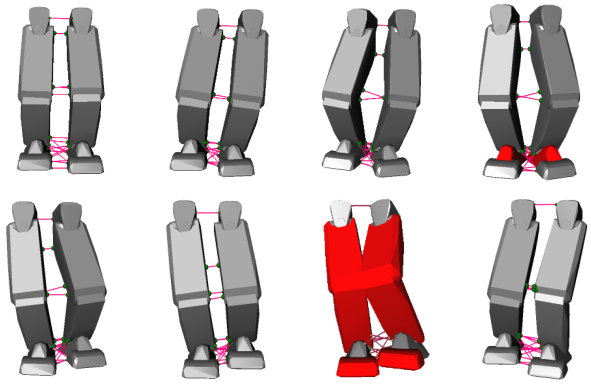

Figure 5: *Collision detection between the links of two legs.* [11]

Validation after trajectory generation is carried out for the following points;

- Range limitation of leg joint angle,
- Limitation of leg joint angle velocity,
- Collision between leg links.

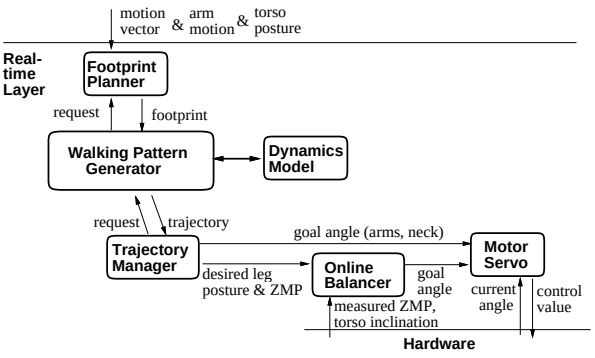

Figure 6: *Online Walking Control System.*

A fast mimimum distance calculation system[11] is adopted for collision detection (Figure 5). We set some safety margin for each check, because the trajectories are modified by sensor feedback when executed. If a trajectory is judged not realizable, the update is abandoned and the robot stops executing the rest of the current trajectory.

4.3 Calculation Time

The step cycle can be changed from 0.7 to 2.0[s] in this implementation. Then the longest trajectory is 5.2[s]. (Last one step is always 1.2[s].) And the maximum calculation time for generating the trajectory was about 215[ms]. The time for collision check was 15 to 25[ms] (Sampling time is 50[ms] and the number of checked pairs is reduced to 19 considering the possibility of collision.) Calculation time for other parts is negligible comparing with these two. Start time of generation is set to 250[ms] before the start of execution.

4.4 Software System

The walking control system is implemented in the realtime layer of RT-Linux (Figure 6). *Foorprint Planner* is a module that generates footprint locations from desired torso motion. *Walking Pattern Generator* is the module that generates three step realizable trajectories. Generated trajectories are stored in *Trajectory Manager*, and this module updates the control value of *Motor Servo* and *Online Balancer* modules periodically (1[ms] cycle). It also notify the *Walking Pattern Generator* of the time for the next generation.

4.5 Controlling Experiment using Joystick Interface

Controlling system with joystick interface is implemented as an application. Joystick is connected to a PC outside, and commands are sent via wireless LAN. Forward and sideward axes and twisting axis of the joystick are assigned to the desired translation and rotation of torso motion. Desired torso posture is controlled by combination of thrust lever and buttons. Several arm postures are also assigned to buttons. The robot successfully walked while changing its arm and torso posture (Figure 7, 8, 9). Changing dynamics model online, walking with shaking a 1[kg]-bottle was also realized (Figure 10).

5 Conclusion

Online walking control method by updating position based trajectory is proposed. A fast generation method of motion pattern that follows desired ZMP was explained. The method is suitable for dynamically stable trajectory generation for online walking control. Performance constraints as well as dynamics constraints are taken into consideration in the control sysytem. One step cycle controlling system was implemented, and controlling experiments with joystick interface showed the validity of the method.

Utilizing the modification caused by sensor feedback for later trajectory generation is the next reserch topic.

References

[1] Kazuo Hirai. Current and future perspective of Honda humanoid robot. In *In Proceeding of 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'97)*, pp. 500–508, 1997.

[2] Jin'ichi Yamaguchi, Sadatoshi Inoue, Daisuke Nishino, and Atsuo Takanishi. Development of a bipedal humanoid robot having antagonistic driven joints and three dof trunk. In *Proc. of the 1998 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 96–101, 1998.

[3] Ken'ichirou Nagasaka, Masayuki Inaba, and Hirochika Inoue. Walking pattern generation for a humanoid robot based on optimal gradient method. In *Proc. of 1999 IEEE Int. Conf. on Systems, Man, and Cybernetics*, 1999.

[4] Samuel Agus Setiawan, Sang Ho Hyon, Jin'ichi Yamaguchi, and Atsuo Takanishi. Physical interaction between human and a bipedal humanoid robot – realization of human-follow walking –. In *Proceddings of the 1997 IEEE International Conference on Robotics and Automation*, pp. 361–367, 1999.

[5] Kazuhito Yokoi, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Shuji Kajita, and Hirohisa Hirukawa. A honda humanoid robot controlled by aist software. In *Proceddings of the IEEE-RAS International Conference on Humanoid Robots*, pp. 259–264, 2001.

[6] Shuji Kajita, Osamu Matsumoto, and Muneharu Saigo. Real-time 3D walking pattern generation for

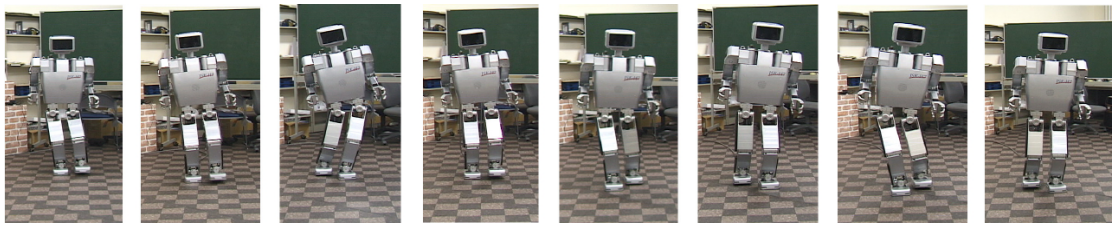

Figure 7: *Online Walking Control. (Changing Torso Posture around Roll Axis.)*

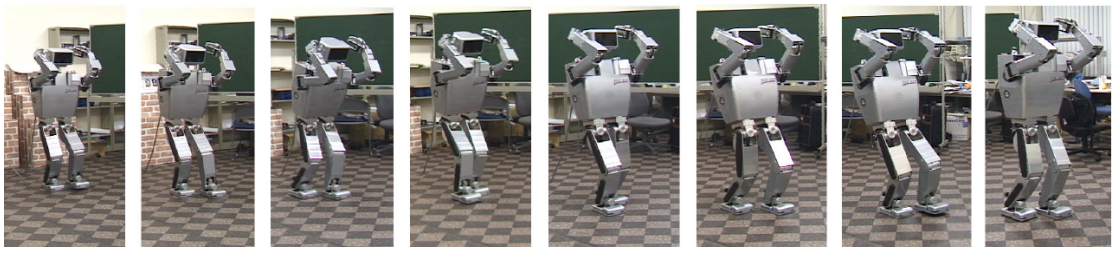

Figure 8: *Online Walking Control. (Changing Torso Posture around Pitch Axis.)*

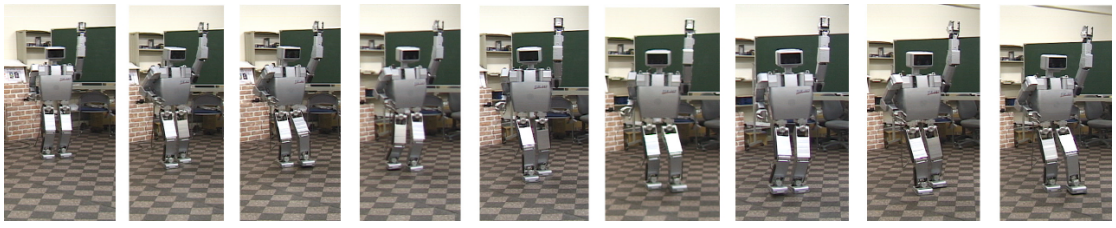

Figure 9: *Online Walking Control. (Changing Torso Posture around Yaw Axis.)*

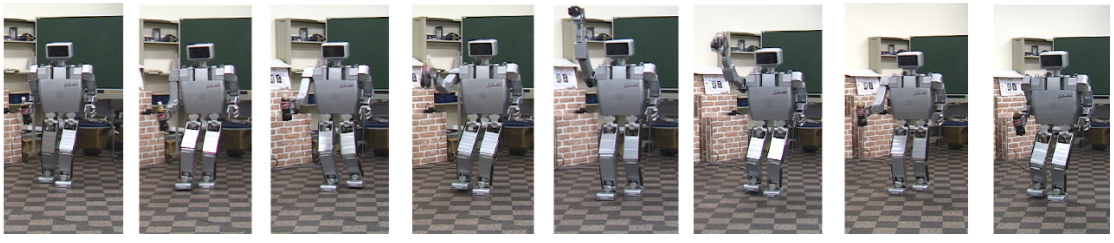

Figure 10: *Online Walking Control. (Shaking a Bottle (about 1 [kg]).)*

a biped robot with telescopic legs. In *Proceddings of the 2001 IEEE International Conference on Robotics and Automation*, pp. 2299–2306, 2001.

[7] Hun-ok Lim, Yoshiharu Kaneshima, and Atsuo Takanishi. Online Walking Pattern Generation for Biped Humanoid Robot with Trunk. In *Proc. of IEEE International Conference on Robotics and Automation*, pp. 3111–3116, 2002.

[8] K. Nishiwaki, T. Sugihara, S. Kagami, M. Inaba, and H. Inoue. Realtime generation of humanoid walking trajectory by mixture and connection of pre-designed motions – online control by footprint specification –. In *Proc. of International Conference on Robotics and Automation (ICRA'01)*, pp. 4110–4115, 2001.

[9] S. Kagami, K. Nishwiaki, T. Kitagawa, T. Sugihara, M. Inaba, and H. Inoue. A fast generation method of a dynamically stable humanoid robot trajectory with enhanced zmp constraint. In *Proc. of IEEE International Conference on Humanoid Robotics (Humanoid2000)*, 2000.

[10] Michael Barabanov. A linux-based real-time operating system. Master's thesis, New Mexico Institute of Mining and Technology, 1997.

[11] James Kuffner, Koichi Nishiwaki, Satoshi Kagami, Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Self-collision detection and prevention for humanoid robots. In *Proc. of International Conference on Robotics and Automation (ICRA'02)*, pp. 2265–2270, 2002.