

自律走行可能なロボットの制御手法の開発（第2報）

中込広幸・布施嘉裕・保坂秀彦・宮本博永・中村卓・米山陽・神村明哉^{※1}・渡辺寛望^{※2}・丹沢勉^{※2}・小谷信司^{※2}

Development of Crawler Robot System for Autonomous Mapping (2nd Report)

Hiroyuki NAKAGOMI, Yoshihiro FUSE, Hidehiko HOSAKA, Hironaga MIYAMOTO, Takashi NAKAMURA, Akira YONEYAMA, Akiya KAMIMURA, Hiromi WATANABE, Tsutomu TANZAWA and Shinji KOTANI

要 約

県内中小企業のロボット試作開発支援のため、ロボットの設計・制御・自己位置推定に関するノウハウの蓄積、および技術移転が容易な自律走行システムの開発が求められている。本研究では、製品への取り付けが容易なセンサであるステレオカメラおよびIMUによるオドメトリ算出手法の開発を行い、占有格子地図とレーザレンジファインダの情報をもとにパーティクルフィルタを用いた自己位置推定を行うシステムの構築を行った。評価として、並進時に滑りが発生しない環境において、ロータリエンコーダを利用した自己位置推定手法と本手法を利用した自己位置推定を行い、目標位置に対する誤差を評価した。その結果として、ロータリエンコーダによる自己位置推定と同程度の精度を得ることが可能であり、本手法をロボットの自律移動に利用することが有効であることを確認した。

1. 緒 言

県内中小企業の特徴として、大手企業のOEMや、製造現場専用の産業用ロボット、その他組込技術を応用した製品の開発を受託する企業が多く存在する。一部の企業では、このような受託開発で蓄積した技術を、走破性の高いクローラ型ロボットに搭載し、自律移動を行うことで、付加価値の高い自社製品の開発を行うことを検討している。これら中小企業では、人材育成・期間・資金等の余裕が少ないため、ロボットの設計・制御・自己位置推定に関するノウハウの提供が求められている。

このため、本ロボットシステム開発では、技術移転が容易、かつ路面状態の影響を軽減可能な汎用性の高いロボット開発を目指している。本研究を通して、クローラ型汎用プラットフォームの試作開発、および路面状態の影響を受けにくいセンサ類やレーザレンジファインダ等を用いた地図作成および地図上での自己位置推定を行うシステムを構築する。

本報は、第1報¹⁾にて開発したシステムを用いて地図作成を行い、地図上における位置推定および自律移動を行うシステムについて報告する。ロボットの自己位置推定において、レーザレンジファインダの測定範囲よりも広い空間での自己位置推定は困難であることから、車軸に取り付けられたロータリエンコーダと、IMUによる位置姿勢の変化量（オドメトリ）の測定が補助的に利用

される。ロータリエンコーダによるオドメトリの測定は、デッドレコニングと呼ばれ、自己位置推定に利用されるパーティクルフィルタ（時系列処理を行う粒子フィルタ）の更新に利用される。このデッドレコニングを利用した自己位置推定は、自律移動ロボットに良く利用されるが、県内企業への技術移転を考慮すると、ロータリエンコーダは、路面状態の変化に影響される上、取り付けのための設計開発が発生することから、技術移転時に大きな労力を伴うことが予想される。

そこで、本研究では、ステレオカメラを利用したオドメトリ（ステレオビジュアルオドメトリ法）とIMUを組み合わせたオドメトリ算出手法を提案する。ステレオカメラは取り付けが容易であり、かつ非接触にオドメトリ取得が可能のため、企業への技術移転において有効である。一方で、ステレオビジュアルオドメトリ法は、旋回時のような視線が大きく変化する際、オドメトリ算出が困難になるといった課題がある。そこで旋回時においても正確なビジュアルオドメトリの算出を目的に、これら手法とIMUによる姿勢推定手法を組み合わせた新たな手法を提案した。

2. ハードウェア構成

実験用クローラロボットとして、Nexus Automation社製 New Tracked Mobile Tank Robot Kit 10022を使用した。各種処理に、PC（CPU: 1.6 GHz Intel Core i5, コア数: 4メモリ: 4 GB）を利用した。センサとして、レーザレ

※1 国立研究開発法人産業技術総合研究所

※2 国立大学法人山梨大学

レンジファインダ（北陽電機製 UHG-08LX），ステレオカメラ（PointGray 社製 BumbleBee2），IMU（SparkFun Electronics 製 9DOF Razor），車軸に搭載されているロータリエンコーダを使用した（図 1）。ステレオカメラは水平に対して約 30° 下向きに設置した。本ステレオカメラの視差は 12 cm であるため，1 m から 3 m ほどの範囲内において距離測定精度が高い。水平よりやや下向きに設置することで，この範囲内の床，壁等の物体を優先的に撮像し，自己位置推定の精度向上が可能となる。

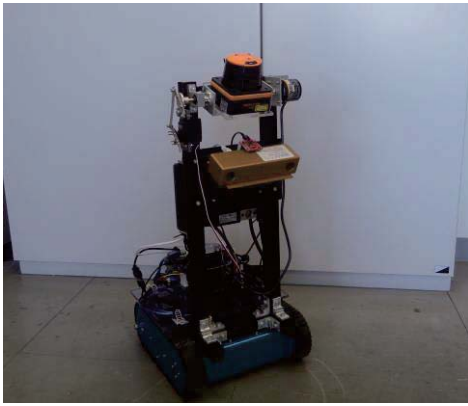


図 1 実験に利用したクローラ型ロボット

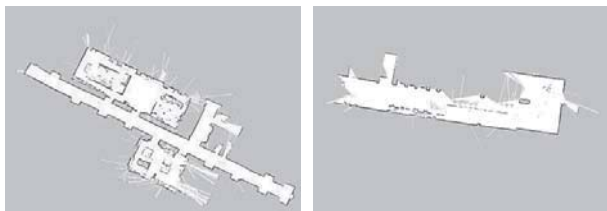
3. ソフトウェア構成

3-1 O.S.とミドルウェア

コンピュータの O.S.には，Ubuntu 14.04 を用いている。また，ロボット用ミドルウェア ROS（Robot Operating System）²⁾を採用し，自律移動に関するさまざまな処理を実現する。

3-2 地図作成

ROS にて利用可能な SLAM パッケージ gmapping³⁾を利用し，レーザレンジファインダの出力値から，占有格子地図（図 2）を作成し，自己位置推定・移動経路設計に利用する。



(a) 研究管理棟 4F (b) 研究管理棟 2F ロビー
図 2 事前作成した占有格子地図

3-3 オドメトリの算出

自己位置推定のため，ロボットのオドメトリを算出する。一般的には，デッドレコニングが利用されるが，

クローラ型ロボットの場合，接地面との滑りに起因する誤差が大きい。そこで，本研究では，IMU による 3 軸の姿勢推定およびステレオカメラによる並進量推定を行うことで，オドメトリの算出を行う。比較のため，デッドレコニングによる並進量推定についても実装を行う。

3-3-1 IMU の角速度情報を用いた姿勢情報の算出

時刻 $t = t_n, t_{n+1}$ における姿勢を表すクォータニオンをそれぞれ ${}^0q_n, {}^0q_{n+1}$ とする。IMU に搭載されているジャイロスコープから， $t = t_{n+1}$ における 3 軸角速度 $\omega_{n+1} = (\omega_{n+1,x}, \omega_{n+1,y}, \omega_{n+1,z})^T$ を取得し，サンプリング間隔 $\Delta t = t_{n+1} - t_n$ の微小回転 ${}^nq_{n+1}$ を算出する。各軸の微小回転を表すクォータニオンを，

$$\dot{q}_{n+1,x} = \begin{pmatrix} \cos \frac{\omega_{n+1,x} \Delta t}{2} \\ \sin \frac{\omega_{n+1,x} \Delta t}{2} \\ 0 \\ 0 \end{pmatrix}, \dot{q}_{n+1,y} = \begin{pmatrix} 0 \\ \cos \frac{\omega_{n+1,y} \Delta t}{2} \\ \sin \frac{\omega_{n+1,y} \Delta t}{2} \\ 0 \end{pmatrix}, \dot{q}_{n+1,z} = \begin{pmatrix} \cos \frac{\omega_{n+1,z} \Delta t}{2} \\ 0 \\ 0 \\ \sin \frac{\omega_{n+1,z} \Delta t}{2} \end{pmatrix}$$

として，クォータニオン積を用いて，微小回転 ${}^nq_{n+1}$ は下式により算出される。

$${}^nq_{n+1} = \dot{q}_{n+1,z} \dot{q}_{n+1,y} \dot{q}_{n+1,x} \quad (1)$$

また， $t = t_{n+1}$ における推定姿勢 ${}^0\hat{q}_{n+1}$ は，クォータニオン積を用いて，

$${}^0\hat{q}_{n+1} = {}^nq_{n+1} {}^0q_n \quad (2)$$

により求められる。

3-3-2 加速度情報を用いた姿勢の補正

ジャイロスコープはドリフト等により誤差が蓄積するため，加速度センサ値を用いて姿勢を修正する。加速度センサから得られた各軸の加速度を $\mathbf{a} = (a_x, a_y, a_z)^T$ ，静止時の重力ベクトルを $\mathbf{g} = (0, \theta \mathbf{g}^T$ と仮定し，ノルムが 1 となるように正規化したベクトルをそれぞれ $\bar{\mathbf{a}}$ ， $\bar{\mathbf{g}}$ とする。3-3-1 により求めた ${}^0\hat{q}_{n+1}$ を初期値とした最小化問題を解くことで， $t = t_{n+1}$ における姿勢 ${}^0q_{n+1}$ の最適化を行う。

$${}^0q_{n+1} = \arg \min \left\| {}^0q_{n+1} \begin{pmatrix} \bar{\mathbf{a}} \\ 1 \end{pmatrix} {}^0q_{n+1}^{-1} - \begin{pmatrix} \bar{\mathbf{g}} \\ 1 \end{pmatrix} \right\|^2 \quad (3)$$

ただし，クォータニオンを修正する際は，姿勢表現をクォータニオンから対数クォータニオンに変換する。これは，修正量を加算すると，姿勢を表す単位クォータニオンのノルムが 1 となる拘束条件を満たせないためである。クォータニオン $\mathbf{q} = (q_w, q_x, q_y, q_z)^T$ の対数

クォータニオンを $\mathbf{e} = (e_x, e_y, e_z)^T$ とすると、クォータニオンから対数クォータニオンへの変換は下記により行われる。

$$\mathbf{e} = \left(q_x \frac{\theta}{\sin(\theta)}, q_y \frac{\theta}{\sin(\theta)}, q_z \frac{\theta}{\sin(\theta)} \right)$$

$$\theta = \cos^{-1}(q_w) \quad (4)$$

一方で、逆変換は下記により行われる。

$$\mathbf{q} = \left(\cos(\theta), e_x \frac{\sin(\theta)}{\theta}, e_y \frac{\sin(\theta)}{\theta}, e_z \frac{\sin(\theta)}{\theta} \right)$$

$$\theta = \|\mathbf{e}\| \quad (5)$$

最適化は、対数クォータニオンの数値微分によるヤコビ行列の算出を行い、最急降下法を利用する。これにより、ロール・ピッチ方向のドリフト等の影響を除去した姿勢推定が可能となる。本手法は原理上、並進加速度が生じた際に精度が低下する。これを回避するため、本最適化の過程で最急降下法の重みパラメータを、加速度ベクトルのノルムが重力加速度 9.806 m/s^2 から離れるほど減少させている。

3-3-3 ステレオカメラを用いた並進量の算出

ステレオビジュアルオドメトリパッケージ Viso2⁴⁾を利用することで、 t_n, t_{n+1} 間のカメラ座標系に対する3次元移動量 $\Delta \mathbf{x}_{c,n+1} = (\Delta x_{c,n+1}, \Delta y_{c,n+1}, \Delta z_{c,n+1})^T$ を算出する。 $t = t_{n+1}$ における全体の座標系における3次元位置移動量 $\Delta \mathbf{x}_{w,n+1} = (\Delta x_{w,n+1}, \Delta y_{w,n+1}, \Delta z_{w,n+1})^T$ は、姿勢クォータニオン ${}^0\mathbf{q}_n$ とのクォータニオン積により、

$$\Delta \mathbf{x}_{w,n+1} = {}^0\mathbf{q}_{n+1} \begin{pmatrix} \Delta \mathbf{x}_{c,n+1} \\ 1 \end{pmatrix} {}^0\mathbf{q}_{n+1}^{-1} \quad (6)$$

により求められる。

3-3-4 デッドレコニングを用いた並進量の算出

ステレオカメラを利用する手法との比較のため、車軸に搭載されているロータリエンコーダから t_n, t_{n+1} 間の並進量を推定し、オドメトリを算出する。エンコーダから得られるキャリブレーション済みの左右クローラの速度を、それぞれ v_l, v_r 、ロボットの前後方向をロボットの座標系における x 軸方向とすると、3次元上での並進速度 \mathbf{v} は、

$$\mathbf{v} = \left(\frac{(v_l + v_r)}{2}, 0, 0 \right)^T \quad (7)$$

により表される。ロボットの姿勢を考慮した3次元の

移動量 $\Delta \mathbf{x}_{w,n+1}$ は、姿勢 ${}^0\mathbf{q}_{n+1}$ とクォータニオン積を利用し、下記により求めることができる。

$$\Delta \mathbf{x}_{w,n+1} = {}^0\mathbf{q}_{n+1} \begin{pmatrix} \mathbf{v}\Delta t \\ 1 \end{pmatrix} {}^0\mathbf{q}_{n+1}^{-1} \quad (8)$$

3-3-5 3自由度姿勢情報への変換

上記により、時刻 $t = t_{n+1}$ における6自由度のオドメトリ値 $\mathbf{u}_{n+1} = (\Delta \mathbf{x}_{w,n+1}, \boldsymbol{\omega}_{n+1} \Delta t)^T$ が得られるが、本研究では2次元の自己位置推定を想定しているため、6自由度のオドメトリ値を3自由度のオドメトリ値に変換する。 $\mathbf{u}'_{n+1} = (\Delta x_{w,n+1}, \Delta y_{w,n+1}, \boldsymbol{\omega}_{n+1,z} \Delta t)^T$ として、3-4で記すパーティクルフィルタに利用する。

3-4 パーティクルフィルタによる自己位置推定

オドメトリのみを用いた位置推定では、時刻 $t = t_{n+1}$ における3自由度位置姿勢は、各座標値を $\boldsymbol{\xi}'_{n+1} = (x_{w,n+1}, y_{w,n+1}, \theta_{n+1})^T$ として、 $\boldsymbol{\xi}'_{n+1} = \boldsymbol{\xi}'_n + \Delta \mathbf{u}'_{n+1}$ により更新されるが、誤差が蓄積されてしまう。

この対策として、事前作成した占有格子地図とレーザレンジファインダの観測値を比較し、自己位置の修正を行う。この際、冗長性を高めるため、パーティクルフィルタを利用する。

パーティクルフィルタでは、オドメトリ値をもとに、パーティクルと呼ばれる複数の位置姿勢候補が生成される。パーティクルごとに、事前作成した占有格子地図と、パーティクル位置姿勢でのレーザレンジファインダの出力値を比較し、尤度と呼ばれる類似度を求める。尤度の低いパーティクルを除去し、尤度の高いパーティクル周辺に再度パーティクルを撒くことで、パーティクルが更新され、尤度の高いパーティクルをもとに推定された自己位置が出力される。本システムでは、ROSにおけるパーティクルフィルタのパッケージ AMCL⁵⁾を導入し、パーティクル数5000個での自己位置推定を行った。

3-5 経路生成および経路に沿った走行制御

ROSにおける自律走行用パッケージ navigation_2d⁶⁾を利用し、自己位置推定結果および地図上のゴール情報からA-Star法を利用した経路計画を行う。地図情報に対して車体の大きさを考慮したコストマップを作成し、指定されたロボットの目標位置と自己位置を結ぶ経路をA-Star法により算出する。経路情報からロボットの目標速度・角速度指令を出力することで、ロボットの走行制御を行った。

4. 評価

4-1 試験環境

異なる路面環境での自己位置推定精度を比較するため、試験環境には、山梨県工業技術センター内 (a)研究管理棟 4F 廊下, (b)研究管理棟 4F 廊下および室内, (c)研究管理棟 2F ロビーにて実施する。(a)の路面はリノリウム材が用いられており, (b)では, リノリウム材およびタイルが用いられている。(c)においては, 全面カーペットが敷設されている。

オドメトリの取得方法は, (1)デッドレコニング+IMU, (2)ビジュアルオドメトリ+IMU, の 2 通りの構成にて行う。並進量のキャリブレーションは試験環境 (a)において実施した。また, オドメトリ手法の比較のため, 1 度の走行にて全センサ値をメモリ上に記録し, 走行後に記録されたセンサ情報を利用して(1), (2)それぞれの手法にて自己位置推定結果を出力する。これにより, (1), (2)による自己位置推定結果は, ロボットが全く同じ経路を走行した場合の自己位置推定結果として比較することができる。

4-2 自己位置推定精度の評価

事前作成した占有格子地図をもとに, 3-4 までの手法により自己位置推定を行い, 精度の評価を行った。結果として, 走行終了地点における自己位置推定誤差は, 全ての試験環境(a)(b)(c)において, 0.5 m 以内であった(表 1)。また, 走行時に並進方向の滑りがほぼ無かったため, デッドレコニングによる自己位置推定は良好な条件での結果であった。比較対象であるビジュアルオドメトリによる手法は, 良好な条件下で行われたデッドレコニングを利用する手法に対して大きく劣ることは無く, 走行終了時の自己位置推定精度についても同程度の精度であるため, 本研究にて提案したビジュアルオドメトリを用いる手法は, 自己位置推定に十分利用可能といえる。

試験環境(b)の位置推定誤差が(a)(c)よりも抑えられた原因としては, 走行終了地点の周辺においてロボットを取り囲むように壁が存在しており, パーティクルが十分に収束したためと考えられる。

4-3 2手法間の差異に関する考察

試験環境(b)において, デッドレコニング・ビジュアルオドメトリそれぞれを利用した自己位置推定の結果に差異が見受けられた(図 3)。差異の大きい地点の拡大映像を図 4 に表す。各地点での 2 手法間の位置推定結果の差は最大, 1) 46.3 cm, 2) 25.8 cm, 3) 21.9 cm となった。

図 3 地点 1)における位置推定結果の差異は, ビジュアルオドメトリ法の並進量推定誤差と, パーティクルフィルタの更新頻度に起因するものであった。ビジュアルオドメトリにより蓄積した誤差は, パーティクルフィルタにより逐次修正されるが, 地点 1)付近では十分に行われていなかったため, デッドレコニングの経路に対して差異が出現したと考えられる。改善方法としては, キャリブレーションを十分に行い, ビジュアルオドメトリの精度を高くする, もしくはパーティクルフィルタの更新頻度を上げることにより, 自己位置推定精度の改善が可能である。

図 3 地点 2), 3)における差異は, ステレオ視が困難な位置が存在したことが原因である。これはビジュアルオドメトリ特有の, texture-less と呼ばれるカメラのピントが合っていない場合や撮影した周辺物の模様が少ない場合に正しくオドメトリを取得できない状態が存在することに起因している。具体的に発生していた状況を検証してみると, 地点 2)では机, 椅子等がステレオカメラに接近したため, ピントの合っていない画像が読み込まれていた。また, 地点 3)では, 周辺物の模様が少ないフレームが含まれていた。改善方法としては, オドメトリ算出時に慣性項を導入する, または IMU から取得した加速度値を利用することで, texture-less により欠損するフレームのオドメトリを補完することが考えられる。

表 1 各試験環境における自己位置推定誤差

試験環境	床面材質	走行距離 m	デッドレコニング利用時	ビジュアルオドメトリ利用時
			走行終了時の位置推定誤差 m	走行終了時の位置推定誤差 m
(a)	リノリウム	37.12	0.392(1.055%)	0.269(0.724%)
(b)	リノリウム+タイル	74.54	0.098(0.131%)	0.201(0.269%)
(c)	カーペット	43.03	0.414(0.962%)	0.326(0.757%)

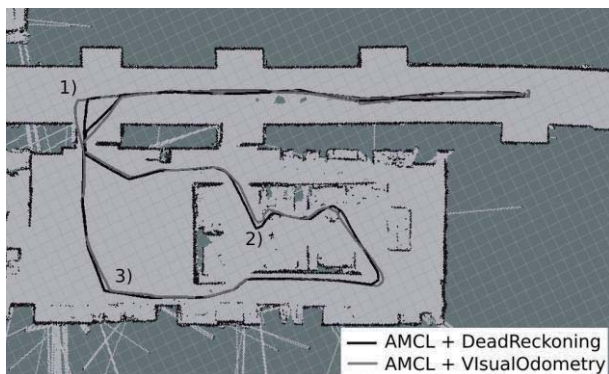


図3 試験環境(b)における自己位置推定結果

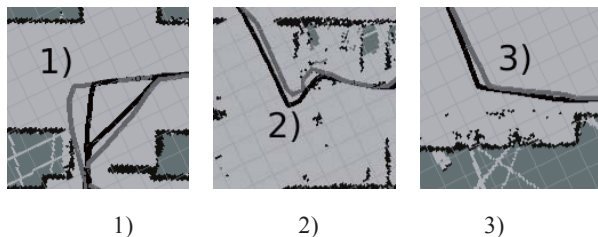


図4 2手法間の差が大きい地点の拡大画像

4-4 自律移動の評価

試験環境(a)において経路計画を行い、自律移動を実施した。走行距離約 37 m を自動走行し、所要時間は 2 分 48 秒であった。実験環境(b)(c)における自律移動は、(b)においては狭い空間での障害物回避に問題があり、(c)においては、経路計画の出力結果が、走行不可能な吹き抜け部分に及ぶため走行不能であった。障害物回避システムの開発およびユーザにより経路設計を修正可能なシステムの構築が必要である。

5. 結 言

本研究では、前年度の研究開発であるレーザレンジファインダを用いた地図作成技術を利用し、地図情報をもとに自己位置推定および自律移動を行うシステムの構築を行った。また、ステレオカメラと IMU によりオドメトリを測定する手法を開発した。その評価として、デッドレコニングを利用する手法との比較を行った。異なる床面材質（リノリウム、タイル、カーペット）をもつ環境において走行し、自己位置推定を実施した結果、本手法とデッドレコニングを利用する手法との誤差は同程度であり、床面状態の異なる状況に対しても安定した自己位置推定が可能であることを確認した。

今後の展開としては、自律移動のための、障害物の回避を行うシステムの開発、およびユーザにより移動経路を修正可能なシステムの開発を行う。また、本報にて作成したシステムを、第1報にて開発した汎用プラットフォーム Y-CRoSAM¹⁾に移植し、床面状態の異なる環境における自律走行について検証を行う。

参考文献

- 1) 布施嘉裕, 中込広幸, 宮本博永, 油井誠志, 中村卓, 米山陽, 神村明哉, 渡辺寛望, 丹沢勉, 小谷信司: 自律走行可能なロボットの制御手法の開発, 山梨県工業技術センター研究報告, No.30, pp.27-29 (2016)
- 2) ROS.org : ROS Wiki <<http://www.ros.org/wiki/>> (2017-3-1 参照)
- 3) OpenSLAM : GMapping <<http://openslam.org/gmapping.html>> (2017-3-1 参照)
- 4) Kitt, Bernd, Andreas Geiger, and Henning Lategahn : Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme, Intelligent Vehicles Symposium (IV), 2010 IEEE. IEEE, pp. 486-492 (2010)
- 5) ROS.org : AMCL Wiki <<http://wiki.ros.org/amcl>> (2017-3-1 参照)
- 6) ROS.org : Navigation2d Wiki <<http://wiki.ros.org/navigation>> (2017-3-1 参照)