

不連続な状態遷移を考慮した学習最適制御による 歩行軌道の生成手法

佐藤 訓志*¹ 藤本 健治*² 玄 相 昊*³

A Gait Generation Framework via Learning Optimal Control Considering Discontinuous State Transitions

Satoshi Satoh*¹, Kenji Fujimoto*² and Sang-Ho Hyon*³

This paper is concerned with a gait generation framework for legged robots based on iterative learning control (ILC) of Hamiltonian systems. This method allows one to obtain solutions to a class of optimal control problems by iteration of laboratory experiments and, furthermore, precise knowledge of the plant model is not required for it by taking advantage of a symmetric property of Hamiltonian systems. Generally in walking motion, there are discontinuous state transitions caused by collision between the foot and the ground. The proposed framework can also deal with such state transitions without using the parameters of the transition model by combining ILC method and the least-squares. It is applied to a compass-like biped robot to generate optimal gait on the level ground. Some numerical examples demonstrate the effectiveness of the proposed method.

Key Words: Iterative Learning Control, Gait Generation, Biped Robots, Hamiltonian Systems

1. はじめに

近年、歩行ロボットに関する研究が活発に行われ、歩行を実現するための様々な制御手法が提案されている。その代表的な手法として、Zero Moment Point (ZMP) を規範とする手法が知られている [1]~[3]。その基本的な手順は、まず ZMP と呼ばれる姿勢安定性に関する指標を考慮して重心軌道を設計する。次に、逆運動学から各関節角の目標軌道を算出し、PID 制御などで軌道追従を行うものである。また、重力のみを利用して緩斜面を歩き下る受動歩行 [4] という運動が、歩行の力学的原理の理解や高効率歩行の実現に繋がるものと期待され注目を集めている [5]~[8]。適切な閉ループ系を設計し、斜面上と同様の挙動を実現する手法や、歩行中の力学的エネルギーの回復に着目した手法など、受動歩行を規範とした制御手法が数多く提案されている [9]~[11]。これらのアプローチに対し、未知の歩行軌道を獲得するものとして、確率論に基づき試行錯誤的にロボットの駆動パターンを探索する手法 [12] や、消費エネルギーを最小と

する歩行軌道を、最適制御問題として定式化し、計算機を用いた数値計算により獲得する手法 [13] [14] なども提案されている。

一方、筆者らのアプローチは、制御対象をハミルトン力学系 [15] として統一的に扱い、最適制御の意味での最適歩行軌道を学習により獲得するものである。筆者らは、ハミルトン系の変分対称性という力学的対称性を利用した学習最適制御 [16] [17] に基づく、最適な歩行軌道の生成手法を提案してきた [18]~[20]。この方法はまず、与えられた初期状態に対して、駆動入力と周期軌道を生成するための拘束条件から成る評価関数を設計する。次に、試行実験を繰り返すことで、この評価関数を (局所的に) 最小化するという意味で最適な歩行軌道と、これを生成する最適フィードフォワード入力を学習的に生成する。本手法は、 L_2 空間という無限次元空間上の最適制御問題を、変分対称性をうまく利用することで、制御対象の詳細な情報を必要とせずに学習的に解くことができる。本アプローチの特徴を、前述の手法との比較を交えて述べる。まず ZMP 規範の手法は、非常に安定な歩行軌道が生成でき、自由度の大きなロボットの歩行軌道も比較的容易に設計できる。この手法は、もともと安定な歩行軌道を設計するために考案されたものであり、その意味では完成度は高いが、設計された軌道は設計者による人為的なものであり、基本的にエネルギー効率は考慮されていない。次に受動歩行規範の手法の一番の利点は、エネルギー効率が良い歩行が実現できることである。単位質量を単位距離移動させるのに必要なエネルギーを表す Specific resistance をエネルギー効率の評価指標として、様々な歩行制御手法との比較によりその有効

原稿受付 2009 年 11 月 5 日

*¹広島大学大学院工学研究院

*²名古屋大学大学院工学研究科

*³立命館大学ロボティクス学科

*¹Division of Mechanical Systems and Applied Mechanics, Faculty of Engineering, Hiroshima University

*²Department of Mechanical Science and Engineering, Graduate School of Engineering, Nagoya University

*³Department of Robotics, Ritsumeikan University

■ 本論文は学術性で評価されました。

性が示されている [21]. しかしながら, この手法が適用できるロボットの構造に制限があることと, 受動歩行が有効な指標を与えることは間違いないものの, 本来斜面上でしか実現し得ない受動歩行の平地における最適性はいまだ不明である. 確率的アプローチによる手法は, 制御対象の情報を必要とせず未知の軌道が生成できる利点をもつが, 探索空間全域をランダムに探索するため, 自由度が大きなロボットにはそのままでは適用できない. また, 文献 [13] [14] の手法は, 計算機を用いるため様々な定式化を行うことができるが, 制御対象の詳細な情報が必要となる. それに対して本手法は, 様々な構造や自由度をもつ歩行ロボットに対して, 最適な軌道を自動的に生成する統一的手法を与えることができる. さらに力学的特性を利用することで, 制御対象の詳細な情報を利用せずに, 評価関数の最急降下方向のみを探索できる. 扱える評価関数に制限があるため, Specific resistance を直接最小化することは現時点ではできていないが, 駆動入力 L_2 ノルムが最小となる軌道が生成できる.

次に本論文の動機付けとして, 筆者らの過去の結果に関する課題を述べる. 一般に歩行運動は, 着地時に足と地面との衝突により, 角速度に不連続な遷移を伴うが, 筆者らのこれまでの枠組みでは不連続な状態遷移を直接扱うことはできなかった. 文献 [18] [19] では, 生成される軌道を対称歩行軌道という特別な軌道近傍に限定し, この軌道に関する性質をうまく利用することでこの問題を回避していた. そこで本論文では, 学習最適制御と最小二乗法を組み合わせることで, 不連続な状態遷移を扱うことができる手法を提案する. 歩行ロボットの衝突現象の解析には, しばしば角運動量の保存則から導かれる衝突方程式などが用いられるが, 本論文のアプローチでは制御対象のモデルパラメータは利用しないため, このような衝突方程式のモデルも利用できない. そこで提案手法は, 筆者らの過去の結果を用いて着地直前までの歩行軌道の学習を行いながら, さらに各試行実験で得られた実験データを利用し, 最小二乗法により状態遷移写像のヤコビアンを推定することで, 着地時の状態遷移に関する学習を同時に行う. これにより, 制御対象だけでなく, 衝突方程式のモデルパラメータも必要とせずに, 不連続な状態遷移を考慮した学習を行うことが可能となる. 最後に, 適用例としてコンパスロボットモデル [5] を用いたシミュレーションによりその有効性を確認する.

2. 変分対称性を利用した学習最適制御

本章では, ハミルトン系の変分対称性を利用した学習最適制御 [16] [17] について簡単に述べる. 以降では, $\mathcal{D}(\cdot)$ を (\cdot) に関する微分作用素を表すものとし, 時間微分 d/dt と区別するために L_2 空間の Fréchet 微分[†] は δ と表すものとする.

2.1 ハミルトン系と変分対称性

本論文で対象とするハミルトン系 (1) は, 電気-機械系や非ホロノミック系などの応用上重要なクラスのシステムを統一的に扱うことができる [15]. さらに, 後述の変分対称性だけでなく, 受動性やハミルトン関数がリアプノフ関数の候補になるなど, 非線形システムでありながら比較的制御しやすい特徴を持つ.

$$(x_{t^1}, y) = \Sigma(x_{t^0}, u) :$$

$$\begin{cases} \dot{x} = (J - R) \frac{\partial H(x, u, t)}{\partial x}^\top, & x(t^0) = x_{t^0} \\ y = -\frac{\partial H(x, u, t)}{\partial u}^\top \\ x_{t^1} = x(t^1) \end{cases} \quad (1)$$

ここで, $x(t) \in \mathbb{R}^n$, $u(t), y(t) \in \mathbb{R}^m$ はそれぞれシステムの状態, 入力, 出力を表す. また, $H(x, u, t) \in \mathbb{R}$ はハミルトン関数であり, システムの全エネルギーを表している. 構造行列 $J \in \mathbb{R}^{n \times n}$ は歪対称行列であり, 受動性などのシステムの性質を表す. 散逸行列 $R \in \mathbb{R}^{n \times n}$ は半正定対称行列であり, 摩擦などのエネルギー散逸を表している. 本論文では, 有限時間区間 $[t^0, t^1]$ 上でのこの制御系の挙動を考え, この系を $\Sigma: X \times U \rightarrow X \times Y$ の作用素ととらえる. ただし X, U, Y はヒルベルト空間を表し, それぞれ $X = \mathbb{R}^n$, $U = Y = L_2^m[t^0, t^1]$ とする. 固定された初期状態 x_{t^0} の下で, 入力から出力への写像に注目する場合は $\Sigma^{x_{t^0}}: U \rightarrow Y: u \mapsto y$ と表記する.

Σ の変分系 $\delta\Sigma$ とは, システム Σ を作用素と捉えたときの Fréchet 微分を表すシステムである. このシステムは, J, R が定数行列であるとき, 任意の (x_{t^0}, u) に対して次式のような線形のハミルトン系で表せることが知られている [16].

$$(x_{v, t^1}, y_v) = (\delta\Sigma(x_{t^0}, u))(x_{v, t^0}, u_v) :$$

$$\begin{cases} \dot{x} = (J - R) \frac{\partial H(x, u, t)}{\partial x}^\top, & x(t^0) = x_{t^0} \\ \dot{x}_v = (J - R) \frac{\partial H_v(x, u, x_v, u_v, t)}{\partial x_v}^\top, & x_v(t^0) = x_{v, t^0} \\ y_v = -\frac{\partial H_v(x, u, x_v, u_v, t)}{\partial u_v}^\top, \\ x_{v, t^1} = x_v(t^1) \end{cases} \quad (2)$$

ただし, ハミルトン関数 $H_v(x, u, x_v, u_v, t)$ は次式である.

$$H_v(x, u, x_v, u_v, t) = \frac{1}{2} \begin{bmatrix} x_v \\ u_v \end{bmatrix}^\top \frac{\partial^2 H(x, u, t)}{\partial (x, u)^2} \begin{bmatrix} x_v \\ u_v \end{bmatrix}$$

随伴系は最適制御において重要な役割を果たし, 本論文においても後述する学習則の導出過程で, その入出力を計算する必要が生じる. しかしながら, 随伴系の計算には一般に制御対象のモデルが必要となる. この問題を解決する一つの方法として, 文献 [17] の学習最適制御手法では, ハミルトン系の変分対称性 [16] を利用している. 変分対称性とは, 変分系のダイナミクス (2) と, その随伴系である変分随伴系のダイナミクスとを関連付けるものであり, この性質を利用することで, 変分随伴系の入出力が制御対象のモデルを用いることなく計算できる.

補題 1 文献 [16] ハミルトン系 Σ (1) について, J, R は定数行列であり, 正則行列 $T \in \mathbb{R}^{n \times n}$ が存在して次式を満たすと仮定する. ただし, I は単位行列を表す.

$$J = -TJ T^{-1}, \quad R = TR T^{-1} \quad (3)$$

$$\frac{\partial^2 H(x, u, t)}{\partial (x, u)^2} = \begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix} \frac{\partial^2 H(x, u, t)}{\partial (x, u)^2} \begin{bmatrix} T^{-1} & 0 \\ 0 & I \end{bmatrix} \quad (4)$$

[†]作用素 f の Fréchet 微分とは, 各 ξ に対して $\delta f(\xi)(\eta) = f(\xi + \eta) - f(\xi) + o(\|\eta\|)$ を満たす η に関して線形な作用素をいう.

さらに $J - R$ が正則ならば, 変分随伴系は変分系 (2) を時間反転した状態空間表現 (5) をもつ. これを変分対称性と呼ぶ.

$$(x_{a,t^0}, y_a) = (\delta\Sigma(x_{t^0}, u))^*(x_{a,t^1}, u_a) : \begin{cases} \dot{x} = (J - R) \frac{\partial H(x, u, t)}{\partial x}, & x(t^0) = x_{t^0} \\ \dot{\bar{x}}_v = -(J - R) \frac{\partial H_v(x, u, \bar{x}_v, u_a, t)}{\partial \bar{x}_v}, \\ \bar{x}_v(t^1) = -(J - R)T x_{a,t^1} \\ y_a = -\frac{\partial H_v(x, u, \bar{x}_v, u_a, t)}{\partial u_a} \\ x_{a,t^0} = -T^{-1}(J - R)^{-1}\bar{x}_v(t^0) \end{cases} \quad (5)$$

定理 1 文献 [17] 補題 1 の仮定を満たす式 (1) のハミルトン系を考える. 二つの入力 $v, w \in U$ に対して, 系の状態の軌跡がそれぞれ $\phi(t), \psi(t) \in X$ で与えられ, かつ次式を満たすとす.

$$\mathcal{R} \left(\frac{\partial^2 H(x, u, t)}{\partial(x, u)^2} \Big|_{\substack{x = \phi \\ u = v}} \right) = \frac{\partial^2 H(x, u, t)}{\partial(x, u)^2} \Big|_{\substack{x = \psi \\ u = w}} \quad (6)$$

ただし, \mathcal{R} は次式で定義される時間区間 $[t^0, t^1]$ 上での時間反転作用素を表している.

$$\mathcal{R}(u)(t) = u(t^1 - t + t^0), \quad \forall t \in [t^0, t^1] \quad (7)$$

このとき, 次の関係が成り立つ.

$$\mathcal{S}(\delta\Sigma(\phi(t^0), v))^* = (\delta\Sigma(\psi(t^0), w))\mathcal{S}, \quad (8)$$

$$(\delta\Sigma^{\phi(t^0)}(v))^* = \mathcal{R}(\delta\Sigma^{\psi(t^0)}(w))\mathcal{R} \quad (9)$$

ただし, $\mathcal{S} : X \times U \rightarrow X \times U$ は次式で定義される.

$$\mathcal{S}(x_{t^0}, u) := -(J - R)T x_{t^0}, \mathcal{R}(u)$$

注意 1 配位座標 q と一般化速度 \dot{q} が, $\forall t \in [t^0, t^1]$ において

$$q(t) = q(t^1 - t + t^0), \quad \dot{q}(t) = -\dot{q}(t^1 - t + t^0) \quad (10)$$

を満たす軌道は, 時間区間 $[t^0, t^1]$ の中心 $t = (t^0 + t^1)/2$ に関して対称な運動を表すため, 本論文では対称軌道と呼ぶ. ある入力 v に対する状態の軌跡 ϕ が対称軌道ならば, $w \equiv v, \psi \equiv \phi$ のもとで定理 1 の条件 (6) を満たすため, 次式が成立する.

$$\mathcal{S}(\delta\Sigma(\phi(t^0), v))^* = (\delta\Sigma(\phi(t^0), v))\mathcal{S}, \quad (11)$$

$$(\delta\Sigma^{\phi(t^0)}(v))^* = \mathcal{R}(\delta\Sigma^{\phi(t^0)}(v))\mathcal{R} \quad (12)$$

2.2 学習則の導出

文献 [17] では, 評価関数を最小化する入力を勾配法を用いて逐次的に求めることで, 最適制御問題の学習的解法を与えている.

式 (1) の制御対象 Σ と, 評価関数 $\hat{\Gamma}(x_{t^0}, u, x_{t^1}, y) : X^2 \times U \times Y \rightarrow \mathbb{R}$ を考える. $(x_{t^1}, y) = \Sigma(x_{t^0}, u)$ に注意して, $\Gamma(x_{t^0}, u) : X \times U \rightarrow \mathbb{R} := \hat{\Gamma}(x_{t^0}, u, \Sigma(x_{t^0}, u))$ と書くと, 評価関数の Fréchet 微分は次式のように計算できる.

$$\begin{aligned} \delta\hat{\Gamma}(x_{t^0}, u, x_{t^1}, y)(\delta x_{t^0}, \delta u, \delta x_{t^1}, \delta y) &= \delta\hat{\Gamma}((x_{t^0}, u), \Sigma(x_{t^0}, u))((\delta x_{t^0}, \delta u), (\delta\Sigma(x_{t^0}, u)(\delta x_{t^0}, \delta u))) \\ &= \langle \nabla\hat{\Gamma}((x_{t^0}, u), \Sigma(x_{t^0}, u)), \begin{pmatrix} \text{id} \\ \delta\Sigma(x_{t^0}, u) \end{pmatrix} (\delta x_{t^0}, \delta u) \rangle_{X^2 \times U \times Y} \\ &= \langle (\text{id}, (\delta\Sigma(x_{t^0}, u))^*) \nabla\hat{\Gamma}(x_{t^0}, u, x_{t^1}, y), (\delta x_{t^0}, \delta u) \rangle_{X \times U} \\ &= \underbrace{\langle \nabla_{x_{t^0}} \hat{\Gamma} + \pi_X \circ (\delta\Sigma(x_{t^0}, u))^* (\nabla_{x_{t^1}} \hat{\Gamma}, \nabla_y \hat{\Gamma}), \delta x_{t^0} \rangle_X}_{=:\nabla_{x_{t^0}} \Gamma(x_{t^0}, u)} \\ &\quad + \underbrace{\langle \nabla_u \hat{\Gamma} + \pi_U \circ (\delta\Sigma(x_{t^0}, u))^* (\nabla_{x_{t^1}} \hat{\Gamma}, \nabla_y \hat{\Gamma}), \delta u \rangle_U}_{=:\nabla_u \Gamma(x_{t^0}, u)} \end{aligned} \quad (13)$$

ただし id は恒等写像を表し, $\pi_{(\cdot)}$ は (\cdot) への射影写像を表す. また Riesz の表現定理より, 式 (13) を満たす関数 $\nabla\hat{\Gamma}(x_{t^0}, u, x_{t^1}, y) \equiv (\nabla_{x_{t^0}} \hat{\Gamma}^\top, \nabla_u \hat{\Gamma}^\top, \nabla_{x_{t^1}} \hat{\Gamma}^\top, \nabla_y \hat{\Gamma}^\top)^\top$ の存在が保証されている. 式 (13) 中の $\nabla_{x_{t^0}} \Gamma(x_{t^0}, u), \nabla_u \Gamma(x_{t^0}, u)$ はそれぞれ評価関数の初期状態と入力に関する勾配を表していることから, 入力を次式のように更新すれば評価関数の値を最も減少させることができる. ただし, 添字 i は i 回目の試行実験を表し, $K_{(i)}$ は学習ゲインを表す適当な正定行列である.

$$u_{(i+1)} = u_{(i)} - K_{(i)} \nabla_u \Gamma(x_{t^0(i)}, u_{(i)}) \quad (14)$$

しかしながら, 更新則 (14) は変分随伴系 $(\delta\Sigma)^*$ を含むため, このままでは制御対象のモデルなしでは計算できない. そこで, 式 (8) より $(\delta\Sigma)^*$ を変分系 $\delta\Sigma$ を用いて表し, 微小正定数 ϵ を用いて, 次式のように制御対象 Σ の出力の差分で近似する.

$$\begin{aligned} &\mathcal{S}(\delta\Sigma(x_{t^0}, u))^* (\nabla_{x_{t^1}} \hat{\Gamma}, \nabla_y \hat{\Gamma}) \\ &= \delta\Sigma(\psi(t^0), w) (- (J - R)T \nabla_{x_{t^1}} \hat{\Gamma}, \mathcal{R}(\nabla_y \hat{\Gamma})) \\ &\approx \frac{\Sigma(\psi(t^0) - \epsilon(J - R)T \nabla_{x_{t^1}} \hat{\Gamma}, w + \epsilon \mathcal{R}(\nabla_y \hat{\Gamma})) - \Sigma(\psi(t^0), w)}{\epsilon} \end{aligned} \quad (15)$$

式 (15) より, 変分随伴系の入出力が, 2 回の実験による出力の差から近似的に計算でき, これと式 (14) より次の実験に用いる入力を得る. この手順を繰り返すことで, 制御対象のモデルを用いることなく, 最適制御問題を学習的に解くことができる.

3. 制御対象

本論文の制御対象は, Fig. 1 に示すコンパスロボットモデル

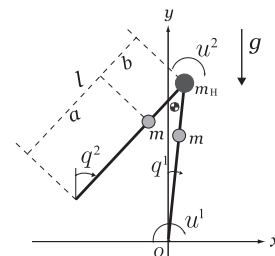


Fig. 1 The compass gait biped

Table 1 Parameters

Notation	Meaning	Unit
$m_H = 10$	hip mass	kg
$m = 5$	leg mass	kg
$a = 0.5$	length from m to foot	m
$b = 0.5$	length from hip to m	m
$l = a + b$	total leg length	m
$g = 9.807$	gravity acceleration	m/s ²

Table 2 Some notations

Notation	Meaning
$q := (q^1, q^2)^\top$	angles of legs
$\dot{q} := (\dot{q}^1, \dot{q}^2)^\top$	angular velocities of legs
$p := (p^1, p^2)^\top$	generalized momentum
$x := (q^\top, p^\top)^\top$	state
$Q := (q^\top, \dot{q}^\top)^\top$	angles and their velocities
$(q_{t^0}, p_{t^0}) := (q(t^0), p(t^0))$	initial state
$(q_{t^1}, p_{t^1}) := (q(t^1), p(t^1))$	terminal state
$(\cdot)_{-/+}$	just before/after transfer
Note that $x_- \equiv x_{t^1}$.	

ル [5] である。ここでは、このモデルに関する仮定の中で特に重要と考えられるものだけを示す。詳細は文献 [5] に準ずるものとする。

仮定 1 衝突は完全非弾性衝突とし、滑りは生じない。

仮定 2 衝突時、支持脚と遊脚は瞬間的に入れ替わる。

仮定 3 歩行運動に影響を与えない伸縮機構を仮定し、遊脚が支持脚とすれ違う際に起こる地面との干渉は無視できる。

ロボットの物理パラメータを Table 1 に示す。Fig. 1 中の q^1, q^2 はそれぞれ鉛直方向からの支持脚と遊脚の角度を表し、 u^1, u^2 はそれぞれ支持脚と遊脚に作用するトルクを表している。本論文では、ロボットのダイナミクスをハミルトン系で表した際に、入出力の対応を簡単にするために、文献 [5] と入力定義が異なる。文献 [5] において用いられているトルクを v^1, v^2 とすると、 $(u^1, u^2) \equiv (v^1 + v^2, -v^2)$ となっている。Table 2 に本論文で用いられる状態に関する表記をまとめる。

一般的な機械系は、式 (1) のハミルトン系において、状態を $x = (q^\top, p^\top)^\top \in \mathbb{R}^{2m}$ 、出力を $y = q \in \mathbb{R}^m$ とし、次式のハミルトン関数、構造行列および散逸行列として表現される [17]。

$$H(q, p, u) = \frac{1}{2} p^\top M(q)^{-1} p + V(q) - u^\top q, \quad (16)$$

$$J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix} \quad (17)$$

ただし、 $M(q) > 0 \in \mathbb{R}^{m \times m}$ は慣性行列であり、 $p := M(q)\dot{q} \in \mathbb{R}^m$ は一般化運動量、半正定行列 $D \in \mathbb{R}^{m \times m}$ は摩擦項、スカラー関数 $V(q)$ は位置エネルギーを表す。Fig. 1 のコンパスロボットは、式 (16) のハミルトン関数 $H(q, p, u)$ 、摩擦項 $D = 0$ 、次式の慣性行列と位置エネルギーを持つ機械系として記述できる。

$$M(\alpha(q)) = \begin{bmatrix} m_H l^2 + ma^2 + ml^2 & -mbl \cos(\alpha(q)) \\ -mbl \cos(\alpha(q)) & mb^2 \end{bmatrix}$$

$$\alpha(q) := q^1 - q^2$$

$$V(q) = \{(m_H l + ma + ml) \cos q^1 - mb \cos q^2\} g \quad (18)$$

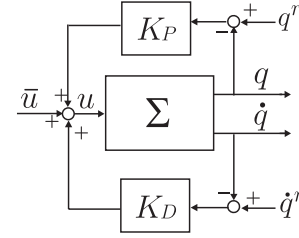


Fig. 2 Feedback system

本論文では、遊脚が地面と衝突した直後から、次に初めて遊脚が地面と衝突するまでの運動を 1 歩の歩行と考える。地面との衝突の瞬間に、ロボットの角速度は不連続に遷移する。この衝突直前と直後の角速度の関係を表す方程式を本論文では衝突方程式と呼ぶ。仮定 1 と仮定 2 から両脚支持期は存在しないため、衝突方程式は角運動量保存則から導出できる。ただし、誌面の都合上本論文では結果のみを示す。導出の詳細は文献 [5] を参照されたい。衝突直後の角度 q_+ は、支持脚と遊脚の交換を表す行列 C と衝突直前の角度 q_- より次のように書ける。

$$q_+ = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} q_- =: Cq_- \quad (19)$$

式 (19) の関係を用いれば、衝突方程式は次式で与えられる。

$$\Pi_+(\alpha(q_-))\dot{q}_+ = \Pi_-(\alpha(q_-))\dot{q}_- \quad (20)$$

ただし、 $\Pi_-(\alpha(q_-))$ と $\Pi_+(\alpha(q_-))$ は次式で与えられる。

$$\Pi_-(\alpha(q_-)) = \begin{bmatrix} (m_H l^2 + 2mal) \cos(\alpha(q_-)) - mab & -mab \\ -mab & 0 \end{bmatrix},$$

$$\Pi_+(\alpha(q_-)) = \begin{bmatrix} ml(l - b \cos(\alpha(q_-))) + ma^2 + m_H l^2 & mb(b - l \cos(\alpha(q_-))) \\ -mbl \cos(\alpha(q_-)) & mb^2 \end{bmatrix} \quad (21)$$

ここで、状態遷移行列を $\bar{\Pi}(\alpha(q_-)) := \Pi_+^{-1} \Pi_-$ と定義すると、式 (20) の衝突方程式は次式のように書き直すことができる。

$$\dot{q}_+ = \bar{\Pi}(\alpha(q_-))\dot{q}_- \quad (22)$$

学習最適制御によるフィードフォワード入力は、制御対象にそのまま入力せずに、ローカルなフィードバックで系を漸近安定化したあとに参照入力として加えることが多い。その際、閉ループ系がハミルトン系の構造を保存する必要がある。例えば文献 [22] では一般化正準変換としてそのようなフィードバックの設計法が与えられている。一般的な機械系の場合は、Fig. 2 に示すような PD フィードバックが一般化正準変換となっている [16]。PD フィードバックを加えた新しい入力を

$$u = -K_P q - K_D \dot{q} + \bar{u} \quad (23)$$

とすると、閉ループ系は次式で表されるハミルトン関数 H_c 、構造行列 J_c 、散逸行列 R_c を持つ新たなハミルトン系となる。

$$H_c(q, p, \bar{u}) = \frac{1}{2} p^\top M(q)^{-1} p + V(q) + \frac{1}{2} \bar{q}^\top K_P q - \bar{u}^\top q,$$

$$J_c = J, \quad R_c = \text{diag}(0, D + K_D)$$

ただし、ここでは原点の漸近安定化が目的であるため、 q と \dot{q} への参照軌道を表す q^r と \dot{q}^r はそれぞれ $q^r \equiv \dot{q}^r \equiv 0$ としている。この系は、もし慣性行列 M が q に依存しなければ、補題 1 において $T = \text{diag}(I, -I)$ と選ぶことで、仮定 (3), (4) が満たされる。さらに、式 (18) のコンパスロボットのように慣性行列が q に依存する場合でも、PD ゲイン K_P, K_D を十分大きくとることで、近似的にこの仮定を満たすことが文献 [16] に示されている。そこで本論文でも、Fig. 2 のように PD フィードバック (23) を施した閉ループ系を考え、新しい入力 \bar{u} を学習入力として、次章で更新則を導出していく。

4. 提案手法

本章では主結果である、不連続な状態遷移を考慮した最適歩容生成法を提案する。まず 4.1 節で、与えられた初期状態に対して、周期的で入力が最小となるような最適軌道を生成するための評価関数を定義し、2.2 節で述べた入力の更新則を導出する。この評価関数は、初期状態と衝突直後の状態が等しくなる 1 周期軌道の必要条件を表す拘束項と、入力の L_2 ノルムを小さくするための項から成る。しかしながら、このままでは更新則の計算に状態遷移写像と慣性行列に関する情報が必要となる。そこで 4.2 節では、歩行運動を歩行開始から衝突直前までの連続的部分と、着地による状態遷移という不連続な部分に分けて考え、前章の学習最適制御による連続部分に関する学習と、最小二乗法による状態遷移写像に関する学習とを組み合わせた手法を提案する。これにより、状態遷移写像と慣性行列の情報を必要とせず、不連続な状態遷移を考慮した学習が可能となる。

4.1 学習最適制御に基づく最適歩容生成

まず、次式のような評価関数を定義する。

$$\hat{\Gamma}_Q(Q_{t_0}, \bar{u}, Q_{t_1}) := \frac{1}{2} (Q_{t_0} - \Phi(Q_{t_1}))^\top \Lambda_x (Q_{t_0} - \Phi(Q_{t_1}))$$

$$+ \frac{1}{2} \int_{t_0}^{t_1} \bar{u}(\tau)^\top \Lambda_{\bar{u}} \bar{u}(\tau) d\tau$$

$$= \frac{1}{2} (\Psi_0(x_{t_0}) - \Psi_1(x_{t_1}))^\top$$

$$\times \Lambda_x (\Psi_0(x_{t_0}) - \Psi_1(x_{t_1}))$$

$$+ \frac{1}{2} \int_{t_0}^{t_1} \bar{u}(\tau)^\top \Lambda_{\bar{u}} \bar{u}(\tau) d\tau$$

$$=: \hat{\Gamma}_x(x_{t_0}, \bar{u}, x_{t_1}) \quad (24)$$

ただし、 Λ_x と $\Lambda_{\bar{u}}$ は重みを表す適当な正定行列である。

注意 2 ここでは歩容を生成する学習入力 \bar{u} の L_2 ノルム $\|\bar{u}\|_{L_2}$ を最小化しているが、式 (23) のローカルなフィードバックを含めた入力 u のノルムを直接最小化することはできない。しかしながら、例えば評価関数 (24) に $\kappa_y/2 \int_{t_0}^{t_1} y(\tau)^\top y(\tau) d\tau + \kappa_{\dot{y}}/2 \int_{t_0}^{t_1} \dot{y}(\tau)^\top \dot{y}(\tau) d\tau$ のような拘束を加え、重みを表す正定数 $\kappa_y, \kappa_{\dot{y}}$ を適切に定めることで、周期軌道の拘束を満たしながら $\|q\|_{L_2}, \|\dot{q}\|_{L_2}$ の大きさを抑えることができる ($y = q$ であること

に注意)。これにより、間接的に $\|u\|_{L_2}$ を抑えることができる。

$\Phi(Q_{t_1})$ は脚の切り替えを考慮した衝突直後の角度と角速度であり、次式で定義される (状態の表記 Q は、Table 2 を参照)。

$$\Phi(Q_{t_1}) := \begin{bmatrix} q_+ \\ \dot{q}_+ \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & \bar{\Pi}(\alpha(q_-)) \end{bmatrix} Q_{t_1} \quad (25)$$

x と Q の関係式は、次式のように書ける。

$$x = \begin{bmatrix} q \\ p \end{bmatrix} = \begin{bmatrix} q \\ M(q)\dot{q} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & M(q) \end{bmatrix} Q \quad (26)$$

式 (25), (26) から $\Psi_0(x_{t_0})$ と $\Psi_1(x_{t_1})$ は次式で定義される。

$$\Psi_0(x_{t_0}) := \begin{bmatrix} I & 0 \\ 0 & M(q_{t_0})^{-1} \end{bmatrix} x_{t_0}$$

$$\Psi_1(x_{t_1}) := \begin{bmatrix} C & 0 \\ 0 & \bar{\Pi}(\alpha(q_-))M(q_-)^{-1} \end{bmatrix} x_{t_1} \quad (27)$$

評価関数 (24) に対して、勾配法に基づく更新則を導出する。まず $\hat{\Gamma}_x(x_{t_0}, \bar{u}, x_{t_1})$ の Fréchet 微分は次式のように計算できる。

$$\delta \hat{\Gamma}_x(x_{t_0}, \bar{u}, x_{t_1})(\delta x_{t_0}, \delta \bar{u}, \delta x_{t_1}, \delta y)$$

$$= \langle (\Lambda_x (\Psi_0(x_{t_0}) - \Psi_1(x_{t_1})), \Lambda_{\bar{u}} \bar{u}),$$

$$(\delta \Psi_0(x_{t_0})(\delta x_{t_0}), \delta \bar{u})_{X \times U}$$

$$+ \langle (\Lambda_x (\Psi_0(x_{t_0}) - \Psi_1(x_{t_1})), 0),$$

$$(-\delta \Psi_1(x_{t_1})(\delta x_{t_1}), \delta y) \rangle_{X \times Y}$$

$$= \langle (\delta \Psi_0(x_{t_0})^* \Lambda_x (\Psi_0(x_{t_0}) - \Psi_1(x_{t_1})), \Lambda_{\bar{u}} \bar{u}) + \delta \Sigma(x_{t_0}, \bar{u})^*$$

$$(-\delta \Psi_1(x_{t_1})^* \Lambda_x (\Psi_0(x_{t_0}) - \Psi_1(x_{t_1})), 0), (\delta x_{t_0}, \delta \bar{u}) \rangle_{X \times U}$$

$$=: \langle (\nabla_{x_{t_0}} \hat{\Gamma}, \nabla_{\bar{u}} \hat{\Gamma}) + \delta \Sigma(x_{t_0}, \bar{u})^* (\nabla_{x_{t_1}} \hat{\Gamma}, 0), (\delta x_{t_0}, \delta \bar{u}) \rangle_{X \times U} \quad (28)$$

次に、定理 1 を用いて変分随伴系の出力 $\delta \Sigma(x_{t_0}, \bar{u})^* (\nabla_{x_{t_1}} \hat{\Gamma}, 0)$ を計算する。文献 [17] では、制御対象が一般的な機械系の場合に、定理 1 の条件を満たす軌道 ψ の生成法を与えている。ある軌道 $\phi(t)$ が対称軌道でない場合は、Fig. 2 の閉ループ系において PD フィードバックの参照軌道として、 $\phi(t)$ の時間反転軌道

$$(q^r, \dot{q}^r) = (\mathcal{R}(q|_{x=\phi}), -\mathcal{R}(\dot{q}|_{x=\phi})) \quad (29)$$

を与えることで、 $\psi(t)$ を生成する。また、そのとき得られる入力を w とする。このとき式 (8), (15) より、

$$\delta \Sigma(x_{t_0}, \bar{u})^* (\nabla_{x_{t_1}} \hat{\Gamma}, 0)$$

$$= \mathcal{S}^{-1} \delta \Sigma(\psi(t^0), w) (- (J_c - R_c) T \nabla_{x_{t_1}} \hat{\Gamma}, 0)$$

$$\approx \mathcal{S}^{-1} \frac{1}{\epsilon} \left(\Sigma(\psi(t^0) - \epsilon (J_c - R_c) T \nabla_{x_{t_1}} \hat{\Gamma}, w) - \Sigma(\psi(t^0), w) \right) \quad (30)$$

となり、変分随伴系の出力信号 $\delta \Sigma(x_{t_0}, \bar{u})^* (\nabla_{x_{t_1}} \hat{\Gamma}, 0)$ が、 w, ψ 周りの変分系の近似計算により求まる。

結局、式 (28) (30) より、更新則は次式のように求まる。

$$\begin{cases}
x_{t^0(3i+1)} = (q_{t^1(3i)}^\top, -p_{t^1(3i)}^\top)^\top \\
\bar{u}_{(3i+1)} \equiv 0 \\
(q_{(3i+1)}^r, \dot{q}_{(3i+1)}^r) = (\mathcal{R}(q_{(3i)}), -\mathcal{R}(\dot{q}_{(3i)})) \\
x_{t^0(3i+2)} = x_{t^0(3i+1)} + \epsilon_{(i)}(J_c - R_c)T\delta\Psi_1(x_{t^1(3i)})^* \Lambda_x \\
\quad \times (\Psi_0(x_{t^0(3i)}) - \Psi_1(x_{t^1(3i)})) \\
\bar{u}_{(3i+2)} \equiv 0 \\
(q_{(3i+2)}^r, \dot{q}_{(3i+2)}^r) = (\mathcal{R}(q_{(3i)}), -\mathcal{R}(\dot{q}_{(3i)})) \\
x_{t^0(3i+3)} = x_{t^0(3i)} \\
\bar{u}_{(3i+3)} = \bar{u}_{(3i)} - K_{(i)} \left(\Lambda_{\bar{u}} \bar{u}_{(3i)} + \frac{1}{\epsilon_{(i)}} \mathcal{R}(y_{(3i+2)} - y_{(3i+1)}) \right) \\
(q_{(3i+3)}^r, \dot{q}_{(3i+3)}^r) \equiv (0, 0)
\end{cases} \quad (31)$$

ただし、 $\epsilon_{(i)}$ は適当な微小正定数、学習ゲイン $K_{(i)}$ は適当な正定行列を表す。初期状態 $x_{t^0(0)}$ は設計者が任意に与え、初期入力に関する情報がない場合は $\bar{u}_{(0)} \equiv 0$ とする。更新則 (31) は、3組の試行実験で1回の学習となっている。まず $3i+1$ 回目の試行で、式 (29) で示した参照軌道を用いて軌道 $x_{(3i)}$ の時間反転軌道 $\psi_{(3i)}$ を生成する。次に $3i+2$ 回目の試行で、式 (30) における $\Sigma(\psi(t^0) - \epsilon(J_c - R_c)T\nabla_{x_{t^1}}\hat{\Gamma}, w)$ の出力情報である $y_{(3i+2)}$ を獲得し、式 (30) を用いて $\delta\Sigma(x_{t^0}, \bar{u})^*(\nabla_{x_{t^1}}\hat{\Gamma}, 0)$ の出力情報を生成する。これで入力に関する勾配が求められ、 $3i+3$ 回目の試行で、式 (14) に基づき、新しい学習入力を生成する。**注意 3** 式 (10) を満たす対称歩行軌道の近傍で学習を行う場合は、更新則 (31) は式 (11) より次式のように簡単になる。筆者らはこれまでに対称歩行軌道の近傍で学習を行い、受動歩行軌道や、平地における歩行軌道の生成を報告している [18].

$$\begin{cases}
x_{t^0(2i+1)} = x_{t^0(2i)} + \epsilon_{(i)}(J_c - R_c)T\delta\Psi_1(x_{t^1(2i)})^* \Lambda_x \\
\quad \times (\Psi_0(x_{t^0(2i)}) - \Psi_1(x_{t^1(2i)})) \\
\bar{u}_{(2i+1)} = \bar{u}_{(2i)} \\
(q_{(2i+1)}^r, \dot{q}_{(2i+1)}^r) \equiv (0, 0) \\
x_{t^0(2i+2)} = x_{t^0(2i)} \\
\bar{u}_{(2i+1)} = \bar{u}_{(2i)} - K_{(i)} \left(\Lambda_{\bar{u}} \bar{u}_{(2i)} + \frac{1}{\epsilon_{(i)}} \mathcal{R}(y_{(2i+1)} - y_{(2i)}) \right) \\
(q_{(2i+2)}^r, \dot{q}_{(2i+2)}^r) \equiv (0, 0)
\end{cases} \quad (32)$$

注意 4 本手法は、与えられた初期状態に対する最適軌道を生成するため、初期状態の最適性は評価しておらず、また初期状態が変われば得られる軌道も変化する。しかし筆者らは過去の結果 [19] において、慣性行列の情報を利用できるという仮定の下では、最適な初期状態を学習できる手法を提案している。また、いくつかの初期状態に対して本手法により最適軌道を求めておけば、その内挿により任意の初期状態に対して最適軌道に近い軌道が生成できると考えられ、これを初期軌道として学習することで任意の初期状態に対する最適軌道も容易に生成できる。

更新則 (31), (32) は、式 (27) で定義される $\Psi_1(x_{t^1})$, $\Psi_0(x_{t^0})$ を含み、このままでは衝突方程式や慣性行列の情報が必要となるため、次節でこれらの情報を必要としない手法を

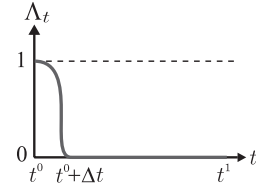


Fig. 3 Illustration of $\Lambda_t(t)$

提案する。

4.2 学習最適制御と最小二乗法を組み合わせた最適歩容生成

本節からは、状態遷移を表す写像を、角運動量保存則から導出された式 (22) ではなく、次式のような一般的な q と \dot{q} に関する一階微分可能な非線形写像と仮定し、これを各試行実験のデータから最小二乗法を用いて推定する手法を提案する。

$$\dot{q}_+ = f_{\Pi}(q_-, \dot{q}_-) \quad (33)$$

式 (24) のように評価関数に状態 x が含まれる場合、運動量 p を求める際に慣性行列の情報が必要となる。そこで、出力 $y = q$ とその時間微分 $\dot{y} = \dot{q}$ の関数に、時間に関する重み関数を加えることで、初期時刻と終端時刻近傍のみを評価し、式 (24) と同様の効果が期待できる次式の評価関数を定義する。

$$\begin{aligned}
\hat{\Gamma}_y(y, \dot{y}, \bar{u}) := & \\
& \frac{1}{2} \int_{t^0}^{t^1} (y(\tau) - C\mathcal{R}(y)(\tau))^\top \Lambda_y(\tau) (y(\tau) - C\mathcal{R}(y)(\tau)) d\tau \\
& + \frac{1}{2} \int_{t^0}^{t^1} (\dot{y}(\tau) - \mathcal{R}(f_{\Pi}(y, \dot{y}))(\tau))^\top \\
& \quad \times \Lambda_{\dot{y}}(\tau) (\dot{y}(\tau) - \mathcal{R}(f_{\Pi}(y, \dot{y}))(\tau)) d\tau \\
& + \frac{1}{2} \int_{t^0}^{t^1} \bar{u}(\tau)^\top \Lambda_{\bar{u}} \bar{u}(\tau) d\tau
\end{aligned} \quad (34)$$

ここで、 $\Lambda_{\bar{u}}, \Lambda_y, \Lambda_{\dot{y}}$ はそれぞれの項に関する重みを表し、 $\Lambda_{\bar{u}}$ は適当な正定行列である。 $\Lambda_y, \Lambda_{\dot{y}}$ は適当な正定数 $k_{(\cdot)}$ と次式で定義される重み関数 $\Lambda_t(t)$ (Fig. 3) を用いてそれぞれ $\text{diag}(k_{y_1}, k_{y_2})\Lambda_t(t)$ と $\text{diag}(k_{\dot{y}_1}, k_{\dot{y}_2})\Lambda_t(t)$ で定義される。

$$\Lambda_t(t) := \begin{cases} \frac{1}{2} \left(1 - \cos \left(\frac{t^0 + \Delta t - t}{\Delta t} \pi \right) \right) & (t^0 \leq t \leq t^0 + \Delta t) \\ 0 & (t^0 + \Delta t < t \leq t^1) \end{cases} \quad (35)$$

設計パラメータ Δt を十分小さくすることで、前章の評価関数 (24) と同様に初期時刻と終端時刻近傍のみを評価できる。

2章で述べた学習最適制御手法 [16] [17] が適用できる評価関数は、入力と出力の汎関数であり、このままでは出力の時間微分 \dot{y} を含む式 (34) には適用できない。そこで筆者らは、擬似共役微分作用素を導入することで、出力の時間微分を含む評価関数を扱える方法を提案した [19]。本節では、更新則の導出に必要な事柄のみを示す。詳細は文献 [19] を参照されたい。

補題 2 文献 [19] ξ, η をそれぞれ $\xi, \eta \in L_2[t^0, t^1]$ であり、少なくとも一階微分可能な信号であるとする。また、 ξ が $\xi(t^0) = \xi(t^1) = 0$ を満たすと仮定する。このとき、次式が成立する。

$$\langle \eta, \mathcal{D}_t(\xi) \rangle_{L_2} = -\langle \mathcal{D}_t(\eta), \xi \rangle_{L_2} \quad (36)$$

補題 2 の結果を利用するため、以下の仮定をおく。

仮定 4 $y_{(i)}$ を i 回目の試行実験における出力とすると、以下の二つの条件が満たされる。

$$\|y_{(i)}(t^0) - y_{(i-1)}(t^0)\| \ll 1, \quad \|y_{(i)}(t^1) - y_{(i-1)}(t^1)\| \ll 1$$

一般に反復学習制御では、各試行実験における初期状態は常に一定であると仮定されており [16] [23]、仮定 4 の一つめの条件は満たされる。本論文では、さらに二つめの条件が必要となるが、これが満たされない場合でも、最適化で用いられるペナルティ関数法を用いて、評価関数に終端状態に関する拘束項を付加することで、近似的にこの条件を満足させることができる [19]。

補題 2 と、時間反転作用素が $\mathcal{R}^* = \mathcal{R}$ を満たすこと [19] を用いて、 $\hat{\Gamma}_y(y, \dot{y}, \bar{u})$ の Fréchet 微分は次式のように計算できる。

$$\begin{aligned} & \delta \hat{\Gamma}_y(y, \dot{y}, \bar{u})(\delta y, \delta \dot{y}, \delta \bar{u}) \\ &= \langle \Lambda_y(y - C\mathcal{R}(y)), \delta y - C\mathcal{R}(\delta y) \rangle \\ & \quad + \langle \Lambda_{\dot{y}}(\dot{y} - \mathcal{R}(f_{\Pi}(y, \dot{y}))), \delta \dot{y} - \mathcal{R}(\partial_y f_{\Pi}(y, \dot{y})(\delta y)) \rangle \\ & \quad + \partial_y f_{\Pi}(y, \dot{y})(\delta \dot{y}) + \langle \Lambda_{\bar{u}} \bar{u}, \delta \bar{u} \rangle \\ &= \langle (\text{id} - \mathcal{R}C)\Lambda_y(\text{id} - C\mathcal{R})(y), \delta y \rangle \\ & \quad + \langle \Lambda_{\dot{y}}(\dot{y} - \mathcal{R}(f_{\Pi}(y, \dot{y}))), \delta \dot{y} \rangle \\ & \quad - \langle \partial_y f_{\Pi}(y, \dot{y})^* \mathcal{R} \Lambda_{\dot{y}}(\dot{y} - \mathcal{R}(f_{\Pi}(y, \dot{y}))), \delta y \rangle \\ & \quad - \langle \partial_{\dot{y}} f_{\Pi}(y, \dot{y})^* \mathcal{R} \Lambda_{\dot{y}}(\dot{y} - \mathcal{R}(f_{\Pi}(y, \dot{y}))), \delta \dot{y} \rangle + \langle \Lambda_{\bar{u}} \bar{u}, \delta \bar{u} \rangle \\ &= \langle (\text{id} - \mathcal{R}C)\Lambda_y(\text{id} - C\mathcal{R})(y) - \partial_y f_{\Pi}(y, \dot{y})^* \mathcal{R} \Lambda_{\dot{y}}(\dot{y} - \mathcal{R}(f_{\Pi}(y, \dot{y}))) - \mathcal{D}_t((\text{id} - \partial_{\dot{y}} f_{\Pi}(y, \dot{y})^* \mathcal{R} \Lambda_{\dot{y}}(\dot{y} - \mathcal{R}(f_{\Pi}(y, \dot{y}))))), \delta y \rangle + \langle \Lambda_{\bar{u}} \bar{u}, \delta \bar{u} \rangle \\ &=: \langle \nabla_y \hat{\Gamma}_y, \delta y \rangle + \langle \nabla_{\dot{y}} \hat{\Gamma}_y, \delta \dot{y} \rangle \\ &= \langle \nabla_{\bar{u}} \hat{\Gamma}_y + \delta \Sigma^{x_{t^0}}(\bar{u})^*(\nabla_y \hat{\Gamma}_y), \delta \bar{u} \rangle \end{aligned} \quad (37)$$

ここで、 $\partial_y f_{\Pi}(y, \dot{y})$ 、 $\partial_{\dot{y}} f_{\Pi}(y, \dot{y})$ はそれぞれ y と \dot{y} に関する f_{Π} の偏 Fréchet 微分を表す。次に、 $\nabla_y \hat{\Gamma}_y$ の具体的な計算方法を述べる。まず $f_{\Pi}(y, \dot{y})$ であるが、式 (37) よりこの項には時間反転作用素 \mathcal{R} (7) と $\Lambda_{\dot{y}}$ 中の時間に関する重み関数 $\Lambda_t(t)$ (35) が作用していることから、実際に必要なのは着地時 ($t = t^1$) における情報のみとなる。 $f_{\Pi}(y_-, \dot{y}_-)$ は着地直後の状態であり、実験による計測値から得られる。つぎに、 $\partial_y f_{\Pi}(y, \dot{y})^*$ 、 $\partial_{\dot{y}} f_{\Pi}(y, \dot{y})^*$ の両方の項にも、 \mathcal{R} と $\Lambda_t(t)$ が作用していることと、 $\partial f_{\Pi}/\partial(y, \dot{y}) = (\partial_y f_{\Pi}, \partial_{\dot{y}} f_{\Pi})$ に注意すると、結局 $\nabla_y \hat{\Gamma}_y$ の計算には、着地時における f_{Π} の (y, \dot{y}) に関するヤコビアンのみが必要となる。式 (33) より

$$d\dot{y}_+ = \frac{\partial f_{\Pi}}{\partial(y, \dot{y})} \Big|_{t=t^1} \begin{bmatrix} dy_- \\ d\dot{y}_- \end{bmatrix} \quad (38)$$

が成立するため、この情報を最小二乗法を用いて、学習により得られる実験データから推定する。式 (38) における dy_- 、 $d\dot{y}_-$ 、 $d\dot{y}_+$ を、 $n-2$ 回目までに得られている実験データと、 n 回目の実験値に一番近いと考えられる $n-1$ 回目の実験データとの差分でそれぞれ近似する。そこで、次式のデータセットを定義する。

$$\Delta Y_{-(n-1)} := \begin{bmatrix} \rho^{n-3}(y_{(1)}^{-\top} - y_{(n-1)}^{-\top}, \dot{y}_{(1)}^{-\top} - \dot{y}_{(n-1)}^{-\top}) \\ \rho^{n-4}(y_{(2)}^{-\top} - y_{(n-1)}^{-\top}, \dot{y}_{(2)}^{-\top} - \dot{y}_{(n-1)}^{-\top}) \\ \vdots \\ y_{(n-2)}^{-\top} - y_{(n-1)}^{-\top}, \dot{y}_{(n-2)}^{-\top} - \dot{y}_{(n-1)}^{-\top} \end{bmatrix},$$

$$\Delta \dot{Y}_{+(n-1)} := \begin{bmatrix} \rho^{n-3}(\dot{y}_{(1)}^{+\top} - \dot{y}_{(n-1)}^{+\top}) \\ \rho^{n-4}(\dot{y}_{(2)}^{+\top} - \dot{y}_{(n-1)}^{+\top}) \\ \vdots \\ \dot{y}_{(n-2)}^{+\top} - \dot{y}_{(n-1)}^{+\top} \end{bmatrix} \quad (39)$$

ただし、 ρ ($0 < \rho \leq 1$) は忘却係数を表す。 $\Delta Y_{-(n-1)}$ のサイズは $(n-2) \times 4$ であり、 $\Delta \dot{Y}_{+(n-1)}$ は $(n-2) \times 2$ である。式 (38)、(39) より、次式のように書くことができる。

$$\Delta \dot{Y}_{+(n-1)} \approx \Delta Y_{-(n-1)} \frac{\partial f_{\Pi}}{\partial(y, \dot{y})} \Big|_{t=t^1}^{\top} \quad (40)$$

式 (40) から、着地時の f_{Π} の (y, \dot{y}) に関するヤコビアン の推定値が次式のように計算できる。

$$\frac{\partial f_{\Pi}}{\partial(y, \dot{y})} \Big|_{t=t^1}^{\top} = \left[\begin{array}{c} \partial_y f_{\Pi}(y, \dot{y})^* \\ \partial_{\dot{y}} f_{\Pi}(y, \dot{y})^* \end{array} \right] \Big|_{t=t^1} \approx \Delta Y_{-(n-1)} \# \Delta \dot{Y}_{+(n-1)} \quad (41)$$

ここで、 $(\cdot)\#$ は擬似逆行列を表す。ただし、式 (39) のデータセットさえ用意すれば、例えば MATLAB の行列左除算などを用いて式 (41) は簡単に解ける。以降では、このとき得られる推定値を $\partial_y f_{\Pi}(y_-, \dot{y}_-)^*_{(n-1)}$ 、 $\partial_{\dot{y}} f_{\Pi}(y_-, \dot{y}_-)^*_{(n-1)}$ と表記する。

最後に、本章で提案する学習最適制御と最小二乗法を組み合わせたアルゴリズムをまとめる。

Step 0 : 初期状態 x_{t^0} を適当に定め、適当な参照入力を用いるか、無入力で実験を行い、得られた実験データを $Q_{(0)}$ 、 $y_{(0)}$ 、 $\dot{y}_{(0)}$ とする。さらに、式 (41) を解くために必要な最低 4 回の予備実験を行い、最初の実験と合わせることでデータセット $\Delta Y_{-(5)}$ 、 $\Delta \dot{Y}_{+(5)}$ を用意する。ただし、予備実験の初期状態はそれぞれ独立な x_{t^0} の近傍とし、入力は適当な参照入力を用いるか、無入力で実験を行うものとする。 $i = 0$ として Step 1 へ進む。

Step $3i + 1$: 次式の学習則を用いて $3i+1$ 回目の実験を行う。

$$\begin{cases} Q_{t^0(3i+1)} = (q_{t^1(3i)}^{\top}, -\dot{q}_{t^1(3i)}^{\top})^{\top} \\ \bar{u}_{(3i+1)} \equiv 0 \\ (q_{(3i+1)}^r, \dot{q}_{(3i+1)}^r) = (\mathcal{R}(q_{(3i)}), -\mathcal{R}(\dot{q}_{(3i)})) \end{cases}$$

Step $3i+2$ へ進む。

Step $3i + 2$: 次式の学習則を用いて $3i+2$ 回目の実験を行う。

$$\begin{cases} Q_{t^0(3i+2)} = Q_{t^0(3i+1)} \\ \bar{u}_{(3i+2)} = \bar{u}_{(3i+1)} + \epsilon_{(i)} \mathcal{R}(\widehat{\nabla_y \hat{\Gamma}_y}_{(3i)}) \\ (q_{(3i+2)}^r, \dot{q}_{(3i+2)}^r) = (\mathcal{R}(q_{(3i)}), -\mathcal{R}(\dot{q}_{(3i)})) \end{cases}$$

ここで、 $\widehat{\nabla_y \hat{\Gamma}_y}_{(3i)}$ は、データセット $\Delta Y_{-(5+i)}$ 、 $\Delta \dot{Y}_{+(5+i)}$ から式 (41) を解いた推定値を用いて、次式で与えられる。

$$\begin{aligned} \widetilde{\nabla_y \hat{\Gamma}_{y(3i)}} &= (\text{id} - \mathcal{R}C)\Lambda_y(\text{id} - C\mathcal{R})(y_{(3i)}) \\ &\quad - \partial_y f_{\Pi}(y_-, \dot{y}_-)^*_{(5+i)} \mathcal{R}\Lambda_{\dot{y}}(\dot{y}_{(3i)}) \\ &\quad - f_{\Pi}(y_{-(3i)}, \dot{y}_{-(3i)})\mathbf{1}(t) \\ &\quad - \mathcal{D}_t((\text{id} - \partial_y f_{\Pi}(y_-, \dot{y}_-)^*_{(5+i)} \mathcal{R})\Lambda_{\dot{y}} \\ &\quad \times (\dot{y}_{(3i)} - f_{\Pi}(y_{-(3i)}, \dot{y}_{-(3i)})\mathbf{1}(t))) \end{aligned}$$

ただし $\mathbf{1}(t) := \text{id}, \forall t \in [t^0, t^1]$ である. Step $3i+3$ へ進む.

Step $3i+3$: 次式の学習則を用いて $3i+3$ 回目の実験を行う.

$$\begin{cases} Q_{t^0(3i+3)} = Q_{t^0(3i)} \\ \bar{u}_{(3i+3)} = \bar{u}_{(3i)} - K_{(i)} \left(\Lambda_{\bar{u}} \bar{u}_{(3i)} + \frac{1}{\epsilon_{(i)}} \mathcal{R}(y_{(3i+2)} - y_{(3i+1)}) \right) \\ (q_{(3i+3)}^r, \dot{q}_{(3i+3)}^r) \equiv 0 \end{cases}$$

$i = i + 1$ として, Step $3i+1$ へ進む.

注意 5 注意 3 と同様に, 式 (10) を満たす対称歩行軌道の近傍で学習を行う場合は, 前述のアルゴリズムにおける Step 0 を実行後, Step $3i+1$ から Step $3i+3$ の代わりに, 次式の更新則を用いた Step $2i+1$ から Step $2i+2$ の実験を繰り返せばよい.

$$\begin{cases} Q_{t^0(2i+1)} = Q_{t^0(2i)} \\ \bar{u}_{(2i+1)} = \bar{u}_{(2i)} + \epsilon_{(i)} \widetilde{\nabla_y \hat{\Gamma}_{y(2i)}} \\ (q_{(2i+1)}^r, \dot{q}_{(2i+1)}^r) \equiv 0 \\ Q_{t^0(2i+2)} = Q_{t^0(2i)} \\ \bar{u}_{(2i+2)} = \bar{u}_{(2i)} - K_{(i)} \left(\Lambda_{\bar{u}} \bar{u}_{(2i)} + \frac{1}{\epsilon_{(i)}} \mathcal{R}(y_{(2i+1)} - y_{(2i)}) \right) \\ (q_{(2i+2)}^r, \dot{q}_{(2i+2)}^r) \equiv 0 \end{cases} \quad (42)$$

注意 6 提案手法は, 学習の初期状態として非零の初期速度を含むが, 実験ではその実現が難しい場合が考えられる. しかしながら, 制御対象としてフルアクチュエータな機械系を考えているため, 例えば十分大きなゲインを用いた PD フィードバックにより任意の状態を実現できる. さらに, 本アルゴリズムはオフラインであるため, ある種の助走区間を設けて実験を行えばよい. つまり, 零初期速度から実験を開始し, はじめは PD フィードバック等を用いて学習則で実現すべき初期状態を近似的に達成したあと, 入力を学習入力に切り替えて, その時刻からの出力情報のみを学習に用いればよい. また, 筆者らは最近本論文の結果と仮想拘束とを組み合わせ, 自動的に学習を継続できる手法を発表している [24]. この方法は, ロボットが拘束により歩き続けながら学習を継続できるため, 人間が試行実験を繰り返す必要がなく, 上述の問題を回避できる. このような発展的な手法においても, 本論文の結果が重要な役割を果たす.

5. シミュレーション結果

前章のアルゴリズムを用いて, コンパスロボット (Fig. 1) の最適な平地歩行軌道を生成するシミュレーションを行った. $K_P = \text{diag}(4, 4), K_D = \text{diag}(2, 2)$ を用いて式 (23) の PD フィードバックを施した. 評価関数 (34) 中の重み関数

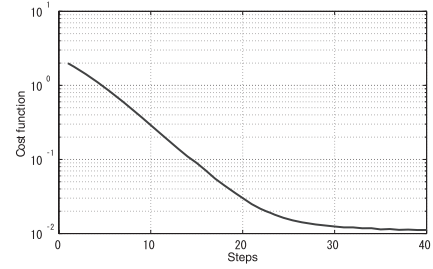


Fig. 4 Cost function

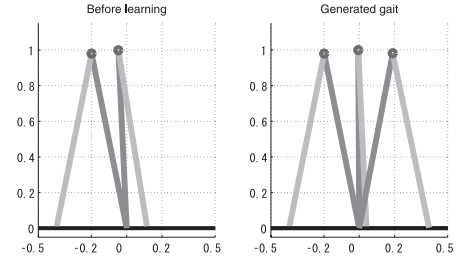


Fig. 5 Stick diagrams before and after learning

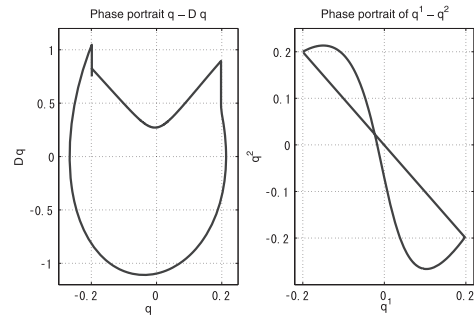


Fig. 6 Phase portraits

を $(k_{y1}, k_{y2}, k_{\dot{y}1}, k_{\dot{y}2}) = (30, 30, 1 \times 10^{-3}, 1 \times 10^{-3}), \Delta t = 5 \times 10^{-3}$ [s], $\Lambda_{\bar{u}} = \text{diag}(1 \times 10^{-6}, 1 \times 10^{-6})$ とし, 学習に関するパラメータを $K_{(\cdot)} = \text{diag}(400, 400), \epsilon_{(\cdot)} = 1$, 忘却係数を $\rho = 0.7$ とした. 筆者らの過去の報告 [18] などから, コンパスロボットの歩行軌道は十分対称軌道に近いことが期待できたため, 注意 5 で述べた 2 ステップのアルゴリズム (42) を用いた. 3 ステップの学習も同様に適用したが, どちらの場合も同様の結果が得られたため, 2 ステップの適用結果のみを示す.

Fig. 4~Fig. 9 に, 初期状態 $(-0.20, 0.20, 0.82, 0.35)$ の下での, 40 回の学習つまり予備実験 5 回を含む 85 回のシミュレーション結果を示す. 予備実験で用いた初期状態は, Matlab の rand 関数で生成した乱数ベクトルを足すことで生成し, $(-0.16, 0.22, 0.88, 0.43), (-0.10, 0.29, 0.84, 0.47), (-0.17, 0.24, 0.88, 0.49), (-0.16, 0.30, 0.86, 0.47)$ を用いた.

Fig. 4 は, 各学習ステップに対する評価関数 (34) の値をプロットしたものであり, 値が単調減少し, 40 回目付近から一定となっていることから, 速やかに (局所的な) 最適軌道に収束していることが定量的に分かる. また, Fig. 5 は左が学習前, 右が学習後のロボットの運動を表す. 学習前では直立状態付近で停止しているが, 学習後は歩行と見られる運動が獲得できてい

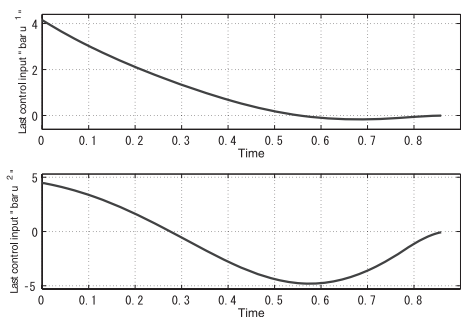
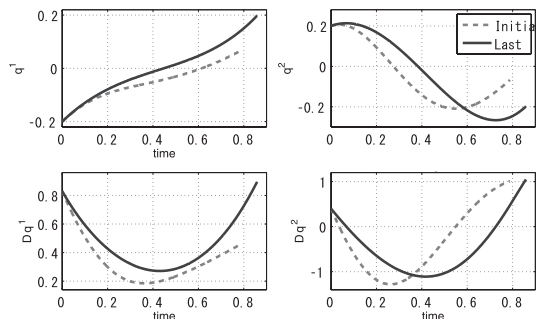
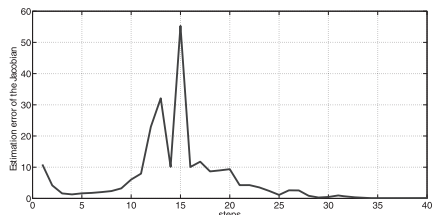
Fig. 7 Generated control inputs \bar{u} Fig. 8 Responses of q and \dot{q} before and after learning

Fig. 9 Estimation error of the Jacobian at each step

る。さらに、Fig. 6 は、学習で得られた歩行軌道の相図であり、十分閉軌道に近いことから周期軌道に近い歩行軌道が得られたことが視覚的に分かる。Fig. 7 に生成された学習入力 \bar{u} を示すが、これを用いて 200 歩の連続歩行を確認した。実際には、学習で得られた軌道に対して軌道安定化を施して最終的な周期軌道とするため、相図が完全な閉軌道ではないが、十分利用できると思われる。Fig. 8 は学習前 (点線) と学習後 (実線) の $q^1, q^2, \dot{q}^1, \dot{q}^2$ の時間応答を示している。シミュレーションの作成にあたり、状態遷移写像のモデルの真値として、式 (22) の衝突方程式を用いた (学習則の設計には用いていないことに注意)。学習後の歩行軌道における状態遷移写像のヤコビアン $\partial(\bar{\Pi}(\alpha(q_-)\dot{q}_-))/\partial(q_-, \dot{q}_-)$ の真値 $\partial(\bar{\Pi}(\alpha(q_-)\dot{q}_-))/\partial(q_-, \dot{q}_-)$ と、式 (41) を用いて学習により推定した状態遷移写像のヤコビアン $\partial\widetilde{f_{\Pi}}(q_-, \dot{q}_-)/\partial(q_-, \dot{q}_-)$ は次式のようにになった。

$$\frac{\partial(\bar{\Pi}(\alpha(q_-)\dot{q}_-))}{\partial(q_-, \dot{q}_-)} = \begin{bmatrix} -0.6117 & 0.5830 & 0.9715 & -0.1047 \\ -1.7571 & 1.6554 & 0.8096 & -0.1941 \end{bmatrix},$$

$$\frac{\partial\widetilde{f_{\Pi}}(q_-, \dot{q}_-)}{\partial(q_-, \dot{q}_-)} = \begin{bmatrix} -0.5828 & 0.5828 & 0.9616 & -0.1042 \\ -1.6546 & 1.6546 & 0.7744 & -0.1923 \end{bmatrix}$$

各学習ステップに対する、ヤコビアン $\partial\widetilde{f_{\Pi}}(q_-, \dot{q}_-)/\partial(q_-, \dot{q}_-)$ の真値と推定値との差の誘導ノルムを Fig. 9 に示す。直立姿勢で停止していた軌道が、

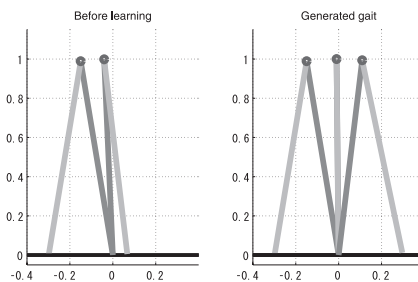


Fig. 10 Stick diagrams before and after learning

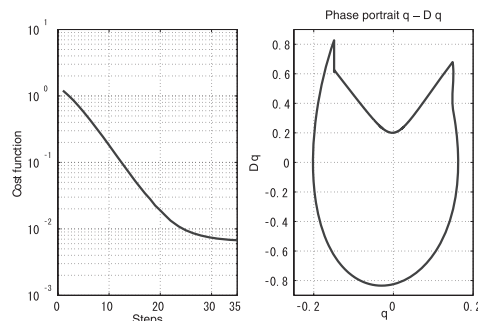


Fig. 11 Cost function and phase portrait

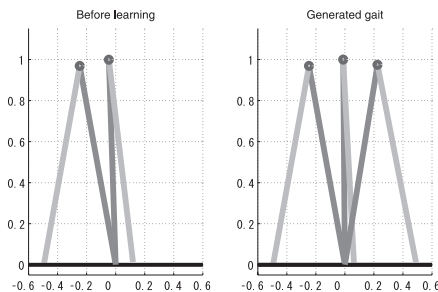


Fig. 12 Stick diagrams before and after learning

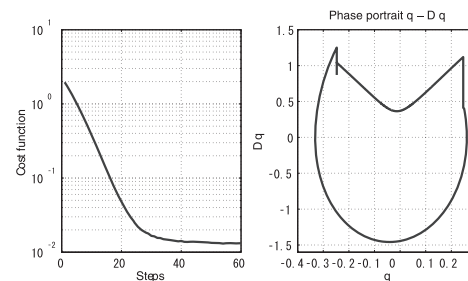


Fig. 13 Cost function and phase portrait

学習が進み遊脚を前へ踏み出す軌道へと変わるため、途中で最小二乗の誤差がいったん大きくなっていると考えられる。これらの結果より、ヤコビアンが十分推定できていることが分かる。

次に、異なる初期状態から学習を行ったいくつかの結果を示す。Figs. 10, 11 は初期状態 $(-0.15, 0.15, 0.63, 0.35)$ の下で、35 回の学習を行った結果であり、Figs. 12, 13 は初期状態 $(-0.25, 0.25, 1.05, 0.4)$ の下で、60 回の学習を行った結果である。どちらも得られた入力による 200 歩の連続歩行を確認した。

本章では、重み関数や学習ゲインなどの設計パラメータは以

下の方針で試行錯誤的に定めており、システマティックな決定手法の考案は今後の課題の一つである。まず歩行の生成においてロボットの関節角 q を速やかに周期的な軌道に近づけるため、 k_{y1} , k_{y2} を比較的大きく設定した。平地歩行には何らかの制御入力が必要だが、初期入力を 0 としているため、評価関数の入力に関する項は学習の始めでは大きくなる。さらに、 q, \dot{q} は $\Lambda_t(t)$ (35) により初期時刻と終端時刻近傍の値しか評価されないため、入力の重みを他の項に比べ小さく設定する必要がある。 $\epsilon_{(\cdot)}$ は式 (15) が成立するように小さくすればよいが、小さすぎると学習の進捗が遅くなり、多くの試行実験を要する。一般に勾配法では、次元探索により学習ゲイン $K_{(\cdot)}$ を決定するが、制御対象のモデルが利用できない本手法では使えない。本章のように一定のゲインを用いるか、試行錯誤的に調節する必要がある。

6. おわりに

本論文では、不連続な状態遷移を考慮した学習最適制御に基づく最適歩容生成手法を提案した。本手法は、着地直前までの歩行軌道の学習を行いながら、さらに各試行実験で得られた実験データを用いた最小二乗法により状態遷移写像のヤコビアンを推定し、着地時の状態遷移に関する学習も同時に行う。そのため、制御対象だけでなく、衝突方程式のモデルパラメータも必要とせず、不連続な状態遷移を考慮した学習が可能となる。

現在広く用いられている ZMP 規範の軌道設計においても、逆運動学計算時の拘束条件として目標軌道の設計時に初期状態を与える場合が多い。また、モデルに基づく歩行制御手法において、モデル構築の際に必要なリンクごとの重心位置や慣性行列の正確な同定は一般的に困難である。そのため、与えられた初期状態の下でロボットの物理パラメータを必要とせず、設計した評価関数を最小化する最適軌道が生成できる本手法は、多くの軌道生成問題において有効であると考えられる。本論文では基礎理論の開発に重点をおき、構造が簡単なコンパスロボットモデルでその効果を検証したが、次の段階として多自由度ロボットへの適用が重要である。多自由度ロボットの歩行は一般的に対称軌道とならないため (例えば、膝を有するロボットの軌道)、本論文で述べた 3-Step の学習則を用いる必要がある。本論文で提案した理論の実用的な応用として、筆者らは現在 12 自由度を有するロボットの学習に取り組んでおり [25]、さらに実機試験による検証を行うために、歩行ロボットを製作中である。

参考文献

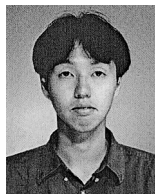
[1] M. Vukobratović and J. Stepanenko: "On the stability of anthropomorphic systems," *Mathematical Biosciences*, vol.15, pp.1-37, 1972.
 [2] 高西, 石田, 山崎, 加藤: "2 足歩行ロボット WL-10RD による動歩行の実現", *日本ロボット学会誌*, vol.3, no.4, pp.325-336, 1985.
 [3] K. Hirai, M. Hirose, Y. Haikawa and T. Takenaka: "The development of honda humanoid robot," *Proc. IEEE Int. Conf. Robotics and Automation*, pp.1321-1326, 1998.
 [4] T. McGeer: "Passive dynamic walking," *Int. J. Robotics Research*, vol.9, no.2, pp.62-82, 1990.
 [5] A. Goswami, B. Thuilot and B. Espiau: *Compass-like biped*

robot part i: Stability and bifurcation of passive gaits, INRIA Research Report, no.2996, 1996.
 [6] K. Osuka and K. Kirihara: "Motion analysis and experiments of passive walking robot quartet II," *Proc. IEEE Int. Conf. Robotics and Automation*, pp.3052-3056, 2000.
 [7] A. Sano, Y. Ikemata and H. Fujimoto: "Analysis of Dynamics of Passive Walking from Storage Energy and Supply Rate," *Proc. IEEE Int. Conf. Robotics and Automation*, pp.2478-2483, 2003.
 [8] 平田健太郎: "受動歩行に内在する安定化機構について", 第 8 回計測自動制御学会制御部門大会予稿集, 2008.
 [9] M.W. Spong: "Passivity-base control of the compass gait biped," *Proc. of IFAC World Congress*, pp.19-23, 1999.
 [10] F. Asano, M. Yamakita, N. Kamamichi and Z.W. Luo: "A novel gait generation for biped walking robots based on mechanical energy constraint," *IEEE Trans. Robotics and Automation*, vol.20, no.3, pp.565-573, 2004.
 [11] F. Asano and Z.W. Luo: "Energy-efficient and high-speed dynamic biped locomotion based on principle of parametric excitation," *IEEE Trans. Robotics*, vol.24, no.6, pp.1289-1301, 2008.
 [12] R. Tedrake, T.W. Zhang and H.S. Seung: "Stochastic policy gradient reinforcement learning on a simple 3D biped," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.2849-2854, 2004.
 [13] 藤井, 大塚: "コンパス型歩行モデルにおけるエネルギー最適軌道の探索", 第 6 回計測自動制御学会制御部門大会予稿集, pp.169-172, 2006.
 [14] 玄, 矢ヶ崎, 藤本, Cheng: "二重振子系の対称軌道族と最適歩行制御への応用", 第 36 回制御理論シンポジウム予稿集, pp.99-102, 2007.
 [15] P.E. Crouch and A.J. van der Schaft: *Variational and Hamiltonian Control Systems (Lecture Notes on Control and Information Science)*. Springer-Verlag, 1987.
 [16] K. Fujimoto and T. Sugie: "Iterative learning control of Hamiltonian systems: I/O based optimal control approach," *IEEE Trans. Autom. Contr.*, vol.48, no.10, pp.1756-1761, 2003.
 [17] 藤本, 堀内, 杉江: "ハミルトン系の変分対称性に基づく学習最適制御", *システム制御情報学会論文誌*, vol.21, no.1, 10-17, 2008.
 [18] 佐藤, 藤本, 玄: "コンパス型ロボットの最適平地歩行軌道の生成について", 第 7 回計測自動制御学会制御部門大会予稿集 CD-ROM, 63-3-3, 2007.
 [19] 佐藤, 藤本, 玄: "ハミルトン系の変分対称性に基づく 1 脚ロボットの最適歩容生成", *計測自動制御学会論文集*, vol.43, no.12, pp.1103-1110, 2007.
 [20] S. Satoh, K. Fujimoto and S. Hyon: "Biped gait generation via iterative learning control including discrete state transitions," *Proc. 17th IFAC World Congress*, pp.1729-1734, 2008.
 [21] P. Gregorio, M. Ahmadi and M. Buehler: "Design, control, and energetics of an electrically actuated legged robot," *IEEE Trans. Systems, Man, and Cybernetics*, vol.27, no.4, pp.626-634, 1997.
 [22] K. Fujimoto and T. Sugie: "Canonical transformation and stabilization of generalized Hamiltonian systems," *Systems & Control Letters*, vol.42, no.3, pp.217-227, 2001.
 [23] S. Arimoto, S. Kawamura and F. Miyazaki: "Bettering operation of robotics," *J. Robotic Systems*, vol.1, no.2, pp.123-140, 1984.
 [24] S. Satoh, K. Fujimoto and S. Hyon: "A framework for optimal gait generation via learning optimal control using virtual constraint," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.3426-3432, 2008.
 [25] 佐藤, 藤本, 久場: "力学系の性質に基づく歩行制御について", 第 10 回計測自動制御学会システムインテグレーション部門講演会予稿集, pp.884-887, 2009.



佐藤訓志 (Satoshi Satoh)

2007年名古屋大学大学院工学研究科博士課程前期課程修了。2009年より日本学術振興会特別研究員(DC2)。2010年名古屋大学大学院工学研究科博士課程後期課程修了。同年より広島大学大学院工学研究院助教となり、現在に至る。非線形制御の研究に従事。博士(工学)。2008年IEEE Robotics Automation Society Japan Chapter Young Award, 2010年計測自動制御学会学術奨励賞研究奨励賞などを受賞。IEEE, 計測自動制御学会の会員。
(日本ロボット学会正会員)



藤本健治 (Kenji Fujimoto)

1996年京都大学大学院工学研究科修士課程応用システム科学専攻修了, 1997年同大学大学院博士後期課程を中途退学, 1997年京都大学大学院工学研究科助手等を経て, 2007年より名古屋大学大学院工学研究科准教授。非線形制御の研究に従事。博士(情報学)。IEEE, 計測自動制御学会, システム制御情報学会, 日本機械学会, 日本鉄鋼協会の会員。



玄 相昊 (Sang-Ho Hyon)

1998年早稲田大学大学院修士課程修了, 2002年東京工業大学大学院博士後期課程修了。同年東北大学大学院工学研究科助手, 2005年講師。同年ATR脳情報研究所研究員。2009年度までJST-ICORP計算脳プロジェクト研究員, 2009年からNICT研究員を兼任。2010年4月立命館大学理工学部ロボティクス学科准教授, 現在に至る。運動制御理論, ヒューマノイドロボットの研究開発等に従事。博士(工学)。日本神経科学学会, IEEEなどの会員。
(日本ロボット学会正会員)