

i-Cart mini と ROS による自立移動ロボットの開発

徳安一，西田健

1 はじめに

我々 CIR-KIT は，九州工業大学工学部の学部生を中心としたロボット開発チームであり，屋内外の移動を行う福祉ロボットの開発に取り組んでいる．自律移動ロボットが人間の生活環境内で活動するにあたって必要とされる機能は安全性の確保を始めとして多くがあるが，本稿で報告するロボット (KIT-C5) はその中でも最も基本的な機能と考えられる，地図情報を元とした移動と障害物回避に焦点を当てて，チーム内の新人の技術習得を目的として開発を行ったものである．

2 ロボットの構成

本ロボットでは，ハードウェアに i-Cart mini を，ソフトウェアに ROS (Robot Operating System) を採用した．以下でそれぞれについて述べる．

2.1 ハードウェア

i-Cart mini は筑波大学知能ロボット研究室 T-frog プロジェクトにより開発された屋内外用の小型移動ロボットフレームで，これを用いることで最小構成の移動ロボットシステムを容易に構築することが出来る．また，オープンソースハードウェアであることも大きな特徴で，T-frog プロジェクトのウェブサイト [1] 上で図面・部品リストが公開されている．

i-Cart mini は 2 機のブラシレスモータにより駆動される独立二輪方式として設計されており，公開されている通りに組めば全長 423.50 [mm]，全幅 365.80 [mm]，全高 157 [mm] である．

小型であることは運動性能の観点からは利点となるが，安全性の観点からは歩行者がロボットの存在に気づけない可能性があることや非常停止スイッチの取り付け位置が低くなることが問題点として考え

られた．そのため本ロボットでは i-Cart mini を立っている人間が身を屈めず触れられる高さ (600 [mm]) まで延長し，そこに非常停止スイッチを設置することでこの問題を解決した．また，歩行者からの視認性を考えロボットを周囲の景色との対比から十分に目立つと考えられる赤色の外装で覆った．図 2.1 に実際に競技会で運用した KIT-C5 の外観を示す．



図 2.1: KIT-C5 の外観

2.2 ソフトウェア

i-Cart mini は，二軸モータドライバ TF-2MD3-R6 と走行制御用プログラム Yp-Spur を組み合わせで動作するように設計されている．Yp-Spur は i-Cart mini と同じく筑波大学知能ロボット研究室により開発されている移動ロボット用の走行制御プラットフォームで各モータドライバへの指示値をロボットの移動方向という形に抽象化する [2]．また，Yp-Spur には ROS と自然に協調するために ROS

パッケージ `yvspur_ros_bridge` が用意されている。

今回は Ubuntu 16.04 インストール済ラップトップにおいて ROS Kinetic Kame を運用した。i-Cart mini においては、二軸モータドライバ TF-2MD3-R6 を運用する。ここで、ドライバ側との通信に於いて互換性のある通信を行う必要があり、これは ROS システム標準のメッセージ/トピックに対応していないため、別途パッケージを用意する。

URG 運用に不可欠な LaserProc パッケージは、開発当初は Kinetic Kame 向けにリリースされていなかったために、代位手段を模索していたが、10 月半ばにリリースがなされたので、これを使用して運用を行った。

2.3 センサ

主センサとして当初より機体前部に設けたマウンタによって北洋電気製の URG を 1 基搭載して運用した。計画の上ではカメラ、後方 URG、GPS 測位モジュール、超音波センサなども検討したが、開発期間の都合上今回の採用は見送って大会に臨んだ。

2.4 駆動部

二軸モータドライバ TF-2MD3-R6 によってブラシレスモータを駆動するのに加えて、オドメトリを同時に取得する。ある程度の処理能力のあるマシンによって得た環境地図の精度は、単一のセンサによるものであってもかなり優秀なものが得られた。これは、センサ類の数を省略して開発を進行するという決定の一因にもなっている。

2.4 筐体

強度を担う部分はアルミフレームとボルト留めで構成されている。運用に於いて、標準の構成要素に加えて、URG 搭載用の自作マウンタを装備した。

これは、URG によって得られるデータが水平面方向にあることから、URG の高さ調整を可能とすること、また、複数の高さにおける水平面でデータを取得できるようにしたいと考えたためであった。実際の運用においては、調整が煩雑に生じること、取付強度的に問題があることに加えて、重心と離れた点にセンサ類を搭載したことなどから振動が激しくデータに悪影響を及ぼすなど問題の多い構造となった。

2.5 マシン

一般的なラップトップを用いて ROS Kinetic Kame を運用した。

当初は旧式のラップトップを用いて運用していたが、URG による環境地図の取得中に、著しい処理能力不足でシステムがダウンするおよび地図が正しく取れないという自体に陥り、急遽これを改めた。

3 走行実験結果

制作中のユニットテスト/実働試験では良好な結果を納めていたが、当日になっていろいろな問題が発生した。それらをここに記す。

3.1 外装

当初の予定では、駆動部のみに重厚なカバーを施し、他の部分については極力省略していくことを構想していた。

また、降雨時の対策として、防雨版を施すことを予定していたが、時間の都合で省略されていた。

被発見性/視認性の低さから現場での改修を行った。プラダンボール板により外装をかさ増しして、開口部を閉塞した。センサ類/駆動部との干渉は可能な限り避けるという方針になった。

当日こそ晴れたが、試験日に雨の釣行が見られ、急遽防雨シート(ビニールの切れ端)を取り付けた次第がある。当初の予定では、防雨版は標準で装備しておく予定だったが、進捗の関係で叶わなかった。

3.2 走行特性

振動が激しかった。石畳の上を走行する実験を欠いていたため、当日まで問題にならなかった点である。特に種々の改造に於いてトップヘビー・前重心という状況になっていたことが状況を悪化させた。

状況によっては、ピッチ方向に $\pm 5^\circ$ ほどブレが起こった。揺動に従って、センサの返す値も激しく変動した。これについては、カウンターウェイトなどを配置してある程度まで対処可能である。

それにもかかわらず最終的なオドメトリ取得/地図作成は優秀な結果を出している。これは i-Cart mini 及び ROS Navigation を組み合わせた運用のポテンシャルを物語るものの一つであろう。

車体下部にバッテリーを搭載しているが、縁石や障害物を超える際にこれが引っかかって動けなくなる事態が頻発した。改造が必要だと認められる。

3.3 ソフトウェア

予め用意した環境地図を用いて即生きセンサからの情報を元に ROS Navigation により経路選択を行

い、Publish されたトピックを USB シリアル通信によって二軸モータドライバ TF-2MD3-R6 に指令値として渡すことができる。

特段の致命的なバグはなかったはずではあるが、前日に用意したデータとの差異/地図作成時のノイズが自己位置推定を破綻させるケースが多々あった。

また、ROS Navigation の運用に際して、主要開発メンバの大半が初心者であったということも手伝って、トラブルにも満たないエラーの対処にさえ混乱があった。いかに ROS がユーザフレンドリーかつ一般性のあるシステムであったとしても、使用者の技能如何に於いては、スムーズな開発 / 運用を実現しきたいという場合もありうる。

地図作成に於いて、ところどころノイズと思われる点が地図内に発生、その点に治して回避動作をとっていると思われる動作が所々に見られた。

現場では、手動でノイズと思われる点に対して、画像編集ソフトによる訂正を加えた（白塗りをかけた）。

根本的には、入るノイズを低減するために機体/センサの安定性を向上すること、ソフトウェア的なノイズ低減フィルタを設けるなどを設ける必要性などが感じられた。

起動手順における自動化が徹底していなかったために、起動順序が煩雑となっていた。実際、本走行における失敗に付いては、人的過失による操作ミスの可能性が CIR-KIT 部内から指摘されている。

制作が当日まで継続されたということもあって、十分な調整および成熟が行えなかったことが悔やまれる。

4. 開発体制について

4.1 開発参加者の熟練度 KIT-C5 は主に九州工業大学 CIR-KIT 所属の学部生によって制作された。

開発期間は、準備期間まで含めるとおおよそ三ヶ月程度である。

ROS システムについては筆者含め非常に不慣れであったため、開発の進行途中に ROS システムの勉強をやり直す必要性が度々発生していたが、同時に細部の実装については、低級工程をパッケージ導入という形で省略したうえで完了することが可能となったために、非常に開発を高速化できたという側面もある。

ROS システムへの習熟は重要であるが、如何せん

時間がかかる。しかしながら ROS エコシステムに共通の部分へ理解が深まれば、他のパッケージについても共通に扱える部分は多少なりとあるはずであるから、これらは少なくとも ロボットに関わる団体に所属することがわかった時点ですでに習得を開始すべき要因の一つである。言い換えれば、これらの要素は一度十分に身につけてしまえば、それからの開発速度は、少なくとも共通に扱える部分については、格段に向上できるスキルの一つである。

今後は、開発参加者について、早い時期から ROS の実用を勤めたい。筆者自身も、ROS を運用する中でこそ発見できたことが多々あった。チュートリアル消化はもちろんのこと、行いたいタスクとパッケージ構造をつなげて考えられるようになるには、現場 / 現物での経験が必要になる。これらを次の開発メンバ育成からは進めていきたい。

対症的に、低次の処理を行うために必要な技能については、全くというほど使用しなかった。これについては、煩雑なハードウェアの仕様を論理関係内に落とし込んでいく作業を省略できたという側面と同時に、パッケージの提供されていないハードウェア / 処理について対応する能力の向上、端的に言うとドライバを書く能力については練習になっていないだろうという予測も立つ。

ROS システムを使用することはロボット開発を高速化する。ただし、その高速化はある意味で仕事の省略であり、教育という面からすると省略された部分からも学ぶべきところがあることは注意するべきかもしれない。

5. 今後の課題

5.1 外装

FRP などの一体成型型かつ防雨性のあるものを実装する。また、整備性 / 強度についても大いに考慮する。

5.2 筐体

重心位置と全高とのバランスを考慮した作りにする。また、センサ搭載位置の安定性に考慮する。

5.3 センサ

複数搭載して、安定性を向上する。また、AMCL の 予備策となりうる手段を併用することで安定性を向上する。

5.4 開発体制

開発参加者の技量を向上する。また シミュレーションモデルの不在によりソフトウェアテストとハードウェアテストの独立化が図れなかった。これを改める。

6 結び

今回の開発に於いて，多数の人々の協力により出走まで漕ぎ着けることができましたことをここにお礼申し上げます。

参考文献

- [1] ”ロボットフレーム i-Cart mini” ,
http://t-frog.com/products/icart_mini/
- [2] ”移動ロボット走行制御コマンド系および走行制御系” , http://www.roboken.iit.tsukuba.ac.jp/platform/wiki/_media/ypspur-0.13.2.pdf