



二次元最適速度ロボットの開発と集団走行実験

メタデータ	言語: jpn 出版者: 交通流数理研究会 公開日: 2020-07-07 キーワード (Ja): キーワード (En): 作成者: 川野, 多佳也, 宮島, 高志, 本田, 泰 メールアドレス: 所属:
URL	http://hdl.handle.net/10258/00010215

二次元最適速度ロボットの開発と集団走行実験

川野多佳也¹, 宮島高志¹, 本田泰²

¹ 室蘭工業大学 情報電子工学系専攻

² 室蘭工業大学 しくみ情報系領域

概要

二次元最適速度モデルによる集団走行実験のために、触覚センサと色認識カメラを備えた走行ロボットを開発した。この触覚センサにより、円形のコース内で弾性反射運動を実現できることを確認した。また、色認識カメラによってロボット間の距離と方角を求めた。さらに二次元最適速度モデルから二輪駆動のための離散的アルゴリズムを導出した。4台の最適速度ロボットによる紐状走行状態を実現する条件を発見した。

Development of two-dimensional optimum velocity robot, and experimentations for group travel

Takaya Kawano¹, Takashi Miyajima¹, Yasushi Honda²

¹ Division of Information and Elctronic Engineering, Muroran Institute of Technology

² College of Information and Systems, Muroran Institute of Technology

Abstract

We designed a Two-Dimensional Optimum Velocity (2dov) Robot with a feeler for inertia reflections in a circular course. In order to move 2dov robots we introduced an algorithm for two wheels. We found that four robots with the algorithm bring a formation like a string.

1 研究目的

我々は以前、一次元最適速度モデルを実装したロボットにおける交通流を再現する実験を行った [1]。最適速度モデルには他に、次元を二次元に拡張した二次元最適速度モデルも存在する [2][3]。

迷路上の空間での二次元最適速度粒子は周期境界、あるいは弾性境界を接続するような最適な経路上で準安定な紐状の粒子運動が形成されることが知られている。このモデルの特徴は迷路上でも自律的に粒子が最適な経路を見つけ出すことにある。このモデルをシミュレーションだけでなく実空間上で再現できるようにアルゴリズムを導出し、そのアルゴリズムを用いたロボットを使って実験を行うことで同じような集団行動 [4] が見られるかを調べるのが本研究の目的である。

2 二次元最適速度モデル

二次元最適速度モデルは生物集団の運動を再現する粒子モデルである。このモデルは以下の運動方程式で表される。

$$\ddot{\mathbf{x}}_j = a \left[\mathbf{V}_{s,j} + \sum_{k \in \{r_{kj} < r\}} \mathbf{V}(r_{kj}, \dot{\mathbf{x}}_j) - \dot{\mathbf{x}}_j \right] \quad (1)$$

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

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

\mathbf{x}_j は j 番目の粒子の位置を表す。 $\mathbf{r}_{kj} \equiv \mathbf{x}_k - \mathbf{x}_j$ と $r_{kj} = |\mathbf{r}_{kj}|$ は、それぞれの k 番目と j 番目の粒子の位置の変位ベクトルを表す。 \mathbf{n}_{kj} は \mathbf{r}_{kj} の単位ベクトルである。 $\mathbf{V}(r_{kj}, \dot{\mathbf{x}}_j)$ は j 番目の粒子が k 番目の粒子から受ける相互作用を表している。 θ_{kj} は j 番目の粒子の速度ベクトル $\dot{\mathbf{x}}_j$ と相対位置 \mathbf{r}_{kj} のなす角度である。

$f(r_{kj})$ は最適速度関数と呼ばれ、粒子との距離に応じて引力、斥力を決定する関数である。 c を変化

させることによって引力領域または斥力領域を調整することが出来る。 a は感応度を表している。

しかし、本研究では二輪駆動ロボットで実験を行うのでこのモデルでは直接実験に用いることができない。そのため、本研究ではこのモデルを離散化し、ロボットの速度ベクトルを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。 Δt 秒後の速度ベクトル $\dot{\mathbf{x}}_j(t + \Delta t)$ を離散化して求め、右 (R) と左 (L) の車輪の接地点での速度 v_R, v_L を求める。 d は車輪間距離とし、 $\Delta\theta$ は $\dot{\mathbf{x}}_j(t)$ と $\dot{\mathbf{x}}_j(t + \Delta t)$ のなす角度とする。

$\Delta\theta$ に対して、 z 成分を 0 とした三次元ベクトル $\dot{\mathbf{x}}_j(t)$ と $\dot{\mathbf{x}}_j(t + \Delta t)$ の外積ベクトル $\dot{\mathbf{x}}_j(t) \times \dot{\mathbf{x}}_j(t + \Delta t)$ の z 成分を $\dot{\mathbf{x}}_j(t) \times \dot{\mathbf{x}}_j(t + \Delta t)$ と定義し、以下で求める。

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

$$\Delta\theta = \text{sgn}(\dot{\mathbf{x}}_j(t) \times \dot{\mathbf{x}}_j(t + \Delta t)) \cos^{-1} \frac{\dot{\mathbf{x}}_j(t) \cdot \dot{\mathbf{x}}_j(t + \Delta t)}{\dot{x}_j(t) \dot{x}_j(t + \Delta t)} \quad (5)$$

$$v_R(t + \Delta t) = \dot{x}_j(t + \Delta t) + d\Delta\theta/\Delta t \quad (6)$$

$$v_L(t + \Delta t) = \dot{x}_j(t + \Delta t) - d\Delta\theta/\Delta t \quad (7)$$

3 二次元最適速度ロボット

実空間上で二次元平面上的の周期境界を表現することは難しい。そこで本研究では弾性境界のみを考慮した円形のコースを使って実験を行うことを考える。

3.1 機体構成

図 1 は本研究で用いるロボットである。ロボットはボードコンピュータ、Raspberry Pi 3 を用いて制御を行う。ロボットには大きく分けて 2 種類のセンサと 2 つのアクチュエータを用いている。1 つ目のセンサは CMUcam5 PIXY というカメラで他のロボットのいる方角と距離を検出している。2 つ目のセンサは触覚センサで壁との反射運動のために使われ、弾性境界上で約 90° 方向転換するために用いている。その 2 種類のセンサからの信号を利用してロボットは 2 つのサーボモータを動かして移動する。

3.2 他機体認識

本研究では複数のロボットの集団行動を観測する。そのためにロボットは自分とは別の機体の方角と位置を認識しなければならない。その他機体認識に使用するセンサが CMUcam5 PIXY である (図 2)。このカメラは事前に登録しておいた色の座標と大きさを Raspberry Pi 3 に送ることが出来る。そのため

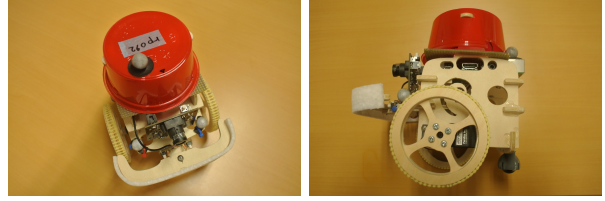


図 1: 二次元最適速度ロボット

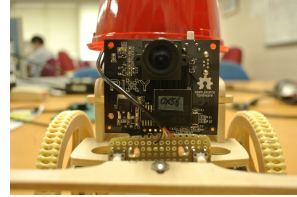


図 2: CMUcam5 PIXY

ボードコンピュータ内で画像処理をする必要はない。本研究ではこのカメラを用いてロボットの上に取り付けられた赤いカップのデータで他機体との位置と方角を認識している。他機体との距離は次の関数を用いて認識している。

$$g(x) = \frac{l}{x^n} + m \quad (8)$$

各パラメータは実際にカップを用いてカップの幅を

記号	意味
$g(x)$	距離の長さ (cm)
x	カメラに写った対象の幅 (pixel)
l, m, n	固定値

表 1: 距離認識のパラメータ

測り、図 3 のように最小二乗法を用いて求めた。この実験の結果から本研究の他機体の距離の認識に用いるパラメータは表 2 のようになった。 θ_{kj} は次の式で導出している。

$$\theta_{(\text{rad})} = (X - 160) \cdot \frac{35}{160} \cdot \frac{\pi}{180} \quad (9)$$

X はカメラの画面の左上から対象の物体画像までのピクセル数を表している。本実験で使用するカメラは横が 320pixel なので 160 との差を求めることによって θ_{kj} を導出している。35 はカメラの画角の $1/2$ である。

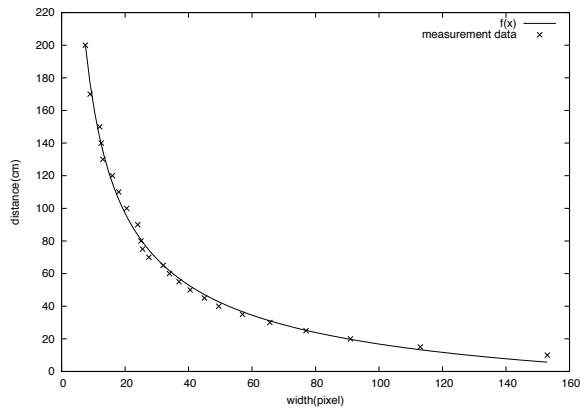


図 3: 最小二乗法を使って求めたグラフ

記号	パラメータ
l	775.928
m	-32.8268
n	0.5969

表 2: 距離認識のパラメータ

3.3 触覚センサ

本研究で弾性境界を考慮する必要があるため、ロボットは壁を検出して反射する必要がある。そのためのセンサが触覚センサである。触覚センサはタクトスイッチによって作られている (図 4 参照)。壁や他のロボットにぶつかりタクトスイッチが押されるとその信号を Raspberry Pi 3 の GPIO ポートから読み込みサーボモータへの制御に影響を与える。タクトスイッチが押されると壁と接触したスイッチと逆側に約 90° 方向転換する。

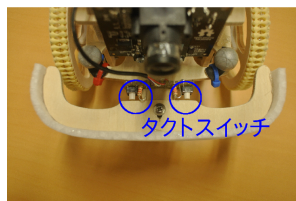


図 4: 触覚センサ

4 走行実験

本研究では図 5 のような円形のコースを用いて走行実験を行う。初期状態としてコースには 4 台のロボットをランダムに配置して実験を行う。

4.1 $V_s \neq 0$ に対する走行

実験に用いたパラメータは表 3 のとおりである。



図 5: 実験コース

内容	パラメータ
α	0.2[m/s]
β	0.5[1/m]
a	15.5[1/s]
b	2.0[m]
c	0.8
r	2.0[m]

表 3: 初期パラメータ

V_s は最適速度関数から外れた時のロボットの速度を表す。つまり、パラメータが $r=2.0\text{m}$ なので本実験では他機体が 2.0m 以上離れた時に機体は V_s によって加速度が決まる。またすべてのロボットの V_s を同一にし、このパラメータを用いると紐状体走行時にロボット同士が衝突することが観測された。実験の結果、図 6 のように先頭のロボットが $V_s=0.2\text{m/s}$ でその他のロボットが $V_s=0.15\text{m/s}$ の時、紐状の集団行動が観測された。図 7 は本実験のロボットの

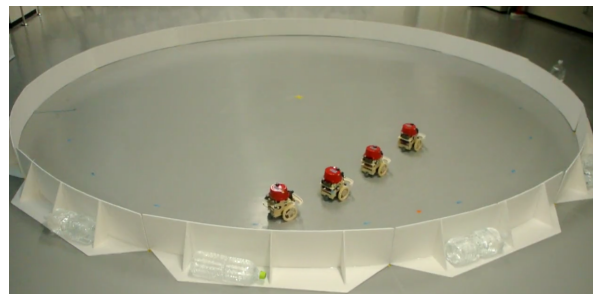


図 6: 先頭ロボット $V_s=0.2\text{m/s}$, その他 $V_s=0.15\text{m/s}$

座標のデータである。左側が集団行動が生成される

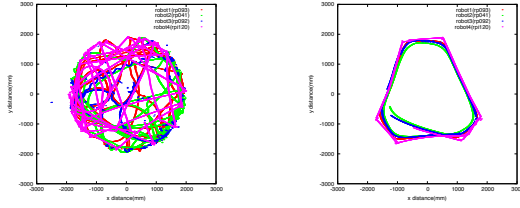


図7: $V_s \neq 0$ の場合の走行ロボットの座標軌道

前までのデータで右側が集団行動が生成された後のものである。本実験では $V_s=0.2\text{m/s}$ のロボットが偶然先頭のロボットになる集団行動が観測されるまでに5分から6分程度の時間がかかってしまった。 $V_s=0.15\text{m/s}$ のロボットが先頭となる集団行動は数秒でくずれた。

4.2 $V_s = 0$ に対する走行

そこで先頭ロボットと後続のロボットのパラメータに差を付けないように $V_s = 0$ とし他ロボットがカメラに写っていない場合、最適速度関数の最大の速度で動くようにプログラムを変更した。すなわち $r_{kj} > r$ の場合 $r_{kj} = \infty$ とした。その上でパラメータを表4のように変更し実験を行った。本実験の結果が図8である。

内容	パラメータ
α	0.1[m/s]
β	0.5[1/m]
a	2.0[1/s]
b	0.3[m]
c	1.0

表4: 実験のパラメータ

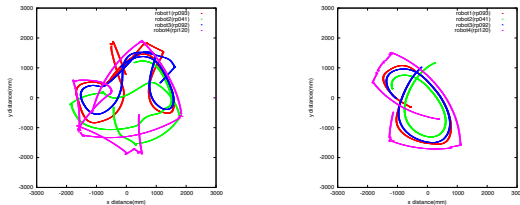


図8: $V_s=0$ の場合の座標軌道

その結果、ロボットは先頭ロボット1台とは決まっていなかったため、複数の紐状の集団行動も見られるよ

うになり、紐状走行内のロボット同士の衝突も見られなくなった。さらに紐状の集団行動が見られるまで40秒から1分ほどと $V_s \neq 0$ の実験よりも圧倒的に短い時間で紐状行動が生成された。

5 まとめと今後の課題

本研究で行った $V_s = 0$ の実験では $V_s \neq 0$ の実験よりも短い時間で紐状の集団行動が生成することが観測された。これは V_s が0以上の値をもつ場合、最適速度モデルによって生まれる速度と V_s による速度が合計されることにより後方のロボットが先頭のロボットに衝突してしまうからと考えられる。この結果を回避するために我々は一台のロボットとそれ以外のロボットの V_s を変えた実験を行った。しかし、ロボットをランダムに配置した初期状態で実験を行うと V_s の値が大きいロボットが偶然先頭にくるまで安定した紐状行動は見られず、集団行動が生成されるまでに数分間も要した。一方で、 $V_s = 0$ の実験では先頭のロボットが指定されているという制限がないため、どのロボットでも紐状行動の先頭に来ることができた。そのため、 V_s に差をつけた実験に比べて約40~60秒で安定した紐状行動が生成された。

先行研究[1]の一次元最適速度ロボットの実験では更に多くのロボットを用いて実験を行っていた。二次元最適速度ロボットも将来的にロボットの数を増やし、大規模な集団行動が見られるようにしていきたい。また、本研究で扱うパラメータの種類が多くまだ実験を行っているパラメータが少ないので他のパラメータを使っても実験を行い、安定した紐状走行の条件を明確にしたい。

参考文献

- [1] 宮島高志, 岩田耕, 佐々木康希, 本田泰, 最適速度ロボット交通流における超音波センサの影響, 第22回交通流と自己駆動粒子系シンポジウム論文集, p21-24, (2016)
- [2] 石渡龍輔, 衣川亮太, 杉山雄規, 迷路内での2次元OV粒子の集団行動 Kantorovic metric を用いた解析, 第20回交通流シュミレーションシンポジウム論文集, p59-62, (2014)
- [3] 石渡龍輔, 衣川亮太, 杉山雄規, Kantorovic 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, p41-22, (2016)
- [4] 浅田稔, 國吉康夫: ロボットインテリジェンス (2006)