

# ロボット用DCモータの安定動作制御とシミュレーション

## Stability Control and Simulation of DC Motor for Robot

○佐藤健太郎\*, 神田和也\*

○Kentaro Sato, Kazuya Kanda

\*鶴岡工業高等専門学校

\*Tsuruoka National College of Technology

キーワード： 電子回路(electronic circuit), 比例制御(Proportional control)

連絡先： 〒997-0842 山形県鶴岡市井岡字沢田104 鶴岡工業高等専門学校 電気電子工学科  
神田和也, Tel.: (0235)25-9095, Fax.: (0235)25-9095, E-mail: kanda@tsuruoka-nct.ac.jp

### 1. 目的

ロボット技術は我が国が最も進んでいるといわれ、安価なロボットキットも発売されるなど、一般の人々にも広く普及し始めている。

そのような中で、本研究ではヒューマノイドロボットキットにも使われている小型サーボモータの制御に注目した。このモータは内部にDCモータ、ギア、ポテンショメータ、制御基板が実装されており、信号により目的の角度まで動作させることができる。しかし、サーボ制御は目的の角度まで高速で動作することで慣性による揺れが発生しやすい。そこで、内部の制御基板を取り外し、直接モータを制御する方法を考察することとした。

制御方法の考察には制御対象の伝達関数の同定が必要であるが、今回使用したサーボモータは内部DCモータの定格が公開されていない。よって、実験的に伝達関数を推定する必要がある。

以上より、本研究では、モータ制御の環境を構築すること、回転角度のデータからモータの伝達関数を同定することを目的とした。伝達関数の同

定にはシミュレーションソフトを用い、パラメータを調整し再現を行う。また同定した伝達関数の評価として、比例制御を実装しゲインを変化させ、シミュレーション結果との比較を行った。

### 2. モータの概要

本研究では制御対象として、小型サーボモータ(近藤科学社, KRS-788)を用いた。先述のとおり、このサーボモータは

- 制御基板 (サーボ制御)
- DCモータ (定格12[V])
- ギア (減速比: 0.0033)
- ポテンショメータ

という要素で構築されており、信号線にパルスを送信することで所望の角度まで動作させることができる。しかし今回の目的は内部DCモータの直接制御であるので、サーボ制御を行う制御基板は取り外して実験を行う。なお、モータはギアを通

しポテンショメータと接続されている。このポテンショメータはフィードバック系におけるセンサとしてそのまま使用する。

### 3. 制御環境

全体の構成をFig.1に示す。マイコンからモータドライバへ信号を送ることでモータを回転させ、回転角度はポテンショメータから電圧値としてマイコンへ送られる。得られる角度信号から制御を行うプログラムを実装する。

本研究におけるモータの制御には、マイコンとしてH8/3052F(ルネサステクノロジー社製)を用いた。クロックが25[MHz]と高速であることや、A/D変換の分解能が10[bit]と高いことなどから、モータ制御には適していると考えた。

モータドライバにはFETによるHブリッジを用いた。Fig.2に回路図を示す。この回路は2つの信号によりモータの正転・逆転を容易に切り替えることができる。また、信号にPWMを用いることで、巨視的な電圧値をマイコン側で設定することができるため、速度制御も行いやすい。今回使用したPWM信号は、H8内クロックを50000[count]した時間を周期とし、その中でパルス幅のカウントを変えることでデューティ比を操作し、速度制御している。

角度信号を得るセンサとしては、内蔵されているポテンショメータから電圧値として取り出し、A/D変換することとした。今回制御する範囲は180[deg]程度であるが、ポテンショメータの最大回転角は180[deg]よりも大きく上回っているため、そのままでは有効に分解能を使用することができない。さらに、0[deg]が0[V]に対応していないため、そのまま増幅するだけでは効果を得ることができない。そこで、差動増幅回路を用いることとした。

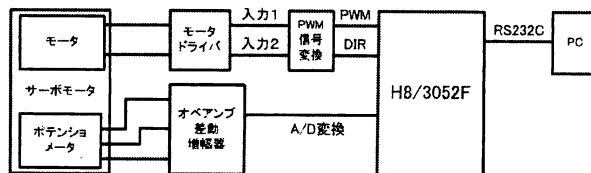


Fig. 1 実験機構

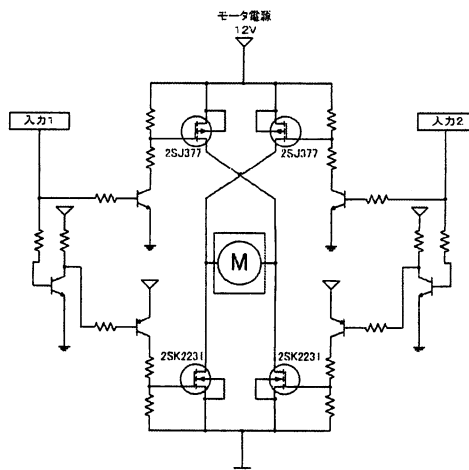


Fig. 2 モータドライバ回路

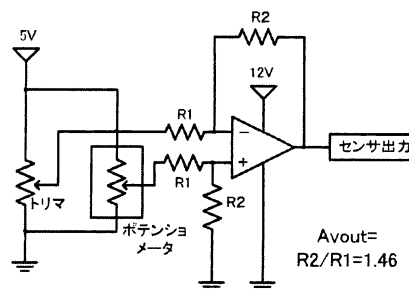


Fig. 3 差動増幅回路

Fig.3に示すように、オペアンプのマイナス端子にトリマを接続し、プラスの方にポテンショメータを取り付ける。トリマからの出力を0[deg]での電圧値に近づけることで、0[deg]での電圧を0[V]付近に落とすことができ、有効な電圧増幅が可能となる。このような回路を用い、出力をH8へ送信することで角度情報を得る。

実際のA/D変換値としては、5[V]を1024としたとき、

0[deg] : 12(0.059[V])  
 90[deg] : 440(2.15[V])  
 180[deg] : 960(4.69[V])

となるように設定した。

#### 4. 比例制御を含めた伝達関数の同定

製作した実験環境を用い応答を測定し、さらにシミュレーションにてシステムの同定を行った。実験方法としては、C言語にて比例制御の環境を記述し<sup>1)</sup>、動作中はポテンショメータの角度情報を得るようにする。この角度の推移と同じ応答となるようなモデルを、シミュレーション上で同定した。後述するが、モータ単体からの同定は行わず、比例制御まで含めた応答からシステムを同定することとした。

C言語にて構築した比例制御ブロック線図をFig.4に示す。動作を説明する。まず、モータは初期値として180[deg]の角度に設定しておく。また、ステップ入力として、目標角度90[deg]のA/D変換値(440)を与える。動作が開始すると、目標値と現在角度の偏差をゲイン $K_p$ で定数倍し、この値をパルス幅時間と設定する。PWM制御により巨視的な電圧が変化するため、速度制御が行われる。偏差が大きいほどパルス幅が大きくなり、より速い速度で動作する。反対に偏差が少なければパルス幅も小さくなり、動作が遅くなる。この方式で角度偏差と速度を比例させるようにした。

この際、ゲインは任意な値とするが、著しく定常偏差が発生するほど小さい値や、発散するほど大きな値にならないよう、いくつか実験してから設定する。本実験では100をゲインとして用いた。

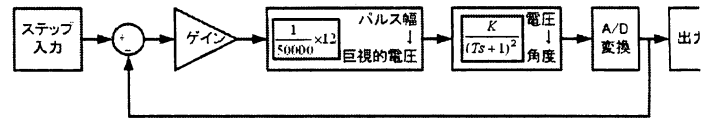


Fig. 4 システム同定のための比例制御概念

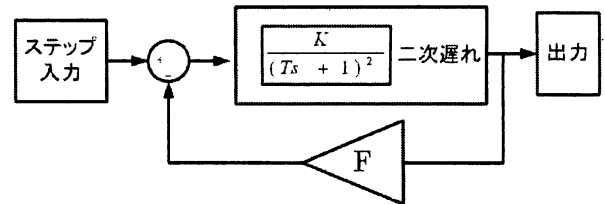


Fig. 5 システム同定モデル

このような環境で応答を測定し、シミュレーションソフト(MATLAB/Simulink)にて制御系まで含めたシステム同定を行った。このソフトはブロック線図を用いてモデルを作成でき、視覚的にわかりやすいモデルの作成を行うことができる。使用した制御は比例制御であり、応答の形から二次遅れであると判断し、モデルとしてFig.5のようなブロック線図を作成し<sup>2)3)</sup>、パラメータを推定することとした。

本来であればFig.4でのプログラムを論理的に考えることで近い応答を計算できるはずであるが、実際には大きな差異が見られた。これは回路やモータに寄生する様々な要素が原因であると考えられ、それらを一つ一つ解析することは困難である。そこで、比例制御であるという最低限の構造を残し、その他の要素は完全にブラックボックスと仮定することで制訂がしやすいと考えたため、Fig.5のようにモデルを設定した。ここで、フィードバックゲインにあたるFは、本来プログラム上では設けていない定数である。この定数は、すべての伝達関数をまとめた際の、二次遅れ系の分母の定数項にのみ影響を与える。この部分を独立して調整するために、便宜的に定数を設けた。

このシステムの中で、パラメータK, T, Fを変

化させ、応答が実測値と近くなる値を制定することでシステム同定とした。

実際に得られた結果とシミュレーションの比較をFig.6に示す。行過ぎ量のピーク、到達時間、定常値などにおいて実測値と近い応答を設定した。しかし、実測値の方は2度の行過ぎ量を経たのち定常状態となっているが、シミュレーションではその後もわずかな振動が残っている。行過ぎ量のピークを合わせようとするとうとうとも発生してしまうものであり、原因はモータの特性であると考へた。即ち、実測値ではモータが動作できる電圧を下回ると、その中で電圧の変動があつても回転せず角度に影響を及ぼさないため、2度の行過ぎ量で定常となつたと考へられる。

## 5. ゲインを変化させた比例制御の応答

様々なゲインを設定し比例制御を行い、シミュレーションの評価を行う。特に、最適ゲインを設定したときにどれほどの誤差があるのかを確認する。ここで、最適ゲインとして、オーバーシュートが2度発生しない範囲で速応性の高いゲインを設定した。

まず、プログラム上で適当なゲインに変化させた実測値と、上記の伝達関数のゲイン部分を変化させたシミュレーション結果を比較してみたが、再現性のある結果は得られなかつた。

そこで、間に最適ゲインが含まれるであろう2つのゲインの範囲を推測し、伝達関数をそれぞれ同定する。さらに、得られた結果のK, T, Fのそれぞれのパラメータを比例ゲイン $K_p$ に関する一次関数と考へることとする。これにより、少なくともその範囲内では、測定結果と大きくずれない伝達関数を得られると考へた。

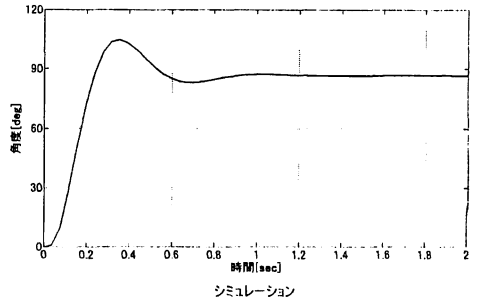
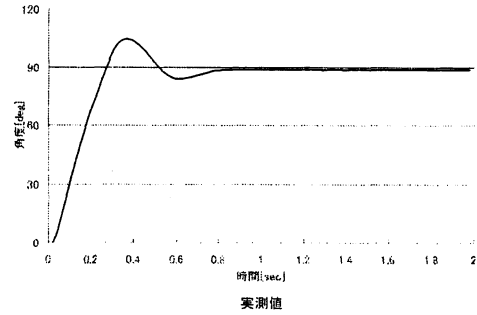


Fig. 6 制御系まで含めた実測値と同定した応答

具体的な制定方法を述べる。まず、大きい方のゲインは前章で使用した $K_p=100$ を用いる。また、小さい方のゲインは $K_p=40$ とした。

$K_p=100$ のとき、伝達関数の各パラメータは

$$\begin{aligned} K &= 4.5 \\ F &= 0.82 \\ T &= 0.21 \end{aligned}$$

であつた。また、 $K_p=40$ のとき、伝達関数の各パラメータは

$$\begin{aligned} K &= 2.9 \\ F &= 0.62 \\ T &= 0.21 \end{aligned}$$

であつた。この中で、Tの値は同じであつたため比例ゲインに影響されないと考へた。

KとFの値については、比例ゲイン $K_p$ の一次関数と考へる。即ち、Kに関しては

$$\begin{cases} 100a_K + b_K = 4.5 \\ 40a_K + b_K = 2.9 \end{cases} \quad (1)$$

を解き、Fについては、

$$\begin{cases} 100a_F + b_F = 0.82 \\ 40a_F + b_F = 0.62 \end{cases} \quad (2)$$

の連立方程式を解くことで、一次関数の係数を求める。結果としては、

$$K = 0.027K_p + 1.83 \quad (3)$$

$$F = 0.0033K_p + 0.49 \quad (4)$$

のようになった。ブロック線図で表すと図7のようになる。

上記の式を用い、比例ゲイン $K_p$ を40から100の範囲で10ずつ変化させたシミュレーション結果と実際の測定値の比較をFig.8に示す。どちらもゲインの上昇に合わせて速応性が上がり、振動が大きくなっている様子が確認できるが、ピークの高さや定常値にはシミュレーションとの差が見られる。これは先述したように、偏差の減少によりモータに入力する電圧が小さくなりすぎ、現実的に動作できなくなったことが原因であると考えられる。また、実測値より最適ゲインは80程度であると判断した。

## 6. まとめ

本研究ではモータ制御の基礎的な回路製作と伝達関数の同定の方法を考察した。

特に伝達関数の同定については、理論的な話を追うだけでは制定できず、結果的にブラックボックスとして扱った。理論との誤差を生む原因としては様々考えられるが、現在は速度制御の方法に問題があるとしている。即ち、速度を変えるとき、純粋に電圧を変えているわけではなく、周期を持ったパルス信号を与えている点から、交流成分をもった信号がDCモータに入ること、寄生要素が付加されるのではないかと考えている。加えて、H8より生成されるPWM信号がしばしば乱れる様子を

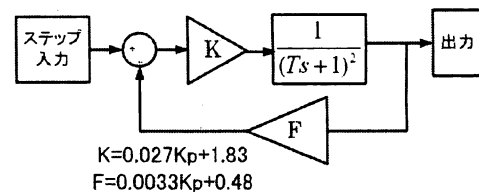


Fig. 7 比例制御ブロック線図

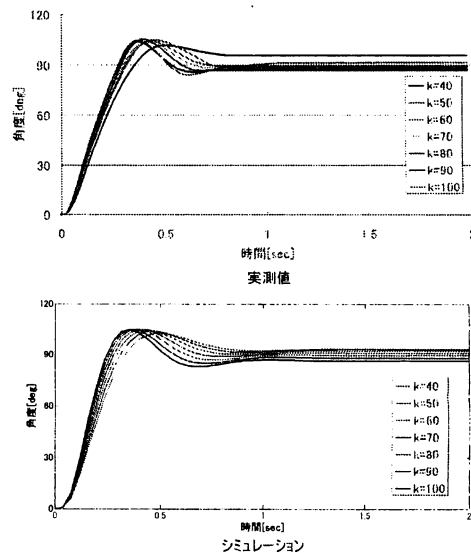


Fig. 8 比例制御実測値とシミュレーション結果

確認しており、これも影響を与えていると考えた。

また、同一のゲインであっても、施行を重ねると他とは異なる応答を起こす例も確認している。外乱による角度信号の急な変化などが原因であると考えている。

今回は比例制御のみであったが、本来はPID制御で一つの信頼できる制御系になるため、今後は積分制御、微分制御についても考察を行う予定である。課題としては、現在比例制御まで含めブラックボックスと考えている伝達関数を、制御対象(モータ)と比例ゲインとに独立させることが挙げられる。積分制御・微分制御は比例制御と並列して配置されるため、比例項の独立が不可欠である。また、微分制御は突発的な外乱に対応する特性上、その評価方法についても考察を行う必要があると考えている。

## 参考文献

- 1) RCサーボをPID制御する実験。 ,  
The Humanoid Robot Wegweiser (オンライン),  
<http://uhiro.fc2web.com/robotone/special/report009.html>, (2009-10-28)
- 2) 熊谷英樹, 大石潔 編著: MATLABと実験でわかるはじめての自動制御, 14/48, 日刊工業新聞社 (2008)
- 3) 三田宇洋 編著 高島博, 宅島章夫, 田中明美  
著: MATLAB/Simulinkとモデルベース設計による二足歩行ロボット・シミュレーション, 33/50, 株式会社 毎日コミュニケーションズ(2007)