Vol.2018-ARC-230 No.25
Vol.2018-SLDM-183 No.25
Vol.2018-EMB-47 No.25
2018/3/8

ニューラルネットによるモデル予測制御高速化

竹松 慎弥[1,a)]　嶋岡 雅浩[2]　道木 慎二[2]　枝廣 正人[1]

概要：大規模・複雑化が進む制御システムのリアルタイム性保証のために，制御アルゴリズム最適化・並列化手法が盛んに研究されている．特にモデル予測制御は不規則に状態が変化する対象にも最適な制御を実現できるが，計算量が多いため，高速化が求められている．そこで，本研究ではモデル予測制御の入出力をニューラルネットを用いて学習させることで，高速化する方法を提案する．永久磁石同期モータ電流制御系モデルを用いた実験の結果，同等の制御結果が得られ，処理時間は約 3700 倍高速になることを確認した．

Shinya Takematsu[1,a)]　Masahiro Shimaoka[2]　Shinji Doki[2]　Masato Edahiro[1]

1. はじめに

ハイエンド制御においては動的かつ不規則に状態が変化するような対象を高精度に制御することが求められる．このような制御を実現する手法のひとつとしてモデル予測制御が知られている．モデル予測制御は数理モデル等から対象の挙動を予測することで常に最適な制御を実現する手法である．様々な分野への適用が期待されている一方で，膨大な計算量を必要とし，短制御周期システムには向いていないとされている．

筆者らはこれまでにモデル予測制御の高速化や並列化手法を適用してきたが，未だ短周期システム適用レベルには程遠い処理時間を要する [1][2]．さらなる高速化にはオフライン時（運転前）に計算しておいた結果をオンライン時（運転中）に使うことによって，オンラインの計算時間を削減するという方法などが考えられる．

そこで，本研究ではニューラルネットワークによってモデル予測制御計算を模倣し，高速化する手法を提案する．ニューラルネットワークは人間の脳構造を模した数学モデルであり，任意の関数を回帰できるとされている．大量の学習データを用いる必要があるため膨大な学習時間がかかるが，学習はオフラインで行う作業である．一方，1 回の出力計算は短時間で実行できるため，オンラインの処理時間は非常に短い．よって，モデル予測制御の入出力関係をニューラルネットワークで回帰し，置き換えることで，制御性能を保ちながら処理時間を短縮する．

また提案手法は，システム稼働中に並行して学習することも可能であるため，制御プラントの経年変化への対応も可能とする．

2. モデル予測制御を用いた永久磁石同期モータ電流制御系

本研究では，永久磁石同期モータ電流制御系にモデル予測制御を適用したシステムの高速化を考える．永久磁石同期モータ制御は一般的に μs オーダー程度の短い制御周期が設定されるため，本研究の対象システムとして選択する．

本章では，モデル予測制御手法および永久磁石同期モータ電流制御系への適用について説明する．

2.1 モデル予測制御

基本的に制御システム設計開発は制御対象の挙動を数理モデル等で表現するところから始まる．うまくモデル化できる場合，ある操作を加えたときの制御対象がどのように動き，システムに対してどのような出力を返すのかを予測できる．これを利用して全ての操作候補に対する挙動を予測し，最も良い挙動を示す操作を実際に使うことで常に最適な制御を実現する手法がモデル予測制御（MPC:Model Predictive Control）である．しかし，操作候補数や予測するステップ（制御周期）数によっては考えなければならない状態数が膨大になり，長い計算時間を要してしまうという問題がある．そのため，分枝限定法 [3] 適応などの様々な高速化手法が提案されている．

[1] 名古屋大学大学院情報科学研究科
[2] 名古屋大学大学院工学研究科
[a)] ts1413@ertl.jp

Vol.2018-ARC-230 No.25
Vol.2018-SLDM-183 No.25
Vol.2018-EMB-47 No.25
2018/3/8

スイッチング状態			電圧
u 相	v 相	w 相	ベクトル
OFF	OFF	OFF	V_0
ON	OFF	OFF	V_1
ON	ON	OFF	V_2
OFF	ON	OFF	V_3
OFF	ON	ON	V_4
OFF	OFF	ON	V_5
ON	OFF	ON	V_6
ON	ON	ON	V_7

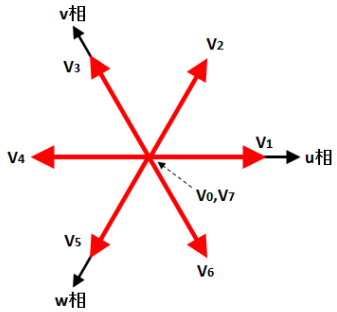

図 1　3 相 PWM による生成電圧ベクトル

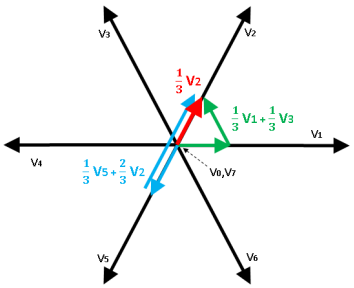

図 2　1 つの指令ベクトルに対するスイッチングパターン例

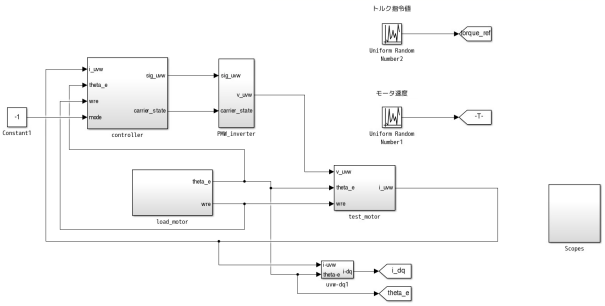

図 3　永久磁石同期モータ電流制御系モデル

2.2　永久磁石同期モータ電流制御系

永久磁石同期モータは回転子に永久磁石を用いたモータである．回転子の磁界と固定子の磁界の引力および斥力を用いて回転させる．一般的に固定子には三相コイル（各相は U 相，V 相，W 相と呼ばれる）を用い，電磁誘導により磁界を生成する．

しかし，三相電流は交流量で各相が 120° の位相差を持ち，かつ時間変化するため扱いにくい．そこで，回転子と同期して回転する d-q 直行二軸回転座標系を取り入れて，時間変化しない直流量として d-q 電流を定義する [4][5]．すると，永久磁石同期モータの発生トルク τ は式 (1) のように数理化できる．

$$\tau = P_n K_E i_q + (L_d - L_q) i_d i_q \qquad (1)$$

ここで，P_n，K_E，L_d，L_q は定数であり，i_d および i_q が電流 i_{dq} の各成分である．この式より，所望のトルクを得るための電流ベクトル i_{dq} を算出できる．

電流ベクトル指令 i_{dq} を生成するために三相 PWM インバータを用いる．PWM（Pulse Width Modulation:パルス幅変調）は直流電源の ON/OFF（比率をデューティ比と呼ぶ）を変更することによって任意の平均電圧を発生させる方式である．瞬間的に見れば，各相それぞれに ON/OFF の状態があるため，図 1 のような 8 パターンの電圧ベクトルが生成できる．各相の ON/OFF を切り替えることでベクトルを組合せ，任意の三相電圧/電流ベクトル生成，ひいてはトルク制御を可能にする．

しかし，ある指令ベクトルを生成するためのベクトルの組合せとしては複数のパターンが考えられる．例えば，1/3 周期分の V_2 を生成するとき，図 2 に示すような複数パターンで実現できる．瞬間的に見れば各パターンによるトルクの挙動はそれぞれ異なり，最適なパターンを選ぶことで制御性能を向上することもできる．そこで，モデル予測制御を用いて各パターンによるトルク挙動を予測し，最適なベクトルの組合せ（各相の PWM スイッチングタイミング）を算出する．

2.3　Simulink モデル

以上の制御手法を MATLAB/Simulink[6] でモデル化したものを本研究で使用する．そのモデルを図 3 に示す．controller サブシステムが制御器であり，計算された各相のデューティ比（sig_uvw）をもとに PWM_inverter で三相電流を生成する．三相電流がターゲットとなる test_motor に印加され，トルクを制御する．ただし load_motor によって test_motor の回転速度は常に指令値通りになっているとしている．また制御周期としては 20μs に設定されている．

controller サブシステムの内部を図 4 に示す．制御器本体が controller という名の S-function サブシステムであり，C コードで記述されている．制御器ではまず，座標変換，制御遅れ対策処理によって現在の制御周期終了時点での状態を計算する．その状態からモデル予測制御計算を行い，各相スイッチングタイミングを求め，さらにはデューティ比を算出する．また，モデル予測制御計算には様々なモードが用意されており，状況によって操作候補数や予測ステップ数を変化させることができる．スイッチングタイミングは，制御周期をさらに細かく分割した周期を作り，何番目の周期で ON/OFF を切り替えるかという 0〜分割数の整数値になる．

Vol.2018-ARC-230 No.25
Vol.2018-SLDM-183 No.25
Vol.2018-EMB-47 No.25
2018/3/8

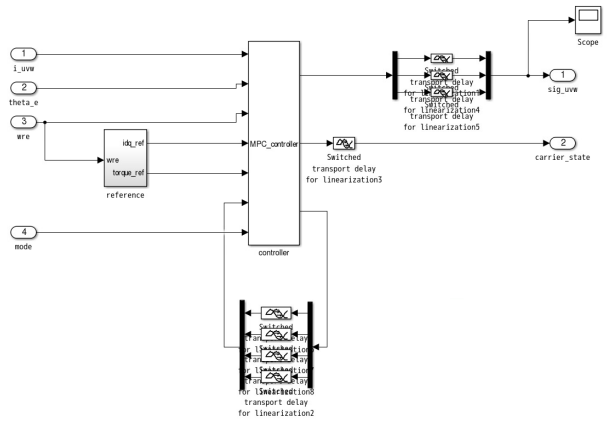

図 4　永久磁石同期モータ電流制御系の制御器

筆者らはこれまでにこのモデルに対して並列分枝限定探索法を適用することによって高速化図った [1][2]．高並列化が実現できた一方，12 並列でも 1 ステップの処理に数百 ms〜数 s 程度の時間を要し，目指す μs オーダーの制御周期には到底間に合わないという結果となった．

3. 提案手法

本提案では，モデル予測制御計算をもっと簡単な計算で模倣することを考える．つまり，同じ入力を与えるとモデル予測制御と同じ出力を返す関数がないかを探す，いわゆる「回帰問題」を考えるということである．

回帰問題には様々な解法があるが，近年急速に注目を浴びるようになったのがニューラルネットワークを用いる手法である．ニューラルネットワークとは人間の脳構造を数学モデル化したものである．脳の基本構成となるニューロンが図 5 のように数式モデル化され（形式ニューロン [7]），形式ニューロン同士をつなぎ合わせることで人工ニューラルネットワークを形成する．特に，並列に並んだ多数のニューロンで 1 つの層を構成し，層と層を繋げていく階層型ニューラルネットワークが広く用いられ，このネットワークは任意の関数を表現できるとされている．また，誤差逆伝播法 [8] という重み更新アルゴリズムを用いることで入出力関係を学習できる．そのため，非線形多入力多出力のような難しい回帰問題に適用され，高い効果が報告されている．ただし，高精度の回帰を実現するためには大量の学習データと学習時間が必要である．一方，1 つの入力に対する出力計算は非常に速く，自動運転などのリアルタイム性が必要なシステムへの応用も検討されている．

そこで，本研究ではモデル予測制御の入出力関係をニューラルネットワークを用いて回帰し，置き換えることで，同様の制御性能を保ちながら高速化する手法を提案する．ニューラルネットワークを用いることで次のようなメリットがある．今回のモデル予測制御は多入力多出力であり，入出力関係を人手で解析するのは非常に困難である．これをニューラルネットワークであれば自動で解析できる．また，学習に必要な大量のデータは Simulink モデルをシミュレーションさせることで取得できる．さらに，入力と重みのベクトル積が多用されるため，GPU による並列化やベクトル演算が有効であり，高並列性も期待できる．

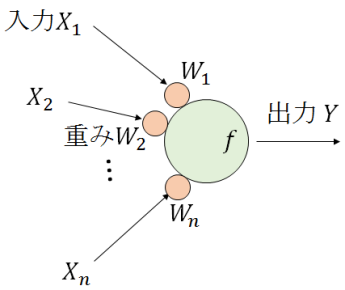

図 5　形式ニューロン

一方で，入出力を完全に再現できず，出力に少なからず誤差を含んでしまうことが予想される．常に最適な操作を出力するモデル予測制御に対して，この誤差が制御性能へもたらす影響については十分に評価する必要がある．

3.1 ネットワーク構造

本研究では，回帰問題に応用され，高い効果が報告されている階層型ニューラルネットワークを用いる．階層型ニューラルネットワークの最初の層は入力層，最後の層は出力層，その他の層は中間層（または隠れ層）と呼ばれる．中間層の層数や各層のニューロン数は回帰精度に大きく影響するため，ネットワーク構成は非常に重要な問題である．よって，本節ではネットワーク構造の決定法について提案する．

入力層および出力層のニューロン数は入出力データ次元数と同じになる．図 4 のモデルにおいて制御器への入力は 13 次元，出力は 8 次元である．しかし，入出力数が多いほど，ネットワークは複雑になり，処理量が増える．そこで，入出力数が最小限になるように以下の変更を行う．

- 制御計算の分割
  2.3 節で述べたとおり，制御器にはモデル予測制御計算以外の前処理や後処理も含まれている．そのため，制御器全体をニューラルネットワーク化するとモデル予測制御には直接関係の無いパラメータが入出力に含まれてしまう．よって，1 つの S-Function ブロックで構成されている制御器部を前処理部，モデル予測制御部，後処理部に分割し，最終的にはモデル予測制御部のみをニューラルネットワーク化する．これによって，ニューラルネットワークの入出力次元数を減らす．
- モード別のネットワーク構築
  モードによってモデル予測制御の探索木構造が大きく変化するため，入出力関係も大きく変わることが予想される．そのため，全てのモードを 1 つのネットワークで実現するとネットワークが大幅に複雑化する

Vol.2018-ARC-230 No.25
Vol.2018-SLDM-183 No.25
Vol.2018-EMB-47 No.25
2018/3/8

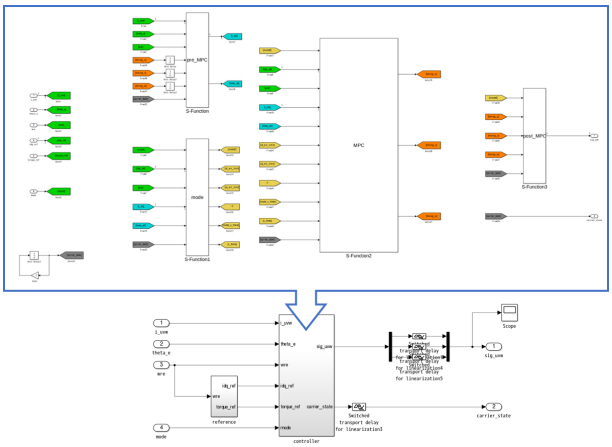

図 6　入出力最適化後の制御器

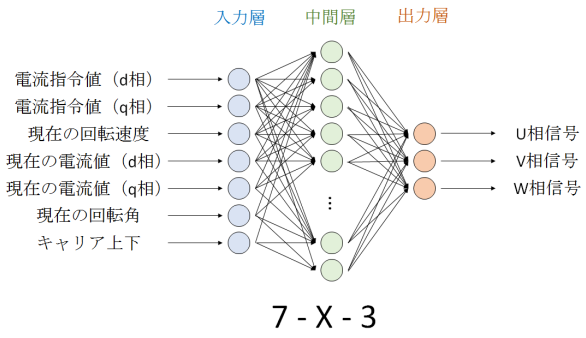

図 7　ネットワーク構造

ことが予想される．よって，モード別にネットワークを構築し，モデル予測制御部へ入力しなければならなかったモード固定パラメータを削減する．

以上の変更を適用すると制御部は図 6 のようになる．pre_MPC が前処理部，MPC がモデル予測制御部，post_MPC が後処理部である．この変更の結果，モデル予測制御部への入力は 7 次元，出力は 3 次元となった．

入出力次元数が決定したため，ニューラルネットワークの入力層および出力層のニューロン数はそれぞれ 7 および 3 と決まる．また，3 層のネットワーク（中間層が 1 層）はあらゆる連続な非線形関数を回帰できることが証明されている [9]．しかし，本システムにおいてモデル予測制御出力となるスイッチングタイミングは必ず整数値であるため，離散である．ただし，不規則な飛び値ではないため連続であるとみなしても十分な精度が得られると考えられ，中間層は 1 層とする．中間層のニューロン数は少ない試行回数で最適値を算出するベイズ最適化手法 [10] を用い，試行錯誤で精度が最大となるニューロン数を決定する．以上から作成するネットワーク構成は図 7 のようになる．

3.2　学習

ネットワークの構築および学習には MATLAB の Neural Network Toolbox[11] を用いる．これを用いる理由としては，Simulink モデルと連携が取れる，回帰用の 3 層のネットワーク（fitnet）が簡単に構築できる，ベイズ最適化の関数が用意されている，学習したネットワークの Simulink モデルを生成できるためである．

表 1　ニューラルネットワーク学習設定

項目	設定内容
ネットワーク構造	7-X-3
データ分割割合	学習用：70% 検証用：15% テスト用：15%
学習アルゴリズム	レーベンバーグ・マルカート法
ネットワーク評価	平均二乗誤差
検証エラー最大回数	6
ベイズ最適化	・対象：中間層のニューロン数 ・範囲：1〜150 ・獲得関数：単位時間あたりの改善期待量 ・目的関数：ネットワーク評価値 ・試行回数：40

今回は操作候補数 21^3，予測ステップ数 2 でモデル予測制御を行うモードの入出力関係を学習させる．このモードでは各相の制御出力は 0〜20 の整数となる．

学習データはモデルをシミュレーションし，モデル予測制御部の入出力を取得することで集める．この時，トルク指令値および速度指令値を長い間隔で変化させる設定と短い間隔で変化させる設定の 2 パターンでシミュレートさせる．これによって，定常状態が多いデータと過渡状態の多いデータを収集できる．集めた 2 パターンのデータはランダムに混ぜ，10 万の入出力組を学習データとして用いる．

学習はベイズ最適化で最適ニューロン数を探しながら行う．主な学習設定は表 1 の通りである．10 万の学習データは 70%をパラメータ更新，15%を収束判定，15%を未知データに対する評価計算のために使用する設定としている．検証データにおいて 6 回連続で評価値（平均二乗誤差）が改善されない場合は学習終了とする．これによって，過学習を防ぐことができる．ベイズ最適化では中間層のニューロン数を 1〜150 の範囲で 40 回試行し，最適値を算出する．Neural Netowork Toolbox の仕様上，ネットワーク出力を整数に限定したり，上限・下限値の設定することはできないが，出力の制限を設けずに学習させ，ネットワーク使用時に出力に飽和や丸めを適用することで対応する．

ベイズ最適化による試行によって得られたニューロン数とネットワーク評価値（平均二乗誤差）の関係を図 8 に示す．ニューロン数を増やすほどネットワークの表現力が高まり，精度が向上している．ニューロン数 130 あたりから精度向上が飽和しており，試行した中で最も精度が高くなったのはニューロン数が 135 の場合であった．また，この時の学習データとの平均二乗誤差は 3.58 であった．よって，0〜20 の値を取る出力に対して，2 未満の誤差に抑えられていると考えられる．

Vol.2018-ARC-230 No.25
Vol.2018-SLDM-183 No.25
Vol.2018-EMB-47 No.25
2018/3/8

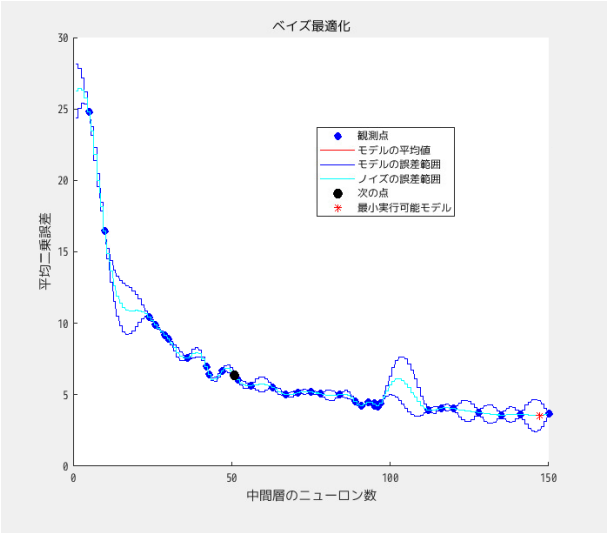

図 8　ベイズ最適化結果

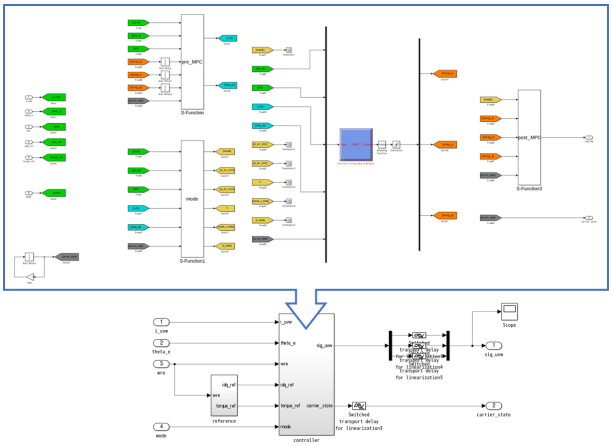

図 9　ニューラルネットワークで置き換えた制御器

上記の考察より，ニューラルネットワークの制御性能や実行時間評価には，ニューロン数 135 の場合の学習結果を用いる．学習したネットワークから Simulink モデルを自動生成し，図 6 のモデル予測制御部を置き換えるたものが図 9 である．入出力はベクトルとなるため，Mux および Demux を用いてベクトル化，ベクトル要素の取り出しを行っている．また，出力は Round ブロックと Saturation ブロックを介すことで，整数丸めおよび範囲制限を適用している．

4. 評価

4.1 評価方法

図 6 のモデル（以下，MPC モデルと呼ぶ）と図 9 のモデル（以下，NN モデルと呼ぶ）に対し，制御性能と処理時間について評価を行う．

まず，制御性能評価では制御出力比較と制御対象パラメータ（トルク）の挙動比較を行う．学習データ収集時とは異なるようにトルクおよび速度指令値を変化させ，MPC モデルをシミュレーションする．この時，モデル予測制御計算部への入出力データを取得しておき，未知データとする．この未知データの入力をニューラルネットワークへ与え，未知データの出力（モデル予測制御計算の出力）と比較する．また，NN モデルを同じ指令値設定でシミュレーションする．MPC モデルと NN モデルのトルクシミュレーション結果を比較することで，制御器をニューラルネットに置き換えることによる制御性能差を確認する．

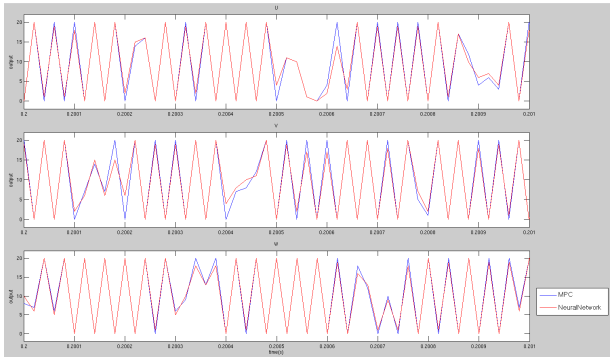

図 10　制御出力

次に，実行時間評価としては MPC モデルと NN モデルにおけるシミュレーション時間を比較し，高速化率を計算する．誤差を小さくするため，十分大きなステップ数実行した時のシミュレーション時間を測定し，ステップ数で割ることで 1 ステップ当たりの実行時間を求める．ただし，モデル予測制御部（またはニューラルネットワーク部）を取り外したモデルを同じシミュレーション設定で動かしたときの時間を測り，先の実行時間から引くことで，制御器のみにかかった時間だけを考える．

なお，評価は以下の計算機環境で行う．

- OS：CentOS 7.4.1708
- CPU：IntelXeon E5-2695 v2 2.40GHz
- メモリ：32GB
- キャッシュ：30MB
- MATLAB：R2017a

4.2 制御性能評価

まず，モデル予測制御計算とニューラルネットワークに同じ入力を与えた場合のある時間範囲における 3 相スイッチングタイミング出力結果を図 10 に示す．各相に対する出力とも非常に良く回帰できていることが確認できた．次に，MPC モデルと NN モデルのトルク制御結果を図 11 に示す．また，指令値との誤差を図 12 に示す．図 11，12 どちらも大差ない結果が示され，ニューラルネットワークに置き換えても同程度の制御性能を実現できていることが確認できた．

4.3 実行時間評価

MPC モデルを逐次実行した時，12 コア並列実行した時 [1][2]，および提案手法（NN モデル）を実行した時の各

Vol.2018-ARC-230 No.25
Vol.2018-SLDM-183 No.25
Vol.2018-EMB-47 No.25
2018/3/8

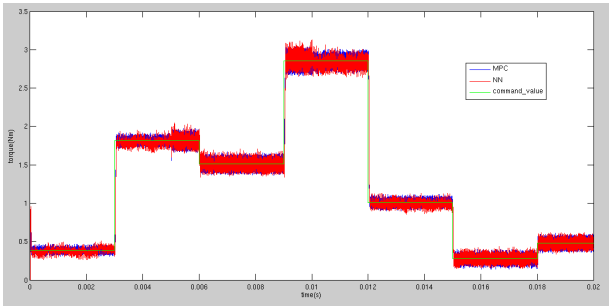

図 11　トルク制御結果

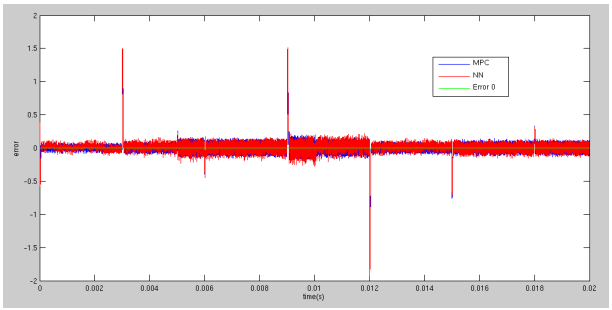

図 12　トルク指令値との誤差

表 2　ニューラルネットワーク実行時間評価結果

	実行時間/ステップ	高速化率
MPC 逐次実行	2.96s	1.0
MPC 12 コア並列	300.92ms	9.84
NN 逐次実行	799.3μs	3705.0

処理時間や高速化率を表 2 にまとめる．実行時間はモデル予測制御逐次実行と比べて 3705 倍高速化し，1 ステップ当たりの実行時間が μs オーダーまで短くなった．

4.4　考察

モデル予測制御を学習させたニューラルネットワークで置き換えることで，高い制御性能を維持したまま μs オーダーまで高速化を達成することができた．よって，ms オーダーの制御周期のシステムに対してはこの手法を用いることでリアルタイムな疑似モデル予測制御が可能であると考えられる．

一方，本モデルの制御周期設定は 20μs であり，これを達成するためにはさらに 40 倍の高速化が必要である．ニューラルネットワークは理論上 1 つの層のニューロンをそれぞれ 1 つずつコアに割り当てて並列計算させることができる．つまり，今回のニューラルネットワークの場合，最大で 135 並列できるということである．よって，理想的には 135 倍高速化できるため，20μs 制御周期を達成できる．ただし，個々のニューロンの処理はとても軽く，データ通信オーバヘッド等が速度向上を阻害する可能性が高い．このような簡単な処理の高並列計算には GPU を用いることが有効である．今後，GPU を用いた並列化による評価が必要であろう．

5.　おわりに

本研究では，モデル予測制御の高速化手法としてニューラルネットワークで制御計算を置き換える手法を提案した．永久磁石同期モータ電流制御系に適用したモデル予測制御のシミュレーション結果を学習データとし，3 層のニューラルネットワークで学習させた．その結果，高精度の回帰に成功し，制御器をニューラルネットワークで置き換えることで制御性能を維持したまま，逐次実行時間比 3705 倍の高速化を達成した．これによって，ms オーダーの制御周期を持つシステムへの適用可能性を示すことができた．

また今後の方針としては，GPU 等による高並列化も期待できるため，さらなる高速化によって μs オーダーレベルの短周期システムへの適用可能性を示す．加えて，稼働中の学習により経年変化への対応も考える．

参考文献

[1] 竹松慎弥他: 永久磁石同期モータ電流制御系のための予測制御アルゴリズム並列化, 2016-EMB-41, No.4, 2016.
[2] 竹松慎弥他: 永久磁石同期モータ電流制御系のための予測制御アルゴリズム並列化, 第 79 回全国大会講演論文集, Vol.3A-03, pp.29-30, 2017.
[3] 大西克実他: 並列分枝限定法における分枝変数の選択に関する考察, 電子情報通信学会論文誌, Vol.J84-D-I, No.9, pp.1318-1326, 2001.
[4] 大沼巧: 新しい座標系を用いた埋込磁石同期モータの位置センサレス制御に関する研究, 名古屋大学, 博士論文, 2011.
[5] 松本純: 新しい数学モデルを用いた永久磁石同期モータの位置センサレス制御系のロバスト化に関する研究, 名古屋大学, 博士論文, 2014.
[6] MathWorks,Simulink, https://jp.mathworks.com/products/simulink.html, 2018.
[7] Warren S. McCulloch et al: A logical calculus of the ideas immanent in nervous activity, The bulletin of mathematical biophysics, Vol.5, No.4, pp.115-133, 1943.
[8] David E. Rumelhart et al: Learning representations by back-propagating errors, Nature, Vol.323, pp.533-536, 1986.
[9] K.Funahashi: On the Approximate Realization of Continuous Mappings by Neural Networks, Neural Netw, Vol.2, No.3, pp.183-192, 1989.
[10] Jasper Snoek et al: Practical Bayesian Optimization of Machine Learning Algorithms, NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems, Vol.2, pp.2951-2959, 2012.
[11] MathWorks,Neural Network Toolbox, https://jp.mathworks.com/products/neural-network.html, 2018.