

ニューラルネットワークを使った倒立振子の制御

The Control of The Inverted Pendulum using Neural Network

○美尾 竜太郎, 大久保 重範

○Ryotaro Mio, Shigenori Okubo

山形大学

Yamagata University

キーワード: ニューラルネットワーク制御 (Neural Network Control), 倒立振子 (Inverted Pendulum)
微分係数 (Differential Coefficient), 連続時間 (Continuous Time), 補償器 (Compensator)

連絡先: 〒992 山形県米沢市城南 4-3-16
山形大学 工学部 機械システム工学科 大久保研究室 美尾 竜太郎

Tel.: (0238)26-3246, Fax.: (0238)24-6445, E-mail: ryomio@mip3470.yz.yamagata-u.ac.jp

1. はじめに

あるプラントに対して制御系を設計する際に、プラントのモデリングを行わなければならない。しかし実際にはプラントが未知であったり、正確なモデリングが難しかったりする場合が多い。正確なモデリングが難しい場合には近似した数式を用いて制御系を設計することになる。ただしこの場合、数式を近似することで実際のプラントとの間に誤差が生じてしまう。これをモデル誤差という。本稿で扱うのは、近似した数式を使うことにより発生したモデル誤差をNN補償器が吸収する制御系を設計していく。近似した式からテーラー展開を用いて得られた微分係数 $(\partial x/\partial u)$ を使ってニューラルネットワーク(以後、NN)を作り、それを補償器としたもので制御系の設計をする。また、それが使われることを考え、シミュレーションによってその有効性を示す。

2. 倒立振子

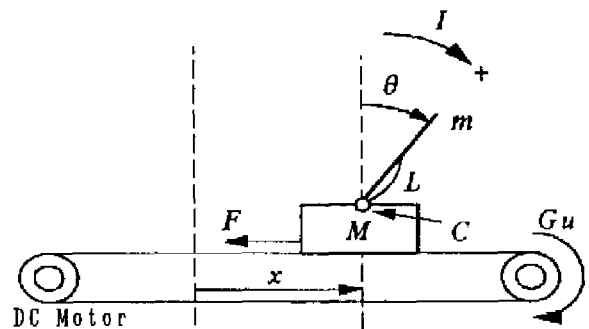


Fig. 1 Inverted pendulum

倒立振子の運動方程式は、ラグランジュの方程式より次のようになる。

$$(M+m)\ddot{x} + mL\ddot{\theta}\cos\theta + F\dot{x} - mL\dot{\theta}^2\sin\theta = Gu \quad (1)$$

$$mL\ddot{x}\cos\theta + (I + mL^2)\ddot{\theta} + C\dot{\theta} - mgL\sin\theta = 0 \quad (2)$$

また、各記号は以下で示すとおりである。

M	[kg]	台車系の等価質量
G	[N/V]	電圧/トルク変換係数
u	[V]	モータへの入力電圧
F	[-]	台車系の等価摩擦係数
C	[-]	軸の粘性摩擦係数
m	[kg]	振子の質量
I	[Nm]	振子の慣性モーメント
L	[m]	軸から振子の重心までの距離
g	[m/s ²]	重力加速度

また、評価関数 $J(t)$ を以下のようにする。式中の a, b は定数である。

$$J(t) = \frac{1}{2}(ax^2 + b\theta^2) \quad (3)$$

これで、 $J(t) = 0$ となれば $x = 0$ で振子が安定に倒立していることが分かる。

3. フィードバックを考慮した倒立振子における微分係数の獲得

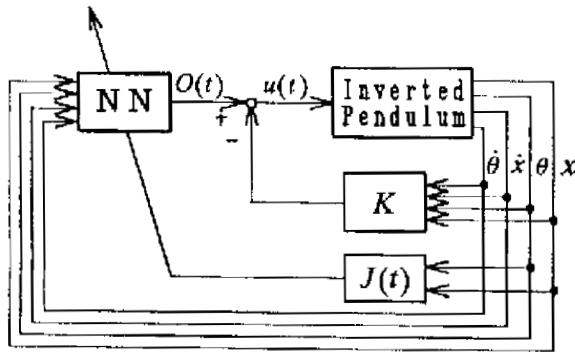


Fig. 2 Block diagram of control system

倒立振子を原点周りで線形近似すると、運動方程式は以下のようなになる。 $(O(t))$ は NN の出力

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4)$$

$$u(t) = -K^T x(t) + O(t) \quad (5)$$

また、上の式の各記号は以下のとおりである。

(K_n はフィードバック係数)

$$A = \frac{1}{\Delta} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a1 & a2 & a3 \\ 0 & b1 & b2 & b3 \end{bmatrix},$$

$$x(t) = \begin{bmatrix} x(t) \\ \theta(t) \\ \dot{x}(t) \\ \dot{\theta}(t) \end{bmatrix}, B = \frac{1}{\Delta} \begin{bmatrix} 0 \\ 0 \\ c1 \\ c2 \end{bmatrix},$$

$$K^T = [K_1 \ K_2 \ K_3 \ K_4] \quad (6)$$

さらに、式(6)内の記号は式(7)のようになる。

$$\left. \begin{aligned} \Delta &= (I + ml^2)(M + m) - m^2 l^2 \\ a1 &= -m^2 l^2 g \\ a2 &= -(I + ml^2)F \\ a3 &= mlC \\ b1 &= (M + m)mgl \\ b2 &= mlF \\ b3 &= -(M + m)C \\ c1 &= (I + ml^2)G \\ c2 &= -mlG \end{aligned} \right\} \quad (7)$$

よって、

$$\dot{x}(t) = (A - BK^T)x(t) + BO(t) \quad (8)$$

プラントのモデル式(8)を平衡点周りでテーラ一展開すると、

$$\begin{aligned} \dot{x}_s + \Delta \dot{x} &= (A - BK^T)x_s + (A - BK^T)\Delta x(t) \\ &\quad + BO_s + B\Delta O \\ &= (A - BK^T)x_s + BO_s \\ &\quad + (A - BK^T)\Delta x(t) + B\Delta O \end{aligned} \quad (9)$$

よって

$$\Delta \dot{x} = (A - BK^T)\Delta x(t) + B\Delta O \quad (10)$$

ここで $\Delta \dot{x} \approx 0$ と仮定し、 Δx を消去すると、

$$\begin{aligned} \frac{\partial x}{\partial O} &= -(A - BK^T)^{-1}B \\ &= \begin{bmatrix} -1/K_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (11)$$

また、式(3)より、 $\frac{\partial J(t)}{\partial \mathbf{x}(t)}$ は以下ようになる。

$$\begin{aligned} \frac{\partial J(t)}{\partial \mathbf{x}(t)} &= \begin{bmatrix} J(t) & J(t) & J(t) & J(t) \\ \mathbf{x}(t) & \theta(t) & \dot{\mathbf{x}}(t) & \dot{\theta}(t) \end{bmatrix} \\ &= [ax \quad b\theta \quad 0 \quad 0] \end{aligned} \quad (12)$$

4. NNの構成

4.1 NN内部

本稿で用いるNNは三層階層型であり、バックプロパゲーション法によって学習を行う。以下NNの数学的学習プロセスを本制御系の構成に従って記述する。NN各層のニューロン数は入力層 N 、中間層 M 、出力層 L とする。ある時刻 t におけるNNへの入力を $I_i(t)$ 、NNからの出力を $O_k(t)$ とする。入力層 i 番目のニューロンと中間層 j 番目のニューロンの間の結合荷重を $W_{ji}(t)$ 、中間層 j 番目のニューロンと出力層 k 番目のニューロン間の結合荷重を $V_{kj}(t)$ とすると、NN内部は次のように表現される。

$$\begin{aligned} I_i(t) &= [I_1(t), I_2(t), \dots, I_N(t)] \\ &= [\mathbf{x}(t), \theta(t), \dot{\mathbf{x}}(t), \dot{\theta}(t)] \end{aligned} \quad (13)$$

$$U_j(t) = \sum_i W_{ji}(t) I_i(t) + \theta_j(t) \quad (14)$$

$$H_j(t) = f\{U_j(t)\} \quad (15)$$

$$S_k(t) = \sum_j V_{kj}(t) H_j(t) + \gamma_k(t) \quad (16)$$

$$O_k(t) = f\{S_k(t)\} \quad (17)$$

ここで応答関数 $f(\cdot)$ は、 $[-L, L]$ の範囲内に出力するように加工したシグモイド関数を用いる。

$$f(x) = L \frac{1 - \exp(-x)}{1 + \exp(-x)} \quad (18)$$

$$f'(x) = \frac{1}{2L} \{L + f(x)\} \{L - f(x)\} \quad (19)$$

4.2 離散時間におけるNNの学習

NNは評価関数 $J(t)$ を最少にするように学習を行う。学習において結合荷重 $W_{ji}(t)$ 、 $V_{kj}(t)$ 、オフセット θ_j 、 γ_k の更新は以下の式によって行われ

る。ここで、 α_{dn} 、 β_{dn} は学習率であり、 Δt はNNのサンプリングタイムである。

$$V_{kj}(t + \Delta t) = V_{kj}(t) - \beta_{d2} \frac{\partial J(t)}{\partial V_{kj}(t)} \quad (20)$$

$$\gamma_k(t + \Delta t) = \gamma_k(t) - \alpha_{d2} \delta_k(t) \quad (21)$$

$$W_{ji}(t + \Delta t) = W_{ji}(t) - \beta_{d1} \frac{\partial J(t)}{\partial W_{ji}(t)} \quad (22)$$

$$\theta_j(t + \Delta t) = \theta_j(t) - \alpha_{d1} \sigma_j(t) \quad (23)$$

ここで式(20)、(21)、(22)、(23)内の $\partial J(t)/\partial V_{kj}(t)$ 、 $\delta_k(t)$ 、 $\partial J(t)/\partial W_{ji}(t)$ 、 $\sigma_j(t)$ は次のように展開できる。

$$\begin{aligned} \delta_k(t) &= \frac{\partial J(t)}{\partial \mathbf{x}(t)} \frac{\partial \mathbf{x}(t)}{\partial O_k(t)} \frac{\partial O_k(t)}{\partial S_k(t)} \\ &= -\frac{a}{K_1} \mathbf{x}(t) (1 + O_k(t)) (1 - O_k(t)) \end{aligned} \quad (24)$$

$$\begin{aligned} \frac{\partial J(t)}{\partial V_{kj}(t)} &= -\beta_{d2} \frac{\partial J(t)}{\partial \mathbf{x}(t)} \frac{\partial \mathbf{x}(t)}{\partial O_k(t)} \frac{\partial O_k(t)}{\partial S_k(t)} \frac{\partial S_k(t)}{\partial V_{kj}(t)} \\ &= -\beta_{d2} \delta_k(t) H_j(t) \end{aligned} \quad (25)$$

$$\begin{aligned} \sigma_j(t) &= \frac{\partial J(t)}{\partial \mathbf{x}(t)} \frac{\partial \mathbf{x}(t)}{\partial O_k(t)} \frac{\partial O_k(t)}{\partial S_k(t)} \frac{\partial S_k(t)}{\partial H_j(t)} \frac{\partial H_j(t)}{\partial U_j(t)} \\ &= \sum_k \delta_k(t) V_{kj}(t) \\ &\quad \times (1 + H_j(t)) (1 - H_j(t)) \end{aligned} \quad (26)$$

$$\begin{aligned} \frac{\partial J(t)}{\partial W_{ji}(t)} &= -\beta_{d1} \frac{\partial J(t)}{\partial \mathbf{x}(t)} \frac{\partial \mathbf{x}(t)}{\partial O_k(t)} \frac{\partial O_k(t)}{\partial S_k(t)} \\ &\quad \times \frac{\partial S_k(t)}{\partial H_j(t)} \frac{\partial H_j(t)}{\partial U_j(t)} \frac{\partial U_j(t)}{\partial W_{ji}(t)} \\ &= -\beta_{d1} \sigma_j(t) I_i(t) \end{aligned} \quad (27)$$

4.3 連続時間におけるNNの学習

ここでは連続時間におけるNNの学習の計算方法を示している。評価関数 $J(t)$ を最少にするという学習の方針は離散時間と同じだが、結合荷重 $W_{ji}(t)$ 、 $V_{kj}(t)$ 、オフセット θ_j 、 γ_k の更新は以下ようになる。ここで、 α_{cn} 、 β_{cn} は学習率である。

$$\dot{V}_{kj}(t) = -\beta_{c2} \frac{\partial J(t)}{\partial V_{kj}(t)} \quad (28)$$

$$\dot{\gamma}_k(t) = -\alpha_{c2} \frac{\partial J(t)}{\partial O_k(t)} \frac{\partial O_k(t)}{\partial S_k(t)} \quad (29)$$

$$\dot{W}_{ji}(t) = -\beta_{c1} \frac{\partial J(t)}{\partial W_{ji}(t)} \quad (30)$$

$$\dot{\theta}_j(t) = -\alpha_{c1} \frac{\partial J(t)}{\partial O_k(t)} \frac{\partial O_k(t)}{\partial S_k(t)}$$

$$\times \frac{\partial S_k(t)}{\partial H_j(t)} \frac{\partial H_j(t)}{\partial U_j(t)} \quad (31)$$

$\delta_k(t)$ 、 $\sigma_j(t)$ は式(24)、(26)と同じなので、式(28)～(31)から

$$\dot{V}_{kj}(t) = -\beta_{c2} \delta_k(t) H_j(t) \quad (32)$$

$$\dot{\gamma}_k(t) = -\alpha_{c2} \delta_k(t) \quad (33)$$

$$\dot{W}_{ji}(t) = -\beta_{c1} \sigma_j(t) I_i(t) \quad (34)$$

$$\dot{\theta}_j(t) = -\alpha_{c1} \sigma_j(t) \quad (35)$$

以上の式より、連続時間での学習制御が可能である。

5. シミュレーション結果

今回のNNの各層のニューロン数は入力層4、中間層10、出力層1で、ここで用いたフィードバック係数の極配置は[-5, -6, -7, -8]である。また、1)～3)までは離散時間による制御で、4)は連続時間による制御である。台車の初期位置は全て0.1[m]とした。

1) 初期角度 30 [deg] Fig.3

2) 初期角度 72 [deg] Fig.4

3) 初期角度 75 [deg] Fig.5

4) 初期角度 30 [deg] Fig.6

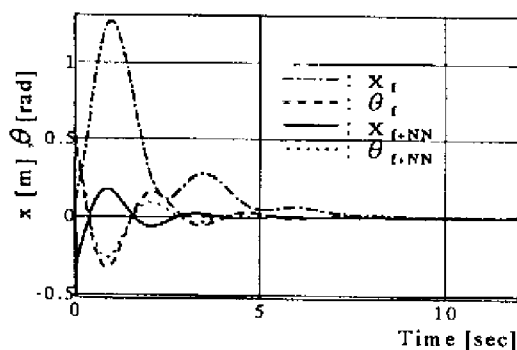


Fig. 3 Response of x and θ at 30[deg] in discrete time

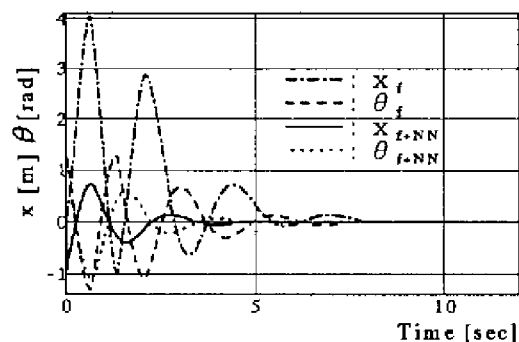


Fig. 4 Response of x and θ at 72[deg] in discrete time

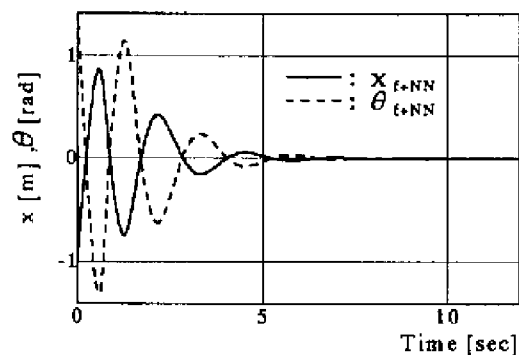


Fig. 5 Response of x and θ at 75[deg] in discrete time

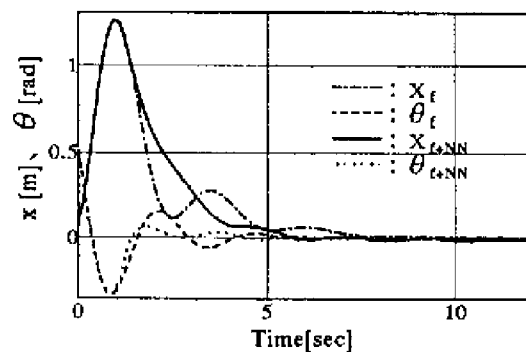


Fig. 6 Response of x and θ at 30[deg] in continuous time

6. おわりに

NNを補償器とした倒立振子の制御はフィードバックのみの制御と比べると、最初の2～3秒はフィードバックと同じような応答をするが、それ以後の過渡応答は明らかに改善されている。これは初期の段階ではNNの学習が不十分で、大部分をフィードバック制御に依存しているからであり、そのため同じような応答をする。しかし、NNの学習がある程度進むと、NNの補償器の効果が明確に現れてくる。よって、本シミュレーションにNNを補償器に使った制御方式が有効であることが確認された。

また、連続時間におけるNNでも今回の制御方式により、応答が改善されることが確認された。

参考文献

- 1) 齊藤 秀夫: 工業基礎振動学, 養賢堂, (1989)
- 2) 金井 喜美雄: 制御システム設計, 槇書店, (1983)
- 3) 大松 繁: ニューラルネットワークによる適応制御, 電気学会雑誌, 7, 503/506, (1992)
- 4) 馬目 聡, 荒 寛, 鈴木 穎二: ニューラルネットワークによる倒立振子の制御, 第33回 SICE 学術講演会予稿集, 335/336, (1994)