

# Pythonでやる前に…運動方程式を手計算で解いてみる

川村 聡

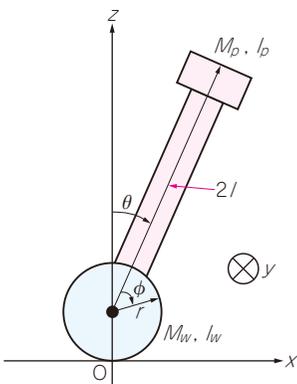


図1 車輪付き倒立振り子ロボットのモデル

第1部では、図1のような倒立振り子の仕組みをベースとした倒立ロボットを作ります。本章では難しい数式がでてきますが、そこでは「中身はこのような数式なんだなあ」と眺めるぐらいで問題ありません。第2章以降へ読み進めると、計算ライブラリが豊富なPythonで楽にさまざまな処理ができることが分かります。それに加えて、シミュレーション、実機製作および実験と、理論から実践まで、制御のトピックを抜粋してお届けします。(編集部)

## ● 第1部でやること

本稿では図1のような振り子の下部に車輪の付いたロボットを倒立状態に制御することを目指します。PC上でのシミュレーションも行いますが、どちらかというと実機の制御に重きを置きます。

## ● 基本の倒立振り子ですら数式がめちゃ難しい

倒立振り子モデルは制御理論を学ぶ学生向けの教材としてよく用いられます。しかし、その挙動は単純ではありません。このモデルの運動方程式は次のようになります。

[振り子の運動方程式]

$$(I_w + I_p + (l^2 + 2rl \cos \theta) M_p + (M_w + M_p) r^2) \ddot{\theta} + (I_w + M_p r l \cos \theta + (M_w + M_p) r^2) \ddot{\phi} - l M_p g \sin \theta - l M_p r \sin \theta \dot{\theta}^2 - \mu_\phi \dot{\theta} = 0 \dots\dots\dots(1-1)$$

[車輪の運動方程式]

$$(I_w + M_p r l \cos \theta + (M_w + M_p) r^2) \ddot{\theta} + (I_w + (M_w + M_p) r^2) \ddot{\phi} - M_p r l \sin \theta \dot{\theta}^2 - \mu_\phi \dot{\phi} = \tau \dots\dots\dots(1-2)$$

式が長く、各項に微分項と三角関数の積が現れる複雑な形をしています。運動方程式は、制御のしやすさを表す式でもあるので、倒立ロボットの制御が簡単ではないことが分かります。

## ● 安心してください! 難しい計算はPythonにお任せ

今回は次のような手順でモデル解析と実装を進めます。

[手順]

- ①運動方程式を求める
- ②運動方程式を数値的に解ける形に変形する
- ③具体的な物理パラメータを機械CADなどで求める
- ④微分方程式を解く
- ⑤解をグラフやアニメーションで表示する
- ⑥制御則を組み込んで倒立状態を維持できるようにする
- ⑦マイコン・プログラムに制御則を組み込んで実機を動作させる

PythonのSymPyやSciPy、Matplotlibパッケージを使えば、①～⑥の運動方程式の導出やシミュレーション結果の表示などを既存の関数の呼び出しだけで行うことができます。特にSymPyのメカニカル・ライブラリによる物理数式の導出は、従来MahtematicaやMapleといった高価な数学ソフトウェアでしかできなかったものが無料で試せる機能です。

⑦の実機への実装には、MicroPythonやCircuitPythonなどが使えます。いずれも基本的なPythonのコードに加え、ハードウェアを制御するため汎用I/Oやペリフェラルを制御するコマンドが用意されていま