

モータコイルの温度評価を考慮した Dynamic Window Approach による 移動体の軌道生成

Trajectory Generation of Moving Object by Dynamic Window Approach Considering Temperature Evaluation of Motor Coil

電気電子情報通信工学専攻 飯塚健太

Department of Electrical, Electronic and Communication Engineering, Kenta IIZUKA

1. 序論

1.1 研究背景

モータ技術は、電気-機械エネルギー変換の要として日々発展を繰り返しており、今日では、我々の日常生活を支えるキーテクノロジーとなっている。そして、近年のモータ技術は、ロボットや自動車などの幅広い分野のみならずサービス分野などでも活用されている [1]。

モータ技術の重要な要素としてモータの高トルク化、小型化、低価格化がある。これからのモータは、これらの要素を同時に満足させることが必要となる。しかし、これらの要求を満たすうえでモータの小型大出力化が進んでおり、熱による破損などの問題が大きな課題となっている。モータの発熱が問題でコイルの絶縁破壊や永久磁石の不可逆減磁を生じることから、モータの信頼性を向上させるために、モータの温度管理、温度制御は重要な問題である。このような背景の下、本研究では特に、モータコイルの過熱を考慮しながら電流を制御することで、電気自動車、IoT などの産業分野の発展を目指すものである。

1.2 研究目的

本稿では、モータコイルの温度に上限を与えることで、定格電流を超えた電流を流し、瞬時最大トルクを上げるとともに、モータコイルの温度範囲内の移動ロボットの障害物回避における軌道生成を目的とする。高電流による熱の問題をモータコイルの温度推定式、熱速度における数式を用いることで解決し、モータを最大限駆動させるための電流範囲の拡大を行う。この電流範囲内の移動体の障害物回避時の軌道生成を Dynamic Window Approach(DWA) を用いて行い、単位時間当たりの移動体の軌道の拡大を研究目的とする。

2. モータコイルの温度評価

2.1 熱伝導を用いた温度推定式

熱の伝わり方は、伝導、対流、放射の 3 タイプがあるが、大気圏の空間では伝導と放射が支配的である [2]。したがって、熱解析においては、伝導と放射のモデルに基づいて行われることが多い。モータの温度には、外部環境に依存する外界温度や機器の熱容量、モータのサイズや形状が影響する。これらのことを踏まえ、以下に熱伝導により算出した温度推定式を示す。

$$C \frac{dT(t)}{dt} = K(Ta - Tc) \quad (1)$$

$$\frac{dT(t)}{T(t) - Ta} = -\frac{K}{C} dt \quad (2)$$

$$\int_{\tau=t}^{\tau=0} \frac{1}{T(\tau) - Ta} dt = -\frac{K}{C} \int_{\tau=t}^{\tau=0} d\tau \quad (3)$$

$$\ln\left(\frac{T(t) - Ta}{T(0) - Ta}\right) = -\frac{K}{C} t \quad (4)$$

となり, さらに指数関数の表現になるよう計算を進め, 最終的に時刻 t における温度は eq(5) で示される.

$$T(t) = Ta - (Ta - T(0)) \exp\left(-\frac{K}{C} t\right) \quad (5)$$

2.2 熱拡散方程式とナビエストークス方程式を用いた熱の移動

熱の1部の推定ではモータコイル全体の動きを推定するものであるが, 実際は熱の移動は存在する. 熱拡散方程式とは, 熱の移動と温度分布をさすものである. また, ナビエストークス方程式では, 圧力, 粘性力を考慮し熱の動きを計算する方程式である. この熱拡散方程式とナビエストークス方程式を応用してモータの熱の移動速度を求める. ここで, 分子運動論の観点から熱速度を導く. 前提として分子運動の速度理論には「速度の3つの成分は独立に分布する」こと, 「分布は速度の大きさだけに依存し, 方向に依存しない」ことの2つが定義としてある. まず, 速度の3つの成分は独立に分布することについては, 以下の式が成り立つ.

$$F(v_x, v_y, v_z) = f(v_x)f(v_y)f(v_z) \quad (6)$$

次に, 分布は速度の大きさだけに依存し, 方向に依存しないことについては, 以下の式が成り立つ.

$$F(v_x, v_y, v_z) = g((v_x)^2 + (v_y)^2 + (v_z)^2) \quad (7)$$

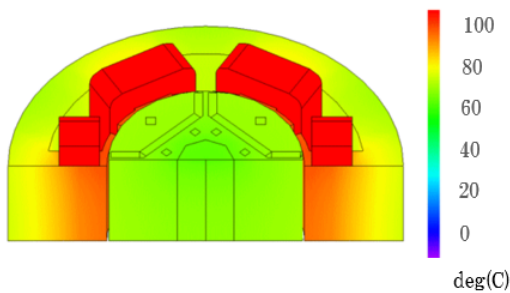


Fig. 1: simulation of heat speed(jmag)

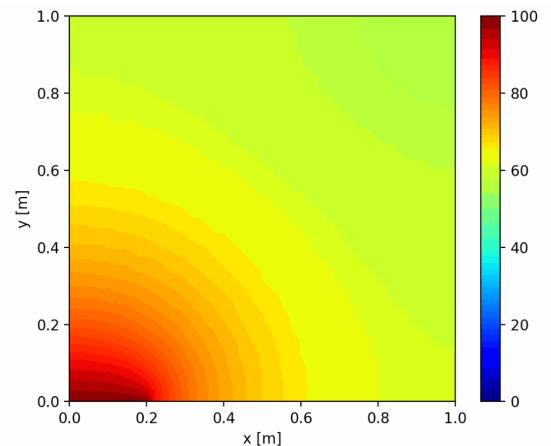


Fig. 2: Simulation of heat speed(Proposal)

モータの熱の速度の分布を eq(6),eq(7) から数式化すると,以下の式が成り立つ.

$$F(v_x, v_y, v_z) = \left(\frac{m}{2k_b T}\right)^{\frac{3}{2}} \exp\left(-\frac{m}{2k_b T}((v_x)^2 + (v_y)^2 + (v_z)^2)\right) \quad (8)$$

上記の方程式と jmag シミュレーションの熱速度を比較をした際の 100[s] 後のモータコイルの熱の速度を Fig.1 と Fig.2 に示す.jmag のシミュレーションに比べ熱速度に必要なパラメータを減らした状態で結果が近似していることを確認できた.

3. DWA を用いた移動ロボットの軌道生成

3.1 Dynamic Window Approach

DWA は、現在の位置情報と速度情報、障害物情報、そしてゴールの位置情報から、ロボットの運動モデルに則した制御入力を計算ができる。本研究では、モータコイルの温度制約条件下での DWA を用いた移動体の軌道生成を行った。この制約条件下での移動体の並進位置応答とトルク応答を従来手法の制約を持った移動体の動きを比較する。

3.2 並進位置応答

移動体の電流制約と温度制約での並進位置応答を Fig.5 と Fig.6 に示す.Fig.5 において、電流制約下では、最小値で-22[m] で、温度制約下では最小値が-46[m] となっていることから制約条件を変えることで x 座標軸だけでも 2 倍の距離を移動することを可能にした。また,Fig.6 でも最大値が電流制約下では 31[m] で、温度制約下では最小値が 47[m] であることから、座標側でも移動距離の拡大を実現した。

3.3 速度応答

移動体の電流制約と温度制約での速度応答を Fig.7 と Fig.8 に示す.Fig.7 と Fig.8 も並進位置応答と同様に電流制約に比べて、温度制約の速度が上昇していることが分かる。このことから、温度制約における単位時間当たりの移動距離の拡大が示せた。

4. 結論

本研究では、発熱の問題を熱の速度を数式化することで解決し、モータの高トルク化の課題を高電流を流すことで実現し、高トルク化を用いた移動体の障害物回避時の軌道生成のシミュレーションを行った。

まず、熱の動きを流体の観点から着目し数式化した式を,jmag シミュレーションと比較をした結果,jmag のシミュレーションに比べ、パラメータを減らし、かつ磁場解析を行わずに近似した結果が得られたことからリアルタ

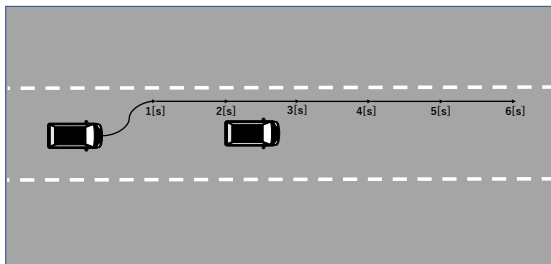


Fig. 3: Moving range of a mobile robot under current constraint

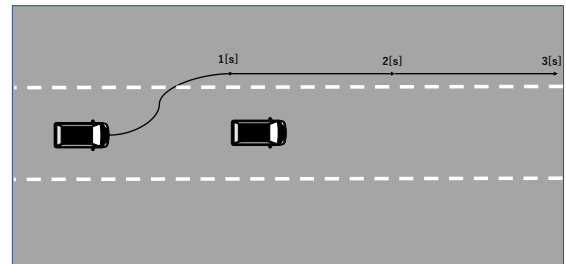


Fig. 4: Moving range of a mobile robot under temperature constraint

イムでの適用を可能とし、有用性が示せた。

また、温度制約下での軌道生成では、単位時間当たりの移動範囲が電流制約に比べて5倍の距離拡大した結果が得られた。また、その後の軌道修正も行えたことから、温度制約下での軌道生成の有用性を示せた。

参考文献

- [1] モータ技術実用ハンドブック編集委員会: "モータ技術実用ハンドブック", 日刊工業新聞社, (2001-3)
- [2] A. N. Diligenskaya, "Method of Parametric Optimization in Problems of Identification of Boundary Conditions of Convective Heat Transfer in Processes of Non-Stationary Heat Conduction," 2018 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), 2018, pp. 1-4, doi: 10.1109/FarEastCon.2018.8602677.

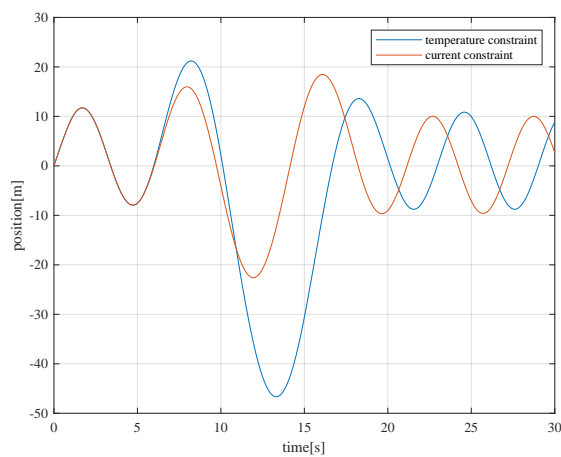


Fig. 5: Position Response(x)

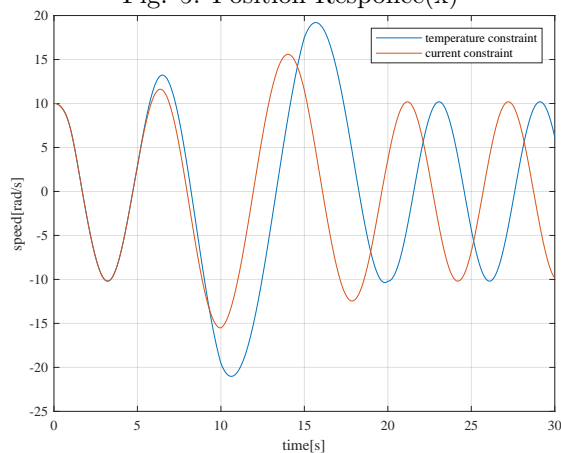


Fig. 7: Speed Response(x)

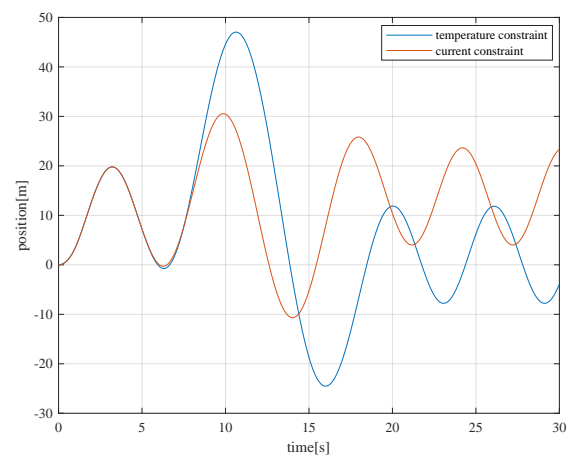


Fig. 6: Position Response(y)

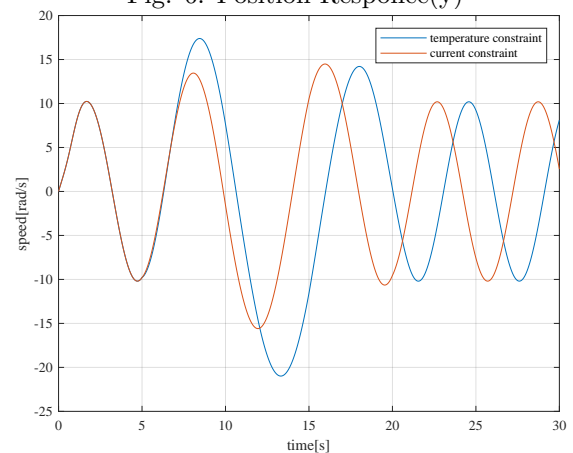


Fig. 8: Speed Response(y)