技術レポート　Technical Report

ステッピングモーターの仕組みとマイコンによる加減速制御

Principles of Stepper Motors and Acceleration Control with a Microcontroller

レポート番号	MTIC-TR-2026-003
団体	NPO 法人松戸技術革新センター M-TIC　松戸ロボット研究会
著者	大城 公平
発行日	2026 年 10 月 25 日（A4 版・改訂）　初版（A3 ポスター）：2026 年 8 月 21 日
対象	ステッピングモーター制御検証系（Raspberry Pi Pico）
分類	技術レポート／駆動・モーター制御
機体状態の基準日	2026 年 9 月 30 日

本レポートの数値と機体状態について　本検証系は研究開発段階にあり随時変更されるため、記載の仕様は基準日時点のものであり、現在の実機と一致することを保証しない。数値には種別を付す：**公称**＝カタログ・設計値、**実測**＝実際に計測した値（記載桁が測定分解能）、**概算**＝目測・推定値、**計算値**＝実測・公称値からの計算。

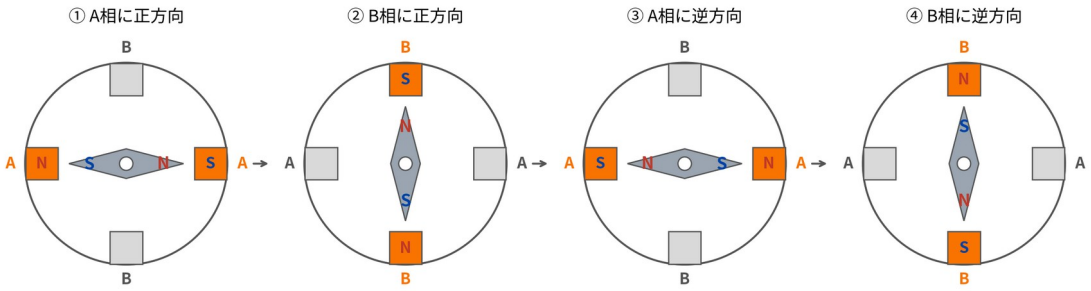
概要

ステッピングモーターは、パルスの数と速さで動きを記述できるモーターであり、位置センサーなしで回転角度を制御できる。本検証では、2 相バイポーラステッピングモーターを STEP/DIR 方式のマイクロステップドライバと Raspberry Pi Pico（MicroPython）で駆動した。ドライバはハーフステップ（400 パルス/回転）・電流 1.0 A に設定しており、モーターの定格電流 1.0 A と一致する。パルスはマイコンのプログラムで 1 本ずつ作り、周波数を 400 Hz から最高 3300 Hz まで 8 秒かけて滑らかに上げ、一定速度で回したのち 10 秒かけて下げる加減速のパターンを実行した。低速側の下限を 400 Hz に置くことで、起動直後の脱調を避ける設計とした。無負荷では電源 12 V・24 V のどちらでも回り続けたが、高速側で速さが伸びていないように見えた。脱調が起きる周波数・負荷時の挙動・発熱はまだ計測しておらず、今後、実回転数を測るセンサーと数値化した負荷を用いて、データをもとに評価したい。

1. 背景と目的

1.1 開発背景

ステッピングモーターは、3D プリンター・CNC・ロボットアームなど「正確な位置決め」が求められる機械に広く使われている。内部に歯車状のローター（回転子）と複数のコイルがあり、コイルに順番に電流を流すことでローターが一定の角度ずつ段階的に回る。この 1 段階をステップと呼ぶ。



2相バイポーラ型：A・B の2つのコイルに、電流の向きを切り替えて流す。1つのコイルは向かい合う2つの極を同時に磁化し（オレンジ、S と N）、
ローター（磁石）の N が S 極に引き寄せられて ①→④ の順に 90° ずつ回る（概念図）。実際のモーターはローターに多数の歯があり、1 ステップ 1.8° のように細かく回る。

図 1　2 相バイポーラ型ステッピングモーターが回る仕組み（概念図）

2 相バイポーラ型では、A・B の 2 つのコイルに電流の向きを切り替えて流す。1 つのコイルは向かい合う 2 つの極を同時に磁化し、磁石のローターが S 極に引き寄せられて一定の角度ずつ回る（図 1）。実際のモーターはローターに多数の歯があり、1 ステップ 1.8°のように細かく回る。

特性	内容
ステップ角	一般的な機種は 1.8°/ステップ（1 回転＝200 ステップ、公称）。パルスを 200 回送れば 1 回転する
オープンループ制御	「何ステップ送ったか」を数えるだけで現在の角度が分かるため、位置センサーが不要。構成が単純でコストと配線を削減できる
保持トルク	停止中も通電していれば軸がその場に保持される。位置決め用途に適する
脱調	パルスが速すぎたり負荷が重すぎたりすると、ローターがパルスに追いつけず止まる。位置がずれてもマイコン側からは分からない

アームの関節にステッピングモーターを使うことの良し悪しを、ほかのモーターと比べる形で表にした。

観点	長所	短所
位置決め	パルス数で角度が決まり、位置センサーなしで位置決めできる	脱調が起きると位置がずれ、ソフトからは分からない
低速・保持	低速でもトルクが出やすく、停止中も軸を保持できる	保持中も電流が流れ続けるため発熱する
高速	—	回転数が上がるほどトルクが下がり、高速には向かない
制御	ドライバがあればマイコンの数本の信号線（STEP／DIR／ENA）で制御でき、構成が単純	同じ出力の DC モーターより大きく重くなりやすい
向く用途	アームの関節・昇降・XY ステージなど、決まった位置に止める動き	車輪の駆動など、速く長く回し続ける用途には向きにくい

1.2 目的と目標

マイコン（Raspberry Pi Pico）による STEP/DIR 方式の制御を検証し、開発中のピッキングロボット（水平多関節アームで物をつかんで運ぶロボット）の Z 軸昇降・アーム関節の制御基盤とする。特に、止まった状態から速度を上げ、また止めるまでの「加減速」を脱調させずに行う方法を確かめる。

目標：マイコンからの信号（STEP・DIR・ENA）でモーターの回転・停止・方向を制御し、加速→一定速度→減速の一連の動きを脱調なく実行できること。

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.

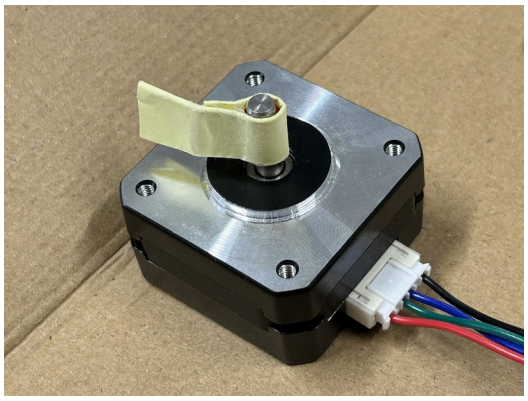


図 2　検証に使用したステッピングモーター（2 相バイポーラ）

2. システム構成

2.1 STEP/DIR 方式

モーターへの電流切り替えは専用ドライバが担い、マイコンはドライバへ信号を送るだけでよい。「速さ＝パルスの周波数」「距離＝パルスの個数」「向き＝DIR の電圧」の 3 つの対応関係で、モーターの基本的な動きを記述できる。

信号	役割
STEP（PUL）	パルス信号。1 パルス＝1 ステップ回転。周波数が速度、パルス数が移動量になる
DIR	回転方向の指定（High/Low で正転・逆転）
ENA	運転許可。プログラムの開始時に有効にし、終了時に無効にする
マイクロステップ	ドライバ側の機能。1 ステップをさらに細分割（例：1/2、1/16）し、滑らかで静かな回転を実現する

2.2 検証システム

本検証系は、マイコン・ドライバ・モーター・電源だけで構成した独立した実験系である。CAN バスの検証（MTIC-TR-2026-001）とは別のマイコンで行っており、CAN 通信基板は載せていない。使用したモーター・ドライバ・配線を図 2〜図 5 に示す。

区分	仕様
マイコン	Raspberry Pi Pico（MicroPython）
ドライバ	STEP/DIR 方式 マイクロステップドライバ（入力 DC 9〜42 V、公称。信号入力は光アイソレーション、過電流・過熱保護付き、公称）
ドライバ設定	DIP スイッチ SW1〜6＝OFF・ON・ON・ON・OFF・ON → ハーフステップ（2/B、400 パルス/回転）、電流 1.0 A（ピーク 1.2 A）
モーター	2 相バイポーラ ステッピングモーター 42DYB311C-21D18（42 mm 角）。保持トルク 0.16 N·m、ステップ角 1.8°、定格電流 1.0 A、巻線抵抗 3.6 Ω（公称、本体ラベル）
信号配線	GP16→PUL（STEP）／GP17→DIR／GP18→ENA／GND
電源	ドライバに DC 12 V（実測、テスター）を別途接続（24 V でも動作を確認）。マイコンからは信号のみ

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.

区分	仕様

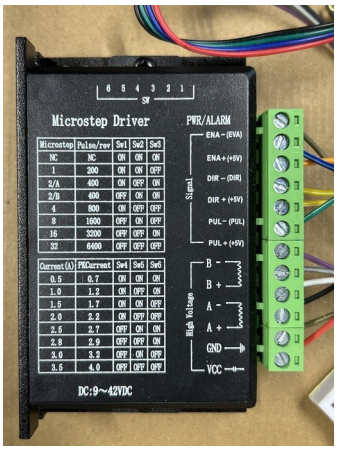


図 3　STEP/DIR 方式マイクロステップドライバ

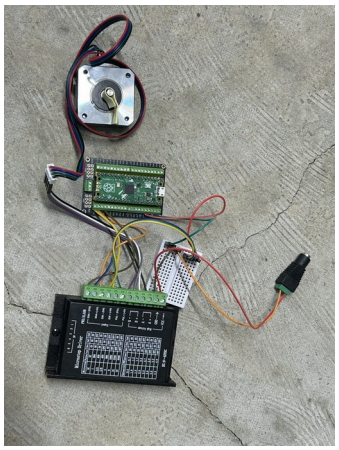


図 4　検証系の全景（マイコン・ドライバ・モーター）

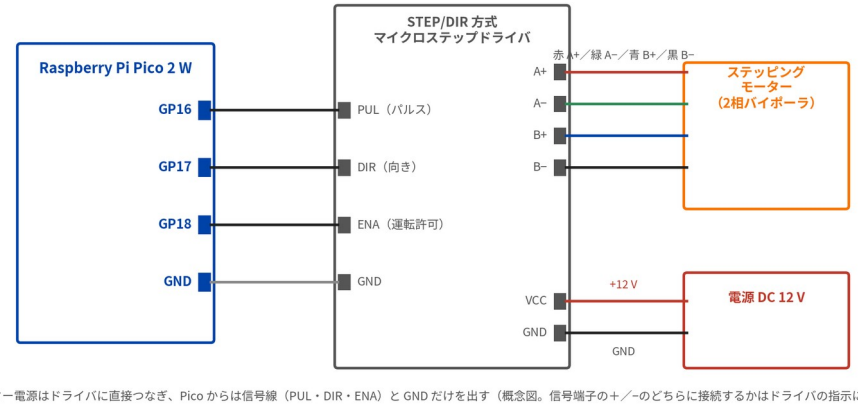


図 5　本検証の配線（概念図）



図 6　ドライバの DIP スイッチ（下側が ON）



図 7　モーター本体のラベル

ドライバの電流設定（1.0 A、図 6）はモーターの定格電流（1.0 A、図 7）と一致しており、定格どおりの条件で駆動している。ハーフステップでは 1 パルスで 0.9°回転するため、1 回転に 400 パルスを要する。

2.3 ソフトウェア：パルスの生成と加減速

STEP パルスはマイコンのプログラムで 1 本ずつ作る。周波数 f［Hz］に対して半周期 1/(2f) 秒ずつ出力ピンを High・Low に切り替えると、周波数 f のパルス列になる。加減速は、開始周波数から目標周波数までを 500 段階に分け、各段階で周波数を少しずつ変えながら短時間ずつ回すことで、滑らかに速度を変える（図 8）。

```
半周期 = 1 / (2 × 周波数)        # 例：400 Hz → 1.25 ms
繰り返し：
  STEP ピン = High → 半周期 待つ
```


Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.

```
  STEP ピン = Low → 半周期 待つ
加減速：開始周波数→目標周波数 を 500 段階に分けて上の繰り返しを行う
```

図 8　パルス生成と加減速の基本的な考え方（擬似コード）

止まった状態から急に速いパルスを送ると脱調しやすいため、低速側の下限を 400 Hz に置き、そこから 8 秒かけて最高周波数まで上げる。減速は 10 秒かけて 400 Hz まで下げ、回転方向（DIR）は一連の動作の間、変えない。

区間	内容（設定値）
加速	400 Hz → 3300 Hz を 8 秒で直線的に上げる
一定速度	3300 Hz で 5 秒
減速	3300 Hz → 400 Hz を 10 秒で直線的に下げる
下限	400 Hz（これより遅いパルスは使わない）

3. 検証方法と結果

3.1 検証方法

無負荷・屋内で、上記の加速→一定速度→減速のパターンをプログラムで実行し、モーターの回転を目視で確認した。DIR の High/Low で方向が反転すること、ENA を無効にすると停止することも確認した。

3.2 結果

項目	結果
回転・停止・方向反転	STEP・DIR・ENA の信号で制御できることを確認
加減速パターン	400 Hz → 3300 Hz → 400 Hz の一連の動作を実行できることを確認（無負荷、電源 12 V と 24 V の両方で回り続けた）
高速側の様子（目視）	周波数を上げても、途中から回転の速さが伸びていないように見えた。実回転数は測っていないため、原因は 4 章で検討する
回転速度（計算値）	400 Hz ＝ 1.0 回転/秒（60 rpm）、3300 Hz ＝ 8.25 回転/秒（495 rpm）。ハーフステップ 400 パルス/回転で換算

脱調する上限周波数、負荷をかけたときの挙動、発熱、位置決めの精度は、まだ測っていない（5.1 節）。

4. 課題と考察

- **定量評価の不足：**本検証は動作確認にとどまり、脱調が起きる周波数・発熱・位置精度は測っていない。3300 Hz で脱調した場合は 2400 Hz 程度に下げる想定でプログラムを組んでおり、実際の上限は計測して決める必要がある。
- **速さが頭打ちに見えた原因：**2 つの可能性が考えられる。①プログラムのループでパルスを作る方式では、処理の遅れがそのまま周期に乗るため、指令した周波数より低いパルスしか出ていない可能性がある（3300 Hz の半周期は約 151 µs で、ループ 1 回の処理時間と同じ桁になる）。②回転が速くなるほどコイルに電流が入りきらずトルクが落ちる、モーター側の限界。実回転数を測っていないため、どちらかは判別できていない。ハードウェア（PWM）でパルスを作る計測プログラムで同じ周波数を試し、12 V と 24 V で頭打ちの位置が変わるかを見れば切り分けられる。

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.

- **パルス生成の方式：**プログラムのループで 1 本ずつパルスを作る方式は単純だが、マイコンがほかの処理（通信・センサー読み取りなど）を同時に行うとパルスの周期が乱れる。多軸化や他機能との同居では、マイコンのタイマーや PWM 機能でパルスを作る方式に移す必要があると考えられる。
- **分解能と速度の関係：**現在のハーフステップ設定では 0.9°単位で位置決めできるが、同じパルス周波数ならマイクロステップを細かくするほど回転は遅くなる。実機では必要な速度と滑らかさに応じて分割数を選ぶ必要がある。

5. 今後の展望

5.1 検証計画

計測項目	方法	判定基準
脱調上限周波数	無負荷・負荷ありで周波数を段階的に上げ、脱調が起きた周波数を記録（PWM 方式の計測プログラムを使用）	実機で必要な速度に対して余裕があること
実回転数の計測と脱調の自動判定	軸に白黒の円盤を付け、反射型フォトセンサー（またはホールセンサー）で 1 回転ごとのパルスをマイコンで数える。指令（周波数÷400）と実回転数を比べ、9 割を下回ったら脱調と判定	実回転数が指令と一致。脱調を目視なしで判定できる
負荷トルクと脱調周波数の関係	軸に糸巻き（半径 r）を付けて糸でおもり（質量 m）を持ち上げる。負荷トルク＝m×9.8×r［N·m］。100・200・500 g で上限周波数を記録	負荷トルクに対する上限周波数の曲線（保持トルク 0.16 N·m に対する余裕）
加減速時間の下限	加速時間を 8 秒から短くしていき、脱調せずに最高周波数に達する最短時間を記録	実機の動作に必要な時間内であること
位置決め精度	400 パルス×N 回送り、軸の実回転角を目印で確認	累積誤差が許容内
発熱	連続運転 30 分後のモーター・ドライバ表面温度を測定	触れられる温度・ドライバの保護動作なし
通信との連携	CAN バス（MTIC-TR-2026-001）で受けた速度指令値で本制御を動かす	指令値どおりの速度で回ること

5.2 計測アプリの開発

脱調が起きる周波数を手早く記録するため、PC からマイコンへ周波数の指令を送り、回ったか脱調したかを記録してグラフにする計測アプリを開発している（図 9）。周波数を一定の刻みで自動的に上げ、結果を表とグラフで残し、CSV と画像に保存できる。現在は動作確認の段階で、回転の判定は目視で行っている。今後は、軸に取り付けたセンサーで実回転数を読み取り、判定を自動化する。

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.

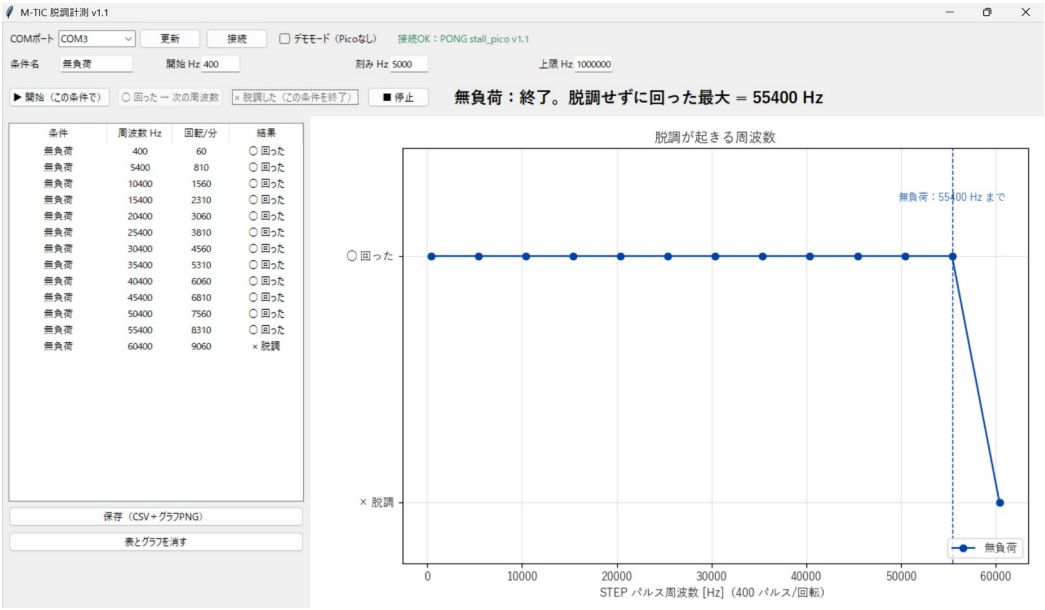


図 9　開発中の計測アプリ（動作確認時の画面。表示されている値は確認用で、本レポートの計測結果ではない）

5.3 応用の可能性

開発中のピッキングロボット（水平多関節アーム）では Z 軸昇降・アーム関節の駆動にステッピングモーターを採用予定であり、本検証はその制御基盤となる。このあとは、一定のトルクをかけた負荷試験と計測アプリを使って数値を集め、アームを安定して動かせる条件を数字で決めていく。

6. まとめ

ステッピングモーターは「パルスの数と速さで動きを記述できる」モーターであり、STEP/DIR 方式によりマイコンの信号だけで制御できることを確認した。パルスをプログラムで 1 本ずつ作り、400 Hz から 3300 Hz まで 8 秒で上げて 10 秒で下げる加減速パターンを無負荷で実行できた。無負荷では 12 V・24 V のどちらでも回り続けたが、高速側で速さが伸びていないように見え、パルス生成の方式かモーターの限界かは判別できていない。脱調が起きる周波数・負荷時の挙動・発熱はまだ計測しておらず、実回転数のセンサーと数値化した負荷を用いて、データをもとに評価したい。

参考資料

Raspberry Pi Ltd, “Raspberry Pi Pico Datasheet”

MicroPython, “machine — functions related to the hardware” ドキュメント

改訂履歴　初版（2026-08-21）A3 ポスターとして発行。　A4 版（2026-09-30）検証系を独立したマイコンでの構成として記述し直し、加減速の制御パターン、回る仕組みの図、長所・短所の整理、ドライバ設定・モーター仕様・配線を追加。開発中の計測アプリについて記載。　2026-10-25　文章を見直し、誤記を訂正。

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.