Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control

Tomomichi Sugihara[*1] Yoshihiko Nakamura[*1*2] Hirochika Inoue[*1]
[*1]Dept. of Mechano-Informatics, Univ. of Tokyo.
7–3–1, Hongo, Bunkyo-ku, Tokyo, 113–0033, Japan.
Email: sugihara@ynl.t.u-tokyo.ac.jp
[*2]CREST Japan Science and Technology Corporation(JST).

Abstract

Humanoid robot is expected as a rational form of machine to act in the real human environment and support people through interaction with them. Current humanoid robots, however, lack in adaptability, agility, or high-mobility enough to meet the expectations. In order to enhance high-mobility, the humanoid motion should be generated in realtime in accordance with the dynamics, which commonly requires a large amount of computation and has not been implemented so far. We have developed a realtime motion generation method that controls the center of gravity(COG) by indirect manipulation of the zero moment point(ZMP). The realtime responce of the method provides humanoid robots with high-mobility. In this paper, the algorithm is presented. It consists of four parts, namely, the referential ZMP planning, the ZMP manipulation, the COG velocity decomposition to joint angles, and local control of joint angles. An advantage of the algorithm lies in its applicability to humanoids with a lot of degrees of freedom. The effectiveness of proposed method is verified by computer simulations.

1 Introduction

Humanoid robot has a high potential being a future form of computer that acts and supports our daily activities in the shared infrastructures with the human beings. In spite of such expectation, humanoid robots at the present substantially lack in mobility. They can neither cope with sudden contacts with the unknown, nor respond to unexpected dicisions due to emergency such as 'stop' or 'avoid'. A technical reason lies in the fact that no control algorithm has been developed or implemented in a responsive form that allows the above-stated high-mobility.

Previous works in motion generation of humanoids can be classified into:

- **Trajectory replaying**
  prepares a joint-motion trajectory in advance, and applies it to the real robot with a little on-line modification [1][2][3]. This method divides the problem into two subproblems, namely, planning and control.
- **Realtime generation**
  generates a joint-motion in realtime, feeding back the present state of the system in accordance with the pre-provided goal of the motion [4][5][6]. This method executes planning and control in a unified way.

Although realtime generation is more promissing than trajectory replaying from the viewpoint of high-mobility, they commonly suffer from a large amount of computations. Realtime generation would requires developments to overcome the difficulty.

The goal of the present paper is to provide humanoids with high-mobility, developing a realtime motion generation method, positively using dynamics of robots. One idea is to use the dynamical relationship between the **ZMP**[7] and the center of gravity(**COG**). The legged system has a similar dynamics to that of inverted pendulum, whose supporting point is equivalently located at the ZMP. We propose the method that controls the COG of the whole humanoid body system in realtime through the ZMP manipulation.

Park et al. proposed an off-line method to design the ZMP trajectory using the fuzzy logic[8]. Sorao et al. also proposed the ZMP manipulation[9], which unfortunately is limited to a specific kinematics. The proposed method in this paper has such an superior advantage over the past works that it is compatible with arbitrary degrees-of-freedom and kinematics, and considers a low dimensional part of the body dynamics to enable realtime computation.

The full scale motion generation and control of humanoid will require a huge computation and a breakthrough, since it is in nature under the nonholonomic constraints due to the underactuatedness. The way to manipulate the ZMP proposed in this paper significantly reduces the complexity of the problem and thus allows realtime implementation. Although it is a pragmatic approach in the sense that all the inertial forces other than the gravitation are not explicitly considered in control, its effectiveness over stabilization for legged motion was remarkable and verified by computer simulation.

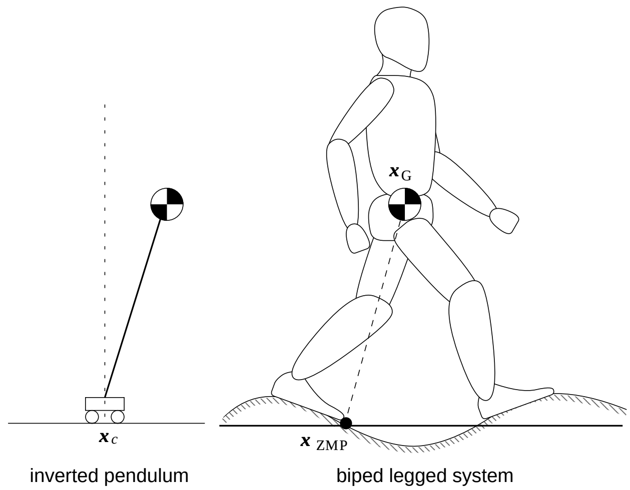


Figure 1: Inverted pendulum and legged system

2 Realtime Motion Generation

2.1 The Principle of Legged Motion and Inverted Pendulum

Legged motion is caused through an interaction between the robot and the environment. A robot generates the internal forces(*i.e.* joint torques) at each joint, and converts them into the external forces through interactions with the environments(floor). A metaphor of inverted pendulum is often used to explain such a system, since it is unstable in nature and controls its motion indirectly using the external forces generated as reaction forces.

Legged robots generally interact with the environment at multiple contact points. They can be equivalently represented by a single point on a virtual floor and its associated force/moments. The point *is* the ZMP. And the virtual floor is called the **VHP** (Virtual Horizontal Plane), more to be explained in the next subsection.

We have the following two approximate equations:

$$\text{Equation of Motion} \qquad m\ddot{\boldsymbol{x}}_G = -m\boldsymbol{g} + \boldsymbol{f} \quad (1)$$

$$\text{Geometric Constraint} \qquad \boldsymbol{f} = k(\boldsymbol{x}_G - \boldsymbol{x}_{ZMP}) \quad (2)$$

where m is the total mass of the robot, $\boldsymbol{x}_G$ is the position of the COG, $\boldsymbol{x}_{ZMP}$ is the position of the ZMP, $\boldsymbol{g}$ is the acceleration of gravity, $\boldsymbol{f}$ is the total external force acting to the robot, and k is an arbitrary positive constant.

Given $z = z_{ZMP}$ (a height of the VHP along z-axis, the direction of gravitation), $\boldsymbol{x}_{ZMP}$ can be get uniquely. Then, we get

$$\ddot{x}_G = \omega_G^2(x_G - x_{ZMP}) \quad (3)$$

$$\ddot{y}_G = \omega_G^2(y_G - y_{ZMP}) \quad (4)$$

where ω_G is defined by

$$\omega_G \equiv \sqrt{\frac{\ddot{z}_G + g}{z_G - z_{ZMP}}} \quad (5)$$

From these equations, we can find out that the system has almost the same dynamics with inverted pendulum whose supporting point is equivalently located at the ZMP(Fig.1). Thus, it is possible to control the COG of the system as well as inverted pendulum through the ZMP manipulation.

2.2 The VHP and The Supporting Polygon

Since the ZMP is the working point of the total external force which acts to the system in the direction to push it, it must exist in *the supporting region*, which is constructed from all of the contact points between the robot and the environment. They are, however, ranged 3-dimensionally. Thus, it is hard to discuss the feasibility of the ZMP in such a 3-dimensional region.

Kitagawa et al. proposed the enhanced ZMP[10] to consider such 3-dimensional contacts, which is applicable only for geometrically simple environments. We show the method to solve this problem; substituting each real contact point to the equivalent point on the

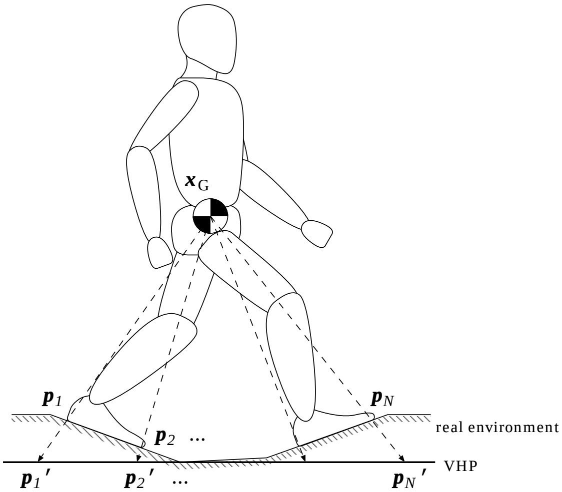


Figure 2: VHP(Virtual Horizontal Plane)

VHP in terms of supporting the COG, and making a convex hull of them. It *is* the feasible region of the ZMP on the VHP.

The equivalent contact points on the VHP can be get with the following way — suppose $\boldsymbol{p}_1 \sim \boldsymbol{p}_N$ are the real contact points between the robot and the environment, a force vector which acts to the COG $\boldsymbol{x}_G$ at the point $\boldsymbol{p}_i$ can be slided along the line which connects $\boldsymbol{x}_G$ and $\boldsymbol{p}_i$. The point of intersection $\boldsymbol{p}'_i$ between the line and the VHP is equivalent to $\boldsymbol{p}_i$. Then, $\boldsymbol{p}'_1 \sim \boldsymbol{p}'_N$ and a convex hull of them are obtained(Fig.2). We call it *the supporting polygon* hereafter.

In accordance with this idea, any situations about contact can be treated in the same way, and the feasibility of the ZMP on the VHP becomes able to be discussed easily.

2.3 The Algorithm

Based on the fact described in 2.1, motion of robots can be generated and controled in realtime with the following algorithm —

1. **Referential ZMP Planning**
   Using the control method for inverted pendulum (for example, PID controller, H∞, and so on), the referential ZMP ${}^{ref}\boldsymbol{x}_{ZMP}$ is decided as an actuating value in order to let the COG velocity converge to a reference ${}^{ref}\boldsymbol{v}$. If ${}^{ref}\boldsymbol{x}_{ZMP}$ is out of the supporting polygon, the nearest point in the polygon to ${}^{ref}\boldsymbol{x}_{ZMP}$ is chosen instead.

2. **ZMP manipulation**
   In the case of inverted pendulum, the supporting point can be manipulated directly, while it is impossible in the case of legged system, that is to say, the ZMP cannot be moved directly. Then, the realtime referential COG velocity ${}^{ref}\dot{\boldsymbol{x}}_G$ (different from ${}^{ref}\boldsymbol{v}$, and called *the strict referential COG velocity* hereafter) could be calculated in order to let the real ZMP coincide with ${}^{ref}\boldsymbol{x}_{ZMP}$. The ZMP is manipulated with it indirectly.

   At first, the factor along z-axis is gotten independently, which is concerned with the motion in the direction of the acceleration of gravity. This is an essencial value to make robots more adaptable or active; they can adapt to rough terrain, absorbing large impulse from the floor(like a compliance control), or can even hop and run, if the value is set adequately.

   Here, we show the simplest way to decide it by the following expression:

   $$ {}^{ref}\ddot{z}_G = K_z({}^{ref}v_z - \dot{z}_G) \tag{6} $$

   where ${}^{ref}\ddot{z}_G$ is the referential acceleration along z-axis, K_z is the proportional gain, ${}^{ref}v_z$ is the given referential velocity along z-axis, and $\dot{z}_G$ is the real COG velocity along z-axis. With this rule, $\dot{z}_G$ converges to ${}^{ref}v_z$.

   From ${}^{ref}\ddot{z}_G$, ${}^{ref}\boldsymbol{x}_{ZMP}$, $\boldsymbol{x}_G$, and the equations (3)(4), the strict referential COG *acceleration* ${}^{ref}\ddot{\boldsymbol{x}}_G = \begin{bmatrix} {}^{ref}\ddot{x}_G & {}^{ref}\ddot{y}_G & {}^{ref}\ddot{z}_G \end{bmatrix}^T$ can be calculated as:

   $$ {}^{ref}\ddot{x}_G = {}^{ref}\omega_G^2(x_G - {}^{ref}x_{ZMP}) \tag{7} $$
   $$ {}^{ref}\ddot{y}_G = {}^{ref}\omega_G^2(y_G - {}^{ref}y_{ZMP}) \tag{8} $$

   where ${}^{ref}\omega_G$ is defined by

   $$ {}^{ref}\omega_G \equiv \sqrt{\frac{{}^{ref}\ddot{z}_G + g}{z_G - {}^{ref}z_{ZMP}}} \tag{9} $$

   Integrating these equations, the strict referential COG velocity ${}^{ref}\dot{\boldsymbol{x}}_G$ can be obtained.

3. **Decomposition of the strict referential COG velocity**
   Suppose $\boldsymbol{\theta}$ is a joint angle vector($n\times 1$, n:DOF), the COG $\boldsymbol{x}_G$ is expressed as a function with an argument $\boldsymbol{\theta}$ like $\boldsymbol{x}_G(\boldsymbol{\theta})$. Thus, there exists a Jacobian $\boldsymbol{J}_G$ which can relate $\dot{\boldsymbol{\theta}}$ to $\dot{\boldsymbol{x}}_G$ as:

   $$ \dot{\boldsymbol{x}}_G = \boldsymbol{J}_G\dot{\boldsymbol{\theta}} \tag{10} $$

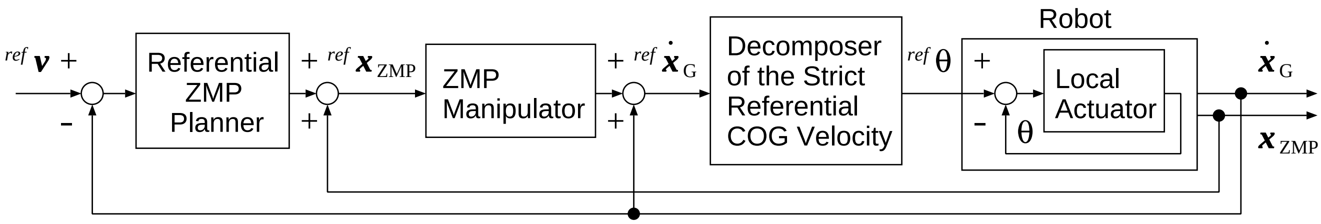


Figure 3: Block chart of the realtime motion generation algorithm

where $\boldsymbol{J}_G$ ($3\times n$) is defined by

$$\boldsymbol{J}_G \equiv \frac{\partial \boldsymbol{x}_G}{\partial \boldsymbol{\theta}} \tag{11}$$

We call $\boldsymbol{J}_G$ *the COG Jacobian* hereafter.

$\boldsymbol{x}_G$ is a quite complex function with multiple arguments. Tamiya et al. proposed the numerical method to calculate it [11], which unfortunately needs a large amount of computation. We developed a fast and accurate calculation method of $\boldsymbol{J}_G$ with the following numerical approach.

Firstly, The relative COG velocity with respect to the base coordinates(which moves with the base link of the robot together) ${}^0\dot{\boldsymbol{x}}_G$ can be expressed as:

$${}^0\dot{\boldsymbol{x}}_G = \frac{\sum_{i=0}^{n-1} m_i {}^0\dot{\boldsymbol{r}}_i}{\sum_{i=0}^{n-1} m_i} = \frac{\sum_{i=0}^{n-1} m_i {}^0\boldsymbol{J}_{Gi}\dot{\boldsymbol{\theta}}}{\sum_{i=0}^{n-1} m_i} \tag{12}$$

where m_i is the mass of link i, ${}^0\boldsymbol{r}_{G,i}$ is the position of the center of mass of link i with respect to the base coordinates, and ${}^0\boldsymbol{J}_{G,i}$ ($3\times n$) is defined by

$${}^0\boldsymbol{J}_{G,i} \equiv \frac{\partial {}^0\boldsymbol{r}_{G,i}}{\partial \boldsymbol{\theta}} \tag{13}$$

Therefore, Jacobian ${}^0\boldsymbol{J}_G$ which relates $\dot{\boldsymbol{\theta}}$ to ${}^0\dot{\boldsymbol{x}}_G$ is

$${}^0\boldsymbol{J}_G = \frac{\sum_{i=0}^{n-1} m_i {}^0\boldsymbol{J}_{Gi}}{\sum_{i=0}^{n-1} m_i} \tag{14}$$

Secondly, suppose link F is fixed in the world coordinates (for example, when the right leg is the supporting leg, the right foot link is fixed), the COG velocity with respect to the world coordinates $\dot{\boldsymbol{x}}_G$ is

$$\begin{aligned}\dot{\boldsymbol{x}}_G &= \dot{\boldsymbol{x}}_0 + \boldsymbol{\omega}_0 \times \boldsymbol{R}_0 {}^0\boldsymbol{x}_G + \boldsymbol{R}_0 {}^0\dot{\boldsymbol{x}}_G \\ &= \boldsymbol{R}_0\{{}^0\dot{\boldsymbol{x}}_G - {}^0\dot{\boldsymbol{p}}_F + ({}^0\boldsymbol{x}_G - {}^0\boldsymbol{p}_F) \times {}^0\boldsymbol{\omega}_F\} \\ &= \boldsymbol{R}_0\{{}^0\boldsymbol{J}_G - {}^0\boldsymbol{J}_F + [({}^0\boldsymbol{x}_G - {}^0\boldsymbol{p}_F)^\times]{}^0\boldsymbol{J}_{\omega F}\}\dot{\boldsymbol{\theta}}\end{aligned} \tag{15}$$

where $\boldsymbol{x}_0$ is the position of the base link in the world coordinates, $\boldsymbol{\omega}_0$ is the rotation velocity of the base link with respect to the world coordinates, $\boldsymbol{R}_0$ is the attitude matrix of the base link with respect to the world coordinates, ${}^0\boldsymbol{p}_F$ is the position of the fixed link in the base coordinates, ${}^0\boldsymbol{\omega}_F$ is the rotation velocity of the fixed link with respect to the base coordinates, ${}^0\boldsymbol{J}_F$ is the Jacobian about linear velocity of the fixed link with respect to the base coordinates, ${}^0\boldsymbol{J}_{\omega F}$ is the Jacobian about rotation velocity of the fixed link with respect to the base coordinates, and $[\boldsymbol{v}^\times]$ means outer-product matrix of a vector $\boldsymbol{v}$ (3×1).

Then, $\boldsymbol{J}_G$ can be calculated as:

$$\boldsymbol{J}_G = \boldsymbol{R}_0\{{}^0\boldsymbol{J}_G - {}^0\boldsymbol{J}_F + [({}^0\boldsymbol{x}_G - {}^0\boldsymbol{p}_F)^\times]{}^0\boldsymbol{J}_{\omega F}\} \tag{16}$$

When the robot executes tasks in the real world, not only the COG velocity of it is controlled, but some kinds of comstraints about its motion must be satisfied in addition, which are classified to —

- **constraints in joint-space**
  constraints giving referential values for certain joint angles $\boldsymbol{\theta}_f$ directly. In this case, the COG velocity would be manipulated by the other angles $\boldsymbol{\theta}_u$, and it is possible to treat ${}^{ref}\dot{\boldsymbol{\theta}}$ as it is ${}^{ref}\boldsymbol{\theta} = \begin{bmatrix} {}^{ref}\boldsymbol{\theta}_u^T & {}^{ref}\boldsymbol{\theta}_f^T \end{bmatrix}^T$
- **constraints in non-joint-space**
  constraints about motions of extremities(end effectors), or physical amounts(momentum around the COG, for example). They are expressed by the equation as:

$$\boldsymbol{J}_C\dot{\boldsymbol{\theta}} = \boldsymbol{c}$$

When the equation and the strict referential COG velocity ${}^{ref}\dot{\boldsymbol{x}}_G$ are given, we can have the following equation:

$$\begin{bmatrix} \boldsymbol{J}_G \\ \boldsymbol{J}_C \end{bmatrix} {}^{ref}\dot{\boldsymbol{\theta}} = \begin{bmatrix} {}^{ref}\dot{\boldsymbol{x}}_G \\ {}^{ref}\boldsymbol{c} \end{bmatrix} \tag{17}$$

where ${}^{ref}\dot{\boldsymbol{\theta}}$ is the referential joint angle vector.

Now, $\boldsymbol{J}_G$ and $\boldsymbol{J}_C$ are decomposed $\begin{bmatrix} \boldsymbol{J}_{Gu} & \boldsymbol{J}_{Gf} \end{bmatrix}$ and $\begin{bmatrix} \boldsymbol{J}_{Cu} & \boldsymbol{J}_{Cf} \end{bmatrix}$, corresponding to each ${}^{ref}\boldsymbol{\theta}_f$ and ${}^{ref}\boldsymbol{\theta}_u$. And then,

$$\begin{bmatrix} \boldsymbol{J}_{Gu} & \boldsymbol{J}_{Gf} \\ \boldsymbol{J}_{Cu} & \boldsymbol{J}_{Cf} \end{bmatrix} \begin{bmatrix} {}^{ref}\dot{\boldsymbol{\theta}}_u \\ {}^{ref}\dot{\boldsymbol{\theta}}_f \end{bmatrix} = \begin{bmatrix} {}^{ref}\dot{\boldsymbol{x}}_G \\ {}^{ref}\boldsymbol{c} \end{bmatrix} \tag{18}$$

which can be simplified as:

$$\boldsymbol{J}_u {}^{ref}\dot{\boldsymbol{\theta}}_u = \boldsymbol{v} \tag{19}$$

The rank of $\boldsymbol{J}_u$ does not always coincide with a degree-of-freedom n of the robot. Thus, using pseudo inverse matrix of $\boldsymbol{J}_u$ with weights $\boldsymbol{J}_u^{\#}$, the equation is solved as:

$${}^{ref}\dot{\boldsymbol{\theta}}_u = \boldsymbol{J}_u^{\#}\boldsymbol{v} \tag{20}$$

Finally, we can get ${}^{ref}\dot{\boldsymbol{\theta}}$. And integrating it, ${}^{ref}\boldsymbol{\theta}$ would be obtained.

4. **Local control for each joint actuator**
Each actuator of joint angle is controled to be converged to the reference ${}^{ref}\boldsymbol{\theta}$. As a result, the whole-body motion of the robot is generated. The inertial forces(or the internal forces) of the system are implicitly considered here.

Fig.3 is a block chart which shows the outline of this algorithm.

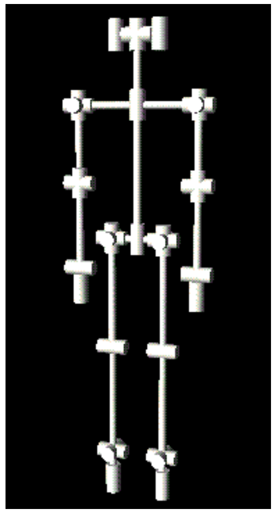
3 Simulation

We generated

- the 3-dimensional COG displacement on both feet
- stepping motion
- one-step-forward motion and walking

in a simulator in order to verify the algorithm. Kinematic structure, size and mass properties of the robot are shown in Fig.4. Each simulation showed superior agility of the robot.

A stepping motion with a disturbance — impact — has also been simulated, and we made sure that the proposed algorithm can give the robot enough adaptability; the robot didn't upset in spite of such a disturbance. Fig.5 is a snapshot of one of the test motions. Fig.6 shows loci of the COG, the referential ZMP and the real ZMP in this motion(the ZMP is transformed to the point with respect to the base coordinates of the robot). We can see that the real ZMP follows the referential ZMP with a good accuracy.



DOF:	30
	(4 for head)
	(14 for arm)
	(12 for leg)
height:	1270 [mm]
weight:	35 [kg]

Figure 4: Kinematic structure, size and mass of the robot

4 Conclusion

We developed a realtime motion generation method that can provide humanoids with superior adaptability and agility, namely, high-mobility essencial for robots to act and support human beings in the real world. It is based on the similarity between the dynamics of robots and the inverted pendulum, and reduses computations for realtime implementation.

The proposed algorithm consists of four subsystems, the referential ZMP planner, the ZMP manipulator, the COG velocity decomposer, and local controller of joint angles, and possesses such algorithmic generality that it is applicable even for humanoids with a lot of degrees of freedom.

We verified usefullness of the method by generating some motions in computer simulations.

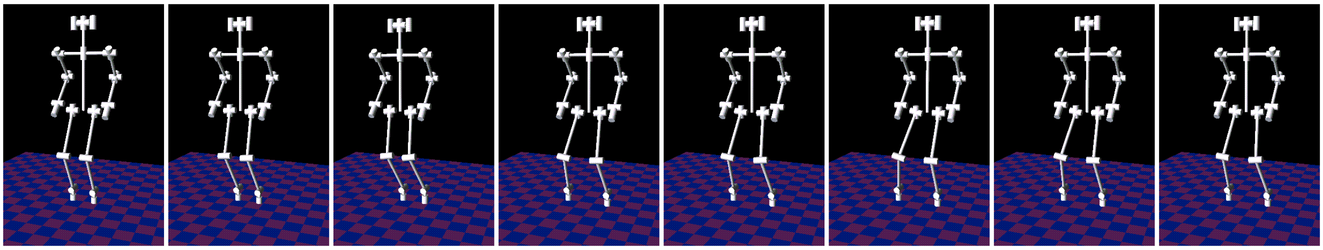


Figure 5: A stepping motion with an impact

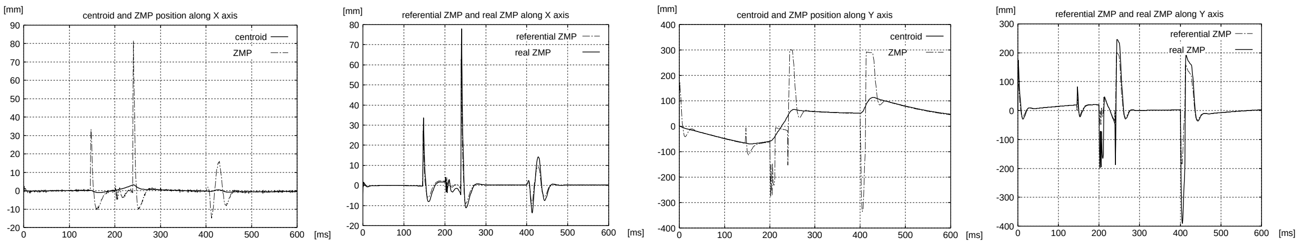


Figure 6: Loci of the COG, the referential ZMP and the real ZMP

References

[1] Kazuo Hirai, Masato Hirose, Yuji Haikawa and Toru Takenaka. The Development of Honda Humanoid Robot. In *In Proceeding of 1998 IEEE International Conference on Robotics & Automation*, pages 1321–1326, 1998.

[2] Jin'ichi Yamaguchi, Eiji Soga, Sadatoshi Inoue and Atsuo Takanishi. Development of a Bipedal Humanoid Robot – Control Method of Whole Body Cooperative Dynamic Biped Walking –. In *Proceedings of the 1999 IEEE International Conference on Robotics & Automation*, pages 368–374, 1999.

[3] Ken'ichirou Nagasaka, Masayuki Inaba and Hirochika Inoue. Walking Pattern Generation for a Humanoid Robot Based on Optimal Gradient Method. In *Proceedings of 1999 IEEE International Conference on Systems, Man, and Cybernetics*, pages VI–908–913, 1999.

[4] Yasutaka Fujimoto and Atsuo Kawamura. Simulation of an Autonomous Biped Walking Robot Including Environmental Force Interaction. *IEEE Robotics & Automation Magazine*, 5(2):33–41, 1998.

[5] Jerry Pratt and Gill Pratt. Exploiting natural dynamics in the control of a planar bipedal walking robot. In *Proceedings of the 36 Annual Allerton Conference on Communication, Control, and Computing*, 1998.

[6] Hiroki Takeuchi. Development of "MEL HORSE". In *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, pages 3165–3171, 2001.

[7] Miomir Vukobratović, B.Borovac, D.Surla and D.Stokić. *Biped Locomotion: Dynamics, Stability, Control and Application*, volume 7 of *Scientific Fundamentals of Robotics*. Springer-Verlag, 1990.

[8] Jong H. Park and Yong K.Rhee. ZMP Trajectory Generation for Reduced Trunk Motions of Biped Robots. In *Proceedings of the 1998 IEEE International Conference on Intelligent Robots and Systems*, pages 90–95, 1998.

[9] K. Sorao, T. Murakami and K. Ohnishi. A Unified Approach to ZMP and Gravity Center Control in Biped Dynamic Stable Walking. In *Proceedings of Advanced Intelligent Mechatronics*, 1997.

[10] S. Kagami, K. Nishiwaki, 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 *Proceedings of the IEEE International Conference on Humanoid Robotics(Humanoid2000)*, 2000.

[11] S. Kagami, F. Kanehiro, Y. Tamiya, M. Inaba and H. Inoue. AutoBalancer: An Online Dynamic Balance Compensation Scheme for Humanoid Robots. In *Proceedings of the 4th International Workshop on Algorithmic Foundation on Robotics(WAFR'00)*, 2000.