令和８年 11 月９日

M5Stack と MicroPython を用いた倒立振子の機械学習による制御

１．はじめに

近年、AI をロボットなどの実際の機械の制御に利用する「フィジカル AI」が注目されています。しかし、AI によって機械を動かすと聞くと、専門的な知識や高性能なコンピュータが必要と思われるかもしれません。そこで今回は、ディスプレイやセンサを内蔵したマイコン「M5Stack Core2」を用いて二輪倒立振子を製作し、機械学習によって制御する実験を行いました。

最初に従来の PD 制御で倒立させ、その動作データをニューラルネットワークに学習させます。さらに、学習した AI をマイコンに組み込み、AI だけで倒立できるか確認しました。プログラムには、Python に近い記述でマイコンを制御できる MicroPython を使用しました。

２．二輪倒立振子の製作

今回製作した二輪倒立振子は、M5Stack Core2、モータ制御ユニット、車輪などで構成されています。Core2 には加速度センサとジャイロセンサが内蔵されており、これらを利用して本体の傾斜角度と角速度を求めます。モータには Roller485 Lite を 2 個使用し、I2C 通信によって制御しました。製作した倒立振子の外観を図 1 に示します。

二輪倒立振子は、車体が傾いた方向に車輪を動かすことで姿勢を維持します。今回は、まず一般的な PD 制御によって倒立できる状態にしました。



図１　製作した二輪倒立振子（倒立状態）

3. MicroPython による PD 制御

PD 制御では、目標姿勢からの傾きと角速度に応じてモータ指令を求めました。今回使用した制御式は次のとおりです。

$$u = -K_p\theta - K_d\omega$$

ここで、u はモータ指令値、θ は傾斜角度、ω は角速度、K_p と K_d は制御ゲインです。今回は、K_p=40、$K_d = 1.0$ としました。

傾斜角度の推定には、加速度センサとジャイロセンサの情報を組み合わせる相補フィルタを使用しました。制御周期は 10 ms とし、MicroPython でモータを制御したところ、二輪倒立振子を安定して倒立させることができました。

4．機械学習による制御器の作成

次に、PD 制御の動作を AI に学習させました。PD 制御で倒立している間、次のデータを学習データとして記録しました。

- 傾斜角度（θ）
- 角速度（ω）
- PD 制御によるモータ指令値（u）

制御周期 5 ms で 1,000 件、約 5 秒間のデータを取得し、CSV ファイルとして保存しました。

取得した CSV ファイルを PC に読み込み、Python の機械学習ライブラリを使用してニューラルネットワークを学習させました。今回のネットワークは、入力層 2 個、中間層 4 個、出力層 1 個という小さな構成です。入力は傾斜角度と角速度、出力はモータ指令値です。今回のような小規模なニューラルネットワークであれば、学習後の計算をマイコン上で実行することができます。

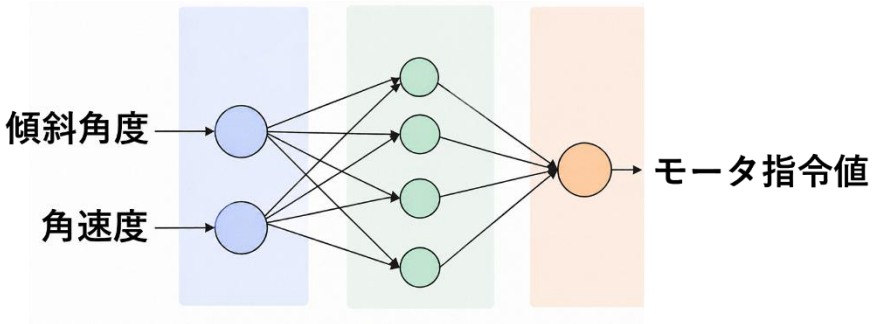


図 2　ニューラルネットワークの構成

学習には最初の 800 件を使用し、残りの 200 件で予測精度を確認しました。その結果、モータ指令値の平均絶対誤差（MAE）は 0.34 となり、PD 制御の指令値を高い精度で予測できることが確認できました。学習後の重みとバイアスを MicroPython のプログラムに組み込むことで、Core2 単体でも AI による推論ができるようにしました。

今回使用したのは、PD 制御の入出力関係をニューラルネットワークに学習させる方法です。

このような方法は、模倣学習の一種と考えることができます。

5. AI による倒立制御実験

最初に、モータの制御は従来の PD 制御で行い、同時に AI にもモータ指令値を計算させました。その結果、AI が予測した指令値は PD 制御の指令値とよく一致しました。

次に、PD 制御と AI の出力を組み合わせた混合制御を試みました。

$$u = (1-\lambda)u_{\mathrm{PD}} + \lambda u_{\mathrm{AI}}$$

ここで、u_{PD}とu_{AI}は PD 制御と AI のモータ指令値で、λ は AI の割合です。AI の割合を 10%（λ=0.1)、50%（λ=0.5）と増やして実験したところ、いずれも倒立できました。

最後に、AI の割合を 100%にして実験しました。この場合、モータの指令値はニューラルネットワークの出力だけで決定されます。なお、安全確認のため、PD 制御による指令値も並行して計算し、AI の出力が大きく異なる場合には停止する仕組みとしました。

まず 2 秒間の倒立に成功したため、続いて制御時間を 60 秒間に延長しました。長時間の実験では、測定データの記録を省略し、メモリ使用量を抑えたプログラムを使用しました。実験中は指で車体を繰り返し軽く押し、外乱を与えましたが、**AI100%の制御によって 60 秒間倒立し続けることができました。**表 1 に実験結果を示します。実験では、最大傾斜角度が 5.2° となりましたが、転倒することなく倒立を維持しました。なお、学習データの取得時は、短時間で多くのデータを収集するため制御周期を 5ms とし、AI による倒立制御では制御周期を 10ms としています。

表 1　AI100%制御による 60 秒間の倒立実験結果

制御時間	60 秒
制御周期	10 ms
制御回数	6,000 回
最大傾斜角度（絶対値）	5.2°
最大モータ指令値（絶対値）	260
AI と PD 制御の平均指令差	0.36
AI と PD 制御の最大指令差	2.43

図 3 に、AI と PD 制御のモータ指令値を比較した 5 秒間の実験結果を示します。AI と PD 制御の指令値は、大きな外乱が加わった際にもほぼ同じ応答を示し、ニューラルネットワークが PD 制御の動作をよく再現していることが確認できました。

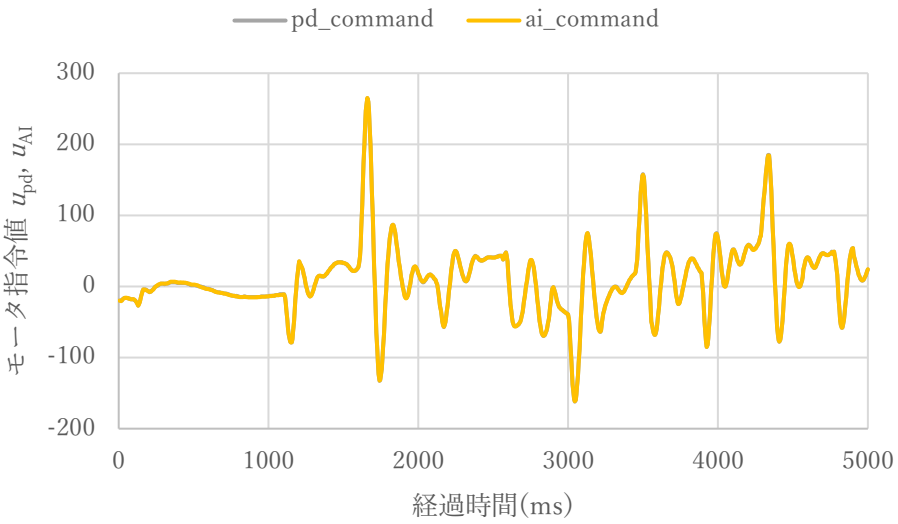


図３　AI と PD 制御のモータ指令値

6．まとめ

M5Stack Core2 と MicroPython を用いて二輪倒立振子を製作し、PD 制御による動作データをニューラルネットワークに学習させました。その結果、学習した AI だけでモータを制御し、外乱を与えながら 60 秒間倒立させることができました。

今回の AI は、PD 制御を模倣するように学習させたものであり、AI が独自に倒立方法を発見したわけではありません。また、PD 制御より高い性能が得られたことを示すものでもありません。しかし、**マイコンによるデータ収集、PC での機械学習、学習済み AI による実機制御**という一連の流れを、比較的簡単な装置で確認することができました。

機械学習は画像認識やデータ分析だけでなく、実際の機械の制御にも利用できます。今回の実験が、製造現場における AI 活用を考えるきっかけになれば幸いです。

参考資料

1) M5Stack Core2 資料：https://docs.m5stack.com/ja/core/Core2

2) Roller485 Unit UIFlow2 API 資料：https://uiflow-micropython.readthedocs.io/en/master/unit/roller485.html

3) scikit-learn MLPRegressor：https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

問い合わせ：新潟県工業技術総合研究所
技術統括センター　斎藤 雄治
TEL：025-247-1320