

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

竹松 慎弥[†] 市村 駿太郎[†] 岩間 拓也[†] 嶋岡 雅浩[‡] 道木 慎二[‡] 枝廣 正人[†]

名古屋大学大学院 情報科学研究科[†] 名古屋大学大学院 工学研究科[‡]

1. はじめに

近年の制御システムは制御対象をより適切に制御するために大規模・複雑化している。また、短い制御周期内で処理を完了することが必要であり、リアルタイム性も求められる。そのため、大規模・複雑な制御システムを高速に処理することが求められており、制御システムへの並列処理の適用が注目されている。

本研究では小型・高効率・高出力を得られるとして近年注目されている永久磁石同期モータ電流制御系を題材とし、ハイエンド制御で有効とされているモデル予測制御アルゴリズムの並列化による高速化を図る。

2. 永久磁石同期モータ

永久磁石同期モータは回転子として永久磁石を用いたものである[1]。固定子には PWM インバータを3つ使用した3相交流が用いられることが多い。

PWM インバータは直流電源の on/off を細かく切り替え、パルス幅（デューティ比）を変えることで擬似的に任意の電圧波形を生成できる。これを3相用いることによって、任意の方向に任意の大きさの磁界を生成でき、この磁界と永久磁石の引力・斥力を利用して回転させる。ゆえに3相のPWM インバータを適切なタイミングでスイッチングすることでモータを任意のトルクや速度に制御できる。

3. モデル予測制御

モデル予測制御は考えうる全ての挙動を予測し、評価関数を基に最も良いと思われる操作を選択するという最適化計算を、制御周期（操作を与えるタイミング。ステップとも呼ぶ。）毎に行うものである[1]。PWM インバータの最適なスイッチングタイミングの導出にモデル予測制御を用いる。3相インバータのスイッチングタイ

Parallelization of Model Predictive Control for PMSM current control system

Shinya TAKEMATSU[†], Syuntaro ICHIMURA[†], Takuya IWAMA[†], Masahiro SHIMAOKA[‡], Shinji DOKI[‡], Masato EDAHIRO[†]

[†] Graduate School of Information Science, Nagoya University

[‡] Graduate School of Engineering, Nagoya University

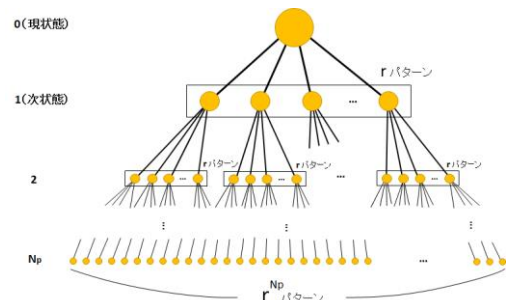


図1 モデル予測制御で探索する木

ミングの組み合わせを r パターン考慮し、 N_p ステップ先の状態までを予測する場合、図1のような全 r^{N_p} パターンの木構造から最適な評価値の葉ノードを探索する最適化問題に置き換えて考えることができる。

このような最適化問題を高速に解く方法として分枝限定法がある。これはそれ以上探索しても最適解が得られないような部分問題（枝）を切り捨て（枝刈り）、探索数を減らすという手法である。しかし、並列分枝限定探索では、逐次実行時と探索順序が変わることにより、本来枝刈りされるべきノードを無駄に探索してしまい、高速にならない場合がある。よって、並列化による探索ノード数の増加の抑制が極めて重要である。

4. 既存の研究

本研究内容に対し、筆者らは解空間の削減やアルゴリズム変更といった逐次最適化、および初期暫定解を利用した並列分枝限定探索手法を提案した[2]。なお、初期暫定解とは、深さ優先探索および最良優先探索によって最初に得られる暫定解のことであり、本研究のアプリケーションでは初期暫定解が最適解に近いことを利用している。この手法ではパターン数 r が多く（ $r = 21^3$ ）、予想ステップ数 N_p が小さい（ $N_p = 2$ ）場合、最適化により6.35倍、並列化により12コアでさらに9.05倍の高速化が達成できた（図2の(1)）。

一方、速度指令値が大きく変化した場合、より短い時間で指令値に達するために予測ステップ数 N_p を大きくしたい。 N_p が大きくなると総探索パターン数は指数的に増加するため、 r を少なくすることで増加を抑える。このとき、探索す

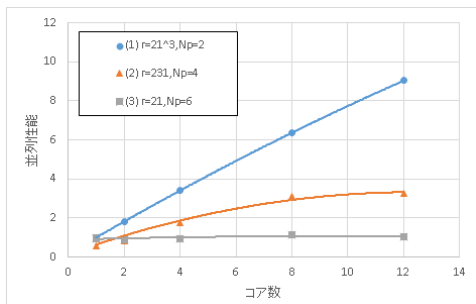


図 2 従来の並列分枝限定探索による並列性能

る木の形状は幅が狭く深さが深いものとなる。図 2 にはこのような形状となる (2) $r = 231, Np = 4$ の場合および (3) $r = 21, Np = 6$ の場合の並列性能も示している。(2), (3) の場合の並列性能は 12 コアでそれぞれ 3.26 倍, 1.02 倍と非常に悪く, コア数に対する性能向上も飽和している。よって, このような形状の木の探索でも高い並列性が得られる必要がある。

5. 改良並列分枝限定探索

文献[2]においては次のステップ時に考えられる操作に対する評価値計算をひとつのタスクとし, このタスクを数多く生成して各コアに割り当てることで並列探索させていた。図 1 の木構造探索のイメージで言い換えるならば, 現在のノードの r 個ある子ノード全てに対する評価値計算をひとつのタスクとしている。よって, r が小さいと個々のタスクの粒度は小さくなる。小さい粒度で並列処理させると並列化オーバーヘッドの強い影響を受け, 並列性能が悪くなる。これが, 図 2 の (2), (3) のような r が小さく, Np が大きい場合において並列性能が悪くなってしまいう原因であった。

これを改善するにはタスクの粒度を大きくする必要があるのである。従来の手法では子ノード, つまり 1 世代分の評価値計算のみをタスクとしていたが, これを複数世代分の評価値計算をひとつのタスクとする。従来の手法が r の粒度であるのに対し, n 世代分をひとつのタスクとすると r^n の粒度となり, 大きくすることができる。一方, 粒度が大きくなるとコア間の負荷不均衡が発生する恐れがあるが, OpenMP の task 構文にはタスクプール機能があり, 処理を行っていないコアへの動的なタスク割り当てを実現しているため, 負荷を均衡させることができる[3]。また, 分枝限定法により枝刈りされる量などによって変化するが, 最大 $r^{Np/n}$ のタスクが生成されることになる。生成タスク数がコア数に対して十分に大きければ, 1 タスク分の負荷の差は無視できる程度になるため, 負荷を均一にできる。

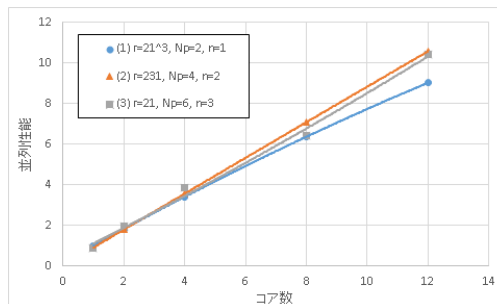


図 3 改良並列分枝限定探索による並列性能

6. 評価

次の環境で MATLAB/Simulink モデルのシミュレーションを行うことにより評価した。

- OS:CentOS
- CPU: IntelXeon E5-1695 v2 2.40GHz
- メモリ: 32GB
- コンパイラ: gcc 4.8.5

以下の 3 つの条件における並列性能の測定結果を図 3 に示す。ただし, n は 1 タスクにする世代 (ステップ) 数を表す。

- (1) $r = 21^3, Np = 2, n = 1$
- (2) $r = 231, Np = 4, n = 2$
- (3) $r = 21, Np = 6, n = 3$

r が小さく, Np が大きい (2) や (3) の場合においても 12 コアで 10 倍程度の高い並列性能が得られた。並列オーバーヘッドの影響の緩和や負荷分散ができていていると考えられる。また, コア数-並列性能の推移は線形的であり, コア数をさらに増やしても性能が飽和せず, 伸び続けるスケールな特性を持っていると考えられる。また, (2) の並列性能が最大になったが, これはタスクの粒度が 231^2 ノード分となり, (1) や (2) の 21^3 と比較して大きいことが要因と考えられる。

7. おわりに

本研究では永久磁石同期モータのモデル予測制御に対する改良並列分枝限定探索手法を提案し, 予測パターン数 r やステップ数 Np に関わらず, 高い並列性が得られることを確認した。

さらなる高速化が今後の課題として挙げられる。

参考文献

- [1] 河合健司: モデル予測制御を用いた PMSM の最適制御に関する研究, 三重大学, 2007.
- [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, 情報処理学会組込みシステム研究会, 2016-EMB-41(10), 1-8, 2016.
- [3] Oracle Solaris Studio 12.2: OpenMP API ユーザーガイド, https://docs.oracle.com/cd/E22054_01/html/8212493/docinfo.html, 2017.