

# ニューラルネットワークを用いた フレキシブルアームの最適フィードバック制御

2020SC091 田代耕大

指導教員：陳幹

## 1 はじめに

産業ロボットアームの軽量化によって剛性が低下し、振動やたわみが発生する [1]. 近年、フレキシブルアームに対してニューラルネットワークを用いて制御する研究が行われている [2]. 本研究では、仮想受動関節モデル [1] のアームの長さを参考にし、剛性などの特性が均一であると仮定して、フレキシブルアームの水平方向の動きをニューラルネットワークを用いてフィードバック制御を行うことである.

## 2 モデリング

本研究で用いるフレキシブルアームの概略図を次に示す.

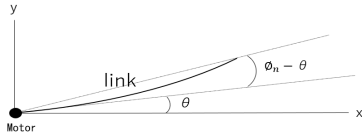


図1 仮想受動関節が n 個の 1 リンクアームの概略図

この図の Moter は、直流モーターを表している.  $\theta$  はモーターの回転角度,  $\phi_n$  はモーターの回転角度を絶対座標とした, モーターに近い方から n 番目の仮想受動関節の角度を表している. VirtualJoint は、仮想受動関節を表している. 仮想受動関節が n 個の 1 リンクアームの位置エネルギー, 並進運動エネルギー, 回転運動エネルギー, 減衰エネルギー, は式 (1), (2), (3), (4) のようになる.

$$K = \frac{1}{2}(k_{12}(\phi_1 - \theta)^2 + \dots + k_{1n+1}(\phi_n - \phi_{n-1})^2) \quad (1)$$

$$T_1 = \frac{1}{2}(m_{11}(x_{11}^2 + y_{11}^2) + \dots + m_{1n+1}(x_{1n+1}^2 + y_{1n+1}^2)) \quad (2)$$

$$T_2 = \frac{1}{2}(I_{11}\dot{\theta}^2 + \dots + I_{1n+1}\dot{\phi}_n^2) \quad (3)$$

$$D = \frac{1}{2}(d_{12}(\dot{\phi}_1 - \dot{\theta})^2 + \dots + d_{1n+1}(\dot{\phi}_n - \dot{\phi}_{n-1})^2) \quad (4)$$

モーターに与える電圧 V と  $\tau$  の関係式は論文 [3] と仕様書より次のようになる.

$$\tau = \frac{\eta_g \eta_m K_t K_g (V - K_m K_g \dot{\theta})}{R_m} \quad (5)$$

式 (1)-(5) から上の式からオイラー・ラグランジュの運動方程式を得て, 状態方程式に表したものを式 (6) に示す.

$$\frac{d}{dt}x(t) = Ax(t) + Bu(t) \quad (6)$$

$$A = M^{-1} \cdot (H + MO_1 + d + k)$$

$$B = M^{-1} \cdot MO_2$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & M_{00} & \dots & M_{0n} \\ 0 & 0 & 0 & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & M_{n0} & \dots & M_{nn} \end{bmatrix}$$

$$MO_1 = \begin{bmatrix} O_{n+1,n+1} & E_{10} \\ O_{n+1,n+1} & mo \end{bmatrix}$$

$$mo = \begin{bmatrix} -\frac{\eta_m \eta_g K_t K_g^2 K_m}{R_M} - Be & 0 & \dots & 0 \\ 0 & 0 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

$$H = \begin{bmatrix} O_{n+1,n+1} & O_{n+1,n+1} \\ O_{n+1,n+1} & h \end{bmatrix}$$

$$h = \begin{bmatrix} H_{00} & H_{01} & \dots & \dots & H_{0n} \\ H_{10} & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & H_{n-1n} \\ H_{n0} & \dots & \dots & H_{nn-1} & H_{nn} \end{bmatrix}$$

$$MO_2 = \begin{bmatrix} 0 & \dots & 0 & \frac{\eta_m \eta_g K_t K_g}{R_M} & 0 & \dots & 0 \end{bmatrix}^T$$

$$d = \begin{bmatrix} O_{n+1,n+1} & O_{n+1,n+1} \\ Spring & O_{n+1,n+1} \end{bmatrix}$$

$$k = \begin{bmatrix} O_{n+1,n+1} & O_{n+1,n+1} \\ O_{n+1,n+1} & Damper \end{bmatrix}$$

$$x(t) = [\theta(t) \quad \phi_1(t) \quad \dots \quad \phi_n(t) \quad \dot{\theta}(t) \quad \dot{\phi}_1(t) \quad \dots \quad \dot{\phi}_n(t)]^T$$

上の式の  $u(t)$  は入力電圧, M は慣性項, H は状態変数を含んだ非線形項,  $E_{10}$  は, 10 行 10 列の単位行列,  $O_{n+1,n+1}$  は, n+1 行 n+1 列のゼロ行列, Spring はバネの項, Damper はダンパの項である. 行列 h は対称行列である.

### 3 ニューラルネットワーク制御器

#### 3.1 ニューラルネットワーク

本研究では、文献 [4] の勾配法の学習法を利用して勾配降下法を用いて最適フィードバック制御を行う。シミュレーションで扱う状態方程式は導出したので、ニューラルネットワークは、入力層、1層の中間層、出力層の3層の階層構造型で学習をする。今回使用するニューラルネットワーク図を下に示す。

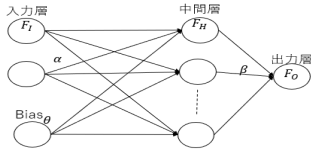


図2 ニューラルネットワーク図

入力層と中間層の重みを  $\alpha$ 、中間層から出力層の重みを  $\beta$ 、バイアスを  $\theta$  とする。  $F_I$  は入力値の活性化関数、  $F_H$  は中間層の活性化関数、  $F_O$  は出力値の活性化関数である。

#### 3.2 制御器設計

文献 [4] の最適制御問題を解き、文献と同等の制御器設計を行い、学習可能な制御器を設計後、フレキシブルアームを制御する。本研究では評価関数  $J$  内にある状態変数に重み  $Q$  をつけた。  $Q$  は正の数である。式 (7) に評価関数を示す。

$$J = \sum_{t=1}^T \left[ \sum_{j=1}^n Q_1 x_1^2(t) + Q_2 x_2^2(t) + \dots + Q_u u_1^2(t-1) \right] \quad (7)$$

勾配を降下するための評価関数  $J$  の勾配を式 (8) に示す。

$$\frac{\partial J}{\partial w} = \sum_{t=1}^T \left[ \sum_{j=1}^n Q \frac{\partial x_j(t)}{\partial w} \frac{\partial J_0}{\partial x_j(t)} + \sum_{j=1}^r Q_u \frac{\partial u_j(t-1)}{\partial w} \frac{\partial J_0}{\partial u_j(t-1)} \right] \quad (8)$$

$T$  は指定したシミュレーションの時間である。重みやバイアスは式 (9) によって更新する。

$$\omega = \omega - \eta \frac{\partial J}{\partial w} \quad (9)$$

$\eta$  は学習率である。

### 4 シミュレーション結果

シミュレーションは仮想受動関節が9個の場合で行う。各関節の初期値は  $0.8[\text{rad}]$ ,  $0[\text{rad/s}]$  とし、目標値を  $0[\text{rad}]$ ,  $0[\text{rad/s}]$  としてシミュレーションを行った。図4にニューラルネットワークの学習を0回、100回、240回したモーターの角度のシミュレーション結果を示す。

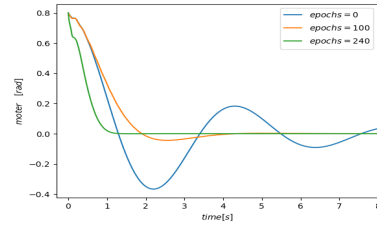


図3  $\theta$  の学習回数が0, 100, 240回のシミュレーション結果

この図の *epoch* は学習回数である。学習を繰り返すことで、モーターの角度がオーバーシュートを抑えて目標値で収束することができた。図4に240回学習したフレキシブルアームのモーターの角度と先端の角度を示す。

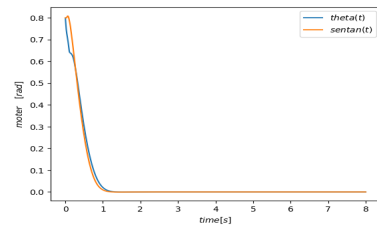


図4 240回学習したモーターの角度と先端角度のシミュレーション結果

図4の結果からフレキシブルアームのたわみと振動を抑制して目標値に収束していることが分かる。

### 5 おわりに

本研究では、仮想受動関節を増やし、フレキシブルアームの水平方向の動きをニューラルネットワークを用いてフィードバック制御器を設計した。ニューラルネットワークを用いて勾配降下法で学習し、学習回数を増やすことで、設定時間内に目標値に収束しない応答結果から、目標値に収束してオーバーシュートが生じない応答結果が得られた。

### 参考文献

- [1] 吉川 恒夫, 田村 正人: フレキシブルアームに対する仮想受動関節モデルの有効性の検討. 日本ロボット学会誌 17-2, 250/259, 1999
- [2] H. Zhang, X. Li, G. Xu, H. Liu: Fuzzy Neural Networks Control of Flexible Robotic Manipulator Based on Lumped Spring-Mass Model. Proceedings of the 40th CCC, Shanghai China, July, 2021
- [3] 水戸健詞: 最適レギュレータによるフレキシブルアームの制振制御, 南山大学数理工学部 2010 年度卒業論文, 2010
- [4] 中西 弘明, 幸田 武久, 井上 紘一: ニューラルネットワークによる最適フィードバック制御系の構成. 計測自動制御学会論文集, 33-9, 882/889, 1997