AN ANALYTICAL METHOD FOR REAL-TIME GAIT PLANNING FOR HUMANOID ROBOTS

KENSUKE HARADA*, SHUUJI KAJITA, KENJI KANEKO and HIROHISA HIRUKAWA
Humanoid Research Group, Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan
*kensuke.harada@aist.go.jp

Received 25 May 2005

This paper studies 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), a 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 a 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.

Keywords: Humanoid robot; biped gait; ZMP; real-time; analytical solution.

1. Introduction

The ultimate goal of research on a biped walking robot is to realize an 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 them. Since the walking motion of a biped robot has been realized by replaying pre-planned trajectories, it was difficult to change the position of the step in real-time. On the other hand, while there has been some research on real-time gait planning of a biped walking robot, this research provides an analytical method which is very simple and effective.

Let us consider what makes the real-time planning of a humanoid robot’s gait difficult. Figure 1 shows a situation where a humanoid robot is changing the walking direction to the left. It would be desirable if the humanoid robot could 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, it 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 robot to maintain its balance. Now, for the purpose of realizing the adaptive and robust gait of a humanoid robot, we have to examine how quickly the change of gait can be realized.

Secondly, in most of the previous research on gait generation, the gait has been calculated numerically. In the numerical method, since it becomes difficult to calculate the gait within one sampling period, a quick change of gait is difficult.

Thirdly, to change 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 an ordinary differential equation, it becomes impossible to specify the initial and 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 real-time gait planning, we propose an analytical solution based framework and show that our approach is very effective when planning gait in real-time. Since our method does not require numerical iterations, the gait can be calculated very quickly. Also, to ensure the continuity between two trajectories of the 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 a smooth real-time change of gait can be realized. We further propose two methods for connecting the newly calculated trajectories to the current ones: the real-time connection method and the quasi-real-time connection method. While a quick change of gait can be realized by using the real-time method, a significant change of the position of the 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 the step is expected to be realized.
This paper is organized as follows. After discussing previous research in Sec. 2, we give an overview of the analytical method of gait planning in Sec. 3. In Sec. 4, we discuss a method for real-time planning of the gait. In Sec. 5, we study the time for changing the gait. Lastly, in Sec. 6, we show our simulation and experimental results.

2. Related Works

In most of the research on biped robots, walking patterns have been generated before the robot actually moves. Takanishi et al. proposed a method for generating the trunk motion of a humanoid robot by transforming the trajectory of the ZMP into a Fourier series. Kajita et al. realized the dynamic walking of a biped robot based on the linear inverted pendulum mode. Nagasaka et al. proposed a method based on the optimal gradient method. Kagami et al. generated the gait by descretizing the differential equation expressing the relationship between the robot’s motion and the ZMP. Kurazume et al. generated the gait based on the analytical solution of the differential equation.

Recently, some research has been done on the real-time planning of a humanoid robot’s gait. The famous one is the “i-walk” by Honda motors. Takenaka realized the online planning of a humanoid robot’s gait by connecting the unit gaits including two steps of the humanoid robot. They realized the smooth connection of the unit gaits by considering the additional dynamics of the inverted pendulum. Lim et al. also considered combining the unit gait patterns generated offline. Nishiwaki et al. proposed a method for modifying the gait pattern online. 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. proposed gait planning by using the preview control of the ZMP.

In contrast, we propose an analytical solution based real-time gait planning, where the solution can be calculated very quickly and the smooth connection of the trajectories can easily be realized by simultaneously calculating the ZMP and the COG trajectories.

3. Basic Idea of Gait Generation

Let us consider a humanoid robot walking on a flat surface. While we focus on the motion of a humanoid robot within the sagittal \((x - z)\) plane, the motion in the frontal \((y - z)\) 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 \(\mathbf{p}_{\text{zmp}}^{(j)} = [x_{\text{zmp}}^{(j)} \ y_{\text{zmp}}^{(j)} \ z_{\text{zmp}}^{(j)}]^T\) be the trajectory of the ZMP belonging to the \(j\)th segment of time. The trajectory of the ZMP in the sagittal plane can be expressed as

\[
x_{\text{zmp}}^{(j)} = \sum_{i=0}^{n} a_i^{(j)}(t - t_{j-1})^i, \quad t_{j-1} \leq t \leq t_j, \quad j = 1, \ldots, m,
\]
where $a_i^j$ ($i = 0, \ldots, n, j = 1, \ldots, m$) 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 finishing 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 $p_G^j = [\bar{x}_G^j \bar{y}_G^j \bar{z}_G^j]^T$ be the trajectory of the COG (Center of Gravity) corresponding to the ZMP trajectory belonging to the $j$th segment of time. Also, let $L_y^j = [L_x^j L_y^j L_z^j]^T$ 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:

\[
x_{zmp}^{(j)} = \frac{-\ddot{x}_G^{(j)} + M \ddot{x}_G^{(j)} (\bar{z}_G^{(j)} + g) - (\bar{z}_G^{(j)} - z_{zmp}^{(j)}) \bar{y}_G^{(j)}}{M(\bar{z}_G^{(j)} + g)},
\]

(2)

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

\[
x_{zmp}^{(j)} = x_G^{(j)} - \frac{z_G^{(j)} - z_{zmp}^{(j)} \bar{x}_G^{(j)}}{g},
\]

(3)

\[
\frac{L_y^{(j)}}{M g} = \Delta x_G^{(j)} - \frac{z_G^{(j)} - z_{zmp}^{(j)}}{g} \Delta \bar{x}_G^{(j)}.
\]

(4)

\(^a\)The extension to motion in the vertical direction is shown in Appendix B.
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) in our framework; the method is shown in Appendix A. 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

\[
x_G^{(j)} = V^{(j)} \cosh(T_c(t - t_{j-1})) + W^{(j)} \sinh(T_c(t - t_{j-1}))
\]

\[
+ \sum_{i=0}^{n} A_i^{(j)}(t - t_{j-1})^i, \quad j = 1, \ldots, m,
\]

where \( T_c = \sqrt{g/(2g - z_{mp})} \), and \( V^{(j)} \) and \( W^{(j)} \) (\( j = 1, \ldots, m \)) denote the scalar coefficients.

Here, Eq. (5) includes sinh and cosh in its homogenous part. To prevent the solution of Eq. (5) from diverging as time goes by, \( V^{(j)} \) and \( W^{(j)} \) (\( j = 1, \ldots, m \)) 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. To plan 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 \( W^{(j)} \) (\( j = 1, \ldots, m \)). To determine these unknowns, we set the following \( 2m \) boundary conditions:

**Initial condition (position of COG):**

\[
x_G^{(1)}(t_0) = V^{(1)} + A_0^{(1)}.
\]

**Connection of two segments (position/velocity of COG):**

\[
V^{(j)} \cosh(T_c(t_j - t_{j-1})) + W^{(j)} \sinh(T_c(t_j - t_{j-1})) + \sum_{i=0}^{n} A_i^{(j)}(t_j - t_{j-1})^i
\]

\[
= V^{(j+1)} + A_0^{(j+1)} ,
\]

\[
V^{(j)} T_c \sinh(T_c(t_j - t_{j-1})) + W^{(j)} T_c \cosh(T_c(t_j - t_{j-1})) + \sum_{i=1}^{n} i A_i^{(j)}(t_j - t_{j-1})^i - 1
\]

\[
= V^{(j+1)} T_c + A_1^{(j+1)} .
\]

**Terminal condition (position of COG):**

\[
x_G^{(m)}(t_f) = V^{(m)} \cosh(T_c(t_f - t_{m-1})) + W^{(m)} \sinh(T_c(t_f - t_{m-1})) + \sum_{i=0}^{n} A_i^{(m)}(t_f - t_{m-1})^i .
\]
By using Eqs. (8)–(11), 2m unknowns $V^{(j)}$, $W^{(j)}$ ($j = 1, \ldots, m$) can be determined as

$$y = Z^{-1}w,$$

(12)

where

$$y = \begin{bmatrix} V^{(1)} & W^{(1)} & \cdots & V^{(m)} & W^{(m)} \end{bmatrix}^T,$$

$$Z = \begin{bmatrix} z_0 & 0 & \cdots & 0 \\
Z_1 & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & Z_j & 0 \\
0 & \cdots & 0 & Z_{m-1} \\
0 & \cdots & 0 & z_{m-1} \end{bmatrix},$$

$$z_0 = [1 \ 0 \ 0 \ 0],$$

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

$$z_{m-1} = \begin{bmatrix} 0 & 0 & \cosh(T_c(t_f - t_{m-1})) & \sinh(T_c(t_f - t_{m-1})) \end{bmatrix},$$

$$w = \begin{bmatrix} \bar{x}_G^{(1)}(t_0) - A_0^{(1)} \cdots A_0^{(j+1)} - \sum_{i=0}^{n} A_i(t_j - t_{j-1})^i \\
A_1^{(j+1)} - \sum_{i=0}^{n} iA_i(t_j - t_{j-1})^{i-1} \cdots \bar{x}_G^{(m)}(t_f) - \sum_{i=0}^{n} A_i^{(m)}(t_f - t_{m-1})^i \end{bmatrix}^T,$$

where we can confirm that the matrix $Z$ is invertible by observing the position of $Z_j$ ($j = 1, \ldots, m - 1$) included in $Z$ and rank $Z_j = 2$ ($j = 1, \ldots, m - 1$).

Substituting $V^{(j)}$ and $W^{(j)}$ ($j = 1, \ldots, m$) obtained by Eq. (12) into Eq. (5), we obtain the trajectory of the COG for a given ZMP trajectory satisfying the initial and terminal conditions 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. Figures 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.
An Analytical Method for Real-Time Gait Planning for Humanoid Robots

Fig. 3. Overview of trajectory connection.
4.1.1. Real-time trajectory connection

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). This 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 \( l_{\text{new}} \) 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 \( l_{\text{new}} \) is small, as shown in the next section.

4.1.2. Quasi-real-time trajectory connection

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 the 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 change of step length does not become very quick since there is enough time until the step length changes. 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 \( l_{\text{new}} \) 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, a discontinuity in the velocity of the COG occurs. In this subsection, to ensure the continuity of the velocity of the COG, the parameters defining the ZMP belonging to the first segment of time are also set to be unknowns and are calculated to smoothly connect to the current one. By setting \( A_i^{(1)} \) \((i = 0, \ldots, n)\) of the new trajectories as unknown constants, the following boundary conditions in addition to Eqs. (8)–(11) are considered:

Initial condition (position of ZMP):
\[
x_{\text{zmp}}^{(1)}(t_0) = a_0^{(1)}.
\] (13)

Terminal condition for the first section (position of ZMP):
\[
x_{\text{zmp}}^{(2)}(t_1) = \sum_{i=0}^{n} a_i^{(1)}(t_1 - t_0)^i.
\] (14)
Initial condition (velocity of COG):
\[ \dot{x}^{(1)}_{G}(t_0) = W^{(1)} T_c + A_1^{(1)}. \]  
(15)

When \( n = 2 \) for the first section, the \( 2m+n+1 \) unknowns \((A_0^{(1)}, \ldots, A_n^{(1)}, V^{(1)}, W^{(1)}, \ldots, V^{(m)}, W^{(m)})\) can be determined by using Eqs. (8)–(11) and (13)–(15) as follows:
\[ \ddot{y} = \bar{Z}^{-1} \dot{w}, \]  
(16)

where
\[
\hat{y} = \begin{bmatrix} A_0^{(1)} & A_1^{(1)} & A_2^{(1)} & V^{(1)} & W^{(1)} & \ldots & V^{(m)} & W^{(m)} \end{bmatrix}^T,
\]
\[
\bar{Z} = \begin{bmatrix}
Z_0 & 0 & 0 & \cdots & 0 \\
Z_{11} & Z_{12} & 0 & \cdots & 0 \\
0 & 0 & Z_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & Z_j & 0 \\
0 & 0 & \cdots & 0 & Z_{m-1} \\
0 & 0 & \cdots & 0 & z_{m-1}
\end{bmatrix},
\]
\[
Z_0 = \begin{bmatrix} 1 & 0 & -2/T_c^2 \\
1 & t_1 - t_0 & (t_1 - t_0)^2 - 2/T_c^2 \end{bmatrix},
\]
\[
Z_{11} = \begin{bmatrix} 1 & 0 & 0 \\
0 & 1 & 0 \\
1 & (t_1 - t_0) & (t_1 - t_0)^2 \\
0 & 1 & 2(t_1 - t_0) \end{bmatrix},
\]
\[
Z_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\
0 & T_c & 0 & 0 \\
cosh(T_c(t_1 - t_0)) & \sinh(T_c(t_1 - t_0)) & -1 & 0 \\
T_c \sinh(T_c(t_1 - t_0)) & T_c \cosh(T_c(t_1 - t_0)) & 0 & -T_c \end{bmatrix},
\]
\[
\hat{w} = \begin{bmatrix} x^{(1)}_{zmp}(t_0) & x^{(2)}_{zmp}(t_1) & \dot{x}^{(1)}_{G}(t_0) & \dot{x}^{(1)}_{G}(t_0) \\
A_0^{(2)} & A_1^{(2)} & \ldots & A_0^{(j+1)} - \sum_{i=0}^{n} A_i^{(j)}(t_j - t_{j-1})^i \\
A_1^{(j+1)} - \sum_{i=1}^{n} iA_i^{(j)}(t_j - t_{j-1})^{i-1} \cdots x^{(m)}_{G}(t_f) - \sum_{i=0}^{n} A_i^{(m)}(t_f - t_{m-1})^i \end{bmatrix}^T.
\]

Here, we numerically confirm that the matrix \( \bar{Z} \) in Eq. (16) is invertible.
4.3. Discussion

We first note that, by using the proposed method, the COG trajectory can be calculated very quickly since the analytical solution is used. The only time-consuming calculation is the inverse of matrix $\tilde{Z}$. Assuming the ZMP trajectory for three steps and $m = 9$, the size of the matrix $\tilde{Z}$ becomes $20 \times 20$, and we confirm that the calculation time is about 0.3 ms using a 2.2 GHz PC. Also, we note that, since the element of $\tilde{Z}$ does not include the step length, the inverse of matrix $\tilde{Z}$ can be calculated offline. Also, in simulation and 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 goes to the edge of the support polygon. When the position of the step is changed, the motion of the robot is compensated since the ZMP trajectory deviates, as shown in the next section.

5. Simulation

We performed a simulation by using OpenHRP. We calculated the ZMP and COG trajectories by using the proposed method. These trajectories are transformed to the trajectories of each joint by using Resolved Momentum Control, and the humanoid robot is driven by the joint angle command. As a model of a humanoid robot, we used HRP-2 whose height and weight are $h = 1.57\,\text{m}$ and $m = 57\,\text{kg}$, respectively. In the simulation, the humanoid robot walks for ten 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 at the fifth step.

First, we show the simulation result of the real-time trajectory connection method. The results of simulation are shown in Figs. 4–7. In the simulation shown in Fig. 4, the time for the single and the double support phases are set as $T_s = 0.8\,\text{s}$ and $T_{dbl} = 0.2\,\text{s}$, respectively. In the fifth step, the step length changes from $l = 0.05\,\text{m}$ to $l_{\text{new}} = 0.15\,\text{m}$. 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 over as shown in Fig. 5.

On the other hand, in the simulation shown in Fig. 6, the step length changes from $l = 0.11\,\text{m}$ to $l_{\text{new}} = 0.12\,\text{m}$. 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 $T_{\text{dbl}} = 1.5\, \text{s}$. 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 trajectory connection method is effective when the amount of change of the step
Fig. 6. Real-time trajectory connection where the step length of the fifth step changes from 0.11 m to 0.12 m ($T_s = 0.8$ s, $T_{dub} = 0.2$ s).

position is small or when the time for the double support phase is long. We note 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.\textsuperscript{16}
Fig. 7. Real-time trajectory connection where the step length of the fifth step changes from 0.05 m to 0.15 m ($T_s = 0.8$ s, $T_{dbl} = 1.5$ s).

Fig. 8. Quasi-real-time trajectory connection where the step length of the fifth step changes from 0.05 m to 0.15 m ($T_s = 0.7$ s, $T_{dbl} = 0.1$ s).
The simulation result of the quasi-real-time trajectory connection method is shown in Figs. 8 and 9. While the step length changes from $l = 0.05\,\text{m}$ to $l_{\text{new}} = 0.15\,\text{m}$ in the fifth step, the robot can maintain its 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.
Fig. 10. Snapshot of the experiment.
6. Experiment

We performed experiments using the humanoid robot HRP-2. In the experiments, an operator holds a hand of the HRP-2. The force/torque sensor is attached at the wrist of 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 times for the single and double support phases are set as $T_s = 0.7$ s and $T_{dbl} = 0.1$ s, respectively. The experimental result is shown in Fig. 10. It is just like navigating a blind person by holding his/her hand. Figure 11 shows the ZMP and the COG trajectory during the experiment.

7. Conclusion

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

For future research, we plan to include visual feedback in the motion of the robot.

Acknowledgment

Finally, we would like to express our sincere gratitude to 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.

Appendix A. Compensation of Angular Momentum

In Sec. 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 omitted. 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 a Fourier series:

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

where

$$c_i^{(j)} = \frac{2}{T} \int_{t_{j-1}}^{t_j} \frac{\ddot{x}^{(j)}}{Mg} \cos \left( \frac{2\pi i (t - t_{j-1})}{t_j - t_{j-1}} \right) dt, \quad (A.2)$$

$$d_i^{(j)} = \frac{2}{T} \int_{t_{j-1}}^{t_j} \frac{\ddot{x}^{(j)}}{Mg} \sin \left( \frac{2\pi i (t - t_{j-1})}{t_j - t_{j-1}} \right) dt. \quad (A.3)$$

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

$$\Delta x_G^{(j)} = \frac{c_0^{(j)}}{2} + \sum_{i=1}^{\infty} C_i^{(j)} \cos \left( \frac{2\pi i (t - t_{j-1})}{t_j - t_{j-1}} \right) + \sum_{i=1}^{\infty} D_i^{(j)} \sin \left( \frac{2\pi i (t - t_{j-1})}{t_j - t_{j-1}} \right) + \Delta V^{(j)} \cosh(T_c(t - t_{j-1})) + \Delta W^{(j)} \sinh(T_c(t - t_{j-1})), \quad (A.4)$$

where

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

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

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

The result of simulation is shown in Fig. 12. In this simulation, the robot HRP-2 walks for three steps without going forward. The ZMP trajectory within the lateral plane is shown in Figs. 12(a) and (b) and the error between the desired and the actual ZMP trajectories is shown in Figs. 12(c) and (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 was compensated for.

**Appendix B. Vertical Motion of COG**

Next, we consider the case where the COG of the robot moves vertically. In our analytical solution based approach, the error in the ZMP position is caused when the COG moves vertically. First, we consider estimating the amount of error of the ZMP position. Let $\Delta x_{zmp}^{(j)}$ be the error in the ZMP position. The ZMP trajectory...
is expressed by the desired trajectory and the error:

\[ x_{\text{zmp}}^{(j)} = \sum_{i=0}^{n} a_i^{(j)}(t - t_{j-1})^i + \Delta x_{\text{zmp}}^{(j)}, \quad t_{j-1} \leq t \leq t_j, \quad j = 1, \ldots, m. \quad (B.1) \]

Removing the assumption of \( z^{(j)}_g = \text{const.} \), and substituting Eqs. (5) and (B.1) into Eq. (3), we obtain

\[ \Delta x_{\text{zmp}}^{(j)} = -f(t)^{(j)}(\dot{\omega}_c^2(t - t_{j-1})^2 + 2\ddot{\omega}_c\omega_c(t - t_{j-1})) \]
\[ - g(t)^{(j)}(\ddot{\omega}_c(t - t_{j-1})) + 2\ddot{\omega}_c \} \]
\[ f(t)^{(j)} = \{ V^{(j)} \cosh(\omega_c(t - t_{j-1})) \}
\]
\[ + W^{(j)} \sinh(\omega_c(t - t_{j-1})) \} / \omega_c^2, \quad (B.2) \]
\[ g(t)^{(j)} = \{ W^{(j)} \cosh(\omega_c(t - t_{j-1})) \}
\]
\[ + V^{(j)} \sinh(\omega_c(t - t_{j-1})) \} / \omega_c^2. \quad (B.4) \]

Now, we can analytically obtain the amount of error in the ZMP position when the COG moves vertically. Considering this equation and using the method proposed in Appendix A, we can obtain the gait pattern considering the motion of the COG in the vertical direction.

References


