

拡散学習を用いた冗長ロボットの 視覚・運動協調の自己組織化

Self-organization of Visuo-motor Coordination for a Redundant Manipulator Using Diffusion-based Learning Approach

○ 伊藤 義治, 羅 志偉, 渡部 慶二, 遠藤 茂

○ Yoshiharu Ito, Zhiwei Luo, Keiji Watanabe, Shigeru Endo

*山形大学大学院 理工学研究科

*Yamagata University

キーワード: 冗長ロボット(Redundant Manipulator)、視覚・運動協調(Visuo-motor Coordination)、自己組織化(Self-organization)、拡散学習(Diffusion-based Learning Approach)

連絡先: 〒992-0038 山形県米沢市城南4-3-16 山形大学 工学部 応用生命システム工学科
渡部研究室 伊藤義治, Tel.: (0238)26-3326, Fax.: (0238)24-2752, E-mail: itou@ewata.yz.yamagata-u.ac.jp

1. はじめに

本研究では、拡散学習理論を用いた冗長マニピュレータの視覚・運動協調の自己組織化について考察する。

はじめに、マニピュレータの逆運動学問題に対し、拡散学習理論を適用して、作業空間において自己組織化マップを形成しロボットアームの手先制御を行うことを示す。

次に、ロボット最適制御の空間汎化問題を扱う。一般に、最適制御問題は2点境界値問題を解く必要があり、解析的にも数値的にも解を求めることが困難である。本研究では、まず、いくつかの初期状態から終端状態までの最適解が既知であると仮定し、それらの最適入力のパラメトリゼーションを行うことによって、基底関数と重み行列の積に分解する。最終的には、拡散学習理論を適用し、有界空間における最適制御の重み行列を求める。

拡散学習方式を適用することにより、最適制御の計算量を大幅に減らすことができ、局所情報のみによる情報の並列分散処理が可能となる。

2. 実験システムの構成

Fig.1に実験に使用するロボットシステムの構成図を示す。

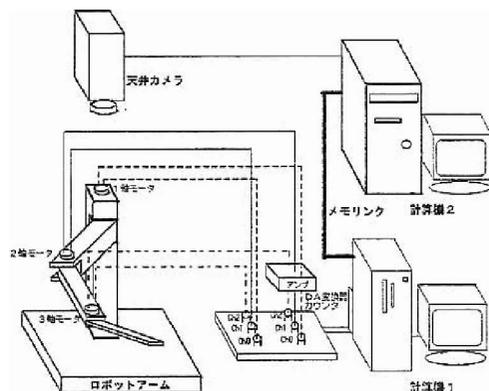


Fig.1 実験環境

この実験装置では、天井に取り付けたカメラで物体の位置情報を計算機2に取り込み、その情報を元にロボットの手先位置、姿勢と目標位置、障害物情報などを検出する。計算機2で処理した目標位置はメモリリンクを介してロボット制御用計算機1に送信する。計算機1では、計算機2からの情報を元に、自己組織化マップを形成し、目標手先位置座標を目標関節角度に変換することによって、ロボットの関節フィードバック制御を行う。

3. 拡散学習方式

3.1. 拡散学習方式

一般に、冗長な非線形関数

$$\mathbf{x} = \mathbf{g}(\mathbf{y}); \mathbf{x} \in R^n, \mathbf{y} \in R^m, m \geq n \quad (1)$$

に対する逆解

$$\mathbf{y} = \mathbf{g}^{-1}(\mathbf{x}) \quad (2)$$

を求めるには拡散学習理論を適用することが有効である。このアルゴリズムでは、 \mathbf{x} 空間を格子状に分割し、いくつかの典型的な \mathbf{x} の点における逆解をあらかじめ教師信号(境界状態)として与えておく。その他の格子点上の逆解については次の関係式を用いて更新をしてゆく。

$$\mathbf{y}_{i,j}^{t+1} = \frac{1}{4} \alpha(t) (\mathbf{y}'_{i,j-1} + \mathbf{y}'_{i-1,j} + \mathbf{y}'_{i,j+1} + \mathbf{y}'_{i+1,j}) + \beta(t) \mathbf{A}_{ij}^t (\mathbf{x}_{i,j}^d - \mathbf{g}(\mathbf{y}_{i,j}^t)) \quad (3)$$

ここで (i,j) は格子点の位置を表し、 t は学習ステップである。最初に $\alpha = 1$ 、 $\beta = 0$ とし拡散によって更新を行う。その後 $\alpha = 0$ 、 $\beta = 1$ とし誤差が減少するように修正してゆく。

行列 \mathbf{A} は式(2)の逆ヤコビ行列であり、次の式を用いて更新してゆく。

$$\mathbf{A}_{i,j}^{t+1} = \mathbf{A}_{i,j}^t + \frac{1}{\|\Delta \mathbf{x}_{i,j}^t\|^2} (\Delta \mathbf{y}_{i,j}^t - \mathbf{A}_{i,j}^t \Delta \mathbf{x}_{i,j}^t) \Delta \mathbf{x}_{i,j}^{tT} \quad (4)$$

ここで $\Delta \mathbf{x}_{i,j}^t = \mathbf{x}_{i,j}^t - \mathbf{x}_{i,j}^{t-1}$ と $\Delta \mathbf{y}_{i,j}^t = \mathbf{y}_{i,j}^t - \mathbf{y}_{i,j}^{t-1}$ は2つの学習ステップにおける誤差である。

最終的に学習理論は次のようにまとめられる

- Step1: 教師学習を用いて $x = f(\theta)$ を学習する。
- Step2: 作業空間を限定し、それを $N \times N$ サイズの格子に分割する。
- Step3: 境界上で試運動を行い対応する関節角度を記憶する。すべての $i, j = 1, 2, \dots, N$ に対し初期角度条件 $\mathbf{y}_{i,j}^0$ と初期逆ヤコビ行列 $\mathbf{A}_{i,j}^0$ を設定する。
- Step4: 時間関数 $\alpha(t)$ と $\beta(t)$ を設定する。
 $\beta(t) = 1 - \alpha(t)$ とする。
- Step5: $\Delta \tilde{\mathbf{y}}_{i,j}^t = \mathbf{y}_{i,j}^t - \mathbf{y}_{i,j}^{t-1}$ 、 $\Delta \tilde{\mathbf{x}}_{i,j}^t = \mathbf{x}_{i,j}^t - \mathbf{x}_{i,j}^{t-1}$ と

$$\frac{\partial E_{i,j}}{\partial \mathbf{A}_{i,j}^t} = -(\Delta \tilde{\mathbf{y}}_{i,j}^t - \mathbf{A}_{i,j}^t \Delta \tilde{\mathbf{x}}_{i,j}^t) \Delta \tilde{\mathbf{x}}_{i,j}^{tT}$$

を計算する。ただし

$$E_{i,j} = \frac{1}{2} \|\Delta \tilde{\mathbf{y}}_{i,j}^t - \mathbf{A}_{i,j}^t \Delta \tilde{\mathbf{x}}_{i,j}^t\|^2$$

である。

Step6: 次のように逆解とヤコビ行列を調節する。

$$\mathbf{y}_{i,j}^{t+1} = \mathbf{y}_{i,j}^t + \alpha(t) \nabla^2 \tilde{\mathbf{y}}_{i,j} + \beta(t) \mathbf{A}_{i,j}^t [\mathbf{x}_{i,j}^d - \mathbf{g}(\mathbf{y}_{i,j}^t)]$$

$$\mathbf{A}_{i,j}^{t+1} = \mathbf{A}_{i,j}^t + \frac{1}{\|\Delta \tilde{\mathbf{x}}_{i,j}^t\|^2} (\Delta \tilde{\mathbf{y}}_{i,j}^t - \mathbf{A}_{i,j}^t \Delta \tilde{\mathbf{x}}_{i,j}^t) \Delta \tilde{\mathbf{x}}_{i,j}^{tT}$$

3.2. 拡散学習方式の利点

拡散学習方式を用いた場合の利点として、次のようなものがある

1. 各節点における出力 \mathbf{y} は近傍の4点だけという局所的な情報によって更新できるため並列分散処理が可能である。
2. ヤコビ行列の逆行列 \mathbf{A} は式(4)を用いて更新してゆくため、既知である必要がない。
3. 少ない教師信号によって学習が可能である。

4. 逆運動学マップの自己組織化

4.1. 逆運動学問題

作業空間で与えられた作業を実行するために、マニピュレータの制御では手先位置から関節空間上の関節角度を求める逆運動学問題を解く必要がある。

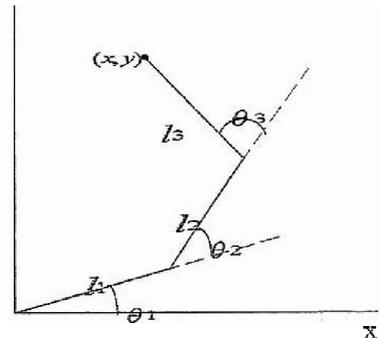


Fig.2 ロボットアームのモデル

今回用いるロボットシステムのモデルをFig.2に示す。平面3リンクマニピュレータなので、関節角度 θ から手先位置 \mathbf{x} までの関係は次のようになり、非線

形である。

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{bmatrix} \quad (5)$$

4.2. 拡散学習による自己組織化

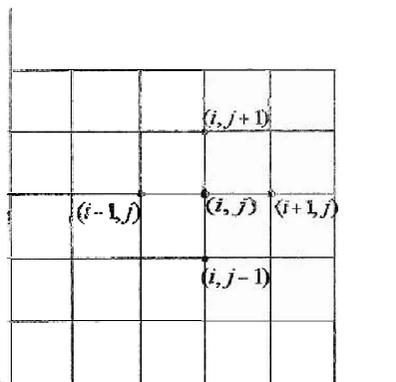


Fig.3 作業空間の分割

冗長自由度を持つ場合、手先位置に対する各関節角度の関係は

$$\mathbf{x} = \mathbf{f}(\boldsymbol{\theta}) \quad \mathbf{x} \in R^n, \boldsymbol{\theta} \in R^m \quad n < m \quad (6)$$

となり、解を一意的に決定することができない。よって、あらかじめ設定した作業空間に対していくつかの教師信号をもとに拡散学習法を用いた自己組織化マップを形成する。

Fig.3のように、あらかじめ作業空間を格子状に分割し、格子の4角の点に対する関節角度を教師信号として与える。その他の格子点に対する各関節角度を、拡散学習方式を用いて、学習していく。

4.3. 自己組織化マップのシミュレーション

実験に用いるロボットアームの自己組織化マップのシミュレーションを示す。

Fig.2のモデルのパラメータとして $l_1 = 0.15[m]$ 、 $l_2 = 0.12[m]$ 、 $l_3 = 0.10[m]$ 、またFig.3における作業空間として、二次元平面を 20×20 の 400 個のセルに分割したものを設定し、シミュレーションを行った結果を次に示す。

Fig.4からFig.6は 20×20 に分割した平面の各格子点に対する関節角度を縦軸にプロットした3次元グラフである。各図から隣り合う格子点の角度が連続的に分布していることがわかり、制御に適したマップとなっている。

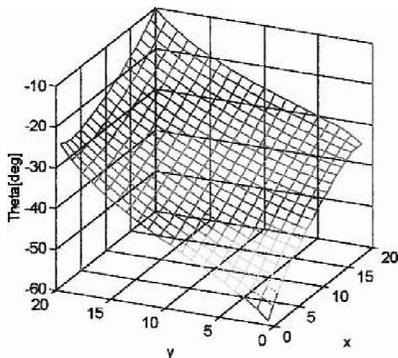


Fig.4 第1関節の角度分布

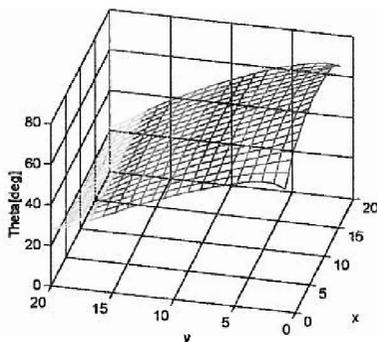


Fig.5 第2関節の角度分布

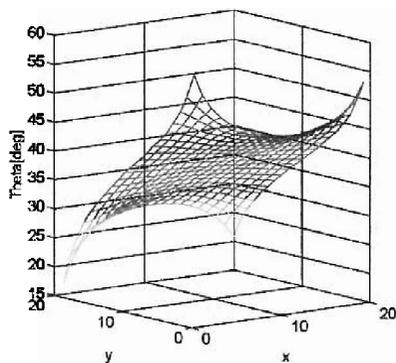


Fig.6 第3関節の角度分布

また算出した関節角度分布より各関節点と手先位置の分布を示したものをFig.7に示す。

0.1 0.2 0.3 0.4 0.5

Fig.7 各関節点と手先位置の分布

5. 最適制御の空間汎化

5.1. 最適制御問題

以下では、作業空間における n 自由度マニピュレータの point to point (PTP)運動の最適制御問題について考える。ロボットの運動方程式は一般に次のように与えられる

$$\mathbf{M}(\theta)\ddot{\theta} + \mathbf{C}(\theta, \dot{\theta}) = \boldsymbol{\tau} \quad (7)$$

ここで、 \mathbf{M} は慣性行列、 \mathbf{C} はコリオリ力を示すベクトル、 $\boldsymbol{\tau}$ は関節トルクベクトルである。初期状態と終端状態における関節角度ベクトルはそれぞれ $\theta(0) = \theta_0$ と $\theta(t_f) = \theta_f$ とする。

式(7)の状態空間表現は次のような一般的な非線形方程式で記述される。

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (8)$$

ただし、 $\mathbf{x} = (\theta, \dot{\theta})^T$ は状態ベクトル、 $\mathbf{u} = \boldsymbol{\tau}$ は制御入力ベクトルである。

最適制御問題はシステム式(8)に対して、評価関数

$$J = \int_0^{t_f} L(\mathbf{x}, \mathbf{u}, t) dt \quad (9)$$

を最小化する最適制御入力 $\mathbf{u}(t)$ を見つけ出すことである。Pontryagin の最小値解法を用いるとこの問

題は次のようなアルゴリズムによって解決される。

5.2. Pontryagin のアルゴリズム

Step 1: Pontryagin の状態関数を作成する

$$\mathbf{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}, \mathbf{u}, t) + L(\mathbf{x}, \mathbf{u}, t) \quad (10)$$

Step 2: 最適制御入力

$$\mathbf{u}^0 = \mathbf{u}^0(\mathbf{x}, \boldsymbol{\lambda}, t) \quad (11)$$

を

$$\frac{\partial \mathbf{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t)}{\partial \mathbf{u}} = 0 \quad (12)$$

より求める。

Step 3: 制御入力である式(5)を式(4)に代入し最適 H 関数

$$\mathbf{H}^0(\mathbf{x}, \boldsymbol{\lambda}, t) = \mathbf{H}(\mathbf{x}, \mathbf{u}^0, \boldsymbol{\lambda}, t) \quad (13)$$

Step 4: $2n$ 個の微分方程式

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{H}^0(\mathbf{x}, \boldsymbol{\lambda}, t)}{\partial \boldsymbol{\lambda}} \quad (14)$$

$$\dot{\boldsymbol{\lambda}} = - \frac{\partial \mathbf{H}^0(\mathbf{x}, \boldsymbol{\lambda}, t)}{\partial \mathbf{x}} \quad (15)$$

を境界条件 $\mathbf{x}(0) = \mathbf{x}_0$ 、 $\mathbf{x}(t_f) = \mathbf{x}_f$ の下に解く。

Step 5: Step 4 で求めた $\mathbf{x}(t)$ と $\boldsymbol{\lambda}(t)$ を式(11)に代入し、ロボットの最適入力 $\mathbf{u}(t)$ を得る。

ここで、最適制御 $\mathbf{u}(t)$ を得るためには、Step 4 の式(14)と式(15)を、与えられた初期状態と終端状態の下で連立して、解かなくてはならない。これは 2 点境界値問題で、一般的に非線形システムでは解析的にも数値的にも解を求めることが困難である。

2 点境界値問題の数値解法として、人間の腕の動きの生物学的解析に基づく解法が考えられる。そこで、次のようなトルク変化の評価関数が最小となることを考える

$$J = \frac{1}{2} \int_0^{t_f} (d\boldsymbol{\tau}/dt)^T (d\boldsymbol{\tau}/dt) dt \quad (16)$$

5.3. 最適制御のパラメトリゼーション

本研究では、まず制御入力を基底関数と重み行列の積に分解することを考える。

$$\begin{aligned}\tau_i(t) &= \sum_{k=1}^m [w_{ik} \phi_k(t) + b_{ik}] \\ &= [w_{i1}, \dots, w_{im}, \sum_{k=1}^m b_{ik}] [\phi_1(t), \dots, \phi_m(t), 1]^T \\ &\equiv \mathbf{w}_i \Phi(t)^T\end{aligned}\tag{17}$$

ここで、 $\Phi(t) = [\phi_1(t), \dots, \phi_m(t), 1]$ は基底関数ベクトルである。

これを用いると、トルクベクトルは以下のように表せる。

$$\boldsymbol{\tau}(t) = \mathbf{w} \Phi(t)^T\tag{18}$$

ここで、 $\mathbf{w} = [w_1, w_2, \dots, w_n]^T$ は重み行列である。

パラメトリゼーションを行うことによって、制御入力は時間に関する基底関数と時不変な重み行列とに分解され、時間関数である最適入力を時不変な重み関数によって特徴付けることができる。

5.4. 拡散学習を用いた空間汎化

最適制御のパラメトリゼーションにより、終端関節角度ベクトルと重み行列の関係は次のようになる。

$$\boldsymbol{\theta} = \mathbf{g}(\mathbf{w})\tag{19}$$

よって最適制御を得るためには最適な重み行列を以下のようにして求めればよい。

$$\mathbf{w} = \mathbf{g}^{-1}(\boldsymbol{\theta})\tag{20}$$

この冗長非線形関数の逆解を求めるために拡散学習方式を適用する。

これによって得られた重み行列を用いて、最終的に、ロボットの最適制御入力を構成できる。

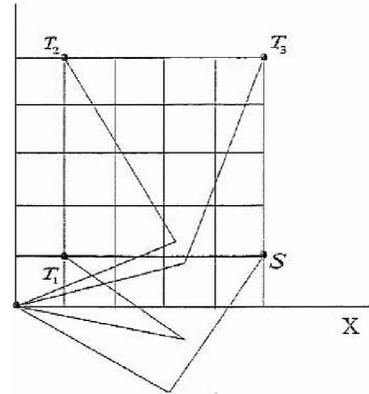


Fig.8 2自由度マニピュレータのモデル

Fig.8のようなモデルに対して、始点Sから各終端 T_1 、 T_2 、 T_3 までの最適入力をあらかじめ教師信号として与え、その他の格子点までの最適入力の重み行列を式(3)、(4)を用いることによって、各格子点までの最適制御入力を得ることができる。

6. おわりに

本研究では、拡散学習理論を用いて、冗長マニピュレータの視覚・運動協調の自己組織化について、運動学制御と動力学最適制御の両面から考察を与えた。今後の課題として、実際のロボットシステムにおけるパラメータ同定を行い、この自己組織化マップの形成シミュレーションと動作実験を検討する必要がある。

参考文献

- 1) Z.W.Luo and M.Ito "Diffusion-based learning theory for organizing visuo-motor coordination" Biol.Cybern,79,279-289,1998.
- 2) M.Kawato Y.Maeda Y.Uno R.Suzuki "Trajectory formation of arm movement by cascade neural network model based on minimum torque-change criterion" Biol.Cybern ,62,275-288,1990
- 3) D.Mgorinevsky "modeling of direct motor program learning in fast human motions" Biol.Cybern,69,219-228,1993