

# 車両型ロボットの自動並走

## —自動スーツケースへの応用を目指して—

M2019SC010 坂巻和実

指導教員：大石泰章

### 1 はじめに

近年、人の歩行に追従するロボットが活用されている。例えば生産現場では、物流支援ロボット「CarriRo (キャリロ)」[1]が人の後方を追従することで重い荷物を運ぶことを可能にしている。

また、より身近な例として自動スーツケース「Ovis」[2]がある。CarriRoは人の後方を追従するが、Ovisは人の横を並走する。並走することで荷物を目視することができ、所有者に安心感を与え盗難の恐れも少なくなる。しかし、このOvisが空港内で実際に走行する動画[3]では、走行中に所有者から遠く離れてしまう現象が見られる。

本研究ではその問題を制御で改善することを考え、自動スーツケースを任意の初期状態から動作させて走行中のターゲットに並走させる制御を目指す。自動スーツケースを2輪型ロボットとしてとらえ、具体的にはレーザ測域センサを搭載したRaspberry Pi Mouseを採用する。このロボットをROS (Robot Operating System) を用いて制御する[4]。

2輪型ロボットの制御方法はいくつかあるが[5]-[6]、本研究ではKanayamaらの方法[7]を用いる。この方法を用いることで、ある速度と角速度で動く対象に実際のロボットを追従させることができる。今回はロボットを並走させるターゲットを2輪型ロボットとし、その右側の一定距離離れた点を対象としてそこに追従させることで並走を実現する。通常の人間の歩行は速度があまり変化せず角速度も零に近いはずであると考え、実験ではターゲットの速度を一定にし角速度を零とする。

本論文の構成は次の通りである。まず2章で今回解決したい課題について述べる。次に3章で課題を解決する方法について説明する。3.1節で今回使用する機器を紹介し、3.2節でセンサの詳しい説明をし、3.3節で制御方法について述べ、3.4節で実験を行い、結果とその精度について述べる。最後に4章で今後の課題について述べる。

### 2 解決したい課題

本研究では、LEGO MINDSTORMS EV3で作製した車両型ロボットをターゲットとして付属のビーコンで操作し、これに車両型ロボットRaspberry Pi Mouseを自動並走させることを目標とする。以降はLEGO MINDSTORMS EV3をターゲット、Raspberry Pi Mouseをロボットと呼ぶこととする。

具体的に解決したい課題について図1を用いて説明する。 $O-xy$ を絶対座標系とする。ターゲットの右側の端の中心から $x$ 軸正方向に15cm離れた点(図の赤の点)を点 $T$ と呼ぶ。点 $T$ の座標を $(x_T, y_T)$ とかく。一方ロボットの重心の $x$ 座標と $y$ 座標、 $x$ 軸正方向を基準とするロボットの方向角(姿勢角)を $(x_r, y_r, \theta_r)$ とかく。ターゲットは一

定の速度 $v_T$ (既知)で $y$ 軸正方向に直進し、姿勢角は常に $\pi/2$ とする。 $(x_r, y_r)$ が $(x_T, y_T)$ に収束し、 $\theta_r$ が $\pi/2$ に収束するようにロボットを制御することを解決したい課題とする。

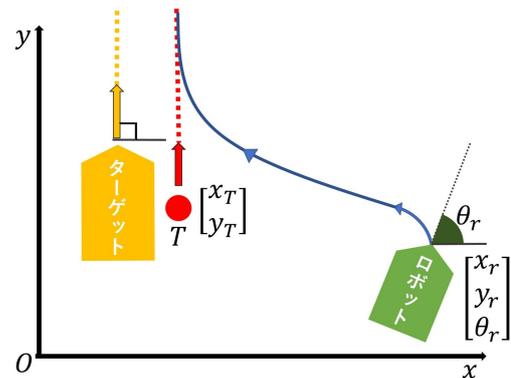


図1 解決したい課題

### 3 実現方法

#### 3.1 使用機器

LEGO MINDSTORMS EV3で作製した車両型ロボットをターゲットとし付属のビーコンで操作する(図2)。ターゲットに自動で並走することを目指すRaspberry Pi Mouseは、メインボードにRaspberry Piを使った左右独立二輪方式の車両型ロボットである。ステッピングモータを使用しており、パルス数を入力することで車輪の速度を指定することができる。ターゲットの位置を検出するためにレーザ測域センサ(北陽電機製URG-04LX-UG01)をロボットに搭載する。このセンサは距離4m、240度の範囲を約0.36度の角度分解能で走査し、ターゲットまでの距離と角度を出力する。センサをロボットに搭載したものが図3である。その大きさは、縦130mm、横100mm、高さ153mmであった。このロボットとセンサはROSを用いて制御する。



図2 使用するターゲット

図3 使用するロボット

### 3.2 周辺情報の取得

物体を認識するために、センサでロボットの周りの状況を読み取る。図4の青い円をロボットとすると、白い四角形が読み取った点を表しており、各点がロボットからの距離と角度のデータを持っている。点が集中しているところは1つの物体であると考え、その始めと終わりの点が図5の青い線の先の点である。2つの点の midpoint を物体の位置と見なす。それが、図5の黄色い線の先の点であると考え。このとき、点の集合が一定の大きさ以下の場合には物体として検出しないものとする。

実験開始直後は、ターゲットの初期位置の範囲を決めておき、その範囲内の物体をターゲットと見なす。走行時は、ターゲットの位置情報を記録しておき、過去のデータと比較してこれに近い物体をターゲットと見なす。

ターゲットの右側の点  $T$  は、センサで検出したターゲットの位置から  $x$  軸正方向に  $0.15\text{m}$  離れた点である。ロボットの方向角  $\theta_r$  がわかれば、ロボットから見てターゲットのどの方向に点  $T$  があるかがわかる。図6の  $R_O - XY$  をロボットに固定された座標系とする。ロボットの重心を原点  $R_O$ 、ロボットの進行方向を  $X$  軸、それに直交する方向を  $Y$  軸としている。このとき、センサで検出したターゲットの位置を座標系  $R_O - XY$  で  $(X_p, Y_p)$  とすると、ロボット座標系における点  $T$  の位置  $X_e = X_p + 0.15 \sin(\pi/2 - \theta_r)$ 、 $Y_e = Y_p - 0.15 \cos(\pi/2 - \theta_r)$  を求められる。

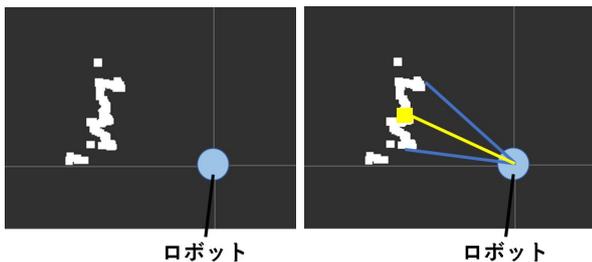


図4 センサで得られたデータの可視化 図5 物体検出の方法

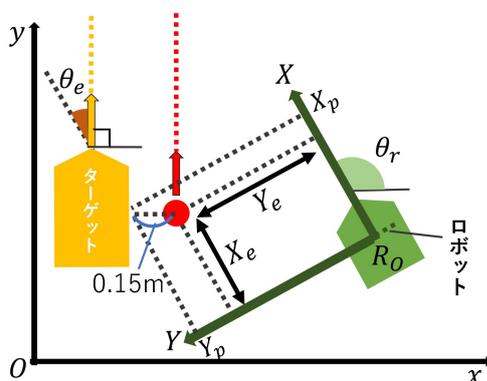


図6 ターゲットの右側の点  $T$  のロボット座標系での座標  $(X_e, Y_e)$

### 3.3 制御方法

ロボットをターゲットに自動並走させる制御方法として Kanayama らの方法 [7] を用いる。この手法は、点  $T$  とロボットの重心との座標差  $X_e, Y_e$  と方向角の誤差  $\theta_e$  を零に収束させるような、ロボットの速度  $v$  と角速度  $\omega$  を与える。  $X_e, Y_e, \theta_e$  を零に収束させるということは2章で述べた課題を解決することと同義である。この制御則では  $X_e, Y_e, \theta_e$  が零に収束することは証明されている [7]。具体的には、ロボットに入力する速度  $v$  と角速度  $\omega$  を式 (1), (2) のように定める：

$$v = v_T \cos \theta_e + K_x X_e, \quad (1)$$

$$\omega = \omega_T + v_T (K_y Y_e + K_\theta \sin \theta_e). \quad (2)$$

前節で述べたように  $X_e, Y_e$  はセンサで検出したターゲットの位置  $(X_p, Y_p)$  とロボットの方向角  $\theta_r$  を用いて計算する。また、ターゲットは  $y$  軸正方向に進むので  $\theta_e = \pi/2 - \theta_r$  である。  $\theta_r$  はロボットに入力した  $\omega$  を数値積分して求める。  $K_x, K_y, K_\theta$  は実験で決定する正の定数である。また、  $v_T$  はターゲットの速度を表しており一定で既知とする。  $\omega_T$  はターゲットの角速度を表しており零で既知とする。この速度と角速度は、一般的な人間の歩行は速度が一定で角速度が零に近いものだと考えて決定した。

### 3.4 実験

実機を用いて実験をする。このとき、ロボットが壊れないようにロボットの最大速度を  $0.4[\text{m/s}]$ 、最大角速度を  $0.8[\text{rad/s}]$  とする。ターゲットの速度は  $v_T = 0.2[\text{m/s}]$ 、角速度は  $\omega_T = 0[\text{rad/s}]$  で一定である。絶対座標系  $O - xy$  は初期時刻  $t = 0$  における点  $T$  の座標が原点になるようにとするものとする。

ロボットの初期状態である図7の  $(x_{r0}, y_{r0}, \theta_{r0})$  を様々な値を選んで実験をする。

$x_{r0} = 0.265[\text{m}]$ 、 $y_{r0} = -0.083[\text{m}]$ 、 $\theta_{r0} = \pi/4$  で実験したときのセンサの計測値が図8, 9, 10である。3つのグラフの横軸は実験の経過時間を表しており、図8のグラフの縦軸は  $X_e$ 、図9のグラフの縦軸は  $Y_e$ 、図10のグラフの縦軸は  $\theta_e$  を表している。  $\theta_e$  は読み取りやすくするために単位を  $[\text{deg}]$  にしている。このとき式 (1), (2) の係数は  $K_x = 2.0$ 、 $K_y = 11.8$ 、 $K_\theta = 6.9$  と設定している。ターゲットの走行距離は  $1.350\text{m}$  である。

センサの計測によると最終時刻では、  $X_e$  は  $-0.007[\text{m}]$ 、 $Y_e$  は  $0.019[\text{m}]$ 、 $\theta_e$  は  $-2.46[\text{deg}]$  となった。  $X_e, Y_e$  はともに零から誤差  $0.020[\text{m}]$  未満、 $\theta_e$  は零から誤差  $3.00[\text{deg}]$  未満まで近づくことができた。

また、センサで計測した  $X_e, Y_e, \theta_e$  がどこまで正確なのか検証するためにモーションキャプチャを用いて計測を行った。計測した結果が図11, 12, 13である。モーションキャプチャの計測によると最終時刻では、  $X_e$  は  $0.001[\text{m}]$ 、 $Y_e$  は  $0.027[\text{m}]$ 、 $\theta_e$  は  $-0.54[\text{deg}]$  となった。

以上の結果から、最終時刻におけるセンサの計測結果とモーションキャプチャの計測結果との誤差は  $X_e$  が  $0.008[\text{m}]$ 、 $Y_e$  が  $0.008[\text{m}]$ 、 $\theta_e$  が  $1.92[\text{deg}]$  だということがわかる。そのほかの時刻でも大きな誤差は見られなかつ

たため、ロボットがある程度の精度で走行できていることが確認できた。

本研究の目的は任意の初期状態のロボットが進行中のターゲットに並走することなので、前述の初期状態以外にも、 $x_{r0}$  を 0.120[m] から 0.420[m],  $y_{r0}$  を  $-0.420$ [m] から 0[m],  $\theta_{r0}$  を 0 から  $\pi/2$  までの範囲の中で変化させて様々な初期状態で実験をした。

ただし、 $x_{r0}$ ,  $y_{r0}$ ,  $\theta_{r0}$  の組み合わせによっては実験を行う場所の広さの制約により実験を行えなかったものもある。また、ロボットの初期状態がターゲットから遠く離れている状態 (開始後数秒の間連続で速度  $v$  が最高速度 0.4[m/s] となるような状態) であると、 $\theta_r$  を数値積分する際の誤差がたまりやすく  $Y_e$  と  $\theta_e$  が零に収束しなくなる。この問題は、ロボットにパルスカウンタを搭載し  $\theta_r$  の推定精度をあげることで改善できると考える。

上記の問題が生じないような初期状態で実験をしたところ、いずれの初期状態からも  $X_e$  は零から誤差 0.01[m] 未満、 $Y_e$  は零から誤差 0.03[m] 未満、 $\theta_e$  は零から誤差 3[deg] 未満まで近づくことができた。 $Y_e$  と  $\theta_e$  は走行時間を長くすればさらに零に近づいていくと考えられる。この計測結果はセンサで計測したものである。

以上のことから、ターゲットを認識することができ十分な走行時間がある場合は進行中のターゲットに並走することが可能だということがわかった。

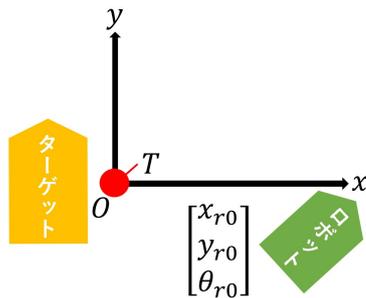


図 7 初期位置

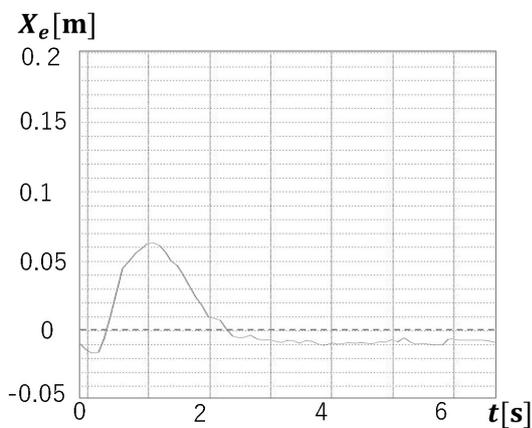


図 8 センサで計測した  $X_e$

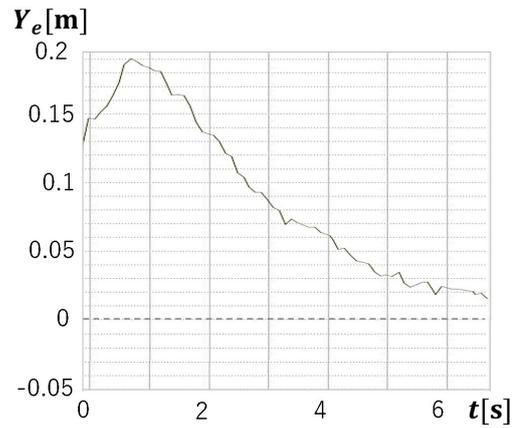


図 9 センサで計測した  $Y_e$

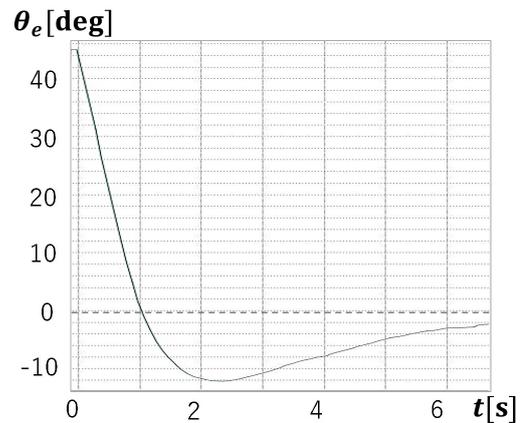


図 10 センサで計測した  $\theta_e$

#### 4 今後の課題

3章ではターゲットの速度  $v_T$  と角速度  $\omega_T$  が一定の場合のみ実験をしたが、速度と角速度が変化するような動きのターゲットにも並走できるようにすべきと考える。理論的には速度  $v_T$  と角速度  $\omega_T$  が計測できれば、計測した  $v_T$  と  $\omega_T$  を式 (1) と (2) に適用することでターゲットの動きに並走することができるはずである [7]。

$v_T$  と  $\omega_T$  を計測するためには、絶対座標系におけるターゲットの座標とロボットの座標を推定する必要がある。ロボットの絶対座標はロボットが出力した速度と角速度を数値積分して求め、ターゲットの絶対座標はセンサを使って得られるロボット座標系におけるターゲットの座標とロボットの絶対座標を用いて計算する。計算したターゲットの絶対座標の変化を用いて  $v_T$  と  $\omega_T$  を計算する。このとき、ターゲットがその場で回転する場合の  $\omega_T$  は座標の変化のみでは計算することはできないため、ターゲットの形を認識する必要がある。

ターゲットの角速度  $\omega_T$  を既知とし、センサでターゲットの速度  $v_T$  を計測して並走を試みたところ、ロボットは前進と停止を繰り返す動きをした。このときの  $v_T$  の推定値を図 14 に示す。真値である 0.2 の周辺で大きく振動しており、正しく推定できていないことがわかる。このこと

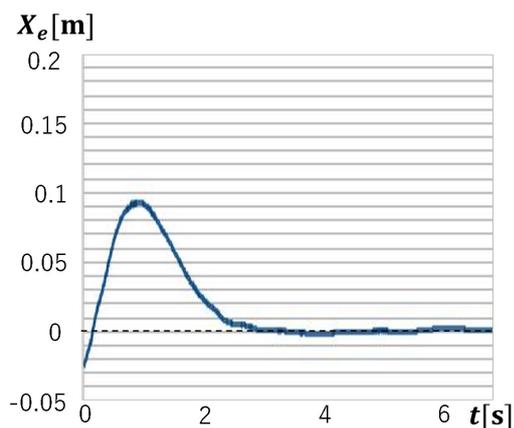


図 11 モーションキャプチャで計測した  $X_e$

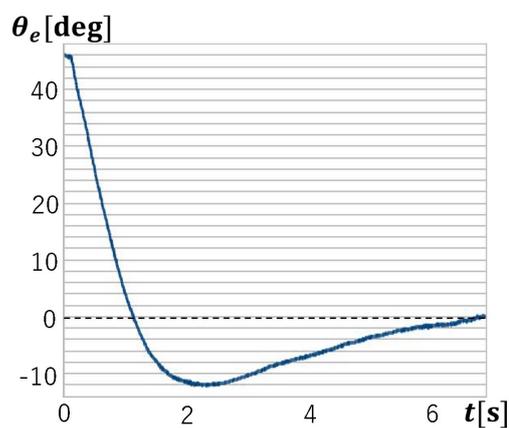


図 13 モーションキャプチャで計測した  $\theta_e$

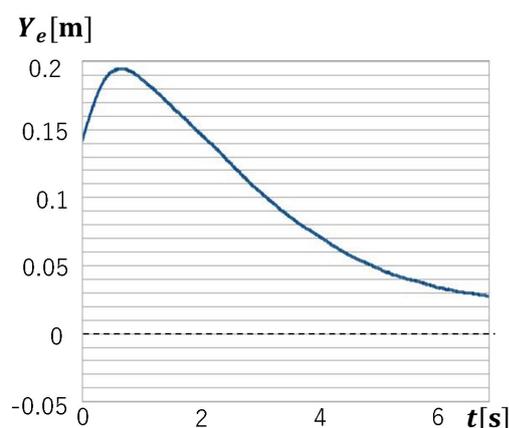


図 12 モーションキャプチャで計測した  $Y_e$

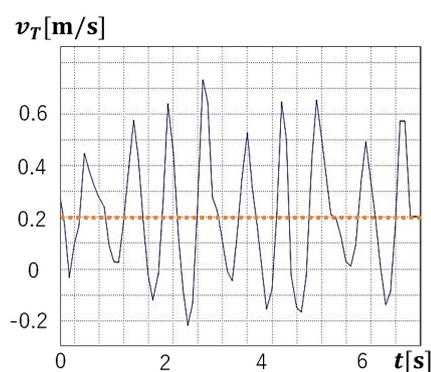


図 14 センサで計測した  $v_T$

から、 $v_T$  の推定が安定していないことがロボットが停止を繰り返す原因であると考えられる。

$v_T$  の推定をより正確にし  $\omega_T$  の推定をすることで、どのような動きのターゲットにも並走できるようにロボットを制御することが今後の課題である。

## 5 おわりに

本研究では、任意の初期状態の 2 輪型ロボットを走行中のターゲットに並走させる制御を目指した。制御方法として、ある速度と角速度で動く対象にロボットを追従させる制御則を採用した。この制御則を用いてターゲットの右側の点に追従させることで、ロボットを並走させることを実現する。

一般の人間の歩行は速度があまり変化せず角速度も零に近いはずであると考え、実験では、ターゲットの速度は一定(既知)で、角速度は零とした。実験の結果、任意の初期状態のロボットをある程度の精度で走行中のターゲットに並走させることができた。

理論的にはターゲットの速度と角速度を計測できるはずだが、今回は正確に計測することができなかった。速度と角速度を正確に計測することで、ターゲットがどのような動きをしたとしても並走できるようにすることが今後の課題である。

## 参考文献

- [1] 物流支援ロボット CarriRo(キャリロ)FD 追従モデル, <https://www.zmp.co.jp/carriro/carriro-fd> (最終閲覧日:2021 年 1 月 19 日)
- [2] Get To Know Ovis, <https://www.ovis.ai/pages/featured> (最終閲覧日:2021 年 1 月 19 日)
- [3] Ben Wang: ForwardX OVIS side follow suitcase first time usage test, <https://www.youtube.com/watch?v=BTaEZvN9g5s> (最終閲覧日:2021 年 1 月 19 日)
- [4] 上田隆一: RaspberryPi で学ぶ ROS ロボット入門, 日経 BP 社 (2017)
- [5] 深尾隆則: 非ホロノミック移動ロボットの適応制御, 計測と制御, 45-7, 602/607 (2006)
- [6] 江上正: 任意曲線経路に対する移動車両の経路制御, 計測と制御, 45-7, 595/601 (2006)
- [7] Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi: A stable tracking control method for an autonomous mobile robot, In *Proc. IEEE Int. Conf. on Robotics and Automation*, 384/389 (1990)