

# 屋外自律移動ロボット Progress-i の開発

帝京大学 山根研究室

遠山 貴則<sup>†</sup>, 横松 秀康, 野村 佳太, 山根 健<sup>‡</sup>

連絡先: <sup>†</sup>136219@uccl.teikyo-u.ac.jp, <sup>‡</sup>yamane@ics.teikyo-u.ac.jp

Report for Tsukuba challenge 2015 (Teikyo University)

Takanori TOYAMA, Hideyasu YOKOMATSU, Keita NOMURA and Ken YAMANE

## 1. はじめに

本稿では、つくばチャレンジ 2015 における帝京大学山根研究室の取り組みについて報告する。我々は初参加であるだけでなく、今まで自律移動ロボットを開発した経験はほぼ無い。過去のレポート[1]と重複する内容も含まれるが、初参加であることを意識して内容を選び報告する。

**研究スケジュール:** 年間スケジュールを図 1 に示す。2015 年 6 月までは、要素技術に関する文献調査と過去のレポートから各チームのロボットを分析した。これらに基づいて、システムを計画、設計し、6 月以降はロボットを製作して、実験をしながらシステムの問題点の把握と改良を行った。結果的に、最も多くの時間を費やした作業はシステム統合の部分であった。また、走行会や交流会では他大学のロボットや技術について調査した。

**研究体制:** 我々のチームは学部 2~4 年生を中心にした 3 名および指導教員 1 名で構成され、それぞれの学生に担当が割り当てられる。1 年間の時限付きプロジェクトとして、週 1 回のミーティングと日々の自主的活動を行う。ミーティングにおける進捗状況の報告、研究室 Wiki を利用した作業の記録などにより、メンバー間で情報を共有する。

**研究の大目標:** 我々は大規模な太陽光発電施設[2]における効率的な点検・故障検知のために、太陽光パネル上に発生する周囲よりも温度が高い部分(ホットスポット)を熱画像から抽出する方法について検討し、自律移動ロボットを用いて熱画像を撮影して解析用サーバに送信するシステムの構築を目指している。平成 26 年度の取り組みでは、市販のラジコンカーと GPS を用いてグローバルな位置情報

に基づいて自律移動するロボットカーを開発し、その性能を調べた。その結果、GPS の精度の問題などにより限界があることがわかった[3]。

**つくばチャレンジにおける目標:** そこで平成 27 年度では、つくばチャレンジなどで広く用いられている、オドメトリや測域センサの情報と環境地図から自己位置を推定して自律走行する方法に変更した。また、つくばチャレンジ 2015 における走行実験や情報交換を通じて、自律移動ロボット Progress-i を製作し、性能を検証する計画を立てた。具体的な目標は、つくばチャレンジ 2015 の確認走行区間(大清水公園内のコース)を安全かつ確実に走破することである。以下ではこの目標達成に向けた我々の取り組みについて述べる。

## 2. 自律移動ロボット Progress-i の開発

基本的な方針として、できる限りオープンソースのものを利用してロボットを製作する。ハードウェアに関しては、T-frog プロジェクトが開発した独立 2 輪駆動方式のロボットフレーム i-Cart mini[4]を利用する。また、ソフトウェアに関しては、筑波大学知能ロボット研究室において開発された走行制御プログラム(Yamabico Project Spur, YP-Spur)、リングバッファ状の共有メモリソフト(Streaming data Sharing Manager, SSM)などのロボットプラットフォーム[5]を、環境地図作成や自己位置推定については C++言語のライブラリである Mobile Robot Programming Toolkit (MRPT) [6]を利用する。

### 2.1 ハードウェアの開発

我々が開発したロボット Progress-i の外観を図 2 に、内部を図 3 に、ハードウェアの構成を表 1

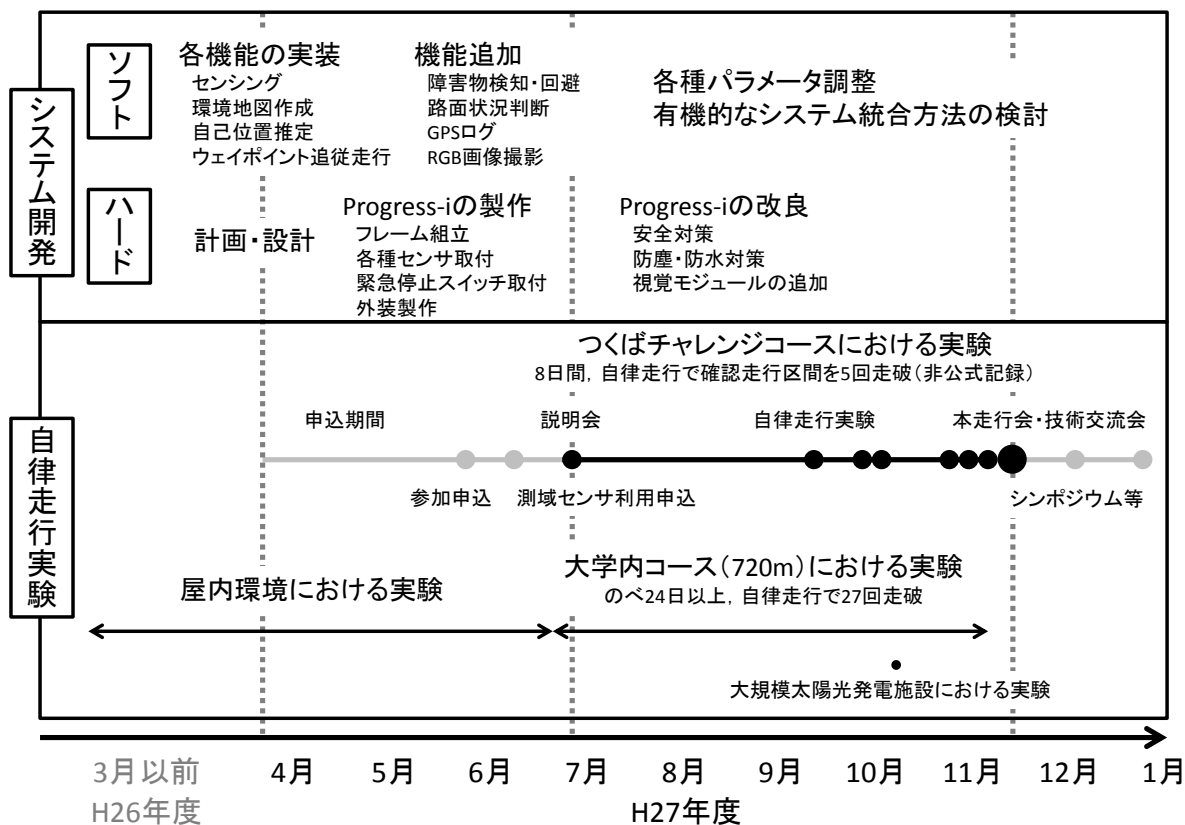


図 1 : 年間スケジュール

に示す。オリジナルの i-Cart mini[4]から主に以下の改造をしている。なお、技術的な情報の収集のために T-frog プロジェクトのフォーラムを利用した。

- ・ キャスタの交換
- ・ アルミフレームの更新
- ・ 各種センサの取り付け
- ・ 緊急停止スイッチの取り付け
- ・ 防塵・防水対策

**アルミフレーム**：ロボットの存在を確認しやすいように 0.6m 以上の高さが必要になる。そこで、外装を含めたロボットのサイズが全長 0.6m×全幅 0.52m×全高 0.8m となるようにアルミフレーム（株式会社ミスミ製）を更新する。

**外装**：外装には、ホームセンタなどで購入可能な厚さ 4mm のプラスチックダンボール（以下、プラダンと呼ぶ）を用いる。加工しやすく、軽量であり、丈夫な点がプラダンを使用する利点である。運搬時に折り畳むことを考慮して 5 つのパーツに分解できるように設計し、ロボット本体への取り付け方を

工夫することで巻き込みを防止する。また、L 字タイプや挟み込むタイプなどの緩衝材を取り付けることで、万が一、歩行者や他のロボットに接触してしまった場合でも相手を傷つけない工夫をする。

**緊急停止スイッチ**：プラダン外装以外の安全対策として、緊急停止スイッチの取り付けが必須である。スイッチを押すと左右のモータへの電源供給が遮断される単純な方式を我々は採用した。なお、赤色のスイッチを選定したので十分目立つと考えていたが、安全確認時にスイッチの周囲を黄色のテープで巻いて目立たせるよう指摘を受け、直ぐに修正した。また、リレーを用いたブレーキ回路などの設計、製作も検討している。

**防塵・防水対策**：モータドライバや電源基板など重要なパーツは全て防水ボックスに収納する。比較的安価なマイコンなどについては、防滴効果を期待して食品保存などに使われるプラスチック製容器を改造したケースに収納する。また、外装のつなぎ目の隙間やケーブルの接続部分に防水テープを貼



図 2 : Progress-i の外観



図 3 : 内部の構成

り、内部を調理用ラップシートで覆う。

**走行機構**：本ロボットの大きな問題は走破性が低いことである。つくばチャレンジの課題の 1 つである横断歩道の段差を乗り越えられないだけでなく、会場のバックヤードからスタート地点へ向かう間にある段差も乗り越えられない。ただし、確認走行区間において問題は生じないため今後の課題と

表 1 : ハードウェアの構成

構成要素	メーカー・型番・スペックなど
Main PC	Apple Mac Book Air Intel Core i7 1.7GHz Memory 8GB
OS	Ubuntu 14.04 LTS
Sub PC	Raspberry Pi 2 Arduino UNO R3 × 2
LRF	Hokuyo UTM-30LX-EW Hokuyo URG-04LX
IMU	Adafruit 10-DOF IMU Breakout
US	Sain Smart HC-SR04 × 3
GPS	Adafruit Ultimate GPS
Camera	Raspberry Pi Camera FLiR Dev Kit
Motor Driver	ツジ電子 TF-2MD3-R6
Motor	ツジ電子 TF-M30-24-3500-G15L/R パルスエンコーダ付
Joystick	Lanjue L600
Battery	GS YUASA NPH5-12 × 2 12V5A

する。この問題に対しては、タイヤの径を大きくし、ギアを変更する計画である。また、坂道の直線走行において、モータドライバのトルク制限などにより、少しずつ左へずれる現象が起こる。これは、モータドライバのファームウェアの更新とパラメータの調整によって少し解消できる。

**測域センサ**：環境情報を取得するため、北陽電機の測域センサ UTM-30LX-EW の貸出を受け（後に同品を購入）、走行面から 41.85cm の高さに水平に取り付ける。また、路面状況を判断するため、北陽電機 URG-04LX を走行面から 31.98cm の高さにピッチ角 30.5deg で取り付ける。それぞれの上部に雨よけのひさしを取り付ける。

**IMU センサ**：また、IMU センサ（加速度、ジャイロ、地磁気、気圧、温度センサ）、超音波距離センサおよび GPS を内部に取り付け、それぞれ 20Hz、20Hz、10Hz のサンプリングレートで AVR マイコン Arduino UNO R3 を介してメインコンピュータに情報が送信される。情報は SSM[5]を用いて複数

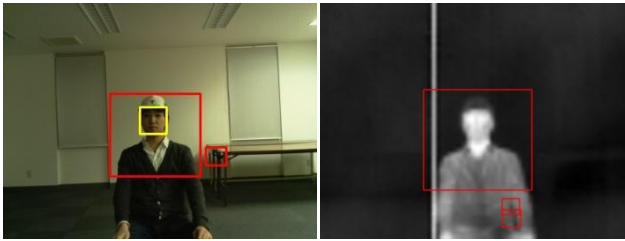


図 4 : RGB および熱画像からの人検出

のプロセス間で共有される。

**視覚モジュール**：さらに、ARM プロセッサを搭載したマイコン Raspberry Pi 2、サーボ 2 つで構成されたパン・チルト機構、RGB カメラ (解像度 2592 × 1944 pixel) および FLiR 製の熱画像カメラ (解像度 80 × 60 pixel) を用いて視覚モジュールを構築し、ロボットの上部に取り付ける。なお、熱画像カメラは我々の大目標を実現するために利用している。また、図 4 のように相対的に表面温度が高い部分を検出できることから、RGB カメラと組み合わせることで人の顔や体を認識する方法について検討している。

**コンピュータ**：制御用のコンピュータとして Apple Mac Book Air (2013 年モデル:1.7GHz 2 コア Intel Core i7, メモリ 8GB, OSX Yosemite, Ver.10.10.5) を用いる。当初の計画では、より小型の B5 サイズの Panasonic Let's note (CF-AX3SEGJR:1.6GHz 2 コア Intel Core i5, メモリ 4GB, Windows8.1) を用いる予定だった。しかし、モータドライバとの通信の不具合やインテルターボ・ブースト・テクノロジー2.0 機能[7]などをうまく利用できない問題が発生し、8 月頃に途中で断念した。

**その他**：ロボットの製作とは無関係であるが、走行会の現場では、コンピュータなどを充電するためのポータブル電源、バックヤード用のテント、椅子、レジャーシートなどの物品が大変役に立つ。テントには強風対策も必要である。

## 2.2 ソフトウェアの開発

メインコンピュータ上に VMware Fusion (バージョン 7.1.3) を使って仮想化された Linux 系 OS・Ubuntu14.04 LTS を用いてシステムを構築する。主に C++言語を用いてソフトウェアの開発を行う。

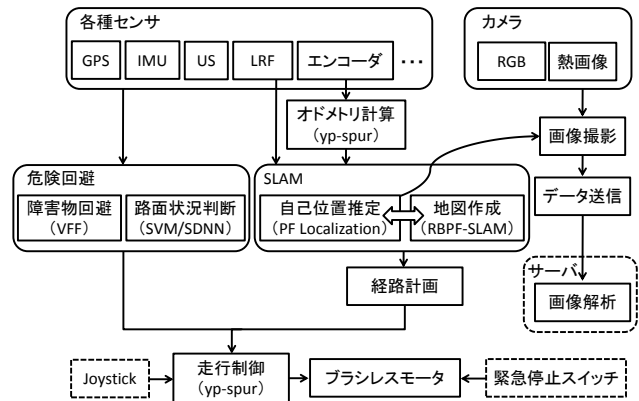


図 5 : 開発するシステム

これらに関する技術的な情報は豊富にあり開発環境を整える際に参考にできる。

開発手順は、まず、各機能を持つモジュールを独立に開発し、研究室など限られた環境においてそれぞれ実験する。次に、その結果に基づいてパラメータ調整を行う。その上で、システムを統合し、最終的に屋外環境において自律走行実験を行い、システム全体としての性能を確認する。

つくばチャレンジ2015のために開発するシステムの構成を図 5 に示す。メインループの計算時間は 1 周期 27ms 程度である。

**環境地図の作成**：測域センサとオドメトリ情報から SLAM 技術[8,9]を用いて 2 次元占有グリッドマップおよびポイントマップを作成する。実験では MRPT[6]の RBPF-SLAM[10]と ICP-SLAM など複数の方法を検討して、RBPF-SLAM を採用した。パラメータは取得したデータとシミュレータを使った繰り返し実験によって決定する。また、走行した軌道上にウェイポイントを一定間隔で設定する。地図の解像度は 0.1m とする。

**自己位置推定**：パーティクルフィルタ [8,9]を用いて自己位置推定を行う。このパラメータも実験的に決定する。計算機の能力を考え、パーティクル数を 10、推定誤差を 0.1m、姿勢誤差を 5 度程度とする。

**ウェイポイント追従走行**：推定された自己位置とウェイポイントまでの距離および方位を計算し、目標方位と現在の姿勢の差に応じて速度制御 (0.6 ~ 0.15m/s) し、現在位置から目標位置までの直線を

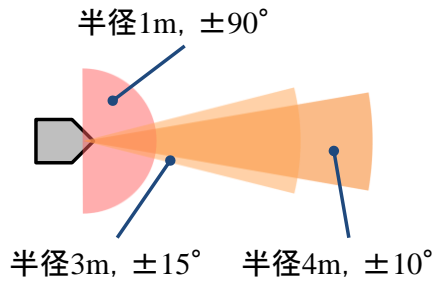


図 6：障害物を検知する範囲

追従するように走行する。ただし、自己位置の推定精度が低いスタート直後 10 秒間だけは速度  $0.15\text{m/s}$  に固定する。また、ウェイポイントの半径  $0.2\text{m}$  以内に入った場合は次のウェイポイントに向かうように再設定する。さらに、目標方位とロボットの姿勢の差が  $11.25\text{deg}$  を超えた場合はその場で回転して姿勢を修正する。なお、回転角速度の最大を  $45\text{deg/s}$  とする。

**障害物検知・回避：**障害物を検知する範囲を図 6 に示す。図中の赤色の領域は衝突の危険性が高いエリア、橙色は衝突の危険性は中程度であるが注意が必要なエリアである。これらの範囲については走行実験を繰り返して調整するため、具体的な角度や距離に関する根拠はない。障害物回避には Virtual Force Field 法 (VFF 法) [11]を用いて、目標とする方位と速度 ( $0.15\sim 0.00\text{[m/s]}$ ) を決定する。

**路面状況判断とそれに基づく速度制御：**測域センサからの情報を用いて、近傍点の値との差分を取ることによって段差を検知する。また、多数のセンサ情報を入力として、走行している路面の状況を識別し、識別した路面状況に基づいて速度制御する方法について検討する。具体的には、入力変数として斜め下向きに設置した測域センサの情報(前方の高さの平均値など 2 次元)、並進速度、回転速度、Z 軸方向の加速度、ヨー角、ピッチ角、ロール角の 8 次元の情報を前処理して正規化したものを用いる。識別器としてサポートベクタマシン (SVM) [12]や選択的不感化法を適用した層状ニューラルネットワーク (SDNN) [13]を検討する。しかし、路面を識別するためには実際にその路面をロボットが走行する

必要がある、路面状況と速度の関係が曖昧であるなどの課題が残されるため、自律走行の時はメインシステムから独立して動作させる。

**自動画像撮影：**画像処理ライブラリ OpenCV[14]を用いて指定した地点で画像撮影して解析する機能を実現する。しかし、学内実験において、Raspberry Pi 2 の電源が不安定である問題や SSM を用いたメインコンピュータとの通信における問題が発生した。そこで、1 秒間隔で画像を撮影してマイコンに保存する機能まで実現し、走行実験ではログ取得用としてロボットに搭載する。

**音声案内機能：**ロボットの進行方向が大きく変化するとき、歩行者へロボットの走行意図を提示する目的で音声案内機能を実現する。音声ライブラリとして Open JTalk[15]を用いて、「出発します」、「停止します」、「左に曲がります」などの音声をスピーカから出力する。しかし、音声案内の頻度によっては周囲の人に煩わしく感じさせる上に、メインプロセスとの干渉などの問題が発生したため、ゴール地点に到達して停止したときのみ音声によってそれを知らせる仕様にする。

### 3. 自律走行実験

#### 3.1 帝京大学トライアルコースにおける実験

つくばチャレンジ 2015 の準備として、帝京大学宇都宮キャンパス内にトライアルコースを設定して実験を行った。具体的には、自動車や人通りがあり、緩やかな坂を含む全長  $710\text{m}$ 、走行時間  $20\sim 30$  分の屋外コースにおいて、2015 年 7 月 23 日 (木) ~ 11 月 2 日 (火) の 24 日間、実験を繰り返して行った。自律走行ソフトウェアの改良を続けた結果、安全に走行できるようになり、この期間中にコースを 27 回走破した。作成した環境地図を図 7 に示す。走行する経路は全体として大きなループを描くため、スタート位置とゴール位置が一致する地図を作成することは困難であった。複数の地図を切り替えて利用する方法を検討する必要がある。

本コースにおいて路面の高さが急激に変化するような段差を検知することができた。その結果を図 8 に示す。図中の緑色部分は溝や起伏がある部分を



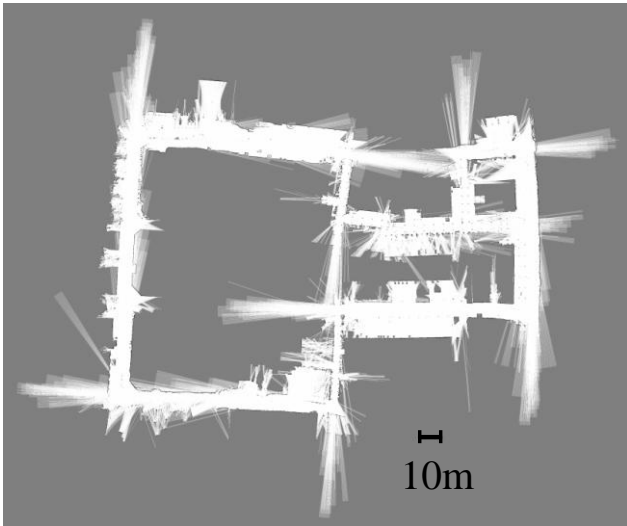


図 7：学内コースのマップ

示している。また、この付近で路面を識別した結果を図 9 に示す。平坦な渡り廊下からアスファルト、そして砂利道を正しく識別できた。未知の場所においても、路面状況を正しく識別可能かどうかは今後の課題である。

### 3.2 つくばチャレンジ 2015 における走行実験

我々は、8 日間の走行会の全てに実験参加した。第 2 回走行会まではコースの確認、他大学の取り組みの調査、マニュアル走行によるデータログの取得、環境地図の作成を行った。3 回以降は主に自律走行実験を行い、各機能のパラメータ調整、システム統合方法の検討を行った。非公式記録ではあるが、確認走行区間を自律走行した全 59 トライアル中 5 回安全に走破することができた。その中で 11 月 8 日に行った 5 回目のトライアルの結果を図 10 に示す。また、我々の実験結果を図 11 にまとめる。図中の棒グラフは各トライアルの到達距離 (m) を、折れ線グラフは各回の平均到達距離 (m) を示す。

本走行では、予定どおり時刻 11 時 39 分にスタートし、先に出発していた 1 台のロボットを 200m 付近で追い抜いた。その後、大清水公園内 270m 地点 (確認走行区間の最後の直線) において芝生エリアに進入したため、緊急停止スイッチがオペレータによって押されて終了した。走行時間は 11 分 9 秒であった。本走行後にデータログを解析した結果、走行中に芝生エリアに進入した理由はノートパソ

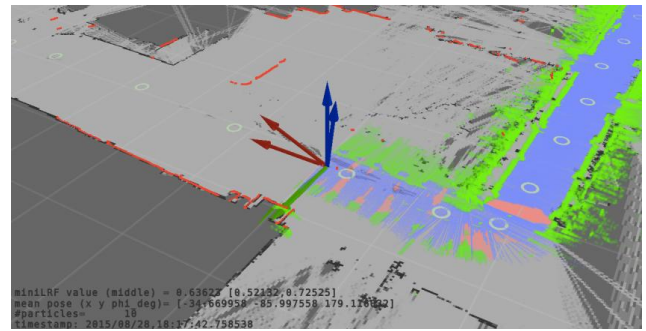


図 8：段差の検知

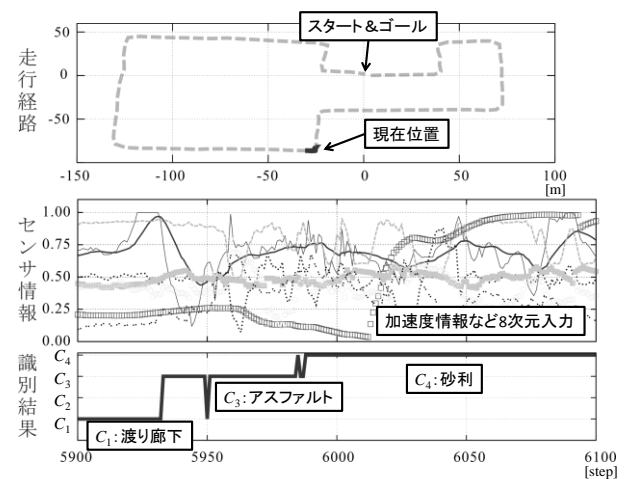


図 9：路面状況判断

コンと測域センサ間の通信が切断され、オドメトリ情報のみで自己位置を推定せざるを得ず、自己位置の精度が低下したためだとわかった。

### 3.3 考察

本走行の結果から、基本的な屋外走行には問題がなかったこと、測域センサにひさしを取り付けたこともあり雨の影響も少なかったことが確認できた。また、走行に関わる機能群の統合、一部の要素に障害が生じた時の対応など今後の課題が明確になった。

**外装・安全対策：**あるロボットから追い抜きをかけた際、後部から側面にかけて衝突された。プラダンに相手のタイヤ痕が付いた程度で、外装が凹むことで衝突が軽減されて内部へのダメージはほぼ無かった。以上からプラダンを用いて最低限の外装を製作できることがわかった。一方で、振動がやや大きく、摩擦による音鳴り、ジョイスティックなどの落下、カメラ画像のぶれ、USB ケーブルの抜

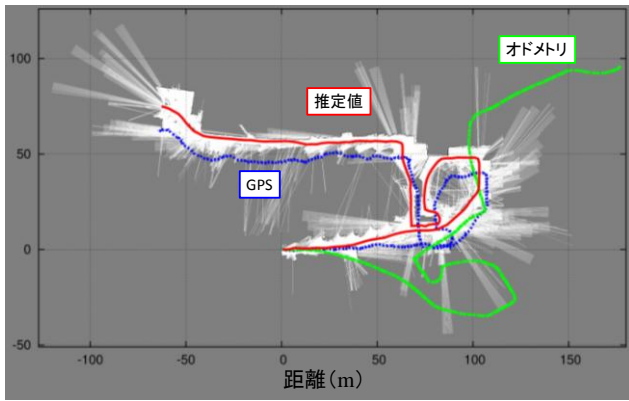


図 10：大清水公園内における自律走行結果

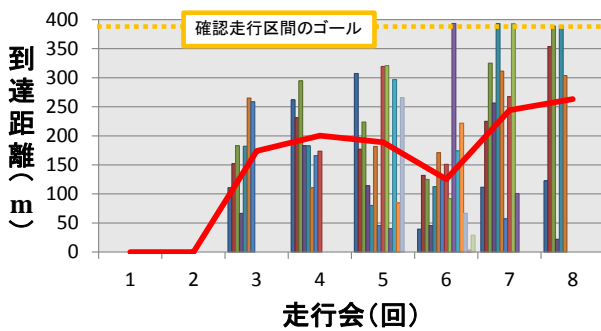


図 11：走行実験結果

けなどが発生した。また、坂道などにおいてロボットの姿勢が変化した場合に後部のキャストと接触して摩擦音が生じた。

**雨の影響：**実験時に小雨が降ったり、本走行では本格的な雨が降ったりした。しかし、浸水による深刻な問題は生じなかった。また、ひさしを取り付けたこともあり、雨粒の測域センサへの影響は小さかった。むしろ、太陽光の悪影響が出なかったため、斜め下向きに取り付けた測域センサから高い精度で路面状況が取得できた。一方で、視覚モジュールについては、取り付け方法、雨対策などが不十分であったため、本走行では取り外した。

**各種パラメータの設定：**2.2 で述べた各機能のパラメータを決定することに多くの時間を割いた。シミュレーションあるいは学内コースにおける事前の実験から最適だと思われるものを使用した。しかし、走行会ではうまく行かないことが多かったため、現場で試行錯誤的に決定した。具体的には、SLAM

のためのパラメータは 30 以上、自己位置推定のものは 20 以上あり、本走行までに何度も繰り返し実験を行ったが、未だ不明な点が多い。

**環境地図の歪み：**坂など傾斜がある路面、凸凹道、正面が階段の場所、大きなループが閉じる場所、公園などのオープンスペースで地図が歪む傾向があった。また、回転量あるいは方位の推定に誤差が蓄積される傾向があった。歪みのない地図を作成するためには、オドメトリ情報の補正や正確な姿勢情報の取得についての検討が必要である。

**自己位置推定の精度：**多くの場合、自己位置推定の精度が低下して、コースを外れて縁石や芝生エリアに進入してスタックすることで失敗していた。これは、コンピュータと測域センサの通信が切断された場合に多発した。通信の問題に関しては、現在、原因を究明している。また、ロボットの周囲に特徴が少ないようなオープンスペースにおいても発生した。具体的には、スタートから 100m 付近のエリアにおいて精度が低下する傾向にあった。その他に、人工的な構造物が少なく自然地形が多い場所、走行路面に起伏があるような場所でも発生した。一方で、学内コースでは自己位置の精度が走行に影響するほど低下することはほとんど無かった。

**障害物回避の戦略：**前方で停滞気味のロボットに近づいた場合の追い抜きが我々にとって難しい課題であった。相手との相対速度が十分大きいか、相手が完全に停止していれば問題は無かった。しかし、そうでない場合の良いアイデアを我々は持っていなかった。本走行後、技術交流会における意見交換から、街中を走行するロボットは高い安全性や確実性が求められるため、障害物回避時の走行速度を極端に遅くする戦略にメリットが多いことを学んだ。

**システム統合方法の検討：**我々が最も時間を費やして検討したことは、各機能をどのように統合していくかということである。例えば、走行制御に関わるウェイポイント追従走行機能、障害物回避機能、路面状況判断機能を独立に調整することはうまく行った。しかし、それらを IF-THEN ルールのような論理を用いて統合する方法はあまりうまく行か

ない。今後、探索対象への接近や横断歩道における走行などへの対応もあり、機能が増加する毎に考えるべき状況は爆発的に増加するため、それら全ての状況を考慮して論理を構築することはとても難しい。現在、複数の機能が相互作用することで機能が柔軟に調整される方法を検討中である。

**その他:** 次のことも考慮する必要がある。落ち葉、枝、どんぐりなどに乗り上げてスリップすることがあった。また、天候によっては水たまり、泥によるぬかるみが広がり大きな障害になる。これらの状況は実験期間中に大きく変化する。さらに、本走行では大勢の観客がスタート地点に集まることから、普段と大きく異なる状況が生じる。これら想定外の状況もできる限り対応する必要がある。

#### 4. まとめ

我々は街中において自律移動ロボットを走行させる実験に初めて参加し、オープンソースのハードおよびソフトウェアを利用して、屋外自律移動ロボット Progress-i の開発を行った。実験の結果、様々な課題が明らかになった。今後これらについて 1 つ 1 つ検討していく予定である。具体的には以下の課題が残されている。

- ・ 安定性・走破性の向上
- ・ センシング方法の工夫
- ・ システム統合方法の検討
- ・ 画像撮影モジュールの改良

我々は今後も安全かつ確実に自律走行できる場所を広げるべく Progress-i の開発を継続する。

#### 謝辞

我々の取り組みを支援して頂いた、帝京大学工学部学部長・波江野勉先生、同大理工学部情報電子工学科長・飽本一裕先生に感謝申し上げます。

#### 参考文献

- [1] つくばチャレンジ 2014 参加レポート集, つくばチャレンジ実行委員会 (編), 2014.
- [2] 帝京スマートエネルギーセンター竣工式, [https://www.teikyo-u.ac.jp/news/2015/0122\\_4133.html](https://www.teikyo-u.ac.jp/news/2015/0122_4133.html), 2015 年 1 月, 2015 年 12 月アクセス.
- [3] 遠山貴則, 野村佳太, 針ヶ谷優, 橋爪悠, “施設

設備点検用ロボットカーの開発,” 第 11 回大学コンソーシアムとちぎ学生&企業研究発表会予稿集, pp.130-131, 2014 年 12 月.

[4] T-frog, <http://t-frog.com>, 2015 年 10 月アクセス.

[5] 知能ロボット研究室 Robot Platform Project, <http://www.roboken.iit.tsukuba.ac.jp/platform/wiki/>, 2015 年 12 月アクセス.

[6] MRPT, <http://www.mrpt.org>, 2015 年 10 月アクセス.

[7] インテル・ターボ・ブースト・テクノロジー2.0, <http://www.intel.co.jp/content/www/jp/ja/architecture-and-technology/turbo-boost/turbo-boost-technology.html>, 2015 年 12 月アクセス.

[8] S.Thrun, W.Burgard and D.Fox, Probabilistic robotics, The MIT Press, 2005.

[9] 友納正裕, “移動ロボットのための確率的な自己位置推定と地図構築,” 日本ロボット学会誌, vol. 29, no.5, pp.423-426, 2011.

[10] A. Doucet, N. de Freitas, K. Murphy and S. Russell, “Rao-Blackwellized Particle Filtering for Dynamic Bayesian Networks,” Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence, pp.176-183, 2000.

[11] J.-L. Blanco, J. Gonzalez and J.-A. Fernández-Madriral, “Extending Obstacle Avoidance Methods through Multiple Parameter-Space Transformations,” Autonomous Robots, vol.24-1, pp. 29-48, 2008.

[12] C.M.ビショップ, パターン認識と機械学習(下), シュプリンガー・ジャパン, 2008.

[13] 野中和明, 田中文英, 森田昌彦, “階層型ニューラルネットの 2 変数関数近似能力の比較,” 電子情報通信学会論文誌(D), vol.J94-D, no.12, pp. 2114-2125, 2011.

[14] OpenCV, <http://opencv.jp/>, 2015 年 12 月アクセス.

[15] Open JTalk, <http://open-jtalk.sourceforge.net/>, 2015 年 12 月アクセス.