

知能制御PBL

第2回 RCR 中間報告

2018年6月6日

西田研究室

13104042 烏谷崇大
14104055 佐々木秀将
15104021 長田駿二郎
15104026 川崎雄太朗
15104050 坂元勇太
15104081 徳野将士
15104113 前田修一
15104117 右田明花
15104134 山福佳
17104311 横田篤紀

目次

1	目的	2
2	RCR (Robot Car Race) 2018	2
2.1	競技概要	2
2.2	コース	2
2.3	競技ルール	2
3	ROS (Robot Operating System)	3
3.1	ROS とは	3
3.2	ROS ノードとトピックの構成	4
4	機体	5
5	回路設計	7
5.1	電源回路	7
5.2	DC/DC コンバータ	8
6	フォトリフレクタ	9
6.1	使用目的	9
6.2	原理	9
6.3	動作確認実験	9
6.4	今後の実験	10
7	距離センサ	11
7.1	ToF 方式	11
7.2	I ² C	12
7.3	ToF センサの特性	12
7.3.1	特性実験方法	12
7.3.2	特性実験結果	12
8	ステアリング、速度の制御	13
8.1	RC サーボモータ	13
8.1.1	目標値生成	13
8.1.2	構造	14
8.2	DC モータ	16
8.2.1	目標値生成	16
8.2.2	制御系	16
9	物品一覧	17

1 目的

学部3年までに学習した制御理論や電気回路・情報工学の知識を使って、競技場内を自律的に走行するロボットの製作を行う。研究室で一丸となってプロジェクトを行なうことで課題を達成することの難しさや楽しさを学び、エンジニアとして仕事を進めるための素養を身に付ける。

2 RCR (Robot Car Race) 2018

2.1 競技概要

ウレタンパネルを用いてレイアウトされる周回コースにおいて、格子模様のコントロールライン手前から走行開始してコントロールラインを3回通過後に停止するまでの時間を競うロボットカー（ロボカー）を製作する。

2.2 コース

一边50 [cm] の正方形及び扇形の黒色ウレタンパネルと白黒格子模様のコントロールライン付ウレタンパネルを組み合わせ、コースを構成する。なお、競技会当日までコースは公表されない。また、コースフェンスの高さが低いため、コースフェンスのコース側に高さ10 [cm] の壁を設置する。

2.3 競技ルール

- (1) コントロールライン手前からの走行開始からコントロールラインを3回通過後に停止するまでの時間を競う。
- (2) 各チームあたり10分以内に最大3回走行し、最短時間の走行を評価する。
- (3) コース2周以上走行すること。
- (4) 競技会当日のコース試走は認めない。
- (5) ロボカーがコース周囲の壁に接触した場合は失格とする。
- (6) コース及び壁に物を設置したり、手を加えてはいけない。
- (7) コース内に足を踏み入れないこと。

3 ROS (Robot Operating System)

3.1 ROS とは

ROS(Robot Operating System) とは Open Source Robotics Foundation によって管理されているソフトウェア開発者のロボット・アプリケーション作成を支援するフレームワークである。具体的には、ハードウェア抽象化、デバイスドライバ、ライブラリ、視覚化ツール、メッセージ通信、パッケージ管理などが提供されている。つまり ROS は汎用コンピュータ向けの OS ではなく、汎用コンピュータ向け OS 上で動作するメタ OS として捉えることができる [1]。

図 1 に示すように ROS ではプロセス(実行プログラム)はノードという単位で扱い、ノード間の通信はトピックと呼ばれる“Publisher/Subscriber”モデルで実現される [2]。

これにより、プログラミング言語や通信相手さえ意識することなく簡単にプロセス間通信を実現できる。これは各ノード間のインターフェース、即ちトピックの名前と型さえ決定すればノードごとに独立して開発を行うことができるという利点もある。

以上の利点を鑑み、本研究室では ROS がインストール可能なマイコンボードである RaspberryPi3 Model B 上に ROS をインストールして開発を進めていくこととした。

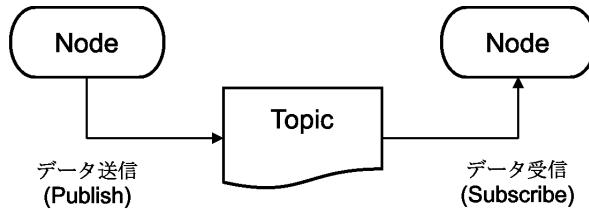


図 1: ROS ノードとトピックの概念

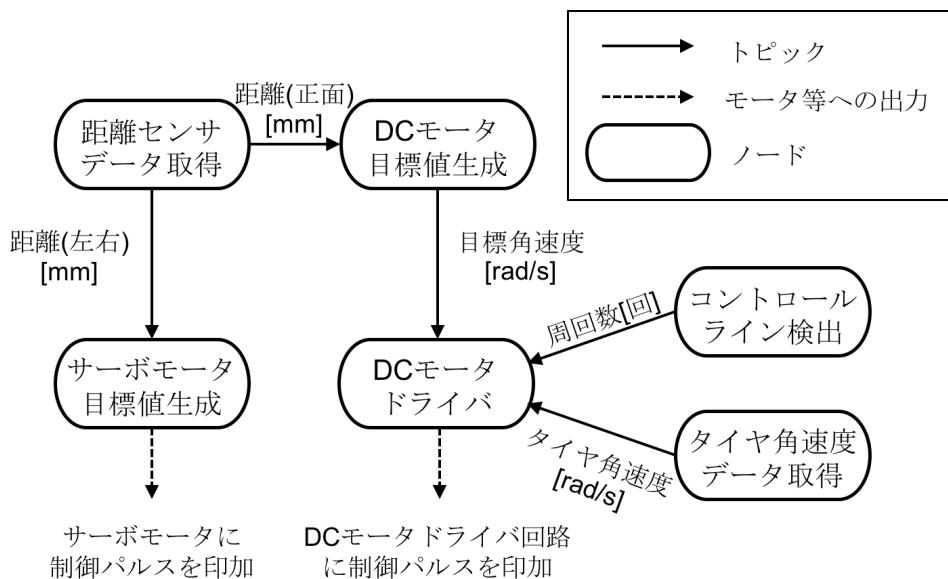


図 2: ROS ノードとトピックの構成

3.2 ROS ノードとトピックの構成

図 2 に開発する ROS ノードとトピックの構成を示す。各ノードの役割は次の通りである。

距離センサデータ取得

機体前方及び両側面に設置した距離センサからシリアルバス規格の一つである I²C を介して距離データを [mm] 単位で取得し外れ値処理や正規化を施した後に Publish する。

コントロールライン検出

機体後方下部に設置したフォトリフレクタによってコントロールラインを通過した回数をカウントし Publish する。

タイヤ角速度データ取得

回転方向は考慮しないため、ロータリーエンコーダの A 相のパルスのみをカウントし、ロータリーエンコーダの一周期あたりの出力パルス数 (500 パルス)、サンプリング周期 (0.01 [s])、ロータリーエンコーダの軸に取り付けたピニオンギアとドライブシャフトに取り付けたギア間のギア比 (2.74) を考慮して k 時点のタイヤの角速度 $\omega(k)$ を [rad/s] 単位で算出する。また、ノイズ対策として算出した角速度を式 (3.1) で表されるデジタルフィルタ (LPF に相当) に通した結果を Publish するようにしている。ただし、 α は 0 ~ 1 の範囲で定める定数である。

$$\omega(k) = \alpha\omega(k) + (1 - \alpha)\omega(k - 1) \quad (3.1)$$

DC モータ目標値生成

機体前方方向の距離データを Subscribe し、それをもとに DC モータに与える目標値を生成し Publish する。

DC モータドライバ

DC モータに与える目標値とタイヤの角速度データを Subscribe し、PI 制御則に基づき DC モータを駆動する。

サーボモータ目標値生成

機体前方の距離データを Subscribe し、それをもとにサーボモータに与える目標値を生成し、その後直接サーボモータを駆動する。また、サーボモータドライバノードが存在しない理由は詳しくは後述するが、サーボモータ内部で目標値に追従する制御系がすでに構成されているためである。

4 機体

図3に配布されたロボカーのベース部を示す。本体の前方には前輪を駆動させ機体を旋回させるためのサーボモータを設置している。そして、本体の後方には後輪を駆動させ機体を動かすためのDCモータを設置し、その上に角速度を計測するためのロータリエンコーダを設置した。

ベース部の上部に他の部品を設置するために、加工と絶縁が容易な点を鑑み、ユニバーサルプレートを二層設置した。図4、図5に搭載する部品のレイアウトを示す。図4より中央にはDCモータを制御するためのモータドライバ、後方には各部品に電力を供給するためのバッテリ、前方にはRaspberry Pi3 model Bに電力を供給するためのモバイルバッテリを設置した。そして、前方と左右にコースの壁の距離を計測するためのToFセンサを設置した。図5より、一番上にはRaspberry Pi3 model BとDC-DCコンバータを含めた電気回路を設置している。

最後に、図6にロボカーの背面の写真を示す。図6より、現在はまだ設置していないがロボカーの背面にゴールラインを読み取るためのフォトリフレクタを設置する予定である。

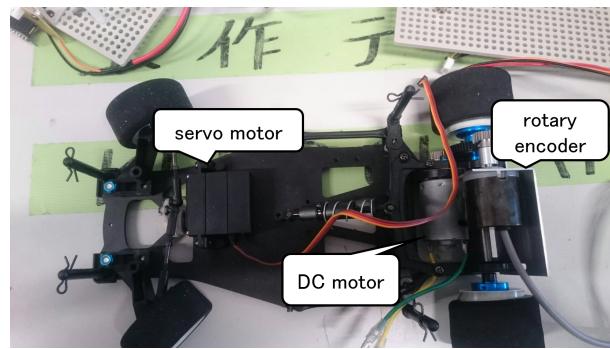


図3: ロボカー本体

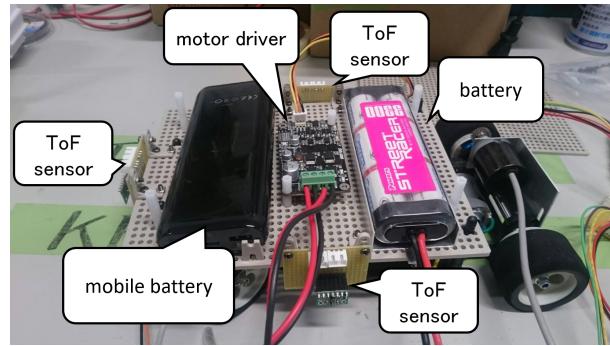


図4: ロボカー 1段目

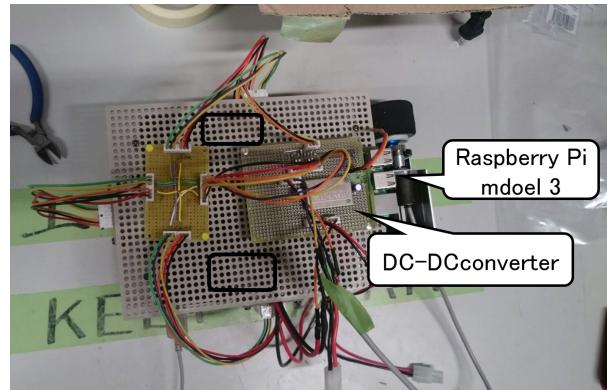


図 5: 口ボカー 2段目

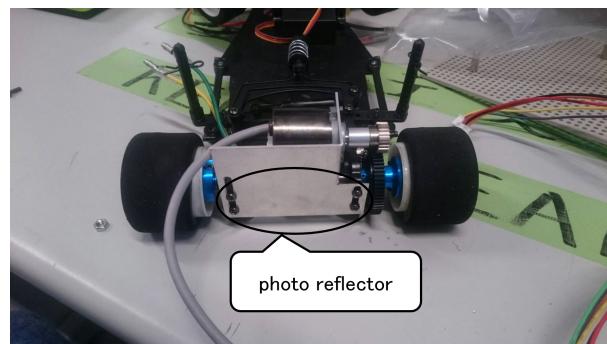


図 6: 口ボカー背面

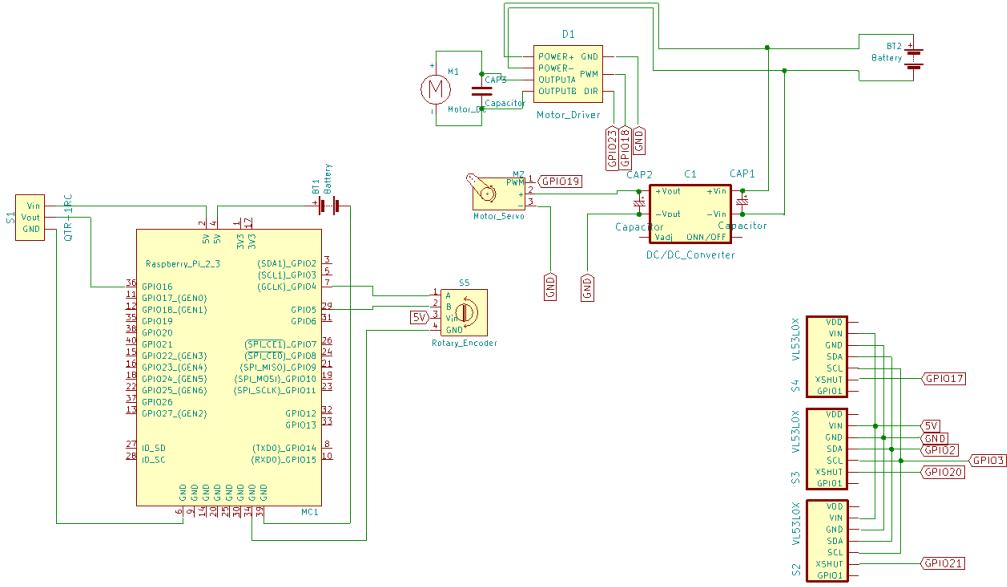


図 7: 回路図

5 回路設計

今回設計した回路を図 7、回路制作に使用した主要な部品の一覧を表 5.1 にそれぞれ示す。さらに各電子部品の仕様や構成について以下に示す。

5.1 電源回路

以下に示す仕様のように RaspberryPi3 Model B とサーボモータ、DC モータドライバではそれぞれ定格電圧が異なるため同一の電源を用いることができない。そこで 7.2 [V] バッテリと 5 [V] バッテリの 2 つを用いている。7.2 [V] バッテリはまず分流を行い、一方を DC-DC コンバータを用いて 7.2 [V] から 5 [V] に降圧してサーボモータに供給し、もう片方を DC モータドライバに供給している。5 [V] バッテリは RaspberryPi3 Model B に電源を供給している。

モータドライバ<MD10CR3>[4]

- モータ電源電圧: DC 5 [V] – 25 [V]
- 最偉大電流 : 13 [A]
- ロジック入力電圧: 3.3 [V] – 5 [V]

サーボモータ<GWS03T/2BBMG>

- 駆動電圧: DC 4.8 [V] – 6 [V]

RaspberryPi3 ModelB[3]

- 電源: 5 [V], 2.5 [A]

表 5.1: 電気回路用部品

タイプ	部品名	数	用途
マイコン	Raspberry Pi3 Model B	1	制御用
DC モータ	RP380-ST	1	後輪モータ駆動用
モータドライバ	MD10C R3	1	後輪モータ制御用
距離センサ	VL53L0X Time-of-Flight	3	距離計測用
フォトリフレクタ	QTR-1RC フォトリフレクタ・モジュール	3	ゴールライン計測用
DC-DC コンバータ	BTD05-05S200D	1	降圧用
コンデンサ	セラミックコンデンサ 1000 [pF]50 [V]	1	後輪モータのノイズ除去用
	OS コンデンサ 10 [V]47 [μF]	2	DC-DC コンバータのノイズ除去用
ロータリーエンコーダ	RE30E-500-213-1	1	ロボカーの速度計測用
バッテリ	Powers Max 4000 Ni-MH 7.2 [V]	1	DC モータ, サーボモータ用の電源
	4000 [mAh] 6CELL ニッケル水素バッテリ	1	Raspberry Pi3 ModelB 用の電源
サーボモータ	GWS03T/2BBMG	1	ステアリング用

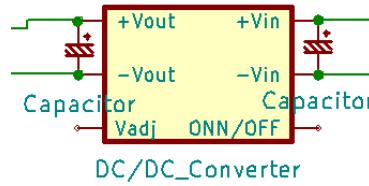


図 8: DC-DC コンバータの回路図

5.2 DC-DC コンバータ

本回路上で降圧を行うために DC-DC コンバータを用いた。以下にその仕様を示す。DC-DC コンバータは内部でデジタルスイッチングを行っているため、ノイズが多い[6]。本回路ではこのようなノイズ成分を除去するために電解コンデンサ (OS コンデンサ 10 [V]47 [μF]) を用いた図 8 の回路を作成した[5]。

DC-DC コンバータ<BTD05-05S200D>[5]

- 入力電圧: DC 4.5 [V] – 9 [V]
- 出力電圧: 0 [mA] – 2000 [mA]
- 出力電流: 3.3 [V] – 5 [V]
- 効率: 84 %

6 フォトリフレクタ

6.1 使用目的

フォトリフレクタはロボカーがコースのコントロールラインを読み取るために使用する。本研究室では図9に示す1RC Reflectance Sensorを用いた。このフォトリフレクタを採用した理由は以下で説明する原理よりAD変換器を使用せずGPIOのデジタル入出力のみで利用可能であるためである。

6.2 原理

このフォトリフレクタの内部の回路図[7]を図10に示す。はじめにRaspberry Pi mobel BのGPIOピンを出力にした後HIGHを出力し回路内のコンデンサを10マイクロ秒充電する。その後先ほどのGPIOピンを入力に変換する。

回路内のLEDが対象物に照射した光の反射光がフォトダイオードに入射することによりフォトダイオードに電流が流れようになりコンデンサの電圧が減衰する。

採用したフォトリフレクタ基板はコンデンサの電圧が一定値より大きければHIGH、小さければLOWという二値出力をするような回路が組み込まれているためパルス幅が長ければコンデンサの電圧減衰時間が長いとみなせる。

反射光の反射率が大きければコンデンサの電圧減衰時間が短くなり、反射率が小さければコンデンサの電圧減衰時間が長くなるためパルス波のHIGHの幅の長さにより反射率の大きさを測定することができる。

反射率の大きさは色によって異なるのでコンデンサの電圧減衰時間、即ちパルス幅を計測することで対象物の色を判別することができる。

仕様

- 作動電圧:5.0[V]
- 電流電源:17[mA]
- 最適な検出距離:3.0[mm]
- 最大検出距離:9.5[mm]

6.3 動作確認実験

色判別

白色の物体と黒色の物体の表面をフォトリフレクタの3mm上にかざし、物体の色を判別できるか実験した。

結果

色判別の実験において正しく白色の物体と黒色の物体を区別することができた。

6.4 今後の実験

- (1) 実際のコースとコントロールラインの色を判別するために各色のフォトトリフレクタが出力するパルス幅について閾値を設定し各色の判定をできるようにする.
- (2) ロボカーが走行しながらコントロールラインを通過した回数を判定できるようにする.

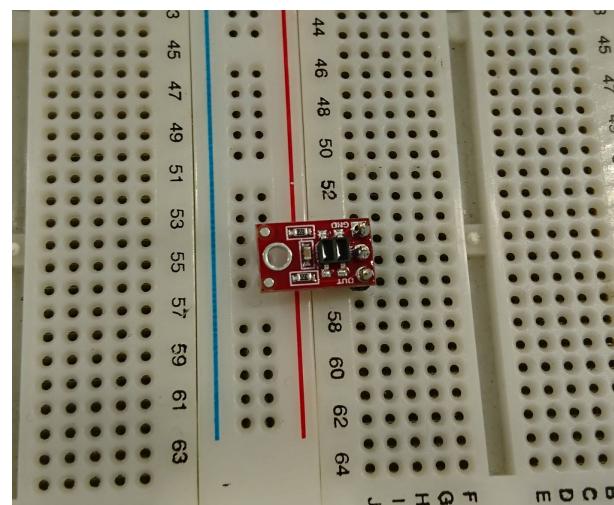


図 9: フォトトリフレクタ

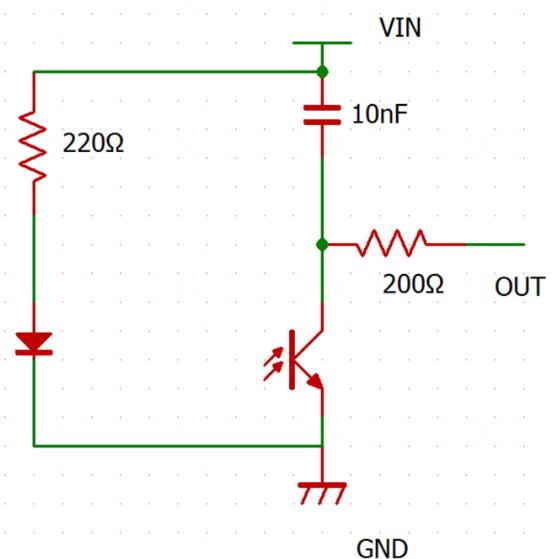


図 10: フォトトリフレクタの内部の回路図

7 距離センサ

今回、距離センサとして Pololu VL53L0X Time-of-Flight 距離センサモジュールを採用し、機体の前方に 1 個、左右に 1 個ずつの計 3 個取り付ける。この距離センサモジュールはデジタル I²C インターフェイスを介して距離を測定しており、2.8[V] のリニア・レギュレータと、入力電圧 2.6[V] から 5.5[V] までに対応する変圧回路を備えている。また、この距離センサモジュールでは赤外線を使用しており、最大 2[m] まで計測することができて分解能は 1[mm] である。図 11 に ToF センサの外観を示す。また、以下に仕様を記す [8]。

- 動作電圧: 2.6V ~ 5.5V
- 消費電力: 10mA (通常動作時の平均値)
- 検出範囲: 3cm ~ 200cm
- 重量: 0.5g (ピンヘッダを除く)
- 出力フォーマット I²C: 16 ビット読み込み (ミリメートル)
- ピンピッチ: 2.54mm

7.1 ToF 方式

ToF とは Time-of-Flight の省略形であり、反射光の時間から距離を計算するので、反射光の強度で距離を計測する一般的な測距センサと異なり、測定対象の色や表面状態に依存せずに距離を計測することが可能であり、安定して高フレームレートで測定が可能である [8]。

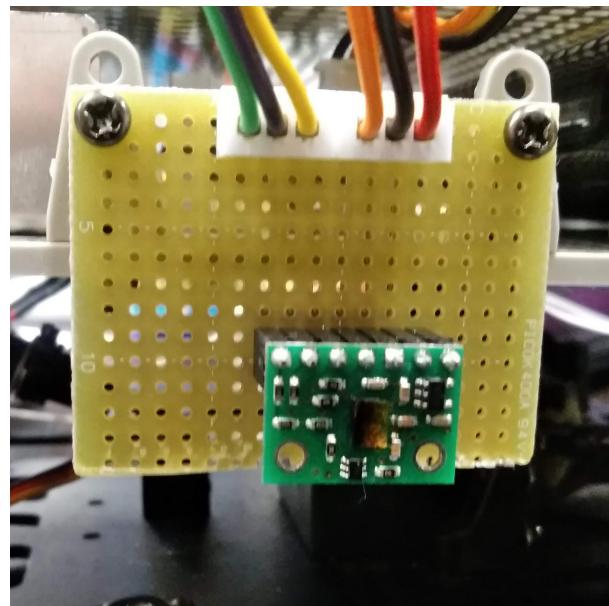


図 11: ToF センサの外観



図 12: ToF センサの実験装置の外観

7.2 I²C

ここで距離センサモジュールの中で使われている I²C(Inter-Integrated Circuit)について説明する。I²C はフィリップス社で開発されたシリアルバスである。I²C の特徴としては、バス・ラインがシリアル・データ・ライン (SDA) とシリアル・クロック・ライン (SCL) の 2 本のみという簡単な構造の双方向性バスとなっていること、バスの静電容量が 400pF 以下であれば、1 つのバス上に IC をいくつでも接続させることができることなどが挙げられる。また、今回の RCRにおいては I²C インターフェイスを介して値を受け渡すので A/D 変換を行わずに測定した距離の値を受け渡すことができるという利点がある [9]。

7.3 ToF センサの特性

7.3.1 特性実験方法

ToF センサの特性実験を行うために、図 12 のような実験装置を準備した。ToF センサとダンボールの壁が平行になるように曲尺で調整しながら、5cm ~ 70cm まで 5cm ずつ ToF センサで測距を 5 回ずつ行ってそれらの値を RaspberryPi3 Model B を通じてモニターに表示し、それらの平均をとった。

7.3.2 特性実験結果

横軸に距離の真値、縦軸に距離センサで計測した際に出力された値とし、距離センサの特性を図 13 のグラフに示す。図 13 の値より、この距離センサを最小二乗法により同定した式は次式となる。なお、 x は実距離 [mm]、 y は出力値 [mm] である。

$$y = 0.96x + 36.9 \quad (7.2)$$

式 (7.2) より、この式の逆関数を取ることによって計測した距離から実際の距離を導出することができる。よって x を計測して出力された距離 [mm]、 $f(x)$ を実際の距離 [mm] とすると次の式となる。

$$f(x) = 1.04x - 38.4 \quad (7.3)$$

よって、実際に距離センサでセンシングする際には、式 (7.3) によって計測値を実距離へ変換して用いることになる。

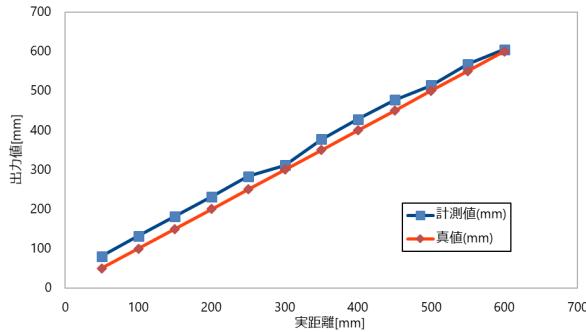


図 13: ToF センサの特性

8 ステアリング、速度の制御

ロボカーがコースを走る際、直進コースでは機体を安定させ可能な限り速く走らせ、カーブに差し掛かったときにはステアリングとステアリングのための減速が必要である。特に、カーブを曲がりきるためにどれだけのステアリング角と速度にすべきか、またその目標角度に追従させるためにはどのような制御系を構成する必要があるのかを考えなければならない。ここでは、ステアリング角と速度の目標値生成および制御方法について説明する。

8.1 RC サーボモータ

ロボカーのステアリング角は RC サーボモータの角度により決められる。すなわち、各時点での RC サーボモータへの目標角度生成と、その目標値に追従させるための制御方法を考える必要がある。

8.1.1 目標値生成

本レースで走行させるロボカーは、機体の左右側面に設置された距離センサにより左右のコースの壁との距離を検出し、その距離の差に応じてステアリング角を変更させる。例えば、右側の壁との距離が左側の壁との距離より大きければステアリング角を時計回り方向に回転させる。ロボカーが直進コースを走行するときに比べ、カーブに差しかかったときとでは図 14 のように左右の壁との距離の差が大きくなる。すなわち、この距離の差が小さいときには RC サーボモータの角度は小さくなり機体を進行方向に向かせるようにふるまい、大きくなるほど角度を大きくすることでカーブに差しかかった場合に必要なステアリング角を実現することができる。また、機体をコースの中央に位置させるようにステアリングを行わせることができる。

ここで、RC サーボモータは PWM 信号に対し図 15 のような特性をもつ。RC サーボモータでの PWM 信号の周期は 16 ~ 23 [msec] であり、その周期中のパルス幅の大きさに比例して角度が決まることがわかる。この特性より、パルス幅 W [msec] に対して目標角度 $\theta_r = f(W)$ [deg] とおく。また、左右の壁との距離の差に比例して目標角度を大きくしていくと、距離の差が小さいときに過剰なステアリングを行ってしまうことが考えられる。これが起こると、機体がコースの中央に位置し直進方向を向いているにもかかわらず小さな距離の差に反応てしまい機体の動きを不安定化させてしまう。よって、距離の差の大きさに応じてステアリング角の変化を大きくし、最大

角度を超えないようにする必要があるためシグモイド関数 $\sigma_a(x)$ を用いた次式で目標角度 θ_r を生成することにした。

$$\theta = f(W) = f(1.5 - 2 \cdot 0.6(\sigma_a(x) - 0.5)), \sigma_a(x) = \frac{1}{1 + e^{-ax}} \quad (8.4)$$

ここで、 a は定数であり、 x は機体の左右のセンサの値の差である。シグモイド関数 $\sigma_a(x)$ は、この式は左右のセンサの値の差により PWM 信号のパルス幅を変えることで目標角度 θ_r を決定している。図 15 に示したように、左右のセンサの値の差がなければ $x = 0$ [deg] となりパルス幅は 1.5 [msec] となって目標角度 θ_r は 0 [deg] となる。右のセンサの値が左のセンサの値より大きい場合を正とすれば、RC サーボモータの角度は右の壁との距離の方が大きくなると時計回りに大きくなり、左の壁との距離のほうが大きくなると反時計回りに大きくなる。定数 a はシグモイド関数の変化の速さにかかわり、試走実験を通して適切な値とする必要がある。

8.1.2 構造

RC サーボモータは内部でフィードバック制御が行われている。その構造は図 16 に示すとおりである。目標角度に対応する PWM 信号を入力すると内部で目標値に追従するように制御が行われる。よって、RC サーボモータの制御については制御系については考えず、目標角度を決めるだけとした。

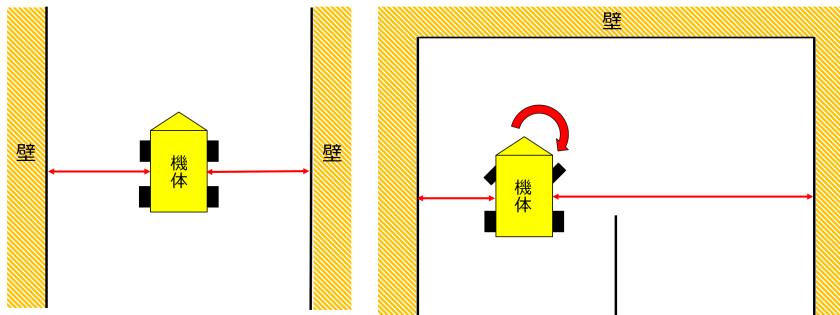


図 14: 左右の壁との距離の差によるステアリング

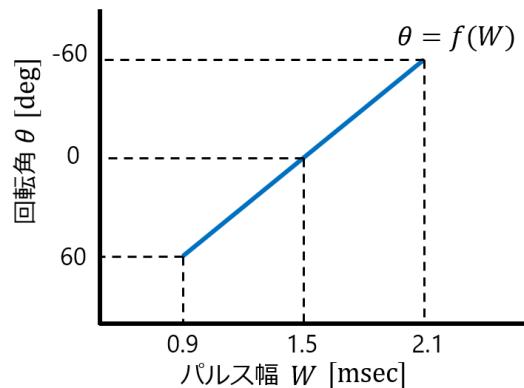


図 15: PWM 信号のパルス幅と RC サーボモータの角度の関係

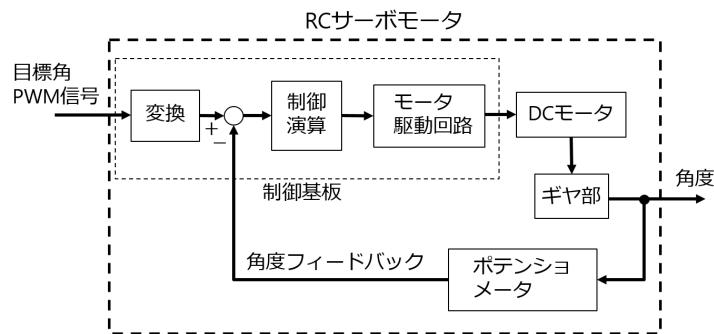


図 16: RC サーボモータの制御系

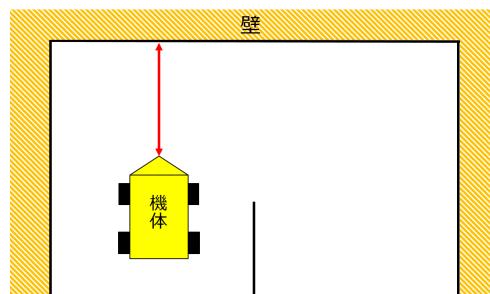


図 17: 前方の壁との距離に対する速度

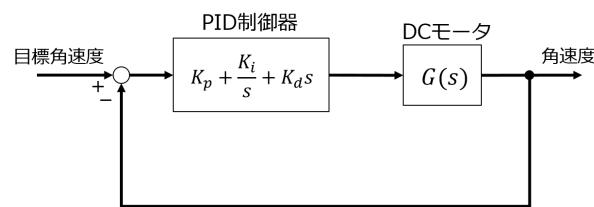


図 18: DC モータの制御系

8.2 DC モータ

ロボカーの速度は DC モータの回転角速度により決められる。すなわち、各時点での DC モータへの目標角速度生成と、その目標値に追従させるための制御方法を考える必要がある。

8.2.1 目標値生成

カーブを曲がりきるために、ステアリングに余裕をもたせる速度が必要である。本レースで走行させるロボカーは、車体の正面に設置された ToF センサにより前方の壁との距離を検出し、その距離に応じて速度を変化させる。図 17 のようにカーブに差しかかれば正面のコースの壁との距離は小さくなる。すなわち、正面のコースの壁との距離が小さくなるほど DC モータの角速度を小さくすることで、カーブでの減速を実現することができる。壁との距離に比例して目標角速度を生成すると、壁との距離が大きくなるほど DC モータの角速度が大きくなりモータの最大角速度まで大きくなってしまう。DC モータの角速度が最大の状態でロボカーを走らせると、消費電力が大きくなり電源の消耗が速くなる、機体がモータの回転に耐えられないなどの問題が生じる。この問題を解消するためには最大角速度をある値まで抑え、前方の壁との距離が一定以上になるとそれ以上角速度が上がらないようにする必要がある。なおかつ、壁との距離が小さくなれば速度を落としたいので、DC モータの角速度の目標値生成には、RC サーボモータと同様にシグモイド関数 $\sigma_a(x)$ を用いることにした。指定する最大角速度を ω_{max} [rad/s] とすれば、DC モータへの目標角速度 ω_r [rad/s] は次式で生成される。

$$\omega_r = 2\omega_{max} (\sigma_a(x) - 0.5), \sigma_a(x) = \frac{1}{1 + e^{-ax}} \quad (8.5)$$

ここで、 a は定数であり、 x は前方のセンサの値である。シグモイド関数 $\sigma_a(x)$ は前方の壁との距離に対し増減する。この式を与えれば、壁との距離がどれだけ離れても目標角速度 ω_r [rad/s] は指定最大角速度 ω_{max} [rad/s] に抑えられ、電力消費を抑えることができる。定数 a はシグモイド関数の変化の速さに関わり、試走実験を通して適切な値に決める必要がある。

8.2.2 制御系

DC モータの制御系は、図 18 に示すように PID 制御器を用いた閉ループ系で構成する。PID 制御器を用いるのは、DC モータの角速度を目標角速度へ速やかに安定して到達させるためである。 $G(s)$ は DC モータの入力電圧から出力角速度への伝達関数であり、実験により求める必要がある。これは DC モータの入力電圧に対する出力角速度の応答を同定することで求められる。また PID 制御器中の K_p, K_i, K_d は順に比例ゲイン、積分ゲイン、微分ゲインである。求めた伝達関数 $G(s)$ のボード線図から、ゲイン余裕、位相余裕などが適切な値になるように、また機体が想定通りの動きをするようにこれらのゲインを調節し、決定する必要がある。

9 物品一覧

今回の配布品、引継ぎ品、新規購入品を以下の表に示す。

表 9.2: 配布品一覧

メーカー	商品名	単価 (円)	数量	価格 (円)
GWS	GWS03T/2BBMG/J	1540	1	1540
日本電産コパル電子	ロータリエンコーダ インクリメンタル 500	7145	1	7145
TAMIYA	22T・23T AV ピニオンギア	378	1	378
TAMIYA	1/12RC トヨタ トムス84C	24624	1	24624
TAMIYA	DCモータ	1250	1	1250
	合計			34937

表 9.3: 引継ぎ品一覧

メーカー	商品名	単価 (円)	数量	価格 (円)
RSコンポーネンツ	Raspberry Pi 3 Model B	6200	1	6200
Bellnix	ISOLATED DC-DC CONVERTER BTD05-05S200D	3368	1	3368
Cytron	10Amp DC Motor Driver	1250	1	1250
Anker	Astro E4 13000mAh Portable Charger	3456	1	3456
Powers	STREET RACER 3300	5076	1	5076
Powers	Powers Max 4000 Ni-MH 7.2V 4000mAh 6CELL ニッケル水素バッテリ	4320	1	4320
TOSHIBA	microSDHCメモリーカード8GB	1058	1	1058
SANYO	OSコンデンサー 10V 47uF 10個	1300	1	1300
協和ハーモネット	耐熱電子ワイヤ	560	1	560
	合計			26588

表 9.4: 購入品一覧

メーカー	商品名	単価	数量	価格
Pololu	VL53L0X Time-of-Flight 距離センサモジュール	1879	3	5637
Pololu	QTR-1RC フォトリフレクタ・モジュール	550	1	550
Herwell Asia Limited	ブレッドボード・ジャンパーウイヤ (オス-メス) 15cm (10本入り)	220	1	220
Supertech Electronic Co., Ltd.	セラミックコンデンサ 1000pF50V	5	10	50
村田製作所	絶縁ラジアルリード型 積層セラミックコンデンサ 3.3μF50V5mm	50	10	500
エーモン工業	ギボシ端子セット	539	1	539
TAMIYA	ユニバーサルブレートL	648	3	1944
OptoSupply	高輝度5mm赤色LED 15度 OSR5CA511A-WY (10個入)	200	1	200
矢島製作所	両面スルーホール ガラスコンポジット・ユニバーサル基板 95×72mm	170	1	170
教室在庫	XHピンヘッダ	10	12	120
教室在庫	XHハウジング	10	15	150
教室在庫	アセンブリケーブル	30	29	870
	手数料			1840
			合計	12790
			残高	7210

参考文献

- [1] 表允, 倉爪亮, 渡邊裕太, ”詳説 ROS ロボットプログラミング-導入から SLAM・Gazebo・MoveIt まで-”, Kurazume Laboratory, pp.15-18, (2015).
- [2] 小倉崇, ”ROS ではじめるロボットプログラミング”, 工学社, pp.8-10, (2015).
- [3] SWITCH SCIENCE, ”10Watt BTD Series”, <<http://www.bellnix.co.jp/pdf/pdf/BTD.pdf>>, 2018 年 6 月 2 日アクセス.
- [4] Cytron technologies, ”MD10C Enhanced 10Amp DC Motor Driver User’s Manual Rev2.0 v1.0”, <<https://www.robotshop.com/media/files/PDF/user-manual-md10c-v2.pdf>>, 2018 年 6 月 2 日アクセス.
- [5] SWITCH SCIENCE, ”10Watt BTD Series”, <<http://www.bellnix.co.jp/pdf/pdf/BTD.pdf>>, 2018 年 6 月 2 日アクセス.
- [6] 後閑哲也, ”作る , できる /基礎入門 電子工作の素”, 技術評論社, pp.84-85, p186, (2009).
- [7] Pololu Corporation, ”QTR-1RC Reflectance Sensor(2-Pack)”, <<https://www.pololu.com/product/2459>>, 2018 年 6 月 1 日アクセス.
- [8] SWITCH SCIENCE, ”Pololu VL53L0X Time-of-Flight 距離センサモジュール” <<https://www.switch-science.com/catalog/2869>> 2018 年 6 月 1 日アクセス .
- [9] Philips Semiconductors, ”I²C バス仕様書バージョン 2.1” <<http://ekousaku.web.fc2.com/doc/I2C.pdf>> 2018 年 6 月 1 日アクセス .