つくばチャレンジ2019における関西大学の自律移動ロボットの開発

貞平紘己，潘晨浩，薛経緯，山本恭輝，高橋智一，青柳誠司，新井泰彦（関西大学）

連絡先：貞平紘己(k635261@kansai-u.ac.jp)

1. 緒言

本研究室では，屋外自律移動ロボット開発の一環として，2014 年よりつくばチャレンジへ参加している．昨年はコースの完走を目指して自律移動に関する部分の改良を行なっていた．今年はコースの完走だけでなく新しく選択課題の達成も目指し，信号認識などに取り組んだ．本稿では，2019 年度のつくばチャレンジにおける本研究室の自律移動ロボットの概要について報告する．

2. ハードウェア構成

本章では，本研究室で開発した自律移動ロボット"KUARO"のハードウェア構成について述べる．

2.1 ロボット本体

Fig.1 に本研究室で開発したロボットの外観を示す．ロボットは長さ 75 cm，幅 55 cm，高さ 120 cm であり，重量は約 70 kg である．前輪駆動の非ホロノミック拘束を持つ．ロボットの筐体は，側面を塩化ビニル板で覆っており，上面はアクリルパネルを蝶番で固定している．筐体とベースの間は 5 本のアルミフレームで締結されており，レーザが通過できるように外装で覆わないようにしている．

2.2 搭載機器

Table.1 にロボットの搭載している機器の構成を示す．自己位置推定にはLiDARを使用し，障害物回避には2DLRFを使用している．2DLRFについては，ロボット前後に設置し，2つからの取得データを合成して使用している．また，3軸の加速度計、ジャイロスコープ、磁力計を内蔵した慣性計測センサを導入することでより精度の高いデータを取得できるようにしている．PC は，Intel 製の NUC5i7RYHを使用しており，PCのOSはUbuntuを使用している．また今回選択課題で信号認識にYOLOのDARKNETを用いるためNVIDIA製のJetson-TX1を搭載している．

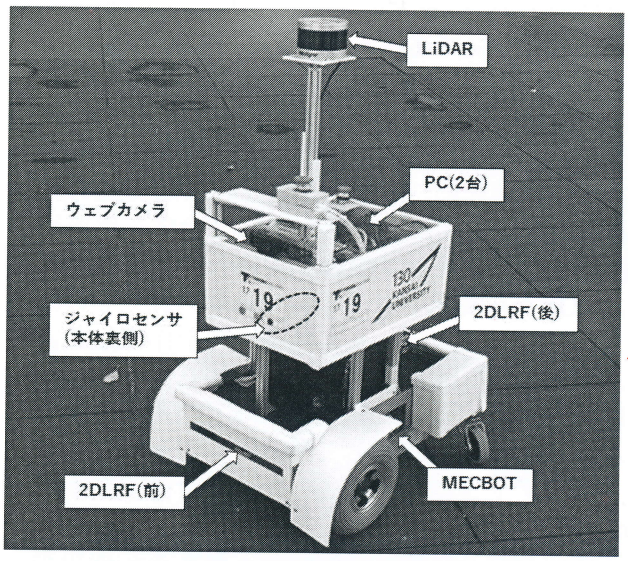

Fig.1 KUARO の外観図

Table 1 搭載している機器

センサ	型番	メーカ
2D LRF × 2	UTM-30LX	北陽電機
LiDAR	VLP16 Puck Hi-Res	Velodyne
ジャイロ	3DM-GX5-25	LORD MicroStrain
ウェブカメラ	HD PRO WEBCAM C920R	logicool
モータ	MECBOT	オカテック
PC	NUC5i7RYH	Intel
PC	Jetson TX1	NVIDIA
ゲームパッド	EDGE 301	ホリ
モニタ	LCD-8000VH	センチュリ

3. 自律走行システムの構成

自律走行システム の作成には,昨年度から引き続きロボット用ミドルウェアであるROS(Robot Operating System)[1]を使用している．システムの機能は大きく分けて，環境地図作成，自己位置推定，障害物検出，経路計画と追従の3つから構成されており，これらの機能をROSから提供されるmove_baseパッケージにより統合している．本章では3つの機能をどのように実現しているか，その方法について述べる．

3.1 環境地図作成

環境地図作成においては昨年度とほとんど変更を行なっていない．本システムでは，自律走行時に自己位置を推定するためのコース全体の環境地図を予めオフラインで作成している．地図作成には，ROSから提供されるgmappingパッケージにより，10 cm 四方の 占有格子地図を作成している．地図作成にはロボット上部に取り付けられたLiDARと，ジャイロセンサによって補正したホイールオドメトリを利用している．LiDAR から取得されるデータは 3 次元の点群データであるが，gmappingはレーザ形式の データを受け取って地図作成を行うため,データ を変換する必要がある．そのため，ROS のパッケージである pointcloud_to_laserscanを使用した．gmappingでは，ロボットに追従していた安全管理責任者やオペレータがノイズとして残ってしまうことが多い．そのため昨年と同様，画像編集ソフト(GIMP)を用いてノイズ箇所を削除した．これに加え，コーンなどのその日によって設置場所が変わる可能性があるオブジェクトについても削除した．
また，昨年同様，自律走行時に辿るべきコース上の点(以下waypoint)を，bagfile を用いて手動走行の経路を基に自動で生成するようにしている．waypointについては，自動生成だけでは精度が悪いため，実際の道を考慮しながら手動で修正を加えている．

3.2 自己位置推定

ロボットの自己位置推定にはROSから提供されるamclパッケージを利用している．amclに用いるデータは，LiDARから得られる全方位のスキャンデータとジャイロで補正したオドメトリである．amclで使用するスキャンデータもgmappingと同様にpointcloud_to_laserscanで変換している．

3.3 障害物検出

障害物は基本的にロボットの前後に取り付けた2DLRFで検出を行う．前方のLRFは高さが低い障害物のデータを取得するために，倒立させて設置している．2DLRFの設置高さにある障害物はmove_base 内のcost mapに登録され，軌道計画の際に利用される．障害物との距離に関してはLRFの取得距離とコストの範囲を調整することにより障害物にどの程度近づくかを調節している．

3.4 経路の計画と追従

自律走行は，大域的な経路計画と局所的な軌道生成を組み合わせて行っている．大域的な経路計画は，ロボットの現在位置から3.1節で述べたwaypointへ向かう経路を生成する．今年度も昨年度同様，carrot_plannerをプランナーとした．carrot_planner は現在地から目的地まで直線で経路を生成するため，長距離の経路を生成することには適していないが，この問題はwaypointを細かく置くことで対処している．
大域的な経路計画だけでは，急な障害物などに対応できない．そこで,大域的な経路に沿いつつ,障害物を回避するような局所的な軌道を生成する必要がある．局所的な軌道計画にはteb_local_plannerを利用している．このプランナーはTimed Elastic Bandと呼ばれる手法によって軌道を生成する．[2][3]

4. 選択課題への取り組み

昨年度から地図作成や軌道計画などについて特に変更点はないが，その代わりに今年から選択課題に取り組んだ．取り組んだ内容としては選択課題Bの信号認識である．そしてその課題にある停止線前での停止については白線を検出出来るプログラムを作成し，白線を認識しそれが停止線だと判別されれば止まるというシステムを作成した．それぞれの検出方法について以下に述べる．
また白線や信号認識については常に認識し続けるのではなく，waypointによってタスクを行なう部分を設定した．これについても詳細を以下に示す．

4.1 waypointによるタスク空間の指示

信号認識では常に認識を行ない続けると，充電などが減りやすくなり，pcでの処理も重くなってしまうため，タスク(課題のためのプログラム処理)を行ないたい付近で認識を行なうことが望ましい．そのため地図作成時に，ゲームパッドで信号を送り，どのwaypointがタスク指示部であるかを設定した．これにより，自律移動を行なっている際に，タスク指示部に近づいた時にタスクを行えるようになる．

4.2 白線検出について

waypointなどを使用して停止線前で止まることも可能であると考えられるが，オドメトリやジャイロの精度に頼る必要がある．そのためスタートから停止線までの距離が長いと，誤差が積み重

なり，停止線を越えて停止してしまう可能性がある．その問題を解決するため本研究室では停止線を検出することで停止する方法を用いた．
検出方法としては，取り込んだ画像をhsv空間に変換し，閾値の範囲を設定して白い部分の二値化を行なう．その後，ハフ変換を利用して白い直線を検出する．画像の半分以上の長さで，傾きが±20°の直線を検出した場合，それを停止線と判別する．hsv空間を用いた理由として，歩行者用の道路には点字ブロックがあり，その色が黄色であるため単純に二値化すると黄色を白であると誤認識してしまうからである．hsv空間に変換することで黄色を含む白は全て黒色となり誤認識率を下げることができた．ただ光が反射して黄色が白色になったときは，まだ誤認識してしまうため，さらに改良が必要であると考えられる．

4.3 信号認識について

学校周辺の同じ信号を学習対象として，信号の赤と青を分けてそれぞれ2000枚の画像を用いて学習を行なった．学習にはDarkNetのYOLOtinyを利用している．20万回学習させている．YOLOtinyを用いた利用した理由としては，認識精度はYOLOよりも劣るが，認識スピードが1秒間に20回の認識ができ，YOLOよりも速いためである．実験走行時の精度としては，20回の内10回ほど認識されていた．誤認識はなく，精度は良かった．しかし，光の反射により信号そのものが認識されないことがあった．

5. 試走会での取り組み

試走会では，11月8日と11月9日の2日間でコース上のデータ取得と地図作成を行なった．また課題のために作成したプログラムの正しく認識できるかを検証した．地図作成は完走を目指していたため，1日目はコース全体のデータ収集を行なった．ただコースが長いことからデータ量が莫大になってしまったため容量不足で途中までしかデータを得ることが出来なかった．またデータ収集を行なった所までの地図作成を行なったが，確認走行区間を越えた横断歩道付近から地図の歪みが生じていることを確認した．2日目は，1日目に全体の地図作成がうまくいかなかったことや白線認識のプログラムの実装が出来ていなかったことから，信号認識の所までの地図作成を重点的に行なった．
作成した地図をFig.2に示す．結果としては2日目の地図も確認走行区間を越えたあたりで歪みが発生してしまった．歪んでいる地点のwaypointを少しずらしたり，地図上にある障害物を消して複数回自律移動を試したがうまくいかなかった．
歪みの原因は特定できなかったが，考えられることとして，昨年の地図作成ではFig.3に示すように，今年より道が歪んでいないことがわかる．このことと，昨年から使用しているプログラムに変化していないことから，ソフトウェアの問題ではなく，ハードウェアから起きた問題だと考えられる．特にオドメトリの出力に異常が見られたことがあったためそれが原因ではないかと考えられる．


Fig.2 今年度作成した信号までの地図

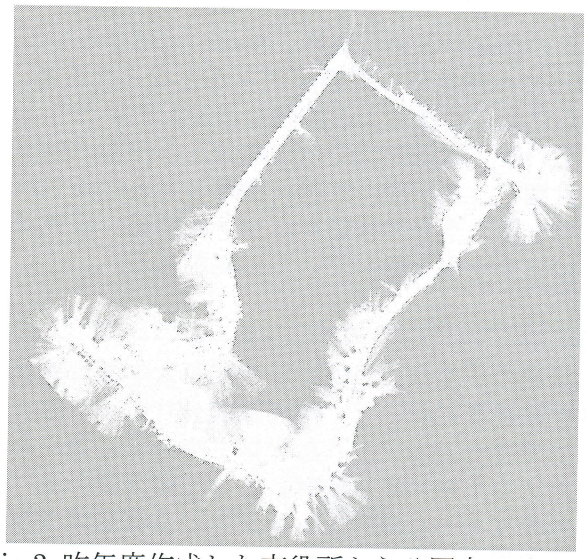
Fig.3 昨年度作成した市役所から公園内の地図

6. 本走行での結果

本走行での結果としては，地図作成が確認走行区間を超えたあたりから歪みが発生し地図作成をうまく行えなかったこと，また停止線の前で停止することが出来なかったため，270mという結果になった．

確認走行区間では，実験走行時に地図上にあるノイズやwaypointの修正を丁寧に行なったことで何の問題もなく進むことが出来た．また地図に歪みが発生してしまったことから確認走行区間を

越えた横断歩道から自己位置推定をうまく行えなくなり，走行不可となってしまった．

7. 結論と今後の課題

本走行の結果は，地図作成がうまくいかなかったため，昨年の結果よりも走行距離が下がってしまった．また信号認識などはうまく認識出来ていたのを確認できたものの，自作プログラムとROSパッケージの複合が間に合わなかったということもあり，不完全なままで終わってしまった．今回は2日間という短い期間でのデータ収集から地図作成を行なわないといけなかったため，非常に効率的な作業が求められた．今後も同じようなことが求められると考えられるため，データ収集などの効率化も考えていく必要があると考えられる．また，今年度から選択課題にも取り組み始め，Pluginを用いてROSに自作プログラムを追加機能として実装しようと考えているため，よりプログラムの理解が求められる．またつくばチャレンジ2019では想定外の問題が明らかとなったため，それらに対応できるよう原因を追及していこうと考えている．本研究室での技術向上と作業の効率化を目指し，来年度には選択課題と完走の両方が達成できるよう自律移動ロボットの開発，研究によりいっそう取り組んでいく必要がある．

謝辞

本研究の一部は，平成28年度関西大学教育研究高度化推進費において，研究課題「ロボット競技会をモチベーションとしたソフトウェアに力点を置いたメカトロニクス教育」として研究費を受けた．今年度もつくばチャレンジという有意義な技術交流の場を提供してくださった，つくばチャレンジ実行委員，大学関係者，協力してくださったつくば市の皆様に感謝いたします．

参考文献

[1] ROS.org , http://wiki.ros.org/ja
[2] C.Rösmann, W. Feiten, T. Wösch, F. Hoffmann and T.Bertram,"Trajectory modification considering dynamic constraints of autonomous robots",Conference on Robotics(ROBOTIK2012), pp.74-79, May 2012.
[3] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann and T. Bertram," Efficient trajectory optimization using a sparse model".Conference on Mobile Robots(ECMR), pp…138–143,September 2013