高校生がつくばチャレンジに出場してみた

※機体解説※

池之上 響、玉木 諒、山田 智生、中田 陸斗

洛星高等学校ロボット研究部

チーム、機体の特徴

本チーム「オリオン」は京都の普通科高等学校、洛星高等学校ロボット研究部の生徒で構成されている。
現状歴代最年少のチームであること、また、関西からのチームであることなどが大きな特徴である。
本研究部では、「自作できるものは基本的には全部自作しよう」という方針がある。
本チームの製作したロボット”Mintaka”に関しても、CAD を使った設計、レーザーカッターなどを使用した機体の製造、自作設計の基板、ROS を使わないプログラムの製作など、ほぼすべてに自作要素が入っている。

ロボットに関しては、「できるだけ多くの災害に対応できる災害支援ロボット」をコンセプトとした。
災害時に活躍するロボットは現在でも多数存在しているが、こういったロボットは地震や火災など、特定の災害時のみを想定して作られているうえ、単価が非常に高いことが多い。
もしも学生が作ることが出来るほど低価格に、かつ 1 台で複数の災害に対応できるロボットを作ることが出来れば、自治体などでより災害支援ロボットを導入しやすくなるのではないかと考えている。

つくばチャレンジの結果について

つくばチャレンジにでは、高校生として初めて確認走行区間を突破した。
本走行の際は課題 A 付近でバッテリー残量通知が出てきて、後述する iPhone のスクリーンショットが出来なくなったため、残念ながら途中でリタイアすることとなってしまった。
初めての出場で、右も左もわからない中、確認走行区間を突破できたことは大きな戦果であると考えている。

以降、本チームの製作した機体”Mintaka”に関する技術的な解説を行う

制御関連

1, 全般

メインの制御には、Panasonic 社の Let's Note CF-NX3 で Python を使用した。
先述した本研究部の方針を達成するために、センサとの通信、制御、画像認識などのすべてのプログラムは ROS を使用せずフルスクラッチで製作した。また、信号や看板認識のモデルの作成も行ったが、そちらに関しては画像認識関連で説明する。

ボタンや LED の制御、無線操縦の受信等の GPIO 入出力用として Arduino を使用した。
Arduino-PC 間は UART 通信で通信している。

2,自己位置推定

自己位置推定には株式会社コアの Cohac-∞ Ten と SoftBank 社の RTK サービス Ichimill を使用した。Cohac-∞ Ten は今年 2 月に新たに始まったみちびきのセンチメーター級測位 CLAS が使用できる数少ない受信モジュールである。これに高精度な RTK サービスの Ichimill を使用することで、誤差 6cm 以下の精度を手に入れることが出来た。

課題 A 建物内、TX 駅構内、高架下、ホテル横、公園の林の中では高精度な RTK Fixed 解を得ることができなかったため、そういった場所でのみエンコーダで移動距離を算出することにした。

また、課題 A の建物(市庁舎)の移動距離の測定に関しては、建物外に市庁舎内にあるものと同じタイルが置かれており、その大きさ(60cm)を測ることが出来たため、タイルの枚数を数えることで移動距離を測定した。

本走行前日に、通信速度の速い USB を機体に取り付けたところ、USB から発せられるノイズがアンテナに干渉し、正常な値が入手できなくなることがあった。

制御に関してはライントレーサを参考にし、「Google Earth の kml データをコースデータとして読み込み、自己位置をマップ上にプロット⇒コースとのずれを計測し補正しながら走行する」といったシンプルなものとした。

3,障害物検知

障害物検知には、北陽電機の UTM30 を使用した。また、同センサは市庁舎内の入口で柱と壁を使用した、ミリメーター級の精密な自己位置補正を行うのにも使用した。

TCP を介して送られてくるデータは圧縮されているため、適切なデコードをする必要があった。

自己位置推定用には 2D で 10m 程度の測定範囲しかなかったため、適さなかった。

障害物を検知すると停止するのみの、シンプルな制御を行っている。

4,機体の方位推定

機体の方位推定には基本的に GNSS コンパスを使用した。しかしながら回転する際や GNSS の Fixed 解を得ることが出来ないような場所では GNSS コンパスの値は不正確となる。そこで、GNSS コンパスに頼らない機体の方位推定が必要であった。

当初、機体の方位推定用の IMU として EPSON の G370 を使用した。同 IMU は高精度だが、ジャイロ+加速度の 6 軸 IMU である為ドリフト対策が大きな問題となる。そこで、方位もついた 9 軸 IMU である、WT906 も使用した。しかしながら、一般的な補正方法であるカルマン補正や Magwick 等では、傾斜

した地形や磁場が多少乱れた(TX の高架下など)場所では信頼できる角度を出すことが出来なかった。

チームで対応を検討した際に、iPhone SE2 のコンパスはアプリ側で適切な補正がされている為に高精度であることが分かった。そこで、多少強引な対応だが、iPhone の画面を Python スクリーンショットし、値を読むことで高い信頼性の角度を入手することが出来た。一方、この方法では 1° 単位でしか値を取ることが出来ないため、最大で 1° の誤差が発生してしまう。課題 A の市庁舎内など、GNSS の補正を使用せず長距離移動する際にはこの誤差は大きな課題となっている。

5,デバッグ性

高いデバッグ性能を実現する為、機体が現在の状態を音声で伝えるようにした。
音源に関しては、弊研究部の広報キャラクターの物を使用した。

機構、ハード関連

1, 新幹線に載せるため…

京都からつくばまで未成年がロボットを輸送するには、自動車を使用できないので、新幹線を使用する必要があった。したがって、機体のサイズ、重量、電源電圧は、新幹線に載せることが出来る数値である必要があった。具体的には、新幹線のドア幅は 56cm であったため、機体の横幅はそれより小さくする必要があった。高さと縦幅に関しても横幅とのバランスを考慮して設計した。
また、荷物の重量が 30kg を超えると新幹線に載せることはできないため、機体の重量はそれより軽い必要があった。

また、無線操縦でロボットを輸送することが出来たが、他の乗客に恐怖感を与えないよう駅構内では手押しで輸送した。しかしながら、動力が無い状態での手押し走行は移動速度が遅く、東京駅などの混雑した場所では余計に迷惑となる為、手押しの電動アシスト機能を搭載した。

2, Cugo-V3+エンコーダモジュール

株式会社 Cubo-Rex よりクローラモジュール”Cugo-V3”とエンコーダモジュールを特別価格にて購入した。足回りにクローラを採用したことで、豪雪、水害などの災害時に迅速な移動を可能としている。

モータの制御方法等に関しては後述する。

3, 荷物入れ

災害時に物資輸送ができるよう、機体の大部分は荷物を搭載することが出来るスペースとなっている。

また、底板は取り外しや交換が容易になっており、底板の代わりに排土板や除雪用の刃、クレーンやジャッキなどのモジュールを取り付けることで物資輸送以外の様々な災害支援を可能としている。

4,今後の課題点

ねじが外れないよう、ロックタイトやばね付きナットを活用したが、それでも輸送時に数本のネジが欠落していた。
また、防水対策に関しては、試験走行の日が全て晴れであったため十分な試験が出来なかったと考えている。
不十分な防水対策や機体の振動でねじが外れ続ける信頼性の低さなどが今後の課題として挙げられる。

電源及び電動機駆動系回路概要

Mintaka に搭載した回路システムについて以下記載する。

1, 主回路電源用 DCDC コンバータ

本体駆動用電源の概要について以下示す通りである。

本体の電源は 12V7Ah の鉛蓄電池から供給する。モータ駆動用適正電圧は 24v のため、12V から 24V へ昇圧する DCDC コンバータを搭載した。

コンバータはユニバーサル基板を用いて制作しており、一時間連続定格出力電力は 96W、最大出力については約 300W としている。回路構成は TL494 を使用した一般的な昇圧チョッパとし、スイッチング周波数は固定 150kHz にて PWM 制御を行う。電圧制御器には Type2 補償機を使用し電圧応答性の改善を図った。保護として電流制限機能を搭載しており、一定以上の電流は流れない。出力短絡時は制御が働かないためヒューズを使用し保護する。

主スイッチング素子には TK100E08N1（東芝セミコンダクター社製、80V,100A 定格）採用し、出力電圧に対し十分な定格電圧を持たせることにより信頼性向上を図っている。なお、電流定格が 100A と大きい理由は、この素子のオン抵抗が 2.6mΩと大変小さく、導通損失の低下が見込まれるためである。電流容量を稼いだ際の寄生容量による損失は一般的に増加傾向にあるが、今回の使用条件での増加量は相通損失低減量に対し十分に小さいことを確認している。

2, モータドライバ

本体の駆動用モータのドライバについて以下示す通りである。

駆動電圧は DC12V~24V、出力電流定格は 20A である。モータドライバ用の電源本体に電流制限機能を備えているため本ドライバに電流監視機能などは搭載しない。ユニバーサル基板を用いて制作している。

回路構成は、FET のフルブリッジ二組からなる一般的なものである。各相低速レグと高速レグを備え、それぞれ別の動作を行う。PWM による出力電圧制御は高速レグを使用し、出力電圧の切り替えのみ低速レグを使用する。

ゲートドライバは全ての素子に TLP250H（東芝セミコンダクター社製）を使用している。高速レグ側では TLP250H によるチャージポンプを構成し（ブートストラップ方式)、フローティング電源によりハイサイド素子のゲートを駆動する。低速レグ側のハイサイド素子のオン時間は長くなることが予想されるため、ブートストラップ方式の使用は信頼性に欠ける。そのため、ハイサイド 2 素子分の絶縁型 DCDC コンバータ(12~24V to 12V) を使用した絶縁電源方式のゲートドライバを採用した。また、各レグのデッドタイムは IRS2003STRPBFTR（International Rectifier 社製）を変則的に使用することにより生成している。

3, モータ制御器

モータの制御方法について以下示すとおりである。

モータの制御には Nucleo F401RE（ST マイクロ社製）を使用した。モータにはエンコーダが搭載されており、これを使用し回転数を検出する。モータドライバと STM マイコン間通信は非絶縁で行った。
STM マイコンからモータドライバに出力される PWM 周波数の設定は 9kHz であり、これに伴いモータドライバ側の駆動周波数は 9kHz である。今回はハーフブリッジによるスイッチング動作を行うためモータドライバからの出力周波数も 9kHz である（フルブリッジによるスイッチング動作だと 18kHz となる）。

4, 課題点

本システムにおける課題点を以下まとめて記載する。

・主 DCDC コンバータの熱設計
主スイッチング素子で 120W 時に約 4W の損失が発生し、放熱機の温度が 80 度を超えている状況となり信頼性が低下していた。設計段階では約 1.5W の損失を見込んでいたが、日光によるヒートシンクの温度上昇及び使用環境の通風状況を十分に考慮していなかったことによるヒートシンクの温度上昇とそれに起因するスイッチング特性の劣化が原因である。屋外高温環境で使用する際は熱設計にかなり余裕を持たせることを考慮したい。

・エンコーダの読み取り処理
今回採用したマイコンは 32bit、クロックは 84MHz であるが、処理速度が不足する場面が多々見られた。これによりエンコーダのミスカウントが発生し結果として機体を停止させる処理落ちが発生した。マイコンの処理速度を向上させる又はエンコーダのカウント方法を見直すなどの対策が考えられる。

画像処理関連

1,信号認識

今回、本機体で課題 B を突破するにあたり YOLOv5 を用いた物体検出と OpenCV ライブラリを用いた色認識を利用した。YOLOv5 を用いた理由は、指定された横断歩道の信号の種類は電球式であり、画像全体から認識するのでは誤検知が増えるだろうと考え、信号の部分を全体の画像から抽出するためである。

YOLOv5 を用いるにあたり、モデルは自身で製作した。実際 YOLOv5 には標準で信号を含めた 80 種類の物体を検出できるモデルが存在するのだが、それでは今回の課題である電球式の信号を認識できず、自作した次第である。モデルの学習には現地で撮影した画像を 400 枚、ある程度の汎用性を担保するために学校周辺で撮影した同型の信号機の画像 200 枚の合計 600 枚を Roboflow でのデータの拡張を行い、1200 枚ほどをトレーニングデータとして学習を進めた

2,コーンの認識

今回 Mintaka には 3DLidar が搭載されていないため人とコーンの区別がつかないため、カメラを用いて目の前にある障害物がコーンなのか人なのかを見分けるプログラムを作成した。

このプログラムでも、YOLOv5 を用いた物体検出を使用した。モデルは自作し、トレーニングデータとしては現地で撮影したおよそ 500 枚を二倍に拡張し、計 1000 枚を 1 と同様の方法で学習を進めた。最終的なモデルの推論結果は以下に示す通りである(参考画像 3,4 を参照)。

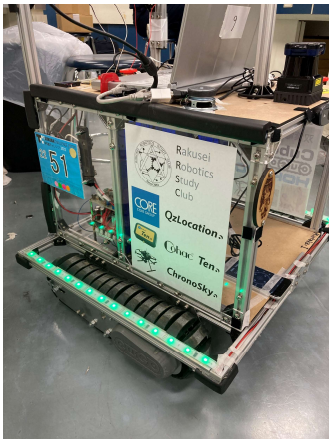
認識に物体検出を用いた理由としては、画像のどの場所にコーンがあるのかというデータと、何個コーンが視野内に存在するのかという情報を得る必要があったからである。

3,課題

認識速度に関しては今後改良の余地があると考えている。

今回の信号及び静止障害物の認識プログラムには共に YOLOv5 を用いたことで認識の精度は非常に高いのだがそれと引き換えに速度の低下が見られ、一回の認識あたり 1.5~1.8 秒の時間がかかった。主な原因としては推論を CPU で行っていたためであると考える。特に信号の認識などの認識速度と精度が重視される場合は GPU を搭載し計算速度を向上させる必要があると考える。

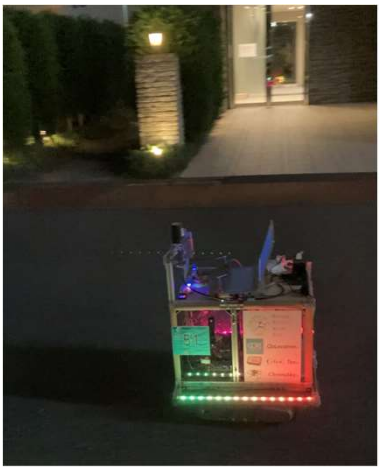
参考画像



⇧機体の全体

⇧確認走行突破時の写真

⇧新幹線に載せられる機体

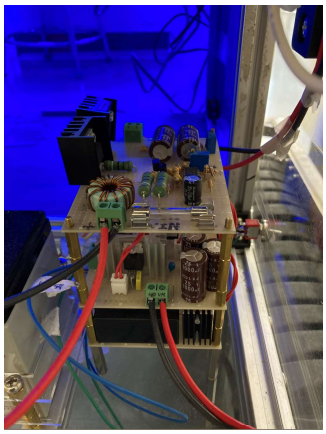


⇧深夜作業で虹色に光る機体

荷物を載せた状態⇩

iPhoneを載せている⇧

自作基板⇩



⇧参考画像1（画像認識）

⇧参考画像2（画像認識）

⇧参考画像3(画像認識)

⇧参考画像4(画像認識)