

深層学習を活用した高精度知能化制御の提案 Proposal of a high-performance intelligent control with deep learning

広津 鉄平[†] 堀口 辰也[†] 中村 敏明[†] 田向 権[‡]
Tepei Hirotsu Tatsuya Horiguchi Toshiaki Nakamra Hakaru Tamukoh

1. はじめに

組み込みハードウェアの性能向上により、深層学習 (DL: Deep Learning) の応用範囲が拡大、特に画像認識分野での性能向上が進んでいるが、他の応用での研究は少ない状況である。本発表では、リアルタイム制御分野において、新しい DL の応用を提案する。本研究は古典制御と比較し制御性能を飛躍的に向上可能であるモデル予測制御を用いる。本手法では、プラント予測モデルを DNN(Deep Neural Network)で高精度化し、さらに、大域的な解を得られる ABC(Artificial Bee Colony)アルゴリズムにより最適操作量を探索することにより、制御性能とロバスト性向上の両立が可能となる。本発表では、開発した制御手法を「DNN モデル予測制御(DNN-MPC: DNN Model Predictive Control)」と命名し、そのコンセプトを提案する。

2. 関連研究

モデル予測制御(MPC)は最適制御の一種で、制御対象に対し、同制御対象のモデルを用いることで一定期間将来までの制御対象の状態最適化を行うものである。MPC の先行研究である GMRES/C[1]では、最初に最適解の初期値を演算し、最適解の差分を漸次的に演算することで、非線形の最適化問題を高速処理している。しかしながら、評価関数の大きな変動が発生した場合、最適解の初期値を再計算する必要があり、制御周期内に操作量が演算出来ないという課題がある。また、最適解の初期値が局所解に収束した場合、制御性能が悪化するがケースも有り得る。DNN-MPCはこれらの課題を解決するものである。

3. DNN-MPC のコンセプト

3.1 全体構成

DNN-MPC の全体構成を図 1 に示す。モデル予測制御の演算は、評価関数 F の演算と、評価関数 F の最適化問題を解く最適化ソルバーに大別される。DNN-MPC では評価関数 F の演算を FPGA で、最適化ソルバーをソフトウェアでそれぞれ実装する。

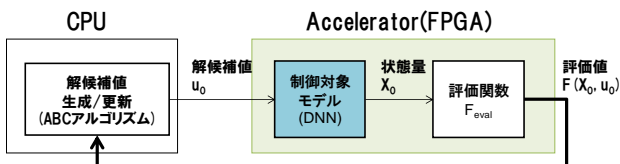


図 1 DNN-MPC の全体構成

[†] (株)日立製作所, Hitachi, Ltd.

[‡] 九州工業大学, Kyushu Institute of Technology

3.2 最適化ソルバーの選択

DNN-MPC では扱える変数量の多さ、局所解問題の回避能力、処理速度、並列化の容易性を考慮し人工蜂コロニーアルゴリズム (以下 ABC アルゴリズム) [2]を選択した(図 2)。

	Iterative Methods	Heuristics		
		GA	PSO	ABC
Number of parameter	×	○	×	◎
Local minimum problem	×	○	×	○
Speed	◎	△	○	◎
Parallelization	×	○	○	○

GA: Genetic Algorithms
PSO: Particle Swarm Optimization
ABC: Artificial Bee Colony optimization

図 2 最適化ソルバーの選択

3.3 DNN を活用した制御対象のモデル化

従来、制御対象を微分方程式でモデル化し、これを積分することで、状態量の予測をしていたが、提案方式では、DNN を用いてモデル化する。DNN を用いたモデルにより、以下のメリットが期待できる。

- 大量の実データを元に学習することにより、定式化の困難な物理現象(ヒステリシス、渦電流、人間の行動、等)をモデル化することが可能であり適用範囲が広い
- 稼動データを元に再学習することにより、制御対象の固体ばらつき、経年劣化に追従した制御が可能

4. ABC アルゴリズムの並列化

DNN-MPC で活用する ABC アルゴリズムは、蜂の捕食行動を模倣したアルゴリズムであり、初期値に依存せず、大域解を探索出来るといった特徴がある。また、蜂をプロセスエレメントに対応させることで、専用ハードウェアによる並列化が可能である。

最適解を餌(蜜)の位置とみなし、蜜の位置を蜂の集団で探索する ABC アルゴリズムの処理概要を以下に示す。

- (1) 蜂の初期化: 全蜂の位置を初期化
- (2) 収穫蜂フェーズ:
 - (2-1) 蜂の位置更新:

$$\text{更新後の位置} = \text{更新前の位置} + \text{Rand} \times (\text{他の蜂の位置} - \text{更新前の位置})$$
 (Rand は $-1 \sim +1$ の乱数)
 - (2-2) 評価関数の演算: 更新後の位置における評価関数値 H を演算

- (3) 追従蜂フェーズ：適用度の良い(評価関数値が小さい)蜂を繰返し更新
- (4) 偵察蜂フェーズ：一定回数改善のない(評価関数値が下がらない)蜂の位置を初期化
- (5) 規定回数 or 評価関数値収束まで(2)-(4)を繰り返す

ここで、並列化のために逐次演算である(4)を省略したアルゴリズムを ABC_para とし、オリジナルと比較した結果を図 3 に示す。他のアルゴリズムと比較し、最速で最適値に収束し、さらに、並列化による収束速度の低下がほとんど見られないことが確認できた。これにより、リアルタイム制御に適用できる見込みを得た。

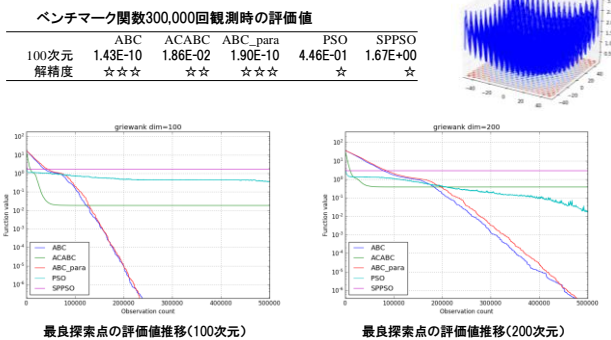


図 3 並列化 ABC の収束速度

5. DNN を活用したモデル化手法と評価結果

5.1 DNN を活用したモデル化手法

従来方式では、制御対象の微分方程式を逐次演算して状態量予測をしていた(図 4)。この場合、1 ステップの予測誤差が蓄積し誤差が増大することや、長いステップを予測する場合、演算サイクル数が増大する課題がある。

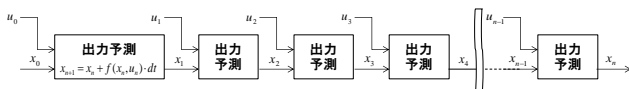


図 4 微分方程式逐次演算による予測手法

ここで、制御対象に印加する操作量 $u(0)\sim u(n)$ と、モデルの状態量初期値 $X(0)$ を一括して入力し、モデルから制御対象の挙動変化予測 $X(1)\sim X(n)$ を一括して出力するネットワークを構成(図 5)し深層学習によるモデル化を行う。これにより、演算速度の向上と誤差の低減が可能となる。

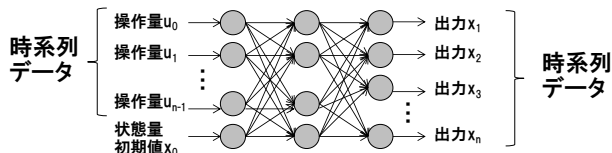


図 5 DNN を活用したモデル化手法

5.2 DNN を活用したモデル化評価結果

2 リンクアームを例に、DNN を用いたモデル化の例を示す。本 2 リンクアームモデルは 2 つのリンク(リンク 1, 2) からなり、入力リンク 1 に取り付けられたモータに印加する電圧 u 、出力は各リンク 1, 2 の角度、および各速度 $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$ である。学習に使うデータは物理式に基づくシミュレーションにて生成した。

まず、比較のため、1 ステップ先の出力を予測する DNN モデルを構築し、これを予測長分逐次演算(図 6)した結果を図 7 示す。予測ステップが進むにつれ誤差が増大し、500 ステップ先では物理モデルと大きく乖離した。

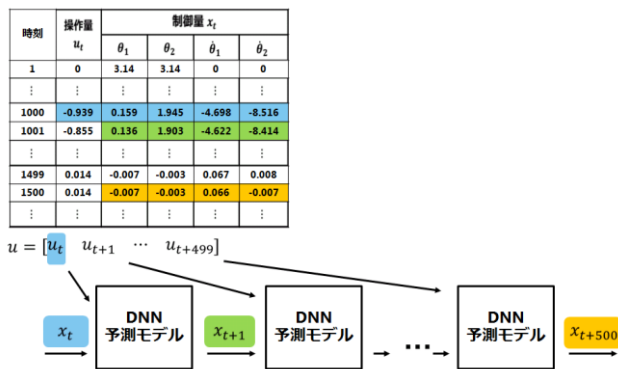


図 7 DNN 逐次演算による予測結果

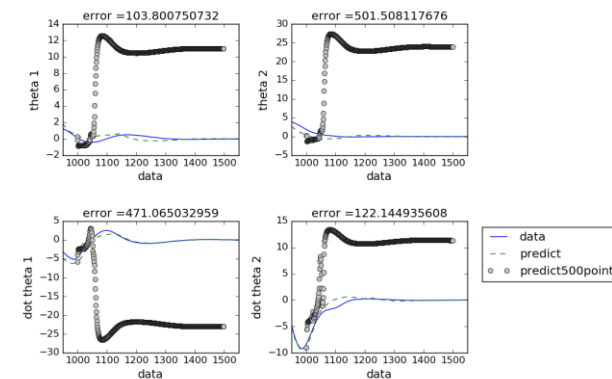


図 6 DNN 逐次演算による予測手法

次に、入力 u の時系列データを入力し、角度、各速度 $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$ の時系列データを出力するネットワークを構成し(図 8)、学習した結果を図 9 に示す。時系列データを入力とすることで、誤差の蓄積を抑止し、500 ステップ先の出力 $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$ を精度良く予測出来ることが確認された。

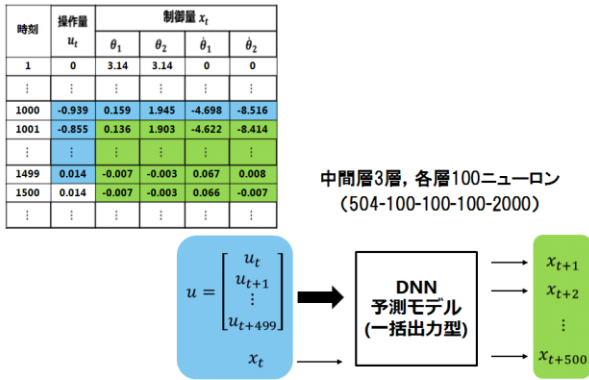


図 8 時系列入出力 DNN による予測手法

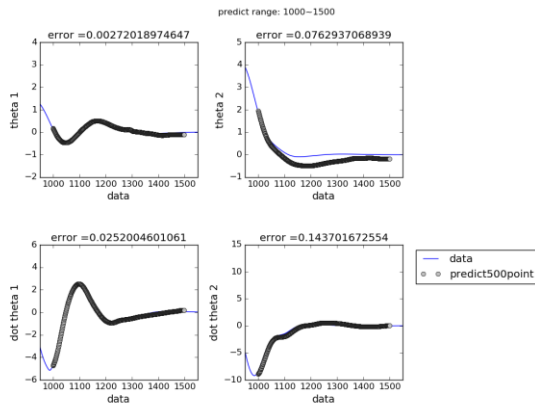


図 9 時系列入出力 DNN による予測結果

6. DNN を活用したモデル化手法の FPGA 実装

本章では、5 章で述べた DNN を活用したモデル化手法の FPGA 実装結果について述べる。

6.1 シフト加算方式

ニューラルネットワークの基本単位であるニューロンは、複数の入力データとそれに対応する重み係数を乗算し、それらの乗算結果を加算し出力する構成である。そのため、FPGA 等の論理回路で実現する場合、多数の乗算器が必要となり回路の大規模化が問題となる。そこで、入力データと重みデータの乗算を、ビットシフトを基本とした近似演算に置き換えて回路実装することで性能と小規模回路を両立するシフト加算方式を提案する。

図 10 にシフト加算方式を適用したニューロンの構成を示す。従来の入力 In と重み係数 W_n の乗算をシフト加算に置き換えて実行する。

図 11 にシフト加算の回路構成を示す。重み係数の仮数部の丸め値と入力データを乗算し、その乗算結果を丸め値のビット数分シフトする回路を共通化し、その出力を重み係数の指数部のビット数分シフトする回路で構成する。特徴として、 $2k \sim 2k-1$ の範囲の重み係数値の個数を一定とし、小さな重み係数値が 0 に丸められてしまうのを防いでいる。また、上記回路方式を一般式化し、DNN の適用対象に応じて性能と回路規模の調整できる方式とした。

図 12 に仮数シフト数 m の値に応じた $2k \sim 2k-1$ の間の重み係数値の個数を示す。例えば $m=2$ の場合、重み係数 W_n の取り得る値は $1.0 \sim 0.5$, $0.5 \sim 0.25$, $0.25 \sim 0.125$ のそれぞれの範囲内で等間隔の 4 点となる。

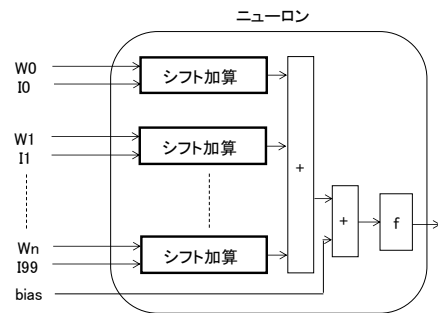


図 10 ニューロンの構成

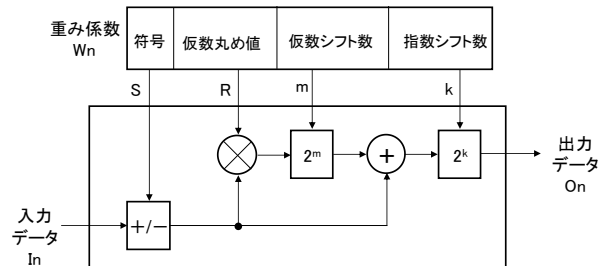


図 11 シフト加算の回路構成

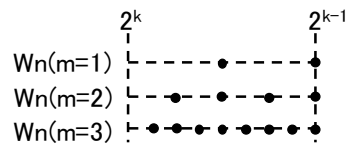


図 12 重み係数の分布

6.2 シフト加算方式の FPGA 実装結果

図 13 に 5.2 節で述べた時系列入出力 DNN モデルを、シフト加算を用いて FPGA 実装した出力を示す。仮数シフト値 m を適切に選ぶことにより、乗算をシフト加算に置き換えても精度を保つことが確認出来た。

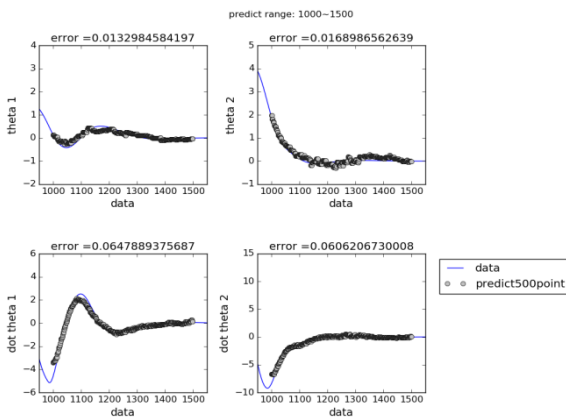


図 13 FPGA 実装した DNN モデルの出力

表 1 に本方式と従来の乗算方式を FPGA に実装した場合の回路規模を示す. 1 ニューロン(入力信号数 10)についてメモリ容量と論理規模を比較した. Target Device は CycloneV(ALTERA 社)とし, 論理は全て LE(Logic Elements)で合成した. シフト加算の仮数シフト数 $m=2$ の場合である. FPGA 実装を試行した結果, 従来の乗算器による回路に対する実装効率として, 重み係数メモリ規模は 2 倍, 論理規模は 5 倍の結果を得た.

表 1 FPGA 実装時の回路規模

	従来	提案方式
ブロック図		
メモリ	20(Byte)	10(Byte)
論理	3725(LE)	692(LE)

7. まとめ

本論文では, MPC におけるプラント予測モデルを DNN で高精度化し, 操作量探索に ABC アルゴリズムを用いることで, 大域的な最適操作量を得ることが可能な DNN-MPC を提案した. 本手法を適用することで, 制御性能とロバスト性向上を両立する制御が可能となる.

プラント予測モデルの小面積実装のため, $2n \sim 2n-1$ の範囲の重み係数値の個数を一定とし, 小さな重み係数値が 0 に丸められてしまうのを回避し性能を維持しながらニューラルネットワークを小規模な回路で実現するシフト加算方式を提案した. 本モデリング手法を FPGA 実装した結果, 従来の乗算器による回路に対する実装効率として, 重み係数メモリ規模は 2 倍, 論理規模は 5 倍の結果を得た.

今後, 実機を用いて DNN-MPC の有効性を確認する予定である.

参考文献

- [1] Ohtsuka T., A continuation/GMRES method for fast computation of nonlinear recording horizon control, Automatica, Vol.40, No. 4, pp.563574(2004).
- [2] 森大輔: 主成分分析を取り入れた Artificial Bee Colony アルゴリズム, 電気学会論文誌 C, Vol.135 No.4 pp.423-435(2015).
- [3] 金川他: 人工知能の制御へのより安全な適用について, FIT2017, 1B-3(2017)
- [4] 中川他: 組み込みシステム向け異常検知方式, FIT2017, 4E-7(2017)
- [5] 西田他: 隠れマルコフモデルを用いた複数個体による高信頼環境情報の推定技術, FIT2017, 1P-5(2017)