International Journal of Humanoid Robotics

An Analytical Method on Real-time Gait Planning for a Humanoid Robot

Kensuke Harada, Shuuji Kajita, Kenji Kaneko, and Hirohisa Hirukawa
Humanoid Research Group, Intelligent Systems Institute
National Institute of Advanced Industrial Science and Technology(AIST)
1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, JAPAN

This paper studies the real-time gait planning for a humanoid robot. By simultaneously planning the trajectories of the COG(Center of Gravity) and the ZMP(Zero Moment Point), the fast and smooth change of gait can be realized. The change of gait is also realized by connecting the newly calculated trajectories to the current ones. While we propose two methods for connecting two trajectories, i.e. the real-time method and the quasi-real-time one, we show that the stable change of gait can be realized by using the quasi-real-time method even if the change of the step position is significant. The effectiveness of the proposed methods are confirmed by simulation and experiment.

1. Introduction

The ultimate goal of research on a biped walking robot is to realize the adaptive and robust gait like a human. For example, when a biped robot moves in an environment including many obstacles, the robot has to detect the obstacles and change the walking direction in real-time to avoid an obstacle. However, the walking motion of a biped robot has been realized by replaying the pre-planned trajectories, and it has been difficult to change the position of the step in real-time. On the other hand, while there have been a few researches on the real-time gait planning of a biped walking robot, this research provides an analytical method which is very simple and effective.

Now, let us consider what makes the real-time planning of a humanoid robot's gait difficult. Fig.1 shows a situation where a humanoid robot is changing the walking direction to the left. It would be desirable if the humanoid robot can change the position of the step as soon as possible. However, as shown in the figure, when a humanoid robot changes the position of the next step, the humanoid robot has to move quickly to compensate for the difference between the pre-planned position of the next step and the newly-planned one. If the difference becomes very significant, it will become difficult for a humanoid robot to keep balance. Now, for the purpose of realizing the adaptive and robust gait of a humanoid robot, we have to examine how quick the change of gait can be realized.

Secondly, in most of the previous research on gait generation, the gait has been calculated numerically. As for the numerical method, since it becomes difficult to

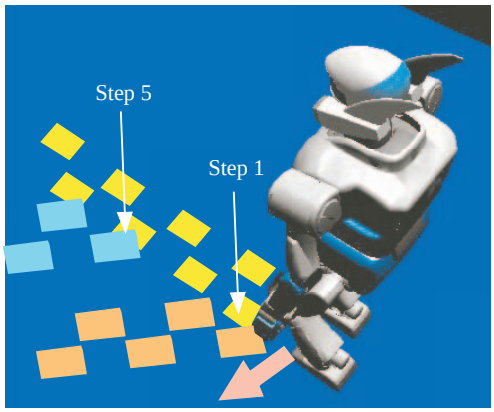


Fig. 1. Explanation of Gait Planning

calculate the gait within one sampling-time, the quick change of gait is difficult.

Thirdly, for changing the position of the step in real-time, let us consider newly calculating the trajectories of the robot's motion and connecting them to the current ones. Here, since the trajectories of the robot's motion are calculated by solving the two-point boundary value problem of the ordinary differential equation, it becomes impossible to specify the initial and the final velocities of the generalized coordinates. Thus, when connecting trajectories, it becomes difficult to ensure the continuity between two trajectories.

To cope with such problems in the real-time gait planning, we propose an analytical-solution based framework. Since our method does not require the numerical iterations, the gait can be calculated very fast. Also, to ensure the continuity between two trajectories of robot's motion, we consider simultaneously planning both the COG(Center of Gravity) and the ZMP(Zero-Moment Point) trajectories in real-time. By using this simultaneous planning method, we show that the smooth real-time change of gait can be realized. We further propose two methods for connecting the newly calculated trajectories to the current ones, i.e., the real-time connection method and the quasi-real-time connection method. While the quick change of gait can be realized by using the real-time method, the significant change of the position of step is difficult. On the other hand, while the quick-change of gait is difficult by using the quasi-real-time method, the significant change of the position of step is expected to realize.

This paper is organized as follows: After discussing the previous works in Section 2, we show the overview of the analytical method of gait planning in Section 3. In Section 4, we discuss a method for real-time planning of the gait. In Section 5, we study the time for changing the gait. Lastly, in Section 6, we show simulation and experimental results.

2. Related Works

In most of the research on a biped robot, the walking pattern has been generated before the robot actually moves. Takanishi et al.[2] proposed a method for generating the trunk motion of a humanoid robot by transforming the trajectory of the ZMP[1] into the fourier series. Kajita et al.[5] realized the dynamic walking of a biped robot based on the linear inverted pendulum mode. Nagasaka et al.[3] proposed a method based on the optimal gradient method. Kagami et al.[4] generated the gait by descretizing the differential equation expressing the relationship between the robot's motion and the ZMP. Kurazume et al.[7] generated the gait based on the analytical solution of the differential equation.

Recently, some researches have been done on the real-time planning of a humanoid robot's gait. The famous one is the "i-walk" by Honda motors. Takenaka[10,11] realized the on-line planning of a humanoid robot's gait by connecting the unit gaits including two steps of a humanoid robot. They realized the smooth connection of the unit gaits by considering the additional dynamics of the inverted pendulum. Lim et al.[8] also considered combining the unit gait patterns generated offline. Nishiwaki et al.[9] proposed a method for modifying the gait pattern on-line. They used the ZMP trajectory with three steps, and the change of gait is realized by connecting the newly calculated trajectories to the current ones. Kajita et al.[6] proposed the gait planning by using the preview control of the ZMP.

On the other hand, our method follows the one proposed by Nishiwaki et al.[9]. Here, different from their method, the analytical solution of the differential equation is used where the solution can be calculated very fast. Also, the smooth connection of the trajectories can be realized by simultaneously calculating the ZMP and the COG trajectories.

3. Basic Idea of Gait Generation

Let us consider the humanoid robot walking on the flat floor. While we focus on the motion of a humanoid robot within the sagittal $(x-z)$ plane, the motion in the lateral $(y-z)$ plane can also be treated in the same fashion. Let us assume that the ZMP trajectory of a humanoid robot is given by a spline function. Let $\boldsymbol{p}_{zmp}^{(j)} = [x_{zmp}^{(j)}\ y_{zmp}^{(j)}\ z_{zmp}^{(j)}]^T$ be the trajectory of the ZMP belonging to the j-th segment of time. The trajectory of ZMP in the sagittal plane can be expressed as:

$$x_{zmp}^{(j)} = \sum_{i=0}^{n} a_i^{(j)} (t - t_{j-1})^i, \qquad t_{j-1} \le t \le t_j, \quad j = 1, \cdots, m \tag{1}$$

where $a_i^{(j)}$ $(i = 1, \cdots, m,\ j = 1, \cdots, m)$ are scalar coefficients. An example of the ZMP trajectory is shown in Fig.2. As shown in the figure, the robot begins to step at $t = t_0$. After finished stepping at $t = t_1$, the robot stands on two legs until $t = t_2$. Then, the robot begins to step at $t = t_2$.

Let $\boldsymbol{p}_G^{(j)} = [x_G^{(j)} \ y_G^{(j)} \ z_G^{(j)}]^T$ be the trajectory of the COG(Center of Gravity) corresponding to the ZMP trajectory belonging to the j-th segment of time. Also, let $\mathcal{L}^{(j)} = [\mathcal{L}_x^{(j)} \ \mathcal{L}_y^{(j)} \ \mathcal{L}_z^{(j)}]^T$ be the angular momentum of the robot about the COG. The relationship between the ZMP and the COG within the sagittal plane is expressed by the following ordinary differential equation:

$$x_{zmp}^{(j)} = \frac{-\dot{L}_y^{(j)} + M x_G^{(j)}(\ddot{\bar{z}}_G^{(j)} + g) - (\bar{z}_G^{(j)} - z_{zmp}^{(j)})\ddot{\bar{x}}_G^{(j)}}{M(\ddot{\bar{z}}_G^{(j)} + g)}. \tag{2}$$

Next, we assume that the motion of the COG in the vertical direction is small enough. By setting $\bar{x}_G^{(j)} = x_G^{(j)} + \Delta x_G^{(j)}$, eq.(2) can be split into the following two equations:

$$x_{zmp}^{(j)} = x_G^{(j)} - \frac{z_G^{(j)} - z_{zmp}^{(j)}}{g}\ddot{x}_G^{(j)} \tag{3}$$

$$\frac{\dot{\mathcal{L}}_y^{(j)}}{Mg} = \Delta x_G^{(j)} - \frac{z_G^{(j)} - z_{zmp}^{(j)}}{g}\Delta\ddot{x}_G^{(j)} \tag{4}$$

In this research, we assume that the effect of eq.(4) is small and can be compensated by the stabilizing controller installed in our experimental setup. However, it is possible to consider the effect of eq.(4), and the meheod is shown in the appendix. Substituting eq.(1) into eq.(3) and solving with respect to $x_G^{(j)}$, we obtain the analytical solution of the position of the COG as

$$x_G^{(j)} = V^{(j)}\cosh(T_c(t - t_{j-1})) + W^{(j)}\sinh(T_c(t - t_{j-1})) + \sum_{i=0}^{n} A_i^{(j)}(t - t_{j-1})^i, \qquad j = 1, \cdots, m \tag{5}$$

$$a_i^{(j)} = A_i^{(j)} - \frac{1}{T_c^2}(i+1)(i+2)A_{i+2}, \qquad i = 0, \cdots, n-2 \tag{6}$$

$$a_i^{(j)} = A_i^{(j)}, \qquad i = n-1, n \tag{7}$$

where $T_c = \sqrt{g/(z_G - z_{zmp})}$, and $V^{(j)}$ and $W^{(j)}$ $(j = 1, \cdots, m)$ denote the scalar coefficients.

Here, eq.(5) includes sinh and cosh in its homogenous part. To prevent the solution of eq.(5) to diverge as time goes by, $V^{(j)}$ and $W^{(j)}$ $(j = 1, \cdots, m)$ should be determined by using the two-point boundary value problem where both the initial and the final position of the COG are specified. On the other hand, since the initial velocity of the COG cannot be specified, it becomes difficult to ensure the continuity between two COG trajectories. For planning the gait in real-time, the newly calculated trajectories of the COG are connected to the current ones. Here, the continuity of the velocity is guaranteed by using the method shown in the next section.

There are $2m$ unknowns in eq.(5), i.e., $V^{(j)}$ and $W^{(j)}$ $(j = 1, \cdots, m)$. To determine these unknowns, we set the following $2m$ boundary conditions:

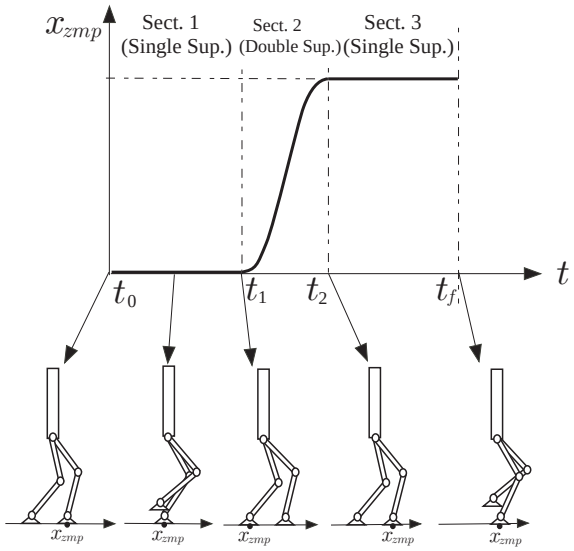


Fig. 2. ZMP Trajectory

Initial Condition (Position of COG)

$$x_G^{(1)}(t_0) = V^{(1)} + A_0^{(1)} \tag{8}$$

Connection of two Segments (Position/Velocity of COG)

$$j = 1, \cdots, m-1$$

$$V^{(j)} \cosh(T_c(t_j - t_{j-1})) + W^{(j)} \sinh(T_c(t_j - t_{j-1})) + \sum_{i=0}^{n} A_i^{(j)} (t_j - t_{j-1})^i = V^{(j+1)} + A_0^{(j+1)} \tag{9}$$

$$V^{(j)} T_c \sinh(T_c(t_j - t_{j-1})) + W^{(j)} T_c \cosh(T_c(t_j - t_{j-1})) + \sum_{i=1}^{n} i A_i^{(j)} (t_j - t_{j-1})^{i-1} = W^{(j+1)} T_c + A_1^{(j+1)} \tag{10}$$

Terminal Condition (Position of COG)

$$\bar{x}_G^{(m)}(t_f) = \tilde{V}^{(m)} \cosh(T_c(t_f - t_{m-1})) + \tilde{W}^{(m)} \sinh(T_c(t_f - t_{m-1})) + \sum_{i=0}^{n} \tilde{A}_i^{(m)} (t_f - t_{m-1})^i \tag{11}$$

By using eqs.(8), (9), (10), and (11), $2m$ unknowns $V^{(j)}$, $W^{(j)}$ $(j = 1, \cdots, m)$ can be determined as

$$\boldsymbol{y} = \boldsymbol{Z}^{-1} \boldsymbol{w}, \tag{12}$$

where

$$\boldsymbol{y} = [V^{(1)}\ W^{(1)}\ \cdots\ V^{(m)}\ W^{(m)}]^T,$$

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_0 & \mathbf{0} & \cdots & & \mathbf{0} \\ \boldsymbol{Z}_1 & \mathbf{0} & \cdots & & \mathbf{0} \\ \mathbf{0} & \ddots & & & \vdots \\ \vdots & & \boldsymbol{Z}_j & & \\ & & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & & \mathbf{0} & \boldsymbol{Z}_{m-1} \\ \mathbf{0} & \cdots & & \mathbf{0} & \boldsymbol{z}_{m-1} \end{bmatrix},$$

$$\boldsymbol{z}_0 = [1\ 0\ 0\ 0],$$

$$\boldsymbol{Z}_j = \begin{bmatrix} \cosh(T_c(t_j - t_{j-1})) & \sinh(T_c(t_j - t_{j-1})) & -1 & 0 \\ T_c \sinh(T_c(t_j - t_{j-1})) & T_c \cosh(T_c(t_j - t_{j-1})) & 0 & -T_c \end{bmatrix}$$

$$\boldsymbol{z}_{m-1} = [0\ 0\ \cosh(T_c(t_f - t_{m-1}))\ \sinh(T_c(t_f - t_{m-1}))]$$

$$\boldsymbol{w} = [\bar{x}_G^{(1)}(t_0) - A_0^{(1)}\ \cdots\ A_0^{(j+1)} - \sum_{i=0}^{n} A_i(t_j - t_{j-1})^i$$

$$A_1^{(j+1)} - \sum_{i=0}^{n} iA_i(t_j - t_{j-1})^{i-1}\ \cdots \bar{x}_G^{(m)}(t_f) - \sum_{i=0}^{n} A_i^{(m)}(t_f - t_{m-1})^i]^T,$$

where we can confirm that the matrix $\boldsymbol{Z}$ is invertible by observing the position of $\boldsymbol{Z}_j$ $(j = 1, \cdots, m-1)$ included in $\boldsymbol{Z}$ and $\mathrm{rank}\boldsymbol{Z}_j = 2$ $(j = 1, \cdots, m-1)$.

Substituting $V^{(j)}$ and $W^{(j)}$ $(j = 1, \cdots, m)$ obtained by eq.(12) into eq.(5), we obtain the trajectory of the COG for a given ZMP trajectory satisfying the initial and terminal condition for the position of the COG.

4. Real-Time Gait Planning

4.1. *Trajectory Connection*

We will explain the method for planning the gait in real-time by using Fig.3. Fig.3(a) and (b) show the trajectories of the ZMP and the COG, respectively, within the sagittal plane. Let us consider changing the step length from l_2 to l_{new} between $t = t_2$ and t_f. The change of the step length is realized by connecting the newly calculated trajectories of both the ZMP and the COG to the current ones. While we can consider several methods for connecting two trajectories, we show two of them in the following. In both cases, we assume that the new trajectories are connected to the current ones just when the double-support phase begins.

4.1.1. *Real-Time Trajectory Connection*

As for the real-time trajectory connection, the new trajectories are connected to the current ones at $t = t_2$ as shown in Fig.3(c). It means that the trajectories are connected just before the step length changes to the new one. By using this method, the new step length has to be determined before $t = t_1$ since the robot begins to step at $t = t_1$. Here, if the difference between l_2 and l_{new} is large, the robot has to move quickly to compensate for the difference. Thus, this method is considered to be effective when the time for the double-support phase is long or when the difference between l_2 and l_{new} is small as shown in the next section.

4.1.2. *Quasi-Real-Time Trajectory Connection*

As for the quasi-real-time trajectory connection, the new trajectories are connected to the current ones at $t = t_0$ as shown in Fig.3(d). By using this method, the new step length has to be determined before $t = t_0$. In the new trajectory, the length of the first step is same as that of the current one while the step length changes to the new one in the second step. While the new step length has to be determined earlier than that of the real-time method, we can expect that the motion of the robot compensating the difference of the step length is not so quick since the length of the first step is same. Thus, we can also expect that this method is effective when the time for the double-support phase is short or when the difference between l_2 and l_{new} is large.

4.2. *Simultaneous COG and ZMP Planning*

In this subsection, we consider the method for smoothly connecting the new trajectories to the current ones.

As shown in the previous section, since the differential equation expressing the relationship between the ZMP and the COG has been solved by using the two-point boundary value problem, the initial velocity of the COG cannot be taken into consideration. Thus, if we simply connect the new trajectories to the current ones, the discontinuity of the velocity of the COG occurs. In this research, to ensure the continuity of the velocity of the COG, the parameters defining the ZMP belonging to the first segment of time is also set to be unknowns and is calculated smoothly connecting to the current one. By setting $A_i^{(1)}$ $(i = 0, \cdots, n)$ of the new trajectories as unknown constants, the following boundary conditions in addition to eqs.(8), (9), (10), and (11) are considered:

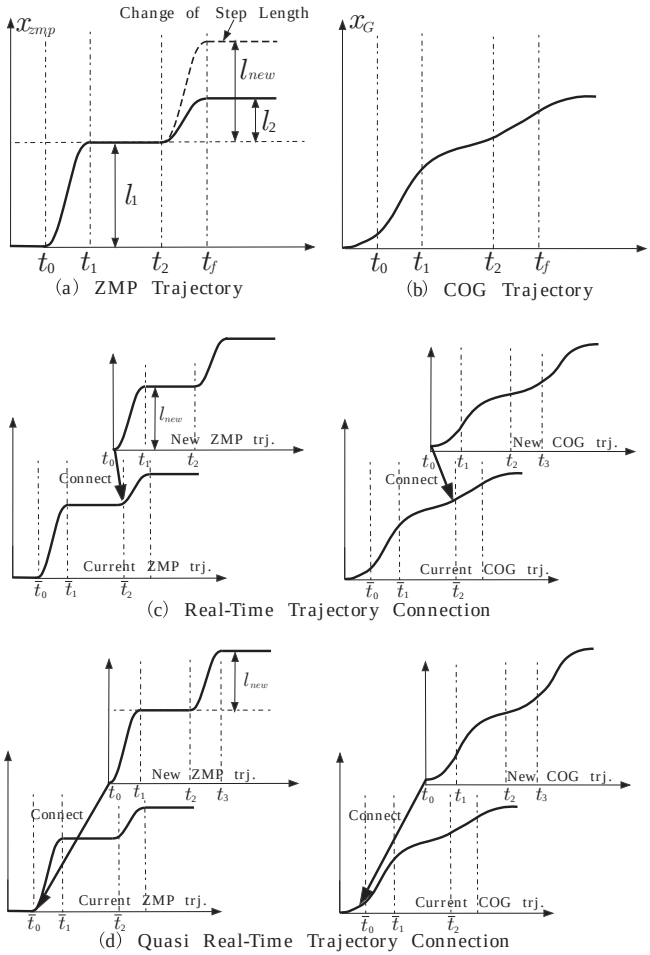


Fig. 3. Overview of Trajectory Connection

Initial Condition (Position of ZMP)

$$x_{zmp}^{(1)}(t_0) = a_0^{(1)} \tag{13}$$

Terminal Condition for the 1st Section (Position of ZMP)

$$x_{zmp}^{(2)}(t_1) = \sum_{i=0}^{n} a_i^{(1)}(t_1 - t_0)^i \tag{14}$$

Initial Condition (Velocity of COG)

$$\dot{x}_G^{(1)}(t_0) = W^{(1)}T_c + A_1^{(1)} \tag{15}$$

When $n = 2$ for the first section, the $2m + n + 1$ unknowns $(A_0^{(1)}, \cdots, A_n^{(1)}, V^{(1)}, W^{(1)}, \cdots, V^{(m)}, W^{(m)})$ can be determined by using eqs.(8), (9), (10), (11),

(13), (14), and (15) as follows:

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{Z}}^{-1}\tilde{\boldsymbol{w}}, \tag{16}$$

where

$$\tilde{\boldsymbol{y}} = [A_0^{(1)}\ A_1^{(1)}\ A_2^{(1)}\ V^{(1)}\ W^{(1)}\ \cdots\ V^{(m)}\ W^{(m)}]^T$$

$$\tilde{\boldsymbol{Z}} = \begin{bmatrix} \boldsymbol{Z}_0 & \mathbf{0} & \mathbf{0} & \cdots & & \mathbf{0} \\ \boldsymbol{Z}_{11} & \boldsymbol{Z}_{12} & \mathbf{0} & \cdots & & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{Z}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ & & \mathbf{0} & \ddots & & \vdots \\ \vdots & \vdots & & \boldsymbol{Z}_j & & \\ & & & & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & & \mathbf{0} & \boldsymbol{Z}_{m-1} \\ \mathbf{0} & \mathbf{0} & \cdots & & \mathbf{0} & \boldsymbol{z}_{m-1} \end{bmatrix}$$

$$\boldsymbol{Z}_0 = \begin{bmatrix} 1 & 0 & -2/T_c^2 \\ 1 & t_1 - t_0 & (t_1 - t_0)^2 - 2/T_c^2 \end{bmatrix}$$

$$\boldsymbol{Z}_{11} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & (t_1 - t_0) & (t_1 - t_0)^2 \\ 0 & 1 & 2(t_1 - t_0) \end{bmatrix}$$

$$\boldsymbol{Z}_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & T_c & 0 & 0 \\ \cosh(T_c(t_1 - t_0)) & \sinh(T_c(t_1 - t_0)) & -1 & 0 \\ T_c \sinh(T_c(t_1 - t_0)) & T_c \cosh(T_c(t_1 - t_0)) & 0 & -T_c \end{bmatrix}$$

$$\tilde{\boldsymbol{w}} = [x_{zmp}^{(1)}(t_0)\ \ x_{zmp}^{(2)}(t_1)\ \ x_G^{(1)}(t_0)\ \ \dot{x}_G^{(1)}(t_0)$$
$$A_0^{(2)}\ \ A_1^{(2)}\ \ \cdots A_0^{(j+1)} - \sum_{i=0}^{n} A_i^{(j)}(t_j - t_{j-1})^i$$
$$A_1^{(j+1)} - \sum_{i=1}^{n} i A_i^{(j)}(t_j - t_{j-1})^{i-1} \cdots x_G^{(m)}(t_f) - \sum_{i=0}^{n} A_i^{(m)}(t_f - t_{m-1})^i]^T$$

Here, we numerically confirmed that the matrix $\tilde{\boldsymbol{Z}}$ in eq.(16) is invertible.

4.3. *Discussion*

We first note that, by using the proposed method, the COG trajectory can be calculated very fast since the analytical solution is used. Only the time-consuming calculation is the inverse of the matrix $\tilde{\boldsymbol{Z}}$. Assuming the ZMP trajectory for 3 steps

and $m = 9$, the size of the matrix $\tilde{\boldsymbol{Z}}$ becomes 20×20, we confirmed that the calculation time is short enough. Also, we note that, since the element of $\tilde{\boldsymbol{Z}}$ does not include the step length, the inverse of the matrix $\tilde{\boldsymbol{Z}}$ can be calculated off-line. Also, in the simulation and the experiment, in addition to the initial condition for the position of the ZMP (eq.(13)) and the terminal condition for the position of the ZMP in the first section (eq.(14)), we also took the initial condition for the velocity of the ZMP and the terminal condition for the velocity of the ZMP in the first segment of time into consideration.

We also note that, while we explained the proposed method by using the motion within the sagittal plane, we can easily extend the method to the 3D cases. As for the 3D motion of the robot, we can find one of the features of our proposed method. In our method, the ZMP trajectory of the first section is calculated along with the COG trajectory. When the robot is tracking the trajectory of the first section, the robot is in the double support phase. Thus, in our method, the ZMP trajectory in the double support phase is deviated from a line including two positions of steps. When the position of the step is changed, the motion of the robot is compensated since the ZMP trajectory is deviated as shown in the next section.

5. Simulation

We performed simulation by using the OpenHRP[12,13]. We calculated the ZMP and the COG trajectories by using the proposed method. These trajectories are transformed to the trajectories of each joint by using the Resolved Momentum Control[14], and the humanoid robot is driven by the joint angle command. As a model of a humanoid robot, we used the HRP-2 [15] whose height and weight are $h = 1.57$[m] and $m = 57$[kg], respectively. In the simulation, the humanoid robot walks for 10 steps. At the beginning of the double support phase of the first eight steps, the new trajectories are connected. Among these steps, the step length changes in the fifth step.

First, we show the simulation result of the real-time trajectory connection method. The results of simulation are shown in Figs. 4, 5, 6, and 7. In the simulation shown in Fig.4, the time for the single and the double support phases are set as $T_s = 0.8$[sec] and $T_{dbl} = 0.2$[sec], respectively. In the fifth step, the step length changes from $l = 0.05$[m] to $l_{new} = 0.15$[m]. In the fifth step, the ZMP trajectory deviates from a line(Fig.4(e)). This is because, since the step length becomes larger, the COG has to be accelerated. To accelerate the COG, the ZMP shifts backward. Since the amount of deviation is large, the robot falls down as shown in Fig.5.

On the other hand, in the simulation shown in Fig.6, the step length changes from $l = 0.11$[m] to $l_{new} = 0.12$[m]. Since the amount of change of the step length is small, the deviation of the ZMP position is also small.

In the simulation shown in Fig.7, the time for the double support phase is $T_{dbl} = 1.5$[sec]. Since the time for the double support phase is long, the deviation of the ZMP position in the fifth step is small. Thus, we can see that the real-time

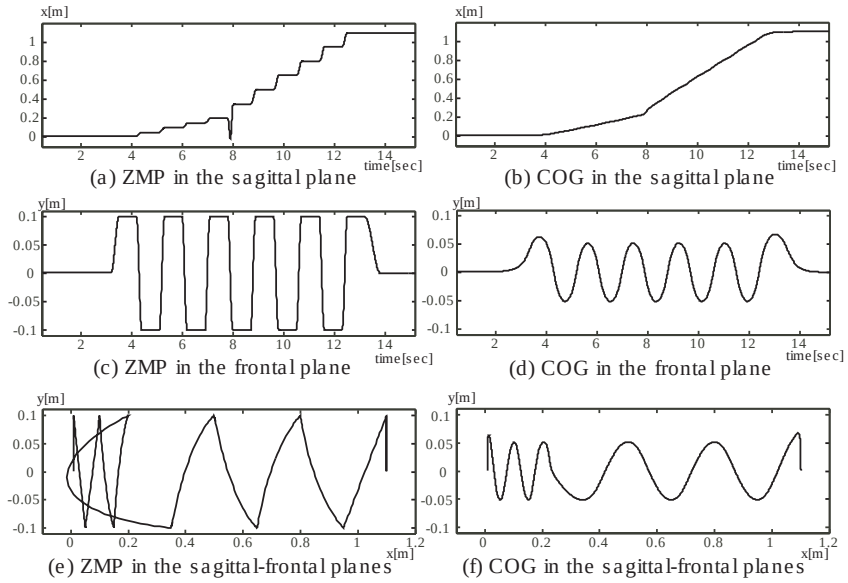


Fig. 4. Real-time trajectory connection where the step length of the 5th step changes from 0.05[m] to 0.15[m]($T_s = 0.8$[sec], $T_{dbl} = 0.2$[sec])

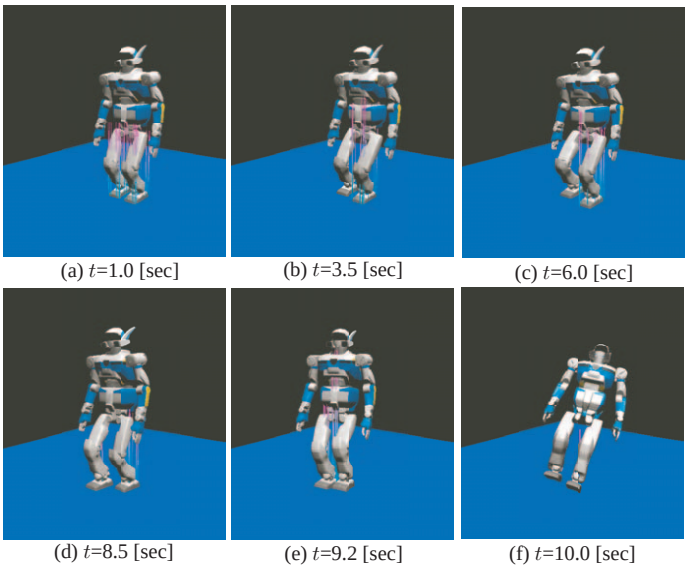


Fig. 5. Snapshot of the Real-Time Trajectory Connection

trajectory connection method is effective when the amount of change of the step position is small or when the time for the double support phase is long. We note

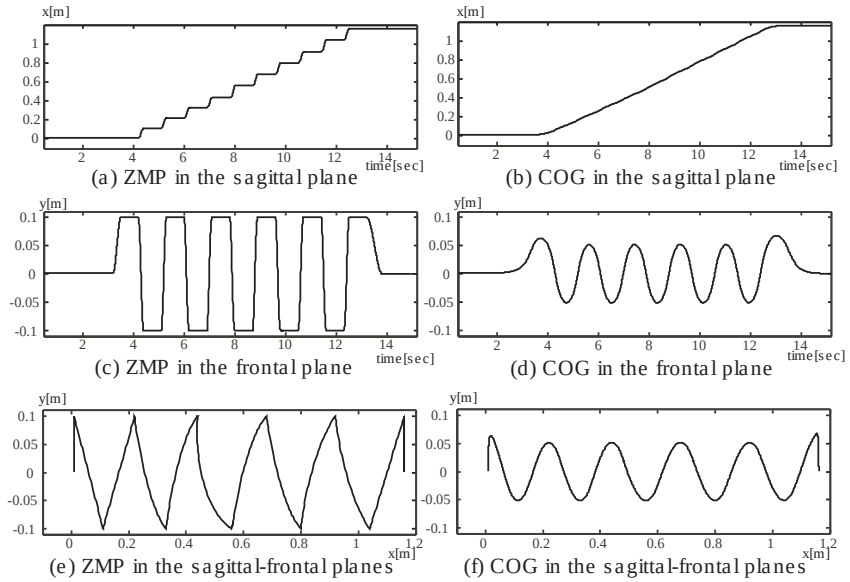


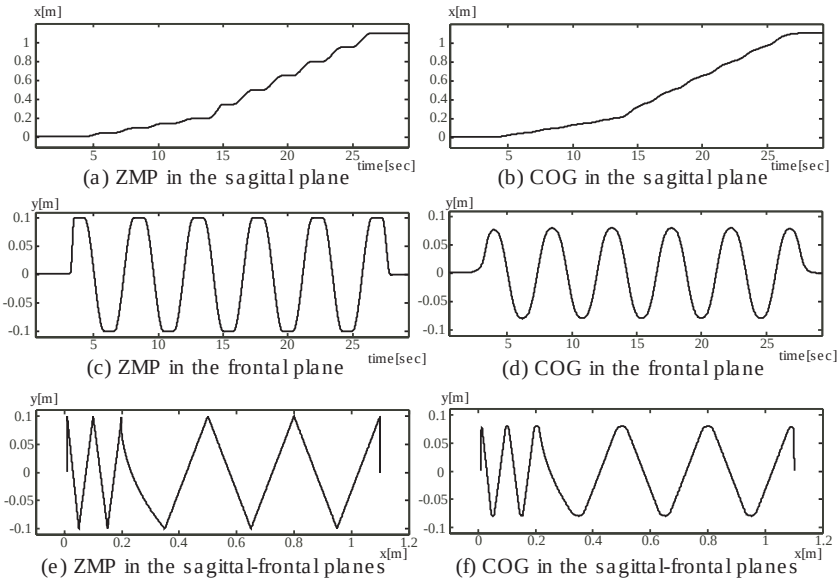
Fig. 6. Real-time trajectory connection where the step length of the 5th step changes from 0.11[m] to 0.12[m]($T_s = 0.8$[sec], $T_{dbl} = 0.2$[sec])

that, in our previous paper, we used the real-time trajectory connection method and the pushing manipulation of a large object was realized stably regardless of the mass of the object[16].

The simulation result of the quasi-real-time trajectory connection method is shown in Figs.8 and 9. While the step length changes from $l = 0.05$[m] to $l_{new} = 0.15$[m] in the fifth step, the robot can keep balance as shown in Fig.9. As for the quasi-real-time trajectory connection method, the new step length has to be determined earlier than that of the real-time trajectory connection method. The difference of time is same as the time for the double support phase. Thus the quasi-real-time method is considered to be effective when the time for the double support phase is short.

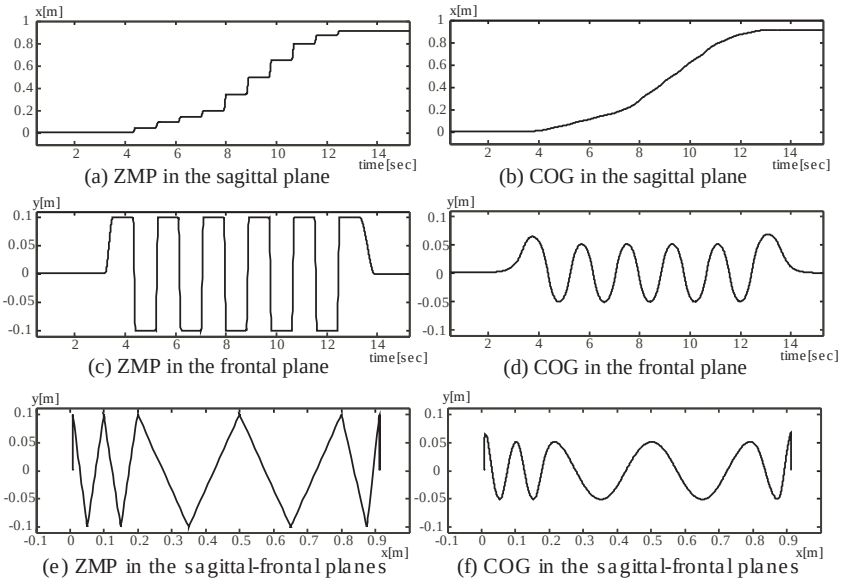
6. Experiment

Then we performed experiment using the humanoid robot HRP-2[15]. In the experiment, the operator holds a hand of the HRP-2. The force/torque sensor is attached at the wrist of the HRP-2, and the position of the step is changed depending on the information from the force/torque sensor. The quasi-real-time trajectory connection method was used. The time for the single and double support phases are set as $T_s = 0.7$[sec] and $T_{dbl} = 0.1$[sec], respectively. The experimental result is shown in Fig.10. It is just like navigating a blind person by holding his/her hand.



(a) ZMP in the sagittal plane (b) COG in the sagittal plane
(c) ZMP in the frontal plane (d) COG in the frontal plane
(e) ZMP in the sagittal-frontal planes (f) COG in the sagittal-frontal planes

Fig. 7. Real-time trajectory connection where the step length of the 5th step changes from 0.05[m] to 0.15[m]($T_s = 0.8$[sec], $T_{dbl} = 1.5$[sec])



(a) ZMP in the sagittal plane (b) COG in the sagittal plane
(c) ZMP in the frontal plane (d) COG in the frontal plane
(e) ZMP in the sagittal-frontal planes (f) COG in the sagittal-frontal planes

Fig. 8. Quasi-real-time trajectory connection where the step length of the 5th step changes from 0.05[m] to 0.15[m]($T_s = 0.7$[sec], $T_{dbl} = 0.1$[sec])

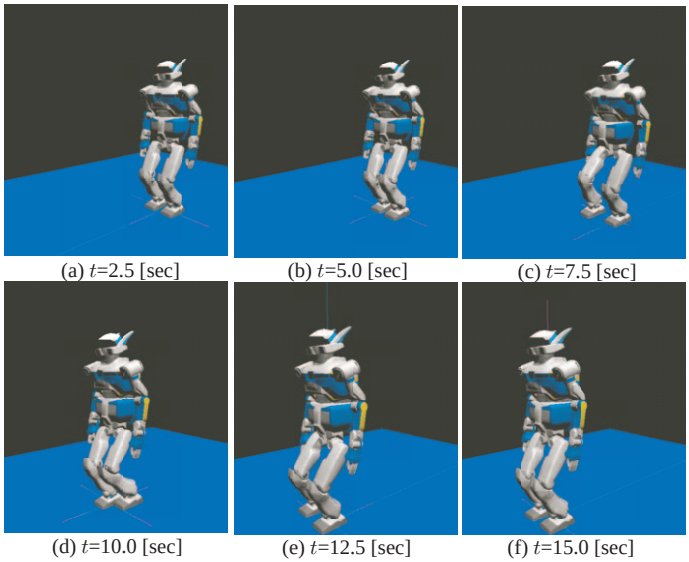


Fig. 9. Snapshot of the Quasi-Real-Time Trajectory Connection

7. Conclusion

In this paper, we proposed a new style of real-time gait planning by a humanoid robot. In our proposed method, the ZMP and COG trajectories are simultaneously planned. Also, for planning the gait in real-time, two methods for connection of trajectories are considered. The effectiveness of the proposed method is confirmed by simulation and experimental results.

As a future research, we plan to include the visual feedback for the motion of a humanoid robot. Finally, we would like to express our sincere gratitude for Dr.Fumio Kanehiro, Mr. Kiyoshi Fujiwara, Mr. Hajime Saito, Dr. Mitsuharu Morisawa, and Dr. Kazuhito Yokoi who are the humanoid robotics researchers in AIST for their valuable discussions.

References

1. M. Vukobratovic and D. Juricic: "*Contribution to the Synthesis of Biped Gait*", IEEE Trans. on Bio-Med. Eng., vol. BME-16, no. 1, pp. 1-6, 1969.
2. A. Takanishi, M. Ishida, Y. Yamazaki, and I. Kato: "*The Realization of Dynamic Walking by the Biped Walking Robot*", Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 459-466, 1985.
3. K.Nagasaka, H.Inoue, and M.Inaba: "*Dynamic Walking PatternGeneration for a Humanoid Robot Based on Optimal Gradient Method*", Proc. of IEEE Int. Conf. on Systems, Man., and Cybernetics, pp. VI-908-913, 1999.
4. 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*", Proc. of the 2000 IEEE-RAS Int. Conf. on Humanoid, 2000.
5. S. Kajita, T. Yamaura, and A. Kobayashi: "*Dynamic Walking Control of a Biped Robot*

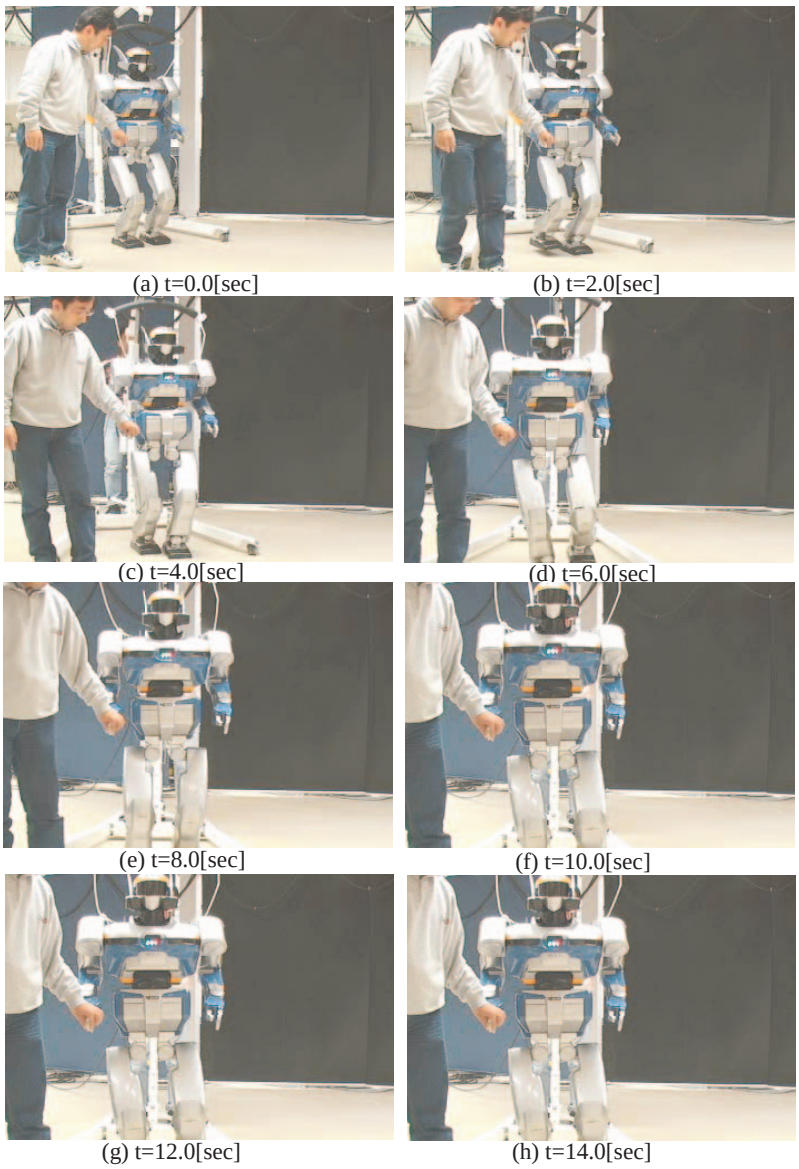


Fig. 10. Experimental Result

Along a Potential Energy Conserving Orbit", IEEE Trans. Robot. and Automat., vol. 8, no. 4, pp. 431-438, 1992.

6. S.Kajita et al.: "*Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point*", Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1620-1626, 2003.
7. R. Kurazume, T. Hasegawa, and K. Yoneda: "*The Sway Compensation Trajectory for a Biped Robot*", 'Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 925-931,

2003.
8. H.-O. Lim, Y. Kaneshima, and A. Takanishi: "*Online Walking Pattern Generation for Biped Humanoid Robot with Trunk*", 'Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 3111-3116, 2002.
9. K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue: "*Online Generation of Humanoid Walking Motion based on a Fast Generation Method of Motion Pattern that Follows Desired ZMP*", 'Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2684-2689, 2002.
10. T. Takenaka: "*Gait Generation Device for a Legged Mobile Robot*", Japanese Patent Application H10-217161, 1998.
11. T. Takenaka: "*Motion Generation Device for a Legged Mobile Robot*", Japanese Patent Application 2002-217161, 2002.
12. F. Kanehiro et al.: "*Virtual humanoid robot platform to develop controllers of real humanoid robots without porting*", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2001.
13. H. Hirukawa et al.: "*OpenHRP: Open Architecture Humanoid Robot Platform*", Proc. of Int. Symp. on Robotics Research, 2001.
14. S.Kajita et al.: "*Resolved Momentum Control: Humanoid Motion Planning based on the Linear and Angular Momentum*", Proc. of IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 1644-1650, 2003.
15. K. Kaneko et al.: "*The Humanoid Robot HRP2*", Proc. of IEEE Int. Conf. on Robotics and Automation, 2004.
16. K. Harada et al.: "*Real-Time Planning of Humanoid Robot's Gait for Force Controlled Manipulation*", Proc. of IEEE Int. Conf. on Robotics and Automation, 2004.

Appendix A. Compensation of Angular Momentum

In Section 3, we split eq.(2) into eqs. (3) and (4), and gait planning was performed based on eq.(3). Here, the effect of eq.(4) is assumed to be small and is omit. In this section, we consider the effect of eq.(4).

In most cases, since the left-hand side of eq.(4) is complex, we cannot solve analytically for $\Delta x_G^{(j)}$ in its original form. Thus, we consider expressing the left-hand side of eq.(4) by the fourier series[2]:

$$\frac{\dot{\mathcal{L}}_y^{(j)}}{Mg} = \frac{c_0^{(j)}}{2} + \sum_{i=1}^{\infty} c_i^{(j)} \cos\left(\frac{2\pi i(t-t_{j-1})}{t_j - t_{j-1}}\right) + \sum_{i=1}^{\infty} d_i^{(j)} \sin\left(\frac{2\pi i(t-t_{j-1})}{t_j - t_{j-1}}\right), \tag{A.1}$$

where

$$c_i^{(j)} = \frac{2}{T}\int_{t_{j-1}}^{t_j} \frac{\dot{\mathcal{L}}_y^{(j)}}{Mg} \cos\left(\frac{2\pi i(t-t_{j-1})}{t_j - t_{j-1}}\right) dt, \tag{A.2}$$

$$d_i^{(j)} = \frac{2}{T}\int_{t_{j-1}}^{t_j} \frac{\dot{\mathcal{L}}_y^{(j)}}{Mg} \sin\left(\frac{2\pi i(t-t_{j-1})}{t_j - t_{j-1}}\right) dt. \tag{A.3}$$

Substituting eq.(A.1) into eq.(4) and solving with respect to $\Delta x_G^{(j)}$, we obtain

$$\begin{aligned}\Delta x_G^{(j)} = \frac{C_0^{(j)}}{2} &+ \sum_{i=1}^{\infty} C_i^{(j)} \cos\left(\frac{2\pi i(t-t_{j-1})}{t_j - t_{j-1}}\right) \\ &+ \sum_{i=1}^{\infty} D_i^{(j)} \sin\left(\frac{2\pi i(t-t_{j-1})}{t_j - t_{j-1}}\right) \\ &+ \Delta V^{(j)} \cosh(T_c(t-t_{j-1})) + \Delta W^{(j)} \sinh(T_c(t-t_{j-1})), \end{aligned} \tag{A.4}$$

where

$$C_i = c_i \Big/ \left(1 + \left(\frac{2\pi i}{T_c(t_j - t_{j-1})}\right)^2\right), \tag{A.5}$$

$$D_i = d_i \Big/ \left(1 + \left(\frac{2\pi i}{T_c(t_j - t_{j-1})}\right)^2\right). \tag{A.6}$$

While eq.(A.4) includes infinite series, we consider approximating them by cutting at the certain order. For example, approximating them by the n-th order series, we can determine the unknown parameters $\Delta V^{(j)}$, $\Delta V^{(j)}$, $C_i^{(j)}$ $(i = 0, \cdots, n)$, $D_i^{(j)}$ $(i = 1, \cdots, n)$ by using the method shown in Section 3.

The result of simulation is shown in Fig.11. In this simulation, the humanoid robot HRP2 walks for three steps without going forward. The ZMP trajectory within the lateral plane is shown in Fig.11(a),(b), and the error between the desired and the actual ZMP trajectories is shown in Fig.11(c),(d). Although we roughly approximate by setting $n = 3$, the error of the ZMP trajectory is small. This is because the error itself was not large before it is compensated.

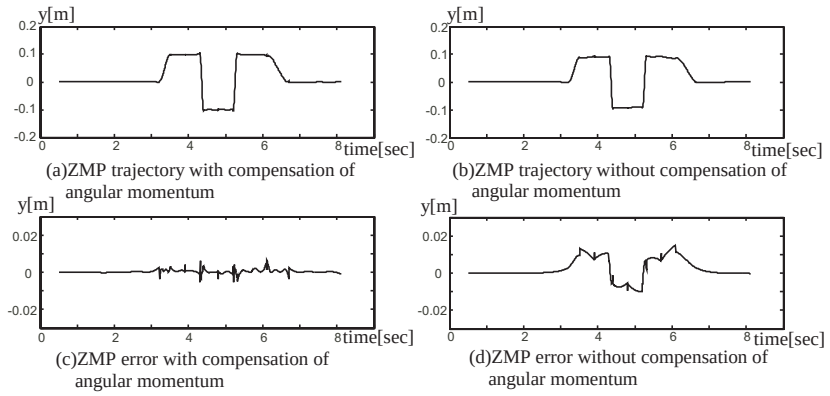


Fig. 11. Simulation Result of Angular Momentum Compensation