

二輪車両の制御

2015SC001 足立竜也 2015SC037 各務琴音

指導教員：中島明

1 はじめに

近年、自動運転技術の進歩は目覚ましい。EV 車の導入や IT 技術の発展などから、自動車業界は 100 年に 1 度といわれる変革期を迎えている [1]。

車体の自動走行は、様々な現場において省人化に寄与する。その一例として農業用ロボットを挙げることができる。自動走行に車体の位置推定は不可欠であるが、ビニルハウスなど施設型の圃場で自律走行をさせる場合にはカメラ映像を用いられた例がある。[2] まず現在位置から目標地点までどのような軌道にしたがって動くのか決める、すなわち軌道生成をする。そしてその軌道から外れて走行しているか否かを位置推定によって把握し、外れていたら修正をしなくてはならない。

本研究では、車体として TETRIX、コンピューターとして myRIO を用いて、場所と時間を指定することで、目標まで車体を自動で走らせることを目指している。

2 使用した実機

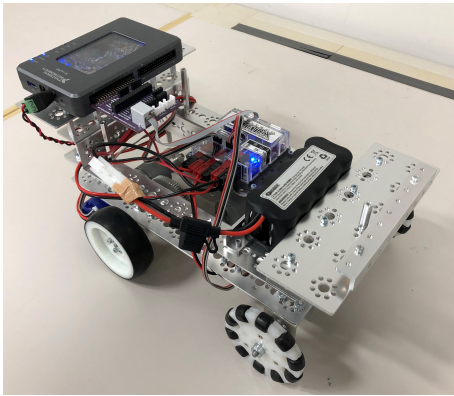


図 1 車体 (TETRIX と myRIO)

本研究では、図 1 に示した自動走行車両を用いて実験を行う。これは文献 [3] の米国の Pitsco Education 社製の TETRIX に文献 [4] の National Instruments 社製の myRIO を搭載したものである。TETRIX とは、アルミ製フレームやギアなどの多数のパーツ、モーターなどを組み立てロボットを制作するものである。myRIO とは 3 つの I/O コネクタや Wi-Fi 機能、デュアルコア ARM リアルタイムプロセッサ、カスタマイズ可能な Xilinx FPGA を備えた組込開発デバイスである [5]。

3 運動学モデルの生成

この章では文献 [6] を参考に車体の運動学モデルを導出する。

3.1 座標系

車体の位置を示す座標系を定義する。

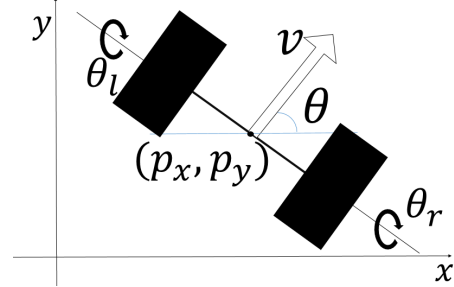


図 2 車体と座標

3.2 変数の定義

座標系から見た、車体の位置を $P = [p_x \ p_y]^T$ とする。車体の進行方向速度を v 、車体と x 軸がなす角 θ とする。右車輪の回転角を θ_r 、左車輪の回転角を θ_l とする。また、車体の各パラメータを表 1 に記す。

表 1 車体の各パラメータ

パラメータ	記号
車体の重量 [kg]	M
車輪の半径 [m]	R_w
車体の慣性モーメント [$\text{kg} \cdot \text{m}^2$]	J_c
車輪の慣性モーメント [$\text{kg} \cdot \text{m}^2$]	J_w
車輪間の間隔 [m]	T_c

3.3 関係式

車体の進行方向の速度 v 、車体の回転速度 ω 、 x 軸、 y 軸の速度ベクトル \dot{P} は以下ようになる。

$$v = \frac{1}{2}(R_w \dot{\theta}_r + R_w \dot{\theta}_l) \quad (1)$$

$$\omega = \frac{1}{T_c}(R_w \dot{\theta}_r - R_w \dot{\theta}_l) \quad (2)$$

$$\dot{P} = v \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \quad (3)$$

3.4 運動方程式

運動エネルギーを T 、ポテンシャルエネルギーを U 、ラグランジアンを L とする。車体は平面上を走るため、ポテンシャルエネルギーは $U = 0$ である。 $L = T - U$ に代入す

れば $L = T$ となるため、ラグランジアンは運動方程式と等しい。

T は以下の式で表される。

$$T = \frac{1}{2}M(\dot{p}_x^2 + \dot{p}_y^2) + \frac{1}{2}J_c\dot{\theta}^2 + \frac{1}{2}J_w\dot{\theta}_l^2 + \frac{1}{2}J_w\dot{\theta}_r^2 \quad (4)$$

一般化座標 q を $q = [p_x \ p_y \ \theta \ \theta_r \ \theta_l]^T$ とする。式 (1) ~ 式 (3) から拘束条件 $A(q)$ とし、 $A(q)\dot{q} = 0$ と表す。

$$\begin{bmatrix} 1 & 0 & 0 & -\frac{R_w}{2} \cos \theta & -\frac{R_w}{2} \cos \theta \\ 0 & 1 & 0 & -\frac{R_w}{2} \sin \theta & -\frac{R_w}{2} \sin \theta \\ 0 & 0 & 1 & -\frac{R_w}{T_c} & \frac{R_w}{T_c} \end{bmatrix} \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\theta} \\ \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} = 0 \quad (5)$$

運動方程式は以下のように表すことができる。

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right)^T = \left(\frac{\partial L}{\partial q} \right)^T + \Upsilon + A^T \lambda \quad (6)$$

3.5 拘束式の消去

$\nu = [v, \dot{\theta}]^T$ とする。 $A(q)\dot{q} = 0$ を解いて、

$$\dot{q} = S(q)\nu \quad (7)$$

にすることができる。ただし、

$$S(q) = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \\ \frac{1}{R_w} & \frac{T_c}{2R_w} \\ \frac{1}{R_w} & -\frac{T_c}{2R_w} \end{bmatrix} \quad (8)$$

である。ここで q について時間微分すると

$$\dot{q} = S(q)\nu \quad (9)$$

となる。

4 軌道の生成

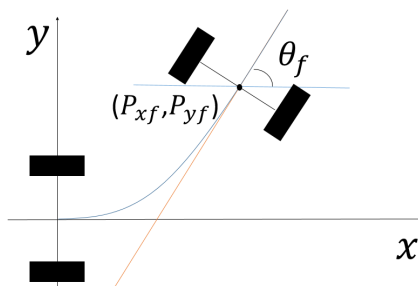


図3 軌道

図3のロボットの軌道生成について考える。 (x, y) 平面で描く軌跡 $y(x)$ を設計し、軌跡が時間経過と共にどのように変化するかを把握するために位置 $x(t)$ を求める。求

めた $y(x), x(t)$ から車体の速度、角速度を導出する。

時刻 t での車体の角度を θ_c と定義すれば、導出した速度、角速度は以下の式で表すことができる。

$$v = \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2}$$

$$\dot{\theta}_c = \frac{d}{dt} \left(\frac{dy}{dx} \right) \cos \theta_c$$

5 軌道追従

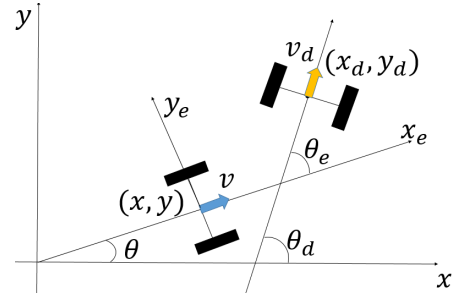


図4 ターゲットと追従するロボット

図4に2つの車体間の軌道追従が示されている。この軌道追従問題を、実機と matlab 上のシミュレーションで作成した理想の軌道に当てはめてフィードバック制御を実現する。この章では文献 [7] を参考に式を導出した。

5.1 二輪車両の運動学

(x, y) は移動ロボットの座標を表し、 θ はロボットの x 軸を基準としたロボットの回転角を表す。また、 (x_d, y_d) は目標とする移動ロボット (以降ターゲットと呼称する) の座標を表し、 θ_d は x 軸を基準としたターゲットの回転角を表す。

移動ロボットの運動学を以下の式で表す。

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} \quad (10)$$

また、ターゲットの運動学は以下ようになる。

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix} = \begin{bmatrix} \cos \theta_d & 0 \\ \sin \theta_d & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_d \\ \dot{\theta}_d \end{bmatrix} \quad (11)$$

ロボット及びターゲットは以下のような速度拘束を満たす。

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0, \quad \dot{x} \cos \theta + \dot{y} \sin \theta = v$$

$$\dot{x}_d \sin \theta_d - \dot{y}_d \cos \theta_d = 0, \quad \dot{x}_d \cos \theta_d + \dot{y}_d \sin \theta_d = v_d \quad (12)$$

制御の目的は時間経過とともに

$$x \rightarrow x_d, y \rightarrow y_d, \theta \rightarrow \theta_d \quad (13)$$

を満たすことである。

5.2 制御則の原型の導出

ロボットの中心から進行方向に l だけ離れた点に着目し座標を (x_1, y_1) とすると

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x + l \cos \theta \\ y + l \sin \theta \end{bmatrix} \quad (14)$$

と表され、同様にターゲットの中心から進行方向に l 離れた座標を (x_{d1}, y_{d1}) とすると

$$\begin{bmatrix} x_{d1} \\ y_{d1} \end{bmatrix} = \begin{bmatrix} x_d + l \cos \theta_d \\ y_d + l \sin \theta_d \end{bmatrix} \quad (15)$$

となる。ここでは制御目的として

$$x_1 \rightarrow x_{d1}, y_1 \rightarrow y_{d1} \quad (16)$$

を満たす制御則を求める。

ロボットの制御入力 $v, \dot{\theta}$ と \dot{x}_1, \dot{y}_1 の関係式は式 (14) の微分と式 (10) を用いて

$$\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \end{bmatrix} = \begin{bmatrix} \dot{x} - l\dot{\theta} \sin \theta \\ \dot{y} + l\dot{\theta} \cos \theta \end{bmatrix} \quad (17)$$

$$= \begin{bmatrix} \cos \theta & -l \sin \theta \\ \sin \theta & l \cos \theta \end{bmatrix} \begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} \quad (18)$$

$$= R \begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} \quad (19)$$

と表すことができる。ただし、

$$R = \begin{bmatrix} \cos \theta & -l \sin \theta \\ \sin \theta & l \cos \theta \end{bmatrix} \quad (20)$$

であり、任意の θ に対して

$$|R| = l > 0 \quad (21)$$

であるので、 $v, \dot{\theta}$ によって \dot{x}_1, \dot{y}_1 は操作可能である。ここで

$$\begin{bmatrix} \dot{x}_{d1} - \dot{x}_1 \\ \dot{y}_{d1} - \dot{y}_1 \end{bmatrix} \rightarrow 0 \quad (22)$$

となるような制御について考える。そこで正の定数 K を用いて、

$$\begin{bmatrix} \dot{x}_{d1} - \dot{x}_1 \\ \dot{y}_{d1} - \dot{y}_1 \end{bmatrix} = -K \begin{bmatrix} x_{d1} - x_1 \\ y_{d1} - y_1 \end{bmatrix} \quad (23)$$

式 (26) を達成する制御則として $v, \dot{\theta}$ を求める。

式 (27) を満たす制御則は以下ようになる。

$$\begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} = R^{-1} \left(\begin{bmatrix} \dot{x}_{d1} \\ \dot{y}_{d1} \end{bmatrix} + K \begin{bmatrix} x_{d1} - x_1 \\ y_{d1} - y_1 \end{bmatrix} \right) \quad (24)$$

6 シミュレーション

6.1 式の定数の決定

上記の制御則に含まれる正の定数 l, K をシミュレーションの結果から求める。 l を大きくした場合、モーターの最大速度は小さいが軌道の修正は遅くなり、 l を小さくした場合では軌道の修正が早くなるがモーターの最大速度は大きくなる。 K を大きくした場合、軌道の修正は速いが初期の x 軸方向の運動にずれが有り、 K を小さくした場合、初期の運動に問題はないが軌道の修正が遅い。

検証の結果、 $l = 0.3, K = 0.25$ の時が最適だと判断した。

6.2 matlab でのシミュレーション

入力を、25 秒後に x 軸方向に 2.00m, y 軸方向に 1.00m, 進んだ所で x 軸との角度が 30° になるように与える。

フィードバック制御を考えるため、初期位置を (0.10, 0.10), 角度を 10° ずらしてシミュレーションを行った。図 5, 図 6 にはそれぞれフィードバック制御と軌道生成から求めた車体の軌道, 車両の速度と角速度を示している。グラフの実線はフィードバック制御の結果を示し, グラフの破線は軌道生成から求めた理想の状態を示している。図から初期位置のずれは時間経過と共に補正されることが分かる

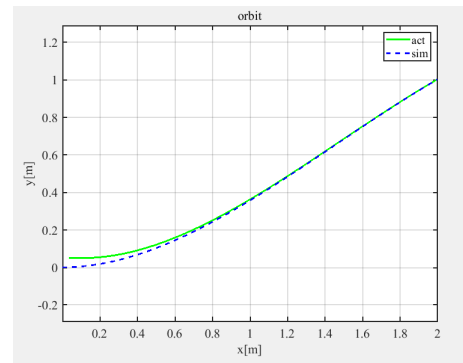


図 5 理想の軌道とフィードバック制御を用いた軌道

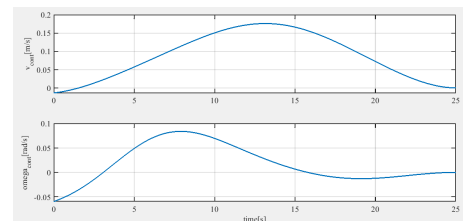


図 6 車両の速度, 角速度

7 実験結果

入力を 25 秒後に x 軸方向に 2.00[m], y 軸方向に 1.00[m], 進んだ所で x 軸との角度が 30° になるように与える。図 7~ 図 9 にはフィードバック制御を用いた場合での軌道, 速度, 角速度の時間変化を示す。フィードバック制御を用いた場合では初期位置 (0.01, 0.01) から到達点

(2.01, 0.97) へ、角度は -0.07° から 33.11° に変化した。実験結果から多少の誤差はあるものの目標を達成した運動ができていると考えられる。

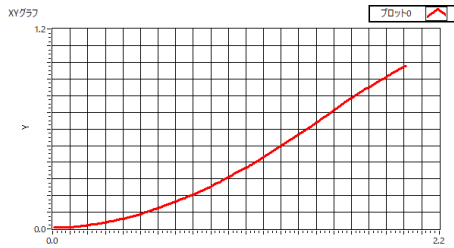


図7 フィードバック制御を用いた軌道

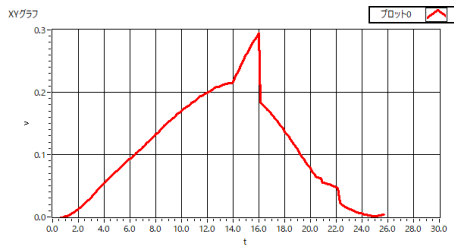


図8 車両の速度の時間変化

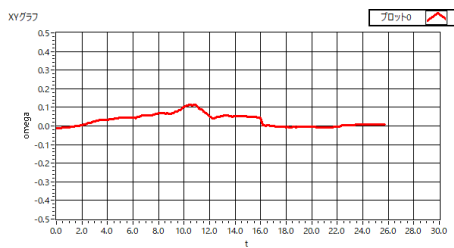


図9 車両の角速度の時間変化

図10～図12には軌道生成から求めた初期入力を用いた場合での軌道、速度、角速度の時間変化を示す。軌道生成から求めた初期入力のみを用いた場合では初期位置(0.02, 0.01)から到達点(1.89, 1.08)へ、角度は 0.08° から 32.21° に変化している。フィードバック制御を用いた場合と比較すると座標のずれは大きいが車体の傾きはずれが小さいことが分かった。

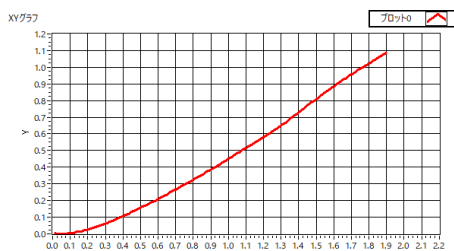


図10 初期入力のみ軌道

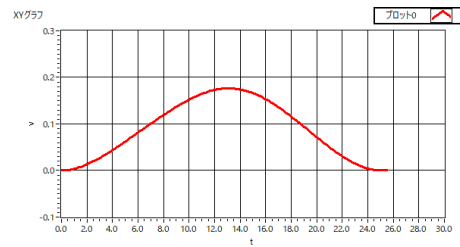


図11 車両の速度の時間変化

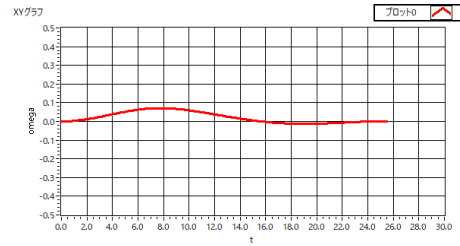


図12 車両の角速度の時間変化

8 実験の課題

マーカーの位置を計測するプログラムと車体を動かすプログラムは別にある関係上、時系列に沿ったデータを取ることが難しかった。また車体にマーカーが固定しきれていないと座標が上手く取れなかったり車体を動かした際に電源の接触が悪くなり接続が切れるなど実機の方にも問題を感じた。最終到達点での座標、角度のずれは初期位置を可能な限り(0,0)に近づけることと制御則の式の係数を調整するなどすれば今より改善出来ると思われる。

参考文献

- [1] 100年に1度の変革に挑む自動車産業。日本経済新聞(2017.11.4), 社説。
- [2] 角谷和宣, 繁田亮, 川原圭博, 浅見徹。カメラ映像による自己位置推定を用いた精密農業のための低価格自律移動ロボットの試作と評価。情報処理学会第79回全国大会, 5T-02。
- [3] 『PITSCO TETRIX MAX』。Pitsco Education 社, 米国。
- [4] 『LabVIEWで動かす myRIO × TETRIX ガイド ver.2』。株式会社アフレル, 2018。
- [5] 製品詳細『TETRIX』(株)アフレル <https://afrel.co.jp/product/tetrix>
- [6] 渡邊亮二。LEGO MIND-STORMSを用いたPID制御によるビジュアルフィードバック制御。南山大学卒業論文。
- [7] 佐藤瞳, 村松鋭一。移動ロボットの軌道追従制御。計測自動制御学会東北支部 第275回研究集会, 資料番号275-10, 2012。