

視覚を備えた移動ロボットの自己位置推定と 追従制御シミュレーション

○榊原和人 村松鋭一 渡部慶二 有我祐一 遠藤茂 (山形大学)

Simulation of Vision-based Localization and Tracking Control of Mobile Robot

* K. Sakakibara and E. Muramatsu, K. Watanabe, Y. Ariga, S. Endo (Yamagata University)

Abstract— Mobile robots which can track walking human are useful in many cases, such as carriage of heavy objects. In this paper, vision-based control of mobile robot to track a moving target is considered. We make a simulator to evaluate the control system, which generates the image of robot cameras utilizing computer graphics software.

Key Words: Tracking Control, mobile robot, vision-based Localization and Tracking control

1 はじめに

近年、作業パターンや指示を事前に入力しなくても、自ら判断して人の役に立つ仕事ができる「次世代知能ロボット」の将来性が注目され、環境内を自由に動き回る自律型の移動ロボット開発が企業や大学の研究室などで活発に行われている。このようなロボットの具体例として、「案内ロボット」や「警備ロボット」などが挙げられる。移動ロボットが環境内を自由に動き回るにはロボット自身が環境を理解、判断し行動に反映させる制御が不可欠である。環境を認識する方法は多々あるがその中で画像を用いるロボットビジョンといったカメラシステムは、1回に得られる情報量が大きいため、今後最も期待される環境センシング法の1つと考えられ、現在ではカメラを搭載したシステムの研究が増えている。そして、カメラを利用した多くの研究の中には経路探索システムや経路追従システムなどが行われている。

経路追従システムに対する基礎研究として、

(1) 現在地点から目標地点まで移動

(2) 目標対象物 (以下: ターゲット) を追跡

などがある。(1)は主にライントレースといわれ、床に引かれた線に沿って移動する。

一方で、(2)は移動ロボットが歩く人間と一定距離を保ちながら追従走行が可能となることを意味している。そのような移動ロボットは人間に代わって重い荷物を運搬する場合などにおいて、車椅子で生活

している方や重い荷物を持つことが難しい高齢者、医療器具や入院患者の食事を運ぶ看護師など福祉や医療現場において有用であると考えられる。また、上記で述べた「案内ロボット」においても、案内を行う人間のペースに合わせながら経路案内を行わなければならないため、人間と一定距離を保つような制御が必要不可欠である。そこで本研究では、カメラを搭載した移動ロボットがその視覚情報を用いて動くターゲットに追従するための制御則について考察する。カメラ画像の生成と画像処理を用いた制御のシミュレーションを行うため、コンピュータグラフィックスを用いたシミュレータを開発し、制御則の有効性を検証する。

2 シミュレータ概要

ロボットの制御アルゴリズム開発のため、シミュレータソフトを開発した。この節ではその概要を説明する。カメラに写る画像の生成と画像処理をシミュレータに取り入れるため、本研究ではOpenGLによる3次元コンピュータグラフィックスを用いた。シミュレータの画像を Fig.1 に、シミュレーションの流れを Fig.2 に示す。

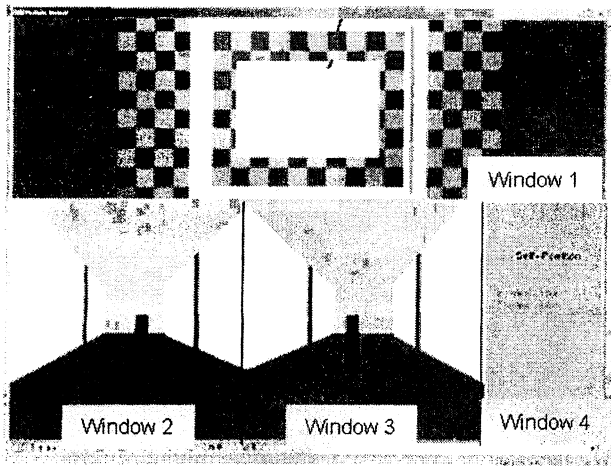


Fig.1 Simulator Screen

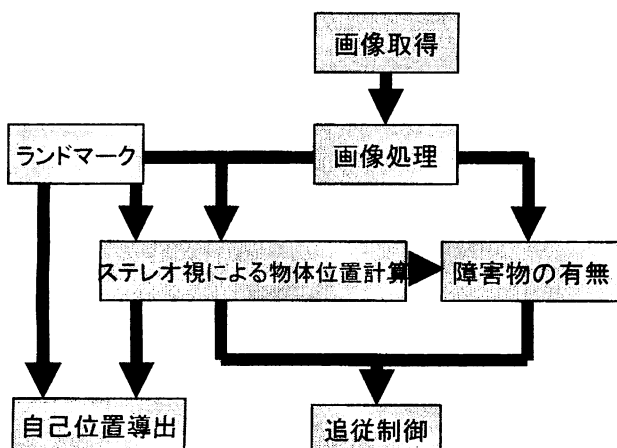


Fig.2 Operation Flow

Fig.1 はシミュレータのウィンドウである。ウィンドウは全部で4つあり、ウィンドウ上部 (Window1) は環境内を上から見た様子を表している。ウィンドウ下部の2つ (Window2, Window3) は、それぞれ移動ロボットに搭載された左右のカメラに映る画像を表している。移動ロボットモデルに関しては次節について述べる。Window4 はシミュレート中の処理に伴うデータを示す。

OpenGL では画像を生成する際、視点設定を絶対座標系の上部に設定することによって上部画像を、任意のカメラ位置に設定することにより、設定したロボットに搭載されたカメラのように画像を生成することができる。これら生成された画像に対し、画像処理を施すことは可能である。OpenGL の関数である `glReadPixels` を用いて、これによって対象画像の1ピクセル毎のRGBAの値を取得することができる。

本研究では、ターゲットの色が既知であるとし、色情報に基づく画像処理を行っている。画像処理からロボット自身やターゲットの3次元位置を求めるステレオ視を制御に利用する。

移動ロボットの走行は次節に基づき計算され、上記で述べた画像生成と画像処理、走行の運動学制御、およびデータ処理結果をループさせることにより、シミュレーションが可能となる。本研究では4節で述べる走行制御も反映させ、制御のシミュレーションを行った。

3 移動ロボットモデル

本研究で想定している移動ロボットについて説明する。移動ロボットモデルを Fig.3 に、想定しているロボットの仕様を Table.1 に示す。

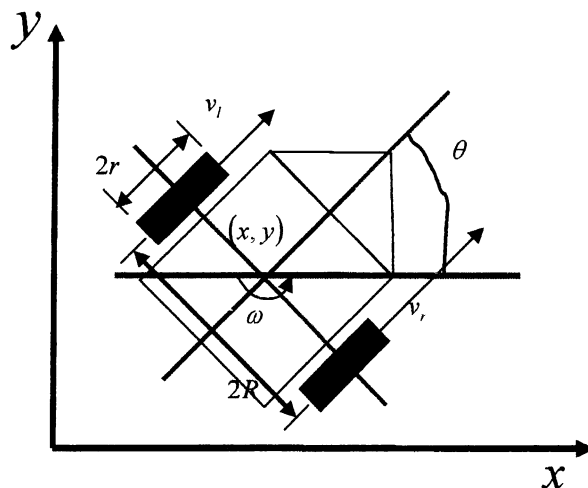


Fig.3 Mobile Robot Model

Table.1 Mobile Robot Spec

| | |
|---------------------|-------------------|
| 方式 | 2輪駆動型 |
| サイズ (幅×奥行き×高さ) [mm] | 400×400×360 |
| センサ | 外角センサ (カメラ)、内角センサ |

移動ロボットは二輪駆動型で平面内を移動する。移動ロボットの車輪半径を r [m]、2つの車輪間隔を $2R$ [m]、左右の車輪の角速度を ω_l 、 ω_r [rad/sec]、 xy 平面におけるロボットの位置 (x,y) 、 x 軸を基準と

したロボットの姿勢を θ とすると、

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r \cos \theta}{2} (\omega_r + \omega_l) \\ \frac{r \sin \theta}{2} (\omega_r + \omega_l) \\ \frac{r}{2R} (\omega_r - \omega_l) \end{bmatrix} \quad (1)$$

という関係が成り立つ。

移動ロボットには前節で述べたように2つのカメラを搭載している。これを Fig.4 に示す。カメラ画像は 50msec のサンプリング間隔で取得され、制御に用いることができるとする。

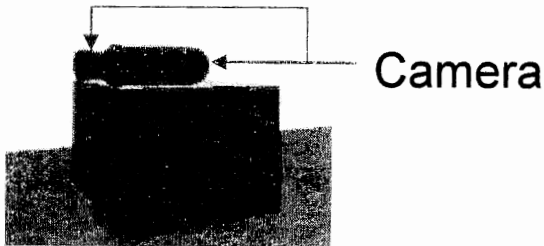


Fig.4 Simulation Mobile Robot Model

目標とする動作は、Fig.5 に示したようにターゲットとある程度の距離を保ちながら追従走行することである。

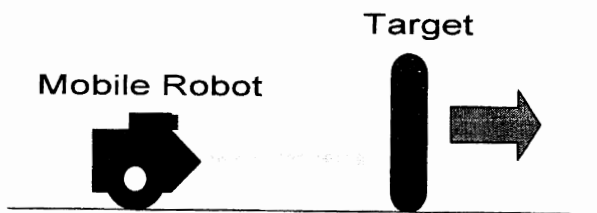


Fig.5 Tracking running

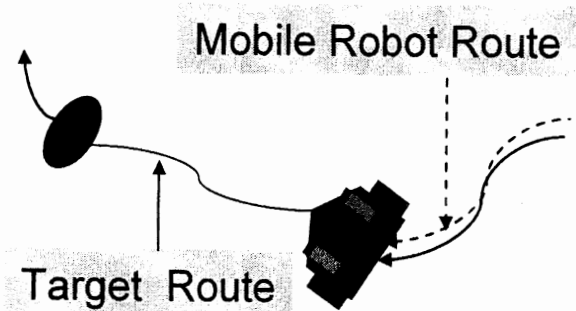


Fig.6 Running that Trace Route

本稿では、目標動作を行うために Fig.6 に示したようなターゲットが通った経路を移動ロボットが辿る走行を主として行う。

4 自己位置推定

自律移動を実現するためには、自己位置の把握が重要である。本節では自己位置の導出法について検討する。

自己位置の導出について、ランドマーク法とデッドレコニング法を用いる。

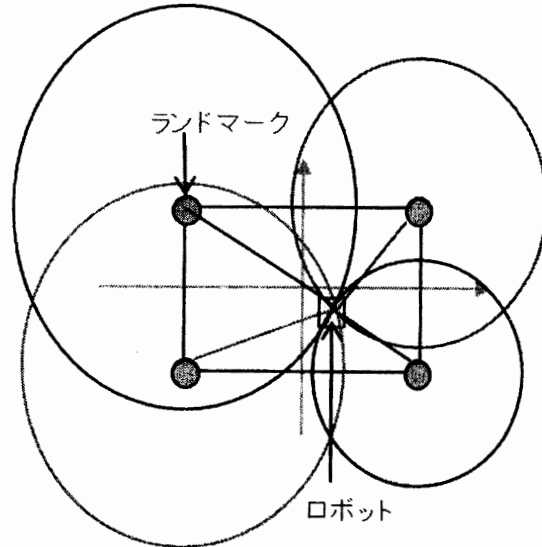


Fig.7 randmark

ランドマーク法とは、Fig.7 に示したように外界を観測して自己位置を認識する方法であり、その条件は、

1. ランドマーク座標は既知
2. 位置計算
3. 計算にはランドマークが3点必要

である。3節で述べたように、移動ロボットのカメラは前方2つのみなのでランドマークを3点以上観測するには回転を行わなければならない。そこで今回はランドマーク2点で自己位置を推定する方法を考察した。それが、Fig.8 と Fig.9 である。

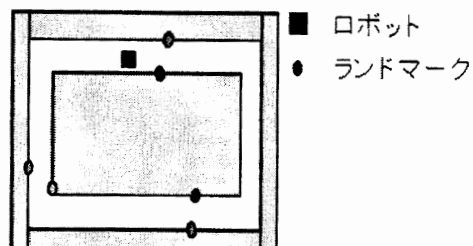


Fig.8 Randmark Arrangement

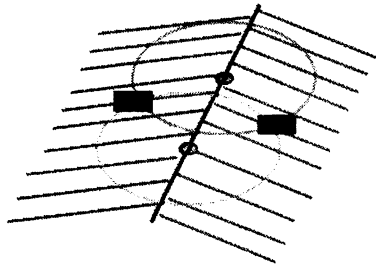


Fig.9 Area Distinction

先の条件 1, 2 は変わらないが、そのあとにランドマークの見える領域で判別を行う。問題点としては、ランドマークが観測できなければ、自己位置の導出ができない点である。

次に、デッドレコニング法とは内角センサを用いて、ロボットの車輪回転角速度を逐次計測して現在位置を推定する方法であるが、問題点としては、積分での算出となるため、移動距離が長ければ長いほど誤差が増加する点である。

そこで、2 つの方法を併用することで自己位置をより正確に行う。それが Fig.10 である。

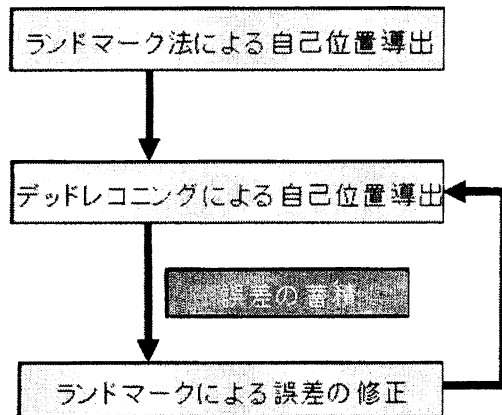


Fig.10 Localization Flow

5 追従制御則

前節、Fig.5、Fig.6 で示すように、ターゲットと距離を保ちながら追従する制御について検討する。二輪駆動による軌道追従制御則については文献[4]が知られている。

ターゲットの座標を、

$$q_r = (x_r, y_r, \theta_r)^T \quad (2)$$

移動ロボットの座標を、

$$q = (x, y, \theta)^T \quad (3)$$

追従偏差を、

$$\begin{aligned} x_e &= (x_r - x)\cos\theta + (y_r - y)\sin\theta \\ y_e &= -(x_r - x)\sin\theta + (y_r - y)\cos\theta \\ \theta_e &= \theta_r - \theta \end{aligned} \quad (4)$$

とする。制御入力は $u_1 = v$ 、 $u_2 = \dot{\theta}$ であるが、新しい入力 μ_1 、 μ_2 を導入し、

$$\mu_1 = v_r \cos\theta_e - v \quad (5)$$

$$\mu_2 = \omega_r - \dot{\theta} \quad (6)$$

と定義する。すると u_1 、 u_2 は、

$$u_1 = v_r \cos\theta_e - \mu_1 \quad (7)$$

$$u_2 = \dot{\theta}_r - \mu_2 \quad (8)$$

と表わされる。ただし、 $\dot{\theta}_r = \omega_r$ である。

また、ターゲットの速度を v_r ($v_r \neq 0$) とするとき(5)、(6)式における μ_1 、 μ_2 を $K_x > 0$ 、 $K_\theta > 0$ を用いて、

$$\mu_1 = -K_x x_e \quad (9)$$

$$\mu_2 = -K_y y_e v_r - K_\theta \sin\theta_e \quad (10)$$

と定める。(7)、(8)式に(9)、(10)を代入して得られる u_1 、 u_2 を用いた車輪角速度を、

$$\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} 1 & R \\ r & r \\ 1 & -R \\ r & r \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (11)$$

とする。(11)式によって駆動を行えば、追従誤差は 0 に漸近する。

ただし、上記のような方法をそのまま適用させるとターゲットに移動ロボットが追いついてしまい衝突を起こす。そこでターゲットの経路をバッファに記憶させておき、過去の履歴バッファの値を追従させることにより衝突を回避し、かつ一定の距離を保つことにした。過去どれだけの情報を取り出すかは、現段階においては定数としているため、結果的にターゲットの走行履歴を時間的に遅らせた経路をロボットが走行することになる。この方法はターゲットが停止した際 ($v_r = 0$) に衝突の可能性が出てくるが、これは別途衝突回避アルゴリズムを組み込むこ

とによって衝突を回避している。

6 おわりに

視覚を用いてターゲットと一定の距離を保ちながら走行する移動ロボットに関して、シミュレータの開発と走行における制御方法の検討を行った。今後の課題として、

- ① ターゲットの走行履歴バッファのより有効な利用法（データ処理、速度履歴バッファの利用など）
- ② 経路における静的、動的障害物の回避アルゴリズムの組み込み

といったものが挙げられる。

参考文献

- [1] 美多勉：非線形制御入門，昭晃堂，(2000)
- [2] 橋本、小林：OpenGLによる3次元CGアニメーション，オーム社，(2005)
- [3] 日本ロボット学会：ロボット工学ハンドブック，コロナ社，(2005)
- [4] Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi: A Stable Tracking Control Method for an Autonomous Mobile Robot, Proc. IEEE Int. Conf. on Robotics and Automation, pp.384-389, 1990.