カメラセンサを用いた小型車両の自動走行制御

2017SC045 長沼圭真
指導教員：大石泰章

1　はじめに

近年，自動車をはじめとした乗り物の自動運転に対する関心が高まっている．新型ウイルスの蔓延による在宅時間の増加や，インターネット通販の普及に伴い, 物流への需要が高まっているが，特にトラックでの輸送では長距離, 長時間の運転が必要である．そこで，より安全，快適に長時間運転するためのサポート技術が必要であると考える．

本研究ではマイコンボード Arduino leonard を搭載した小型車両 Zumo にカメラセンサ Pixy2 を取り付け, 画像処理により車線に沿った自動走行を実現する．具体的にはカメラセンサにより黒線を検知することで，一本の黒色の曲線上に沿って自動走行させること，および，Zumo の両脇に引かれた二本の曲線の中間を自動走行させることを目的とする．

2　使用する機器

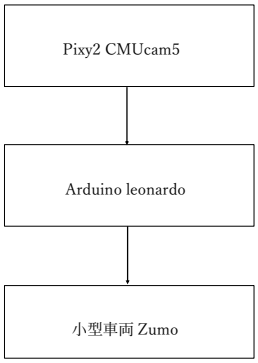

図 1　システムの構成

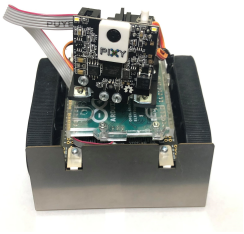

図 2　小型車両 Zumo

2.1　システムの構成

図 1 は構成するシステムである．カメラセンサ Pixy2 が地面の黒色のラインを検出し, その位置を計測する．そして，マイコンボード Arduino で計測結果に基づいて，指令を出すことによって小型車両 Zumo を動かす．小型車両 Zumo を図 2 に示す．

2.2　小型車両 Zumo

本実験では小型車両 Zumo 一台を制御対象として使用する．Zumo は長さ 10cm，横幅 10cm，高さ 3.5cm 程度の大きさの無限軌道式ロボット用プラットフォームであり，マイコンボード Arduino leonard が Zumo の上にかぶさるように取り付けてある．また，Zumo は車両左右に二つずつモータを搭載しており，左右のモータが回転することでキャタピラが動いて走行できる．左右のモータは独立して回転させられるためカーブを曲がることやその場で回転することができる．

2.3　カメラセンサ Pixy2

Pixy2 CMUcam5 (以下 Pixy2 と記す) は各辺 4cm 以下の大きさであり，高速イメージセンサによって対象物体を検出し, 追尾することができるカメラセンサである．Pixy2 は左折，右折，減速などの指示を 60fps で実行することができる．本実験で使用する Zumo に Pixy2 が土台とサーボモータを介して取り付けられており，Pixy2 は Zumo 上で正面方向を中心に ±90 度の範囲で回転させたり，上下方向に 90 度の範囲で傾けたりすることが可能である．今回 Pixy2 は地面の黒線が見えるように約 45 度下方向に傾け，正面方向に向けた状態で使用する．

3　曲線に沿った走行

3.1　Pixy2 から得られる座標情報

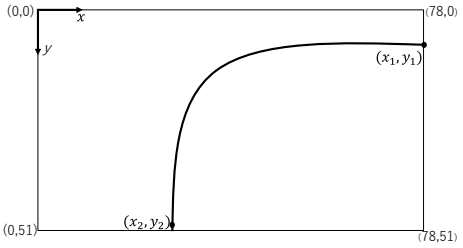

図 3　カメラの画像から得られる座標情報

まず一本線上を走行させる場合について考える．図 3 は Pixy2 に写っている画像を座標を使って表現したものである．この座標情報は関数 getMainFeatures を使うことで Pixy2 から取得できる．座標は画面上の左上を原点として，x 軸は原点から右向きを正，y 軸は原点から下向きを正ととる．x 軸の最大座標は 78，y 軸の最大座標は 51 である．関数 getMainFeatures では Pixy2 に写っている画像の中の一本の曲線の両端点の座標を取得することができる．一本の曲線上を走行させる場合，制御に必要な座標は端点のうち y 座標が小さい (x_1,y_1) のみである．

3.2　Zumo の制御

曲線の両端点うち y 座標が小さい方の x 座標を $x_1(t)$ とし，画像の中心の x 座標 (39,0) との誤差 $e_1(t)$ を求めると

$$e_1(t) = x_1(t) - 39. \tag{1}$$

となり，求めた誤差 $e(t)$ に基づいてゲイン k_p，k_d を用いてモータへの入力 $u(t)$ は次のように求められる:

$$u(t) = k_p e(t) + k_d(e(t) - e(t-1)). \quad (2)$$

符号が逆の入力 $u(t)$ を左右のモータに加えることで Zumo の方向を変化させ曲線に沿って走行させる．

3.3　二曲線の中間の走行について

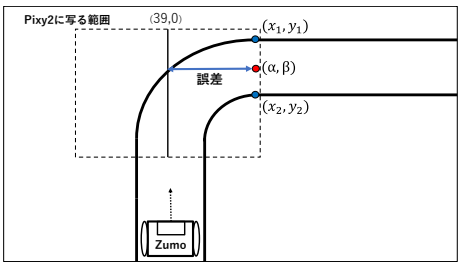

図 4　二本の曲線の中間を走行させるための方法

Zumo を二本の曲線の中間を自動走行させる方法を図式化したものを図 4 に示す．二本以上の曲線の端点の座標を取得するには関数 getAllFeatures を使用する．まず二本の曲線の端点の座標 (x_1,y_1) と (x_2,y_2) を取得し中点 (α,β) を求める．次に時刻 t での中点 (α,β) の x 座標を $\alpha(t)$ とし，x 軸の中点 (39,0) との誤差を $e(t)$ として

$$e(t) = \alpha(t) - 39. \quad (3)$$

とすることで一本線上を走行させる場合と同じく式 (2) によって Zumo を制御して走行させる．

4　実験結果

4.1　一本線上の走行

右向きにカーブしている一本の曲線上を走行させたときの Pixy2 が取得した曲線の両端点 (x_1,y_1)，(x_2,y_2)(ただし $y_2 > y_1$) の x 座標，x_1 と x_2 の変化を図 5 に示す．

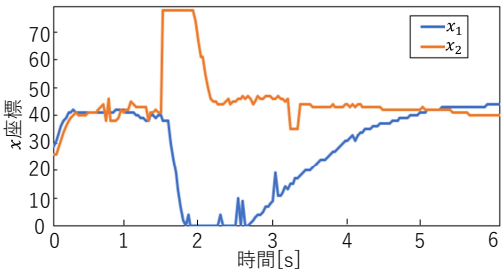

図 5　一本の曲線上を走行した際の座標データ

図 5 より，1 秒を過ぎたあたりから x_2 と x_1 の座標が広がっているのが分かる．これは曲線上を走行するときに曲線の端点 (x_1,y_1) に向かって走行するため，曲線の真上を走行するのではなく曲線の少し内側を走行するためである．

4.2　二曲線の中間の走行

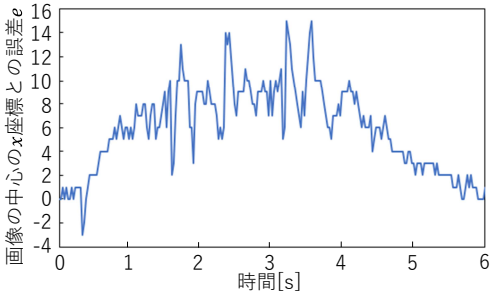

図 6　二本線の中間を走行した際の座標データ

右向きにカーブしている二本の曲線の中間を走行させた際に取得した誤差 $e(t)$ のデータを図 6 に示す．図 6 のグラフより，曲線に沿って走行する時は誤差 $e(t)$ が増加しているのがわかる．これは一本線上を走行する時と同様に曲線の端点に向かって走行するためカーブの少し内側を走行するためである．一方，カーブを通過した 5 秒あたりから $e(t)$ がほぼ零になり，二本の曲線の中間を走行していることが分かる．

5　おわりに

本研究ではカメラセンサ Pixy2 を小型車両 Zumo に取り付けて使用し，座標情報を利用した制御によって一本の曲線上と，二本の曲線の中間を走行させることを実現した．本実験では赤外線を使用したライントレースと比較してより滑らかに線に沿って走行することができたが，進行方向に設定した目標と現在地との誤差を使用したフィードバック制御によるものと考えられる．今回の実験では曲線上、または二本の曲線の中間を走行する際，正確に曲線に沿った動きではなくショートカットする動きをしていたが，実際に自動車がこのような動作を行いながら安全に走行することはできない．この問題を解決するためには，カメラセンサによって取得する座標情報を端点のみではなく，中点の座標も加えて位置制御を行うことで Zumo の位置制御を行うことが有効であると思われる．

参考文献

[1] pixy2 Line tracking quickstart
`https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:line_quickstart`

[2] pixy2 Line tracking for line-following
`https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:line_tracking`