

解 説

力学的情報処理による運動の表現と設計

Motion Representation and Design based on Dynamics-based Information Processing

岡田 昌史* *東京工業大学大学院理工学研究科機械物理工学専攻

Masafumi Okada* *Department of Mechanical Sciences and Engineering, Tokyo TECH

1. はじめに

一般にロボットの運動は、運動を定める目標運動パターンとそれに追従させるための安定化コントローラによって実現されてきている。これら二つはロボットの身体と環境に依存して変化するため、異なるロボット、環境に対しすべて設計しなければならない。これはすべての場合での書き下しに相当し、情報爆発を引き起こす。

これに対し、信号処理の分野では時系列データの圧縮法に力学系が用いられる場合がある。ある時系列データがあるダイナミクスから発生している場合、データには何らかの時間的相関が存在しており、これを利用することでデータの圧縮が可能である。つまり、

$$x[k+1] = f(x[k]) \quad (1)$$

と表される離散時間力学系（差分方程式）に対して、ある初期値 $x[0]$ から順番に得られる $x[k]$ ($k = 1, 2, 3, \dots$) が時系列データ

$$\Xi = \left[\xi_1 \quad \xi_2 \quad \xi_3 \quad \dots \right] \quad (2)$$

に一致するよう $f(x)$ を設計すれば、この力学系は Ξ を記憶したものとなり、しかも、時系列データの情報に比べ関数 f のパラメータは少なく、大きなデータ圧縮となる。これは時系列データの時間的な流れを力学系のパラメータという形で圧縮していると考えられ、データの時間的圧縮であると言えるであろう。Kawashima らは線形力学系を用いて人の唇の動き（画像情報）を力学系に記憶させ、力学系が予測した次時刻データと実際のデータのマッチングをとることでデータの認識を行っている [1]。ただし、線形力学系の利用であるため、データの認識を可能としても生成は難しい。また、Heikkonen らの方法も自己組織化マップを用いて時系列データの時間的圧縮を行い、データの分類

を行うものである [2]。リカレントニューラルネットワークを用いて時系列データを学習させる方法も、基本的にはこの考え方と同一であり、 $f(x)$ の関数の構造定義の差はあるものの、力学系を用いたデータ圧縮技術が有効な成果をあげてきた。これらの結果は信号の時間的相関に基づくものであると言える。また、 x は一般に n 次元ベクトルであり、そのデータ量は少なくない。そのすべてを用いて力学系を設計することは計算量の膨大さと逆問題の不安定性を導き、適切な結果を得にくい。そこで、 x の空間的相関を用いてデータを圧縮（低次元化）する方法も提案されている。Lee らはニューラルネットワークを用いた非線形主成分分析（NLPCA）によって人間のデータの空間的低次元化を行い [3]、Tatani らはこれを階層化することで人間の全身運動の低次元化を行った [4]。

一方、ロボットの制御という観点からすると、目標運動パターンは時間のタイムスタンプを持ったデータであるため、ロボットのダイナミクスとの整合性を持つ必要がある。2足歩行ロボットにおける ZMP の軌道計画とそれへの安定化制御はその典型的な例である。さらに、アクチュエータのパワー、消費電流、骨格の強度、静止摩擦力などの拘束条件を加えると、満たすべき制約条件の数は少なくない。そのため、すべての状況において、すべての場合を想定して制約条件を満たす目標運動パターンを計算することは困難であると言えるであろう。ダイナミクスの整合性の条件を緩和させるために、コントローラのロバスト性向上の研究も多くなされているが、その限界は近く、ロボットと環境の精密なモデル化と逆動力学解析といった作業を毎回行わなければならないのが現状であろう。これに対し、人間の運動では目標の運動パターンは存在しない。我々は「歩く」という運動に対して、手足をどのように動かせば良いのかを常に考えているわけではなく、また、地面が凸凹のときに「このように歩こう」といった考えも持たない。我々の「歩く」という概念と身体が結合することで歩く動作をしはじめ、そのときのセンサ信号（身体の傾き、足裏の接触）からその時々に対応した動きで「歩く」動作をし続けていると解釈することができる。そのため、まったく同じ運動を再現することはできず、一定周期の運動をし続けることでもで

原稿受付 2007年3月9日

キーワード: Dynamics-based Information Processing, Attractor Design, Nonlinear Dynamics, Motion Reduction

*〒152-8552 目黒区大岡山 2-12-1

*Meguro-ku, Tokyo

きない。これは数学的には、概念と身体が結合し、身体のダイナミクスがある軌道アトラクタへと引き込まれているという現象によって説明できる。すなわち、運動パターンはダイナミクスの動きが我々の存在する世界に現れた「結果」であって、目標運動パターン先にありきという制御手法とは異なる現象であると言える。これにより、人間は環境に応じた目標運動パターンすべてを記憶しておく必要はなく、情報量の減少をもたらしているのみならず、アトラクタへの引き込みをもって安定化を達成している。この解釈は、運動は身体を通して実体化する (embodiment) ということから「身体性」[5] の概念とも一致しており、さらに、記号 (概念) が身体を通して実体化するという Deacon の「言語と脳の共進化」[6] の考え方も一致している。この考え方をロボットに応用することで、ロボットは目標運動パターンを持たずに動き、結果としてその動きが「運動」へとつながる制御系の設計を可能とする。

本解説では、筆者らがこれまでに提案してきた、力学系を用いたデータの圧縮法、ロボットの制御法に関して紹介する。具体的には、

- (1) 主成分分析によってロボットの関節角データを空間的に低次元化し、
- (2) 低次元化されたデータと力学系を用いて時間的圧縮を行う方法に関して紹介する。特に、ここでは運動を力学系のアトラクタとすることで、適当な初期値からでも目標の運動を生成する差分方程式を設計する。
- (3) また、この方法はロボットのダイナミクスとの親和性が高く、この方法を応用したロボットの制御法を紹介する。ここでは、ロボットの身体ダイナミクスをある閉曲線へと安定化させるものであり、アトラクタの設計によってロボットの安定な運動を創発させるものである。

以下に、それぞれの章でその詳細を紹介する。

2. 主成分分析による時系列データの空間的圧縮

本章では、主成分分析によってロボットの全身運動データを空間的に圧縮する方法について紹介する。なお、この内容は文献 [7] に詳しい。

いま、ロボットの全身運動データ X

$$X = \begin{bmatrix} x[1] & x[2] & \cdots & x[N] \end{bmatrix} \in R^{n \times N} \quad (3)$$

が与えられているとする。これは例えばロボットの全身角度データであり、 $x \in R^n$ である。このとき、例えば歩行データであれば膝関節角度と股関節角度の相関は大きく、これに基づいてデータの空間的圧縮が可能ならずである。ここで、 X の特異値分解

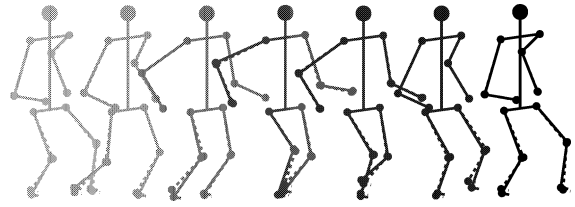


図 1 歩行データとそれを低次元化・復元したデータ

$$X = USV^T \quad (4)$$

を考える。 S は特異値を大きい順に並べた行列

$$S = \text{diag} \left\{ \sigma_1 \quad \sigma_2 \quad \cdots \quad \sigma_n \right\} \quad (5)$$

なので、 $\sigma_m \gg \sigma_{m+1}$ となる m で

$$X = \begin{bmatrix} U_m & U_{n-m} \end{bmatrix} \begin{bmatrix} S_m & 0 \\ 0 & S_{n-m} \end{bmatrix} \begin{bmatrix} V_m^T \\ V_{n-m}^T \end{bmatrix} \quad (6)$$

$$\simeq U_m S_m V_m^T \quad (7)$$

と分解、近似すると

$$X \simeq A \hat{X} \quad (8)$$

$$A = U_m S_m \in R^{n \times m} \quad (9)$$

$$\hat{X} = V_m^T = \begin{bmatrix} \hat{x}[1] & \cdots & \hat{x}[N] \end{bmatrix} \in R^{m \times N} \quad (10)$$

として n 次元ベクトル x が m 次元ベクトル \hat{x} へと低次元化される。なお、 A は復元を行う行列であり、 $A^\# = S_m^{-1} U_m^T$ は低次元化を行う射影行列である。

この方法を用いて、実際にヒューマノイドロボットの歩行データ (関節角度データ) の圧縮を行ってみる。ロボットの歩行 1 周期データ $X \in R^{20 \times 120}$ (左右各脚 6 自由度、腕 4 自由度で合計 20 自由度) を前述の方法を用いて $\hat{X} \in R^{3 \times 120}$ に圧縮し、これを復元することで再び歩行データを得た。このときの結果を図 1 に示す。鎖線は元のデータ、実線は低次元化・復元されたデータである。二十次元から三次元という大きな圧縮にも関わらず、元とほぼ同じデータが再現できることが分かる。これは本来歩行データが手足の動きの相関が大きいために可能となっているもので、どの程度まで低次元化できるかは元のデータに依存する。

3. 力学系を用いた時系列データの時間的圧縮

空間的に低次元化されたデータを力学系に埋め込み、アトラクタとしてデータを復元する。いま、低次元化されたロボットの全身運動データ \hat{X}

$$\hat{X} = \begin{bmatrix} \hat{x}_1 & \hat{x}_2 & \cdots & \hat{x}_N \end{bmatrix} \in R^{m \times N} \quad (11)$$

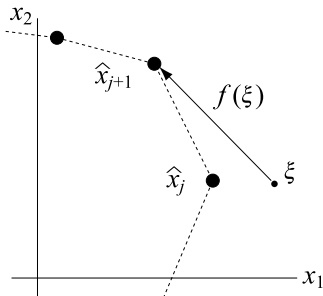


図2 ベクトル場の定義

が与えられているとする．この運動が周期的で1周期分のデータであるとする．このデータは m 次元空間の中の一つの閉曲線 C を描く．そこで，次式の m 次元力学系

$$x[k+1] = x[k] + f(x[k]) \quad (12)$$

が C を軌道アトラクタとするよう $f(x)$ を設計しよう．ここで，力学系が閉曲線を軌道アトラクタとするということは，ある初期値 $x[0]$ から出発した $x[k]$ が $k \rightarrow \infty$ とともに C へと収束し，その後， C の上を動き続けることを意味するものであり，ここでは $x[0]$ が C 上にあるときに $x[k]$ が C の上を動き続けるリミットサイクルとは区別するものとする．すなわち，リミットサイクルは平衡軌道であり，アトラクタは安定な平衡軌道とする． $f(x)$ 設計のために，ここでは m 次元空間の中に場を定義し，これを関数近似することで $f(x)$ を求めるといった方針をとる．いま，図2のように x 空間の中に点 ξ を定め，これに一番近い \hat{X} 上の点を \hat{x}_j とする．このとき，点 ξ が次の時刻に \hat{X} に近付くためには

$$f(\xi) = \hat{x}_{j+1} - \xi \quad (13)$$

となればよい．この関係から $\{\xi, f(\xi)\}$ の組み合わせが一つ定まる．そこで，空間の中に多くの ξ_i ($i = 1, 2, \dots$) を定義することで $\{\xi_i, f(\xi_i)\}$ の組み合わせが数多く得られ，ベクトル場が定義されるのでこれを関数近似する．具体的に， $f(\xi)$ を ξ の l 次多項式として近似すると，

$$f(x[k]) = \Theta \phi(x[k]) \quad (14)$$

で表される関数となる．ただし， Θ は多項式の係数からなる行列で $\phi(x)$ は x を多項式に展開する関数である． $m = 2$ ， $l = 2$ の場合であれば，

$$\phi(x) = \begin{bmatrix} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \end{bmatrix}^T \quad (15)$$

$$x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \quad (16)$$

である．このとき， Θ は

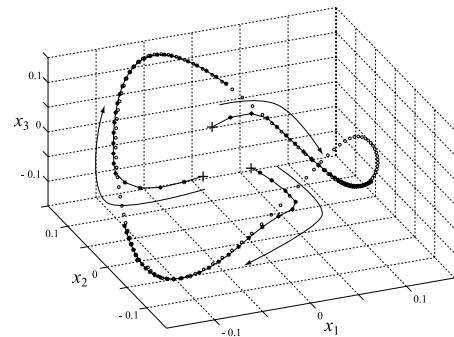


図3 設計した力学系の動き

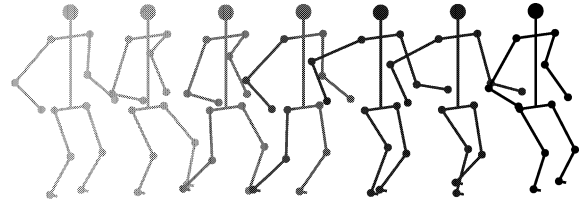


図4 力学系から復元したデータ

$$\Theta = F\Phi^\# \quad (17)$$

$$F = \begin{bmatrix} f(\xi_1) & f(\xi_2) & f(\xi_3) & \dots \end{bmatrix} \quad (18)$$

$$\Phi = \begin{bmatrix} \phi(\xi_1) & \phi(\xi_2) & \phi(\xi_3) & \dots \end{bmatrix} \quad (19)$$

で求められる．これにより，閉曲線 \hat{X} をアトラクタとする力学系が x の多項式として求められることになる．実際に，2章で求めた，歩行の三次元低次元化データを用いて力学系を設計した．ただし， $l = 4$ とした．このときの力学系の動きを図3に示す． \circ は \hat{X} を示し，実線が力学系の動きを示す． $+$ は初期値を表している．この結果からアトラクタを持つ力学系が設計できていることが分かる．

次に，この力学系の動きからロボットの動きを復元した．この結果を図4に示す．力学系の初期値等の理由から図1と運動の位相は合っていないが，歩行運動が生成される様子が理解できる．ここで特記すべきことはこの運動が力学系のアトラクタとして埋め込まれていることにある．すなわち，適当な初期値を与えても力学系はアトラクタに引き込まれ，運動を生成する．ニューラルネットワークにも引き込みの効果があることが知られているが保証されないため，結果はリミットサイクルを形成するに留まり，アトラクタまで議論することは難しい．

4. 力学系を用いたロボットの運動創発

前章で設計した力学系は単に時系列データの情報を埋め込んだもので，ロボットのダイナミクス情報は含んでいない．そのため，得られた運動パターンをそのままロボットに流し込んでも，安定化コントローラがなければ実際には運動を実現することはできない．ただし，力学系はロボッ

トのダイナミクスと親和性が高く、この方法を基礎としてロボットをアトラクタで制御する方法を文献[8]で提案した。また、設計法の修正を[9][10]で行った。本章ではこの方法を紹介し、運動創発システムの設計法を示す。

ここでは簡単のため線形ダイナミクスの場合について述べる。いま、ロボットのダイナミクスが

$$x[k+1] = Ax[k] + Bu[k] \quad (20)$$

の n 次の差分方程式で表されたとする。これはロボットの運動方程式を離散化することで得られる。これに対し、コントローラを

$$u[k] = h(x[k]) \quad (21)$$

としたとき、式(20)と(21)の連立差分方程式がある軌道

$$\hat{X} = \begin{bmatrix} \hat{x}_1 & \hat{x}_2 & \cdots & \hat{x}_N \end{bmatrix} \in R^{n \times N} \quad (22)$$

をアトラクタとするよう、 $h(x)$ を求める。3章では仮想的な力学系を設計したため、式(12)の $f(x)$ は自由に設計できたが、ここではこれが式(20)のような条件を持ち、設計できるのは u に関する部分のみである点が大きく異なる。ただし、設計の基本指針は同じであり、 $\{x, u\}$ の組を数多く求め、これに多項式の関数近似を行うものである。

式(20)から次式が導ける。

$$X_{k+1} = \Gamma x[k] + HU_k \quad (23)$$

$$X_{k+1} = \begin{bmatrix} x^T[k+1] & \cdots & x^T[k+p] \end{bmatrix}^T \quad (24)$$

$$\Gamma = \begin{bmatrix} A \\ \vdots \\ A^p \end{bmatrix}, \quad H = \begin{bmatrix} B & \cdots & 0 \\ \vdots & \ddots & \vdots \\ A^{p-1}B & \cdots & B \end{bmatrix} \quad (25)$$

$$U_k = \begin{bmatrix} u^T[k] & \cdots & u^T[k+p-1] \end{bmatrix}^T \quad (26)$$

これは式(20)のダイナミクスに初期値 $x[k]$ から入力列 $u[i]$ ($i = k, k+1, \dots, k+p-1$) を入力すると、 x が $x[j]$ ($j = k+1, k+2, \dots, k+p$) と変化することを表したものである。これより、式(22)の \hat{X} 上の点 \hat{x}_k の近傍から出発した $x[k]$ が \hat{X} に収束しながら進むための入力列は

$$U_k = H^\# (\hat{X}_{k+1} - \Gamma x[k]) \quad (27)$$

$$\hat{X}_{k+1} = \begin{bmatrix} \hat{x}^T[k+1] & \cdots & \hat{x}^T[k+p] \end{bmatrix}^T \quad (28)$$

で与えられる。また、そのときの $x[j]$ ($j = k+1, k+2, \dots, k+p$) は得られた U_k を式(23)に代入することで求められる。これにより、 p 組の $\{x, u\}$ が得られる。前章

と同様に、空間内に多数の $x[k]$ を定義し、多数の $\{x, u\}$ の組を求めることで、式(14)と同様にコントローラ

$$u[k] = \Theta \phi(x[k]) \quad (29)$$

が得られる。式(27)の最小二乗近似は評価関数 J

$$J = \sum_{i=k+1}^{k+p} \|\hat{x}[i] - x[i]\| \quad (30)$$

の最小化であるが、本来 x には単位の異なる変数が含まれているため重み行列 W を用いて

$$J = \sum_{i=k+1}^{k+p} \|W(\hat{x}[i] - x[i])\| \quad (31)$$

とする必要がある。文献[9]ではダイナミクスの保存量に基づいて W を決定する方法を提案している。また、一般に \hat{X} は x 空間の中で細長い形をしている場合が多い。これは、多項式による関数近似を行う際に逆問題の不安定さを引き起こす。例えば、 x の要素 x_i が $x_i < 1$ であれば $x_i^\ell \rightarrow 0$ ($\ell \rightarrow \infty$) であるし、 $x_i > 1$ であれば $x_i^\ell \rightarrow \infty$ ($\ell \rightarrow \infty$) となり、関数近似の精度をあげるために多項式の次数 ℓ を大きくとることがコントローラ設計に悪影響を及ぼす。そこで、文献[10]では目標軌道が球体に近い形になるよう、 x に座標変換を施してからコントローラを設計する方法を提案している。

この方法の特徴について述べる。この設計手法では、コントローラの設計時に式(22)の目標軌道が用いられるが、ロボットが運動を実現する際には目標運動パターンは存在していない。式(20)のロボット身体と式(21)のコントローラの連立差分方程式の解として運動が現れ、得られる運動は時間の拘束を持った周期一定の運動ではなく、外乱・環境の影響によって常に周期は異なる。すなわち、身体・環境・コントローラの相互作用から創発される運動である。これは人間の運動に類似した運動創発システムである。また、ここでは式(22)は周期的であることを仮定したが、これはコントローラ設計アルゴリズムからすると必要な条件ではない。ただし、その場合 \hat{x}_N が平衡点である必要がある(運動終了時に停止が可能)。さらに、ここでは式(20)のように線形系として話を進めたが、非線形系の場合は式(20)を \hat{x}_i 周りでの線形化モデル

$$x[k+1] = A_i x[k] + B_i u[k] + C_i \quad (32)$$

とすることで同様のアルゴリズムからコントローラが設計可能である。

では、実際にこの方法を用いてロボットを制御した例を示そう。ここでは、図5に表されるタップダンスロボット

を対象としよう. このロボットは図5右図にあるように頭部でバランスをとりながら足踏み動作を行うものである. このロボットの運動方程式は図6にあるように右足接地時, 左足接地時に場合分けすることで得られる. ここで, 腰部の回転角を θ とし, 頭部の回転角を ϕ とした.

まず, \hat{X} を得るために頭部を 2.0 [Hz] のサイン関数で動かした. この結果, 図7に表されるような運動が得られた. なお, ロボットの状態変数 x は

$$x = [\theta \quad \dot{\theta} \quad \phi \quad \dot{\phi}]^T \quad (33)$$

であるが, ここでは $\theta, \dot{\theta}, \phi$ のみ示した. このとき, ロボットは安定化制御されていないため運動の成功・不成功は姿勢の初期値の選び方に大きく依存し, また, 途中で倒れてしまう現象も多く見られた. ここでは, 運動が成功した適当な時間を切り取って \hat{X} とした. 次に, 上述の方法を用いてアトラクタを設計し, ロボットの運動創発を行った. この結果を図8に示す. $+$ は初期値を表し, $x=0$ からタップダンス運動へと引き込まれる運動が生成されている様子が分かる. また, このときの実験の様子を図9に示す. なお, ここでは1周期分の様子を示した. 足を切り替えながら足踏み動作を実現していることが分かる.

先にも述べたように, 図7では頭部を 2.0 [Hz] で動かすため実現される運動は必ず 2.0 [Hz] となる. 足の滑りなどの外乱が加わった場合でも 2.0 [Hz] の運動をし続けようとするため運動が不安定になる. これに対し, 図8の結果では目標運動パターンがないため, 足の滑りが起こった場合には, 体勢の立て直しを行い, そこから運動を開始する. そのため, この結果の周期は一定ではない. その例として, 図10に θ のグラフを示す. これは, 図8の θ を横軸を時間にして表示したものである. グラフ内のはじめの周期を t_1 とし, 各ピーク間の時間と比較した. これより, 運動の周期が一定とはならないことが読み取れる. これはロボットの運動を非線形力学系の引き込み (アトラクタ) として実現した結果であり, 人間の運動創発と類似した結果を示していると言える.

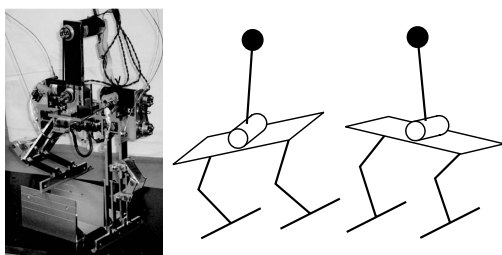


図5 タップダンスロボットとその動き

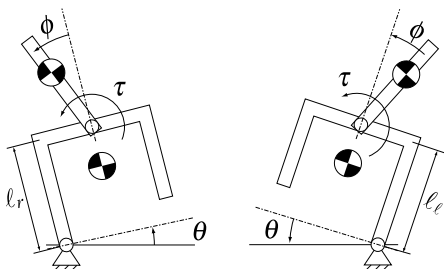


図6 タップダンスロボットのモデル化

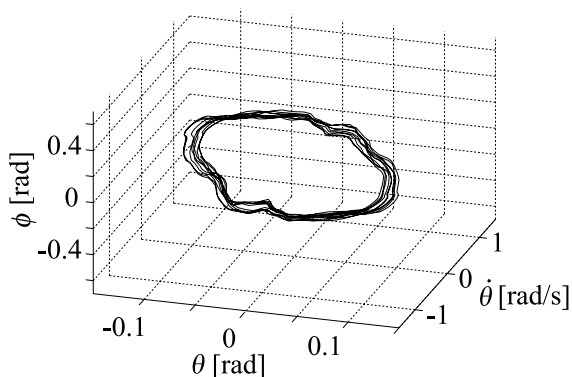


図7 目標運動パターンの獲得

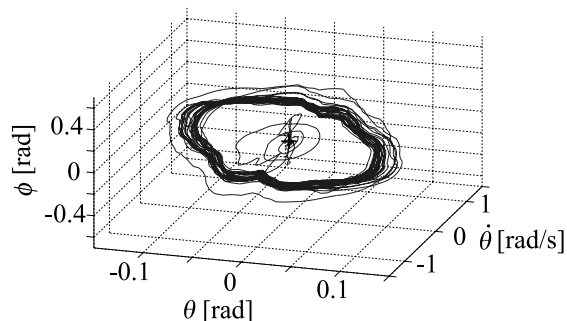


図8 アトラクタ設計によるタップダンス運動

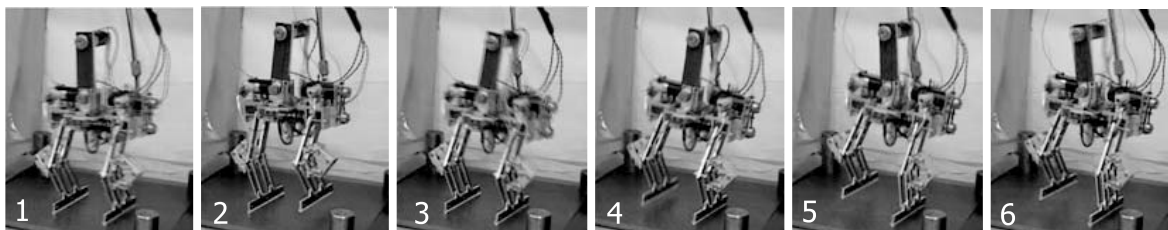


図9 タップダンス運動の実験の様子

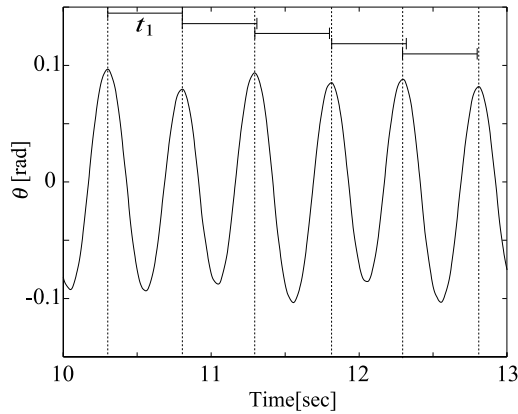


図 10 タップダンス運動の非周期性

5. おわりに

本解説では、力学系を用いたロボットの情報処理、制御を力学的情報処理として述べた。具体的には、信号処理の観点から主成分分析を用いたデータの空間的圧縮法（低次元化法）、力学系を用いた時間的圧縮法を示し、さらに、これに基づいたロボットの制御法を紹介した。特に、ロボット制御法ではロボットの身体ダイナミクスをアトラクタに引き込ませるようなコントローラを設計することで、ロボットが運動を創発するシステムの設計法について述べた。

今後の発展として、式 (29) のコントローラはロボットの身体ダイナミクスと結びついて運動を創発するが、このときの身体の情報、運動の情報が Θ に圧縮されてつまっている。ここからその情報を分離抽出することでそれらの組み合わせによって新たなコントローラの設計が可能となると予想され、その基礎的研究が文献 [11] で行われている。ここでは、 Θ は単なる数字の羅列でこれを見ただけではその意味は理解できず、身体と結びついて初めて「運動」となって現れることに注目し、 Θ が記号（概念）に相当し、これより身体記号と運動記号を抽出することでその組み合わせを可能としている。これは人間の記号操作の能力を実現することを目指したものであり、ロボットの知能設計のための指針を示している。

If-then ルールのようなアルゴリズムとしての情報処理は、人間の情報処理を文章化しこれを計算機に実行させるための方法である。フローチャートのようにその流れを見るのに適した方法で作り込みに適している。ただし、すべてを文章化する必要があるため決まった作業の実行に適しているものの応用性に欠く。これに対し、力学的情報処理は非線形力学を用いてその設計アルゴリズムを導いているものであり、情報処理自体は単なる計算として実現される

ことから、計算機に適した情報処理手法であると言えよう。今後、この方法がロボットの知能の新たな設計法を導くことを期待する。

参考文献

- [1] H. Kawashima and T. Matsuyama: "Hierarchical Clustering of Dynamical Systems based on Eigenvalue Constraints," 3rd International Conference on Advances in Pattern Recognition (S. Singh et al. (Eds.): ICAPR 2005, LNCS 3686), pp.229–238, 2005.
- [2] J. Heikkonen, K. Kaski, T. Koskela and M. Varsta: "Temporal sequence processing using Recurrent SOM," Proc. of 2nd International Conference on Knowledge-Based Intelligent Engineering System (KSE'98), pp.290–297, 1998.
- [3] C. Lee and Y. Xu: "Reduced-dimension Representations of Human Performance Data for Human-to-robot Skill Transfer," Proc. of IEEE/RSJ International Conference on Intelligent Robotic Systems, vol.3, pp.1956–1961, 1998.
- [4] K. Tatani and Y. Nakamura: "Dimensionality Reduction and Reproduction with Hierarchical NLP Neural Networks Extracting Common Space of Multiple Humanoid Motion Patterns," Proc. of IEEE International Conference on Robotics and Automation (ICRA2003), pp.1927–1932, 2003.
- [5] R. Pfeifer and C. Scheier: Understanding Intelligence. Bradford Books, 2001.
- [6] T.W. ディーコン (金子隆芳 訳): ヒトはいかにして人となったか—言語と脳の共進化—. 新曜社, 1999.
- [7] 岡田, 中村: "脳型情報処理を行う力学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.1050–1060, 2004.
- [8] 岡田, 大里, 中村: "非線形力学系のアトラクタ設計によるヒューマノイドロボットの運動創発", 計測自動制御学会論文誌, vol.41, no.6, pp.533–540, 2005.
- [9] M. Okada: "Attractor Design of Dynamics based on Energy Distance in State-Space for Linear System," International Journal of Simulation Systems, Science and Technology—Special Issue on Mechatronics Technology—, vol.7, no.8, pp.47–54, 2006.
- [10] 村上, 岡田, 岩附: "軌道アトラクタ設計によるロボットの運動創発とタップダンスロボットの制御—力学特性の大きな変化を伴う運動への適用—", 第 12 回日本 IFToMM 会議シンポジウム前刷集, pp.37–42, 2006.
- [11] 岡田昌史: "軌道アトラクタによるロボットの運動記号と身体記号の設計", 第 24 回日本ロボット学会学術講演会予稿集 CD-ROM, 1D31, 2006.



岡田昌史 (Masafumi Okada)

1969年3月21日生。1992年3月京都大学工学部精密工学科卒業。1996年9月同大学大学院応用システム科学専攻博士課程修了、博士(工学)。1996年10月日本学術振興会特別研究員(PD)。1997年2月東京大学大学院工学系研究科リサーチ・アソシエイト。2000年4月同大学大学院工学系研究科講師。2004年4月東京工業大学大学院理工学系研究科助教授、2007年4月准教授となり現在に至る。ヒューマノイドロボットの機構開発、力学系を用いた情報処理システムの研究に従事。計測自動制御学会、IEEEの会員。(日本ロボット学会正会員)