

# モーションキャプチャシステムを用いた小型車両の自動走行制御

2014SC025 加賀勇汰

指導教員：大石泰章

## 1 はじめに

制御工学において、制御対象に接触せずにセンシングや制御ができるということは有益である。実際、制御対象にセンサを実装する手間が省け、制御対象の複数の物理量を一つのセンサで測定でき、制御対象に影響を与えることもない。

本研究では、観測対象の位置と姿勢を非接触で測定できるモーションキャプチャシステムと、非接触で制御指令を伝えられる無線通信を使って、非接触でセンシングと制御を行う制御系の構築を目指す。具体的には、光学式モーションキャプチャシステム「OptiTrack」を用いて、マイコンボード Arduino を搭載した小型車両 Zumo の位置と姿勢を測定し、無線モジュール XBee による無線通信を行うことで、Zumo の位置制御を行う。

### 1.1 システム

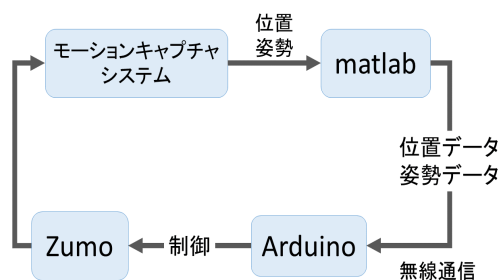


図1 システムの概要

図1は今回構築したシステムの全体像である。モーションキャプチャシステムにより算出される Zumo の位置と姿勢の情報を matlab にストリーミングし、そこから無線化したシリアル通信で位置と姿勢のデータを Zumo に搭載した Arduino へ送る。Arduino では送信された位置と姿勢のデータをもとに制御の計算処理を行い、Zumo を自動走行させる。

今回制御の計算処理は Zumo 搭載の Arduino で行った。当初はモーションキャプチャシステムからデータのストリーミングを受ける matlab 上に制御則を実装しようとしたが、この場合、matlab が制御則を実行してから Zumo が自動走行し始めるまでに数秒の時間差が生じ、リアルタイム性が失われる結果となった。そのため Arduino で制御則を組み、計算処理を行う。以下 2, 3, 4 章でシステムの個々の要素について詳述する。

## 2 モーションキャプチャシステム

Zumo の位置と姿勢のデータを取得するために NaturalPoint 社の光学式モーションキャプチャシステム「OptiTrack」を使用した [1]。このモーションキャプチャシ

テムは、対象に取り付けた球体のマーカの位置を 3 次元座標として誤差 1mm 以下の精度で測定できるシステムである。この測定には、Motive というソフトウェアを使用した。Motive により自動的に 4 点のマーカの重心座標とマーカが取り付けられた剛体の姿勢角が算出される。

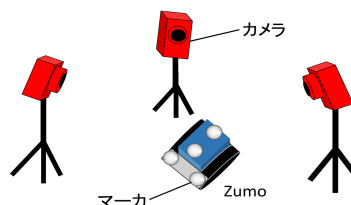


図2 モーションキャプチャシステムの実験環境

本研究では、PC に接続された 3 台のカメラ (Flex3) で Zumo を囲み、Zumo に取り付けられた 4 つのマーカの重心の位置および Zumo の姿勢角を測定する。このときのシステムの概要を図 2 に示す。今回は、高さ方向の位置座標は無視し、Zumo の位置の 2 次元座標およびヨー角を測定する。

## 3 XBee による無線通信

XBee は、無線の近距離通信規格 ZigBee に対応する無線モジュールである。この XBee には、シリーズ 1 とシリーズ 2 の 2 つのハードウェアがある。本研究では、シリーズ 2 の「XBee S2C ZigBee」を使用した。このタイプの XBee は、室内での最大通信範囲が 60m で、消費電力はシリーズ 1 の XBee よりも少なくなっている [2]。PC から USB ケーブルで接続された XBee (送信側) と Zumo に搭載した Arduino に取り付けられた XBee (受信側) の間で無線通信を行う。

## 4 小型車両 Zumo

本研究では、制御対象として小型車両 Zumo を使用する。この Zumo は制御部としてマイコンボード Arduino Uno を搭載している。

今回使用する Zumo は、10cm×10cm 未満の小型車両ロボットで、Arduino の開発環境である「Arduino IDE」に適切なプログラムを書き込み、Zumo の左右のモータを制御することで、それぞれのモータに接続された駆動輪を思い通りに動かすことができる [3]。また、Zumo には 3 軸加速度センサや 3 軸磁場センサ、3 軸ジャイロセンサが内蔵されているが、今回はこれらの内蔵センサは使用しない。

## 5 位置制御

### 5.1 問題設定

図 3 のような  $xz$  平面の原点に、Zumo を  $x$  軸正方向を向くように置き、目標点  $(x_g, z_g)$  まで自動走行させる制

御を行う。図3は Zumo が時刻  $t$  に位置  $(x, z)$  にいるときの様子であり、Zumo の正面方向が  $x$  軸正方向となす角 (Zumo のヨー角) を反時計回りを正として  $\varphi(t)[\text{deg}]$  で表す。また、Zumo と目標点との距離を  $r(t)$  で表し、Zumo を基準とした目標点の方向が  $x$  軸正方向となす角を同じく反時計回りを正として  $\varphi_g(t)[\text{deg}]$  で表す。

本研究で用いたモーションキャプチャシステムは、ヨー角  $\varphi(t)$  の値を  $-90^\circ \leq \varphi(t) \leq 90^\circ$  の範囲で与え、ある角度とそれに  $180^\circ$  を加えた角度とを区別しない。このため、目標点の  $x$  座標  $x_g$  は正であるものとする。

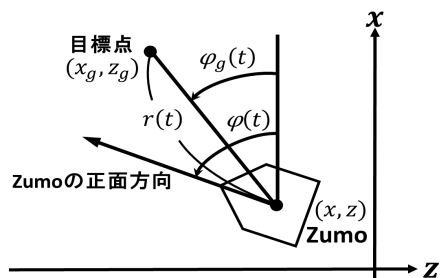


図3 Zumo の位置と角度の関係

モーションキャプチャシステムはリアルタイムで位置データ  $(x, z)$  と姿勢データ  $\varphi(t)$  を計測する。これをもとに、次節で説明する比例制御を行う。

## 5.2 比例制御

距離と角度の比例制御を組み合わせることにより、Zumo を目標点に走行させる制御を考える。Zumo の位置から目標点までの距離  $r(t)$  にゲイン  $k_1$  を掛けることで距離の比例制御を行う。試行錯誤に基づきゲイン  $k_1$  の値を 1.1 に設定した。左右のモータに与える入力  $a(t)$  を次のように定める：

$$a(t) = k_1 r(t). \quad (1)$$

次に角度の比例制御を行うことで Zumo を目標方向に向けることを考える。Zumo の左右のモータに与える入力の差を  $1/2$  倍したものを  $b(t)$  を次式のように定める。試行錯誤によりゲイン  $k_2$  の値を 7.8 に設定した：

$$b(t) = k_2 (\varphi(t) - \varphi_g(t)). \quad (2)$$

式 (1) と式 (2) を組み合わせた値を左右のモータに入力として与える。次式の  $u_r(t)$  は右モータの入力値を表し、 $u_l(t)$  は左モータの入力値を表している：

$$u_r(t) = a(t) - b(t), \quad (3)$$

$$u_l(t) = a(t) + b(t). \quad (4)$$

## 6 実験結果

5章で作成したプログラムを実装した結果を図4に示す。目標点は  $(x_g, z_g) = (400\text{mm}, 400\text{mm})$  に設定した。図4より、Zumo は  $t = 0[\text{s}]$  から弧を描くように走行し始め、 $t = 1.8[\text{s}]$  で目標点である×印に到達していることを確認できる。

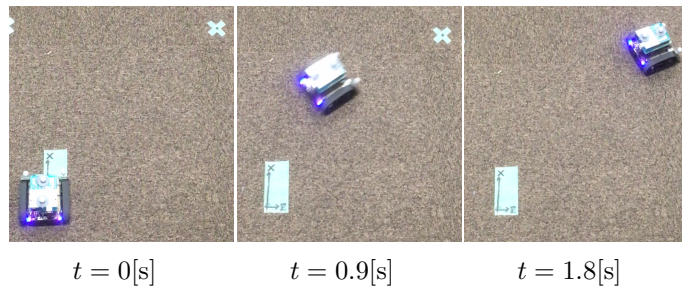


図4 実験結果

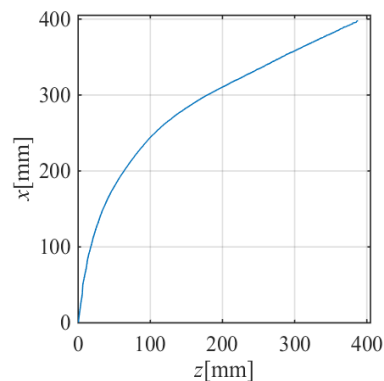


図5 取得した Zumo の位置データ

図4のときの Zumo の位置データをモーションキャプチャシステム「OptiTrack」で取得した結果を図5に示す。図5のグラフより、Zumo の停止位置と目標点との誤差が少しあるものの、十分な精度で位置制御ができていることがわかる。この誤差は、床の凹凸の影響であると考えられる。

## 7 おわりに

本研究では、モーションキャプチャシステムと無線通信を用いて、リアルタイムで Zumo の位置制御を行った。この研究は、例えば車両の自動車庫入れに応用できると考えられる。また、今回は目標点の位置を  $x_g > 0$  の場合に限定した Zumo の自動走行制御を行ったが、 $x_g < 0$  の場合には Zumo を後進させる制御を行うことで解決できると考えられる。

## 参考文献

- [1] OptiTrack  
<https://www.optitrack.co.jp/>
- [2] Robert Faludi (小林茂 監訳, 水原文 訳): 『XBee で作るワイヤレスセンサーネットワーク』. オライリー・ジャパン, 東京, 2011.
- [3] Pololu Robotics & Electronics  
<https://www.pololu.com/docs/0J57>