概要

私たちのチームは、レスキューラインのすべてのタスクを完遂するために、一番重要なことは早く正確にライントレースをすることだと考えています。そのために私たちは、ラインの未来位置を予測しその位置に正確にロボットを動かすことが必要であると考え、25個のラインセンサ及びカメラによってラインをベクトルで検知することによりラインの未来位置を予測することができます。これによりメカナムホイールの運動方程式にベクトルを代入することによって正確にラインをトレースすることができます。またレスキューゾーンにおいては、被災者を効率よく回収するために、８個のレーザー距離センサを用いることで三次元の地図を作り、短時間でボールを回収することができます。そして軽く頑丈にするためにフレームは炭素繊維強化プラスチック(CFRP)を使用したロボットになっています。

このようにして私たちは制限時間内にすべてのタスクを終えることができ、故障の少ないロボットを作ることができました。

1 はじめに

A チーム

私たちのチームは、海陽学園海陽中等教育学校ロボット部レスキュー部門の５年生で構成されたチームであり、今までにロボカップジュニアワールドリーグ東海大会に二回出場しています。私たちのチームにはウェブサイトなどはありませんが、Githubのリンクを添付しておきます。

B チームメンバー

Mitsuyoshi Sugaya

ソフト、ハード、電子回路担当

Ryouta Obara

ソフト、デバッグ担当

2 プロジェクト計画

A 全体的なプロジェクト計画

A-1 コンテストの目的

私たちのチームがこのコンテストに出場する理由は、私たちがこの一年間で学んできたロボットの制御アルゴリズム、設計、電子回路などの成果をこのコンテストでロボットに組み込み、これまで学んできたことをより良いものにし、未来につなげるためです。また他のチームの良いところを学ぶことによって私たちのチームの知識を深め、自分たちの技術などを部活の後輩に伝承するためでもあります。

A-2 プロジェクトの計画

私たちのチームではプロジェクト計画が三つあります。一つ目はハードの計画です。私たちのチームではこれまでレゴマインドストームEV3で大会に出ていましたが、ルールの改変によりEV3ですべてのタスクをこなすことは困難であると考え、一から設計することにしました。設計要件として私たちは、「大きい衝撃を加えられても壊れず、メンテナンスがしやすく、軽い」という３つを目標に設計をしました。２つ目は電子回路の計画です。これまでEV3でしかRCJに出場してこなかったので、電子回路とは無縁でしたが、ロボットを一から設計するようになって電子回路を設計する必要が出てきました。初めのころは、ユニバーサル基盤にジャンパワイヤで配線をしていましたが、今回のロボットからはプリント基板を自分で設計し企業に発注することで信頼性の高い電子基板をロボットに乗せることができるようになりました。三つ目はソフトの計画です。EV3からシフトしたためにメインマイコンをArduinoにする必要がありました。ArduinoはEV3とは違いC言語でプログラミングする必要があり、まずはC言語を学ぶ必要がありました。しかしArduinoに移行することによって比較的自由なプログラムを書くことができ、プログラムの誤作動やバグを減らすことができるようになりました。

A-3 マイルストーン

私たちのチームでは次のようなマイルストーンを設定しています。

１、設計要件、システム要件の決定及び各種センサ、部品の選定

２、機体設計

３、回路設計

４、メカナムホイールの運動方程式の算出

５、カメラの機能を理解

６、プログラム設計

７、機体、回路の不具合の修正およびプログラムの各変数の調整

８、最終調整

A-4 タスクと制約の分析及びプロジェクトへの影響

2018、2019年のルールから2020年のルールに変わったことによりタスクにシーソー、レスキューキットなどが追加されレスキューゾーンの改変も行われました。これにより私たちのチームでは、EV3による全タスクの完遂は非常に難しいと考え、EV3から機体は3DCadで設計しマイコンをArduino mega、Arduino nano、esp32に変更し、EV3からのシフトにより電子回路をKiCadで設計しプリント基板をロボットに搭載しています。またレスキューキットを距離センサ、カラーセンサなどで判別することは難しいと考え、カメラを搭載することにしました。レスキューゾーンの改変については、8個のレーザー距離センサを搭載することによって三次元の地図を作り対応しています。

B 統合計画

私たちのロボットでは3個のマイコン、37個の各種センサ、9個のモータ、1個のカメラを三個のマイコンの中のメインマイコンであるArduino megaで一括制御しています。マイコン同士はI2C通信でデータを交信し、すべてのセンサの値はメインマイコンか読み取り、カメラのデータはもう一つのマイコン(Arduino nano)で必要な情報だけをピックアップした後、メインマイコンにデータを送信しています。このようにして私たちのロボットは、各マイコンやセンサがバラバラに動いているのではなく連携して動いています。

C テスト

私たちのチームではこのロボットの他にBluetooth双方向シリアル通信システムを開発しています。このシステムを私たちのロボットに搭載することによってパソコン上ですべてのセンサーの値やカメラのデータ、モーターの動きを確認することができます。これにより得られた走行、センサ、モータ、カメラのデータを分析することによって、プログラムの中の変数やモータなどの誤差修正値を正確に算出することができます。

3 ソフトウェア

A 一般的なソフトウェアアーキテクチャ

私たちのロボットに搭載されているソフトウェアは主に三つの部分に分かれています。一つ目はライントレースのプログラムです。今まではライントレースをするのにラインセンサまたはカラーセンサしか使わなかったのに対し、今回のロボットにはカメラを搭載しているので、ラインセンサ及びカメラのデータを同時に処理できるようになっています。二つ目は、レスキューキットを回収するためのプログラムです。レスキューキットの判別にはカメラを使い、回収にはジャイロセンサを用いてアームの角度を常時監視しながら慎重に回収することができます。三つめは、避難者の回収のプログラムです。レスキューゾーンでは8個のレーザー距離センサとカメラが常時動いており、ロボットの位置を常に把握しています。また3次元の地図を生成することにより、どこにどの種類のボールがあるかを常に把握しているため、効率的に被災者及び死者を回収することができます。

B 革新的なソリューション

B-1 メカナムホイールの運動方程式を応用したモーター制御関数

私たちのロボットのホイールはメカナムホイールを採用しています。メカナムホイールの詳細や、採用理由については「ハードウェア」で説明します。このメカナムホイールは前を向いたまま横方向に移動でき、超信地旋回を行えるといった戦車のような機動ができる特性を持っており、これを応用することによって全方向に移動することができます。この特徴を活用するために私たちはメカナムホイールの運動方程式を算出しました。運動方程式の各変数を右図のようにして計算すると、下のような式になります。この方程式に動きたい方向のベクトルや、回りたい角度を代入することで、直線上を進みながら90度回転するというような動きをすることができます。

B-2 ラインセンサとカメラによるラインの読み取り

これまでのライントレースで使うセンサはラインセンサだけでした。ラインセンサでライントレースを正確に行うため、僕たちのチームは今までPID制御を用いていましたが、ゲイン値の調整などがうまくいかず、まともにライントレースができていたとしても精度はあまりいいものではありませんでした。PID制御の場合、ゲイン値の調整を正確に行わなければ、ラインの未来位置の予測は困難であるため、僕たちはカメラを用いてラインの未来位置を予測するのではなく、未来位置を未来ではないようにする、つまりロボットがライントレースする前にトレースするラインをカメラで検出すればいいのではないかと考えました。このようにすれば不正確なデータでモーターを動かさなくて済むようになります。

前項で示した通りメカナムホイールを制御するためには、運動方程式に動きたい方向ベクトルを代入しなければなりません。そのために、25個のラインセンサのデータをそれぞれ分析するのでは方向ベクトルを出すのにとても複雑なアルゴリズムが必要になるため、25個のデータを最小二乗法を用いてラインを一次関数に変換し、方向ベクトルに直しています。しかし、最小二乗法で正確な値が出るときは、ロボットが交差点や鋭角のラインの上に位置していないときであり、最小二乗法は実質、数値の平均であるため、ラインが複数本検知されていると不正確な値が出力されます。カメラは、２本や３本に分かれているラインも識別でき、それぞれ方向ベクトルとして出力できるためロボットが直線や曲線のライン上に位置していない間は、カメラのデータから得られた方向ベクトルでモーターを制御します。このようにしてラインが直線や曲線の時はラインセンサとカメラ、ラインが多数存在するときにはカメラだけというような使い方をすることによって正確なライントレースをすることができます。

B-3 レーザー距離センサによる効率的な被災者の回収

私たちのロボットには、レーザー距離センサが８個搭載されています。ロボットはレスキューゾーンに入った後、一回転します。これは一回転することによって8個の距離センサで避難ゾーン、被災者、ロボットの位置を極座標を用いて三次元の地図に目標をプロットするためです。しかし、被災者と被災者が重なっているときは同じ目標として捉えられてしまうため、最低でも2回回転して地図に目標をプロットするようにしています。ロボットは、目標を回収するときは壁にボールを当てないと回収できないため、ロボット自身の位置を知ることが重要になってきます。この距離センサの分解能は1㎜なのですが、1㎜単位の地図を作ってしまうとオーバーフローしてしまうので1cm単位の地図でロボットの位置を管理しています。また、どのボールから回収すれば一番効率がいいのかという問題は、ボールの数はそこまで多いわけではないので、すべての経路の移動距離を算出し、各ルートの移動時間を分析することによって、最短ルートを通って回収できるようにしています。このようにして効率的に被災者の回収をすることができます。

B-4 プログラム実行時間の最小化

Arduino系列のマイコンをプログラムするにあたって、ArduinoIDEが指定している関数、つまりIOピンなどの入出力関数は、ArduinoのAPIを見る限りプログラムの実行時間を短くするようには作られていません。ラインセンサを1サイクル動かすにはIOピンの入出力関数を10回以上使用することになりますが、この入出力関数は私たちが期待するほど高速には動かないので、時間当たりに何サイクル動かせられるかが重要になるラインセンサのプログラムにはあまり向いていないと私たちのチームは考えました。そこで、Arduinoのレジスタを直接操作しIOピンの入出力を行うことで既存の関数よりもプログラムの実行時間を短縮することができ、時間当たりで実行できるプログラムの回数を増やすことができます。しかしレジスタのピンの番号はマイコンに搭載されているチップの種類によって異なるため、他のマイコンでレジスタを直接操作しているプログラムが動かせなくなってしまうというデメリットも存在します。このようにマイコンのレジスタを直接操作することによってプログラムの実行時間を最小化することができます。

4 ハードウェア

A ハードウェア設計の概要

私たちのロボットのハードウェア設計は「プロジェクトの計画」で示したように「大きな衝撃を加えても壊れず、メンテナンスがしやすく、軽い」ということを目標にして設計しています。しかし一つ目と三つ目の目標は矛盾しているため、私たちはロボットの主な重量を支える構造の材料を研究する必要がありました。そこで私たちは、軽く強度が高い材料として強化プラスチック（FRP）に注目しました。強化プラスチックに多くの種類がありますが、その中でも軽さと大きな耐荷重性を誇る炭素繊維強化プラスチック（CFRP）をロボットに使用することにしました。なぜならCFRPは他のFRPよりも軽いという特性があるからです。しかしながらCFRPは他のFRPよりもコストが高いというデメリットがあります。この問題を解決するために、私たちは、3DCadで設計することによって、ロボット全体の重心の位置やどこにどのような力がかかるのかなどといったことを知ることができ、正確に肉抜きをすることができます。これにより使用するCFRPの重量を削減することができるため、コストカットにつながります。右の写真はロボットの一番下の部分のCFRP板です。二つ目の目標については、3DCadで設計することにより各部品を3Dプリンターで作ることができるため、バッテリーなどを積載するスペースを広く作ることができメンテナンス性が市販の部品を使った場合よりも向上しています。このようにして私たちのロボットは三つの目標に沿った構造になっています。

また、次の写真のように私たちのロボットは主に二つの階層に分かれています。一つ目の階層はロボットを支えるためのホイールを付けたモータ4個、距離センサ4個、メインの電子基盤で構成されています。二つ目の階層は被災者やレスキューキットを回収するモジュール、サブの電子基板、4個の距離センサ、3個のジャイロセンサ、カメラで構成されています。各階層の基盤やモーターは3DCadにより設計されているため他の部品と干渉することはありません。また、CFRPで作成されている部品以外のフレームは3Dプリンタによって作成されているため、構造上の重量を分散し、重量を軽くすることができています。

B 機械設計と製造

B-1 炭素繊維強化プラスチックと3Dプリンターの利用

私たちのロボットの主なフレームにはCFRPが使われています。これはロボットを軽くし、剛性を高めるためです。CFRPはCNC加工時に発生する粉塵が発がん性物質なのですが、この粉塵が加工時に空気中に飛散しないように、CFRP板を液体中に沈めて加工することによって粉塵が出ないようにしています。またCFRP以外の部品、つまりモーターマウントなどは3Dプリンタを使用して作成しています。3Dプリンタを使用することによって、私たちが3DCadで作った部品などを短時間で作成することができるというメリットがあります。このようにして私たちのロボットのフレームや各種ジョイント、部品などはすべて私たちのチーム内で作成されています。

B-2 レスキューキットの展開およびレスキューメカニズム

私たちのロボットでは被災者のレスキューのために5個のモーターを使用しています。被災者の救助の仕方は、左上、右上、左下、右下の順番でアームを展開していきます。アームを展開した後、ロボットは距離センサを用いて壁との距離を計測し、被災者を壁に押し当てて被災者を籠の中に押し入れます。被災者を回収した後アームは先ほどと逆の動きをして元の位置に移動します。そのあとロボットは他の被災者の救助に向かいます。アームを元に戻すときのアームの位置、角度などのデータは常にジャイロセンサで読み取り監視しているため、アームについている籠は被災者を落とさないように常に地面と平行になるようにプログラムされています。被災者をリリースするときはアームの位置を左下の写真の状態のに移した後、籠を傾けることで被災者を避難ゾーンに入れることができます。またロボットは最大3人まで被災者を運ぶことができます。このようにして被災者を短時間で効率的に避難させることができます。

レスキューキットの回収は、まずカメラでレスキューキットとロボットの位置関係を把握します。そのあとロボットは距離センサを用いてレスキューキットを回収できる位置まで移動します。回収の際は、アームを被災者の救助の時と同じように伸ばします。しかしレキューキットは正方形であり、周りに押し込むことができるような壁がないため、上の二枚の写真のように籠についた爪でひっかけて籠に入れます。たとえるならアイスを食べるときのスプーンのように。（下の二枚の写真は上の写真を横から見たものです。）展開については、被災者を避難ゾーンに入れるのと変わりありません。このようにして私たちのロボットはレスキューキットを回収、展開しています。

　 私たちは、今までのような普通のタイヤでは私たちが目標とする正確なライントレースができないと考えました。そこで私たちはメカナムホイールを使うことにしました。メカナムホイールは、任意の方向に移動できるように設計されたホイールです。車輪円周上に45度の角度でローラーが取り付けられています。駆動力の伝達により、従来の車輪と同じ動きをすることに加え、フリーになっているローラーで45度の方向に移動できます。4つのモーターの回転方向と速度を調整することで、車輪の回転とローラーの動きによって全方向移動が可能です。一般的な4輪構成では、各輪が車輪が取り付けられているホイールベースの対角線にほぼ直角に力を加えます。各車輪の回転速度と回転方向を変えることで任意の方向に移動し、回転させることができます。4つのモーターを全て同じ方向に動かすと、前後に動きます。 一方の車輪を反対側の車輪と反対方向に動かすと回転運動になり、一方の車輪をもう一方の斜めの車輪と反対方向に動かすと横方向に動きます。これらの車輪の動きの組み合わせにより、ロボットの回転を伴うまたは伴わないなど、あらゆる方向の動きが実現できます。

私たちは多くのタスクをこなすためにPixy2というカメラを使用しています。Pixy2は、オブジェクト検出にカラーベースのフィルタリングアルゴリズムを使用しています。この方法は高速で効率よく、比較的安定性もあるポピュラーなものです。イメージセンサを通して各RGBピクセルの色相と彩度を計算し、それらをプライマリフィルタリングパラメータとして使用します。通常照明や露出を変化させると、オブジェクトの色相は大きく変化しませんが、カラーフィルタリングアルゴリズム上では深刻な影響を与える可能性があります。しかしながらPixy2のフィルタリングアルゴリズムは照明や露出の変化に対しても強く、安定性が高いです。このようにして私たちはPixy2を的確に利用することができます。

このロボットに搭載されている8個距離センサはレーザー距離センサです。なぜ私たちがレーザー距離センサを使用しているかというと、超音波センサに比べて正確だからです。超音波は距離に比例して拡散するため、被災者やレスキューキットなどの小さい目標を検出するのには向いてませんし、たとえ検出できたとしても、検出距離は少し曖昧といえます。しかしレーザーであれば、拡散することなく直進し分解能も１㎜と正確であるため小さな目標も検出することができます。私たちが使っているレーザー距離センサはSTマイクロ社のレーザー距離センサVL53L0Xを使用したもので、最大200㎜までの距離を計測することができます。

私たちのロボットの電子基板にはプリント基板が使われています。私たちはEV3からArduinoにシフトした後、自分で基盤を作らなければなりませんでした。最初のころは左の写真のようにユニバーサル基盤でジャンパワイヤを使って配線していました。しかしジャンパワイヤを使った配線では、ジャンパワイヤ同士のはんだ付け不良や基盤をどこかにひっかけるなど断線が数多く起こりました。またジャンパワイヤによる配線はどこにどの線がつながっているか分かりにくく、つながっていないところを見つけるのに何時間もかかります。またユニバーサル基盤はSMD部品をつけることができないため基盤の大きさが大きくなってしまうという欠点もありました。そこで私たちは基盤をパソコン上で設計し、基盤会社に発注することで結線不良などがなくコンパクトな電子基板を作ることができました。しかしプリント基板にすることで生じる問題もあります。プリント基板は間違えた配線をしていた場合修正することが難しいという点です。しかしこの問題も基盤をモータードライバ、電源回路、LEDドライバなどに分けてモジュール化することによって回避できます。このようにして私たちは信頼性の高い電子基板をロボットに搭載しています。

このロボットに搭載されているのは右の写真のメイン基板です。この基盤は主に7個のモジュールで構成されています。モジュールごとに分けることによってどこかが壊れたとしても、壊れたモジュールだけを交換することによって修理できるという利点があります。そしてロボットを作り変えたとしても各モジュールを再利用することができます。

まず初めにモータードライバです。基盤に搭載されているモータードライバは二種類あり、一種類目はTB67H450FNG、二種類目はTB6612FNGを使用したモジュールになっています。なぜモータードライバを使っているのかというと、私たちが使っているArduino megaのIOピンは一ピンあたり40mAしか電流を出力できませんがモーターは最大3Aの電流を消費するためモーターを直接ArduinoにつないでしまうとArduino が壊れる可能性があるからです。またモータードライバを中継することによってモーターの回転する速さや、回転の向きを制御することができます。モータードライバを二種類使っている理由は、ロボットのホイールにつながっているモーターと被災者やレスキューキットを回収するモーターでは要求される電流の大きさが違うためです。また、モータードライバを使うことによってArduino などのマイコンを過電流から守ることもできます。前のロボットでは市販のモータードライバを使っていましたが、プリント基板を自分で設計するようになり、モーターに相性のいいモータードライバを使ったモータードライバモジュールを作ることができるようになりました。

二つ目にラインセンサです。ラインセンサは25個のフォトトランジスタ、16個の白色LED、4個のRGBLEDで構成されています。ラインセンサはメイン基盤の裏側に直接ピンヘッダで接続されています。なぜなら25個ものデータとLEDの信号線をコードでつなぐとなるととても多くのコードが必要な上に、コードのコネクタをつける面積がメイン基板の面積を圧迫してしまします。またLEDを制御するLEDドライバモジュールもメイン基板に搭載されており、16個のLEDを一括操作することができます。フォトトランジスタのうち12個はアナログで読み、後の13個はデジタルで読んでいます。アナログでデータを読み込む理由は25個の内12個はラインセンサとカラーセンサの両方の機能を持っているからです。ラインセンサの場合、半固定抵抗で閾値を決めることができデジタルで読み込んでも問題ありませんが、カラーセンサの場合RGBそれぞれの色を順番に当ててその反射光の強さを読み込むためアナログである必要があるからです。このように私たちのロボットは25個のラインセンサと12個のカラーセンサでラインを読み取っています。

三番目は電源回路です。電源回路はヒューズとDCDCコンバータで構成されています。このDCDCコンバーターは5Vの電圧を6Aまで流すことができます。6Aまで流すことができるため、ラインセンサの28個のLEDを最大光量まで光らせることができ、ラインを周りの環境に影響されることなく読み取れます。また過電流防止機能や、逆流防止回路が組み込まれているため、ショートしたとしても被害を抑えることができます。

最後はマイクロコントローラです。この基盤にはArduino nano,Arduino megaが搭載されています。Arduino magaはメインマイコンで、Arduino nanoはサブマイコンです。二つのマイコンはI2C通信でつながっています。またサブ基盤の方には、M5stack(esp32)が搭載されています。このマイコンは液晶が付いているのでデータを表示するために使っています。

このようにして私たちのロボットに搭載されている電子基板はすべてプリント基盤であり、各機能がモジュールとして交換可能であるため故障したとしてもすぐに交換でる信頼性の高い基盤になっています。

プログラムおよび3DデータはGithubにて順次公開しています。

私たちのロボットは当初の目標どうり正確にかつ素早くライントレースができるロボットになっています。またロボットに搭載されている基盤は壊れることがなく、マイルストーンで示した設計要件を十分達成いていると思います。しかしレスキューラインのすべてのタスクを完遂するためにはライントレースだけできるということではまだ難しいため、コンテストまでの期間でプログラムに磨きをかけていきたいと思います。

このTDPの結論として、私たちのチームは次のように考えます。ロボットにおいてソフトウェア、ハードウェア、電子回路は個々に動くのではなくそれぞれが連携して動いているということです。これら三個の要素の中でどれか一つの要素でもかけていたらロボットは最高の性能を出してはくれません。この三個の要素はそれぞれが補っていますが補える範囲もまた存在します。レスキューラインのすべてのタスクを完遂するためには、どれか一つの要素が特に優れていても完遂することはできず、すべての要素が一様に完璧に動くことによって完遂されるものであると私たちのチームは考えています。

このロボットのハードウェア設計、ソフトウェアアルゴリズムがプロジェクトに与える影響は大きいと考えます。基礎基本のハードウェア設計、ソフトウェアアルゴリズムが私たちのロボットの底辺を支えており、これらがなければこのロボットに搭載されている各種部品や電子基板、モジュール、各関数は成り立ちません。またハードウェア設計、ソフトウェアアルゴリズムの開発がうまくいくことによってマイルストーンに沿ったプロジェクト計画を円滑に進めることができます。

チームの学習経験は大きく3つに分けられます。一つ目は3DCadによるハードウェア設計です。3DCadをで設計できるようになったことで、3Dプリンターなどを使って自分の思い描いた部品や市販されていない部品を短時間で作ることができるようななりました。また、ロボットの重心の位置関係や、重量の軽量化などのテクニックを学べることができました。二つ目はKiCadによる基板設計です。EV3からシフトしたことにより回路の基礎から学ぶ必要がありました。これにより回路設計に必要な法則や物理現象などを学ぶことができました。また今まで使っていた基盤のし国などを理解することができました。三つ目はソフトウェアアルゴリズムです。プログラムの実行時間を短縮化するためにマイコンのレジストリを操作する方法や、無駄なプログラムを省略するテクニックを学びました。

今後はコンテストに向けて各プログラムの変数の調整や3Dプリンタ製の部品の強度を高くすることなどの三つの要素でまだ足りていない所をコンテストまでに強化していきたいと思います。