

再利用性を考慮した移動ロボット用ソフトウェアモジュールの開発 -目的地までの指定経路走行に適した障害物回避手法-

○石田卓也 山田大地 岡村公望 (富士ソフト(株))

大矢晃久 油田信一 (筑波大学)

Development of Reusable Software Modules for Mobile Robots.

-Obstacle Avoidance Method Suitable for Designated Path Traveling to Destination-

*Takuya ISHIDA, Taichi YAMADA, Kimimochi OKAMURA (FUJISOFT Inc.)

Akihisa OHYA, Shin'ichi YUTA (Univ. of Tsukuba)

Abstract — We are developing reusable software modules for mobile robots in METI project “Intelligent RT Software Project”. This paper describes an obstacle avoidance method that suitable for designated path traveling to destination. We implemented this method, and evaluate this software system.

Key Words: Mobile robot, Reusable software modules, Obstacle avoidance

1. はじめに

我々は経済産業省(平成20年度からはNEDO)が実施する「次世代ロボット知能化技術開発プロジェクト」においてサービス産業分野での移動ロボットの汎用基本ソフトウェアモジュールの開発に取り組んでいる[1][2]。本モジュール開発では人や障害物が存在する実環境中で目的地に到達するとともに、ロボットが人や障害物に衝突することなく移動するための汎用的な機能を提供するモジュールの開発を目的とし、自己位置の管理や、走行の制御などの機能を提供するモジュールを開発している。

ここで、開発するソフトウェアは再利用性を高く保つため、処理の流動性により3つに分割して開発を行った。流動性が低いと考えられる部分をモジュールとしてライブラリ化した。また、OSに依存する部分をプラットフォームとして切り出した。そしてモジュールを内包し、プラットフォーム上でメッセージをやり取りして実際のロボットの動作を実現する、動作モデルにより処理内容が流動する部分をコンポーネントとして切り出した。コンポーネントはプラットフォーム上で、プロセスかスレッドとして独立したコンテキストとして動作する。ここで、モジュールの粒度は小さくしており、ロボットの動作モデルが変更になっても、コンポーネントからのモジュールの呼び出し方法を変更するだけでモデル変更への対応が可能となる。また、「次世代ロボット知能化技術開発プロジェクト」においてRTミドルウェア上で動作するRTCとしての提供も視野に入れており、上記のようなソフトウェア構成となっているため、RTC化もモジュールをそのまま使用可能となり移植が容易となる。

本稿では、目的地までの指定経路走行に適した障害物回避手法の検討、障害物回避モジュールの設計、開発したモジュールを利用した車輪型ロボットによる障害物回避実験結果について報告する。

2. 走行方式

2.1. 目的地までの指定経路走行

目的地を指定することにより、事前に与えられる経路地図上で最短となる経路を算出する。算出された指定経路を実現するため、ロボットの自己位置を逐次同定し、ロボットの速度・角速度を算出し走行を行う。

目的地までの指定経路走行のために必要な機能を下記に示す。本プロジェクトでは他に自己位置の同定のための機能や地図情報の管理機能なども必要であり開発を進めているが、今回は走行関連の機能のみを記載する。

- ・ **経路計画機能**

事前に与えられる経路地図上で、指定される目的地へ現在の自己位置から最短となる経路を算出する。算出した経路をロボットが走行可能な軌跡の羅列である動作計画に変換する。作成した動作計画を動作計画管理機能へ指示する。

- ・ **動作計画管理機能**

指定される動作計画を実現するように、ロボットの自己位置を監視し、タイミングを計り走行制御機能へ軌跡を指示する。

- ・ **走行制御機能**

指定される軌跡に追従するような速度・角速度を、逐次ロボットの自己位置から算出する。軌跡は、直線追従・円弧追従・旋回に対応する。

2.2. 障害物回避

Fig.1に示す通り、走行指示の階層は目的地、経路、軌跡、速度・角速度を指定する論理階層となっている。障害物回避の論理階層は、走行指示と同様の論理階層を取り、衝突回避、局所的回避、大域的回避の3つの回避方法をとることとした。

ここで障害物回避は、指定経路を可能な限り遵守するという制約の中で回避を実施する方針とした。

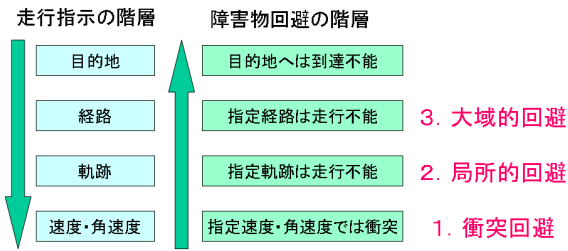


Fig. 1 障害物回避の論理構造

3階層の障害物回避イメージを Fig. 2 に示す。

・衝突回避

予定走行軌跡上に障害物を検出した場合に、減速し衝突を回避する。障害物までの距離が一定の距離以下になった場合に、速度は0となり停止する。障害物が解消した場合は再度走行を始める。

・局所的回避

本来の動作計画から逸れて障害物を回避する。このとき本来の動作計画から逸れる幅は、経路地図で与えられる道幅内に収まるようにする。障害物回避後には本来の動作計画に出来るだけはやく戻るようにする。

・大域的回避

経路が障害物により完全に塞がれたと場合、その経路を使用せずに目的地に到達可能な他の経路を再計画する。使用不能とした経路は記憶しておき、再度大域的回避を実施する際にも、1度使用不能とした経路は使用しないように経路を計画する。

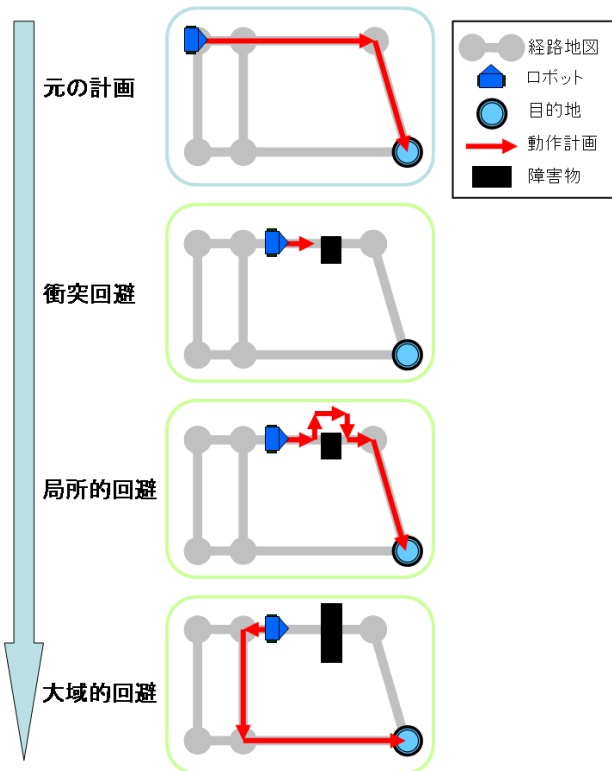


Fig. 2 障害物回避の3階層のイメージ

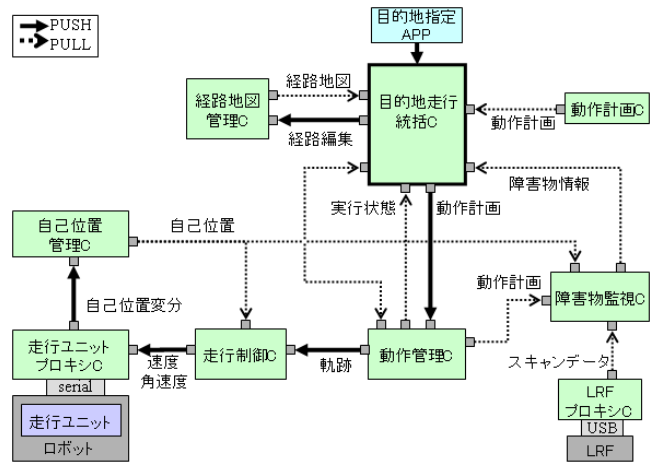


Fig. 3 障害物回避を含む目的地までの指定経路走行のコンポーネント構成

3. ソフトウェア実装

3.1. コンポーネント構成

Fig. 3 に障害物回避を含む目的地までの指定経路走行のコンポーネント (以下Cと略す)の構成を示す。障害物回避の機能は下記のように実現される。ここで、3つの障害物回避の実行は目的地走行統括Cが判断するが、その処理内容は3.2状態遷移にて示す。

・衝突回避の実現

障害物監視Cは動作計画とスキャンデータから、予定走行軌跡上の障害物検出処理を行う。障害物を検出した場合は、衝突予測距離を算出し障害物情報として目的地走行統括Cへ送信される。目的地走行統括Cは、衝突予測距離から安全速度を算出し、動作計画の速度を安全速度で書き換えて動作管理Cへ送信する。動作管理Cは指定された速度で走行を行うため減速・停止となり衝突回避が実現される。

・局所的回避の実現

障害物監視Cはスキャンデータから周辺障害物のグリッドマップを作成し、障害物情報として目的地走行統括Cへ送信する。目的地走行統括Cは局所的回避が必要と判断した場合、障害物グリッドマップと動作計画の道幅情報から走行可能領域を示すグリッドマップを作成する。そして走行可能グリッドマップから、隣接する走行可能グリッドマップ間を接続する経路を接続し、局所的な経路地図を作成する (Fig. 4)。このときの局所的な経路地図は、障害物が存在しなく道幅以内の部分に網目状に経路が引かれたものとなっている。さらに障害物の向こう側となる局所的な目的地を算出し、局所的な経路地図と局所的な目的地を動作計画Cへ送信する。動作計画Cは受信した経路地図と目的地から、最短経路を算出し動作計画に変換し目的地走行統括Cへ返信する。この局所的な経路地図上で最短経路を探索することにより局所的に障害物を回避する経路を算出できる (Fig. 5)。目的地走行統括Cは返信された動作計画と本来の動作計画を接続して局所的回避が可能な動作計画を作成し (Fig. 6)、動作管理Cへ送信する。動作

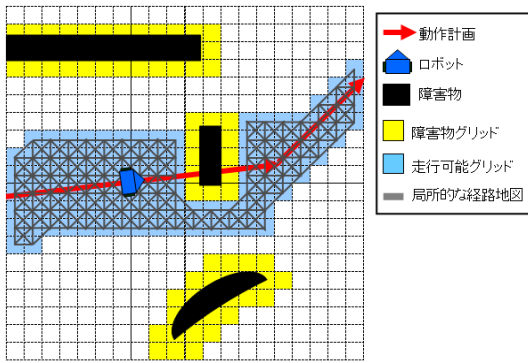


Fig. 4 グリッドマップと局所的な経路地図

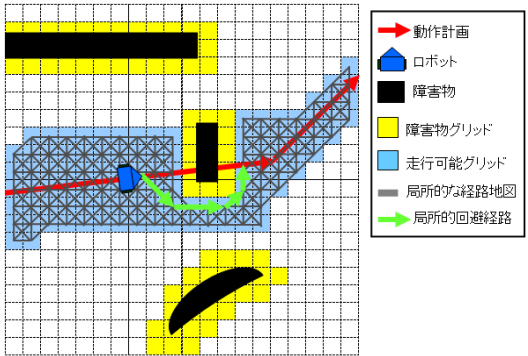


Fig. 5 局所的な回避経路

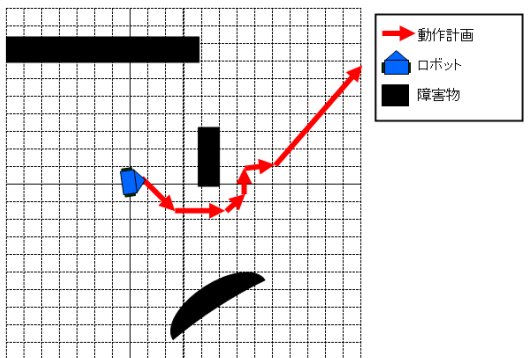


Fig. 6 局所的回避計画後の動作計画

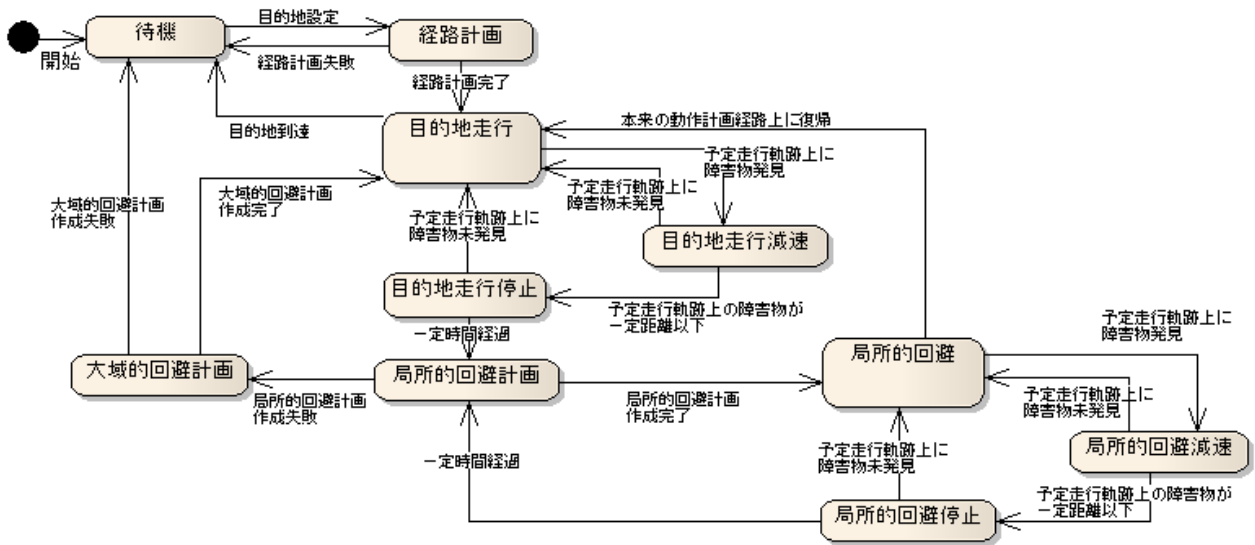


Fig. 7 目的地走行統括Cの状態遷移図

管理Cは指定された動作計画を実行する。

・大域的回避の実現

目的地走行統括Cは大域的回避が必要と判断した場合、衝突予測距離から障害物が存在する経路を算出し、その経路を走行不能であるとの経路編集指示を経路地図管理Cへ送信する。そして障害物が存在する経路を走行不能とした経路地図と本来の目的地を動作計画Cへ送信する。動作計画Cは受信した経路地図と目的地から、最短経路を算出し動作計画に変換し目的地走行統括Cへ返信する。ここで、障害物が存在する経路は使用しないため、大域的に障害物を回避する経路を算出できる。目的地走行統括Cは返信された動作計画を動作管理Cへ送信する。動作管理Cは指定された動作計画を実行する。

3.2. 状態遷移

障害物回避の実行は、目的地走行統括Cの内部の状態遷移によりすべて管理される。Fig. 7に目的地走行統括Cの状態遷移図を示す。

目的地が指定され指定経路の計画が完了し通常の走行を行っている状態は、目的地走行状態となる。目的地走行状態では予定走行軌跡上に障害物が発見されると減速を行う目的地走行減速状態に遷移し、さらに障害物への距離が短くなると、停止する目的地走行停止状態に遷移する。

目的地走行停止状態では、任意の時間を経過した場合に局所的回避計画を行う局所的回避計画状態に遷移する。局所的回避計画が完了した場合は、局所的回避を実行する局所的回避状態に遷移する。局所的回避状態では目的地走行状態と同様に、予定走行軌跡上の障害物検出処理を行い、障害物を発見した場合には減速・停止処理を行う。

局所的回避計画状態にて局所的回避計画の作成に失敗した場合には、大域的回避計画を行う大域的回避計画状態に遷移する。大域的回避計画が完了した場合は、通常走行と同様となるため目的地走行状態に遷移する。大域的回避計画が失敗した場合は目的地に到達不能なので待機状態に遷移する。

4. 障害物回避実験

開発したソフトウェアを使用し、目的地までの指定経路走行の実験を行った。経路上には障害物を配置し、衝突回避、局所的回避、大域的回避が発生するようにした。

4.1. ハードウェア構成

実験に使用したロボットの仕様を Table1 に、ロボットの外観を Fig. 8 に示す。LRF は北陽電機製 TOP-URG (UTM-30LX) を搭載している。コントローラとして、モータ・エンコーダと接続される SH2 ボードと、全体の制御を行うノート PC を搭載している。

Table1 ロボットの仕様

外形寸法	W=306 D=229 H=295 [mm] (筒体のみ)
総重量	約 3 [kg] (筒体のみ)
最大速度	約 50 [cm/sec]



Fig.8 ロボットの外観

4.2. 実験内容

経路地図は Fig. 9 に示すものを使用した。実線が経路、点線が各経路の道幅を示している。

実験では、Fig. 9 中の Start が示す地点をロボットの初期位置、Goal が示す地点を目的地とした。このような情報をロボットに与え、目的地までの指定経路走行を行わせた。そして走行中に予定走行経路上に障害物を置いて障害物回避の実験を行った。

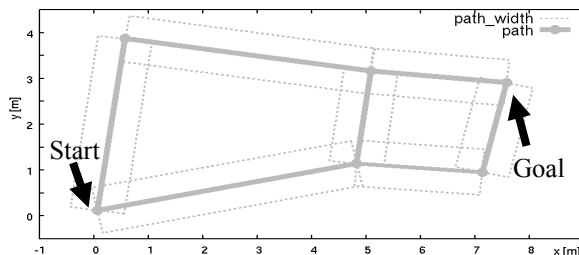


Fig.9 経路地図

4.3. 実験結果

Fig. 10 にロボットが算出した最短経路である当初の動作計画を示す。Fig. 11 にロボットが走行した結果の自己位置を示す。×が自己位置を示しており、※が LRF にて測定された障害物を示している。

Fig. 11 の A で示す地点では、予定走行軌跡上に障害物を認識して減速・停止を行い、その後、局所的障害物回避を実施している。回避のために指定した

経路から逸れているが、指定した道幅内で走行しており障害物回避後は本来の計画に戻っている。B で示す地点では、道幅一杯に障害物が存在しており、局所的障害物回避は不可能となり大域的障害物回避を実施し、別の経路を計画し一度折り返して別の経路から目的地に到達している。

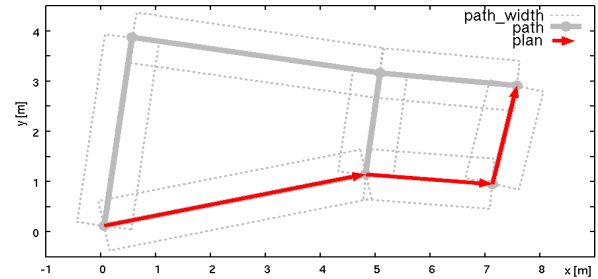


Fig.10 当初の動作計画

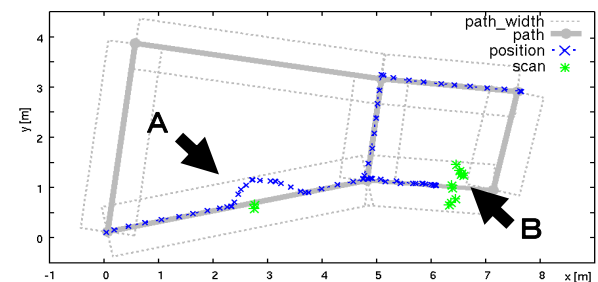


Fig.11 走行結果

5. まとめ

本稿では、目的地までの指定経路走行に適した障害物回避手法、ソフトウェア構成の紹介と障害物回避実験の結果を示した。開発したソフトウェアは再利用性が確保しており、プラットフォームの変更、走行仕様の変更に柔軟な構成となっている。

実験では、開発したソフトウェアモジュールを使用して、障害物回避を含む目的地への指定経路走行が可能なロボット制御システムを構築した。実際の走行結果では、予定経路上の障害物を検出して停止する衝突回避、指定した道幅内での局所的回避、指定した経路地図上での大域的回避の実現が確認できた。

今後は、移動障害物への対応などの機能追加や、実環境での走行試験などによるソフトウェアモジュールのロバスト性の向上を目指す。

参考文献

- [1] 油田信一, 大矢晃久, 吉田光治, 岡村公望: “移動ロボット用基本知能のモジュール化 一次世代ロボット知能化技術開発プロジェクトにおける取り組み”, 第 26 回ロボット学会学術講演会, 2008.
- [2] 岡村公望, 福永和海, 大矢晃久, 油田信一: “移動ロボット用基本知能のモジュール化～開発モジュールの紹介と RTC 化への取り組み～”, 第 27 回ロボット学会学術講演会, 2009.