

ニューラルネットワークを用いた移動物体の視覚追跡処理

函館高専 ○藤原 真吾 浜 克己

要 旨

移動物体を視覚追跡するロボットアームシステムを設定し、入力としての位置に関する視覚情報と出力としての関節角変化量の対応づけに教師あり学習のニューラルネットワークを用いて、軌道追跡のシミュレーションを行う。

1. 緒 言

一般に、実ロボットの制御における大きな欠点として、環境やタスクの変化に対する柔軟性の欠如がある。一例として、本研究では平面上を任意に移動する物体が、マニピュレータの先端に取り付けられたカメラによって観測されるシステムを取り扱う。実際のタスクは、物体の動きに追従して、物体上の一定の高さに常にカメラを位置づけることである。しかし、実現のためには、移動物体に対する視覚情報処理とマニピュレータの関節変化量の計算並びにその操作によって生ずる時間遅れの問題を解決する必要がある。そこで、本研究ではこれらの対応づけにニューラルネットワークを導入し¹⁾²⁾、学習点の与え方に着目しながら、本手法の有効性について確認する。

2. 問題設定

本研究で取り扱うシステム構成を図1に示す。また、それぞれについて以下を仮定する。

① マニピュレータ

- ・ウエストの回転角 θ_1 を $0^\circ \sim 180^\circ$ とする。
- ・ハンド先端の向きは常に移動平面と垂直で、その高さが一定となるような姿勢を取る。

② 移動物体

- ・追跡される物体の移動範囲は、上記の姿勢におけるマニピュレータの可動範囲とする。
- ・移動物体は等速で移動する。

③ 視覚センサ (カメラ)

- ・ハンドの先端に取り付ける。
- ・常にカメラ座標系の中心位置で物体を捕らえるように追跡する。

以上より、任意時間ステップにおける処理は以下の手順で行われる。

- 1) ハンド先端のカメラから得られる移動物体の入力画像に基づき、このカメラ座標系において中心位置と移動物体の現在位置との変位を求める。
- 2) この変位を0にするようなマニピュレータの各関節の相対的な角度変化を、ニューラルネットワークの出力より獲得する。
- 3) これをマニピュレータに与え、以上の処理を繰り返しながら、連続的な追跡を行う。

図1において、基準座標系XYZにおけるハンド先端の位置と、マニピュレータの各関節角との間には次のような関係がある。

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \begin{bmatrix} 0 \\ 0 \\ L_4 \\ 1 \end{bmatrix} \quad (1)$$

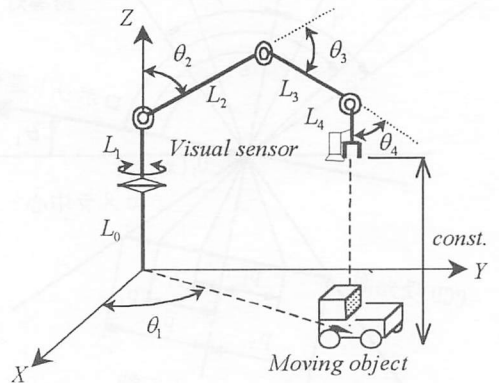


Fig.1 System configuration

$$A_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & L_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_i & -\sin \theta_i & 0 \\ 0 & \sin \theta_i & \cos \theta_i & L_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (i = 2, 3, 4)$$

これより、各関節角を定めればマニピュレータの姿勢が決まり、ハンド先端の位置は一意に決定されることがわかる。しかし、逆にハンド先端の位置を与え、これを満たすようなマニピュレータの姿勢、すなわち各関節角を求めるには、式(1)の逆変換マトリックスを解く必要がある。しかし、特に冗長性を有するマニピュレータの場合、自由度が増すにつれ、その解を解析的に求めるには多大な時間を必要とし、実時間処理が困難となる。

そこで、近似解ではあるが、次に示すニューラルネットワークを用いて位置情報と関節角度の対応づけを行い、計算量を軽減して追跡処理の高速化を図る。

3. ニューラルネットの適用

本研究では、階層型のフィードフォワードニューラルネットワークを用い、その学習パターンとして入力ベクトル $\mathbf{x} = (x_1, x_2, \dots, x_N)$ と理想出力ベクトル $\mathbf{t} = (t_1, t_2, \dots, t_M)$ の対が与えられる。中間層および出力層の出力関数には、以下に示すシグモイド関数を使用する。

$$\sigma(s) = \frac{1}{1 + e^{-s}} \quad (2)$$

いま、学習パターン数を P とすると、ネットワークの実際の出力ベクトル $\mathbf{z} = (z_1, z_2, \dots, z_M)$ と理想出力ベクトル \mathbf{t} との残差平方和を以下のような評価関数として設定する。

$$E = \frac{1}{2} \sum_{p=1}^P \sum_{m=1}^M (z_m^p - t_m^p)^2 \quad (3)$$

学習は、以上の評価関数を最小にするように進められ、その学習則には最急降下法を用いた誤差逆伝播法を使用する。

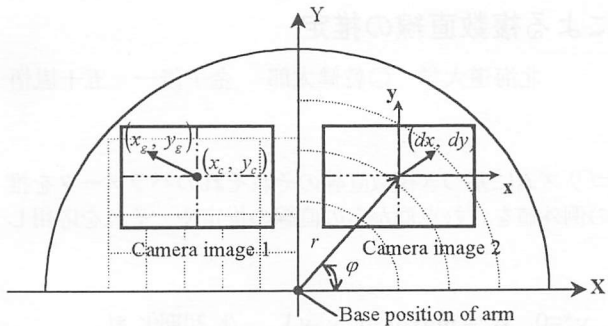


Fig.2 Methods for giving input data

これより、ネットワークの荷重更新は以下のようになる。

$$w(k+1) = w(k) + \Delta w(k) \quad (4)$$

$$\Delta w_{ij}(k) = -\eta \frac{\partial E(k)}{\partial w_{ij}(k)}$$

ここで、 k は時間ステップ、 η は学習率を表す。
 本研究で使用するニューラルネットワークは教師あり学習のタイプに属し、教師データ(理想出力ベクトルに相当)を必要とする。出力はマニピュレータを動かすのに必要な各関節角度の変化量であるが、望ましい出力を得るためには、対応づけを行う入力の与え方に十分考慮する必要がある。そこで、ここでは学習点を含む入力データの与え方に関して、以下に示す2つの方法を試みる(図2)。

1) 直交座標入力(カメラ画像1)

マニピュレータのベースを原点とする基準座標系におけるカメラ画像中心の絶対位置 (x_c, y_c) と、カメラ画像中の移動物体の位置、つまりハンド先端の目標となる絶対位置 (x_g, y_g) の4入力を設定する。これより、写像関数を F 、各関節角度の変化量ベクトルを $\Delta\theta$ とすると、その関係は以下のように表現される。

$$\Delta\theta = F(x_c, y_c, x_g, y_g) \quad (5)$$

2) 極座標入力(カメラ画像2)

基準座標系の原点からカメラ画像中心までの距離 r とその角度 ϕ 、カメラの画像中心位置から画像中の移動物体の位置までの相対変位 (dx, dy) の4入力を設定する。写像関数による関係づけは以下ようになる。

$$\Delta\theta = F(r, \phi, dx, dy) \quad (6)$$

4. 計算機実験

第1象限($0^\circ \leq \phi \leq 90^\circ$)に対し、可動範囲(太線)内に極座標入力を用いて学習するために設定したすべての学習点を図3に示す。なお、ここで使用する各パラメータの値は次の通りである。

- ・ 入力層: 4, 中間層: 40, 出力層: 4
- ・ 学習係数: $\eta = 0.3$
- ・ 学習データ数: 72
- ・ 学習回数: 10,000
- ・ 入力値正規化範囲: $[0, 1]$
- ・ 移動物体速度: 10 mm/step (一定)
- ・ ハンド先端高: 300 mm (一定)

以上の学習データおよび各パラメータを用いて学習した結果に基づき、矩形軌道と楕円軌道に関して追跡を行った結果を図4と図5にそれぞれ示す。いずれの場合も、与えられた軌道にかなりよく追従していることがわかる。

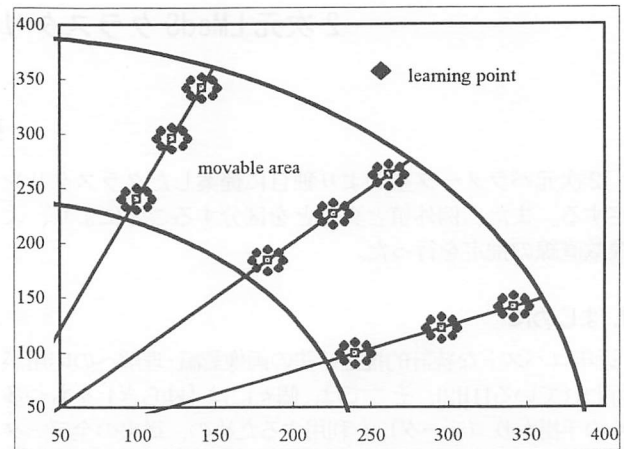


Fig.3 Setting of learning points

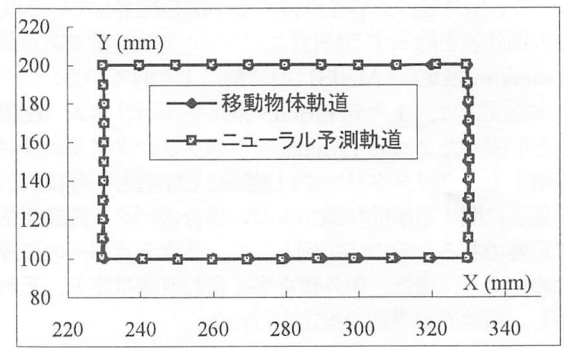


Fig.4 Results of tracking (1)

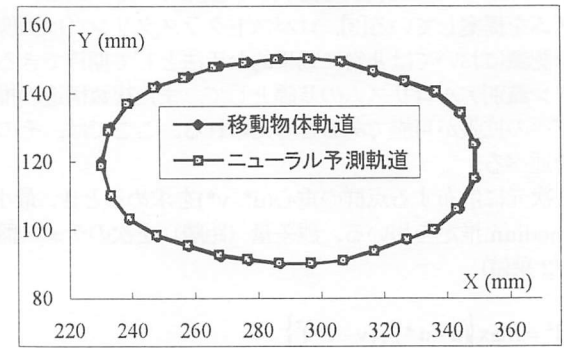


Fig.5 Results of tracking (2)

5. 結言

本研究では、マニピュレータの関節角と移動物体に対する視覚情報の対応づけにニューラルネットワークの導入を試みた。速度が変化する物体に対しても追跡ができるように、学習データの与え方などを工夫することが今後の課題である。

参考文献

- 1) Tanaka, K. et al., An Autonomous Robot System controlled by Neural Networks, Proc. of IMACS/SICE International Symposium on Robotics, Mechatronics and Manufacturing Systems '92, 1257-1262, 1992
- 2) Schram, G. et al., Visual Tracking of moving objects using a neural network controller, Robotics and Autonomous System, 18, 293-299, 1990