

GPS とセンサの組み合わせによる自己位置推定システムの開発（第2報）

Development of localization system by GPS and other sensors (2nd report)

技術開発部 生産・加工科 吉田英一 菅野雄大 稲葉勉 近野裕太 清野若菜

GPS等の測位衛星からの信号が受信できない環境下でも利用可能な、安価なセンサの組み合わせによる自己位置推定システムを開発する。今年度は北陽電機製のレーザレンジファインダ UTM-30LX（以後、「LRF」と呼ぶ）による自己位置推定システムを構築し、センサ評価ロボットに搭載して屋内外で環境地図の作成や自律走行実験を行った。また、障害物を回避するため、YOLOによる障害物の認識実験を行った。その結果、屋内においてLRFによる自律走行ができることや、ロボットに搭載したカメラで障害物を認識できることが確認できた。

Key words: GPS、自己位置推定、障害物検出、YOLO

1. 緒言

近年、自律走行ロボットの研究開発が盛んに行われており、GPSトラクターや農作物の無人運搬ロボットなど屋外の様々な環境を走行可能なロボットの実用化が期待されている。

しかし、屋外で自律走行するロボットの多くは位置情報を取得するためにGPSを利用するが、GPSアンテナ周辺に壁等の障害物がある環境では位置情報の誤差が大きくなる課題がある。また、自律走行を実現するためには多くの要素技術を開発する必要があり、開発コストや技術者不足の面から中小企業だけですべて開発することは難しく、ロボット関連産業の参入障壁の一つとなっている。

そこで本研究では、自律走行を実現するために必要な自己位置推定技術において、GPSからの位置情報による位置推定のほか、位置情報が正確に取得できない環境でも、各種センサを組み合わせることでロボットの位置を推定するシステムを開発する。

また、深層学習による画像認識技術を活用して障害物の認識と回避を行うシステムも併せて開発する。

本研究の概要を図1に示す。

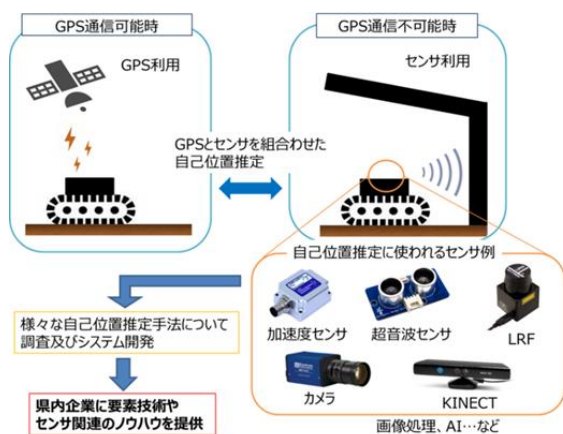


図1 研究概要

昨年度は、自己位置推定に利用するGNSS受信機を使い、単独測位やRTK測位などの測位方式の違いによる測位実験を行い、測位精度を比較した。また、センサ評価基板及びロボットを製作し、ロボットに搭載した状態でのセンサ評価、自己位置推定システムの評価を行う開発環境を構築した。さらに、自律走行ロボットに物体検出機能を付加する目的で、深層学習による物体検出システムYOLOを用いて屋外における物体の学習モデルの作成と評価を行った。

今年度はLRFによる自己位置推定システムを構築し、昨年度製作したセンサ評価ロボットに搭載して、屋内外で環境地図の作成や自律走行実験を行った。

本稿では、LRFによる自己位置推定システムの開発と、動作を検証するために行った環境地図作成や自律走行実験について述べる。また、AIを活用した障害物の認識実験について述べる。

2. 実験と結果

2. 1. 自己位置推定システムの概要

本研究ではGNSS、ベイズフィルタ、自律航法による位置推定手法を組み合わせた自己位置推定システムの開発を行う。自己位置推定とは既存の地図やセンサデータから走行中のロボットの位置や向きを推定する技術のことで、自律走行に必須の技術となっている。

ロボットの自律走行の実現には自己位置推定のほかに、地図作成、障害物検出と回避、経路・動作計画などの機能を実装する必要があり、膨大なソフトウェアすべてを自前で実装することは現実的ではない。

そこで本研究ではROS^{1) 2)} (Robot Operating System)を使用し、自律走行機能を持つROSパッケージ(任意の機能を含むソフトウェアの集合)とGNSSや自律航法の位置推定手法を組み合わせることで自己位置推定システムを構築する。

また、利用するセンサは、位置推定や障害物検出に

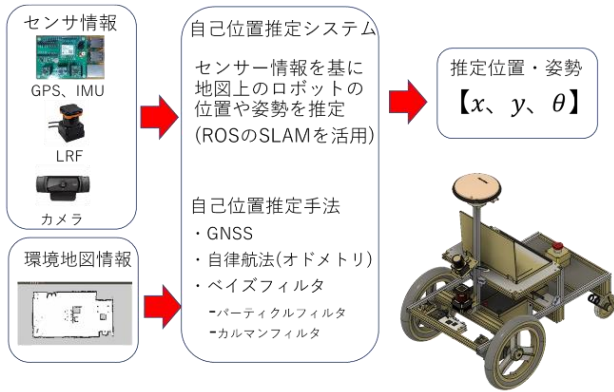


図2 自己位置推定システムの概要

使用する GPS、LRF、慣性センサ、カメラ、超音波センサとし、ROS のセンサドライバやデータ送受信機能を利用して、センサデータの取得や送信を行うこととする。自己位置推定システムの概要を図2に示す。

センサ情報と環境地図情報から自己位置推定手法によりロボットの推定位置座標 (x, y) とロボットの向き θ を推定し、目標地点までの経路を計画して自律走行する。また障害物を AI により画像認識し、ロボットは停止や回避行動を行う。

2. 2. 自律走行実験

2. 2. 1. センサ評価用ロボットの概要

ROS を用いた自己位置推定システムの動作検証をするため、LRF を搭載したセンサ評価用ロボットでの環境地図の作成や、環境地図を利用したロボットの位置推定と自律走行実験を行った。

センサ評価用ロボットは i-Cart middle³⁾ をプラットフォームとしたロボットで、ツジ電子社製の T-frog 二軸ブラシレスモータドライバ「TF-2MD3-R6」を搭載している。センサ評価用ロボットの外観を図3に示す。

TF-2MD3-R6 は、オープンソースの移動ロボット走行制御プラットフォーム「YP-Spur」と組み合わせることで、2 台のブラシレスモータを制御して二輪差動駆動の移動ロボットを容易に構築できる³⁾。ロボットの仕様を表1に示す。

自己位置推定のためのセンサとして、ロボット前方上段に GNSS アンテナ、下段に 2 次元 LRF を搭載した。



図3 センサ評価ロボットの外観

表1 センサ評価用ロボットの仕様

最大外形	L700×W490×H730[mm]
重量	21[kg]
使用センサ	GNSS 受信機、LRF、カメラ
車輪	4 輪（前方 2 輪が駆動輪、 後方 2 輪が従動輪） 前輪直径：300[mm]
ロボット主電源用バッテリー	廣隆光電社製密閉式鉛蓄電池 WP12-12（電圧 12[V]、容量 12[Ah]）を 2 個直列
ブラシレスモータ（2 個）	ツジ電子社製 TF-M30-24-3500-G200L、 TF-M30-24-3500-G200R ギヤ比 200：1
モータドライバ基板	ツジ電子社製 TF-2MD3-R6

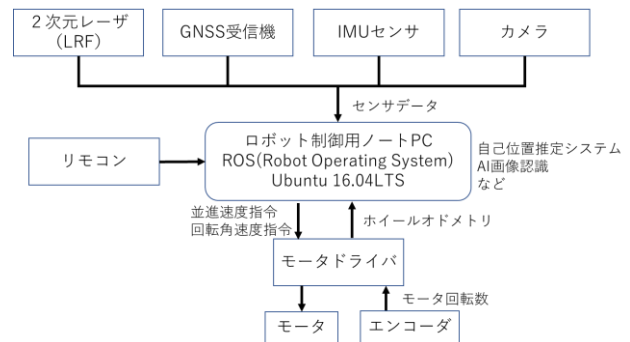


図4 システム構成図

ロボット前方中段にはカメラを搭載した。下段の LRF で周囲の建造物などの形状データを取得し、ロボットのホイールオドメトリと併せて自己位置推定を行う。ロボット走行中の前方画像を中段のカメラで取得し、AI により障害物を認識し、ロボットの停止や回避動作に利用する。システム構成図を図4に示す。

センサデータやリモコンからの信号は Ubuntu16.04 LTS と ROS を搭載したロボット制御用ノート PC で処理し、ROS ノード（ROS パッケージ内の複数の独立したプログラム）を介してロボットの速度指令をモータドライバに送信して制御を行う。

2. 2. 2. LRF による自己位置推定システムの構築

ROS を利用して北陽電機製 LRF「UTM-30LX」による自己位置推定システムを構築した。ロボットの自律走行は、2 次元の占有格子地図と呼ばれる環境地図を使用することが基本となっている。2 次元占有格子地図の作成には gmapping を使用した。2 次元占有格子地図を用いたロボットの自己位置推定には amcl を使用した。amcl はパーティクルフィルタを利用した自己位置推

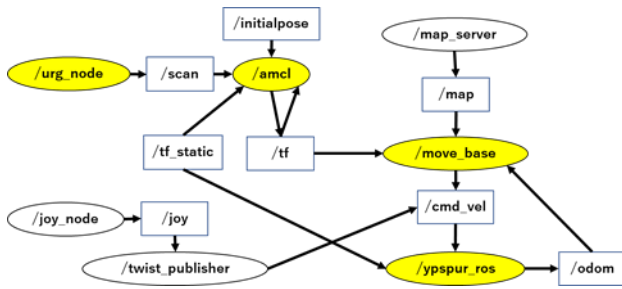


図5 自律走行時のデータ入出力

定手法である Monte Carlo Localization (MCL) に Kullback-Leibler Divergence (KLD) サンプリング等のテクニックを加えたものである²⁾。ロボットが走行する経路計画には ROS パッケージの move_base を使用した。

ROS を利用してロボットが自律走行する時のノードとトピック（ノード間の一方方向のデータ送受信経路）の関連図を図5に示す。図ではノードを丸囲み、トピックを四角囲みで表している。

yvspur_ros ノードはモータドライバからロボットのオドメトリ情報を取得し、odom トピックを出力する。amcl ノードは LRF からのスキャン情報 scan トピックを入力し、オドメトリから計算される座標系情報 tf トピックを出力する。move_base ノードはロボットへの速度指令 cmd_vel トピックを yvspur_ros ノードに送信し、ロボットのモータ制御を行っている。

2. 2. 3. 環境地図の作成

ロボットの自律走行に利用する環境地図を事前に作成するため、LRF を搭載したロボットをリモコンで操作し、研究室の2次元占有格子地図を作成した。作成した環境地図を図6に示す。約7[m]四方の研究室の環境地図を作成したところ、歪みのない環境地図が作成できた。

ハイテクプラザ敷地内の中庭において屋外での環境地図の作成実験も行った。作成した環境地図を図7に示す。約58[m] × 約22[m]の中庭の環境地図を作成したところ、地図に歪みが生じた。屋内に比べ屋外では路面が滑りやすく、広範囲の地図作成のため走行距離が長くなり、車輪の回転数から求めるオドメトリの誤差が蓄積したことが原因と考えられる。また、今回の実験では初期パラメータの gmapping で地図を作成したことも原因と考えられる。

今後、GPS や IMU センサによる補正やシミュレーションを利用した最適な gmapping のパラメータの調整により、歪みの少ない環境地図の作成を目指す。

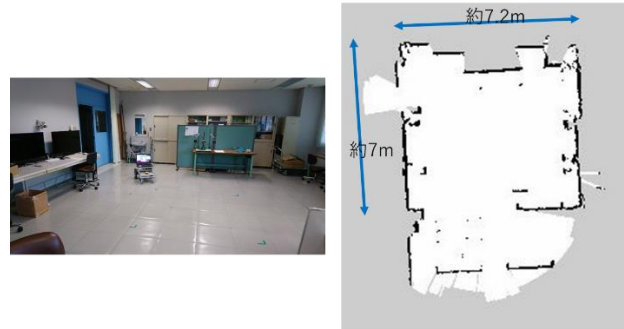


図6 研究室の環境地図

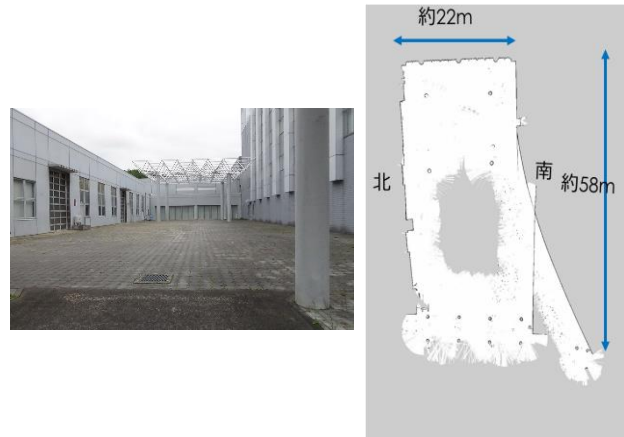


図7 中庭の環境地図

2. 2. 4. 自律走行実験

位置推定の実験では、LRF により検出した研究室内の形状と環境地図の壁やロッカーのなどの特徴的な形状をマッチングして、ロボットの位置推定ができることが確認できた。また、目標地点を設定することで現在位置から目標地点までの経路を計画して自律走行できることや、通過地点であるウェイポイントを4点設定し、4点間を自律走行することも確認した。

位置推定と自律走行実験の様子を図8に示す。

2. 3. GNSS 受信機による測位実験

2. 3. 1. GNSS 測位の課題

昨年度は GNSS を利用した自己位置推定の際に、衛星の遮蔽やマルチパスの影響を受けやすい建物の壁際や木の付近で、どの程度測位精度が劣化するかを検証するため、u-blox 社製1周波 RTK-GNSS 受信機「C94-M8P」を使用して単独測位方式と RTK (Real Time Kinematic) 測位方式の測位精度を比較した。GNSS 受信機1台で測位する単独測位に対し、RTK 測位は、基準局と移動局の2つの GNSS 受信機を使用し、リアルタイムに2点間で情報をやり取りすることで測位精度を高める測位方式である。

その結果、衛星の遮蔽やマルチパスの影響を受けやすい建物や木の付近では、単独測位方式に比べて RTK 測位方式のほうが高精度で測位できることが分かった。

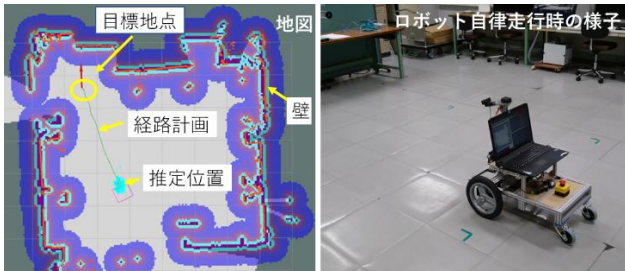


図8 位置推定と自律走行実験

しかし、RTK 測位方式は精度数 cm の Fix 解が得られるまでの初期化時間が長いことや、静止した状態では Fix 解が得られるものの移動しながらの測位では精度が数 m の Float 解に落ちてしまうという課題があることが分かった。

今年度は1周波受信機と同様に数万円で入手可能な u-blox 社の低価格2周波対応 RTK-GNSS 受信機「C099-F9P」を使用してC94-M8P との比較を行った。C099-F9P は測位衛星から発信される2つの周波数帯の電波を受信して測位する。1周波対応 C94-M8P に比べ、Fix 解を求めるまでの初期化時間が数秒と短いことや Fix 持続性が向上することが特長である。

2. 3. 2. GNSS 測位実験方法と結果

ハイテクプラザ敷地内テニスコートのオープンスカイ環境で、u-blox 社製1周波 RTK-GNSS 受信機 C94-M8P と2周波 RTK-GNSS 受信機 C099-F9P を使用してテニスコートの白線に沿って歩いた軌跡を取得し、初期化時間と Fix 持続性の比較を行った。

以下に実験条件を示す。

<実験条件>

日時：令和元年7月16日（火）15:33 - 15:52

場所：ハイテクプラザ敷地内のテニスコート（図9）

使用 GNSS 受信機：u-blox 社製 C94-M8P

u-blox 社製 C099-F9P

（RTK 測位時は2台使用）

衛星：GPS、BeiDou、QZSS

測位方式：RTK 測位

オープンスカイ環境で測位実験を行った時の受信衛星数は18、PDOPは1.39だった。位置精度低下率PDOP（Position Dilution Of Precision）は衛星数の減少や衛星配置による精度低下率の指標を表す数値である。

測位結果の解析にはu-blox社が無償提供しているGNSS 評価ソフトウェア「u-center」を使用した。測位結果を図9に示す。グーグルアース上に緑と赤の実線で示す歩行軌跡の測位結果を張り付け、2受信機の測位結果を比較した。基準局は三脚にアンテナを取り付け、テニスコートの中央（図9の×印）に設置した。

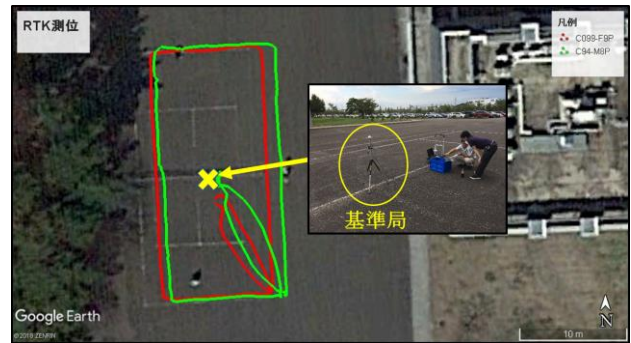


図9 RTK 測位結果（緑線：C94-M8P 赤線：C099-F9P）

実験より、測位精度の大きな違いは見られなかったが、初期化時間やFix 持続性に大きな違いがみられた。1周波 RTK-GNSS 受信機 C94-M8P は静止状態では Fix 解が得られて高精度測位できたが、移動しながらの測位では複数回 Float 解になり、静止しないと Fix 解に戻らなかった。一方、2周波 RTK-GNSS 受信機 C099-F9P では Float 解から Fix 解になるまでの初期化時間が M8P に比べて短く、移動中でも Fix 解を維持して高精度測位することが確認できた。

この実験結果より、ロボットには2周波 RTK-GNSS 受信機を搭載するほうが高精度の測位データを安定して取得できることが分かった。今後は、LRF と組み合わせた自己位置推定システムを開発する予定である。

2. 4. YOLO を用いた物体検出システムの開発

自己位置推定システムを搭載した自律走行ロボットは、搭載された LRF 等を用いて周囲に存在する物体を高精度で検出することができる。この検出物が何であるかを理解することは、ロボットの走行上において、回避や停止の動作を判断するために重要である。

本研究では、自律走行ロボットに物体検出機能を付加する目的で、畳み込みニューラルネットワーク（CNN）による動画の物体検出システム YOLO を用いて、屋外における物体検出の学習モデルを作成しその物体検出度を評価する。

YOLO^{4) 5)} は、リアルタイム物体検出アルゴリズムで、Darknet⁶⁾ や Keras/TensorFlow、PyTorch など様々なディープラーニングフレームワーク上で実装されたものが提供されており、検出精度も非常に高い。YOLO の大きな特徴は、候補領域（Bounding Boxes）の切り出しとその候補領域のクラス確率（Class Probability）の算出を一回の推測で同時に行う点にある⁷⁾。このため、YOLO は非常に高速に物体検出を行うことができる。

本研究では、Darknet で実装された YOLO-v2⁸⁾ を github のサイト⁹⁾ からダウンロードして利用した。

YOLO による画像認識実験の様子を図10に示す。実験によりロボットに搭載したカメラでリアルタイムに人や物体の認識ができることを確認できた。

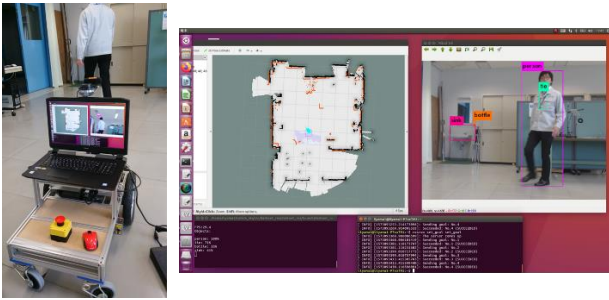


図10 YOLOによる画像認識実験

2. 5. 開発環境の構築

2. 5. 1. 自律走行評価用ロボットの製作

屋内外環境においてGNSS受信機やLRFなどの各種センサを利用した自律走行を評価するため、自律走行評価用ロボットを製作した。自律走行評価用ロボットの外観を図11に示す。

このロボットは平成30年度に製作したセンサ評価用ロボットの小型軽量化と、比較的平坦な屋内外環境でのモータ駆動走行を目指して、車輪径を300[mm]から150[mm]、ロボットの幅(トレッド長)を490[mm]から365[mm]に小型化し、ロボット重量を21[kg]から11[kg]に小型軽量化を図った設計とした。ロボットの仕様を表2に示す。

2. 5. 2. AI 認識評価システムの製作

ロボットに搭載したカメラの画像からYOLOにより物体を認識できるか評価するため、GPU搭載組み込み装置 NVIDIA Jetson AGX Xavier 開発者キット(以後、「エグゼビア」と呼ぶ)と深度カメラ Intel Realsense D435i(以後、「深度カメラ」と呼ぶ)で構成されるAI認識評価システムを製作した。

エグゼビアには NVIDIA 社から提供される JetPack をインストールして Ubuntu 18.04 ベースの OS を動作させ、CUDA、cuDNN、TensorRT、OpenCV など AI 関連のソフトウェアやドライバをインストールした。また、USB 接続した深度カメラや WEB カメラからの画像を AI 認識評価できる仕様とした。



図11 自律走行評価用ロボットの外観

表2 自律走行評価用ロボットの仕様

最大外形	L423×W365×H480[mm]
重量	11[kg]
使用センサ	GNSS 受信機、LRF、カメラ
車輪	3 輪 (前方 2 輪が駆動輪、 後方 1 輪が従動輪) 前輪直径: 150[mm]
ロボット主電源 用バッテリー	GS ユアサ製鉛蓄電池 NPH5-12 (電 圧 12[V]、容量 5[Ah]) を 2 個直列
ブラシレス モータ (2 個)	ツジ電子社製 TF-M30-24-3500-G15L、 TF-M30-24-3500-G15R ギヤ比 15 : 1
モータドライバ 基板	ツジ電子社製 TF-2MD3-R6

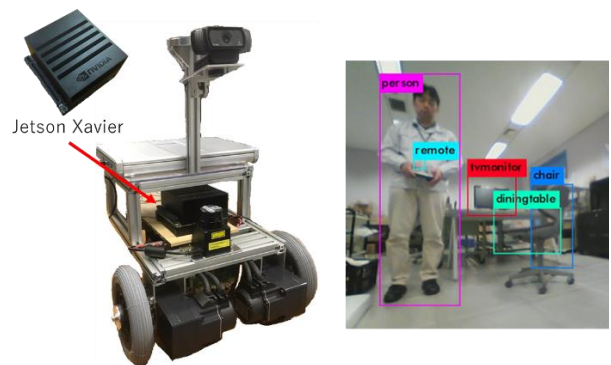


図12 YOLOでの画像認識実験

ロボットにエグゼビアを搭載して認識実験を行った様子を図12に示す。ロボットを走行させながらカメラで撮影した画像をYOLO-v2 tiny モデルで認識させたところ、約50フレーム/秒の処理スピードでリアルタイムに画像内の人や複数の物体を認識できた。今後、エグゼビアを使い、YOLO-v3での認識実験や屋外障害物の学習データを使った認識実験を行う予定である。

2. 5. 3. シミュレーション環境の構築

2. 2. 3. 章で述べたように、歪みの少ない地図作成には gmapping の最適なパラメータ設定が必須であるが、屋外ではロボット車体の振動など不特定のノイズが多く困難である。そこで、シミュレーションを活用し、最適なパラメータ値を求める。本章では、ロボットのモデル作成と自律走行シミュレーション環境の構築について述べる。

シミュレータには Gazebo を使用した。Gazebo は、ロボットの実機がなくてもバーチャルな三次元空間で姿勢や動作などを確認できるオープンソースの三次元シミュレータである。また、ROS のデータ可視化ツールである RViz を使用し、LRF のデータなど ROS 上の

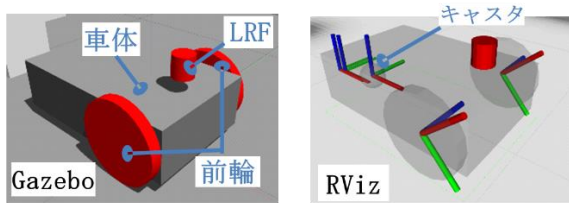


図13 シミュレーション用ロボットモデル

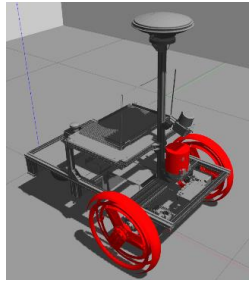
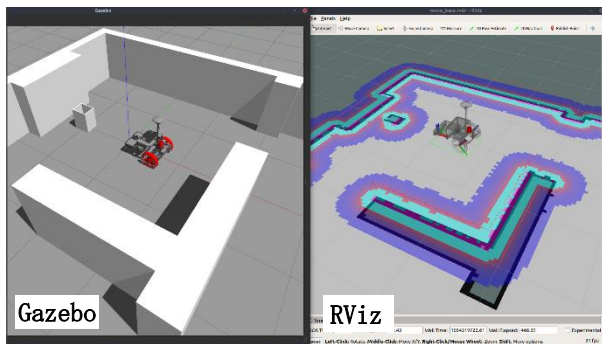


図14 ロボットモデルの外観

図15 左：構築したシミュレーション環境
右：作成した地図の表示

データを三次元空間上に可視化した。

図13に自律走行シミュレーション用に作成したロボットのモデルを示す。ロボットのモデルは、図3のロボットと同様に、車体とモータのついた前輪2輪、車体を支えるキャスト、LRF からなる二輪差動駆動型とした。ロボットの構造は、ロボットの記述ファイルである URDF (Unified Robot Description Format) と呼ばれる XML フォーマットにより記述した。URDF には車輪半径や車輪間距離等の車体パラメータや、慣性モーメント等の動力学パラメータを設定することで、シミュレーション環境上で実際のロボットの自律走行に近い動作を模擬できる^{10) 11)}。

また、図14、図15のようにロボットの外観や部屋の壁といった周囲の障害物は、各々の三次元 CAD のメッシュデータ (STL) から RViz や Gazebo の画面に表示させた。図15のように、ロボットによる地図作成や事前に作成した地図を使用したロボットの自律走行をシミュレーション可能な環境を作成した。今後は2.2.3.章で述べた図7のような地図の歪みを改善す

るため、作成したシミュレーション環境上で gmapping のパラメータの調整を行っていく予定である。

3. 結言

今年度は LRF による自己位置推定システムを構築し、このシステムを搭載したセンサ評価用ロボットで屋内の環境地図作成と走行実験を行った。その結果、屋内の壁や机などの障害物を LRF でスキャンした環境地図を作成し、LRF で環境地図とスキャンデータをマッチングしながらセンサ評価用ロボットが自律走行できることを確認した。また、屋外で LRF とホイールオドメトリのみで環境地図を作成した結果、環境地図に歪みが生じることが分かった。これは屋内に比べ屋外では車輪の滑りや LRF のノイズの影響が大きく、歪みが生じたためと考えられる。今後は、シミュレーション上での gmapping のパラメータの調整や、GPS、IMU センサによる補正を行い、歪みの少ない環境地図の作成を目指す。

自己位置推定に利用する GNSS 受信機の実験では、1周波と2周波対応 GNSS 受信機を使ってオープンスカイ環境で RTK 測位実験を行い、測位精度、初期化時間や Fix 持続性について比較した。その結果、2周波対応 GNSS 受信機では初期化時間及び Fix 持続性が改善されることが分かった。

さらに、自律走行ロボットに物体検出機能を付加する目的で、深層学習による物体検出システム YOLO を用いて人や物体をリアルタイムで認識することができた。

今年度得た知見を利用して、今後は GNSS と LRF を組み合わせた ROS による自己位置推定システム及び AI による物体認識の開発を行い、屋内外で自律走行の実験を行っていく予定である。

参考文献

- 1) 原 祥堯, “ROS を用いた自律走行”, 日本ロボット学会誌, Vol.35 No.4, p.286~290, 2017.
- 2) 上田隆一, “移動ロボットのための ROS パッケージの紹介と実機への導入方法”, 計測と制御, 第57巻 第10号 2018年10月号, p.715~720.
- 3) 渡辺敦志, “移動ロボット用二軸ブラシレスモータドライバの開発とオープンソースプラットフォームを用いた走行制御”, <http://www.roboken.esys.tsukuba.ac.jp/~ohya/pdf/RSJ2013-AWD.pdf>.
- 4) Joseph Redmon. “YOLO”. YOLO:Real-Time Object Detection. <https://pjreddie.com/darknet/yolo/>. (参照 2019-04-03).
- 5) Redmon, Joseph, et al. “You only look once: Unified, real-time object detection.”. arXiv

- preprint arXiv:1506.02640,2015.
- 6) Joseph Redmon. “Darknet” .Darknet: Open Source Neural Networks in C. <https://pjreddie.com/darknet/>. (参照 2019-04-05).
 - 7) 藤田一弥, 高原歩:実装ディープラーニング, オーム社, (2016).
 - 8) Joseph Redmon and Ali Farhadi. “Yolo9000: Better, faster, stronger”.arXiv preprint arXiv:1612.08242, 2016.
 - 9) GitHub. “pjreddie” . pjreddie (Joseph Redmon). <https://github.com/pjreddie>. (参照 2019-04-03).
 - 10) srs. “ROS講座33 車輪ロボットを作る1(rvizで表示)” . Qiita. <https://qiita.com/srs/items/c1065543228e95be1a0f>.
 - 11) 西田健, “実用ロボット開発のための ROS プログラミング”, 森北出版株式会社 p.75～88, 2018.