赤外線測距センサを用いた車両ロボットの自動走行制御

2013SE242 山田啓人
指導教員：大石泰章

1　はじめに

近年，交通事故や交通渋滞を解消するために，自動車の自動走行制御が盛んに行われている．高速道路の単一車線での，巡航走行時のアクセル，ブレーキ，ステアリングを自動制御するシステムを搭載した自動車が発売されるまでに至っている．

本研究では，見通しの悪い小道での自動走行を実現するために，小型車両 Zumo を用いて障害物の検知と，その後の自動走行を行わせる．Zumo とは，マイクロコンピュータ Arduino によって制御可能な 10cm×10cm 未満の小型の実験機である [1]．左右 2 つのモーターにより，キャタピラを独立して動かすことができる．初期状態の Zumo では周囲環境を把握するセンサがないため，赤外線測距センサを新たに取り付ける．前方に 2 つ，後方に 2 つ配置することにより，実験機の 4 方向の環境を知ることができる．これを用いて，前方の障害物の検知と，検知後に障害物に沿う自動走行を行うことを目標とする．

2　初期状態の実験機

図 1　初期状態の Zumo

まず，既製品の Zumo について説明する．Zumo の概観を図 1 に示す．Zumo の大きさは 10cm×10cm×5cm 未満である．単 4 電池 4 本により電力供給されており，モータにより，左右 2 本のキャタピラを独立して動かすことができる．左右のキャタピラをともに正転させることで前進，逆転させることで後退ができる．右のキャタピラだけを正転させると左に曲がり，左のキャタピラだけを正転させると右に曲がる．Zumo には，単音を流せるブザー，3 軸加速度センサ，3 軸磁場センサ，3 軸デジタルジャイロセンサ，そして車体下部にはラインやエッジ検出のための赤外線センサが初期搭載されている．制御部にはマイクロコンピュータ「Arduino Uno」を用いた．以下，Arduino と表記する．Arduino とは，AVR マイコン，入出力ポートを備えた基盤であり，C 言語風の Arduino 言語とそれの統合開発環境が用意されている．

3　測距センサの取り付け

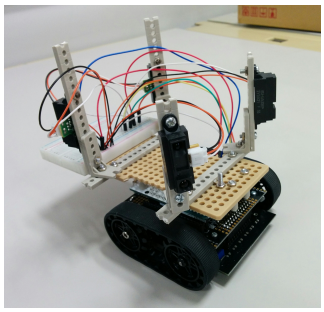

図 2　赤外線測距センサを取り付けた Zumo

赤外線測距センサを取り付けた小型車両 Zumo の概観を図 2 に示す．初期状態の Zumo には Zumo Shield という基盤があり，今回の研究では Zumo Shield にソケットを増設した．これにより，A0〜A5，GND，5V などのピンが使えるようになった．A0〜A4 ピンはそれぞれ赤外線測距センサ（SHARP 製，2Y0A21F) とつながっている．測距センサの接続方法は文献 [3] の 100〜104 ページを参考にした．その他，ユニバーサルプレートセット（TAMIYA 製），ユニバーサルアームセット（TAMIYA 製）を使い，測距センサの位置を固定している．

4　センサの性能

本研究で使う測距センサの特性を調べた．測距センサは出力電圧と距離との関係がほぼ反比例になるよう設計されている．文献 [4] より，出力電圧 x[V] と距離 y[cm] の関係は次のように表される：

$$y = \frac{a}{x+b} + c. \tag{1}$$

ここで，a, b, c は定数である，実際に出力電圧と距離のデータを取り，最小二乗問題を解いたところ，a, b, c の各値は次のようになった：

$$a = 6.10 \times 10^3, \tag{2}$$
$$b = -17.4, \tag{3}$$
$$c = -3.66. \tag{4}$$

また，測距センサには雑音が多く混ざっており，そのままの値を制御に利用するのは難しい．そこで，今回は式 (5) のローパスフィルタを利用した：

$$z(n) = (1-k)s(n) + kz(n-1). \tag{5}$$

k は忘却係数であり，今回は $k = 0.7$ とした．また，n はサンプル時間 28.5[ms] における離散時間を，$s(n)$ は時

間 n において測距センサが読み取った値を，$z(n)$ は時間 n でのローパスフィルタの出力をそれぞれ指している．このローパスフィルタのカットオフ周波数は約 2Hz である．

5　障害物に沿う自動走行制御　—ON-OFF 制御—

障害物に沿う走行を実現させるために，センサの反応に応じて表 1 のような動作を行わせることにした．ここで，○ はセンサが 15cm 以内に障害物を感知したことを示し，× はセンサが障害物を感知しなかったか 15cm 以上の距離があると判定したことを示している．センサに反応がないときは障害物がないとみなし，前進させた．センサに反応がある場合，前後のセンサから障害物から近づいている状態なのか離れている状態なのかを判定し，障害物と一定距離を取るように旋回させた．センサの反応とモータの関係を表 1 に示す．

実際に，Zumo を障害物にむかって走らせてみた．図 3

表 1　センサと反応と動作の関係

センサの反応				モータ	
左前	左後	右前	右後	右	左
×	×	×	×	正転	正転
○	×	×	×	停止	正転
○	○	×	×	正転	正転
×	○	×	×	正転	停止
×	×	○	×	正転	停止
×	×	○	○	正転	正転
×	×	×	○	停止	正転

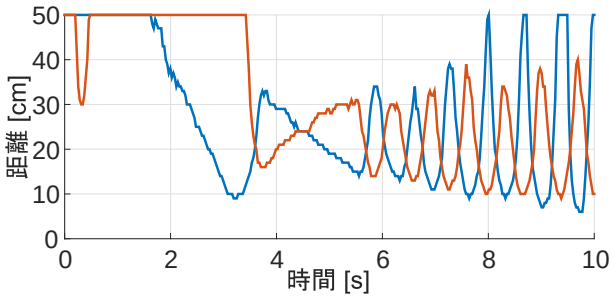

図 3　ON-OFF 制御による前後のセンサの距離関係

において，青線が左前のセンサ，赤線が左後のセンサが示す距離関係である．ただし，50cm 以上の距離を与えたときは 50cm に丸めて表示している．図 3 より，左前のセンサから障害物を 15cm 以内に検知するまでは前進し障害物との距離を縮めていき，障害物との距離が十分近いときには障害物との距離をほぼ一定にしながら走行していることがわかる．なお，障害物との距離は平均的にはほぼ一定に保っているものの，ジグザグ状に走行して動作が滑らかでない．

6　障害物に沿う自動走行制御　—比例制御—

動作が滑らかでないという欠点を補うため比例制御を行う．壁からの距離と，壁に沿う方向と Zumo の進行方向とのなす角度にゲインを掛けることで壁に沿う制御を行うことを考える．壁に近い側の前方の測距センサの出力値に式 (1) と式 (5) を適用して得た値を F，後方の測距センサの出力値に式 (1) と式 (5) を適用して得た値を B とする．r は壁からの目標距離とする：

$$C = k_1 \left(\frac{F+B}{2} - r \right) + k_2 (F - B). \qquad (6)$$

式 (6) 中の $(F+B)/2$ が壁との距離を表し，$F-B$ が壁との角度をそれぞれ表している．k_1 と k_2 は試行錯誤で求め，$k_1 = 0.5, k_2 = 2.55$ に設定する．ここで得られた C を左右のモータの速度差とすることで壁に沿う制御を行う．実際に走行させた結果が図 4 である．

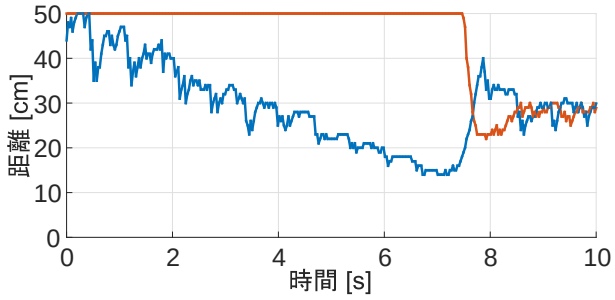

図 4　比例制御による前後のセンサの距離関係

図 4 より ON-OFF 制御を利用した場合に比べ動作が滑らかであることがわかる．

7　おわりに

本研究では，Zumo に測距センサを取り付け，測距センサの特性を調べ，障害物を検知してその障害物に対して一定の距離を保ち走行する実験を行った．5 章で行った ON-OFF 制御ではジグザグ状に走行したが，6 章で行った比例制御では滑らかに走行することができた．今後は壁が曲線を描いている場合への対応が課題である．

8　参考文献

参考文献

[1] Pololu Robotics & Electoronics
https://www.pololu.com/product/2510

[2] GitBook-Arduino docs
https://www.gitbook.com/book/fabkura/arduino-docs/details

[3] 鈴木美朗志：『Arduino でロボット工作をたのしもう！』．技術評論社，東京，2007.

[4] 平田光男：『Arduino と MATLAB で制御系設計をはじめよう！』．TechShare 株式会社，東京，2012.