

# 初学者を対象としたロボット制御プログラムの開発支援フレームワーク

菅野 志穂<sup>†</sup> 武田 敦志<sup>†</sup>

ロボット制御を題材としたプログラミング教育では作成するプログラムの目的や効果が明確であるため、ロボット制御はプログラミング初学者を対象とした導入教育の教材として最適である。しかし、ロボット制御のプログラムを作成する場合、ロボットの制御方法だけではなく、プログラムの制御構造やデータ構造を考える必要があるが、これらのすべてを考えることはプログラミング初学者にとって簡単なことではない。そこで、本稿では、開発者がロボットを動かすための制御プログラムを作成することなく、状況に応じて最適なパラメータを入力するだけでロボットに目的の動作を行わせることができる開発支援フレームワークを提案する。この開発支援フレームワークでは、ロボット制御のための基本的な制御構造をフレームワークの機能として提供し、設定パラメータを変更することによりロボットの動作を変更することを可能とする。このフレームワークを使うことにより、プログラム開発者は制御構造に関するプログラミングを必要としないため、ロボットの制御方法に注目して開発を進めることができる。

## Development Framework of Robot Control Program for Programing Beginners

Shiho Kanno<sup>†</sup> Atsushi Takeda<sup>†</sup>

Robot control programing is a good practice for programing beginners because a target of the program is clear and it is easy to understand about the results. Robot control programing, however, is not easy for programing beginners because programmers must consider algorithms and data structures as well as robot control. Therefore, in this paper, we propose a development framework of robot control program for programing beginners. Our framework provides an algorithm of robot control, so programmers must only set the parameters in order to change the robot's actions. Therefore, programmers do not have to consider algorithms, so they can focus on robot control scheme.

## 1. はじめに

ロボット制御を題材としたプログラミング教育では作成するプログラムの目的や効果が明確であるため、ロボット制御はプログラミング初学者を対象とした導入教育の教材として最適である。そのため、小中学生を対象としたプログラミング体験イベント[1]や高校生を対象としたプログラミング講座[2]の題材としてロボット制御が採用されている。ロボット制御のプログラムを作成する場合、ロボットの制御方法だけではなく、プログラムの制御構造やデータ構造を考える必要があるが、これらのすべてを考えることはプログラミング初学者にとって簡単なことではない。そのため、プログラミング初学者を対象としたプログラミング教育では、ブロック線図を作成することでプログラムを作成できる支援ツールを使うなどの導入教育の障壁を低くするための工夫がなされている[3][4]。しかし、これらのプログラミング支援ツールはプログラムの制御構造を学習することを目的としており、プログラミング初学者はロボットの制御方法とプログラムの制御構造を同時に学習することになる。

そこで、本稿では、開発者がロボットを動かすための制御プログラムを作成することなく、状況に応じて最適なパラメータを入力するだけでロボットに目的の動作を行わせることができる開発支援フレームワークを提案する。一般的なロボット制御プログラムは、ロボットの周りの状況を認識し、その状況に適した動作を繰り返し実行することでロボット制御を実現している。そこで、本稿で提案する開発支援フレームワークでは、上記のロボット制御のための基本的な制御構造をフレームワークの機能として提供し、設定パラメータを変更することによりロボットの動作を変更することを可能とする。このフレームワークを使うことにより、プログラム開発者は制御構造やデータ構造に関するプログラミングを必要としないため、ロボットの制御方法に注目して開発を進めることができる。これは、プログラムの制御構造やデータ構造に対する理解が低いプログラミング初学者がロボットの制御方法を学ぶために適した開発支援フレームワークであると考えられる。本稿では、提案する開発支援フレームワークが提供するロボット制御のための基本的な制御構造を述べ、それに設定するパラメータについて説明する。また、ET ロボコンにおける小型ロボットの制御プログラムを題材とし、上記のフレームワークを用いた制御プログラムによって目的の制御が行うことができることを検証する。

## 2. ロボット制御プログラムの開発支援フレームワーク

### 2.1 開発支援フレームワークの概要

現実世界でロボットを正確に動かすためには、ロボットのまわりの状況を判断しそ

<sup>†</sup> 東北学院大学教養学部情報科学科  
Department of Information Science, Tohoku Gakuin University

の状況に応じた動作を指示する必要がある。一般的なロボット制御プログラムでは、常に変化するまわりの状況に対応するため、上記の状況判断と動作指示を繰り返す制御構造となっている。ロボットを自律動作させるためのプログラムを開発するためには、上記の制御構造をプログラムとして作成し、そのプログラムで用いるパラメータを設定する必要がある。しかし、ロボット制御プログラムの制御構造には共通化や再利用が可能な部分が多く含まれており、これらの部分をライブラリとして提供する開発支援フレームワークを用いれば従来よりも容易に制御プログラムを開発できる。特に、教育を目的としたロボット制御プログラミングの場合、ロボット制御の目標がことなってもプログラムの基本的な制御構造は同じことが多い。

そこで、本稿では、開発者がロボットを動かすための制御プログラムを作成することなく、状況に応じて最適なパラメータを設定するだけでロボットに目的の動作を行わせることができる開発支援フレームワークを提案する。このフレームワークを用いて開発するプログラムでは、ロボットのまわりの状況に応じた行動を選択し、次に選択された行動を終了条件に達するまで繰り返し、終了条件に達したら行動を終了し再び行動選択をする手順を繰り返し行う。ロボットの制御プログラムの基本構造は変更しないが、設定されるパラメータを変更することにより様々な状況に対応した制御プログラムを実現する。

## 2.2 制御プログラムの基本構造

一般的なロボット制御プログラムは行動選択・行動実行・終了条件の確認を繰り返すことでロボットの動作を制御している。図1にロボット制御プログラムの基本構造を示す。行動選択ではどの行動をするべきか判断するためにまわりの状況を確認し、それに応じて行動を選択する。行動実行では、まわりの状況に応じて行動を実行する。状況に応じてパラメータを変更することでロボットに適切な動作を指示する。終了条件の確認では行動を実行しながらまわりの状況を確認し、行動終了の判断をする。これを終了条件の確認とする。まわりの状況が終了条件に達するまではその前に選択された行動を繰り返し実行し、終了条件に達すると行動選択へ移る。これらの手順を繰り返し行うことで、ロボットはまわりの状況に応じた行動ができる。例えば、ロボットが決められたコースを走行する場合、走っているコースがカーブか直線か、段差があるか、コースの色などのロボットのまわりの状況を判断しその状況に応じた適切な行動を選択する。ロボットが動作している際にも状況を判断し行動を終了する。この手順を繰り返すことでロボットは決められたコースに合わせた動作を行うことができる。

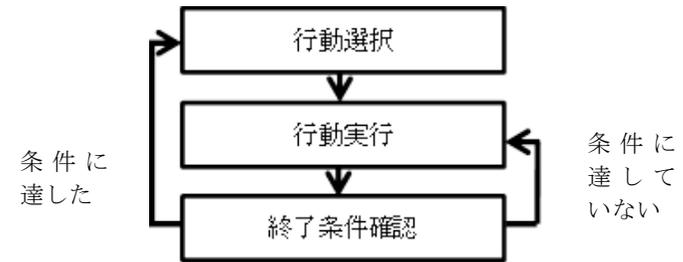


図1 ロボット制御プログラムの基本制御構造

## 2.3 制御プログラムのパラメータの設定

ロボットを制御するためには、状況に応じてロボットのパラメータを変更する必要がある。開発者はロボットの状況の判定に用いる閾値やロボットに付いているそれぞれのモータに与える力などを変更していくことで、ロボットの細かい制御を行っている。ロボットを正確に動かすために、パラメータの調整を正確に行う必要がある。例えば、ロボットに決められたコースをできるだけ早く走行させたい場合、ロボットが走行している場所に合わせた速度を設定することでロボットはより早く走行することができる。ロボットの走行コースに直線とカーブがある場合、直線はできるだけ速度をあげることで早く走ることができる。しかし、カーブの場合は直線と同じ速度で走行するとカーブを曲がりきれず転倒する可能性がある。そのためカーブのときは直線よりも遅い速度で走行させる必要がある。また、カーブの中でも急なカーブとゆるやかなカーブでは速度を変える必要がある。走行場所に応じて、転倒やコースを外れてしまうことなく走ることができる一番早い速度に常に変更していくことでロボットは走行コースを最速で走ることができる。本研究の制御プログラムでは上記のような速度の変更を、ロボットの制御プログラムを変えることなくパラメータのみ変更できるようにする。

## 3. ETロボコンへの実装

本研究ではETロボコンにおけるロボットの制御プログラムを題材とし、本研究での提案フレームワークを用いたプログラムで正確にロボットを動かすことができるかを確認した。ETロボコンとは黒のラインが引かれた決められたコースをスタートからゴールまで走行し、その走行タイムを競うものである[5]。図2に2015年度ETロボコ

ンで使用したコースを示す。2015年度のETロボコンのコースにはRコースとLコースがあり、出場チームのロボットはそれぞれのコースを1回ずつ走行し、2つのコースの走行タイムの合計からボーナスタイムを引いたタイムで競う。ボーナスタイムはゴール後に用意された難所と呼ばれるコースをクリアすることで獲得できる。難所には、Rコースのゴールの後に用意されているフィギュアLというものがある。フィギュアLとは図3のような黒のラインが引かれた木の板で、この板がコースの上に置かれている。図3の④側からロボットは段差を上り、板の上で一回転し図3の③側から降りることによってクリアとなる。

また、ETロボコンで使用するロボットは図4に示す小型の2輪ロボットである。左右のモータでタイヤをそれぞれ動かすことができ、モータに与える数値を変えることで前進や後退、スピードを変えることができる。後ろのモータではロボットのしっぽの上げ下げを行う。他に傾きを検知するジャイロセンサ、ロボットの下の光の反射を読み取る光センサもついている。これらを使い分けコースを走行していく。

ETロボコンにおける状況判断は、ロボットの走行距離や段差の検知、灰色の検知、指示された行動が終了したかなどで状況を判断していく。状況に応じた行動を実行し、また状況判断により行動を終了させ次の行動に移る。スタートからゴールまでのコースは走行距離やカーブ、直線などを判断してそれぞれに応じたパラメータで走行していく。ゴール後の難所のひとつであるフィギュアLをクリアするための設定パラメータを図5に示す。図1で示した制御プログラムと図5に示すパラメータを用いることによりフィギュアLをクリアすることができた。



図3 フィギュアL

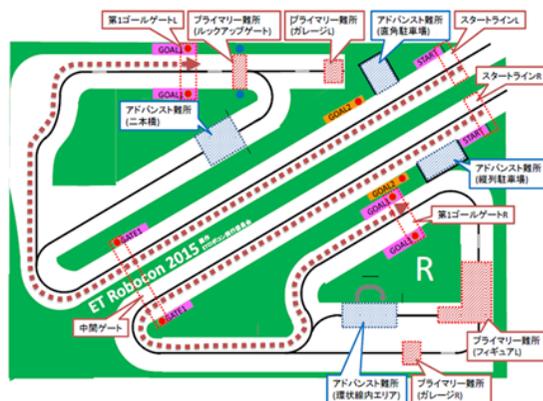


図2 ETロボコン走行コース



図4 ETロボコンで使用するロボット  
 左：前面 右：後面

## 参考文献

- 1) 中井 智己, 内山 豊, 水島 聡哉, 富永 浩之: 小中学生への LEGO ロボットのプログラミング体験イベントの運営方法の検討とゲーム課題の再構成, 情報教育シンポジウム 2015 論文集, Vol.2015, pp.67-72 (2015).
- 2) 高橋 知希, 富永 浩之: 高大連携の LEGO プログラミング講座におけるゲーム課題とシミュレーション教材, エンタテインメントコンピューティングシンポジウム 2013 論文集, Vol.2013, pp.301-304 (2013).
- 3) 四折 直紀, 宮内 悠仁, 杉原 一臣, 大熊 一正, 山西 輝也, 魚崎 勝司: マイクロロボットとスクラッチによる組込みシステムの学習に向けて, 第 74 回全国大会講演論文集, No.1, pp. 651-652 (2012).
- 4) 井戸坂 幸男, 兼宗 進, 久野 靖: 中学校における自律型制御ロボット教材の評価と授業～新学習指導要領の「計測・制御」授業に向けて～, 研究報告コンピュータと教育, Vol. 2010-CE-103, No.22, pp.1-7 (2010).
- 5) ET ロボコン <http://www.etrobo.jp/>

| スタート    | 行動内容          | 終了条件(状況判断) | 終了処理  |
|---------|---------------|------------|-------|
| 初期(f=0) | ラインを読み取って前進   | 段差検知       | f = 1 |
| f = 1   | 前進速度をあげる      | 段差を上った     | f = 2 |
| f = 2   | しっぽを下す        | しっぽが下りた    | f = 3 |
| f = 3   | 回転            | 360度回った    | f = 4 |
| f = 4   | ロボットをラインの上に戻す | ラインの上に戻った  | f = 5 |
| f = 5   | ラインを読み取って前進   | 指定した距離を走った | f = 6 |
| f = 6   | しっぽを下す        | しっぽが下りた    | f = 7 |
| f = 7   | 真っ直ぐ前進        | 指定した距離を走った | f = 8 |
| f = 8   | しっぽを上げる       | しっぽが上がった   | f = 9 |
| f = 9   | ロボットをラインの上に戻す | ラインの上に戻った  | 終了    |

図 5 フィギュア L 攻略のための設定パラメータ

## 4. おわりに

ロボット制御を題材としたプログラミング教育では, 学習者はロボットの制御方法だけではなく, プログラムの制御構造やデータ構造を考える必要があるが, これらのすべてを考えることはプログラミング初学者にとって簡単なことではない. そこで, 本稿では, 開発者がロボットを動かすための制御プログラムを作成することなく, 状況に応じて最適なパラメータを入力するだけでロボットに目的の動作を行わせることができる開発支援フレームワークを提案した. 提案した開発支援フレームワークでは, ロボット制御のための基本的な制御構造をフレームワークの機能として提供し, 設定パラメータを変更することによりロボットの動作を変更することを可能とする. このフレームワークを用いることにより, プログラム開発者はロボットの制御方法に注目して開発を進めることが可能となる. また, ET ロボコンにおける小型ロボットの制御プログラムを題材とし, 上記のフレームワークを用いた制御プログラムによって目的の制御が行うことができることを検証した. 今後の課題として, 提案した開発支援フレームワークを用いて初学者教育を行うための教育用アプリケーションや教材の開発が挙げられる.