

平面2足歩行ロボットのグラフ型モデル予測制御

田崎 勇一* 井村 順一*

Graph Based Model Predictive Control of a Planar Bipedal Robot

Yuichi Tazaki* and Jun-ichi Imura*

In this paper, a new control method for a planar bipedal robot, which we call here Graph-based Model Predictive Control, is proposed. This method consists of two phases: the graph construction phase and the realtime control phase. In the graph construction phase, a directed graph on the state space of the control target is constructed off line. In the realtime control phase, the controller drives the state of the control target so as to make it move through graph nodes connected by directed edges. By tracing directed edges, a model predictive control is achieved in some sense. Moreover, since the directed graph is constructed in advance, the realtime computational cost is dramatically reduced compared with the ordinary MPC. In addition, by constructing multiple directed graphs based on different cost functions, one can design multiple motions and switching trajectories among them in a uniform way. The proposed method is applied to the speed changing control problem of a bipedal walker on a 2-dimensional plane and its effectiveness is verified by numerical simulation.

Key Words: Bipedal robot, MPC, Motion planning

1. はじめに

実環境に適用可能な歩行ロボットには、定常歩行だけでなく、定位置での静止や方向転換といった様々な動作を行う能力が求められる。このような要求のもと、様々な歩行制御則やモーションプランニング手法が提案されてきた [1] [2] [3] [4] [5] [6]。この種の問題では、複数の動作を個別に行う能力に加えて、ある動作から別の動作へスムーズに切り替える機能が制御系に求められる。このためには、ロボットがいかなる姿勢・速度にあるときも、その状態から速やかに新たな目標動作へ遷移できなければならない。これを達成するには異なる動作間を結ぶ過渡軌道をいかに生成するかが重要となる。Kovarら [3] はモーションキャプチャデータに対し、十分に近い姿勢を表すキーフレームを節点としてグラフ構造を生成するモーショングラフと呼ばれる手法を考案した。岡田ら [5] は、あらかじめ与えられた状態の周期的時系列データに対し、これを attractive な閉軌道とするようなベクトル場を生成する手法を提案した。ベクトル場を変化させることである閉軌道から別の閉軌道へ滑らかに遷移できるとしている。

非線形ダイナミクスや拘束条件を陽に取り扱える制御手法として、モデル予測制御 (以下 MPC) が近年活発に研究されている [7]。MPC は制御手法として一般性が高く、適用対象を選ば

ない。このため歩行ロボットのような複雑な制御対象にも理論的には自然に適用できる。しかしながら、通常の MPC は実行時に最適制御問題を繰り返し解く必要があり、このことが歩行ロボットといった比較的高次元なシステムへの適用に対する障害となっていた。

これに対し、井村ら [8] [9] はピークルの障害物回避問題に対して、ピークルの状態空間に経由点を配置し、障害物を回避しつつ与えられた評価関数を最小化するパスをリアルタイムで探索する手法を提案した。これにより、MPC において繰り返し解くべき最適制御問題は、経由点の最適配置問題と最適経路の探索問題に分離できることが示された。しかしながら経由点の配置と経路探索をリアルタイムに行うため、高次元のシステムに対して計算量の問題は残されている。

本論文ではこのアイデアを発展させ、経由点の配置と経路設計を設計時に予め行うモデル予測制御 (以下グラフ型モデル予測制御と呼ぶ) を提案する。本手法は経由点を固定するものの、従来の MPC と比べてリアルタイムでの計算量が劇的に低減されるため、歩行ロボットのような高次元のシステムに対しても適用可能となる。また、物理的な拘束条件に加えて異なる複数通りの評価関数を用いて経路設計を行うことにより、複数の動作およびこれらの間の切り替え動作を統一的な枠組みで設計することができる。これらの特性を有効に利用した適用例として、本論文ではグラフ型モデル予測制御を用いて平面上の2足歩行ロボットに対する定常歩行制御および歩行スピードの切り替え制御を実現する。

*東京工業大学

*Tokyo Institute of Technology

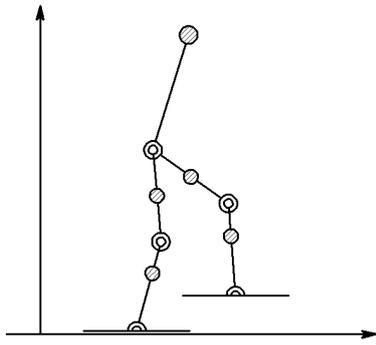


Fig. 1 Model of Planar Bipedal Robot.

なお、作業空間をグラフ化する手法はパスプランニングの分野でロードマップ法という名前でも古くから用いられている [10] [11] [12]。しかしこれらの研究では与えられた出発点と目標点を結ぶパスを得ることを目的としており、制御対象の動特性やパスの最適性は基本的に考慮されていない。また、歩行のような周期運動は対象とされていない。

本論文の流れを以下に述べる。まず第 2 節では、本論文で取り扱う平面上の 2 足歩行ロボットをハイブリッドシステムモデルとして定式化する。第 3 節では、提案手法であるグラフ型モデル予測制御を説明する。第 4 節では、本手法を 2 足歩行ロボットの変速制御に適用した数値結果を示す。最後に第 5 節で本研究の成果と将来の課題について述べる。

2. 平面 2 足歩行ロボットのハイブリッドシステムモデル

Fig. 1 に本研究で対象とする平面 2 足歩行ロボットを示す。足 (foot) の質量は 0 とする。このため、振り脚の足部はダイナミクスに寄与しない。整合性を保つため、振り脚の足首関節トルクは 0 とする。ロボットは必ず足全体で接地することとし、踵やつま先での点接地は起こらないとする。さらに、足部と地面との間の摩擦係数は ∞ とする。足と地面との衝突は完全非弾性衝突とする。したがって、衝突時には振り脚の足の速度は必ず 0 となる。

ロボットの運動を記述する変数は次の通りである。

体幹重心の位置	(x, y)
体幹の傾き	θ_{body}
股関節角度	$\theta_{\text{hip}_i} (i \in \{l, r\})$
膝関節角度	$\theta_{\text{knee}_i} (i \in \{l, r\})$
股関節トルク	$\tau_{\text{hip}_i} (i \in \{l, r\})$
膝関節トルク	$\tau_{\text{knee}_i} (i \in \{l, r\})$
足首関節トルク	$\tau_{\text{ankle}_i} (i \in \{l, r\})$
位置変数	$\mathbf{q} = (\theta_{\text{body}}, \theta_{\text{hip}_l}, \theta_{\text{hip}_r}, \theta_{\text{knee}_l}, \theta_{\text{knee}_r})^T$
速度変数	$\dot{\mathbf{q}} = \frac{d}{dt} \mathbf{q}$
入力変数	$\mathbf{u} = (\tau_{\text{hip}_l}, \tau_{\text{hip}_r}, \tau_{\text{knee}_l}, \tau_{\text{knee}_r}, \tau_{\text{ankle}_l}, \tau_{\text{ankle}_r})^T$
状態変数	$\mathbf{x} = (\mathbf{q}^T, \dot{\mathbf{q}}^T)^T$

本論文では $(\cdot)_l, (\cdot)_r$ はそれぞれ左脚、右脚に関する量を意味する。また、平面上の位置ベクトルについて $(\cdot)_x, (\cdot)_y$ はそれぞれ x 方向成分、 y 方向成分を表す。さらに、体幹重心を基準とする左右足先の位置および速度を $\mathbf{p}_i, \mathbf{v}_i (i \in \{l, r\})$ と書くこと

とする。前述の仮定より、足部はロボットの運動に寄与しない。したがって足首関節の角度および角速度は状態変数に含まれない。また、体幹重心の位置 (x, y) およびその時間微分も状態変数に含まれないことに注意されたい。

ロボットの運動は、L, R, LR, RL の 4 つのモードに分けられる。L モードは、ロボットが左足のみで支持している場合を表す。逆に R モードは、ロボットが右足のみで支持している場合を表す。次に LR モードは、L モードにおいて振り足である右足が地面と衝突することで速度変化が生じ、R モードへ遷移する事象を表す。これに対し RL モードは R モードにおいて左足が地面と衝突することで L モードへ遷移する事象を表す。それぞれのモードにおけるロボットの運動は次のような常微分方程式と代数方程式によって記述される。

$$\left\{ \begin{array}{l} \dot{\mathbf{q}}(t) = X_L(\mathbf{q}(t))\mathbf{u}_L(t) + Y_L(\mathbf{q}(t), \dot{\mathbf{q}}(t)) \\ I(t) = L \\ \dot{\mathbf{q}}(t) = X_R(\mathbf{q}(t))\mathbf{u}_R(t) + Y_R(\mathbf{q}(t), \dot{\mathbf{q}}(t)) \\ I(t) = R \\ \dot{\mathbf{q}}(t)^+ = Z_{LR}(\mathbf{q}(t))\dot{\mathbf{q}}(t) \\ I(t) = LR \\ \dot{\mathbf{q}}(t)^+ = Z_{RL}(\mathbf{q}(t))\dot{\mathbf{q}}(t) \\ I(t) = RL \end{array} \right. \quad \begin{array}{l} (\mathbf{q}(t), \dot{\mathbf{q}}(t), \mathbf{u}(t)) \in S_L \\ (\mathbf{q}(t), \dot{\mathbf{q}}(t), \mathbf{u}(t)) \in S_R \\ (\mathbf{q}(t), \dot{\mathbf{q}}(t)) \in S_{LR} \\ (\mathbf{q}(t), \dot{\mathbf{q}}(t)) \in S_{RL} \end{array} \quad (1)$$

$(\cdot)^+$ は次の瞬間の値を表す。また、 $\mathbf{u}_L(\mathbf{u}_R)$ は、入力変数 \mathbf{u} から振り足である右足 (左足) の足首トルク (仮定より常に 0 である) の成分を除いたものである。 $I(t)$ は時刻 t におけるモードを表す。 S_I は、モードが I となるための条件を満たす \mathbf{q} と $\dot{\mathbf{q}}$ 、あるいは $\mathbf{q}, \dot{\mathbf{q}}$ 、および \mathbf{u} の成す領域を表す。詳しくは付録を参照されたい。

3. グラフ型モデル予測制御の提案

3.1 アルゴリズムの概要

提案手法であるグラフ型モデル予測制御のコントローラ構成は、Fig. 2 に示すように状態空間上に構築された有向グラフとリアルタイムコントローラからなる。有向グラフの個々のノードはロボットのある状態 (姿勢と速度) を持つ。グラフ上の複数のノードの連なり、すなわち経路を考えることにより、ノードを経由点とする様々な動作軌道を構成することができる。ここで、歩行運動などの動作を実現する経路と、ある動作から別の動作へ移行するための切り替え経路を設計することを考える。この目的に対して、本手法では実現すべき動作毎にこれに対応する経路の評価関数を設計する。次に、モデル予測制御の原則にもとづいて各動作を実現する有向グラフを構築する。Fig. 2 では動作の種類は 2 つ (例えば低速歩行動作と高速歩行動作) であり、これらの動作を担う 2 つの有向グラフは、ひとつは実線、もう一方は破線で描いてある。各有向グラフは対応する動作自体を実現する周期経路と、周期経路外の経路点から周期経路へ合流する切り替え経路によって構成される。各時刻において、複数の有向グラフの内のいずれか 1 つが選択された状態にある。リアルタイムコントローラは、選択されている有向グラフの有向エッジを辿るようにロボットの状態を経由点から次の経路点

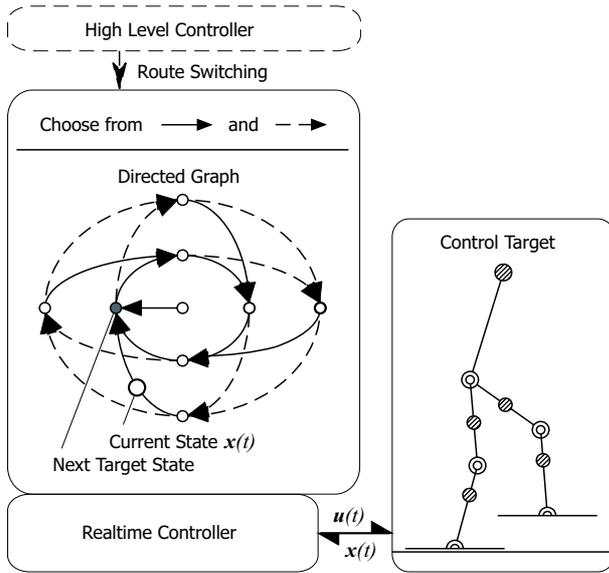


Fig. 2 Controller architecture.

へ遷移させる制御入力を生成する．実現する動作を切り替えるには，これに対応する有向グラフを新たに選択すればよい．

このような有向グラフの構築方法については第 3.3 節で述べるが，これに先立ってロボットの状態を与えられた初期状態から目標状態へ，所定の遷移時間で遷移させる制御入力の生成方法について次節で説明する．また，リアルタイムコントローラのロバスト性を向上するための工夫について第 3.4 節で述べる．

3.2 経路点間制御

グラフ型モデル予測制御では，与えられた 2 つの経路点 $\mathbf{x}_k = (\mathbf{q}_k^T, \dot{\mathbf{q}}_k^T)^T$, $\mathbf{x}_{k+1} = (\mathbf{q}_{k+1}^T, \dot{\mathbf{q}}_{k+1}^T)^T$ を結ぶ制御則が不可欠である．ここで \mathbf{x}_k が S_{LR} か S_{RL} の要素ならば，式 (1) に従い制御入りに依らずに瞬間的にジャンプ後の状態へ遷移する．したがってここでは \mathbf{x}_k が S_L か S_R の要素である場合のみ考える．このとき，経路点間制御は次のように定式化される．

[問題 1]

システム (1) の状態 $\mathbf{x}(t)$ が時刻 0 において経路点 \mathbf{x}_k にあるとき，与えられた時刻 h_k と目標経路点 \mathbf{x}_{k+1} について $\mathbf{x}(h_k) = \mathbf{x}_{k+1}$ とし，かつ評価関数

$$J(\mathbf{u}; h_k) = \int_0^{h_k} \mathbf{u}(\tau)^T R \mathbf{u}(\tau) d\tau \quad (R > 0) \quad (2)$$

を最小とする制御入力 $\mathbf{u}(t)$ を求めよ．

ここでは一般性を失うことなく出発時刻を 0，目標経路点への到達時刻を h_k とした．この問題は以下のようにして解析的に近似解を得ることができる．まず，次の制約条件を設ける．

[経路点間でのモード不変条件]

時刻 $t \in [0, h_k]$ においてモード変化は生じない．

もしこの制約条件が満たされるならば，経路点間でのシステムの振る舞いはある $I \in \{L, R\}$ に対する次の運動方程式に従う．

$$\dot{\mathbf{q}} = X_I(\mathbf{q}) \mathbf{u}_I + Y_I(\mathbf{q}, \dot{\mathbf{q}}) \quad (3)$$

I がいずれのモードをとるかは出発・目標経路点の状態の組合

せにより以下のように決まる．

$$I = \begin{cases} L & \text{if } p_{ly}(\mathbf{q}_k) < p_{ry}(\mathbf{q}_k), p_{ly}(\mathbf{q}_{k+1}) \leq p_{ry}(\mathbf{q}_{k+1}) \\ R & \text{if } p_{ly}(\mathbf{q}_k) > p_{ry}(\mathbf{q}_k), p_{ly}(\mathbf{q}_{k+1}) \geq p_{ry}(\mathbf{q}_{k+1}) \end{cases} \quad (4)$$

上記のいずれにも該当しない組合せについては遷移不能とする．ここで

$$\mathbf{u}_I = X_I(\mathbf{q})^{-1}(\mathbf{v} - Y_I(\mathbf{q}, \dot{\mathbf{q}})) \quad (5)$$

として入力変換すると以下の二重積分系が得られる．

$$\dot{\mathbf{x}} = A \mathbf{x} + B \mathbf{v} \quad A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (6)$$

式 (2) と式 (5) を用いると \mathbf{v} に関する評価関数

$$J_{\mathbf{v}}(\mathbf{v}; h_k) = \int_0^{h_k} \|\mathbf{v}(\tau) - Y_I(\mathbf{q}(\tau), \dot{\mathbf{q}}(\tau))\|_{\hat{R}_I(\mathbf{q})}^2 d\tau \quad (7)$$

$$R_I(\mathbf{q}) = X_I(\mathbf{q})^{-T} \hat{R} X_I(\mathbf{q})^{-1}$$

を得る．ここで $\|\mathbf{x}\|_M^2 = \mathbf{x}^T M \mathbf{x}$, \hat{R} は R から振り足の足首トルクに対する行と列を取り除いたものである．評価関数 (7) は $\mathbf{q}(\tau), \dot{\mathbf{q}}(\tau)$ に関する非線形項を含んでいるので，これらを $(\mathbf{x}_k + \mathbf{x}_{k+1})/2 = (((\mathbf{q}_k + \mathbf{q}_{k+1})/2)^T, ((\dot{\mathbf{q}}_k + \dot{\mathbf{q}}_{k+1})/2)^T)^T$ を用いて定数近似する．

$$\tilde{J}_{\mathbf{v}}(\mathbf{v}; \mathbf{x}_k, \mathbf{x}_{k+1}, h_k) = \int_0^{h_k} \|\mathbf{v}(\tau) - \bar{Y}_I\|_{\bar{R}_I}^2 d\tau \quad (8)$$

$$\bar{Y}_I = Y_I((\mathbf{q}_k + \mathbf{q}_{k+1})/2, (\dot{\mathbf{q}}_k + \dot{\mathbf{q}}_{k+1})/2)$$

$$\bar{R}_I = X_I((\mathbf{q}_k + \mathbf{q}_{k+1})/2)^{-T} \hat{R} X_I((\mathbf{q}_k + \mathbf{q}_{k+1})/2)^{-1}$$

以上より，問題 1 は次のような二重積分系に対する固定端問題に近似的に変形される．

[問題 2]

システム (6) に対して， $\mathbf{x}(0) = \mathbf{x}_k$, $\mathbf{x}(h_k) = \mathbf{x}_{k+1}$ ($\mathbf{x}_k = (\mathbf{q}_k^T, \dot{\mathbf{q}}_k^T)^T$, $\mathbf{x}_{k+1} = (\mathbf{q}_{k+1}^T, \dot{\mathbf{q}}_{k+1}^T)^T$) を満たしつつ評価関数 (8) を最小化する制御入力 \mathbf{v} を求めよ．

[定理]

問題 2 の最適制御入力 \mathbf{v}^* ，最適状態軌道 \mathbf{x}^* および評価関数の最小値 $\tilde{J}_{\mathbf{v}}^*$ は以下のような形で解析的に得られる．

$$\mathbf{x}^*(t; \mathbf{x}_k, \mathbf{x}_{k+1}, h_k) = \left(\mathbf{q}^*(t; \mathbf{x}_k, \mathbf{x}_{k+1}, h_k)^T \quad \frac{d}{dt} \mathbf{q}^*(t; \mathbf{x}_k, \mathbf{x}_{k+1}, h_k)^T \right)^T \quad (9)$$

$$\mathbf{v}^*(t; \mathbf{x}_k, \mathbf{x}_{k+1}, h_k) = \frac{d^2}{dt^2} \mathbf{q}^*(t; \mathbf{x}_k, \mathbf{x}_{k+1}, h_k) \quad (10)$$

$$\mathbf{q}^*(t; \mathbf{x}_k, \mathbf{x}_{k+1}, h_k) = \begin{bmatrix} 1 & 0 & -\frac{3}{h_k^2} & \frac{2}{h_k^3} \\ 0 & 1 & -\frac{2}{h_k} & \frac{1}{h_k^2} \\ 0 & 0 & \frac{3}{h_k^2} & -\frac{2}{h_k^3} \\ 0 & 0 & -\frac{1}{h_k} & \frac{1}{h_k^2} \end{bmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix} \quad (11)$$

$$\tilde{J}_v^*(\mathbf{x}_k, \mathbf{x}_{k+1}, h_k) = \begin{pmatrix} \mathbf{q}_k \\ \dot{\mathbf{q}}_k \\ \mathbf{q}_{k+1} \\ \dot{\mathbf{q}}_{k+1} \\ \bar{Y}_1 \end{pmatrix}^T \begin{bmatrix} \bar{R}_1 \frac{12}{h_k^3} & \bar{R}_1 \frac{6}{h_k^2} & -\bar{R}_1 \frac{12}{h_k^3} & \bar{R}_1 \frac{6}{h_k^2} & 0 \\ \bar{R}_1 \frac{6}{h_k^2} & \bar{R}_1 \frac{4}{h_k} & -\bar{R}_1 \frac{6}{h_k^2} & \bar{R}_1 \frac{2}{h_k} & \bar{R}_1 \\ -\bar{R}_1 \frac{12}{h_k^3} & -\bar{R}_1 \frac{6}{h_k^2} & \bar{R}_1 \frac{12}{h_k^3} & -\bar{R}_1 \frac{6}{h_k^2} & 0 \\ \bar{R}_1 \frac{6}{h_k^2} & \bar{R}_1 \frac{2}{h_k} & -\bar{R}_1 \frac{6}{h_k^2} & \bar{R}_1 \frac{4}{h_k} & -\bar{R}_1 \\ 0 & \bar{R}_1 & 0 & -\bar{R}_1 & \bar{R}_1 h_k \end{bmatrix} \begin{pmatrix} \mathbf{q}_k \\ \dot{\mathbf{q}}_k \\ \mathbf{q}_{k+1} \\ \dot{\mathbf{q}}_{k+1} \\ \bar{Y}_1 \end{pmatrix} \quad (12)$$

証明は $e^{At} = \begin{bmatrix} I & It \\ 0 & I \end{bmatrix}$ に注意しつつ、文献[13]等を参照すれば得られる。

ところで、本制御則は、経路点間でのモード不変条件を陽に考慮するものではない。したがって、この制御則を用いた結果制約条件が破られる、つまり経路点間で予期せぬモード変化が生じてしまう可能性は十分にある。そこで、有向グラフを構築する過程において、上述の制御則を用いた結果モード不変条件が破られてしまうような経路点の組合せは遷移不能と見なすこととする。幸い、経路点間の状態軌道は式(9)より解析的に得られるので、中間軌道上のいくつかの点をサンプルしてモード不変条件をチェックすることにより、十分高い信頼性で不正な経路点組を振るいにかけることが可能である。

また、本手法では多くの経路点の組み合わせについて問題1の解である経路点間制御入力を計算する必要がある。このために汎用の非線形最適化手法を用いて制御入力を計算しては、いくらオフライン設計であっても計算時間がかかりすぎてしまう。これに対し本節で述べた方法を用いれば、近似解ながらも非常に短時間で制御入力値を得ることができる。さらに、3.6節で述べるように、この利点はオンラインでのロバスト性の向上にも役立てられる。

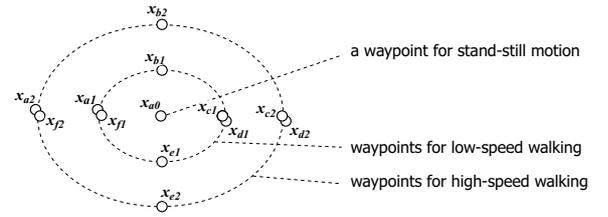
3.3 有向グラフの設計

前節において、与えられた経路点間を近似最適に遷移する制御則の生成方法を述べた。続いて本節では、状態空間上に経路点を配置し、モデル予測制御の規則にもとづいて経路点同士を結ぶ有向エッジを生成することによって有向グラフを構築する方法を述べる。有向グラフ構築の流れを Fig. 3 に示す。

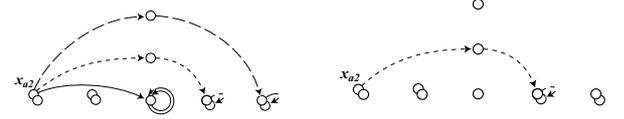
初めに、状態空間上の所定の位置に経路点を配置する (Fig. 3(1))。歩行運動などの動作は経路点の連なり、すなわち経路として表現される。このため、経路点間の軌道が良好であっても、経路点自体が適切に配置されていなければ総体としての運動が良好になることは望めない。したがって、状態空間上にどのように経路点を配置するかが非常に重要となる。本論文の目的である歩行ロボットの変速歩行は、静止動作および異なる複数通りのスピードでの定常歩行動作と(これらを主動作と呼ぶ)、これらの間の切り替え動作から成る。そこで本論文では個々の主動作を良好に実現することを目標として経路点の配置および経路点間の遷移時間の設計を行う。

上述の方針に従い、まずロボットが所定の歩幅と周期で定常歩行を行うための経路点の設計方法を述べる。問題を簡単にするために、以下の仮定を設ける。

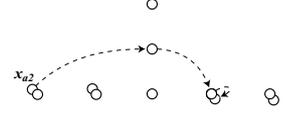
[仮定 2]



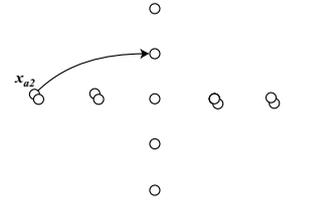
(1) Place waypoints for a stop motion and steady walk motions.



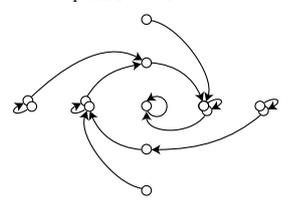
(2a) Candidate paths starting from x_{a2} .



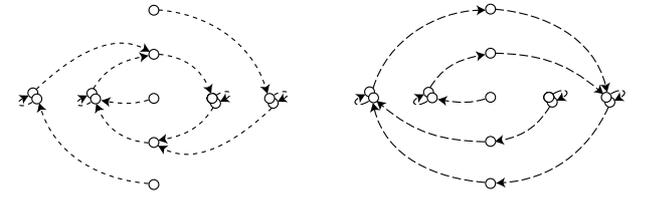
(2b) The optimal path starting from x_{a2} under the cost function $J_{\text{path}}(\cdot; \gamma_0)$.



(2c) Mark the first edge of the optimal path as a directed edge of the graph.



(2d) Repeat step 2a-2c for every other waypoints.



(3) Do 2a-2d under the cost function $J_{\text{path}}(\cdot; \gamma_1)$ and $J_{\text{path}}(\cdot; \gamma_2)$.

Fig. 3 Design flow of the directed graph.

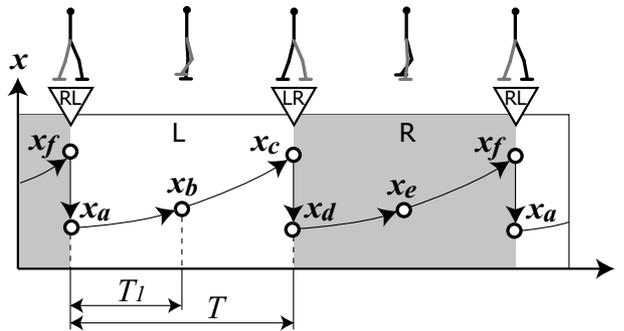


Fig. 4 Waypoint configuration.

(a) 定常歩行動作でのモードの時間変化は $L \rightarrow LR \rightarrow R \rightarrow RL \rightarrow L \rightarrow \dots$ となる。

(b) 定常歩行動作の軌道は左右の脚に関して対称である。

Fig. 4 は仮定 2(a) を踏まえてロボットが定常歩行動作を行うた

めの経路点と遷移時間を図示したものである。図下部のグラフはロボットが定常歩行動作を行う際の状態の時間変化を示した模式図である。一方で図上部の絵は対応する時刻でのロボットの姿勢を表す。モード不変条件を満足するために、モード境界には必ず経路点を設けなければならない(図中の経路点 x_a, x_c, x_d, x_f)。さらに、振り足と地面との間に適度な間隔を確保するために、経路点 x_b, x_e を設ける。 $x_i = (q_i^T, \dot{q}_i^T)^T \quad i \in \{a, b, \dots, f\}$ とすると、仮定 2(b) より次の関係が成り立つ。

$$\begin{aligned} q_d &= Cq_a \\ \dot{q}_d &= C\dot{q}_a \\ q_e &= Cq_b \\ \dot{q}_e &= C\dot{q}_b \\ q_f &= Cq_c \\ \dot{q}_f &= C\dot{q}_c \end{aligned} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (13)$$

また、 x_c と x_d 、 x_f と x_a は次式の関係にある。

$$\begin{aligned} q_d &= q_c, \quad \dot{q}_d = Z_{LR}(q_c)\dot{q}_c \\ q_a &= q_f, \quad \dot{q}_a = Z_{RL}(q_f)\dot{q}_f \end{aligned}$$

結局、決定すべき変数は x_b と x_c 、加えて時間 T_1, T となる。さて、歩幅や着地周期、振り足高度(振り足と地面との間隔)は設計者が明示的に指定できる方が都合が良い。そこでこれら 3 つの設計パラメータを歩容パラメータと呼ぶこととし、これを決定する位置変数 q_b, q_c と着地周期 T は設計者が与え、残る速度変数 \dot{q}_b, \dot{q}_c と T_1 を決定変数として最適化を行う。ここで最適化の評価関数は経路点 a から c へ到達するまでの移動コストとし、前節の結果より近似的に $\tilde{J}_v^*(x_a, x_b, T_1) + \tilde{J}_v^*(x_b, x_c, T - T_1)$ と表せる。具体的な最適化アルゴリズムとしては Downhill-Simplex 法 [14] を用いる。

上述の方法を踏まえ、設計者は初めに必要な複数通りの歩容パラメータを用意する。次に、それぞれの歩容パラメータについて上述の手法により定常歩行動作の経路点と遷移時間を設計する。一方で静止動作は、単一の経路点から成る周期経路によって実現する。静止動作の経路点は特に最適化は行わず、設計者が明示的に与えるものとする。

以上のようにして設計した経路点集合を Fig. 3(1) に示す。図には、ある状態におけるロボットの左足先から右足先へのベクトルを考え、その x 成分を紙面右向き、 y 成分を紙面上向きとなるようにプロットした。この方法では経路点 c と d、f と a は同じ位置にプロットされてしまうので若干ずらして図示してある。また、図は低速歩行 (x_{a1}, \dots, x_{f1}) と高速歩行 (x_{a2}, \dots, x_{f2}) の 2 種類の歩容パラメータを与えた場合を表している。静止動作の経路点は x_{a0} である。

このようにして設計した経路点の集合を \mathcal{X} 、遷移時間の集合を \mathcal{T} とする。加えて組 $(x_0 \in \mathcal{X}, x_1 \in \mathcal{X}, h \in \mathcal{T})$ で、前節で述べた経路点間のモード不変条件を満たすものの集合をエッジ候補集合 \mathcal{E} とする。

次に、モデル予測制御の原則に基づいて経路点同士を結び有向エッジを設計する。まず、ある経路点 x_0 を始点とする経路長 N の経路 (N ステップ経路という) を考える。

$$(x_0, x_1, \dots, x_N, h_0, h_1, \dots, h_{N-1}) \quad (14)$$

$$x_i \in \mathcal{X}, h_i \in \mathcal{T}$$

h_k は経路点 x_k, x_{k+1} 間の遷移時間である。Fig. 3(2a) に例として x_{a2} を始点とする経路長 $N = 3$ の経路のいくつかを示す。次に、 N ステップ経路の望ましさを指標として以下のような経路評価関数を導入する。

$$J_{\text{path}}(x_0, x_1, \dots, x_N, h_0, h_1, \dots, h_{N-1}; \gamma) = \sum_{k=0}^{N-1} \left\{ \tilde{J}_v^*(x_k, x_{k+1}, h_k) - \gamma Pr^x(x_k, x_{k+1}) \right\} \quad (15)$$

ここで右辺中括弧内の第 1 項は経路点間の移動コストを表す。同じく右辺中括弧内の $Pr^x(x_k, x_{k+1})$ は経路点を移動することによるロボットの推進距離を表す。状態変数 x は静止座標系におけるロボットの位置を含まないが、ロボットの接地足が滑らない条件から姿勢の変化より推進距離を計算することができる。すなわち

$$Pr^x(x_k, x_{k+1}) = \begin{cases} p_{l_x}(q_k) - p_{l_x}(q_{k+1}) & \text{if } I = L \\ p_{r_x}(q_k) - p_{r_x}(q_{k+1}) & \text{if } I = R \end{cases} \quad (16)$$

ここで q_i は状態 x_i における位置変数、 I は経路点間のモードである。 p_{l_x}, p_{r_x} の定義は第 2 節を参照されたい。以上より、経路評価関数 (15) は「なるべく小さな消費エネルギーでなるべく大きな推進距離を得よ」という制御目標を定式化している。スカラーパラメータ $\gamma > 0$ を調整することによって消費エネルギーと推進距離のいずれを重視するかを設定することができる。

上述の経路評価関数を用い、各経路点について、その経路点を始点とする N ステップ経路の中で経路評価関数を最小とする最適経路を求める (Fig. 3(2b))。次に、得られた最適経路の一本目の有向エッジに注目し、これを有向グラフを構成する有向エッジとする (Fig. 3(2c))。この操作を全ての経路点について行うことにより、Fig. 3(2d) のような、各経路点を出発する最適経路の 1 本目の有向エッジにより構成される有向グラフが得られる。さて、制御時においてこれらの有向エッジが指し示す経路点へと次々に移動していくことは、経路点へたどり着く度に、改めてそれを始点とする N ステップの最適経路をたどることに他ならない。これはモデル予測制御における receding horizon の概念に対応している。

有向グラフは、いくつかの経路点上を周回する周期経路と、他の経路点からこの周期経路へ合流する経路から成る。周期経路をたどることによって定常歩行動作が、合流経路をたどることによって切り替え動作が実現される。このような有向グラフが経路評価関数とモデル予測制御の原則にもとづいて自動的に得られるのである。

一方、周期経路がどの経路点上に形成されるかは γ の値によって決まる。そこで、異なる値の γ を用いて複数の有向グラフを生成する。これにより、先に設計した複数の歩容パラメータにもとづく経路点集合や静止動作経路点の上に周期経路を形成する有向グラフを得る (Fig. 3(2d), (3))。

ところで、各頂点を始点とする最適経路の計算は、最適経路

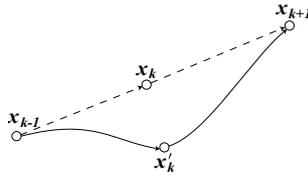


Fig. 5 Feedback mechanism.

が持つ次のような再帰的性質を利用することにより動的計画法の要領で効率的に行うことができる。

$$J_{\text{path}}^*(\mathbf{x}_0; N, \gamma) = \min_{(\mathbf{x}_0, \mathbf{x}_1, h) \in \mathcal{E}} \{J_{\text{path}}(\mathbf{x}_0, \mathbf{x}_1, h; \gamma) + J_{\text{path}}^*(\mathbf{x}_1; N-1, \gamma)\} \quad (17)$$

ここで $J_{\text{path}}^*(\mathbf{x}_0; N, \gamma)$ は \mathbf{x}_0 を始点とする N ステップの最適経路の評価値である。まず全経路点について 1 ステップの最適経路を求める。これはエッジ候補集合 \mathcal{E} の要素数 $|\mathcal{E}|$ に比例する計算量を要する。次に全経路点について 2 ステップの最適経路を求めるが、この時点で 1 ステップの最適経路は得られているので式 (17) を利用することで同じく $|\mathcal{E}|$ に比例する計算量で済む。以下同様に N ステップの最適経路が得られるまで繰り返す。結局この手続き全体を $N|\mathcal{E}|$ に比例する計算量で行うことができる。

3.4 リアルタイムコントローラ

本論文では初期時刻 t_0 におけるロボットの状態は任意の経路点 \mathbf{x}_0 上に設定されているものと仮定する。これに対しリアルタイムコントローラはサンプル時刻 t_k 毎に、その時点で選択されている有向グラフの有向エッジ $(\mathbf{x}_k, \mathbf{x}_{k+1}, h_k)$ を参照して、次のサンプル時刻 t_{k+1} に目標経路点 \mathbf{x}_{k+1} へ到達する制御入力を式 (10) および式 (5) より計算し印加する。ここで $k \geq 1$ については $t_k = t_0 + \sum_{i=0}^{k-1} h_i$ である。

ところで、式 (10) の \mathbf{v}^* に対して \mathbf{x}_k の代わりに $\mathbf{x}(t_k)$ を代入すると、 $\mathbf{v}^*(t; \mathbf{x}(t_k), \mathbf{x}_{k+1}, h_k)$ は状態 $\mathbf{x}(t)$ に対する時刻 t_k 毎のフィードバック制御則となる。これにより、リアルタイムコントローラに外乱やモデル化誤差に対してある程度のロバスト性を与えることができる。Fig. 5 のように、時刻 t_{k-1} に経路点 \mathbf{x}_{k-1} から出発した状態が、外乱などの影響を受けて時刻 t_k において目標経路点 \mathbf{x}_k から若干ずれた状態 \mathbf{x}'_k へ到達したとする。この場合、 \mathbf{x}_k の有向エッジを参照し $((\mathbf{x}_k, \mathbf{x}_{k+1}, h_k)$ とする)、制御入力 $\mathbf{v}(t; \mathbf{x}(t_k), \mathbf{x}_{k+1}, h_k)$ を計算する。これにより、微小な誤差であればサンプル時刻 t_k 毎に修正することができる。既に述べたように、このような制御入力は式 (10) の形で解析的な近似解が得られるので十分にリアルタイム計算可能である。

ところが、 $\mathbf{x}(t_k)$ の値によっては軌道 $\mathbf{x}^*(\mathbf{x}(t_k), \mathbf{x}_{k+1}, h_k)$ がモード不変条件を破る可能性がある。また、大きな外乱が加わった場合などには、次の経路点到達時刻に達する前に予期しないモード変化が生じてしまうこともあり得る。現状ではこれらのような場合にはエラーとなる。

Table 1 Physical Parameters

body mass	20.00 [kg]		
shank mass	10.00 [kg]	thigh mass	10.00 [kg]
body length	0.50 [m]		
shank length	0.40 [m]	thigh length	0.40 [m]
ankle to toe	0.15 [m]	ankle to heel	0.15 [m]

Table 2 Waypoint parameters

	stride[m]	period[s]	clearance[m]
low speed	0.20	0.40	0.03
high speed	0.40	0.50	0.05

Table 3 Weight parameters

steady state	γ_0	0
low speed	γ_1	6000
high speed	γ_2	12000

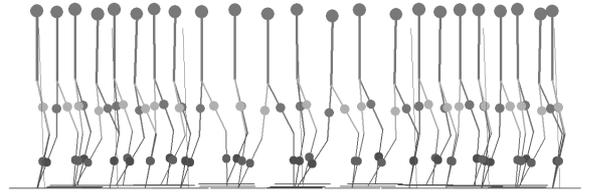


Fig. 6 Gait of bipedal walker.

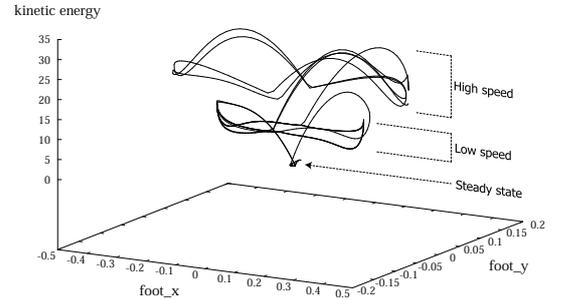


Fig. 7 A phase curve in variable speed locomotion.

4. 数値シミュレーションによる検討

4.1 変速歩行制御

提案手法の有効性を 2 足歩行ロボットの変速制御の数値シミュレーションにより検討した。ロボットの主な物理パラメータを Table 1 に示す。

第 3.3 節で述べた方法により、静止動作と低速歩行、高速歩行の 3 動作を実現するべく経路点を設計した。各動作の歩容パラメータを Table 2 に、経路評価関数の重み係数 γ の値を Table 3 に示す。また、経路長 N は 100 とした。Fig. 6 はロボットが静止状態から低速歩行、高速歩行へ段階的に遷移した後、減速して再び静止状態へ戻る様子を示したものである。初期時刻でのロボットの状態を静止動作用経路点上に設定し、静止動作用の有向グラフを選択した。その後有向グラフを低速歩行用有向グラフに切り替えて歩行を開始し、ロボットが 3 歩歩く毎に高速歩行、低速歩行、静止と有向グラフを切り替えた。Fig. 7 は

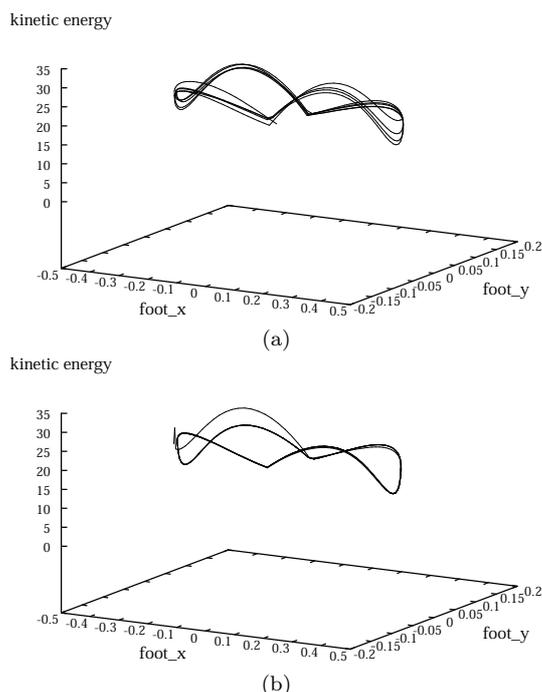


Fig. 8 Phase curves under noises and errors. (a)with noise, (b)with model error.

このときの状態軌道を射影プロットしたものである。高さ方向を運動エネルギーとし、左右足先位置の差分ベクトルの x, y 成分をそれぞれ横方向、奥行き方向にプロットした。

4.2 ロバスト性の検証

外乱やモデル化誤差に対して本手法が有するロバスト性を数値シミュレーションによって定性的に検証した。Fig. 8 は外乱やモデル化誤差の存在する環境でのロボットの状態軌道を Fig. 7 と同様の方法でプロットしたものである。Fig. 8(a) はコントローラとロボットを結び信号にノイズを付加した場合である。具体的には、コントローラが観測するロボットの状態 $x(t)$ およびロボットに加える入力 $u(t)$ の全成分に対し、その絶対値 $\pm 1\%$ の範囲で擬似乱数によるノイズを加えた。一方、(b) はノイズは加えず、動力学計算に用いるロボットモデルの全物理パラメータに対して $\pm 3\%$ の範囲で誤差を与えた場合である。

本手法がこのようなロバスト性を有する仕組みについては第 3.4 節で述べた。しかしながら、上に提示した例以上に大きなノイズやモデル誤差が加わると、経路点間において予期しないモード変化が生じる傾向が強まり、現状のリアルタイムコントローラはこのような予期しないモード変化に対処することができない。将来的には、このようなモード変化が生じた際に、向かうべき経路点をリアルタイムで再探索する仕組みを組み込むことによってロバスト性をさらに向上することが望まれる。

5. おわりに

本論文ではグラフ型モデル予測制御と呼ぶ制御手法を提案し、これを平面 2 足歩行ロボットの変速制御に適用した。本手法では、状態空間上に経路点を配置し、与えられた経路評価関数に基づく最適性およびモデル予測制御の原理を用いて経路点をノー

ドとする有向グラフを構築する。また、経路点間を結び軌道は固定端の最適制御問題の近似的解析解によって与えられる。有効グラフを設計時に事前計算し、リアルタイムでは有効グラフのエッジをたどるように状態を制御することによって、モデル予測制御の有用な性質を保持しつつリアルタイム計算量を劇的に低減することができる。さらに、複数の経路評価関数を用いることによって複数動作の切り替え制御を効果的に実現できる。転倒の回避という見地から述べると、本手法はサンプル時刻毎に誤差を修正するフィードバック制御を実装することによって定常歩行軌道近傍においては外乱などに対するロバスト性の向上を図っている。しかしながら、大域的な転倒回避（上体によるバランス補償、着地位置の調整など）あるいは転倒回復は现阶段で取り扱っていない。また、本研究では歩行運動を実現するための経路評価関数および動的計画法による経路生成法を述べたが、今後、より複雑な動作を実現するためにどのような形の評価関数や経路探索法を用いるべきかを解明しなければならない。次に、今回は経路点間制御に厳密な線形化手法を用いたが、その過程で評価関数を近似したため近似最適なものとなっている。また、つま先立ちや跳躍などの劣駆動モードを取り扱うには厳密な線形化はそのままでは適用できない。走行などの動作を実現するにはこのような劣駆動モードにも対応できるよう手法を拡張する必要がある。

謝辞 本研究にあたり、(財)三豊科学技術振興協会の研究助成から一部助成していただいた。

参考文献

- [1] S.Kajita, F.Kanehiro, K.Kaneko, et al.: "The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation", Proc. of 2003 IROS, pp.239-246, 2001.
- [2] 梶田, 金広, 金子, 藤原, 原田, 横井, 比留川: "分解運動量制御: 運動量と角運動量に基づくヒューマノイドロボットの全身運動生成", 日本ロボット学会誌, Vol.22, No.6, pp.772-778, 2004.
- [3] L.Kovar, M.Gleicher, and F.Pighin, "Motion Graphs", In Proceeding of SIGGRAPH 02, 2002, pp. 473-482
- [4] K.Yamane, Y.Nakamura: "Dynamics Filter - Concept and Implementation of On-Line Motion Generator for Human Figures", Proc. of IEEE ICRA, pp688-695, 2000
- [5] 岡田, 中村: "脳型情報処理を行う情報処理系の多項式設計法とそのヒューマノイドの全身運動生成への応用", 日本ロボット学会誌, Vol.22, No.8, pp.1050-1059, 2004.
- [6] K.Terada, Y.Ohmura, Y.Kuniyoshi: "Analysis and Control of Whole Body Dynamic Humanoid Motion - Towards Experiments on Roll-and-Rise Motion", Proc. of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.
- [7] 大塚: "非線形モデル予測制御の実時間アルゴリズムと機械系への応用", 第 32 回制御理論シンポジウム, pp.1-6, 2003.
- [8] J.Imura: "Optimal control of sampled-data piecewise affine systems", Automatica, Vol.40, No.4, pp.661-669, 2004.
- [9] H.L. Hagenaaers, J. Imura, and H. Nijmeijer: "Approximate continuous-time optimal control in obstacle avoidance by time/space discretization of non-convex state constraints", Proc. of IEEE Conf. on Control Applications, pp.878-883, 2004
- [10] D.Hsu, R.Kindel, J.C.Latombe, S.Rock: "Randomized Kinodynamic Motion Planning with Moving Obstacles", The International Journal of Robotics Research, Vol. 21, No. 3, 233-255 (2002)
- [11] S.M.Lavalle, "From dynamic programming to RRTs: Algorithmic design of feasible trajectories". In A. Bicchi, H. I.

Christensen, and D. Prattichizzo, editors, Control Problems in Robotics, pages 19–37. Springer-Verlag, Berlin, 2002.

- [12] L.E.Kavraki, P.Svestka, J.C.Latombe, and M.Overmars. IEEE Transactions on Robotics and Automation, 12(4):566-580, 1996.
- [13] 片山: “線形システムの最適制御 デスクリプタシステム入門”, 近代科学社.
- [14] W.H.Press, S.A.Teukolsky, W.T.Vetterling, B.P.Flannery: “Numerical Recipes in C The Art of Scientific Computing Second Edition”, Cambridge University Press.

付録 A. モデルの導出

式 (1) の導出過程の概略を示す。はじめに静止座標系における体幹重心の位置を含めた一般化座標を以下のように定義する。

$$\hat{q} := \begin{pmatrix} q_d \\ q \end{pmatrix} \quad q_d = \begin{pmatrix} x \\ y \end{pmatrix}$$

すると \hat{q} に関する運動方程式は次のように表現される。

$$\hat{M}(q)\ddot{\hat{q}} = \hat{h}(q, \dot{q}) + \hat{g}(q) + \hat{J}_i(q)^T \lambda_i + \hat{N}u \quad (\text{A.1})$$

ここで $\hat{M}(q)$ は慣性行列, $\hat{h}(q, \dot{q})$ はコリオリ・遠心項, $\hat{g}(q)$ は重力項である。 \hat{N} はアクチュエータトルクから一般化力を得る定数行列である。また $\hat{J}_i(q)\lambda_i$ ($i \in \{1, r\}$) は床反力項を表し, $J(q)_i$ は \dot{q} から支持足の足先速度を得るヤコビ行列, λ_i は床反力である。両足が一度に床反力を受けるケースは考えないこととする。接触点における加速度が 0 となる条件から

$$\hat{J}_i \ddot{\hat{q}} + \dot{\hat{J}}_i \dot{\hat{q}} = 0 \quad (\text{A.2})$$

式 (A.1) と式 (A.2) より

$$\lambda_i = -(\hat{J}_i \hat{M}^{-1} \hat{J}_i^T)^{-1} (\hat{J}_i \hat{M}^{-1} (\hat{h} + \hat{g} + \hat{N}u) + \dot{\hat{J}}_i \dot{\hat{q}}) \quad (\text{A.3})$$

ここで $\dot{\hat{J}}_i \dot{\hat{q}}$ には q_d, \dot{q}_d は含まれないことに注意すると, λ_i は q, \dot{q}, u の関数となる。したがって

$$\ddot{q} = \begin{bmatrix} 0 & I_{5 \times 5} \end{bmatrix} \ddot{\hat{q}} = \begin{bmatrix} 0 & I_{5 \times 5} \end{bmatrix} \hat{M}^{-1} (\hat{h} + \hat{g} + \hat{J}_i^T \lambda_i + \hat{N}u)$$

も q, \dot{q}, u の関数となり, しかも以下のように書くことができる。

$$\ddot{q}(t) = X_i(q(t))u(t) + Y_i(q(t), \dot{q}(t)) \quad (\text{A.4})$$

続いて状態ジャンプの式を導出する。仮定より, 振り足と床との衝突による速度変化は次式で表せる。

$$\hat{M}(q^-)(\dot{\hat{q}}^+ - \dot{\hat{q}}^-) = \hat{J}_i^T \Lambda_i \quad (i \in \{1, r\}) \quad (\text{A.5})$$

ここで \hat{J}_i は \dot{q} から床に衝突する足の足先速度を得るヤコビ行列, Λ_i は床から足先に作用する撃力である。衝突後に着地した足先の速度が 0 となる条件から

$$\hat{J}_i \dot{\hat{q}}^+ = 0 \quad (\text{A.6})$$

式 (A.5) と式 (A.6) より

$$\Lambda_i = -(\hat{J}_i \hat{M}^{-1} \hat{J}_i^T)^{-1} \hat{J}_i \dot{\hat{q}}^-$$

となる。ここで $\hat{J}_j \dot{\hat{q}} = 0$ (j は i と反対側の足を示す) に注意す

ると, 上式は

$$\Lambda_i = -(\hat{J}_i \hat{M}^{-1} \hat{J}_i^T)^{-1} (J_i - J_j) \dot{\hat{q}}^-$$

と書ける。したがって Λ_i は q, \dot{q}^- の関数である。ゆえに

$$\begin{aligned} \dot{q}^+ &= \begin{bmatrix} 0 & I_{5 \times 5} \end{bmatrix} \dot{\hat{q}}^+ = \begin{bmatrix} 0 & I_{5 \times 5} \end{bmatrix} (\hat{M}^{-1} \hat{J}_i^T \Lambda_i + \dot{\hat{q}}^-) \\ &= \begin{bmatrix} 0 & I_{5 \times 5} \end{bmatrix} \hat{M}^{-1} \hat{J}_i^T \Lambda_i + \dot{q}^- \end{aligned}$$

も q, \dot{q}^- の関数であり, しかも以下のように書ける。

$$\dot{q}^+ = Z_i(q) \dot{q}^- \quad (\text{A.7})$$

を得る。

次にシステムの状態を決定する領域 S_L を記述する。

L モードでは, まず床反力の y 方向成分は必ず正でなければならないから

$$\lambda_{1y} := \begin{bmatrix} 0 & 1 \end{bmatrix} \lambda_1 > 0 \quad (\text{A.8})$$

となる。また, 足が前後の端点を支点として浮き上がらない条件から

$$-w_h \lambda_{1y} \leq \tau_{\text{ankle}1} \leq w_t \lambda_{1y} \quad (\text{A.9})$$

が必要である。ここで w_h, w_t はそれぞれ足首関節からかかと, つま先までの長さである。これらに加えて, 左足, 右足の相対的位置・速度関係を考慮すると S_L は次のように与えられる。

$$\begin{aligned} S_L = \{q, \dot{q}, u \mid &\lambda_{1y} > 0 \wedge (-w_h \lambda_{1y} \leq \tau_{\text{ankle}1} \leq w_t \lambda_{1y}) \wedge \\ &(p_{1y} < p_{ry} \vee (p_{1y} = p_{ry} \wedge v_{1y} < v_{ry}))\} \end{aligned} \quad (\text{A.10})$$

S_{LR} については, 衝突が生じる条件, 激力が y 方向に正となる条件, そして衝突後に左足が離陸する条件より以下のようになる。

$$S_{LR} = \{q, \dot{q} \mid p_{1y} = p_{ry} \wedge v_{1y}^- > v_{ry}^- \wedge \Lambda_{ry} > 0 \wedge v_{1y}^+ > v_{ry}^+\} \quad (\text{A.11})$$

S_R および S_{RL} については上記について左右を反転するだけでよいので省略する。

田崎 勇一 (Yuichi Tazaki)

1980 年 5 月 21 日生。2005 年 3 月東京工業大学大学院情報理工学研究科情報環境学専攻修士課程卒業。同年 4 月同専攻博士課程進学, 現在に至る。歩行ロボットの研究に従事。計測自動制御学会学生会員。

井村 順一 (Jun-ichi Imura)

1964 年 10 月 11 日生。92 年京都大学大学院工学研究科博士課程中途退学。同年, 京都大学工学部助手。96 年広島大学工学部助教授。2001 年東京工業大学大学院情報理工学研究科助教授。2004 年同教授, 現在に至る。98 年 5 月トエンテ大学客員研究員。非線形制御, ハイブリッドシステム論などの研究に従事。博士 (工学)。IEEE, 計測自動制御学会, システム制御情報学会, 日本鉄鋼協会などの会員。