

車型ロボット'Beauto chaser'と日本語オブジェクト指向  
プログラミング言語ドリトルを利用した、「モデル化と  
シミュレーション」分野についての実践的研究

野部 緑 中田雅士 大島嘉一

要約

情報 B の「モデル化とシミュレーション」は教えるのに自信がないと答える教員が多く、また教えていないといわれる单元でもある。

そこで、理解しやすくまた興味や関心を引き起こす教材として、車型ロボットと日本語プログラミングを利用した授業を考案した。単に「ロボット制御」の教材にしないために、「モデル化とシミュレーション」の单元で、ドリトルと車型ロボットを利用し、生徒にとって身近なシミュレーションの体験をさせることを目的とした。

「モデル化とシミュレーション」だけであれば、実際に動く車型ロボットは必要ないが、画面だけで物事を考えるより、考え予想したことが実際に動きとなってあらわれるのは、生徒の興味をひき、難しいといわれるモデル化とシミュレーションの理解の助けになる。

「モデル化とシミュレーション」は、現実の社会で実験等を行っていくものをモデル化し、コンピュータ上で事象を再現するということが多い。しかし、このモデル化という作業の過程で、現実の要素をすべて取り込むことはできない。

そこで、プログラミング言語を利用して、コンピュータ画面上と、車型ロボットで同じ問題を扱い、これを比較することで、現実との違いを考えさせる。

実際の教材には、ヴィストン社の車型ロボット'Beauto chaser'を利用し、日本語で記述できるプログラミング言語「ドリトル」を用いた。USB ポートを利用して、命令を転送し、その命令がコンピュータの画面と同様に動くかどうかということを確認させ、またなぜ違いが生じるのかということを考えさせた。

授業で取り扱ったモデル化の課題は簡単なものであったが、実際に車型ロボットが動くことが面白いという感想が多く、興味や関心をひく教材を考えるという目的は果たせたと考える。また、思うようにならない試行錯誤が面白かったという感想もあった。

一方で、「ドリトル」からの車型ロボットへ転送できる命令の数が少なく、当初考えていたモデル化の問題を授業で行うことができなかつたのは今後の課題である。助成期間は終わったが、引き続き、これらを使った教材の改良を行っていきたい。

## 1. 研究の動機・目的について

### 1.1 研究の動機

情報Bの単元のうち、「モデル化とシミュレーション」は実際には授業で行われていないことが多い。財団法人 コンピュータ教育開発センターが平成20年度に行った「高等学校等における情報教育の実態に関する調査」<sup>1)</sup>によると、教えていると回答したのは 6.9%である。もっとも、情報 B の開講の割合も低い、それでも 10%より少ないということは、情報 B を教えていてもこの單元には触れていないという可能性が高い。また同じ調査では、「モデル化とシミュレーション」は 58.1%、「コンピュータによる計測・制御」は 48.8%が自信がないと答えており、「情報の科学的な理解・問題解決」に分類される場所は、教える自信がないという教員が多いという結果がでている。

ひとつには、「モデル化とシミュレーション」や「コンピュータによる計測・制御」は日常生活とかかわりが薄いと思われていることにあるだろう。たとえば、教科書<sup>2)</sup>に載っている「モデル化とシミュレーション」の例は次のようなものである。

- 「紙人形販売のシミュレーション」
- 「水槽の水量変化」
- 「害虫の増加する様子」
- 「推薦タンクにおける水量変化」
- 「モンテカルロ法を利用した円周率の計算」
- 「太陽光発電による発電量」
- 「日本カモシカの生息数の変化」
- 「乗車券購入にかかる待ち行列」

シミュレーションが現実では確かめにくい、自然現象や人工的な現象をモデル化して、計算において、検討するための材料であるという点においては、これらの例は問題ないだろうが、「文化祭における紙人形の販売」以外は、高校生にとって身近ではなく、興味がわかないだろう。

また、学習指導要領<sup>3)</sup>の解説では、簡単にモデル化できる内容をと書かれているが、教科書を見ている限り、モデル化が難しいものが多い。それは、ひいては、シミュレーションとして扱う時に、ソフトウェアの利用が難しくなるという結果になるだろうし、教えるににくい原因にもなっていると考えられる。また、数式が少し複雑になると、学力があまり高くない生徒には、コンピュータでシミュレーションを行う以前に、モデル化の段階で理解しにくいことも多い。

しかし、モデル化といっても何も難しい現象をなぞる必要はなく、現実の動きをコンピュータ上でいかに再現するかということ、そのためにはどのような手順があるかということが生徒に理解されればよいのである。生徒が関心を持ち、また理解をしやすい教材を作りたい。その考えに基づいて、分かりやすい例を使ったモデル化、さらにそれコンピュータで計算し再現するだけでなく、実際の機械を動かして体験することでモデル化への理解を進めたいというのが動機である。

「モデル化とシミュレーション」では、実際の機械を動かすことまでは求められていない。しかし、理解をふかめるために、あえて、制御誤差を体験して欲しいと考えた。なぜなら、モデル化やシミュレーションは何のために行うのか確認して欲しいためである。

教科書に載っている例はさまざまであるが、「モデル化とシミュレーション」の目的は、問題を解決するために、現象をモデル化し、その現象をコンピュータ上で再現することである。

しかし、現実の事象のすべてがモデル化されているわけではない。数学的なモデルであれば、モデル化の際に省略する要素がなく、そのまま再現できることもある。しかし、物理的な事象などは多くの場合、モデル化のときに要素のいくつかが省かれている。したがって、コンピュータ上ではうまく再現できても、実際にはその通り行かないこともあるということも知っておく必要がある。

これは、そのようなこともあるという知識でもよいが、やはり実際に体験する方がより効果的であると考え、車型ロボットを導入したいと考えた。

## 1.2 研究の目的

高等学校の情報では、情報Bの科目に「モデル化とシミュレーション」という単元がある。この単元における目的は、高等学校学習指導要領解説によると、次のようになっている。

### (3) 問題のモデル化とコンピュータを活用した解決

#### ア モデル化とシミュレーション

身のまわりの現象や社会現象などを通して、モデル化とシミュレーションの考え方や方法を理解させ、実際の問題解決に活用できるようにする。

### (4) 情報社会を支える情報技術

#### ア 情報通信と計測・制御の技術

情報通信と計測・制御の仕組み及び社会におけるそれらの技術の活用について理解させる。

そして、内容の取扱いについては、さらに下記のように記述されている。

#### ア モデル化とシミュレーション

ここでは、コンピュータを用いて予測問題などを効果的に解決する方法として、モデル化とシミュレーションの考え方や方法を扱う。モデル化とシミュレーションそのものは、必ずしもコンピュータを利用することを前提としてはいない。したがって、コンピュータの活用という観点からは、どのような場面でどのように行えばコンピュータを用いたシミュレーションが有効なのか、コンピュータを用いたシミュレーションの特性や活用上の留意点の理解などについて学ぶ必要がある。ただし、既に確立されている定型的なモデルを知識として理解させることだけで終わらないように留意する。

特に、モデル化の仕方が異なるとシミュレーション結果が異なることについて認識させることは必要である。同じ課題に異なるモデル化を行い、生徒の分担によるシミュレーションを行わせ、その結果を比較させるなどの活動が考えられる。

また、モデル化の過程やシミュレーション結果の適切さを判断する場合に、実験や調査によるデータ収集を行い、それを活用して検証することの有効性などに触れることも考えられる。

(中略)

さらに、内容(4)のアで扱う制御や通信の学習と関連付けて、身近な制御・通信システムの動

作をモデル化し、コンピュータでシミュレーションしてみることも考えられる。その際、制御誤差やノイズの影響についてシミュレーションすることにより、内容(4)のAの学習を深め、内容(4)のイなどで発展させることもできると考えられる。

いずれにしても、ここではモデル化の方法を学ぶことが目的であり、生徒が各教科等で学習済みの知識を生かして簡単にモデル化できる程度の題材を取り上げることが望ましく、数理的、技術的な内容に深入りした題材は扱わない。

教科書を見ると「現象－要素を取り出しモデル化－シミュレーション」という流れになっていて、時間的な変化のモデルや確率的なモデル化などが取り上げられている。そして、コンピュータを使って実際に確かめてみようという構成になっている。

しかし、シミュレーションできることも必要であるが、より大切なことは、「要素を取り出しモデル化」という段階で、省かれている要素があること、モデル化を簡単にしすぎてもいけないし、一方すべての要素を取り込むことは不可能でもあることである。つまり物理的な現象を再現することは無理であるが、なるべく現象と近くなるようにモデル化をする必要があることを気付かせることだと考えた。学習指導要領の解説にも、「制御や通信の学習と関連付けて」とあり、実際の制御誤差などについても発展できると考えられている。

したがって、本研究は、「現象－要素を取り出しモデル化－シミュレーション－実際の現象の確認」として、学習指導要領の『(3)のA「モデル化とシミュレーション」』と『(4)のAである、制御』を結びつけ、実際の制御誤差などを体験させるための教材の開発を行うことを目的とした。

制御誤差などを体験させるには、上手いかない要素は少ない方が理解しやすい。

コンピュータから操作できるものとしては、それ以前に「マジカルスプーン」と「飛行船」という組み合わせで情報Bの中で制御の授業を行ったことがある。「コンピュータの計測・制御」という単元のみであれば、飛行船を制御するという事は、生徒にとって興味深く、「面白かった」という感想がほとんどであった。一方で、飛行船が3次元の動きであり、またエアコンの風などでも上手く動かないなど不安定要素が大きく、制御誤差を認識するのが生徒にとって難しいという問題があった。また、スプーンを叩いて飛行船を制御するという方法では、制御自体も安定していないので、モデル化の段階での誤差なのか、実際の制御がうまくいかないのかわからないという部分も見受けられた。飛行船に関しては、ドリトルから飛行船を制御するという事を京都情報大学院大学が行っており、「総合的な学習の時間」で実演してもらった。このとき、前述のマジカルスプーンの授業を受講していた生徒が、「信号を安定して送ることができるのでわかりやすかった」という感想を書いている。しかし、「飛行船」自体が持っている、電池の消耗、空調、バランスなども問題も大きく、また数を多く用意することも難しい。このような点から、「モデル化とシミュレーション」という単元と組み合わせるには適当ではないと判断した。<sup>4)</sup>

機械である限り、毎回同じようにはならないが、比較的安定した動きをするということから、制御する機械は車型ロボットと決定した。また、制御するコンピュータのプログラム言語も、より分かりやすいものとして日本語で記述することができるオブジェクト指向言語「ドリトル」を使うこととした。

## 2. 研究方法

実証的研究であり、実際に授業を行いその結果をアンケート等で確認する。

そのため、「モデル化とシミュレーション」の単元における教材を開発し、それに基づいて授業を行う。授業に関しては、生徒等のアンケートで目的が達せられているかどうか、確認をするという手順になる。なお、後述

するが、受講生徒が少ないため、統計的な結果を出すことは無理であり、感想などが中心となる。

本校(大阪府立桃谷高等学校)は、前後期制であり、単位制をとっている。情報Bの講座は平成21年度は後期のみ、22年度は前後期に開講した。それぞれ1講座ずつであるが、この講座で授業を行った。

### 3. 研究内容

#### 3.1 大阪府立桃谷高等学校の特徴と授業について

大阪府立桃谷高等学校は、完全な単位制であり、必修科目以外は、生徒の興味や関心、また進路に応じて、自分で教科・科目を選ぶことが可能である。また、単位の登録数も自由であり、午前や午後だけ登校して年数を掛けて卒業することもできるし、一方、3年で卒業することも可能である。さらに、別の高校からの編入や転入の場合、ほとんどすべての単位を認定している。そのため、中学まであまり学校へ行っていない生徒や、別の高校からの編入や転入生も多い。さらに、社会人を経て入学する生徒もいる。生徒の、学力の幅は広く、中学程度の数学の理解が難しい生徒もいれば、学区の上位クラスの学校から転入した生徒もいる。また、登校が難しい生徒もおり、出席状況も一定ではない。

したがって、この「モデル化とシミュレーション」に限らず、教科書を用いながらも題材に工夫をし、体験を通して学ぶことを多く取り入れた授業を行っている。

単位制であるため、学年により受ける授業は決まっていない。この研究を行った情報Bの授業を選択している生徒の状況はさまざまであり、どのような科目を習ってきたかという共通の前提もない。情報科として、「情報A」を履修したことが望ましいとしているが、その情報Aも本校での履修でない場合もある。さらに、「モデル化とシミュレーション」では必要と思われる、数学や物理(理科総合)などの修得についても履修状況はさまざまである。

授業については、前後期制でなおかつ半期認定を実施しており、2単位の情報の授業は週4時間で行っている。

#### 3.2 使用教材について

##### (1) 車型ロボット'Beauto chaser'

ヴィストン社<sup>5)</sup>が開発した、学校教材用の車型ロボットであり、パソコンとロボット間の命令の転送にUSBポートを利用できるということが、授業では扱いやすいと考え採用した。CPUにH8マイコンを利用している。

本研究を始めた時点では、USBポートを利用できるものとしては、この'Beauto chaser'だけであり、すぐ後に廉価なBeauto Racer<sup>6)</sup>が発売された。しかし、後者はCPUに関する情報が公開されていないため候補から除いた。

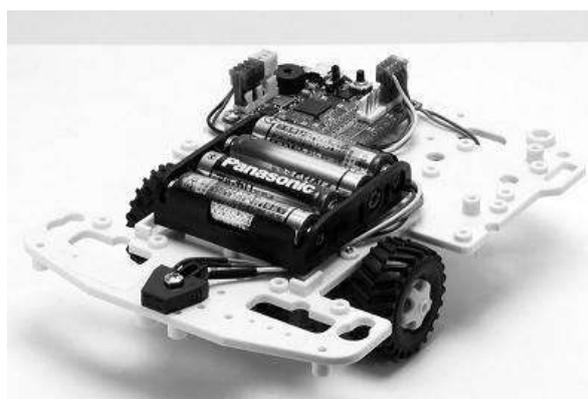


図1 Beauto chaser

パソコンからの制御については付属のプログラムである「Beauto Builder NEO」を用いる。なお、「Beauto Builder NEO」は、タイルプログラムであり、作成したプログラムから命令をつくり、それらをUSBケーブルをつないで、CPUに書き込む。その後、'Beauto chaser'をUSBケーブルから離し、ロボットを動かすことができる

現在では、この'Beauto chaser'の後継機種である'Beauto Rover'が販売されているが、本研究は'Beauto

chaser'で行った授業である。

## (2) プログラミング言語「ドリトル」による制御

コンピュータ上でのシミュレーション画面と現実のロボットを動かす言語は同じであることが望ましい。なぜなら、「なぜうまく動かないのか」といった問題に生徒がぶつかったとき、使用している言語が違くと、解決へのハードルがひとつ増えることになるからである。

'Beauto chaser'に付属している「Beauto ビルダーNEO」ではシミュレーション機能を持っていないため、パソコン上のシミュレーションに「Beauto ビルダーNEO」を利用することは適当でないと考え、プログラミング言語からの制御を用いることとした。

プログラミング言語を用いる場合、ある程度修得している言語か、修得が簡単な言語がよい。'Beauto chaser'はC言語での制御が可能であるから、これを用いることもできるが、C言語に対しての敷居が高いと、制御までは難しい。

本校の情報B では、「モデル化とシミュレーション」の前の章である「アルゴリズム」において、プログラミング言語「ドリトル」(以下「ドリトル」)を用いた授業を行っているので、「ドリトル」を利用することが自然であり、生徒にもわかりやすいと考えられたので、これを用いた。

### 3.3 「モデル化とシミュレーション」の授業

授業としては、いきなり車型ロボットを使うのではなく、簡単な数式モデルなど利用し、「問題をどのようにモデル化するのか」ということを考えさせた。

その後、発展的な問題解決の授業として、ドリトルと車型ロボットを利用した。そのほか、「ドリトル」を利用したアルゴリズムの学習なども事前に行っている。(表1)

表1 授業計画

「ドリトル」を利用したアルゴリズムの学習	6時間
モデル化とシミュレーション 数式モデル等の作成と実行	2時間
ドリトルと車型ロボットを利用した学習	3時間

#### (1) 数学モデルを用いたモデル化とシミュレーションの授業

##### ① 鶴亀算を利用した問題

児童会でおやつに250 円のケーキを買うことにしましたが、40 人分買うには、予算がたりません。そこで、何個かは200 円のクッキーにすることにしました。予算は全部で9000 円です。ケーキとクッキーをそれぞれいくつずつ買うことにすればよいでしょうか？  
ただし、なるべくケーキを多く買うことにします。

この問題を連立方程式で解くのではなく、表計算ソフトを利用し、プログラムで計算することで、予算内のケーキやクッキーの個数を検討できることを理解させる。

また、最後の1 行がない場合、答えは数多くあるが、「なるべくケーキを多く買う」という条件によって、答えを決めることができるということも気づかせたい。つまり、「モデル化とシミュレーション」では、無数に解が出てきた場合、条件によって答えが変わることを理解させるのが重要である。

## ② 線形計画を利用した問題

文化祭で砂糖と小麦粉を使ってカップケーキとクッキーを作って売ることにしました。必要な材料は、カップケーキ 1 個につき 砂糖 9g, 小麦粉 10g、クッキー 1 枚につき 砂糖 6g, 小麦粉 9g、で、カップケーキは1 個10 円、クッキーは1 枚8 円で売ります。当日、うっかり材料をこぼしてしまい、砂糖 200g、小麦粉 250g だけ残りました。さて、残った材料で作ったカップケーキとクッキーが全部売り切れると仮定すると、売上を最大にするには、カップケーキとクッキーをそれぞれ何個つくっておけばいいでしょうか？

こちらも、数式として表現することが可能である、

①と比較すると数学的に解くのは難しいが、表計算やプログラムを使うとこのような問題も比較的簡単に解けるというのを理解するのが目的である。また、実際にカップケーキやクッキーを作って確かめるわけにはいかないなので、このように計算をして確認することでシミュレーションができるという理解にもつながる。



図 2 線形計画のモデル化のプログラム実行画面

## (2) 時間の経過を考えるモデル化とシミュレーション

正月に、ネズミのつがい（夫婦）が現れ、子を12 匹産む。そして親と合わせて14 匹になる。このネズミは、二月に子ネズミがまた子を12 匹ずつ産むため、親と合わせて98 匹になる。このように、月に一度ずつ、親も子も孫もひ孫も月々に12 匹ずつ産むとき、12 ヶ月でどれくらいになるだろうか

時間の経過を利用したシミュレーションということで、この問題を取り上げた。

生存率などを変えることで、12 ヶ月後の予測が変わることも確認できる。この問題もネズミに子どもを産ませて確認するのは、難しいということが大事である。

ここまでの授業の内容は、平成21年度、22年度も共通している。

## 3. 4 「ドリトル」と'Beauto chaser'の通信を使った授業

### (1) 平成21年度の情報Bにおける授業

平成21年度の授業を行う段階では、「ドリトル」からUSBポートを通じて、「Beauto chaser」を制御することは可能であった。しかし、リアルタイムの制御であったため、USBケーブルを長くし、その可動範囲内でしか「Beauto chaser」を動かすことができない状態であった。そのため授業内容は、「モデル化とシミュレーション」というより「コンピュータによる計測・制御」の要素が強く、同じプログラム言語を利用して、画面のタートルを動かすこと、タートルの代わりに「Beauto chaser」を実際に動かすということの比較を行った。

授業で実際に行った課題は次のようなものである。

[問題]

右のように配置されている、チューリップを拾うためには、車をどのように動かせばよいらうか。

車を接続せずに、タートルを動かす場合は次のようなプログラムになる。なお、この授業ではわかりやすいように、タートルの絵を車に変更している。

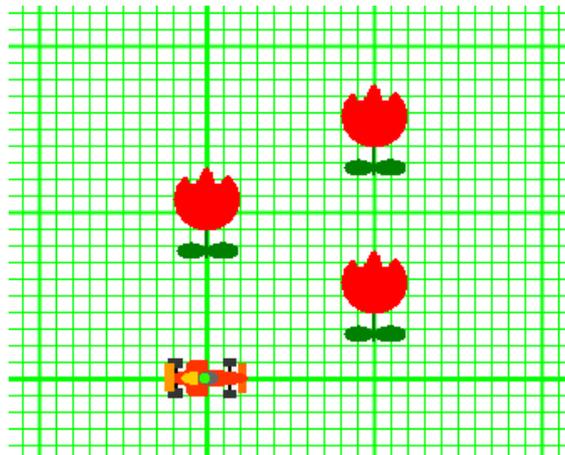


図 3 問題1のプリント図

```
タートル! 作る "tulip.png" 変身する ペンなし 100 50 位置。
タートル! 作る "tulip.png" 変身する ペンなし 0 100 位置。
タートル! 作る "tulip.png" 変身する ペンなし 100 150 位置。
ビュート = タートル! 作る。
ビュート! "raceRed.png" 変身する。
ビュート: 衝突=「|相手| 相手! 消える」。
ビュート! 30 左回り。
ビュート! 110 歩く。
ビュート! 60 左回り。
ビュート! 100 歩く。
ビュート! 120 左回り。
ビュート! 110 歩く。
```

車型ロボットを接続して、制御する通信プログラムは以下の通りである。(一部省略)

```
//=====
// 初期設定
//=====
VSWRC003=シリアルポート! 1024 115200 8 1 "even" 作る "com3" 開く。
時計=タイマー! 作る。
時計! 0.1秒 間隔。
受信文字=フィールド! 作る 300 40 大きさ -390 240 位置。
受信文字! (VSWRC003! (VSWRC003! データ数?) 値?) 書く。
動作完了=0。
尺度=1。 //画面で動き回る範囲に合わせて数値を調整
VSWRC003! (13) 出力。
「動作完了 == 0」!の間「
    (VSWRC003! データ数?) > 0」!なら「
```

受信文字！ (VSWRC003！ (VSWRC003！ データ数?) 値?) 書く。

「((受信文字！読む)！ ((受信文字！読む)！ 長さ?) 1 部分) == ">"！なら「  
動作完了=1。

」実行。

」実行。

」実行。

動作完了=0。

ビュート=タートル！ 作る。

ビュート！ "raceRed.png" 変身する。

VSWRC003！ ("w 04 01") 出力。

VSWRC003！ (13) 出力。

時計！ 10 回数 「ビュート！ "raceRed.png" 変身する。」実行。

時計！ 待つ。

「動作完了 == 0」！の間「

「(VSWRC003！データ数?) > 0」！なら「

受信文字！ (VSWRC003！ (VSWRC003！ データ数?) 値?) 書く。

「((受信文字！読む)！ ((受信文字！読む)！ 長さ?) 1 部分) == ">"！なら「  
動作完了=1。

」実行。

」実行。

」実行。

動作完了=0。

//=====

// 前進 「ビュート！ 秒数 (1秒単位) 前進。」

//=====

ビュート：前進=「 | x |

！ "raceBlue.png" 変身する。

VSWRC003！ ("w 09 7F81") 出力。

VSWRC003！ (13) 出力。

時計！ (x) 時間 「 ! (尺度) 歩く。」実行。

時計！ 待つ。

VSWRC003！ ("w 09 0000") 出力。

VSWRC003！ (13) 出力。

「動作完了 == 0」！の間「

「(VSWRC003！データ数?) > 0」！なら「

受信文字！ (VSWRC003！ (VSWRC003！ データ数?) 値?) 書く。

「((受信文字！読む)！ ((受信文字！読む)！ 長さ?) 1 部分) == ">"！

なら「

```

                動作完了= 1。
            」実行。
        」実行。
    」実行。
    動作完了= 0。
    ! "raceRed.png" 変身する。
}。

    . . . . . 中略 . . . (左回転、右回転などの制御部分)
//=====
//    終了処理 「ビュート! 終了。」
//=====
ビュート: 終了= 「
    受信文字! (VSWRC003! (VSWRC003! データ数?) 値?) 書く。
    VSWRC003! 閉じる。
}。
//=====
//    ここから下が生徒や先生が自作する部分。//
//=====
//各自が作成するプログラム部分
ビュート! 1 左回転。
ビュート! 2 前進。
ビュート! 2 左回転。
ビュート! 2 前進。
ビュート! 4 左回転。
ビュート! 2 前進。
ビュート! 終了。

```

実際に生徒がプログラムするのは、最後の6行である。

この場合の動きは、「2秒左回転し、2秒前進、2秒左回転・・・」となる。

この2つのプログラムにより、コンピュータ上の動きと実際の動きには誤差が生じるという点を体験させることはできた。予想された、床の摩擦や電池の消耗といった要素以外に、USBケーブルがからむといった問題も生じた。

一番の問題は、車型ロボットを制御するプログラムの数値は秒であるのに対し、画面上では座標で動いているということであり、そのため、制御誤差以上に動きの幅が一致しないという問題が生じた。

’Beauto Racer’ という安価な車型ロボットでは、2009年7月にシミュレーション機能が公開されたので、平成22年度の「総合的な学習の時間」において、こちらを用いた授業も行ったが、同じように秒数でカウントする問題が生じた。すなわち、シミュレーション画面では画面内で動きがおさまっているのに、実際に’Beauto Racer’を動かすと、用紙をはみ出すという問題である。ライントレースであれば条件判断で制御をおこなうが、

「前進」や「右回転」「左回転」などの場合、モーターの回転数を落とすというだけでは、対応しにくい。

授業後の感想では、「思ったように動かなかった」と書いているものが多かったが、コンピュータから命令を送りロボットが動くことに興味を持ち「また機会があればやりたい」という回答もあった。「画面では簡単に対象物が消えるが、実際に車と衝突しても空き缶は消えない」と答えた生徒もいた。これは、情報という教科だけでなく、バーチャルとリアルという問題を考える上で大事なことであると思われる。

無線通信モジュールXBeeを利用した無線で動く‘Beauto chaser’も用意した。USBケーブルの接続なしに、コンピュータからの信号をリアルタイムで受け取り動作する。自由に動くので、生徒は喜んでた。しかし、ファームウェアの書き換えを必要とすることと、改造費が1台につき1万近くかかるため、多くの台数を用意することは難しく、これを使った授業の展開は断念した。

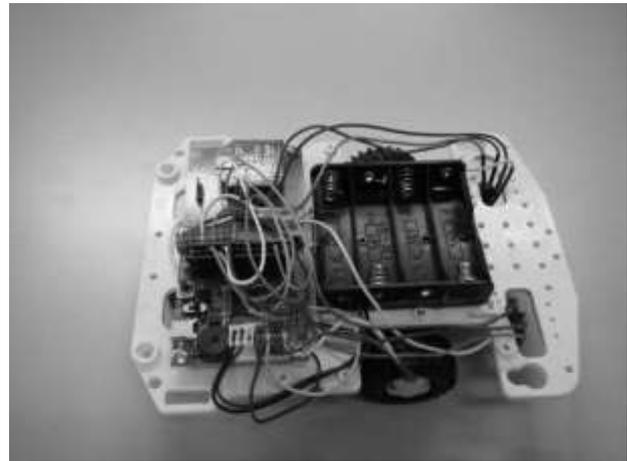


図 4 Xbee で改造した Beauto chaser

## (2) 平成22年度の情報Bの授業

‘Beauto chaser’の授業を「モデル化とシミュレーション」の中に位置づけるには、単にコンピュータを使って動かすだけではなく、‘Beauto chaser’を制御することで問題を解決するという必要性がある。

一方、コンピュータと実際の機械の動きとの差を体験させるには、動きはシンプルでなおかつ命令による動作が近い方がいい。

前年度の問題は、効率よく対象を拾いに行く(衝突させる)という点では、シンプルであるが、動作そのものは、片方は秒で動き、片方は座標の距離となっていたのでわかりにくい。

そこで22年度は、ボタンでタートルを制御する「1時間で学ぶソフトウェアの仕組み」からヒントを得て、プログラム中で動く距離を指定するのではなく、ボタンをクリックする回数により、タートルの動きを指定するようにプログラムを変更した。

① ボタンにより逐次制御でタートルを動かす。

**[問題1] 右の図の画面のチューリップを拾うためには、どのボタンを何回利用すればよいだろうか。効率のよい、回り方を考えなさい。**

チューリップの配置は前年度と同じであるが、ドリトルの画面では、「前」「右」「左」ボタンが用意されている。このボタンをクリックすることで、チューリップを拾う。

第一段階の[問題1]では、ボタンをクリックすると、画面のタートルも一緒に動くようにプログラムを作り、動きを確認させた。

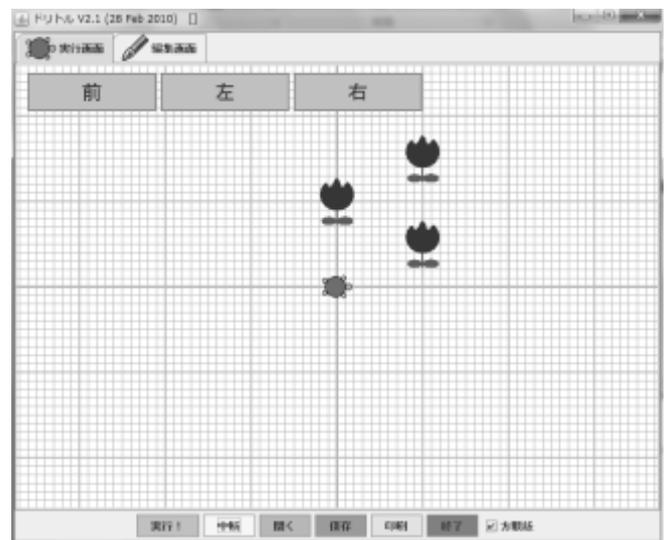


図 5 問題1の実行画面

このプログラムは次のようになる。

```
かめた=タートル！作る。
前ボタン=ボタン！“前”作る。
前ボタン：動作＝「かめた！10 歩く」。
左ボタン=ボタン！“左”作る。
左ボタン：動作＝「かめた！30 左回り」。
右ボタン=ボタン！“右”作る。
右ボタン：動作＝「かめた！30 右回り」。
タートル！作る “tulip.png” 変身する ペンなし 100 50 位置。
タートル！作る “tulip.png” 変身する ペンなし 0 100 位置。
タートル！作る “tulip.png” 変身する ペンなし 100 150 位置。
かめた：衝突＝「|相手| 相手！消える」。
```

このプログラムの2、4、6行目を変更することで、ボタンによるタートルの動きが変わる。

授業では、ボタンの動きを次のように変えて、確認を行った。

- ・ 前ボタン 10 歩く      右ボタン 30右回り      左ボタン 30左回り
- ・ 前ボタン 10 歩く      右ボタン 45右回り      左ボタン 45左回り
- ・ (自分で、設定を行う)

同じ右ボタンであっても、30度と45度のように設定が変わっていることで、解決方法も変わることに、ボタンを設定できることをここでは確認している。

## ② ボタンにより動きを指定し、一括実行

平成22年度の授業を行った時点では、「ドリトル」のプログラムではなく、「ドリトル」を記述しているJAVAプログラムを書きかえることで、USBポートを通じて、CPUへの命令を書きこむことが可能となった。対応している命令は「前進」「右回転」「左回転」のみであるが、これにより、命令をCPUに書きこんだあと、'Beauto chaser'をパソコンから切り離して動かすことができる。

したがって、「ドリトル」を利用したプログラムも、逐次実行ではなく、「開始ボタン」によりタートルが動く一括実行に変更した。これには、「ドリトル」のeval機能を利用している。

[問題2] 画面のチューリップを拾うためには、どのボタンを何回利用すればよいだろうか。考えた後、ボタンをクリックして、実行して確認しなさい。

プログラムは次の通りである。

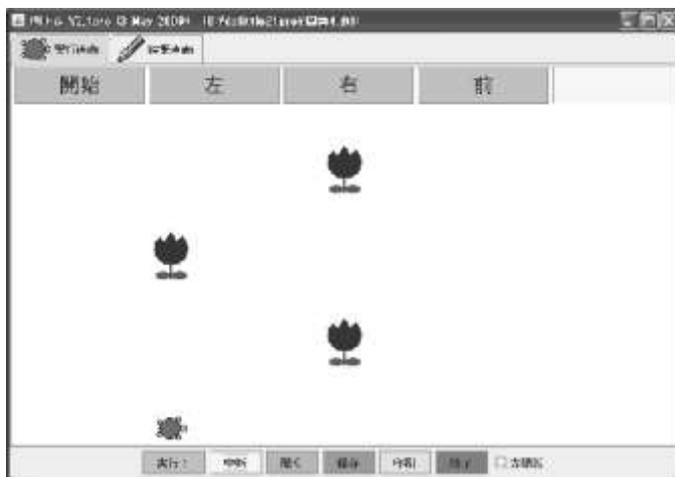


図 6 問題 2 の実行画面

かめた=タイトル!作る ペンなし 0 0 位置。

かめた! ペンあり。

ボタンF=フィールド! 0 作る 50 50 大きさ。

左ボタン=ボタン!“左” 作る 70 50 大きさ。

右ボタン=ボタン!“右” 作る 70 50 大きさ。

前ボタン=ボタン!“前” 作る 70 50 大きさ。

左ボタン:動作=「moji ! ((moji!読む)+“かめた!30 左回り。”) 書く。ボタンF! 増やす。」。

右ボタン:動作=「moji ! ((moji!読む)+“かめた!30 右回り。”) 書く。ボタンF! 増やす。」。

前ボタン:動作=「moji ! ((moji!読む)+“かめた!10 歩く。”) 書く。ボタンF! 増やす。」。

開始ボタン=ボタン!“開始” 作る 100 50 大きさ。

moji = フィールド! 作る "" 書く。

もじれつ=""。

開始ボタン:動作=「もじれつ= (moji!読む)。もじれつ!実行。」。

「命令 = ラベル! 作る。命令! (moji) 書く。」。

宝物たち=配列! 作る。

宝物たち! (タイトル! 作る “tulip.png” 変身する ペンなし 0 100 位置) 書く。

かめた:衝突=「|相手| 相手! 消える。宝物たち! (相手) 消す。「(宝物たち!要素数?) <=0」!なら「ラベル!“クリア!“作る」実行」。

[問題1]と似ているが、「開始ボタン」をクリックするまでタイトルは動かない。

[問題1]の場合、考えることが苦手な生徒が適当にボタンをクリックしているうちに、チュールリップに到達して手順を覚えていないということがあった。しかし、[問題2]は、開始ボタンを押すまで動かないので、必ず手順を考える、すなわち問題を解く必要がでてくる。

授業では、「問題1」で逐次動作を考えタイトルの動きを理解させ、その後、「問題2」による一括での実行を行った。

### (3) 車型ロボット'Beauto chaser'を利用した授業(平成22年度前期)

「モデル化とシミュレーション」だけであれば、(2)の授業だけでもよいが、今回の目的はこのシミュレーションの内容を実際に車型ロボット'Beauto chaser'で行うことにある。

なぜなら、次の2点が大切だと考えるからである。

- ・問題解決においては、その検討した結果を現実の実験で行うことが困難な場合、プログラムを利用して、パソコンなどで、シミュレーションを行う。

- ・パソコンのシミュレーションで行った結果を実装したときに、コンピュータの計算通りに行くとは限らない。

また、うまくいかない原因を考え解決方法を考える必要がある。

このような考えに基づき、チュールリップを空き缶、タイトルを車型ロボットとして、ドリトルから制御を行う授業を行った。なお、'Beauto chaser'については2人で1台割り当てている。(2)の授業とあわせて3時間を割り当てていたが、実際には、復習なども行ったため4時間となった。以下、授業内容について記述する。

① (2)の[問題2]を復習も兼ねて、ボタンの動作(移動距離、移動角度)を変えながら、いくつかの問題を解くとこから始めた。これにより、ボタンの角度や移動距離の変更の仕方について、確認をおこなった。

② 'Beauto chaser'で同じ動きになるようにするために次の手順で授業をした。

・'Beauto chaser'を接続し、それぞれのボタンを1回クリックしたときに動く距離や角度を測定する。ボタンを1回クリックしたのち、図7のようなマス目模造紙の上で、動かし距離を測定した。「ドリトル」での最初のタートルはX軸方向に正の向きに配置されているので、この計測も同じように置かなければいけないのであるが、最初の配置でとまどっていた生徒もいた。

このとき、前進でまっすぐ動かないものについては、モーターの調整を行った。

・この測った値を(2)で利用したプログラムに戻し、ボタンの動作を変更。そののち、画面上でシミュレーションを行った。マス目模造紙の1cmを「ドリトル」の画面の1マスとしている。

・シミュレーションで動きを確認後、マス目模造紙の上で実際の'Beauto chaser'を動かした。



図 7 'Beauto chaser'の制御

動かす手順は次の通りである。

まず、シミュレーションと同じ方法(ボタンをクリックする回数)を、車型ロボット用のドリトルの画面で実行し、プログラムを転送した。

この動きが転送されて書き込まれた車型ロボットを床の上で実際に動かす。

このときの課題は、空き缶を1つだけ取りに行くものと2つ取りに行くものの2種類を用意したが、ほとんどの班が2種類を終えることができていた。

(4)車型ロボット'Beauto chaser'を利用した授業(平成22年度後期)

平成22年度の後期も前期とほとんど同じ授業を行った。

手順を考えて取りに行くということを体験させるため、「アルゴロジック」を取り入れた。

これにより、問題解決のため手順を考えるということが理解できたようであった。

プログラム上変更したのは、次の2か所である。



図 8 'Beauto chaser'の制御画面

'Beauto chaser'を制御するプログラムのボタンの名称が、「実行」となっていたのを「開始」、「前進」を「前」とし順序もシミュレーション画面と一致させた。

シミュレーション画面のチューリップを小さな点にした。これは、チューリップのアイコンはある程度の幅があり、計算上中心をはずれていても衝突したときに消えるためである。より小さな点にすることで、シミュレーションでの計算を少しでも緻密なものにすることができる。

## 4. 研究結果

### 4.1 授業についての考察

#### (1) 生徒の反応

##### ① 平成 21 年度の授業について

授業後の感想は次のようなものである。

- ・同じように動いてうれしかった
- ・画面をみているとコードが絡まって困った
- ・コンピュータやからできることがある。
- ・タイヤの摩擦などがあって実際にはうまくいかない
- ・タイヤの調子が悪かったら動かない
- ・コンピュータではうまく動くけれど、実際にはタイヤの調子が悪かったりすると動かない。
- ・実際には動かすにくい。
- ・行ってほしいところに行かない。タイヤの調子がわるいとうまく動かない。

これらを見ると、画面上で動いても、'Beauto chaser'ではうまくいかないということには理解ができたようである。動かない理由については、タイヤの調子と書いている生徒が多いが、これは、秒と距離の調整がうまくいかず、数字を調整するのに手間取っていたということがある。

##### ③ 平成 22 年度の授業について

平成 22 年度の授業については、3.4の(2)および(4)の終了後アンケートを取った。

前期・後期とも有効回答は 10 枚である。

結果は以下のようになっている。(人数は、前期・後期の順)

Q モデル化に興味を持ちましたか？

- ・興味を持った(3 人、1 人)
- ・わりに興味を持った(5 人、8 人)
- ・あまり興味がない(1 人、1 人)
- ・わからない(1 人)

Q 画面のシミュレーションが実際の車を動かすのに役に立ったと思いますか？

- ・役に立つ(5 人、4 人)
- ・少し役に立つ(4 人、4 人)
- ・あまり役に立たない(1 人、2 人)

Q 画面のシミュレーションと実際に車を動かすのとどちらが面白かったですか？

- ・車(4人、7人)
- ・画面(2人、2人)
- ・両方(4人、1人)

(理由)

画面の方がいい

車が思うようにならない。(前期)

画面は考えたとおりに動く(後期)

車の方がいい

思い通りにならないが、試行錯誤が面白い(後期)

動かすたびに結果が変わるので悔しい半面とても楽しい(後期)

思いもよらない要因で動かないが面白い(後期)

実際に動いているので実感がわく(後期)

両方

画面で確認したことが実際に動くのが楽しい(前期)

Q 画面上でシミュレーションと、実際の車を動かすことで違うことは何だと思いますか？

(前期)

- ・モーターの動きの違いでまがったりしていたから、パソコンとはちょっと違う。(前期)
- ・実際では毎回同じに動かない(前期)
- ・床や電池の影響で動きが変わってしまう。(前期)
- ・画面上のシミュレーションはやり直しがきくけど、実際の車を動かすのはやり直しがきかない(前期)

(後期)

- ・摩擦電池の問題
- ・曲がり方が機械によって違う
- ・角度や距離が実行するたびに少し違う
- ・摩擦の抵抗など細かい部分がシミュレーションされていないので動きが変わったりする
- ・コンピュータはそのとおり動くが、多少だが車は動きが変わる
- ・画面は計算通りに必ず動くけど、実際は電池や絨毯など計算外のことが多い。
- ・予定の角度にまがらないので、誤差の修正が必要。

全体的な感想

思い通りに動かないが意外と白熱していた。

悔しくてとっても向きになってしまいましたが、予想して考えることの楽しさを改めて実感しました。

このような結果から、実際の車を動かすことが、面白いと感じる生徒が多く、興味を引くこと。さらに、画面と同じ動きをすることがより面白いと感じる生徒もいるので、画面上でのモデル化とシミュレーションだけでなく、実際の動きをする車を取り入れたことはよかったといえるのではと思っている。

また、画面上のシミュレーションの違いについても言及した生徒がいたが、これは、実際に存在する車を動かすことで、気がつくことが増えたと考えられる。

さらに、「実際の車を動かすのはやり直しがきかない」という解答は、今回の'Beauto chaser'の課題が空き缶を取りに行こうであるため、つぶれてしまったりすると次がとれない。何回もやりなおしている間に電池が消耗する、本当の車だったら、さらに困るということであった。

予想外だったのは、うまく動かないことが面白いと感じた生徒がいたことである。思い通りにいかないことに興味を感じているということで、やはりこれも実際の車を動かした効果だと考えられる。一人ではあるが、「予想して考えることが楽しい」という回答があった。この回答は「モデル化とシミュレーション」における問題解決の目的に合っている。簡単な例を用いたからこそ、予想が楽しいという回答が得られたのではとと思っている。

なお、鶴亀算等の授業とビュートチェイサーの授業について、どちらが面白いかという質問については以下の解答であった。(なお、片方を受けていない生徒がいるため、回答数は減っている)

- ・両方面白い(4人、3人)
- ・ビュートチェイサーが面白い(2人、4人)
- ・鶴亀算が面白い(0人、1人)

このことから、鶴亀算等を使ったモデル化の授業についても、後に車型ロボットを利用した授業をすることで、理解が深まるとも考えられる。

#### 4. 2 今後の課題

うまくいかない原因を考えさせるときに、車型ロボットの左右の車輪の動きが違ふ、電池が減ってきている等、いくつかの要素があり、これを一つ一つ解決していくのが難しいようであった。

授業では、'Beauto chaser'そのものには深入りさせないように、モーターの制御は教師側で行っていたが、解決する力を身につけるという点では、ロボットそのものを扱う必要もあるかもしれない。

また、「ドリトル」からの制御はできるようになったが、実装している命令の数が少ない。センサーを利用するための命令群などはほとんど未対応であり、センサーを利用したライントレースなどが無理である。

命令数を増やすなどして、より複雑な動きや2台以上の複数台を使ったロボットの制御ができると、車の追い越しであるとか、より現実に近いモデル化とシミュレーションを行うことが可能である。

現実に近いモデル化を行う場合、ライントレースは必要であるか、距離センサーを使った動きを利用するのかなど、どのようなモデル化を考えていくべきかといったことが課題となる。それに伴い、必要な命令を増やしていく必要が生じる。

#### 5 おわりに

高等学校の「モデル化とシミュレーション」は最初にも述べたように、と教えられていない分野であり、また教える自信がないとされているところである。その原因はいろいろあるだろう。

現実にあるシミュレーション「地震シミュレーション」「気象シミュレーション」等については、授業で教える内容としてはかけはなれている。また、教科書の例なども身近ではなく生徒になじみにくいといったこともあげられるだろう。

助成期間は終了したが、車という身近なものを現実に動かしながら、シミュレーションを体験するといった授業がひとつの例となるように、さらに開発をしたいと考えている。

## 参考資料

- 1) 平成 20 年度「高等学校等における情報教育の実態に関する調査」  
財団法人 コンピュータ教育開発センター, <http://www.cec.or.jp/CEC/>
- 2) 岡本敏雄,山極隆,ほか 10 名,最新情報B.実教出版(平成 19 年)  
水越敏行,村井順,ほか 25 名,新・情報B,日本文教出版(平成 20 年) など
- 3) 文部科学省,高等学校学習指導要領解説 (情報編)(2000 年)
- 4) 野部緑,教科「情報」でのマジカルスプーンの実践,2008PC カンファレンス論文集, CIEC(コンピュータ利用教育協議会)(2008 年 8 月)
- 5) ヴィストーン株式会社,<http://www.vstone.co.jp/>
- 5) 稲川孝司,台車型ロボットによる問題解決学習, ICT・Education No.43,日本文教出版(平成 22 年 2 月)
- 6) 久野靖,辰己丈夫,他,「情報科教育法」改訂 2 版,オーム社(2009)
- 7) 野部緑,中田雅士,大島嘉一,ほか 教科「情報」における車型ロボットの利用, 情報処理学会コンピュータと教育研究会(2009 年 12 月)
- 8) 野部緑,ロボットと日本語プログラミングを利用した、情報Bの授業の試み,2010PC カンファレンス(2010 年 8 月)
- 9) 野部緑,モデル化とシミュレーションについての実践～数式モデル、ドリトルを利用した車型ロボットの制御等の実践報告～, 全国高等学校情報教育研究会(2010 年 8 月)
- 9) 野部緑,中田雅士,大島嘉一 ロボットと日本語プログラミングを利用した、情報 B「モデル化とシミュレーション」について, 情報処理学会コンピュータと教育研究会(2010 年 11 月)