

教材の共同利用を可能にする中学校向け制御学習システムの提案*

井戸坂幸男[†] 中野 由章^{††} 紅林 秀治^{†††} 兼宗 進^{††††}

A Control Learning System Sharing Control Devices for Junior High School*

Yukio IDOSAKA[†], Yoshiaki NAKANO^{††}, Shuji KUREBAYASHI^{†††},
and Susumu KANEMUNE^{††††}

あらまし 中学校の技術・家庭科(技術分野)では, 学習指導要領の改定により, 2012年度から計測・制御の学習が必修になった。生徒が1人1台ずつの制御教材を使う環境での授業は, 教材を準備する経済的な負担だけでなく, 授業で教材を管理する指導者の負担も大きい。少ない制御教材で授業ができるようにするために, 数台の制御教材を生徒全員で共有し利用する学習環境の仕組みを考えた。ネットワークで接続された環境で制御教材を共有する学習システムを開発し, 中学生を対象とした授業で有効性を確認した。このシステムは, 一般的な開発システムの仕組みを体験的に学ぶ教材としても活用可能である。教室内で使用する制御教材の数を少なくし, 複数台の端末から制御教材を共有して利用できる計測・制御の学習システムを提案する。

キーワード 計測・制御学習, 制御教材, ネットワーク環境, プログラム学習, 教材共有

1. ま え が き

中学校の技術・家庭科(技術分野)では, 学習指導要領[1]の改定により, 2012年度から計測・制御の学習が必修になった。

実習には自律型の制御ロボット教材が多く使われている。この教材は, プログラム転送時のエラーや制御ロボットのハード面での故障など, 不具合が発生する場合も多い[2]。生徒が1人1台の制御ロボットを使う学習環境では, このような不具合に対応する指導者の負担は大きい。また, 1人1台(学級全体で40台)の制御ロボット教材を準備することは, 学校の予算の面, 維持管理の面からも難しいと考えられる。授業に

おける指導者の負担や教材を準備する経済的な負担から考えると, 制御ロボットの台数は少ない方が望ましい。一方, プログラムの制作や制御ロボットの操作など生徒の学習効果から考えると, 1人1台の学習環境が望ましいと考えられる。

そこで, 数台の制御ロボットを生徒全員で共有して利用し, プログラムの制作や制御ロボットの操作などは個人で学習を進める学習環境を設計・開発し, 中学生の実験授業で有効性を確認した。この学習環境を実現するために, 教室内で少数の制御ロボットを複数台の端末から共有して利用できる計測・制御の学習システムを提案する。なお, 本論文では, 中学校学習指導要領と同じ表記にするために計測・制御の学習としているが, 扱う内容は制御学習のみである。

2. 計測・制御の学習環境

2.1 中学校における計測・制御の学習内容

中学校の計測・制御の学習は, 技術・家庭科の「D 情報に関する技術」領域で扱われる[1]。この領域では, プログラムによる計測・制御について, 「コンピュータを利用した計測・制御の基本的な仕組みを知ること」と「情報処理の手順を考え, 簡単なプログラムが作成できること」を指導する。更に, 中学校

[†] 松阪市立飯高東中学校, 松阪市
Iitakahigashi Junior High School, 927 Miyamae, Iitaka-cho,
Matsusaka-shi, 515-1502 Japan

^{††} 神戸市立科学技術高等学校, 神戸市
Kobe Municipal High School of Science and Technology, 1-
4-17 Wakihamacho, Chuo-ku, Kobe-shi, 651-0072 Japan

^{†††} 静岡大学教育学部, 静岡市
Faculty of Education, Shizuoka University, 836 Oya,
Suruga-ku, Shizuoka-shi, 422-8529 Japan

^{††††} 大阪電気通信大学, 寝屋川市
Osaka Electro-Communication University, 18-8 Hatsucho,
Neyagawa-shi, 572-8530 Japan

* 本論文はシステム開発論文である。

学習指導要領解説[3]には、「ものづくりを支える能力を育成する観点から、実践的・体験的な学習活動を通して、情報を収集、判断、処理し、発信したり、プログラムにより機器等を制御したりする喜びを体験させる」とあり、体験的な実習を取り入れた学習を通して学ばせる必要がある。

本研究は、機器を使った体験的な実習を通して、計測・制御の基本的な仕組みを学ぶ学習を対象とする。

2.2 学習環境の検討

中学校の計測・制御の学習で使われる教材や学習形態を調査した。制御教材は車型で自律型の制御ロボット教材が多く使われており、1～2名に対して1台の割合で使用し、制御プログラムを制作する学習を展開している[4]～[8]。学習形態は、個人で学習を進める場合とグループで学習を進める場合がある。そこで、個人とグループの学習形態について、生徒の実習、授業運営、学習評価、教材管理などの観点から比較検討した。

生徒の実習に関しては、個人学習が適している面が大きい。個人学習では個人でプログラムを作り、自分の制御ロボットにプログラムを転送し、自分で制御ロボットを操作して学習を進める。全員が全ての作業を体験できることが、個人学習の利点である。一方、グループ学習では、グループ内の作業の分担に課題がある。生徒に任せて授業を進めると、ロボット製作を1人が独占してしまう場合や、遊んでしまう生徒が生じる場合がある。グループ内でプログラムやメカニックなどの担当を決める作業分担制をとることは作業を分担して作業できる利点があるが、担当する作業によって生徒に知識の偏りを生じさせる可能性がある[9]。生徒ごとに行う作業が異なることは、学習評価の面からも問題が生じる可能性がある。教材管理に関しては、使用する制御ロボットの台数を少なくできるグループ学習が適している。使用する台数が少ないということは、制御ロボットに発生する不具合への対応が容易になり、指導者の負担も軽減される。また、経済的な負担も小さい。以上の検討より、学習形態は個人学習で行い、使用する制御ロボットの台数が少ない学習環境が適していると考えた。

使用する制御ロボットを少なくする方法としては、ロボットを共有することが考えられる。共有は、グループ学習を行う場合はグループで共有する形になるが、今回は個人学習を行うことから、指導者の近くで制御ロボットを管理でき、不具合に対する対応が容易になると思われる教室全体で共有する方法を検討した。

2.3 教室で共有するための機能の検討

前節の学習環境を実現するために、必要になる機能を検討した。

一つ目は、生徒の端末画面でプログラムを確認するためのシミュレーション機能である。制御プログラムを学習するためには、作ったプログラムを実行して動作を確認する必要がある。動作を確認する方法は実機で実行することが基本だが、転送に時間を要するという問題がある。そこで、生徒の画面上で動作をシミュレーションすることで時間を短縮することを検討した。生徒はプログラムの意味を理解するために何度も動作を確認する必要があるため、限られた時間で繰り返し確認できる利点は大きいと考えられる。

二つ目は、機体ごとの補正を行うキャリブレーション機能である。実機にプログラムを転送して走行させる学習は、シミュレーションによる「画面という理想的な世界で誤差なく動く」学習に続いて行う「現実世界では正確な距離を動いたり正確な角度を曲がることは難しい」ことを体験的に学習する場である。この体験は、生徒は現実世界での制御の難しさや、フィードバックのある制御の必要性を考える学習につなげることができる。しかし、授業で使用する教材は安価なモーターを使用しており、機体ごとの移動距離や回転角度の誤差が大きすぎて授業に支障があるという問題がある。そこで、機体ごとの補正を行い、学習に支障のない程度に機体ごとの誤差を小さくするキャリブレーション機能が必要であると考えた。

三つ目は、生徒のプログラムを指導者の近くにあるサーバーへ送る転送機能である。サーバーにプログラムを集めることにより、生徒の学習状況を把握することができる。共有された制御ロボットを公平に使わせ、それぞれの生徒の学習の進捗を把握することは、授業運営の面から重要性が高い。端末のプログラム転送機能とともに、サーバーには生徒の学習状況が把握できる機能が必要となる。

3. 関連研究

制御機器を共有する学習環境に関しては、初等中等教育向けの機器の共有に関する研究は非常に少ない。それは、学習指導要領[1]の改訂以前は、計測・制御学習は選択教科の技術・家庭科の中で行われていたため、実践が非常に少なかったこと、選択教科のため少人数での実践が可能であったことが理由として考えられる。中学校において教室内のネットワークを使った

報告として、チャットプログラムを作り、チャットやメールの仕組みが学習できる機能を実装したプログラム言語「ドリトル」がある [10]。プログラミング学習の中で、教室内のネットワークを活用した事例であり、計測・制御学習に関して教室内のネットワークを活用した事例ではない。ネットワークを使ったシステムとしては、大学生を対象とした工学教育用の遠隔実験システムが開発されている [11]。制御機器を実際に見て操作するのではなく、離れたところにある制御機器の動作をカメラ画像で判断するなど、初心者が使用するには高度なシステムとなっている。このようなシステムは大学の専門教育には適していると考えられるが、基本的な制御プログラムや制御機器の操作の修得を目指す初心者の中学生にとっては適しているとはいえない。しかし、このようなネットワークを使った開発システムは、中学生にも必要な学習であると考えられる。

シミュレーション機能に関しては、シミュレーションをロボットプログラミングの学習に生かす研究事例が報告されている [12]。また、個人もち教材として市販されている車型の制御ロボット教材に搭載されているものもある^(注1)。しかし、この教材を中学校の授業で使用したところ、実機とシミュレーションの両方が使える環境では、生徒は実機でプログラムを動かしたいという気持ちが強く、シミュレーション機能を使う生徒は少数であった。実機が手元にある環境では、生徒はシミュレーション機能を使うことは少ないと考えられる。

キャリブレーション機能に関しては、個人もち教材として市販されている車型の制御ロボット教材の多くに搭載されている [2]。これは、安価なモーターを使用しているため左右のモーターの回転速度を調整して車型の制御ロボットを直進させるために使われる。本システムに必要なキャリブレーション機能は、個々の制御ロボットの直進性の調整だけでなく、制御ロボット間の個体差をなくす機能である。

4. 教材を共有する学習システムの設計

本システムは、中学校の技術・家庭科における計測・制御の授業で使用される学習システムとして設計した。中学校の一般的な授業は、ネットワーク設備のあるコンピュータ室で、40名の生徒が一斉に学習する。そ

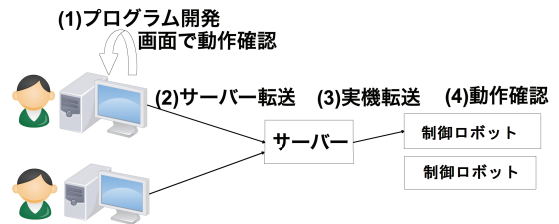


図1 制御教材を共有する学習システム
Fig. 1 Robot sharing learning environment.

こで、教室内にあるネットワークを利用して、制御ロボットを共有できるように設計した。

学習の流れは、次のように考えた。まず、学習者は自分の席で、課題のプログラムを作り、シミュレーション機能を使って動きを確認する。シミュレーションで課題が達成できていることを確認した後、プログラムをサーバーに転送する。次に、指導者の近くにあるサーバーのところまで移動し、用意された実機に自分のプログラムを転送する。そして、実機を走行させるコースまでもって行き、動作を確認する。

図1に提案する学習システムのモデル図を示す。

学習の流れをまとめると、次のようになる。

- (1) 端末でプログラムを作成し、シミュレーション機能で動作を確認する。
- (2) 確認したプログラムをサーバーに転送する。
- (3) サーバーで自分のプログラムを実機(制御ロボット)に転送する。
- (4) 実機(制御ロボット)で動作を確認する。

この学習環境を実現するためには、生徒の端末とサーバーでは、次の処理が必要となる。

生徒の端末で必要な処理：

- プログラムの作成。
- シミュレーション画面での動作確認。
- サーバーへのプログラム転送。

サーバーで必要な処理：

- プログラムの受信と保持。
- 端末から送られたプログラムの待ち行列の管理。
- 実機(制御ロボット)ごとの補正。
- 実機(制御ロボット)への転送。

そして、上記の処理を実行するためには、次の機能を実装する必要がある。

- 端末におけるシミュレーション機能。
- サーバーへのプログラム転送機能。
- 実機(制御ロボット)へのプログラム転送機能。

(注1) : Beauto Racer (Vstone 社) http://www.vstone.co.jp/products/beauto_racer/index.html, 参照 May. 1, 2013.

- 実機(制御ロボット)の個体差をなくすキャリブレーション機能.

本システムの利点として、シミュレーションで動作を確認することにより、実機を使う回数や時間を大幅に減らすことができる。また、制御プログラムができた生徒から順に実機を使うため、実機使用の順番待ちの時間を減らすことができる。更に、ネットワークを使ってサーバーにプログラムを転送するため、生徒が何回実機を使ったかをサーバーのログから知ることができる。ログにある実機の使用回数から、多く使用している生徒には回数を制限したり、逆に少ない生徒に優先的に使用させたりするなど、公平に実機を使用させることができる。また、ネットワークを使ってサーバーに生徒のプログラムを集め、プログラムを補正して転送することができるため、制御ロボットの固有の特性情報をあらかじめ登録すれば、容易にキャリブレーション機能が実装できる。

以下の節では、これらの機能の実装について説明する。

5. システムの実装

5.1 システムの概要

プログラム言語には、ドリトル [13] を使用した。ドリトルは教育用に開発された言語であり、Windows, Mac OS X, Linux など動作可能である。プログラムを日本語によるオブジェクト指向で記述できる特徴をもち、小中学校から大学までの幅広い授業で利用されている。グラフィックスなどの画面内で実行されるプログラムに加え、ネットワークでの通信やロボットなどの外部機器制御が可能である。

制御教材には、ミュウロボ^(注2)を使用した [14]。図 2 に示すミュウロボは車型の自律型制御ロボット教材であり、中学から大学までの制御の授業で利用されている [15]。左右の車輪に DC モーターが組み込まれ、このモーターを制御することで、前進、後退、左右の回転の動作をすることができる。基板には、CPU として PIC(16F688) が搭載されており、255 バイトの命令を格納できる^(注3)。作成したプログラムをコンピュータからシリアルケーブルを使い、実機のメモリに転送することで、自律型の制御ロボットとして動作する。セ

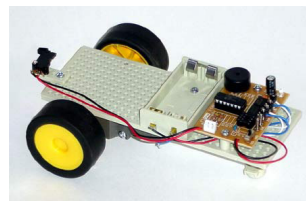


図 2 使用した制御ロボット「ミュウロボ」
Fig.2 Myu-robot.

ンサとしては、触覚センサ(タッチセンサ)、赤外線センサ、距離センサ等を取りつけることができ、計測値を使ったフィードバック制御が可能である。

ドリトルは標準でミュウロボに対応しており、記述されたプログラムを制御ロボット教材に合わせた命令コード(バイトコード)にコンパイルし、実機に転送できる。今回は図 1 のように、生徒が端末で作成したプログラムをサーバーに転送し、サーバーからロボットに転送するようにした。

本システムの利用にあたっては、教室内にあるネットワークを使い、生徒用端末と教師用端末(サーバー)にドリトルをセットアップすれば使用でき、特別な設備は必要としないように工夫した。サーバーが複数台必要な場合は、空いている生徒端末をサーバーとしてセットアップすることが可能である。また、ドリトルは無償で配布されており、必要な経費は使用する制御ロボットの費用のみである。必須の機器ではないが、バーコードリーダーを利用することで、機体番号の入力や転送の実行にキーボードやマウスを使う操作を不要にすることができ、サーバーから実機への転送作業を容易に行うことが可能になる。バーコードリーダーは安価な市販の製品に対応している。

本システムのセットアップ方法は、生徒用端末とサーバーの両方にドリトルをインストールし、本システムの端末用のプログラムとサーバー用のプログラムを読み込めば完了する。その後、端末のドリトルでサーバーの IP アドレスを設定する。キャリブレーションは、機体ごとに「前進距離」「回転角度」を測り、サーバーのドリトルに入力しておく。

5.2 端末でのシミュレーション機能

図 3 にシミュレーション機能を搭載した場合の端末側の構成図を示す。本システムのために、コンパイルされたバイトコードを解釈するインタプリタを開発し、画面上でロボットに見立てたキャラクタを移動して実行することで、シミュレーションを実現した。インタ

(注2)：ロボット工房スタジオミュウ <http://www.studiomyu.com/>, 参照 May. 1, 2013.

(注3)：今回は 255 バイトを格納できる機種を使用した。127 バイトや約 2K バイトの機種も存在する。

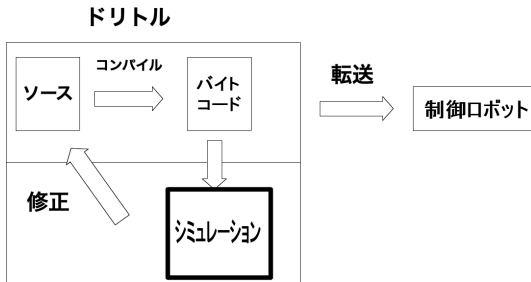


図 3 端末でのシミュレーション機能の実装
Fig.3 Extended procedure with the simulator.

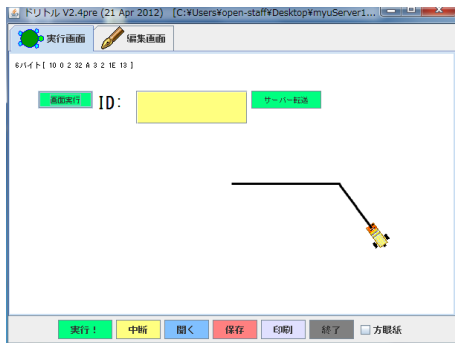


図 4 端末でのシミュレーション例
Fig. 4 On simulator.

表 1 バイトコードの格納例
Table 1 Sample of byte code.

ユーザー ID	バイトコード
wada	10 0 2 A A 5 8 A 13
nakano	10 0 2 F B 5 2 A 13

表 2 バイトコードの命令の例
Table 2 Sample of byte code.

命令	使用例	説明
2	2 A	「前進 10」を実行する
8	8 1E	「後退 30」を実行する
A	A 5	「右回り 5」を実行する
B	B 14	「左回り 20」を実行する
53	53 A	飛び先 (10) を指定する
54	54 A	指定した飛び先 (10) に飛ぶ



図 5 サーバーの画面例
Fig. 5 Screen shot of server.

プリタはドリトル言語で実装した。配列に格納したバイトコードを1バイトずつ解釈し、1パス目でジャンプテーブルを作成し、2パス目で命令を実行する2パス方式で動作する。

図 4 に端末でのシミュレーション例を示す。シミュレーション機能は、学習者がプログラムを作成した後、実機を動作させる前に画面上で動作確認を行うために使用する。シーケンス制御の課題を考えた場合、移動する距離や角度は実機と近い動作にしておく必要がある。今回は複数台の実機で単位時間あたりに移動する距離や回転する角度を測定し、標準的な動作を行うようにシミュレーションのパラメータを調整した。

5.3 サーバーへのプログラム転送機能

端末からサーバーへのプログラムの転送には、ドリトルの通信機能を利用した。転送では、プログラムに加え、端末にログインしているユーザーごとの ID(ログイン名または端末の IP アドレス) を送るようにした。このことで、サーバーに送られたプログラムの作成者を識別できるようにした。

表 1 にサーバーに転送されたプログラム (バイトコード) の例を、表 2 にミュウロボのバイトコードの

命令の一部を示す。

5.4 実機へのプログラム転送機能

図 5 にサーバーの画面例を示す。サーバーの画面には、クライアントから転送されたプログラムのリストがログイン名で表示される。学習者は画面のリストから選択または ID を入力することで、プログラムを実機に転送できる。

今回のシステムでは、サーバーから実機に転送する際、他人のプログラムを誤って選択することを防止し、短時間で転送できるようにするために、バーコードリーダーを利用した。ログイン名をバーコード表示したカードを用意し、ログイン名はカードから読み込まれるようにした。また、転送する制御ロボットにもバーコードを貼り、ロボット番号もバーコードリーダーで読み取れるようにした。この工夫により、プログラムの選択ミスやロボット番号の入力ミスを防ぎ、スムーズな実習を行うことができるようにした。

5.5 個体差をなくすキャリブレーション機能

機体ごとの誤差を小さくするためのキャリブレーション機能をサーバー側に実装した。端末から送られてきた前後進、回転命令の引数にあたる部分に、補正

係数を掛ける形で補正した。補正係数は実機ごとに異なるため、全ての実機の動作距離と回転角度を測定し、補正係数を割り出している。共有する制御ロボットには番号がつけてあり、その番号ごとに補正係数を変更して制御ロボットに出力するようにしている。

6. 授業での評価

6.1 実施した授業

大学で実施された中学生向けセミナーで、開発した学習システムの評価実験を行った。対象は、一般公募で集まった中学生 15 名 (1 年 7 名, 2 年 6 名, 3 年 2 名) である。中学校での授業を想定し、指導は教員 1 名が担当し、他の教員や TA (ティーチングアシスタント) は授業中の指導には参加しない方針で行った。TA は正確なデータを取るために配置し、課題達成の判定や達成時間、試行回数等の記録を行った。他の教員は生徒の観察やビデオによる記録を担当した。

実験方法は、この 15 名を無作為に二つのグループ (A 8 名, B 7 名) に分け、1 人 1 台の制御教材を使用して学習を進める個人学習の環境 (個人環境) と今回開発したネットワークを利用して制御教材を共有する学習環境 (共有環境) の両方を用意し、比較実験をした。また、二つのグループは、両方の環境が体験できるように、前半の実習と後半の実習で使用する環境を入れ替えた。共有環境では、サーバーを 1 台、制御ロボット (ミュウロボ) を 3 台使用した。また、課題のコースは、10 コース以上用意し、実機でコースを試すための待ち行列はできないようにした。

午前中はドリトルを使ったグラフィックスを中心とする基本的なプログラミング学習、午後はミュウロボを使った計測・制御学習を行った。

午後の授業の展開を次に示す。

- (1) 操作説明 (50 分): 制御命令, 転送方法の説明
- (2) 実習 1 (50 分): 課題 1
 - A グループは個人環境, B グループは共有環境
- (3) 実習 2 (50 分): 課題 2
 - A グループは共有環境, B グループは個人環境

最初の操作説明 (50 分) は、個人環境と共有環境の両方に必要なミュウロボの制御命令やプログラムの転送方法等の基本的な知識を学習する。共有環境のみに必要なシミュレーターの使い方、プログラムをサーバーへ転送する方法、サーバーから実機へ転送する方法は、実習の最初に 5 分程度で説明した。

学習する課題としては、次にあげる 2 種類のものを

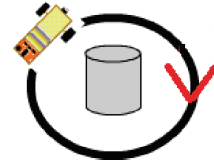


図 6 課題 1 の例
Fig. 6 Task 1.

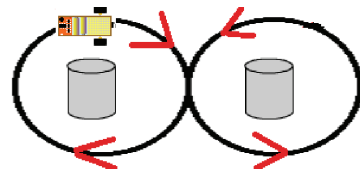


図 7 課題 2 の例
Fig. 7 Task 2.

表 3 課題の達成割合 (達成人数/全人数)
Table 3 Result of task1, 2.

グループ	環境	課題 1	環境	課題 2
A	個人	8/8 人	共有	3/8 人
B	共有	5/7 人	個人	3/7 人

表 4 課題達成者の実機での平均試行回数 (標準偏差)
Table 4 The trial number of times with the machine.

グループ	環境	課題 1	環境	課題 2
A	個人	6.0 回 (4.03)	共有	5.7 回 (1.89)
B	共有	2.2 回 (1.47)	個人	12.7 回 (8.22)

用意した。

課題 1: 柱状の障害物の周囲を (右回りで) 回るプログラム (図 6)

課題 2: 二つの柱状の障害物を (左回りからスタートして) 8 の字に回るプログラム (図 7)

課題 1 は、前進と回転を組み合わせて障害物を回る課題である。前進と回転の時間を厳密に設定しなくても周回は可能である。課題 2 は、同じ方向の回転だけでなく、反対方向の回転も必要な課題である。

6.2 授業の結果

課題が達成されたかどうかの判断は、TA の立ち会いの下で、ミュウロボをコース上で走行させ、与えられた条件のとおり走行するかを TA が判定する形で行った。TA を複数配置し、TA の立ち会いを待つ生徒が出ないように配慮した。

それぞれの課題に対する達成割合を表 3 に、課題達成者の実機での平均試行回数を表 4 に、課題達成までの平均時間を表 5 に示す。

課題 1 については、個人環境では 8 名全員が達成で

表 5 課題達成までの平均時間 (標準偏差)
Table 5 The average time of subject achievement.

グループ	環境	課題 1	環境	課題 2
A	個人	25.6 分 (2.50)	共有	34.3 分 (3.40)
B	共有	27.0 分 (8.27)	個人	27.0 分 (15.90)

きたが、共有環境では 5 名が達成でき、2 名は達成できなかった。達成できなかった 2 名は、実機でのコース試行をそれぞれ 3 回は実施している。試行回数は、個人環境では最少 2 回、最多 15 回、課題達成者の平均試行回数は 6.0 回であった。共有環境においては、最少 1 回、最多 5 回、課題達成者の平均試行回数は 2.2 回であった。個人環境における試行回数の少ない生徒は、実際にコースで試行する前に実機を自分の座席付近で動作させてからコースを使っていたため、実際に実機を動作させた回数をもっと多いと考えられる。共有環境においては、手元の実機がないため、試行回数分だけの実機走行である。個人環境での最短の達成者は 22 分、最長で 30 分、課題達成者 8 名の平均は 25.6 分であった。共有環境では、最短で 16 分、最長で 40 分、課題達成者 5 名の平均は 27.0 分であった。

課題 2 については、個人環境では 7 名中 3 名、共有環境では 8 名中 3 名が達成できた。試行回数は、個人環境では最少 2 回、最多 22 回、課題達成者の平均試行回数は 12.7 回であった。共有環境においては、最少 3 回、最多 7 回、課題達成者の試行回数は 5.7 回であった。個人環境での最短の達成者は 5 分、最長で 42 分、課題達成者 3 名の平均は 27.0 分であった。共有環境では、最短で 31 分、最長で 39 分、課題達成者 3 名の平均は 34.3 分であった。

次に、授業後のアンケートに書かれていたシミュレーターに関する生徒の感想を紹介する。

- シミュレーターのある場合は手軽に動作が確認でき、短時間でできますが、ない場合と比べてシミュレーションと実際の動きにズレがあります。
- シミュレーションの方が、頭と画像を照らし合わせられるから正確にできてよかった。
- シミュレーターがない方が想像しにくいので、ある方がやりやすかった。
- シミュレーションはあまり信じられないから、ない方がよい。
- シミュレーターは実際に走らせなくていいので便利だが、実際にやってみないとわからないところもあった。

6.3 授業の考察

課題の達成割合から、開発したシステムの有効性を考察する。

課題 2 については、個人環境も共有環境もほぼ同じ達成割合である。しかし、課題 1 については、個人環境では全員が達成できているにもかかわらず、共有環境では 2 名の生徒が達成できなかった。共有環境で課題 1 が達成できなかった生徒 2 名を詳しく調べた。片方の生徒は、後半の実習の個人環境でも課題を達成することができなかった生徒であり、どちらの環境においても課題を達成できなかった生徒である。もう一方の生徒は、個人環境の課題では最も長い時間の 42 分もかかって課題 2 を達成している生徒であることから、課題 1 に関しても時間さえあれば達成できた可能性があると思われる。以上より、開発した制御ロボット教材を共有する共有環境は、1 人 1 台の制御ロボット教材を使用する個人環境と同等の学習達成率があり、中学生が使用することは可能であると考えられる。また、複数の教員による生徒の観察より、共有環境の生徒はシミュレーターで試行錯誤を繰り返してプログラムを作り、できたプログラムを容易にサーバーに送る様子が観察され、操作に戸惑っている生徒がいなかったことから端末側は問題なく使用できていたと考えられる。また、サーバーではバーコードリーダーを使い、自分のカードから ID を読みとり、続いて制御ロボットの番号を読み取ることによって、瞬時にプログラムが制御ロボットに転送されていた。これらの観察より、本システムが中学生にとって適切に使用できるシステムであると判断した。

比較的やさしい課題 1 に対して一部の生徒が達成できなかった理由は、時間が足らなかったことが原因とも考えられるが、見方を変えるとシミュレーターを使用しているため、シミュレーターと実機の動きを結びつけて考えることが、中学生にとっては難しく、慣れるまでに時間がかかったとも考えられる。共有環境では作ったプログラムをサーバーに転送し、サーバーで自分のプログラムを実機に転送するという一連の流れを理解する必要がある。このことは、共有環境ではシミュレーターを使って動きを確認した上で実機に転送し、個人環境の毎回実機で動作確認をする場合に比べ、達成までの時間が短縮されるという予想に反して、実際には共有環境の方が時間が多くかかっていたことから理解できる。共有環境では、実

機への試行回数は個人環境の半分以下であるにもかかわらず、共有環境の方が課題達成までの平均時間が個人環境よりも多くなっている。

実機での平均試行回数より、制御ロボット教材を共有する学習環境では、1人1台の学習環境に比べて、実際に実機を使ってコースで試行する回数がおおよそ1/2以下になることがわかった。このことは、コースの待ち行列の問題に対しても有効な解決法であることを示している。

今回の実験授業では、学習者8名に対して、サーバー1台、制御ロボット3台で行ったところ、生徒の待ち行列も発生せず、順調に実機への転送が行えた。この経験から、与える課題や実機を走らせるコースの数にも関係するが、学習者8～13名に対してサーバーを1台程度、それぞれのサーバーに制御ロボットを3～5台程度用意すれば授業運営が可能であることを確認できた。40人の授業であれば、サーバーは3～5台程度、実機は全体で10～15台程度を用意すればよいと考えられる。

7. むすび

ネットワークを利用して教材を共有する制御学習システムを提案し、実際に中学生の授業で評価した。その結果、1人1台の制御ロボット教材を使う環境と同等の学習効果を得ながら、使用する制御ロボットの台数を少なくすることができた。使用する制御ロボットが少ないということは、指導者の負担だけでなく、経済的な負担も軽減される。少数の教材を集中的に管理することで、手間を減らし安定した授業運営が行えることにつながる。実機での試行は、プログラムを転送し、実際にコースを走行させるための時間が必要となる。限られた授業時間で実機での試行が多くなると、プログラムを考える時間が少なくなることになる。シミュレーターを使った共有環境の方が、実機での平均試行回数が少なかったことから、実機を何度も走らせながらの個人環境よりも、プログラム作成に集中できる環境と考えられる。

実機を使った体験的な実習は、経験的に学ぶ意義も多い。例えば、実機には個体差があることを学ぶこともその一つではあるが、本システムで実装したキャリアブレイク機能は、個体差があることを学ぶ機会をなくしてしまう側面がある。しかし、個体差が大きいためその修正に時間がかかり、本来学習すべきプログラムから離れてしまうことを防ぐ効果があるとも考

えられる。

提案したシステムは、ネットワークを利用する点に特徴がある。本システムでは、生徒が作成した制御プログラムがサーバーに集められ、使用できる実機にそれぞれ転送される。手元のコンピュータがネットワークで接続されており、先生のコンピュータ(サーバー)に通信によってプログラムを送るという「システムやモデルの理解」も学習できる。このようなネットワークを利用して制御機器を共有、または遠隔操作する一般的な開発システムを体験的に学習できるシステムともいえる。

今回は教材を共有することを目的としたが、本システムはプログラムの共有や共同制作に発展させることが可能なシステムである。また、仕組みを応用することで、ネットワークや情報システムの学習が可能である。中学校技術・家庭科(技術分野)の学習内容には、ネットワークに関する学習も含まれており、今回の実習はネットワークの仕組みを理解するための実習としても利用できる。更に、バーコードリーダーやICカードリーダーなどの入力装置の仕組みや利用を考えることで、POSシステムなど身近な情報システムの学習としても利用できる。また、制御だけでなく、計測機器を共有する学習など、今回の仕組みを拡張したさまざまな教材への発展の可能性を検討していきたい。

文 献

- [1] 文部科学省, 中学校学習指導要領, 2008.
- [2] 井戸坂幸男, 久野 靖, 兼宗 進, “自律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.53, no.1, pp.9-16, 2011.
- [3] 文部科学省, 中学校学習指導要領解説, 技術・家庭編, 2008.
- [4] 紅林秀治, 青木浩幸, 室伏春樹, 江口 啓, “自律型3モーター制御ロボット教材による学習効果の検討,” 日本産業技術教育学会誌, vol.51, no.3, pp.195-202, 2009.
- [5] 古平真一郎, 坂本弘志, 針谷安男, “自律型ロボット教材を用いた「プログラムによる計測・制御」学習の授業実践に基づく学習効果の検証,” 日本産業技術教育学会誌, vol.51, no.4, pp.285-292, 2009.
- [6] 伊藤陽介, 石塚仁志, 大泉 計, 菊地 章, “ロボカップジュニア・レスキューを題材とする情報技術学習の提案,” 日本産業技術教育学会誌, vol.50, no.2, pp.59-67, 2008.
- [7] 嶋田彰子, 山管和良, 針谷安男, 鈴木道義, “自律型ロボット教材を活用したプログラムと計測・制御学習に関する授業方法の開発と評価,” 日本産業技術教育学会誌, vol.49, no.4, pp.297-305, 2007.
- [8] 伊藤陽介, 森 誉範, 菊地 章, 大泉 計, “「プログラムと計測・制御」のためのロボット学習材の開発と実践,” 日本産業技術教育学会誌, vol.49, no.3, pp.213-221, 2007.

- [9] 嶋田彰子, 柴崎 寿, 山菅和良, 針谷安男, “技術科教育における自律型ロボット教材を用いた授業実践,” 宇都宮大学教育学部教育実践総合センター紀要, 第 28 号, pp.117-125, 2005.
- [10] 兼宗 進, 中谷多哉子, 御手洗理英, 福井眞吾, 久野 靖, “初中等教育におけるオブジェクト指向プログラミングの実践と評価,” 情処学論, vol.44, no.SIG13(PRO18), pp.58-71, 2003.
- [11] 吉澤大輔, 平野秀樹, 米川 揮, 橋本政宏, 岩月正見, “工学教育用遠隔実験システムの開発と評価,” 信学技報, ET, 教育工学 104(534), 2004.
- [12] 吉田恵美, 山本茂樹, 西野洋介, 早川栄一, “ロボットプログラミング学習支援環境の実現,” 信学技報, ET, 教育工学 103(536), pp.123-128, 2003.
- [13] 兼宗 進, 久野 靖, ドリトルで学ぶプログラミング, 第 2 版, イーテキスト研究所, 東京, 2011.
- [14] 紅林秀治, 青木浩幸, ドリトル, eBASIC による計測・制御とプログラミング, イーテキスト研究所, 東京, 2009.
- [15] S. Kurebayashi, S. Kanemune, H. Aoki, T. Kamada, and Y. Kuno, “Proposal for teaching manufacturing and control programming using autonomous mobile robots with an arm,” Lecture Notes in Computer Science, LNCS 5090, pp.75-86, 2008.

(平成 25 年 1 月 14 日受付, 5 月 8 日再受付)



兼宗 進 (正員)

2004 年筑波大学大学院ビジネス科学研究科博士課程修了。博士(システムズ・マネジメント)。企業勤務後、一橋大学准教授を経て 2009 年から大阪電気通信大学医療福祉工学部教授, 2013 年から同大学情報学科教授。プログラミング言語, 情報科学

教育に興味をもつ。



井戸坂幸男

三重大学教育学部卒業。1984 年より中学校技術・家庭科の教諭として三重県の公立学校に勤務。2011 年より松阪市立飯高東中学校勤務。2013 年 3 月大阪電気通信大学大学院医療福祉工学部研究科博士後期課程修了。博士(工学)。



中野 由章

技術士(総合技術監理・情報工学)。企業においてソフトウェア開発プロセスの研究に従事し、その後、高校において情報教育、大学において情報教育・教員養成に従事。現在、神戸市立科学技術高等学校電気情報工学科教諭。専門は情報教育。



紅林 秀治

静岡大学大学院教育学研究科修了。1987 年より静岡県内国公立中学校, 工業高校に勤務。2005 年より静岡大教育学部へ現在に至る。博士(学校教育学)。