

力学的情報処理におけるアトラクタ設計法に関する研究

—逐次設計による可塑性の導入と階層化設計による大規模化—

岡田昌史^{*1} 中村大介^{*2} 門根秀樹^{*3}
梶山博史^{*4} 中村仁彦^{*3}

Research on Attractor Design for the Dynamics-based Information Processing System

—On-line Design for Plastic Property and Hierarchical Design for Large-scale Systems—

Masafumi Okada^{*1}, Daisuke Nakamura^{*2}, Hideki Kadone^{*3},
Hiroshi Kajiyama^{*4} and Yoshihiko Nakamura^{*3}

Because robot motions are generated through the interaction between the robot body and its environments, it is necessary for the robot to modify the motion pattern that is prepared a priori, depending on the situation. We have proposed dynamics-based information processing system in which a dynamical system memorizes and replays the robot whole body motion using the entrainment phenomenon of the nonlinear dynamics. In this paper, we propose the motion pattern modifying method based on the re-design of the dynamics using the on-line least square method, and the hierarchical design method of the dynamical system for the large-scale system to generate the context dependent motion sequence.

Key Words: Dynamics-based Information Processing, Hierarchical Structure, On-line Design Method, Motion Generation and Transition, Humanoid Robot

1. はじめに

ロボットの運動生成では、運動パターンが用意され、これに動きを追従させる安定化フィードバックコントローラによって運動を実現させるのが一般である。しかし、実際にはどの初期姿勢から運動を開始するか、ある運動からある運動へ変化する場合にどのタイミングで実行するかによってその遷移過程は多種にわたり運動は変化する。一般には運動パターンは始点と終点を持つため遷移のタイミングは固定され、その遷移パターンはフィードバックコントローラのロバスト安定性にゆだねられてきた。そのため、ロボットが運動を変える場合には特定の姿勢を経由したり、運動の切り換え時に無理な動作をとったりする。これは、本来連続的なロボットの運動を、離散的な情報処理系で操作しようとする結果である。

これに対し、Freeman らのウサギの嗅球電位における実験的

な結果 [1] や Tsuda が提唱するカオスの遍歴 [2] により、力学系と生体内における情報処理系との密接な関係が示された。この結果をもとにして、力学系を用いた情報処理系が注目を浴びている。力学系は連続時間システムであるため、時間の流れの中にある運動を扱うものとして適しているといえる。関口らはカオスの持つ位相推移性を利用して移動ロボットの運動制御を行い [3] [4]、琴坂らは神経振動子を用いてロボットのリズム的なドラム打撃動作を生成した [5]。木村らや土屋らは神経振動子を用いて 2 足・4 足ロボットの運動生成を行った [6] [7]。これらは主に現象論からのアプローチであり、パラメータの調整が情報処理系実現のための大きな役割を担っている。また、川嶋らは力学系によって時系列信号の認識を行う方法を提案している [8]。これは力学系の出力する予測値と実際の値の比較を用いるもので、力学系が写像関数として用いられている。この中で、筆者らはこれまでに N 次元空間内の閉曲線をアトラクタとする力学系の設計法と、これを用いたロボットの運動生成法を力学的情報処理系として提案した [9]。この方法では、(1) 力学系へのアトラクタ (閉曲線) の埋め込みによる運動の記憶、(2) 力学系の自律的運動 (アトラクタへの収束) による運動の生成、(3) 入力信号によるアトラクタへの引き込みに基づく運動の認識、が実現されている。運動パターンの時系列データを力学系のアトラクタとして埋め込むことで運動の始点・終点は存在せず、運動の遷移過程は力学系の引き込みの過程という連続的な

原稿受付 2004 年 6 月 9 日

^{*1} 東京工業大学大学院理工学研究科機械物理学専攻

^{*2} マッキンゼーアンドカンパニー

^{*3} 東京大学大学院情報理工学系研究科知能機械情報学専攻

^{*4} 東京大学大学院工学系研究科産業機械工学専攻

^{*1} Department of Mechanical Science and Engineering, Tokyo Institute of Technology

^{*2} McKinsey and Company Inc.

^{*3} Department of Mechano-Informatics, University of Tokyo

^{*4} Department of Engineering Synthesis, University of Tokyo

現象として表現される．この方法では，力学現象が情報処理であるという観点に立ち，アトラクタを有する力学系の創発を待つのではなく，数理的方法によって設計するという立場をとっている．足立らは，リアプノフ関数に基づいて大域的な安定性を保証しながらアトラクタを有する力学系の設計法を示し，ロボット運動生成を実現した [10]．しかし，これは次に述べる力学系の構造変化や階層化を容易としない．

実際にロボットが実環境の中で運動を実行する場合，環境の変化により運動の修正が必要であろう．力学的情報処理では運動がアトラクタとして記述されているので，ロボットの運動を変化させるためには力学系の構造を変化させ，アトラクタを変形させる必要がある．これは力学系の可塑性を意味する．情報処理系が実世界の中で機能するためには，環境の変化とともにその構造を変化させるといった可塑性を持つことが重要な意味をなす．

一方，より多くの情報を処理するためには情報処理系の大規模化が必要であろう．しかし，実際に大規模な情報処理系を一度に設計することは困難であるうえ，系の構造が不明確になり修正や機能の追加・削除を難しくする．これに対し，人間の脳皮質は感覚性情報を統合して認識し，過去の経験により得た学習内容を照合するなど，中枢神経の最も中心的な神経機能を果たすものであり，前頭葉，側頭葉，頭頂葉，後頭葉の領域に大別される．側頭葉では味覚・聴覚・嗅覚・言語を，頭頂葉では体性感覚，後頭葉では視覚といったように外部からの感覚信号が処理されている．前頭葉ではこれらの情報の集結，理解，意思決定を行い，また，小脳では骨格筋の緊張調整を行い運動を制御する [11]．それぞれの領域は外部からの入力信号に対して固有のダイナミクスで変化し，一つの周期状態へと落ち着く．さらに，アトラクタから特徴量を抽出し他の領域へ情報を出力している [12]．このように，各領域それぞれが役割を持ち固有の情報処理を行うといったネットワーク構造が見られており，これにより実世界の中の膨大な量の情報を柔軟に処理していると考えられる．また，このネットワークがダイナミクスで変化することから外部からの信号の時間変化も情報として取り入れられ，過去の経歴が反映した文脈依存性も膨大な量の情報処理に大きな役割を担っているといえる．さらに，学習効果によってその構造を変化させ，変わりゆく実環境への対応を可能にしているものと考えられる．

本論文では，ロボットの行動獲得，学習，認識，生成といった知能の実装ではなく，これらを力学的に実現する際に基礎となる要素技術の確立を目指し，力学的情報処理における力学系の設計法を考察する．力学的情報処理系が上述のような知能としての機能を持つためには，

- (1) 力学系をデータや目的に応じて逐次的に変化させる漸化的設計法
 - (2) スケイラブルな力学系の設計法，すなわち，力学系の大規模化による計算量の増加によって破綻しない設計アルゴリズム
 - (3) 膨大な情報の組み合わせに時間的な流れを考慮した文脈依存型情報処理系の設計法
- などの数学的な要素技術を確立させる必要がある．そこで，本

論文では力学的情報処理系への可塑性の導入，大規模構造の設計を主目的とし，

- (1) 力学系へのアトラクタの埋め込みを逐次的に行う方法を提案し，さらに，これに忘却係数を設定することで力学系の設計に学習的な効果を持たせ，古い情報を忘れ新たな情報を記憶するといった可塑性を持たせる．
- (2) すでに設計した力学系のアトラクタの形を拘束条件に基づいて変形させ，ロボットの運動を修正する．
- (3) 力学的情報処理系の階層化設計を行い，センサ信号の認識や運動の生成を行うような役割分担を行う力学系とそのネットワーク化により，大規模情報処理系の設計を行う．
- (4) 構成したネットワークを統括する力学系を設計し，過去の情報を用いた文脈依存の運動決定機構を設計する．

に関する力学系およびシステムの設計法について述べる．また，上半身型のヒューマノイドロボットに実装し運動の生成と遷移を観察する．これらの組み合わせにより，力学的情報処理系による行動獲得，行動学習，行動認識，行動生成を行う知能機械システムの設計への展開が期待される．

2. 力学的情報処理の逐次設計と構造の変化

2.1 力学的情報処理

本節では力学的情報処理に関して述べる [9]．いま，ロボットの全身運動 \mathcal{M} を考える．この運動の時系列データ $\xi[k]$ (例えば関節角度) から得られる行列 Ξ

$$\Xi = \begin{bmatrix} \xi[1] & \xi[2] & \cdots & \xi[m] \end{bmatrix} \quad (1)$$

$$\xi[k] = \begin{bmatrix} \xi_1[k] & \xi_2[k] & \cdots & \xi_N[k] \end{bmatrix}^T \quad (2)$$

は N 次元空間内における曲線 C 上の点の集合として表現される．ただし， m はデータの個数， N は例えばロボットの自由度を意味する． \mathcal{M} が周期的な運動ならば C は閉曲線となり，ここでは C が閉曲線であるとする．

次に，以下の離散時間差分方程式で表される力学系を考える．

$$x[k+1] = x[k] + f(x[k]) \quad (3)$$

これが閉曲線 C をアトラクタとするとき，すなわち，ある初期値 $x[0]$ から出発して $k \rightarrow \infty$ で $x[k]$ が閉曲線 C に収束するとき，この力学系はロボットの運動 \mathcal{M} を記憶しており，力学系が自律的に動くときの状態変数 $x[k]$ は運動 \mathcal{M} の時系列データ Ξ を再生する．力学的情報処理は式 (3) の力学系が与えた閉曲線をアトラクタとするように設計することで，ロボットの全身運動を記憶・再生・認識する情報処理系の設計を行うものである．

2.2 力学系の逐次設計

式 (3) の力学系は $f(x[k])$ が N 次元空間内のベクトル場を定義する関数とみなすことで設計可能である．文献 [9] ではこれを多項式として近似する方法を提案しており，この設計アルゴリズムを付録 A に示す．式 (A.10) では， $f(x[k])$ を定義するパラメータ行列 Φ を最小二乗法で求めた．本論文では逐次最小二乗法 [13] を用いることで力学系の逐次計算を行う．これにより，常に入力される信号を記憶する力学系が設計され，運

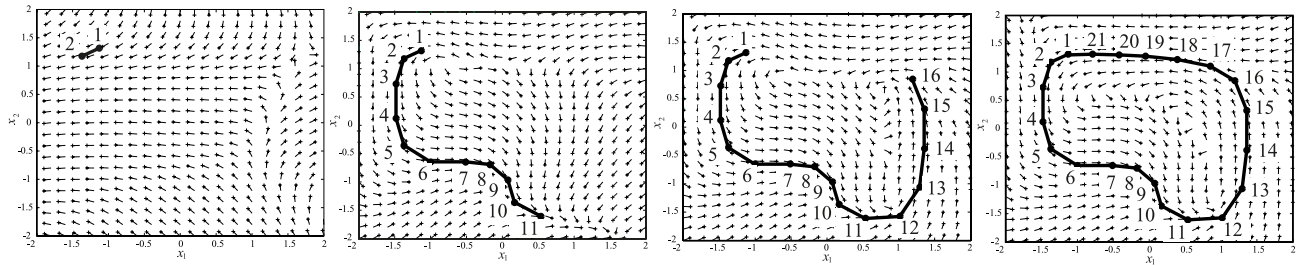


Fig. 1 On-line design of dynamics

動の学習が可能となる。

時刻 m において, ある正則な対象行列 P_m とパラメータ行列 Φ_m , アトラクタとする軌道上の点 $\xi[m+1]$ と $\xi[m]$ が与えられているとする. このとき, $\xi[m]$ を中心として $\xi[m+1] - \xi[m]$ に垂直な半径 r の $N - 1$ 次元の球体を考え, この中に適当な ℓ 個の点 $\eta_m^i (i = 1, 2, \dots, \ell)$ を設定する. さらに, 各 η_m^i に関して, 式 (A.1) に基づいて $f(\eta_m^i)$ を定義する. ただし, $\xi^{\eta}[m] = \xi[m]$ とする. 逐次最小二乗法は以下の式 (4), (5) の繰り返しにより実行される.

$$P_m^{i+1} = P_m^i - \frac{P_m^i \theta(\eta_m^i) \theta^T(\eta_m^i) P_m^i}{1 + \theta^T(\eta_m^i) P_m^i \theta(\eta_m^i)} \quad (4)$$

$$\Phi_m^{i+1} = \Phi_m^i + (f(\eta_m^i) - \Phi_m^i \theta(\eta_m^i)) \theta^T(\eta_m^i) P_m^{i+1} \quad (5)$$

ただし, $i = \ell$ のときは $P_m^{\ell+1} = P_{m+1}$, $\Phi_m^{\ell+1} = \Phi_{m+1}$ となる. ここでは, 以下の逆行列補題を用いており (A は正則),

$$(A + BC)^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1} \quad (6)$$

P_m は式 (A.12) の Θ を用いると

$$P_m = (\Theta \Theta^T)^{-1} \quad (7)$$

に相当する.

Fig. 1 は $N = 2$ のときに逐次的に力学系を設計したときのベクトル場 $f(x[k])$ の変化の様子を示している. 「+」はベクトル先端を表し, ベクトルの始点におけるベクトル場を表現した. また, 見やすさのためベクトルの長さは正規化した. アトラクタを持つ力学系が設計されていく様子が理解できる. また, Fig. 2 に設計された力学系の動きを示す. 五つの初期値「●」から出発した力学系はアトラクタへと引き込まれている.

2.3 忘却係数による力学系の構造変化

次に力学系の逐次設計法に忘却係数を設定する. いま, 式 (A.10) に対して,

$$\Phi = \hat{F} \hat{\Theta}^{\#} \quad (8)$$

$$\hat{F} = \begin{bmatrix} \alpha F & f(\eta_{m+1}) \end{bmatrix} \quad (9)$$

$$\hat{\Theta} = \begin{bmatrix} \alpha \Theta & \theta(\eta_{m+1}) \end{bmatrix} \quad (10)$$

を考える. ここで, α は重み係数である. これは重み付き最小二乗であり,

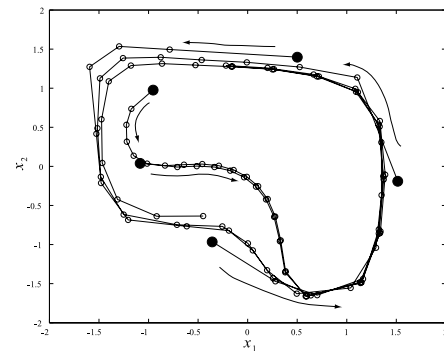


Fig. 2 Motions of the dynamics from initial positions

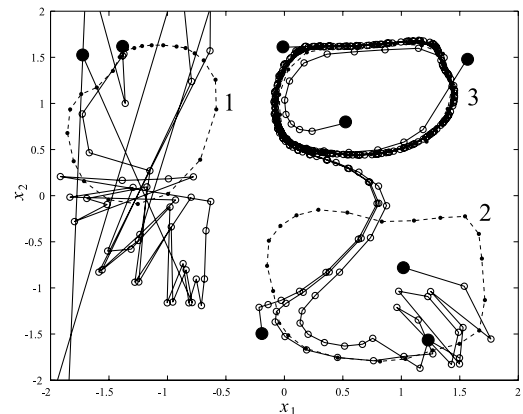


Fig. 3 Motions of dynamics with weighted on-line algorithm

$$P_m = \alpha^{-2} P_m \quad (11)$$

とおいた後に, 式 (4), (5) によって Φ を計算することで解が得られる. 証明を付録 B に示す. α を $0 < \alpha < 1$ とし, 運動の時系列データから式 (11), (4), (5) の繰り返し計算を用いて力学系を計算することにより, α を忘却係数とした逐次設計が可能となる. 実際の計算では, 式 (4), (5) の計算において, $i = 1, 2, \dots, \ell - 1$ のときは $\alpha = 1$ とし, $i = \ell$ のとき $0 < \alpha < 1$ とする. これにより行動の変化に伴ったアトラクタの修正や, 昔の行動を忘却し最近の行動を記憶する力学系の設計が可能となる.

Fig. 3 は忘却係数を設定し, 1→2→3 の順番に閉曲線を埋め込んだときの力学系の動きを示す. 「●」は初期値を表す. 初めに埋め込んだ閉曲線 1 は忘却効果によりアトラクタとなっていない. 閉曲線 2 は一部がアトラクタとして残っており, 最後に

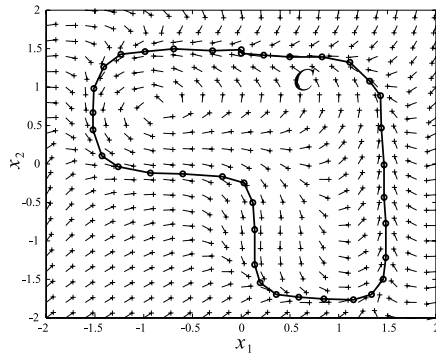


Fig. 4 Designed dynamics using on-line least square method

記憶した閉曲線 3 のみがアトラクタとなっている．この結果は力学系の可塑性を示し，アトラクタの形を変化させることを可能としている．

2.4 拘束条件に基づくアトラクタの変化

上記のように忘却係数を用いた逐次最小二乗法によりアトラクタを変形させることが可能であるが，本節では，すでに設計された力学系に拘束条件を加えて再設計し，アトラクタを変形させてロボットの運動を変化させる方法を提案する．例えば，運動するロボットの軌道に障害物が入り，これを避けるように運動を変えろといった場合に有効である．

いま， m 個のデータ $f(\eta_1), \dots, f(\eta_m)$ と $\theta(\eta_1), \dots, \theta(\eta_m)$ に基づいて最小二乗法により Φ_m が得られているものとする． Φ_m の行数を j として以下の $\lambda (< j)$ 個の拘束条件を考える．

$$F_\lambda = \Phi_\lambda \Theta_\lambda \quad (12)$$

$$F_\lambda = \begin{bmatrix} f(\eta_{m+1}) & \cdots & f(\eta_{m+\lambda}) \end{bmatrix} \quad (13)$$

$$\Theta_\lambda = \begin{bmatrix} \theta(\eta_{m+1}) & \cdots & \theta(\eta_{m+\lambda}) \end{bmatrix} \quad (14)$$

これは λ 個の点におけるベクトル場の向きを拘束条件としていることを意味する．このとき，式 (12) を満たしながら，

$$J = \|F - \Phi_\lambda \Theta\|^2 \left(= \text{trace} \left([F - \Phi_\lambda \Theta] [F - \Phi_\lambda \Theta]^T \right) \right) \quad (15)$$

を最小化する Φ_λ は次式で与えられる．

$$\Phi_\lambda = \Phi_m + (F_\lambda - \Phi_m \Theta_\lambda) \left(\Theta_\lambda^T P_m \Theta_\lambda \right)^{-1} \Theta_\lambda^T P_m \quad (16)$$

ただし， P_m は式 (7) で定義されるもので，これは逐次最小二乗法によって得られる．証明を付録 C に示す．

この方法を用いて，力学系のアトラクタを変化させる例を示す．まず，逐次最小二乗法を用いた方法により，力学系を設計した．Fig. 4 は軌道 C と設計されたベクトル場を示す．なお，Fig. 1 と同様にベクトル場の長さは正規化した．次に，このアトラクタに対してある 4 点におけるベクトル場の向きを拘束条件として与えた．Fig. 5 の太線が拘束条件を表し，4 点 a, b, c, d におけるベクトル場の向きに制約を与えることでアトラクタを変形する．この結果，Fig. 4 で表されるベクトル場が Fig. 5 に

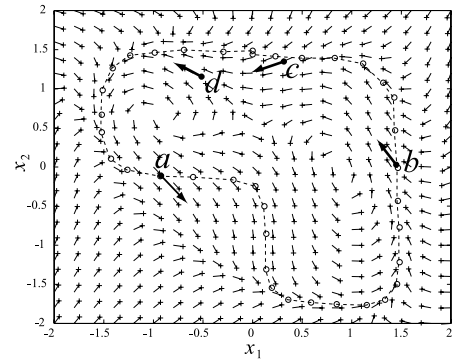


Fig. 5 Restriction and modified vector field

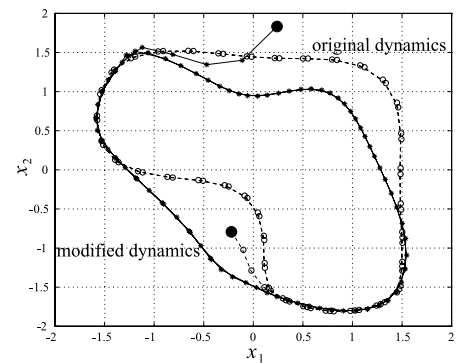


Fig. 6 Trajectory of the original and modified dynamics

表されるベクトル場に変形した．Fig. 6 は初めに設計した Φ_m に基づく力学系の軌跡（鎖線と「○」）と拘束条件を与えた Φ_λ に基づく力学系の軌跡（実線と「*」）を表す．「●」はそれぞれの初期値を表す．提案手法によりアトラクタが変化の様子が分かる．

3. 情報処理系の階層化設計

3.1 力学系の階層化

力学的情報処理系における力学系は時系列データをアトラクタとして記憶しており，自律的に動いた力学系はアトラクタに収束後，ある決まった閉軌道上を動く．力学系が複数のアトラクタ間を遷移し，異なる時系列データを生成するためには力学系の構造を変化させるメカニズムが必要である．ここではセンサ信号をトリガ信号とすることを考える．センサ信号の空間と運動を定める空間を合わせた，より多次元な空間での力学系設計によりセンサ信号と運動の結合を実現することも可能であるが，次第に次元の増える空間の中で力学系を設計することは，計算機の能力上現実的ではない．多次元での力学系設計はその構造を不明確にし，設計の指針を曖昧にする．そこで，外部からの信号を処理する層と運動を処理する層を個別に設計し，これらを結合する方法を考える．各層で個別の力学系を設計し，層間での結合を設定することで構造は明確化し，目的を考慮した設計を可能とするばかりでなく，アトラクタの追加や削除も容易にする．

3.2 階層化システムの構造

ここでは Fig. 7 に示されるようなセンサ空間とモータ空間を

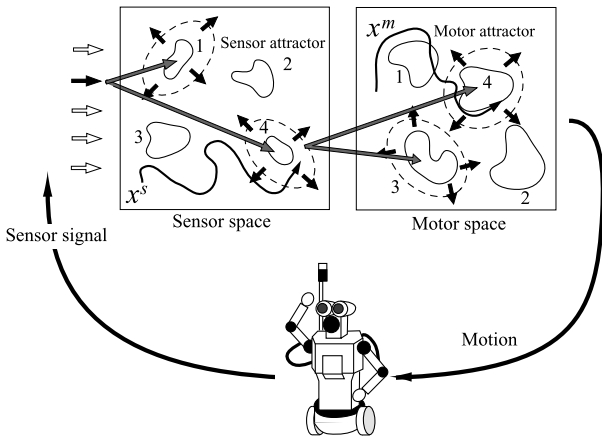


Fig. 7 Hierarchical design of sensor space and motor space

設定する．センサ空間は仮想的な空間であり，後に述べるカオスの挙動を導入するために二次元トラス空間とした．センサ空間内には複数のセンサアトラクタが存在し，それぞれのアトラクタの引き込み領域の大きさがセンサ信号によって決定される．この空間内における状態ベクトル $x^s[k]$ は力学系に従って運動する．センサ信号入力がない場合には $x^s[k]$ はカオス的な挙動をとり，探索を行うものとする．引き込み領域内に入るとそのアトラクタへと引き込まれる．センサ信号が変化し，引き込み領域が変化すると， $x^s[k]$ は他のアトラクタへと遷移する．

モータ空間にはロボットの全身運動の時系列データを記憶する複数のアトラクタが埋め込まれており，センサ空間における $x^s[k]$ のアトラクタへの引き込みがモータ空間のアトラクタの引き込み領域の大きさを決定する．モータ空間における状態ベクトル $x^m[k]$ は力学系に従って動き，アトラクタに引き込まれることでロボットは全身運動を行う．運動によりセンサ信号が変化し，これは環境を通じたセンサ空間へのフィードバックとなる．このように二つの空間によって力学系を階層化し，その結合に引き込み領域を変化させるという手法をとる．

3.3 力学系の設計

上記の階層化達成のために，複数アトラクタを持つ力学系と各アトラクタの引き込み領域を変化させる構造が必要となる．式(3)の一つの力学系が複数のアトラクタを持つように設計することも可能であるが，そのためには多項式の次数 ℓ を極めて大きくする必要があり，計算上望ましくない．そこで，ここでは文献[9]の方法に以下のような修正を加えて設計を行う．

ベクトル場の有効領域とアトラクタの引き込み領域

アトラクタを定義するベクトル場がアトラクタの近傍に限られるよう，式(3)の力学系を次式のように変形する．

$$x[k+1] = x[k] + w_1 f(x[k]) \quad (17)$$

ここで， w_1 はある定数 $a_1 > 0$ を用いて

$$w_1 = 1 - \frac{1}{1 + \exp\{a_1(\omega_1(x[k]) - 1)\}} \quad (18)$$

$$\omega_1(x[k]) = (x^T[k] - X_0^T)Q(x[k] - X_0) \quad (19)$$

と定義する． Q は X_0 を中心とし，閉曲線 C を内包する楕円

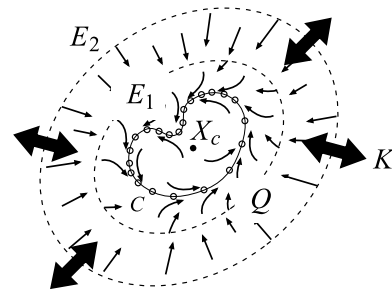


Fig. 8 Change of basin of attractor

体を定義する正定値行列である． $x[k]$ が楕円体の内部に存在するとき w_1 は 1 となり外部では 0 となる．さらに， C の内部の点(例えば C の中心)を X_c とし，次式によって引き込み領域を定義する．

$$x[k+1] = x[k] + w_2 (w_1 f(x[k]) + (1 - w_1)\delta(X_c - x[k])) \quad (20)$$

ここで， δ ($0 < \delta < 1$) は定数， w_2 はある定数 $a_2 > 0$ を用いて

$$w_2 = 1 - \frac{1}{1 + \exp\{a_2(\omega_2(x[k]) - 1)\}} \quad (21)$$

$$\omega_2(x[k]) = K(x^T[k] - X_0^T)Q(x[k] - X_0) \quad (22)$$

によって定義する．これにより， K が引き込み領域の大きさを決定するパラメータとなる．これらの内容を模式的に表すと Fig. 8 のようになる． E_1 は式(19)の右辺 = 1 によって定義される楕円体で E_2 は式(22)の右辺 = 1 によって定義される楕円体である． w_1 は $x[k]$ が E_1 の内部にあるときに 1，外部で 0 となり， w_2 は E_2 の内部で 1，外部で 0 となる． E_1 の内部かつ E_2 の内部でアトラクタを定めるベクトル場 $f(x[k])$ が有効で， E_1 の外部かつ E_2 の内部が引き込みのベクトル場 ($X_c - x[k]$) が有効となる． K が次第に大きくなり $E_1 \supset E_2$ となったとき引き込み領域は存在せず， $K \rightarrow \infty$ でアトラクタは消滅する．

複数のアトラクタを持つ力学系

ここで設計した力学系はベクトル場として定義されているため，複数のアトラクタを埋め込むためにはベクトル場の加算をとればよい．また，特にセンサ空間ではセンサ信号がない場合やセンサ空間における状態ベクトル $x^s[k]$ がどのアトラクタの引き込み領域にも存在しない場合にはカオス的な挙動をとり，空間の中を探索するように設定する．これより，力学系を次式によって定義する．

$$x[k+1] = x[k] + \sum_i w_{2i} \{w_{1i} f_i(x[k]) + (1 - w_{1i})\delta_i(X_{ci} - x[k])\} + \prod_i (1 - w_{2i})x^c[k] \quad (23)$$

ここで，右辺第三項 $x^c[k]$ はカオス的な挙動を発生させる項であり，他のカオス力学系によって生成される． $x[k]$ がどの引き

込み領域にも属さないときに

$$\prod_i (1 - w_{2i}) = 1 \quad (24)$$

$$w_{2i} = 0 \quad (i = 1, 2, \dots) \quad (25)$$

となり、カオス的な挙動となる。なお、モータ空間においてはこの項は付加していない。

3.4 ヒューマノイドロボットへの実装

3.4.1 ヒューマノイドロボットの概要

提案した力学系をヒューマノイドロボットへ実装し、運動の生成と遷移を実現する。ここで用いたヒューマノイドロボット Robovie を Fig. 9 に示す。このロボットの自由度は Fig. 9 右図にあるように首に 3, 肩に左右それぞれ 3, 肘に左右それぞれ 1 の合計 11 自由度である。また、頭, 肩, 胸, 上腕, 下腕, 手首に合計 16 個の接触センサ (オンオフのみ) と両眼としてのカラー NTSC カメラを持つ。カラーカメラからは画像中の赤, 緑, 青の占める割合を連続値で求めている。これより, センサ信号は 19 個となる。このロボットに対して 10 個の全身運動を設計した。Fig. 10, 11 にその例として, Motion 1 と Motion 2 を示す。

3.4.2 力学系の設計と階層化

このロボットは 11 自由度を有するが, ここでは文献 [9] の方法に基づき特異値分解に基づく主成分分析法によって 10 個の

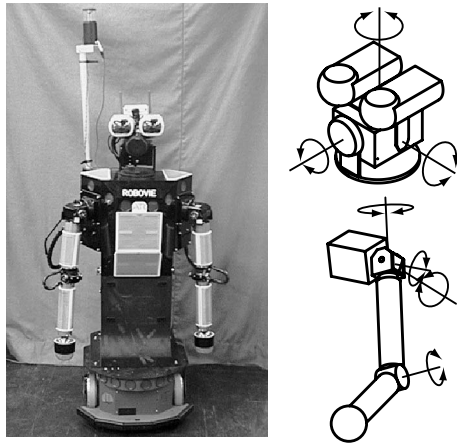


Fig. 9 Humanoid robot Robovie

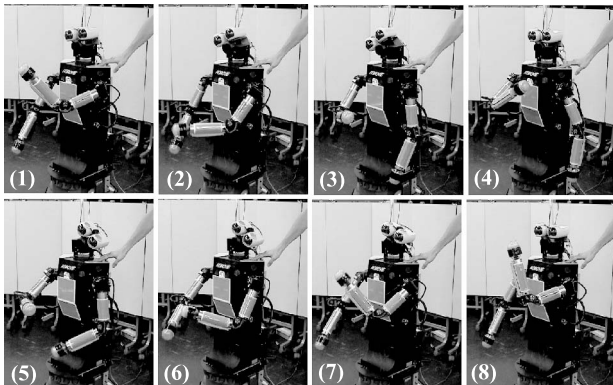


Fig. 10 Humanoid motion (Motion 1)

運動をすべて三次元に低次元化し, この低次元化された空間の中で次式の力学系を設計した。

$$\begin{aligned} \mathbf{x}^m[k+1] &= \mathbf{x}^m[k] \\ &+ \sum_{i=1}^{10} w_{2i}^m \{ w_{1i}^m \mathbf{f}_i^m(\mathbf{x}^m[k]) + (1 - w_{1i}^m) \delta_i^m(\mathbf{X}_{ci}^m - \mathbf{x}^m[k]) \} \end{aligned} \quad (26)$$

$$\mathbf{x}^m[k] \in \mathbf{R}^3 \quad (27)$$

ここで, m はモータ空間を意味し, $\mathbf{x}^m[k]$ はモータ空間内における状態ベクトルである。また, $\mathbf{x}^m[k]$ から十一次元の関節角空間へ復元する関数を $F_i(\in \mathbf{R}^{11 \times 3})$ とするとき, ロボットの運動 $\mathbf{y}[k](\in \mathbf{R}^{11})$ は

$$\mathbf{y}[k] = \sum_{i=1}^{10} w_{2i}^m w_{1i}^m F_i \mathbf{x}^m[k] \quad (28)$$

によって生成した。ここで, $w_{2i}^m w_{1i}^m$ を i 番目のアトラクタに引き込まれていることを判断するパラメータとして用いている。

モータ空間内における i 番目のアトラクタの引き込み領域を定めるパラメータ K_i^m はセンサ空間における力学系の引き込みによって決定される。ここでは, センサ空間内に 10 個のアトラクタを埋め込んだ。タッチセンサからの入力 (0 か 1 の値) $u_i^t (i = 1, 2, \dots, 16)$ から

$$\mathbf{u}^t = [u_1^t \quad u_2^t \quad \dots \quad u_{16}^t]^T \quad (29)$$

とし, カラー画像の RGB の割合を

$$\mathbf{u}^v = [u_R^v \quad u_G^v \quad u_B^v]^T \quad (30)$$

とする。このとき, センサ空間におけるアトラクタの引き込み領域の大きさを決定するパラメータ

$$\mathbf{K}^s = [K_1^s \quad K_2^s \quad \dots \quad K_{10}^s]^T \quad (31)$$

を, 重み行列 ${}^s W_s$ を用いて

$$\mathbf{K}^s = {}^s W_s \begin{bmatrix} \mathbf{u}^t \\ \mathbf{u}^v \end{bmatrix} \quad (32)$$

によって定めた。さらに, モータ空間での引き込み領域の大き

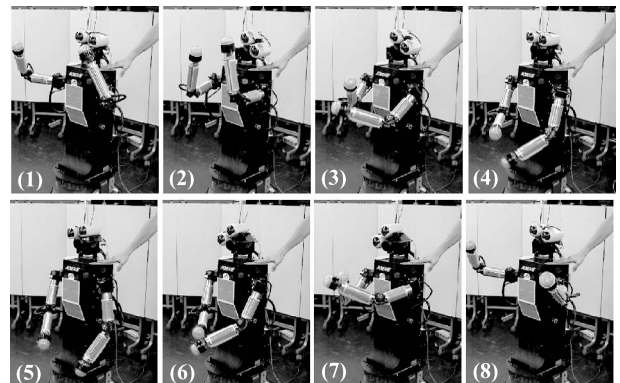


Fig. 11 Humanoid motion (Motion 2)

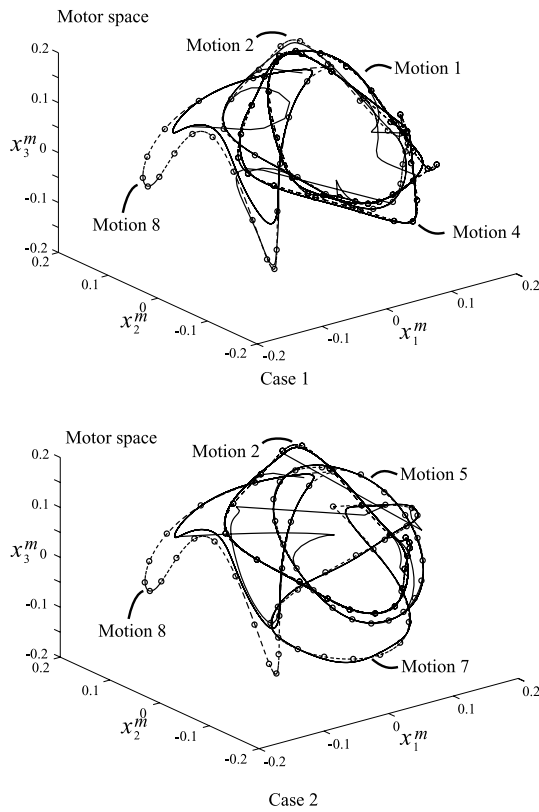


Fig. 12 Motion of dynamics in motor space

さを決定するパラメータ

$$K^m = \begin{bmatrix} K_1^m & K_2^m & \cdots & K_{10}^m \end{bmatrix} \quad (33)$$

を、重み行列 mW_s を用いて

$$K^m = {}^mW_s w^s \quad (34)$$

$$w^s = \begin{bmatrix} w_{21}^s w_{11}^s & w_{22}^s w_{12}^s & \cdots & w_{2-10}^s w_{1-10}^s \end{bmatrix}^T \quad (35)$$

とした．式(28)と同様に， $w_{2i}^s w_{1i}^s$ をアトラクタに引き込まれたことの判断指標として用いている．なお， sW_s はセンサ信号とセンサ空間におけるアトラクタ番号を結び付けを行うもの， mW_s はセンサ空間におけるアトラクタ番号とモータ空間におけるアトラクタ番号の結び付けを行うもので，例えば，右肩の接触センサが on になったときは，2 番目と 3 番目のアトラクタの引き込み領域を大きくする，といった具合に設計者が与えた．

3.4.3 ロボットの運動生成

設計した力学系を用いてロボットの運動生成と遷移を行った．ここでは，右肩(時刻 $t_1 < t < t_2$) → 頭部右側(時刻 $t_3 < t < t_4$) → 右腕(時刻 $t_5 < t$)の順番に接触センサの入力信号を加え，その時間タイミングを変化させて 2 回の行動生成を行った．Fig. 12 はそのときの生成された運動を表す．センサ信号のタイミングが異なることにより，異なる運動が生成されている．Fig. 13 に引き込み領域の大きくなったアトラクタ(ここでは発火という言葉で表現するものとする)と実際に生成された運動の時間変化を示す．グラフの濃度が発火の大きさを表しており，同時刻

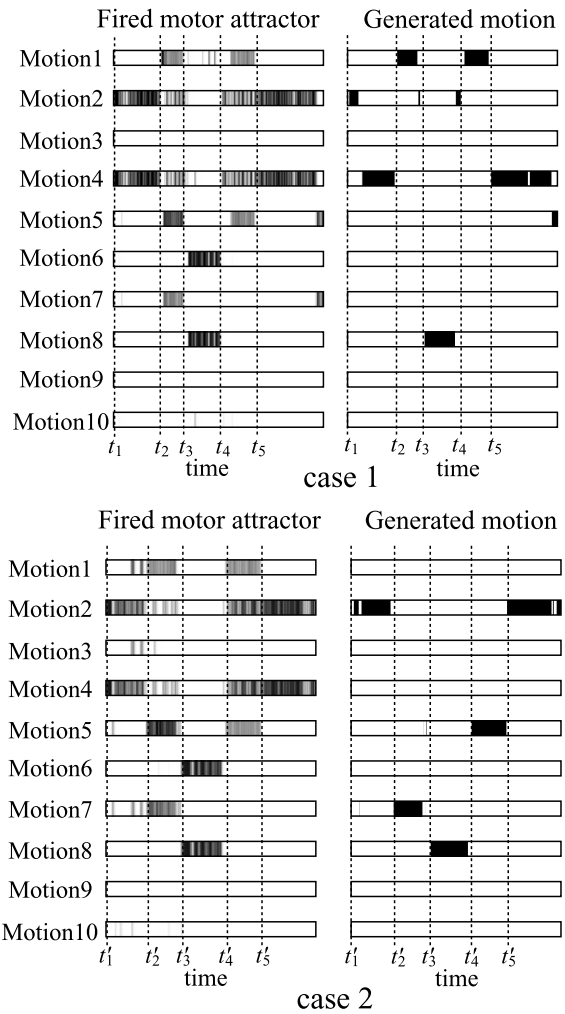


Fig. 13 Fired attractors and generated motions

に複数のアトラクタが発火している．力学系はその中のいずれかのアトラクタに引き込まれ運動が生成される．接触センサからの入力によりセンサ空間におけるアトラクタの発火パターンは同じであるが，時間のタイミングによって異なる運動が生成される．これはセンサ空間，モータ空間における力学系の状態(内部状態)に依存した行動が生成されることを示している．

4. モジュール化ネットワークの構成

4.1 文脈依存の行動の設計

前章の方法によりセンサ空間とモータ空間が結合され，センサ信号に基づく力学系の構造変化とロボットの運動遷移を実現した．ロボットの運動遷移は式(34)の K^m によって決定され， mW_s が二つの空間の結合を定義している．一方，我々の運動は現在のセンサ信号のみによって決定するのではなく，現在自分が実行している運動や過去の運動，センサ信号の時間的な流れに依存する．これは，文脈依存性として多視点からの知識に必要な要素である．本章では mW_s を力学系によって動的に変化させ，文脈依存の運動生成機構を設計する．ここではニューラルネットワークの連想記憶モデル[14]を用いる．センサ空間，モータ空間に加えて内部状態空間を設定し，モジュール化を行

いこれらの結合によってセンサ信号，運動の時系列データに基づいた行動を決定する．

4.2 モジュール化ネットワークの設計

本章で設計するモジュール化ネットワークの構造を Fig. 14 に示す．Fig. 7 と比較して内部状態空間を設定した．また，各空間の結合は各空間におけるアトラクタへの引き込みを表現するパラメータ $w[\tau]$ と各空間における引き込み領域の大きさ $K[\tau]$ を用いて，上層にある

$$K[\tau+1] = g(w[\tau]) \quad (36)$$

によって定義されている．これはネットワークの文脈依存性を実現することを目的としている．次にネットワークの構成法について述べる．まず，センサ空間において式 (35) と同様に j 番目の内部状態空間において w_j^i ($j = 1, 2, \dots, n$)，モータ空間に対して w^m を定義する．これらは 0 から 1 の連続値で構成されるベクトルである．さらに，式 (31), (33) と同様に j 番目の内部状態空間におけるアトラクタの引き込み領域の大きさを定めるパラメータ K_j^i を定義する．このとき，時刻 τ の $w^s, w_1^i, \dots, w_n^i, w^m$ が時刻 $\tau+1$ の内部状態空間，モータ空間の K_j^i, K^m を

$$\begin{bmatrix} K^i[\tau+1] \\ K^m[\tau+1] \end{bmatrix} = g \left(\begin{bmatrix} w^s[\tau] \\ w^i[\tau] \\ w^m[\tau] \end{bmatrix} \right) \quad (37)$$

$$K^i[\tau] = \begin{bmatrix} K_1^i T[\tau] & \dots & K_n^i T[\tau] \end{bmatrix}^T \quad (38)$$

$$w^i[\tau] = \begin{bmatrix} w_1^i T[\tau] & \dots & w_n^i T[\tau] \end{bmatrix}^T \quad (39)$$

によって決定するものとする．すなわち，時刻 $\tau+1$ の引き込み領域の大きさはセンサ空間，内部状態空間，モータ空間の力学系が時刻 τ にどのアトラクタに引き込まれているかによって決まるといえるものである． K^s は式 (31) と同様にセンサ信号のみによって決定されるとした．これにより，力学系の動きが次の時刻の引き込み領域を決定し，センサ信号に基づいて自律的に運動遷移を行う機構が設計できる．

いま，運動計画において 1. 動き始める，2. ランドマークを見つければ右へ曲がる，3. 壁を発見して左へ曲がる，4. ... という

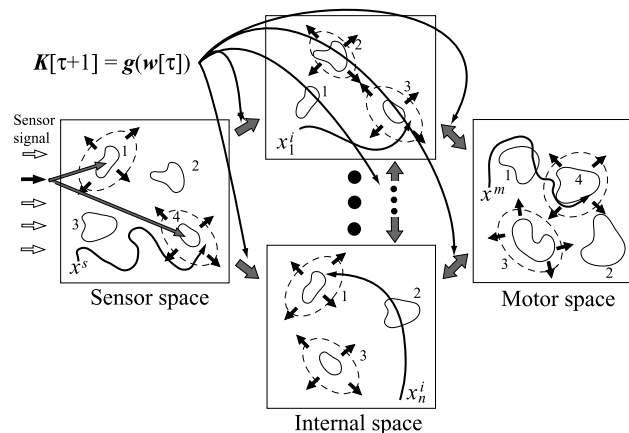


Fig. 14 Modularized network architecture

例のように，センサ信号と行動のシーケンスを

$$\begin{bmatrix} w^s[\tau] \\ w^i[\tau] \\ w^m[\tau] \end{bmatrix}, \begin{bmatrix} w^s[\tau+1] \\ w^i[\tau+1] \\ w^m[\tau+1] \end{bmatrix}, \begin{bmatrix} w^s[\tau+2] \\ w^i[\tau+2] \\ w^m[\tau+2] \end{bmatrix}, \dots \quad (40)$$

と定める．ただし，理想状態を考え， w^s, w^i, w^m はすべて 0 か 1 の 2 値によって構成されるとする．また，内部状態空間 $w^i[\cdot]$ は各時刻における $\begin{bmatrix} w^s T[\cdot] & w^i T[\cdot] & w^m T[\cdot] \end{bmatrix}^T$ が等しくならないように定めるものとする．これを用いて，各時刻 τ において

$$\begin{bmatrix} w^i[\tau+1] \\ w^m[\tau+1] \end{bmatrix} = g \left(\begin{bmatrix} w^s[\tau] \\ w^i[\tau] \\ w^m[\tau] \end{bmatrix} \right) \quad (41)$$

を満たす関数 g をニューラルネットワークによる連想記憶モデルを用いて設計し，これを式 (37) の g とすることで，アトラクタの引き込み領域 K^i, K^m を決定する．これにより，同じセンサ信号に対しても内部状態の影響により過去の経歴による文脈依存の運動が生成される．

4.3 力学系の動きと遷移

上記の構造を持つ力学系のネットワークを構成し，各空間における力学系の動きを観察した．ここでは，Fig. 9 で表されるロボットを想定し，センサ空間として接触センサ空間，ビジュアルセンサ空間を設定した．それぞれの空間には 10 個のアトラクタ，3 個のアトラクタを設定している．また，それぞれ 3 個のアトラクタを有する 5 個の内部状態空間と，5 個の運動を記憶しているモータ空間を設定した．適当な $\begin{bmatrix} w^s T & w^i T & w^m \end{bmatrix}^T$ のシーケンスを設定し， $g \left(\begin{bmatrix} w^s T & w^i T & w^m \end{bmatrix}^T \right)$ をニューラルネットワークの連想記憶モデルを用いて設計した．ここで，運動は運動 1 \rightarrow 運動 2 \rightarrow 運動 3 \rightarrow 運動 4 \rightarrow 運動 5 \rightarrow 運動 1 $\rightarrow \dots$ というシーケンスで設定した．さらに，式 (41) の力学系の刻み幅 τ に対して，各モジュール内の力学系の刻み幅 k は $\tau = 10k$ とした．このときの力学系の動きを Fig. 15 に示す．上から順にタイムステップ k が 30, 140, 270 のときの様子を示す．各空間における点線は各アトラクタの引き込み領域を，鎖線はアトラクタを，実線は学習に用いた引き込まれるべきアトラクタ (シーケンスに従ったもの)，一点鎖線は力学系の軌跡，「●」は各時刻における力学系の状態変数を意味する．実線で示されたアトラクタの引き込み領域が大きくなり，力学系が引き込まれていれば，これは学習したシーケンス通りの手順に沿って力学系が動いていることを意味する．例えば，タイムステップが 30 のとき，モータ空間において引き込まれるべきアトラクタは 1 番であるが，3 番のアトラクタの引き込み領域が大きくなり，力学系は学習した通りの運動を表していないことを意味する．また，Fig. 16 にモータ空間における学習した運動の番号と生成した運動の番号を示す．実線が生成された運動，鎖線が学習した運動のシーケンスである．初期値の影響で，初めは学習した運動が生成されていないが時間の経過とともに学習した運動へ収束する様子が理解できる．ここでの結果にお

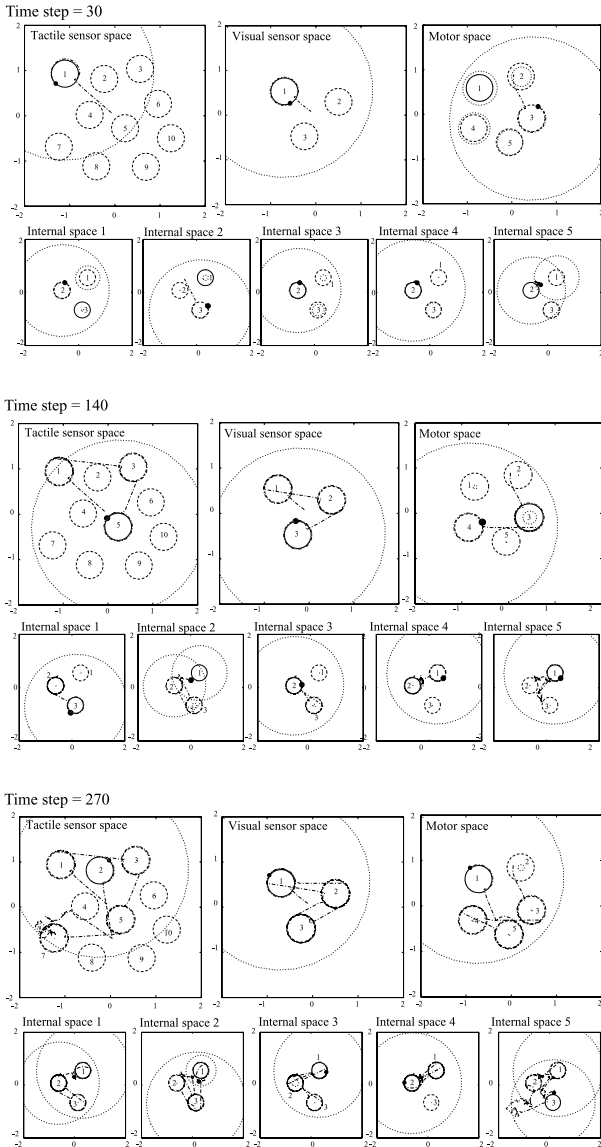


Fig. 15 Trajectory of the dynamics

いて運動のシーケンスの学習にニューラルネットワークの連想記憶モデルを用いたため学習したシーケンス通りに運動が生成されることは保証されない。しかし、ニューラルネットワークの連想記憶モデルには引き込み効果があることが知られており、この効果によりシーケンスが実現されている。これらの結果から、力学系をモジュール化しそのネットワークを構成することで大規模な情報処理系が設計されたことが分かる。

5. おわりに

本論文では、可塑性を持つ力学系の設計と大規模力学系の設計を目的とし、これまでに提案した力学的情報処理系に対して、
 (1) 逐次最小二乗法を用いることにより、アトラクタを持つ力学系を逐次的に設計する学習効果を持たせた。
 (2) 逐次最小二乗法に忘却係数を設定し、可塑性を持つ力学系の設計法を示した。

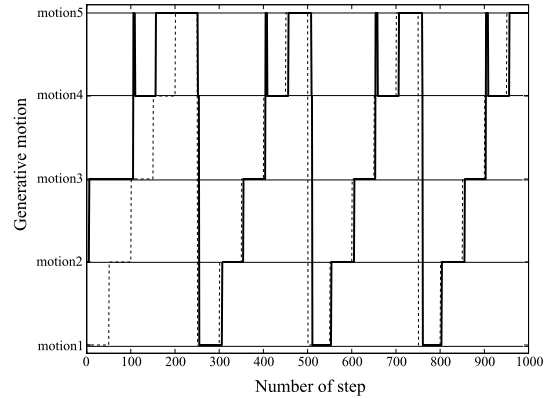


Fig. 16 Generative motion

- (3) 逐次最小二乗法を応用し、拘束条件を加えてアトラクタの形を変形させる方法を提案した。
- (4) 力学系を階層化し、そのネットワーク化の手法について述べた。
- (5) 上半身型のヒューノイドロボットに階層化された力学系を実装し、センサ空間とモータ空間を設計することで、内部状態に依存した運動の生成を行った。
- (6) 階層化された力学系の結合を統括する力学系の設計法を示し、文脈依存性を持つ運動生成法を提案した。

本論文で示した方法では時系列データの信号処理法に焦点が当てられており、ロボットの安定性に関する議論は行われていない。提案した手法をもとにして、

- (1) ロボットの身体の力学を導入することで与えられたおおまかな運動から独自の運動を学習・獲得する機構
- (2) 環境の変化に対しても最終目的を達成するための行動選択といった知能ロボットに向けての発展が期待される。特に、(1)では情報処理に力学系を用いていることから、ロボットの身体の力学を導入しやすい、(2)では力学系の引き込みを利用しているといった点から、力学系を用いた情報処理系の大きな利点が期待される。

謝辞 本研究は科学技術振興事業団「自律行動単位の力学的結合による脳型情報処理機械の開発」(プロジェクトリーダー: 中村仁彦) および文部科学省科学研究補助金特定領域研究(A03)「力学的情報処理の階層化に基づく知能ロボットのリアルタイムカーネルの構築」(研究代表者: 岡田昌史)の支援を受けた。

参考文献

- [1] W.J. Freeman: "Simulation of Chaotic EEG Patterns, Non-linear Dynamic model of the Olfactory Systems," Biological Cybernetics, vol.56, pp.139-150, 1987.
- [2] I. Tsuda: "Chaotic itinerancy as a dynamical basis of Hermeneutics in brain and mind," World Futures, 31, pp.105-122, 1991.
- [3] A. Sekiguchi and Y. Nakamura: "The Chaotic Mobile Robot," Proc. of Systemics, Cybernetics and Informatics 2000, vol.9, pp.463-468, 2000.
- [4] A. Sekiguchi and Y. Nakamura: "Behavior Control of Robot Using Orbits of Nonlinear Dynamics," Proc. of IEEE International Conference on Robotics and Automation, pp.1647-1652, 2001.

- [5] 琴坂, S. Schaal: “神経振動子を用いたロボットのリズム的な運動生成”, 日本ロボット学会誌, vol.19, no.1, pp.116–123, 2001.
- [6] 木村, 秋山, 桜間: “神経振動子を用いた四足ロボットの不整地歩行と整地走行”, 日本ロボット学会誌, vol.16, no.8, pp.1138–1145, 1998.
- [7] 土屋, 辻田: “Central Pattern Generator モデルに基づく 4 脚歩行ロボットの歩行制御”, 日本ロボット学会誌, vol.20, no.3, pp.243–246, 2002.
- [8] H. Kawashima and T. Matsuyama: “Integrated Event Recognition from Multiple Sources,” Proc. of International Conference on Pattern Recognition, vol.2, pp.785–789, 2002.
- [9] 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.
- [10] M. Adachi, T. Ushio and S. Yamamoto: “Application of Lyapunov Function Based Synthesis of Nonsmooth Limit Cycles to Motion Generation for Humanoid Robots,” Proc. of 2nd International Symposium of Adaptive Motion of Animals and Machines, SaP-II-4, 2003.
- [11] 甘利俊一, 外山敬介編集: 脳科学大辞典. 朝倉書店, 2000.
- [12] 武田暁: 脳と力学系. 講談社, 1997.
- [13] 足立修一: ユーザのためのシステム同定理論. 計測自動制御学会, 1993.
- [14] 中野馨: ニューロコンピュータの基礎. コロナ社, 1991.

付録 A. 力学系の設計法 [9]

閉曲線 C アトラクタとする力学系の設計法を示す.

Step 1 N 次元空間上に閉曲線 C を描く.

Step 2 N 次元空間上に引き込み領域 D を定義し, その領域内の点 η_i におけるベクトル場 $f(\eta_i)$ をある γ_i を用いて以下の式で定義する.

$$f(\eta_i) = (\xi^{\eta_i}[k+1] - \xi^{\eta_i}[k]) + \gamma_i[k] \quad (\text{A.1})$$

$$\xi^{\eta_i}[k] = \arg \min_{\xi[k]} \|\eta_i - \xi[k]\| \quad (\text{A.2})$$

$\xi^{\eta_i}[k]$ は $\xi[k]$ のなかで η_i に最も近い点を意味する. ここで, $\delta[k]$ を

$$\eta_i = \xi^{\eta_i}[k] + \delta[k] \quad (\text{A.3})$$

によって定義する. このとき, ベクトル場 $f(x[k])$ に従って式 (3) の力学系が動いたときに C をアトラクタとするための十分条件は

$$\|\delta[k] + \gamma_i[k]\| < \|\delta[k]\| \quad (\text{A.4})$$

で与えられるため, γ を式 (A.4) を満たすように選び, これを用いて式 (A.1) の $f(\eta_i)$ を定義すれば $\delta[k]$ が $k \rightarrow \infty$ のときに, 0 に収束する.

Step 3 引き込み領域 D 内に存在する多くの点 $\eta_1, \eta_2, \dots, \eta_m$ を定義し, ベクトル場 $f(\eta_1), f(\eta_2), \dots, f(\eta_m)$ を定める.

Step 4 定義したベクトル場 $f(\eta_i)$ を以下のような最大次数 ℓ の多項式の形で近似する.

$$f(\eta) = \sum_{P=0}^{\ell} \sum_{p_1, \dots, p_n}^{\ell} a_{(p_1 p_2 \dots p_n)} \prod_{i=1}^n \eta_i^{p_i} \quad (\text{A.5})$$

$\sum p_i = P$
 p_i : positive integer

$$\eta = \begin{bmatrix} \eta_1 & \eta_2 & \dots & \eta_N \end{bmatrix}^T \quad (\text{A.6})$$

$a_{(p_1 p_2 \dots p_n)}$ は定数である. 例えば, 二次元空間上 ($N = 2$) で $\eta = \begin{bmatrix} \eta_1 & \eta_2 \end{bmatrix}$ の場合) のベクトル場を三次の多項式 ($\ell = 3$) で近似する場合,

$$\begin{aligned} f(\eta) = & a_{(30)}\eta_1^3 + a_{(21)}\eta_1^2\eta_2 + a_{(12)}\eta_1\eta_2^2 + a_{(03)}\eta_2^3 \\ & + a_{(20)}\eta_1^2 + a_{(11)}\eta_1\eta_2 + a_{(02)}\eta_2^2 \\ & + a_{(10)}\eta_1 + a_{(01)}\eta_2 + a_{(00)} \end{aligned} \quad (\text{A.7})$$

となる. このとき, $f(\eta)$ を

$$f(\eta) = \Phi(a_{(p_1 p_2 \dots p_n)})\theta(\eta) \quad (\text{A.8})$$

$$\theta(\eta) = \begin{bmatrix} \eta_1^\ell & \dots & \eta_N^\ell & \eta_1^{\ell-1}\eta_2 & \dots & 1 \end{bmatrix}^T \quad (\text{A.9})$$

とすると, Φ は最小二乗法によって以下のように計算される.

$$\Phi(a_{(p_1 p_2 \dots p_n)}) = F\Theta^\# \quad (\text{A.10})$$

$$F = \begin{bmatrix} f(\eta_1) & f(\eta_2) & \dots & f(\eta_m) \end{bmatrix} \quad (\text{A.11})$$

$$\Theta = \begin{bmatrix} \theta(\eta_1) & \theta(\eta_2) & \dots & \theta(\eta_m) \end{bmatrix} \quad (\text{A.12})$$

これにより, Φ が定義したベクトル場を良く近似していれば, D は引き込み領域として定義され, D の中から出発した $x[0]$ は $k \rightarrow \infty$ で閉曲線 C に引き込まれる.

付録 B. 重み付き逐次最小二乗法

次の重み付き最小二乗問題を考える.

$$\Phi = \arg \min_{\Phi} \left\| \begin{bmatrix} \alpha F & f \end{bmatrix} - \Phi \begin{bmatrix} \alpha \Theta & \theta \end{bmatrix} \right\| \quad (\text{B.13})$$

逆行列補題から $(\alpha^2\Theta\Theta^T + \theta\theta^T)^{-1}$ は

$$\begin{aligned} (\alpha^2\Theta\Theta^T + \theta\theta^T)^{-1} &= (\alpha^2\Theta\Theta^T)^{-1} \\ &\quad - \frac{(\alpha^2\Theta\Theta^T)^{-1}\theta\theta^T(\alpha^2\Theta\Theta^T)^{-1}}{1 + \theta^T(\alpha^2\Theta\Theta^T)^{-1}\theta} \end{aligned} \quad (\text{B.14})$$

で表される. ここで,

$$P_{m+1} = (\alpha^2\Theta\Theta^T + \theta\theta^T)^{-1} \quad (\text{B.15})$$

$$P_m = (\alpha^2\Theta\Theta^T)^{-1} \quad (\text{B.16})$$

とおくことで, 式 (4) が得られる. さらに,

$$\Phi_{m+1} = \begin{bmatrix} \alpha F & f \end{bmatrix} \begin{bmatrix} \alpha \Theta & \theta \end{bmatrix}^\# \quad (\text{B.17})$$

$$\Phi_m = F\Theta^\# \quad (\text{B.18})$$

であるから,

$$\begin{aligned} \Phi_{m+1} &= (\alpha^2 F\Theta^T + f\theta^T)(\alpha^2\Theta\Theta^T + \theta\theta^T)^{-1} \\ &= \alpha^2 F\Theta^T(\alpha^2\Theta\Theta^T)^{-1}(\alpha^2\Theta\Theta^T)(\alpha^2\Theta\Theta^T + \theta\theta^T)^{-1} \end{aligned} \quad (\text{B.19})$$

$$+ \mathbf{f}\boldsymbol{\theta}^T P_{m+1} \quad (\text{B.20})$$

$$= \Phi_m (\alpha^2 \boldsymbol{\theta} \boldsymbol{\theta}^T + \boldsymbol{\theta} \boldsymbol{\theta}^T - \boldsymbol{\theta} \boldsymbol{\theta}^T) (\alpha^2 \boldsymbol{\theta} \boldsymbol{\theta}^T + \boldsymbol{\theta} \boldsymbol{\theta}^T)^{-1} + \mathbf{f}\boldsymbol{\theta}^T P_{m+1} \quad (\text{B.21})$$

$$= \Phi_m - \Phi_m \boldsymbol{\theta} \boldsymbol{\theta}^T P_{m+1} + \mathbf{f}\boldsymbol{\theta}^T P_{m+1} \quad (\text{B.22})$$

$$= \Phi_m + (\mathbf{f} - \Phi_m \boldsymbol{\theta}) \boldsymbol{\theta}^T P_{m+1} \quad (\text{B.23})$$

となり、式(5)と同様の結果が得られる。

付録 C. 拘束条件付き最小二乗法

式(12)の拘束条件を満たし、式(15)を最小化する Φ は、ラグランジュの未定乗数 Λ を用いることで、以下の評価関数 J を最小化する Φ として求められる。

$$J = \frac{1}{2} \text{trace}((F - \Phi_\lambda \Theta)(F - \Phi_\lambda \Theta)^T) + \text{trace}(\Lambda^T (F_\lambda - \Phi_\lambda \Theta_\lambda)) \quad (\text{C.24})$$

この解は、

$$\frac{\partial J}{\partial \Phi_\lambda} = -(F - \Phi_\lambda \Theta) \Theta^T - \Lambda \Theta_\lambda^T = 0 \quad (\text{C.25})$$

$$\frac{\partial J}{\partial \Lambda} = F_\lambda - \Phi_\lambda \Theta_\lambda = 0 \quad (\text{C.26})$$

を満たすものとして与えられる。式(C.25)より、

$$\Phi_\lambda = (F \Theta^T + \Lambda \Theta_\lambda^T) (\Theta \Theta^T)^{-1} \quad (\text{C.27})$$

$$= \Phi_m + \Lambda \Theta_\lambda^T (\Theta \Theta^T)^{-1} \quad (\text{C.28})$$

となり、これを式(C.26)に代入して

$$F_\lambda - \Phi_m \Theta_\lambda = \Lambda \Theta_\lambda^T (\Theta \Theta^T)^{-1} \Theta_\lambda \quad (\text{C.29})$$

となる。これを Λ について解いて、

$$\Lambda = (F_\lambda - \Phi_m \Theta_\lambda) (\Theta_\lambda^T (\Theta \Theta^T)^{-1} \Theta_\lambda)^{-1} \quad (\text{C.30})$$

となり、これを式(C.28)に代入することで

$$\Phi_\lambda = \Phi_m + (F_\lambda - \Phi_m \Theta_\lambda) (\Theta_\lambda^T (\Theta \Theta^T)^{-1} \Theta_\lambda)^{-1} \Theta_\lambda^T (\Theta \Theta^T)^{-1} \quad (\text{C.31})$$

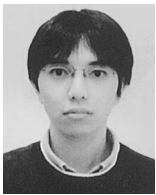
$$= \Phi_m + (F_\lambda - \Phi_m \Theta_\lambda) (\Theta_\lambda^T P_m \Theta_\lambda)^{-1} \Theta_\lambda^T P_m \quad (\text{C.32})$$

を得る。これは式(16)と一致する。



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。2001年4月同大学大学院情報理工学系研究科講師、2004年4月東京工業大学大学院理工学系研究科助教授となり現在に至る。ヒューマノイドロボットの機構開発、脳型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)



門根秀樹 (Hideki Kadone)

1980年3月20日生。2002年3月東京大学工学部機械情報工学科卒業。2004年3月同大学大学院情報理工学系研究科知能機械情報学専攻修士課程修了。2004年4月同博士課程進学。現在に至る。力学系を用いた脳型情報処理システムの研究に従事。(日本ロボット学会学生会員)



中村仁彦 (Yoshihiko Nakamura)

1954年9月22日生。1977年京都大学工学部精密工学科卒業。1982年同大学大学院博士課程単位取得退学。1982年同大学助手。1987年カリフォルニア大学サンタバーバラ校助教授、准教授。1991年東京大学工学部機械情報工学科助教授。現在、同大学大学院情報理工学系研究科教授。工学博士。ロボットの運動学、動力学、制御、および知能の問題、特に非ホロノミックロボット、脳型情報処理、ヒューマノイドやCGの運動・力学計算、外科手術用ロボットなどの研究に従事。システム制御情報学会、計測自動制御学会、日本機械学会、日本コンピュータ外科学会、日本IFTToMM会議、IEEE、ASMEなどの会員。(日本ロボット学会正会員)



中村大介 (Daisuke Nakamura)

1979年8月3日生。2002年3月東京大学工学部機械工学科卒業。2004年3月同大学大学院情報理工学系研究科知能機械情報学専攻修士課程修了。2004年4月マッキンゼー・アンド・カンパニー入社。現在に至る。



梶山博史 (Hiroshi Kajiyama)

1980年7月3日生。2003年3月東京大学工学部機械情報工学科卒業。2003年4月同大学大学院工学系研究科機械工学専攻修士課程に進学。現在に至る。