

ゼロバイアスを考慮したファジィニューラルネットワーク

M2022SC015 吉田裕基

指導教員：陳幹

1 はじめに

近年では知的制御が注目されている．この知的制御にはニューラルネットワークを用いたものや [1]，ファジィ推論を用いたものが存在する [2]．最近ではファジィニューラルネットワークと呼ばれる制御手法が提唱されている [3] [4]．これはファジィの解釈可能な制御器に対してニューラルネットワークの自動チューニング機能を応用する事で，両手法のメリットでデメリットを解消したハイブリッドな制御器である．しかしこの制御器は学習に際して 1 つの問題を抱えている．それは学習された制御器が必ずしもその制御対象に対して最適なものを学習しない事である．これは制御において注意すべき問題である．本論文では従来のファジィニューラルネットワークに対して改修を加えることで学習される不必要なバイアス（ゼロバイアス）の解消を試みる．またモデルだけでなく，学習されるパラメータについて一部の学習を制限する事でより不要なバイアスの学習の防止を図る．修正されたファジィニューラルネットワークをライントレースでシミュレーションし，その制御性能の変化を確認する．

2 プラントモデリング

今回使用する 2 輪自動ロボットのモデリングを行う．車両のモデリングは一般的な手法を使う [5]．2 輪自動ロボットの簡略化した図を図 1 に示す．また，それぞれの物理

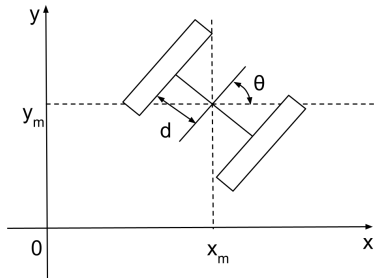


図 1 2 輪ロボットのダイナミクス

定数を以下の式で求める．

$$v_r(t) = r\dot{\phi}_r, v_l(t) = r\dot{\phi}_l, \quad (1)$$

$$\omega(t) = \frac{(v_r(t) - v_l(t))}{2d}, v(t) = \frac{(v_r(t) + v_l(t))}{2} \quad (2)$$

$$\dot{x}_m = v(t) \cos \theta(t), \dot{y}_m = v(t) \sin \theta(t), \dot{\theta} = \omega(t) \quad (3)$$

この時，式 (1) 式 (2) 式 (3) で使用される変数について表 1 に示す．

車輪の角加速度の導出は一般的なモーターの速度制御の運動方程式を用いる．

$$\ddot{\phi}_r = -K_1\dot{\phi}_r + K_2u_r(t), \ddot{\phi}_l = -K_3\dot{\phi}_l + K_4u_l(t). \quad (4)$$

表 1 2 輪モバイルロボットの物理変数の一覧とその意味

記号	内容
$v_r(t)[\text{m/s}]$	右車輪速度
$v_l(t)[\text{m/s}]$	左車輪速度
$r[\text{m}]$	車輪半径
$\phi_r(t)[\text{rad}]$	右車輪角度
$\phi_l(t)[\text{rad}]$	左車輪角度
$v(t)[\text{m/s}]$	本体速度
$\theta(t)[\text{rad}]$	車体向進角度
$\omega(t)[\text{rad/s}]$	車体向進角速度
$(x_m(t), y_m(t))[\text{m}]$	本体重心の位置
$d[\text{m}]$	車体重心と車輪中心までの距離

K_1, K_2, K_3 そして K_4 は左右のモータそれぞれの特性である． u_r と u_l はそれぞれ左右のモータに印加される入力電圧であり，その計算は以下の様に行う．

$$u_r(t) = u_0 + u(t), u_l(t) = u_0 - u(t) \quad (5)$$

u_0 はモバイルロボットが進み続けるために印加されるバイアス電圧である．これに対して制御入力 $u(t)$ を加えることでロボットを操作する．

3 ファジィニューラルネットワークコントローラ

本論文では堀川のファジィニューラルネットワークを用いる [6]．

3.1 ファジィ推論

本研究で引用したファジィニューラルネットワークの構造を図 2 に示す．図の A 層から D 層が前件部ファジィ

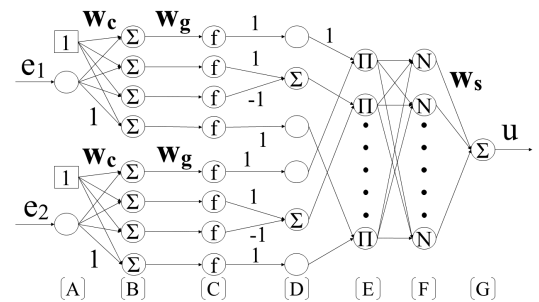


図 2 ファジィコントローラーの計算フロー

演算を，D 層から G 層が後件部ファジィ演算を表している．まず A 層に目標値との誤差 e_1 と誤差の時間差分 e_2 を入力する． e_2 は以下の用に求める．

$$e_2(k) = \frac{(e_1(k) - e_1(k-2))}{2} \quad (6)$$

図の丸いノードは入力されたデータを計算を加えて次層へ出力する。四角いノードは常に定数を出力するノードであり、A 層の四角いノードはどちらも常に 1 を出力する。B 層には線に沿って A 層から入力される。Σ ノードは足し合わせを意味し、この場合入力 e_1, e_2 に対して定数バイアスの重み行列 w_c が足し合わされる。次に C 層には、B 層からの入力に対して重み行列 w_g との積が入る。またこの f のノードは関数を表しており、入力データに対しグモイド関数を使用する。D 層で C 層からの一部入力を合成する。以上の A 層から D 層までの計算処理を以下にまとめる。

$$L1 : \mu_{L1}(e_1(k)) = \frac{1}{1 + \exp(w_{g1}(e_1(k) - w_{c1}))} \quad (7)$$

$$C1 : \mu_{C1}(e_1(k)) = \frac{1}{1 + \exp(w_{g2}(e_1(k) - w_{c2}))} - \frac{1}{1 + \exp(w_{g3}(e_1(k) - w_{c3}))} \quad (8)$$

$$R1 : \mu_{R1}(e_1(k)) = \frac{1}{1 + \exp(w_{g4}(e_1(k) - w_{c4}))} \quad (9)$$

入力 e_2 における L2, C2, R2 に関しても式 (7), 式 (8), 式 (9) と同様である。 $\mu_n(x)$ はそれぞれ入力に対する前件部ファジィ適合度の関数である。 [] のノードは掛け合わせを表しており、E 層では対応する適合度同士で以下の様な積をとる。

$$\mu_1 = L1 \cdot L2, \mu_2 = L1 \cdot C2, \mu_3 = L1 \cdot R2, \quad (10)$$

$$\mu_4 = C1 \cdot L2, \mu_5 = C1 \cdot C2, \mu_6 = C1 \cdot R2, \quad (11)$$

$$\mu_7 = R1 \cdot L2, \mu_8 = R1 \cdot C2, \mu_9 = R1 \cdot R2, \quad (12)$$

$$\mu_{all} = [\mu_1, \mu_2, \dots, \mu_9]^T. \quad (13)$$

F 層の N のノードは正規化処理を行う。E 層で作成された最終適合度 μ_{all} に対して以下の計算を行う。

$$\mu_{allt} = \frac{1}{\mu_1 + \mu_2 + \dots + \mu_9} \cdot \mu_{all}. \quad (14)$$

最後に G 層で後件部シングルトン w_s との内積を取ることによって最終的な制御出力 u を得る。

$$u = w_s \cdot \mu_{allt}, \quad w_s = [w_{s1}, w_{s2}, \dots, w_{s9}]. \quad (15)$$

3.2 バックプロパゲーション

パラメータ w_c, w_g そして w_s に対してバックプロパゲーションを用いる事でファジィニューラルネットワークコントローラの学習を行う。改めて一部パラメータを以下のように定義する。

$$w_c = [w_{c1}, w_{c2}, \dots, w_{c8}]^T \quad (16)$$

$$w_g = [w_{g1}, w_{g2}, \dots, w_{g8}]^T \quad (17)$$

バックプロパゲーションには評価関数が必要である。以下に本研究で仕様する評価関数を示す [1]。

$$J(k) = J_1(k) + J_2(k) \quad (18)$$

$$J_1(k) = \frac{1}{2}e_1(k)^2, \quad J_2(k) = \frac{1}{2}e_2(k)^2 \quad (19)$$

この評価関数に対して以下の様な偏微分が考えられる。

$$\frac{\partial J(k)}{\partial w_i(k-1)} = \frac{\partial J_1(k)}{\partial w_i(k-1)} + \frac{\partial J_2(k)}{\partial w_i(k-1)}, \quad i = c, g, s \quad (20)$$

また、連鎖率によって式 (20) は以下となる。

$$\frac{\partial J_1(k)}{\partial w_i(k-1)} = \frac{\partial J_1(k)}{\partial e_1(k)} \frac{\partial e_1(k)}{\partial u(k-1)} \frac{\partial u(k-1)}{\partial w_i(k-1)} \quad (21)$$

$$\frac{\partial J_2(k)}{\partial w_i(k-1)} = \frac{\partial J_2(k)}{\partial e_2(k)} \frac{\partial e_2(k)}{\partial u(k-1)} \frac{\partial u(k-1)}{\partial w_i(k-1)} \quad (22)$$

この時式 (21) と式 (22) の右辺内部の $\frac{\partial e_1(k)}{\partial u(k-1)}$ の項は該当する式が存在せず厳密には微分できない。そこで今回は以下の様に近似する [1]。

$$\frac{\partial e_j(k)}{\partial u(k-1)} \rightarrow \text{sgn} \left(\frac{e_j(k) - e_j(k-1)}{u(k-1) - u(k-2)} \right), \quad j = 1, 2 \quad (23)$$

以上からパラメータの学習は、バックプロパゲーションを用いて次の式で表せる。

$$w_i(k) = w_i(k-1) - \eta_i \frac{\partial J(k)}{\partial w_i(k-1)} \quad (24)$$

η_i は学習率である。

4 ゼロバイアスを考慮したファジィニューラルネットワーク

従来のファジィニューラルネットワークは、ファジィ演算に重要なパラメータの自動調整が可能であり、非線形な制御対象や状況に対してもチューニングできる事が分かっている [7]。しかしながらこの制御器はゼロ入力においてバイアスのある出力を学習する事がある。この問題を解決するために、まず制御器の学習対称化を試みた [8]。これによって制御器の非対称問題と無駄出力問題の解決を図った。しかしこの手法は非対称な制御対象や軌道に対しフィットするチューニングができない。そこで本研究では、0 入力時に 0 を出力し、かつ非対称な制御対象にも対応可能なコントローラの作成を試みる。そのために後件部シングルトンの学習を一部制限し、前件部のメンバーシップ関数を他の関数へ変更する。

4.1 前件部学習の修正

従来のファジィニューラルネットワークは学習の結果、前件部適合度を求めるファジィ関数が中央 0 入力の時点で適合度にズレを生じさせていた。これはシグモイド関数を用いる以上必ず発生してしまう問題である。この関数は構造的にどの様な入力でも出力が 0 にならない。これは非対称的な制御対象の場合問題にならない。しかし位置制御系の制御対象に対して問題となる。そこでこのズレの発生を抑制するために、前件部ファジィ関数を次式に置き換えた。

$$L1 : \mu_{L1}(x) = \begin{cases} 1 & (x \leq -\frac{\pi}{2a_{l1}}) \\ \sin^2(a_{l1}x) & (-\frac{\pi}{2a_{l1}} < x < 0) \\ 0 & (0 \leq x) \end{cases} \quad (25)$$

$$R1 : \mu_{R1}(x) = \begin{cases} 0 & (x \leq 0) \\ \sin^2(a_{r1}x) & (0 < x < \frac{\pi}{2a_{r1}}) \\ 1 & (\frac{\pi}{2a_{r1}} \leq x) \end{cases} \quad (26)$$

$$C1 : \mu_{C1}(x) = cl1(x) + cr1(x) \quad (27)$$

$$cl1(x) = \begin{cases} 0 & (x \leq -\frac{\pi}{2a_{cl1}}) \\ \cos^2(a_{cl1}x) & (-\frac{\pi}{2a_{cl1}} < x < 0) \\ 1 & (0 \leq x) \end{cases} \quad (28)$$

$$cr1(x) = \begin{cases} 0 & (x \leq 0) \\ -\sin^2(a_{cr1}x) & (0 < x < \frac{\pi}{2a_{cr1}}) \\ -1 & (\frac{\pi}{2a_{cr1}} \leq x) \end{cases} \quad (29)$$

従来シグモイド関数で行っていたファジィ演算を三角関数で代用した。この時、 a_{l1} は学習してメンバーシップ関数を変化させるパラメータである。このパラメータが変動する事で図3の様に0入力時の不要な適合度の学習を解消した。L2, C2 および R2 も同様である。以上から前

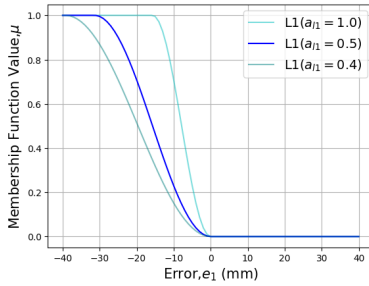


図3 学習される $\mu_{L1}(x)$ の例

件部の問題は解消された。

4.2 後件部学習の修正

前節で適合度の修正ができた。しかし、最終的な出力は後件部シングルトン w_s で決まる。故に、最終的に出力の調整を行う後件部も調整する必要がある。前節の修正の結果、入力 $e_1 = 0, e_2 = 0$ の時、前件部最終適合度は μ_5 のみが1となり他の適合度は0となる。ゆえにこれに対応する後件部定数 w_{s5} が0となるように修正する必要がある。そこで後件部シングルトンを以下の様に修正する。

$$w_s = [w_{s1}, w_{s2}, w_{s3}, w_{s4}, 0, w_{s6}, w_{s7}, w_{s8}, w_{s9}]. \quad (30)$$

その上で式(30)の中央0の値は学習しない事とする。これにより0バイアスの学習を防ぎ、非対称な制御対象への適応力の維持を図る。

5 シミュレーション

従来のファジィニューラルネットワークコントローラ(FNNC)と0バイアスに考慮したファジィニューラルネットワークコントローラ(Proposed FNNC)の制御性能を

比較するために、両コントローラを用いてライントレースのシミュレーションを行った。

図4にシミュレーションに用いたコースマップを示す。学習率 η は全て 0.01 とし、バイアス電圧 u_0 は 3.0V と

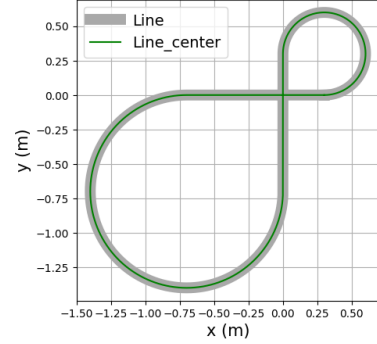


図4 コースマップ

した。右モータが左モータより速くなるように物理特性 $K_n, (n = 1, 2, 3, 4)$ を調整した。シミュレーション時間は 3000 秒とし、シミュレーションの際の刻み幅は 0.01 秒とした。これらの条件を元にシミュレーションを行った。図5は車体中心とコースライン上の中心との誤差をプロットしたものである。図5から、どちらも最終的に

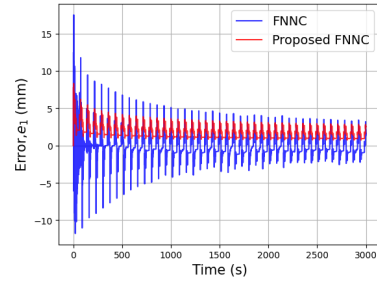


図5 車体とライン中心との誤差

誤差が 5mm の範囲に収まっていることが分かる。また、従来法は 0 を中心として振動しているが、提案法は誤差が正に偏っている。また誤差の収束速度において提案した FNNC の法が従来の FNNC より速い事が分かる。次にシミュレーションステップ毎の評価値のグラフの比較を図6に示す。評価値の減少率からも提案した制御器の

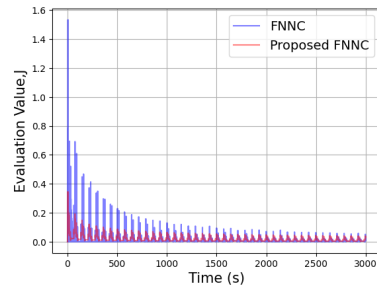


図6 シミュレーションステップ毎の評価値の比較

方が制御性能に優れている様に見受けられる。次に図7

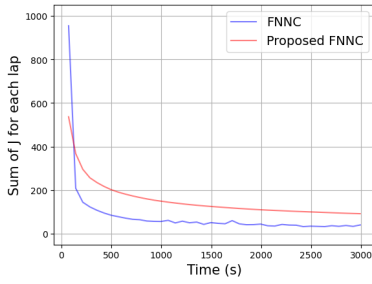


図 7 1 周毎の評価値の合計の比較

に 1 周毎に評価値を合計したグラフを示す。図 7 から総合的なコントロールパフォーマンスでは従来の FNNC が優れている事が分かる。つまり、提案した FNNC は従来の FNNC に比べ初期の乱雑さが少なく収束が速いが、最終的なパフォーマンスでは従来の FNNC が優れていると考えられる。提案した FNNC は従来の FNNC と比べメンバーシップ関数の自由度が減少しているため、制御性能の劣化は予測されていた。しかし、図 5 から提案法において、ある程度知覚できる誤差には大きな差が見られない。故に提案した FNNC の制御性能の劣化は、従来の FNNC と比べある程度抑えられていると考えられる。次に学習された両コントローラにおいて入力と出力の関係を表したグラフを図 8、図 9 に示す。図 8 は e_2 を自由にし、 e_1 を 0 で固定した時の出力の拡大図であり、図 9 は e_1 を 0 で固定し、 e_2 を自由にした時の出力の拡大図である。どちらの図も従来の FNNC は入力 0 の原点からズ

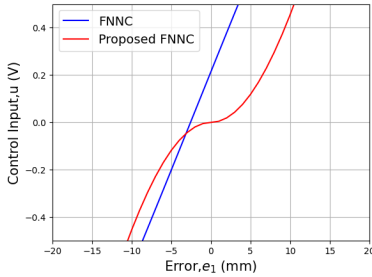


図 8 $e_2(k) \equiv 0$ にした入出力関係

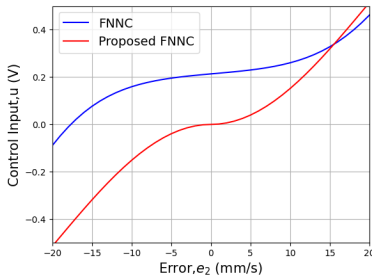


図 9 $e_1(k) \equiv 0$ にした入出力関係

れており、提案した FNNC は原点を通っていることが見て取れる。つまり、提案した FNNC はバイアスの学習を抑制できたと考えられ、入力 0 の時正しく出力しており、コントローラの目的を達成している。

6 おわりに

本研究では 0 バイアスに考慮したファジィニューラルネットワークコントローラの構築を行った。そのために従来のシグモイド関数を使ったメンバーシップ関数を三角関数で置き換えた。また後件部の学習仕様を一部制限した。この修正した制御器に対してライントレースのシミュレーションを行い従来の FNNC との制御性能を比較した。その結果、提案法を従来法と比較した際の性能劣化が少ないことが分かった。また、初期の収束速度や 0 入力地点でのバイアスの学習を抑える事に成功した。

参考文献

- [1] M. Asai, G. Chen, and I. Takami, “Neural network trajectory tracking of tracked mobile robot,” Proceedings of 2019 16th International Multi-Conference on Systems, Signals & Devices (SSD’19), pp. 225–230, 2019.
- [2] 星野孝総・瀧本浩志: “ライントレースカーの二段ファジィ制御による速度制御と差分進化アルゴリズムによる加速制御器の最適化”, 知能と情報 (日本知能情報ファジィ学会誌), vol.25, no.3, pp.760-771, 2013.
- [3] A. Genno and W. Wang, “An Adaptive Neuro-Fuzzy Controller for Vibration Suppression of Flexible Structures,” IEEE/ASME Transactions on Mechatronics, pp. 1–10, 2023.
- [4] H. F. Asres and A. T. Kassie, “Design of Neuro Fuzzy Sliding Mode Controller for Active Magnetic Bearing Control System,” IEEE Access, Vol. 11, pp. 103189–103200, 2023.
- [5] 山川聡子: “移動ロボットを用いて制御理論を学ぶ実験カリキュラム”, 計測自動制御学会論文集, Vol.55, No.7, pp.457-465, 2019.
- [6] 堀川慎一・古橋武・内川嘉樹: “ファジィニューラルネットワークの構成法と学習法 (<特集論文>ファジィ・ニューラルネットワークとその応用)”, 日本ファジィ学会誌, Vol.4, No.5, pp.906-928, 1992.
- [7] 吉田裕基・陳幹: “ファジィニューラルネットワークを用いたライントレース”, 令和 4 年度 電気・電子・情報関係学会東海支部連合大会要旨集, 2022.
- [8] 吉田裕基・陳幹: “対称性を考慮したファジィニューラルネットワークを用いたライントレース”, 日本機械学会東海支部第 72 基総会・講演会予稿集, 2022.