

フィードバック制御を理解するための シミュレータとライブラリの開発

奥本 拓哉¹ 清水 智矢¹ 大村 基将¹ 疋田 真一¹ 兼宗 進¹

概要：現在、中学校等の計測・制御の学習では、決められた動作を順に行うシーケンス制御と、ライントレース等の基礎的なフィードバック制御が扱われている。フィードバック制御の題材としては「ラインから外れたかどうか」を2値で扱うライントレースが多い。一方、実社会では「外れているか否か」という真偽値だけでなく、「どれくらい外れているか」という連続値を扱う制御も多く使われている。そこで本研究では古典制御理論のひとつであるPID制御に着目し、目標値との差分を出力に反映するP制御（比例制御）を画面のシミュレーションにより理解したり、実機に転送して動作を確認できる教材を開発した。

キーワード：フィードバック制御, PID 制御, シミュレーター, ドリトル

OKUMOTO TAKUYA¹ SHIMIZU TOMOYA¹ OMURA MOTOMASA¹ HIKITA SHIN'ICHI¹
KANEMUNE SUSUMU¹

1. はじめに

中学校技術・家庭（技術分野）では、平成20年7月に公表された学習指導要領より「プログラムによる計測・制御」の学習内容を必修とした。本学習では、フローチャートなどの図示表現を交えての順次・反復・分岐等の構造化文の学習や、センサやアクチュエータをもつ装置の制御実習などが行われている。実習ではライントレースカーの制御が取り扱われることが多く、その内容は、走行ラインを「白と黒」の2値で判別する二値化制御によるフィードバック制御が題材とされる傾向がある [1] [2]。ただし、生徒らを取り巻く製品には、2値化制御以外にもPID制御理論などの制御理論が活用されているケースも多い。このため、中学生における「プログラムによる計測・制御」においては、複数の制御理論に生徒らが触れ、それぞれの制御理論の特徴などを直感的に理解できる仕組みが必要であると考えた。

そこで、本研究では、アルゴリズムや制御方式による違いを可視化する制御教材を提案する。本教材は、ライントレースカー教材をコンピュータ上に再現するシミュレータと、プログラミングによりシミュレータ上のライントレースカーを制御するためのライブラリからなる。シミュレー

タは、単体でも制御方式、目標値、操作量などの違いを確認できる。さらに、ライブラリを利用して制御プログラムを記述することで、シミュレータ上でプログラムの動作を確認できる。

本稿では、教材が対象とするフィードバック制御の概要と、提案するシミュレータとライブラリの設計方針、および、プロトタイプの開発状況について報告する。

2. フィードバック制御について

フィードバック制御系の基本構造を図1に示す。フィードバック制御とは、フィードバックによって制御量を目標値と比較し、それらを一致させるように操作量を生成する制御である [8]。コンピュータが操作量を算出し、アクチュエータを制御する。制御した結果をセンサで計測し、計測結果と目標値を比較し、再度一連の処理の流れを行う。フィードバック制御系の制御方式は様々、存在する。

2値化制御は、入力大きさによって、出力が二つの定まった値のどちらかをとる制御動作である。現在値と目標値の偏差が「+か-」で判別して、「+」の状態の時の操作量と「-」の状態の時の操作量を使い分ける決定する制御方式である。

P制御は、目標値と現在値の偏差を比例ゲイン k_p に積算して操作量を決定する制御方式である。特徴として、操作

¹ 大阪電気通信大学
Osaka Electro-Communication University

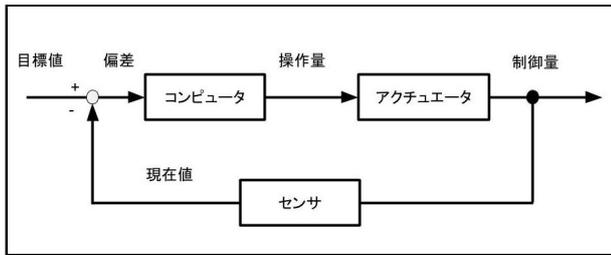


図1 フィードバック制御の基本構造

量が偏差に比例しているため、偏差が大きい場合は操作量が大きくなり、偏差が0に近づくほど操作量も0に近づき偏差を減らすことができる。

2 値化制御は簡単な制御方式のため、P 制御に比べて処理速度が速いのが特徴である。しかし、P 制御は目標値と現在値との偏差を収束していくため、処理速度が遅いが偏差をなくし安定した制御が可能と言える。

3. 提案教材

3.1 計測・制御教材における課題

計測・制御の学習として、ロボット等の実機を用いたシーケンス制御やフィードバック制御を行う場合、3つの課題がある。

1 つ目は、ロボットの動作精度に関する問題である。教材のロボットは自身の動作精度や環境などの要因により、ロボットがプログラム上で想定した動きと異なる動きをしたり、再現性のない動作を行ってしまう場合がある。ロボットの自車位置の推定や移動精度に着目した報告としては、紅林らのロボットの移動量から自車の位置を推定するロボット教材 [6] や西ヶ谷らの距離を計測するロボット [5] などがある。これらの教材は単純なシーケンス制御の範囲であれば、自車位置や移動量を十分な精度で得ることができた。しかし、連続的かつ微細な移動量や位置を検出する必要があるフィードバック制御では、精度に課題がある。精度が十分でない場合、ロボットが想定した動作にならない原因が、制御プログラムにあるのか、ハードウェアの問題であるのか特定しにくくなるなどの問題が予想される。

2 つ目は、制御方法の違いによる動作の変化を比べにくいことがある。生徒が制御プログラムの改良やアルゴリズムの変更など実施した時、その変更がロボットにどの程度の変化を与えたかは目視での確認に頼ることになる。差異が小さいほどその変化を認識することは難しい。したがって、改良等によって本来は異なる動作をしているにもかかわらず、生徒がその変化を知覚できないなどの問題が予想される。

3 つ目は制御理論のプログラム記述が難解な場合である。制御理論のいくつかには高等数学の理解を要する場合がある。したがって、どのような制御がしたいかが明確にもかかわらず、既有知識の有無によりプログラムでの表現がで

きない場合がある。

以上から、我々は、様々な制御方式に対応する計測・制御教材には、それぞれ以下の要素が必要であると考えた。

- (1) 制御プログラムの命令に従い、制御対象が正確に動作する
- (2) 制御方式等による動作の変化を認識できる
- (3) 制御理論を抽象化してプログラム上に表現できる。

3.2 提案教材の設計

今回は、ライントレースカーをシミュレートする環境と、シミュレータ上で制御プログラムを記述するためのライブラリを開発する。ライントレースカー開発は、コースの黒と白をセンサで識別し、コースに沿って走行するために、アクチュエータの操作量を状況に応じて調整をおこなう、計測・制御の中でも代表的な教材の一つである。また、制御方式による制御内容の変化がわかりやすいことから、ライントレースカーを制御対象とした。

実際のロボットが制御プログラムの命令に従い、正確に動作する教材は、工作精度を高めセンサ等を多用することで可能となる。しかし、コストの問題等により用意できる実機数の制限が発生することが想定された。また、実機での実習では、プログラム転送時のエラーやロボットのハード面での不具合も多い [10]。そこで、本教材は、コンピュータ内にライントレースカーを再現するシミュレータとすることにした。シミュレータとすることで、センサに影響を及ぼす外乱、アクチュエータの個体差、および、ギアボックス等の機構の工作精度などの影響を、考慮する必要がなくなる。これにより、生徒が純粋な制御アルゴリズムの検討のみに注力することができるようになると考えた。シミュレータとすることで、制御中の自車位置のみならず、走行軌跡、動作中の制御量、偏差、現在地なども得ることができる。この結果を用いて、視覚的にも、数値的にも制御方式の違いによる動作の違いを分析することができるようになる。

また、シミュレータ上のライントレースカーの制御プログラムについては、極力、制御方式の特徴や違いに気づくところから学習が進められるように配慮する必要がある。このため、各制御方式の制御量の計算などを抽象化し、単純な命令実行のみで実現できるライブラリをあらかじめ提供することにした。なお、本教材は中学生を対象としていることから、制御に用いるプログラム言語については教育用プログラミング言語「ドリトル」[7]で開発できるようにした。

4. ライントレース・シミュレータ

4.1 シミュレータの設計

ライントレース・シミュレータは、生徒が作成したライントレースカーの制御プログラムをコンピュータ上で再現するための仕組みである。ライントレースの再現のためには、走行コース、制御対象となる車体、および、コースを

検出するセンサ（以後、擬似センサ）を再現する必要がある。制御方式毎の動作の違いや制御方式の特徴は、走行するコースの形状により特徴的に現れたり、他の制御方式と同じように現れたりする。この確認作業を、学習者が主体的におこなえるようにするため、再現するコースの設定は学習者が容易に行えることが望ましい。そこで、本シミュレータでは、ボタン操作により画面上に線を引き、その線をコースとして設定できるようにすることで、任意のコースでライントレースを実施できるようにした。また、センサによるコースの検出については、センサを模した図形とコースを模した線との重なり量を計測した現在値とする方法を採用した。一般的なライントレースカーでは、センサとして赤外線センサを用いている。赤外線センサは計測範囲の赤外線反射量を現在値とする。赤外線センサでの現在値とは、コース部分の反射量とコース外での反射量の総和であることから、擬似センサと赤外線センサの計測結果は似通った特性となる。そして、この擬似センサを図形オブジェクトの要素として持たせたものを車体とした。これにより、ドリトルで図形描画したオブジェクトを動かすのと同じ要領で、車体の制御プログラムの記述が可能になる。

学習者が制御プログラムを作るにあたっては、利用する制御方式の特徴などを理解しておく必要がある。このため、プログラムを書かず、制御方式、目標値、およびゲイン等のパラメータを設定するだけで、ライントレースシミュレーションをできる機能を追加した。

上記の設計方針を踏まえ、必要とする機能を以下と定めた。また、これら機能を踏まえて開発したシミュレータの実行画面を図 2 に示す。

- 擬似センサ機能
- コースの描画機能
- 制御方式の選択機能
- ロボットの軌跡が見える機能
- 現在値・偏差・制御量の情報が見える機能
- プログラムから確認するためのライブラリ

擬似センサ機能は、センサを模した図形とコースを模した線との重なり量から、制御に用いる現在値を提供する機能である。コースの図形に対してセンサの図形が衝突した時、図形同士の重なりをピクセル数で取得し、取得したピクセル数とセンサ図形全体のピクセル数を割算し 255 段階で数値を返すようにした。

制御方式の選択機能は、ボタンを押すことで各制御方式の動作をシミュレートする機能である。2 値化制御では、現在値と目標値を比較し、ロボットが右回転、左回転を行う。P 制御では、式 (1) により操作量を計算する。

$$\text{操作量} = K_p \times (\text{目標値} - \text{現在値}) \quad (1)$$

シミュレータで 2 値化 (図 3)・P (図 4) の実行結果を示す。ロボットの軌跡を確認することで 2 値化制御では振幅が収

まっていないことや、P 制御では直線上を時間が追うごとに収束していることがわかる。

ロボットの軌跡が見える機能は、ロボットが動いた軌跡を表示する機能である。

現在値・偏差・制御量の情報が見える機能は、フィードバック制御中の、現在値、目標値、偏差、操作量、および、目標値を表示する機能である。

なお、パラメータ設定によるシミュレーションは次のステップで実施することができる。

- (1) ライントレースを動かすプログラムを作成および実行する
- (2) コースの描画する
- (3) ロボットの開始位置を設定する
- (4) ゲインを調整する
- (5) 「2 値」または「P (P 制御)」のボタンを押す

5. ライブラリ

シミュレータ上でのライントレースの制御プログラムにおいては、制御量の計算に必要なゲイン・目標値・現在値を用いて計算する必要がある。これを各制御方式で共通化するため、計算処理を制御方式ごとのオブジェクトとした。これに計算に必要な情報をパラメータとして与える。現在値に関しては、状況によって変動するため、繰り返し処理の中に記述するようにした。また、操作量の限度指定に関しても制御方式に対してパラメータを与えることで設定できるようにした。図 5 に、2 値化制御のプログラムを示す。

ライブラリを使用するため、1 行目に「システム! Simulator”使う。」を記述する。2 行目では 2 値化制御の計算処理用のオブジェクトを生成し、制御という名前にしている。3 から 5 行目では、制御量の最大値と最小値、計算に必要な目標値を設定している。8 行目と 12 行目が実際にロボットカーに対しての処理命令である。10 行目ではセンサの計測値を取得している。11 行目では現在値をパラメータとして与え、操作量を計算している。

図 6 に、ライブラリを使用した場合の P 制御のプログラムを示す。P 制御のプログラムでは、2 行目で P 制御の計算用オブジェクトを生成している。6 行目では、P 制御に必要なゲインの設定をしている。11 行目では現在値をパラメータとして与え、操作量を計算している。

6. おわりに

フィードバック制御系の「2 値化」と「P」の動作を確認できるシミュレータをドリトルで開発した。ドリトルで記述したライントレースのプログラムをシミュレータ上でシミュレーションできるほか、2 値化制御と P 制御の動作を比較することが可能である。開発した教材を活用することで中学校や高等学校での計測・制御で P 制御を扱うことが可能になると期待している。

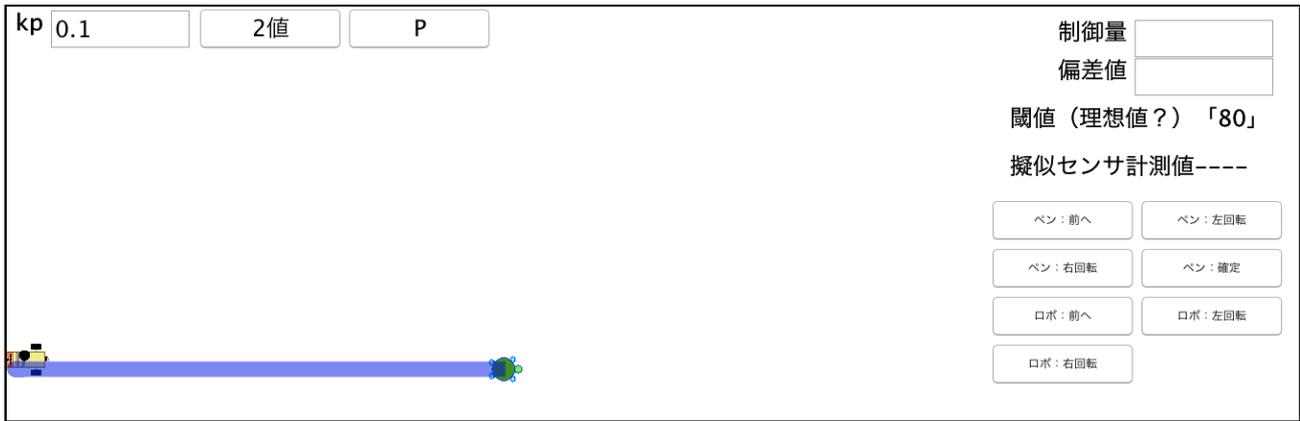


図 2 シミュレータの実行画面

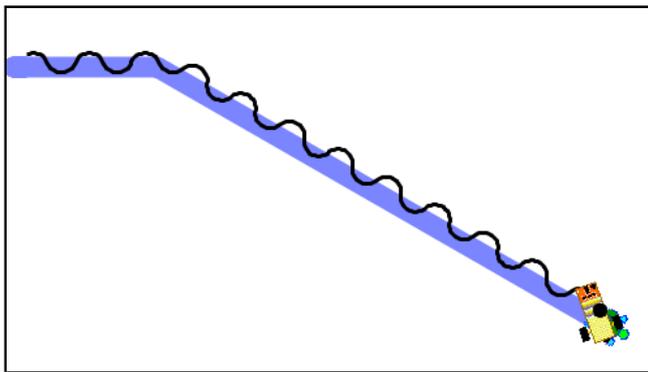


図 3 2 値化制御の実行結果

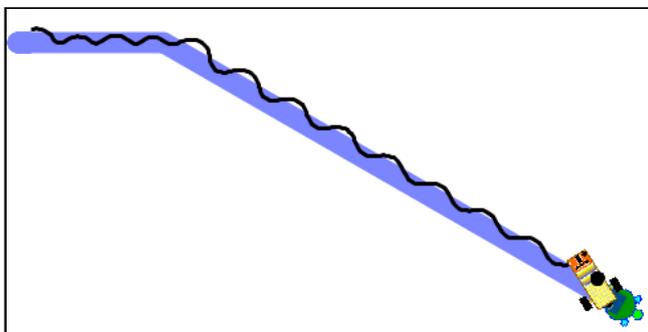


図 4 P 制御の実行結果

```

1 システム! "Simulator" 使う。
2 二値化=2値化制御! 作る。
3 二値化! (50) 最大値。
4 二値化! (-50) 最小値。
5 制御! (80) 目標値。
6
7 「
8     ロボットカー! 前進。
9     現在値=センサ! 読む。
10    操作量=制御! (現在値) 計算。
11    ロボットカー! (操作量) 左回転。
12 」! 繰り返す。
    
```

図 5 2 値化制御のプログラム

```

1 システム! "Simulator" 使う。
2 P=P 制御! 作る。
3 P! (50) 最大値。
4 P! (-50) 最小値。
5 P! (80) 目標値。
6 P! (0.1) ゲイン設定。
7
8 「
9     ロボットカー! 前進。
10    現在値=センサ! 読む。
11    操作量=P! (現在値) 計算。
12    ロボットカー! (操作量) 左回転。
13 」! 繰り返す。
    
```

図 6 P 制御のプログラム

[2] 佐竹隆顕:技術・家庭 技術分野, 教育図書株式会社 (2014)

[3] 文部科学省:中学校学習指導要領, 2008

[4] 文部科学省:高等学校学習指導要領, 2009

[5] 西ヶ谷浩史, 青木浩幸, 井上修次, 江口啓, 紅林秀治. 自律型 3 モータ制御ロボット教材を用いた計測の授業. 情報処理学会研究報告. Vol.2009-CE-098, No.17. (2009)

[6] 紅林秀治, 高山大輝. ボール式マウスを用いた位置を把握できる教材用自律型移動ロボットの開発. 日本産業技術教育学会誌. 2011, Vol.53, No.4, pp.243-253.

[7] プログラミング言語「ドリトル」.
<http://dolittle.eplang.jp/>

[8] 日本工業規格 JIS Z8116-1994.
<http://kikakurui.com/z8/Z8116-1994-01.html>

[9] 紅林秀治, 井上修次, 江口啓, 鎌田敏之, 青木浩幸, 兼宗進. 自律型 3 モータ制御ロボット教材の開発. 日本産業技術教育学会誌, Vol.51, No.1, pp.7-16 (2009).

[10] 井戸坂幸男, 久野靖, 兼宗進. 自立型ロボット教材の評価と授業. 日本産業技術教育学会誌, Vol.2010, No.532010

[11] 紅林秀治, 井上修次, 江口啓, 鎌田敏之, 青木浩幸, 兼宗進. 自律型 3 モータ制御ロボット教材の開発. 日本産業技術教育学会誌, Vol.51, No.1, pp.7-16 (2009).

[12] 和田佑介, 竹川貴博, 仲石資紀, 兼宗進. フィジカルデバイスを利用した大学での組込ソフトウェア教育. 情報処理学会, 第 73 回全国大会, Vol.2011, No.1, pp.465-466 (2011).

参考文献

[1] 加藤幸一, 永野和男:新しい技術・家庭 技術分野, 東京書籍株式会社 (2014)