

# 最強巨神兵 II

川寄 洋平 (東京海洋大学)

森本 春香 (東京海洋大学)

## 1. はじめに

第一回大会では、ロボットカーの車体を自作したが、ハードウェアのトラブルが多く、大会では十分な成果を出すことができなかった。そこで、今回は車体にラジコンカーを用いることでハードウェアの製作を簡略化し、GPS 受信機による航法やフィードバック制御による軌道修正など、ロボットカーの制御プログラムに集中することにした。

## 2. ロボットカーの製作

ロボットカーの制御には、ベストテクノロジー社から販売されている ATmega128 マイコンボードを用いた。このボードは、Atmel 社製 8 ビットマイコンを搭載した小型マイコンボードである。このマイコンボードで、サーボの制御、GPS 受信機との通信、フィードバック制御など、ロボットカー制御の全てを行う。

ATmega128 は 12 ビット ADC が 8 チャンネル搭載されている。ADC には、加速度計やジャイロを接続している。これらの情報を GPS 受信機からの測位結果と複合することで、測位精度を向上させることが可能である。

GPS 受信機の製作には、u-blox 社の LEA-4T モジュールを使用した。この受信機は、一般的な NMEA フォーマットでの出力以外に、擬似距離や搬送波位相などのデータも出力することができる。また、出力頻度も最大 10Hz まで可能である。

今回製作したロボットカーを Fig. 1 に示す。

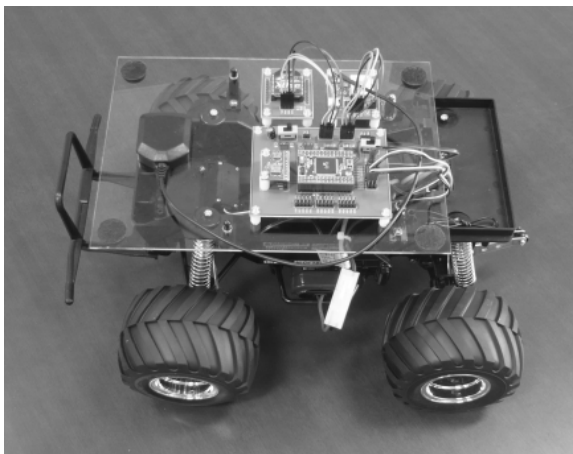


Fig. 1 ロボットカーの外観

## 3. マイコンによるサーボの制御

ラジコンカーのサーボは、PWM 信号によって制御される。サーボは、左右にほぼ 90 度ずつ回転し、その回転角は、PWM 信号のパルス幅に比例する。スピードコントローラも同様に、サーボの左右が駆動モーターの前進・後退に対応する。

一般的に、サーボに入力する PWM 信号の周期は 10ms から 20ms であり、サーボが回転角ゼロとなるパルス幅は 1.5ms 程度である。そこから  $\pm 0.5ms$  の範囲でパルス幅を変化されることで、左右の回転角を指令することができる。

ATmega128 による PWM 信号の生成には、Timer 1 を利用している。このタイマーは、10 ビットカウンタがオーバーフローすることで PWM 信号の周期を生成する。ATmega128 マイコンボードには、16MHz のクロックが搭載されているため、プリスケイラーに 256 を選択することで、PWM 信号の周期は次式で求められる。

$$16\text{MHz}/256/2^{10} = 60.04\text{Hz} \quad (1)$$

これは、約 16.4ms に相当する。

さらに、Timer 1 からは 2 つの異なるパルス幅の PWM 信号を生成することができる。これらを用いて、ロボットカーのステアリングとスピードを制御する。Eq. (1) で示されるカウンタの 1 単位は、次式で求められる。

$$16.4\text{ms}/2^{10} = 16\mu\text{s} \quad (2)$$

したがって、1.5ms のパルス幅は次式の値に対応することになる。

$$1.5\text{ms}/16\mu\text{s} \approx 94 \quad (3)$$

同様に、パルス幅を制御する範囲は  $\pm 0.5ms$  であるため、ATmega128 のプログラムでは Eq. (3) の値から  $\pm 31$  単位の範囲で、サーボの回転角を制御することになる。

PWM 信号の生成の概要を Fig. 2 に示す。

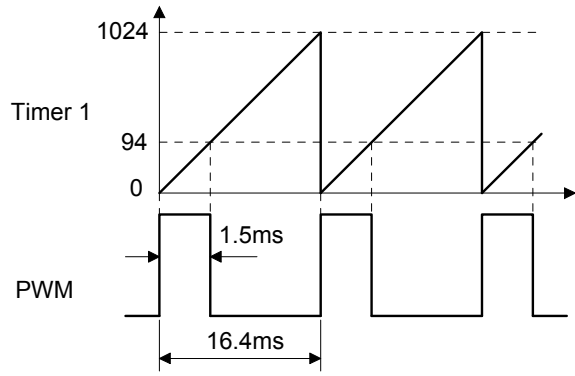


Fig. 2 PWM 信号の生成

#### 4. 局地座標変換の簡略化

マイコンの演算負荷を軽減するために、ロボットカーのナビゲーションには、簡略化した局地座標変換を用いている。

厳密な局地座標変換では、基準点の緯度 ( $\phi_0$ ) と経度 ( $\lambda_0$ ) から、次式によって ECEF 座標系から ENU 座標系に変換する回転行列が求められる<sup>(1)</sup>。

$$\mathbf{R} = \begin{bmatrix} -\sin \lambda_0 & \cos \lambda_0 & 0 \\ -\sin \phi_0 \cos \lambda_0 & -\sin \phi_0 \sin \lambda_0 & \cos \phi_0 \\ \cos \phi_0 \cos \lambda_0 & \cos \phi_0 \sin \lambda_0 & \sin \phi_0 \end{bmatrix} \quad (4)$$

さらに、GPS 受信機からはロボットカーの現在位置として緯度 ( $\phi$ )、経度 ( $\lambda$ )、高度 ( $h$ ) の情報が出力されるため、局地座標に変換するためには、これらを次式によって ECEF 座標系に変換する必要がある。

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (N+h) \cos \phi \cos \lambda \\ (N+h) \cos \phi \sin \lambda \\ (N(1-e^2)+h) \sin \phi \end{bmatrix} \quad (5)$$

ここで、 $e$  は楕円体の離心率である。また、 $N$  は次式で求められる。

$$N = \frac{a}{\sqrt{1-e^2 \sin^2 \phi}} \quad (6)$$

ここで、 $a$  は楕円体の長半径である。

このように、厳密な局地座標変換では、三角関数や浮動小数点演算が多くなり、8 ビットマイコンの

処理能力では演算負荷が大きすぎる。そこで、ロボットカーのナビゲーションでは、その移動範囲が十分に狭いものと仮定して、次式に示す簡略化された局地座標変換を行っている。

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (\lambda - \lambda_0)N \cos \phi_0 \\ (\phi - \phi_0)N \end{bmatrix} \quad (7)$$

ここで、局地座標の  $x$  軸と  $y$  軸は、それぞれ東方向と北方向を示している。また、ロボットカーの移動範囲において、高度の変化は無視できるものと仮定している。

Eq. (7)において、唯一の三角関数である  $\cos \phi_0$  は、基準点の座標から求められるため、定数として扱うことができる。したがって、局地座標への変換は、2つの加算と3つの乗算の浮動小数点演算で行うことができる。

第一回大会の走行競技で利用されたフィールドの代表的な点の座標を用いて、Eq. (4)による厳密な座標変換と Eq. (7)に示す簡略化された座標変換の比較を行った (Fig. 3)。フィールドはテニスコート3面分であり、50m×50mほどの面積となる。このとき、Eq. (7)による座標変換誤差は10cm以下であり、ロボットカーのナビゲーションには十分な精度が得られることが確認できた。

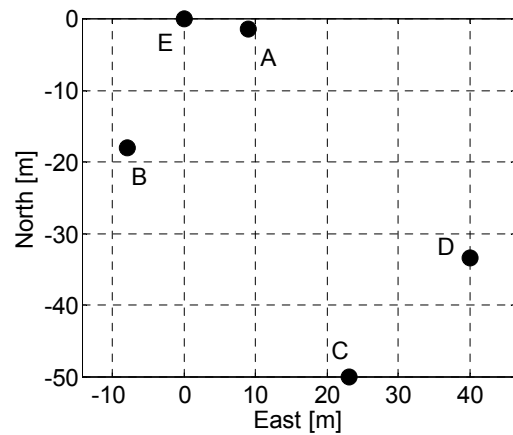


Fig. 3 走行競技フィールドの局地座標変換

#### 5. フィードバック制御による軌道修正

ロボットカーの制御では、GPS 受信機から得られる位置、速度、進行方向の情報を用いて、フィードバック制御による軌道修正を行う<sup>(2)</sup>。

Fig. 4 にロボットカーの運動モデルを示す。モデルを単純化するために、後輪で駆動し、前輪で操舵

を行う二輪モデルを用いる。

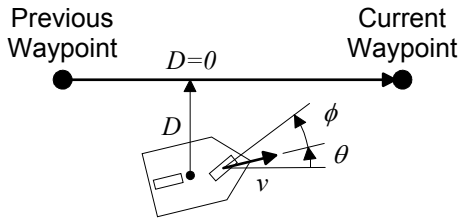


Fig. 4 ロボットカーの運動モデル

ここで、 $D$  は追従すべきラインからの偏差、 $\theta$  はラインに対するロボットカーの姿勢、 $v$  は速度、 $\phi$  は操舵角である。

ロボットカーの軌道修正では、 $v$  は一定であると仮定して、定常状態において  $D = \theta = 0$  となるよう、 $\phi$  を制御することになる。ここで、ロボットカーに働く外乱が十分に小さいものと仮定して、フィードバックによる制御系の設計には、次式で表される PD 制御を用いることにする。

$$\phi = K_1 D + K_2 \dot{D} + K_3 \theta + K_4 \dot{\theta} \quad (8)$$

ここで、 $K_1$  と  $K_3$  は比例ゲインであり、 $K_2$  と  $K_4$  は微分ゲインである。

Eq. (8)において、比例ゲインは  $D$  と  $\theta$  をゼロにするためのものである。例えば、 $D$  がゼロでないときには、ロボットカーの進行方向をラインの方向に向ける必要がある。微分ゲインは、 $D$  と  $\theta$  の速度を制御するためのものである。例えば、ロボットカーがラインに近づいて  $D$  が十分に小さくなったとしても、ロボットカーが勢い良くラインに近づいている場合（つまり、 $\theta$  が大きい場合）、ロボットの進行方向を変える必要がある。

ロボットカーの制御では、これら 4 つのパラメータを調整し、スムーズにラインを追従できるようにする。

## 6. 直線と点の最短距離

Fig. 4 において、 $\theta$  は GPS 受信機から出力される heading angle から容易に求めることができる。しかし、 $D$  の算出では、与えられたラインと任意の点の最短距離を求める必要がある。

現在向かっている waypoint の座標を  $(x_1, y_1)$ 、直前の waypoint の座標を  $(x_0, y_0)$  とすると、追従すべきラインベクトル  $(x_a, y_a)$  は次式で表される。

$$\begin{cases} x_a = x_1 - x_0 \\ y_a = y_1 - y_0 \end{cases} \quad (9)$$

ここで、点  $(x_0, y_0)$  を通り、ベクトル  $(x_a, y_a)$  に平行な直線と、任意の点  $P(x_p, y_p)$  との最短距離を求めることを考える。点  $P$  から直線へ向かうベクトルは、媒介変数  $t$  を用いて次式で表すことができる。

$$(x_a t + x_0 - x_p, y_a t + y_0 - y_p) \quad (10)$$

このベクトルと直線の方法ベクトルの内積がゼロとなる点が直行する点、すなわち最短距離となる。このときの条件は次式で表される。

$$x_a(x_a t + x_0 - x_p) + y_a(y_a t + y_0 - y_p) = 0 \quad (11)$$

これを解くことで、Eq. (10)のベクトルが最短距離となる媒介変数  $t$  は、次式で表される。

$$t = \frac{x_a(x_p - x_0) + y_a(y_p - y_0)}{x_a^2 + y_a^2} \quad (12)$$

これより、 $D$  の絶対値は次式で求めることができる。

$$|D| = \sqrt{(x_a t + x_0 - x_p)^2 + (y_a t + y_0 - y_p)^2} \quad (13)$$

Eq. (13)だけでは、 $D$  の方向までは判らないため、ロボットカーが追従すべきラインのどちら側にいるのか求める必要がある。直前の waypoint に対するロボットカーの位置  $(x_b, y_b)$  は、次式で表される。

$$\begin{cases} x_b = x_p - x_0 \\ y_b = y_p - y_0 \end{cases} \quad (14)$$

この位置ベクトルと、追従すべきラインベクトルの外積は、次式で求めることができる。

$$c = x_a y_b - x_b y_a \quad (15)$$

このとき、外積の符号から、 $D$  の方向を次式で求め

ることができる。

$$D = \begin{cases} +|D| & (c \geq 0) \\ -|D| & (c < 0) \end{cases} \quad (16)$$

## 7. シミュレーション結果

ロボットカーのフィードバック制御を、Matlab によるシミュレーションで確認した。

Fig. 4 に示すロボットカーの運動方程式は次式で表すことができる<sup>(3)</sup>。

$$\begin{bmatrix} \dot{D} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{v \sin \theta}{L} \\ \frac{v \tan \phi}{L} \end{bmatrix} \quad (17)$$

ここで、 $L$  はトレッド（軸距）である。シミュレーションでは実際のロボットカーのサイズに合わせて 20cm とした。さらに、速度は一定と仮定し、1m/s とした。

Eq. (17) に示す運動方程式に Eq. (8) の制御則を代入し、それをルンゲクッタ法により数値積分することで、フィードバック制御されたロボットカーの軌道修正をシミュレーションした。シミュレーションでは、制御則を簡略化するために、P 制御のみを用いている。

Fig. 5 にシミュレーション結果の例を示す。ここでは、 $y$  方向に 4m の初期位置誤差を与え、(10, 0) に設置された waypoint に向けてロボットカーを制御している。このときの P 制御ゲインは、 $K_1 = -0.08$ 、 $K_3 = -0.3$  である。

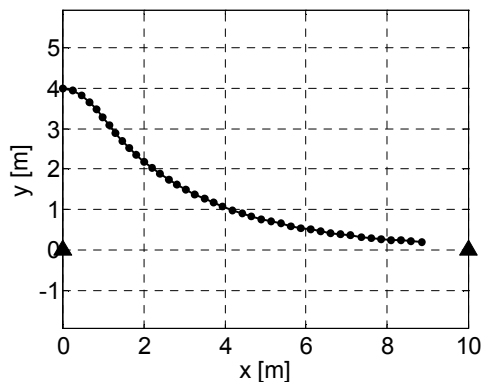


Fig. 5 安定したロボットカーの制御

Fig. 5 の結果では、ロボットカーは waypoint に向けて漸近的に収束している。しかし、制御ゲインの設定によっては、フィードバック制御系が不安定に

なり、ロボットカーが不要な旋廻動作などをしてしまうことがある。その様子を Fig. 6 に示す。

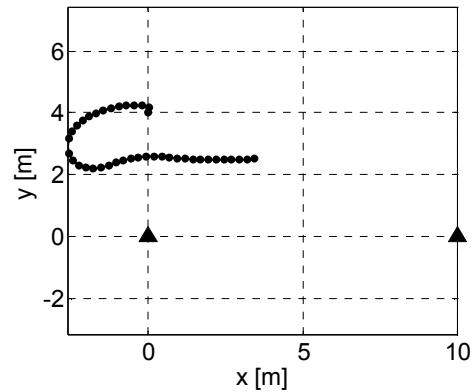


Fig. 6 不安定なロボットカーの制御

このときの P 制御ゲインは、 $K_1 = -0.5$ 、 $K_3 = -0.3$  であった。

制御則の設計では、Fig. 6 のような状態にならないように、シミュレーションにより最適なパラメータを選択する。それらを基にロボットカーの走行試験を行い、実際のロボットカーの走行状態を見ながら微調整を行う。

## 8. まとめ

今回のロボットカーコンテストでは、ラジコンカーをベースとして、マイコンによってサーボを制御するロボットカーとした。

GPS によるロボットカーのナビゲーションでは、マイコンの演算負荷を軽減するために、簡略化した局座標変換アルゴリズムを作成した。座標変換誤差は 10cm 以下であり、ロボットカーのナビゲーションには十分な精度が得られることを確認した。

ロボットカーの軌道修正には、PD 制御によるフィードバック制御を用いる。ロボットカーの運動方程式を用いたシミュレータを作成し、制御則の設計を行った。

これらの成果をロボットカー制御のマイコンに組み込むことで、自律走行を実現する。

## 参考文献

- (1) P. Misra and P. Enge, 精説 GPS: 基本概念・測位原理・信号と受信機, 正陽文庫, pp. 103-107, 2004.
- (2) 出村公成, 簡単! 実践! ロボットシミュレーション, 森北出版, pp. 134-137, 2007.
- (3) 高野政晴, ロボットの運動学, オーム社, pp. 264-270, 2004.