

# 計測制御システムの状態遷移を可視化する教材開発 と情報教育への応用に関する研究

著者	鎌田 敏之
学位名	博士(工学)
学位授与機関	大阪電気通信大学
学位授与年度	2019
学位授与番号	34412甲第53号
URL	<a href="http://id.nii.ac.jp/1148/00000245/">http://id.nii.ac.jp/1148/00000245/</a>



# 博士学位論文

題 目

計測制御システムの状態遷移を可視化する教材開発と  
情報教育への応用に関する研究

---

担当指導教員名 松村 雅史 印

申請年月日 2019年8月5日

申請者専攻名 医療福祉工学専攻

学 生 番 号 DL15B001

氏 名 鎌田 敏之 印

博士学位論文  
計測制御システムの状態遷移を可視化する教材開発と  
情報教育への応用に関する研究

2019年9月

鎌田 敏之

大阪電気通信大学大学院  
医療福祉工学研究科

# 目次

第1章 序論.....	1
第2章 プログラミング教育の現状と課題.....	6
2.1 中学校における計測制御プログラミング教材とその課題.....	6
2.2 新教育課程で求められる計測制御システム.....	9
2.3 プログラミング教育を担う教員養成の課題.....	14
第3章 状態でのプログラミングを可能にする学習環境の開発と学習効果.....	16
3.1 計測制御プログラミングの学びを深める教材のありかた.....	16
3.2 状態遷移をプログラムする学習環境 Rtoys の設計と開発.....	17
3.3 中学生を対象とした検証授業と結果.....	21
3.4 評価.....	28
第4章 計測値の可視化による計測制御学習支援教材の開発と学習効果.....	29
4.1 中学生に計測制御システムを理解させるためのアプローチ.....	29
4.2 開発した教材における可視化の詳細.....	31
4.3 中学生を対象とした検証授業と結果.....	34
4.4 評価.....	43
第5章 教員養成課程での状態遷移の教授法.....	46
5.1 複数の状態を持つ計測制御システムの学習における課題.....	46
5.2 状態遷移と手続きを接続するためのアプローチ.....	47
5.3 状態遷移と手順の実行を関連付ける方法.....	48
5.4 今後の課題.....	51
第6章 結論.....	52
謝辞.....	54
参考文献.....	55
関連発表.....	59

# 第1章 序論

現代の社会は、あらゆる分野でシステム化が進んでいる。計測制御システム、すなわち、計測を行い、得られた値をコンピュータ等により分析と判断を行い、制御を行うシステムも例外でなく、社会のあらゆる場所に組み込まれている[1][2]。センサ技術の進展により、デバイスの小型化が進み精度も向上したことで、これまでは計測装置の設置が困難であった場所にもセンサを置くことが可能となった[3][4]。また、センサ利用範囲の拡大により、デバイスも低価格となり、一般家庭や個人向けの小型製品にもさまざまなセンサが利用されるようになった。医療機器においても、患者の生体情報をモニタリングして蓄積し、異常の早期発見、治療方針の判断、患者への適切なアドバイスに利用されてきた。この範囲を拡大するための新たな計測装置の開発、特に使用者への負担の少ない計測装置の研究も盛んに行われている[5]。

技術の発展に伴い、学校教育にも変化が及んでいる。小学校から高等学校までの教育課程では、10年ごとに改訂される学習指導要領の内容に沿った教育が行われる[6][7][8]。これまでの日本の情報教育は、小学校段階では情報機器の活用の推進にとどまり、コンピュータを授業で用いる場面はアプリケーションを用いた表現活動や、総合的な学習の時間において、調査した結果をまとめるための道具としての利用が中心であった[9][10][11][12][13][14]。2020年度からの新教育課程では、小学校の新学習指導要領[15]において、プログラミングの考え方（プログラミング的思考）が導入され、各学年の各教科と課外活動を含めた多様な学習が可能になった[16][17]。これは、政府が進める「Society 5.0」のための人材育成として重要な意味を持っている[18][19]。表 1.1 に、新教育課程における初等教育から高等教育までのプログラミング教育の段階を示す。

表 1.1 新教育課程での初等教育から高等教育までのプログラミングと計測・制御教育の段階

学校段階	プログラミング教育	計測・制御教育
小学校	ブロック記述での体験 初歩的な反復、分岐など	入力値の有無による出力制御
中学校	反復と分岐による構造化	入力値の変化による計測制御 複数の状態を持つ計測制御
高等学校	関数、配列	(中学校と同等)
専門教育（高等学校工業科/情報科、高等専門学校、専門学校、大学の専門課程など）	モジュール、オブジェクト	伝達関数、遅れ要素、PID 制御、 運動学など

諸外国では、比較的早くから、情報科学を初等教育段階から体系的に高等教育へ接続する政策が進められてきた。しかし、先行する各国においても、特別なプロジェクトや大学等の高等教育機関との連携では事例があるものの[20][21][22]、義務教育及び専門家養成を目的としない普通教科において、計測制御システムのプログラミングを取り扱う授業はほとんど行われていないことが文部科学省委託研究による現地調査により明らかとなっている[23]。これに対し、中学校技術・家庭科（技術分野）（以下、中学校技術科と呼ぶ）の新学習指導要領においては、計測制御システムの仕組み理解を踏まえた、生徒自身による計測制御システムの構想と具体化を行う[24]とされており、システムの学習、特に計測制御システムの学習は、国際的にみても先進的であることがわかる。

中学校技術科は、諸技術に関連する初歩的な学問知識を学ぶ場面と、それを応用した「ものづくり」活動を行う実習が設定されている。電気・機械・情報の技術について学ぶ科目は他にないことから、工学の基礎を学ぶ科目といえる。計測と制御は、情報の技術のなかで取り扱われ、プログラミングにより計測と制御を学ぶ。プログラミング教育は、中学校では技術科でのみ行われる。

表 1.2 に、中学校技術科の現行と新教育課程の概要と、計測・制御についての抜粋を示す（太字は筆者による）。現行の学習指導要領では、「計測・制御の基本的な仕組みとプログラム作成」が求められていたが、新教育課程では「計測・制御システムの仕組みを理解し、計測・制御システムを構想し、計測・制御のプログラミングによって生活や社会の問題を解決」することが求められている。

表 1.2 中学校技術科の学習内容（抜粋）

現行（2011-2020）	新教育課程（2021-2030）
A 材料と加工に関する技術	A 材料と加工の技術
B エネルギー変換に関する技術	B 生物育成の技術
C 生物育成に関する技術	C エネルギー変換の技術
D 情報に関する技術 (3) プログラムによる計測制御 ア 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 イ 情報処理の手順を考え、簡単な <b>プログラムが作成できる</b>	D 情報の技術 (3) 生活や社会における問題を、計測・制御のプログラミングによって解決する活動 ア <b>計測・制御システムの仕組みを理解し、安全・適切なプログラムの制作、動作の確認及びデバッグ等ができる</b> イ 問題を見出して課題を設定し、入出力されるデータの流れを元に <b>計測・制御システムを構想</b> して情報処理の手順を具体化するとともに、制作の過程や結果の評価、改善及び修正について考える

現在、中学校で一般的に行われているプログラミングは、問題を手順に分解して考え、フローチャートを描いて手順の組み合わせを設計し、プログラムを記述して実行し、意図した動作ができているかを確認、デバッグする考え方で授業が行われている。正しい手順を組み立てることはプログラミングの基本であるが、フローチャートは単純な処理の記述に適しており、新学習指導要領が求める複雑な計測制御システムのプログラミングやデバッグには適していないことが課題となっている。

新学習指導要領解説編に記載された例では、「自動灌水を行う栽培ロボット」に並んで「買物の際に、高齢者の方を目的の売り場に誘導しながら荷物を運搬したり、障害物や路面状況などをセンサで確認し、危険な状況となった場合には注意を促したりする生活サポートロボットのモデルを開発する」例が挙げられている。後者は、工場などでの運搬作業を無人で行う「無人搬送車」を日常生活の場面に応用した事例と考えられる。無人搬送車は、移動する床に貼られた走行経路を示すテープに沿って走るが、経路が交差する箇所があるため、車は交差点を識別し、衝突しないよう他の車の接近を確認して、決まった手順で道を譲り合う必要がある。また、車に経路上で特定の動作を指示するために、マーカーと呼ばれるテープが経路近くに貼られ、車は経路以外にも、マーカーを識別して、所定の動作をしなければならない[25]。これらの例は従来のフローチャートによる状態を持たない記述では表現することが難しく、中学生が理解して活用できる新しい教育法が求められている。

本研究では、中学校技術科で扱ってきたフローチャートなどの手続きの考え方による「計測・制御のプログラミング」の教育方法を状態遷移の考え方をを用いて発展させることで、新教育課程の「計測・制御システムのプログラミング」に対応した教育を可能にする教材および教育法を提案する。

状態遷移モデルそのものは、状態を考え、イベント発生に対してガード条件を満たせば遷移するというシンプルな考え方であるため、企業におけるシステム開発では盛んに利用されているが、従来の高校までのプログラミング教育では、状態遷移はほとんど扱われてこなかった。図 1.1 と図 1.2 に、これからの中学校でのプログラミング教育において状態遷移モデルが必要となる意味を、ソフトウェア開発プロセスの観点から整理した図を示す。

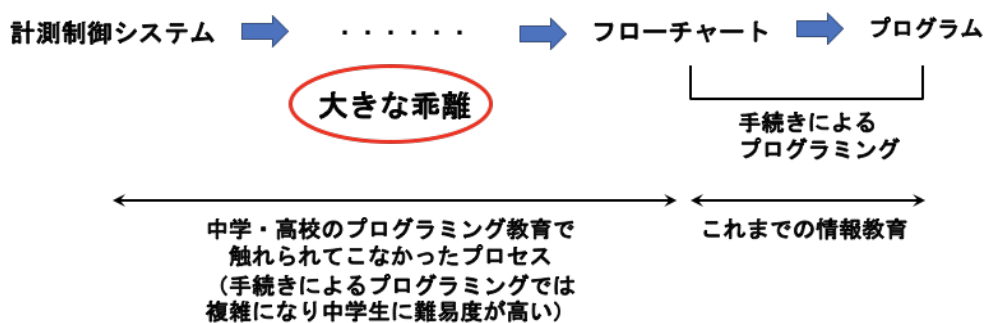


図 1.1 計測制御システムのプログラミングにおける従来の課題

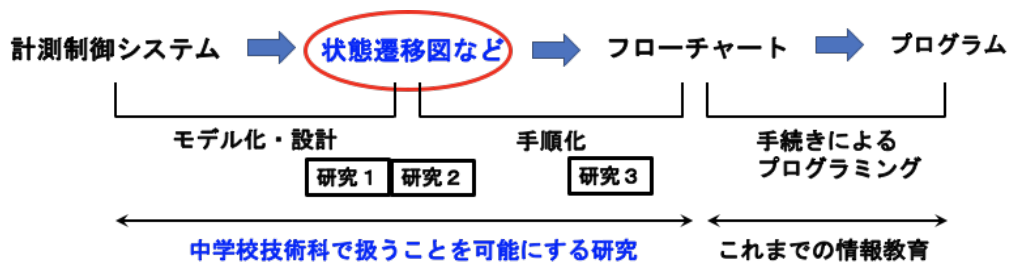


図 1.2 本研究における提案

すなわち、従来の手続きのみによるプログラミング教育では、手順をフローチャートによって設計し、プログラミング言語により手続きを記述してコンピュータに実行させ、デバッグさせる、という、図 1.5 の右側の範囲のみを扱ってきた。しかし、計測制御システムのプログラミングには、システムが持つ複雑性を克服するためのプロセスが必要であり、これが従来は欠けていた。一般に、ソフトウェア開発では、問題を分析してモデル化するプロセスが存在し、モデル化による分析結果をもとにプログラムの設計を行う。しかし、中学校・高校のプログラミング教育には、今後必要となる、複雑さを持つ計測制御システムについて考えられる有効な手がかりがなく、フローチャートのみで考えなければならないという課題がある。このことで、生徒は、センサ入力から車を思い通りに制御したり、プログラムを修正してデバッグしたりすることが容易でない状況に置かれている。

そこで、本論文では、中学校の計測制御システムの学習環境として、中学生が、連続的な値の入力や、複数の状態を持つ計測制御システムの振る舞いを可視化によって理解し、状態遷移を考えながらプログラムできる教材の開発を目的とする。具体的には、次のような内容を中学生が学習できるようにするための教材開発である。

1. 中学生が計測値の連続的な変化を理解して制御するシステムを理解できる
2. 中学生が複数の状態を持つ計測制御システムを理解できる
3. 中学生が複数の状態を持つ計測制御システムをプログラミングできる

本論文は、以下の 6 章で構成される。

第 1 章は序論であり、以上のような研究の背景、研究の必要性につながる計測制御のプログラミング教育における課題と、それを解決するための研究のアプローチについて述べる。

第 2 章では、中学校技術科で使われている教材と、国内及び海外の研究を調査することで、技術教育の現状と課題について述べる。また、この状況に対して果たすべき教員養成の役割について述べる。

第 3 章では、複数の状態を持つ対象をプログラミングできる言語 Rtoys を提案する。その学習効果



について公立中学校において検証した結果をもとに、中学生が複数の状態遷移を考え、プログラミングできることを示す。

第4章では、「センサによる計測値をコンピュータが処理・判断し、アクチュエータを制御する」という計測制御システムの基本的動作をプログラミングによって体験的に学習する新しい教材開発について述べる。その学習効果について公立中学校において検証した結果をもとに、中学生が計測値の連続的な変化により制御するシステムを理解でき、複数の状態を持つ計測制御システムの動作についても理解できることを示す。

第5章では、状態遷移の考え方を学校教員が理解して授業で扱えるようになることを目的として、また状態遷移モデルと手続き型プログラミングの関係を教師・生徒双方が理解させるために、状態遷移図から手続き的なフローチャートへの変換手順を提案する。これを教職課程で学ぶ大学生への実践的教育に導入し、その教育効果について述べる。

第6章で、全体を総括し、結論を記す。

## 第2章 プログラミング教育の現状と課題

本章では、本論文が対象とする中等教育における情報教育、特に、中学校におけるプログラミング教育と、中学校において必修である計測制御プログラミングに関わる現状と課題について、従来の中学校で計測と制御のプログラミングに利用されてきた教材と、取り扱われる題材について、先行研究を踏まえ、課題を明らかにする。一方で、新学習指導要領が求める計測制御システムの具体化を指導できるよう、教員側が状態遷移の考え方を理解し、活用できる必要がある、したがって、現職教員の研修を見据えつつ、教員養成大学生を対象とした、プログラミング教育における課題の解決方法について検討を行った結果を示す。

### 2.1 中学校における計測制御プログラミング教材とその課題

#### (1) フローチャートによるライントレースの記述

図 2.1 に山崎教育システム「制御学習プロロボ USB プラス」を示す[26]。フォトリフレクタを用いたライントレース制御など、ロボットの処理の流れをフローチャートで記述する教材である。「プロロボ」は、検定教科書[27][28]にて複数の単元学習で使用教材として取り上げられ、技術科教員の間では標準的教材と認知されている。

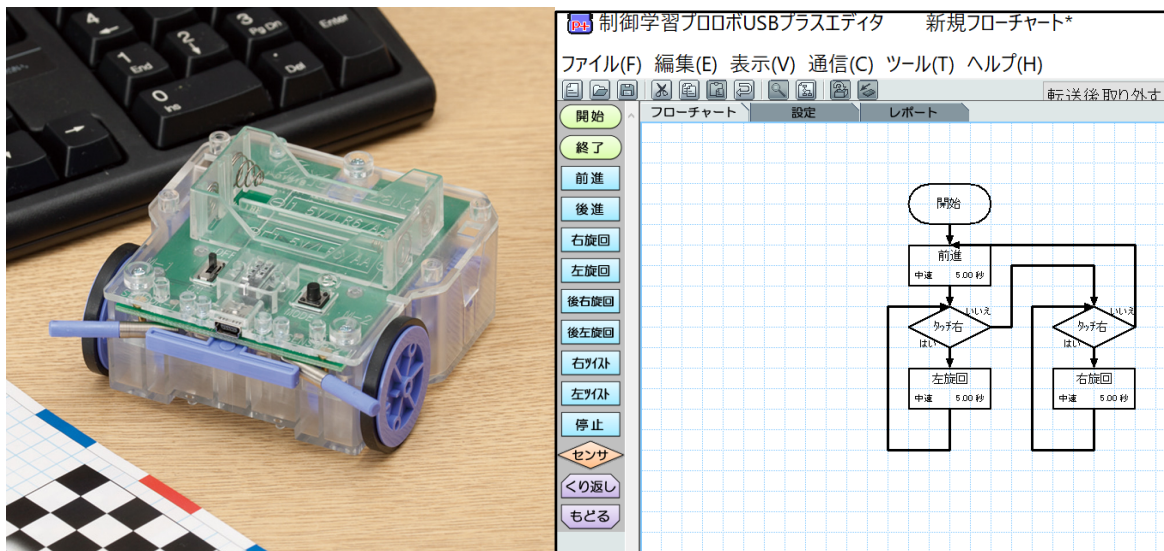


図 2.1 プロロボ USB プラス (ヤマザキ教育システムの Web ページより)

この教材は、スイッチを用いた接触センサもしくは別売の光センサを用いた光反射の有無を 2 値の計測値として用い、PC 上でフローチャートとして描いたプログラムを制御基板にコード化して書

き込み、PC から切り離して実行するマイコン搭載自走車である。光センサの計測値の 2 値化は基板上で行われており、利用者がしきい値を変えることはできない。図 2.1 (左) にプログラミングとプロロボへのコード書き込みを行うソフトウェアの画面を示す。プログラミング操作は、画面左側のアイコンからフローチャート記号を引き出して配置し、それらを線で結合するという、手で紙に作図する順序に似た手順である。フローチャート記号に対応した命令は、処理はロボットの走行に限られ、反復は回数指定のみ、条件は 2 値センサの入力の有無に限られている。フローチャートの作図がプログラミングにあたるため、プロロボのプログラムは、センサ入力による分岐と反復の入れ子を図で描くことになる。生徒にとっては、線は自由に描いていくことができるため、分岐先や反復が増えると、行き先を示す線をどこへ向かわせればよいのかわかりづらい可能性がある。

## (2) ブロックの並びによる計測値に対する動作の記述

図 2.2 に久富電機産業の学習教材「オーロラクロック」[29]を示す。図 2.3 左に示すように、アラームクロックの形状をしており、温度・光・音のセンサを内蔵し、高輝度のフルカラーLED を内蔵している。この教材を用いた授業として、複数の本教材を縦横に並べた、カラーイルミネーション作品の制作が広く行われている。教室の生徒が分担して 6~10 台の本教材を用いたカラフルなイルミネーションをプログラムし、生徒の手拍子を音センサで感知すると、同時に光の点滅と音の演奏を始めるものを作成するという内容である。この授業では、音センサはイルミネーション開始の手がかりのみに使われている。ひとりの生徒の活動は、作品制作する班の一員として、作品全体の表現のために、他の教材とタイミングを合わせて光の色や明滅を考えることが中心となる。図 2.3 中と図 2.3 右にオーロラクロックのプログラミング画面を示す。プログラミングのためのソフトウェアには、フローチャート作成画面との往復も可能な、LED の点灯・消灯時間と反復のみからなるタイルを並べるプログラム作成画面があり、分岐のない、LED イルミネーション作成に使いやすい画面である。実践の多くは、このタイル画面を利用していることが考えられる。

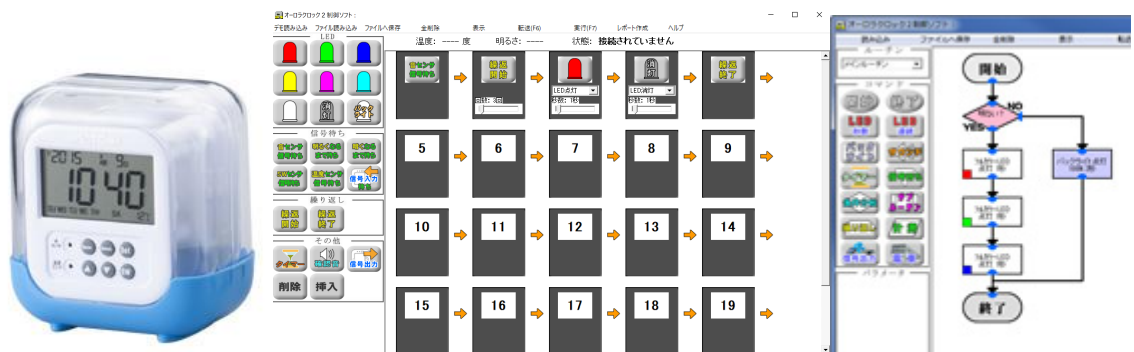


図 2.2 オーロラクロック (久富電機産業の Web ページ[29]より) とプログラミング画面

### (3) ブロックの並びによる計測・制御の記述

複雑な計測制御のプログラミングが可能な既存教材として、図2.3にLEGO MindStorms EV3[30][31]を示す。EV3は、RCX, NXTに続く第3世代のLEGO MindStormsであり、4入力3出力の端子を持つほか、USBポートとSDカードスロットがあり、USB DongleによりBluetoothやWi-FiによりPCとの通信を行うことができる。赤外線通信機能も搭載している。ドットマトリクス液晶パネルと6ボタン入力があり、液晶画面に絵を表示するほか、ボタン操作によって、センサ計測値の実時間表示のほか、計測値のデータログをグラフ表示することもできる。標準のセンサとして、ジャイロ、超音波距離センサ、焦電型赤外線センサ、カラーセンサがあり、電氣的・機械的規格さえ合わせればCで開発した独自のモジュールをプログラミングで用いる新たなブロックとして追加させることもできるほか、C、Java、C#、Python、LabVIEWを含め30以上のプログラミング言語がコミュニティや提携企業により提供されているため、大学の教育研究において、汎用のコンピューティングモジュールとしての利用も行われている。これを中学校の授業で利用する場合の最大の問題は価格である。また、教師がLEGOによる組み立てを学ぶ時間の不足や、できることが多すぎるため、何が授業に利用できるのか考えることが難しいという問題もある。プログラミングの面でも、複雑なプログラムを作成する場合に、標準のブロックプログラミング環境がフローチャート風の作図であることが、教師にとって、授業設計の困難さをもたらすことも想定される。



図 2.3 LEGO MindStorms EV3 による製作例 (LEGO education 社のユーザーガイド[31]より)

## 2.2 新教育課程で求められる計測制御システム

### 2.2.1 状態を持つ計測制御の必要性

床のテープを経路として、その上を走ることをライントレース (line following) と呼び、現在の中学校の計測制御プログラミングでは、フィードバック制御の学習教材として広く実習に取り入れられている。ライントレースさせるために、模型自動車型ロボット (以下、ロボットと呼ぶ) が使われる。図 2.4 に、開隆堂書店の検定教科書に掲載された例を示す。

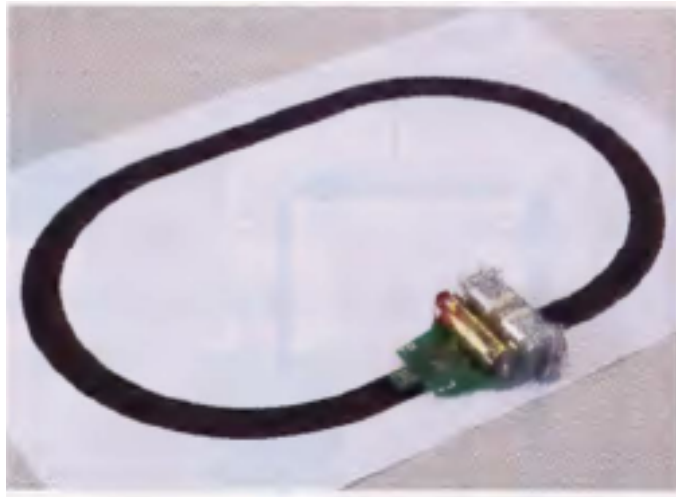


図 2.4 ライントレースロボット (文献[6]より)

ここでのライントレースロボットは、ロボット前方下部に取り付けられた、2つの光センサを入力として、ラインをたどるよう方向を調整しながら走行する。このとき、2つの光センサがラインを挟むよう設置されており、図 1.1 のように、白地に黒のラインが描かれた上を走行する場合は、以下の3つの動作からなる。

1. 前進しながら左右の光センサの計測値を取得する
2. 左センサが黒であれば、ロボットは左に回転する
3. 右センサが黒であれば、ロボットは右に回転する

ロボットがコースを外れた状態から回復する必要がある場合に 2.または 3.の動作をし、そうでない場合は 1.を実行することを繰り返す、という、非常にシンプルな動作である。ひとつのセンサを使い、ラインの右端に沿って走るライントレースの場合の動作は、以下のようになる。

センサの計測値が白であれば右前進し、黒であれば左前進する。これを繰り返す

床から得られた光センサの値が、白ならばライン端に近づくよう左寄りに前進し、黒ならばライン

の右端に近づくように右寄りに前進する。この動作により、ラインの右端を目指しながら前進する。図 2.5 に、フローチャートで描いたラインレースの手順を、図 2.6 に、C 言語的に記述したプログラムを示す。

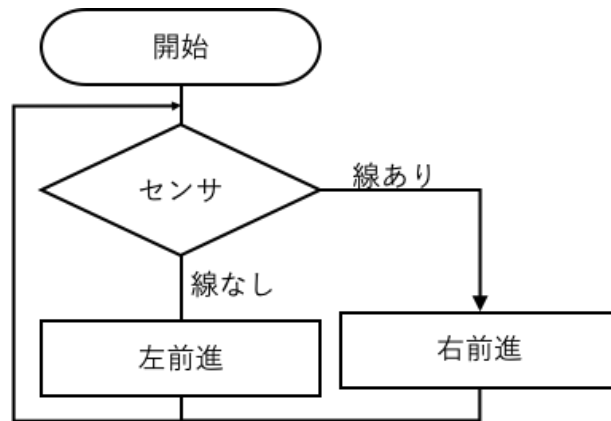


図 2.5 ライントレースの手順をフローチャートで描いた例

```

While() {
    if (センサ()==0){
        右前進();
    } else {
        左前進();
    }
}
  
```

図 2.6 ライントレースを C 言語的に記述した例

センサの値として得られる制御量の 2 値を 0 (黒) と 1 (白)、操作量として右に寄せることを 1、左に寄せることを -1 として表し、ロボットの軌道を描いたのが、図 2.7 である。

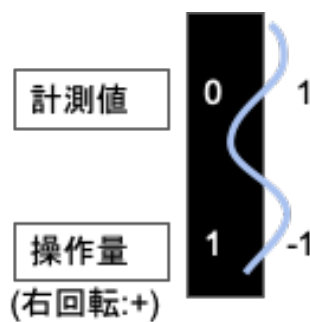


図 2.7 センサ 1 つのラインレースにおける制御のモデル



このモデルは、ライン右端を目標値とした、On-Off 制御であるといえる。計測値は 2 値であり、制御量も 2 値である。この学習によって、エアコンや冷蔵庫の動作をライントレースするロボットの軌道のように、設定温度を目標値として、冷却であれば、目標温度を超えた場合に冷却器を作動させ、下回った場合に冷却器を停止するものとして無理なく理解させることができる (図 2.8)。

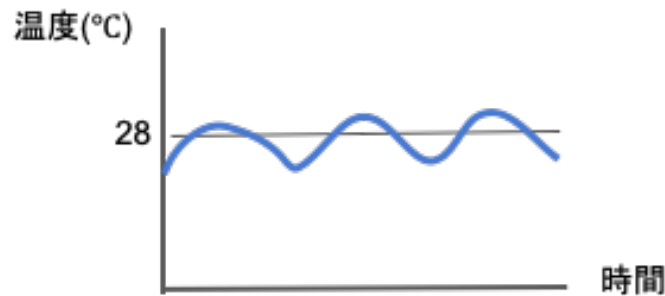


図 2.8 エアコンの温度制御のモデル

しかし、新学習指導要領解説編に示された無人搬送車のモデルは、ライントレースを基本とするものの、交差点での道の譲り合いや、マーカーを読み取って所定の動作をするという、ひとつの制御では実現不可能な複雑な制御が必要となる。さらに、On-Off 制御を 2 値の計測値で行うことは、フィードバック制御における「目標値に近づくよう操作量を変化させる」ことが困難であり、目標値の近くで振動が生じ、安定した制御とならない。そこで、連続的な計測値に応じた制御を行うよう、センサの計測値と操作量の両方を連続量として扱う必要がある。

無人搬送車のマーカーに応じた動作を実現するには、マーカー検出をイベントとして、対応する動作をするような論理的な挙動をするように、コンピュータのプログラムを作成する必要がある。交差点やマーカー、あるいは人の接近を検出して停止するなど、イベントの発生は不定期に発生する。そして、イベントが発生した場合には、すぐにイベントに応じた動作を実行しなければならない。新学習指導要領解説編で示された例は、このようなシステムの設計と具体化を求めている。フローチャートのみで、正しく動作するように設計することは非常に難しく、設計の誤りにも気づきにくいいため、デバッグも容易でない。このような離散イベントを扱うには、「状態遷移モデル」が適している。状態遷移モデルは、トランスデューサ有限状態オートマトンによって意味づけられた、コンピュータの動作を状態と状態の遷移によって説明するモデルである。以下、無人搬送車がマーカー入力によってライントレースを止め、必要な仕事をする例を用い、状態遷移モデルを説明する。状態遷移は図表現することができ、分析の強力なツールとなる。本論文では、状態遷移図の表記に、UML [8] のステートマシン図を用いる。

最初に、図 2.4 のような楕円コースのライントレース動作をひとつの状態「走行」とする (図 2.9)。

図上部の黒丸が「スタート疑似状態」であり、即座に矢印の先、「走行」状態に遷移する。ひとつの状態は、他の状態に遷移しない場合、状態に定義された処理（アクション）を反復実行する。したがって、この状態遷移図は、ライントレースを実行し続けるプログラムに対応する状態遷移モデルである。

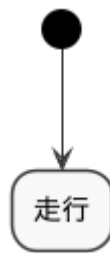


図 2.9 ライントレースしている状態（1 状態のみ）

次に、無人搬送車がマーカーによって対応する制御を変える動作を加える。楕円コースの途中の 2 箇所にマーカーを置き、一方では荷待ち、他方では荷降ろしを行うとする。荷待ち、荷受け、いずれも完了するまで停車しているとし、荷が台車に乗っているかどうかは、光センサによって検知できるとする。このイメージを、図 2.10 に示す。

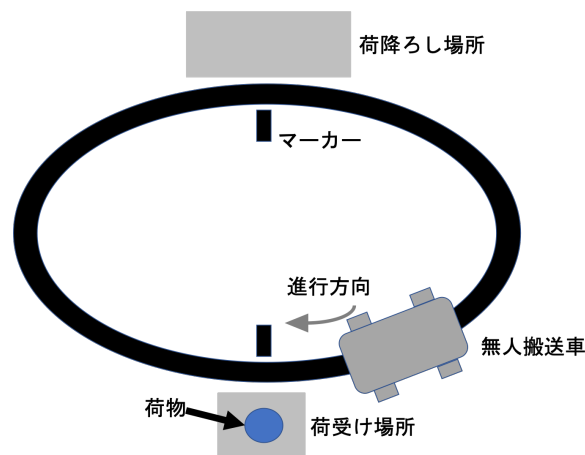


図 2.10 荷物を無人搬送するコースのイメージ

この動作には、「走行している・停車している」の 2 種類の動作と、それぞれに対して台車に「荷がある・荷がない」の 2 つの状態の区別がある。したがって、状態数は 4 となる。最初は荷がなく、マーカーもない場所から始まるとすると、「走行」状態から始まることになる。マーカーの検出と、荷の積み下ろしがイベントとなるので、コースに沿って、ライントレースと停車を交互に行いながら、一方のマーカーの場所で荷を積み、他方のマーカーの場所で荷を下ろすことを繰り返す動作となる。一般に状態遷移図で状態名に重複は許されないので、荷を載せてライントレース走行する状態を、「搬送中」状態とした。また、荷の有無はセンサの値により検出されるため、荷台の変化というイベントに対し、センサの値がどちらであるかの条件をガードとして加えることにした。その結果、



状態遷移図は、図 2.11 のような、コースと同様の円環する遷移となる。

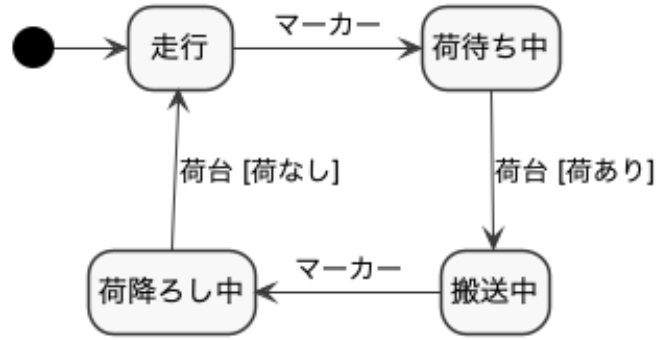


図 2.11 自動搬送車の簡易動作モデル

さらに、同じコースとマーカーで、荷積みの場所で3秒経過したとき、荷があろうとなかろうと発車してしまう動作に変更した場合を考える。この場合、荷待ちをしたあと荷がないまま車が走る可能性がある。すると、荷の有無に関わらず、ライントレースであることは同じであるため、荷の有無で状態を区別することができなくなる。また、荷降ろしする場所でマーカーを検出しても、荷がなければ停車せず通過すべきである。したがって、状態は、ライントレースをしている状態、荷積みする場所で荷を待つ状態、荷降ろしする場所で荷降ろしをする状態、の3状態となる。イベントは、さきのマーカーの検出と荷の積み下ろしに加え、3秒経過の4つとなる。ここで、「荷待ち中」状態と「荷降ろし中」状態への状態遷移が、マーカー検出のイベントによって生じるが、「荷がない・荷がある」というセンサの値によって「荷待ち中」状態か「荷降ろし中」状態のいずれかへ分岐する必要がある。UML ステートマシン図では、このような場合、「choice 疑似状態」を経由させる。分岐後の遷移はイベントの発生を待たずにセンサの値を調べる必要があるため、ガード条件のみ記述する。イベント `after(時間)` は、括弧内に指定した時間が経過した後に遷移するイベントを表す UML での記法である。その結果、図 2.12 のような状態遷移図となる。

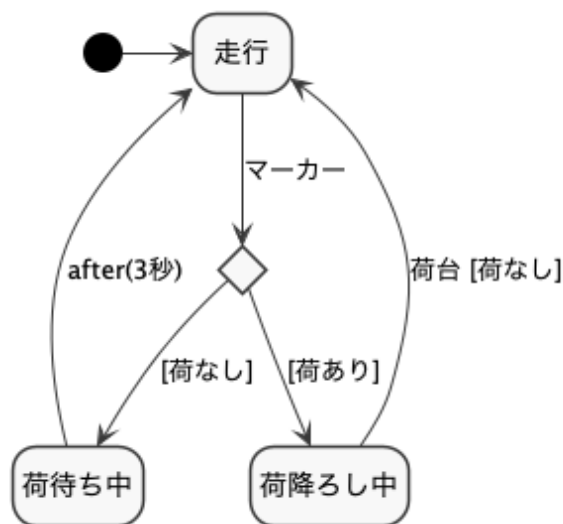


図 2.12 荷積みを持たずに発車する無人搬送車のモデル

## 2.2.2 状態遷移モデルとモデリング技能の学習困難さ

これまで述べてきたように、状態遷移は有用な考え方であるが、実際の利用場面で、正確なモデルを作成し、ハードウェアやソフトウェアの作成に活用できるまでの学習は、職業エンジニアにとっても困難といわれている[32]。

現在のソフトウェア開発において、モデルの記述にはオブジェクト指向分析を前提としたモデリング言語として統一モデリング言語 (UML) [33]が用いられる。UML は Object Management Group により策定された図形による形式的表現であり、現在も改訂が進められているが、ISO/IEC 及び JIS にも採用されている標準規格でもある。UML の図は、システムの静的な構造を表す構造図と、動的な振舞を表す振舞図の2つに大別される。

UML では状態遷移の図表現として、Harel による StateChart [34]記法を用いた「ステートマシン図」が定義されている。また、処理手順の図表現にはステートマシン図との対応を考慮した図形を用いる、「アクティビティ図」が定義されている。

モデリングの困難さは、問題の抽象化という思考能力の他に、実際のソフトウェアやハードウェアに求められる動作を満たす、適切な抽象化を問題の性質に応じて考える能力及び経験が必要だからである。香山ら[35]もまた、初学者向けのモデリング教育方法論の確立が未だなされていないとして、プログラミング経験のない初学者の大学生を対象に、日常の事象をステートマシン図で描かせる課題を与え、作成する際に生じる誤りの分析を通して、有効と考えられる教育方法について考察を行っている。ただし、香山のこの研究では、自身の研究グループによる学習ツール[36]により自動生成された LEGO MindStorms NXT のプログラムの動作と対比し修正する活動により作成されるモデルの品質が向上したとしており、机上で思考法を学ぶだけではモデリングの学習は難しく、実際のプログラムの挙動を観察しながらモデルを修正する活動が、モデリングの正確な理解に不可欠であることを示唆していると考えられる。

## 2.3 プログラミング教育を担う教員養成の課題

新学習指導要領が求める、生徒自ら課題と考えたことをプログラムによって解決する学習のプロセスは、情報科学に関わる専門的教育を受けている、ソフトウェア開発の実務経験を持つなどの、ごく一部の教員を除いて、現役の教員にとって未知の経験である。また、授業として成立させるために有効な、状態遷移の考え方を会得している教員もまた、ごくわずかな人数にとどまっている状況と考えられる。授業のなかで、困難に陥った生徒が、何につまずいているのかを見極め、問題をどのように捉えるか、もしくは生徒の能力に見合った問題に変更すれば解決しやすくなるのかを支援するこ

とは、教師の仕事である。よって、プログラミングを指導する教員は状態遷移の考え方を理解し、使いこなせる状態になっていなければならない。

そこで、教師にとって必要な理論と実践を学んでいる途上にある、教員養成段階の大学生を対象とした効果的な教授法を開発し、その効果を検証しながら、教員研修への応用を探ることが目下の重要課題であるといえる。

本章の内容をまとめる。中学校技術科では、計測制御システムの理解を目標に設定していたものの、これまで、フローチャートにより処理手順を考えさせ、プログラムを作成して実行するという手段のみが用いられ、手続きの考え方のみで成立する基本的な内容にとどまっていた。このため、生徒が計測制御システムの仕組みを理解できていたかは疑問である。状態遷移の考え方を中学校段階で取り組む研究は限られており、指導法も確立していない現状がある。したがって、生徒が状態遷移の考え方を使って計測制御をプログラミングできる教材と指導法の開発と、状態遷移の基本的な考え方を教員や教員志望の大学生が身につけられる教授法の開発が必要である。さらに、コンピュータ動作の仕組み理解に直結する手続きの考え方を完全に捨て去ることはできない。そこで、状態遷移モデルにより問題を分析したのちに、それを手続きに変換していく効果的な指導法の開発も必須の課題である。

# 第3章 状態でのプログラミングを可能にする学習環境の開発と学習効果

本章では、生徒が状態遷移概念を理解しながらプログラミングを行う学習環境として、状態遷移をプログラムとして直接記述できる言語を提案する。この言語では、マウスのドラッグアンドドロップにより、状態を示すウィンドウ内に動作を指示する命令語を配置することが可能である。この言語と車型の自律型制御ロボット教材を使った授業で、学習効果を検証する。

## 3.1 計測制御プログラミングの学びを深める教材のありかた

第2章でみたように、状態遷移概念を、最初に抽象的な知的活動であるモデリングから正確に理解させることは難しく、実際にプログラムの動作を確かめながら、徐々に正しい概念の形成を学習していくのが現状において最善の効果を発揮する方法であると考えられる。一方、通常の手続き型プログラミング言語を操作しながら状態遷移モデルとの対応を理解することは、特に初学者にとっては適切な助言なしでは難しい。そこで、仮説として、「プログラミング言語を、状態を図形として表現し、そこに振舞と状態遷移の命令を配置していくビジュアルプログラミング言語として学習者に提供することにより、状態遷移が必要となるような複雑さを持つプログラムを、学習者自らが作成できるようになる」と設定した。

これを検証するために、比較対照実験を行うことにした。手順として、まず、学習の課程を考慮して、徐々に手続き型プログラミング言語では課題達成が困難になる課題を作成した。次に、提案する状態遷移プログラミング言語とほぼ同じ操作の手続き型ビジュアルプログラミング言語を用意した。

ただし、学校の授業で比較対照実験を行うことは、生徒の学習の権利を奪う倫理的問題が発生するため、以下の措置により問題を最小限にとどめる対応をとる。

1. 状態遷移概念を説明する資料を授業の同じ時点で配布する
2. 手続き型プログラミング言語で状態遷移と対応づける仕組みを命令として提供する
3. 配布資料は、状態遷移用の資料にほぼ対応するような説明を記述する

すなわち。資料からその使い方が理解できれば、状態遷移プログラミング言語と同程度の難易度で課題解決ができる配慮である。

## 3.2 状態遷移をプログラムする学習環境 Rtoys の設計と開発

### 3.2.1 言語の設計方針

計測制御プログラミングの教材として、すでに中学校で広く利用されている市販教材と同様に、車型ロボットにプログラムを転送しながら課題解決させる形の学習環境が適当であると考えた。しかし、市販教材はプログラムされる模型車の動きを転送前に確かめることができないため、生徒の考えた手順が物理的な理由で実際の動作と一致しない場合、試行錯誤を余儀なくされていた。学習の初期段階において試行錯誤は必要であるが、時間に制約のある授業内では、生徒への結果のフィードバックは早いほどよい。そこで、画面上でロボットの動作をシミュレーションできる機能を持たせることで、早いフィードバックが実現できると考えた。

計測制御システムの学習を効果的に行うためには、生徒が当初困難と感じる課題を自ら解決できたという達成感を与えることが重要である。このような課題では、状態遷移の考え方が解決に有利である。そこで、状態遷移をプログラミング言語の仕様に加え、状態を直接プログラミングしながら、状態遷移を含む手続きを状態内での動作として記述する、ハイブリッド言語が中学生に理解しやすいと考えた。

これらの条件を満たすプログラミング言語を検討したところ、Squeak Etoys [37]が注目に値することがわかった。この言語は、子どもが体験を通して数理的な思考を学ぶための道具として設計された LOGO [38]言語の思想を受け継いだ新しい言語、Squeak [39]の上に作られたビジュアルプログラミング言語であるためである。Etoys はシミュレーションのために並行処理が言語に追加されており、意図している教材の開発に適していると考えた。また、ビジュアルプログラミング言語による操作も生徒に受け入れられやすいと予想した。

そこで、Etoys をもとに、Etoys の図形に付加するプログラム「スクリプト」に着目した。Etoys で、ひとつの図形は複数のスクリプトを持つことができ、同時実行すなわち並行処理も可能である。よって、スクリプトを状態に見立て、状態の遷移は該当スクリプトへ処理を引き渡すことで、状態遷移を実現する方針とした。さらに、画面上を走り回る MYUCAR オブジェクトを定義し、ロボットカーの走行やビーブ音を発するメソッドを定義し、Etoys の命令図形「タイル」として扱えるようにした。MYUCAR のスクリプトをまとめてミュウロボ制御基板のバイトコードプログラムに変換し、保持する機能と、シリアルポートを介して基板へバイトコードを書き込む機能を追加した。ロボット制御の学習には不要なタイルを表示しないよう調整し、ロボット用ビジュアルプログラミング言語とした。Etoys の派生であることを明確にするため、Rtoys と名付けた[40]。

比較対照実験のために、Rtoys から状態遷移に関連する変更を表示から隠し、手続きの考え方ながら、状態遷移の考え方に対応づけられる、無限ループと、ループからの脱出命令を追加した Rtoys も開発した。以下では、状態遷移の考え方に基づく方を「状態 Rtoys」、手続きの考え方に基づく方を「手続き Rtoys」と記して区別する。

### 3.2.2 状態 Rtoys

状態 Rtoys を用いて、開発した Rtoys の画面構成と、操作の概要を説明する。

図 3.1 が、Rtoys の画面である。プログラミングと実行、どちらもこの画面内で行う。起動時には図の左上にいる MYUCAR の図形のみが、画面左側に存在するのみである。MYUCAR は教材のハードウェアである模型車にあたり、状態遷移やロボット制御のタイルを持っている。MYUCAR をマウスオーバーすると、図 3.2 のように周囲に「ハロ」（後光）と呼ばれる、拡大縮小などの操作をするアイコンが周囲に現れる。左列中央の「目」が描かれたアイコンをクリックすることで、画面右側に、タイルの一覧である「ビューワ」が現れる。ビューワ内にある各タイルの左にある「！」アイコンをクリックすることで、前進など、対応するロボットの動きを画面上で確認することができる。ビューワからタイルを画面内にドラッグアンドドロップして配置するのが、Rtoys のプログラミングの基本操作である。

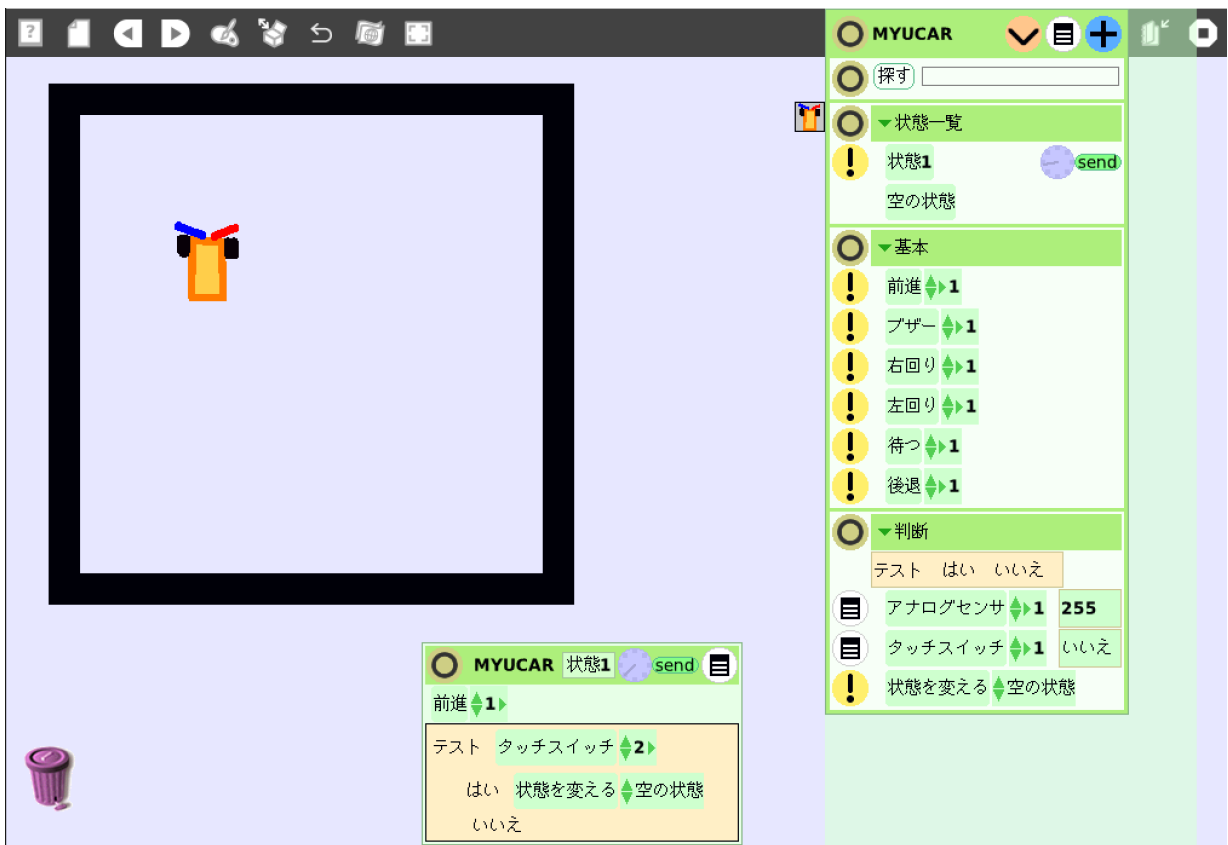


図 3.1 状態 Rtoys のプログラム画面



図 3.2 「ハロ」と Rtoys の「ビューワ」

タイルのどれかひとつを画面の背景上にドラッグアンドドロップすることで、「状態 1」の名前のウィンドウが同時に現れ、内部に、そのタイルが最初の動作として配置される。不要なタイルは、画面左下のゴミ箱にドロップすることで消すことができる。ビューワから追加のタイルを枠のなかに加え、実行順をドラッグアンドドロップ操作で入れ替えることができる。数値や文字列を左クリックすると色が変わり、直接キーボードから入力した内容に置き換えることもできる。

状態の追加は、任意のタイルを画面の背景にドロップすることで作成され、「状態 2」などの連番付きの名前が自動で与えられる。状態の名称もクリックして、キーボード入力した内容に変更できる。状態遷移の命令は、ビューワの最下部にあるタイル「状態を変える」である。引数「空の状態」は実行終了を意味するので、矢印のクリックで遷移先の状態名に変更する必要がある。

状態遷移しない場合は、枠内の動作を上から下へ向かって順に繰り返すため、利用者が混乱しないよう、反復命令のタイルは表示していない。

図 3.3 にプログラム例を示す。障害物回避である。プログラムは 3 つの状態からなり、「直進」は直進したあと、左右のタッチスイッチを調べ、障害物を検出した場合、それを回避するどちらかの状態に遷移する。回避の状態では回避行動のあとタッチスイッチを調べ、障害物がなければ直進状態へ遷移する。

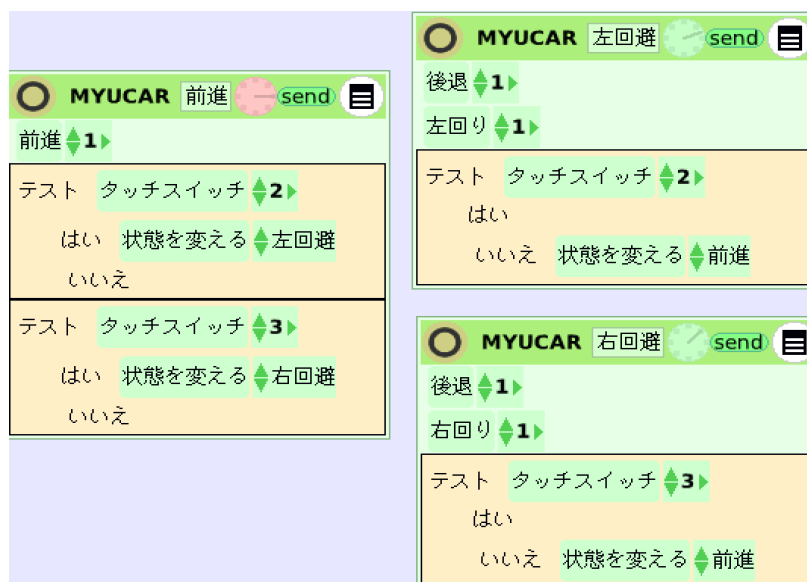


図 3.3 状態 Rtoys による障害物回避のプログラム

### 3.2.3 手続き Rtoys

手続き Rtoys では、手続きによるプログラムの反復を意識させるために、無限ループのタイル「繰り返し」ブロックを作成し、ビューワで表示できるようにした。状態はひとつの制御の反復であることを明示するため、スクリプトの実行を変更した。Etoy では状態 Rtoys のようにスクリプトを反復実行するが、「繰り返し」ブロックを使うよう、スクリプトは一度実行した後、停止する。また、「繰り返し脱出」ブロックを作成し、ループからの飛び出しにより次の処理の実行ができるようにした。手続き Rtoys は検証授業で、状態 Rtoys と同等な学習を保障することを重視した設計であり、ビューワ内の位置も同じ最下部である。ウィンドウの名称は、Etoys と同じ「スクリプト 1」が自動的につけられる。

図 3.4 は、手続き Rtoys による、ラインレースのプログラムである。1 センサによりラインの右端を走るモデルである。光センサのアナログ値としきい値の比較により、左の光センサが黒であれば右回りし、白であれば左回りしながら前進する。

ロボットを動かし続けるための無限ループがあり、そのなかでセンサの値を調べながら動きを変化させている。



図 3.4 手続き Rtoys によるラインレースのプログラム



### 3.3 中学生を対象とした検証授業と結果

本教材の学習効果を検証するために、中学生を対象とした、状態 Rtoys と手続き Rtoys による比較対象実験の授業結果について述べる。

#### 3.3.1 開発した教材のハードウェアとソフトウェア

ハードウェアは、2006 年に開発し商品化も行った計測制御基板[41]に、左右との接触を検知できるよう、ワイヤとビスを接点に用いた接触センサを入力端子に取り付け、2 個のモータを用いた車型の自律型制御ロボットに仕立てた「ミュウロボ」[42][43]を用いた。図 3.5 にその外観を示す。基板の入力端子はアナログ値の計測もできるが、接触の有無をデジタル値として扱うことにした。

ソフトウェアは、前述した状態 Rtoys と手続き Rtoys である。ミュウロボ制御基板の単位時間が 0.1 秒であるため、Rtoys のロボット移動命令の引数は、0.1 秒単位の時間を表す。シミュレーション画面の動作もほぼ同じであるが、移動する距離や角度について、物理的な慣性や摩擦などに基づく挙動については、ミュウロボ個体ごとのばらつきが大きいため、一切考慮しないことにした。

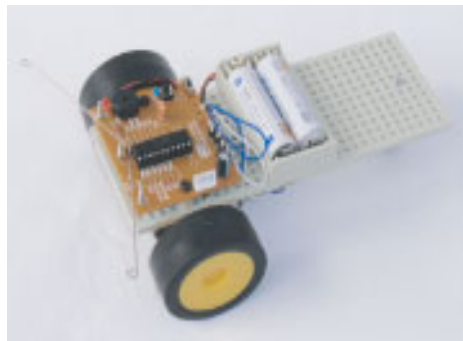


図 3.5 接触センサを取り付けたミュウロボ

#### 3.3.2 授業の構成と内容

検証授業は、公立中学校で、3 年生の技術科の授業 2 クラスで実施した。時間割が違うため、2 つのクラスの生徒が同席することはない。各クラスに同じ制御ロボットと課題を与え、ソフトウェアだけを変えた比較実験とした。データは 欠席者を除き、学力テストのクラス平均点が等しくなるよう抽出した生徒からなる、各クラス 15 名ずつを分析した。

授業の構成を、表 3.1 に示す。ロボットが持つ左右 2 つの接触センサを用いて、5 つのステップを踏んで難易度が高まる構成とした。

表 3.1 検証授業の構成

ステップ	状態	課題
1	なし	接触の有無で回転方向が変わる
2	なし	接触がない間は前進、ある間は停止
3	一方向の2状態	左回転の間に接触があれば右回転し始める
4	往復する2状態	接触した方向に回転方向を変える
5	往復が2つある3状態	障害物回避しながら走行

学習を円滑に進めるため、Rtoys の操作や状態遷移の理解を補助する資料を作成し、課題の設定やロボットの動きを詳しく説明した。

### 3.3.3 検証授業の詳細

#### ステップ1: 多角形を描くように走行

ステップ1は、Rtoys の操作に慣れることを目的に、反復だけのプログラムを課題とした。具体的内容は、前進と右回りをを用いて、多角形を描くように走る動作である。図 3.6 に示すように、状態 Rtoys では、状態名の右にある円（時計）をクリックすることで反復実行が始まる。手続き Rtoys では、時計が実行開始の意味しかなく、「繰り返し」ブロックを置いて反復実行をプログラムに含める必要がある。多角形については、ミュウロボで閉じた多角形の軌道を走らせることは難しいため、軌道の形状については問わないことにする。



図 3.6 多角形を描くプログラム

#### ステップ2: 接触がない間は前進、ある間は停止

ステップ2は、反復に加え、条件分岐を扱うプログラムを課題とした。Rtoys では、条件分岐には

「テスト」タイルを用いる。具体的な課題の内容は、左のタッチスイッチ（以後、左センサと呼ぶ）に接触がない間は前進、接触のある間は停止するプログラムが課題の内容である。図 3.7 に示すように、テストの条件に「タッチスイッチ 3」を置き、「いいえ」の右にロボット制御のタイルを置く。前進する時間に 0.1 秒間の指定があるため、「はい」が空欄ならばロボットは停止する。

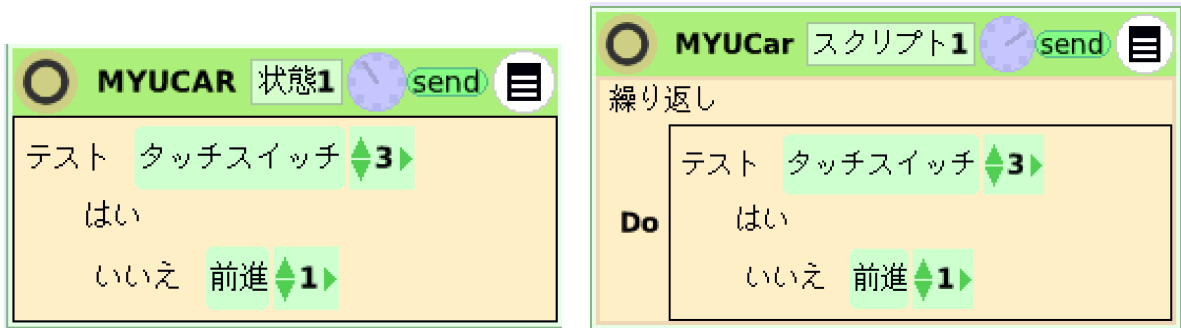


図 3.7 接触の有無で前進または停止するプログラム

### ステップ 3: 左回転の間に接触があれば右回転し始める

ステップ 3 からは、状態遷移を扱う。図 3.8 のように、2 つの状態を、センサ入力により順に遷移する単純な状態遷移である。課題は、左回転する状態から、左センサに接触があると、右回転する状態へ遷移する、一方向の遷移とした。

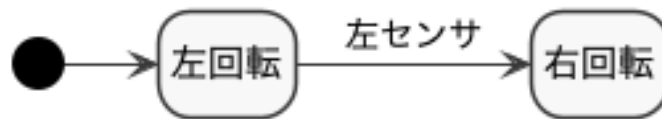


図 3.8 一方向の状態遷移

状態 Rtoys では、「左回転」と「右回転」の 2 つの状態を作成し、左センサを意味する「タッチスイッチ 2」によって状態遷移するプログラムが直接作成できる。手続き Rtoys では、状態を表すひとつの反復から次の状態（反復）への遷移に、「繰り返し脱出」を用いることで、状態遷移に対応付けている。これを図 3.9 に示す。一方向の遷移を手続き Rtoys で作成する場合、ループ内部の条件文のなかに次のループを入れ子にしても成り立つ。これを図 3.10 に示す。

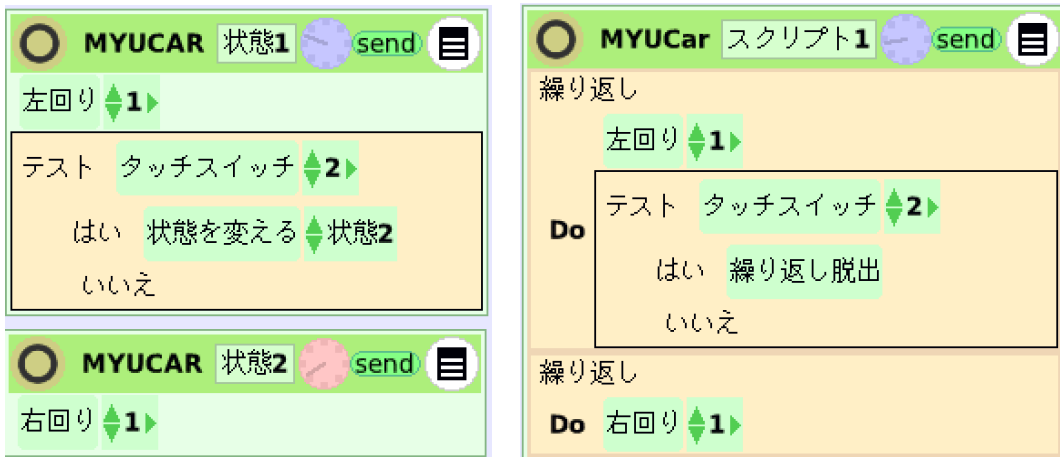


図 3.9 左回転の間に接触があれば右回転し始めるプログラム



図 3.10 反復の入れ子により同じ動作をするプログラム

#### ステップ 4: 接触した方向に回転方向を変える

ステップ 4 では、2 状態を往復する状態遷移を扱う。そこで、課題は、左回転していて、右センサに接触があれば右回転し始め、そこで左センサの接触があれば左回転し始めるプログラムとした。この状態遷移を図 3.11 に示す。

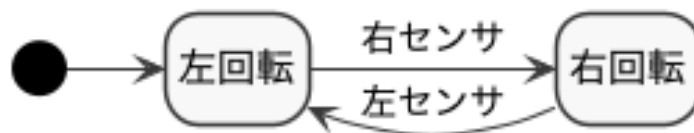


図 3.11 センサが接触した方向に回転方向を変える状態遷移

すなわち、ステップ 3 の図 3.8 に、右側の状態から左側の状態への遷移を追加した形となる。状態 Rtoys では「状態を変える」をひとつ追加すればよいが、手続き Rtoys では、センサの接触で「繰り返し脱出」する条件文を追加するだけでは、全体の反復がなくなってしまう。そこで、外側に反復を

追加しなければならない。結果として、入れ子が一段深くなる。これらのプログラムを、図 3.12 に示す。ステップ 4 の状態 Rtoys で、反復を内側に入れた場合には、条件文がプログラムの末尾であるため、「繰り返し脱出」を入れることで、外側の反復に戻り、意図した動作にすることができる（図 3.13）。

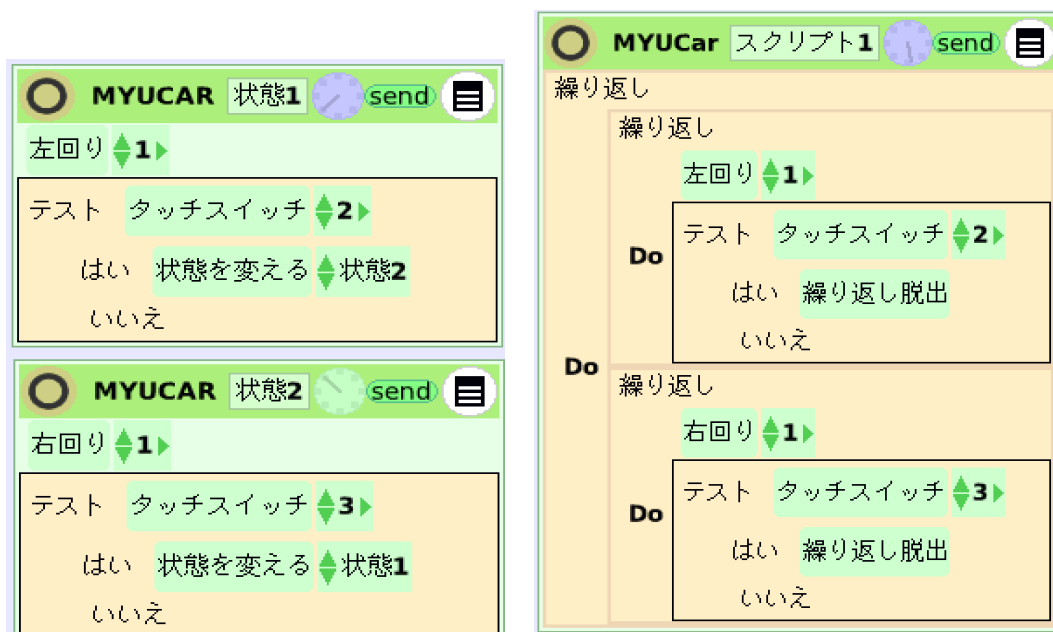


図 3.12 接触のあった方向に回転を変えるプログラム



図 3.13 内側に反復を入れ子にして同じ動作をするプログラム

## ステップ 5: 障害物回避しながら走行

最後のステップ 5 では、往復する 2 状態を 2 つ重ねた 3 状態を扱う。課題は、障害物回避をしながら前進するプログラムとした。障害物回避は、センサの接触により、いったん後退して反対方向に回転することを繰り返すことになる。この動作を状態遷移図にしたものを、図 3.14 に示す。

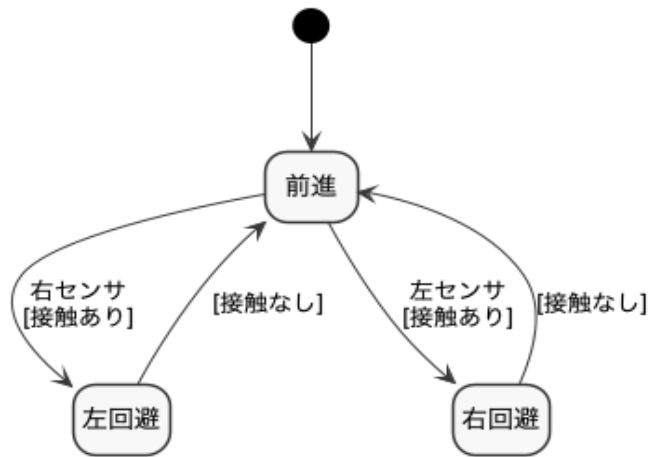


図 3.14 障害物回避走行の状態遷移図

状態 Rtoys では、状態遷移図と同じ形でプログラムを作ることができる。プログラムは図 3.3 に示した。手続き Rtoys では、「繰り返し脱出」を使うことで、「前進」のループの内側にある「左回避」相当のループや「右回避」相当のループから脱出するプログラムとなる。しかし、この課題は、全体をひとつの状態と考えると、3 分岐により処理を判定する手続き的なプログラムが最もシンプルな解となる。これを、図 3.15 に示す。

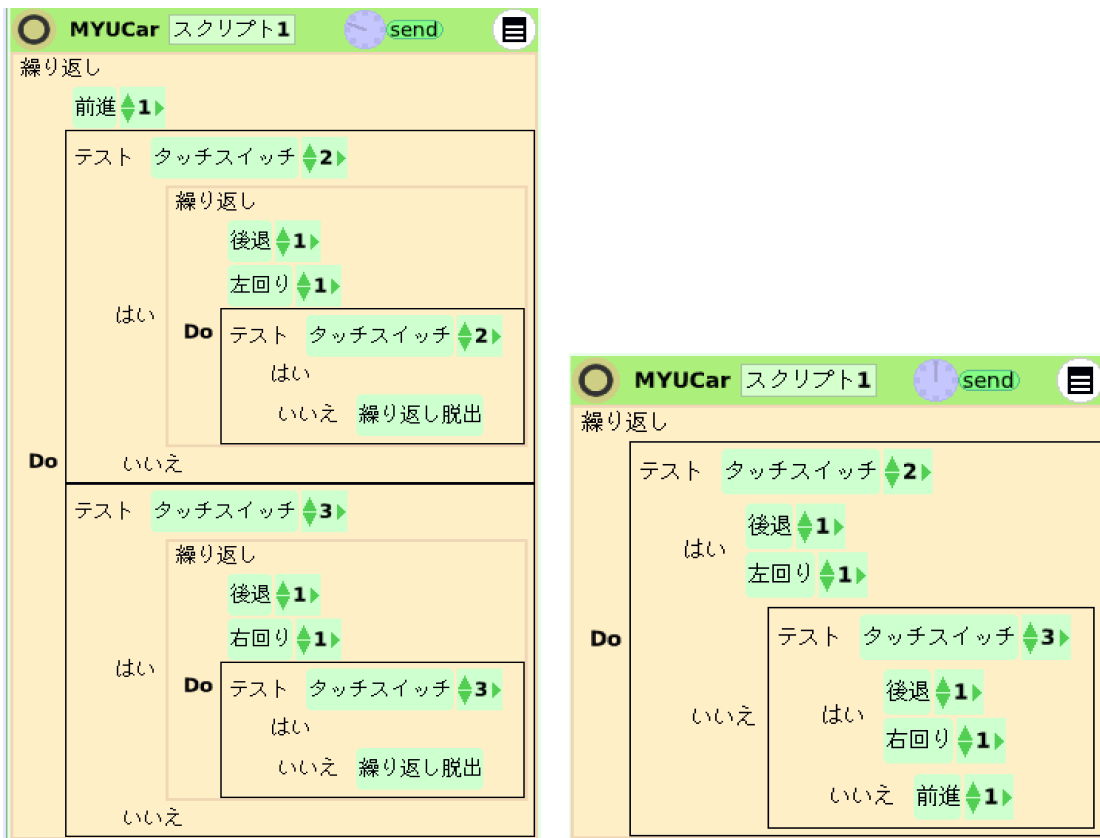


図 3.15 障害物回避の手続き Rtoys による 2 つのプログラム

### 3.3.4 検証授業の結果

5ステップの各課題について、自らの力のみでできた生徒から順に動作をたしかめ、意図通りの動きであれば課題を終えたとし、保存させた生徒のプログラムで正誤を確認した。両方を満たしている生徒の割合を、表 3.2 に示す。 $p$ 値は、片側の Fisher の正確確率検定による。

表 3.2 課題を正解した生徒の割合（両群ともに $n = 15$ ）

課題	状態	状態 Rtoys	手続き Rtoys	$p$
1	なし	86.7%	46.7%	0.008
2	なし	80.0%	73.3%	0.150
3	一方向 2 状態	6.7%	33.3%	0.200
4	往復 2 状態	60.0%	20.0%	0.002
5	往復 3 状態	46.7%	0.0%	0.003

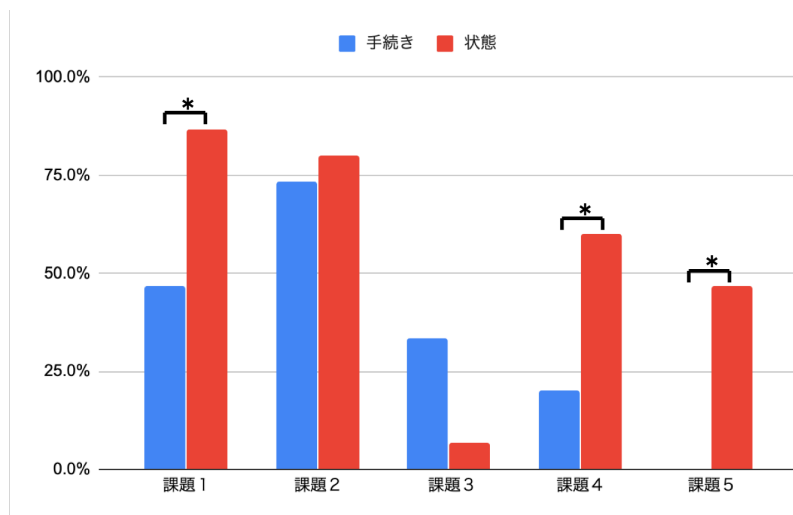


図 3.16 課題正解者の割合（\*は $p < .05$ ）

図 3.16 は、このグラフである。群間で、 $p < .05$  で有意差が認められるものを明示した。ここから、以下のことを読み取ることができる。

1. 状態のない課題 1 及び課題 2 は、以後と比較してどちらの群も高い正答率である
2. 最初に状態を扱った課題 3 では、正答者が両群で急減した
3. 状態遷移が以前より複雑な課題 4 及び課題 5 では、手続き Rtoys の群は正答がほとんどない
4. 課題 5 は状態 Rtoys 群のみ半数弱の生徒が正答できた

課題 3 とそれ以後の違いは、状態遷移について教師による一斉指導の有無による。この集計は生徒自身の思考で、または自ら資料を読んで、課題を解決した生徒のみを対象としている。しかし、課

題 3 を十分な時間をとっても解決できないことから、残りの時間で状態遷移について資料に基づき一斉指導した。これを受けた後に、生徒は課題 4 と課題 5 に取り組んだ。

### 3.4 評価

課題正解者の集計（表 3.2 及び図 3.16）から、複数の状態を持つ、異なる動作の反復処理を切り替える必要のある課題では状態 Rtoys の正解者数が有意に高かった。このことは、状態 Rtoys は、生徒の状態遷移の理解と、具体的問題の全体的な動作を分解して考え、遷移によって結合する思考とのギャップが少なかったことが推測される。課題 3 の後に行った一斉指導は、課題 3 を理解するための補足であり、以後の課題である往復する遷移について特に情報は与えなかった。

状態 Rtoys は、状態の反復動作は、状態遷移モデルに則り暗黙的に行われるため、状態内の動作のみを考えることができる。手続き Rtoys は反復のなかに条件を入れ子にする必要があるため、課題 3 の成功体験により、遷移先の状態を、入れ子の内側に記述する解答が目立った。用意した「繰り返し脱出」によって次の反復へ処理を移すという考え方が、生徒の直感と異なっていたことが考えられる。

以上のことから、手続きのみで状態遷移を考えてプログラムさせることは、手続き Rtoys の場合のような一定の規則を与えても、中学生の直感にそれを訴えることは難しく、規則を理解して状態遷移のプログラミングに使うことができる生徒は限られることが明らかになったと言える。すなわち、状態遷移の説明を受けたとしても、手続きで中学生がプログラムすることは難しい。

一方、状態 Rtoys で導入した「状態」のウィンドウ表現と、状態の遷移を遷移先の状態名で指定する記述は、状態遷移図に対応付けられる表現である。このことと、プログラムして実行できる言語であることで、本章での提案が、中学生が複数の状態遷移を持つプログラムを考えるための教材になり得たものとする。



# 第4章 計測値の可視化による計測制御学習 支援教材の開発と学習効果

前章では、適切な学習環境によって生徒が複数の状態を持つ状態遷移を理解してプログラミングを行うことが可能であることを実証した。その上で、本章では、計測制御システムの基本的動作である「センサからの計測値を、コンピュータが判断して、制御を行う」を理解させる可視化教材を提案する。教材では、計測値と判断、制御の内容のほかにも、状態遷移についても実時間で可視化する。この教材を用いてシステムの動作を状態遷移の理解を伴って理解させる授業を設計し、中学校での授業を通し、教材の学習効果を検証する。

## 4.1 中学生に計測制御システムを理解させるためのアプローチ

### 4.1.1 研究仮説

前章では、ビジュアルプログラミング環境 Rtoys によって、生徒は状態遷移を考えプログラミングが可能であることを明らかにした。一方、中学校段階は小学校で今後行われるビジュアルプログラミングからテキストプログラミングの転換期にあたる。したがって、テキスト言語であっても状態遷移を直接プログラムできる言語により、中学生が複数の状態遷移を持つプログラムを考えられるかを確かめる必要がある。また、これからの中学校に求められる計測制御システムの学習では、計測制御システムの基本的仕組みを理解して、設計と具体化が必要である。このとき計測値及び制御のための操作量が連続値であることが必要になる。

そこで本章における研究仮説を、「状態遷移の考え方を使いながらロボットのプログラミングを行い学習を進める過程で、ロボットが計測している連続的な値、計測値に基づく判断の状況、制御と状態遷移を、それぞれを実時間で可視化表示すると、生徒はそれを見て、ロボットの置かれている状況や実際の動きと対比することで、計測制御システムの仕組みを踏まえた、複数の状態遷移を持つプログラムの動作を説明できるようになる」と設定した。

### 4.1.2 研究の方法

計測値の連続した変化や状態遷移を実時間で可視化表示できる教材を新たに開発し、検証授業に用いることにした。

最初に教材の設計方針を検討する。ロボットの機構は、前章からの連続性を保つため、同じミュウ

ロボ[42]を用いることとした。ただし、実時間で計測値と状態遷移及び制御命令の実行状況を表示させるためには、ミュウロボ制御基板に生徒が作成したプログラムを書き込む方法では実現できない。そこで、Bluetoothによる無線通信によりロボットとPCを常時接続することを考えた。常時無線接続により、ロボットが得た計測値をPCに連続して送信するとともに、制御命令をPCから受信して即時に実行することができる。したがって、制御基板自身は、いわばリモコン動作だけを行う簡易なソフトウェアで十分である。ミュウロボ制御基板が持つ自律動作のためのプログラムは不要であるため、同じサイズの新たな基板を設計し、搭載することとした。プログラミング言語は、中学校の授業で広く用いられるテキスト言語として、ドリトル[44][45]に状態オブジェクトを定義したものを利用することとした。

図 4.1 に、①制御基板、②測距センサ、③電池ボックス、をミュウロボに実装した状態と、④PC用無線基板を示す。

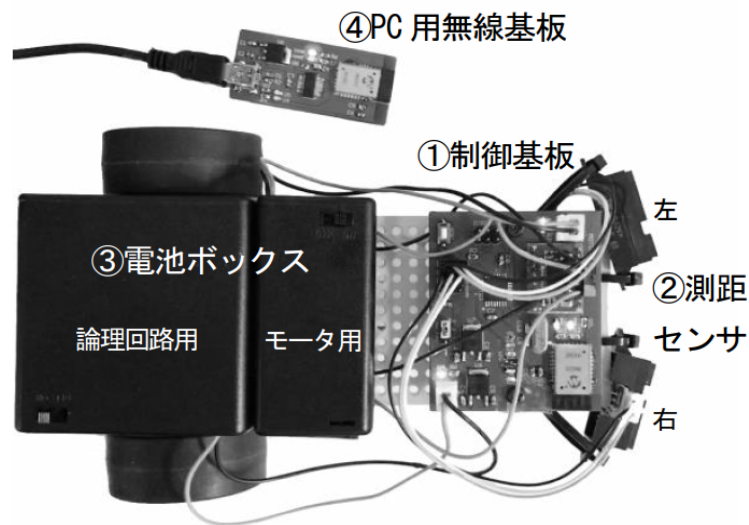


図 4.1 作成した基板を搭載したロボットと PC 用無線基板

可視化の方針と開発した可視化画面については、次節で述べる。プログラミング言語をドリトルの拡張としたことで、イベント発生 の条件式を記述するために、不等号の入った式を手で入力する必要が生じるようになった。中学校 1 年の数学で不等式を学習するが、プログラムで用いる条件式のなかの不等号とは性質が異なる。数学の不等式は静的な関係で中学生は理解するが、プログラムでは動的に比較を行い、式の値は都度変化する。そこで、授業ではまず「計測値をつねに測定していて、いつも変化している」ことに注目させた上で、その後、教師が実機を用いて「計測値である距離の値と、条件式のしきい値の大小を、センサにかざす手を徐々に近づけながら、繰り返し『どちらが大きいか』を問う」ことで、不等号の意味とプログラムの実際の動作とのつながりを確実にするよう指導ができるような場面を設定する。検証授業では、状態の必要のない課題から、最終的に、2 センサ 3 状態の遷移の課題までを解決させる構成とする。学習効果を検証するため、事前・事後調査では、計測制御

システムの理解度の変容と、状態遷移の理解に基づき、図で示した模型車の周囲の状況とセンサ計測値から、次の挙動が推測できるようになるかを問う用紙法による調査を行うこととした。

## 4.2 開発した教材における可視化の詳細

本研究における可視化のポイントは、以下の3つである。

1. 連続的な計測値をコンピュータがつねに得ていて、値はつねに変化している
2. 条件判断により複数の状態の間に遷移が生じている
3. プログラムが現在実行している制御命令は何であるか

これらを学習者が見てわかるようにすることで、次のような計測制御システムの学習を可能とする。まず、計測制御システムのなかで、計測と制御はつねに行われていて、プログラムにより目的の動作を計測値に基づいた制御によって実現していることに生徒の意識を向けさせる。制御に伴う計測値の変化から、「障害物にぶつかりそうだ」といった予測をさせる。予測に対処するための制御プログラムを記述し、「たしかにそれが動いている」または「意図と違った命令が動いている」ことを確認しながら、プログラムに必要な改良を施す。

上記の学習を念頭に、2つの可視化画面を設計した。ひとつは、計測値の可視化を行い、人によるロボットの操作や簡単なシーケンス制御の間の計測値変化を観察させるものである。もうひとつは、状態遷移を含む計測制御システムのプログラムを入力して実行する場面で利用する、システムの動的な振る舞いを計測値から制御命令までリアルタイムに表示するものである。状態遷移を可視化することに特徴がある。前者を以後「計測値可視化画面」、後者を「状態遷移可視化画面」と呼ぶ。

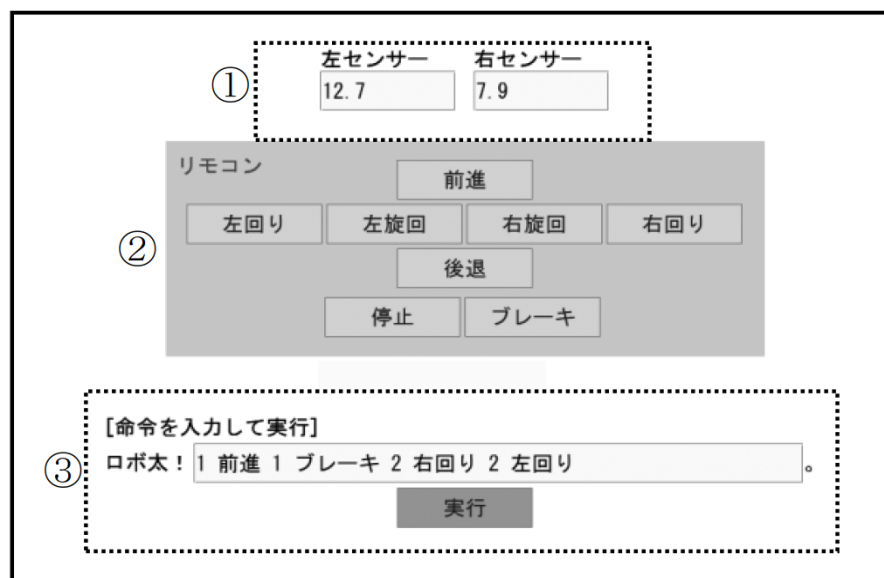


図 4.2 計測値可視化画面

図 4.2 は、設計した計測値可視化画面である。図に加えた各部の内容は、番号ごとに、①センサの値変化の観察、②ロボット制御命令を体験的に学ぶためのリモコンボタンの操作、③学んだ制御命令を命令文として入力してシーケンス制御を実行、に対応する。内部の並行処理により、②のボタン操作と③の命令文実行中にも、①のセンサ値は更新され続ける。ボタン操作は、リモコンのイメージ通り、学習者がマウス操作する。例えば「前進」ボタンを押した後、ロボットが前進を続け、「停止」ボタンを押すことで停止するように、ボタンに対応するロボットの挙動が連続して実行される。ボタンの名称は、ロボットの制御命令に対応しており、③にそのまま命令として入力し実行できる。この画面は、例えば、表示されている計測値の目標値を決め、それに合わせるような学習に利用することを想定している。

図の②に示した 8 つの制御命令のうち、「停止」と「ブレーキ」の違いについて補足する。「停止」はモータを回路から開放するため、慣性によりロボットがすぐに停止しない場合がある。一方、「ブレーキ」はモータを短絡させるため、モータの逆起電力により、強い力で制動がかかる。「左回り」「右回り」と、「左旋回」「右旋回」の違いについて補足する。「左回り」「右回り」は、片方のモータのみを前進させるため、自動車のハンドル操作に似た、前進しながら回転する軌跡を描く。「左旋回」「右旋回」は左右のモータを逆回転させるため、その場所で回転する動作を行う。

よって、③のシーケンス制御命令文「ロボ太！ 1 前進 1 ブレーキ 2 右回り 2 左回り。」を実行した場合、ロボットはまず 1 秒間前進し、1 秒間急ブレーキをかけた後、2 秒間右回り(やや前進しつつ右方向に向きを変更)し、2 秒間左回りし、最後に停止する。

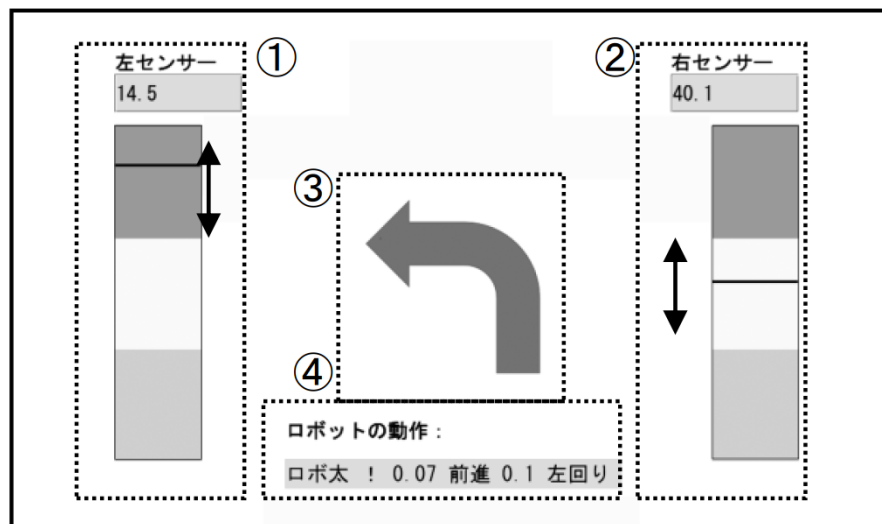


図 4.3 状態遷移プログラム実行時の可視化

図 4.3 に、状態遷移可視化画面を示す。ここでは、センサの計測値変化と、生徒がプログラム実行中、現在、どの状態に定義された制御命令が実行されているのかが直感的に視認できるよう、可視化を施している。①が「左センサ」の値変化、②が「右センサ」の値変化である。数値とともに、帯

の内部で、横棒が上下する。横棒は、ロボットが壁など障害物に近いほど衝突の危険があることを表現するため、センサが計測可能な最小値である 7cm の場合上端へ、最大値である 80cm の場合、下端へ至ることとした。③は「左回り」命令に対応するアイコンであり、8 個の制御命令に対応する、異なる図案のアイコンを用意した。アイコンは、制御命令の実行ごとに描き直される。④は、現在の状態に設定されている動作の表示であり、状態が遷移するごとに描き換えられる。

図 4.3 では、④から「現在の状態では、前進と左まわりを順に実行する動作を行う」ことが示されており、③のアイコンから「その後半にあたる左回り命令を実行中である」ことが示されている。同じ状態にとどまる場合は動作を反復するため、④の表示は変わらない。表示の変化は状態遷移が生じた際に起こるため、①②の計測値、③の制御命令と合わせ、どの状態のなかで、どのような制御を行っていて、計測値がいくつであったために遷移したのかを総合的に判断できることを想定した。

本研究で「状態遷移をプログラム」させるために、ドリトルを拡張して実現したテキスト言語による記述の設計について述べる。

状態遷移の考え方をういたプログラムを「状態を示すオブジェクト」と「遷移させる命令」の組で表現することの学習効果が、前章のプログラミング言語 Rtoys を用いた検証授業によって確かめられた。そこで本教材においても、ドリトルに状態オブジェクトを定義し、状態内で実行される動作を、ドリトルの構文と同等の、状態オブジェクトに対する動作設定命令とした。

「状態」や「遷移」の語は Rtoys では「状態を変える」だったが、中学生の語彙を考え直し、状態を「モード」、遷移を「移る」に置き換えることにした。語選択が適切であったかは、検証授業を通して考察する。

その結果、ドリトルを拡張した状態遷移の構文は、以下通りとなった。構文要素はゴシック体で、識別子は下線で表記した。網掛け部分には、任意の文を記述できる。

(状態の定義)

モード名 = **モード!** 作る「」動作。

(遷移の定義)

モード名! 「遷移先モード名」**移る**。

(プログラムの起動)

**はじめ!** 「起動時のモード名」**移る**。

図 4.4 に、この構文を用い、前進していたロボットが、10cm より近くの距離に障害物がある場合

に停止し、障害物が取り除かれた場合に再び前進を始めるプログラムの例を示す(なお、距離が 10cm と等しい場合には状態は遷移しない)。プログラムの末尾で、プログラムを起動させるために、最初の状態である「はじめ」からの状態遷移を記述している。

```

前へ=モード！作る「ロボ太！前進」動作。
前へ！「左センサ<10」なら「とまる」移る。

とまる=モード！作る「ロボ太！停止」動作。
とまる！「左センサ>10」なら「前へ」移る。

はじめ！「前へ」移る。
    
```

図 4.4 拡張したドリトルでの状態遷移プログラム例

この例のように、制御命令「前進」「停止」に動作時間を指定していない場合は、基板上のモータドライバの設定を変更した後、すぐに次の処理が実行される。このプログラムでは、この性質を利用してタイムラグの少ない状態遷移を実現している。

## 4.3 中学生を対象とした検証授業と結果

### 4.3.1 単元の構成と授業の内容

愛知県内の公立中学校 3 年生を対象に検証授業を行った。この学校は全国学力試験結果が全国平均点位であり、標準的学力の公立中学校と考えられる。

表 4.1 は、検証授業の構成である。対象の生徒は 2 年生のときに、ドリトルを用いたタートルグラフィックスとミュウロボを用いた手続き型の制御学習を経験していることから、プログラミングの基本やドリトルの構文に慣れる学習について、授業計画には含めていない。各時限の授業時間は、中学校では短縮授業 45 分となる日が多いため、全て 45 分として計画した。

表 4.1 検証授業の構成

時限	内容
1	センサ計測値変化の観察・理解
2	ロボットのリモコン操作とシーケンス制御の比較
3	計測と制御の並行処理・状態遷移導入
4	自動運転車モデル① 左の壁のみ 1 センサ 2 状態
5	自動運転車モデル② 左の壁と前方 2 センサ 3 状態

第 1 時限では、センサにより計測される値が周囲の状態により時間変化することを体験し、意識化させる。第 2 時限では、人の操作とロボットのプログラムを関連付けさせ、時間によるシーケンス制御をさせるなかで、人手による調整には限界があり、センサ利用の必要性を意識させる。第 3 時限では、計測制御システムの身近な例を取り上げ、センサの値を読み取り続ける動作が内部で行われていること、しきい値との比較も繰り返し行われていて、条件に合った制御が選ばれている、という、計測制御システムの動作を簡単なロボット制御により理解させる。第 4 時限では、コースを走る自動運転車をモデルに、ひとつのセンサだけでコースを守って走らせる。第 5 時限ではもうひとつのセンサを追加し、正面の障害物の手前でも止まる課題を扱うことで、より複雑な計測制御の処理を考えさせる。

### 4.3.2 検証授業の詳細

#### 第 1 時限：センサ計測値変化の観察・理解

生徒が初めて触れるロボットに対する興味を活用し、計測と制御の学習へ意識を向けるさせるため、センサそのものの性質を理解させ、計測・制御を行う授業全体の導入とすることをはかった。

そのため、ロボットに搭載される 2 つの距離センサとフォトトランジスタ（光センサ）をアナログポートに接続しておき、①値を数値のみで表示するプログラム②値を折れ線グラフでリアルタイムに表示するプログラムの 2 つを用いる。生徒には、①のプログラムが表示する 3 つの計測値を観察しつつロボットに触れさせることで、計測の対象と値の範囲を考えさせ、②のプログラムの時系列グラフ表示と合わせ、センサ計測値が周囲の状況により時間変化することを意識化させる方法をとった。

その結果、生徒は数値が素早く変化する様子に強い関心を示し、距離センサが cm 単位で距離を計測していることを素早く言い当てた。グラフ表示に対する大きな反応はなかったが、ロボットを手を持って動き回すと周囲の状況が大きく変わり、距離と明るさセンサ双方の値が変わることを楽しみ、目標の数値となるロボットの状態を自ら探す様子がみられた。授業のまとめとして、センサの計測値は時々刻々と変化すること、変化は周囲の状況変化によって生じることを再度確認した。

#### 第 2 時限：リモコン操作とシーケンス制御

ロボットの操作と時間によるシーケンス制御双方を体感させることで、自動化の手順や意味を考えさせた。また、時間を指定したシーケンス制御の限界を体感させることで、次回以後のセンサ利用への導入をはかった。

そのため、図 4.2 に示した、「計測値可視化画面」を用いた。人によるリモコン操作と、制御命令入力欄へ命令を書き込み実行させた後、両者の違いについて生徒に発言させるとともに、考えをワークシートに記入させた。また、位置により徐々に曲率が変わり、制御する角度もそれに合せ調整が必要な楕円コースを正確に周回する課題に取り組ませた。

その結果、生徒からリモコン操作では「ボタン操作と車の動きにわずかな遅れがあるので操作が難しい」「画面と車の位置が離れると一方が見えないからうまく動かせない」等の発言があった。命令を入力して制御する場面では、楕円コースに取り組むとき、生徒の多数が「命令では少しずつ命令を追加することで完成の動きに近づけられるので、こちらのほうが楽だった」という意味の回答をした。自動化により省力化できる面と難しさの両方が把握されており、授業の意図は十分に達成されたと考えられる。

### 第 3 時限：計測と制御の並行処理・状態遷移

計測制御システムがつねに計測を行っているイメージと、条件を満たしたときにイベントが発生するイメージをつかみ、状態遷移としてシステムを捉える発想をもたせるため、生徒にとって身近な、自動ドアから導入することをはかった。

そのため、図 4.5 のワークシートを用い、制御システムの多くに、条件を満たしたときに動作が切り替わる考え方が使われていることを気づかせた。具体的には、生徒に対し、「ドアが開く」動作は、その前に「ドアが閉じている」必要があるということに気づかせた。そして「閉じている」状態から「開いている」状態へ切り替わり、再び「閉じている」状態へ切り替わることを確認した。

「閉じている」状態が図の上と下の両方にあることを示し、これらは同じであるから先頭に戻ることと同じだね、と、板書を用い、矢印を加筆することで、繰り返しの動作であることを認識させた。この後、「閉じている」状態から「開いている」状態への切り替えに注目させ、「人が近づいた」時にこのイベントが生じることに気づかせた。再び「閉じている」状態へ戻る条件は、生徒の言葉を尊重し、「時間がたった」としてまとめた。

その後、ロボットの正面で前後に動かす手との距離を一定間隔に保つロボットの動きを考えさせ、自らプログラムし、動作させた。特に、プログラムを考えさせる場面で、生徒に距離センサの値を注視させながら手を徐々に近づけ「10cm より近いか?」と繰り返し問う演示を行った。この意図は、条件を逐次比較するイメージを可視化し、しきい値に基づき変化するセンサの値に対して制御を切り替える考え方を印象づけるためである。

その結果、時間内に全生徒が自ら課題のプログラムを完成させ、正しい実行結果を得ることができたことで、授業の意図を達成されたと考えられる。



1 自動ドアの仕組みを考えて、□と（ ）に言葉をいれましょう。

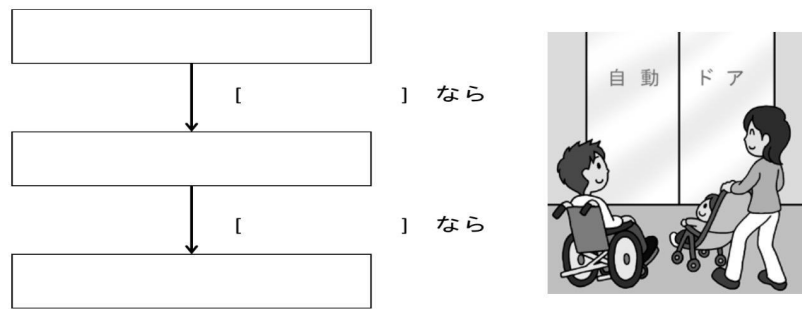


図 4.5 「計測と制御」導入用ワークシート（部分）

#### 第 4 時限：自動運転車モデル① 1 センサ 2 状態

自動運転をイメージし、左センサのみを使い、左側に壁を立てたコースとの距離を一定の範囲に保ちながら前進を行う課題に取り組ませることで、2 状態遷移の認識の定着をはかった。

導入として「左の壁に近づきながら前に進む」動作と、「壁から避ける」動作からなることを、教師が手のおおまかな動きにより示唆した。実際のロボットの動きは各生徒独自の考えにより試行錯誤的に探求させた。特に、「壁に近づく」「壁を避ける」のロボット視点となる語を用い、状況に応じ得られる計測値を正確に解釈しプログラムに活かすことを意識した。

その結果、ほぼ全員が紙上のプログラムを論理的、構文的に正確な形として完成させ、実機で時間パラメタ調整に苦労した一部の生徒を除き、課題を達成した。

よって、前時と合わせ、生徒は、2 状態の遷移に関し、具体的な理解が得られたと考えられる。

#### 第 5 時限：自動運転車モデル② 2 センサ 3 状態

前時の課題を拡張した「前方への衝突回避」追加した課題とした。2 つのセンサを利用する 3 状態の状態遷移となる。特にある状態から 2 分岐する状態遷移を、生徒自ら気づきプログラムとして表す能力の確認をはかった。

2 つのセンサとして、左センサを左の壁との距離測定に、右センサを前方の障害物との距離測定に用いることのみを伝えた。

その結果、全員が図 4.6 のような状態遷移の形を、図ではないが、生徒自身の言葉を用いノートに表現していた。授業中、生徒に問題の難易度を問うたところ、「この問題は難しいと思わない」と多くの生徒が回答した。授業実践者によれば、実践校のなかで平均前後またはそれ以上の学力と判断される生徒全員がそう答えたとのことであった。したがって、1 センサ 2 状態の理解を得た中学生にと

って、2つのセンサの役割を別々に考え、結合する順序で考えさせるならば、2分岐が含まれる2センサ3状態への拡張は、無理のない設定であったと考えられる。

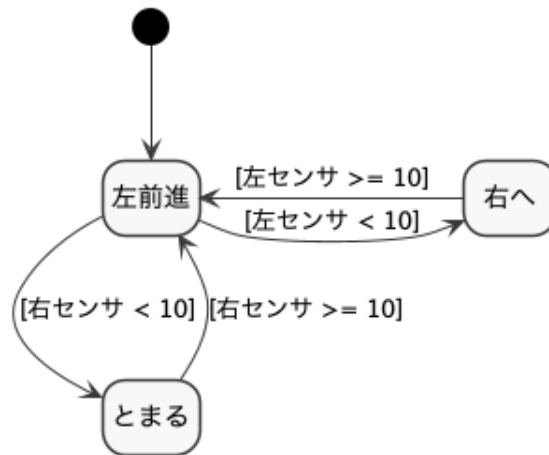


図 4.6 課題の状態遷移図

### 4.3.3 事前調査と事後調査

事前調査と事後調査は、表 4.1 に示した授業計画の前と後に、それぞれ実施した。

事前調査は、学習前の生徒が、身近にあるセンサを用いてコンピュータにより自動制御されるシステムに対する認識状況を知るために実施した。具体的には、自動運転車を題材に、「コンピュータ制御で、衝突しないよう走る乗り物」が、「途中でやや右に曲がる細い裏通りを走る」という設定を説明した。このとき、通りの左右には様々な大小さまざまな障害物があることを補足説明した。調査用紙には、自由利用可能として公開されている写真を添えた質問・回答用紙を作成した。図 4.7 に、事前調査の質問内容、図 4.8 に事前調査の質問 2 で使用した写真を示す。

1. 次の文が、自動運転する乗り物の仕組みにあてはまるかどうかを考えて、○か×の印を付けてください。それを選んだ理由も書いてください。
  - ・センサが乗り物に、走る速さや向きを変える命令を出す。
  - ・コンピュータが乗り物に、走る速さや向きを変える命令を出す。
2. 次の文が、自動運転する乗り物の動作として正しいかどうかを考えて、○か×の印を付けてください。それを選んだ理由も書いてください。
  - ・前半はまっすぐ前に走り、後半は少し右に曲がってから、まっすぐ前に進む。
  - ・乗り物の左右前方をみる距離センサの値を調べながら、走る速さや向きを調整して走る。

図 4.7 事前調査の質問内容



図 4.8 事前調査の質問 2 で使用した写真

質問 1 では、計測制御システムの仕組みに対する認識、計測から制御へ至る機序概念の把握状況に焦点を当てた。センサ及びコンピュータの語は認知されている前提に立ち、制御をするのは何かを問う形式をとり、第 1 文にセンサを、第 2 文にコンピュータを置き、第 1 文と対置させた。

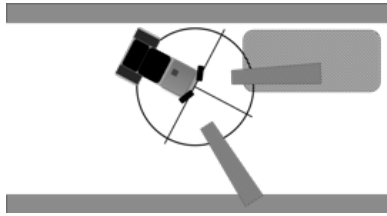
質問 2 では、計測しながら制御する認識、すなわち計測と判断は反復して行われており、それを応じた制御が行われているというシステムの概念の形成状況に焦点を当てた。同様に対立する 2 文を配置し、第 2 文に計測・判断と処理の関係を意図した「ながら」の語を用いた。また、調査時に「乗り物はコンピュータのプログラムで運転されています」と口頭で補足することにより、生徒が乗り物内部の処理を考え、回答するよう促した。

事後調査は、第 5 時限の授業で扱った内容を踏まえ、検証授業単元の到達目標とした「センサ計測値の時間的変化をコンピュータが読み取り判断をする」こと、「プログラムにより設定された条件に応じ、予め用意された制御動作を実行すること」の認知状況を知るために設定した。図 4.9 に、質問内容を示す。

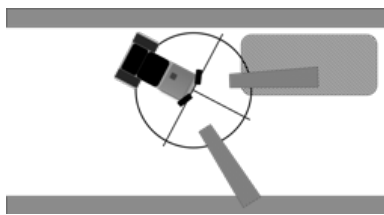
質問文に含まれる図の要素と意味を説明する。上下の太いラインは、第 5 時限の授業で用いたコースの、「左の壁」「右の壁」を表す。図の右方向がゴールであるため、上が「左の壁」、下が「右の壁」に対応する。ロボットには先頭左角に「左センサ」、右角に「右センサ」が取り付けられており、距離を計測する。ロボットに重ね書きされている十字と円は、十字がロボットの向きを表し、円の直径は 10cm を表す。左右の距離センサの先に伸びるグレーの帯は、センサが距離を計測可能なおよその範囲を示している。作図の都合上、最短の 7cm は表現しているが、最長の 80cm より短く、20cm 程度までを描いた。ロボットの動作は、図 4.4 の状態遷移図である。左センサが「左の壁」との距離

を 10cm 程度に保ちつつ左右に首を振りながら前進する動作を担当する。右センサが距離 10cm より小さな値の場合は停止し、大きな場合は再び動き出す動作を担当する。

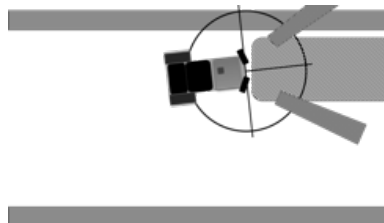
下の図は、ロボットが左の壁を避けるため、右に 45 度、回転したところです。



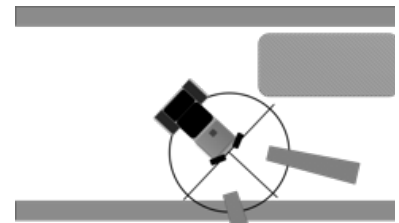
ロボットはこの先、どのように動きますか。予想される動きを、図(A)~(C)から選び、その理由を説明してください。



(A)そのまま停止



(B)左前に進み、障害物に衝突する



(C)右に回転したあと左前に進み、  
右の壁で停止

図 4.9 事後調査の質問内容

質問文に含まれる最初の図は、左センサが 10cm の円内に入っているため、計測値は 10cm より小さい。右センサは約 20cm である。この図では、左センサは遷移条件を満たし、右センサは遷移条件を満たさない。質問文は、現在ロボットは、「右へ」の状態にあることを述べ、ロボットがこの後どのような動作により図(A)、(B)、(C)いずれの位置に移動するかを回答させている。

図(A)は、本来「とまる」状態へ遷移、すなわち停止するのは右センサの距離であるにも関わらず、左センサでも停止できると誤解した場合を想定している。図(B)は、左センサの値が 10cm より近いにも関わらず、「左へ」へ遷移すると誤解した場合を想定している。図(C)は「右へ」状態を保った後、「左へ」へ遷移して前進すると正しく考えた場合である。

#### 4.3.4 結果

授業を受け調査対象となる生徒数は 225 名であった。結果を集計するにあたり、全 6 クラスのうち、授業進行が同等と判断できる 3 クラスを選び、欠席や回答未記入等、欠損データとなるものを除いた結果、 $N = 105$ となった。調査対象に対する意図的な選択操作は行っていない。

#### 事前・事後調査の結果

事前・事後ともに、回答を点数化して集計した。質問ごとに、正しい解答を選択した場合に 1 点を与え、選択の正誤にかかわらず、理由が明確な場合に 1 点を加点した。したがって、0 点から 3 点の 4 段階となる。

事前と事後回答における点数ごとの人数を、表 5.2 に示す。①質問 1 と事後、②質問 2 と事後、それぞれについてカイ二乗検定を行ったところ、①では $\chi^2(3) = 71.749, p < .001$ 、②では $\chi^2(3) = 20.065, p < .001$ より、有意差のあることがわかった。そこで、事前と事後における各点ごとの度数増減をみるために、残差分析を行ったところ(表 5.3)、調整済み残差は、質問 1 では、0 点では 5%水準、他の採点では 1%水準で、事後において期待値より有意に大きな値となり、誤答 (0 点または 1 点) から正答 (2 点または 3 点) への変化が示された。質問 2 では、0 点に有意差はなかった( $p = 0.261$ )が、1 点と 2 点では、5%水準で事後において期待値に比して有意に小さく、3 点もまた、1%水準で期待値に比して有意に大きな値となり、3 点が有意に増加したことが示された。

生徒個別の変化を集計したところ、事後が 3 点の生徒のうち、事前の 2 つの質問が共に正答(3 点または 2 点)であった者は 13 名(12.4%)であり、92 名(87.6%)が事前の質問いずれかを誤答していた。逆に、事前の 2 つの質問が共に誤答(1 点または 0 点)であった生徒のうち、事後も誤答した者は 7 名(6.7%)であった。

また、質問 1 の回答が 1 点であった 54 名(51.4%)について、記述された理由を確認したところ、「センサが乗り物に制御の指示を出す」(26 名)を肯定するものと、「人が操作すればよい」(28 名)の 2 つであった。

表 4.2 検証授業事前・事後の採点とその人数

採点	事前質問 1 (N=105)	事前質問 2 (N=105)	事後 (N=105)
3	17	26	57
2	8	42	27
1	55	17	7
0	26	20	14

表 4.3 事前・事後の調整残差

(a) 事前質問 1 と事後			(b) 事前質問 2 と事後		
採点	事前質問 1	事後	採点	事前質問 2	事後
3**	-5.78	5.78	3**	-4.38	4.38
2**	-3.52	3.52	2*	2.20	-2.20
1**	7.14	-7.14	1*	2.17	-2.17
0*	2.11	-2.11	0	1.12	-1.12

\*  $p < .05$ , \*\*  $p < .01$

### 学習前の生徒の思考分析

事前調査の理由記述から、自動運転システムに関する生徒の思考を分析した。大学教員と大学生の4名が分類し、結果を相互に確認した。分析の目的は、学習前の生徒が自動運転について、何を理解していないか、何を誤解しているのか、その傾向を定量的に明らかにするためである。生徒の記述は省略が多く、通常の記事解析が困難であると判断したため、同じ回答すべてを複数が個別に通読し、記述の解釈に対する仮説を突合しつつ総合的判断を導き分類していく方法で、主観的判断を減少させる手段をとった。その結果、4点が主要な要素として挙げられると結論づけた。結果を表 4.4 に示す。

表 4.4 学習前の生徒の自動運転に対する理解

割合	分類	回答例
78.2%	センサの役割	「センサが制御を判断する」 「センサはコンピュータより速く判断できる」
45.5%	コンピュータの役割	「センサがあればコンピュータは不要」
42.7%	自動運転の理解	「思ったとおりに動く」「衝突してから障害物をよける」 「安全そう」「センサの値を人が見て運転する」
20.9%	自動運転への不安	「人が運転すべき」「機械にまかせてはいけない」

質問 1 で「センサが車に命令を出す」に○をつけた生徒の割合が最も高く、78.2%の生徒がセンサの役割について誤解していることがわかった。類似した内容と判断される記述をまとめたところ、「センサが制御の判断をする」意味の文と、「センサはコンピュータより速く判断して制御をする」意味の文が多数を占めた。前者の回答では「コンピュータが車に命令を出す」に×をつけており、セ

ンサのみが制御の指示を出していると考えていたが、後者の回答では両者に○をつけており、緊急時の対応はセンサが担当し、平常時の制御はコンピュータが行っている役割分担を意図した記述であった。

それに続いて割合が高かった（45.5%）のが、自動運転におけるコンピュータの役割を誤って説明している生徒であった。質問1で「コンピュータが車に命令を出す」に×をつけた理由として「センサがあればコンピュータは不要」と理由を書き添えており、センサの役割の誤解との重複が主要な回答であるが、コンピュータの役割から判断が抜け落ちて制御だけを担当する認識を示した回答や、少数であるが、「目で見ればいいから」「自分ががんばる」のように、センサを否定した上でコンピュータに○をつけ、人が判断し操作する記述があった。

自動運転のイメージが漠然としている、または非現実的な考えを述べた回答が42.7%あった。「思った通りに動く」は万能の機械を想定していると考えられ、「衝突してから障害物をよける」という安全とは正反対の記述とともに、非現実的な回答としてここに加えた。多数の「安全そう」も、機械に過度の期待を寄せているか、自動運転のイメージが定まっていないと判断した。「センサの値を人が見て運転する」の記述は、自動運転は人が操作するものであり、人を補助するために、車にセンサの値表示パネルのある状況をイメージしていると判断した。

「人が運転すべき」「機械にまかせてはいけない」のように、選択理由でなく、自動運転に対する不安や忌避感を明確に記述する回答が20.9%あった。

これらの分類には、総合的な判断として因果関係が推定された。センサが自動運転車の制御を支配する考えからコンピュータの存在が認識から抜け落ち、しかしセンサの働きの認識が曖昧なため、自動運転に過度の期待をしたり、曖昧に良い悪いの考えを記述したりすることにつながっていると考えられた。自動車は人が運転すべきとの主張も、自動運転ができることとできないことの判断がつかないことから生じているように思われた。

## 4.4 評価

事前・事後調査すべてにおいて3点と採点された13名(12.4%)の生徒は、事前の段階で、本研究が対象とした、複数の状態を持つ計測制御システムに関する一定の理解を何らかの形で既に得た状態であったことが考えられる。

逆に、質問1の回答が1点であった54名(51.4%)の具体的記述は、「コンピュータはいらない」「センサが命令を出すからいい」「人が操作しないと危ない」等であり、事前の段階では計測制御システムの仕組みについての認識は乏しいことが伺えた。

生徒のこうした認識、すなわち検証授業前の「センサが指示を出して機械をうまく動かしている」という漠然とした解釈から、「センサの値変化をコンピュータが逐次調べながら、条件に合致する制御を選択的に実行する」という複数の状態を持つ計測制御システムの動作に対する正確な理解へと変容したことが確かめられた。その理由は、事後調査における回答理由に記述された、計測から条件判断への動作機序と、その結果生じる新たな状況変化に対する条件判断を述べる回答(3 点の 57 名)から裏付けられる。

検証授業、特に第 4, 5 時限の自動運転車モデルに基づく課題のプログラミングでは、左の壁に近づくように前進するため、時間を引数として前進と左回転を交互に繰り返すが、大きな数値を与えた生徒が、左の壁との距離が画面上では 10cm を下回っていても、設定時間経過まで右回転で壁から離れる状態に遷移しないことに気づき、数値を小さくするよう調整していた。ここで、計測制御システムの理解を支える認識として設定した、①ロボットの動きにより搭載された距離センサの値が時々刻々変化すること、②センサの計測と制御が「つねに」行われており、③プログラムとして記述した条件文により状態遷移が生じ動作が切り替わること、の 3 つが生徒に定着していたことの傍証になると考える。

また、第 5 時限において、与えた課題を日本語で構想した後プログラムに置き換える際、図 6.4 の状態遷移図における、「左へ」状態から左右のセンサの値により他の 2 状態のいずれかに遷移する箇所について、「同じ状態に」対して遷移の条件文を 2 行書いた場合、「下の行の影響で上の行の定義が消えてしまうのではないか」、と当初の予想を超える疑問を持ち、授業実践者へ質問する場面が、どのクラスにもあった。本教材におけるドリトルの拡張では属性を追加するよう実装したため問題ないが、オブジェクトによっては属性を上書きすることもある。このことから、ドリトルを用いたプログラミングの学習を通し、プログラムへの深い理解を持つ生徒が多数おり、ドリトルが中学生の教材として十分機能することと、本教材での拡張がドリトル学習経験を損なわず、受容可能なことを示したと考える。

ただし、これらの生徒を含め、本教材で「状態」の言い換えに用いた「モード」の語に関する違和感の声は少なからずあった。今後の検討を要する。

本研究の成果は、計測値として「距離」という連続量を扱い、複数の状態を持つ計測制御システムの仕組み理解を伴ったプログラミングが、中学生にも可能であることを示したことにあると考える。時間的に変化する連続量を制御に利用するモデルとして、コンピュータによる模型自動車の制御を用い、検証授業では、自動運転をイメージした課題を解決するためのプログラミングを、状態遷移によって考えさせ、正しくプログラムできることを確かめた。計測値の変化、コンピュータが行う判断と状態遷移、制御の状況を可視化する画面を開発した。また、ドリトルに状態遷移オブジェクトを追



加し、状態遷移を直接プログラミングできるようにした。可視化画面を観察しながらプログラムした実機の動きを見比べることで、状態遷移を伴う計測制御システムの仕組みを体験的に理解することができたと考えられる。事後調査において、次の動作を予測させたところ、正解かどうかによらず、理由の文にシステムの動作機序を整然と記述する記述が 6 割に及んだ。このことが、本研究の成果を裏付ける根拠になると考える。

なお、本研究の検証授業では、本来ひとつの制御であるところを状態に分割して生徒に考えさせた。「左壁に沿って走る」を 2 状態としたことがその例である。本研究で状態遷移を扱うにあたり、計測値の変化をイベントと同一視したモデルを用いたことによる制約である。今後、一般的なモデルを扱う可視化教材について研究を展開していきたい。

# 第5章 教員養成課程での状態遷移の教授法

前章までの研究では、中学生を対象に、簡単な問題から状態遷移モデルを考え、状態遷移を直接プログラムする言語を用いる経験を通して、複数の状態を持つ計測制御の課題について解決できることを実証した。これからの中学校の計測制御プログラミングの学習に求められる内容は、状態遷移によって問題をモデリングすることが必須である。しかし、状態遷移を直接プログラムする経験だけでは、状態遷移が適した部分を考えることには役立っても、手続きによって考えるべき多くの部分の理解にはつながらない。手続き型のモデルとして状態遷移を表す方法を理解することは、こうした相互のギャップの解消に必要である。特に、教員養成課程は、教師に必要な様々な知識技能を学んでいる段階であることから、これから必要となる新しい概念を学ぶべき位置にある。そこで本章では、状態遷移を教えるにあたり、これを手続き的に理解させ、手続きのプログラムで状態遷移を扱う方法について述べる。

## 5.1 複数の状態を持つ計測制御システムの学習における課題

これまで中学校で行われてきた計測制御の学習では、プログラミングするにあたり、計測制御する問題を手順に分解して考え、フローチャートを描いて手順の組み合わせを設計し、プログラムを記述して実行し、意図した動作ができているかを確認、デバッグする考え方で授業が行われている。しかし、フローチャートによるプログラムの設計は、問題を正しく表しているかを確認するには表現力が乏しく、誤った手順の組み合わせに気づく手段もなく、デバッグに役立たないことも多かった。また、多少複雑な手順に対しては、フローチャートでは流れを示す線が交差を避けられないことがあり、作成にも、読み取りにも困難さが生じてしまう。しかし、中学校段階の教育においてはフローチャートが教師や生徒にとってのプログラムを考える道具であり、それに代わるものが乏しかった。

しかし、前節までの議論で、複数の状態を持つ問題に対しては状態遷移で問題をモデリングすることで問題を分解して考えることができ、この考え方を中学生が理解して使うことができることが示された。特にこれからの中学校技術の学習で求められている、複数の状態を持つ計測制御システムを扱うために、複雑さへの対応のため、状態遷移を授業に導入することが避けられないといえる。

これに対し、第3章、第4章での提案は、状態遷移を直接記述するプログラミング言語を用いることで、状態遷移図など、状態遷移を考える道具を用いてモデルを考え、そのまま言語に置き換えることで、複数の状態を持つ問題を解決できることを示した。ただし、提案した言語であっても、状態における動作を手続きで記述するハイブリッド言語であり、手続きについて考えることは必要である。

すなわち、手続きによるプログラミングの学習の重要性は変わらない。

したがって、状態遷移と手続きによるプログラミングを接続するための方法論が求められているといえる。

## 5.2 状態遷移と手続きを接続するためのアプローチ

コンピュータのプログラムのなかで、状態を扱う場面は珍しくない。変数の値によって処理の分岐をすることは、現在では代替手段があるため推奨されないが、小さな全体が見えるプログラムのなかで誤りなく扱う限りにおいて、有用な手段である。かつては複数のフラグをループのなかで組み合わせて使い、例えば通信プロトコルを解釈して実行するプログラムなどが書かれ、こうしたプログラムは状態機械と呼ばれてきた。状態機械は有限状態オートマトンでモデル化できることから、有限状態オートマトンの図表現である状態遷移図が状態機械の設計に用いられてきた。

研究の分野では、状態遷移を記述できるプログラミング言語は、無人ローバーやロボットなど、複雑かつ誤りが許されない対象の行動を記述する目的で提案され、利用されてきた[46][47]。組み込みシステム開発において、状態遷移は避けることができないため、組み込み開発で一般的な C 言語に状態遷移を記述する機能を組み込む試み[48]や、手続きで書かれた過去のプログラムを解析して状態機械を発見し、保守に役立てる研究[49][50]、状態遷移により設計され状態機械として書かれたプログラムの正しさを自動検証するソフトウェア[51][52]など、ソフトウェアの品質に関わる研究が行われている。

組み込みシステムのソフトウェア開発では、状態遷移図による設計では遷移条件の見落としが致命的な不具合を起こし、特に自動車など人の生命に関わるソフトウェア開発では、設計に際し、状態遷移図ではなく、伝統的な状態表を用いて状態遷移を表現し、未定義部分に適切な処理を割り当てる教育が行われている[53]。ソフトウェアが持つ状態遷移の規模が大きくなるにつれ、伝統的な状態表に対する改善も模索されている[54]。

このように、状態遷移と手続きを接続する必要性は研究よりもソフトウェア開発の実務からの需要が大きく、効果的な方法の探求は業務のなかで行われる技術的専門性を持つ人々の間で行われていることが推察される。ここで状態遷移と手続きの接続は、状態遷移モデルを手続き型プログラミング言語での記述に変換することである。

本研究の関心は、中学校教師という、プログラミングを専門としない人々に、状態遷移と手続きの接続を学ばせる方法にある。現在行われている、厳密性を求め、プログラミングの知識と経験を持つ人を対象とした技術的方法論は適切とはいえない。

そこで本研究の目的を、中学校教師が技術の授業で複数の状態を持つ計測制御システムのプログラミングを指導する際に、状態遷移でモデル化したシステムを、手続き型プログラミング言語に変換して記述することを可能とする、中学校教師が理解しやすい教授法の開発とする。ただし、こうした新しい試みは、すでに現場に出ている教師を対象に検証する機会がほとんどないことが問題となる。そこで、教師になるための学問と実践的知識技能を学ぶ途上であり、将来教師として授業を担当することが高く見込まれる教員養成大学生を対象として研究を行う。

## 5.3 状態遷移と手順の実行を関連付ける方法

### 5.3.1 状態遷移図からフローチャートへの変換

現在、中学校の技術科では、プログラムの記述にフローチャートが用いられている。よって、状態遷移図をフローチャートに変換することができれば、授業で活用することが可能と考えられる。

計測制御システムに関わる状態遷移において、最もシンプルなモデルは 2 状態を往復する遷移と考えられる。その例を図 5.2 に示す。

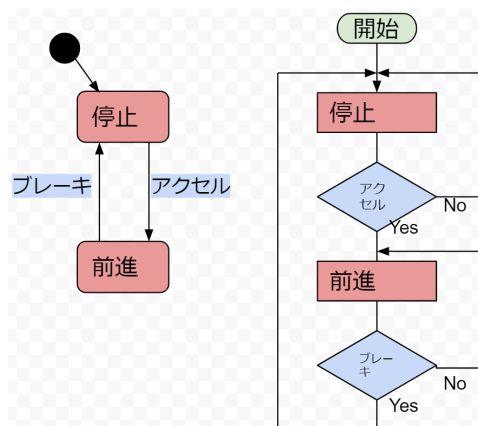


図 5.1 2 状態を往復する状態遷移と対応するフローチャート

しかし、これを 3 状態に延長しただけであっても、図 5.2 のように、3 分岐や、交差するループの構造が現れ、フローチャートに一定の習熟がなければ対応関係を理解することも困難となる。ここでは、「弱風」状態が、他の 2 つの状態への遷移に加え、同じ状態に留まるループを含めて 3 つの接続先を持つことが分かりにくさの原因となっている。

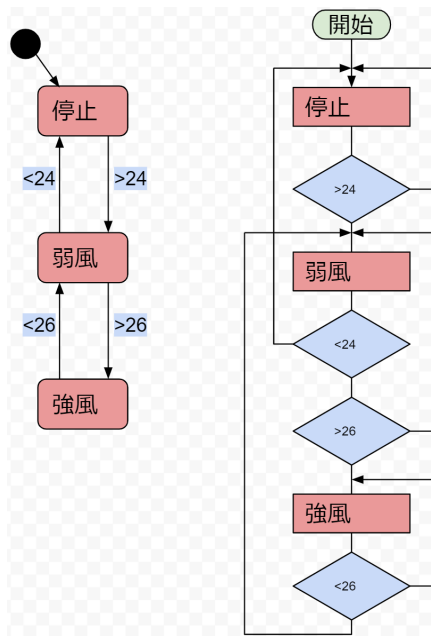


図 5.2 3 状態を往復する状態遷移と対応するフローチャート

現状では直感的に理解させられる変換方法の開発には至っていないが、基本的な発想として、計測制御システムでは多くの場合、状態は図 5.3 の構造が基本となっている。

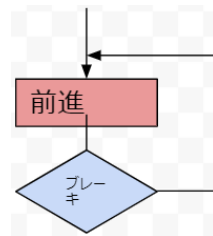


図 5.3 計測制御システムで一般的な状態の手順的な構造

したがって、少なくとも図 5.4 のように、2 状態を往復する状態遷移図は、状態を図 5.3 の構造の手続き構造モジュールに置き換え、遷移条件に基づく状態の結合関係を考慮して、これらを接続するという説明が、本研究が対象とする教員や教員養成大学生に理解させやすい順序と考えている。以後の変換手順を考えるにあたって予備調査を少数の大学生に行った際にも、図 5.1 については容易に変換できたが、図 5.2 については正解に至ることができなかった。

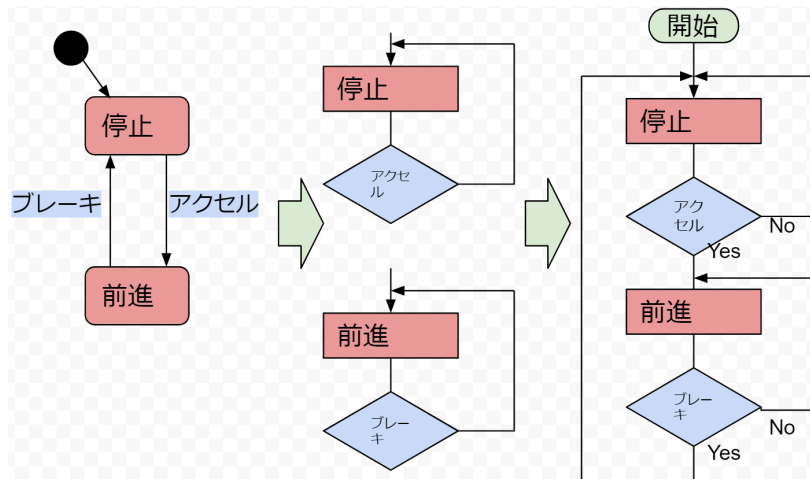


図 5.4 2 状態を往復する状態遷移図をフローチャートに変換する考え方

### 5.3.2 状態遷移図から手続きプログラミングへの変換

前節の考察により、「状態を、動作と継続判断の反復でとらえることができる」「状態遷移図からフローチャートへの手順化は、状態が 3 個以上になると複雑になり大学生でも理解が困難になってしまう」ことがわかった。このことは、状態の考え方が計測制御システムに適しているにも関わらず、中学校技術科で利用されてこなかった一因と考えられる。

そこで、複数の状態を持つ計測制御システムのプログラムを、通常の手続きプログラミングに変換する手順を検討する。図 5.5 に、提案する変換手順を示す。

ステップ 1 は状態遷移図であり、計測制御システムの複数の状態をモデル化する。4 章で示したように、中学生は 3 状態から 4 状態のモデルを理解し、活用することが可能である。

ステップ 2 は、状態を直接記述できる言語で表現したプログラム記述である。3 章と 4 章で示したように、中学生は状態によるプログラム記述を行うことが可能である。

ステップ 3 は、状態から手続き的なコードの断片を生成する。コードの内容は、「その状態で行う動作と、他の状態に遷移する条件」の記述である。この例では状態を変数 **st** で表現した。

ステップ 4 は、最終的な手続き的なプログラムを作成するためのひな型の準備である。先頭行には開始状態を定義し、その後は反復の中に、状態の数だけ状態ごとの分岐を準備する。このコードは教師が用意しておくことも考えられる。ここでは **while()** の継続条件を省略しているが、基本的には無限ループを想定している。

ステップ 5 は、ステップ 3 で生成した状態を表すコードを、ステップ 4 のひな型のコードに入れる。このプログラムを実行すると、最初に状態 **st** がセットされ、**while** による反復の中で状態 **st** に対応する処理が **if** で表現された分岐によって実行される。

この変換手順は、前節で示したフローチャートに変換する手順と比べて、状態の数が増えたときも線が交差するような複雑さが生じない、そして状態ごとの反復を省略できる利点がある。また、状態

によって思考を要する点はステップ3のコード断片の生成だけである。ステップ4とステップ5は示された手順通りに作業を行えばよいことから、この変換手順を中学校の授業で活用することは可能と考えることができる。

ステップ1 状態遷移図	ステップ2 状態遷移プログラム (Rtoysなどで記述)	ステップ3 状態ごとの部品化 (変換作業)	ステップ4 手続き化のひな型 (教師が提示)	ステップ5 手続きプログラム化 (ひな型に部品を埋める)
<pre> graph TD     Start(( )) --&gt; Stop[停止]     Stop -- "&lt;24" --&gt; Weak[弱風]     Weak -- "&gt;24" --&gt; Stop     Weak -- "&gt;26" --&gt; Strong[強風]     Strong -- "&lt;26" --&gt; Weak     </pre>	<p>状態1： テスト 温度&gt;24 はい 状態2</p> <p>状態2： 弱風 テスト 温度&lt;24 はい 状態1 テスト 温度&gt;26 はい 状態3</p> <p>状態3： 強風 テスト 温度&lt;26 はい 状態2</p>	<pre>// 状態1 if (温度())&gt;24 {   st=2; }  // 状態2 弱風(); if (温度())&lt;24 {   st=1; } if (温度())&gt;26 {   st=3; }  // 状態3 強風(); if (温度())&lt;26 {   st=2; } </pre>	<pre>st=1; while() {   if (st==1) {     // 状態1   }    if (st==2) {     // 状態2   }    if (st==3) {     // 状態3   } } </pre>	<pre>st=1; while() {   if (st==1) {     if (温度())&gt;24 {       st=2;     }   }    if (st==2) {     弱風();     if (温度())&lt;24 {       st=1;     }     if (温度())&gt;26 {       st=3;     }   }    if (st==3) {     強風();     if (温度())&lt;26 {       st=2;     }   } } </pre>

図 5.5 状態移図を手続きプログラムに変換する提案手順

## 5.4 今後の課題

教員養成大学生 10 名程度を対象とした 2 回の予備実験では、図 5.4 を用いた説明に対する十分な理解が得られている。今後、様々な状態遷移モデルを検討し、Rtoys 等の授業で得られた状態遷移の教育利用の知見を多くの授業で活用できるように研究を進めたい。

## 第6章 結論

本論文では、2021年以後の中学校段階のプログラミング教育に求められている、「複数の状態を持つ、複雑な振る舞いを持つ計測制御システムを生徒が主体的に考えてプログラミングに取り組むことができる学習環境の開発」と、その学習効果について述べた。具体的成果は、次の3点である。これまでフローチャートのみでプログラムを設計しなければならなかった中学生が、状態遷移によるモデル化について理解し、プログラムできたこと。システムの仕組みを可視化することで、計測制御システムの基本的な仕組みを理解できたこと。計測値の可視化により、計測値の連続的な変化を理解して、システムのプログラムを作成できたこと、である。

第1章では、研究の背景、研究の必要性につながる計測制御のプログラミング教育における課題と、それを解決するための研究のアプローチについて述べた。

第2章では、中学校技術科で使われている教材と、国内及び海外の研究を調査することで、現在の技術教育の現状と課題を明らかにした。また、情報教育において、中学校技術科が担う情報技術やプログラミングに対する期待が増していることを示し、特に、新学習指導要領が求める計測制御システムの具体化には、状態遷移の理解が中学校段階で求められることを示した。さらに、これに対応できる教員養成の重要性を明らかにした。

中学校技術科で使われている教材と取り扱われる題材について、先行研究を踏まえ、技術教育の現状と課題を明らかにした。また、これからの計測制御システムのプログラミングで必要となる、状態遷移モデルの学習について、従来の取り組みと研究上の課題を示した。さらに、これに対応できる教員養成の重要性を明らかにした。

第3章では、複数の状態を持つ計測制御のプログラムを状態遷移として直接記述可能な学習環境 Rtoys を提案し、公立中学校において、Rtoys と従来の手続き型言語を用いた授業結果の比較を行った。Rtoys は状態をひとつのウィンドウとして画面表示し、命令語をマウスのドラッグアンドドロップ操作により状態内に配置していく、ビジュアルプログラミング言語である。状態はウィンドウであり、画面内を自由にマウスで移動でき、そこへ命令語を配置する。また、画面内にロボットの画像を表示し、センサの反応を含めてプログラムしたロボットの動作を対話的にシミュレーションする機能を持つ。検証授業では、条件を揃えるため、Rtoys と、手続き型プログラミング向けに機能を入れ替えた手続き Rtoys を用いた。状態のない課題から3状態まで5ステップの難易度を持つ課題により学習を行った。中学校の2クラスで Rtoys を用いるクラスと手続き Rtoys を用いたクラスで授業を行い、課題を生徒自身で考え正答に至った人数の集計を行ったところ、状態 Rtoys では3状態で半数を下回る正答者数となったものの、すべての課題で正答に至ったのに対し、手続き Rtoys では状態を



含む課題において有意に低い差があった。このことから、Rtoysにより中学生が状態を理解し、状態遷移を直接プログラムできることが示された。

第4章では、「センサによる計測値をコンピュータが処理・判断し、アクチュエータを制御する」という計測制御システムの基本的動作を学習する新しい教材を提案した。この教材は、計測制御の対象として車型のロボットを状態遷移を用い、プログラム実行時に「センサ値、状態、実行中のプログラム」の変化をリアルタイムで表示する可視化画面を表示する。公立中学校で検証授業を行い、開発した教材を用いて状態遷移を学習し、ロボットがセンサの値に応じて状態遷移する課題に取り組ませた。事前調査の結果より、学習前の生徒は計測制御システムについてのイメージが乏しく、特にコンピュータの役割が理解できていなかった。事後調査では、最後の課題の設定のもと、ある場面を示し、次にどのような動作をするのかを3つから選択させ、その理由を記述させた。事後調査で生徒の8割が正解を選び、6割が解答の正誤に関わらず、計測制御システムの動作機序を正確かつ整然と記述できた。このことから、本教材の利用を通し、計測制御システムの仕組みについて体験的に理解し、計測値の連続的な変化を見ながら状態遷移を持つシステムの挙動を考えてプログラムできることが実証された。

第5章では、教員が状態遷移を理解して授業で扱えるようになることを目的として、また状態遷移モデルと手続き型プログラミングの意味的なつながりを教師・生徒の両方が自ら考えられることを目的として状態遷移図から手続き的なフローチャートへの変換手順を検討した。教職課程で学ぶ大学生への実践的教育に導入したところ、状態のモジュールとその接続がはっきりしている2状態について効果のある方法であることが明らかになった。

本研究では、2021年度からの新教育課程の実施に向けて、複数の状態を持つ計測制御システムを体験・理解してプログラムを考えられる教材を提案し、その効果を検証することができた。今後は研究成果を公開して全国での利用を可能にしたい。また、授業を行う教師の教育も必要であるため、本務である教員養成の中で研究成果を生かした教育用研修資料などを開発していきたいと考えている。

# 謝辞

本論文をまとめるにあたり、大阪電気通信大学松村雅史教授をはじめ、吉田正樹教授、藤川智彦教授、兼宗進教授に御助言と御指導をいただきました。感謝いたします。

また、株式会社イーテキスト研究所代表原久太郎氏には、たくさんの研究の場を提供していただきました。いろいろな方との出会いを通して、研究のための人とのつながりができました。深く感謝いたします。本論文中の第 3 章の研究は、国際基督教大学青木浩幸助教と三重県大台町立宮川小学校井戸坂幸男校長の研究成果をもとに検証を進めました。第 4 章の研究は、愛知教育大学本多満正教授と愛知県春日井市立柏原中学校恩田健司先生に協力をいただきました。研究に関して、御示唆をいただきました高麗大学李元揆教授、静岡大学紅林秀治教授に感謝いたします。

検証授業におきましては、三重県松阪市立飯南中学校、愛知県春日井市立柏原中学校の生徒の皆さんに協力をいただきました。授業だけでなく、アンケートや検証テストにも積極的に協力していただいたおかげで有用な知見を得ることができました。感謝いたします。

# 参考文献

- [1] 野波 健蔵: 世界のドローン開発動向と農業応用からみた課題と展望, 計測と制御, Vol. 55, No. 9, 計測自動制御学会, pp. 780-787 (2016).
- [2] Dimitrios Serpanos: The Cyber-Physical Systems Revolution, Computer, Vol. 51, No. 3, IEEE, pp. 70-73 (2018).
- [3] Kevin Ashton: That 'Internet of Things' Thing - In the real world, things matter more than ideas., RFID Journal, Jun 22 (2009).  
<https://www.rfidjournal.com/articles/view?4986>
- [4] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami: Internet of Things (IoT): A vision, architectural elements, and future directions, Future Generation Computer Systems, Vol. 29, Elsevier, pp. 1645-1660 (2013)
- [5] 松村雅史, 辻童之介: 笑い声の無拘束・長時間モニタリング: 爆笑計, 電子情報通信学会技術研究報告, SP, 音声, Vol.105, No.370, pp.7-12 (2005).
- [6] 岡本敏雄, 高橋参吉, 西野和典: 情報科教育法 第2版, 丸善 (2015).
- [7] 伊藤秀郎: 情報社会における中学校技術科の情報領域の在り方の考察, 学校教育学研究, 2 第 26 卷, pp 75-81 (2014).
- [8] 林向達: 日本の教育情報化の実態調査と歴史的変遷, 日本教育工学会研究報告集 , 12(4), pp.139-146 (2012).
- [9] 岡 俊博, 吉田 誠, 葉山 泰三: 「情報基礎」の授業における BASIC プログラミングの教授方法の関する研究, 奈良教育大学教育実践研究指導センター研究紀要, 第 2 巻, pp. 1-11 (1993).
- [10] 須曾野仁志, 木谷康司. 下村勉: Logo プログラミングを主体とした中学校「情報基礎」の実践, 日本教育情報学会第 10 回年会, pp. 106-107 (1994).
- [11] 荒木直美, 斉藤俊則, 大岩元: 義務教育課程における情報教育~ 中学校技術家庭科 「情報基礎」の現状より~, 情報処理学会研究報告コンピュータと教育 (CE) Vol. 1996-CE-042, pp. 1-8 (1996).
- [12] 古平真一郎, 石島隆志, 坂本弘志, 高橋脩太, 宮川こずえ, 伊藤秀哲, 針谷安男: 自律型ロボット教材を用いたプログラム学習に対する効果, 宇都宮大学教育学部教育実践センター紀要, 第 30 号, pp. 539-548 (2007).
- [13] 村松浩幸: 中学校技術・家庭科技術分野における情報の学習の動向—プログラムによる計測と制御学習の必修化を受けて—, ICT・Education, No.50, 日本文教出版, pp.6-9 (2013).
- [14] 川俣純: 「なぜ, 計測・制御を学ぶのか?」生徒の疑問に答える授業展開の工夫—身近な製品のプ

- プログラムをフローチャートで考える, ICT・Education, No.53, 日本文教出版, pp.22-25 (2015).
- [15] 文部科学省: 平成 29・30 年改訂 学習指導要領、解説等,  
[http://www.mext.go.jp/a\\_menu/shotou/new-cs/1384661.htm](http://www.mext.go.jp/a_menu/shotou/new-cs/1384661.htm)
- [16] 原田康德, 勝沼奈緒実, 久野 靖: 公立小学校の課外活動における非専門家によるプログラミング教育, 情報処理学会論文誌, Vol. 55, No. 8, pp. 1765–1777 (2014).
- [17] 渡邊景子, 時川えみ子, 辰己 丈夫: 小学校におけるクラブ活動でのプログラミング実践報告, 情報処理学会研究報告コンピュータと教育 (CE) Vol. 2016-CE-136 No.15, No. 15, pp. 1-6 (2016).
- [18] 内閣府: 未来投資戦略 2018 — 「Society 5.0」「データ駆動型社会」への変革— (2018),  
[http://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/miraitousi2018\\_zentai.pdf](http://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/miraitousi2018_zentai.pdf)
- [19] 文部科学省: Society 5.0 に向けた人材育成 ～ 社会が変わる、学びが変わる ～, (2018),  
[http://www.mext.go.jp/b\\_menu/activity/detail/2018/20180605.htm](http://www.mext.go.jp/b_menu/activity/detail/2018/20180605.htm)
- [20] Sertac Karaman, Ariel Anders, Michael Boulet, Jane Connor, Kenneth Gregson, Winter Guerra, Owen Guldner, Mubarik Mohamoud, Brian Plancher, Robert Shin, and John Vivilecchia: Project-based, Collaborative, Algorithmic Robotics for High School Students: Programming Self-driving Race Cars at MIT, 2017 IEEE Integrated STEM Education Conference (ISEC). IEEE, pp. 195-203 (2017).
- [21] Illah R. Nourbakhsh, Emily Hamner, Kevin Crowley, and Katie Wilkinson. Formal measures of learning in a secondary school mobile robotics course. Proceedings of IEEE International Conference on Robotics and Automation 2004. IEEE, pp. 1831-1836 (2004).
- [22] Wiesner, Bernhard, and Torsten Brinda: Using Robots as Teaching Aids in Early Secondary Informatics Education (2008).
- [23] 大日本印刷株式会社: 諸外国におけるプログラミング教育に関する調査研究 : 文部科学省平成 26 年度・情報教育指導力向上支援事業 : 報告書, 平成 26 年度文部科学省委託事業 (2015).
- [24] 文部科学省: 中学校学習指導要領解説 技術・家庭編 (2018).
- [25] 津村俊弘: 無人搬送車とその制御, 計測と制御, Vol. 26, No. 7, 計測自動制御学会, pp. 593-598 (1987).
- [26] ヤマザキ教育システム: プロロボ USB プラス.  
<http://www.yamazaki-kk.com/technique/>
- [27] 東京書籍: 新版 新しい技術・家庭 技術分野, pp.246-249 (2015).
- [28] 開隆堂出版: 技術・家庭 [技術分野] , pp.230-233 (2015).
- [29] 久富電機産業, オーロラクロック 2N,  
<http://www.hisatomi-kk.com/seihin/joho/UC-7/UC-7.html>

- [30] LEGO Mindstorms EV3,  
<https://www.legoedu.jp/ev3/>
- [31] LEGO Eduction: LEGO MindStorms EV3 ユーザーガイド,  
<https://education.lego.com/ja-jp/support/mindstorms-ev3/user-guides>
- [32] Jon Whittle, John Hutchinson, and Mark Rouncefeld: The State of Practice in Model-Driven Engineering, IEEE Software, Vol. 31, Issue 3, pp.79-85 (2014).
- [33] OMG: OMG Unified Modeling Language (OMG UML) version 2.5.1 (Dec. 2017),  
<https://www.omg.org/spec/UML/2.5.1>
- [34] David Harel. Statecharts: A visual formalism for complex systems, Science of computer programming, Vol.8, No.3, pp.231-274 (1987).
- [35] 香山瑞恵, 小形真平, 増元健人, 伊東一典, 橋本昌巳, 大谷真: 状態遷移図作成に際する初学者の誤り分析とそれに基づく教育方法の検討, 研究報告コンピュータと教育 (CE), Vol. 2012-CE-117, No. 7, 情報処理学会, pp.1-9 (2012).
- [36] 香山瑞恵, 小形真平, 永井孝, 横田寛明, 増元健人, 橋本昌巳, 大谷真: 初学者向けの状態遷移図による振舞に関する概念モデリング教育へのモデル駆動開発方法論に基づく学習環境導入の効果, 組込みシステムシンポジウム 2014, 情報処理学会, pp.108-113 (2014).
- [37] Alan Key: Squeak Etoys, Children & Learning (2005).  
[http://www.vpri.org/pdf/rn2005001\\_learning.pdf](http://www.vpri.org/pdf/rn2005001_learning.pdf)
- [38] Seymour Papert: MindStorms: Children, Computers, and Powerful Ideas, Basic Books (1980).
- [39] Mark J. Guzdial, and Kimberly M. Rose: Squeak: Open personal computing and multimedia, Prentice Hall PTR (2001).
- [40] Hiroyuki Aoki, DongHee Park, and WonGyu Lee: Robot Programming with Squeak "Rtoys" – Connection between Drawing Worlds and the Real World, International Conference on Technology Education in the Asia Pacific Region Conference, pp.390-398 (2009).
- [41] 西ヶ谷浩史, 青木浩幸, 井上修次, 鎌田敏之, 兼宗進, 紅林秀治: 3 軸自律制御ロボットを用いた制御の学習, 情報教育シンポジウム(SSS2006), 情報処理学会, pp.319-324 (2006).
- [42] スタジオ ミュウ: ミュウロボ,  
<http://www.studiomyu.com/>
- [43] 井戸坂幸男, 青木浩幸, 李元揆, 久野靖, 兼宗進: 状態遷移概念を利用した制御プログラミングの学習効果, 日本産業技術教育学会誌, Vol.53, No.3, pp.179-187 (2011).
- [44] 兼宗進, 村松浩幸: ドリトルによるプログラミング学習, 東京書籍 (2017).

<https://ten.tokyo-shoseki.co.jp/detail/101646/>

- [45] 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 (2011).
- [46] Michel Ingham, Robert Ragno, and Brian Williams: A Reactive Model-based Programming Language for Robotic Space Explorers, Proceedings of ISAIRAS-01 (2001).
- [47] Philipp Allgeuer, and Sven Behnke: Hierarchical and State-based Architectures for Robot Behavior Planning and Control, Proceedings of 8th Workshop on Humanoid Soccer Robots, International Conference on Humanoid Robots (Humanoids), arXiv preprint arXiv:1809.11067, (2018).
- [48] 小倉信彦, 谷川郁太, 渡辺 晴美 : 状態遷移言語による組込みソフトウェア開発, 情報処理学会研究報告システム LSI 設計技術 (SLDM) , Vol. 2011-EMB-20, No. 48, pp. 1-6 (2011).
- [49] 山本涼太, 吉田則裕, 竹田彰彦, 舘伸幸, 高田広章: 組込みソフトウェアを対象とした状態遷移表抽出手法, 電子情報通信学会技術研究報告, 116 (128), pp. 13-18 (2016).
- [50] 竹田彰彦, 状態遷移表のリバースモデリングへの適用, ET2013 セミナー資料, 組込みシステム技術者協会 (2013).
- [51] 金周慧, 松原豊, 高田広章: 状態遷移図に着目した安全分析手法, 電子情報通信学会論文誌 A Vol. J95-A No. 2, pp.198-209 (2012).
- [52] 大貫智洋: 制御ソフトウェアの仕様整合性検証技術の調査と分析, 情報処理学会研究報告ソフトウェア工学 (SE) , Vol. 2012-SE-175, No. 10, pp. 1-8 (2012).
- [53] 組込みシステム技術協会: 状態遷移表設計における S P L E 実践ガイド (2013)  
[http://www.jasa.or.jp/TOP/download/technical/24\\_design.pdf](http://www.jasa.or.jp/TOP/download/technical/24_design.pdf)
- [54] 紫合治: 状態指向の状態遷移表, 情報処理学会研究報告ソフトウェア工学 (SE) , Vol. 2007-SE-157, No. 97, pp. 79-86 (2007).

# 関連発表

## 主論文

### 論文誌（査読あり）

1. 鎌田敏之, 恩田健司, 本多満正, 兼宗進: 中学・高校生に対し並行処理を意識させる自動化システム教材の開発, 日本情報科教育学会誌, Vol. 10, No. 1, pp. 33-44 (2017).
2. Hiroyuki Aoki, JaMee Kim, Yukio Idosaka, Toshiyuki Kamada, Susumu Kanemune and WonGyu Lee: Development of State-Based Squeak and an Examination of Its Effect on Robot Programming Education, KSII Transactions on Internet and Information Systems, Vol. 6, No. 11, pp. 134-153 (2012).
3. 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モータ制御用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).

### 国際会議（査読あり）

1. Toshiyuki Kamada, Mitsumasa Honda, Susumu Kanemune: Development of Learning Material toward Basic Understanding of Information Systems for Secondary Students, IPSJ-CE 141 International Session, Taipei, pp. 1-7 (Nov. 2017).
2. Toshiyuki Kamada, Hiroyuki Aoki, Shuji Kurebayashi, Yoshikazu Yamamoto: Development of an Educational System to Control Robots for All Students, Lecture Notes in Computer Science, Vol. 5090, Springer, pp. 63-74 (2008).

### 研究紀要（査読あり）

1. 鎌田敏之, 田端将太郎, 本多満正: プログラミング学習における状態遷移概念の利用とその効果, 愛知教育大学研究報告, Vol. 68, No. 1, pp. 89-95 (2019)
2. 鎌田敏之, 本多満正, 木下崇, 秋山大翼: ネットワークを用いた双方向性のあるプログラミングの授業へ向けた教員支援の試みとその評価, 愛知教育大学研究報告, Vol. 67, No. 1, pp. 31-36 (2018).
3. 鎌田敏之: 自律ロボット制御を通じたプログラミング教育の試み, 愛知教育大学教育実践総合センター紀要, Vol. 5, pp. 53-59 (2002).

## 参考論文

### 国際会議（査読あり）

1. Tomoko Yoshida, Toshiyuki Kamada, Ryota Nakamura and Toshio Matsuura: Development and Use of a Programming Environment for Learning the Mechanism of Measurement and Control by Programs, The Second International Conference on Computer Science, Computer Engineering, and Education Technologies, Kuala Lumpur, Malaysia, pp. 32-40 (Sep. 2015).
2. Hiroyuki Aoki, Toshiyuki Kamada and WonGyu Lee: Development of State-Based Squeak to Support Robotics Programming, The 3rd International Conference on Internet, Sepang, Malaysia, (Dec. 2011).
3. Shuji Kurebayashi, Susumu Kanemune and Toshiyuki Kamada: Learning Computer Programming with Autonomous Robots, Lecture Notes in Computer Science, Vol. 4226, Springer, pp. 138-149 (2006).
4. Shuji Kurebayashi, Hiroyuki Aoki, Toshiyuki Kamada, Susumu Kanemune and Yasushi Kuno: Proposal for Teaching Manufacturing and Control Programming Using Autonomous Mobile Robots with an Arm, Lecture Notes in Computer Science, Vol. 5090, Springer, pp. 75-86 (2008).
5. Shuji Kurebayashi, Susumu Kanemune and Toshiyuki Kamada: The Effect of Learning Programming with Autonomous Robots for Elementary School Students, Proceedings of EuroLogo 2007, Bratislava, Slovakia, pp. 1-9 (Aug. 2007).

### 口頭発表

1. 田端将太郎, 鎌田敏之, 本多満正: 計測と制御のプログラミング学習における状態遷移の概念形成に関する研究—技術科教員養成における実践と分析—, 日本産業技術教育学会第35回東海支部大会, 静岡大学, pp. 11-14 (2017).
2. 小島正嗣, 鎌田敏之, 重堅斗: 教員養成におけるプログラムによる計測と制御の考察～2つのアプローチの比較～, 日本産業技術教育学会第34回東海支部大会, 三重大学 (2016).
3. 鎌田敏之: 時系列計測データを活用する制御プログラミング学習システムの構築, 日本産業技術教育学会第34回東海支部大会, 三重大学, pp. 1-4 (2016).
4. 鎌田敏之, 大村基将, 兼宗進: 「プログラムによる計測と制御」概念形成を意識したコースカリキュラムとその評価, 情報処理学会研究報告コンピュータと教育(CE), 2015-CE-131(10), 静岡大学, pp. 1-6 (2015).
5. 鎌田敏之: Raspberry Pi を用いた中学校教員養成における計測・制御プログラミング環境の構築と実践, 情報処理学会研究報告コンピュータと教育(CE), 2014-CE-126(10), 奈良女子大学, pp. 1-6 (2014).



6. 鎌田敏之: 教員養成におけるロボットを用いた計測制御の授業実践, 情報処理学会研究報告コンピュータと教育(CE), 2009-CE-99(11), 静岡県立大学, pp. 1-6 (2009).
7. 青木浩幸, 西ヶ谷浩史, 鎌田敏之, 原久太郎, 李元揆, 紅林秀治: 中学校技術・家庭科における計測制御プログラムモデル, 情報処理学会情報教育シンポジウム第10回記念大会, 済州大学校, 大韓民国, pp. 1-6 (Aug. 2008). (査読あり)
8. Toshiyuki Kamada, Hiroyuki Aoki, Shuji Inoue and Shuji Kurebayashi: Structured Robot-Control Language on Dolittle Syntax, 2nd International Workshop on Information Science Education and Programming Languages, Korea University, pp. 1-8 (Jan. 2007).
9. Toshiyuki Kamada and Susumu Kanemune: Toward the Web-based Dolittle Programming Environment with High Usability, International Workshop on Information Science Education and Programming Languages, Korea University, pp. 1-6 (Jan. 2006).
10. 鎌田敏之, 青木浩幸, 井上修次, 紅林秀治: 教材用自立型制御ロボット基板の開発と授業実践, 日本産業技術教育学会第49回全国大会, 高知大学, p.100 (2006).
11. 鎌田敏之, 青木浩幸, 井上修次, 紅林秀治: ロボット制御のための構造化プログラミング言語の設計, 日本産業技術教育学会第24回東海支部大会, 愛知教育大学, pp. 1-4 (2006).
12. 鎌田敏之, 井上修次, 紅林秀治: 教材用自立型ロボット基板の開発, 日本産業技術教育学会第23回東海支部大会, 静岡大学, pp. 137-140 (2005).