

修士論文

環境を含む自律移動ロボットの
マルチエージェントシステムの開発

指導教員 加藤 典彦 准教授

平成23年度

三重大学大学院

工学研究科 博士前期課程 機械工学専攻

410M132 原田 祥太

平成 23 年度 修士論文

環境を含む自律移動ロボットの マルチエージェントシステムの開発

専攻 三重大学大学院 工学研究科 機械工学専攻
研究室 メカトロニクス研究室

平成 22 年度入学 410M132

氏名 原 田 祥 太

目次

第1章	まえがき	1
第2章	ロボットのシステム構成	3
2.1	ロボットのハードウェア構成	3
2.2	ロボットのソフトウェア構成	4
第3章	自律移動プログラム	8
3.1	エンコーダエージェント	8
3.2	LRF エージェント	9
3.2.1	パーティクルフィルタを用いた地図マッチングによる自己位置推定	9
3.2.2	前進時における前方障害物の監視	16
3.3	GPS エージェント	17
3.4	統合エージェント	19
3.5	制御エージェント	21
3.6	画面表示エージェント	23
第4章	環境調査	25
4.1	環境の機構面に関する調査	25
4.2	環境の認識面に関する調査	27
4.3	環境調査に関するまとめ	32
第5章	むすび	37

図目次

1.1 各アプローチの比較	2
2.1 実験に使用するロボット: MURV-2 三翠	4
2.2 MURV-2 三翠 システム構成図	6
2.3 マルチエージェントシステム構成図	7
3.1 100m の直線走行時のデッドレコニングのみによる自己位置推定結果	10
3.2 100m の直線走行時の角度変化	11
3.3 二値格子地図の例	12
3.4 LRF から各グリッドまでの角度及び距離	13
3.5 各グリッドまでの角度に近い角度インデックスを持つ計測点	14
3.6 グローバルマップ生成におけるローカルマップ	15
3.7 グローバルマップ生成のフローチャート	16
3.8 パーティクルの初期化	17
3.9 尤度計算	18
3.10 パーティクルフィルタを用いた地図マッチングの流れ	19
3.11 危険領域の回避	20
3.12 緊急停止	21
3.13 GPS の精度比較 1	22
3.14 ウェイポイント間で引いた直線との角度及び距離によるステアリング制御	23
3.15 ロボット操作用インタフェース	24
3.16 パーティクル表示ウィンドウ	24
4.1 機構的に移動可能な領域の判定方法	26
4.2 調査結果 (機構的に移動可能な領域)	26

4.3	実験環境	27
4.4	実験に使用したグローバルマップ	29
4.5	地図マッチングがうまく機能しない環境 (各ローカルマップ及びグローバルマップ) . .	30
4.6	自律走行試験結果	33
4.7	認識面での環境調査を行う環境	34
4.8	各建物の移動目標地点	35
4.9	調査結果 (環境の認識面における調査)	36

表 目 次

2.1	ML200 の諸元	5
2.2	使用するセンサなどの機器	6
3.1	NMEA-0183 フォーマットの例	18
3.2	GPS の精度比較 2	19
4.1	パーティクルフィルタの各パラメータ	28
4.2	各経路走行中に出現する問題のある環境	31
4.3	各環境の出現確率	31
4.4	環境調査に関するまとめ	32

第 1 章

まえがき

現在, ロボット技術の発展とともに人々が生活している実空間上でロボットが自律的に行動する技術の追求が盛んに行われている. しかし, 実際に実空間上でロボットを自律的に行動させるにはハードウェア, ソフトウェアの両面から解決しなければならない技術的課題が非常に多く存在し, 実用化されている例は多くない. そこで, それらの技術の確立のためにも実機を用いて実環境で実証実験を行っていくことが重要であると考えられる. そのような実証実験の場として, 海外では DARPA Urban Challenge, Grand Challenge[1] などがあり, 国内では, つくばチャレンジといったものが行われている. そのどちらにおいても, 屋外の実環境を場所として定義し, その中で達成すべき課題を定め, ロボットの機能のみでそれを実現するロボットを研究開発することで自律移動の技術の進展を図っている.

一方, 工場内などの整備された環境では, 自律移動ロボットはすでに実用化されている [2][3]. これらの研究では, 磁気タグや反射板などのセンサやマーカを周囲環境に設置し, 自律移動ロボットあるいは環境を統括するシステムによりそれらの情報を処理することで, ロボットの走行を確実にすることを目標としている. これらのように, 周囲環境に手を加えてロボットの走行を支援するものを「環境情報化」といい, 近年研究されている [4][5]. しかし, これらの研究では, 環境からは十分すぎるほどの情報が与えられており, ロボットの確実な自律走行のための最低限の環境操作に関しては議論されておらず, 環境の導入負荷が大きい.

そこで上記のロボットの機能を向上させることで自律走行を可能にするアプローチ (Fig. 1.1 左) や走行環境を変化させることで環境からロボットを支援するアプローチ (Fig. 1.1 中央) に対して, 本研究では想定される環境の出現確率に焦点を当て, 環境の有効度を考慮し, 自律移動ロボットの機能の改良のみでなく周辺環境も適宜変更することが重要であるとする (Fig. 1.1 右). この考えに基づ

いてシステムを構築することにより, ロボットの機能の複雑化を防ぐとともに, 導入負荷の小さな環境操作をすることで, 確実な走行の実現を目指す. また, 将来的には自律移動ロボットの実用化のために有効な環境操作も見つけ出す.



Fig. 1.1: 各アプローチの比較

そのためにまず実際につくばチャレンジのトライアル走行が達成できるレベルを一定の基準として自律移動ロボットのシステムの開発をしていき, その後に有効な環境操作提案のための初期段階としてそのシステムを利用して実際に環境を走行させ, 環境を調査していく.

以降, 第 2 章では, 本研究で用いるロボットシステムについて述べる. 第 3 章では, 作成した自律移動システムの概要について述べる. 第 4 章では, 作成した自律移動システムを用いてモデルとして選んだ環境を実験結果を基に調査した結果を述べる. 第 5 章では, まとめとして達成した課題と今後の課題を述べる.

第 2 章

ロボットのシステム構成

2.1 ロボットのハードウェア構成

環境の出現確率の調査及びシステム変更の評価実験をするため、自律移動ロボットを作成する。ロボットを作成するにあたり、簡単な走行機構となるようにする。理由は、簡単な走行機構のロボットでも確実な走行が可能となる有効な環境操作を見つけ出すためである。本研究では、三重大学工学部機械工学科システム設計研究室が制作した Fig. 2.1 に示すロボット MURV-2 三翠を使用する。これは、ホンダ技研製高齢者用電動四輪カート「ML200」をベースとして、自律移動ロボットとして改造を施したものである。ML200 は他社の電動カートに比べ重量は重い、アクセル用モータのノイズ対策がしっかり行われておりバッテリーの容量が大きい、またタイヤ全体がゴムであることからデッドレコニングの際に両車輪のパルス数のズレが少ないなどの優位点がある。ロボットの寸法等の諸元を Table 2.1 に、使用するセンサなどの構成を Table 2.2 に示す。また、ロボットの搭載機器のシステム構成図を Fig. 2.2 に示す。

ML200 への動作指令は我々が開発してきた小型自律移動ロボット「MieC」が制御用 PC の代替として行う。MieC に関して、簡単に紹介する。MieC は、CPU ボードを持ち、USB メモリに OS として Debian GNU/Linux 5.0 が入っている Linux コンピュータである。カーネルバージョンは 2.6.30 を使用し、本システムにおいて制御面での必要性がなかったことから OS のリアルタイム化は行っていない。USB 接続の無線 LAN キーを通して、外部とデータを通信する。また、CPU ボードから PCI 接続された FPGA が DC モータやエンコーダと信号をやり取りし、動作を実現している。MieC は、USB 接続によりセンサを取り付けることが可能であり、本研究に用いる MieC には、外界センサとして多くの移動ロボットが標準的に持つ、LRF(Laser Range Finder)、GPS 等を取り付けた。



Fig. 2.1: 実験に使用するロボット: MURV-2 三翠

2.2 ロボットのソフトウェア構成

自律移動ロボットのシステムを開発する際、複数のセンサからの情報を取得し、それぞれ処理を施し、その情報を基にロボットを移動させるため、様々な処理を加えていく毎にシステムの複雑化が懸念される。またつくばチャレンジなどでは、機器トラブルになどよりセンサの構成をその場で変更する必要性に迫られる可能性がある。そのような際の対応のシステムのモジュール性を高めておくことは非常に重要といえる。そこで、本研究では、システムのモジュール性を向上させるために、各センサ毎にプログラムを分割し、そのそれぞれをエージェントとするマルチエージェントシステムとして設計する。今回は、Fig. 2.3 に示すように 6 つのエージェント (エンコーダエージェント, GPS エージェント, LRF エージェント, 統合エージェント, 制御エージェント, 画面表示エージェント) によりシステムを構築した。それぞれのエージェントにおける詳細な説明を次章にて述べる。

Table 2.1: ML200 の諸元

名称		モンパル ML200	
型式		UDAJ	
諸元	寸法	全長 × 全幅 × 全高	1190x595x1045mm
	シート寸法	幅 × 奥行 × 背もたれ高	550x550x375mm
	本体重量 (バッテリー含む)		115kg
	タイヤサイズ		80/80-6
	バッテリー (5 時間率容量)		12-35Ah × 2 個
	モータ (30 分定格出力)		DC ブラシレスモータ (24V 430W) × 1 個
	操舵方式		ループハンドルによる前輪操舵
	駆動方式		後 2 輪直接駆動方式 (デフ付)
	制動方式		回生制動・電磁ブレーキ・手動ブレーキ
	制御方式		走行レバーによる無段階制御
	充電器		定電圧/定電流方式 (温度補正付)
性能	最高速度		1 ～ 6 km
	実用登坂角度		10°
	最小回転半径		1430
	段差乗り越え高さ		90mm
	溝乗り越え幅		120mm
	連続走行距離		約 25km(6km/h), 平坦路
	使用者最大体重		100kg(積載物を含む)

Table 2.2: 使用するセンサなどの機器

	搭載物	メーカー	型番	電源電圧 (V)	数量
	搭載 PC	AMP	XL800	+5V	1
	前方用 LRF	北陽電機	UTM-30LX	+12V	1
	リアタイヤ用エンコーダ	KEYENCE	FS-N10	-	2
	エンコーダ用アンプ	KEYENCE	FS-N10	12V	1
	ファイバースенса	AEC	5507	-	1
	ファイバセンサアンプ	AEC	5507	+12V	1
	GPS	Hemisphere	A100	+12V	1
	ハンドル制御用モータ	SAWAMURA	SS40E4	-	1
	モータドライバ	SAWAMURA	MS-100V05	+12V	1
	モータ位置制御アンプ	SAWAMURA	SAP-1210A	+5V	1
	DC/DC(+24V \rightarrow +12V)	COSEL	ZW102412	-	1
	DC/DC(+24V \rightarrow -12V)	COSEL	MGS5152405	-	1
	DC/DC(+24V \rightarrow +5V)	COSEL	MGS5302412	-	1

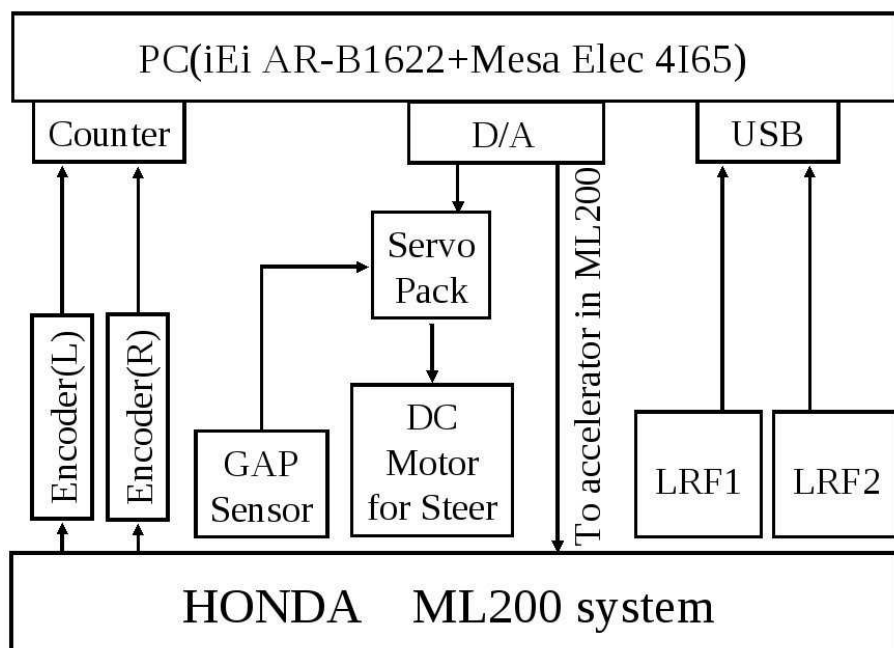


Fig. 2.2: MURV-2 三翠 システム構成図

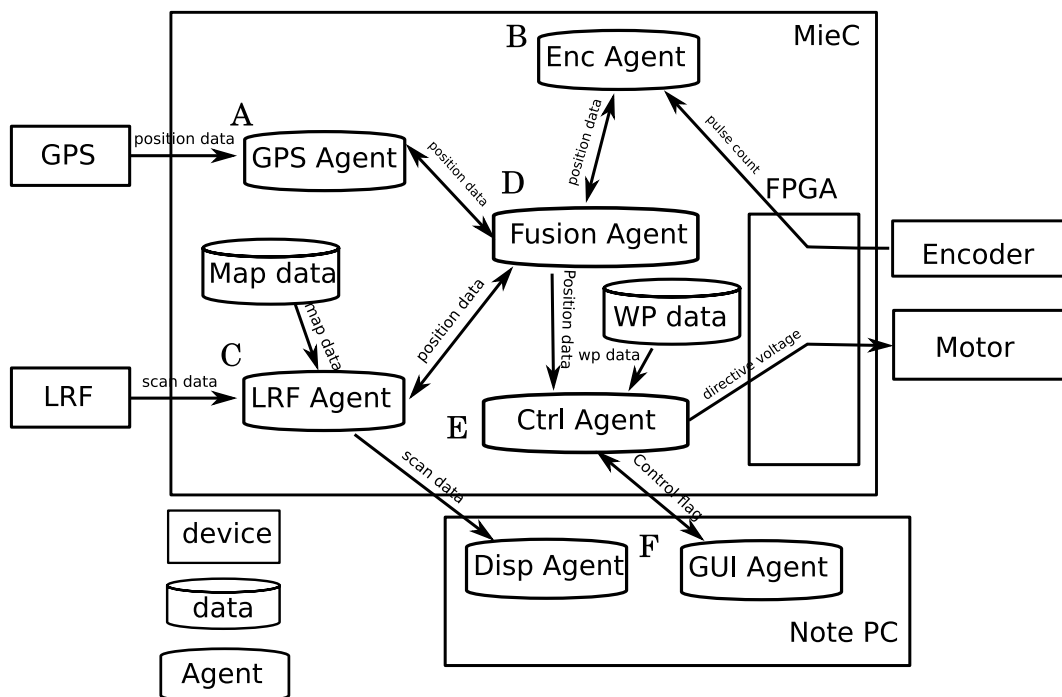


Fig. 2.3: マルチエージェントシステム構成図

第3章

自律移動プログラム

本研究では, 実際に自律移動ロボットを走行させて環境の出現確率を調査するにあたり, 以下に示す処理に基づく自律移動プログラムを作成した. 確実な走行を目指しロボットの機能と環境を適宜変更していくため, 最初のプログラムは比較的シンプルなもので構成する. そこで基準とするレベルとしては, つくばチャレンジにおけるトライアル走行がクリアできるものとし, 実際に各チームで実際に採用されているセンサや手法を取り入れ各処理を構成する.

3.1 エンコーダエージェント

ロボット後輪に取り付けられたエンコーダから FPGA を通じて車輪回転数を取得し, 以下に示す式を用いてデッドレコニングによる自己位置推定をした後にその推定結果を統合エージェントに送る.

$$L = \frac{(P_L + P_R)}{2} \pi D_w \quad [\text{m}]$$

$$\theta_R = \frac{(P_L - P_R) \arctan(\pi D_w)}{N W_R} \quad [\text{rad}]$$

$$X_{t+1} = X_t + (L_t - L_{t-1}) \sin \theta_R$$

$$Y_{t+1} = Y_t + (L_t - L_{t-1}) \cos \theta_R$$

ここで, L : 走行距離, θ_R : 車体角, X : ロボット座標系の x 座標, Y : ロボット座標系の y 座標, P_L : 左後輪エンコーダのパルス数, P_R : 右後輪エンコーダのパルス数, D_w : 車輪直径, N : 車輪 1 回転で得られるパルス数, W_R : トレッド幅, 添字 t : 現在時刻, 添字 $t-1$: 1 つ前の時刻である

また 100m の直線を走行させどの程度の誤差が生じるかの検証を行った。その結果を Fig. 3.1, Fig. 3.2 に示す。図より距離が増えるほどに角度がずれ、それにともない自己位置推定結果に誤差が生じる様子が見てとれるが、三本の走行で傾向は似ていることから左右の校正をとることで改善することができると思われる。よって実際に走行をする際には左右の値に補正係数を掛けることで対応している。

3.2 LRF エージェント

LRF は用途毎に計 2 つ搭載され、以下の役割を担う。

1. パーティクルフィルタを用いた地図マッチングによる自己位置推定
2. 前進時における前方障害物の監視

以下にそれぞれの詳細を示す。

3.2.1 パーティクルフィルタを用いた地図マッチングによる自己位置推定

富沢らは、自由空間モデルに基づくペナルティ関数や物体の光反射特性を考慮した環境地図生成法を提案し、パーティクルフィルタ及び二値格子地図を用いた地図マッチングを利用してつくばチャレンジ 2010 などで良好な結果を示した [12][13]。

本研究では、それらの手法を取り入れ自己位置推定のために利用する。以下に詳細を示す。

A. 二値格子地図の生成

走行環境をある一定の大きさを持つグリッドで区切りロボットの車体前方に取り付けられた LRF から得られる環境情報を用いて 2 次元の二値格子地図を作成する。

この二値格子地図では、LRF のレーザ光が一度でも通過した領域を走行可能領域として白で表現し、逆にレーザ光で観測したよりも奥の領域を侵入不可領域として黒で表現する。この二値格子地図をロボットの自己位置を中心とした 20x20m の範囲で作成し、それを重ね合わせていくことで大域地図を作成する。その後、走行中に逐次作成した局所地図と事前に作成した大域地図の間にロボットの自己位置付近でテンプレートマッチングをとることで自己位置を推定する。この手法を用いることで画像の単純な平行移動や回転で扱うことができる利点がある。

B. 局所地図の生成

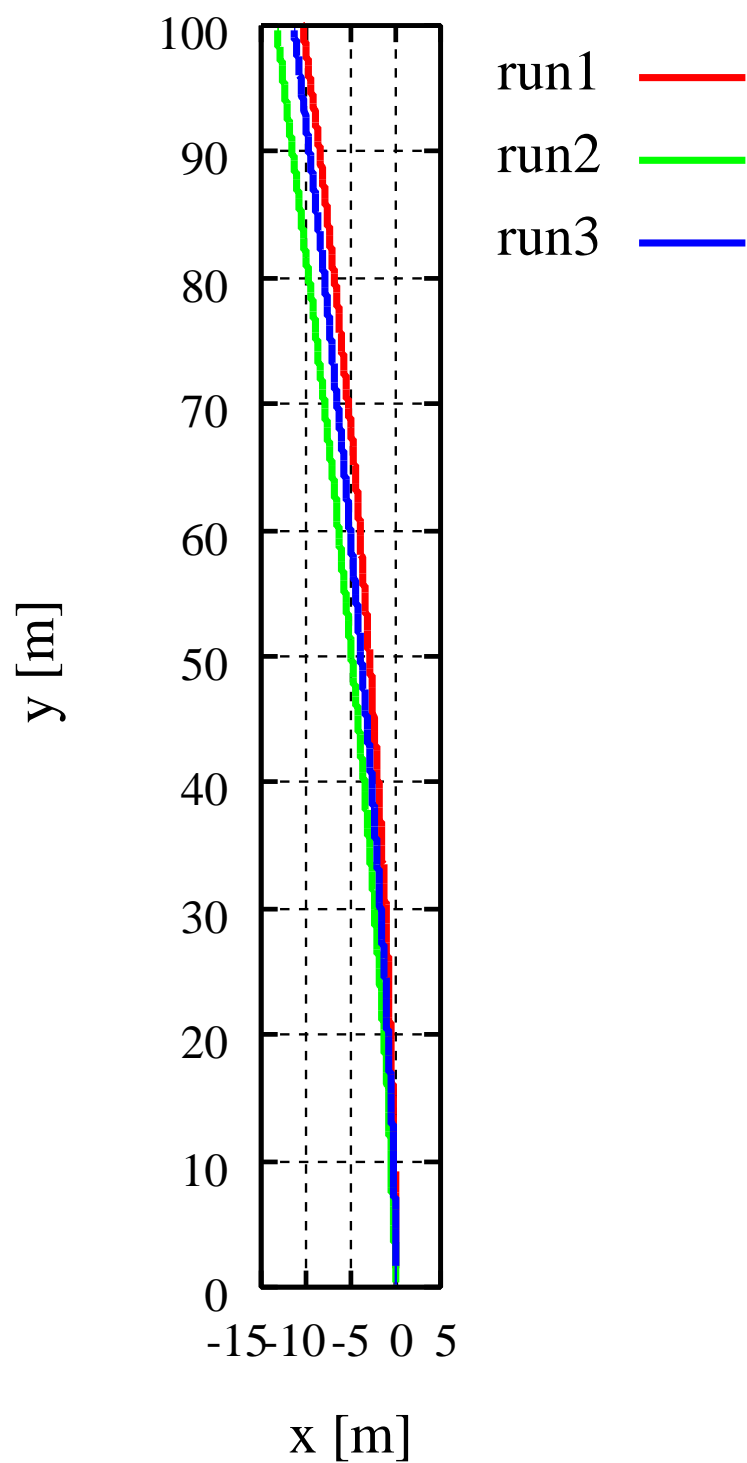


Fig. 3.1: 100m の直線走行時のデッドレコニングのみによる自己位置推定結果

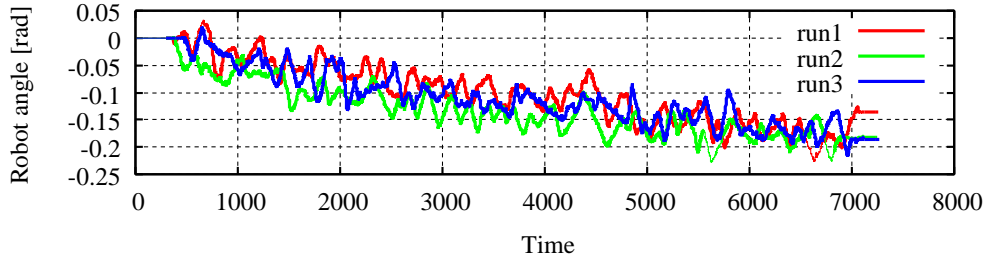


Fig. 3.2: 100m の直線走行時の角度変化

本研究では格子地図に用いるグリッド 1 つの大きさを 10cm 四方とし, そのグリッドを縦横 201 グリッド並べた局所地図を作成する. 局所地図作成の具体的な流れは以下のである.

a) LRF から各グリッドまでの距離と角度を算出する

Fig. 3.4 に示すように各グリッドへの距離と LRF 正面を原点とするグリッドへの角度をそれぞれ求める.

b) 各グリッドまでの角度に近い角度を持つ計測点を 2 点探索し, LRF から計測点までの距離を算出する

Fig. 3.5 に示すように現在計算しているグリッドへの角度 $\theta_{i,j}$ に近い角度を持つ 2 つの計測点 P, Q を算出する. ここでセンサからの角度が大きい点を P , 小さい点を Q とする. そして点 P と点 Q の角度と距離を $\theta_p, \theta_q, d_p, d_q$ として 2 点を通る直線の方程式から計測点への距離を算出する.

$$d_{obj} = \frac{d_Q - d_P}{\theta_Q - \theta_P} (\theta_{i,j} - \theta_P) + d_P$$

c) グリッドまでの距離 $d_{i,j}$ と計測点までの距離 d_{obj} を比較し, 局所地図のグリッドに白か黒を記入

上で求めた d_{obj} と $d_{i,j}$ を比較して対象としているグリッドが計測された物体よりも手前にあるか奥にあるかを判断する. 対象グリッドが計測点よりも奥にある場合にはグリッドを黒で塗り潰し, 逆に計測点よりも手前にあった際にはグリッドを白で塗る. これを全てのグリッドについて行うことで二値格子局所地図を作成する.

C. 大域地図の生成



Fig. 3.3: 二値格子地図の例

次にグローバルマップの生成について述べる. 生成するグローバルマップはローカルマップと同様に各グリッドの 1 辺の大きさが 10cm の二値格子地図である. また次に述べる地図マッチングと同様にグローバルマップの生成にはパーティクルフィルタを使用する.

グローバルマップの生成には事前に地図を生成する区間をセンサを搭載したロボットで手動走行させ取得した LRF の測域データとエンコーダのログデータを使用する. まずはじめに LRF から取得する測域データは計測点までの距離 r と角度 θ で表される極座標系であるため, これを xy 座標系に変換して使用する.

グローバルマップは初期段階では全体が黒で塗り潰されており, パーティクル数は 100 個である. 全てのパーティクルは等しい初期位置, 初期角度の値を持つ. 地図生成ループのカウンタ $t = 0$ の場合, グローバルマップ上の初期の自己位置からみた LRF データの計測点に該当する格子を白に変更する.

$t \geq 2$ の場合, 前回のパーティクルから尤度の高い上位 11 個を残留パーティクルとし, その中から尤度の比により上位のパーティクル程多くのコピーを残すように元の 100 個へ分裂させる.(詳細は地

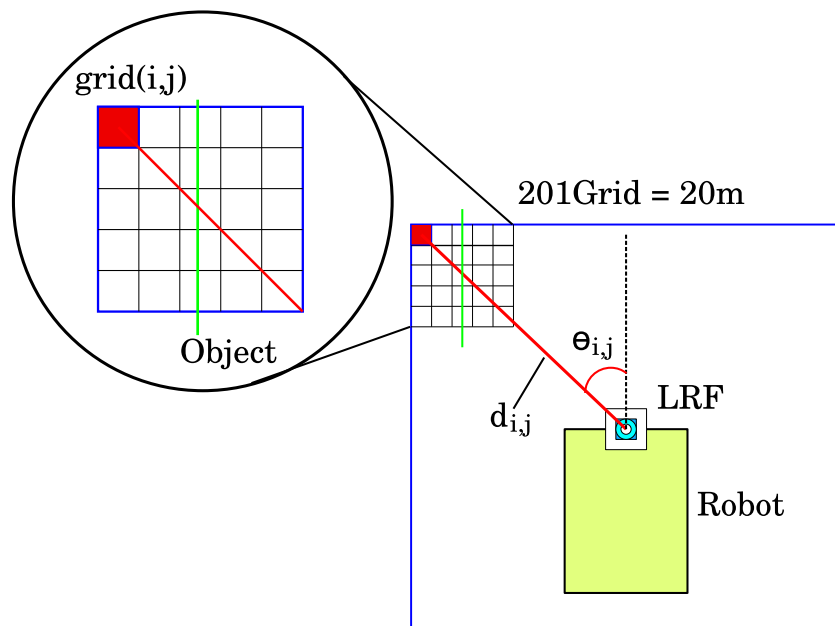


Fig. 3.4: LRF から各グリッドまでの角度及び距離

図マッチングの部分で述べる。) 次に各パーティクルをエンコーダデータに基づき移動させる。このとき値はひとつ前のエンコーダデータパルス値と現在のパルス値の変化量を使用し、そこにエンコーダの計測誤差を考慮した乱数を加算した値を用いて各パーティクル毎にデッドレコニングをすることで移動量を決定する。

次に LRF データから Fig. 3.6 に示す 100×100grid のローカルマップを物体を検出した格子を白に変更することで作成する。

各パーティクルの尤度の計算は、グローバルマップ上における各パーティクルの位置からグローバルマップをローカルマップと同じ大きさ (100×100grid) で切り出したパーティクルマップをローカルマップとの間で比較することにより決定する。実際には比較した両者の間で一致している白グリッドの数を 100×100grid のマップ全域で数えあげ、その数を尤度とする。このパーティクルマップの作成、ローカルマップとの比較という流れを全てのパーティクルに対して繰り返し行い全てのパーティクルに対する重みを決定する。

最後に尤度の高い上位 $N\%$ のパーティクルの値を使用して加重平均を計算し、それを自己位置としたのちに、決定した自己位置をにローカルマップを配置することでグローバルマップを更新する。この流れを繰り返し行うことでグローバルマップを生成する。

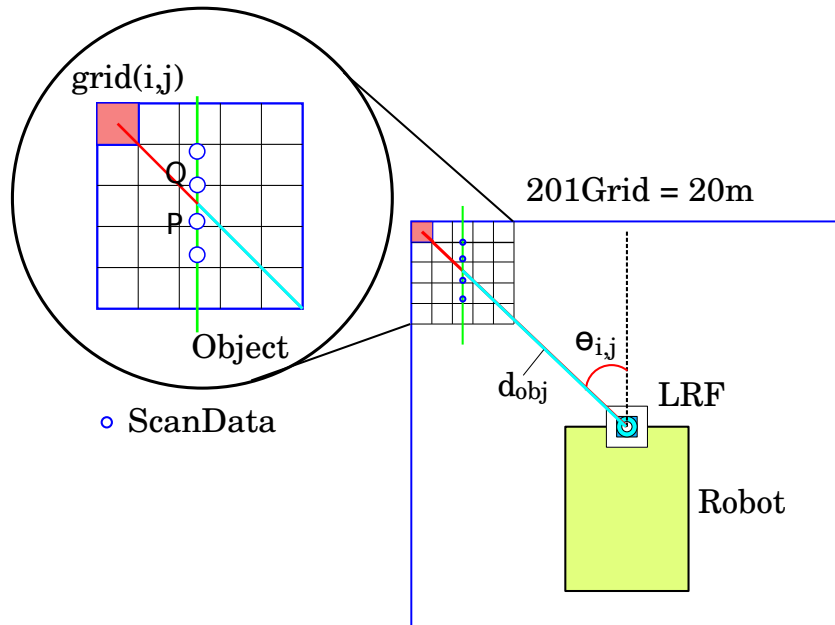


Fig. 3.5: 各グリッドまでの角度に近い角度インデックスを持つ計測点

全体の流れを Fig. 3.7 のフローチャートに示す. この流れでグローバルマップを生成した後に実際に走行する環境に合わせるためにペイントツール等を使用して人手で修正を加え, それを用いて地図マッチングを行う.

D. パーティクルフィルタを用いた地図マッチング

前章で作成した局所地図及び大域地図のマッチングをパーティクルフィルタを使用して行う流れを以下に示す. まず始めに各パーティクルの持つ位置・姿勢の値を一様分布を使用して初期化する (Fig. 3.8). x, y の位置を $-0.5 \sim 0.5[\text{m}]$ の間で, θ を $-5 \sim 5[\text{deg}]$ の間で分散させる.

次にエンコーダエージェントから送られてくる左右エンコーダのパルス数変化から車体の移動モデルに基づいてデッドレコニングを行いパーティクルを移動させる. その際にデッドレコニングの誤差を考慮して, 平均 μ を 0, 分散 σ をエンコーダ値変化量の $\alpha\%$ とする正規分布を加算してパーティクルを分散させている. そして各パーティクルが持つ位置の値を基に局所地図と同サイズのパーティクルマップを切り出し, 局所地図をそのパーティクルマップと角度が合うように回転させた後に尤度計算を行う. (Fig. 3.9)

尤度計算の基本的な流れは同サイズのローカルマップとパーティクルマップの画像を重ね合せ, 各ピクセル毎に and を取る形となる. パーティクルマップが黒の部分にローカルマップが黒で重なるピ

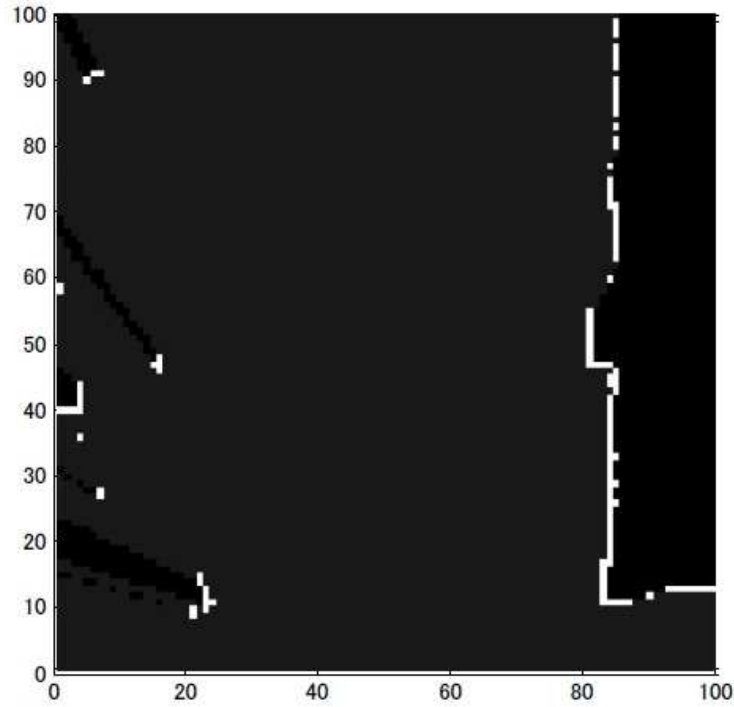


Fig. 3.6: グローバルマップ生成におけるローカルマップ

クセルの数を W_a , パーティクルマップが黒の部分にローカルマップが白で重なるピクセルの数を W_b , パーティクルマップが白の部分にローカルマップが黒で重なるピクセルの数を W_c , また 3 つのパターンの重み係数をそれぞれ C_a, C_b, C_c とすると各パーティクルの尤度 W_i は以下の式で計算される.

$$W_i = C_a W_a - C_b W_b - C_c W_c$$

このような尤度計算を行なうことで既知の障害物に近づきすぎたり, 離れすぎたパーティクルに対してペナルティをかけることができるため, より真値に近い推定結果を出すことが可能となる. このようにして求められた尤度を基に尤度の正規化を行い, その加重平均を推定結果とする. またパーティクルを尤度順にソートした後に上位 N_{lp} 個のパーティクルを残留パーティクルとして残し, その他のパーティクルは消滅させる. その後, 残留パーティクルが自身のコピーを増やす数 N_{cp} を決定し, 分裂させる.

$$N_{cp} = \frac{W_i}{\sum_{i=1}^{N_{lp}} W_i} \cdot (N_p - N_{lp})$$

以上の計算を繰り返し行うことで, 自己位置の推定をする. 全体の流れを Fig. 3.10 に示す.

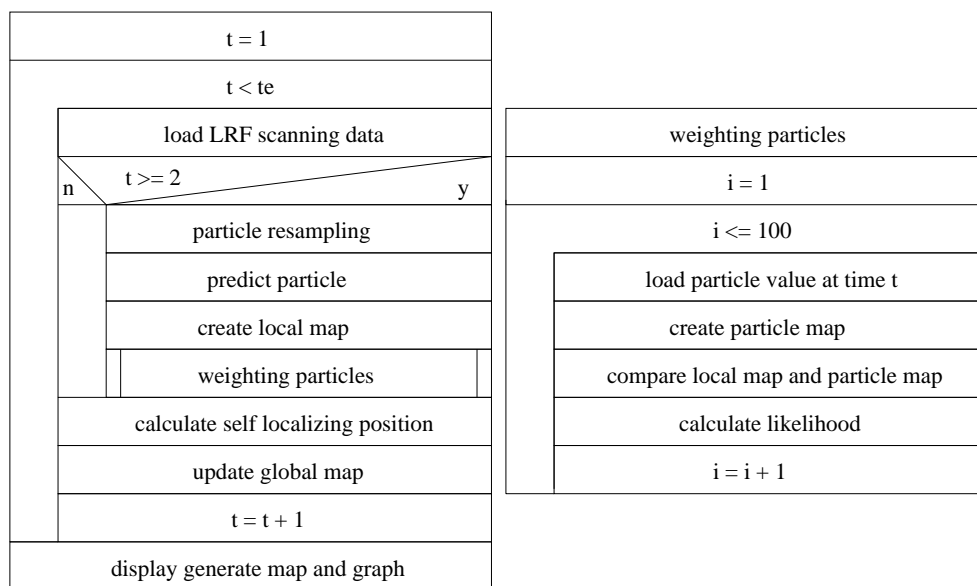


Fig. 3.7: グローバルマップ生成のフローチャート

3.2.2 前進時における前方障害物の監視

移動ロボットの前進時における前方障害物の監視には, 車体全面に斜め下を向けて搭載させた LRF を使用する. LRF から得られる距離データを用いて, 以下に示す 2 つの処理を導入し簡単な障害物回避を実現した.

以下の説明では LRF の走査平面において x 軸正方向を LRF の右側, y 軸正方向が LRF の正面とする.

A. 危険領域の回避

y 座標が道路の LRF データと閾値以上異なる点を危険点とし, 左右の危険点の個数に応じて危険点の多い領域を回避する.(Fig. 3.11) また, 左右の危険点数の合計が閾値以上となった場合, 移動不可能と判断して停止する. Fig. 3.11 中の A により危険点の探索範囲が変更可能である. 道路領域の検出では, y 軸方向に一定の幅を持った領域を移動させ, 領域内に含まれる LRF データの個数が最も多い領域を道路領域とする.

B. 緊急停止

Fig. 3.12 に示すロボットの前方にある範囲を緊急停止領域とし, この領域内に LRF データが検出された場合, 回避不可能であるとして停止する. Fig. 3.12 中の A ~ D により, 緊急停止領域が変更可

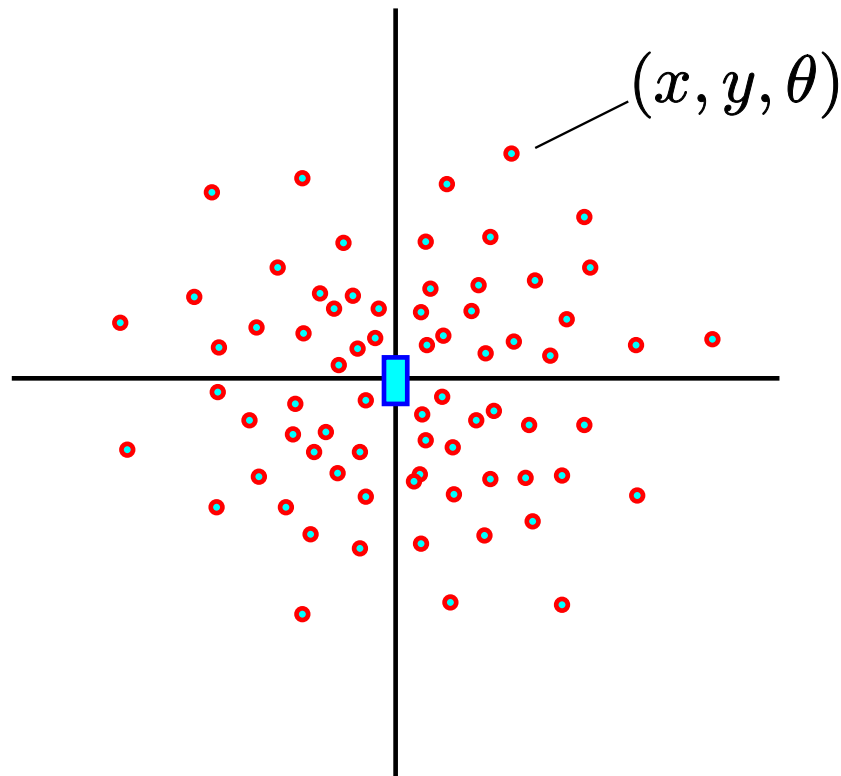


Fig. 3.8: パーティクルの初期化

能である.

自己位置推定結果及び障害物検知等の状態を統合エージェントに送る.

3.3 GPS エージェント

搭載される GPS からは NMEA-0183 フォーマットで文字列としてデータが送られてくる. その中から必要なデータを切り出し座標データを取得後統合エージェントに送る. GPS から送られてくるデータの一例を Table 3.1 に示す.

また本研究で使用する Hemisphere 製の A100 では, 衛星からの電波のみで位置を計測する単独測位モードと, 位置が既知の基準局が発信する補正情報を利用して精度を高める DGPS モードを利用することが可能である. GPS をナビゲーションに利用するにはその位置情報の精度が非常に重要となるため事前に A100 の精度実験を実施した. また, 同時に A100 の優位性を確認するため, 他の GPS ユニットと精度を比較した. 比較に使用した GPS は, Globalsat 製の ND-100 と CANMORE ELECTRONICS

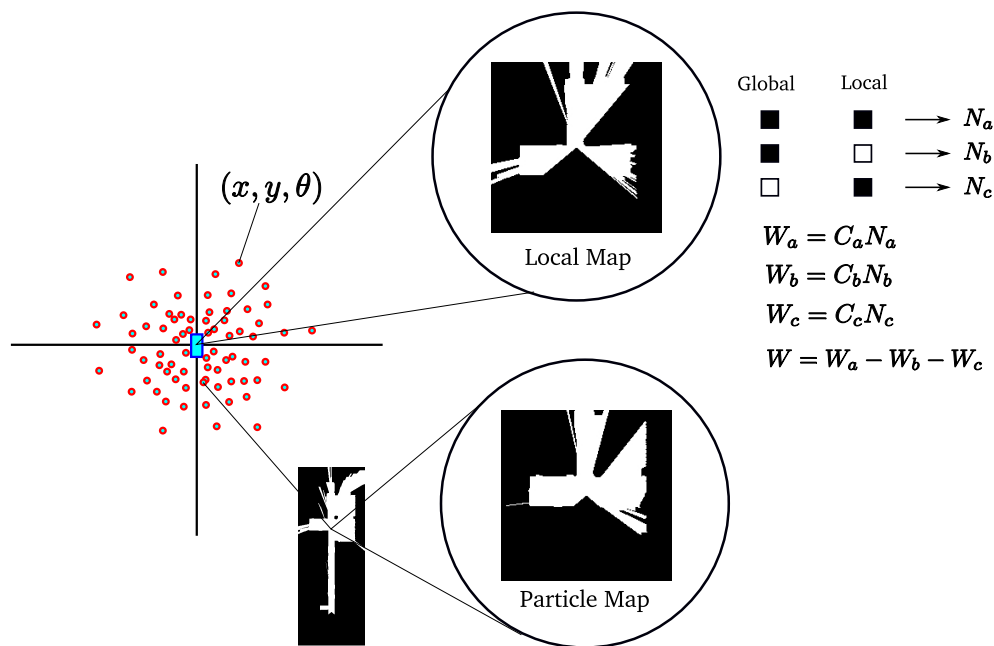


Fig. 3.9: 尤度計算

製の GT-730F/L であるこれらは市販されている GPS ユニットであり, A100 が 20 万円程度であるのに対し, 4,000 円程度であり, 低価格となっている. 実験は, 三重大学機械工学棟の屋上にて各 GPS ユニットを固定し, 1 分毎に 300 回の測位データを記録した. また, A100 に関しては, 屋上にて位置を変更し, 単独測位モードと DGPS モードの双方を測定した. その結果を Fig. 3.13, Table 3.2 に示す. なお, 図は x 軸正方向が東, y 軸正方向が北である. 結果より, 市販されている GPS ユニットでは, 自律移動ロボットのナビゲーションのための十分な精度が得られないことが分った.

一方, A100 では, DGPS モードの場合, 80cm 以内の誤差しか生じないため, 2m 程度の狭い道路の中心

Table 3.1: NMEA-0183 フォーマットの例

\$GPGGA,060306.00,4344.77894,N,14223.38857,E,2,11	
060306.00	測位時刻 (UTC)
4344.77894,N	緯度
14223.38857,E	経度
2	GPS ステータス
11	受信衛星数

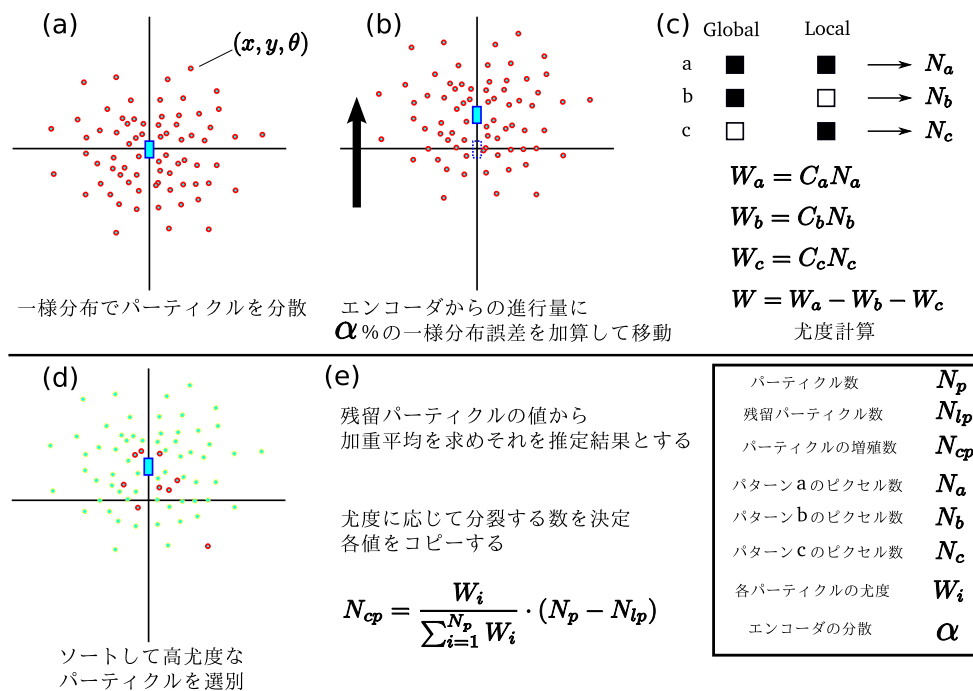


Fig. 3.10: パーティクルフィルタを用いた地図マッチングの流れ

の測定した場合でもほぼ全ての測定点が道路上を捉えることができ、ナビゲーションに十分な精度が得られた。

3.4 統合エージェント

統合エージェントは各センサから得られる値を他のエージェントへ渡す際の中継点としての役割を担う。統合エージェントに集められる情報は主に以下に示す 2 つのデータである。

- エンコーダエージェントからのデッドレコニングによる自己位置推定結果

Table 3.2: GPS の精度比較 2

	A100(DGPS)	A100	ND-100	GT-730F/L
平均	0.263	0.569	2.610	1.692
標準偏差	0.149	0.354	1.874	1.074
$E + 3\sigma$	0.710	1.631	8.232	4.914
最大誤差	0.727	1.940	11.927	6.155

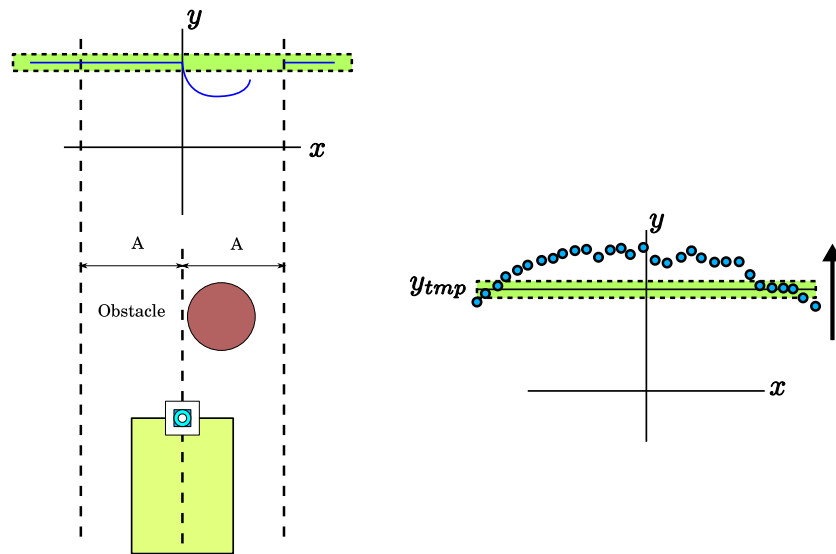


Fig. 3.11: 危険領域の回避

- LRF エージェントからのパーティクルフィルタを用いた地図マッチングによる自己位置推定結果

また、地図マッチングによって得られた推定結果とエンコーダエージェントが行なうデッドレコニングでの推定結果を統合する方法について述べる。

本研究で使用するロボットに搭載される MieC が持つ CPU は省電力性には優れるが、反面処理性能に関しては劣る面があり、パーティクルフィルタを用いた地図マッチングによる自己位置推定のサンプリングタイムは $1 \sim 2[\text{sec}]$ 程度となる。そこで以下に示すような手順でエンコーダエージェントのデッドレコニングによる推定結果を地図マッチングによる推定結果で補正することでそれぞれの結果を統合し、最終的な自己位置推定結果とする。

A. LRF エージェント

- パーティクルフィルタを用いた地図マッチングをサンプリングタイム毎に行う。
- 推定結果 (タイムスタンプ, x 座標, y 座標, 車体角度) を送信
- 以上を繰り返す

B. エンコーダエージェント

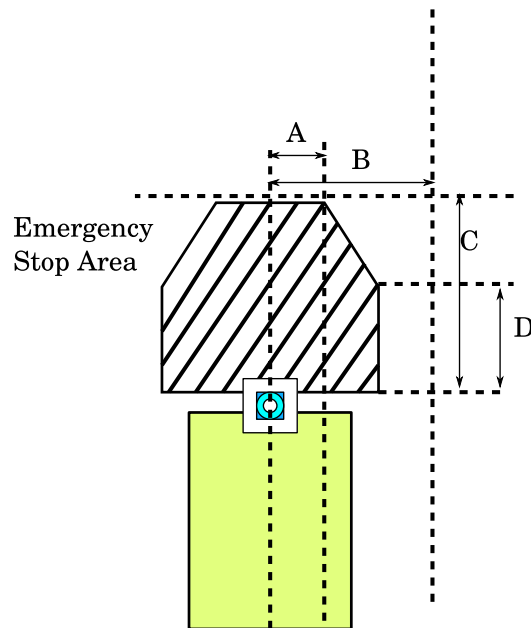


Fig. 3.12: 緊急停止

- LRF エージェントが先に起動されるので LRF エージェントからの推定値 $Pos_{pf}(x \text{ 座標}, y \text{ 座標}, \text{車体角度})$ を受け取り, その値を保存してデッドレコニングの一時積算値 $Pos_{tmp}(x \text{ 座標}, y \text{ 座標}, \text{車体角度})$ を $(0,0,0)$ で初期化する.
- エンコーダパルスから Pos_{tmp} を算出
- LRF からのデータが更新されたかを判定する. タイムスタンプが異なり, かつ位置・姿勢の内どれかが変化していれば最新の値として保存し, LRF からの推定結果更新フラグを ON にする.
- 更新フラグが ON ならば, 保存した最新の推定値にデッドレコニングの一時積算値 Pos_{tmp} を加算して現在値 $Pos(x \text{ 座標}, y \text{ 座標}, \text{車体角度})$ として制御エージェントに送信する.
- 更新フラグが OFF であれば現在値 Pos に一時積算値 Pos_{tmp} を加算して現在値として制御エージェントに送信する.

3.5 制御エージェント

A. 経路計画

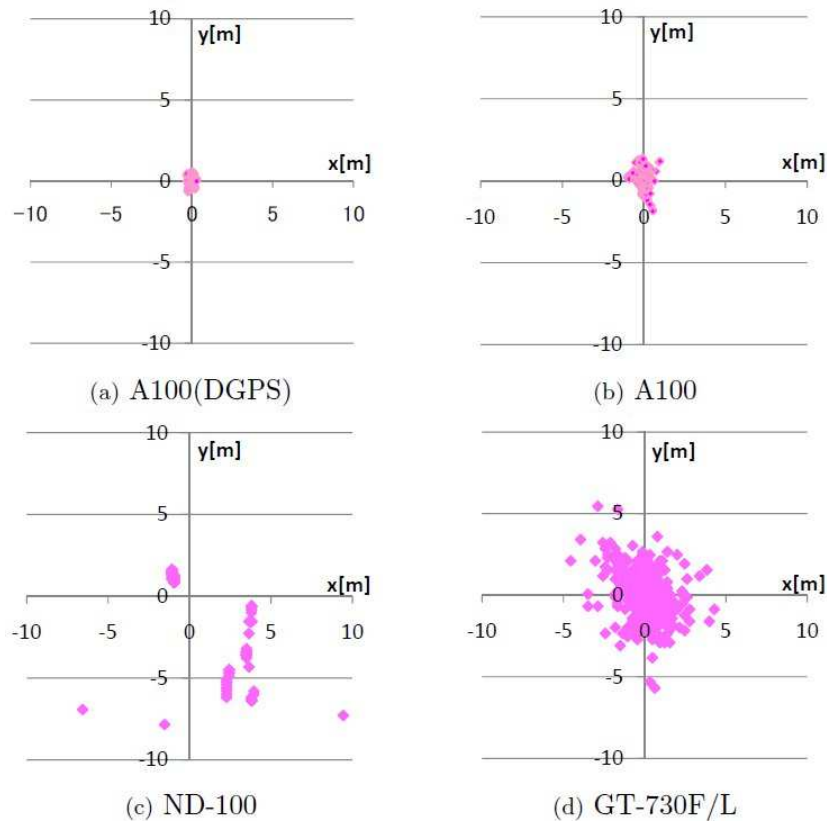


Fig. 3.13: GPS の精度比較 1

ロボットの経路計画については制御エージェントが担当をする．経路計画には，一般的な Point-to-Point 制御を用いる．これは，目的地までの経路上に複数のウェイポイントを設定し，ウェイポイントの到達及び次のウェイポイントを目標値に設定といった処理を繰り返すことで意図した目的地までナビゲートする方法である．本研究では，事前に作成した各ウェイポイントの座標や一旦停止の有無などを記したテキストファイルを経路データとしてロボットにあらかじめ与え，この制御を実行する．

B. ステアリング制御

ステアリングの制御では，現在目標としているウェイポイントと直前のウェイポイントとの間で引いた直線上をなぞるように制御を行う．

Fig. 3.14 に示すように現在のウェイポイント P_i と直前のウェイポイント P_{i-1} 間で引いた直線とロボットの進行方向との間の角度を θ ，ロボットと直線との距離を d とする．そのそれぞれにゲイン G_a, G_d を掛け，足し合せたものを S としてそれが小さくなるようにステアリングを制御することで，

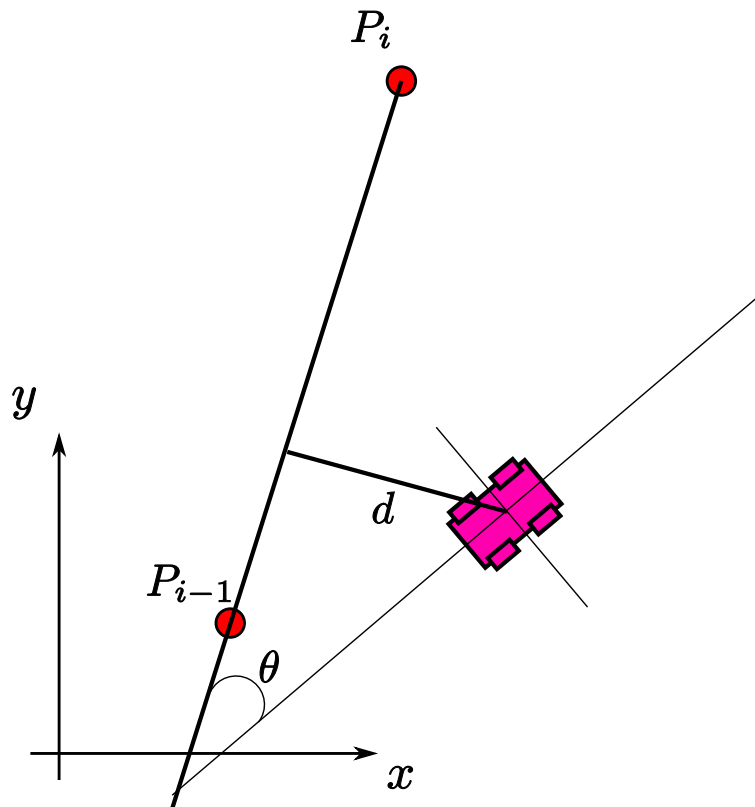


Fig. 3.14: ウェイポイント間で引いた直線との角度及び距離によるステアリング制御

直線を追従させる.

$$S = G_{\theta}\theta + G_d d$$

3.6 画面表示エージェント

ロボットの状態の表示や, ロボットの発進, 一時停止等を画面上のボタンから行えるようなインターフェイスを提供する (Fig. 3.15).

表示する情報としては, ロボットの車体角及び x 軸, y 軸速度, ステアリングモータ及び後輪モータへの出力電圧などが表示され, 画面左には周辺の地図が表示される.

また, パーティクルフィルタを用いた地図マッチングを使用中にはデバッグの効率化などを考慮して, 別ウィンドウに車体周辺のパーティクルの分散の様子を可視化して表示している (Fig. 3.16).



Fig. 3.15: ロボット操作用インタフェース

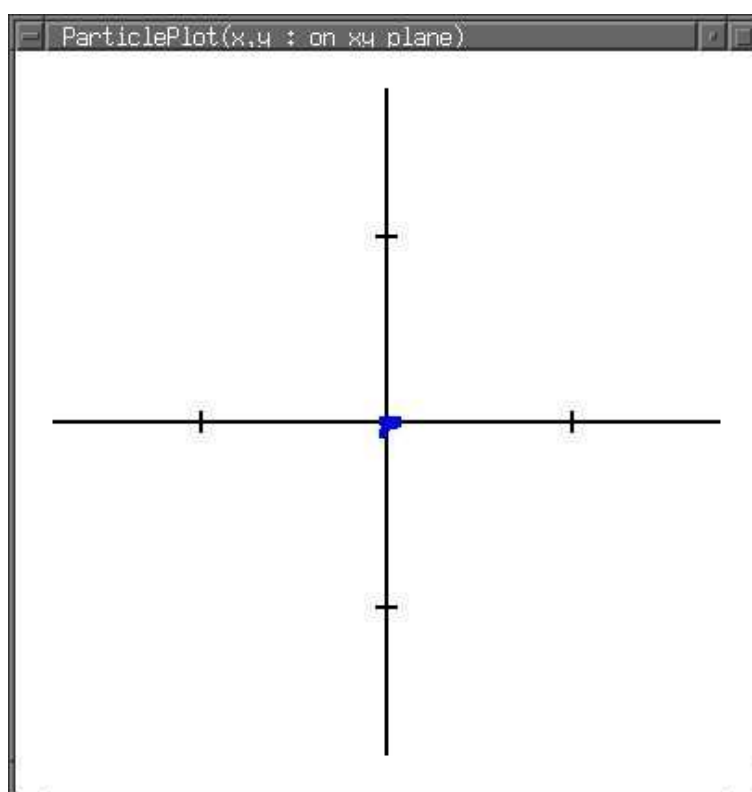


Fig. 3.16: パーティクル表示ウィンドウ

第 4 章

環境調査

今回, 自律移動ロボットの周囲環境を調査する場所として三重大学工学部を対象とした. 周囲環境による問題には機構系によるものと認識系によるものに大別できる. 機構系の問題とは, 自律移動ロボットが段差などにより機構的にそもそも移動することができないことによる問題とする. 用いる自律移動ロボットは電動四輪カートであり, 一般的な構造をしているためこれが移動可能であれば一般の自律移動ロボットも移動可能といえる.

4.1 環境の機構面に関する調査

実施した機構系の問題の調査に関して述べる. 実施内容としては, ロボットを手動で操作し三重大学工学部内を走行させ, 機構的に走行が不可能な場所を調査した. その結果を Fig. 4.2 に示す. 調査結果より, ロボットが通れない領域も多数存在しているしかし, 建物間の移動をというものを考えた場合には各建物の入口間の経路で移動が可能であればその領域における建物間の移動が可能であるとして問題はない (Fig. 4.1). よってそのような観点で調査結果をみるとほとんど全ての建物間の移動に問題がないことが分った. 建物によってはドアに段差があり侵入できないものもあったが工学部内にはそのような建物は 1 つしか存在していないため, 入口の段差という環境の出現確率は非常に低いと言える. 段差という機構面での問題についての解決案としては自律移動ロボットの足周りを改良してより大きな段差を越えられるようにする方法も考えられるが, 前述のようにこのような出現確率の低い環境についての対策としてはロボットの機構の変更という解決策は不適當であり, より容易に対策をすることができる段差スロープなどを問題のある箇所に設置するという解決がこのような場合にはより適していると考えられる.

段差スロープの設置等により, 問題の建物の入口段差が解消されれば三重大学工学部内の建物間の移動については機構的には問題がなくなると推測できる.

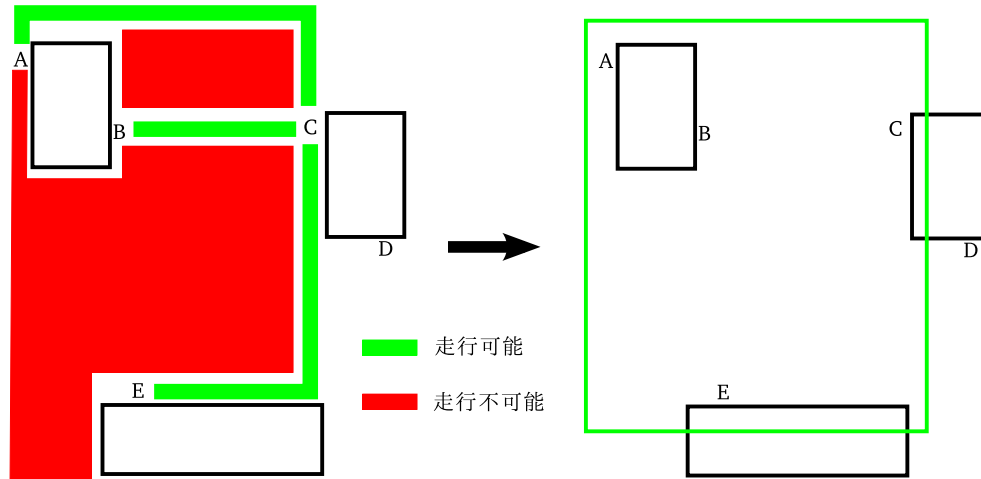


Fig. 4.1: 機構的に移動可能な領域の判定方法

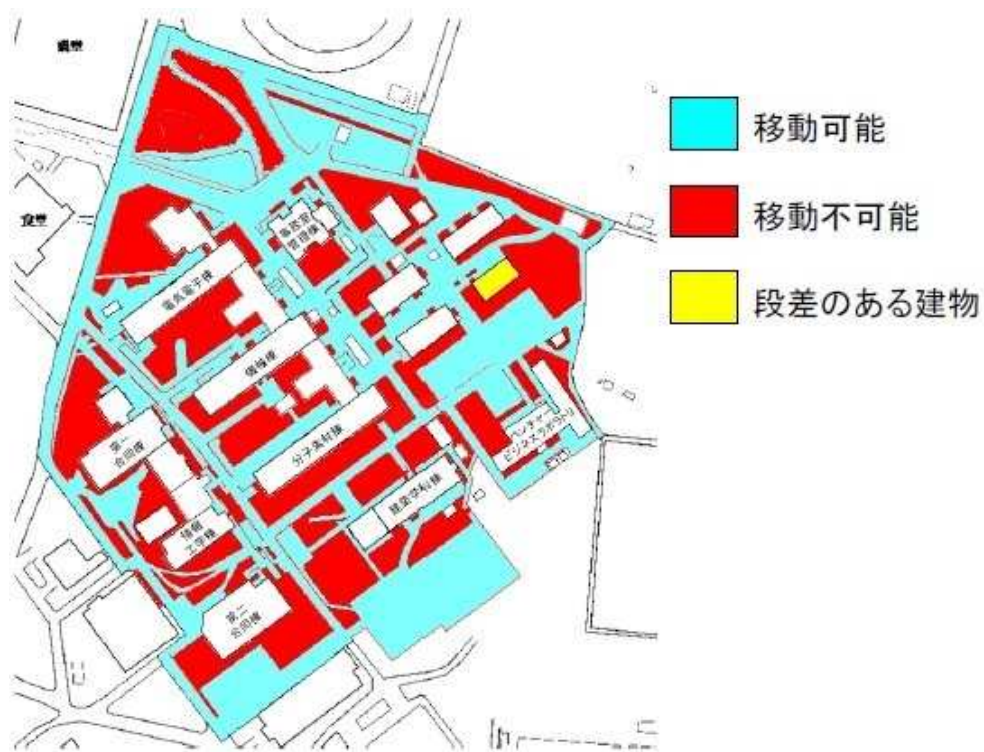


Fig. 4.2: 調査結果 (機構的に移動可能な領域)

4.2 環境の認識面に関する調査

本節では、自律移動ロボットの認識系の問題の調査について述べる。認識系の問題の調査では、実際にロボットを自律移動させ発生した問題を整理する。

まず始めにこれまでに構築した自律移動ロボットのシステムを使用して、自律走行が可能であるかの検証を行った。実験走行の環境としては、三重大学工学部学務周辺の道路を使用した。環境の外観を Fig. 4.3 に示す。

実験に使用するロボットは前述の電動四輪カートであり、使用するセンサは LRF, エンコーダである。エージェントは、極力シンプルな構成とすることを目的として LRF エージェント, エンコーダエージェント, 統合エージェント, 制御エージェントを使用し、パーティクルフィルタを用いた地図マッチングをする際の各パラメータは Table 4.1 に示す値である。また、マッチングに使用するグローバルマップを Fig. 4.4 に示す。



Fig. 4.3: 実験環境

今回の実験では Fig. 4.3 中の A から C の区間において自律走行の検証を行った。A から B の区間はスタート地点付近は右手側に建物があり、その輪郭をマッチングに利用できるため安定した走行が可能であるが、建物を通じた直後あたりから左手側には明確な輪郭を持つ領域が存在しないことや、右手側に自転車置き場があり (Fig. 4.5) 地図マッチングの難易度が高く、自己位置を誤って認識してしま

Table 4.1: パーティクルフィルタの各パラメータ

パーティクル数 N_p	50
残留パーティクル N_{lp}	10
パターン a の重み係数 C_a	1.0
パターン b の重み係数 C_b	1.3
パターン c の重み係数 C_c	0.1
エンコーダの誤差見積 α	1 [%]

うことが判明した. このように地図マッチングの難易度が高い理由としては, 自転車置場のような環境においては物体が存在し, レーザが通過しない領域と物体が存在しないためにレーザが通過する領域が繰り返し現れることで結果として楕状の二値画像が生成されてしまうからだと考えられる (Fig. 4.5 左側上 1 段目右から 1 番目の画像など). こういった環境では誤った位置でのマッチング尤度が高くなり, 自己位置の認識に問題が生じる可能性がある.

よって本研究ではこのような環境を認識面で問題のある環境として取り扱い, 今回の自律走行試験ではこの区間を除外した B から C の区間で走行させた. その結果を Fig. 4.6 に示す.

青線がエンコーダからの値を用いたデッドレコニングによる推定結果を, 赤線がパーティクルフィルタを用いた地図マッチングによる推定結果, + 印が目標軌道をそれぞれ示している. 区間 B から C の間にかけての環境には主に進行方向左側に植木が存在し, それが明確な輪郭を持つために図のように目標軌道に沿って自己位置推定をしつつ目的地まで走行することが可能であった.

次にこの結果を基にして走行環境の認識面での調査をした. 調査対象とする環境は第一合同棟から工学部事務管理までをベンチャビジネスラボラトリ (h)(以下 VBL とする) 方向経由で行く経路及び第二食堂方向経由 (g) で行く経路に含まれる領域であり, 領域内に存在し移動を考慮する建物としては電気電子棟 (a), 第一合同棟 (b), 機械棟 (c), 分子素材棟 (d), 情報工学棟 (e), 事務管理棟 (f)(Fig. 4.7) である. またそれぞれの建物間の移動目標地点として Fig. 4.8 を使用する. なお各経路が選択される事前確率は全ての経路で等しいとする.

調査の方法としては, ロボットに LRF を搭載し手動で該当する範囲を走行させてセンサデータを取得した後に, 前述の手順で二値格子地図へと変換した画像を用いて自律走行の結果を基に認識面で問題がないかを判断した. 調査した結果を Fig. 4.9 に示す. また以下に問題のある環境について述べる.

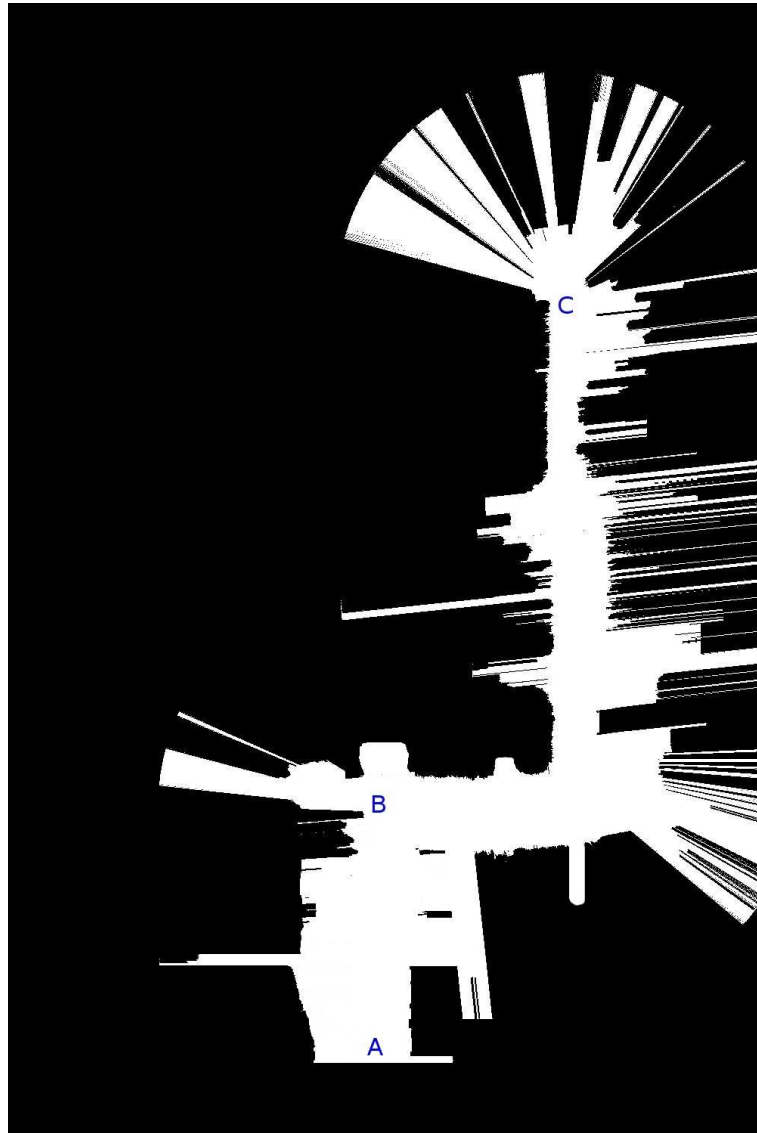


Fig. 4.4: 実験に使用したグローバルマップ

A. 車体近隣におけるマッチング対象の一時的欠如

VBL から情報工学棟前への直線などの区間ではマッチングの対象となる生垣が存在するもののその密度が低く、また両側に生垣が無い部分があるなど走行中一時的に車体近隣のマッチング対象が無いという状況が発生する。このような環境中においてもその区間が距離的に短いものであれば一時的に地図マッチングを停止させ、デッドレコニングのみで自己位置推定することで問題なく走行することが可能であるが、マッチング対象が欠如する区間が長距離におよぶと自己位置を大きく見失う可

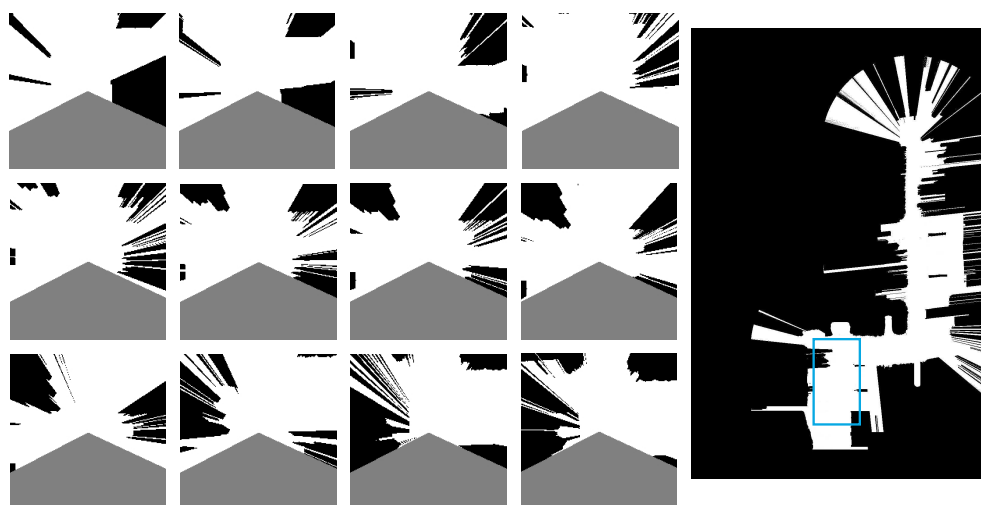


Fig. 4.5: 地図マッチングがうまく機能しない環境 (各ローカルマップ及びグローバルマップ)

能性がある。事前に 100m の直線をデッドレコニングのみで走行させたときの自己位置推定結果では、50m の地点で 5m ほど車体横方向の推定結果に誤差があったことから学内の道路幅などを考慮すると、20m 程度の区間であれば地図マッチングによる補正なしでも道路上にとどまりつつ走行することが可能と考えられる。

B. 路上に駐車されている自転車などの不規則に位置が変化する物体の影響

電気電子棟、第一合同棟、情報工学棟などに面する長い直線の区間に関しては普段多くの自転車が駐車されており、またその配置も不規則に変化しているため地図マッチングによる自己位置の補正は難しいと考えられる。またその区間の距離も 50m 以上と長いことから、デッドレコニングのみによる走行も困難である。

調査した領域に含まれる建物は計 6 棟であり、そのそれぞれの建物間の移動経路は計 15 通りである。各経路と各経路を走行中に発生すると考えられる上述の問題をまとめたものを Table 4.2 に示す。ここで電気電子棟 (a)、第一合同棟 (b)、機械棟 (c)、分子素材棟 (d)、情報工学棟 (e)、事務管理棟 (f) であり、経路が複数考えられる移動に関してはより問題の少ない経路を採用した。

この表を基にパターン A,B それぞれの出現確率を算出すると以下ようになる。

A のパターンは今回の調査範囲における結果では 80%と出現確率が高くなっている。また今回の調査範囲が狭いものであったにもかかわらず複数の箇所で問題のある環境が出現していることから今後調査範囲を拡大した際にも出現確率は高くなることが予想される。従ってこのような問題に対しては

Table 4.2: 各経路走行中に出現する問題のある環境

移動する建物	移動経路	出現する問題のある環境のパターン
a → b	最短経路	A,B
a → c	最短経路	A,B
a → d	第二食堂方面経由	A
a → e	第二食堂方面経由	A
a → f	第二食堂方面経由	A
b → c	最短距離	A,B
b → d	最短距離	B
b → e	最短距離	B
b → f	第二食堂方面経由	A,B
c → d	最短距離	A,B
c → e	最短距離	A,B
c → f	VBL 方面経由	A,B
d → e	最短距離	-
d → f	VBL 方面経由	A
e → f	VBL 方面経由	A

Table 4.3: 各環境の出現確率

パターン	問題点	出現確率
A	マッチング対象の一時的欠如	12/15 = 80[%]
B	自転車の不規則な配置の変化	9/15 = 60[%]

ロボットの機能の向上による解決が望ましいと考えられる。具体的な対応策としては、現在のシステムでは地図マッチングに用いるローカルマップは車体前方に搭載された LRF で計測した車体前方方向のデータのみを使用して生成している。そこで車体後方にも LRF を搭載し車体後方を計測したデータも使用して地図マッチングを行うことでより安定して自己位置推定をすることが可能になる。この方法では既存の前方のみを監視しているプログラムを一部修正するだけで比較的容易に導入できるという利点もある。また、GPS を搭載することでマッチング対象がほとんど存在しないような環境においても自己位置の認識が可能になるため問題を解決が可能と考えられる。

次に B のパターンでは今回の調査結果では出現確率が中程度となっている。このような結果となっ

たのは今回の調査範囲が狭く、それぞれの建物間の移動経路が似通ったものとなったことで問題のある同一の区間が複数の移動経路で出現するためであり、調査範囲内において問題となる環境は主に 1 箇所しか存在していない従って、調査対象を拡大した際には B のパターンのように不規則に配置が変化する物体のある区間が長く続くという環境の出現確率は少ないと予想される。従ってこのパターンでは、環境を操作することでロボットを支援する仕組みが望ましいといえる。具体的な対応策としては、不規則に配置が変化する物体のある位置に該当するグローバルマップの部分で灰色で塗り潰すことで地図マッチングを無効にすることが挙げられる。その上でマッチング対象となる目標物をデッドレコニングが破綻しない間隔で経路上に配置し、グローバルマップに目標物を追加することで作業的・金銭的に少ないコストで対策が可能と考えられる。

4.3 環境調査に関するまとめ

以上の三重大学工学部に関する機構面及び認識面での環境調査により以下の結果が判明した。

Table 4.4: 環境調査に関するまとめ

問題の分類	問題点	出現確率	対策
機構面	乗り越え高さを超える段差	低	段差スロープの設置
認識面	マッチング対象の欠如	高	後方用 LRF の追加, GPS の追加
認識面	駐車されている自転車	中	大域地図の加工, 目標物の設置

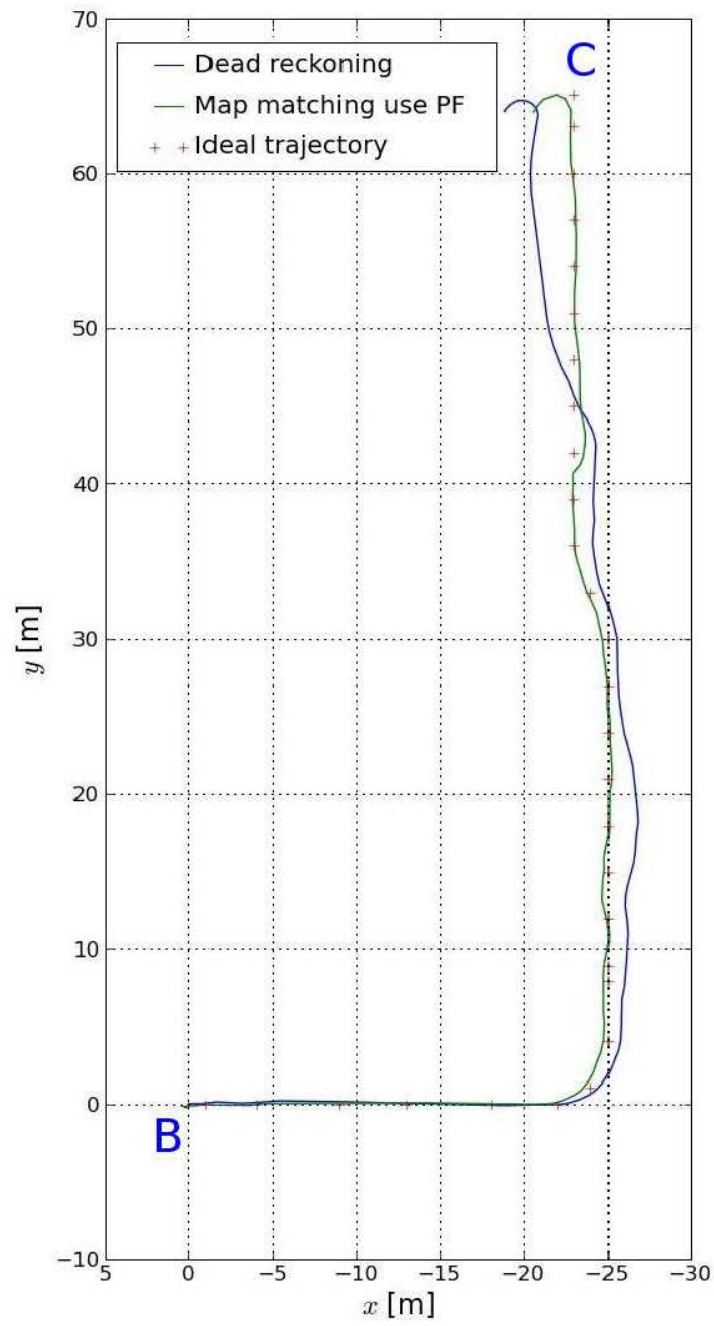


Fig. 4.6: 自律走行試験結果

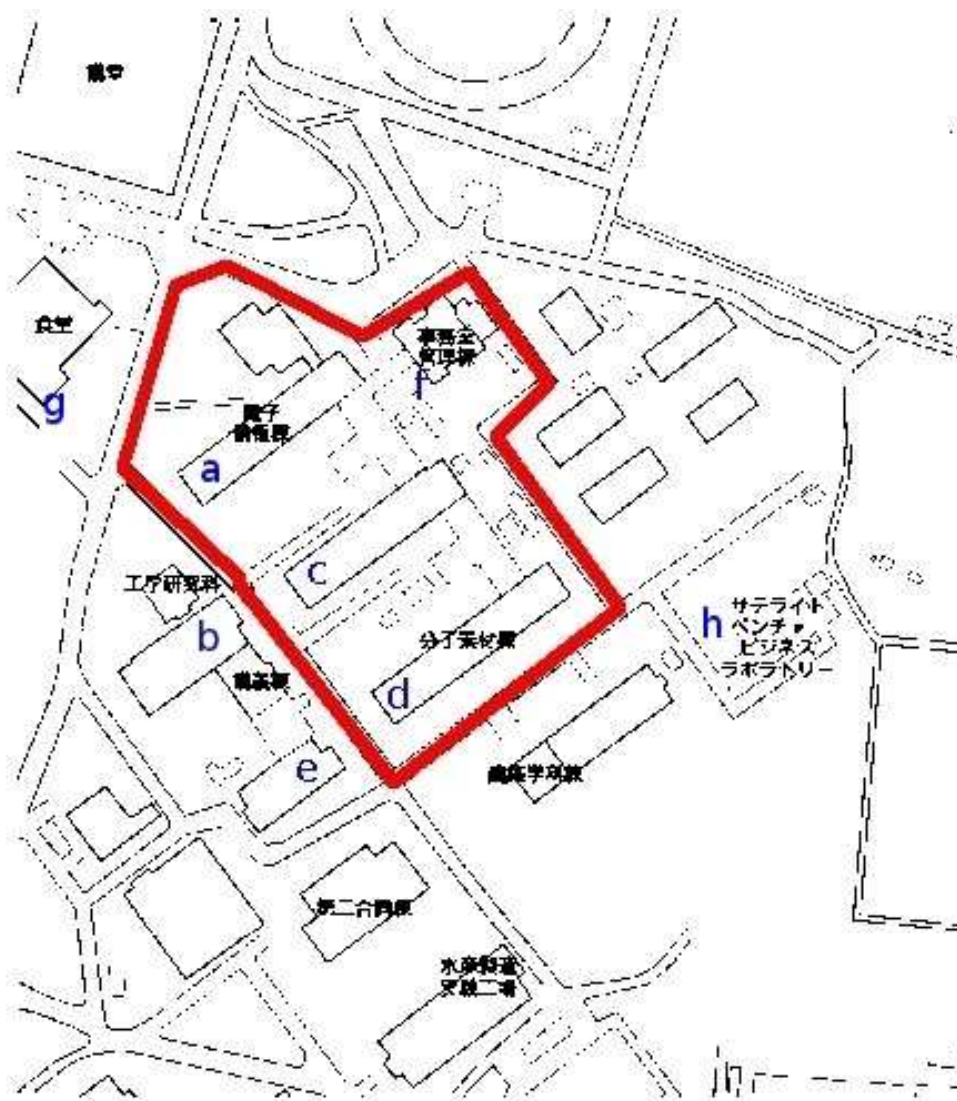


Fig. 4.7: 認識面での環境調査を行う環境

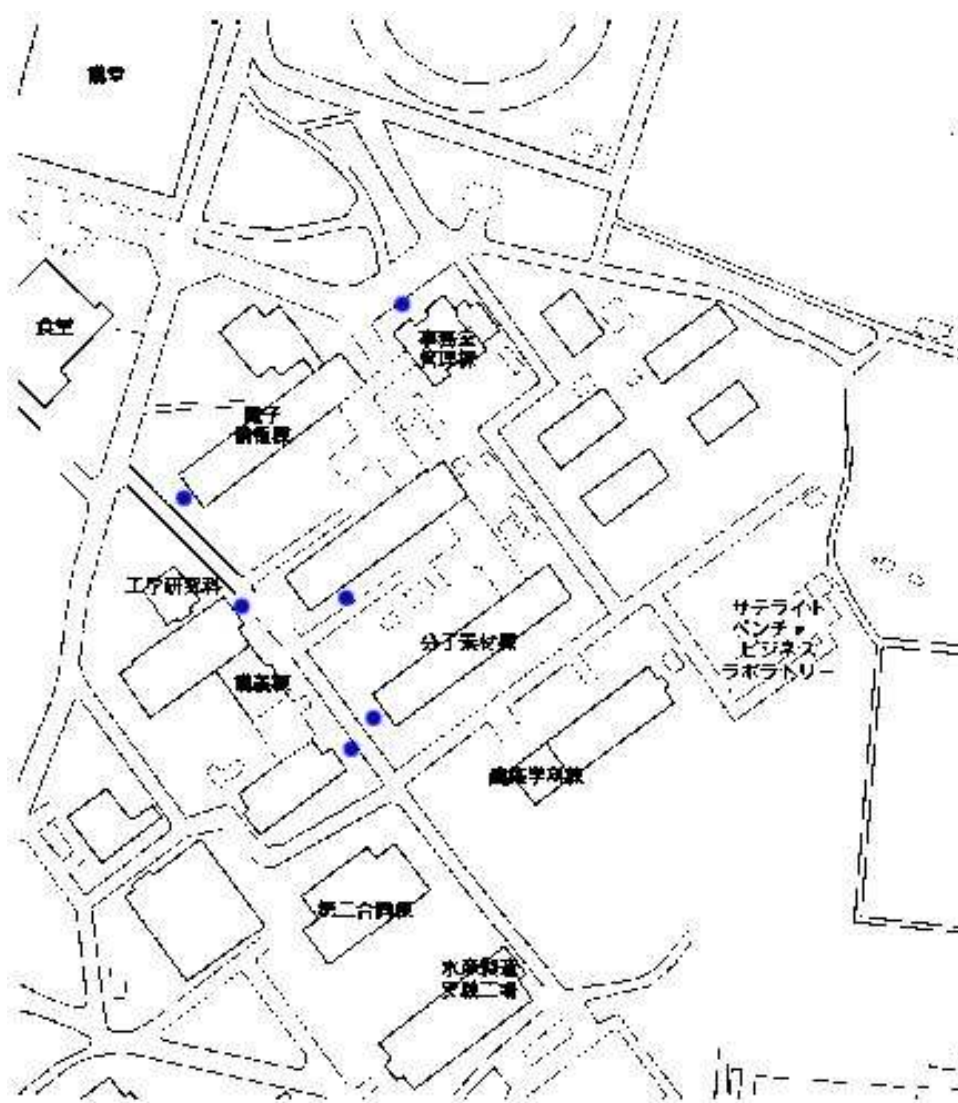


Fig. 4.8: 各建物の移動目標地点

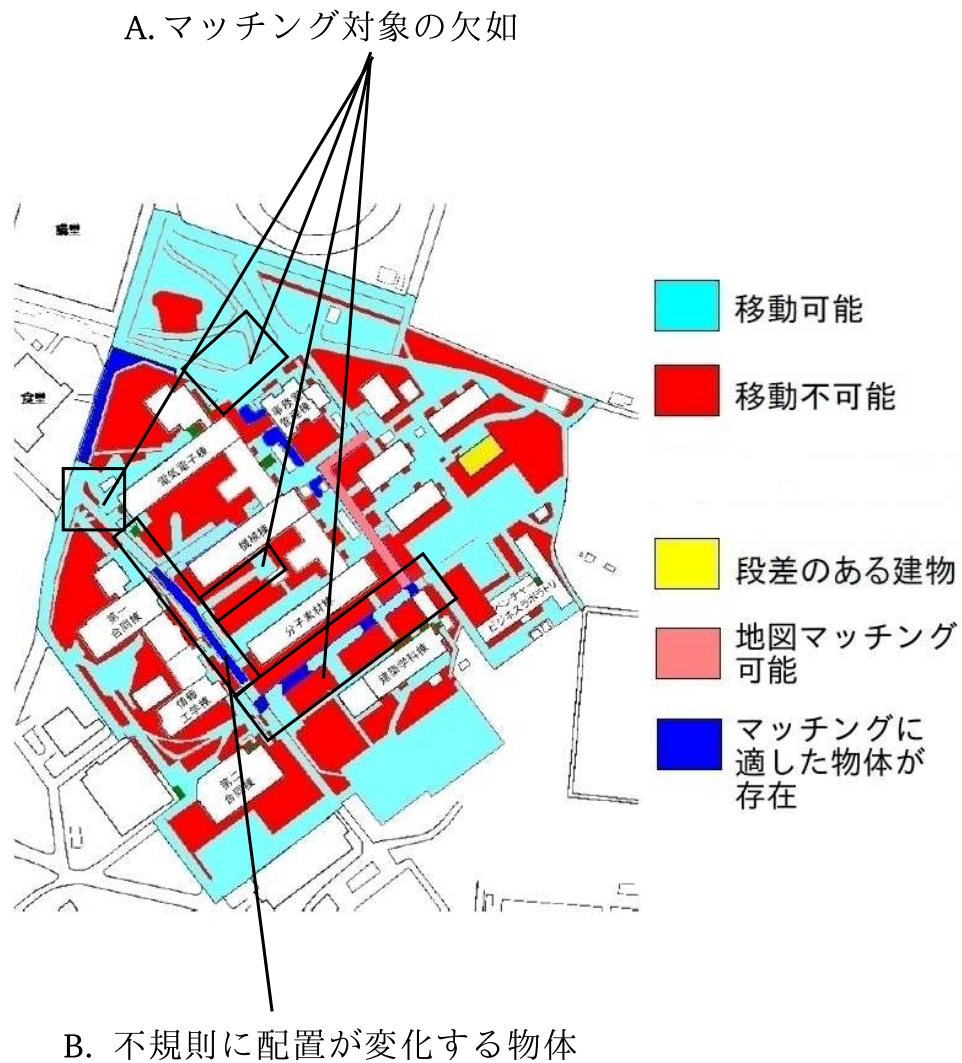


Fig. 4.9: 調査結果 (環境の認識面における調査)

第5章

むすび

移動ロボットが実環境中において自律走行を実現するためには、ロボット自身の機能を向上させる方法や、走行環境を操作することで環境側からロボットを支援する方法の 2 通りが考えられる。しかしそのどちらの方法においても導入負荷の問題などが存在しているため容易ではない。

そこで本研究では、野外実環境において活動する自律移動ロボットの実現の際により少ない労力でシステムを構築するために環境の出現確率に着目して問題を場合分けし、環境を操作するのか、ロボットの機能を向上させるのかそれぞれの環境に適した方法を適宜選択することでより少ない労力でシステムを構築できるロボットが走行をする環境もロボットの機能と同様にシステムの一部として捉える自律移動システムの提案を目指す。

そのために環境の出現確率や問題のある環境の調査をするための基準となるシステムとしてマルチエージェントシステムをベースとしたシンプルな自律移動システムを構築した。また、環境調査の初期段階としてモデルとする実環境中を実際に走行させるなどして走行時の問題点を調査し、環境の出現確率に着目した際に望ましい解決策の提示をした。

今後の課題として、ほとんどの建物間の移動は機構的には問題が無いことが環境調査により判明したが、実際に建物の入口を通過するためにはスロープを走行する必要がある。そこで、そのためのプログラムをまず作成する。また、今回の環境調査は工学部内の一部の領域であり、自律走行による調査ができたのはさらにその中の一部であった。従って今後は調査の範囲を工学部全体や、三重大学全体などのように拡げていき、より広範囲を調査した結果としての問題のある環境の出現確率を算出していくことが重要である。そして最終的・将来的には、少ない導入負荷で移動ロボットが安定して自律走行をすることが可能になるためにどのような環境操作をすればよいのかという環境構築に関する有効

な提案をする.

謝辞

本研究を遂行するにあたり，終始適切なご指導を頂き，熱心に御討論いただきました，三重大学大学院工学研究科准教授 加藤典彦 先生に深く感謝いたします。また，研究を行う上で数多くの貴重なアドバイス等をいただきました，三重大学大学院工学研究科助教 松井博和 先生にも深く御礼を申し上げます。

三重大学大学院工学研究科教授 矢野賢一 先生には，研究を行う上で必要となる知識や姿勢などについて御指導をいただきました。深く御礼を申し上げます。

三重大学大学院工学研究科准教授 早川聡一郎 先生及び三重大学大学院工学研究科技術職員 澤井秀樹 先生にはつくば関連で数多くの貴重なアドバイスや助力を頂き深く御礼を申し上げます。

また研究室諸氏には，本研究に対する御討論を頂きました。それぞれの皆様に深く感謝いたします。

参考文献

- [1] DARPA “URBAN CHALLENGE” <http://www.darpa.mil/grandchallenge/index.asp>, 2009
- [2] 村井亮介, 酒井龍雄, 上松弘幸, 中嶋久人, 三谷宏一, 北野齊 “自律移動ロボット群による搬送システムの実用化” 日本ロボット学会誌, Vol.28 No.3 pp.311-318, 2010
- [3] 関淳也, 青山元, 石川和良, 石村左緒里, 和田迫鉄矢, 薩見雄一, 横田和隆, 尾崎功一, 山本純雄 “視覚によるライン・トレースと磁気タグの併用による走行制御を用いた搬送ロボットの開発” 日本ロボット学会誌, Vol.27 No.8 pp.833-841, 2009
- [4] 村上剛司, 長谷川勉, 木室義彦, 千田陽介, 家永貴史, 有田大作, 倉爪亮 “情報構造化環境における情報管理の一手法” 日本ロボット学会誌, Vol.26 No.2 pp.192-199, 2008
- [5] 木室義彦, 荒屋亮, 有田大作, 家永貴史, 村上剛司, 楊智梅 “ロボット地理空間情報システム R-GIS” 日本ロボット学会誌, Vol.27 No.8 pp.868-876, 2009
- [6] 油田, 水川, 橋本 “つくばチャレンジのねらいと成果” 計測と制御, Vol.49 No.9 pp.572-578, 2010
- [7] 根岸, 三浦, 白井, “全方位ステレオとレーザレンジファインダの統合による移動ロボットの地図生成” 日本ロボット学会誌, Vol.21 No.6 pp.690-696, 2003
- [8] 子安, 三浦, 白井 “不確かさを考慮した複数のスキャンマッチング結果の統合による移動ロボットの移動量推定” 日本ロボット学会誌, Vol.23 No.2 pp.263-271, 2005
- [9] 倉爪, 戸畑, 村上, 長谷川, “CPS SLAM の研究—大規模建造物の高精度三次元幾何学形状レーザ計測システム” 日本ロボット学会誌, Vol.25 No.8 pp.1234-1242, 2007
- [10] 滝田, 伊達, “Smart Dump 3 で実現した自律機能” 計測と制御, Vol.49 No.9 pp.636-639, 2010
- [11] 岡村, 石田, 大矢, “つくばチャレンジ 2009 への取り組み [富士ソフト・筑波大学チーム]” 計測と制御, Vol.49 No.9 pp.604-607, 2010

- [12] 岩井, 村松, 富沢, 末廣, 工藤 “物体の光反射性を考慮した自由空間モデルに基づく環境地図生成” 日本機械学会ロボティクス・メカトロニクス講演会 2011, 2A1-O03, 2011
- [13] 富沢, 村松, 平井, 岩井, 佐藤, 御堂丸, 工藤, 末廣 “公共空間を自律移動する電動カートの開発 自由空間形状の画像マッチングによる未知障害物にロバストな自己位置推定” 日本機械学会ロボティクス・メカトロニクス講演会 2011, 2A1-M04, 2011