

樹脂摩擦界面画像の AI 学習による未学習荷重・速度条件下に対する摩擦力推定

Friction Force Prediction under Unlearned Load and Speed Conditions via AI Learning of Polymer Interface Images

兵庫県大・工（正）*前田 悠斗 兵庫県大・工（非）田村 慧智 兵庫県大・工（正）田中芹奈
 兵庫県大・工（正）松本直浩 兵庫県大・理，ダイセル（正）須貝幸廉 兵庫県大・工（正）木之下博
 Yuto Maeda*, Keito Tamura*, Serina Tanaka*,
 Naohiro Matsumoto*, Noriyuki Sugai*, **, Hiroshi Kinoshita*
 *University of Hyogo, ** Daicel

1. はじめに

近年，機械的接触部における摩擦現象の解明と制御は輸送機械や精密機器の高効率化・長寿命化に直結する重要課題となっている．特に，プラスチック材料の一種であるポリアセタール（POM）は，優れた機械的強度と自己潤滑性から機械の摺動部品に広く用いられている．しかし，POM の摩擦係数は荷重や速度に依存して変化し，耐摩耗性や振動・騒音特性に大きく影響するため，実使用条件下での摩擦機構を定量的に把握し，摩擦挙動を予測・制御する手法が求められている．

一方，機械学習，特にニューラルネットワークやデータ駆動型モデルの活用は，材料の構造情報や原子スケールの特性から摩擦エネルギーや密着・ファンデルワールス力といった摩擦挙動に関連する物性を効率よく予測できる点で注目されている．最近では，多次元シミュレーションデータをもとに単層材料などの摩擦特性を機械学習で高精度にモデリング・推定する手法も報告されている．この手法を応用すれば，POM - 鋼摺動面や表面特徴量から荷重や速度条件に応じた摩擦係数を自動的かつ高効率に予測することが可能となる．

したがって，本研究では，POM の乾燥摺動における摩擦機構をレビューするとともに，PIN-ON-DISC 試験で得られた荷重・速度別の摩擦データを CNN モデルに学習させ，未知条件下の摩擦挙動の推定を行った．

2. シミュレーション方法

2.1 データセット作成，データ前処理

本研究では，POM（ポリアセタール） - ガラス摩擦界面の画像と荷重・速度などの数値条件を組み合わせた教師ありデータセットを独自開発の Python スクリプトで構築した．荷重 5 N および 10 N としゅう動速度 150 rpm, 300 rpm, 600 rpm, 900 rpm の計 7 通りの条件（荷重 10 N・900 rpm では急激な摩耗が生じ測定データは得られなかった）での PIN-ON-DISC 摩擦試験から得られたデータに対してデータセットを作成した．試験中に光学顕微鏡で摩擦界面を連続観察し，取得したフレームごとに①画像ファイルパス②摩擦力(N)③荷重(N)④回転速度 (rpm) ⑤摩擦距離 (mm) をひとつのレコードとして CSV 形式のファイルに記録した．画像はフレーム間引き後に保存し，摩擦力と荷重・速度を対応付けた教師データ用 CSV を同時出力し，最終的に約 30 万枚の画像サンプルを収集・整理した大規模データセットを得た．構築した CSV をもとに，以下の前処理を実施した．リサイズ：256×256，グレースケール変換による 1 チャネル画像への変換，Normalize：ピクセル値を[0, 1]に正規化後平均 0.5・標準偏差 0.5 で標準化．CSV 中の荷重，回転速度カラムについて，訓練データ全体から平均・標準偏差を算出し， $x' = (x - \mu) / (\sigma + \epsilon)$ の式で標準化した．ここで， ϵ はゼロ除算を防ぐため $1e-8$ を設定した．摩擦力推定モデルは，pytorch を用いて以下のような最適化フローと学習戦略を採用した¹⁾²⁾．まず，GPU 上での計算効率とメモリ使用量を抑えるために torch. cuda. amp. GradScaler を利用した自動混合精度訓練を導入し，オプティマイザには Adam（初期学習率 10^{-3} ）を用い，重みの過学習を防ぐ目的で L2 正則化項を $\lambda = 10^{-4} \sim 10^{-2}$ の範囲で付与した．学習が停滞した際には ReduceLROnPlateau スケジューラ（patience=5, factor=0.1）が学習率を段階的に減衰させる仕組みを備えている．モデル構造としては，摩擦界面画像と数値条件を同時に扱える CNN+FiLM 層のマルチモーダルアーキテクチャを採用し，データセット全体に対して 5-fold 交差検証を実施してハイパーパラメータを探索した³⁾．本研究のモデルは，摺動面画像と荷重・速度といった数値条件の二つの入力を同時に受け取る構造をもつため本来の CAM ライブラリをそのまま適用することができない．そこで，元の CNN2-FiLM モデルを内部に保持し，CAM 実行時にはあらかじめセットしておいた数値条件を内部で結合してから元モデルに渡すラッパークラスを実装した．このラッパーにより，XGrad-CAM の呼び出し時には画像テンソルだけを渡せばよくなり，ライブラリの単一入力想定をクリアしている⁴⁾．次に，回帰タスク用としてモデルの出力スカラーをそのまま返す機能を自作し，クラス確率ではなく回帰値を可視化対象とするよう定義した．続いて，ラッパー付きモデルと最終畳み込み層（FiLM 後の特徴マップ）を XGrad-CAM クラスに渡してセットアップを行った．可視化実行時は，各サンプルに先に数値条件をラッパーにセットし，続けて画像テンソルのみを渡して勾配を計算，得られた 2 次元 CAM マップをヒートマップ化して元画像と重ね合わせた．これにより，画像と数値条件を同時に扱うマルチモーダルモデルであっても，XGrad-CAM の高精度な可視化機能を安定的に適用できる環境を確立した．

3. シミュレーション結果、考察

Fig. 1 には荷重 5N, 回転速度 600rpm のデータセットを摩擦係数推定の評価用, それ以外の条件のデータセットを摩擦係数推定のための学習用としてシミュレーションを行った結果を示している. ここで示しているデータセットに対する平均二乗誤差による損失は約 6.06×10^{-3} , R^2 値は約 0.937 であった. 図中の近似直線が $y=x$ の直線に近づくほど理想的な推定ができていることを示している. Fig. 2 は Fig. 1 で示した推定に対して CAM による可視化を行った結果である. Fig. 2 の CAM で示している画像では, CAM による可視化では問題点があることが確認できる. 左に示している画像では特に問題がなく, POM の摩擦部分に注目しているように見受けられる. しかし, Fig. 2 の右側に示している画像では画面全体が青く示されており, 摩擦部分に注目して推定を行っているわけではないことが分かる. CAM 画像の割合は Fig. 2 の右側のような画像が多く, これでは摩擦面画像から摩擦力の推定を行っているとは言えない. このような結果となったのはデータセット作成の際に使用した実験データにおいて, Fig. 5 に示すように摩擦距離と摩擦係数に正の相関があったからであり, 学習の際に摩擦面画像ではなく摩擦距離に対して過度に適応してしまったからであると考えられる. Fig. 1 で結果を示した推定から摩擦距離の入力をなくして学習を行った際の推定結果を Fig. 3 に示す. この推定における平均二乗誤差による損失は約 1.73×10^{-2} , R^2 値は約 0.899 であった. 摩擦距離を入力に加えた場合と比べて R^2 値はわずか 0.04 ほどの低下であるが, 損失は 3 倍ほど増加するという結果となった. また, Fig. 4 にはこの推定における CAM の可視化結果を示しており, すべてのデータにおいて摩擦部分にヒートマップが反応していることが確認できた. 以上のことを踏まえると, Fig. 1 で示した推定結果は一見, R^2 値が約 0.937 と高い精度を誇っているように見えるが, 画像から摩擦係数を推定しているというより, 摩擦距離の増加に応じて摩擦係数の推定値を出力しているだけであると考えられる. これは私たちが目指しているような推定ではないため, あくまで摩擦距離を補助データとして扱うといったような改善が必要となってくる. 具体的には, SHAP 解析で距離寄与率を算出し, 距離依存が残る場合のみデコリレーション損失を追加検討すること, 現在使用している CNN から ResNet へのモデルのアップグレードを検討していく.

4. おわりに

問題点

- ・ Fig.1 は $R^2=0.937$ と高精度だが, CAM (Fig.2) が摩擦面に反応せず, 摩擦距離に大きく依存していると判明.

結論と対策

- ・ モデルを ResNet へアップグレードし画像特徴抽出を強化.
- ・ SHAP で距離寄与率を測定し, 過度依存が残る場合はデコリレーション損失を追加.

文献

- 1) A. Paszke et al. : PyTorch : An Imperative Style, High-Performance Deep Learning Library, NeurIPS 32 (2019).
ライセンス : BSD 3-Clause License.
- 2) C. R. Harris et al. : Array programming with NumPy, Nature 585 (2020) 357–362.
ライセンス : BSD 3-Clause License.
- 3) E. Perez et al. : FiLM : Visual Reasoning with a General Conditioning, arXiv : 1709. 07871 (2017).
実装ライセンス : MIT License.
- 4) R. Fu et al. : Axiom-based Grad-CAM : Towards Accurate Visualization and Explanation of CNNs, arXiv : 2008. 02312 (2020). 実装ライセンス : MIT License.

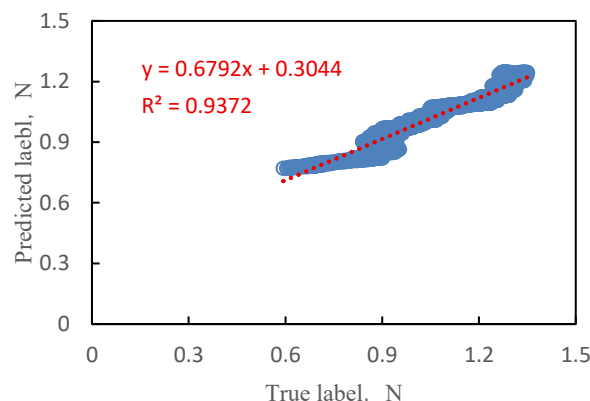


Fig. 1 Friction force estimation results

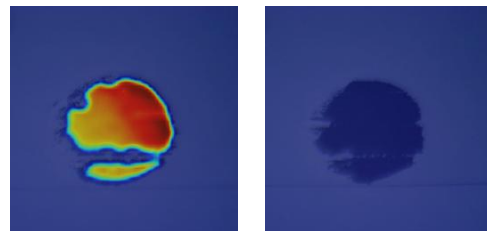


Fig. 2 Visualization results by CAM

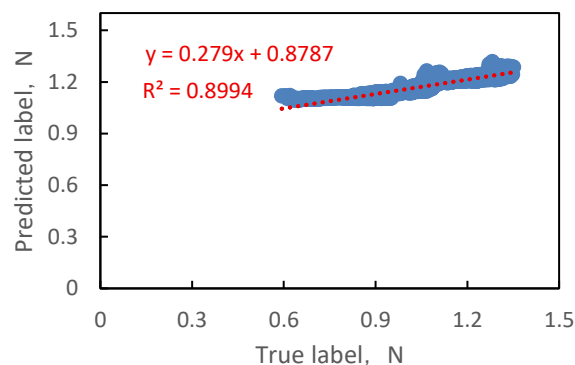


Fig. 3 Friction force estimation results without friction distance input

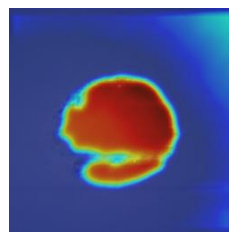


Fig. 4 Visualization results without friction distance input by CAM

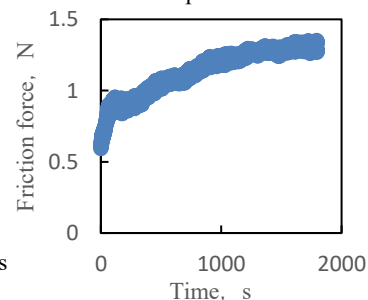


Fig. 5 Time evolution of the friction force