

場の重ね合わせと状態空間写像に基づくロボットの運動創発

Robot Motion Emergence based on Superposition of Vector Field and State-space Mapping

渡辺将旭, 岡田昌史 (東工大)

*Masaaki WATANABE and Masafumi OKADA (Tokyo TECH)

Abstract— Robot control system consists of a feedback controller and reference motion pattern. They are designed based on the robot dynamics and coupled with each other, and it requires lots of calculation to obtain them. A new controller design method based on the existing controller will lead useful way to overcome this problem. So far, we have proposed attractor design method of nonlinear dynamics. This method is based on the vector field in state-space and it enables the new controller design based on existing controllers. In this paper, we propose a modified controller design method using definition of "realizable" vector field and a new controller design method based on the superposition of vector field and state-space mapping.

Key Words: attractor design, vector field, nonlinear state-space mapping, motion emergence

1. はじめに

一般に, ロボットの運動制御では Fig.1 にあるように, 安定化コントローラ K と目標運動パターン x_{ref} を設計する必要がある. また, 異なる運動を生成させ

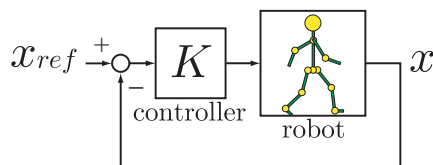


Fig.1 Robot motion control

るためにはこれらを設計し直す必要があり, これらはロボットの運動方程式に基づくことから, 多くの計算を要する. これに対し, 岡田ら [1] はロボットの状態変数を軌道に引き込ませるコントローラ $h(x)$ を設計することで, 安定化とパターン生成を同時に行う方法を提案し, ロボットの運動創発システムの設計を行った. この方法で構成される閉ループ系は Fig.2 で表され, x_{ref} が存在していない. さらに, 渡辺ら [2] はシ

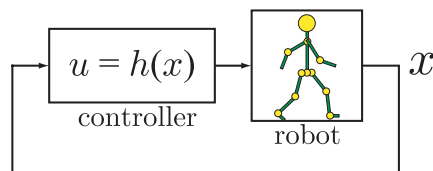


Fig.2 Robot motion emergence

ンボル [3] の概念に基づいて, コントローラを「身体を安定化させる要素」「運動を生成する要素」に分離する方法を提案し, それらの要素の物理的な意味をデータとして実体化する方法を提案した. この方法では, コントローラがそれぞれの要素から構成されているものとしているため, 身体を安定化させる要素を変化させれば他のロボットのコントローラが設計でき, 運動を生成する要素を変化させれば他の運動のコントロー

ラが設計できる可能性を示唆している. そのため, 既存のコントローラに基づいた新たなコントローラ設計を実現し, ロボットの運動制御のための計算コストを大幅に減少させる利点が考えられる.

本研究では, 既存のコントローラの情報を用いて新たなコントローラを設計する方法を提案する. この方法は, 場の重ね合わせに基づいており, 適当な座標変換を考えることでロボットの間のダイナミクスの差を補間するものである. まず, 従来のコントローラの設計法に改良を加え, よりコントローラを設計しやすいアルゴリズムを開発する. また, この方法を用いたタップダンスロボットの運動制御を行い, 既存の情報を用いた新たなコントローラ設計法の有効性を検証する.

2. 軌道と入力のカラクリを考慮したアトラクタ設計法

2.1 アトラクタ設計法

参考文献 [1] ではロボットの運動方程式の解を軌道アトラクタに引き込ませることで, ロボットの運動創発を設計する手法が提案された. これについて簡単に説明する. 離散時間におけるロボットの運動方程式を

$$x_{k+1} = f(x_k) + g(x_k, u_k) \quad (1)$$

とし, コントローラを

$$u_k = h(x_k) \quad (2)$$

とする. 式 (1) と式 (2) の連立方程式の解が

$$\Xi = \left[\xi_1 \quad \xi_2 \quad \xi_3 \quad \dots \right] \quad (3)$$

に引き込まれるように, すなわち, Ξ がアトラクタとなるように $h(x)$ を設計する. 実際に, 式 (2) は x の多項式を用いて,

$$u_k = \Theta \phi(x_k) \quad (4)$$

として設計される. Θ は多項式の係数行列, $\phi(x)$ は x を多項式に展開したベクトルである. この方法では,

ロボットの運動が式 (1) と (2) の連立方程式の解として現れており、運動が物理現象として現れていると言える。また、運動の目標パターンを与える必要がない利点により、環境の変化によってそれに適応した運動へと自動的に遷移する。

Θ の設計では (x_k, u_k) の組を多く求め、これを関数近似する。そのため、現実的な (x_k, u_k) の組 (力学的に整合性の取れた組) を求める必要がある。ある初期値 x_1 から出発した状態変数の時間遷移は、式 (1) を x_k 周りで線形近似した系

$$x_{k+1} = A_k x_k + B_k u_k + C_k \quad (5)$$

を用いて (ただし、以後簡単のため $C_k = 0$ とする)、

$$\begin{bmatrix} x_2 \\ \vdots \\ x_{k+1} \end{bmatrix} = \begin{bmatrix} A_1 \\ \vdots \\ \prod_{i=1}^{k-1} (A_i) \end{bmatrix} x_1 + \begin{bmatrix} B_1 & & 0 \\ \vdots & \ddots & \\ \prod_{i=1}^{k-2} (A_i) B_1 & \cdots & B_k \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_k \end{bmatrix} \quad (6)$$

で表される。これを改めて

$$X = AX_1 + BU \quad (7)$$

と書くことにする。これより、 x の時間遷移を Ξ に沿わせ、しかも、 Ξ に収束させるための入力列は

$$U = B^\# (\Xi - AX_1) \quad (8)$$

によって得られる。また、実際に得られた U を式 (7) に代入することで、 x の時間遷移 X が得られ、これより (x, u) の組が得られる。さらに、多くの初期値 x_1 を設定することで、多くの (x, u) の組が得られ、これを関数近似することで、式 (2) の Θ が設計される。

2.2 コントローラ設計方法の改良

上記のアルゴリズムでは、 A_i, B_i, C_i を求めるためには x_i が必要であるが、 x_i は後から決まるためこれを先に用いることはできない。そこで、参考文献 [1] では x_1 から出発した状態変数が ξ_i に沿って動くことを仮定し、 A_i, B_i, C_i は式 (1) を ξ_i 周りで線形近似した値を用いた。しかし、実際にはこの仮定は満たされにくく、 x_i が ξ_i から大きく離れる場合が多々ある。そのため、コントローラ設計のためのパラメータに多くの調整が必要であり、計算コストが大きかった。そこで、これを以下のように変更する。

1. 式 (8) から得られた U を用いて式 (7) より、実際に入力列 \hat{X} を求める。
2. \hat{X} に基づいて、 A_i, B_i, C_i を計算し直し、新たに \hat{U} を求める。
3. これを繰り返し、現実的な (x_k, u_k) の組を求める。

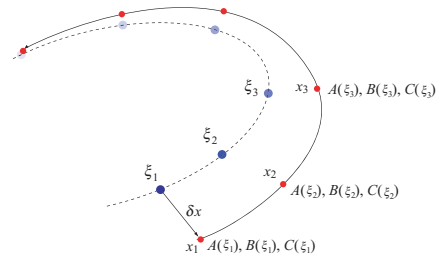


Fig.3 Conventional controller design method

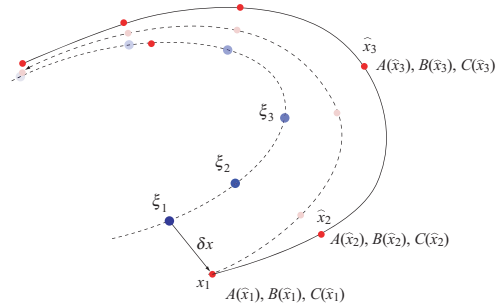


Fig.4 Modified controller design method

この方法を概念的に表すと、従来の方は Fig.3 のように ξ_i に基づいた A_i, B_i, C_i を用いて (x, u) の組を求めていたのに対し、新たな方法では、Fig.4 のように実現される \hat{x}_i に基づいた A_i, B_i, C_i を用いて現実的な (\hat{x}_k, \hat{u}_k) の組を求めるものである。

この方法を用いて、Fig.5 にあるロボットの足踏み動作を設計した。なお、このロボットの運動方程式等

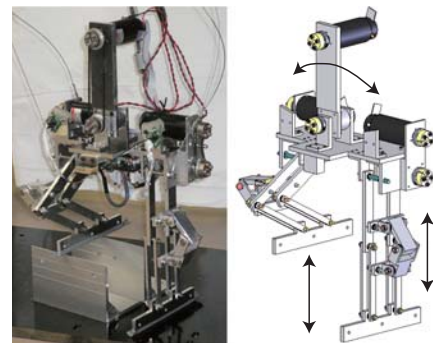


Fig.5 Tapping dance robot

については参考文献 [1] に詳しい。運動方程式の状態変数は、頭の回転角 ϕ 、腰の回転角 θ を用いて、

$$x = \begin{bmatrix} \theta & \dot{\theta} & \phi & \dot{\phi} \end{bmatrix}^T \quad (9)$$

となる。シミュレーション結果を Fig.6 に示す。この設計法により、設計パラメータの試行錯誤的な調整が減少し、コントローラが設計しやすくなった。また、コントローラの設計段階における \hat{x} の様子を Fig.7 に示す。従来の方では、 x に基づいてコントローラを設計していたのに対し、新たな方法を用いることで、現実的な \hat{x} の列が求められている。

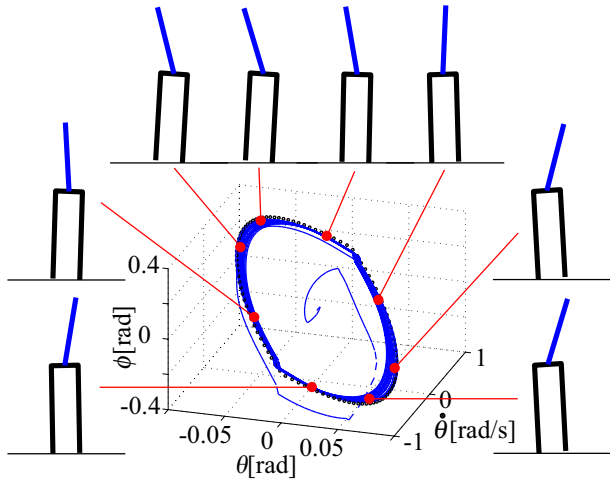


Fig.6 Simulation result of attractor design for the tapping dance robot

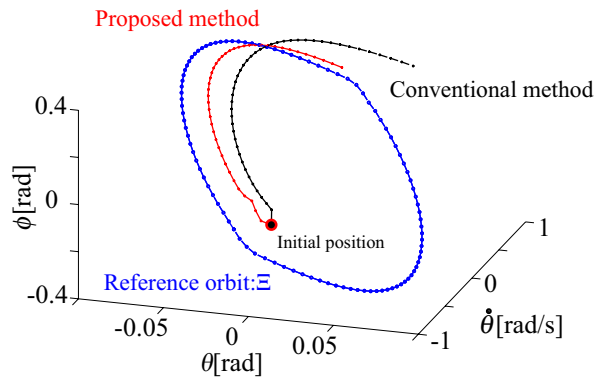


Fig.7 Locus of the obtained state value

3. 既存コントローラに基づく設計法

3.1 コントローラのパラメタライズ

いま、ロボット A の運動 1 を生成するコントローラ Θ_1^A とロボット B の運動 1 を生成するコントローラ Θ_1^B が与えられていたとする。このとき、二つのコントローラの間を次式で定義する。

$$\Theta_1^B \phi(x) = \Theta_1^A \phi(\tilde{x}) \quad (10)$$

$$\tilde{x} = F(x) \quad (11)$$

ここで、 F は x の関数であるが、これは x の非線形な座標変換とみなすことができる。また、コントローラが x の多項式で表されたように、 Λ を係数行列として F を x の多項式を用いて次式で定義する。

$$F(x) = \Lambda \phi(x) \quad (12)$$

これにより、式 (10) は

$$\Theta_1^B \phi(x) = \Theta_1^A \phi(\Lambda \phi(x)) \quad (13)$$

となる。このような Λ は状態空間内に多くの x_i ($i = 1, 2, \dots$) を定義し、式 (13) を満たす解としてニュートン

法によって求められる。なお、具体的に式 (13) の右辺の、 Λ の m 行 n 列要素 λ_{mn} での偏微分は

$$\frac{\partial \Theta_1^A \phi(\Lambda \phi(x))}{\partial \lambda_{mn}} = \Theta_1^A \frac{\partial \phi(\tilde{x})}{\partial \tilde{x}} \frac{\partial \Lambda}{\partial \lambda_{mn}} \phi(x) \quad (14)$$

で表され、 $\frac{\partial \Lambda}{\partial \lambda_{mn}}$ は (m, n) 要素を 1 とし、他の要素を 0 とする行列になる。また、 $\phi(x)$ は x を多項式に展開したベクトルなのでその x での偏微分値 $\frac{\partial \phi(x)}{\partial x}$ は容易に求められる。

3.2 新たなコントローラの設計

Λ を用いて新たなコントローラの設計を行う。いま、 Θ_1^A 、 Θ_1^B に加え、ロボット A の運動 2 を生成するコントローラ Θ_2^A が与えられたとすると、Fig.8 にあるロボット B の運動 2 を生成するための新たな入力 u_2^B は

	Motion 1	Motion 2
Robot A	$u_1^A = \Theta_1^A \phi(x)$	$u_2^A = \Theta_2^A \phi(x)$
Robot B	$u_1^B = \Theta_1^B \phi(x)$ $= \Theta_1^A \phi(\Lambda x)$	$u_2^B = ?$

Fig.8 New controller design

$$u_2^B = \Theta_2^A \phi(\Lambda \phi(x)) \quad (15)$$

で与えられる。この意味について詳しく見てみる。本来、式 (4) のコントローラは入力 u を x の場として求めていることに相当する。そのため、ロボット A においてコントローラ Θ_2^A による場合は、

$$\Theta_2^A \phi(x) = \Theta_1^A \phi(x) + {}^1\Delta_2 \phi(x) \quad (16)$$

$${}^1\Delta_2 = \Theta_2^A - \Theta_1^A \quad (17)$$

のように $\Theta_1^A \phi(x)$ に対して ${}^1\Delta_2 \phi(x)$ で表される場が加わったものとみなすことができる。そこで、ロボット B に関しても、同様の場を加えることで運動を 1 から 2 へ変化させる、すなわち、

$$\Theta_2^B \phi(x) = \Theta_1^B \phi(x) + {}^1\Delta_2 \phi(x) \quad (18)$$

とする。ただし、 ${}^1\Delta_2$ は Fig.9 にあるように、ロボット A の状態空間での場の定義であるため、これをロボット B の状態空間に写像する関数が式 (13) で定義される写像であるとみなすことができる。この結果が

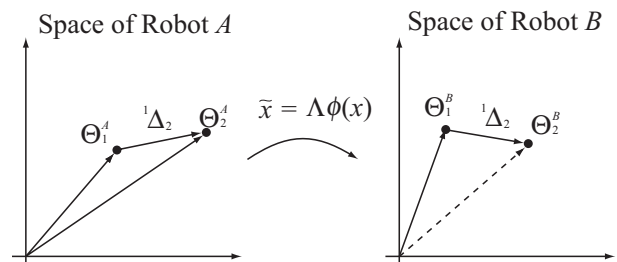


Fig.9 Space mapping for Robot A to Robot B

ら, (θ_1^A, θ_1^B) のみに基づいて Λ を求めるのではなく, (θ_i^A, θ_i^B) ($i = 2, 3, \dots$) も用いて Λ を求めた方が写像の精度が良く, これは既存のコントローラの情報が多い方が精度が向上することと等価である.

4. コントローラ設計の有効性検証

4.1 Λ に基づくコントローラ

本章では, 前章で提案したコントローラ的设计法の有効性をシミュレーションによって検証する. 制御対象としては, Fig.5 にあるタップダンスロボット A と, これよりやや寸法の大きいタップダンスロボット B を用い, 足踏み動作に対して動作の周波数によって制御系を評価する. まず, ロボット A, B の運動 1~3 を生成するコントローラ θ_i^A , ($i = 1 \sim 3$) および θ_1^B を設計した. このときの運動の周波数の対応を Table.1 に示す. ただし, アトラクタ設計法ではモデル化され

Table 1 Robot motion and its frequency

Robot	Motion No.	Controller	Frequency
A	1	θ_1^A	0.99Hz
	2	θ_2^A	1.51Hz
	3	θ_3^A	2.05Hz
B	1	θ_1^B	1.03Hz

ない外乱 (ここでは脚の踏み替えによる衝撃) の影響により, 運動の周期は一定ではないので, 平均を求めることで周波数とした. 次に, θ_1^A と θ_1^B に基づいて, 式 (13) の Λ を求め,

$$u_1^B = \theta_1^A \phi(\Lambda \phi(x)) \quad (19)$$

によって入力を決し, ロボットの運動を生成した. このときの運動は 1.0Hz となり, 状態変数の軌跡を Fig.10(b) に示す. 比較として, θ_1^B を用いたときの運動を (a) に示す. この結果がほぼ等しいことから, Λ を用いたコントローラ的设计が有効であることが示される.

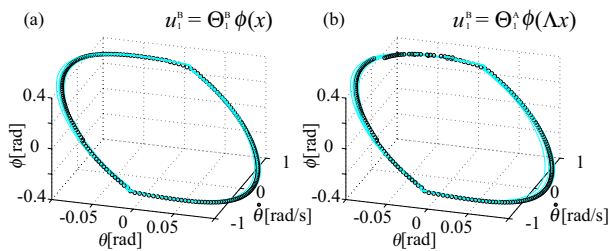


Fig.10 Comparison of robot motions

4.2 新たな運動の生成

次に, 新たな運動の生成を行う. 上で設計した Λ を用いロボット B の運動 3 を

$$u_3^B = \theta_3^A \phi(\Lambda \phi(x)) \quad (20)$$

によって生成した. このとき, 運動は 2.11Hz となった. さらに, θ_1^A と θ_1^B , および, θ_2^A と θ_2^B を用いて

$\hat{\Lambda}$ を求め,

$$\hat{u}_3^B = \theta_3^A \phi(\hat{\Lambda} \phi(x)) \quad (21)$$

を用いてロボット B の運動生成を行った. このとき, 運動は 1.89Hz となった. これらの運動における状態変数の軌跡を Fig.11 に示す. なお, 参考として, 実際

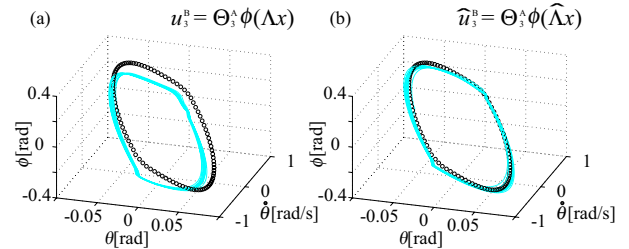


Fig.11 New motion generation using space mapping

にロボット B の運動 3 のためのコントローラ θ_3^B を設計したときの, アトラクタを \circ 印で示す. Fig.11 の (a) と (b) を比べると, (b) の多数のコントローラを用いて Λ を設計する方がアトラクタとなる軌道により合致する結果が得られた. これらの結果から, 既存のコントローラの情報をもっと多く使うことで運動の精度が向上することが理解できる.

5. おわりに

本研究では, ロボットの状態変数を軌道アトラクタに引き込ませることで運動を作り出すコントローラ設計法の改良と, 既存のコントローラを用いた, 場の重ね合わせと状態空間写像に基づいた新たな運動の生成法を提案した. 以下に, 研究の成果を示す.

1. アトラクタを生成するコントローラ的设计方法に関し, 現実的な場の定義に基づいて設計する方法を提案した.
2. 状態空間写像に基づいてコントローラのパラメタライズを行う方法を提案し, これを用いて新たな運動を創発させるコントローラ的设计法を提案した.
3. これらの方法をタップダンスロボットの足踏み動作に応用し, シミュレーションによって有効性を検証した.

謝辞

本研究は文部科学省科学研究補助金若手研究 (A) 「力学的観点に立ったロボットの身体と知能の相互発達と運動の創発」の支援を受けた.

参考文献

- [1] 岡田, 村上: 軌道アトラクタを用いたロボットの同調運動制御によるコミュニケーション原理の表現, 日本ロボット学会誌, Vol.25, No.4, pp.545-553, 2007
- [2] 渡辺, 岡田: 軌道アトラクタに基づく身体・運動記号の設計とその物理的意味表現, 第 25 回 日本ロボット学会学術講演会, 1D14(CD-ROM), 2007
- [3] T.W.Deacon: The Symbolic Species - The Evolution of Language and the Brain -, W. W. Norton & Company, 1997.