きまぐれロボット

— ロボット知能をプログラミング —

2019 年 6 月 27 日

概 要

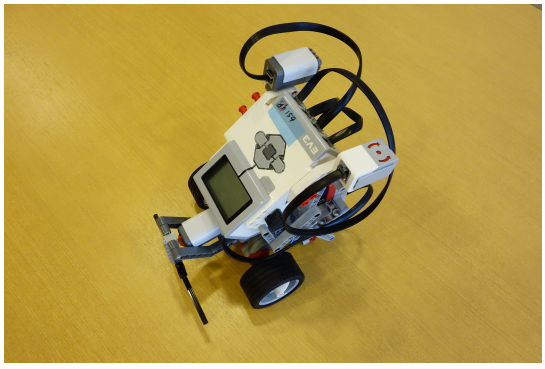
このセミナーでは，初歩的なロボット知能をゼロからプログラミングします．ロボットの身体（筐体）はすでに組み立ててあります．超音波センサーやタッチセンサーなどのさまざまなセンサーをつかって障害物を感知し，それを自律的に回避しながら進むことを目標にします．

決められた動きをするプログラムを使うと，障害物の配置が変わったときに，ロボットはそれをうまく回避できないかもしれません．確率的に，方向を変えながら進むロボット，つまり「きまぐれロボット」に，うまく障害物を回避させるプログラムを作ってみよう．

1 はじめに

このセミナーでは，LEGO MINDSTORMS という小さな標準ロボット（図 1(a) 参照）をコンピューター (PC) からプログラミングすることで，動かします．このロボットを以下では省略して LEGO ロボットと呼ぶことにします．

ロボットというと，アトムやアシモなどのようなヒューマノイド型のロボットがすぐに思い浮かぶかもしれません．アトムやアシモの身体（筐体）はハードウェア．いっぽう，その頭脳や神経系といえる部分がソフトウェアです．それをロボットの「知能」と言い換えることもできます．具体的には図 1(b) に示されたような，プログラミング言語で記述された**ソースコード，すなわちソフトウェアがロボットの「知能」にあたります．**これらのハードウェアとソフトウェアの両者



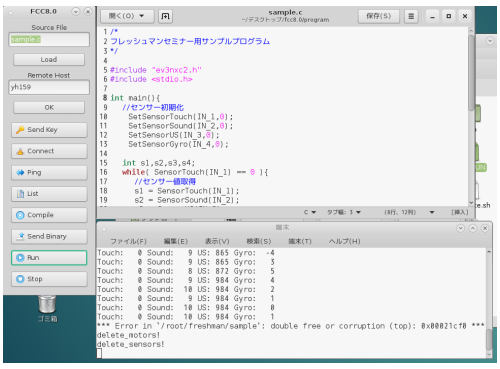


図 1: (a)LEGO ロボット外観　　(b) 統合開発環境

が調和して機能することによってはじめてロボットの自律的な行動が可能となります．このセミナーではハードウェアの構成には変更を加えず，ソフトウェア（アルゴリズムやプログラム）を工夫することのよって，自律的障害物回避を行います．

2 LEGO ロボット

LEGO ロボットには定められた形はなく，パーツをブロックの様に組み合わせることで，さまざまな形のロボットを作ることができます．３つのモーターが動きを生み出します．その他に超

音波センサー (US)，ジャイロセンサー (Gyro)，タッチセンサー (Touch)，音センサー (Sound) の 4 つのセンサーを使って，環境を感じることができます．

このセミナーでは，2 つのモーターと 4 つのセンサーを使って，標準的なロボットをすでに準備してありますので，**自分たちで組み立てる必要はありません．**

LEGO ロボットはモーターとセンサー以外に小さなコンピューター (EV3) を持っています．ロボットの頭脳部分です．EV3 にはモーターを接続するインーターフェイスが 4 つ (A,B,C,D)，センサーを接続するインターフェイスが 4 つ (1,2,3,4) あります．それらを通じてモーターを制御したり，センサーから情報を受け取ったりします．また， 無線 LAN インターフェイスを使って後述するプログラムを他のコンピューター (PC) から受け取ります（図 2 参照）．

このセミナーでは，**プログラム自体は，PC で製作し，それを LEGO ロボットに搭載された EV3 に転送**して実行することによって，ロボットに「知能」を与えます．

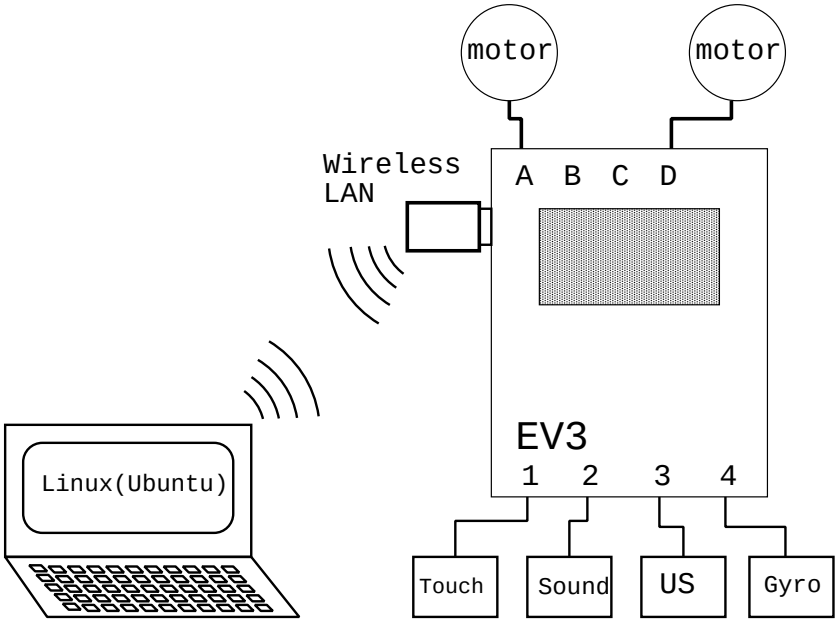


図 2: EV3 にはモーターをつなぐインターフェイス (A,B,C,D) と 4 種類のセンサーをつなぐインターフェイス (1,2,3,4) がある．PC と EV3 は無線 LAN で接続されている．

3 アルゴリズム

ロボットの知能にあたるプログラムはどのような構造をしているのか，直接プログラミング言語で記述するまえに，大まかな流れ（アルゴリズム）を設計します．アルゴリズムの記述には PAD 図 (Problem Analysis Diagram) を使います．図 3 に PAD 図で使われる構成要素を例示しました．

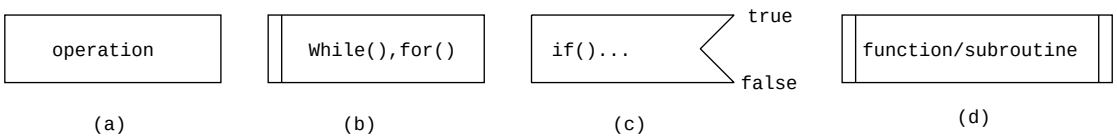


図 3: PAD の構成要素．(a) 命令，(b) 繰り返し，(c) 条件分岐

図 3(a) は**命令**，つまり代入や四則演算などを表します．図 3(b) は**繰り返し**を表します．図 3(c)

は**条件分岐**を表します．図 3(d) は**関数**を表します．つまりセンサーのセットやモーターの駆動などを表します．

例を使って説明します．二つのモーター AD を回して 1 秒間前進し，その後，モーター AD を反転させて 1 秒間後退させるアルゴリズムを，図 4(a) に示しました．このアルゴリズムが実行さ

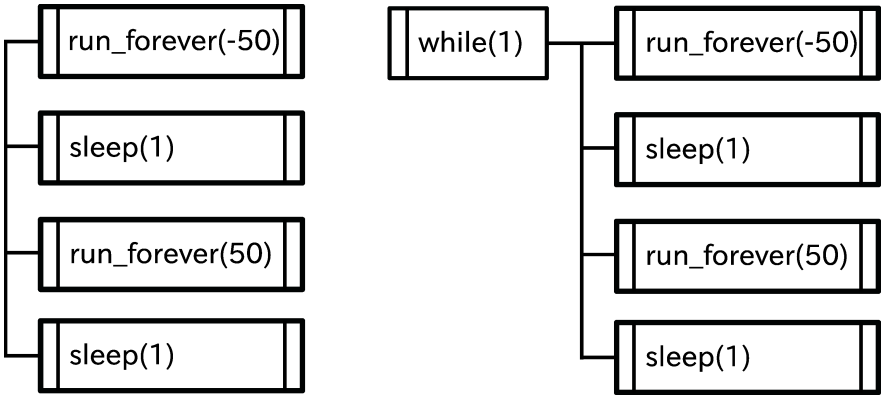


図 4: (a)1 秒前進して 1 秒後退するためのアルゴリズム． (b)1 秒前進して 1 秒後退を無限に繰り返すためのアルゴリズム

れると，合計 2 秒間モーターが動作してすべて終了します．いっぽう，図 4(b) に，1 秒前進，1 秒後退を無限に**繰り返す**アルゴリズムを示しました．このアルゴリズムが実行されると，強制的にプログラムが終了されるまで，LEGO ロボットは動きつづけます．

4 LEGO ロボットを動かす Python

前節で示したアルゴリズムを実際に Python のプログラムにすることによって，LEGO ロボットを動かしてみます．図 5 に図 4(a) のアルゴリズムに対応するプログラムを示しました．1 ～ 2

```
1  import ev3dev.ev3 as ev3 # モジュールをインポートする ev3dev
2  import time
3
4  mA = ev3.Motor('outA') # オブジェクトを生成する Motor
5  mD = ev3.Motor('outD')
6  mA.run_forever(duty_cycle_sp=-50) # 出力%で回転させるメソッドを実行する 50
7  mD.run_forever(duty_cycle_sp=-50) # 出力%で回転させるメソッドを実行する 50
8  time.sleep(1)
9  mA.run_forever(duty_cycle_sp=50)
10 mD.run_forever(duty_cycle_sp=50)
11 time.sleep(1)
12
13 mA.stop()
14 mD.stop()
```

図 5: 1 秒間前進，1 秒間後退のプログラム．図 4(a) のアルゴリズムに対応

行目は必要なモジュールの読み込みです．

`run_forever` メソッドの引数にある 50 という数字は，モーターの出力を 50%にすることを指定しています．−50 はモーターが 50%の出力で逆回転することを意味します．`time.sleep(1)` は１秒間その状態を保つことを意味します．

つぎに，図 6 に図 4(b) のアルゴリズムに対応するプログラムを示しました．while 文によって図 4(a) の内容を繰り返します．while 文は while(条件):という形をしており，条件が成り立つ (ture

```
1  import ev3dev.ev3 as ev3 # モジュールをインポートする ev3dev
2  import time
3
4  mA = ev3.Motor('outA') # オブジェクトを生成する Motor
5  mD = ev3.Motor('outD')
6
7  while(1):
8      try: # 例外処理をおこなうために必要
9          mA.run_forever(duty_cycle_sp=-50) # 出力 50 % で回転させるメソッドを実行する
10         mD.run_forever(duty_cycle_sp=-50)
11         time.sleep(1)
12         mA.run_forever(duty_cycle_sp=50)
13         mD.run_forever(duty_cycle_sp=50)
14         time.sleep(1)
15     except KeyboardInterrupt: # Ctrl-が押された時実行される c
16         mA.stop()
17         mD.stop()
18         break
19
20 mA.stop()
21 mD.stop()
```

図 6: 1 秒間前進，1 秒間後退を繰り返すプログラム．図 4(b) のアルゴリズムに対応

or 1) 限りブロックの内容が繰り返されます．ブロックとは，while 文の下にあるインデント（字さげ）された部分の 6 行のことです．

Python では，ブロック（処理のかたまり）をインデントで表します．つまりインデントがずれていたりすると，文法エラー (syntax error) がおこりますので注意して下さい．

この例の場合，つねに true(=1) なので，無限に繰り返されます．プログラムを強制的に終了するには，ストップボタン (■) を押して下さい．CTRL-C を押すのと同じです．図 6 の例では，例外処理 (except) として stop メソッドを実行しています．

5　センサーで感じる

ここまでのプログラムでは，LEGO ロボットは一定の決まった動きだけをしました．センサーによって，障害物や光を感じることによって，状況に応じた動きが可能となります．そのために条件文 (if 文) によって，センサーの感知した情報を判断します．

5.1　センサーを有効にする

四種類のセンサーをつかえる状態にするための関数を図 7 に示ます．これらのオブジェクトをプログラムの最初の部分で一度生成すれば，その後値を読むたびに生成する必要はありません．

#で始まる行はコメント行です．省略してもプログラムは正常に動作します．

5.2　タッチセンサーからの出力

タッチセンサーからの出力は value メソッド，

$$\text{ts.value()} = \begin{cases} 0 & (\text{タッチセンサーにものが触れていないとき}) \\ 1 & (\text{タッチセンサーにものが触れているとき}) \end{cases} \tag{1}$$

```
ts = ev3.TouchSensor('in1')
    # タッチセンサーを1番のインターフェイスに.
p = ev3.LegoPort('in2')
p.set_device='lego-nxt-sound'
ss = ev3.SoundSensor('in2')
    # 音センサーを2番のインターフェイスに
us = ev3.UltrasonicSensor('in3')
    # 超音波センサーを3番のインターフェイスに. 0：長さの単位(mm)
gs = ev3.GyroSensor('in4')
gs.mode='GYRO-G&A'
    # ジャイロセンサーを4番のインターフェイスに.
```

図 7: 4 種類のセンサーを有効にする関数

を用いて得ることができます．図 6 の例で，while(1): の部分を while(ts.value()==0): に変えれば，タッチセンサーに物が触れていない間は，前後運動をつづけ，タッチセンサーに物が触れると while による繰り返しループが終了します．

5.3 超音波センサーからの出力

超音波センサーからの出力は us.value() で知ることができます．us.value() の値は，超音波センサーから物体までの距離を mm 単位で表しています．たとえば，図 8 は，超音波センサーから障害物が 30cm 以内に近づいたら，モータ A,D を反転させて，1 秒間バックするためのコードです．

```
    if us.value()<300:
        mA.run_forever(duty_cycle_sp=−50)
        mD.run_forever(duty_cycle_sp=−50)
```

図 8: 超音波センサーからの出力で条件分岐する

5.4 音センサーからの出力

音センサーからの出力は図 9 のように利用することができます．if 文の中の 300 は音の強さの

```
    sound=1000−ss.value()
    if sound>300:
      ...
```

図 9: 音センサーからの出力で条件分岐する

閾値を表しています．値を変えて，ロボットの反応の変化を観察しましょう．

5.5 ジャイロセンサー

ジャイロセンサーの値は，最初に電源を投入したときの角度がゼロに初期化されています．

```
    ...
    angl=gs.value(0) # 角度の値
    vel=gs.value(1) # 角速度の値
    ...
```

図 10: ジャイロセンサーの値を読み込む例

5.6 値を記憶する（変数）

ここまでは，センサーから帰ってくる値をそのまま利用しました．その値は，今現在の環境から得られる値です．たとえばジャイロセンサーの場合，gs.value(0) の値は，今現在の角度を意味します．過去の角度を記憶しておきたい場合どうすれば良いでしょうか？

過去の角度を記憶しておいて，現在の角度との差を利用して行動することなどが可能です．変数にセンサーからの値を記憶することによってそれが実現できます．たとえば，ange という変数を使うとすると，例えば図 10 のような使い方が可能です．

例えば，ロボットの回転角度を指定したい場合には，変数にジャイロセンサーの出力値を初期値として記憶しておき，そこからの増加分を使って条件分岐などを行う必要があります．

6 確率的な動き

確率的なロボットの動きは，乱数を利用することで，作り出すことができます．random() という関数は，0〜1 までの数値をランダムに返します．これを利用して，確率 1/2 で，右回転と左回転をするプログラムの例を以下に示します．

```
import random as rdm

if rdm.random()>0.5:
    mA.run_forever(duty_cycle_sp=50)
    mD.run_forever(duty_cycle_sp=−50)
    time.sleep(0.5)
else:
    mA.run_forever(duty_cycle_sp=−50)
    mD.run_forever(duty_cycle_sp=50)
    time.sleep(0.5)
```

図 11: 確率的に方向を変える

7 課題

超音波センサーなどからの入力を利用することによって，LEGO ロボットが障害物を回避しながら運動するためのプログラムを作成し，実際に LEGO ロボットを動かしてみよう．

7.1 きまぐれロボットアルゴリズム

超音波センサーの前に 300mm の距離以下に障害物が感知された場合に，確率 1/2 で右あるいは左にランダムに方向転換するアルゴリズムの PAD 図を図 12 に示した．

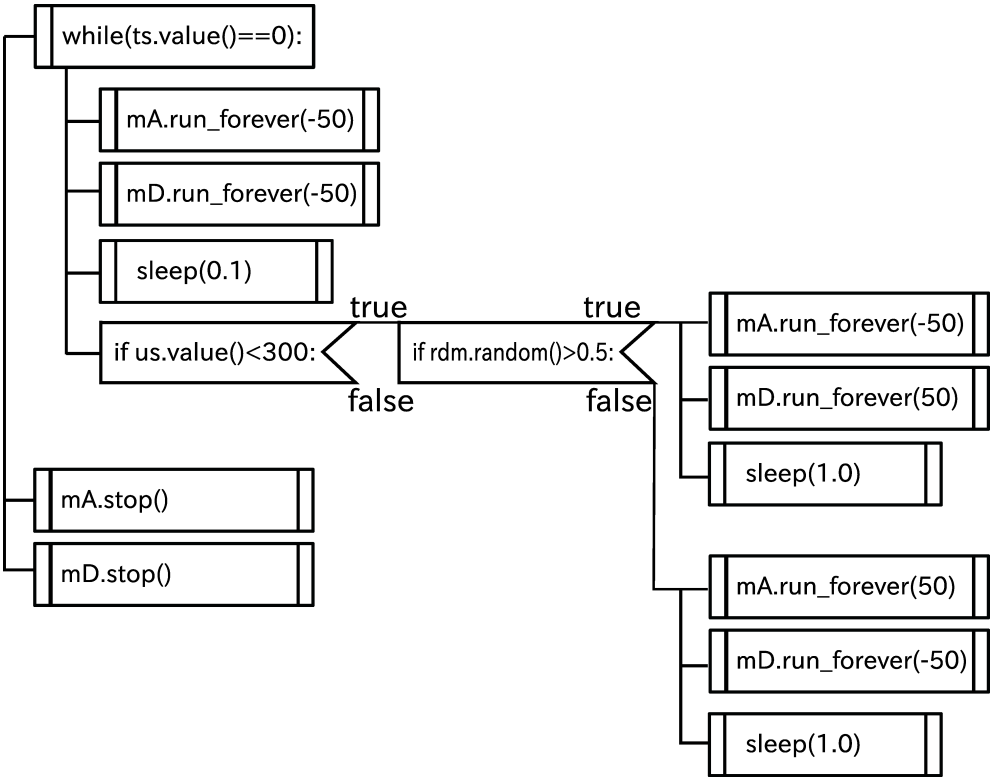


図 12: 確率 1/2 で左右に曲がる気まぐれロボットのアルゴリズム

8　さらに知りたい人は

毎週金曜日 14:30 から VR シアターにて「きまぐれロボットゼミ」を行います．興味がある方，この授業の予習復習したい方は自由に集まってください．

また，本実習に関連する web ページを下記に示しますので，ぜひ参考にして下さい．

`https://github.com/HondaLab/Robot-Intelligence/wiki/python-ev3dev`

図 13: python-ev3dev でロボット知能をプログラミング