ヒューマノイドロボットの実時間歩容計画における 解析的アプローチ

原 田 研 介* 梶 田 秀 司* 森 澤 光 晴* 金 広 文 男*
藤 原 清 司* 金 子 健 二* 比 留 川 博 久*

Analytical Approach on Real-time Gait Planning for a Humanoid Robot

Kensuke Harada*, Shuuji Kajita*, Mitsuharu Morisawa*, Fumio Kanehiro*,
Kiyoshi Fujiwara*, Kenji Kaneko* and Hirohisa Hirukawa*

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.

Key Words: Humanoid Robot, Gait Planning, ZMP, Analytical Solution

1. 緒 言

2足歩行ロボット研究の究極的な目標は、ヒトのように適応的でロバスタな歩行を実現することである。例えば、2足歩行ロボットが多数の障害物が置かれた環境を移動する場合を考える。このような環境において、ロボットは実時間で障害物を検知し、その障害物を避けるための動作軌道を計算しなくてはならない。従来、2足歩行ロボットの歩容を計画する研究では、ロボットの動作軌道はあらかじめ計画されており、実時間で歩容を変更することは困難であった。それに対して、近年2足歩行ロボットの歩容計画を実時間で実行する研究が行われている。

ここで、2足歩行ロボットの実時間歩容計画を困難にしているいくつかの問題点について考える。Fig. 1は、直進しているヒューマノイドロボットが進行方向を左に変更する場合を示している。適応的でロバスタな歩行を実現する観点より、ヒューマノイドロボットは出来るだけ早く計画された動作を変更できることが望ましい。しかしながら、次にステップする位置を変更しようとする、ロボットは現在の動作軌道を急に新しい軌道に変更しなくてはならない。特に、ステップする位置の変更量が多い場合など、このような急な動作は実現困難であり、ロボットが転倒する危険性が増す。このような問題に対しては、数

Fig. 1 Explanation of gait planning

歩先にステップする位置を変更し、ステップする位置を変更するまでの時間で徐々に動作軌道を変更することが有効ではないかと想像できる。ただし、ステップの変更はなるべく早いほうが望ましいと考えられる。

次に、従来2足歩行ロボットの歩容計画に関する研究では、ロボットの動作軌道は繰り返し計算により導出されてきた。しかし、繰り返し計算によって動作軌道を導出する場合、計算に時間がかかるため、実時間で軌道を変更することは困難である。

さらに、2足歩行ロボットの歩容計画は、与えられたZMP (Zero Moment Point) [1]の軌道を実現するようなロボットの動作軌道を計算することにより行われる。ここで、ZMPとロボット重心の関係は、倒立振子の支点と振子の重心の関係に置き換えると考えやすい。つまり、倒立振子において支点の軌道を与え、それを実現するような振子の重心の軌道を求める場合、

原稿受付 2004年8月24日

*産業技術総合研究所

*National Institute of Advanced Industrial Science and Technology

重心の初期姿勢のみを与えたのでは、時間が経つに従って倒立振子が倒れないことを保証することは困難である。つまり、質点の初期姿勢と終端の姿勢を与える必要がある。このような問題は2点境界値問題として知られている。しかしながら、微分方程式の2点境界値問題では初期速度や終端の速度を与えることが困難であるため、この解を実時間歩容計画に用いる場合、二つの軌道の間で速度の連続性を保証するのが困難である。

以上のような問題に対して、本論文では解析解をベースにした実時間歩容計画のフレームワークを提案する。実時間歩容計画における軌道間の速度の連続性を保つために、ロボットの動作軌道と同時にZMP軌道も計画することを考える。さらに、軌道を接続する方法として、実時間軌道接続と準実時間軌道接続の二つの方法を提案する。実時間軌道接続では素早い歩容の変更が可能である反面、ステップの位置を大きく変更することは困難である。それに対して、準実時間軌道接続では歩容の変更が遅れる反面、ステップの位置を大きく変更することが可能である。

本論文では、2章で従来研究を紹介した後、3章では解析解を求める基本的なアイデアを示す。さらに、4章において実時間歩容計画の手法を提案し、5、6章においてシミュレーションや実験結果を示す。

2. 関 連 研 究

従来、2足歩行ロボットの研究において、歩行パターンはロボットが実際に動作する以前にオフラインで計画されてきた。Takanishiら[2]は脚が運動することにより生じるZMPの誤差を胴体の運動により補償するために、ZMP[1]軌道をフーリエ級数展開する手法を提案した。Kajitaら[3]は線形倒立振子モードを提案し、それに基づく歩容計画手法を提案した。Nagasakaら[4]は最適勾配法を用いることにより、ヒューマノイドロボット全身の軌道を計算する手法を提案した。Kagamiら[5]はZMPに関する微分方程式を離散化することにより、動作計画を行った。Kurazumeら[6]はオフラインの歩容計画において解析解を用いたアプローチを提案している。ここで、ステップ間を直線で結び、その間のZMPの時間軌道も同時に導出している。

一方、近年ヒューマノイドロボットの実時間歩容計画に関する研究が行われている。竹中[7][8]は2歩から構成されるロボットの動作軌道を倒立振子モデルにより接続することで、実時間での歩容計画を実現した。また、Setiawanら[9]はオフラインで生成した歩容をデータベースの形で蓄えておくことで、実時間での歩容計画を実現した。Nishiwakiら[10]はZMPに関する微分方程式を離散化する手法を応用し、数歩の軌道を接続する手法を提案した。

これに対して、本研究では解析解を用いた実時間歩容計画のフレームワークを提案する。提案する手法には、他の従来手法と比較して以下のような利点がある。まず、数値計算の負荷が小さいため、計算が非常に高速であり、実時間での歩容計画に馴染みやすいことが挙げられる。また、解析解を用いることで、どのような歩容が実現可能で、どのような歩容が実現困難であるかが明確に分かる。本手法の場合、目標重心軌道とZMP軌道

を同時に計画するが、生成された目標ZMP軌道が支持多角形に含まれないような実時間歩容計画は実現不可能である。また、歩容のバリエーションを増やしても、歩容を実現するために必要なデータの数を増やさなくてよいことも利点の一つである。

また、梶田ら[11]は予見制御理論を応用した歩容生成手法を提案している。この理論を用いると、予見時間より先の目標軌道は、現在のロボットの状態に依存して設計することが可能である。それに対して、本研究では素早い歩容の変更を実現する観点より、現在ある目標軌道を変更することによって実時間歩容計画を実現する。

3. 基本的アイデア

本論文では、ヒューマノイドロボットが平らな床面上を歩く場合を想定する。また、床面が基準座標系の $x-y$ 平面と平行で、 $-z$ 方向に重力が働く場合を考える。初期姿勢においてロボットは x 方向に動き出すと仮定する。本論文では、特に $x-z$ 面内のロボットの運動に着目するが、 $y-z$ 面内の運動も全く同様に扱うことが可能である。

目標ZMP軌道が時間に関するスプライン関数であると仮定する。ここで、第 j 時間区間におけるZMPの位置を $\mathbf{p}_{zmp}^{(j)} = [x_{zmp}^{(j)} \ y_{zmp}^{(j)} \ z_{zmp}^{(j)}]^T$ とおくと、特に $x-z$ 面内の成分は以下のように表すことができる。

$$x_{zmp}^{(j)} = \sum_{i=0}^n a_i^{(j)} (t - t_{j-1})^i \quad (1)$$

$$t_{j-1} \leq t \leq t_j, \quad j = 1, \dots, m$$

ここで、 $a_i^{(j)}$ ($i = 1, \dots, m, j = 1, \dots, m$) は係数を表すスカラーである。歩行においては、両方の足が同時に床面と接する両脚支持期と、片方の足のみが接する単脚支持期が交互に現れる。本研究では、両脚支持期と単脚支持期で時間区間を分けて、単脚支持期でZMPを一定値にする場合は、対応する時間区間において $n = 0$ とした。また、平らな床面上での歩行を想定しているので、 $z_{zmp}^{(j)}$ は一定である。ZMP軌道の一例をFig. 2に示す。図においてロボットは $t = t_0$ において単脚支持期を開始し、 $t = t_1$ において完了する。その後 $t = t_2$ まで2足で床面

Fig. 2 ZMP trajectory

と接し、 $t = t_2$ において次の単脚支持期を開始する。

第 j 時間区間における ZMP に対応するロボット全体の重心位置を $\mathbf{p}_G^{(j)} = [\bar{x}_G^{(j)} \ \bar{y}_G^{(j)} \ \bar{z}_G^{(j)}]^T$ とおく。また、ロボット全体が持つ重心まわりの角運動量を $\mathbf{L}^{(j)} = [\mathcal{L}_x^{(j)} \ \mathcal{L}_y^{(j)} \ \mathcal{L}_z^{(j)}]^T$ とおく。このとき、 $x-z$ 面内における ZMP の位置とロボット全体の重心は以下のような微分方程式により関係づけられる。

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

次に、 $\bar{x}_G^{(j)} = x_G^{(j)} + \Delta x_G^{(j)}$ とおく。また、ロボット重心は鉛直方向に運動しないと仮定し $z_g^{(j)} = \text{const}$ とおくと、式 (2) は次のような二つの線形な微分方程式に分けることができる。

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

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

ここで、本論文では主に式 (3) に着目し、式 (4) の影響は大きくなく、シミュレーションや実験において導入されるフィードバック制御系 [12] において吸収可能であるとする。ただし、式 (4) の影響を陽に考慮することは可能であり、その手法は付録に示す。また、 $z_g^{(j)} = \text{const}$ とはならない場合、本論文で提案する手法では ZMP に誤差が生じる。この点についても付録に示す。

式 (1) を式 (3) に代入し、 $x_G^{(j)}$ について解くことにより、第 j 時間区間におけるロボット重心の軌道は次のような形で導出される。

$$x_G^{(j)} = V^{(j)} \cosh(\omega_c(t - t_{j-1})) + W^{(j)} \sinh(\omega_c(t - t_{j-1})) + \sum_{i=0}^n A_i^{(j)}(t - t_{j-1})^i \quad (5)$$

$$a_i^{(j)} = A_i^{(j)} - \frac{1}{\omega_c^2}(i+1)(i+2)A_{i+2} \quad (6)$$

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

ここで、 $\omega_c = \sqrt{g/(z_G - z_{zmp})}$ であり、 $V^{(j)}$ 、 $W^{(j)}$ ($j = 1, \dots, m$) は係数を表すスカラーである。式 (5) には $\sinh$ や $\cosh$ が斉次解に含まれるため、初期値問題として係数 $V^{(j)}$ 、 $W^{(j)}$ ($j = 1, \dots, m$) を求めると、多くの場合解が発散する。そこで、2点境界値問題を用いなくてはならない。2点境界値問題では、重心の初期位置と終端位置が与えられる反面、重心の初期速度と終端速度を与えることができない。このため、実時間歩容計画において、動作軌道を接続する場合に、速度の連続性を保証するのが困難になる。この手法については、次章で議論する。

式 (5) には、 $V^{(j)}$ 、 $W^{(j)}$ ($j = 1, \dots, m$) の $2m$ 個の未知の係数が存在する。二点境界値問題として、次のような $2m$ 個の境界条件を与える。

重心位置に関する初期条件：

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

重心位置・速度に関する二つの領域の接続：

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

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

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

重心位置に関する終端条件：

$$x_G^{(m)}(t_f) = V^{(m)} \cosh(\omega_c(t_f - t_{m-1})) + W^{(m)} \sinh(\omega_c(t_f - t_{m-1})) + \sum_{i=0}^n A_i^{(m)}(t_f - t_{m-1})^i \quad (11)$$

式 (8)～(11) を用いることで、 $2m$ 個の係数 $V^{(j)}$ 、 $W^{(j)}$ ($j = 1, \dots, m$) を以下のように決定することができる。

$$\mathbf{y} = \mathbf{Z}^{-1} \mathbf{w} \quad (12)$$

ここで、

$$\mathbf{y} = [V^{(1)} \ W^{(1)} \ \dots \ V^{(m)} \ W^{(m)}]^T$$

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

$$\mathbf{z}_0 = [1 \ 0 \ 0 \ 0]$$

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

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

$$\mathbf{w} = \begin{bmatrix} x_G^{(1)}(t_0) - A_0^{(1)} & \dots & A_0^{(j+1)} - \sum_{i=0}^n A_i^{(j+1)}(t_j - t_{j-1})^i \\ A_1^{(j+1)} - \sum_{i=0}^n i A_i^{(j+1)}(t_j - t_{j-1})^{i-1} & \dots \\ x_G^{(m)}(t_f) - \sum_{i=0}^n A_i^{(m)}(t_f - t_{m-1})^i \end{bmatrix}^T$$

ここで、行列 $\mathbf{Z}$ が正則であることは、 $\mathbf{Z}$ に含まれる行列 $\mathbf{Z}_j$ ($j = 1, \dots, m-1$) の位置、ならびに $\text{rank} \mathbf{Z}_j = 2$ ($j = 1, \dots, m-1$)

であることより明らかである。

式 (12) により得られた $V^{(j)}, W^{(j)}$ ($j = 1, \dots, m$) を式 (5) を代入することで、与えられた ZMP 軌道を実現し、与えられた初期条件や終端条件を満足するロボット重心の軌道を導出することが出来る。

4. 実時間歩容計画

4.1 軌道の接続

実時間で歩容を計画するため、現在実行している ZMP 軌道に新しい歩幅を持つ ZMP 軌道を接続することを考える。新たな軌道を接続するのは両脚支持期の始めとする。このとき、実時間軌道接続と準実時間軌道接続の 2 種類の手法を考える。足が床につく位置を変更する場合、実時間軌道接続では足が床に接する直前までに、位置の変更量が分かればよい。それに対して、準実時間軌道接続では、前の両脚支持期の間に次の足をつく位置が分かっているなければならない。

Fig. 3 において、(a) は ZMP 軌道を示し、(b) はロボット重心の軌道を示す。 $t = t_2$ から t_f の時間区間において歩幅を l_2 から l_{new} に変更する問題により、軌道接続法についてもう少し詳しく説明する。

Fig. 3 Overview of trajectory connection

4.1.1 実時間軌道接続

この手法では、Fig. 3 (c) に示すように、 $t = t_2$ において軌道が接続される。両脚支持期において ZMP の位置が変化する直前に、歩幅が l_2 から l_{new} に変更される。

この手法では、歩幅の変更量である l_2 と l_{new} の差が大きい場合、 t_2 から t_f の間の時間区間で加速度が過大になることが予想される。つまり、この手法は l_2 と l_{new} の差が小さい場合や、 t_2 から t_f の間の両脚支持期が長い場合に有効であると考えられる。

4.1.2 準実時間軌道接続

Fig. 3 (d) に示すように、現在の時刻が $t = t_0$ 以前の場合、 $t = t_0$ において軌道を接続することができる。また、この手法では、新たに計算される ZMP 軌道で最初の歩幅は、現在実行している ZMP 軌道で対応する歩幅と同様であるが、2 番目の歩幅が l_2 から l_{new} に変更される。

この手法では、新たな歩幅は実時間軌道接続の場合よりも早く決定されなくてはならない。その反面、新たに計算される ZMP 軌道において最初の歩幅が現在の ZMP 軌道と同様であるため、 $t = t_2$ から t_f の間で加速度が過大にならないことが期待できる。つまり、この手法は l_2 と l_{new} の差が大きい場合に有効であると考えられる。

4.2 重心と ZMP の同時計画

新たに計算された軌道を、現在実行している軌道に接続する。ここで、前章において述べたように、ZMP とロボット重心の関係を表す微分方程式は 2 点境界値問題により解かれるため、初期速度を指定することができない。このため、軌道を接続する場合に速度の連続性を保証できない。このような問題に対して、新たに重心軌道を計算するときに、第一区間の ZMP 軌道も同時に計画することを考える。つまり、式 (5) において、 $A_i^{(1)}$ ($i = 0, \dots, n$) も未知の定数とおく。ここで、未知数の数が増加するため、対応して次のように速度に関する境界条件を追加できる。

ZMP の位置に関する初期条件：

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

ZMP の位置に関する第一区間の終端条件：

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

重心速度に関する初期条件：

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

第一区間において $n = 2$ であるとき、合計 $2m + n + 1$ 個の未知数があるのに対して、同じ数の条件式が存在することになる。つまり、未知数 $A_0^{(1)}, \dots, A_n^{(1)}, V^{(1)}, W^{(1)}, \dots, V^{(m)}, W^{(m)}$ は、式 (8) ~ (11), (13) ~ (15) より以下のように導出される。

$$\tilde{y} = \tilde{Z}^{-1} \tilde{w} \quad (16)$$

ここで、

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

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

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

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

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

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

である。行列 $\tilde{\mathbf{Z}}$ の正則性は数値的に確認した。その結果、少なくとも本論文のシミュレーションや実験においては、すべての場合に正則となった。

4.3 議論

本節では、提案する手法の特徴について議論する。まず、提案する手法は解析解を用いているため、計算が高速である点の特徴的であるが、 $\tilde{\mathbf{Z}}$ の逆行列を計算するのに計算時間を要する可能性がある。例えば 3 歩の軌道を計算し $m = 9$ である場合、行列 $\tilde{\mathbf{Z}}$ のサイズは 20×20 であるが、シミュレーションや実験においては、1 サンプル時間内に軌道計画を含めたすべての計算を行うことができることを確認した。また、 $\tilde{\mathbf{Z}}$ の要素には歩幅が含まれないため、両脚支持期と単脚支持期の時間を固定すると、 $\tilde{\mathbf{Z}}$ の逆行列はあらかじめ計算しておくことが可能である。

また、本章では、新たに計算される ZMP 軌道の第一時間区間においては、ZMP の位置に関する連続性のみを考えた。ここで、本論文のシミュレーションや実験においては、式 (13)、

(14) の ZMP の位置に関する境界条件以外に、ZMP の速度に関する連続性も考慮し、 $n = 4$ として歩容計画の計算を行った。

さらに、本章では主に Sagittal 面内の運動について提案する手法を説明したが、全く同様な方法で Lateral 面内の運動も議論できる。ここで、Sagittal 面と Lateral 面の両方を考慮した場合、新たに計算する軌道において、最初の両脚支持期における ZMP 軌道が直線になるとは限らない。この点については、次章においてより詳細に議論する。

5. シミュレーション

OpenHRP [13] [14] を用いてシミュレーションを行った。シミュレーションにおいては、提案する手法により ZMP とロボット重心の軌道を導出した。また、これらの軌道を、分解運動量制御 [15] により関節軌道に変換し、ロボットの関節角度指令とした。ロボットのモデルとして、HRP-2 [16] を用いた。HRP-2 は身長が $h = 1.57$ [m]、体重が $m = 57$ [kg] で、30 自由度を持つヒューマノイドロボットである。シミュレーションにおいてロボットは 10 歩歩くが、そのうち最初の 8 歩において、両脚支持期の最初に新しい軌道が計算されて現在実行している軌道に接続される。そして、5 歩目において歩幅が変更される。

まず、実時間軌道接続に関するシミュレーション結果を Fig. 4~7 に示す。Fig. 4 に示すシミュレーションでは、片脚支持期の時間を $T_s = 0.8$ [sec]、両脚支持期の時間を $T_{dbl} = 0.2$ [sec] と設定した。また、5 歩目においてあらかじめ設定された歩幅は $l_5 = 0.05$ [m] であるが、これを $l_{new} = 0.15$ [m] に変更した。5 歩目において、ZMP 軌道は Fig. 4 (e) に示すように大きく湾曲している。これは、歩幅を大きくしたため、それに合わせて胴体を前方向に加速させるために、ZMP 軌道が後ろ方向に移動したためである。しかしながら、ZMP の軌道が直線から大きくずれたため、Fig. 5 に示すようにロボットは転倒する。

この ZMP の移動量は、歩幅が大きく変化した場合や、両脚

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

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])

支持期の時間が短い場合に顕著になると考えられる。そこで、Fig. 6 には 5 歩目に歩幅を $l_5 = 0.11$ [m] から $l_{new} = 0.12$ [m] に変更した場合のシミュレーション結果を示す。この場合、ZMP 軌道の直線からのずれは小さく、ロボットは転倒しない。

また、Fig. 7 には、両脚支持期の時間を $T_{dbl} = 1.5$ [sec] に設定した場合のシミュレーション結果を示す。この場合も ZMP 軌道の直線からのずれは小さく、ロボットは転倒しない。ここで、筆者らは床に置かれた対象物の押し操作において、実時間軌道接続を用いることで対象物の物理パラメータに依存せずに安定な押し操作を実現した [17]。この際、両脚支持期において対象物を押すことにし、両脚支持期の時間は十分に長く設定した。

一方、準実時間軌道接続に関するシミュレーション結果を Fig. 8, 9 に示す。5 歩目において、歩幅を $l_5 = 0.05$ [m] か

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])

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])

ら $l_{new} = 0.15$ に変化させたが、ZMP 軌道の直線からのずれは小さく、ロボットは転倒しない。これは、新たに計算された軌道において、歩幅の変化が反映されるのが第 2 番目のステップであるため、それまでの間に胴体を加速できるためである。つまり、この手法では歩幅の変化が反映されるのが、実時間軌道接続よりも遅いが、ロボットはより容易に平衡を維持できる。

6. 実 験

ヒューマノイドロボット HRP-2 [16] を用いて実験を行った。実験では、ヒトが HRP-2 のハンドを握る。HRP-2 の手首には力・トルクセンサが内蔵されており、ヒトが HRP-2 のハンドを押したり引いたりする情報を得ることが可能である。力・

Fig. 9 Snapshot of the quasi-real-time trajectory connection

Fig. 10 Snapshot of experiment

Fig. 11 Experimental results

トルクセンサの出力に係数をかけることで、変位の情報に変換し、これによりロボットの進行方向を準実時間軌道接続を用いて計画した。実験において、片脚支持期の時間は $T_s = 0.7$ [sec] で、両脚支持期の時間は $T_{dbl} = 0.1$ [sec] とした。実験の様様を Fig. 10 に示す。あたかも、目をつぶっているヒトの手を握って、誘導するような動作が実現される。また、実験結果として ZMP 軌道と重心軌道を Fig. 11 に示す。この実験では、ロボットを左前方に誘導したが、ロボットが意図したように動作していることが分かる。なお、手先の力センサを用いた歩容計画は、Setiawan ら [9] によって前後方向の運動に対して最初に行われた。本実験は、彼らの実験結果を拡張したものである。

7. 結 言

本論文では、解析解を用いたヒューマノイドロボットの実時間歩容計画について議論した。提案する手法では、ロボットの動作を変更するため、実行中のロボット重心と ZMP の軌道に、新たに計算した軌道を接続させる。このとき、軌道を接続させる方法として実時間軌道接続と準実時間軌道接続を考えた。実時間軌道接続は素早い動作の変更が可能である反面、大きな変更は困難である。それに対して、準実時間軌道接続は動作の変更が遅れる反面、大きな変更が可能である。また、提案する手法は解析解を用いているため、非常に速く動作軌道が計算され、実時間歩容計画に馴染み易い。提案する手法の有効性をシミュレーション、ならびに手先の力センサの値により実時間で歩容を計画する実験により示した。

今後の課題としては、エネルギー消費等の評価関数を最小にする歩行を実現すること等が考えられる。

参 考 文 献

- [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] S. Kajita, T. Yamaura and A. Kobayashi: "Dynamic Walking Control of a Biped Robot Along a Potential Energy Conserving Orbit," IEEE Trans. Robot. and Automat., vol.8, no.4, pp.431-438, 1992.
- [4] K. Nagasaka, H. Inoue and M. Inaba: "Dynamic Walking Pattern Generation for a Humanoid Robot Based on Optimal Gradient Method," Proc. of IEEE Int. Conf. on Systems, Man, and Cybernetics, pp.VI-908-913, 1999.
- [5] 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.

- [6] 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.
- [7] T. Takenaka: "Gait Generation Device for a Legged Mobile Robot," Japanese Patent Application H10-217161, 1998.
- [8] T. Takenaka: "Motion Generation Device for a Legged Mobile Robot," Japanese Patent Application 2002-217161, 2002.
- [9] S.A. Setiawan, S.H. Hyon, J. Yamaguchi and A. Takamishi: "Physical Interaction between Human and a Bipedal Humanoid Robot—Realization of Human-follow Walking—," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.361-367, 1999.
- [10] 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.
- [11] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa: "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.
- [12] 金子ら: "HRP-2の歩行安定化制御", 第21回日本ロボット学会学術講演会予稿集 CD-ROM, 2003.
- [13] 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.
- [14] H. Hirukawa et al.: "OpenHRP: Open Architecture Humanoid Robot Platform," Proc. of Int. Symp. on Robotics Research, 2001.
- [15] 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.
- [16] K. Kaneko et al.: "The Humanoid Robot HRP2," Proc. of IEEE Int. Conf. on Robotics and Automation, 2004.
- [17] 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.
- [18] 森澤ら: "パラメトリック曲面上に拘束された2足歩行パターン生成", 第21回日本ロボット学会学術講演会予稿集 CD-ROM, 2004.

付録 A. 角運動量の考慮

3章では, ZMP とロボット重心との関係を表す式 (2) を, 式 (3) と式 (4) の二つの微分方程式に分けた. そして, 式 (4) の影響は小さいとして, 式 (3) のみに着目した. 次に, 式 (4) の影響を考慮することを考える. 式 (4) は, 左辺の関数が複雑な形をしているため, そのままの形では $\Delta x_G^{(j)}$ について解析的に解くことはできない. そこで, 左辺をフーリエ級数展開 [2] する. つまり, 左辺を次のようにおく.

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

ここで,

Fig. 12 Simulation result of angular momentum compensation

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

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

である. 式 (A.1) を式 (4) に代入し, $\Delta x_G^{(j)}$ について解くことにより, 次のような式が得られる.

$$\Delta x_G^{(j)} = \frac{C_0^{(j)}}{2} + \sum_{i=1}^{\infty} C_i^{(j)} \cos\left(\frac{2\pi i(t-t_{j-1})}{t_j-t_{j-1}}\right) + \sum_{i=1}^{\infty} D_i^{(j)} \sin\left(\frac{2\pi i(t-t_{j-1})}{t_j-t_{j-1}}\right) + \Delta V^{(j)} \cosh(\omega_c(t-t_{j-1})) + \Delta W^{(j)} \sinh(\omega_c(t-t_{j-1})) \quad (\text{A.4})$$

ここで,

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

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

である. 式 (A.4) は無限級数を含むが, 適当な次数で打ち切ることによって近似をする. 例えば, n 次で打ち切る場合, 3章に示した方法により係数 $\Delta V^{(j)}$, $\Delta W^{(j)}$, $C_i^{(j)}$, $D_i^{(j)}$ ($i = 1, \dots, n$) を決定することができる.

シミュレーション結果を Fig. 12 に示す. このシミュレーションでは, ロボットは移動せずに3歩足踏みをしている. Lateral 面内の ZMP 軌道を Fig. 12 (a), (b) に, また目標 ZMP からの誤差を Fig. 12 (c), (d) に示す. シミュレーションでは $n = 3$ とした. 比較的粗い近似であるが, 補償すべき ZMP の誤差自体が元々小さいため, 誤差の補償が有効に働いていることが分かる.

付録 B. 重心の鉛直方向の運動

次に, 重心が鉛直方向に運動する場合を考える. 本論文で提案

する手法を重心が鉛直方向に運動する場合に適用すると, ZMP の位置に誤差が生じる. 本章では, この誤差を見積もることを考える. 重心が鉛直方向に運動することによる ZMP の誤差を $\Delta x_{zmp}^{(j)}$ とおき, ZMP の位置を目標からの誤差分を含めて

$$x_{zmp}^{(j)} = \sum_{i=0}^n a_i^{(j)} (t - t_{j-1})^i + \Delta x_{zmp}^{(j)} \quad (B.7)$$

$$t_{j-1} \leq t \leq t_j, \quad j = 1, \dots, m$$

とおく. $z_g^{(j)} = \text{const}$ という仮定を取り除き, 式 (5), (B.7) を式 (3) に代入することで, 次式が得られる.

$$\Delta x_{zmp}^{(j)} = -f(t)^{(j)} \{ \dot{\omega}_c^2 (t - t_{j-1})^2 + 2\dot{\omega}_c \omega_c (t - t_{j-1}) \} - g(t)^{(j)} \{ \ddot{\omega}_c (t - t_{j-1}) + 2\dot{\omega}_c \} \quad (B.8)$$

$$f(t)^{(j)} = \{ V^{(j)} \cosh(\omega_c(t - t_{j-1})) + W^{(j)} \sinh(\omega_c(t - t_{j-1})) \} / \omega_c^2 \quad (B.9)$$

$$g(t)^{(j)} = \{ W^{(j)} \cosh(\omega_c(t - t_{j-1})) + V^{(j)} \sinh(\omega_c(t - t_{j-1})) \} / \omega_c^2 \quad (B.10)$$

つまり, 本手法では式 (B.8) で示される ZMP の誤差が支持多角形を出るような動作パターンは実行不可能である. ここで, 森澤ら [18] はロボット重心の上下への運動を許容した歩行動作の生成手法を提案している.

原田研介 (Kensuke Harada)

1968 年 9 月 28 日生. 1997 年 3 月京都大学大学院工学研究科機械工学専攻博士後期課程修了. 博士 (工学). 同年 4 月広島大学工学部助手, 2002 年 4 月独立行政法人産業技術総合研究所研究員. ヒューマノイドロボット, ロボットハンド, ロボットマニピュレータ等, ロボット機構の力学と制御に関する研究に興味を持つ. IEEE, 計測自動制御学会, 日本機械学会などの会員. (日本ロボット学会正会員)

森澤光晴 (Mitsuharu Morisawa)

1976 年 7 月 17 日生. 2005 年 3 月慶應義塾大学大学院理工学研究科総合デザイン工学専攻博士課程修了. 博士 (工学). 同年 4 月より独立行政法人産業技術総合研究所ヒューマノイド研究グループ研究員. ヒューマノイドロボット, パラレルマニピュレータ, モーションコントロールなどの研究に従事. 電気学会の会員. (日本ロボット学会正会員)

藤原清司 (Kiyoshi Fujiwara)

1972 年 4 月 11 日生. 1997 年筑波大学大学院工学研究科修了. 同年電子技術総合研究所入所. 現在独立行政法人産業技術総合研究所知能システム部門ヒューマノイド研究グループ研究員. 生体制御, 医療福祉工学, マンマシンインタフェース, ヒューマノイドロボットなどの研究に従事. (日本ロボット学会正会員)

比留川博久 (Hirohisa Hirukawa)

1959 年 1 月 15 日生. 1987 年神戸大学大学院自然科学研究システム科学専攻博士課程修了. 学術博士. 同年通産省工業技術院電子技術総合研究所に入所. 1994~1995 年米国スタンフォード大学計算機科学科客員研究員. 2001 年改組より独立行政法人産業技術総合研究所知能システム研究部門ヒューマノイド研究グループ長. ロボットの動作計画アルゴリズム, ネットワークロボティクス, ヒューマノイドロボット等の研究に従事. (日本ロボット学会正会員)

梶田秀司 (Shuuji Kajita)

1961 年 1 月 26 日生. 1985 年東京工業大学大学院修士課程修了 (制御工学専攻). 同年通産省工業技術院機械技術研究所に入所. 2 足歩行ロボット等の動的制御技術の研究に従事. 1996 年 2 月より 1 年間米国カリフォルニア工科大学客員研究員. 2001 年より組織改変に伴い独立行政法人産業技術総合研究所主任研究員, 現在にいたる. 1996 年 3 月東京工業大学より学位取得 (工学博士). 1996 年度計測自動制御学会論文賞受賞. 計測自動制御学会, IEEE (Robotics and Automation Society) の会員. (日本ロボット学会正会員)

金広文男 (Fumio Kanehiro)

1971 年 7 月 22 日生. 1999 年東京大学大学院工学系研究科情報工学専攻博士課程修了. 博士 (工学). 1998 年より日本学術振興会特別研究員. 2000 年電子技術総合研究所入所. 現在産業技術総合研究所知能システム研究部門研究員. ヒューマノイドロボットのシステム構成法, 全身行動制御に興味がある. IEEE, 人工知能学会, 日本ソフトウェア科学会, 情報処理学会の各会員. (日本ロボット学会正会員)

金子健二 (Kenji Kaneko)

1965 年 9 月 22 日生. 1990 年慶應義塾大学大学院理工学研究科電気工学専攻修士課程修了. 同年工業技術院機械技術研究所に入所. 1995 年 9 月より半年間米国カーネギーメロン大学客員研究員. 1999 年 9 月より 1 年間仏国 CNRS パリロボット研究所客員研究員. 2001 年より組織改変に伴い独立行政法人産業技術総合研究所主任研究員, 現在にいたる. 博士 (工学). モーションコントロール, マイクロマシン, 遠隔制御, ヒューマノイドロボット等の研究に従事. 日本機械学会, 電気学会の各会員. (日本ロボット学会正会員)