

ロボットプログラミング学習支援環境の開発

Development of a support environment for learning robot programming

吉田恵美†
Megumi Yoshida

早川栄一‡
Eiichi Hayakawa

1. はじめに

ロボットは、学習者の興味を持ちやすいことから、プログラミング教材として多くの教育機関で用いられている。その一方で、学習者にとって動作とプログラムとの関係が把握しにくいという問題がある。

この問題を解決するには、学習者がデバッグ、テストなどを容易に行える環境を構築する必要がある。また、今日のプログラム言語教育でオブジェクト指向が用いられはじめているが、これらをロボットに応用した場合のサポートはない。

本研究の目的は、オブジェクト指向言語を用いたロボットプログラミングの学習支援環境を提供することである。本研究の対象者はオブジェクト指向の概念を学習済みの学習者であり、情報工学科2～3年生を対象とする。

2. 問題分析

本学情報工学科3年生のカリキュラムである情報工学実験Ⅱでは、自走ロボットを用いたライントレースや経路探索を行っている。この実験から、既存のロボットプログラミングの問題点は次のとおりだと考えた。

(1) ロボットの動作とプログラムを対応づけられない

この原因は二つあり、一つ目は外部環境に影響されたセンサなどの入力値の変化を類推できないこと、二つ目は複雑なソースのトレースに慣れていないことである。

(2) デバッグがしづらい

ロボットの表示デバイスは貧弱であり、学習者が見たい変数やセンサの値をすべて表示することはできない。

(3) テストするのに手間がかかる

通信にシリアルケーブルや赤外線を用いるものが多く、テストに手間や時間がかかり学習者のモチベーションが下がってしまう。また、実機がないとテストが行えない。

3. 設計方針

これらの問題を解決するために、次のような方針で設計を行う。

(1) 動作とセンサの値とソースを対応付ける

ロボットの動きを PC カメラで撮影し、動画として保存する。その動画、ソース、センサの取得した値（以下センサ入力値）を時間軸で対応付けることにより、センサ入力値の変化の類推やソースのトレースの手助けを行う。

(2) 表示用 API を提供する

ロボットの表示デバイスの変わりに、PC 側のディスプレイに表示する表示用 API を提供する。標準出力 API と同じインタフェースにすることで、チェックプリントによる

† 拓殖大学大学院工学研究科,
Graduate school of Engineering, Takushoku University

‡ 拓殖大学情報工学科,
Faculty of Engineering, Takushoku University

デバッグが可能となる。

(3) テストを行うためのシミュレータを提供する

ロボットが走行時に取得したセンサの値を、シミュレータの入力値として扱える。

4. システム構成

4. 1. 全体構成

本システムは、開発環境と走行ロボットにそれぞれモジュールを置くことで構成し、図 1 にその構成図を示す。ロボット走行とシミュレータ利用時の手順があり、それぞれの例を次に示す。

(1) ロボット走行時

学習者は、実行させるユーザプログラムを指定し、カメラの撮影及びロボットの走行を開始する指示を与える。

学習者が上記の操作を行うことで、制御モジュールが動画撮影とロボットの走行を開始し、ロボットからセンサの値などのログデータを取得し、ファイル管理モジュールへ渡す。ファイル管理モジュールは、ログデータをファイルに保存し、ロボットの走行終了後にこれらの項目を学習者に見せる。

(2) シミュレータ利用時

学習者は、実行させるユーザプログラム及び、入力値が書かれたファイルを指定する。学習者から指定されたファイルを、シミュレータモジュールが読み込み、結果を学習者に表示する。

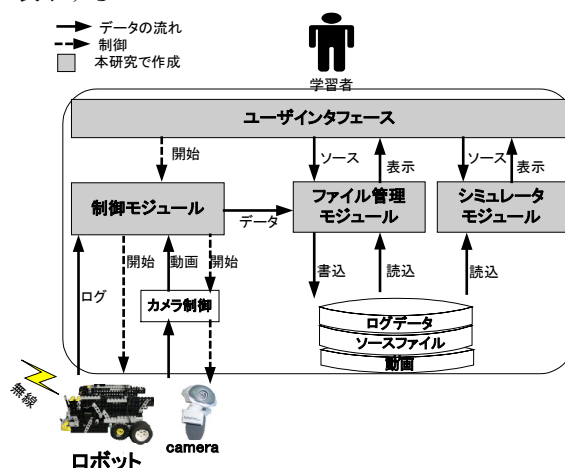


図 1 全体構成図

4. 2. 開発環境の詳細設計

開発環境の設計について述べる。

4. 2. 1. ユーザインタフェース

ソース、動画、ログデータを時間軸で対応付け表示する。ユーザインタフェース（以下 UI）の表示例を図 2 に示す。

(1) ログデータ

ログデータとは、ロボットから送信されたセンサやモータの値であり、この表示方法は表 1 のとおりである。また、

動画との対応付けはグラフに縦のラインをつけることで表示する。

表 1 ログデータの表示方法

I/O の種類	表示方法
すべて	値のみ
フォトセンサ	グラフ、色の濃淡
タッチセンサ	グラフ、push か release
モータ	グラフ、向き及び ON/OFF

(2) ソース

表示用 API が実行されたとき、マークが表示される。

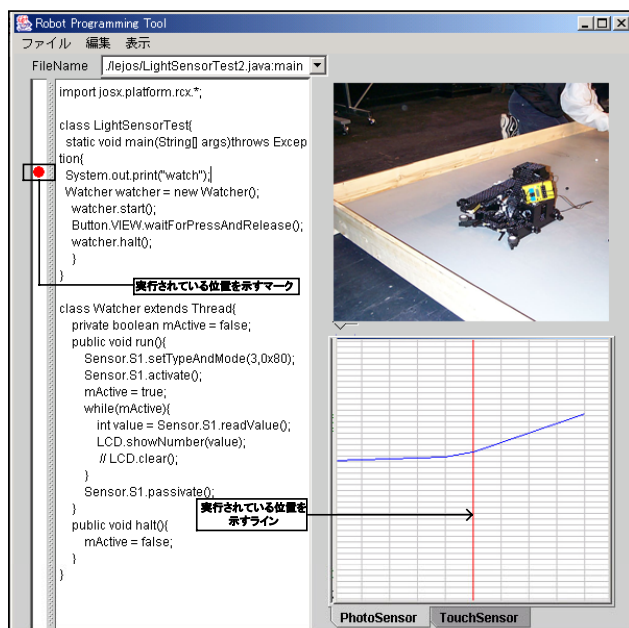


図 2 ユーザインタフェースのイメージ

4. 2. 2. 制御モジュール

ロボットとの通信、制御を行うためのモジュールであり、機能は次のとおりである。

(1) ロボットへ走行開始の命令を出す

開発環境のファイル管理モジュールの準備が終わるまでロボットを待機させ、終わり次第ロボットに走行命令を出す。

(2) ロボットからログデータを受信する

ロボットから送信されたログデータを受信し、ファイル管理モジュールへ渡す。

(3) ロボットと PC の時刻を合わせる

動作、ソースなどを時間軸で対応付けるため、ロボットと開発環境の時刻を合わせる必要がある。

4. 2. 3. ファイル管理モジュール

学習者から指定されたソース、PC カメラで撮影した動画、制御モジュールから送信されたログデータの保存及び管理を行う。また、UI の指示に従いファイルの読み込みを行い、それらを時間軸で対応付け表示する。

4. 2. 4. シミュレータモジュール

シミュレータは、学習者が自宅でプログラミングを行う場合に利用することを前提とし、その設計は次のとおりである。

(1) 仮想入出力 API を提供

学習者がロボットを制御するために使う入出力 API と同じインタフェースを提供することで、プログラムを書き換えなくてもシミュレーションできる。

(2) センサの値の入力方法

センサの値を入力する方法は二通りあり、一つは、対話型に学習者が入力する方法であり、もう一つはロボットが取得したログデータを扱う方法である。

(3) 表示方法

走行時と同じ UI を使い、センサ値、モータ値をグラフなどで表示する。

4. 3. ロボットのハードウェア設計

本システムを実現するには、ログデータの送信と Java の実行が必要なため、自走ロボットは次の条件を満たしている必要がある。

(1) Java VirtualMachine (以下 VM) の搭載が可能なこと

(2) センサの取付けに自由度があること

(3) 無線 LAN の利用が可能なこと

これらの条件を満たすものとして、本研究では図 3 のような構成を考えた。

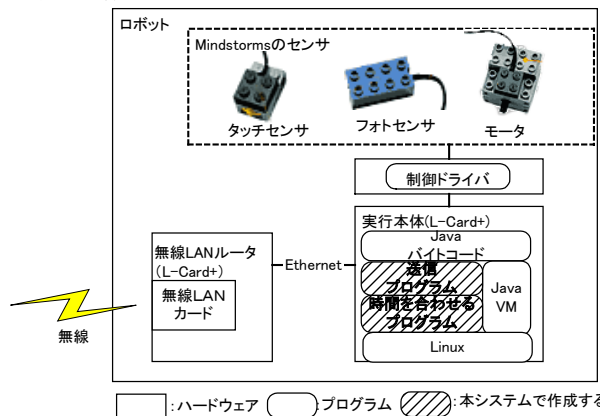


図 3 ロボットの構成図

(1) 無線 LAN ルータ

コンパクトフラッシュ型の無線 LAN カードを搭載した、無線 LAN ルータである。実行本体から Ethernet 経由で届いたログデータを開発環境へ送信する。

(2) ユーザプログラム実行本体

Linux 上に JavaVM を搭載し、ユーザプログラムを実行する。ここにロボット側の送信モジュールがあり、ログデータの送信を行う。

(3) センサとモータ

センサの作成に手間をかけないため、LEGO 社の Mindstorms というロボット用に作られたセンサ類を利用する。

5. おわりに

本稿では、従来のロボットプログラミングの学習環境の問題を述べ、その問題を解決する支援環境の設計について述べた。現在は、プロトタイプを作成を行っており、ユーザインタフェース及び制御モジュールのロボット制御を作成している。今後はプログラム開発をすすめ評価をとっていく。

参考文献

[1] Jose Solorzano:lejOS, <http://lejos.sourceforge.net/>