

# ニューラルネットワークによる線形写像行列の算出に基づく マニピュレータの動作生成

山崎 公俊<sup>\*1</sup>

## Motion Generation of Manipulators Based on Linear Mapping Calculation by Neural Networks

Kimiotoshi Yamazaki<sup>\*1</sup>

<sup>\*1</sup> Shinshu University  
4-17-1 Wakasato, Nagano, Nagano, 380-8553, Japan

In this paper, we describe a motion generation method for articulated manipulators. The proposed method is composed of two phases: similar experience search and motion modification. In the latter phase, a neural network is used to calculate the amount of the modification. This process roughly ensures trajectory reproducibility. The effectiveness of the proposed method was proven by means of simulation assuming a planar manipulator and a 3D manipulator.

### 1. はじめに

本研究の目的は、過去の動きの経験を生かして、目標姿勢を実現するためのマニピュレータの動きを効率的に生成することである。すなわち、事前にマニピュレータに様々な動きをさせておき、それを知識として蓄えておく。そして、目標姿勢が与えられたら、過去におこなった動きを参考にして、適切な動きをその場で素早く生成する。

本稿で提案する手法は、大きく二つの部分からなる。一つ目は、目標手先姿勢を入力として、それと似た手先姿勢を目標としていたときの過去のマニピュレータの動きを検索する部分である。もう一つは、検索された動きを微調整して、目標手先姿勢を実現するためのマニピュレータの動きを生成する部分である。ポイントは、後者における微調整の方法である。提案手法では、手先姿勢と関節角度列の関係を記述する線形写像行列の算出を、ニューラルネットワークによって近似する。これを利用することで、目標手先姿勢に至るための関節角度列のセットを、その場での明示的な干渉チェック等をおこなわずに得る。

### 2. 動作生成のアプローチ

動作生成のための事前準備として、動作データの収集をおこなう。手順は次の通りである。まず、ロボットが動作する環境の地図  $map_{ref}$  を様々なパターンで用意する。その後、各地図において様々な手先目標

姿勢  $\mathbf{x}_{ref}$  を設定して、それぞれにおけるロボットの関節角度列  $Q_{ref} = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M)$  を生成し、さらに各  $\mathbf{q}$  におけるヤコビ行列を計算し、 $Q_{ref}$  についての  $J_{ref} = \{\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_M\}$  を得る。また、 $Q_{ref}$  によって実現される手先姿勢の列  $X_{ref} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$  を計算する。なお、 $\mathbf{x}_M = \mathbf{x}_{ref}$  である。そして、 $Q_{ref}$ ,  $X_{ref}$ ,  $J_{ref}$ ,  $map_{ref}$  を一組として記録する。こうして収集したデータセットは、後の動作生成処理において、過去の類似した状況を検索するために用いられる。

提案手法は、次のように書くことができる。

$$Q = G(\mathbf{x}_{tar}, X_{ref}, map_{tar}, map_{ref}) \quad (1)$$

すなわち、手先の目標姿勢、現在の環境マップ、過去の類似データ  $(X_{ref}, map_{ref})$  を利用する。動作生成の手順は次の通りである。まず、手先目標姿勢  $\mathbf{x}_{tar}$  と現在の環境地図  $map_{tar}$  が与えられる。そして、検食用データセットの中から、手先目標姿勢が  $\mathbf{x}_{tar}$  に近く環境地図が  $map_{tar}$  に似ているデータの組  $(X_{ref}, map_{ref})$  を検索する。これらのデータをニューラルネットワークの入力とし、手先姿勢の修正量  $\Delta X = \{\Delta \mathbf{x}_1, \Delta \mathbf{x}_2, \dots, \Delta \mathbf{x}_M\}$  と、ヤコビ行列のセット  $J_{tar} = \{\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_M\}$  を出力として得る。最後に、 $X_{ref}$ ,  $\Delta X$  と  $J_{tar}$  を利用して、新たなマニピュレータの動作列  $Q_{tar}$  を求める。

### 3. 関節角度列の計算

ニューラルネットワークへの入力には  $\mathbf{x}_{tar}$ ,  $X_{ref}$ ,  $map_{tar}$ ,  $map_{ref}$  である。ここで、手先姿勢パラメータの次元を  $P$  とすると、 $X_{ref}$  は  $P \times M$  の行列である。

<sup>\*1</sup> 信州大学学術研究院工学系 (〒380-8553 長野県長野市若里4-17-1) [kyamazaki@shinshu-u.ac.jp](mailto:kyamazaki@shinshu-u.ac.jp)

また、環境地図は、 $D$ 次元の空間を一辺 $L$ 個ずつに区切ったグリッドもしくはボクセルで表すものとして、 $D$ 次元のテンソル表現にしておく。

これらを並列にニューラルネットワークへ入力し、畳み込み層、プーリング層、全結合層などを通して、二つの二次元配列を得る。一つは $J_{tar}$ である。これは $(N \times P) \times M$ の行列であり、各列は $N \times P$ のヤコビ行列 $\mathbf{J}_m$ を $(N \times P) \times 1$ に並び替えたものである。もう一つは $\Delta \mathbf{X}$ であり、 $P \times M$ の行列である。

マニピュレータの動きを算出するためには、 $J_{tar}$ を1列ずつ参照し、各列を $N \times P$ の行列 $\mathbf{J}_m$ に成形する。それと $\Delta \mathbf{X}$ 内の1列 $\Delta \mathbf{x}_m$ を利用して、次の計算をおこなう。

$$\Delta \mathbf{q} = \mathbf{J}_m^+ \Delta \mathbf{x}_m \quad (2)$$

ここで、 $\mathbf{J}_m^+$ は $\mathbf{J}_m$ の逆行列もしくは疑似逆行列である。このようにして得た $\Delta \mathbf{q}$ を、 $Q_{ref}$ 内の各 $\mathbf{q}$ に加えることで、手先目標姿勢へ至るまでのマニピュレータの動作列を得る。

#### 4. シミュレーションによる検証

3次元4関節のシリアルリンクマニピュレータを仮想的に構築し、提案手法の有効性を調べた。マニピュレータの全長は1.0mとした。作業空間は、原点から各軸正方向に広がる一辺1.0mの空間とし、マニピュレータの根元はその原点に固定されているものとした。この空間の任意の位置に、所定サイズの立方体を障害物として挿入し、マニピュレータはそこへ侵入できないものとした。また、マニピュレータの初期姿勢は、各関節角度をすべて0 radとし、まっすぐ伸びた状態であるととした。

学習処理では、マニピュレータの動きをRRT\*(<sup>1</sup>)を用いて生成した。そして、 $(\mathbf{x}_{tar} - \mathbf{x}_{ref})$ ,  $J_{ref}$ ,  $map_{tar}$ ,  $map_{ref}$ を入力として、 $\Delta X_{tar}$ および $J_{tar}$ を出力として、ニューラルネットワークの学習をおこなった。このときのネットワークの構成は、シンプルさを重視して、畳み込み層2層以下、プーリング層2層以下、全結合層2層以下として様々に試した。基本的には畳み込み層が2層あれば比較的精度の良い回帰が可能であった。

比較手法として、入出力の異なる三種類の手法(A), (B), (C)も実装した。

**手法(A):** 入力を $(\mathbf{x}_{tar} - \mathbf{x}_{ref})$ ,  $X_{ref}$ ,  $map_{tar}$ ,  $map_{ref}$ として、出力を $Q_{tar}$ もしくは $X_{tar}$ とする。これは、式(1)を直接的に実装したものである。

**手法(B):** 入力を $\mathbf{x}_{tar}$ ,  $map_{tar}$ ,  $map_{ref}$ として出力を $Q_{tar}$ もしくは $X_{tar}$ とする。すなわち、現在の地図

Table 1 The results of motion generation by a 3D-4DoF manipulator. 50k and 25k mean 50000 and 25000 training data, respectively. SI means Sampling Interval[mm], EP means End-effector Pose, and JA means Joint Angles.

	50k, SI=50	25k, SI=33
Proposed	1.61(2.28)	1.14(1.72)
Method (A)	JA:7.33(3.95), EP:6.28(3.83)	JA:7.55(4.97), EP:7.22(9.64)
Method (B)	JA:7.05(3.83), EP:5.57(4.36)	JA:7.33(4.89), EP:9.05(19.25)
Method (C)	JA:8.32(3.58), EP:3.56(1.53)	JA:9.25(4.12), EP:7.75(14.95)

と類似した地図の違いに基づいて動きを決定する。

**手法(C):** 入力を $\mathbf{x}_{tar}$ ,  $map_{tar}$ として出力を $Q_{tar}$ もしくは $X_{tar}$ とする。式(3)をそのまま実装したものであり、類似データの検索処理を必要としない。

表1にシミュレーションの結果をまとめた。表中のSIは、検索用データの取得間隔である。セル内の数値は、手先位置誤差の平均および標準偏差(mm)である。それぞれ絶対値を採った。方法(A)~(C)については、出力を $X_{tar}$ とする場合(EP)と $Q_{tar}$ とする場合(JA)の二パターンを実装した。

表1での誤差平均の大きさは、いずれも1mm台であり、一定の精度と再現性が見て取れる。また、当然ながら、検索用データの取得間隔を小さくしたほうが、生成された動作の誤差も小さく抑えられた。また、方法(A)~(C)では、出力を $X_{tar}$ とする場合の方が $Q_{tar}$ とする場合よりもやや良い性能になる傾向が認められた。

#### 5. まとめ

本稿では、過去の動きの経験を生かして、目標手先姿勢を実現するためのマニピュレータの動きを生成する方法について述べた。線形写像行列の算出をニューラルネットワークを用いておこなう方法を提案し、マニピュレータの全長に対して手先位置の誤差が0.1%程度に抑えられることを確認した。

#### 謝 辞

本研究の一部はJSPS 科研費(JP18K19809)の補助を受けたものである。

#### 参 考 文 献

- (1) S. Karaman, E. Frazzoli: "Sampling-based algorithms for optimal motion planning," The International Journal of Robotics Research, Volume 30 Issue 7, pp. 846-894, 2011.