LEGOロボットによる協調性の考察

2002MM025 市川　晃
指導教員　陳　幹

1　はじめに

1.1　背景

近年、様々なロボットが作られ、2 足歩行するものも開発され、生活にロボットというものが身近になってきている。しかし、ロボットの作製はプログラミングが難しく、難解な機械が多い為、一部の人にしか作ることができなかった。そこで、容易にロボット作りを体験できる『LEGO MINDSTORMS』を使い、過去に前例が少ない LEGO ロボットの協調性について研究を始めた。

1.2　アプローチ

本研究では、Master Slave の関係を持たせ、協調性を持たせる。低コストかつ簡単なプログラミングにより、高度な自立性ロボットを作り、ロボットの協調動作の確立を目指していく。RCX 間の赤外線通信によりプログラムが正しく実行されているかを検証する。

2　LEGO ロボットについて

LEGO MINDSTORMS とは 1988 年に LEGO 社とマサチューセッツ工科大学とが共同開発した LEGO ブロックキットの 1 種である。特徴は 3 つある。1 つ目は工具を使わずに、LEGO ブロックを使って、ボディを作ることが可能であること。2 つ目は車体部（RCX) にマイクロコンピュータを内臓しておりコントロールプログラムを自由に作成することができること。3 つ目は RCX には 3 入力（センサ類）・3 出力（モータ駆動など）があり、様々な動きをさせることができることである [2]。

3　プログラミング

3.1　ロボットのプログラミング

基本的には、パソコンで作成したプログラムを赤外線トランスミッタで RCX に転送（ダウンロード）する方式をとっている。LEGO ロボットを動かすためのプログラム開発環境には様々な種類があり、それぞれに対してプログラム言語仕様が異なる。代表的なものとしては次の 3 つが知られている [3]。

1. ROBOTICS INVENTION SYSTEM 2.0 (RIS2.0)
   LEGO MINDSTORMS に付属しているシステムで、命令アイコンをマウスで配置しビジュアルにプログラミングできる。Windows で動作する。
2. NOT Quite C (NQC)
   C 言語に似た文法を持つテキストベースのプログラミング言語で Windows、Mac、Linux で動作する。
3. ROBOLAB
   RIS2.0 同様、命令アイコンを順に配置してビジュアルにマウスだけでプログラミングできる。Windous,Mac で動作する。[1]

4　ロボット間の通信

協調動作を行なう上で、ロボット間の通信が必要となる。RCX は赤外線ポートで PC および他のロボットと交信できる。この特徴を用いれば共同作業ロボットなどの作成が可能になる。

2 つ以上のロボットを使用するとき、1 つをリーダーにするのが通常である。ここではこのロボットをマスタと呼ぶ。他方のロボットをスレーブにする。マスタが命令を出し、スレーブがそれに従う。場合によっては、センサーの値のような情報をスレーブからマスタへ転送することもできる。この場合、マスタ用およびスレーブ用の 2 つのプログラムが必要になる。プログラムの例を以下に示す [4]。

```
task main() // SLAVE
{
while(true)
{
ClearMessage();
until () != 0);
if (Message() == 1){OnFwd(OUT_A+OUT_C);};
if (Message() == 2){OnRev(OUT_A+OUT_C);};
if (Message() == 3){Off(OUT_A+OUT_C);};
}
}
```

マスタのプログラムの方は簡単である。マスタが命令の送信後 2 秒待ち、次の命令を送信する。そのプログラムを以下に示す。

```
task main() // MASTER
{
SendMessage(1) ;wait(200) ;
SendMessage(2) ;wait(200) ;
SendMessage(3) ;
}
```

5　ロボットの制動距離測定

モーターパワー値に対応したロボットの制動距離を測定した。各 10 回測定を行なった平均値を示している。

表 1 より、ズレを少なくするため、モータパワー値は 2 以下に設定した。

表 1　　ロボットの制動距離特性

モーターパワー値	制動距離
パワー 5	7.84
パワー 4	4.07
パワー 3	3.45
パワー 2	1.51

(単位：cm)

6　協調動作

本研究における目的は 2 つある。1 つはロボットの「自律性」である。知的な動作を行なうロボットには、この「自律性」は欠かすことができない重要な要素である。しかし、いかに高度な自律性を有するロボットがあろうとも、1 台では限界がある。もちろん複数台で協調して作業を行なったほうが、より効率的な作業を行なうことが考えられる。そこで、本研究で「自律性」と共に重要視していることが、ロボットの「協調動作」の確立である。複数台が自律的に協調し合い、1 台のロボットではなし得なかった動作を行なうことが可能となれば、より効率的になる。これはシステムを向上させる上で重要な要素の 1 つであると考えられる。

6.1　協調動作を行う上での問題点

協調動作を確立する条件のもとで次のような問題が浮上する。

1. RCX の通信に限界があるため、任意の位置、対象物により通信が不可能になる。そのために協調動作も不可能になる。
2. PC から複数のロボットへプログラムをダウンロードしようとして失敗する可能性がある。
3. 複数のロボットが同時にメッセージを送信するとその情報が失われる危険性がある。

6.2　問題点を考慮した対応策

それぞれの問題を考慮し、次のような対応策を考えた。

1. RCX 間の通信距離は短いので、遠くに送信を行なうために、SetTxPower(Tx_POWER_HI) の命令を使う。
2. ダウンロードは PC—ロボット (1 台) の 1 組で行なうものなので、プログラムのダウンロードを行なうときは、電源オンのロボットを 1 つにしておく。
3. プログラムにおいて通信 (メッセージ) を受け取ったらそれに対応する返事を返すようにする。このとき送信は 1 つのロボットのみにする。

7　自律移動実験

7.1　実験概要

「自律性」と「協調動作」の確立を示すために LEGO ロボット 2 台で対象物を運ぶ自律移動実験を行なった。全体の流れとしては以下の通りである。LEGO ロボットが対象物の方向へ向かう。LEGO ロボットがタッチセンサに触れて察知したとき対象物を掴む。対象物を移動させる。一度対象物を放してから対象物を持ち換えることによって方向転換を行なう。対象物を移動させる。目標地点で対象物を放す。

7.2　実験結果

自律移動実験を行なう前にタスク毎に分けて動作の確認をした。

1. 一度対象物を置いてから方向転換を行なう方法を適用した。曲がる方向にばらつきがあり、プログラムを見直し精度を高めた。
2. LEGO ロボット同士が動きを合わせて動くことができるか検証した。メッセージを送信する、メッセージを受ける時間に若干ズレがあり、待機時間を調節した。

実験結果は以下の通りである。

1. ロボット間の動作の同調に度々ズレが生じる。
2. 対象物を掴むことができた。
3. 方向転換を行なう際、始動からズレが生じている。目標は 90 ° 方向転換を行なうことだが実験結果では、約 45 ° から約 90 ° 未満と目標の角度へ方向転換を行なうことができなかった。
4. 実機 (LEGO ロボット) 自体にお互いの性能の違いが生じた。

一連の動作は動かすことができたが､様々な問題が残った。

8　おわりに

今後に課題が残るが簡易なロボットを作製することにより高度なロボットと比較し、協調性のあるロボットが作製できたことに満足度の高い成果が得られた。

参考文献

[1] ROBOLAB スタートガイド,LEGO DACTA A/S(2000).

[2] MINDSTORMS 情報局, http://www.mi-ra-i.com/JinSato/MindStorms/ .

[3] Jin Sato：LEGO MindStorms 鉄人テクニック, 総合科学技術出版　株式会社オーム社 (2000).

[4] Mark Overmar ：NQC を使った LEGO ロボットのプログラミング, 英語版 3.03 (1999.10.2). http://ns1.hep.scitec.kobe-u.ac.jp/~kurasige/lectures/Robotics/Manual/NQCTu2v1.pdf.