

ニューラルネットワークを利用した ボールスクリーシステムの経路制御

M2017SC012 横山 壮太

指導教員：高見 勲

1 はじめに

本研究では二軸のボールスクリーシステムに対して、経路制御設計を行う。二軸のボールスクリーシステムの制御においては、X軸とY軸を独立に制御し、目標値と追従値の位置誤差を最小化する手法が一般的である。しかしこの手法では目標速度が高速である場合やX軸とY軸の特性が大きく異なっている場合、経路誤差が大きくなってしまふ。そこで、本論文では、各軸独立ではなく、両軸を互いに考慮に入れた経路制御を行う。未来の目標経路を考慮に入れながら両軸で発生する経路誤差を最小化する手法の一つとして、ニューラルネットワークを用いた経路制御が考えられる。ニューラルネットワークを用いた経路制御では未来の目標経路を考慮に入れつつ、経路誤差を評価関数に取り入れ、新たな目標経路を探索することができる。そして、シミュレーションからその有用性を示す。

2 制御対象とモデリング

本研究で使用しているボールスクリーシステムとは、モータと接続されているスクリーとナット内部のボールを回転させることによりモータの回転運動をテーブルの前後運動に変換するものである。このボールスクリーシステムのモデルを図1に示す。

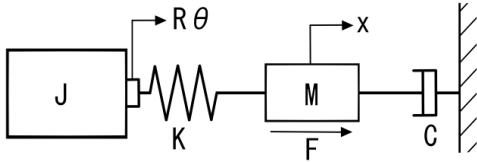


図1 システムのモデル

次に、制御対象の運動方程式を導出する。モータの回転角を $\theta(t)$ [rad]、テーブルの変位を $x(t)$ [m]、モータからの入力トルクを $T(t)$ [Nm] とすると、モータに関する運動方程式が以下のように求められる。

$$J_m \ddot{\theta} = T(t) - R_b K_s (R_b \theta - x(t)) \quad (1)$$

また、テーブルに関する運動方程式は

$$M_t \ddot{x}(t) = K_s (R_b \theta - x(t)) - C_v \dot{x}(t) \quad (2)$$

となる。ここで、回転系の全慣性モーメントを J_m [Nms²]、直線系ばね定数を K_s [N/m]、直線系の粘性摩擦係数を C_v [Ns/m]、テーブルの質量を M_t [kg]、ボールねじ定数を R_b [m/rad] とする。テーブルの運動に比べモータの運動が

速いことからモータの回転運動の遅れは無視する。式(1)、(2)より以下の式が得られる。

$$M_t \ddot{x}(t) + C_v \dot{x}(t) = \frac{1}{R_b} T(t) \quad (3)$$

状態変数を $x_p(t) = [x(t), \dot{x}(t)]^T$ 、入力を $u(t) = T(t)$ とすると、制御対象の状態空間表現は

$$\begin{aligned} \dot{x}_p(t) &= \begin{bmatrix} 0 & 1 \\ 0 & -\frac{C_v}{M_t} \end{bmatrix} x_p(t) + \begin{bmatrix} 0 \\ \frac{1}{M_t R_b} \end{bmatrix} u(t) \\ &= A_p x_p + B_p u(t) \end{aligned} \quad (4)$$

$$\begin{aligned} y(t) &= [1 \ 0] x_p(t) \\ &= C_p x_p(t) \end{aligned} \quad (5)$$

となる。表1に制御対象の物理パラメータを示す。

表1 ボールスクリーシステムの物理パラメータ

J_m	回転系全慣性モーメント	1.147×10^{-5} [Nms ²]
K_s	直線系ばね定数	1×10^8 [N/m]
C_v	直線系粘性摩擦定数	1083[Ns/m]
M_t	テーブルの質量	2.988[kg]
R_b	ボールねじ定数	3.184×10^{-4} [m/rad]

3 制御系設計

3.1 最適サーボシステムの設計

最適サーボシステムを設計する。式(4)、(5)のシステムに対し評価関数

$$J = \int_0^{\infty} (\tilde{x}(t)^T Q \tilde{x}(t) + r \tilde{u}(t)^2) dt \quad (6)$$

を最小化するような積分型コントローラを設計する。ここで、 Q 、 r は重み行列である。以下のような拡大偏差システムを定義する。

$$\begin{aligned} \dot{\tilde{x}}(t) &= \begin{bmatrix} A_p & 0 \\ -C_p & 0 \end{bmatrix} \tilde{x}(t) + \begin{bmatrix} B_p \\ 0 \end{bmatrix} \tilde{u}(t) \\ &= A_e \tilde{x}(t) + B_e \tilde{u}(t) \end{aligned} \quad (7)$$

$$\begin{aligned} e(t) &= [-C_p \ 0] \tilde{x}(t) \\ &= C_e \tilde{x} \end{aligned} \quad (8)$$

評価関数 J を最小化するようなコントローラゲイン K_e は最適レギュレータ理論により、拡大偏差システムに対するリカッチ方程式とその正定対称解により唯一に定まる。

4 ニューラルネットワーク (NN) を用いた経路制御

4.1 経路制御

一般的に行われている経路制御手法としては、目標値と追従値の位置誤差を最小化することにより、結果的に経路誤差を最小化するものがある。しかし、この手法では目標値の速度が大きいときに位置誤差が大きくなり、その修正に大きな加速度が必要となる。位置誤差は各時刻における目標値と追従値の距離であり、経路誤差は最終的な目標軌道と応答軌道の誤差である。位置誤差は経路誤差を直接評価し得るものではなく、位置決め制御と経路制御は別の制御であると言える。特にこの位置誤差と経路誤差の差異は湾曲した経路など複雑な経路で大きくなる。当然、完全な位置決め制御は完全な経路制御となり得るが、ゲインが非常に大きなものとなり現実的ではない。そのため、両軸を考慮に入れた経路誤差を評価できる関数を用いる。その評価関数を最小化するような目標軌道と応答軌道が最小化される新たな仮想目標軌道を導出する。

4.2 ニューラルネットワークの役割

NN の役割の図を図 2 に示す。本研究の目的は仮想目標軌道 $R^*(t)$ を導出することである。仮想目標軌道 $R^*(t)$ が満たす条件はシステムの入力軌道に $R(t)$ を入れたとき、出力軌道である応答軌道 $P(t)$ が目標軌道 $R(t)$ と一致することである。NN の役割は目標軌道 $R(t)$ から仮想目標軌道 $R^*(t)$ を導出することである。

4.3 ニューラルネットワークの学習

図 2 において、NN の役割は目標軌道 $R(t)$ から仮想目標軌道 $R^*(t)$ を導出することである。仮想目標軌道が満たす条件より、NN は出力軌道 $P(t)(= R(t))$ から入力軌道 $R^*(t)$ を導出することと同じと考えることができる。

NN の学習の図を図 3 に示す。図 3 における目標軌道 $R(t)$ と応答軌道 $P(t)$ を学習データとする。NN の出力 $R'(t)$ が、 $R(t)$ に近づくように学習する。そのため、評価関数は次の式を用いる。

$$E(t) = \frac{1}{2} [\{R'_x(t) - R_x(t)\}^2 + \{R'_y(t) - R_y(t)\}^2] \quad (9)$$

ここで、下つき文字 x, y は各 x 軸と y 軸の軌道を表す。学習後では、NN はシステムの入力軌道から入力軌道を出力できる逆システムとなるため、図 2 における出力軌道 $P(t)(= R(t))$ から入力軌道 $R^*(t)$ を出力できる。

4.4 ニューラルネットワークの構造

仮想目標軌道 $R^*(t)$ を求める NN の内部構造を図 4 に示す。図 4 における入力層、第 1 中間層、第 2 中間層、出力層における各ノードは完全相互結合されている。中間層の数は 1 つでは学習効率が低く、多いと冗長になるため 2

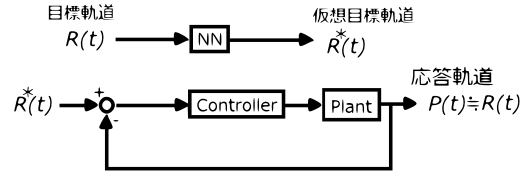


図 2 NN の役割の図

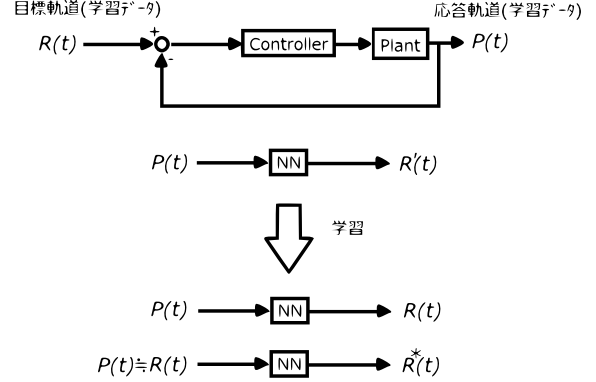


図 3 NN の学習の図

つとした。この NN において入力層のベクトルを $I(t)$ 、入力層と第 1 中間層の間の重み行列と閾値ベクトルを W_a と B_a 、第 1 中間層ベクトルを H_a 、第 1 中間層と第 2 中間層の間の重み行列と閾値ベクトルを W_b と B_b 、第 2 中間層ベクトルを H_b 、第 2 中間層と出力層の間の重み行列を W_c 、出力層ベクトルを $O(t)$ とする。出力層の要素は仮想目標軌道 $R^*(t)$ とする。この NN において第 1 中間層ベクトル H_a 、第 2 中間層ベクトル H_b 、出力層ベクトル $O(t)$ は各々以下の式で表される。

$$\begin{aligned} O_{ha} &= W_a I(t) - B_a \\ H_a &= F(O_{ha}) \end{aligned} \quad (10)$$

$$\begin{aligned} O_{hb} &= W_b H_a - B_b \\ H_b &= F(O_{hb}) \end{aligned} \quad (11)$$

$$O(t) = W_c H_b \quad (12)$$

ここで $F()$ はベクトルの各要素に下式の活性化関数 (Leaky ReLU 関数)

$$f(z) = \begin{cases} z & (z \geq 0) \\ 0.3z & (z < 0) \end{cases}$$

を適用する関数とする。学習中に勾配消失問題を防ぐために活性化関数には Leaky ReLU 関数を用いている。

4.5 ニューラルネットワークの重みと閾値の更新

NN の最適な重みと閾値の更新方法として評価関数 $E(t)$ を用いた誤差逆伝播法を使う。第 2 中間層と出力層の間の

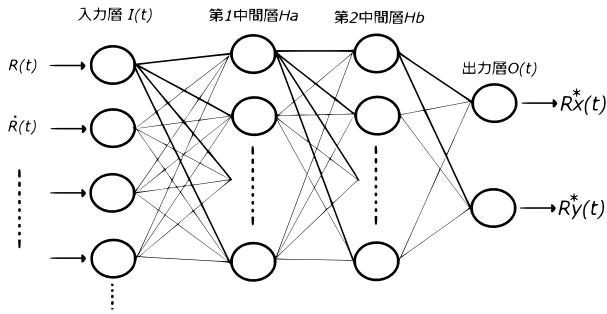


図4 NNの構造図

重み W_c の更新式は

$$\begin{aligned} W_c(t+1) &= W_c(t) - \alpha \frac{\partial E(t)}{\partial W_c(t)} \\ &= W_c(t) - \alpha \frac{\partial E(t)}{\partial R^*(t)} \frac{\partial R^*(t)}{\partial W_c(t)} \end{aligned} \quad (13)$$

となり、同様に第1中間層と第2中間層の間の重みと閾値の更新式は

$$\begin{aligned} W_b(t+1) &= W_b(t) - \alpha \frac{\partial E(t)}{\partial W_b(t)} \\ &= W_b(t) - \alpha \frac{\partial E(t)}{\partial R^*(t)} \frac{\partial R^*(t)}{\partial H_b(t)} \frac{\partial H_b(t)}{\partial W_b(t)} \end{aligned} \quad (14)$$

$$\begin{aligned} B_b(t+1) &= B_b(t) - \alpha \frac{\partial E(t)}{\partial B_b(t)} \\ &= B_b(t) - \alpha \frac{\partial E(t)}{\partial R^*(t)} \frac{\partial R^*(t)}{\partial H_b(t)} \frac{\partial H_b(t)}{\partial B_b(t)} \end{aligned} \quad (15)$$

となり、同様に入力層と第1中間層の間の重みと閾値の更新式は

$$\begin{aligned} W_a(t+1) &= W_a(t) - \alpha \frac{\partial E(t)}{\partial W_a(t)} \\ &= W_a(t) - \alpha \frac{\partial E(t)}{\partial R^*(t)} \frac{\partial R^*(t)}{\partial H_b(t)} \frac{\partial H_b(t)}{\partial H_a(t)} \frac{\partial H_a(t)}{\partial W_a(t)} \end{aligned} \quad (16)$$

$$\begin{aligned} B_a(t+1) &= B_a(t) - \alpha \frac{\partial E(t)}{\partial B_a(t)} \\ &= B_a(t) - \alpha \frac{\partial E(t)}{\partial R^*(t)} \frac{\partial R^*(t)}{\partial H_b(t)} \frac{\partial H_b(t)}{\partial H_a(t)} \frac{\partial H_a(t)}{\partial B_a(t)} \end{aligned} \quad (17)$$

となる。ただし α は学習率であり任意に値の調整が可能とする。これらの更新式を用いてシミュレーションを行い、NNの学習をすることで最適な重みと閾値が導出される。

4.6 ニューラルネットワークの入力層

NNの役割は目標軌道 $R(t)$ から仮想目標軌道 $R^*(t)$ をつくることである。そのためNNの入力層には目標軌道 $R(t)$ を用い、また制御系の入力軌道は出力軌道の速度がフィードバックされているため、目標軌道の速度 $\dot{R}(t)$ も用いる。

次の図5に目標軌道にステップ関数を用いた片軸の図

を示す。図5より、入力軌道の変化の影響が出力軌道の2.5[second]まで影響している。そのため、目標軌道の未来の軌道を考慮に入れる必要がある。NNの入力層には2.5秒先の目標軌道まで0.025秒間隔で $R(t)$, $R(t+0.025)$, $R(t+0.050)$, $R(t+0.075)$... $R(t+2.500)$ の101組の目標軌道の値を用いることとする。

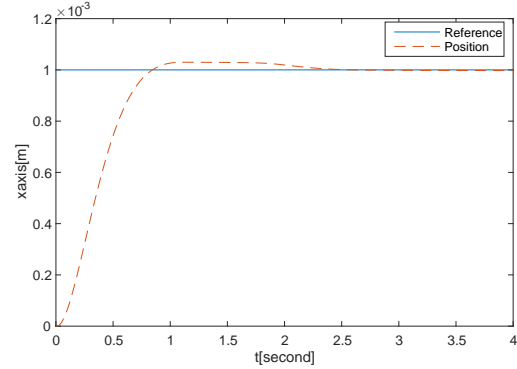


図5 片軸のステップ目標軌道に対する応答軌道

5 シミュレーションと実験結果

設計したNNを用いた経路制御と従来用いられてきた通常の位置決め制御を用いてシミュレーションと実験を行い、結果を比較する。NNの第1中間層と第2中間層のノード数は100個と50個とした、また学習率は0.3とした。NNの学習に用いた目標軌道は1周6[s]の円形軌道とする。またサンプルタイムは 10^{-3} [s] である。

NNを用いた x 軸の応答軌道と従来の応答軌道を図6と図7に示す。図6と図7より、NNを用いた目標軌道は従来の目標軌道より位相が進んでおり、NNを用いた応答軌道は従来の応答軌道に比べて時間遅れが抑制されている。

NNを用いた応答軌道と従来の応答軌道を次の図8と図9に示す。図8と図9より、6秒後の従来の応答軌道は円軌道の始点である原点に到達していないが、NNを用いた応答軌道は始点である原点にほぼ到達している。また、象限突起が抑制されていることも確認できる。

図9における各時刻の目標軌道と従来の応答軌道の差 $|Reference(t) - Position(t)|$ である経路誤差と、目標軌道とNNを用いた応答軌道の経路誤差を次の図10に示す。図10の各RMS値は、Error(LQ)が 2.4052×10^{-4} 、Error(LQ+NN)が 1.7629×10^{-5} となる。NNを用いた応答軌道は従来の応答軌道に比べ経路誤差が抑制されていることが確認できる。

6 おわりに

本研究では経路誤差に対し、ニューラルネットワークから導出した仮想目標軌道を用い、目標軌道への追従性能を高めた。これにより、時間遅れによる誤差発生抑制、及びその誤差の修正性能の向上を実験から確認した。

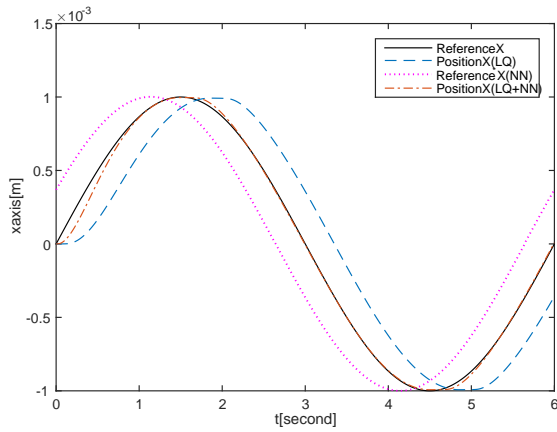


図 6 NN を用いた片軸の軌道の図 (シミュレーション)

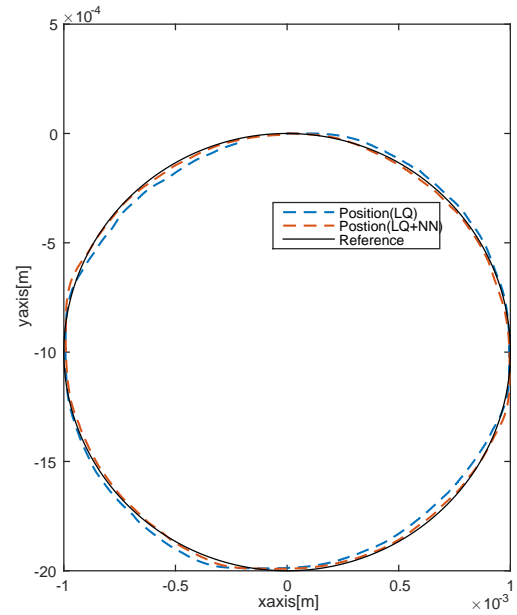


図 9 NN を用いた両軸の軌道の図 (実験)

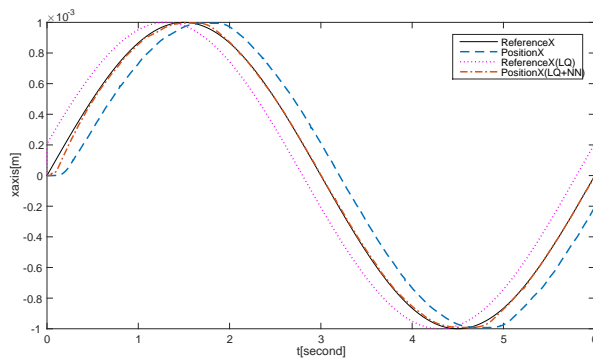


図 7 NN を用いた片軸の軌道の図 (実験)

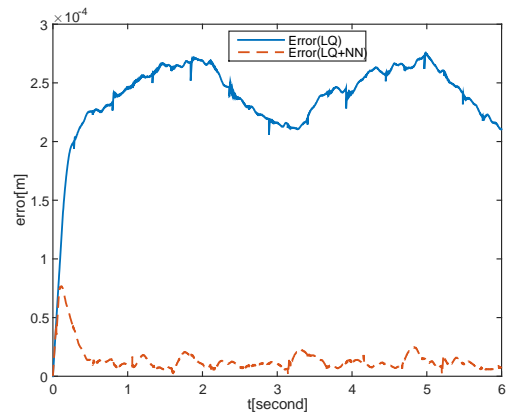


図 10 経路誤差の図 (実験)

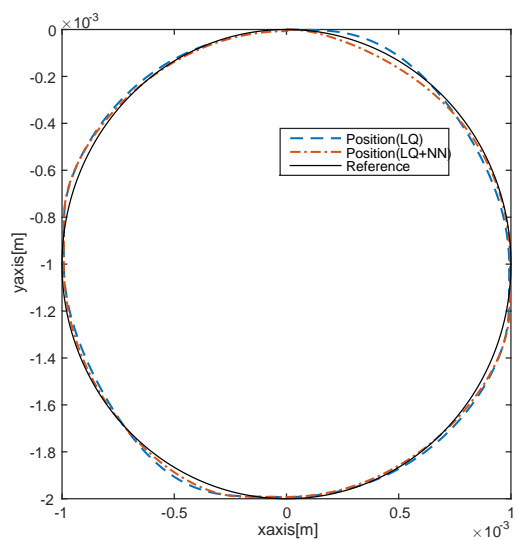


図 8 NN を用いた両軸の軌道の図 (シミュレーション)

参考文献

- [1] 江上 正, 豊田 修, 土谷 武士: 協調経路制御とそのリニア X-Y テーブルへの応用, 電気学会論文誌 D, Vol. 113, No. 12, pp. 1395-1402, 1993.
- [2] 江上 正, 天野 桂介: 最短目標点探索を用いた経路制御手法の NC フライス盤への適用, 計測自動制御学会論文集, Vol. 48, No. 12, pp. 898-906, 2012.
- [3] 藤中 透, 清水 真弥, 吉岡 理文, 大松 繁: ニューロ PID 制御を用いたハードディスクドライブの位置制御, 電気学会論文誌 C 分冊, Vol. 121-C, No. 3, pp. 580-585, 2001.
- [4] 山田 巧: 逆システムの構成法と制御への応用, 計測と制御, Vol. 36, No. 6, pp. 417-425, 1997.