

二輪駆動移動ロボットのオフラインプログラミング

日高良和*、篠原毅**

Off-Line Programming of A Two-Wheel Drive Mobile Robot

Yoshikazu HITAKA* Takeshi SHINOHARA**

Abstract:

One of characteristics to robot is according to work for the operation change. However, complex work robot teaching and checking of operation is difficult. Thereat, operating of the robot is programming at off-line and operating of the robot can confirm important. This paper report on development of programming of a two-wheel drive mobile robot on PC screen sort item only playable system.

Keywords: off-line programming, two-wheel drive mobile robot

1. はじめに

ロボットは、その動作をプログラミングによって変更できる柔軟性をもつ機械である。これまで社会で利用されて来たほとんどのロボットは産業用ロボットであり、その操作は専門知識を持つ作業員が行っている。しかし、近年、ペットロボットやアミューズメントロボットなど産業用以外の利用を目的としたロボットが登場している。これら一般向けのロボットはその動作の変更ができないか、変更可能であっても一般人には難しいことが多い。これからのロボット利用分野の拡大を考えると、ロボットプログラミングの簡易化が重要になると思われる。

本レポートは、掃除ロボットや案内ロボットなど一般施設内を移動しながら作業する移動ロボットを想定し、その走行動作プログラムがパソコン上で容易に作成できるオフラインプログラミングシステムについて述べている。

プログラミングの対象とした移動ロボットは、前輪がキャスタ、後輪の二輪が独立に駆動できる三輪型であり、本体前方には外界認識用に赤外線センサを備えている。この移動ロボットは、前進、後進、左旋回、右旋回、停止の5つの行動とロボット前方の障害物の有無が判断できるだけである。そのため、動作のプログラミングは、これらの行動を指定し、条件分岐のためのセンサ情報を付加したモジュールを組み合わせるだけで作成できるようにした。プログラミングの簡易化を図るために、プログラムの記述は開いているウィンドウ画面上に表示された項目をマウスのクリック操作によって選択するだけで行えるようにした。また、ロボットの速度やセンサの感度など、ロボットを動かしてからでない設定が難しい項目は、シミュレーション画面中で調整できるようにしている。今回開発したオフラインプログラミングシステムを使うことによって、ロボット操作の専門的知識がなくても移動ロボットの障害物回避などの複雑な動作のプログラミングも容易に行えた。また、マウス操作だけでプログラミングができるので、短時間でプログラミング方法を習得することができ、プログラミング作業も楽に行うことができた。

(2001年12月4日受理)

*宇部工業高等専門学校電気工学科

**宇部工業高等専門学校専攻科生産システム工学

2. 移動ロボットの構造

移動ロボットは、全長25cm、最大幅10cmの大きさで、マイコン(PIC16F84)を搭載したNEC宮城ビジネス社製の「梵天丸」を想定している。図1はその外観を示しており、キャスター式の前輪と左右独立に駆動できる2つの後輪をもち、車体上にはマイコンや車輪駆動回路などを実装した基盤と乾電池を使用した電源が置かれている。また、ロボット前方に赤外線センサが取り付けられ、進行方向にある障害物の有無を判断できる。赤外線センサは、車体前方の左右に取り付けられた2個の赤外線LEDと前方中央にある受光ユニットからなっている[1]。このセンサは、例えば、受光ユニットが左側の赤外

線LEDから出た信号を感知すればロボットの前方左側に、左右のLEDからのそれを感知すればロボットの前方中央に障害物があると判断できる。また、各赤外線LEDの出力はこれらの後部にある半固定抵抗器によって、受光ユニットの感度はプログラミングによって調整可能となっている。車輪駆動回路への指令信号は、このような赤外線センサの出力などを利用して決定される。各車輪用モータの駆動はモータドライバICによって行われており、ドライバICへの指令によって正転、逆転と停止、および回転速度の変化が可能である。したがって、この移動ロボットの走行動作は、センサによる判断と車輪駆動信号による行動をプログラミングすることによって実現できる。

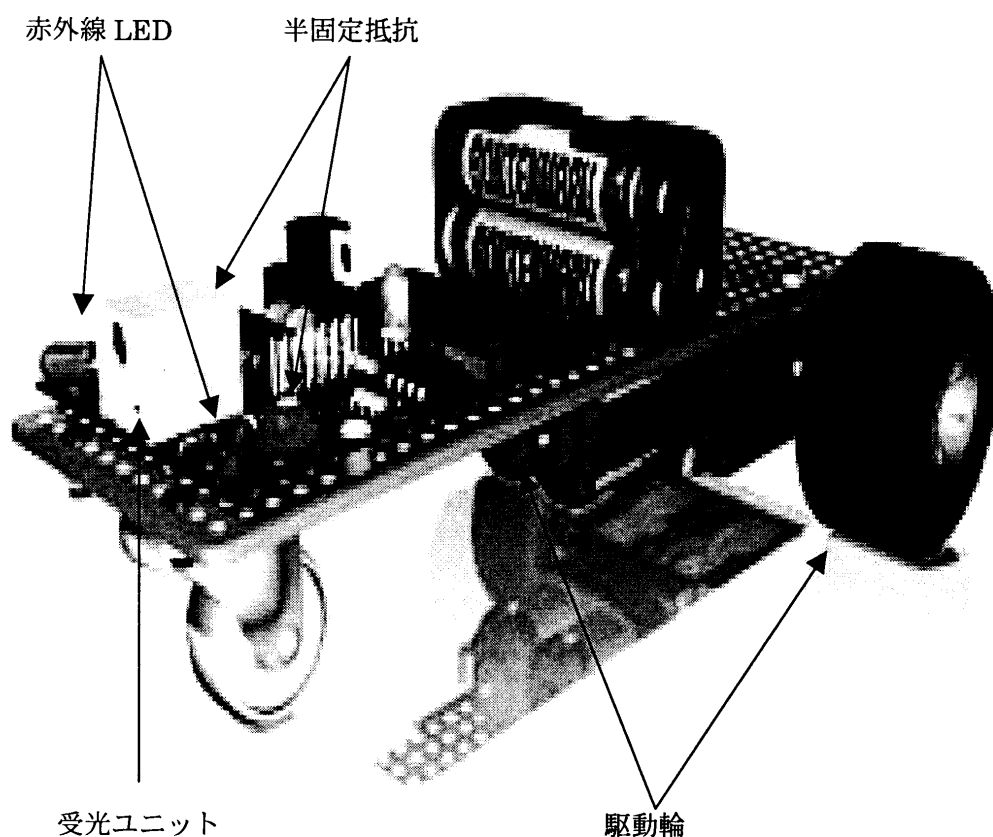


図1 移動ロボット「梵天丸」の外観図

3. オフラインプログラミング法

オフラインプログラミングの方法としては、作業の位置や姿勢などのデータを直接入力する数値入力法、作業環境をグラフィカルに模擬した画面を使用するグラフィック表示法、また、ロボット言語を用いてプログラミングする言語法がある。数値入力法や言語法は組み立てロボットなどの、グラフィック表示法は溶接用ロボットなどのプログラミングに使用されている。このような用途からわかるように、これらのプログラミング方法は作業順序が決まっている場合に有効である。しかしながら、移動ロボットは予期しない走行経路上の障害物への対応などがあり、走行動作のプログラミングは必ずしも順序に従って行っておけばよいと言うものではない。

そこで、移動ロボットのオフラインプログラミング法として、ロボットの走行動作は用意された項目を使ってプログラミングし、走行動作の詳細はシミュレーション画面を見ながら調整することを考える。シミュレーションは、現段階ではロボットの走行動作プログラムを作成するだけなので、ロボットの移動経路を上部から描く二次元のグラフィックによる表現とした。これにより、大きなCPUパワーを必要としない開発環境となったので、プログラミングシステムは一般のDOS-Vパソコンを使用することにした。また、開発プログラミング言語はウィンドウ画面の作成が容易なマイクロソフト社製のVisual-Basicを使用した[2]。開発したプログラミングシステムは、プログラミングの段階によって作業に使うウィンドウ画面を切り替えるようにしている。このシステムによるプログラミング作業の大まかな流れは次のようである。

- a) システムを起動させ「エディタ」画面を開く
- b) 「エディタ」中の作成を選び、「行動選択」画面を開く

- c) 「行動選択」画面でロボットの走行動作をプログラミングする
- d) 「エディタ」中のシミュレートを選び、「シミュレーション」画面を開く
- e) 「シミュレーション」画面でロボットの動作を確認し、速度などを調整してプログラミングを終了する

これらのプログラミング作業はキー入力を選び、ウィンドウ画面に表示されている項目やボタンをマウスでクリックすることなどで進めることができるようにした。

3. 1 「エディタ」画面

システムを起動させると図2のような「エディタ」画面が現れる。エディタ画面は、プログラムの表示部、プログラムの追加作成や新規作成をするときに選ぶ[作成]ボタン、および作成したプログラムの確認を行うときに選ぶ[シミュレート]ボタンがある。

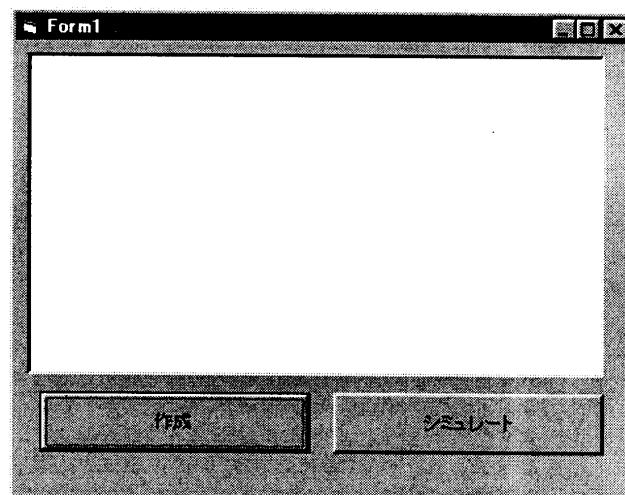


図2 プログラミング起動時に開かれるエディタ画面

それぞれのボタンはクリックされると新たなウィンドウ画面を開き、各作業ができる状態になる。作成中のプログラムは、図3に示すようにプログラムが表示部に書き表される。このシステムでは、

例えば [ロボットが前進していて、前方に障害物
が出現した] のように、移動ロボットの前進など
のひとつの動きと赤外線センサによる障害物の有
無判断の組み合わせをひとつの行動ユニットとし
ている。センサによる障害物の判断は条件分岐と
なるので、実行されている行動ユニットは条件が
成立すると別の行動ユニットへ移すことができる。
そのため、行動ユニットはそれ自身にラベルを付
けられ、条件成立時の行先ラベルの指定が必要と
なる。これらユニット自身のラベルや行先ラベル
は、行動ユニットが作成されたときに、図3の例
のように0番目や1番と「番号」を自動的にエデ
ィタが付けるようになっていく。

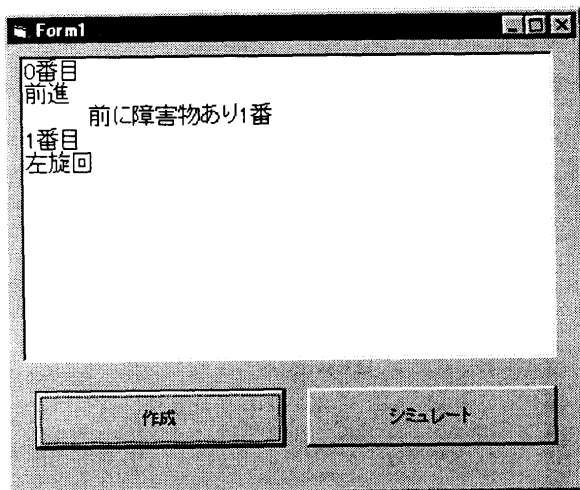


図3 エディタ画面に表示されるプログラムの例

3. 2 「行動選択」画面によるロボットの走行動作プログラミング

この移動ロボットが行える行動は、前進、後進、左右それぞれの旋回、および停止である。また、赤外線センサの出力信号により、ロボット前方と左側、右側の障害物の有無がわかる。このように、行動決定の要素はロボットの走行に必要な最小限の行動とセンサによる判断しか用意していない。そ

のため、プログラミングは、前進などのロボットの動きを決める「行動」の項目と赤外線センサによって障害物の有無を判断する「センサ」の項目を選択することで行えるようにした。

プログラムの作成は、エディタ画面の [作成] ボタンをクリックすると現れる図4のような「行動選択」画面を開いて行う。行動選択画面は、「行動」の項目として前進や後進、左・右それぞれの旋回、および停止の5項目と、「センサ」の項目として障害物がロボットの前方・左側・右側の3方向にそれぞれ有るか無いかの6項目が用意されている。

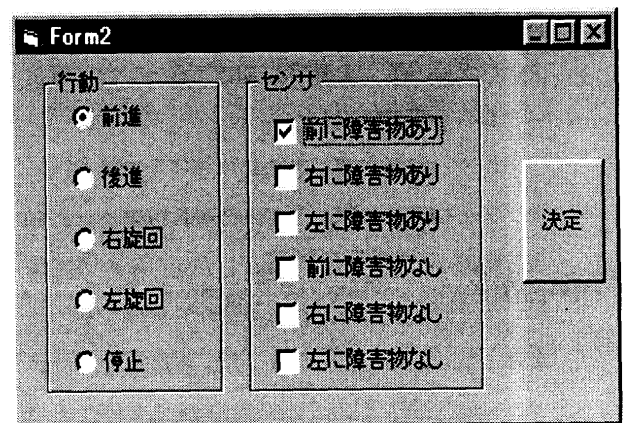


図4 走行動作をプログラミングするための「行動選択」画面

ロボットのプログラミングは、「行動」と「センサ」の項目をマウスで選択しながら、前節で説明した行動ユニットを作成することで行われる。ただし、「センサ」の項目は選択しなくてもひとつの行動ユニットとすることができる。

プログラミングの例として、図3のプログラム表示部に示した [ロボットが前進していて、前方に障害物が出現したとき、ロボットは左まわりに旋回し続ける] という障害物回避のプログラムを作成してみる。この場合、ロボットが前進と左旋回という2つの行動をするので、まず前進についてプログラムを作成する。プログラミングは、ま

ず始めに「行動」の項目から「前進」を選び白丸をクリックして印を付ける。次に「センサ」の項目から「前に障害物あり」を選び、前と同様に白角に印を付ける。項目選択が終了し、ウィンドウ上にある「決定」ボタンをクリックすれば、エディタ画面に図3に示した上から3行までのプログラム [0番目 前進 前に障害物あり1番] が書かれ、前進についての行動ユニットのプログラムができあがる。次にロボットの行動は、ロボットの前に障害物が現れると [1番目] のラベルをもつ行動ユニットへ移る必要がある。そのため、「センサ」の項目を選んだ場合は、選択した項目の数だけ行動選択画面が順番に現れるようになっている。図3の例では、「決定」のボタンをクリックした直後に新たな行動選択画面が現れている。そして、「行動」の項目から「左旋回」を選んだ後、「決定」をクリックして全てのプログラミングを終了している。

3. 3 「シミュレーション」画面による動作確認と補正

移動ロボットの全プログラミングが終了したら、パソコンのディスプレイ上には、再び図3に示したエディタ画面のみが表示される。この画面のプログラム表示部に書かれたプログラムが問題なければ、「シミュレート」ボタンをクリックしてロボットの動作確認をする「シミュレーション」画面を開く。

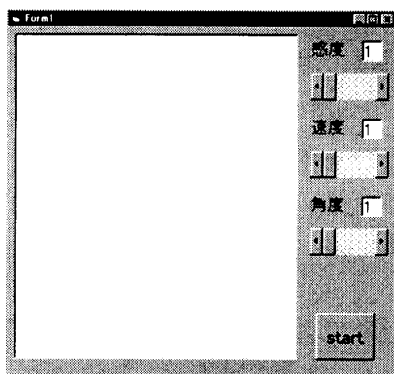
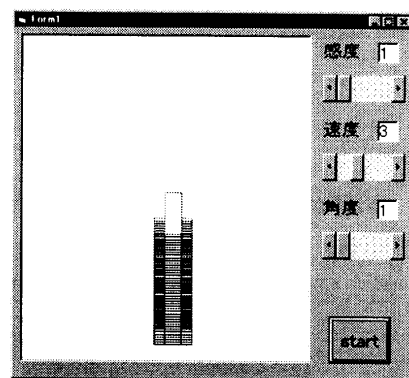


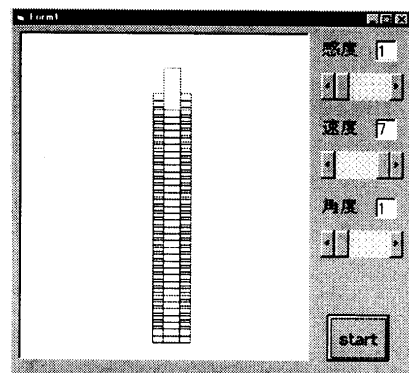
図5 「シミュレーション」画面

シミュレーション画面は、図5に示されるようなもので、移動ロボットの走行の様子を上部から眺めた2次元表現のシミュレーション表示部と赤外線センサの感度、ロボットの走行速度、およびロボットの旋回角度を調整するための各調整バーがある。シミュレーション表示部は、マウスの操作によってロボットのスタート位置の指定と円形をした障害物の設置を画面の任意の場所に行うことができる。

感度は、センサ感度の最大値を1段階から7段階までに分け、「1」が最低感度、「7」が最大感度とした。同様に速度もモータの定格速度値が7となるように分割した。角度は、単位時間内にロボットが旋回する角度を表しており、角度も速度と同様に7段階の設定にした。これら感度、速度、および角度は、それぞれの調整バーをマウスによってドラッグすることで設定ができる。



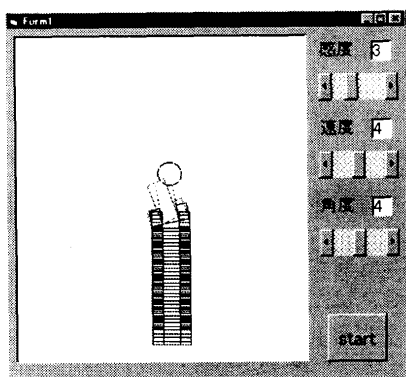
(a) 速度が遅い場合



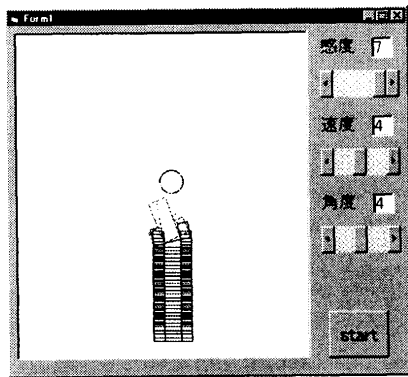
(b) 速度が速い場合

図6 速度を変化させた場合の前進走行シミュレーション

図6は速度を変化させて移動ロボットを前進走行させたときのシミュレーション結果である。図6(a)はロボットの速度を3として、図6(b)は7の定格速度として走行させている。ロボットの描画は、図6(a)、(b)共に10秒毎にロボットの様子を重ね合わせて行っているため、その描画の粗密によってロボットの速度の違いがわかる。



(a) 感度が低い場合



(b) 感度が高い場合

図7 感度を変化させた場合の障害物回避シミュレーション

図7は、3.2節で作成した「ロボットが前進して、前方に障害物が出現したとき、ロボットは左に旋回する」という障害物回避プログラムにおいて感度を変化させた場合のシミュレーション結果である。

図7(a)は感度3と低く設定しているため、障害物の発見が遅れて衝突している。それに対して図7(b)は感度を7と最大にしているため、障害物を遠くから認識して衝突回避ができています。このような調整作業を各動作ユニット毎に行うことで、プログラムを完成させることができる。

4. まとめ

ロボットの利用分野が拡大して行くと共に利用者が増加するため、ロボットのプログラミング作業をいかに効率的に実施するかが問題となる。そのため、二輪駆動型移動ロボットの走行動作プログラムの作成がパソコン上で容易にできるオフラインプログラミングシステムを開発した。このシステムは、ロボットの動作プログラムをウィンドウ画面上の項目選択だけで作成でき、そのプログラムの動作確認と速度などのパラメータ調整をシミュレーションによって実施できるものである。また、その作業もマウスの簡単なクリック動作を使って行えるようにしてある。そのため、パソコンに不慣れなユーザーでも短時間でプログラミング方法を習得できることができた。

5. 参考文献

- [1] NEC 宮城ビジネス “梵天丸 マニュアル”, 1999
- [2] 河西朝雄 “VisualBasic5.0 中級 テクニック編” 技術評論社, 1998