

ロボットを用いた組込みシステム学習環境「港 ver.R」における シミュレータの開発

川口貴弘[†] 青山誠一[†] 西野洋介[‡] 早川栄一[†]

[†] 拓殖大学 〒193-0985 東京都八王子市館町 815-1

[‡] 東京都立府中工業高校 183-0005 東京都府中市若松町 2-19

Email: ocha@os.cs.takushoku-u.ac.jp, hayakawa@cs.takushoku-u.ac.jp

あらまし 本稿では、ロボットを用いた組込みシステム学習環境「港 ver.R」の概要と、その一環として開発したシミュレータについて述べる。

本研究は、組込みシステム学習をサポートするシミュレータの開発に関するものである。本システムの特徴は、学習にシミュレータを用いることで実機では困難な、動作の再現が容易に行うことができること、ハードウェア自体の不具合の心配がなくなるため、学習者は学習に専念することができる点にある。さらに、ロボットの概観をシミュレートした3Dロボットシミュレータ、ロボットの動作をデバッグするシステムモニタリングツールと連動することで、ロボットが存在しない環境においても組込みシステムの学習ができる。

Development of Simulators for Embedded Systems Learning Environment “MINATO ver.R” using Robot

Takahiro KAWAGUCHI[†], Seiichi AOYAMA[†], Yosuke NISHINO[‡],
and Eiichi HAYAKAWA[†]

[†] Takushoku University 815-1 Tatemachi, Hachioji City, Tokyo, 193-0985, Japan

[‡] Tokyo Metropolitan Institute Fuchu Technical High School 2-19, Wakamatuchou,
Huchu City, 183-0005, Japan

Email: ocha@os.cs.takushoku-u.ac.jp, hayakawa@cs.takushoku-u.ac.jp

Abstract This paper describes the outline of Development of Simulators for Embedded Systems Learning Environment “MINATO ver.R” using Robot and the development of the Simulators.

A characteristic of this system is that reproduction of the movement is possible easily. And one more, it is to be able to devote itself to learning because a learner does not have to worry by trouble of the hardware. Furthermore, the learning of Embedded System without robot enabled it by linking System Monitoring Tool which debugs the movement using log to be provided from a robot and Three-Dimensional Robot Simulator which simulated the general view of the robot.

1. はじめに

近年、携帯電話や情報家電といった組込みシステムの需要が増加しており、その重要性が高くなっている。その一方で、開発技術者数の不足や技術者のスキル不足といった問題があり、技術者の育成が重要な要素となっている^[1]。

組込みシステムを学習する上では、ハードウェアや組込み OS、制御プログラムといった幅広い内容の知識が必要となる。しかし、既存の学習環境の多くは、制御プログラミングに特化したもの^[2]であり、上記に

あげた項目について一貫して学習できるものが少ない。また、学習教材としてロボットやワンチップボードといった実機を用いたものが多く存在するが、実機は動作の再現が難しく、検証したい動作環境を作り出すためには手間が掛かる。

そこで、シミュレータを用いた学習環境に着目した。学習にシミュレータを用いることで場所やハードウェアの有無にかかわらず、PC上で学習できる環境を提供する。さらに、本研究室で開発が進められているロボットを用いた組込みシステム学習環境「港 ver.R」^[3]

と連動することで組込みシステムの学習支援を行う。

2. 問題分析

(1) 連続的に学習可能な環境が不在

従来の実機を用いた組込みシステム学習環境は、制御プログラミングの学習に重点を置いたものが多く組込み OS などのシステム学習を考慮していない。また、学習の項目ごとに異なる教材を用意し、複数の学習環境を用いて学習するなど、一つの学習環境で一貫して学習できる環境が少ない。ここでは、学習者が教材ごとに仕様を覚えなおさなければならず、また、各学習教材で学んだ内容を関連付けるのが困難であり、学習した内容を反映することができない。

(2) 実機を用いた学習教材

組込みシステムの学習において、実機を用いた学習環境は動作を実際に目で見て確認できるため非常に有効である。しかし、その反面、動作の再現が困難である。また、不具合が発生した場合、ハードウェアとソフトウェアの両方をデバックする必要があるため、バグの特定に手間がかかる。さらに、ロボットは、非常に高価であり、すべての学習者に対して提供するためにはコストがかかってしまう。

(3) シミュレータを用いた学習環境

シミュレータを用いた学習環境では、ハードウェアの制約を受けないため、自由度の高い学習が可能である。しかし、デバイスの誤動作を調整するといった、ハードウェア特有の体験をすることができない。

3. ロボットを用いた組込みシステム学習環境

本節では、ロボットを用いた組込みシステム学習環境「港 ver.R」について述べる。

3.1 概要

「港 ver.R」では、ロボットを中心に、学習を支援するツールを用意し、単一の環境で制御プログラミングや組込み OS の学習をサポートする環境を提供する。学習の対象としてロボットを利用することで学習者のモチベーションを高めることができ、ロボットの挙動から、学習者が予定通り動作しているかを留意に確認することができる。用いるロボット本体は、RoboCup サッカー⁽⁴⁾にも用いられているサッカーロボ 915⁽⁵⁾を使用し、ターゲットボードとして SH7709S (SH3) を搭載した CAT709⁽⁶⁾を用いる。このロボットは、光学センサ、タッチセンサ、モータ、LED といったデバイスを搭載しており、CPU の入出力から直接制御で

きるようになっている。ロボットの外観を図 1 に示す。

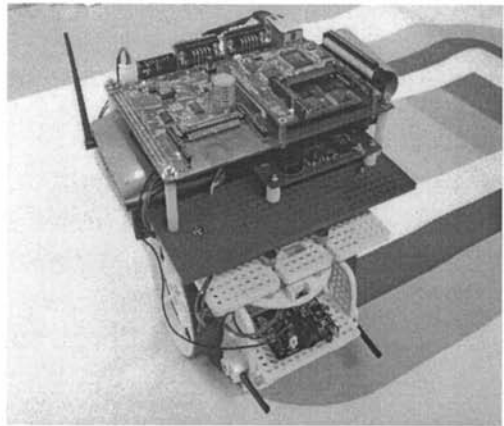


図 1 ロボットの外観

3.2 学習のコースウェア

制御プログラムや組込み OS の学習には、C 言語およびアセンブリ言語を用いる。これは、組込みシステムの開発環境では C 言語が標準的に利用されていることや、プログラムの演習に多く取り入れられていることから採用した。学習者の対象は、前提知識として、OS の講義や C 言語によるプログラミングを学習したことのある人を想定している。学習のコースウェアを図 2 に示す。

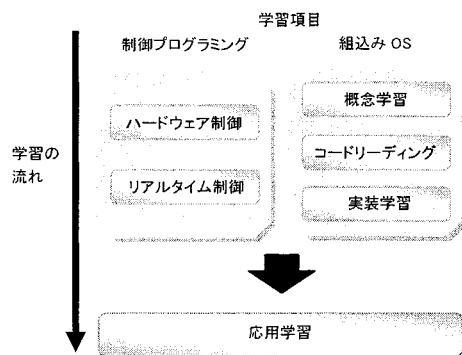


図 2 コースウェア

3.2.1 制御プログラムの学習

ここでは、ハードウェアの制御やリアルタイム制御について学習する。学習項目は次のとおりである。

(1) ハードウェア制御

ロボットに搭載されているモータやセンサといったデバイスの動作について学習する。モータやセンサを

用いて制御プログラムを作成し、実際にロボットを動作させてみることでデバイスの制御や接続方法について学習を行う。

(2)リアルタイム制御

組込み OS ではリアルタイムの制御が求められる。そこで OS のスケジューリング部分の変更することを通してリアルタイム制御の学習を行う。

3.2.2 システムソフトウェアの学習

ここでは、組込み OS に関する学習を行う。学習項目は次のとおりである。

(1)概念学習

この段階では、OS の概念を学習する。OS の複雑な動作や構造を抽象化し、アニメーションによって可視化することで学習者に動作のイメージを理解させる。

(2)コードリーディング

実際に OS のコードを読むことで組込み OS の機能を理解する。概念学習で学んだ動作イメージをもとに、実際のコードに触れることで OS の構成や機能について理解する。

(3)実装学習

学習者が OS のコードを改良、追加することで実際にロボットの制御を行う。具体的には、ライントレースや簡単なコース走行などを行うプログラムを作成することで組込み OS の動作やデバイスの制御について学習を行う。

3.2.3 応用学習

制御プログラミング学習、組込み OS の学習を通して段階的に学習してきた内容をもとに、ロボットを用いた応用学習を行う。具体的には、複数台のロボットを用いた協調動作や、ロボサッカーといったネットワークを用いた応用動作について学習を行う。

3.3 学習サポートツール

「港 ver.R」で提供する学習支援ツールについて述べる。コースウェアにおける各段階について学習をサポートするツールを用意することで、連続的に学習が可能となる。各学習ツールは次のとおりである。

3.3.1 OS 可視化ツール

ロボットの動作時に取得したログデータをもとに、タスクの遷移、タスクの実行時間、割り込み処理を時間変化のグラフやタスクの動作をアニメーションとして可視化する。これにより動作のイメージがつかみにく

いシステム内部の動作を実際に目で確認することができるため、学習者は動作概念を直感的に理解することができる。コースウェアにおける概念学習に用いることで学習の支援を行う。時間変化のグラフとタスク遷移のアニメーションが連動して動作している画面を図 3 に示す。



図 3 OS 可視化ツールの利用画面

3.3.2 組込み OS

ロボットの制御に用いる OS は、コードリーディングを想定し、コメントやドキュメントを豊富に用意した約 2000 行程度の小規模なものである。小規模化することにより学習者が全体の把握をしやすく、また、可読性を重視し、容易に OS の改良ができるようにすることで学習者の理解を深めることができる。また、汎用的に利用されている μ ITRON 仕様の OS である、TOPPERS/JSP^[7]を利用することも可能である。

3.3.3 システムモニタリングツール

ロボットの走行時に取得したログをもとに、ロボットのデバイス及びシステムの内部状態を GUI で表示するツールである。ロボットの外部動作を Web カメラで撮影し、そのビデオと連動してセンサやモータといったデバイスの内部データをグラフにして示すことで、デバッグやシステムの動作を容易に理解することができる。コースウェアにおいて、実装学習、応用学習に用いることで学習の支援を行う。システムモニタリングツールの利用画面を図 4 に示す。

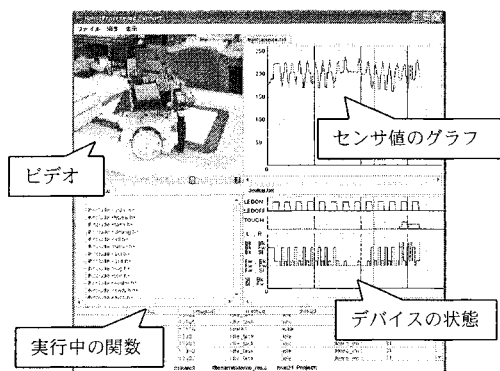


図4 システムモニタリングツールの利用画面

4. 設計方針

前述した問題点を解決するために、本システムでは次のように設計方針を定める。

(1)二つのシミュレータの統合

本研究では、ロボットの制御に用いる CAT709 ボードをシミュレートした、ボードシミュレータと、ロボットの外観をシミュレートした、3D ロボットシミュレータを提供する。この二つのシミュレータを連携させて学習者に提供することで、ロボットが存在しない環境においても学習の支援を行う。

また、シミュレータを用いて学習する内容は、「港 ver.R」で提供するコースウェアにおける、コードリーディング、実装学習、ハードウェア制御である。それぞれ、ロボットを用いて学習することが可能であるが、学習対象となる組込みシステム初学者にとって、突然ロボット等の実機を扱うことは困難であり、学習者の不安も大きい。そこで、実機を扱う前段階としてシミュレータを用いることで実際に実機を扱う際のイメージをつけることができる。このように、ロボットとシミュレータの二つの環境で学習が可能となることで段階的に学習することが可能となる。

(2)入出力制御を可能にする

組込みシステムは、外部からの入力により動作を行う。よって、組込みシステム学習においてデバイスの入出力制御の学習はなくてはならないものである。そこで、ロボットに用いられているデバイスを仮想デバイスとして実装し、それらへのアクセスを可能とすることでライントレースなどの具体的なロボットの制御を想定したプログラムの作成が可能となる。これにより、シミュレータ上においてもデバイスの入出力制御の学習が可能となる。

5. 設計

各シミュレータの設計について述べる。

5.1 ボードシミュレータ

ロボットの制御に用いるボードのシミュレータを提供する。これにより、学習者はロボットが存在しない環境においてもコードの書換えによる、組込み OS や制御プログラムの学習が可能となる。

また、5.2 で記述する 3D ロボットシミュレータと連動することで、組込みシステムにおいて重要である、センサに対する外部入力を得ることができる。さらに、ボードシミュレータだけでは困難な、外部動作のイメージを学習者に提供することが可能である。ボードシミュレータの全体構成を図5に示す。

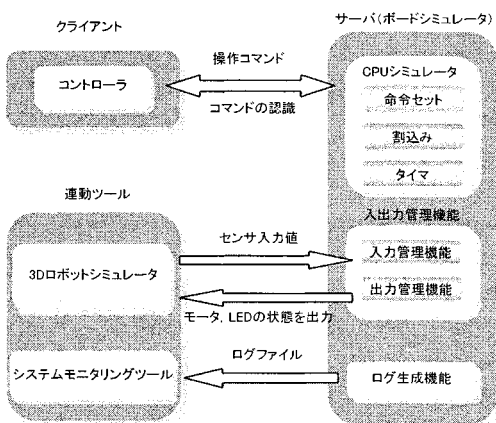


図5 ボードシミュレータの全体構成

ボードシミュレータは、クライアントサーバ方式で実装する。これにより、教材開発者は、対象に応じてユーザインタフェースを複数用意することができる。また、ボードシミュレータの本体となる CPU シミュレータとデバイスの入出力管理を行う機能を分離することで、多様な入出力装置に対応することができる。また、システム内の時間の管理は、命令セットの実行による経過クロック数で管理を行う。シミュレータ上で1命令の実行にかかる時間は実際の CPU より長くなってしまふ。そこで仮想時間を設け、実時間におけるクロックを粒度を落として扱い、実時間との整合性を図る。これは、組込み OS や制御プログラムを動作させることを目的としているため、命令セットを一つずつ実行し、正確にエミュレーションする必要があるためである。

5.1.1 CPU シミュレータ

ロボットの制御に用いられている SH3 マイコンのシミュレートを行う。これにより、学習用 OS をシミュレータ上で動作させることが可能となる。必要となる機能として命令セット、割込み、タイマの三つの機能を実装する。命令セットは、SH3 マイコンに用意されているものをすべて実装する。割込み機能には例外処理、一般割込み、NMI 割込みがあるが、本システムでは命令実行の際に必要な例外処理とタイマによる割込みが分類される一般割込みの処理を実装する。

5.1.2 入出力管理機能

3D ロボットシミュレータとデータの送受信を行う。これにより、組み込みシステムでは必要不可欠な入出力の制御が可能となる。入出力管理機能は、次に示す二つの機能から成り立つ。

(1) 入力管理機能

入力管理機能では、3D ロボットシミュレータ側から光学センサ、タッチセンサの入力値を受け取り、CPU シミュレータの入出力ポート部分に書き込む。それぞれのデータは一定時間ごとに受信する。これは実機において、外部環境は常に変化しているため、シミュレータ上においても常に新しいデータを得られるようにするためである。

(2) 出力管理機能

出力管理機能では、モータ、LED のデータを 3D ロボットシミュレータ側へ送信する。モータ、LED の状態は命令セットを一つ実行するごとに確認する。データを送信するタイミングはモータか LED のいずれかの状態が変化したときに行う。これは通信にかかるオーバーヘッドを軽減するためである。

5.1.3 ログ生成機能

シミュレータ上における動作ログをログファイルとして生成する。ログとして保存する情報は、システム時間、光学センサとタッチセンサの入力値、モータの状態、LED の状態である。

5.2 3D ロボットシミュレータ^[8]

学習環境の題材として用いるロボットの外観を3次元化した、3D ロボットシミュレータを提供する。可視化する内容は次の二点である。

- (i) ボードシミュレータ上におけるプログラムの動作結果

- (ii) ハードウェア上でプログラムを実行した際に生成されたログファイル

ユーザは GUI を操作し、3D 画面を確認することでプログラムの実行結果の確認や動作のデバッグを行う。3D ロボットシミュレータの全体構成を図 6 に Java3D^[9]による仮想空間の階層構造図を図 7 に示す。

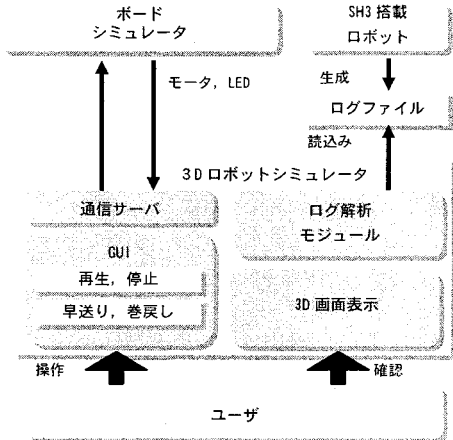


図 6 3D ロボットシミュレータの全体構成

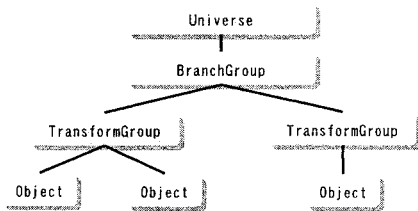


図 7 Java3D の階層構造

5.2.1 3D 画面表示部

ロボットの動作を Java3D で構築した 3D 空間に表示する。これにより、実機での動作がイメージしやすくなり、ユーザの学習理解を促す。

(1) 3D による画面表示

3D アニメーションの表示に必要な、ロボットのデバイス(モータ、LED)の値は、通信サーバ又はログ解析モジュールから受け取る。タイマタスクによって周期的にロボットの位置情報を更新し、同時に描画によるアニメーションを行う。また、ロボットが走行するコース、障害物の表示も行う。

(2) 障害物の配置

仮想空間内の障害物はユーザが自由に配置することができる。ユーザが使用できる機能には、障害物の生成、障害物の除去、障害物及びロボットの回転、水平移動がある。

(3) 3D 空間構造

3D 空間はツリー構造になっており、空間を定義する上位の Universe クラス、カメラの視点やフィールドを定義する中間層の BranchGroup クラス、個々の物体、または物体の集まりを定義する下位の TransformGroup クラスから成っている。

5.2.2 通信サーバ

ボードシミュレータとの通信で行う処理は次の二つの処理である。

- (i) ボードシミュレータからモータ、LED の状態を受け取る
- (ii) 光学センサ、タッチセンサの値をボードシミュレータへ送る

可視化するデータを受け取り、センサフィードバックを実装することにより、ボードシミュレータ単体では実行できないプログラムを実行可能にする。

(1) ボードシミュレータからの入力

モータ、LED の値の受取りは、ボードシミュレータとの接続と同時に入力待機用のスレッドを生成し、その内部でポーリングを行う。通信サーバが受け取ったデータは 3D 表示部に受け渡す。

(2) ボードシミュレータへの出力

光学センサ値、タッチセンサ値は、仮想空間内のロボットの座標データが更新されるたびにボードシミュレータ側へ送信する。光学センサは、ロボットの光学センサ部分の座標と MAP データの照合を行い、床の色を送信する。タッチセンサは、ロボットと障害物、外壁の接触があった場合に、センサ値を送信する。

5.2.3 ログ解析モジュール

ハードウェアが生成するログファイルから、ロボットの動作の再生、早送り、巻戻しを行う。これによりハードウェア上でプログラムを実行した際に起きる、再現性の低いバグの発見を容易にする。また、ログファイルの任意の時点でのロボットの動きを繰り返し確認することができ、デバッグの効率を高める。

(1) 再生機能

ロボットが生成するログファイルには、時間(ms)ご

とのデバイスの状態が記録されている。ログ解析モジュールはログファイルに記録されている時間とモータの状態を 3D 表示部に受け渡す。

(2) 早送り、巻戻し機能

ログ解析モジュールはログファイルに記録されているデータから移動距離を計算し、1 秒ごとのロボットの座標を格納しておく。これを利用することで任意の位置からの早送り、巻戻しを行うことができる。

6. 実現

本研究で実現したシステムにおける実現規模と開発言語を表 1 に示す。

表 1 システムの開発言語と規模

システム	開発言語	行数
ボードシミュレータ	C 言語	8207
3D ロボットシミュレータ	Java 言語	2746

6.1 ボードシミュレータの開発

ロボットの制御に用いられている CAT709 ボードのシミュレータを開発した。これにより、次に示す点が可能となった。

(1) 組み込み OS の実行や制御プログラム

実機を用いなければ動作の確認ができなかった、組み込み OS や制御プログラムの動作確認が PC 上で行うことができる。

(2) 学習ツールとの連携

シミュレータ上における動作ログを取得できる。これにより、システムモニタリングツールによるデバッグができる。

6.2 3D ロボットシミュレータの開発

ロボットの外観をシミュレートした、3D ロボットシミュレータを開発した。3D ロボットシミュレータの利用画面を図 8 に示す。これにより、次に示す点が可能となった。

(1) 3D によるロボットの動作の可視化

プログラムの実行を可視化することで、ロボットの動作確認が容易にできる。

(2) センサフィードバック

仮想空間内における床の色や接触判定を、外部入力と得られるようにすることで、ボードシミュレータだけでは実行できない、ライトレースといったプログラムの実行ができる。

(3) ログファイルの再生

実機がプログラム実行時に生成するデバイスのログの再生、早送り、巻戻しを可能とすることでハードウェア上で動作させた際のデバッグができる。

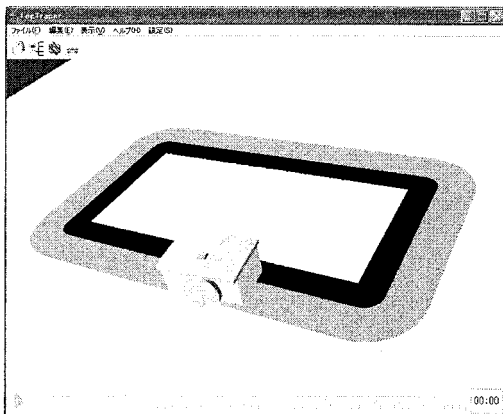


図8 3D ロボットシミュレータ利用画面

また、システムモニタリングツールではビデオの変わりに3D ロボットシミュレータによる可視化が可能である。システムモニタリングツールと3D ロボットシミュレータを統合した環境を図9に示す。

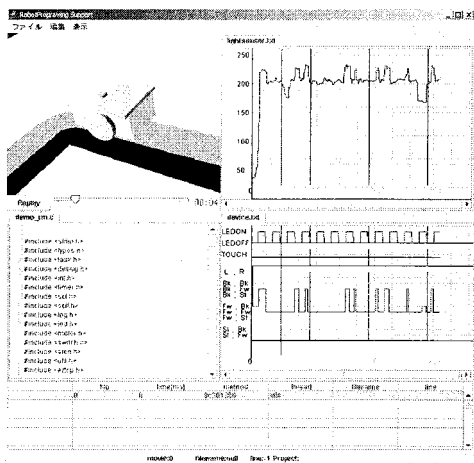


図9 3D ロボットシミュレータとシステムモニタリングツールの統合環境

6.3 シミュレータの利用方法

各シミュレータの具体的な利用方法を次に示す。

(1) ソフトウェアのデバッグ

ロボットを動作させる際に生じる不具合は、ロボッ

ト自体に原因がある場合と、ソフトウェア側に原因がある場合の二通りが考えられる。組込みシステムを学習する際、実際にロボットの調節することは必要不可欠であるが、原因の追究にかかる時間はできる限り少ないものとしたい。そこで、実際にロボットを使う前段階としてシミュレータを使うことで原因追求を容易にする。利用手順は次の通りである。

- (i) ロボットを制御するためのプログラミングを行う。
- (ii) 作成したプログラムをボードシミュレータと3D ロボットシミュレータを用いて動作の確認をする
- (iii) 作成したプログラムがイメージどおりに動作した場合は、実際にロボットを使って動作させる。プログラムに誤りがある場合は(i)に戻り、プログラムの修正を行う。

(2) ロボットが存在しない環境における利用

ロボットが存在しない環境、自宅での自主学習や、複数人で一つのロボットを共有して使う場合に用いる。利用手順は次の通りである。

- (i) ロボットを制御するためのプログラミングを行う。
- (ii) 作成したプログラムをボードシミュレータと3D ロボットシミュレータを用いて動作の確認をする。
- (iii) 得られたログデータをもとにシステムモニタリングツールを用いてデバッグすることでプログラムの動作確認を行う。
- (iv) (i)に戻る。

(3) ログデータを用いたプログラムのテスト

一度ロボットで動作させた環境を何度も繰り返し検証したい場合に用いる。また、ログデータは用意に改良、修正可能であるため、理想的な環境を用意に作成することができる。利用手順は次の通りである。

- (i) ロボットを制御するためのプログラミングを行う。
- (ii) ロボットを用いて作成したプログラムの動作を検証する。
- (iii) ロボットを動作させた際に得られたログデータをもとにボードシミュレータ、3D ロボットシミュレータを用いて動作の検証または比較を行う。
- (iv) ロボット制御用のプログラムを変更、また、ログデータの改良、修正を行う
- (v) (iii)へ戻る。

7. 関連研究

本研究と関連のあるシミュレータを用いた学習環境として次のものがある。

(i) Robocode^[10]

RobocodeはIBMが提供しているロボット対戦シミュレータである。Java言語を用いてプログラミングを行い、PCのスクリーン上で対戦させる。世界中でトーナメントやコンテストが行われており、企業や教育機関においてはJava言語の学習にも用いられている。また、プログラムした結果が明確に目で見て確認できるという利点があるが、外部の環境に影響を受けるといった、実機特有の問題を体験することができない。

(ii) サッカーロボットシミュレータ^[11]

本研究でターゲットとなっている、サッカーロボ915の制御プログラムが出力できるシミュレータで、ボタン操作により、容易にプログラムの作成が可能である。また、ロボットが走行するコースの作成など自由度の高いシミュレートを行うことが可能である。しかし、デバイスの入出力に関するサポートが不十分であり、組込みシステムの学習教材としては不向きである。

(iii) 計算機シミュレータとラジコンカーを用いた学習環境^[12]

計算機の動作原理をシミュレータとラジコンカーを用いて学習する環境である。計算機の動作原理をアニメーションを用いて可視化しているため、学習者が理解しやすいという利点がある。また、計算機シミュレータからラジコンカーを制御できるため、作成した制御プログラムが実際に実機上でどのように動作するかを目で見て確認することができる。しかし、制御プログラムに特化した環境であるため、組込みOSの学習が不可能であり、また、外部からの入力に対応していないため、実機の動作パターンが限られてしまう。

8. おわりに

本稿では、ロボットを用いた組込みシステム学習環境「港 ver.R」の概要と、その一部であるシミュレータについて述べた。本研究を通して、実機の存在しない環境における組込みシステム学習環境の提供が可能となった。

今後の課題として、3Dロボットシミュレータ上における複数台のロボットを用いた協調動作への対応、各シミュレータを用いた学習環境における教育的評価があげられる。

参考文献

- [1] 経済産業省商務情報製作局, 2007年版組込みソフトウェア産業実態調査報告書
- [2] 上野真路ほか, “教育用マイコン制御システムの開発研究”, SWEST6, Jul., 2004
- [3] 田中裕樹ほか, “ロボットを用いた組込みシステム学習環境における学習支援環境の開発”, SWEST8, pp59-66, 2006
- [4] robocup,
<http://www.robocup.or.jp/>
- [5] エレキット株式会社,
<http://www.elekit.co.jp/>
- [6] シリコンリナックス,
<http://www.si-linux.co.jp/>
- [7] TOPPERS プロジェクト,
<http://www.toppers.jp/>
- [8] 幸坂晋ほか, “組込みシステム学習環境「港」における3次元ロボットシミュレータの開発”, 情報処理学会第69回全国大会, 2007
- [9] Sun's Java3D Homepage,
<http://java.sun.com/products/java-media/3D>
- [10] IBM, robocode,
<http://www-06.ibm.com/jp/event/robocode/home/>
- [11] 財団法人九州システム情報技術研究所,
<http://www.isit.or.jp/~kimuro/rcupj/>
- [12] 木室義彦ほか, “計算機シミュレータとラジコンカーを用いた計算機の動作原理教育”, メディア教育研究, 第3巻, 第1号, pp125-131