

ロボット最適運動制御の空間汎化

Diffusion-based Spatial Generation of Optimal Control for Redundant Manipulators

○浅黄 義昭*, 羅 志偉**, 渡部 慶二*, 遠藤 茂*

○Yoshiaki Asagi*, Zhi-wei Luo**, Keiji Watanabe*, Shigeru Endo*

*山形大学大学院理工学研究科 **理化学研究所

*Yamagata University **Riken

キーワード: 拡散学習(Diffusion-based Learning), 最適制御(Optimal Control), 基底関数(Radial Basis Function), 冗長マニピュレータ(Redundant Manipulators)

連絡先: 〒992 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
浅黄 義昭, Tel: (0238)26-3178, E-mail: asagi@ewata.yz.yamagata-u.ac.jp

1. はじめに

複雑な作業を行うロボットの手本として、人間の動作が挙げられる。人間はその自由度の多さのためにさまざまな作業が柔軟に実現できると考えられる。実際に、人間の上肢だけを見ても7自由度が存在する。自由度が多ければ手先を固定してもさまざまな姿勢をとることができるので、障害物を回避しながら目標位置まで手先を移動させることが可能となる。

しかしながら、冗長自由度を持つマニピュレータでは、手先位置から関節角度(姿勢)を求める逆運動学問題を解くにあたって、手先を固定しても何通りもの姿勢をとることができるため、その姿勢を決めなければならないという問題がある。また、ある点からある点までのアームの到達運動の場合には、最適な軌道を描くように制御入力を求めなければならない。本研究では、図1のようなスカラ型3リンクマニピュレータを例に取り上げ、拡散学習方式を用いてロボットアームの最適制御入力を求める手法について検証する。

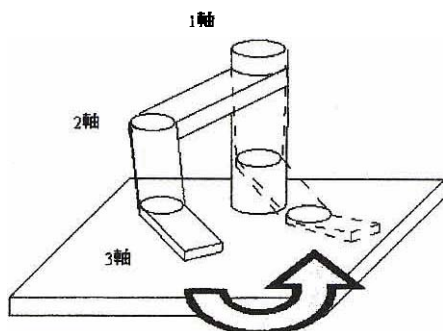


図1 3リンクマニピュレータ

2. 最適運動制御問題

制御対象(ロボットマニピュレータ)の状態方程式は一般的に次のように与えられる。

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (2.1)$$

ここで、最適の動作特性を議論するために、次のような一般的な形をもつ積分形の評価関数

$$PI = \int_0^T L(\mathbf{x}, \mathbf{u}, t) dt \quad (2.2)$$

を用いる。この評価関数の値が最小(最大)であれば、 $\mathbf{u}$ が最適制御入力となる。最適制御 $\mathbf{u}$ を求めるために、Pontryagin の状態関数

$$H(\mathbf{x}, \mathbf{u}, \lambda, t) = \lambda^T \mathbf{f}(\mathbf{x}, \mathbf{u}, t) + L(\mathbf{x}, \mathbf{u}, t) \quad (2.3)$$

を導入する。この状態関数を用いた最適制御入力 $\mathbf{u}$ の計算アルゴリズムは、以下のようになる。

Step1 式(2.3)のPontryaginのH関数を作る。

Step2 方程式 $\frac{\partial H(\mathbf{x}, \mathbf{u}, \lambda, t)}{\partial \mathbf{u}} = 0$ を解いて

$$\mathbf{u}^0 = \mathbf{u}^0(\mathbf{x}, \lambda, t) \quad (2.4)$$

を求める。

Step3 Step2で求めた $\mathbf{u}^0$ をStep1のH関数に代入して、最適Pontryagin関数 H^0 を求める。

Step4 1対の $2n$ 個の1階微分方程式

$$\begin{aligned} \dot{\mathbf{x}} &= \frac{\partial H^0(\mathbf{x}, \lambda, t)}{\partial \lambda} \\ \dot{\lambda} &= - \frac{\partial H^0(\mathbf{x}, \lambda, t)}{\partial \mathbf{x}} \end{aligned} \quad (2.5)$$

を $2n$ 個の境界条件 $\mathbf{x}(t_i)$ と $\mathbf{x}(t_f)$ のも

とで解く。

Step5 Step4で求めた解を式(2.4)に代入して最適制御入力を求める。

このアルゴリズムのStep4において式(2.5)の解は、一般に初期値 λ^i を与え、 $\mathbf{x}$ が最終値 $\mathbf{x}^f$ と一致するように λ^i を変えていくことにより求めることが必要で、解析的にも数値的にも求めるのが困難な2点境界値問題である。

これに対して、生体における到達運動動作は

$$J = \frac{1}{2} \int_0^t (d\tau/dt)^T (d\tau/dt) dt \quad (2.6)$$

というトルク変化が最小になる特徴を持つことから、以下の数値計算法が提案されている：

Step1 状態変数 $\mathbf{x}$ の初期状態 $\mathbf{x}(0) = \mathbf{x}_0$ と初期トルク $\tau(t)$ を $t=0$ から $t=t_f$ まで与える。

Step2 終端時刻 t_f までのロボット関節の時間応答を計算し、終端関節角度ベクトル θ_N を得る。

Step3 関節角度誤差 $\theta_f - \theta_N$ より

$$\frac{d\tau_j}{ds} = -\frac{\partial J}{\partial \tau_j} = \frac{\partial \theta_N}{\partial \tau_j} (\theta_f - \theta_N) + \lambda (\tau_{j-1} - \tau_j + \tau_{j+1})$$

for all $j=1,2,\dots,N$ ($N\Delta t = t_f$)

として、誤差 $\theta_f - \theta_N$ が減少するように $\tau(t)$ を修正してStep2に戻る。

ここで、 $\frac{\partial \theta_N}{\partial \tau_j}$ は感度行列と呼ばれ、時刻 j にお

るトルク τ_j を変化したときの終端値の変化をあらわすもので、その計算に、終端値から逆伝播を行うことが必要となる。それを時系列のすべてのトルク入力に対して行わなくてはならないため、膨大な計算量になってしまう。

3. 最適運動制御の空間汎化

前章で示した最適制御入力の計算には膨大な計算量を必要とするため、作業平面における任意の二点間の最適制御入力を一つ一つ求めていくことは容易には実行できない。そこで、本研究では、最適制御の空間汎化問題を考え、いくつかの点までの最適入力があらかじめ既知であるという前提で、拡散学習方式を応用した計算手法を考案する。

3.1. 制御入力のパラメトリゼーション

まず、最適制御の空間汎化を行うために有効とな

る制御入力のパラメトリゼーションについて述べる。

パラメトリゼーションの方法として、基底関数分解(RBF)を用いる。

まず、時間関数 $\tau(t)$ を次式のように分解していく。

$$\begin{aligned} \tau_i(t) &= \sum_{k=1}^m [w_{ik} \phi_k(t) + b_{ik}] \\ &= \left[w_{i1}, \dots, w_{im}, \sum_{k=1}^m b_{ik} \right] [\phi_1(t), \phi_2(t), \dots, \phi_m(t), 1]^T \\ &\equiv \mathbf{w}_i \Phi(t)^T \end{aligned} \quad (3.1)$$

ここで、 τ_i は制御入力ベクトルの i 番目の要素である。 $\Phi(t) = [\phi_1(t), \phi_2(t), \dots, \phi_m(t), 1]$ は基底関数ベクトルであり、 ϕ_k ($k=1,2,\dots,m$)は時間 t に関する基底関数で、その一例として、次のように与える。

$$\phi_k = e^{-\frac{(t-t_k)^2}{\sigma_k^2}} \quad (3.2)$$

ここで、 t_k と σ_k は平均と変化のパラメータである。

また、 $\mathbf{w} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m]^T$ は重み行列であり、ロボットの初期状態と終端状態によって決まる。

このように制御入力が基底関数と時不変な重み行列とに分解され、図2で示すように、重み行列 $\mathbf{w}$ から制御入力 $\tau(t)$ を合成することができる。

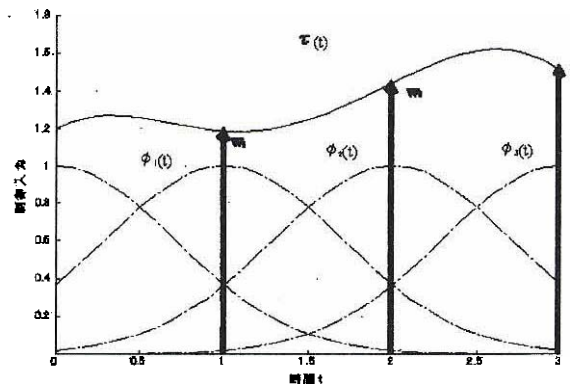


図2 制御入力のパラメトリゼーション

3.2. 空間汎化

ロボットの一般的な運動方程式は次のように表される。

$$\mathbf{M}(\theta)\ddot{\theta} + \mathbf{C}(\theta, \dot{\theta}) + \mathbf{g}(\theta) = \tau(t) \quad (3.4)$$

ここで、 $\mathbf{M}$ は慣性行列、 $\mathbf{C}$ は遠心力・コリオリ力ベクトル、 $\mathbf{g}(\theta)$ は重力項、 $\tau(t)$ 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。こ

こで $\mathbf{x} = (\theta, \dot{\theta})^T$ 、 $\mathbf{u} = \tau(t)$ とすると、式(3.4)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。

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (3.5)$$

式 (3.5) と前節 3.1 の制御入力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な非線形関数となる。

$$\mathbf{x}_N = \mathbf{g}(\mathbf{w}) \quad (3.6)$$

よって最適制御入力を得るための重み行列は次式より求めることができる。

$$\mathbf{w} = \mathbf{g}^{-1}(\mathbf{x}_N) \quad (3.7)$$

しかしながら、この非線形関数の逆解を求めるのは困難であるため、拡散学習方式を適用する。

まず、図 3 のように作業平面を格子状に分割した後、平面のある点をスタートポイント S として、そこから 4 角の点 T_1 、 T_2 、 T_3 と T_4 までの最適制御入力を前章のような方法で数値計算によって求める。そして、最適制御入力のパラメトリゼーションを行って各点 T_1 、 T_2 、 T_3 と T_4 までの最適制御入力の重み行列を求める。

求めた重み行列を教師信号として、スタートポイント S から残りの格子点までの重み行列を拡散学習方式で求めていく。

ここで、 $\mathbf{w}_{i,j}$ は S 点から (i, j) 点までの制御入力の重みで、式 (3.1) に代入することによって、 S 点から (i, j) 点までの最適制御入力求められる。

なお、2 点間到達運動の最終的な目標は作業平面の格子点へ手先位置を到達させることである。しかし、逆運動学は非線形であるので、式 (3.10) で用いる角度ベクトルの目標値 $\theta_{i,j}^d$ を拡散学習による逆運動学解法を用いてあらかじめ求めておくことによって手先の終端位置での誤差がなくなる。

4. シミュレーション

まず、図 3 に示すように、各リンクの長さは $l_1 = 0.15[m]$ 、 $l_2 = 0.12[m]$ 、 $l_3 = 0.10[m]$ とし、 $0.25[m] \leq x \leq 0.35[m]$ 、 $-0.05[m] \leq y \leq 0.05[m]$ の作業平面を 20×20 の 400 個の格子に分割する。

そして、拡散学習方式を用いて、各格子点におけるマニピュレータの姿勢を決定しておく。

従来の方を用いて、拡散学習に必要なとなる教師信号を求める。図 4、図 5、図 6 に始点を (10,10) とした場合の教師信号となる手先軌道、最適制御入力と関節角度変化を示す。

図 3 3 リンクマニピュレータのモデル

具体的には、まず、

$$\mathbf{w}_{i,j}^{t+1} = \frac{1}{4} \alpha(t) (\mathbf{w}_{i,j-1}^t + \mathbf{w}_{i,j+1}^t + \mathbf{w}_{i-1,j}^t + \mathbf{w}_{i+1,j}^t) \quad (3.8)$$

の空間拡散による重み行列の更新を行い、また、つぎの誤差修正に用いる逆ヤコビ行列は、

$$\mathbf{A}_{i,j}^{t+1} = \mathbf{A}_{i,j}^t + \frac{1}{\|\Delta \tilde{\theta}_{i,j}^t\|} (\Delta \tilde{\mathbf{w}}_{i,j}^t - \mathbf{A}_{i,j}^t \Delta \tilde{\theta}_{i,j}^t) \Delta \tilde{\theta}_{i,j}^{tT} \quad (3.9)$$

で更新していく。ここで、 $\Delta \tilde{\theta}^t = \theta^t - \theta^{t-1}$ は関節角度の誤差、 $\Delta \tilde{\mathbf{w}}^t = \mathbf{w}^t - \mathbf{w}^{t-1}$ は重み行列の誤差を表す。解が収束した後、次式を用いて誤差修正を行う。

$$\mathbf{w}_{i,j}^{t+1} = \mathbf{w}_{i,j}^t + \beta \mathbf{A}_{i,j}^t \{\theta_{i,j}^d - \mathbf{g}(\mathbf{w}_{i,j}^t)\} \quad (3.10)$$

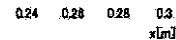


図 4 従来の手法で求めた教師信号となる手先軌道

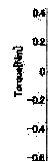


図 5 従来の手法で求めた始点(10,10)から終点(1,20)までの最適制御入力

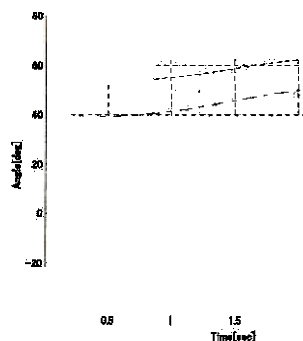


図6 従来の手法で求めた始点(10,10)から
終点(1,20)までの関節角度変化

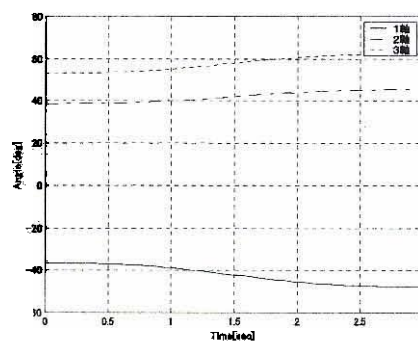


図9 空間汎化による始点(10,10)から
終点(5,5)までの関節角度変化

図7、図8、図9に、始点を(10,10)とした場合における空間汎化後の手先軌道、最適制御入力と関節角度変化を示す。

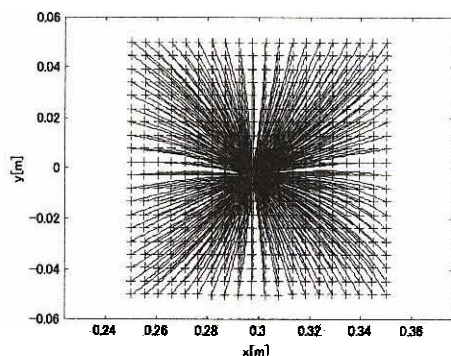


図7 (10,10)を始点とした手先軌道

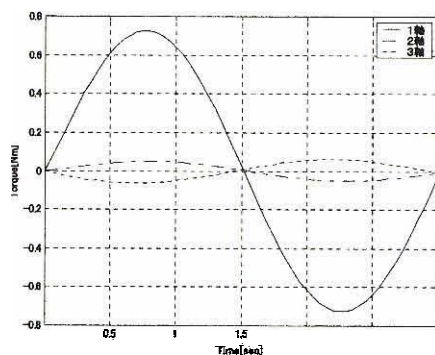


図8 空間汎化による始点(10,10)から
終点(5,5)までの最適制御入力

5. おわりに

本研究では、コンピュータシミュレーションを通して拡散学習方式を応用した最適運動制御の空間汎化法について検証した。結果として無駄の少ない最適な制御入力が見られることと、計算量の大幅な減少を確認した。

今後の課題として、実験システムを用いた検証を行う必要がある。また、作業空間に障害物が存在する場合の軌道について考察していく必要がある。

参考文献

- 1) 伊藤 義治:修士学位論文「拡散学習を用いた冗長ロボットの視覚・運動協調制御」
2001年 山形大学大学院理工学研究科博士前期課程電子情報工学専攻
- 2) 須山 郁夫:修士学位論文「冗長マニピュレータにおける逆運動学マッピングの並列分散学習」1995年 豊橋技術科学大学大学院工学研究科 修士課程 情報工学専攻
- 3) 「図解 ロボット制御入門」1997年 株式会社オーム社 川村 貞夫