

補助線を用いたレーン走行アルゴリズムの開発

神谷享昌† 築地立家‡

東京電機大学大学院 理工学研究科

1. 研究概要

本研究では、実車の10分の1サイズのロボット車を使用して、カメラ画像認識処理によって車体の位置と回転角度を計算し、障害物を回避しながら走行ラインに沿って自動走行するアルゴリズムの開発を行う。その際、カメラ画像処理によって目標点を設定し、カメラ画像処理によって安全ライン（仮想的な直線）を設定する。そのうえで、安全ラインに触れないようにレーンキープ走行することによって障害物回避を実現する。これによって、安全ラインを障害物から十分離れた位置に設定して、安全走行を実現する。

2. 実験環境

本研究で使用した RoboCar のスペックは以下の通りである。[1]

CPU:AMD Geode®LX800 Processor 500MHz

OS:Linux

画像認識モジュール:ZMP 製モジュール(IMAPCAR®:NEC 製並列プロセッサ

開発環境:gcc

3. 従来研究

安全ラインを考慮した自動走行の実験研究として、レーザレンジセンサーによって壁を認識して壁と平行な安全ラインを設定して、そのラインを回避しながら走行するための研究があげられる。[2]一方、本研究では、壁に限らず、任意に配置された障害物に対する安全ラインを設定して回避することができている。また、目標点をあらかじめ固定しておいて、そこに向けて滑らかに走行させるようなアルゴリズムの研究もある。[3]一方、本研究では、目標点を走行しながら自動検出して安全ライン走行を行うため、コース上の変化にリアルタイムで対応できる。

4. 今回の提案

本研究では、レーンキープ走行と、安全ライン（仮想線）を用いて障害物回避を処理する新しい画像処理システムを提案して、RoboCar による実装実験を行う。そのために、レーンキープ走行と障害物回避のアルゴリズムを提案する。レーンキープ走行の方法としては、走行中、走行レーンである白線をエッジ検出により点の集合として2次元配列に抽出し、最小2乗法によって近似した直線の傾きにより操舵角を決定する。

また、障害物回避の方法としては、認識した障害物の端の座標と車体とを結ぶ仮想的な直線を用いて操舵角を決定する。

5 安全ラインを用いた障害物回避手順

RoboCar はカメラ画像の解析によって得られる xz 平面のヒストグラム画像から、安全ラインを設定して、それを回避するように運動する。その手順を以下に述べる。

5.1 障害物発見時

図1は、コース全体を上空から眺めた座標系であり、これを絶対座標系と呼ぶ。図1において、絶対座標系の原点を (x_0, y_0) とし、 x 軸、 y 軸を設定し、目標点 (x_1, y_1) 、および安全ラインを設定する。図1において、緑と赤のラインの交点が存在している間、右に曲がり、交点が存在しなくなれば左に曲がる。

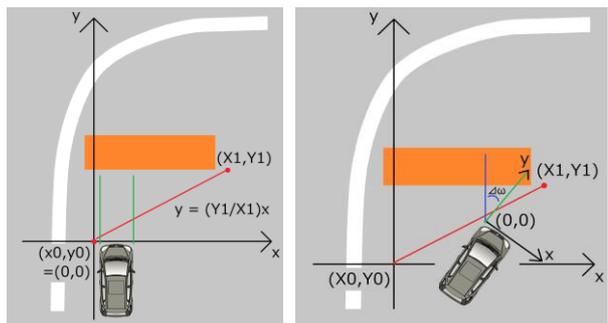


図1 障害物発見時の絶対座標系 (上から見た図)

5.2 走行中

車体が走行して移動した結果、カメラから見える座標系は毎処理ごとに変化する。そこで、本研究では、車体が移動中であっても、安全ラインの方程式を絶対座標系上では不動となるようにしておきたい。そのために、走行中に対座標の原点 (x_0, y_0) を常時更新することとする。これより、図2における目標点 (x_1, y_1) も、絶対座標系上で安全ラインも、正しく更新できる。すなわち、安全ラインの方程式を更新しながら走行を行う。そして、緑と赤のラインの交点が存在している間、右に曲がり、交点が存在しなければ左に曲がる。すなわち、緑のラインが赤いラインに触れないように走行することで、安全な障害物回避を行う。(図2)

6. 原点を求めるための走行実験例

タイヤの角度 $\theta = 30^\circ$ 、ループ回数 $n = 500$ を初期値として、走行実験を行うことにより、 (x_0, y_0) を求めるためのパラメータを以下のように確定させた。まず、走行前と

Development of a lane-keeping control algorithm using virtual lanes for vehicle.

† Takaaki Kamiya ‡ Tatsuie Tsukiji

Graduate School of Tokyo Denki University School of Science and Engineering

走行後における (x_1, y_1) の変化量は, $(\Delta x, \Delta y) = (-44, -7)$ であった(図3).

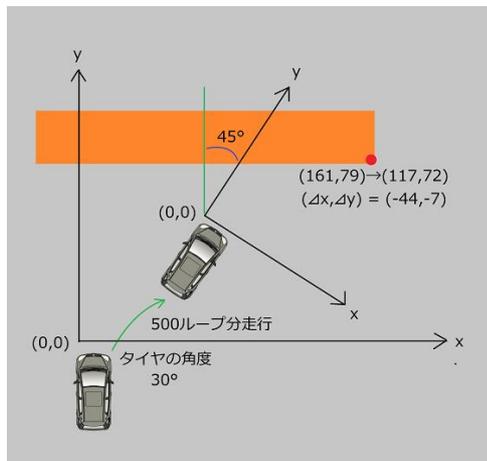


図3 走行実験の例

初期値を $n = 500$ としたので, 1 ループあたりでは
 $(\Delta x, \Delta y) = (-44/500, -7/500) = (-0.088, -0.014)$
 と算出されるため, n ループあたりでは
 $(\Delta x, \Delta y) = (-0.088n, -0.014n)$
 と予想される. また, 初期値を $\theta = 30^\circ$ としたので, n
 ループあたり 1° あたりでは,
 $(\Delta x, \Delta y) = (-0.088/30, -0.014/30)$
 $= (-0.0029333333, -0.0004666667)$
 と算出されるため, n ループあたり θ 度あたりでは
 $(\Delta x, \Delta y) = (-0.0029333333n\theta, -0.0004666667n\theta)$
 となることが予想される. そこで, この走行実験によっ
 て定まるパラメータとして,

$$0.0029333333 = A, \quad 0.0004666667 = B$$

を導入して,

$$(\Delta x, \Delta y) = (-An\theta, -Bn\theta)$$

と想定したうえで, 走行中の原点は

$$(x_0, y_0) = (An\theta, Bn\theta)$$

と予想する. ただし, この予想値は, 車体の角度の変化量
 $\Delta\omega = 0^\circ$ であった場合である. ところで, 実験における
 車体の角度の変化量 $\Delta\omega \approx -45^\circ$ であったので, 求めた
 (x_0, y_0) を, 原点を中心に 回転させると, 走行中の原点座
 標は下記となる. (1)(2)

$$x_0 = An\theta \cos \frac{45}{\theta n} - Bn\theta \sin \frac{45}{\theta n} \quad \dots(1)$$

$$y_0 = An\theta \sin \frac{45}{\theta n} + Bn\theta \cos \frac{45}{\theta n} \quad \dots(2)$$

7. 走行実験

走行実験によるパラメータ設定により, 走行中の原点
 座標を上記のように想定した上で, 障害物の形状, 配置を
 変えた走行実験をおこなった. その結果の安全ラインの
 設定と車体の軌跡を図4に示す.

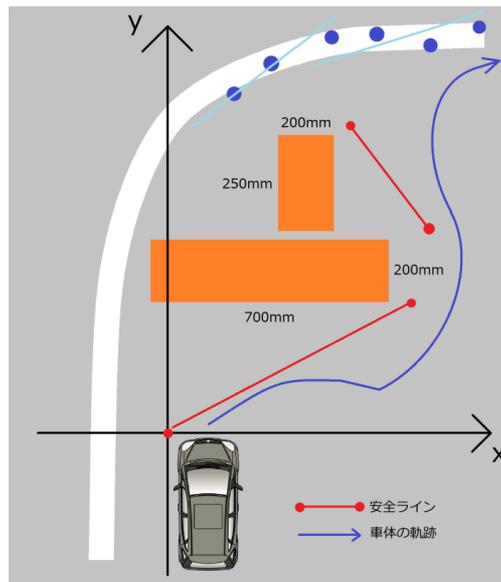


図4 走行実験パターン

8. 結果

障害物の安全な回避のために, 安全ラインを作成し, それに
 関して回避を行う手法提案した. すなわち, 車体の走行中
 に, 現在の車体位置を原点としたときの (x_0, y_0) を座標変換
 で求め, 安全ラインの方程式をリアルタイムで導く手法を
 とった. 様々な位置に複数の障害物を設けた実証実験の結
 果, 障害物発見時に, 安全に回避を行うための安全ライン
 の方程式を作成し, その安全ラインを疑似的に固定したま
 ま走行することにより, 安全ラインが実際のコース上に存
 在するかのように扱うことで, よりスムーズで安全な走行
 が可能であることが確かめられた.

9. 今後の課題

今後は, 外部環境の明るさに応じて, 白線認識処理の際
 の障害物認識処理に用いるしきい値を自動的に調整する
 ことにより, 様々な外部環境に対応した画像処理系を確
 立したい. また, $\theta = 30^\circ, n = 500$ 以外の様々な初期値で
 走行実験を行ってパラメータを改良することにより, より
 安全でスムーズな走行を目指したい. また, より複雑
 なコース上での走行実験を行いたい.

参考文献

- [1] ZMP RC Library Reference Manual ドキュメント
RoboCarVer1.5, ZMP 社
- [2] 橋本雅文『狭小屈曲通路における自立走行車の制御ア
ルゴリズム構築』(2007)
- [3] 斎藤, 安信『ファジィ目標を用いた四輪車の自動走
行』(2009)
- [4] ZMP RoboCar Reference Manual, ZMP 社