MAV 支援型知能移動ロボットの自律走行制御と SLAM

Autonomous Control and SLAM of Intelligent Mobile Robot for MAV

○ NGUYEN DUY HINH (千葉大) 鄔熙乾 (千葉大)
学 岩倉大輔 (千葉大) 王偉 (千葉大) 正 野波健蔵 (千葉大)

Duy Hinh NGUYEN, Chiba University, Xiqian WU, Chiba University,

Daisuke IWAKURA, Chiba University, Wei WANG, Chiba University, Kenzo NONAMI, Chiba University

In this study group, we are developing a micro air vehicle (MAV), which is equipped with an automatic flight control system and a camera, for the purpose of performing rescue/relief tasks in natural disasters, security surveillance and observation tasks. MAV, however, has a limited performance due to its low payload or short duration of flight. On the other hand, unmanned ground vehicle (UGV) research and development is being pursued actively in Japan and overseas. Unlike MAV, UGV has a long continuous run time and a relatively bigger payload. It can carry many kind of equipment and play an important role in rescue. Therefore, we proposed a UGV that can assist the MAV, to make effective usage of our MAV in scenes. The UGV needs WayPoint and Mapping system. This paper shows Mapping using SLAM and WayPoint of UGV.

Key Words: Small autonomous flight machine, Rescue robot, SLAM, MAV

1. 緒言

Fig. 1 Overview of MAV

Fig. 2 Overview of UGV

本研究グループでは、災害現場における救助活動支援を目的とした Fig.1 に示すような MAV(Micro Air Vehicle) の研究を行ってきました。この MAV の特徴として、情報収集に向いている、積載量が小さい、駆動時間が短い事などがあげられる。

また、Fig.2 に示すような UGV(Unmanned Ground Vehicle) の研究も行なわれています。この UGV の特徴として、積載量が大き、様々な装置が搭載できる、駆動時間が長い事などがあげられる。

以上の事から、本研究グループでは MAV を災害現場で最大限に活用するためには UGV のアシストが必要だという見解にいたった。そこで、本研究では MAV 支援型 UGV の開発を行なった。

開発する UGV には次のような能力を必要とする。

- GPS を用いた WayPoint 走行
- SLAM を用いた屋内環境のマッピング

前者は屋外において MAV アシストを行なう際に必要となる。具体的には、MAV のバッテリーが減った際に MAV の位置を GPS で取得し UGV がその地点まで自律で走行しバッテリー充電を行なうものである。また、後者は GPS データを取得できない屋内環境で必要となる。具体的には、MAV が GPS データを用いることができない屋内環境で自己位置を把握するための地図を UGV が生成するものである。よって、本論文では UGV の WayPoint 走行と SLAM によるマッピングについて論じる。

2. UGV のシステム構成

2.1 制御対象

本研究グループが開発した UGV を Fig.2 に示す。仕様を Table 1 に示す。研究対象となっている UGV は MOBILE ROBOTS 社の Pioneer3-AT をプラットフォームとして開発したものである。Pioneer3-AT は屋外自律ロボットの研究用として販売されている。また、開発に関するソースコードが公開されていることで非常に研究を行いやすいため、本研究ではこの Pioneer3-AT を採用した。Pioneer3-AT の操作は搭載されているマイコンに指定されたコマンドをシリアル通信で送信することで行い、これによって様々な動作をさせることができる。UGV の車体前方にはレーザーレンジファインダーが、車体上部には通信ボードが搭載されている。通信ボードには、基地局のホストコンピュータと通信するための機器、方位磁気センサ、GPS の 3 つが装備されている。

Table 1 Specifications of UGV

Height×Width×Length	700 mm×490 mm×500 mm
Weight	15 kg
Continuous run time	4~ 6 h
Drive	4wheel drive
Wheel diameter	215 mm
Maximum speed	0.7 m/sec
External sensors	GPS
	direction sensor laser range finder

2.2 制御装置

制御装置は UGV に搭載されている PC(以下オンボード PC)、通信ボード、センサボード、レーザーレンジファインダーの 4 つで構成されている。搭載されている PC は市販されている Panasonic 社製のノート PC で、OS は Windows XP、CPU は Intel 社製の Core2Duo(2.26 GHz) である。開発環境には Microsoft

Fig. 3 System architecture

社製の VisualStudio2008 を用いている。

通信ボードのサイズは 250 mm(D) × 110 mm(W) × 80 mm(H), 重さは 400 g 程度である。通信装置は GPS と磁気方位センサが搭載されたマイコンボード (CPU は SH7211), PC とデータを通信するための RS232 変換ポート, 基地局の PC と通信するための通信機器ボード (XBee) で構成されている。

レーザーレンジファインダーは北陽電機社製の UTM-30LX というセンサで, レーザー ($\lambda=905$ nm) 光線により半円状のフィールドをスキャンし, 角度ごとの距離データを出力するセンサである。スキャン範囲は前方から ± 135 deg, ステップ角は 0.25 deg, 最大検出距離 30 m, 重量 210 g である。

2.3 制御システムの構成

制御システムの構成を Fig.3 に示す。UGV を操作する指令はすべてオンボード PC から UGV のマイコンに送信される。GPS データと方位データはセンサボードのマイコン上で処理され, RS232 ポートで PC と通信できるように電圧レベルを変換したのち, オンボード PC に送信される。また, プロポのデータは基地局を通して通信機器ボードに送信され, オンボード PC に取り込まれる。UGV にはマニュアルモードと自律モードが存在する。まず, UGV がマニュアルモードの際にはプロポが接続された基地局の PC から, プロポのデータを送信して UGV を操作する。自律モードの際には GPS データ, 方位データ, レーザーレンジファインダーのデータをオンボード PC で処理し, UGV のマイコンに指令を送信して自律制御を行なう。

3. WayPoint

3.1 WayPoint アルゴリズム

以下に, WayPoint アルゴリズムを示す。また, その概要を Fig.4 に示す。

step1 UGV の方位を WayPoint 座標に向ける。

step2 WayPoint 座標半径 1 m のエリアに進入するまで UGV を前進させる。

step3 進入後は, UGV を 1 m 直進させる。

Step1 では方位制御のみを行い UGV を旋回させる。Step2 では方位制御と位置制御を行う。Step3 では位置制御のみを行う。Step3 で方位制御を行わない理由は後述する。

Fig. 4 Overview of WayPoint motion

3.2 WayPoint の実験

Way Point の実験は,

- 方位制御のみの実験を運動場とアスファルトの 2 種類の地面で行なったもの (実験 1,2).
- WayPoint 座標付近で距離制御に移行したものと移行しなかったもの 2 種類 (実験 3,4).
- WayPoint 座標を連続的に送信したもの (実験 5).

の合計 5 種類行なった。

3.2-1 2 種類の地面で行なった方位制御のみの実験

Fig.5 は UGV を運動場で (実験 1), Fig.6 はアスファルトの上で (実験 2) 方位制御のみを行なった結果である。実験を行なった日の前日は雨が降っていたため運動場の土はぬかるんでいた。逆にアスファルトの上は綺麗に清掃されており乾燥していた。結果からわかるように運動場の土のほうが粘性があるためオーバーシュートを起こすことなく UGV の方位が目標値に収束している。アスファルト上での実験ではオーバーシュートが確認された。このときのオーバーシュートの量は 1.5 deg であった。今回の UGV の方位制御では UGV のモーターに負担をかけないように目標値の ± 3.0 deg の不感帯を設けている。よって, 今回のオーバーシュートは方位制御にはあまり影響しなかったと言える。

3.2-2 WayPoint 座標を連続的に入力した実験の方位制御

Fig.7 は運動場にて WayPoint 走行を行なった結果 (実験 5) である。今度は WayPoint 走行を行なうため運動場の中でも乾燥している場所だけを選択して実験を行なった。WayPoint 座標は合計で 4 つ入力した。毎回オーバーシュートしているが目標値に追従していることがわかる。アスファルトの上でも同様の実験を行なったが結果はほぼ同じであった。

3.2-3 制御切り替えを行なわなかった実験の方位制御

Fig.8 はアスファルトの上で WayPoint 走行を行なったもの (実験 4) であるが, WayPoint 座標から半径 1 m の範囲に入っても速度制御から距離制御に切り替えなかった場合の結果である。図からわかるように 30 sec から目標値が急に発散していることがわかる。この現象が起こったのは, GPS データに誤差があるため目標座標付近では目標地までの角度が変化しやすいためであると考えられる。

3.2-4 UGV の位置制御

Fig.9 と Fig.10 に UGV の位置制御の実験結果を示す。Fig.9 は WayPoint 座標を 1 度だけ入力した実験 3 の結果である。最初に方位を定めてから前進するため、理想的には軌跡がほぼ直線になるはずだが Fig.9 を見ると明らかに軌跡はゆるい円弧を描いている。これは方位センサの収束速度が原因である。UGV が停止または直進している際は正しい値を示していたが、UGV が旋回した時に、センサの値が収束せず誤差を含んだまま方位制御を行なってしまった。実験中に基地局の PC には WayPoint 座標と UGV の向きが表示されるが、こちらは一致していた。しかし、PC に表示されている UGV の方位と現実の UGV の方位は一致しておらず最大で 30 deg 近くの誤差が生じていた。

Fig.10 は実験 4 の UGV の軌跡である。先ほど述べたように制御の切り替えを行なわなかったため、GPS の誤差により WayPoint に到達したことが認識できずに通過してしまうという結果となった。

3.2-5 UGV の WayPoint 走行

以下の Fig.12 に UGV の WayPoint 走行の実験結果を示す。Fig.12 は実験 5 の UGV の軌跡である。軌跡は GPS データからメートル単位の位置を算出して表示した。はじめに、現在地より北 (Y 軸正の方角) に WayPoint 座標を指定した。以後、西、南、東の合計 4 つの WayPoint 座標を入力した。Fig.11 に実際に基地局の PC から入力した 4 つの Way Point 座標を示す。Fig.12 から見てわかるように、軌跡は曲線を描いている。これは先ほど述べたように方位センサの誤差が原因である。方位センサは時間がたつにつれ正しい値に収束するが、WayPoint 座標を入力されたばかりの時は方位が収束しておらず間違った向きのまま前進を開始する。しかし WayPoint 座標付近ではおおよそ正しい値に収束するため目標地に向かうことができた。

4. SLAM

4.1 SLAM アルゴリズム

SLAM とは、(Simultaneous Localization And Mapping) の略である。これは、ロボットが自己位置を正確に推定しながら同時に、地図を生成するという意味である。

本来、ロボットが自己位置を正確に推定するためにはロボットが置かれた環境の地図を必要とする。また、ロボットが地図

Fig. 5 Angle of UGV (Ex1) Fig. 6 Angle of UGV (Ex2)

Fig. 7 Angle of UGV (Ex5) Fig. 8 Angle of UGV (Ex4)

Fig. 9 Position of UGV (Ex3) Fig. 10 Position of UGV (Ex4)

Fig. 11 Overview of Ex5 Fig. 12 Position of UGV (Ex5)

を生成する際にロボットは正確な自己位置を必要とする。このような矛盾を解決するのが SLAM アルゴリズムである。

一般にマッピングを行なう際には、UGV のタイヤから得られる軌跡データを基に地図を生成するが、得られるデータはタイヤのスリップなどの要因で誤差を多く含んでしまう。しかし、SLAM はマッピングしたデータを用いて軌跡データを修正するため、UGV は正確な自己位置を得ることができる。よって正確な地図生成を行なうことができる。

本研究では、Duke 大学の Eliazar らによって研究された DP-SLAM[1] を UGV に実装し、マッピングの実験を行なった。

4.2 UGV の自律走行アルゴリズム

以上の実験は全て UGV を自律走行させマッピングを行なった。UGV は前方を 0 deg として ± 15 deg と -45 deg ~ -15 deg, 15 deg ~ 45 deg の 3 つの範囲における距離データを監視し、一定の値を下回った距離データがそれぞれに存在した場合は回避行動をとる。各範囲におけるステップ角は 1 deg である。まず、前方の距離データに反応があった場合は左右の距離データを全て比較し、最大値が含まれている方に UGV を向ける。また、左右の距離データに反応があった場合はそれぞれ反対方向に UGV の向きを向ける。このアルゴリズムによって UGV は常に一番スペースが開いている空間に向かう。次に、行き止まりに UGV が入ってしまった場合は、全ての距離データが一定の値を下回るため、それを判断基準とし UGV を 180 deg 回転させる。このアルゴリズムにより UGV は行き止まりに迷い込んでしまっても自律で脱出することができマッピングを継続できる。

4.3 SLAM の実験

SLAM の実験は、屋内と屋外の両方で行った。

4.3-1 屋内でマッピングを行なった実験

実験は千葉大学自然科学研究棟 2 号棟の 10 階で行なった。

まず、10 階の間取り図を Fig.13 に示す。間取り図の中で水色で表示されているのはガラス製の壁である。Fig.14 は 10 階で UGV を自律で走行させマッピングを行なって生成されたものである。これは占有格子地図である。DP-SLAM で生成される占有格子地図では、白い格子で表現された部分は全くスキヤ

ンされなかった場所を示し、薄い灰色はフリースペースを示し、濃い灰色は障害物を示す。

各廊下の突き当りではロボットは大きく回転しながら走行したが、オドメトリの誤差も正確に修正され、廊下が直角にマッピングされた。階段と通路の間にはガラスの壁があるが、階段を正確にマッピングできたということは、ガラスの壁がレーザーを透過していることが分かる。また、同様な理由で実際は吹き抜けである地図の中心部が間違ってマッピングされてしまっている。しかし、地図上ではガラスの壁を確認することができるため、一部のレーザーはガラスに反射したことが分かる。

4.3-2 屋外でマッピングを行なった実験

実験は千葉大内のさまざまな場所で行なった。実験場所を選ぶ際はなるべく木や自転車、植木などランドマークになりそうなものが多い場所を選んだ。しかし、地図生成はうまくいかなかった。その一例を Fig.17 に示す。DP-SLAM は格子ベースの SLAM なので木や植木はランドマークとして機能しなかった。そのため、UGV は何メートルも移動したにもかかわらず、生成された地図はある一点の UGV がとどまっているかのような地図を生成することもあった。これは、木や植木が目立ったランドマークになり得なかったのが原因だと考えられる。今回の SLAM の格子地図は、辺が 3 cm の格子で形成されている。茂みや自転車はレーザレンジでスキャンする際に連続的に格子を占有することができない。そのため、ランドマークが多くのパラバラの点として存在してしまうため SLAM のアルゴリズムがランドマークを認識できなかったと考えられる。Fig.17 の地図は Fig.15 と Fig.16 をマッピングして生成した地図である。次に Fig.18 をマッピングしなければならないのだが、Fig.17 と Fig.18 には共通する特徴を捉える事ができなかった。Fig.18 の中でも特徴として捕らえられるのは図の下側の部分だが、これは茂みがレーザーが散乱しているため障害物が多く、点として表現されてしまい、目立ったランドマークとして捉える事ができなかった。よって、この時点でマッピング成立しなくなり地図を生成することができなくなってしまった。

Fig. 13 Map of 10th floor

Fig. 14 Map of 10th floor generated by SLAM

Fig. 15 Map generated by SLAM (a)

Fig. 16 Map generated by SLAM (b)

Fig. 17 Map of outdoor experimental field generated by SLAM

Fig. 18 Map generated by SLAM (c)

5. 結言

本研究では MAV 支援型 UGV の研究の足がかりとして、UGV に WayPoint 走行アルゴリズムと SLAM を実装した。以下に結果をまとめる。

- UGV を自律で WayPoint 走行させることに成功した。
- 実装した SLAM で室内のマッピングに成功した。

今後は、WayPoint 走行アルゴリズムを拡張した MAV 追従システムと、生成された地図を用いた MAV 飛行経路計画支援システムを構築する。

文 献

- [1] Eliazar, A., DP-SLAM, PdD thesis, Duke University, 2005.
- [2] 樋山義久, 村尾建治, 山本佳男, 車輪型移動ロボットの柔軟な障害物回避行動, ロボティクス・メカトロニクス講演会 '03 講演論文集, 1A1.1FC4(1)-1A1.1FC4(2), 2003.
- [3] 川崎洋平, 久保信明, 自律型 GPS ロボットカーの制御, 卒業論文, 東京海洋大学, 2008.
- [4] 樽床祐樹, 渡辺嘉二郎, Simultaneous Localization And Map building による自律ナビゲーションの開発, 修士論文, 法政大学院, 2007.
- [5] 野波健蔵, 西村秀和, 平田光男, MATLAB による制御系設計, 東京電機大学出版局, 1998.