

# 最適速度旋回アルゴリズムによるスキッドステアリング 2D ロボットのひも状走行

山田 将司<sup>1</sup>，李 方正<sup>1</sup>，本田 泰<sup>2</sup>

<sup>1</sup> 室蘭工業大学大学院 工学研究科 情報電子工学系専攻

<sup>2</sup> 室蘭工業大学大学院 しくみ解明系領域

## 概要

我々は以前、二次元最適速度モデルという自己駆動モデルを知能として組み込んだ2次元最適速度ロボットを開発した。今回新たに、スキッドステアリングにより4輪で走行するロボットを開発した。以前の2輪走行アルゴリズムをもとに、スキッドステアによる4輪走行のための2次元最適速度旋回アルゴリズムを導出した。また、接触センサから距離センサに変更することで、障害物に衝突する前に方向転換を行うようにアルゴリズムを変更した。本研究の目的は2輪の走行ロボットで観測されたひも状走行が4輪のロボットでも観測できるか実験を行った。結果として、4輪の走行ロボットでもひも状走行は観測された。

## String-like traveling of skid-steering 2D robots by Optimal Velocity Turning Algorithm

Masashi Yamada<sup>1</sup>，Li Fangzheng<sup>1</sup>，Yasushi Honda<sup>2</sup>

<sup>1</sup> Division of Information and Electronic Engineering, Graduate school of Engineering, Muroran Institute of Technology, Japan

<sup>2</sup> College of Information and System, Muroran Institute of Technology, Japan

## Abstract

We have developed the 2D optimal velocity robot which is made by incorporating a self-driven model which is named the 2D optimal velocity model previously. And recently, we also developed a robot that run by four whells with skid steering. Then, we derived a 2D optimal velocity turning algorithm for four-wheels driving by skid steering based on the two-wheel driving algorithm previously. In addition, by changing from the contact sensor to the distance sensor, the algorithm also is changed to another algorithm which operates as robot's direction change before colliding with obstacles. The purpose of this study was to check if the string-like motion can be observed by the four-wheels land-moving robot as same as two-wheels land-moving robot. As a result, string-like motion was also observed with a four-wheel land-moving robots.

## 1 はじめに

歩行者や交通渋滞といった集団行動は、各個体の相互作用によって自己組織的に形成される動きであ

る。我々はその中で交通流モデルである一次元最適速度モデルを2次元に拡張した2次元最適速度ロボットを開発した[1]。

環境と走行ロボットが相互作用することで行動を

創発する。本研究では、スキッドステアリングで走行するロボットを使用して実験を行った。スキッドステアリングとは、スキッド(滑る)とステアリング(操舵)という旋回方法であり、左右のタイヤを滑らせながら旋回する。走行ロボットの身体性が変化しても先行研究[1]と同じひも状走行が観測されるか確かめることを本研究の目的とする。

## 2 二次元最適速度モデル

二次元最適速度モデルは以下の運動方程式(1)式で表される[2]。自己のロボットの速度と最適速度の差から、加速度を求めるモデルである。

$$\ddot{\mathbf{r}}_j = a \left[ \sum_k \mathbf{V}(\mathbf{r}_{kj}, \dot{\mathbf{r}}_j) - \dot{\mathbf{r}}_j \right] \quad (1)$$

$$\mathbf{V}(\mathbf{r}_{kj}, \dot{\mathbf{r}}_j) = (1 + \cos \theta_{kj}) f(r_{kj}) \mathbf{n}_{kj} \quad (2)$$

$$f(r_{kj}) = \alpha [\tanh \beta(r_{kj} - b) + c] \quad (3)$$

$\mathbf{V}(\mathbf{r}_{kj}, \dot{\mathbf{r}}_j)$  は  $j$  番目のロボットが  $k$  番目のロボットから受ける相互作用項である。 $\mathbf{n}_{kj}$  は  $\mathbf{r}_{kj}$  の単位ベクトルを表す。 $\theta_{kj}$  は  $j$  番目のロボットの速度ベクトル  $\dot{\mathbf{x}}_j$  と相対位置  $\mathbf{r}_{kj}$  のなす角であり、 $(1 + \cos \theta_{kj})$  は異方性を表す項である((2)式参照)。モデル上では、 $\mathbf{n}_{kj}$  は  $\mathbf{r}_{kj}$  の単位ベクトルだが、ロボットに組み込む際に、 $\mathbf{n}_{kj} = (\sin(\theta_{kj}), \cos(\theta_{kj}))$  と近似した。 $f(r_{kj})$  は最適速度関数であり、ロボットとの距離  $r_{kj}$  に応じて引力または斥力を決定する関数である((3)式参照)。

本研究ではこのモデルを離散化し、走行ロボットの速度ベクトルを走行ロボットの前進する速度と旋回する速度に変換する必要がある。 $\Delta t$  秒後の  $\dot{\mathbf{r}}_j$  を離散化して求めると(4)式となる。(2)式内の  $\theta_{kj}$  は  $360^\circ$  反応できる範囲があるが、ロボットに搭載されているカメラの画角は約  $104^\circ$  である。よって、 $-52^\circ < \theta_{kj} < 52^\circ$  でなければ、カメラで他機体を認識できない。

4輪の接地面での速度  $v_L$ ,  $v_R$  を(7), (8)式で求める。ここでの  $v_L$  は進行方向左側の車輪の速度、同様に  $v_R$  は右の車輪の速度を表す。

先行研究[1]では角加速度を用いて計算していたが、本研究では角速度  $\omega$  を用いることとした((6)式参照)。角加速度を用いて計算した場合、走行ロボットはスキッドステアで旋回するため、アルゴリズム上の旋回角度とロボットが旋回した角度にずれが生じてしまう。旋回方法の変更により(7), (8)式内の  $d$  は車輪とロボットの中心との間の距離(図1参照)

だが、 $d$  を実際の値よりも大きくすることでより強い旋回力を得られるように変更した。先行研究[1]では  $d = 0.065[\text{m}]$  としていたが、本研究では、 $d$  を旋回力を調整するゲインとして扱うことにして、 $d = 5$  に調整した。

$$\dot{\mathbf{r}}_j(t + \Delta t) = \dot{\mathbf{r}}_j(t) + \Delta t \ddot{\mathbf{r}}_j(t) \quad (4)$$

$$\dot{\mathbf{r}}_j(t + \Delta t) = (\dot{x}_j(t + \Delta t), \dot{y}_j(t + \Delta t)) \quad (5)$$

$$\omega = \arctan\left(\frac{\dot{x}_j(t + \Delta t)}{\dot{y}_j(t + \Delta t)}\right) \quad (6)$$

$$v_L(t + \Delta t) = \dot{r}_j(t + \Delta t) + d \cdot \omega \quad (7)$$

$$v_R(t + \Delta t) = \dot{r}_j(t + \Delta t) - d \cdot \omega \quad (8)$$

## 3 2次元最適速度ロボットの身体性

本研究では4輪駆動の走行ロボットで実験を行う(図1参照)。先行研究[1]では2輪とポールキャスターで走行するロボットであったが、本研究ではスキッドステアリングにより4輪で走行するロボットを使用する。左右の車輪はそれぞれ中央のギヤに繋がっていて、2つのサーボモータを用いて独立に回転させる。2輪から4輪へと変更した理由としては、2輪とポールキャスターでは地面に段差や凹凸がある場合ポールキャスターが引っかかるてしまう。4輪の場合だと走破性が高く、段差や凹凸がある場合でも2輪走行と比べ、比較的安定して走行が可能である。

本研究で使用するロボットにはtof距離センサとカメラ、2つのモータを用いる。カメラはRaspberry Pi Cameraというカメラで、他のロボットの方角と距離を検出している。tof距離センサは走行ロボットと壁や他機体との間の距離を測定している。進行方向正面に1つ、進行方向正面から左右  $45^\circ$  外側に向けたセンサを左右に1つずつ、合計3つ装着している。tof距離センサとカメラから得られた情報を利用して走行ロボットは2つのサーボモーターを動かし走行する。

### 3.1 他機体認識

本研究では複数の走行ロボットの集団行動を観測する。そのため、自己のロボットとは別のロボットの相対角度と位置を認識しなければならない。他機体認識に使用するカメラはRaspberry Pi Cameraである。本研究ではこのカメラを用いてロボット上部に取り付けられたピンク色のカップから、他機体との相対角度と距離を認識している。距離を求める際

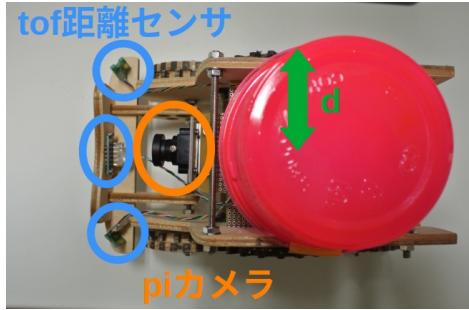


図 1: 使用する走行ロボットを真上から見た写真

に，カメラに写ったピンク色のカップの幅から最小二乗法を用いてフィッティングを行った。

### 3.2 相乗平均距離と疑似的弹性散乱

本研究では，弹性境界を考慮する必要性があるため，ロボットは壁や他機体を検出して反射する必要がある。壁や他機体に近づいた際には，弹性散乱を行うために tof 距離センサから得られる距離データの相乗平均を (9), (10) 式で求める。ここで  $\gamma = 0.33$  とし，3 つの tof 距離センサから得られる距離データを等加重で計算を行う [3]。相乗平均距離を使用する理由としては，障害物に近づいた際に，より反応しやすくするために用いる。

$d_C$  は中央の距離センサの距離データ， $d_L$  は左の距離センサの距離データ， $d_R$  は右の距離センサの距離データである。tof 距離センサから得られる距離データの単位は [m] である。

$$\bar{d}_L = d_C^\gamma \times d_L^{1-\gamma} \quad (9)$$

$$\bar{d}_R = d_C^\gamma \times d_R^{1-\gamma} \quad (10)$$

本研究では， $\bar{d}_L$  または  $\bar{d}_R$  が 0.2 以下になった場合はその場で  $\bar{d}_L$  と  $\bar{d}_R$  の値が大きい方へ 0.3 秒間，出力 100[%] で方向転換する。この方向転換を疑似的弹性散乱とみなす。

## 4 走行実験

本研究では，図 2 のドーナツ型コース，走行ロボット 8 台を用いて走行実験を行う。外側の壁は半径 2[m] の円，内側の壁は半径 1[m] の円である。初期状態として，コース上に 4 台を時計回りに配置，残りの 4 台を半時計回りに配置して実験を行う。走行実験では，2 種類のアルゴリズムで走行させ流量を比較する。1 つ目のアルゴリズムは，弹性散乱のみで走行す

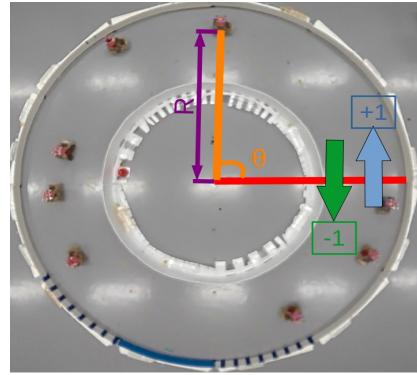


図 2: 実験コースをコース中心真上から見た写真  
(ひも状走行になっている状態)

るアルゴリズムである。 $\bar{d}_L < 0.2$  もしくは  $\bar{d}_R < 0.2$  であれば，弹性散乱を行う。それ以外の場合には進行方向正面に走行する。

2 つ目のアルゴリズムは，2 次元最適速度旋回アルゴリズムである。他機体を発見した場合は近づいていき，それ以外の場合は進行方向正面に進み，相乗平均距離で弹性散乱も行う。パラメータは  $a = 3.0[1/s]$ ， $\alpha = 0.15[m/s]$ ， $\beta = 8.0[1/m]$ ， $b = 0.3[m]$ ， $c = 1.0$  で実験を行う。

### 4.1 評価方法

走行実験は流量という観測量を用いて定量的に評価する ((11) 式参照)。(11) 式内のパラメータは  $l$ :全走行ロボットの通過回数， $t[min]$ : 計測時間， $w[m]$ : 使用したコースの道幅である。

$$f_{rate} = \frac{l}{tw} \quad (11)$$

実験コース (図 2) の右側にある赤色の直線を基準線とする。その基準線を時計回り (緑矢印) に通過すると通過回数を -1 回，反時計回り (水色矢印) に通過すると通過回数を +1 回する。

## 5 実験結果

弹性散乱のみと 2 次元最適速度旋回アルゴリズムでの走行を行った。走行状態を表すグラフを図 3～図 6 に示す。

図 3，図 4 のグラフは走行ロボットの  $\theta$  と時間の関係図である。横軸は時間 [秒]，縦軸は角度  $\theta[\text{rad}]$  である。図 5，図 6 のグラフは走行ロボットの時間変化におけるコース中心からの直線距離を表している (以下  $R$  と表記する)。

図3のグラフでは、実験開始から約80秒までは $\theta$ のグラフが途中で折れ曲がっている。これはロボットが方向転換している状態である。約80秒以降は $\theta$ の変化が一定であることから、このロボットは時計回りに走行している。

弾性散乱のみで走行した場合のグラフである図4を見ると、実験終了まで $\theta$ のグラフが折れ曲がっている。2次元最適速度旋回アルゴリズムでの走行に比べると不安定な走行状態である。

図3から、2次元最適速度旋回アルゴリズムでは、約80秒以降で時計回りに走行している。図5をみると、実験開始から約80秒まではコース内を約60cm移動していることがわかる。これは、対面走行時に他のロボットと衝突しないよう回避行動をとっているため、 $R$ の値が大きく変動している。約80秒以降だと、 $R$ の変化が減少傾向にあることから、他のロボットを回避する行動が少ない走行状態だと言える。約230秒以降はひも状走行が形成された。

弾性散乱のみで走行した場合には2次元最適速度旋回アルゴリズムと比べて、実験終了時まで $R$ の値が大きく変動している。本研究で使用するコースは図2の円形コースであり、比較的密度が低いことから走行時には、コースの内側や外側に接近する。すなわち、全ロボットが一方向で走行しているのではなく、対面走行状態になっている可能性が高いと言える。このように、 $\theta$ と $R$ のグラフ両方から、走行ロボットの走行状態を予測することが可能である。

合計8回の走行実験の平均流量は、弾性散乱のみで走行した場合は6.9、2次元最適速度旋回アルゴリズムで走行した場合は8.2となった。ひも状走行が形成されるため、2次元最適速度旋回アルゴリズムの方が弾性散乱のみと比べて流量が増加する。

## 6 まとめと今後の課題

本研究は先行研究との相違点として、センサとカメラの画角を変更した。走行ロボットはスキッドステアで旋回することから、2次元最適速度旋回アルゴリズムを開発した。2種類のアルゴリズム(2次元最適速度旋回アルゴリズム、弾性散乱のみを行うアルゴリズム)で走行させ、流量という観測量で比較を行った。結果として、2次元最適速度旋回アルゴリズムでは、ひも状走行が創発されるため弾性散乱のみの場合と比較して流量が増加する。本研究で扱う2次元最適速度旋回アルゴリズムのパラメータは種類が多く、他のパラメータを使用し実験を行って

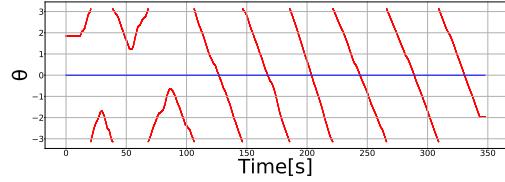


図3: 走行中の時間変化における角度変化 (2dovr)

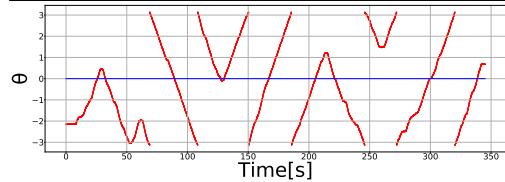


図4: 走行中の時間変化における角度変化 (弾性散乱)

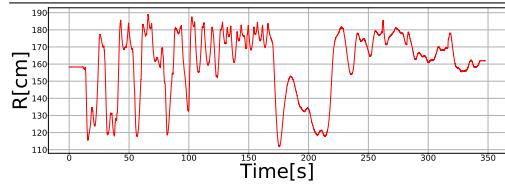


図5: 走行中の時間変化における半径の変化 (2dovr)

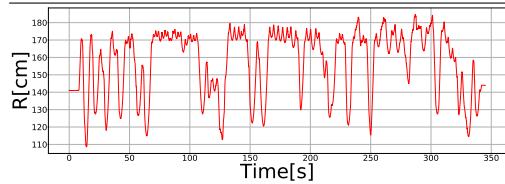


図6: 走行中の時間変化における半径の変化 (弾性散乱)

いきたい。

## 参考文献

- [1] 川野多佳也, 宮島高志, 本田泰, 二次元最適速度ロボットの開発と集団走行実験, 第23回交通流と自己駆動粒子系のシンポジウム論文集, p63-p66, (2018)
- [2] 石渡龍輔, 衣川亮太, 杉山雄規, Kantorovich-metric を用いた2次元OV粒子の集団流の感応度依存性の解析, 第22回交通流と自己駆動粒子系シンポジウム論文集, P41-44, (2016)
- [3] 李方正, 橋爪晋平, 本田泰, 非線形感覚運動写像ロボットの対面流-1方向走行流への転移と流量のコース幅依存性-, 第26回交通流と自己駆動粒子系のシンポジウム, (2020)