

自動車模型のライトレースと信号検知 —専用道路におけるバス自動運転を目指して—

2017SC008 東 国繁 2017SC017 岩田 航輝

指導教員:大石 泰章

1 はじめに

1.1 研究背景

日本各地でトロリーバスや、BRT (Bus Rapid Transit) と呼ばれる専用道路を走るバスシステムが普及している。自家用車や歩行者が立入ることのできないバス専用の走行空間を有することにより、渋滞の影響を受けず円滑に運行できるという利点がある。

BRT を含めた一部のバス路線では、近年自動運転の試験が行われている。背景として、バス運転手の不足により増便または続行便を設けることが困難になっていることがある。さらに生産年齢人口の減少に伴い、バス運転手の確保が困難な状況は今後も続く見通しである。これらのことから人員不足に対応しながら輸送サービスを提供し続けるための手法として、自動運転が求められている。

先述の通りバス専用道路では原則として自家用車や歩行者が立入れないことから、一般道と比較して自動運転を行いやすい。本研究ではその点に着目し、BRT を簡略的に捉え、自動運転技術を確認させることを目的とする。バスの代わりに車両型ロボット Zumo を、また信号機の代わりにカラーボールを用いて、ライトレースと信号検知を自動で行う。

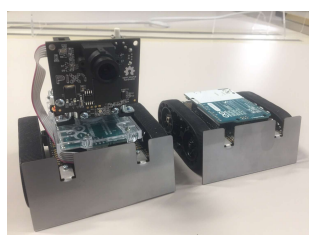


図 1: 車両型ロボット Zumo

1.2 車両型ロボット Zumo について

先述の通り、本研究では Zumo を用いて実験を行う。Zumo の特徴として、左右対称に付けられている 2 個の車輪が、それぞれ独立に駆動する対向二輪型である [1] ことが挙げられる。また、Zumo には Arduino ボードが取り付けられており、ソフトウェア Arduino IDE を使って指示を与えることができる。Zumo に付けられている赤外線センサアレイおよびカメラセンサについての詳細は後述する。

2 ライトレースの実現

2.1 赤外線センサアレイと目標値の設定

Zumo 前方には、地面に赤外線をあて、反射される赤外線の量から地面の色を識別するセンサアレイが取り付けられている。センサアレイは直線状に並んだ 6 つのセンサで構成されており、センサの返り値は 0 (白) から 1000 (黒) である。

Zumo の進行方向に対して垂直にセンサが並ぶようセンサアレイを配置し、図 2 のように各センサに 0 から 5 までの番号をふる。各センサの返り値にセンサの番号をかけて加重平均 y をとると、センサアレイの中央部、すなわちセンサ 2 とセンサ 3 の間でラインを検知した場合は加重平均が $y=2500$ となる [2] ので、これを目標値 R としてライトレースを行う。

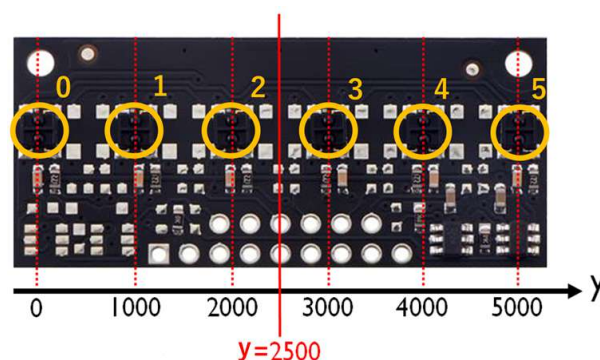


図 2: 赤外線センサアレイ

2.2 偏差と目標値追従

ライトレースにおける各パラメータの定義を表 1 に示す。

表 1: ライトレースにおけるパラメータ

記号	名称
V_{sd}	基準速度
v_L	左車輪の速度
v_R	右車輪の速度
V	車両中心部の速度
t	時刻
e	目標値との偏差
u	車輪速度の変化分 (入力)
y	赤外線センサの出力値

ラインが真っ直ぐな区間では、左車輪と右車輪の速度を基準速度と等しくすることでラインをたどることができる。すなわち、

$$v_L = v_R = V_{sd}$$

とすればよい。曲線においてトレースを行うためには、偏差

$$e = R - y$$

に応じて右車輪と左車輪の速度を変える必要があるため、偏差 e に基づいて適切に入力 u を定めて、左右の車輪の速度を

$$v_L = V_{sd} + u, \quad (1)$$

$$v_R = V_{sd} - u \quad (2)$$

のようにする。ここでは PD 制御を使うことにし、時刻 t における入力 $u(t)$ を次のように定める：

$$u(t) = K_p e(t) + K_d \{e(t) - e(t-1)\}.$$

ただし、 K_p は比例ゲイン、 K_d は微分ゲインである。本研究では次の 3 通りの設計を試みる：

(a) $K_p = \frac{1}{6}$, $K_d = 6$;

(b) $K_p = \frac{1}{4}$, $K_d = 6$;

(c) $K_p = \frac{1}{2}$, $K_d = 6$.

直線のコースを設け、基準速度 V_{sd} を 400 として、上記 (a)~(c) のコントローラに対して赤外線センサの出力値 y の時間変化を調べた。出力値 y の時間変化を図 3 に示す。

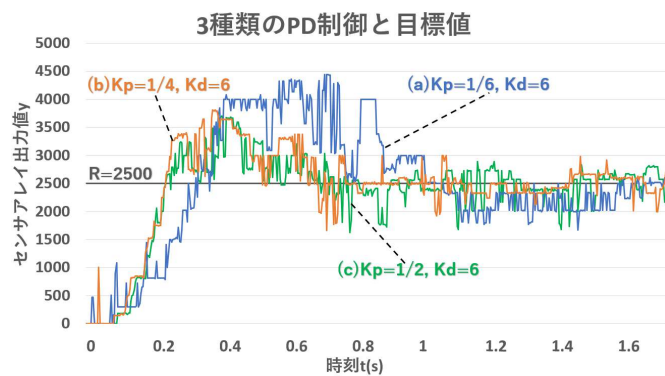


図 3: 出力 y の時間変化

まず (a) の場合、立ち上がり時間に時間を要したほか大きなオーバーシュートも見られ、結果として目標値へ収束するまでに時間を要した。(b), (c) を比較すると、立ち上がりには大きな差は見られなかったものの、(b) の方が振動の少ない定常特性を持つ。以上から、この 3 通りのコントローラのうち適切なものは (b) であると結論付け、信号検知を伴うライトレースでもこのコントローラを用いることとした。

2.3 コースアウト時の停止実験

先述の通り、車両型ロボット Zumo は赤外線センサを用いてライトレースを行うが、何らかの原因でカーブにおいてコースアウトすることがたまに見受けられた。これは、特に急なカーブで起こりやすく、コースアウトしてもそのまま走り続けるといったことが多く見られた。コースアウト時の赤外線センサ出力値 y を調べたところ、出力は連続して同じ値となることから、一定時間出力値 y に変化がなかった際に車両を停止するプログラムを実装した。なお、コースアウトの際には信号検知における段階的な減速 (3.2 節参照) を行わず、速やかに車両を停止させるものとする。プログラムの実行結果を図 4 に示す。

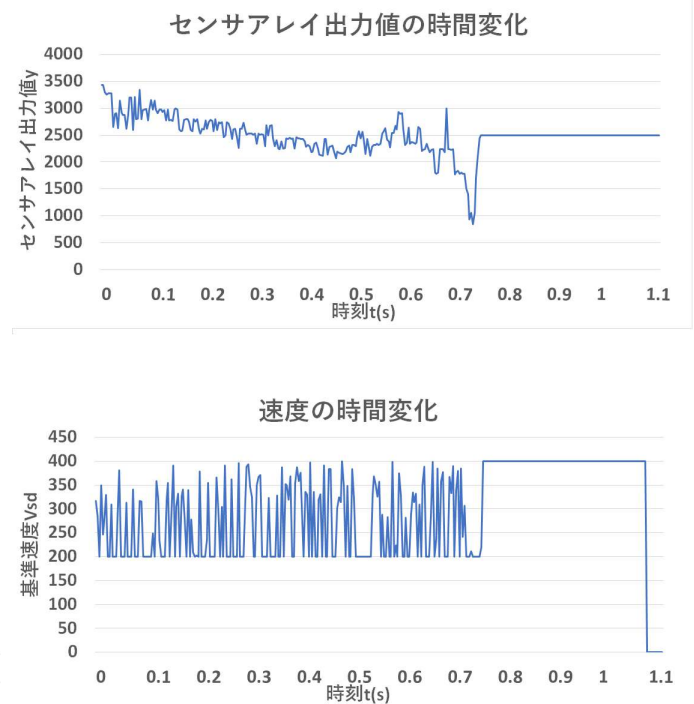


図 4: Zumo の赤外線センサ出力値と速度の時間変化

速度変化のグラフの縦軸は車両中心部の速度 V であり、

$$V = \frac{v_L + v_R}{2}$$

と表せる [1]。この式に 2.2 節の (1), (2) 式を代入すれば

$$V = V_{sd}$$

となることがわかる。

図 4 から、おおよそ 0.3 秒間出力値 y が一定になってから停止していることが確認できる。

3 視覚情報を用いた信号検知

3.1 カメラセンサ PixyCMUcam5

カメラセンサ PixyCMUcam5 を図 5 の左に示す。PixyCMUcam5 は、対象物を検知し、その対象物の色を認識す

るカメラである。黒や白などの色相が不明確な色を除いて、最大で7色を認識することができる。

色の認識の際、PixyMon というソフトウェアを用いる。PixyMon で処理した画像を図5の右に示す。PixyMon は、Pixy カメラと PC を USB ケーブルで接続することで、Pixy カメラで捉えた画像を PC 上で表示できるものであり、画像の明るさや、対象物の色の感度などを設定することができる [3]。なお、本研究では、信号機の代わりとしてカラーボールを使用する。

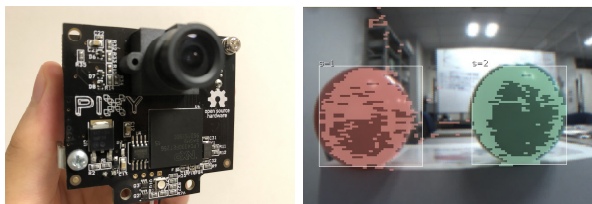


図5: Pixy の概観 (左) と PixyMon で処理した画像 (右)

3.2 各信号に対する動作

本研究では、赤と青の2色を Pixy カメラに記憶させ、それぞれの色に応じて停止または進行させることを試みる。赤のカラーボールを検知したら、徐々に減速して停止、青のカラーボールを検知したら、そのまま直進する。Pixy では画像中の対象物を囲う最小面積の長方形の縦幅 h と横幅 w を画素数を単位として測定できる (図6)。長方形の面積 hw が大きいほど対象物までの距離が近いと考え、表2のように Zumo の基準速度 V_{sd} を変化させる。



図6: PixyMon で処理した画像

表2: Zumo の基準速度の変化

hw	V_{sd}
1500 未満	400
1500 以上 4500 未満	200
4500 以上 14000 未満	100
14000 以上	0

3.3 専用道の走行と信号検知の実現

図7のようなコースを用いて、自動でライトレースと信号検知をさせることを試みる。スタート地点から出発し、

ラインに沿って進んだ後、カーブを曲がり、徐々に減速し、目標地点に置いてある赤のカラーボールの約4cm手前で止まる実験を行った。

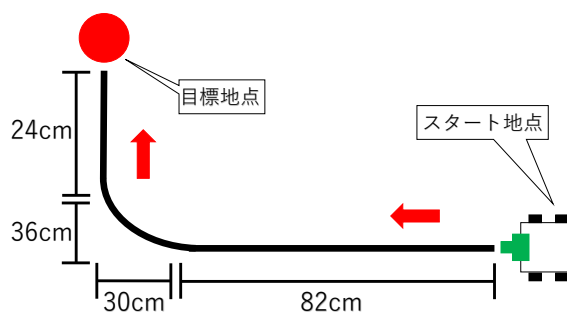


図7: 実験のコース

3.4 信号との距離に応じた速度変化

実際に図7のコースを Zumo で走らせる。走行中に Pixy が検知した赤色のカラーボールの大きさとして Zumo の速度の変化を図8に示す。

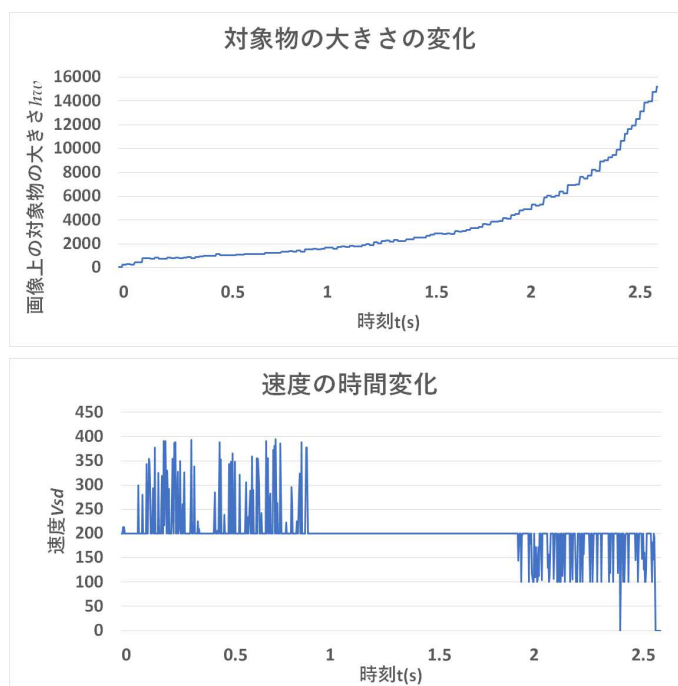


図8: カラーボールの大きさとして Zumo の速度の時間変化

図8の速度変化のグラフ(下)は、対象物の大きさのグラフ(上)に対応している。 hw の値が徐々に大きくなっていくのは、進むにつれて信号との距離が縮まっているためである。

また、図8の速度変化のグラフ(下)を見ると、速度 V_{sd} が激しく変化していることがわかる。これは、赤外線センサの出力に含まれる雑音による影響であると考えられる。しかし、上のグラフと下のグラフを照らし合わせながら見

ると、概ね実験は成功したと言える。カーブを曲がる前の直線を走行している間より、カーブを曲がり終えた後の直線を走行している間の方が hw の値の変化量が大きいため、 $V_{sd}=400$ や $V_{sd}=200$ で走行している時間より、 $V_{sd}=100$ で走行している時間の方が短いことがグラフから読み取れる。そして最後は $V_{sd}=0$ となり、信号の手前で停止していることがわかる。また、今回の実験における Arduino プログラムのサンプリング周期は、計測の結果おおよそ 3.5×10^{-3} 秒であった。

4 待避所の実現

4.1 待避所の概説

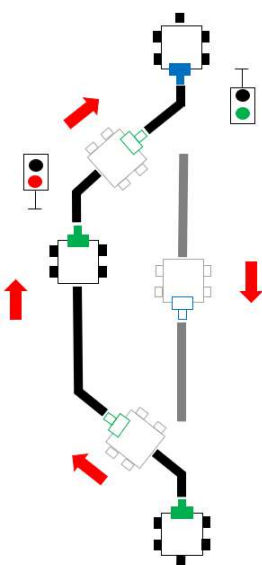


図 9: 行き違いのコース

バス専用道の単路では、バスとバスがすれ違うために、待避所が設けられている。そこには赤と青の信号機が設置されており、赤信号の場合は、待避所に入って停止し、対向車を通さなければならないが、青信号の場合は、そのまま直進できる。このシステムを車両型ロボット Zumo とカメラセンサ Pixy を用いて実証する。図 9 のように Zumo がライン上を走行し、分岐点を左に曲がり、赤信号の手前で一旦停止する。その後、対向車が自車の横を通過し、赤信号が青信号に変わった後、再発進する。なお、この実験においても速度変化を 3.2 節の表 2 の通りとする。

4.2 待避所の実験結果

Zumo を 2 台用いて、図 9 のコース上を走らせたときの様子を図 10 に示す。

先述の通り、赤信号で停止し、対向車が通過した後、再発進することが確認できた。しかし、本研究では、分岐を実線でつなげると、自車が分岐点を左に曲がらずに直進してしまうといった現象が起きたため、分岐点の直線方向は一部実線を消して実験を行った。また、対向車についても赤外線センサを用いず、単に直進させるのみとした。

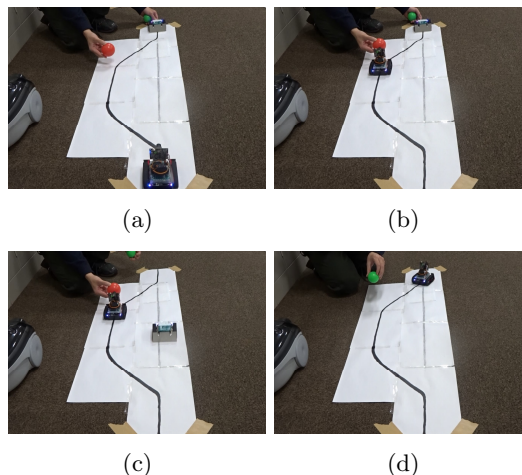


図 10: 行き違いの様子 (a) → (b) → (c) → (d)

5 おわりに

本研究では、Zumo と Pixy を用いてライン上を走行しながら赤または青のカラーボールを検知し、停止または進行することを実現した。ライントレースについては、PD 制御を用いて 3 通りのコントローラを設計し、適切なゲインを求めたことにより精度を向上させた。加えて、コースアウトした際に停止させる機能を追加した。信号検知については、赤のカラーボールを検知した場合、画像内において目標の大きさになったときに、段階的に減速させて停止させることも実現した。また、これらを活用して、簡易的ではあるが待避所での行き違いも実現した。

今後の課題としては、カーブに黄信号を設置し、減速指示を与えることによって、より安全な自動運転が実現できると考えられる。また、実際の専用道路には磁気マーカが一定の間隔で埋められており、それを車両に搭載された磁気センサが検知することで自動運転を行っているケースも多い。そのため、磁気マーカのコースを点線として捉え、実線のコースと同様にライントレースや信号検知が可能となれば、より現実的な自動運転の実装が期待できる。

6 参考文献

- [1] 熊谷正朗 . “車輪移動ロボット”. 東北学院大学 工学部機械知能工学科 熊谷研究室. 2018-04-17. <http://www.mech.tohokugakuin.ac.jp/rde/contents/course/robotics/wheelrobot.html>, (参照 2020-12-19)
- [2] “Arduino Library for the Pololu QTR Reflectance Sensors”. Pololu Robotics and Electronics. <https://www.pololu.com/docs/0J19/all>, (参照 2020-09-27).
- [3] How to use PixyMon —PixyMon の使い方— <http://rcjinput.blog.jp/archives/74297306.html> 2018 年 2 月