

詳細な環境モデルを用いない移動ロボットの  
位置センサ情報に基づく  
位置ナビゲーションに関する研究

1999年8月

安藤 吉 博

11  
1111  
111  
111

詳細な環境モデルを用いない移動ロボットの  
距離センサ情報に基づく  
屋内ナビゲーションに関する研究

1999年3月

安藤 吉伸

寄	贈
	年
	月
	日

# 論文概要

## 本研究の目的

本研究の目的は、自律移動ロボットに詳細な環境モデルを用いることなく移動ロボット自身に搭載した距離センサの情報をたよりに屋内をナビゲーションさせることにある。

一般に移動ロボットのナビゲーションを取り扱う研究では、移動ロボットが、1. 自己の位置を自ら推定できるか否か、2. その走行環境に関する環境モデル(地図)を持っているか否か、3. 環境中のランドマークをセンサにより認識し、かつ地図中に記述されたランドマークとのあいだで対応をとることができるか否か、の組合せによって研究の問題設定がなされてきた。

例えば、オドメトリ(車輪の回転数を計測することにより、移動ロボットの自己位置・方位を推定すること)により自己の位置座標が推定でき、地図を持ち、実環境中のランドマークの認識と地図中のランドマークとの対応付けができる場合には、常に自己位置の把握を行いながらのナビゲーションが可能になる。さらにオドメトリによる推定自己位置をランドマーク認識時に修正して位置の誤差を減らすことができる。すなわち、オドメトリの誤差累積による位置の誤認から、環境中に存在する物体に誤って衝突してしまう危険を著しく減らすことができ、「35m 直進し、90° 時計周りに回転し、20m 直進し、90° 反時計周りに回転し、30m 直進し、停止せよ。」というように姿勢位置や距離を明示的に記述する命令群により走行経路を記述することによるナビゲーションが可能になる。このアプローチでは、移動ロボットに細かい動きを指示できる。しかし、このような明示的な命令は、距離や方位を正確に与える必要があり、これらの命令群を作成するためには、人間があらかじめ巻尺などの距離測定器具を使って環境の距離情報を測定したり、移動ロボットを事前に走行させ環境に関する

情報を得ておく必要があり、経路や環境が変わるたびに準備に大きな手間がかかるという問題がある。

また、オドメトリによる自己位置推定が可能で実環境中のランドマーク認識は可能であるが、地図は事前には与えられずその移動ロボットにとって環境が未知である場合、その環境の中を徘徊することによってロボット自身はその走行環境の情報を地図として学習する、という問題設定がなされる。従来の移動ロボットのナビゲーションに関する研究はここで述べた2つの問題設定のどちらかに属するものが多かった。

一方、人が廊下を歩く時を考えると、まず自分の位置は座標として把握しているわけではなく、むしろ「この廊下に沿って進み、2つ目の交差点を右折して3番目の右の部屋に行く」というような意識で行動していることが多い。この行動意識のなかには既に環境に関するモデルがその行動に必要なだけ埋め込まれており、その行動では立ち寄らない場所まで環境モデルが記述される必要はない。しかし、このような立場で移動ロボットのナビゲーションを論じ、かつ実装まで行われた例は皆無ではないにせよ、事前に地図を与え、オドメトリによる自己位置推定を行いながら環境内を移動するナビゲーションに比較してあまり例を見ない。

そこで、筆者は、事前に詳細な環境モデル(地図)を移動ロボット上に与えずに、屋内の主に廊下環境において移動ロボットにナビゲーションを行わせることを研究テーマとした。事前に詳細な環境モデルを与えないかわりに、環境に対してインタラクティブに、すなわち、廊下に壁づたいに走行する、あるいは正面に障害物があれば迂回する、などの動作を行う必要が生じる。このための環境を認識する手段として移動ロボットの周囲の対象物までの距離を広い角度範囲にわたって知ることのできる距離センサを用いる。本論文は、大きく分けて5つの研究成果を述べている。その第1は、ソナーリング(超音波距離センサを円環状に12個配置したセンサ)からの距離情報により移動ロボットに壁沿い走行を行わせる研究、第2は、ソナーリングの距離情報による移動ロボットの壁沿い走行とオドメトリを利用する位置ベース走行の組合せによりナビゲーションを行う研究、第3は、超広角カメラと扇状光による広い角度にわたって距離を測定できるレーザレンジセンサの開発、第4は、レーザレンジセンサによる距離情報に基づく移動ロボットの廊下長手方向沿い走行の研究、そして、第5は、言語的指示シーケンスによる移動ロボットの広範囲屋内ナビゲーションの研究である。本学位請求論文におけるこれら5つの研究成果は、それぞれ移動ロボットの屋内ナビゲ

ションの研究に貢献するものである。以下、これら5つの成果について概要を述べる。

## ソナーリングによる壁沿い走行

本研究は、自律移動ロボットが与えられた環境内で超音波距離センサによる対象物までの距離情報をたよりに、形状のわかっていない壁に沿いながら走行（ウォール・フォローイング）する問題に対するものである。超音波距離センサを用いる移動ロボットのウォール・フォローイング問題に関して、Turennotらは、ロボットの真横にある左右1つずつの超音波センサを持つロボットが、超音波センサによる左右にある壁との距離情報をもとに壁との間隔を一定にしながら走る制御方法を提案し、実験を行なった。それは、直線の壁を前提にしており、途中に不連続なところがある場合はオドメトリを用いて直線に追従するように制御するものであった。

しかし、一般に環境内にはいろいろな形状の壁（凹凸のある壁、曲率のある壁）が存在する。筆者らは、自律移動ロボットが、ソナーリングを用いて壁の形状にロバストにウォール・フォローイングをするアルゴリズムを検討し、実験を行なってその有効性と問題点を明らかにした。

ロボットが実時間で、センサに基づいて行動する場合、その行動アルゴリズムはロボットの特性・機能に大きく依存する。本研究ではまず移動ロボットとして筆者らの有する研究用の自律移動ロボット「山彦」にソナーリングを搭載したものを想定し、その上で働くウォール・フォローイングのアルゴリズムを検討した。

このアルゴリズムでは、ロボットはその筐体の中心が壁からあらかじめ決められた距離  $D$  をもって走行させるものとした。ソナーリングには、超音波距離センサが12個、円環上に配置されている。前方正面を中心に左右  $22.5^\circ$  おきにこれを配置した。この各々のセンサからの距離データの組み合わせを考察し、起こす行動を4つのケースに分けた。これをある決められた距離  $d$  の走行ごとにそのうちのどの行動をとるかを選ぶことにより壁沿い走行を実現した。

ロボット「山彦」上に上記アルゴリズムを実装し、いろいろな環境の中で壁沿い走行の実験を行った。

この結果、いろいろな環境下で、超音波センサで壁が検出される限り、壁に沿って走行できることが確かめられた。例として、静止している1辺75cmの4角形の壁に沿って走行させた場合、 $D=50\text{cm}$ 、 $d=3\text{cm}$  として直径75cmの円形の壁に沿って  $30\text{cm/sec}$

の速度で走行させられることが実機実験の結果示された。また、壁がゆっくり動いている場合に、上記のアルゴリズムによって壁沿い走行させた結果を示した。なお、この実験ではゆっくり走行する物体は他のロボットを用いて実現した。ロボットを秒速 30cm 程度で走行させる場合、対象物速度が秒速 5cm 程度であれば、静止している場合と同様に壁沿い走行をすることができた。

一方、本アルゴリズムでノコギリ状の壁に沿うようにウォールフォロイングを行なわせると、衝突する場合と衝突しない場合があることがわかった。その理由は超音波の対象物への鏡面反射特性により、反射波が得られず壁の存在が検出できないためである。

## ソナーリングによる壁沿い走行制御と位置制御に基づく移動ロボットのナビゲーション

この研究は、位置（座標値）ベースな走行制御と、前項で記したセンサベースな壁づたい走行を組み合わせることによって自律移動ロボットのナビゲーションを実現する方法を検討・実現したものである。この種の動作の実現のために自律移動ロボットのための走行制御コマンドシステムを提案し、それを用いて実現したナビゲーションの実験例を示した。

はじめにも述べたとおり、移動ロボットが通るべき経路と自己位置推定のためのランドマークの位置を記した正確な地図を移動ロボットにあらかじめ与えれば、移動ロボットの位置を地図上の座標値により表現でき、その座標値をベースに決められた経路を走行させることで、信頼性の高いナビゲーションが実現される。このような走行をここでは「位置に基づく走行制御」と呼ぶ。

一方前項に記したように、ソナーリングとそれを用いて壁に沿って走行できるアルゴリズムを実験用自律ロボット「山彦」に搭載した結果、移動ロボットは連なっている壁に沿って走行することができる。したがって、ロボットに壁づたい走行をさせれば目的地に到達できる環境では、精密な地図を予め与えることなしに目的地まで到達させることもできる。このように走行させることを「壁沿い走行制御」と呼ぶ。

今、移動ロボットに自律的に実環境内をナビゲーションさせる場合を考えると、環境が広々としていて直線あるいは単純な曲線で走行すべき経路が表現できる場合は、

位置ベースで走行制御をするのが良い。しかし、例えば比較的狭くかつ曲がりくねった道に沿って走行させたい場合には、センサに基づいて道路に沿って走行する方が容易である。一方、予め環境の地図を有して「位置に基づく走行制御」を行なっても、走行中に予想外の障害物が発見された場合などには、その障害物を回避するにはその障害物のまわりづたいにセンサ情報に基づいて走行せねばならない。

これらのケースを考えると実環境内のナビゲーションではこれらの2つの制御方式をそれぞれの長所に応じた環境でうまく使い分け、区間や場合によって制御方式を切り換えて全体の動作を実現するのが有効である。この研究成果は、従来からよく研究されてきた問題設定である詳細環境モデルと自己位置推定による位置に基づく走行と、本論文における詳細環境モデルを与えず距離センサの情報に基づく壁沿い走行の両者の長所を生かすかたちで併せて適用することの提案と実験結果となっている。

ここで、提案した走行制御方式の切り換えをナビゲーションレベルのプログラミングのレベルで実現できるようにするために、「壁沿い走行コマンド」を定義し、実験用ロボットシステム「山彦」上に実装した。筆者らが有する自律型移動ロボットでは従来から Spur と呼ぶ走行制御コマンド体系が定義されている。これは、主に「位置に基づく走行制御」の実現のためのコマンド体系であり、具体的な走行軌跡経路が、座標系上の直線または円弧で指定できる。筆者は、このコマンド体系を拡張し、これに壁沿い走行コマンドを追加した。壁沿い走行コマンドが発行されると、ロボットは、ロボット中心から壁表面までの目標距離だけ離れて壁沿い走行する。

本コマンドシステムを用いれば、経路の途中に通り道がジグザグである場合には壁沿いを走行するという指令をロボットに与えるだけで複雑な経路の指定を避けることができる。また、移動ロボットがナビゲーションをしている時に、移動ロボットの進路軌道上に予期せず障害物などが置かれた場合、一時的に移動ロボットに壁沿い走行をさせて障害物回避をさせることができる。ここではその動作のプログラム例と実験結果を示し、その有用性を明らかにした。

## 超広角カメラと扇状光を用いた移動ロボット用レーザレンジセンサの開発

これまでに述べた研究においては、円環状に超音波距離センサを配置したソナーリングを用いた。超音波距離センサは、比較的短時間に距離を得る事ができる長所を持つ。しかし、超音波の鏡面反射の性質により、超音波の指向角の範囲に対象物があっても対象物の表面形状によっては反射波が戻って来ないため対象物の存在とその距離が検出されないことがある。また、一般に超音波トランスデューサの指向角は広いので距離データの方向分解能が悪いという性質がある。この性質によってすでに述べたとおりソナーリングの情報による壁沿い走行では、その壁の形状によってはうまくゆかないことがあった。したがって、対象物が存在すればほぼ確実に検出ができ、また方向分解能の良いセンサが必要であると結論された。このような要求を満足できる可能性のあるセンサとして、筆者は床面に平行な平面内にレーザ光線による扇状の光を投射し、その光が対象物にあたった部分を TV カメラによって検出して三角測量法によって距離を得る光アクティブ式距離センサの方式を採用し、実験用ロボット「山彦」上で使えるシステムを開発した。

移動ロボットのナビゲーションでは、障害物を回避するために前方の環境を調べるためのセンサ、さらには左右の壁に沿って走行するために左右の環境を調べるためのセンサが必要になる。移動ロボットのナビゲーションには前方および左右をカバーする広範囲の環境を調べる必要がある。そこで、移動ロボットの頂部に3個の超広角 CCD カメラ (視野角:1 カメラにつき  $108^\circ$ )、ロボットの腰部に5個の赤色レーザ扇状光投光器 (波長:685nm) を搭載し、視野範囲をカバーした。1個のカメラ毎に1個の画像処理ボード (CPU: T805) を用意した。3個の超広角 CCD カメラは、ロボットの上部高さ 70 cm の位置に、前と左右方向に向けて下向きふ角約  $40^\circ$  で取り付け、それぞれのレンズには、680nm ~ 2600nm の波長の光を通す光学フィルタを取り付けた。扇状光レーザ投光器は、1つで約  $90^\circ$  に広がるスリット光投光器5個で構成され、それぞれの投光器は床面に平行にロボットの前方の周囲約  $260^\circ$  を扇状光がカバーするように取り付けた。

対象物までの距離を計算するためには、床面と平行に広がった扇状光を測距対象物に当て、その反射光をカメラで撮像し、得られた画像上の輝度の高いピクセルの位置



からレーザー平面上での対象物の光の反射位置に変換する。

カメラがピンホールモデルに従う場合この変換は単純な幾何学的な考察により求まるが、ここでは超広角カメラを用いたため、この超広角レンズにより生じる歪みを補正する必要があった。また、ロッカーなどの金属性の光沢のある壁がある場合や、床面がピカピカに磨かれている場合などに扇状光が多重反射するため、物体が無いところに、あたかも物があるかのように、移動ロボットが距離を計算してしまう問題が見つかった。この問題を回避するためにもその画像の走査にも工夫が必要であった。

本研究で開発されたこの距離センサの測距可能範囲は、ロボットの周囲  $260^\circ$  (前方から左右  $\pm 130^\circ$ ) 程度の範囲であり、測距精度は、キャリブレーションの程度に大きく左右されるが、画像上で1ピクセルずれた時、約1mの距離を測定する場合およそ  $\pm 5$  cm、約2mの距離を測定する場合およそ  $\pm 10$  cm である。環境にもよるが、対象物が白色に近ければ、測定可能範囲はおよそ3~4mである。処理時間は、現在のところ  $0^\circ$  から  $180^\circ$  を  $1^\circ$  毎に測定するのにおよそ600msecかかる。

### レーザーレンジセンサによる廊下長手方向走行

前項で述べたように、ロボットの前方から左右に  $\pm 130^\circ$ 、およびおよそ3m以内の範囲においてロボットを取り巻く壁や障害物の様子を同時に知ることができる。もし、直線状に距離値が並んでいる部分があれば線分をあてはめることができる。この線分が長ければ、これは壁である可能性が高く、これを用いればロボットが廊下の長手方向すなわち、廊下の壁に沿う自律的な走行ができる。本研究はこのためのアルゴリズムを考案し、実験システム上に実装し実機実験を行なったことである。ここで想定した走行環境としての廊下の幅は、一般的な建物の廊下の幅の1.5~3m程度であるとする。また障害物がもしあるとすれば、四角柱のように幾何学的に簡単な形状のものとして仮定した。このアルゴリズムの大略は次の通りである。

このアルゴリズムでは、まず得られた対象物への距離データに以下のような方法によりおおまかに線分をあてはめることにより廊下の長手方向を決定し、その長手方向に沿いつつ、次の移動目標となるサブゴールを決定する。

1. (点列の分割) それぞれ隣り合う点の間の距離がしきい値  $T_1$  以下だった場合同じ点列の集合とし、あるしきい値を越えたら、そこで点列の集合を分割する。

2. (慣性モーメントを最小にする線分あてはめ) 点列の集合に対し、その重心を計算し、重心を通り各点列と線分との距離の自乗が最小になる傾きの線分を決定する
3. (線分の分割) あてはめられた線分とそれぞれの点列の間の距離の最大値があるしきい値を越えた場合、その最大値の点で線分を分割し、折れ線にする。この処理を再帰的に行う。

廊下などに箱などの物体が置かれていた場合、移動ロボットは、その物体に衝突しないように走行する必要がある。そのためサブゴールの設定と直線追従走行によって衝突回避を行う。具体的には、得られた距離データに線分をあてはめ、その各線分に対して以下のように処理を行う。

1. 各線分を移動ロボットの半分の幅だけ拡大した長方形をつくり、各線分と移動ロボットの中心から廊下の長手方向に平行に引いた半直線との交差チェックを行い、もし、交差していれば次の処理を実行する。
2. ある線分の端点から他の線分までの距離を計算し、距離が移動ロボットの幅よりも大きいすき間を探す。
3. 先に求めたすき間のうち、交差チェックした半直線に近い側の線分を拡大した長方形の頂点をサブゴールとし、移動ロボットを直線追従走行させる。

上述の廊下長手沿い走行アルゴリズムを実験用ロボットシステム「山彦」上に実装し実験を行った。箱などの障害物がある廊下の環境において衝突することなく廊下の壁に沿って走行することを確認した。

## 言語的な行動シーケンスの指示によるナビゲーション

この研究は、数百メートル四方程度の範囲の中に広がりをもつ建物内の廊下環境 (例えば、筑波大学第三学群 3 階、4 階の廊下環境) における自律移動ロボットのナビゲーションを目的とする。この場合、環境を未知として探索を行うというよりはむしろ移動ロボットに対して何らかの命令を与えて移動の開始点から目的地まで到達できればよいと考えられる。しかし、対象とする環境が広大なため、走行する可能性のある環

境を調べあげて地図を作成することは大変な労力を要する。しかし、「2つめの交差点を右に曲がり、1つめの交差点を左に曲がり、2つめの交差点まで進め。」というような、比較的抽象度の高い命令群により移動ロボットのナビゲーションが実現されれば、このような問題は解決される。本研究は、このような命令群による言語的指示を与えることで移動ロボットのナビゲーションを実現することを目標とした。実現方法としては、(A) 壁沿い走行により廊下の長手方向を走行、(B) 壁の有無の検出により交差点の候補を認識、の2つの機能を移動ロボット上に実現し、これらを組み合わせてナビゲーションを行うこととした。

ここで、環境やロボットの前提条件を以下のようにした。想定している環境は、数百メートル四方の建物内の廊下環境であり、廊下と廊下はほぼ直交し、廊下の幅は1.5～3m程度とする。移動ロボットは、方位を常に推定し、前方から±130°、中心から2m以内の物体までの距離を1°毎に計測できるセンサを装備し、その2次元形状は円形、そして、その初期方位は廊下長手方向、その初期位置は廊下の両壁のほぼ中間であるとする。

この条件の下で、ロボットに行動を言語的指示で与えるための一群の記述をいくつか組み合わせる経路指示言語のシンタクスを定義した。

Keep_on <u>Job</u> . When <u>Condition</u> [and <u>Condition...</u> ] [at <u>Number</u> ], Exec <u>Action</u> .
---

- Job に与えられる命令
  - going\_along\_corridor : 廊下長手方向に沿って走行し続けよ
- Condition に与えられる条件の種類
  - no\_right\_wall : 右壁がない
  - right\_wall : 右壁がある
  - no\_left\_wall : 左壁がない

- ・ left\_wall : 左壁がある
  - ・ no\_front\_wall : 前方に壁がない
  - ・ front\_wall : 前方に壁がある
  - ・ for Dist m : 直前の Job で走行し始めた点から、およそ 距離 Dist m 走行した場所
- Number に与えられる条件の種類
    - ・ 1st: 1 番目の (default)
    - ・ 2nd: 2 番目の
    - ・ 3rd: 3 番目の
  - Action に与えられる命令の種類
    - ・ stop : 停止せよ
    - ・ turn\_right : 右に 90° 曲がり直進せよ。
    - ・ turn\_left : 左に 90° 曲がり直進せよ。
    - ・ go\_straight : 直進せよ。

この指示を実現するための、ロボットによる壁の検出方法は次の通りである。ロボットは一旦停止し、周囲の対象物の距離情報の取得を行なう。この一回のセンシングにより移動ロボットの左右の距離データから、移動ロボットの左右の空間に距離点列があるかを調べ、測距可能範囲に距離点列がなければ壁の無い通れる場所 (交差点) とし、あれば壁のある場所 (交差点でない) とする。

そして、前項で述べた廊下長手方向走行の手法を用いて廊下の長手方向に進みながら、このような壁の有無の検出を行なうことにより本ナビゲーションを実現する。

右折、左折を含むような言語的指示シーケンスの例を用いて、筑波大学第三学群の廊下で、移動ロボットにナビゲーションを行なわせ、移動ロボットがほぼ期待通りに走行することを確認した。

## 結論

本研究では、以下のことを行なった。(1) ソナーリングによる壁沿い走行アルゴリズムを提案し、実験によりその有効性を検証した。ここでは、未知の環境を前提とし、超音波距離センサの情報に対しある決められた行動をとることで壁沿い走行を実現した。壁の形状が円柱形や四角柱形の場合には、安定して壁沿い走行することがわかった。しかし、壁がジグザグな形状の場合には、壁に衝突することがあった。これは、超音波が鏡面反射の性質を有しているため、壁の形状によってはその反射波が返らず、壁を検出できないためであると考えられる。ソナーリングによる壁沿い走行の有効性とその限界を考察したことは、意義があると考えられる。

(2) ソナーリングによる壁沿い走行とオドメトリに基づく走行を切り換えることにより、移動ロボットにナビゲーションさせる手法を提案し、その有効性を実験により検証した。具体的には、オドメトリに基づく直線追従や円弧追従などの走行や、ソナーリングからの距離情報に基づく壁沿い走行を一つの関数として実行できる走行コマンドシステムを開発した。この走行コマンドシステムを用いて、一部詳細な環境モデルが無い環境において、詳細な環境モデルが無い場所では壁沿い走行をし、その他の場所ではオドメトリに基づく走行をすることにより、移動ロボットにナビゲーションをさせる実験を行ない、移動ロボットがほぼ期待通り走行することを確認した。これにより、一部詳細な環境モデルが無い場合においても、移動ロボットをナビゲーションさせることが可能になった。

(3) 超広角カメラと扇状光を用いた移動ロボット用レーザレンジセンサの開発を行なった。我々は、前項までに示したように、以前ソナーリングを用いて様々な研究を行ってきた。しかしソナーリングは、距離を計算するまでの処理時間が短いという長所はあるが、方向分解能が低く、超音波のもつ鏡面反射の性質により対象物の表面形状によっては指向範囲にある対象物が検出できないという問題があった。そこで、そのような問題がほとんどなく、かつ、移動ロボットの周囲を広くカバーする距離センサによりナビゲーションの研究を行なうため、このレーザレンジセンサの開発を行なった。そして、廊下などの環境ではほぼ期待通りの距離値が得られることを確認した。回転型のレーザレンジセンサは既に世の中にあるが、移動ロボットが走行中に計測すると距離値がずれてしまうという問題がある。同時に距離を測定でき、かつ、広い角

度にわたって距離を測定でき、かつ、コンパクトな移動ロボット用レーザレンジセンサは今まで見たことがなかった。このセンサの開発方法やハード構成は、移動ロボット用センサの今後の研究開発の参考になると考えられる。

(4) レーザレンジセンサによる廊下長手方向走行アルゴリズムを提案し、実験によりその有効性を検証した。実環境の廊下は、壁際にゴミ箱やロッカーなどが置かれていたり、壁の形状に凹凸があったりして、単純な壁沿い走行アルゴリズムでは、壁などに衝突することがある。本研究では、前項で示した距離センサの使用を前提に、壁への衝突回避、廊下長手方向の決定などの方法を組み合わせて移動ロボットを廊下の長手方向に沿って走行させるアルゴリズムを提案した。また、実環境の廊下において走行実験を行ない、移動ロボットがほぼ期待通り走行することを確認した。これによりトイワールドではなく、実環境において距離センサ情報により移動ロボットをロバストに廊下長手方向に沿って走行させることが可能になった。

(5) 言語的な行動シーケンスの指示によるナビゲーション方法を提案し、実験によりその有効性を検証した。具体的には、「Keep\_on going\_along\_corridor. When no\_right\_wall, Exec turn\_right. (廊下の長手方向に沿って走行し、右壁が無いところを右折しなさい)」というような、言語的な指示シーケンスにより移動ロボットをナビゲーションさせる手法を提案した。また、筑波大学第三学群の廊下環境において実機実験を行ない、移動ロボットがほぼ期待通りに走行することを確認した。これにより、詳細な環境モデルを用いずに、言語的な指示シーケンスだけを用いて、距離センサ情報に基づいた移動ロボットの広範囲な廊下環境におけるナビゲーションが可能になった。

上記に示した(1),(2),(3),(4),(5)から、本研究は、移動ロボットの詳細な環境モデルを用いない移動ロボットの距離センサ情報に基づく屋内ナビゲーションに関する研究に貢献をした。

# 目次

論文概要 . . . . .	i
目次 . . . . .	xiii
図目次 . . . . .	xvi
表目次 . . . . .	xix
<b>1 序論</b> . . . . .	<b>1</b>
1.1 本研究の背景と目的 . . . . .	1
1.2 実現すべき手段 . . . . .	7
1.3 本論文の構成 . . . . .	8
<b>2 従来の研究</b> . . . . .	<b>9</b>
2.1 はじめに . . . . .	9
2.2 詳細な環境モデルを用いるナビゲーションに関する研究 . . . . .	9
2.3 詳細な環境モデルを用いないナビゲーションに関する研究 . . . . .	10
2.4 レーザレンジセンサに関する研究 . . . . .	12
2.5 2章のまとめ . . . . .	13
<b>3 研究用プラットフォームとしての自律移動ロボット「山彦」</b> . . . . .	<b>14</b>
3.1 機能分散アーキテクチャと行動プログラム . . . . .	14
3.2 走行制御系 . . . . .	18
3.3 位置推定系コマンドと位置推定モジュール . . . . .	20
<b>4 ソナーリングによる壁沿い走行</b> . . . . .	<b>21</b>
4.1 はじめに . . . . .	21
4.2 ソナーリング . . . . .	22

4.3	壁沿い走行手法	22
4.3.1	超音波の鏡面反射特性と仮想壁	22
4.3.2	壁沿い走行アルゴリズム	26
4.4	ソナーリングによる壁沿い走行実験	32
4.4.1	実験 1(四角柱、円柱、台形の内側、動く円柱)	32
4.4.2	実験 2(大小の星型対象物、大小の“*”型対象物)	36
4.4.3	壁沿い走行の限界の検討 1(四角い対象物が動く場合)	39
4.4.4	壁沿い走行の限界の検討 2(ノコギリ状の凹凸のある壁)	42
4.5	4章のまとめ	44
<b>5</b>	<b>ソナーリングによる壁沿い走行制御と位置制御に基づく移動ロボットのナビゲーション</b>	<b>49</b>
5.1	はじめに	49
5.2	Spur コマンド系の拡張(壁沿い走行コマンドの追加)	50
5.3	壁沿い走行コマンドの使用例	51
5.3.1	蛇行経路	52
5.3.2	障害物回避	54
5.4	壁沿い走行から位置ベース走行に切り換える方法	56
5.5	ナビゲーションの実験例	57
5.6	5章のまとめ	63
<b>6</b>	<b>超広角カメラと扇状光を用いた移動ロボット用レーザレンジセンサの開発</b>	<b>64</b>
6.1	はじめに	64
6.2	レーザレンジセンサのハードウェア	65
6.3	距離計算の原理	66
6.4	超広角カメラ画像の歪み補正	70
6.5	画像の走査方法(扇型走査)	71
6.6	実機(山彦)による環境の測定例	72
6.7	6章のまとめ	78
<b>7</b>	<b>レーザレンジセンサによる廊下長手方向走行</b>	<b>81</b>



---

7.1	はじめに . . . . .	81
7.2	壁沿いに廊下長手方向へ走行する手法 . . . . .	83
7.2.1	距離点列への線分あてはめ . . . . .	83
7.2.2	走行アルゴリズムの概要 . . . . .	89
7.3	実機実験 . . . . .	91
7.4	7章のまとめ . . . . .	95
<b>8</b>	<b>言語的な行動シーケンスの指示によるナビゲーション</b>	<b>98</b>
8.1	はじめに . . . . .	98
8.2	行動シーケンスによってナビゲーション経路を指示する言語の仕様 . . .	100
8.3	ナビゲーション経路を指示する言語の実装とナビゲーションの実現 . . .	105
8.4	言語的指示による実環境におけるナビゲーションの実験 . . . . .	108
8.5	8章のまとめ . . . . .	108
<b>9</b>	<b>まとめと結論</b>	<b>111</b>
9.1	各テーマの総括 . . . . .	112
9.2	本研究の産業や社会への応用について . . . . .	115
9.3	結論 . . . . .	116
	謝辞 . . . . .	117
	参考文献 . . . . .	120
	研究業績 . . . . .	127

## 目次

1.1	位置ベース走行の例 . . . . .	2
1.2	直線追従による移動ロボットの経路情報 . . . . .	3
1.3	車輪のスリップなどによる衝突 . . . . .	3
1.4	移動ロボットの位置や方位を修正する情報 . . . . .	5
1.5	詳細な環境モデル (経路情報+位置や方位を修正する情報) . . . . .	5
3.1	LR-1 タイプ「山彦」 . . . . .	15
3.2	「山彦」の機能分散アーキテクチャ . . . . .	17
3.3	GL, LC, FS 座標系 . . . . .	19
4.1	ソナーリングの概観 . . . . .	23
4.2	ソナーリングを搭載した移動ロボットの概観 . . . . .	24
4.3	超音波の鏡面反射特性と仮想壁 . . . . .	25
4.4	Case1(正面または、右斜めに壁が検出された場合) . . . . .	27
4.5	Case2(左斜めに壁が検出された場合) . . . . .	28
4.6	Case3(左横に壁が検出された場合) . . . . .	28
4.7	Case4(壁が検出できない場合) . . . . .	29
4.8	四角柱の壁に沿う実験 (実機実験) . . . . .	33
4.9	円柱の壁に沿う実験 (実機実験) . . . . .	34
4.10	台形の壁の内側の壁に沿う実験 (実機実験) . . . . .	35
4.11	ゆっくり動く円柱の壁に沿う実験 (実機実験) . . . . .	36
4.12	大きい星型対象物の外壁に沿って走行させた場合 (実機実験) . . . . .	37
4.13	小さい星型対象物の外壁に沿って走行させた場合 (実機実験) . . . . .	38
4.14	大きい“*”型対象物の外壁に沿って走行させた場合 (実機実験) . . . . .	40

4.15	小さい“*”型対象物の外壁に沿って走行させた場合 (実機実験)	41
4.16	ゆっくり動く四角柱の壁に沿う実験 (実機実験)	42
4.17	ノコギリ状の凹凸のある壁に沿って走行させる場合の実験環境	45
4.18	zigzag shape (L=45 [cm])	46
4.19	zigzag shape (L=90cm)	47
4.20	衝突の検討	48
5.1	壁沿い走行コマンドのモデル	51
5.2	蛇行経路走行 (実機実験)	52
5.3	蛇行経路走行 (行動プログラムの抜粋)	54
5.4	障害物回避 (実機実験)	54
5.5	障害物回避 (行動プログラムの抜粋)	56
5.6	壁沿い走行から位置ベース走行への切り換え方法	58
5.7	フラットな壁を壁沿い走行させた時の軌跡	59
5.8	ナビゲーション実験 (環境)	60
5.9	ナビゲーション実験 (プログラムの状態遷移)	61
5.10	ナビゲーション実験 (移動ロボットの推定自己位置の誤差楕円)	62
5.11	ナビゲーション実験 (実験風景)	63
6.1	センサの概観図	66
6.2	レーザ取付位置 (平面図)	67
6.3	画像処理モジュールの内部構成	68
6.4	センサが実装された移動ロボットの写真	69
6.5	レーザ平面とピンホールモデル画像の幾何関係	70
6.6	超広角カメラで市松模様を撮像した場合の例	73
6.7	光沢のある壁の配置	73
6.8	前方の超広角カメラ画像	74
6.9	画像の走査方法	74
6.10	画像の走査方法と距離計算処理	75
6.11	画像全体の走査により得られた対象物表面のロボット上に固定された座標系からみた座標位置	76

6.12	前方の超広角カメラ画像 (黒線：右斜め 60° 方向のスキャンライン) . . .	76
6.13	右斜め 60° 方向のスキャンライン上のピクセル輝度値 . . . . .	77
6.14	扇型スキャン方法による距離測定結果 . . . . .	77
6.15	実験環境 (廊下) . . . . .	78
6.16	得られた 3 方向の超広角カメラ画像 . . . . .	79
6.17	得られた距離データ . . . . .	79
7.1	環境 (筑波大学 第三学群) . . . . .	82
7.2	箱などの障害物のある廊下環境 . . . . .	82
7.3	線分あてはめ (点列の分割) . . . . .	84
7.4	線分あてはめ (再帰的な線分の決定) . . . . .	84
7.5	測距環境 . . . . .	85
7.6	実環境測距データ . . . . .	86
7.7	実環境測距データの分割結果 . . . . .	87
7.8	実環境測距データへの線分あてはめ結果 . . . . .	88
7.9	距離測定と線分あてはめ (Step1)、直線 A の決定 (Step2)、および、 サブゴール P の仮決定 (Step3) . . . . .	91
7.10	衝突チェック (Step4) . . . . .	92
7.11	通り抜けられる隙間探し (Step5) . . . . .	93
7.12	サブゴール P の再設定 (Step5)、および、サブゴール P まで直線追従 (Step6) . . . . .	94
7.13	実験環境と走行経路 . . . . .	95
7.14	実験時のオドメトリによるロボットの位置 (・印) . . . . .	95
7.15	実験時のスナップショット . . . . .	96
8.1	想定環境 (筑波大学 第三学群) . . . . .	99
8.2	言語実装 . . . . .	106
8.3	壁の有無検出 . . . . .	107
8.4	実験結果 (軌跡) . . . . .	108
8.5	言語的指示例 . . . . .	109
8.6	実験 1(目的地：研究室 B) のスナップショット . . . . .	109

## 表目次

3.1	走行制御系の主要な関数 (Spur 関数) . . . . .	19
3.2	位置推定系の主要な関数 (POEM-II 関数) . . . . .	20
4.1	Result of experiments . . . . .	43
8.1	言語的な経路指示方法の具体例 (その 1) . . . . .	102
8.2	言語的な経路指示方法の具体例 (その 2) . . . . .	103
8.3	言語的な経路指示方法の具体例 (その 3) . . . . .	104

# 第 1 章

## 序論

### 1.1 本研究の背景と目的

本研究は、自律移動ロボットの屋内環境における自律走行とナビゲーションに関するものである。すなわち、建物内の環境のうち、階段などの立体的な段差を登り降りするような環境を除く、ほぼ平らな床の上において、車輪を有し自律的に移動することが可能なロボットを、環境内の壁やそこにおかれた物体(対象物)のまわりに沿って走行させたり、環境中でスタート地点やゴール地点を与えて移動ロボットを自動的に走行させることを取り扱っている。

移動ロボットをスタートからゴールまでその通過すべき経路を誤ることなく、また環境中の壁や対象物に衝突することなく移動させることをナビゲーションと呼ぶ。この移動ロボットによるナビゲーションのためには、従来、環境に関する詳細な地図を作ってこれをロボットに与え、またロボット自身の位置をできるだけ正確に常時推定し、それに従って目的地まで走行するというナビゲーション手法をとることが一般的であった。この手法をもう少し詳しく説明する。

[詳細な環境モデルを用いるナビゲーション]

いま図 1.1 のような環境で、スタートからゴールまで移動ロボットをナビゲーションさせることを考える。

車輪で移動するロボットは車輪の回転数の累積によって自己の位置と方位を容易に推定できる(この推定機構を本論文では、オドメトリ<sup>1</sup>と呼ぶ)。いま、移動ロボット

---

<sup>1</sup>これをデッドレコニングと呼ぶ習慣も存在するが、本来デッドレコニングとは盲目航法のことを指

を走行させようとする環境において、ある座標系を決め(図 1.2)、スタート地点とゴール地点のロボットの位置(図 1.2中の  $(x_0, y_0)$  および  $(x_3, y_3)$ )と方位を決める。あらかじめ与えられた詳細な環境モデルによってロボットが安全に通過できる領域にスタート地点からゴール地点までの通過経路を決め、走行する方向を変えるべき地点(図 1.2中の  $(x_1, y_1)$ ,  $(x_2, y_2)$ )とその地点での方位の変化(図 1.2中の  $\theta_1$ ,  $\theta_2$ )などのパラメータを与える。これらの地点を線分で結び、オドメトリによるロボット自身の推定位置が常に直線上にのっているように走行制御を行ない(直線経路追従)、また、走行方向を変えるべき位置  $(x_1, y_1)$ ,  $(x_2, y_2)$  において、角度  $\theta_1$ ,  $\theta_2$  回転することを行なえばゴール位置  $(x_3, y_3)$  に到達できるはずである。(このような位置に基づく走行を本論文では「位置ベース走行」と呼ぶ。)

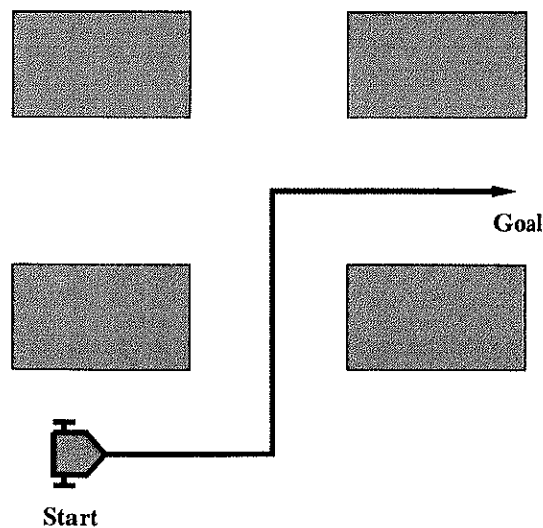


図 1.1: 位置ベース走行の例

しかし、移動ロボットの初期方位が少しずれていたり、オドメトリの推定位置の車輪のスリップなどによる誤差累積により、実際には図 1.3のように壁などに衝突する場合がある。そのため、通常はロボットと壁や対象物までの距離を測定する距離センサ、内界センサによって自己位置と方位を推定する方式の総称でもある。ゆえに航空機などで使われている INS(慣性航法装置) や、人間などが歩幅を数える位置・方位の推定方法もデッドレコニングの範疇に属する。本論文では車輪の回転数の累積によるデッドレコニングを特に示すため、オドメトリという語を用いる。

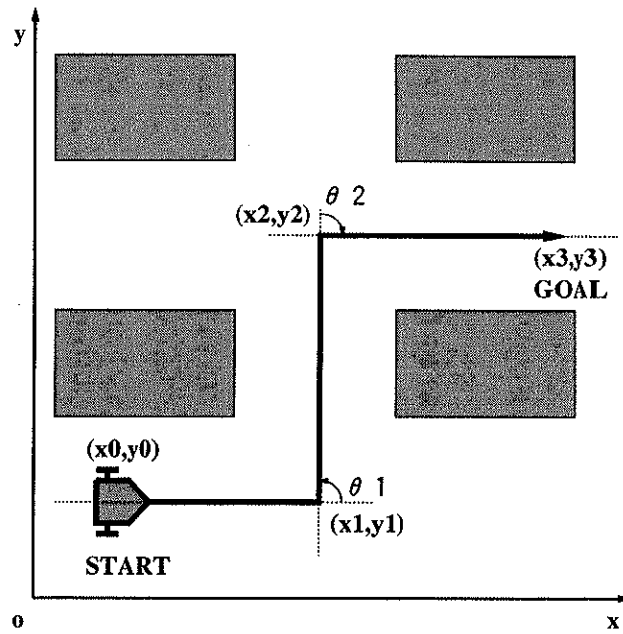


図 1.2: 直線追従による移動ロボットの経路情報

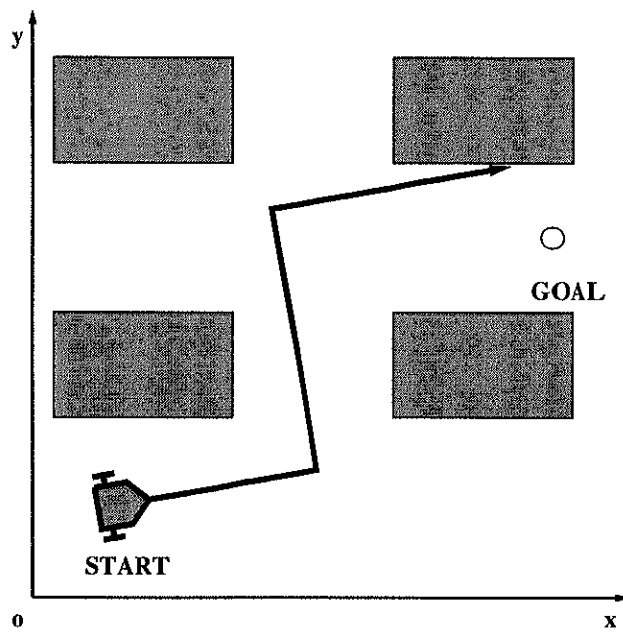


図 1.3: 車輪のスリップなどによる衝突



サを用いて、オドメトリにより推定された移動ロボットの位置や方位に含まれる誤差を修正する。例えば図 1.4 のように、位置  $(x_4, y_4)$ ,  $(x_5, y_5)$ ,  $(x_6, y_6)$ ,  $(x_7, y_7)$  において、その位置における壁までの距離  $L_4, L_5, L_6, L_7$  を付加的に与える。オドメトリにより移動ロボットが  $(x_4, y_4)$  の位置に到達したとみなした時、距離センサにより計測された壁までの距離が  $L_4$  に等しくないことがある。これは、オドメトリによる自己位置に誤差があるため、実際には  $(x_4, y_4)$  からその誤差分だけ離れた位置にあるためである。そこで、実際の壁までの計測値をもとに、実際の移動ロボットの位置を推定し直すことでオドメトリによる推定位置の誤差が修正される。また、このような走行経路に関する情報や位置誤差の推定のための付加的な情報は、環境全体の壁や対象物の配置に関する詳細な地図を持っていれば、移動ロボットのスタート地点とゴール地点の位置を与えることにより自動的に生成することもできる。このような地図、および、図 1.5 に示すような、移動ロボットの走行経路情報と、移動ロボットの推定位置や方位を修正する情報とを合わせて、本論文では詳細な環境モデルと呼ぶ。多くの場合、このような詳細な環境モデルの構築は、実際の走行環境を人手で巻尺を用いて測定する、あるいは建物の図面を入手してこれを計算機上に入力するなどの手間を要していた。移動ロボットを走行させてこのモデルを自律的に取得させることも考えられるが、実用上利用しうる例はまだほとんどない。

#### [詳細な環境モデルを用いないナビゲーション]

上で述べた詳細な環境モデルを用いる位置ベース走行は、いわば “cheap” なセンサ系によるナビゲーションということができる。すなわち、車軸またはモータ軸に取り付けることのできるパルスエンコーダにより容易に実現できるオドメトリと、ある特定の方向に向けて距離を測定する外界センサを用意するのみで、人間の視覚のように周囲の様子をくまなく目視するようなセンサを搭載しなくてもナビゲーションを実現している。逆に言えば移動ロボットのセンサ系を構築する負荷を軽減する代償として、十分に詳細な (“rich” な) 環境モデルを用意する必要があった。

一方、人が歩行して目的とする場所まで移動する場合、自己の位置を座標系の中のある位置座標として意識しているわけではなく、むしろ「この道に沿って進み、2 目の交差点で右折し・・・」などという意識で行動している。これは視覚や脳によるいわば “rich” な環境センス・認識能力によって、「この道に沿う」「この廊下づたいに歩く」「交差点をみつけてそこをまがる」などの行動が可能になっているといえ

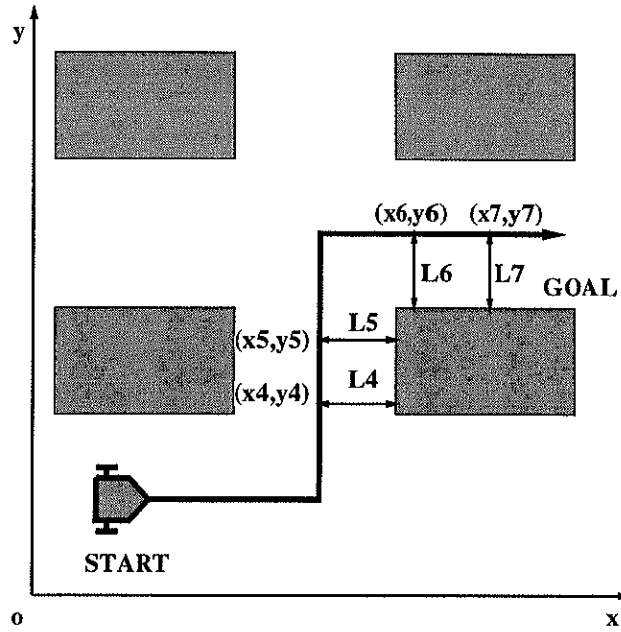


図 1.4: 移動ロボットの位置や方位を修正する情報

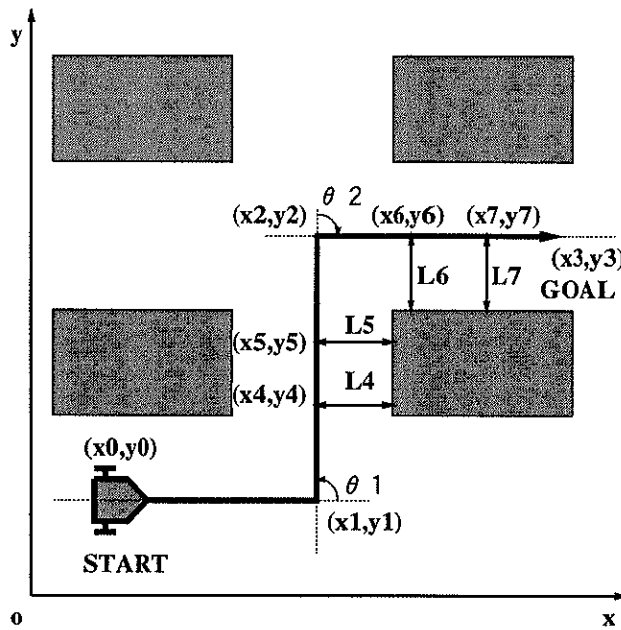


図 1.5: 詳細な環境モデル (経路情報+位置や方位を修正する情報)

る。「道に沿う」「廊下づたいに歩く」などの行動は道の境界や廊下の壁に近付いたら離れ、それらより離れすぎたら少しそれらの近くに寄る、というような行動を歩行(走行)中に短い時間間隔で繰り返すことによってなされる。これを移動ロボットの場合にあてはめれば、一定のサイクル(時間など)毎に、距離センサなどの外界のセンサ情報から、移動ロボットの行動を決定し、それに従い移動ロボットを走行させることになり、センサーベーストビヘービアによる行動の実現ということになる。

このようなセンサーベーストビヘービアを積極的に用いた壁沿い走行手法を実現すれば、建物内の廊下の壁に沿って廊下の長手方向を走行することができ、詳細な環境モデルを用いない移動ロボットのナビゲーションを実現することも可能であろう。そうすれば、必ずしも走行環境に関する詳細な環境モデルを必要とせず、特に移動ロボットを走行させる環境が広範囲に及ぶ場合、その詳細な環境モデルを構築する手間が著しく減るメリットがある。

このようなセンサーベーストビヘービアによる行動を組み合わせたナビゲーション手法を実現するためには、その位置におけるロボット周囲の環境情報を十分に得るセンサ系が必要であり、この情報に基づいて、例えば壁に衝突しないように壁沿い走行できる走行手法を実現することが必要になる。

本研究の目的は、以上に述べたように、詳細な環境モデルを用いずに、十分な環境の情報の得られるセンサを用意し、センサーベーストビヘービアによって屋内ナビゲーションを行なうことであり、上記のナビゲーションを実現できる移動ロボットシステムを構築し、開発することである。

本研究においては、移動ロボット周囲の環境情報を得るためのセンサとしては、(a) 12 個の超音波距離センサを円環上に配置したソナーリング、(b) レーザ光を扇状の平面に拡散させ、その光が対象物にあたった光切断面を超広角 TV カメラでとらえて三角測量を行なうレーザレンジセンサを利用した。

ソナーリングは、筆者らの研究室で用いられている超音波センサを利用して円環状に配置し、またレーザレンジセンサは、本研究のために開発したものである。

本論文で取り扱った具体的なテーマは 5 つである。

まず、ソナーリングを用いて、

#### 1. ソナーリングを用いた不定型な壁に沿う走行(第 4 章)

2. 位置ベースなナビゲーションと上記 1. の壁に沿う走行を組み合わせた屋内ナビゲーション (第 5 章)

を取り扱った。

次に、レーザレンジセンサを用いて、

3. 超広角カメラと扇状光による移動ロボット用レーザレンジセンサの開発 (第 6 章)
4. 上記 3. のセンサを用いた屋内の廊下の壁沿い走行 (第 7 章)
5. 上記 4. を用いた言語的指示による屋内ナビゲーション (第 8 章)

を取り扱った。これらの各テーマは、それぞれ独立した 5 つの研究テーマであるとも考えられるが、いずれも詳細な環境モデルを用いない移動ロボットの距離センサ情報に基づく屋内ナビゲーションに関する研究テーマである。

本研究においては、ロボットに必要な機能を装備し、環境にはできるだけ手を加えないで、ナビゲーションを実現することを目指した。移動ロボットの研究では、実際に移動ロボットを開発し、実際に動作させてみて、はじめて問題点が見つかることも多い。そのため、移動ロボットを実際に開発し、実地に動作させ、評価を行なう立場をとることは、ロボットの研究のひとつの方法論として重要で意義があると言える。

## 1.2 実現すべき手段

本研究を実現するにあたり、研究用プラットフォームとして筆者等の研究グループで用いられている移動ロボット「山彦」を使用することにした。この理由としては、(1)「山彦」が走行制御系やセンサ系を分散して開発しやすいアーキテクチャを有しているため、筆者が開発するセンサを実装しやすい、(2)「山彦」が優れた走行制御系を有し、そのライブラリがすでに実装され使用できる、(3) センサーベーストビヘーピアをプログラミングしやすい移動ロボットの行動記述言語 Robol/0 が用意され、それを使用することができる [52]、の 3 点を挙げるができる。このロボットの詳細については、第 3 章で述べる。

### 1.3 本論文の構成

本論文は、以下のように構成される。第 1 章は序論、第 2 章は従来の研究、第 3 章は研究用プラットフォームとしての自律移動ロボット「山彦」、第 4 章はソナーリングによる壁沿い走行、第 5 章はソナーリングによる壁沿い走行制御と位置制御に基づく移動ロボットのナビゲーション、第 6 章は超広角カメラと扇状光を用いた移動ロボット用レーザレンジセンサの開発、第 7 章はレーザレンジセンサによる廊下長手方向走行、そして、第 8 章は言語的な行動シーケンスの指示によるナビゲーションである。第 9 章は全体のまとめである。

## 第 2 章

### 従来の研究

#### 2.1 はじめに

本研究の目的は、移動ロボットにその周囲の環境の様子を十分に把握することのできる能力の高いセンサを搭載し、詳細な環境モデルを与えなくても環境内を走行できるような走行アルゴリズムを提案し、また、そのようなセンサとアルゴリズムを実際に移動ロボット上に実装することによって実験を行なうことをテーマとしている。本章ではこのテーマに関連する国内・国外の研究についてその概要を述べる。

#### 2.2 詳細な環境モデルを用いるナビゲーションに関する研究

第 1 章で述べたように、特に屋内環境における移動ロボットのナビゲーションの場合には、詳細な環境モデルを用いる方法が一般的であった。実際、Crowley や Cox らは、詳細な環境モデルを用いて移動ロボットの距離センサにより屋内ナビゲーションを実現している [1][2]。彼らの研究では、外界センサとして Crowley は超音波距離センサ、Cox はレーザレンジセンサを用いている。また、Lumelsky らは、病院の廊下における移動ロボットのナビゲーションを実現している [3]。これは、詳細な環境モデルを予め与え、モデル上で与えられたランドマークを実環境中で検出することを超音波距離センサあるいはレーザレンジセンサなどで行なうことにより、それが本来検出されるべき位置と、実際に検出された移動ロボットのオドメトリによる位置や方位との差を利用することで、より正確な位置・方位を求めることでオドメトリによる累積誤差を修正している。従来から、筆者らの研究室でもこのような詳細な環境モデルを

用いるナビゲーションの研究を行ってきた [4]。これらの移動ロボットのナビゲーションの研究は、移動ロボットに、その周囲をとりまく環境の様子、あるいは形状を知ることができるほどの強力な外界センサを用意せず、あらかじめ環境中に固定された座標系に関して決められた走行経路に沿って走行しているかどうかをオドメトリによる現在の推定位置・方位によって常時確認しながらナビゲーションを行なうものである。したがって、この推定位置に誤差が累積すれば、位置の誤認による環境中の物体への衝突もあり得るため、オドメトリの誤差をいかに減らすか、あるいは、自己位置をいかに推定するかの議論が不可欠である。このアプローチでは、情報処理に多大な労力を要する画像処理が必要な視覚センサを用いずともナビゲーションが可能であり、また仮に視覚センサを利用する場合でも、事前にモデルが与えられていることにより画像中の必要な特徴点や特徴物を検出する処理の負荷が軽減できることから、多くの研究でこの詳細な環境モデルを与えるアプローチがとられてきた。

このような詳細な環境モデルを用いるナビゲーション手法では、環境が広範囲になると、詳細な環境モデルを人手で作成するには手間がかかる。そのため、移動ロボットが自律的に環境中を移動して探索を行なうことで移動ロボットが環境中で行動するのに必要な環境モデルを人手を介さず作成する研究 [5] や、その探索をすべてロボットにまかせるのではなく、ある程度人手でそれを助けてやる、すなわち未探索部分を人がロボットを操縦して走行させ、その間にロボットが環境モデルを作つてゆく研究もある [6]。前者のようにロボットが完全に自律的・自動的に環境モデルを作成する研究の場合は、探索範囲を小部屋のようなところに限定せざるを得なかったり、移動ロボットが環境全部を探索するのに非常に多くの時間を費すことが多い。一方、後者のようにある程度人手を介するもの場合は、実際に移動ロボットを現場に持って行き、人が移動ロボットを操作する労力が必要になり、走行環境や走行すべき経路が頻繁にかわるような場合には人的な労力負担がやはり無視できない。

### 2.3 詳細な環境モデルを用いないナビゲーションに関する研究

例えば道路上を走行する自動車を自律化するという試みがある [7][8]。これらは、自動車の前方を監視する TV カメラによって車道の路肩に描かれた白線や境界を監視し、これらがそのカメラから見て常に一定の位置にあるように走行を制御するものである。このように、走行する通路をガイドにして走行できれば、詳細な地図を持たな

いナビゲーションが可能になる。[7]や[8]ではTVカメラの画像処理に専用のハードウェアを用い、自動車の走行方向を決定するステアリングの制御量をリアルタイムに画像処理結果より決定してフィードバックすることで制御を行なっている。通路沿い走行を視覚系を含めた制御により実現している。

一方、Brooksらはサブサンプリングアーキテクチャと呼ばれる、反射的な行動の組合せによる移動ロボットの行動の実現法を提唱し、注目された[9][10]。すなわち、ロボット上のセンサにより得られた情報の値や値の変化によって必要な行動が発現するようなルール、あるいはアーキテクチャを実装することで、センサ情報によって適切な行動が反射的に出てくるような構造をつくっておくものである。これはセンサ情報を制御のフィードバックに用いるのとはまた異なるものである。反射的な行動がうまく発現するようにルールの構造を作りこむことで、詳細な環境モデルを用いずにナビゲーションを行なうことも可能である。センサ情報に基づいて行動が発現されるので、これをセンサベーストビヘービアと呼ぶ。実際、森らは、TVカメラを用いて道路の縁石に沿う走行などのセンサベーストビヘービアを用いて移動ロボットの屋外ナビゲーションを実現している[11]。また、Lumelsky, Noborioらは、壁沿い走行を用いて未知環境における移動ロボットのナビゲーションを検討している[12][13]。すなわち、与えられた現在位置座標よりゴールの位置座標が与えられ、そこに到着するように進む。しかし環境は未知で、どのような障害物があるかはわからないという前提に立つものである。この場合、障害物があればその壁に沿い、障害物がなくなればその壁に沿う走行から離脱して、再びゴールの方向に走行しはじめる。というものである。しかし、提示された実験例はシミュレーションに終始して実機実験までは行なわれておらず、また、提案されたアルゴリズムのテストのために用いられた環境の複雑さが現実離れしている。また、センサベーストビヘービアによってロボットが障害物の壁に沿えることが前提になっているが、実際に移動ロボットを安定に壁に沿わせるにはどうしたらよいか、という技術的な課題には触れられていない。

このような壁沿い走行を実現する試みとしてTurenoutらは、超音波距離センサを用いた壁沿い走行の手法について検討をしている[15][16]。しかし、彼らが前提としている環境は、ある程度の長さの平らな壁が断続的に続く環境であり、不定型な形状の壁に関しては議論がされていない。

松本らは、ステレオ視を用いた移動ロボットの廊下環境での壁沿い走行アルゴリズム



ムを検討している [17]。これは簡単な形状の廊下環境を仮定しており、障害物などがあつたら、それを避けて走行することはできない。

ところで、人が口頭である目的地までの経路を指示する際には、よく、「2 番目の 3 叉路を右折し、最初の T 字路へ移動せよ」というような指示をする。柴田らは、このような指示を移動ロボットに与えることによるナビゲーション方法を検討している [14]。しかし、これは限られた想定環境での計算機上のシミュレーションレベルの議論しかなされておらず、センシング能力の限界のある実際のセンサを用いて、実際の環境において、どれくらい適用できるかはまだ未知数である。

このように従来の研究の中でも、詳細な環境モデルを用いずにセンサーベーストビヘービアを用いて移動ロボットの屋内ナビゲーションを実現している例は確かにすでに存在する。しかし、例えば、用いている超音波距離センサの能力によってどのような形状の壁まで壁沿い走行が可能であるのかというような限界の議論や、壁沿い走行と位置ベース走行を併用するナビゲーション、あるいは柱状の凹凸の多い屋内の廊下において安定な壁沿い走行を行なう実施例などはほとんどなかったのが現状である。

## 2.4 レーザレンジセンサに関する研究

本論文における研究では、超広角カメラと扇状光を用いたレーザレンジセンサを開発した。従来のレーザレンジセンサについては、例えば Sick 社製レーザレンジセンサで、広い角度にわたって距離を得るようなセンサとして LMS と呼ばれるものなどがある。これは装置の規模が大きいいため、小さい移動ロボットに搭載することが難しい。また、カーネギーメロン大学で開発されている Navlab と呼ばれる自律走行車に使用された ERIM レーザレンジセンサは  $30^\circ \times 80^\circ$  の窓の立体的な距離データが得られるが、非常に高価であり、また装置も大きいことが知られている [18]。このように、市販のレーザレンジセンサでは、小型の移動ロボットに搭載でき、かつ、安価なレーザレンジセンサはあまり例をみない。一方、筆者の所属する研究グループでも、水平視野角  $60^\circ$  程度の視野が狭いカメラとスリットレーザを用いて、三角測量の原理によるレーザレンジセンサを開発している [19]。これは、ロボットの前方に存在する障害物を見つけるのには役に立ち、これを用いた障害物回避に関する研究もなされているが、ロボットが障害物を見失うおそれもある。もっと広い視野範囲を有するレーザレンジセンサが必要である。

## 2.5 2章のまとめ

本論文における研究テーマである詳細な環境モデルを用いない移動ロボットの屋内ナビゲーションに関連して、このテーマと対極にある従来からのアプローチである、詳細な環境モデルをもたせる屋内ナビゲーションに関する研究、および、本テーマに直接にかつ密接に関連する従来の研究について概観した。また、本論文の第6章で述べられるレーザレンジセンサに関する研究についても概観した。

## 第 3 章

# 研究用プラットフォームとしての自律移動ロボット「山彦」

本章では本研究に用いた自律移動ロボットの概要を記す。第 4 章以降で述べる本研究において提案するアルゴリズムの実装は、このロボットのアーキテクチャに強く依存している。言い換えれば、このロボットのアーキテクチャがすでに実装され、自由に利用できる環境にあったため、本研究を短時間のうちに遂行できたと言っても過言ではない。第 4 章以降で用いる本ロボットに特有の考え方や術語等についてここで説明を行ない、本論への準備とする。

### 3.1 機能分散アーキテクチャと行動プログラム

使用した移動ロボットのプラットフォームは、我々の研究グループで「山彦」と呼ばれる図 3.1 のような移動ロボットである。ロボットの大きさは、縦 50[cm]、横 35[cm]、高さ 50[cm] である。ロボットの下部にはバッテリーがあり、筐体中央にこのロボットに必要な計算機基板を挿入するラックがある。本研究を進めるのに必要なセンサ群は図 3.1 に示したロボットの上部に搭載した。したがって、行動に必要なエネルギー源をもち、センシングや走行のためにロボットを管理・制御する計算機をすべてこの筐体上に有する自立・自律型の移動ロボットである。

このロボット全体を管理・制御するための計算機は複数用意されており、ロボットの走行制御系、センサ系、位置推定系、それらを統括する統括制御系に割り当てられている(図 3.2)。このように、ロボットに必要な機能をつかさどる計算機を分散させ



図 3.1: LR-1 タイプ「山彦」

たアーキテクチャを機能分散アーキテクチャと呼んでいる [41]。結果としてそれぞれの機能について分散して開発・メンテナンスできる特長がある。このそれぞれの機能をつかさどる計算機基板を、走行制御モジュール (Locomotion Module)、センサモジュール (Sensor Module)、位置推定モジュール (Position Estimation Module)、マスターモジュール (Master Module) と呼んでいる。

実際のロボットの行動を記述するためのプログラムはマスターモジュール上に置く。本研究において筆者が実装したセンサベーストビヘービアに基づく行動プログラムやナビゲーションプログラムもこのマスターモジュール上に置かれるものである。

マスターモジュールと各機能モジュールの間には、状態告知板 (SIMP State Information Monitoring Panel) がある。各機能モジュールは、その内部で持っているロボットの状態に関する最新のデータを SIMP に書き込み、常に更新している。各機能モジュールが SIMP に書き込むデータは、例えばセンサモジュールであれば、そのセンサが自律的に計測しているデータであり、走行制御モジュールであればオドメトリによる現在の位置・方位の推定値である。したがって、マスターモジュールは各機能モジュールが SIMP に書き込んだデータを手に入れることができる (図 3.2)[41]。この最新の情報とは、例えばセンサモジュールであれば、そのセンサが自律的に計測しているデータであり、走行制御モジュールであればオドメトリによる現在の位置・方位の推定値である。言い換えれば、マスターモジュールはロボットのシステム全体の状態を常時監視できるようになっている。マスターモジュール上にユーザが記述する行動プログラムはロボット行動記述言語 Robol/0[52] か、C 言語のどちらかを用いる。どちらの言語を用いるにしても、それぞれの SIMP に各機能モジュールによって書き込まれている情報をこの言語の中から読みとるシステム関数 (3.2節参照) が用意されている。また走行制御系に対しては、実際にロボットが移動するコマンドを発行するためのシステム関数がこの行動プログラムの中で発行できるように用意されている。このコマンドはマスターモジュールの側から SIMP に書き込まれ、走行制御系の側で読みとられるようになっている。ユーザが記述する移動ロボットの行動プログラムにおいて、センサモジュールからの情報を読みとり、必要な走行制御コマンドを発行することを繰り返すことによりセンサベーストビヘービアに基づく行動を記述できる。

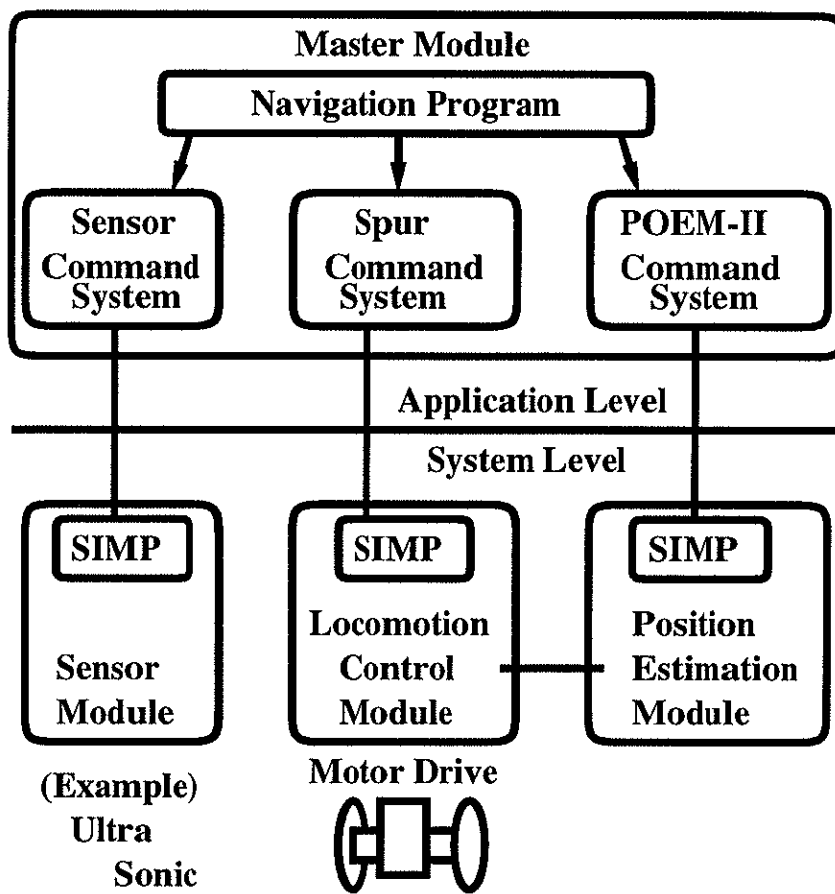


図 3.2: 「山彦」の機能分散アーキテクチャ

## 3.2 走行制御系

マスターモジュール上の CPU で実行される、ロボットの行動を指示するプログラムによって発行することができる走行制御コマンド関数は、表 3.1 のようなものである。筆者らの研究室ではこのコマンド体系とそのコマンドに基づく走行制御モジュール上の走行制御ソフトウェアを Spur と呼んでいるので、これらを Spur 関数と呼ぶことがある。表 3.1 に示したのは Spur 関数のうちの一部であるが、Spur\_line\_LC(), Spur\_line\_GL(), Spur\_line\_FS() は座標系における直線に追従することを指定するものである。関数末尾の LC, GL, FS はそれぞれ引数に与える座標値の座標系を表しており、すぐあとで説明する。引数の  $(x, y, \theta)$  はその座標系において  $(x, y, \theta)$  を通る直線を指定する。走行制御モジュール上で計算されているオドメトリに基づく推定位置・方位も、各々の座標系に関してもっており、マスターモジュール上のナビゲーションプログラムにおいてこのようなコマンドが発行されるごとに、走行制御系はそのコマンドに従った制御を行なうようになっている。この走行制御コマンドは実際に第 6 章を除く各章で述べる走行アルゴリズムを記述する際に利用されている。

### [走行制御コマンドにおける座標系]

走行制御系や位置推定系などでロボットの位置を表現し、また、位置に基づいて走行制御するために、2次元平面上に図 3.3 に示すようなグローバル座標系 (GL 座標系)、ローカル座標系 (LC 座標系)、そして、フロントサイド座標系 (FS 座標系) の 3 つの座標系が定義されている。

- グローバル座標系 (GL 座標系)

この座標系は、地図などを作成する時に最初に決める座標系であり、地上 (床面上) に固定されたものである。地図上で、ロボットの位置、角度、追従させる線などを表現するために必要な基本的な座標系である。

- ローカル座標系 (LC 座標系)

この座標系は、グローバル座標系上の任意の位置に原点をとり、グローバル座標系上で自由に定義される座標系である。これもグローバル座標系と同様に地上に固定されたものである。この座標系は、グローバル座標系以外の座標系でロボットの位置、角度、追従させる線などを表現するための座標系である。

- フロントサイド座標系 (FS 座標系)

この座標系は、上の 2 つの座標系とは異なり、図 3.3 のように、ロボットの前方を X 軸、左方向を Y 軸とするような、ロボットに固定された座標系であり、ロボットの移動とともに動く。

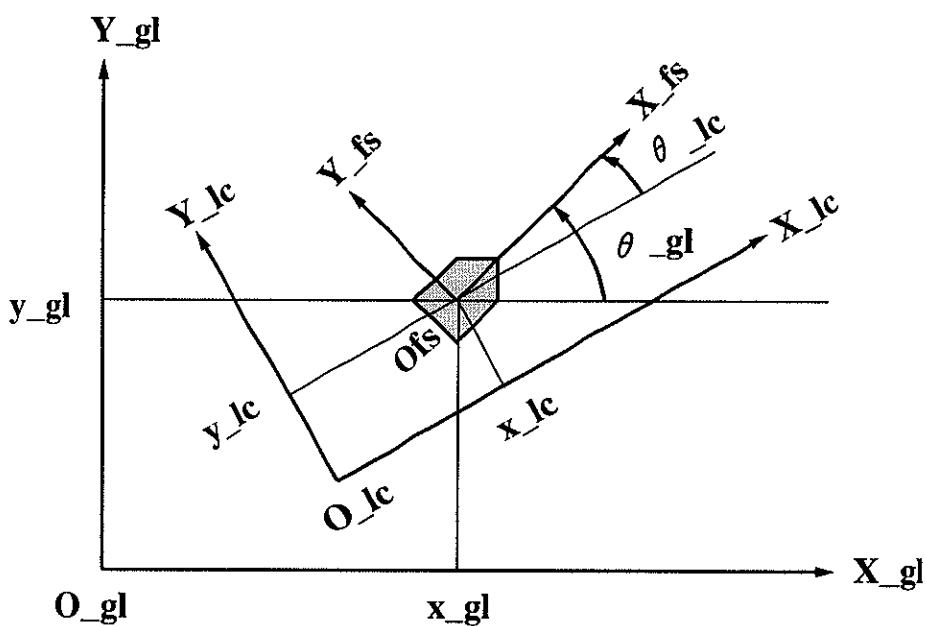


図 3.3: GL, LC, FS 座標系

表 3.1: 走行制御系の主要な関数 (Spur 関数)

関数例 (LC 座標系)	関数例 (GL 座標系)	関数例 (FS 座標系)	効果
Spur_line_LC(x,y,θ)	Spur_line_GL(x,y,θ)	Spur_line_FS(x,y,θ)	直線追従
Spur_arc.t_LC(x,y,θ,r)	Spur_arc.t_GL(x,y,θ,r)	Spur_arc.t_FS(x,y,θ,r)	円弧追従
Spur_stop_LC(x,y,θ)	Spur_stop_GL(x,y,θ)	Spur_stop_FS(x,y,θ)	停止
Spur_spin_LC(θ)	Spur_spin_GL(θ)	Spur_spin_FS(θ)	回転



### 3.3 位置推定系コマンドと位置推定モジュール

位置推定モジュール上には、“POEM-II” と呼ばれるオドメトリベースの位置推定ソフトウェアが稼働している。これは、走行制御モジュール上のオドメトリと同様に、モータ軸にとりつけられたパルスエンコーダ情報を利用して独立にロボットの自己位置を推定しかつ走行中常に更新している。走行制御モジュールのオドメトリでは自己位置と方位を推定しているのみであるが、本モジュールでは、オドメトリによる自己位置・方位とともにこれらに関する推定誤差分散行列の計算も行なっているのが特徴である。また、マスターモジュール上のロボットの行動を指示するプログラムから表 3.2 に定義されたものを発行することができる。例えば、ロボットの走行中、オドメトリによる推定位置付近の左側に壁があり、その位置が事前にわかっているとする。超音波センサによる壁までの測距値  $y$  とその誤差分散  $dev$  を関数  $Poem\_fuse\_y(y,dev)$  で与えることにより、 $y$  および  $dev$  とオドメトリによる推定位置とその推定誤差分散との間で最尤推定を行なわれ、より正確な、すなわち誤差分散の小さな推定位置・方位に修正される。なお、この最尤推定においては、超音波センサの測距値という 1 次元の情報であっても、ロボットの自己位置  $(x,y,\theta)$  の推定誤差分散行列の共分散項によってこの 3 変数に影響し、推定自己位置それぞれの誤差が補正される。すなわち、超音波の測距値という 1 次元の情報であっても、最尤推定によって自己位置  $(x,y,\theta)$  が補正でき、かつその誤差分散も全体として小さくなる [21]。最尤推定によって推定位置が修正されると、走行制御モジュール上のオドメトリ情報も自動的に更新されるようになっている。ここで示した位置推定モジュールの機能と関数は第 5 章でのみ用いられる。

表 3.2: 位置推定系の主要な関数 (POEM-II 関数)

関数	効果
$Poem\_position(&x,&y,&\theta)$	GL 座標上の位置情報を得る
$Poem\_set\_position(&x,&y,&\theta)$	GL 座標上の初期位置を設定する
$Poem\_fuse\_y(y,dev)$	GL 座標上の $y$ を修正する

## 第 4 章

# ソナーリングによる壁沿い走行

### 4.1 はじめに

自律移動ロボットを詳細な環境モデルを用いずに屋内環境で走行させる場合、環境内の対象物を目標と定めてそれに向かう、あるいはその対象物の周囲に沿って走行するなどの手法を開発する必要がある。詳細な環境モデルを移動ロボットに事前に与えなければ、ロボットにとっては未知の環境の中を走行することになり、その場の環境に関する情報をセンサ系によって取得し、その情報に基づいてその都度とるべき行動を決定する様式で行動アルゴリズムを記述することになる。このようなアルゴリズムによって発現するロボットの行動はセンサベーストビヘービアとなる。

さて、本章では、屋内環境において存在する壁面や柱、設置されている消火器、ゴミ箱、などの対象物の周囲の壁に沿って移動ロボットをこれらに衝突せずに安定に走行させることを取り扱った。この対象物の周囲に沿う移動ロボットの走行のために用いる外界センサは超音波距離センサを円環状に並べた距離センサ(ソナーリング)である。これによってロボットの周囲に渡る距離情報を得て、対象物の周囲に沿ってある一定の距離を保ちながらロボットを走行させることが本章における目的である。廊下や室内の壁面、あるいは対象物の周囲外壁面に沿う走行を以後「壁沿い走行」と呼ぶことにする。したがって、本章はソナーリング上の複数の超音波距離センサによる、対象物までの距離情報に基づく壁沿い走行が研究の目的である。

以下、本章では、筆者が製作したソナーリングの概要(4.2節)、ソナーリングからの情報に基づく壁沿い走行のアルゴリズム(4.3節)、このアルゴリズムに基づく壁沿い走

行の実験例(4.4節)を示す。一般に、ソナーリングのような超音波センサを円環状に並べた距離センサを使用すれば、移動ロボットをどんな複雑な形状の壁にも沿って安定に走行させることができると考えられているが、超音波の特性などにより、対象物の形状によって、本質的に壁沿い走行がうまくいかないことがあることも、4.4節で実験例として示す。

## 4.2 ソナーリング

超音波距離センサを円環状に並べたソナーリングを製作し、第3章で述べた自律移動ロボット「山彦」に実装した。本節ではこのソナーリングの概要を説明する。この超音波距離センサは超音波パルスの time of flight を利用する一般的なものであり、その配置は、図 4.1 のように 12 方向とした。隣接するセンサの干渉を防ぐため、30 ミリ秒毎に互い違いの 6 方向の対象物までの距離の計測を行なっている。使用したトランスデューサの指向性の広がり角(指向角)は約  $50^{\circ}$  であり、隣り合うトランスデューサの指向角は、上面から見た場合約  $25^{\circ}$  オーバーラップしており、正面より  $\pm 150^{\circ}$  の範囲では全体をカバーしている。実際にソナーリングを搭載した移動ロボットの概観を図 4.2 に示す。この超音波距離センサの測定可能範囲は約 20cm から 2m、測距精度はほぼ 1cm である。測定可能範囲外に対象物がある場合は、「対象物なし」という値が得られるようになっている。各超音波距離センサのための超音波パルスの発信・制御と受信・距離計測は第3章で述べたセンサモジュールによって行ない、各方向の測距値は SIMP に書き込まれるようになっている。

## 4.3 壁沿い走行手法

### 4.3.1 超音波の鏡面反射特性と仮想壁

本節ではソナーリングによって計測される 12 方向の距離データを用いる壁沿い走行アルゴリズムを説明する。この壁沿い走行のアルゴリズムの設計においては、事前に詳細な環境モデルを与えないのが前提であるので、実際の対象物の壁の形状は未知であり、事前にこの形状を限定しなかった。よって、壁の形状には、平面状の壁、円柱状の壁、不定形な物体等種々のものがあり得る。一方、超音波は、対象物が鏡面であるかのように反射(スペキュラー・リフレクション)する性質を有している。例えば

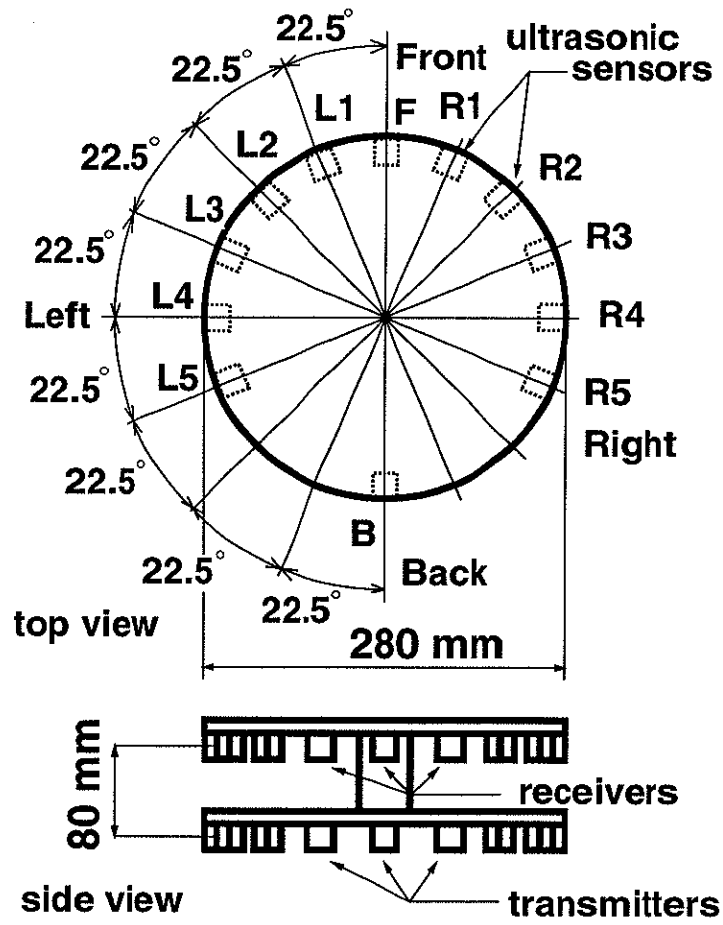


図 4.1: ソナーリングの概観

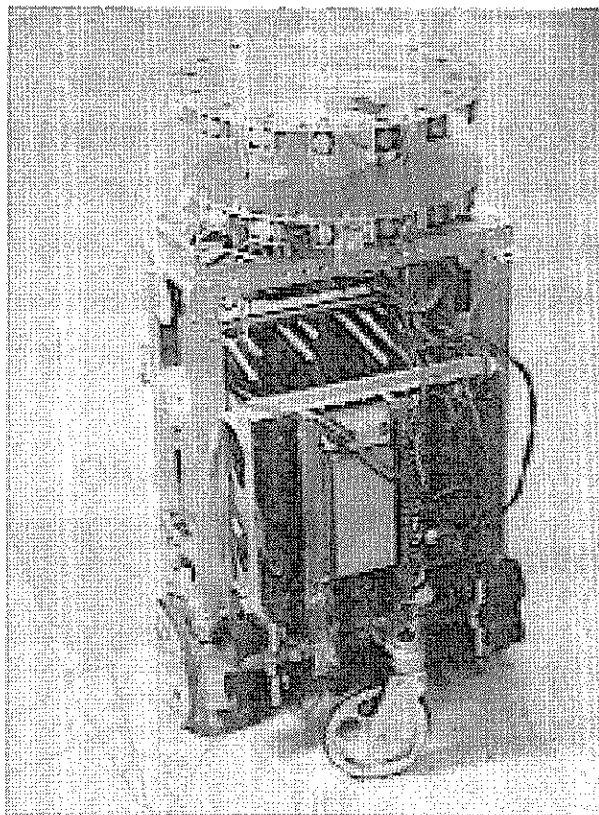


図 4.2: ソナーリングを搭載した移動ロボットの概観

図 4.3 のような不定型な対象物が超音波距離センサのトランスデューサの前にある場合には、トランスデューサに正対する面からの反射を受けることになり、結果的に使用しているトランスデューサの指向角の範囲内にある対象物の表面までの最短距離が得られる。すなわち、図 4.3 の例においては対象物体の [A] 点までの距離が実際には求まる。しかし、ひとつの超音波距離センサはどの方向からの反射波によってその距離が得られたかを判別する分解能はない。また、壁の形状そのものは未知であることが前提である。そのため、次節に示す壁沿い走行のアルゴリズムを考察する際には、以下のような仮想壁を考えた。すなわち、図 4.3 のように超音波距離センサで [A] 点までの距離が検出されても、超音波距離センサのトランスデューサの中心軸方向(図中の点線方向)に、直交する仮想壁が [B] 点にあると考え、トランスデューサから [A] 点までの距離が [B] 点までの距離となるようにおく。次節で示す走行アルゴリズムでは、その仮想壁と平行かつ手前に距離  $D$  離れた直線を考え、その直線に追従するように移動ロボットを走行させる方針をとり、この走行をロボットが距離 3cm 進む毎にソナーリングの距離データを得て処理を繰り返すことにした。

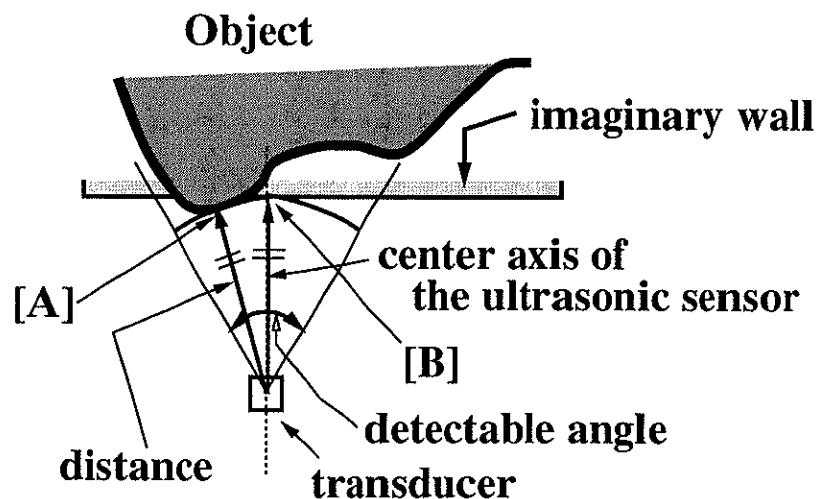


図 4.3: 超音波の鏡面反射特性と仮想壁

### 4.3.2 壁沿い走行アルゴリズム

第 3 章にも述べたように、壁沿い走行を実現するアルゴリズムはマスターモジュール上のプログラムとして記述する。このアルゴリズムとしては、次の方針をとった。すなわち、ソナーリングから得られる 12 方向の距離情報により、まず、次の 4 つのケースに場合分けする。そのケースの各々に対して適切な走行制御コマンドを定め発行する。これを移動ロボットが 3cm 走行するごとに繰り返す。ただし、以下の Case1 の場合以外は停止せず、走行したままでこの繰り返しを行なう。また、以下の場合分けは、ロボット中心から対象物の壁の表面まで、一定の距離  $D$  を保って、壁を左手に見るようにロボットを走行させる場合について示す。また、特に下の Case1 において停止コマンドが発行されてから実際に移動ロボットが停止するまでには時間がかかり、制動距離が存在するため、それを吸収するような一定の距離値  $d$  を予め与える。この値  $d$  は、他の Case の場合も、距離  $D$  に対するマージンとして利用している。また、イタリック体の  $F, R1, R2, R3, R4, R5, L1, L2, L3, L4, L5, B$  はソナーリングのそれぞれのトランスデューサ  $F, R1, R2, R3, R4, R5, L1, L2, L3, L4, L5, B$  から得られた距離値とする。

#### Case1 (図 4.4)

$R3, R2, R1, F$  が距離  $(D + d)$  以下であるかをこの順に調べ、もし、該当した場合、これらのうちで最初に距離  $(D + d)$  以下であることを満足したセンサの中心軸の方向とその距離に仮想の正対する壁があると考え。そして、一時停止してからその仮想の壁がロボットの真左に正対するようにスピンコマンドを走行制御系に送る。続いて、その場から直進するコマンドを走行制御系に送る。

#### Case2 (図 4.5)

Case1 の条件を満たさなかった場合、かつ  $L1, L2$  が距離  $(D + d)$  以下であるかをこの順に調べ、もし、該当した場合、これらのうちで最初に距離  $(D + d)$  以下であることを満足したセンサの中心軸の方向とその距離に仮想の正対する壁があると考え。そして、一時停止せずに、その仮想の壁がロボットの真左に正対するように、その場からその仮想壁と平行な直線に追従するような直線追従コマンドを走行制御系に送る。

#### Case3 (図 4.6)

Case1, Case2 の条件を満たさなかった場合、かつ  $L4, L3, L5$  のどれかが距離  $2D$

未満であった場合、L4,L3,L5 の最短の距離の方向とその距離に仮想の正対する壁があると考え、その壁に距離 D を保って沿うように直線追従するコマンドを走行制御系に送る。

#### Case4 (図 4.7)

Case1,Case2,Case3, のどの条件も満たさなかった場合、半径 D の円弧を描きながら左方向に進むように円弧追従するコマンドを走行制御系に送る。ただし、円弧が一周を描いた時は停止するものとする。

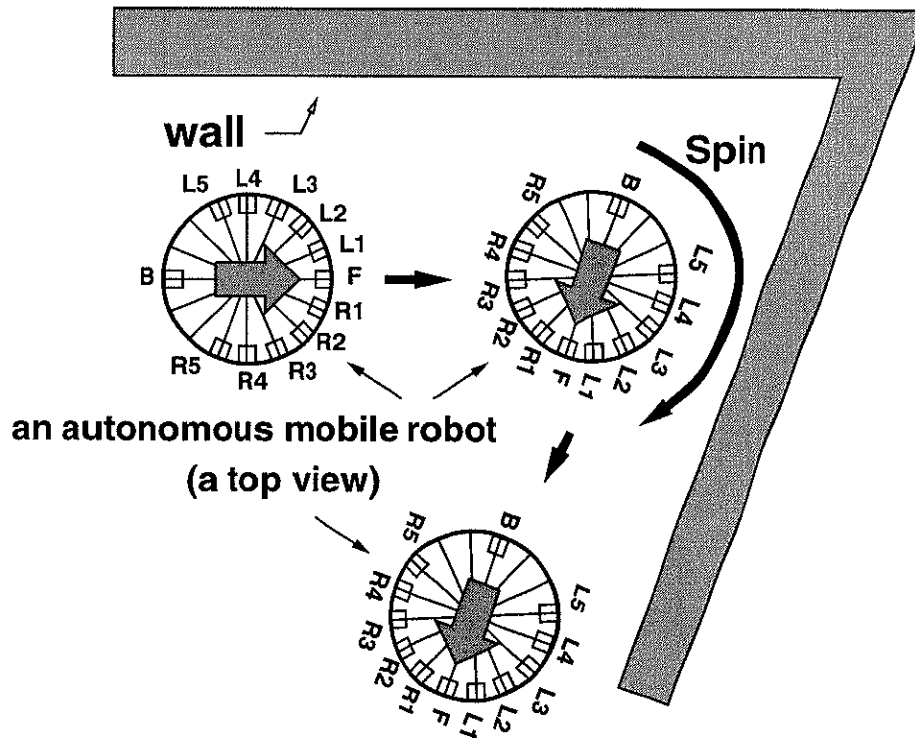


図 4.4: Case1(正面または、右斜めに壁が検出された場合)

#### 走行プログラム

以下に、C 言語様に記述した壁沿い走行のプログラムを示す。以下のプログラムは、ロボット中心が壁から距離 D を保って左壁に沿ってロボットを走行させるプログラム



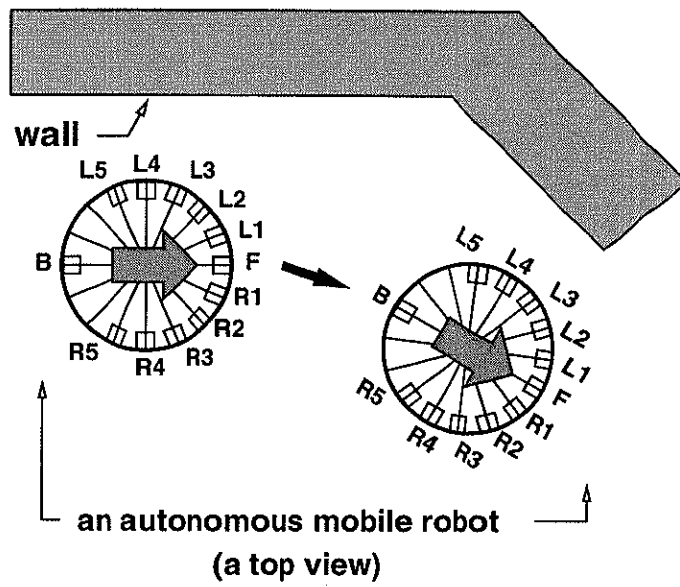


図 4.5: Case2(左斜めに壁が検出された場合)

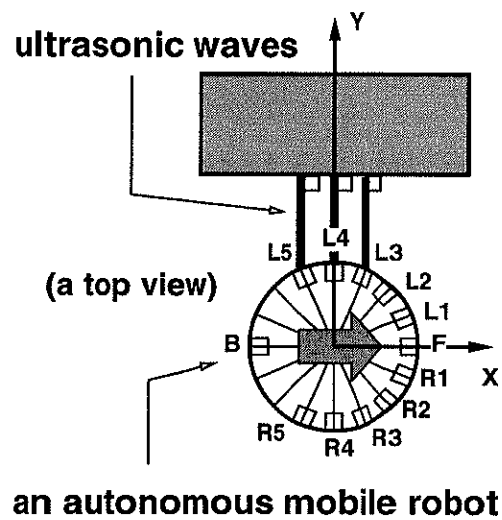
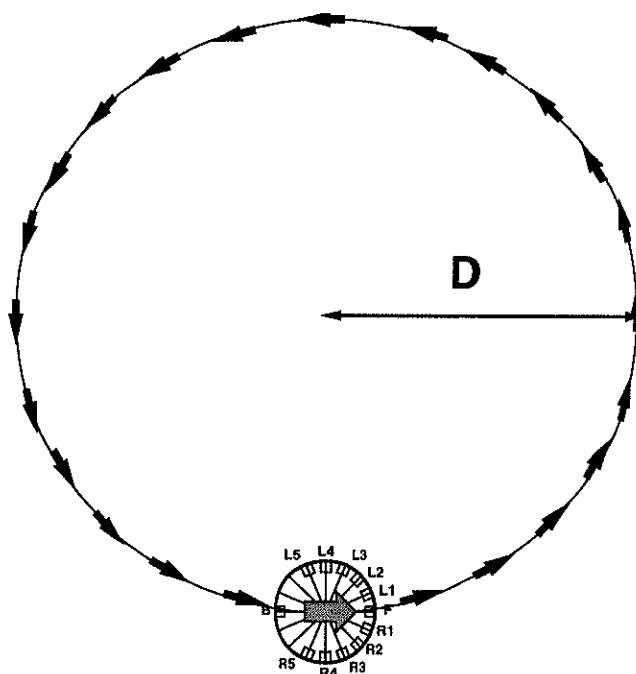


図 4.6: Case3(左横に壁が検出された場合)



**A mobile robot (a top view)**

図 4.7: Case4(壁が検出できない場合)

である。また、 $d$  はあらかじめ与える一定の距離とする。ここで、 $L1, \dots, L5, F, B, R1, \dots, R5$  は各方向の超音波センサの測距値であり、 $\min$  は  $L3, L4, L5$  のうちの最小値を示す。

```

/*****
* 前方か右方向に壁が検出された時、一旦停止して
* スピンした後、壁が見えたセンサに垂直な方向に進む
*****/
if( $R3 \leq D + d$ ){ /*  $R3 \leq D+d$  */
    Spur_stop_Q(); /* 急停止 */
    Spur_spin_FS(-158);
/* 現在の R3 方向が L4 の方向に重なる様、158° 時計方向にスピン */
    Spur_line_FS(0,0,0); /* 前進 */
else if( $R2 \leq D + d$ ){ /*  $R2 \leq D+d$  */
    Spur_stop_Q(); /* 急停止 */
    Spur_spin_FS(-135);
/* 現在の R2 方向が L4 の方向に重なる様、135° 時計方向にスピン */
    Spur_line_FS(0,0,0); /* 前進 */
else if( $R1 \leq D + d$ ){ /*  $R1 \leq D+d$  */
    Spur_stop_Q(); /* 急停止 */
    Spur_spin_FS(-113);
/* 現在の R1 方向が L4 の方向に重なる様、113° 時計方向にスピン */
    Spur_line_FS(0,0,0); /* 前進 */
else if( $F \leq D + d$ ){ /*  $F \leq D+d$  */
    Spur_stop_Q(); /* 急停止 */
    Spur_spin_FS(-90);
/* 現在の F 方向が L4 の方向に重なる様、90° 時計方向にスピン */
    Spur_line_FS(0,0,0); /* 前進 */
/*****
* 左斜めの方向に壁が検出された時、壁が見えたセンサ
* に垂直な直線を目標直線として走行する

```

```

*****/
else if(L1 <= D + d){ /* L1 ≤ D+d */
    Spur_line_FS(0,0,-68);}
/* 現在の L1 方向が L4 の方向に重なる様、ロボットに対して時計方向に 68° 傾いた
直線に追従させる */
else if(L2 <= D + d){ /* L2 ≤ D+d */
    Spur_line_FS(0,0,-45);}
/* 現在の L2 方向が L4 の方向に重なる様、ロボットに対して時計方向に 45° 傾いた
直線に追従させる */
/*****
* 左方向に壁が見えた時、最も近くに壁が見えたセンサ
* に垂直な直線を目標直線として走行する
*****/
else if(L4 == min&&L4 < 2 * D){ /* L4 < 2D、かつ、L4 が最小 */
    Spur_line_FS(0,L4 - D,0);}
/* ロボットの L4 側に距離 L4 - D に L4 方向と直交する直線に追従すべく前進 */
else if(L3 == min&&L3 < 2 * D){
    Spur_line_FS(0,L3 - D,-23);}
/* ロボットの L3 側に距離 L3 - D に L3 方向と直交する直線に追従すべく前進 */
else if(L5 == min&&L5 < 2 * D) {
    Spur_line_FS(0,L5 - D,23);}
/* ロボットの L5 側に距離 L5 - D に L5 方向と直交する直線に追従すべく前進 */
/*****
* どの条件にも該当しない時、現在位置から左方向に
* 半径 D の円弧を描くことを目標軌跡として走行
* (ただし、円弧が一周を描いた時は停止する。)
*****/
else{ /* どの条件にも該当しない場合 */
    Spur_arc_t_FS_cm(0,0,0,D);} /* 半径 D の左円弧に追従 */

```

## 4.4 ソナーリングによる壁沿い走行実験

本章で示した壁沿い走行アルゴリズムがどの程度有効に働くのかを検証するため、ソナーリングを搭載した自律移動ロボット「山彦」上に上記アルゴリズムを実装した。ただし、移動ロボットは壁を左にみて沿うこととし、4.3節で示したアルゴリズムの定数  $D$  と  $d$  をそれぞれ、 $D=50\text{cm}$ ,  $d=10\text{cm}$  とした。また、今後の実験は、全てこのパラメータで行った。以下に示すような様々な形状の壁あるいは障害物を用いて壁沿い走行を実行させた。この結果、超音波センサで壁が検出される限り、ほぼ期待した通りに壁に沿って走行できることが確かめられた。しかし、壁の形状によっては期待に反して超音波センサで壁が検出できない場合もあり、この場合は壁に衝突する場合もあった。以下いくつかの実験例を示す。

### 4.4.1 実験1(四角柱、円柱、台形の内側、動く円柱)

例として静止している1辺約75cmの4角形の壁に沿って実際に走行させて、オドメトリにより記録した軌跡を図4.8に示す。図4.8には、ソナーリングが3cm走行毎に得た環境データがアルゴリズムのどのケースに属したかを同時に示してある。この結果から、移動ロボットが四角柱の壁に沿って走行する場合、移動ロボットは四角柱の角の部分を検出できていないが、その部分はアルゴリズムのCase4によりカバーされ壁沿い走行が実現されていることがわかった。

また、直径75cmの円形の壁に沿って30cm/secの速度で走行させた場合の実機実験の結果を図4.9に示す。この結果から、移動ロボットが円柱の壁に沿って走行する場合、移動ロボットは、常時、壁を検出できていて、アルゴリズムのCase3により壁沿い走行が実現されていることがわかった。

ロボットが正面及び右斜めに壁を検出する場合の例を次に示す。例として囲いの内側を走行する場合の結果を図4.10に示す。この結果から、移動ロボットが台形の内側の壁に沿って走行する場合、壁の角の部分で、アルゴリズムのCase1が満たされ、一旦停止することにより、壁への衝突を回避し、安全な壁沿い走行が実現されていることがわかった。

次に、円柱状の壁がゆっくり動いている場合に、上記のアルゴリズムによって壁沿い走行させた結果を図4.11に示す。ここで、ゆっくり走行する物体は他のロボットを

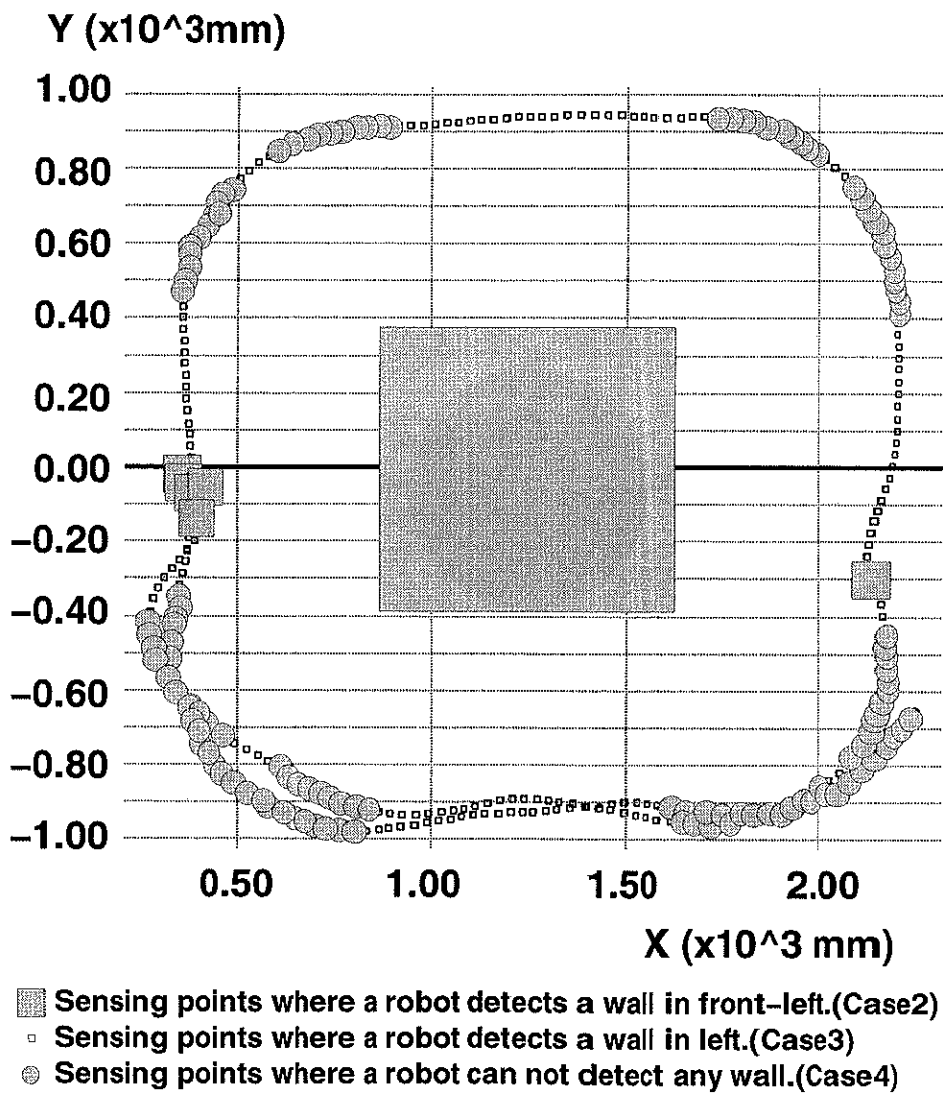


図 4.8: 四角柱の壁に沿う実験 (実機実験)

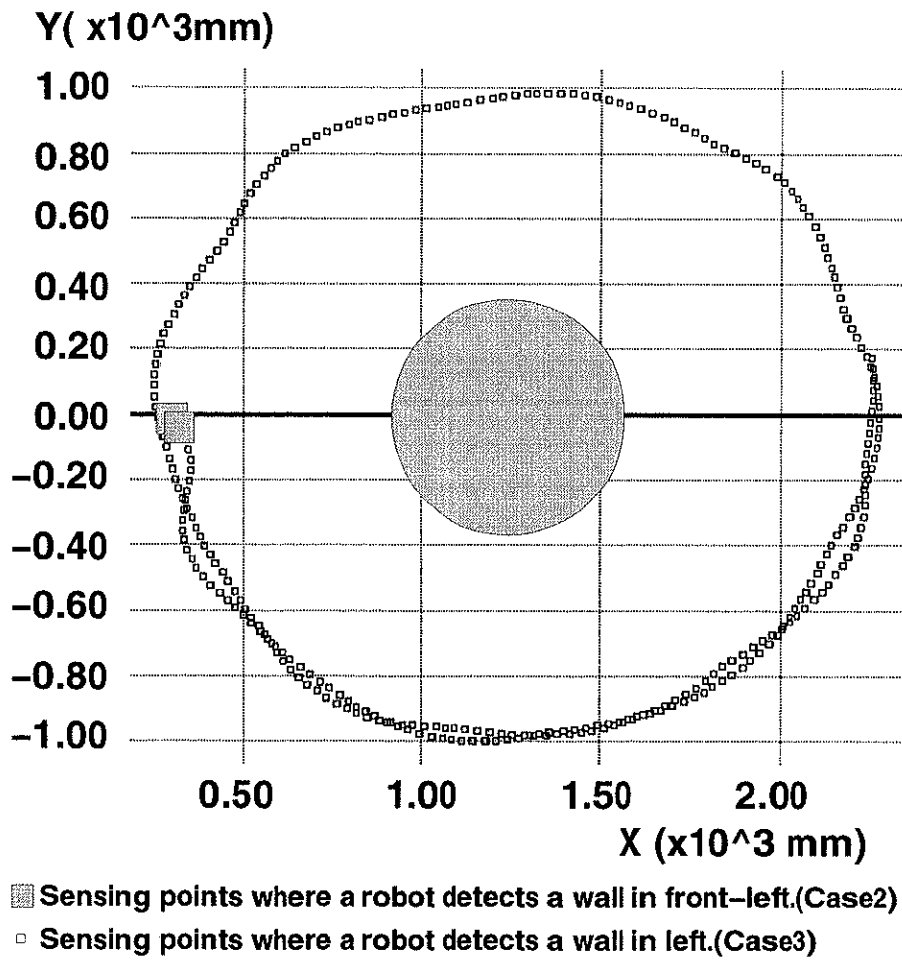


図 4.9: 円柱の壁に沿う実験 (実機実験)

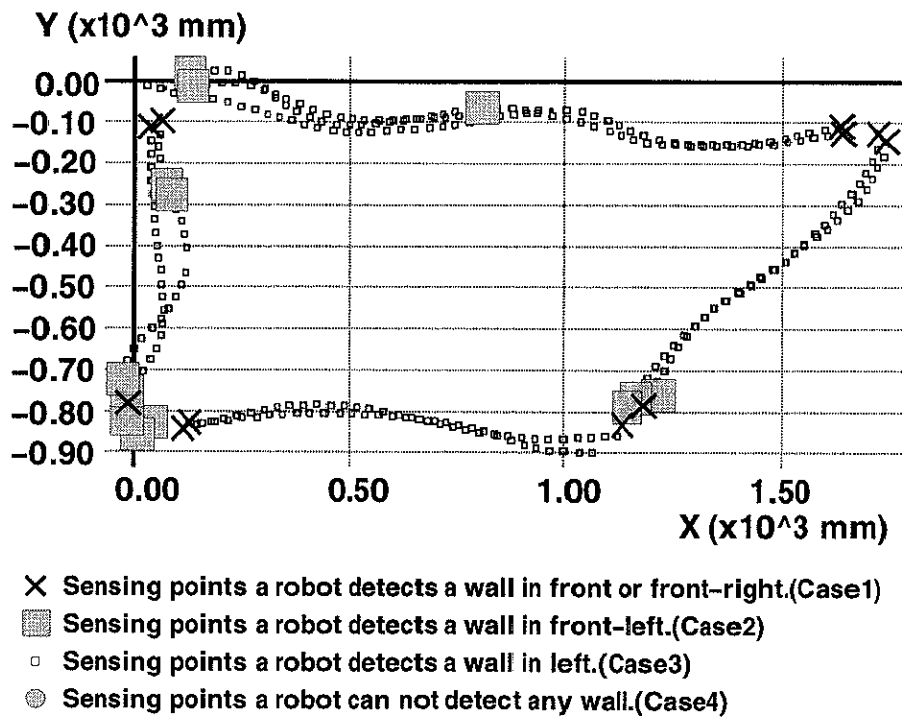


図 4.10: 台形の壁の内側の壁に沿う実験 (実機実験)



用いて実現した。ロボットを秒速 30cm 程度で走行させる場合、対象物速度が秒速 5cm 程度であれば、静止している場合と同様に壁沿い走行をすることができた。図 4.11 をみればわかるように、移動ロボットがゆっくり動くような円柱の壁に沿って走行する場合、壁を検出できないことがまれにあることがわかった。ちなみに、静止している円柱を検出できないことはほとんどなかった。しかし、この場合もこの壁沿い走行アルゴリズムにより壁への衝突を回避して安全な壁沿い走行が実現されていることが確認された。

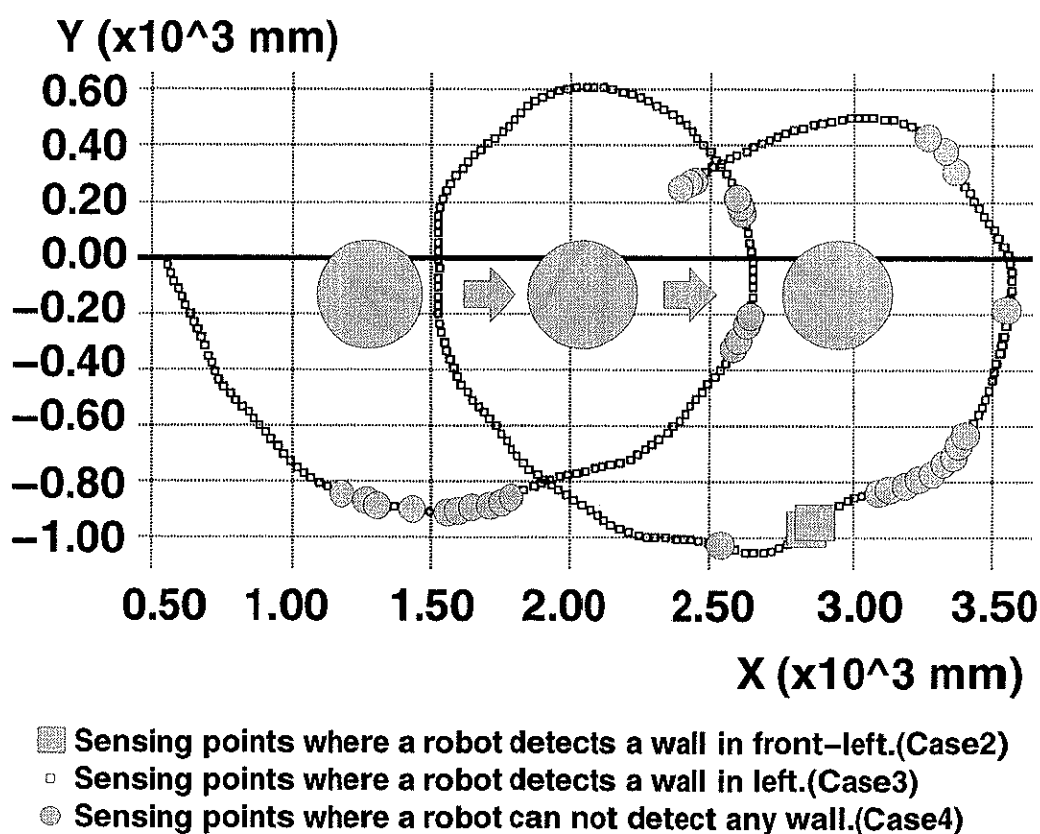


図 4.11: ゆっくり動く円柱の壁に沿う実験 (実機実験)

#### 4.4.2 実験 2(大小の星型対象物、大小の“\*”型対象物)

図 4.12、4.13は、移動ロボットを星型対象物の外壁に沿って走行させた実験結果である。星型の壁の一辺の長さは、図 4.12では 90cm、図 4.13では 23cm である。また、

それぞれのとがった角の部分の辺のなす角度は、大小の星型とも  $30^\circ$  とした。この実験結果から、移動ロボットは衝突せずにこのような星型の対象物でさえも、外壁に沿って走行できることがわかった。このように星型の対象物に沿って走行させた場合、ほとんどアルゴリズムの条件 Case4 があてはまり、ロボットはちょうどロボットの真正面で対象物を検出する傾向があった。移動ロボットの軌跡は、大きい星型対象物の外壁に沿って走行させた場合に比べ、小さい星型対象物の外壁に沿って走行させた場合のほうが安定していなかった。

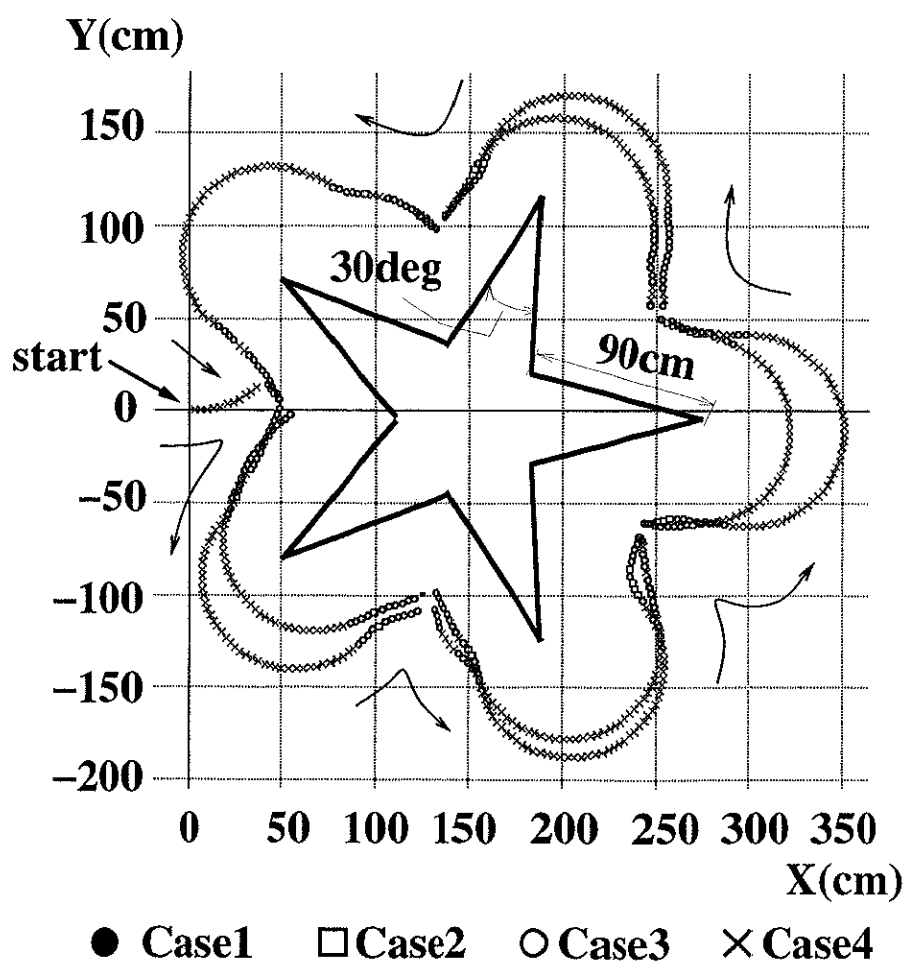


図 4.12: 大きい星型対象物の外壁に沿って走行させた場合 (実機実験)

また、形状が記号“\*”のような対象物の外壁に沿って移動ロボットを走行させた場合の実機実験結果を図 4.14と図 4.15に示す。“\*”型の対象物の壁の一辺の長さは、図

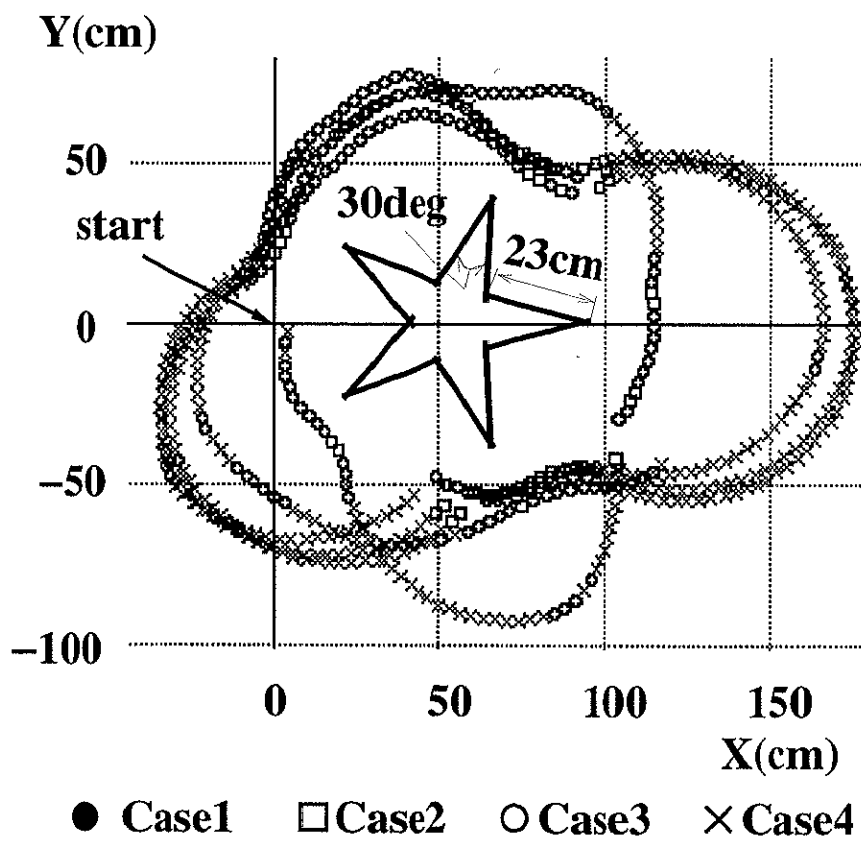


図 4.13: 小さい星型対象物の外壁に沿って走行させた場合 (実機実験)

4.14では135cm、図4.15では45cmである。この実験結果から、移動ロボットは衝突せずにこのような“\*”型の対象物でさえも、外壁に沿って走行できることがわかった。移動ロボットが、大きい“\*”型対象物の外壁に沿って走行する場合、大きい星型対象物の外壁に沿って走行する場合と同様に、ほとんどアルゴリズムの条件 Case4 があてはまっていた。しかし、移動ロボットが、小さい“\*”型対象物の外壁に沿って走行する場合、小さい星型対象物の外壁に沿って走行する場合とは違って、あたかも、小さい円柱の回りをまわっているかのように、安定に走行していた。これは、超音波の多重鏡面反射により、移動ロボットの左方向にほとんど常時、壁が検出され、アルゴリズムの条件 Case3 がほとんど常時あてはまっていたためだと考えられる。

#### 4.4.3 壁沿い走行の限界の検討 1(四角い対象物が動く場合)

上記に提案したアルゴリズムは超音波センサによって自分に正対した壁面が検出できるという仮定に基づいている。この仮定は、数十 KHz の超音波を用いる一般的な超音波センサにとっては、室内環境ではほぼ満たされる。

一方、このことは、図4.8の例のように角のある対象物について、角をまわっている場合は沿うべき対象物が全く見えない場合があることを示している。

図4.8で壁沿い走行が成功しているのは、何も見えない時には、半径50cmの円弧を描いて左へ向かうという Case4 の動作によっている。したがって、この間に対象物が動くとこのアルゴリズムでは対象物と衝突してしまうことがあった。その実験例を図4.16に示す。ロボットの速度は秒速30cm程度、四角い物体の速度は秒速5cm程度である。この例では、ロボットは物体に衝突してしまった。図4.16からわかるようにロボットはほとんど Case4 に従って動いており、この四角形の対象物は見えていない。これは、対象物の表面で超音波が鏡面反射する結果、超音波の反射波がソナーリングに対して戻ってこず、測距に失敗しているためと考えられる。

対象物がどのように動いていてもそれに対処して衝突せずに走行するためには、各時刻において前方の対象物が観察できるか、あるいは、観察できた瞬間の対象物の、その形状と動きを推定できるかの少なくともどちらかが必要である。しかし、それを鏡面反射をする超音波によって観察または推定することは難しいと考えられる。

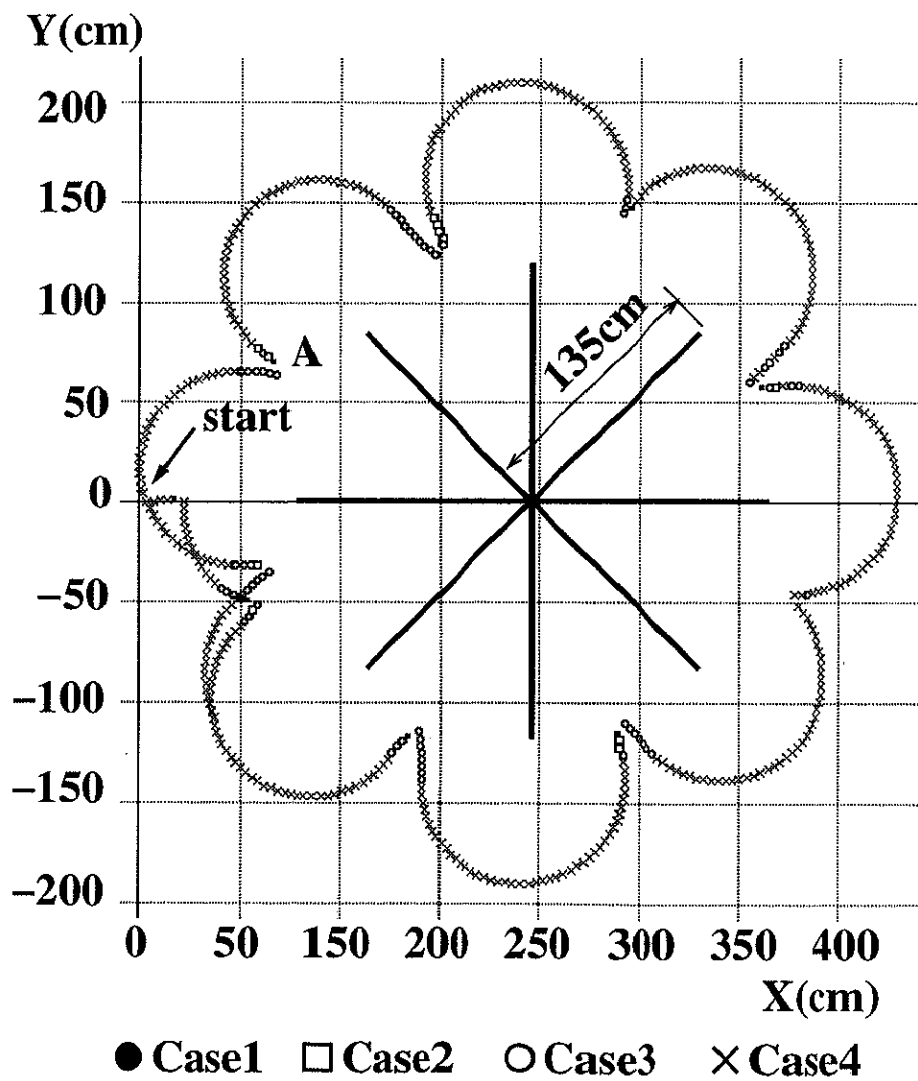


図 4.14: 大きい“\*”型対象物の外壁に沿って走行させた場合 (実機実験)

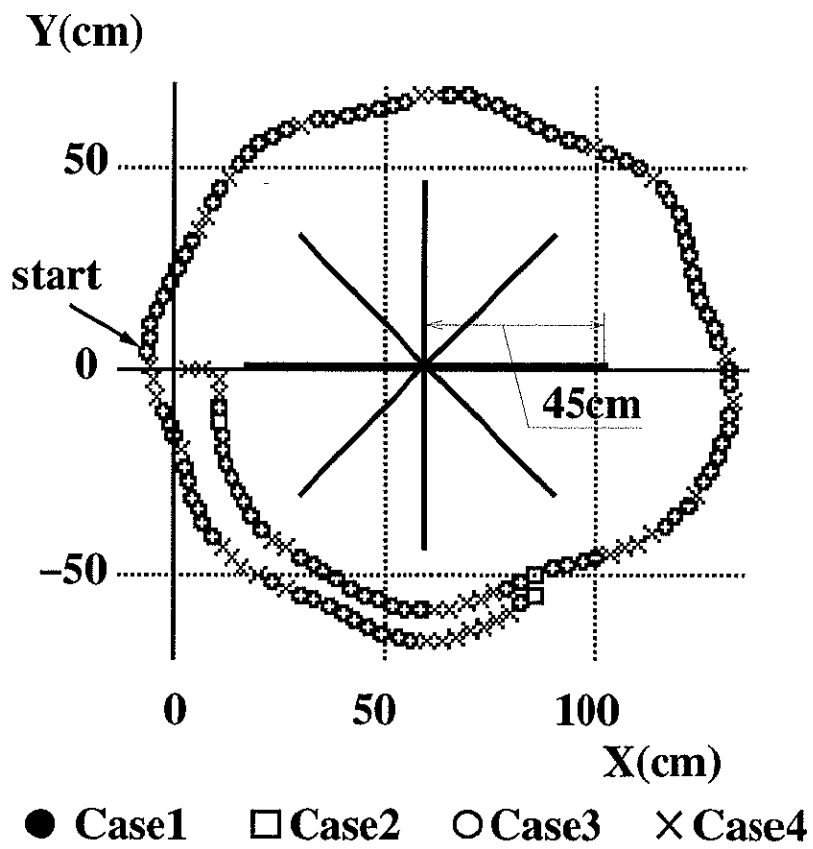


図 4.15: 小さい“\*”型対象物の外壁に沿って走行させた場合 (実機実験)

#### 4.4.4 壁沿い走行の限界の検討 2(ノコギリ状の凹凸のある壁)

前節の実験を行なうまでは、図 4.16 のように対象物に衝突するケースはまれであると予想し、多くの場合、図 4.11 の場合と同様にうまく対象物のまわりをまわることができると考えていた。しかし、実際には図 4.16 のような結果となるが多かった。そこで静止していてもノコギリ状の凹凸のある壁に対してどのようなことがおきるかを調べた。

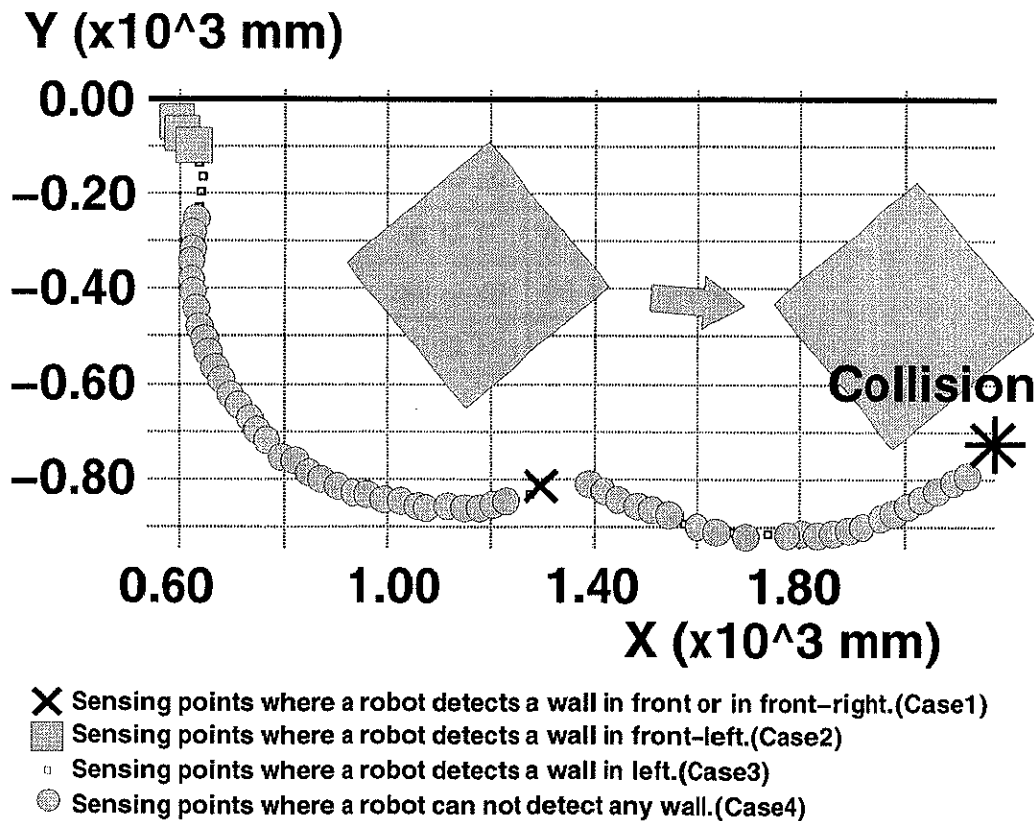


図 4.16: ゆっくり動く四角柱の壁に沿う実験 (実機実験)

ノコギリ状の凹凸のある壁を、上述のアルゴリズムによって壁沿い走行させた結果を図 4.18、4.19 に示す。本章で提案するアルゴリズムでこのようなノコギリ状の壁を壁沿い走行させた時、壁に衝突する場合と衝突しない場合があった。この壁の大きさは図 4.17 のように幅  $L$  の板を板と板とのなす角度が  $a$  になるようにし、表 4.1 に示すようないくつかの条件を与えた。

移動ロボットに、ロボット中心から壁までの目標距離 (D) を 50cm、速度を 30cm/sec とし左壁に沿うようにウォールフォロイングを行なわせた。

以下の図 4.18、4.19 にその実験例を示す。図にはオドメトリにより得られたロボットの位置情報と、アルゴリズムの Case1 から Case4 までのいずれが実行されているかを示す。

表 4.1: Result of experiments

L [cm]	a [deg]	H [cm]	結果
45	180 ~ 130	90 以上	衝突せず
	120 以下	80 以下	衝突した
90	180 ~ 60	90 以上	衝突せず
	50 以下	80 以下	衝突した
135	180 ~ 40	90 以上	衝突せず
	30 以下	80 以下	衝突した

これらの結果から、本アルゴリズムでこのようなノコギリ状の壁をウォールフォロイングさせた時、衝突するかどうかは板の幅 L や板のなす角度 a によらず壁の頂点間距離 H が 80cm 以下の場合に衝突することがわかった。

これは次のような理由によると考察できる。

図 4.20 のような環境では、ロボットが壁 A に沿って走行し、b 点に達したとすると、アルゴリズムの Case4 により点 b を中心に半径 50cm の円弧に追従する。その場合超音波の鏡面反射の性質により、ロボットのセンサが  $50\text{cm} + \alpha$  以内に対象物を検出するのはロボットに垂直な壁が存在した時である (図 4.20 中の a 点)。したがって、その点に達する前にロボットのボディが壁に接触してしまう場合、衝突は避けられないことになる。仮にアルゴリズムの Case4 における追従すべき円の半径を変化させても、衝突するかしらないかの頂点間距離の限界が変化するのみであろう。したがって、ここで提案したアルゴリズムを利用する限り、ここで示した衝突問題は回避できないと考えられる。



#### 4.5 4章のまとめ

本章においては、12方向の距離情報を得るようなソナーリングを移動ロボット「山彦」上に実装し、実環境における壁の形状に制限を設けずに、壁に沿って走行する方法を提案した。この壁沿い走行の方法は、壁から鏡面反射された超音波が検出されている限り、実機実験により安定に壁沿い走行を達成することができることがわかった。しかし、どんな壁の形状でも壁沿い走行が可能であるかといえば否であり、これがうまくゆかない実験例も示した。超音波距離センサは安価であり、対象物に反射した超音波を受信してから距離値を得るまでの計算機の負荷はほとんどないので、よく利用される。また実験用に販売されている移動ロボット NOMAD や Denning robot[10]においてもこの種のソナーリングが搭載されていることも多い。このような多数の超音波センサを用いるソナーリングを用いれば、ほとんどすべての環境でセンサ情報に基づく壁沿い走行が可能であると一般には考えられがちである。本章で示された実験結果はそのような考え方には落ちがあることを示したものであり、ソナーリングによって対象物が検出できない場面があることを示している。このことは、どんなにアルゴリズムを工夫しても、壁の形状や一辺の壁の長さを限定しない限り、本質的に対象物を見落とす可能性があることを示している。ソナーリングを用いて壁沿い走行を破綻なく行なう場合には、その走行環境の対象物の形状にも注意を払わねばならないことを示した。

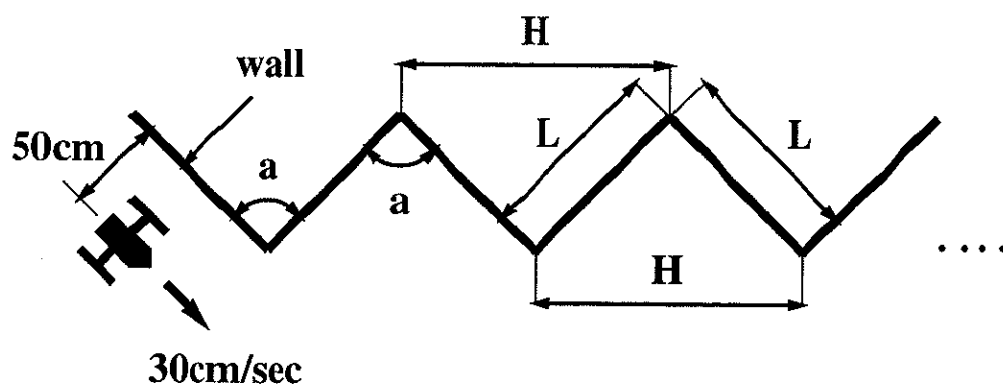


図 4.17: ノコギリ状の凹凸のある壁に沿って走行させる場合の実験環境

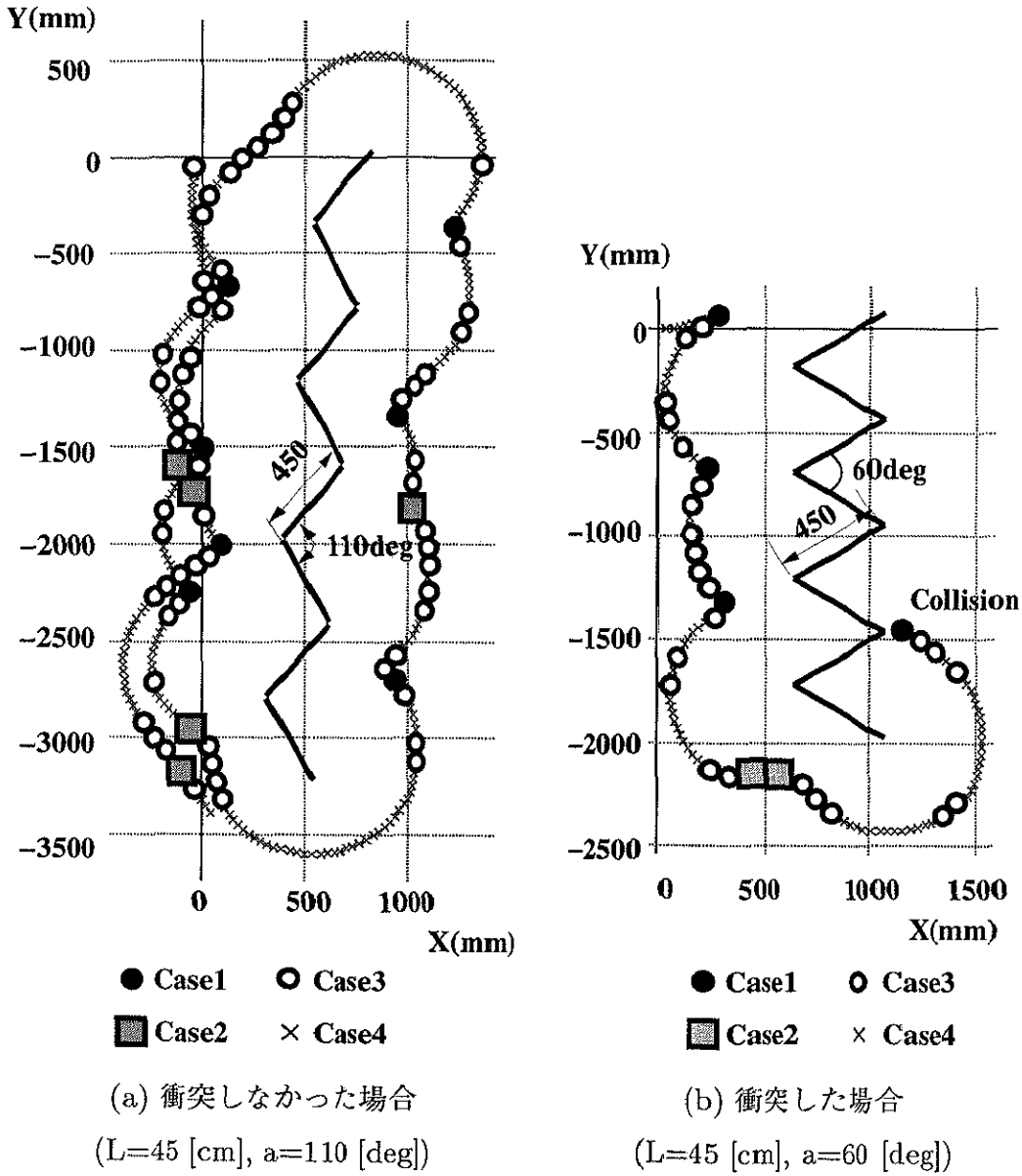
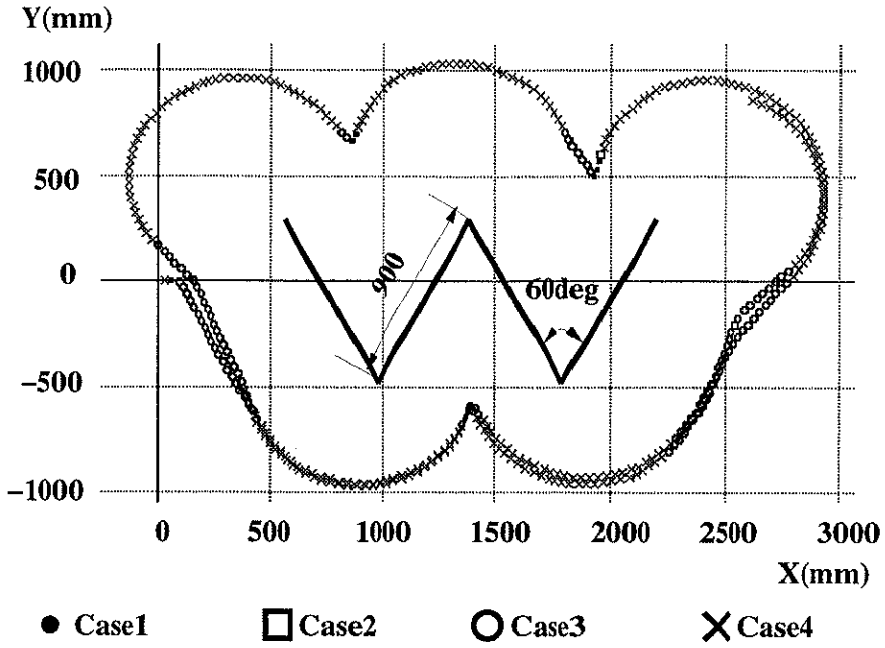
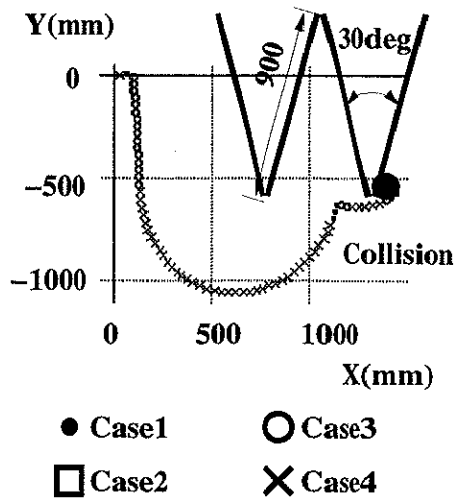


図 4.18: zigzag shape ( $L=45$  [cm])



(a) 衝突しない場合 ( $L=90\text{cm}$ ,  $a=60\text{deg}$ )



(b) 衝突する場合 ( $L=90\text{cm}$ ,  $a=30\text{deg}$ )

図 4.19: zigzag shape ( $L=90\text{cm}$ )

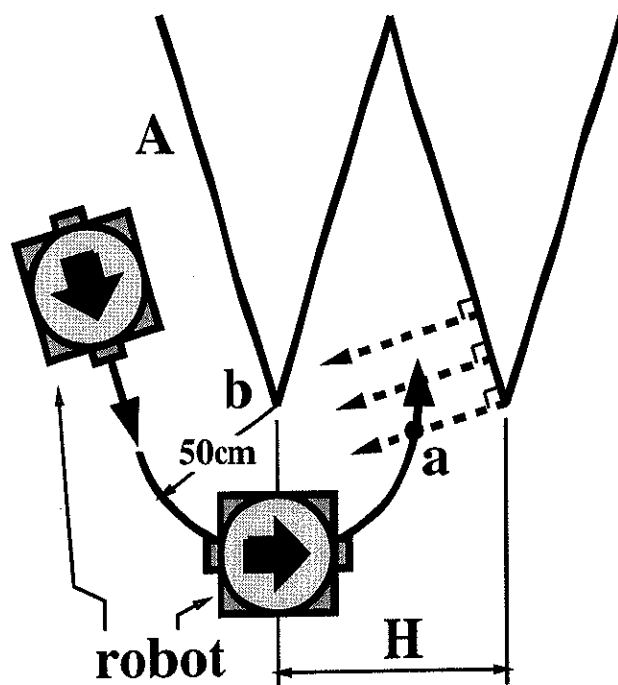


図 4.20: 衝突の検討

## 第 5 章

# ソナーリングによる壁沿い走行制御と位置制御に基づく移動ロボットのナビゲーション

### 5.1 はじめに

本章では、第 4 章で述べたソナーリングを用いたセンサベースな壁沿い走行と従来からよく用いられている地図と位置(座標値)に基づく走行とを適宜切り換えて利用することによる、移動ロボットのナビゲーションについて取り扱った。

自律移動ロボットに屋内の環境でナビゲーションを行なうこと、すなわち現在位置から目的地まで自律的に到達させることを考えた場合、まず考えられるのは、第 1 章でも述べたように、走行環境のモデルを詳細に持ち、目的地まで到達できる走行予定経路を設計し、さらにこの経路を環境中に固定した座標系に関して記述しておくことである。移動ロボットは自己位置を常に推定しながら走行予定経路の上を通過しているかを確認しながら進む(地図・位置ベースな走行)。一方、第 4 章では、超音波距離センサに基づいて、いろいろな形状の壁に沿って走行するアルゴリズムを開発し、実験用自律移動ロボットにそのアルゴリズムを実装して実験を行なった。この手法を利用すると、

(a) ロボットが地図に従って走行中に予定経路上に地図にない障害物を発見した場合、ロボットは障害物を見つけた時点で一旦停止し、ついでその障害物を例えば右側に見ながらそれに沿って壁沿いに走行することによりその障害物を回避し、障害物に

沿った経路が予定された経路と交差した時に再び予定経路に乗り換え、位置ベースな走行を行なうことができる場合がある。すなわち、その障害物がロボットを通さないような通路のふさぎ方をしていなければ、壁沿い走行によって障害物を回避できる。

(b) 例えばゆるやかに連なり、超音波距離センサが常にその壁の存在を知ることができる壁に沿って走行すれば目的地に到達できるような環境では、ロボットに壁沿い走行をさせることにより、詳細な地図を予め与えることなしに目的地まで到達させることができる。

そこで、筆者は、第1章で述べた位置ベース走行と、第4章で述べた壁沿い走行とを使い分け、区間や場合によって走行方法を切り換えながら自律移動ロボットのナビゲーションを実現した。すなわち、地図に基づいて走行予定経路を設計できる場所ではそれに従い移動ロボットを位置ベース走行させ、地図上には記述されない予期せざる障害物がある場合や、詳細な地図はないが壁沿い走行可能な場所では移動ロボットを壁沿い走行させることにした。このような壁沿い走行と位置ベース走行を切り換えることによるナビゲーションを実現するため、ここでは特に、(1) マスターモジュール上でユーザがロボットの行動プログラムを記述する際、壁沿い走行と位置ベース走行の切り換えをユーザが簡単に指定できるようなコマンド系を定義し、実装した。また、(2) 壁沿い走行が10m程度にわたって続いた後、位置ベース走行に切り換えて、その後の走行予定経路に入る際の問題点を指摘し、これに対する対処法を示した。

## 5.2 Spur コマンド系の拡張 (壁沿い走行コマンドの追加)

移動ロボットの行動を記述しようとする者が、前述の2通りの走行方法の双方を組み合わせてロボット自体の行動をプログラミングするためには、マスターモジュール上に実装される行動プログラムにおいて、その双方を指示できる走行制御コマンド系が必要になる。この目的のため、従来の位置に基づく走行制御のため第3章で述べた Spur コマンド系を、壁沿い走行の指示もできるようにした。具体的には、次の形式の関数を既存の Spur 関数群に追加した。既存の Spur 関数については第1章で述べたので、ここでは拡張された部分である壁沿い走行コマンドだけについて示す。

### 1. 壁沿い走行コマンド形式: Spur\_follow(x)

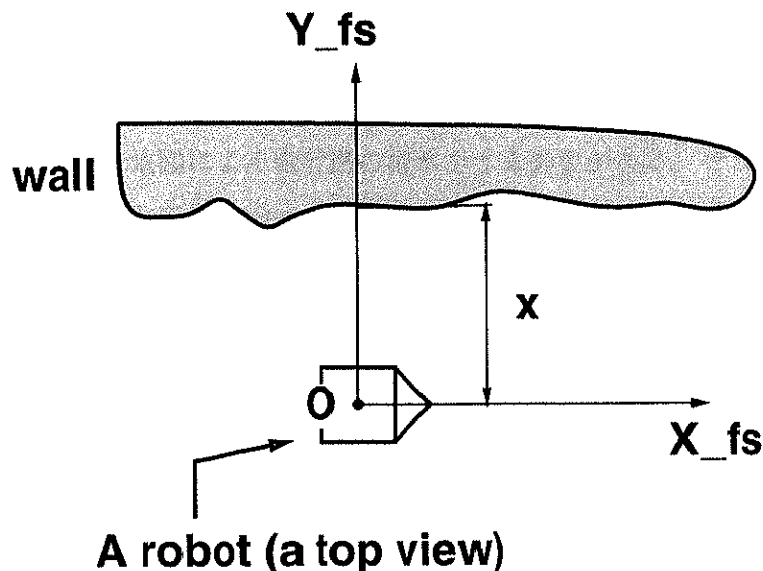


図 5.1: 壁沿い走行コマンドのモデル

この関数がマスターモジュール上の行動記述プログラムで発行されると、図 5.1 のようにロボット中心から壁表面までの目標距離を  $x[\text{mm}]$  として、移動ロボットに壁沿い走行をさせる。右壁に沿う時は  $x$  に負の値を与える。このコマンドが発行された場合、第 4 章で示した壁沿い走行アルゴリズムに従い、Spur コマンドがマスターモジュールから発行される。ただし、行動記述プログラムにおいて、壁沿い走行中に直線追従コマンドなどの違った行動記述を示す Spur コマンドが発行された場合は、それを監視するモニタープロセスにより、壁沿い走行を中止し、指示された行動記述に従うことにする。

### 5.3 壁沿い走行コマンドの使用例

前節で定義した壁沿い走行コマンドの使用例として、図 5.2 のような蛇行経路走行と図 5.4 のような障害物回避の場合を示す。図 5.2 の原点にロボットが Y 軸の方向に向かっておかれている状態から走行を開始し、y 軸上を 3000mm 進んだところで一旦停止し、右に  $90^\circ$  転回し、そこから壁沿い走行をおこなわせた。この行動記述は、図 5.3 に示したとおりである。この記述言語は Robol/0 [52] を用いている。



### 5.3.1 蛇行経路

蛇行経路を壁沿い走行を用いずにオドメトリだけを用い、その曲線を円弧と直線に近似し軌跡追従制御により走行することは可能である。しかし、そのためには正確な位置推定とていねいな地図による走行路の指示をする必要がある。それを避けて蛇行経路を走行する方法として壁に沿って走行の方が容易であろう。そこで、ナビゲーション中に蛇行経路を走行しなければならない場合には、壁に沿って走行することにした。

ナビゲーションレベルのプログラムは拡張されたSpurコマンド系を用いてROBOL/0言語により記述された。拡張されたSpurコマンド系では壁沿い走行を1つの関数で記述することができ、プログラミングをする上で有用であった。実機による実験結果を図5.2に示す。また、図5.3に、ロボット言語ROBOL/0により蛇行経路走行の行動を記述したプログラム例の抜粋を示す。

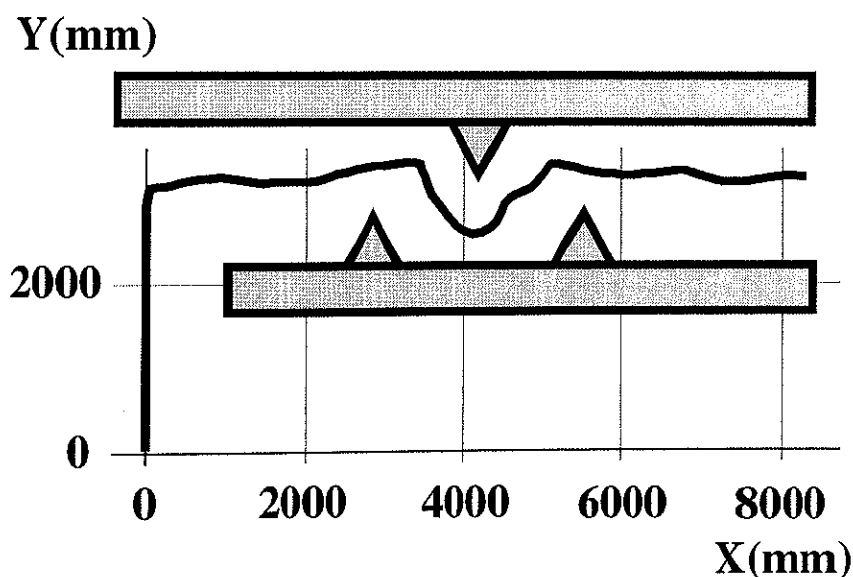


図 5.2: 蛇行経路走行 (実機実験)

```

MODESDEF
STARTM straight1
MODE straight1
    
```

```
    Spur_line_LC(0,0,0);                /* Y軸に沿って直線追従 */
WAIT
    WHEN(Spur_over_line_LC(3000,0,0)==1) /* もし、3m 走行したら、 */
        NEXT stop                       /* MODE stop に飛ぶ */
MEND
MODE stop
    Spur_stop_Q();                      /* 停止 */
WAIT
    WHEN(Spur_near_vel(0,5) ==1)       /* もし、速度が0になったら、 */
        NEXT spin                       /* MODE spin に飛ぶ */
MEND
MODE spin
    Spur_spin_LC(-90);                 /* 時計回りに 90° 回転 */
WAIT
    WHEN(Spur_near_ang_LC(-90,5) ==1) /* もし、X軸と平行になったら */
        NEXT follow                    /* MODE follow に飛ぶ */
MEND
MODE follow
    Spur_follow(600);                 /* 左壁から 60cm 離れて壁沿い走行 */
WAIT
    WHEN(Spur_over_line_LC(0,-7000,-90)==1) /* もし、7m 走行したら、 */
        NEXT straight2                /* MODE straight2 に飛ぶ */
MEND
MODE straight2
    Spur_line_LC(3000,0,-90);         /* 直線追従 */
WAIT
    WHEN(Spur_over_line_LC(0,-8000,-90)==1) /* もし、1m 走行したら、 */
        NEXT stop_end                /* MODE stop_end に飛ぶ */
MEND
MODE stop_end
```

```
    Spur_stop_Q();                               /* 停止 */
    death();                                     /* 終了 */
    WAIT
MEND
MODESEND
```

図 5.3: 蛇行経路走行 (行動プログラムの抜粋)

### 5.3.2 障害物回避

移動ロボットがナビゲーション中に超音波センサにより予定経路上を監視し障害物を発見した場合に、壁沿い走行を用いて障害物を回避するプログラムを作成した。障害物回避の実機による実験結果を図 5.4 に示す。

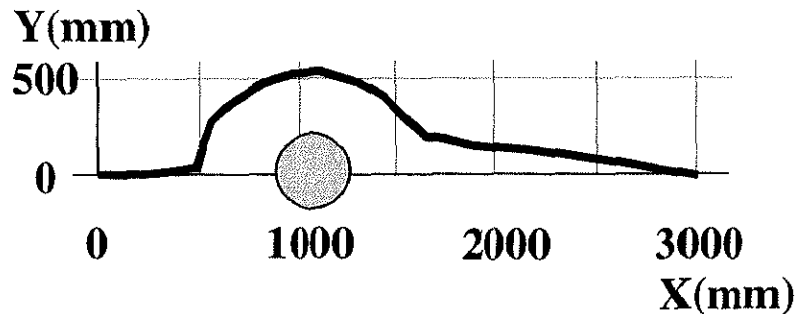


図 5.4: 障害物回避 (実機実験)

図 5.5 に、ロボット言語 ROBOT/0 により障害物回避走行をする行動を記述したプログラムを示す。

```
MODESDEF
STARTM straight
MODE straight
    Spur_line_LC(0,0,0);                          /* X 軸に沿って直線追従 */
```

```

WAIT
  WHEN(Spur_over_line_LC(2970,0,0)==1) /* もし、ゴール付近に到着 */
    NEXT stop_end /* MODE stop_end に飛ぶ */
  WHEN(前方の超音波距離が 50cm 以下) /* もし、超音波距離が 50cm 以下 */
    NEXT kaihi /* MODE kaihi に飛ぶ */
MEND
MODE kaihi
  Spur_follow(-500); /* 右壁から 50cm 離れて壁沿い走行 */
WAIT
  WHEN(Spur_over_line_LC(2970,0,0)==1) /* もし、ゴール付近に到着 */
    NEXT stop_end /* MODE stop_end に飛ぶ */
  WHEN(予定経路と平行) /* もし、現在の移動ロボットの方向と */
    /* ゴールまでの方向が平行になったら、 */
    NEXT stop_spin /* MODE stop_spin に飛ぶ */
MEND
MODE stop_spin
  Spur_stop_Q(); /* 停止 */
WAIT
  WHEN(Spur_near_vel(0,10)==1) /* もし、速度が 0 になったら、 */
    EXEC Spur_spin_FS(90); /* 時計回りに 90° 回転 */
    NEXT straight2 /* MODE atraight2 に飛ぶ */
MEND
MODE straight2
  Spur_line_LC(ゴールの方向を計算した値); /* ゴールの方向に直進 */
WAIT
  WHEN(ゴールの 3cm 手前に来たかどうかを調べる)==1(来た))
    /* もし、ゴール付近に到着したら、 */
    NEXT stop_end /* 停止 */
  WHEN(前方の超音波距離が 50cm 以下) /* もし、超音波距離が 50cm 以下 */
    NEXT kaihi /* MODE kaihi に飛ぶ */

```

```

MEND
MODE stop_end
    Spur_stop_Q();                /* 停止 */
    death();                       /* 終了 */
WAIT
MEND
MODESEND
    
```

図 5.5: 障害物回避 (行動プログラムの抜粋)

#### 5.4 壁沿い走行から位置ベース走行に切り換える方法

位置ベース走行と壁沿い走行を切り換える場合、位置ベース走行から壁沿い走行への切り換えは、特に問題となる点はない。しかし、壁沿い走行から位置ベース走行へ切り換えるためには、ロボットは位置ベース走行を開始する時点で自分の推定位置を有していねばならない。なぜならば、壁沿い走行から位置ベース走行に切り換える際に、予定された走行経路上に乗らなければならないからであり、またその切り換えを行なう位置自体もロボット自身も持っている自己の推定位置にたよらざるを得ないからである。そのためには、壁沿い走行中もロボット上では常時位置の推定が行なわれている必要がある。すなわち、壁沿い走行中もオドメトリにより、常時位置を推定しておく必要がある。

ところが、壁沿い走行が(筆者の用いたロボットの場合)、約 10m 程度続くと、オドメトリの累積誤差が増大しすぎて、スムーズに位置ベース走行に切り換えることができないことがあった。そのため、以下のような切り換え方法をとる。

長距離の壁沿い走行を終了して位置ベース走行にもどるとき、オドメトリの累積誤差が大きくなっているため、このままではその後の地図・位置ベース走行のための予定軌跡に乗ることは困難である。すなわち、オドメトリによる推定位置が壁沿い走行から位置ベース走行に切り換える地点に達したとしても、実際にはオドメトリの累積誤差のためにその位置が期待した位置と著しくずれてしまう。さらにこのずれをとまなまま位置ベース走行を行なっても、本来の予定経路から、オドメトリの誤差分

だけずれて通過することになる。そこで、位置ベース走行にうつる直前でオドメトリの累積誤差をランドマークにより補正してやることにした。具体的には、図5.6のようなフラットな壁が1m以上あるような場所を、切り換えエリアに選び、壁沿い走行中にそのフラットな壁を利用して累積誤差を補正してから、位置ベース走行に切り換える。この補正には、ロボットの正確な位置がわかるような枠組であればどのようなセンサを用いてもよいが、ここではソナーリングを用いているので、位置ベース走行と同様な手法でオドメトリの推定位置を補正する。すなわち、壁沿い走行を終了する付近に平面的な壁があるので、これに沿ってロボットを走行させる。壁沿い走行のアルゴリズムでは、ソナーリング上の超音波センサの距離情報から常に与えた距離  $D$  で壁との間の距離を保つようにしている（第4章）、当然この距離  $D$  が壁との間で保たれている、そこでこの事実を使ってPOEM-II(第3章)による位置修正を行ない（図5.6）、できる限り、誤差の小さな推定位置を求める。こうすれば、位置ベース走行に戻った時に予定軌跡に正しく乗ることが期待できる。図5.7は、実験的にフラットな壁を一定の距離60cmで右壁沿い走行させた時の移動ロボットの軌跡である。図5.7のように実験的にフラットな壁を壁沿い走行させた時のロボットの振舞いから、与えた壁との距離  $D$  に対してどれ位の幅であらつくかを調べておき、これから距離  $D$  に対する分散  $\text{dev}$  を求める。これらを用いて、壁沿い走行終了地点付近で最尤推定を行なう。

## 5.5 ナビゲーションの実験例

図5.8のような実験環境でロボットをスタート（左下）からゴール（右上）までナビゲーションさせる実験を行なった。この実験環境においては、途中で壁づたいに走行すれば通過できる経路が存在している。しかしその壁の形状は不定型なので地図を作成することは困難であり、壁沿い走行をさせる。

スタート地点ではまず最初ロボットに位置ベース走行で目的の経路を走行させる。曲線的壁のある経路の手前で壁沿い走行に切り替え、ロボットに壁沿い走行を行なわせる。曲線的な経路を抜けた後、フラットな左壁の位置でロボットのオドメトリ累積誤差を減少させる。その後地図・位置ベースな走行に切り替えゴールに到達させる。実験で使用したナビゲーションプログラムを状態遷移表現を用いて図5.9に示す。

移動ロボットは第4章で用いたものを使用しており、その時の実験風景が図5.11で

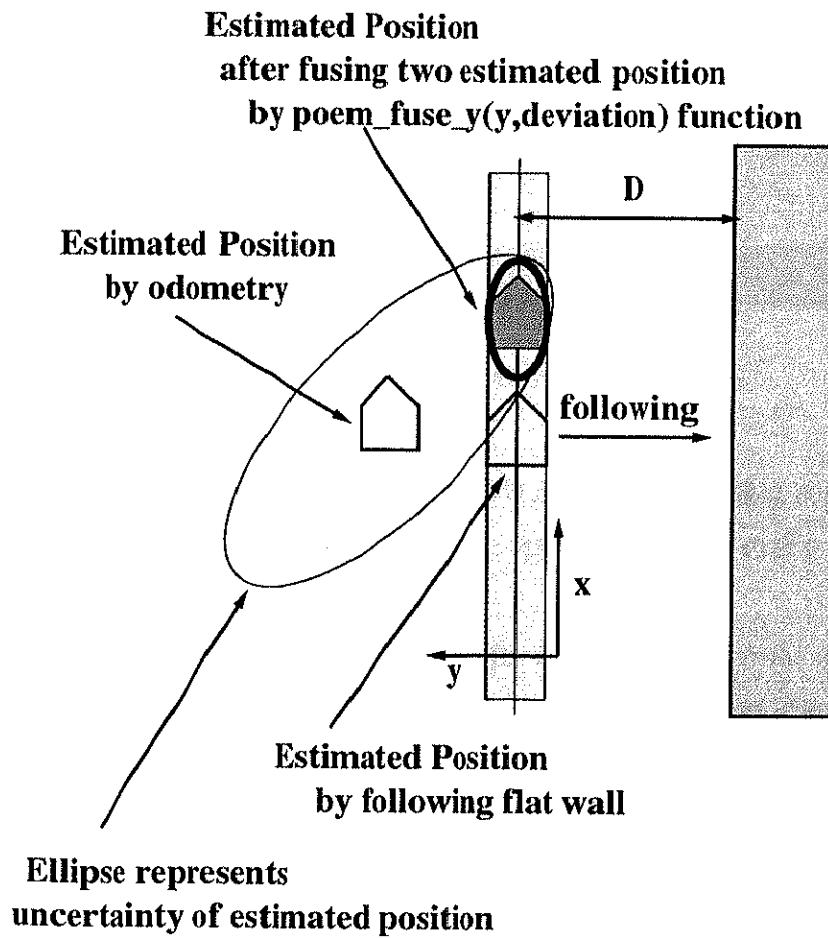


図 5.6: 壁沿い走行から位置ベース走行への切り換え方法

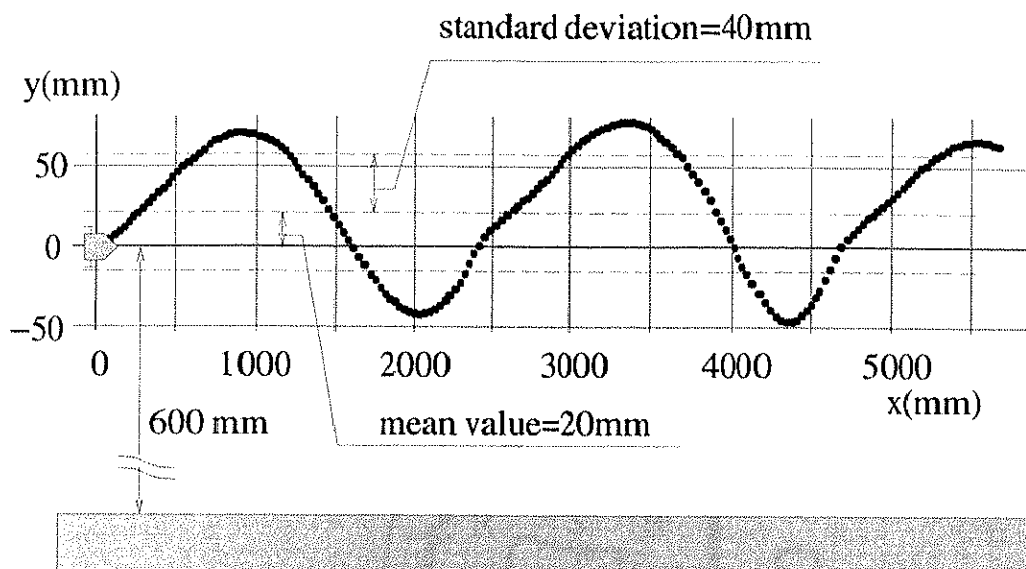


図 5.7: フラットな壁を壁沿い走行させた時の軌跡

ある。また、この実験において壁沿い走行の時のロボットと壁の距離は 60[cm] に保つように指定した。またそのときの不定型な壁は、45[cm] × 60[cm] の板を垂直に立てて並べることにより実現している。

ロボットの推定位置の誤差楕円の推移を図5.10 に示す。この図から、壁沿い走行時の推定位置の誤差は累積しどんどん増加していくが、壁沿い走行終了地点近くのフラットな壁の横を壁沿い走行した際にこのフラットな壁のあらかじめ既知である位置と壁から 60[cm] を保つように指定した事実、および、図 5.7からわかるように、フラットな壁を壁沿い走行した際の 60[cm] からのずれに対する標準偏差がほぼ 4[cm] であるということ、および、オドメトリによる推定位置および誤差分散を用いて最尤推定をほどこした結果、推定位置の誤差楕円は小さくなることが確認された。

このことから、長い距離を壁沿い走行していても、行動アルゴリズムからこの付近を通る時には、ほぼ必ずここを通るという知識を利用することにより、オドメトリで得られた推定位置の累積誤差を小さくできることを確認した。したがって、壁沿い走行から、大きな破綻なくふたたび地図・位置ベース走行にもどることができた。



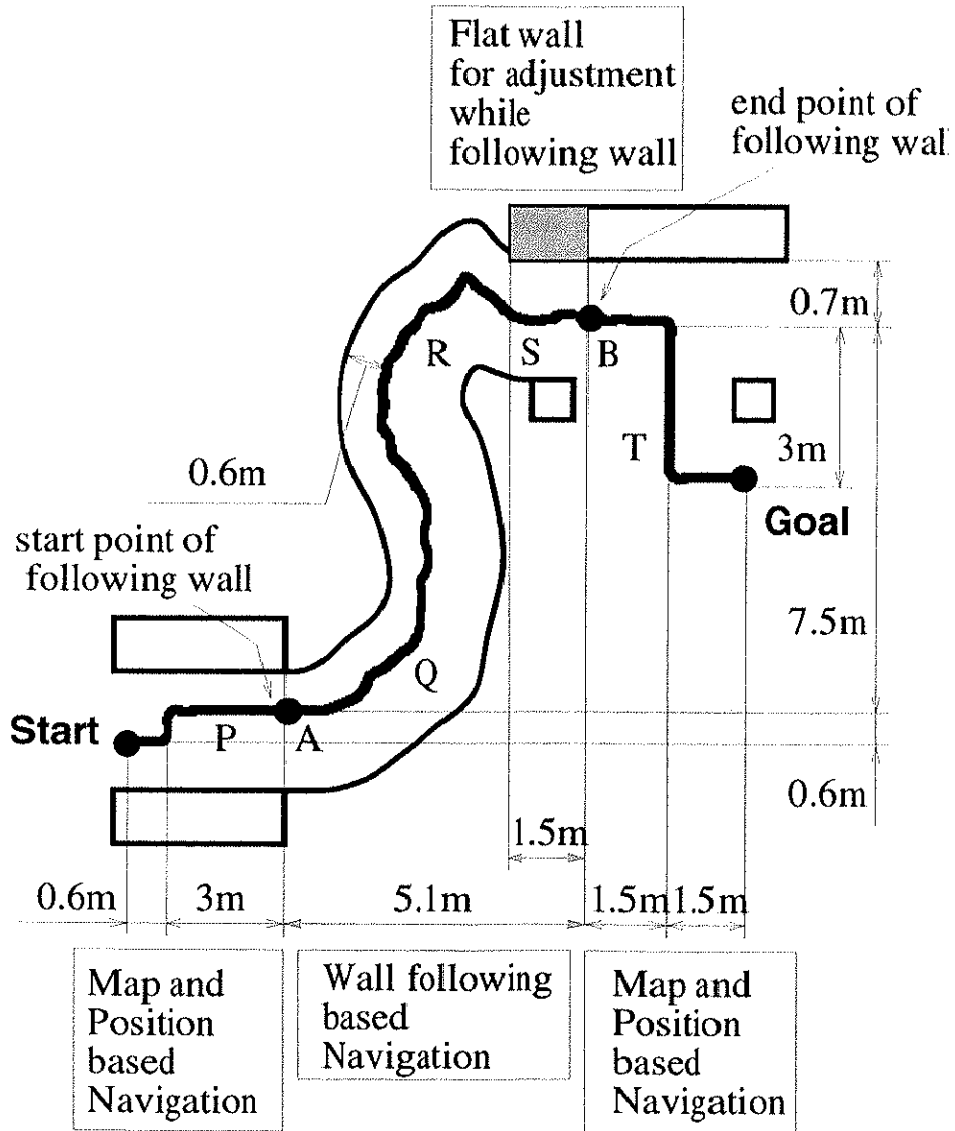


図 5.8: ナビゲーション実験 (環境)

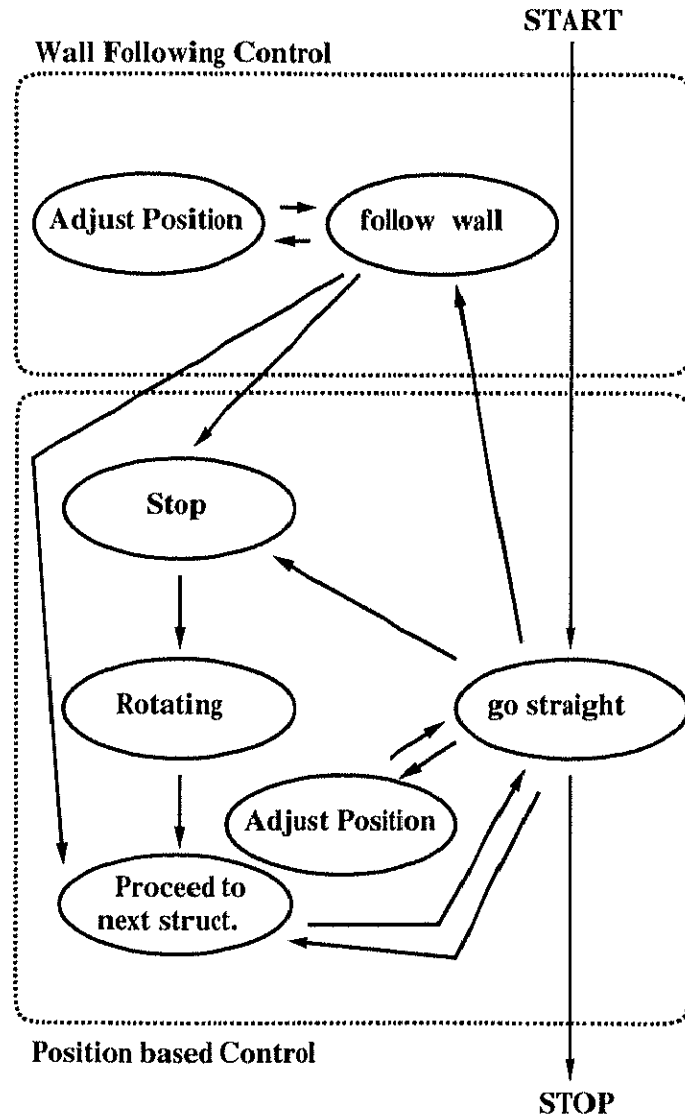


図 5.9: ナビゲーション実験 (プログラムの状態遷移)

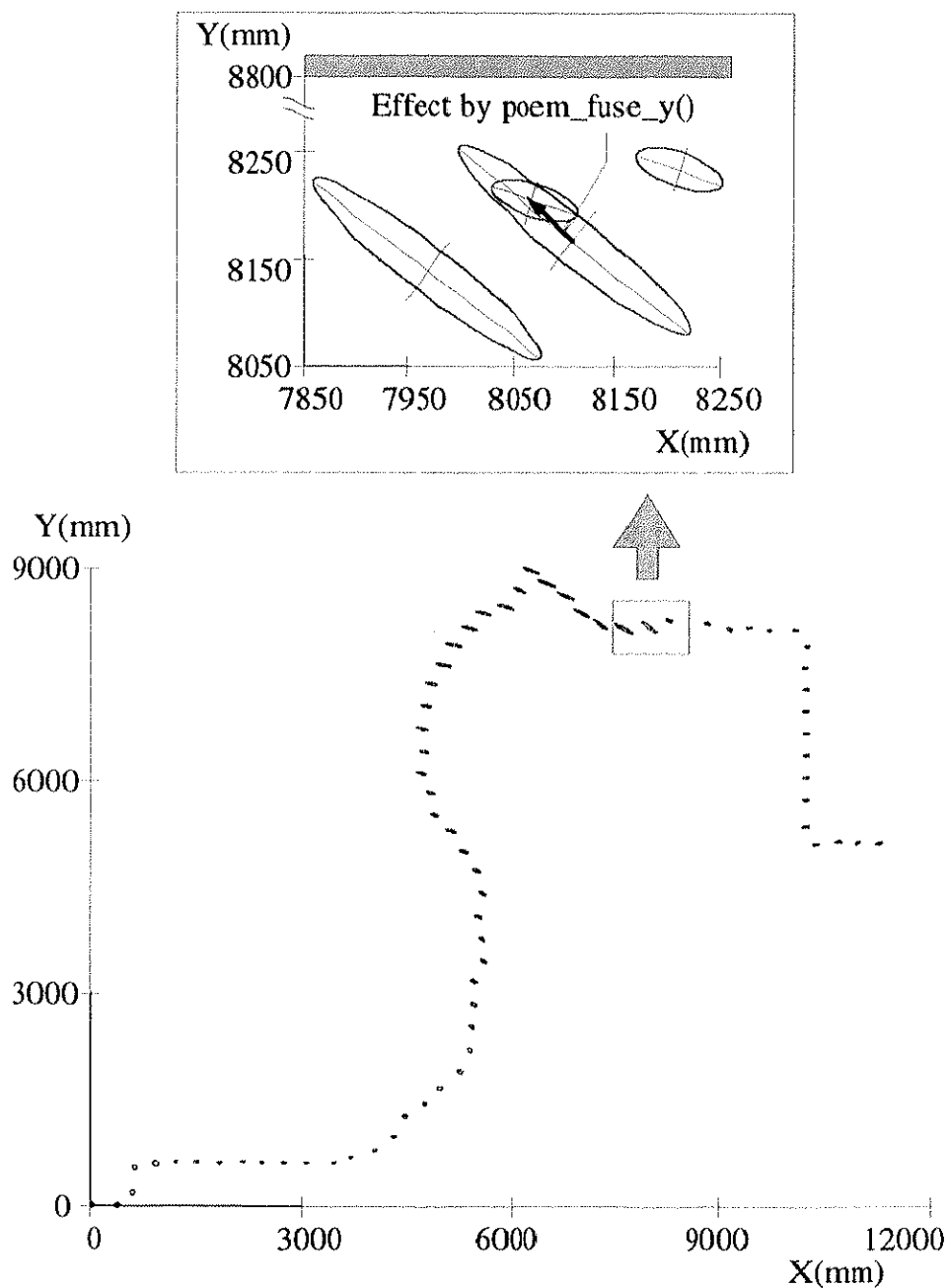
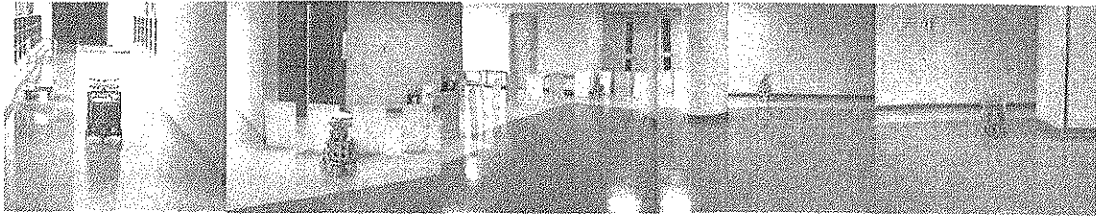


図 5.10: ナビゲーション実験 (移動ロボットの推定自己位置の誤差楕円)



(a) 図 5.8 の P 点における写真 (b) 図 5.8 の Q 点における写真 (c) 図 5.8 の R 点における写真 (d) 図 5.8 の S 点における写真 (e) 図 5.8 の T 点における写真

図 5.11: ナビゲーション実験 (実験風景)

## 5.6 5 章のまとめ

ここでは移動ロボットを 2 次元の座標として表現される自己の位置推定値に基づいて走行制御する方法と、4 章で示した壁沿い走行する方法を切り換えて使用するナビゲーションを実現した。また、ユーザが簡単に壁沿い走行を指定できるような走行制御コマンドを追加した。そして、いくつかの実験を行い、期待通りロボットが走行したことを確かめた。本コマンドシステムを用いることにより、経路の途中にジグザグな通り道がある場合には壁沿いを走行するという指令をロボットに与えるだけで複雑な経路の指定を避けることができた。また、移動ロボットがナビゲーションをしている時に、移動ロボットの進路軌道上に予期せず障害物などがあつた場合一時的に移動ロボットに壁沿い走行をさせて障害物回避をさせることが実現できた。

ここでは、(1) 壁沿い走行と位置ベース走行の切り換えをユーザが簡単にプログラミングできるようなコマンドを定義・実装したことが成果であり、(2) 壁沿い走行が 10m 程度続いた後、位置ベース走行に切り換える際、累積したオドメトリ誤差を修正する方法を示し、その効果を確認したことが成果である。この (1)(2) により位置 (座標値) ベース走行と、センサベースな壁沿い走行を切り換えながら自律移動ロボットのナビゲーションを実現した。

## 第 6 章

# 超広角カメラと扇状光を用いた移動ロボット用レーザレンジセンサの開発

### 6.1 はじめに

本章で説明する研究の目的は、扇状に広げたレーザ光線と超広角カメラを組み合わせた三角測量の原理に基づく移動ロボット用レーザレンジセンサを開発することである。

本章以下、第 7 章、第 8 章では、主として筑波大学第 3 学群棟内の廊下、すなわち、比較的整理はされているが、ゴミ箱等の箱状の物体も比較的多く置かれ、また独立した直方体または円柱状の柱も存在するような環境において、移動ロボットをセンサベースな行動原理によって走行させることにテーマを設定した。第 4 章において取り扱ったソナーリングによってもこのような環境における走行はある程度可能である。しかし、第 4 章で示したように、超音波が対象物の表面において鏡面反射する性質からかなりの頻度で対象物が検出できない場面のあることが確認された。そこで、超音波センサよりも確実に移動ロボットの周囲の対象物の形状を検出できるセンサとして扇状光と TV カメラを用いるレーザレンジセンサを開発した。

廊下環境において、センサベースな行動原理に基づいて障害物回避や壁沿い走行を行なうには、ロボットの正面だけでなく、左右の距離情報も同時に得られるよう、広い角度にわたってロボット周囲の物体の形状の情報が得られるレンジセンサが必要になる。すなわち、ロボットの前方から水平に  $\pm 130^\circ$  程度の角度範囲内においてロ

ロボット上に固定された座標系に関するロボット周囲の対象物の表面の各点の座標、あるいはその点までのロボットからの距離情報が得られるセンサが必要である。開発したレーザレンジセンサはこのような角度範囲において実用的な時間でこれを測定でき、また小型の移動ロボットに搭載できる安価でコンパクトなものとなっている。

## 6.2 レーザレンジセンサのハードウェア

本研究において開発したレーザレンジセンサのハードウェア構成を示す。小型で軽量のものとするため、細い一条のレーザ光線を機械機構を用いて走査するものではなく、扇状にレーザ光線を投射して、その光が対象物にあたった部分(光切断面)を TV カメラでとらえ、三角測量の原理で計測を行なうものとした。一方、ロボットの正面より水平方向にして $\pm 130^\circ$  にわたり計測可能な角度範囲をもたせるためには複数の TV カメラをもちいなければならないが、その数をできるだけ少なくするために、水平視野が広い超広角カメラを使用することにした。具体的には、図 6.1 に示すように、移動ロボットの頂部に 3 個の超広角 CCD カメラ(ひとつのカメラにつき水平視野角 $108^\circ$ )を配置した。3 個の超広角 CCD カメラは、ロボットの上部高さ 70 cm の位置に、前と左右方向に向けて下向きふ角約 $40^\circ$  で図 6.1 のように取り付け、それぞれのレンズには、680nm ~ 2600nm の波長の光を通す光学フィルタを取り付けた。図 6.2 のようにロボットの腰部に 5 個の赤色扇状レーザ投光器(波長:685nm)を配置した。厚さ 5mm 程度のレーザ光が扇状に約 $90^\circ$  に広がるものを用い、床面に平行にロボットの前方の周囲約 $260^\circ$  をこの扇状光がカバーするように各々の投光器を配置している。TV カメラからの画像を処理し、光切断面とロボットとの間の距離あるいは光切断面のロボット上に固定した座標系に関する座標位置を計算する画像処理モジュールは、1 個のカメラ毎に 1 個の画像処理ボード(CPU: T805)を割り当てた。この画像処理モジュールの内部構成は図 6.3 のようになっている。これは、フレームメモリ(64Kbytes, 256 × 240)を有し、同期信号生成回路を通して NTSC 信号をフレームメモリに取り込む機能を有した。また、デバッグなどのためにフレームメモリの内容を確認するため、フレームメモリの内容を NTSC 信号に変換しモニターできる機能を有した。内蔵の RAM(1024K)を用いて歪み計算や距離計算を行ない、計算結果を一定サイクル(約 600msec)毎に Dual Port Memory(2Kbytes)に書き込ませた。この画像処理モジュールを 3 枚用いて、本レーザレンジセンサのセンサモジュールを構成した。前と左右のカメラによって得ら

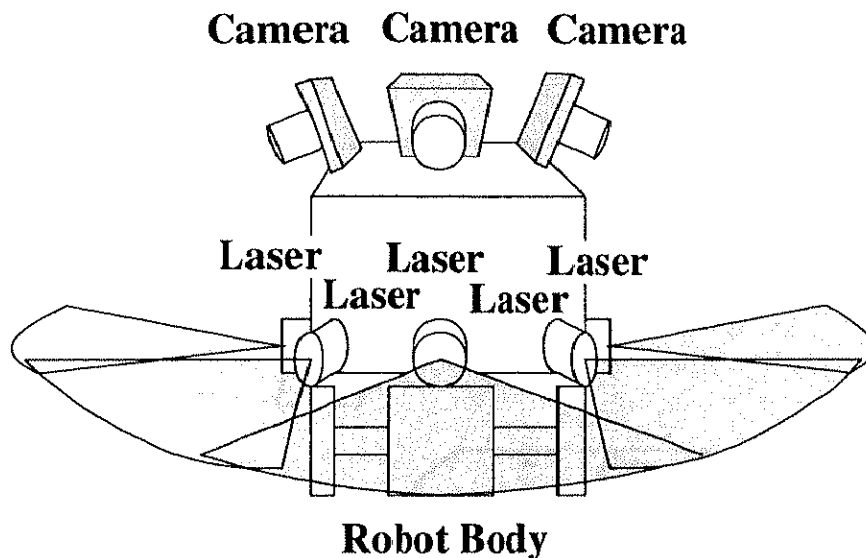


図 6.1: センサの概観図

れた画像に基づいて計算された距離は各々独立にそれぞれの画像処理モジュールの Dual Port Memory に書き込む。この Dual Port Memory は、SIMP(第 3 章参照)の実装となっている。すなわち、マスターモジュールからは各々の距離情報を SIMP となる Dual Port Memory を介して読みとることができる。移動ロボット「山彦」上に実際に実装した様子を図 6.4 に示す。

### 6.3 距離計算の原理

扇状光が対象物にあたった光切断面とロボットの間の距離あるいはロボット上に固定された座標系に関する光切断面の座標を計算する方法について示す。この計算方法は一般的に知られている三角測量に基づくものである。すなわち、床面と平行に広がったレーザ光線による扇状光を対象物に当て、その様子をカメラで撮像し、得られた画像上で輝度の高い画素の位置をもとに扇状光の平面内における対象物の位置座標およびそれまでの距離を計算する。

画像上の輝度の高い画素の位置がレーザ平面上のどの位置に対応するかを求める式は、いま、カメラが図 6.5 のようなピンホールモデルに従うものとする、次の (6.1), (6.2) 式で示される。レーザ平面を  $xy$  平面、平面内における対象物の光切断面の位置を  $P(X_p, Y_p)$ 、

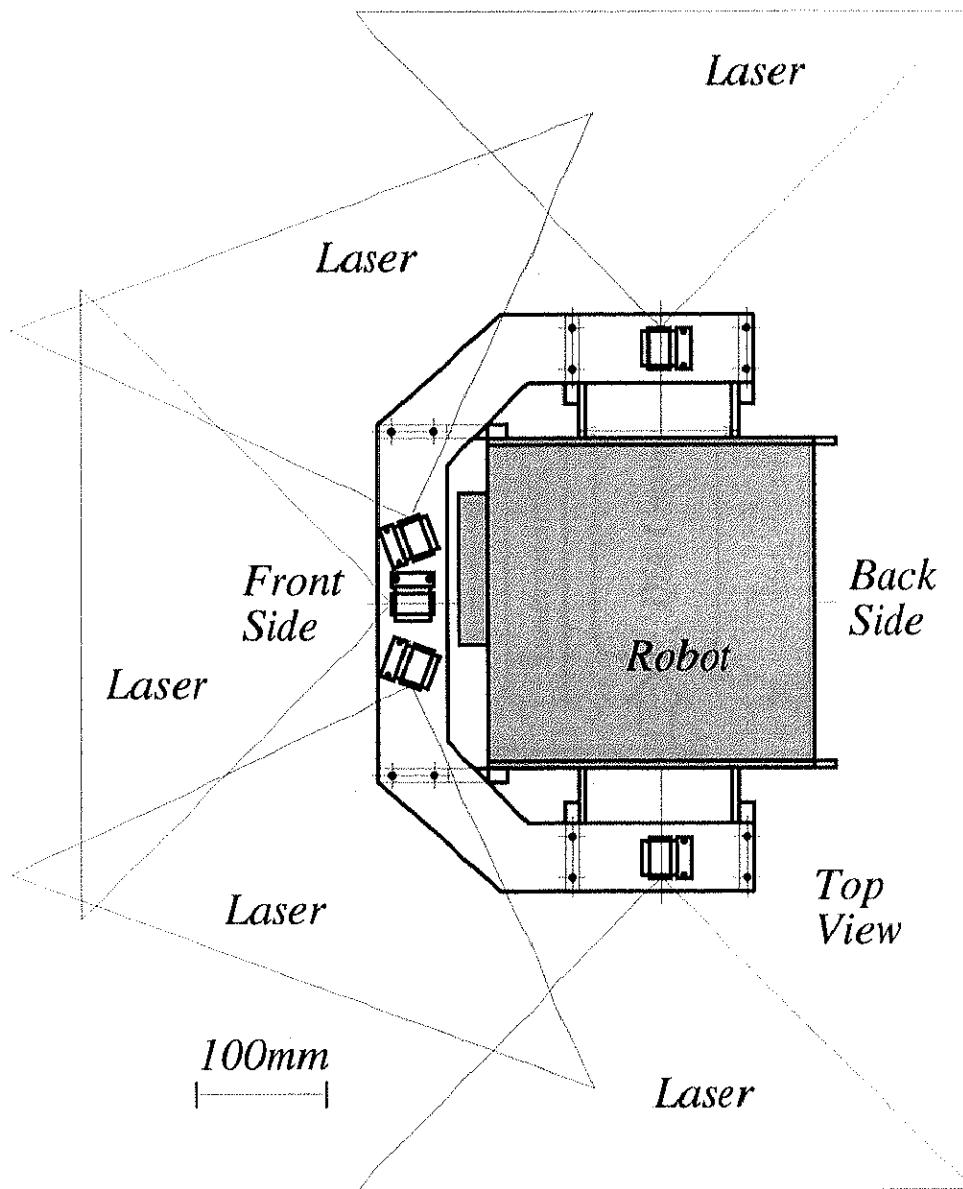


図 6.2: レーザ取付位置 (平面図)



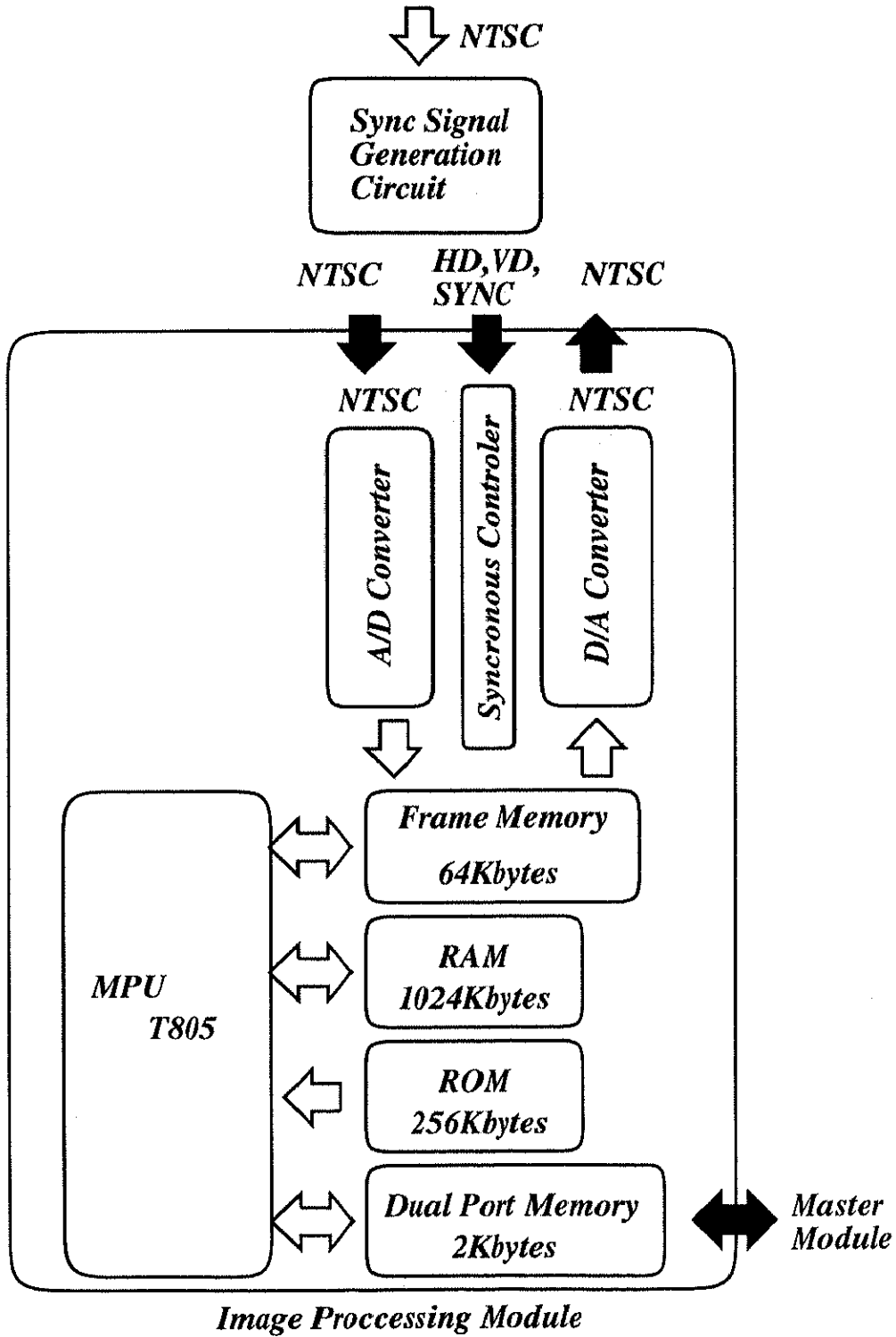


図 6.3: 画像処理モジュールの内部構成

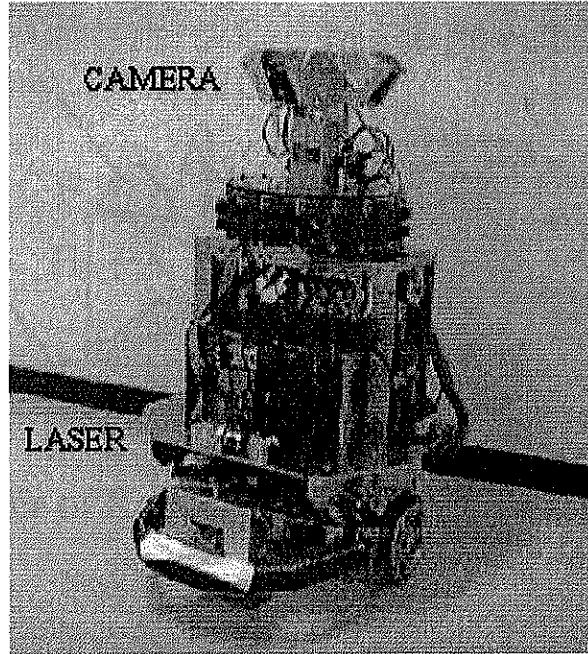


図 6.4: センサが実装された移動ロボットの写真

ピンホールモデル画像の平面を  $uv$  平面、輝度の高い画素の位置を画像平面上で  $P'(U_p, V_p)$ 、基線長を  $H$ 、焦点距離を  $f$ 、レーザ平面と光軸中心線との交わった点を  $O$ 、ロボット中心点と  $O$  の間の距離を  $J$ 、カメラのふ角を  $\theta$  とした。

$$X_p = \frac{\frac{V_p}{f}H + J\sin(\theta)(\sin(\theta) - \frac{V_p}{f}\cos(\theta))}{\sin(\theta)(\sin(\theta) - \frac{V_p}{f}\cos(\theta))} \quad (6.1)$$

$$Y_p = \frac{-\frac{U_p}{f}H\sin(\theta)}{\sin(\theta)(\sin(\theta) - \frac{V_p}{f}\cos(\theta))} \quad (6.2)$$

この時、ロボット中心から対象物までの距離  $D$  は、以下の式 (6.3) で表される。

$$D = \sqrt{(X_p + J)^2 + Y_p^2} \quad (6.3)$$

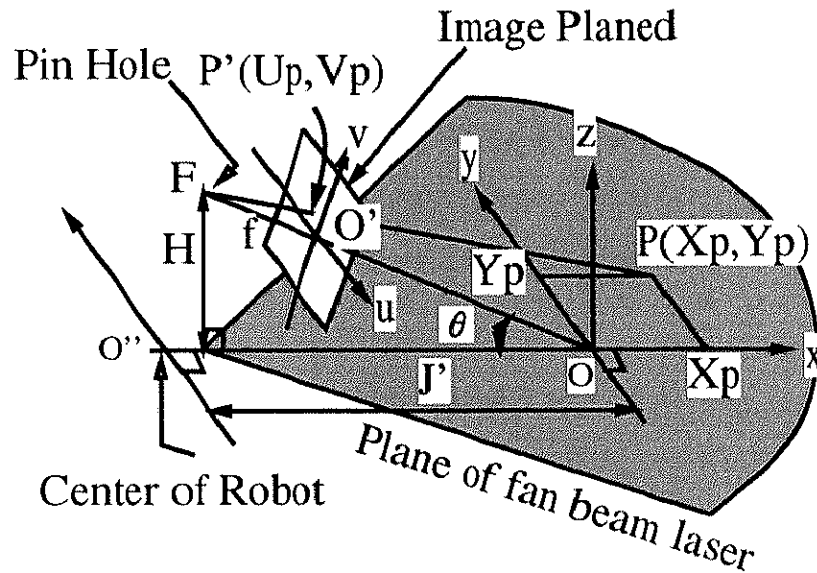


図 6.5: レーザ平面とピンホールモデル画像の幾何関係

#### 6.4 超広角カメラ画像の歪み補正

6.1節で述べたように、ここでは超広角カメラを用いた。しかし、図 6.6(左)の市松模様を超広角カメラで撮像した例でもわかるように超広角カメラで撮像された画像は歪むため、この画像のまま、6.3節で示したピンホールカメラモデルに基づく測距公式を用いることはできない。この歪みを補正してピンホールカメラモデルを用いることができるようにする必要がある。超広角カメラの画像上の点  $(r, \phi)$  をピンホールカメラモデル画像上の点  $(R, \Phi)$  (どちらも極座標表示 [図 6.6]) に変換する歪み補正の変換式は次のようになる [47]。ただし、各座標軸の原点はカメラの光軸が画像平面を通る点である。また、 $a_1, a_2$  はカメラ固有の定数であり、事前にキャリブレーションによって求めておく。

$$\Phi = \phi \quad (6.4)$$

$$R = a_1 \tan(a_2 r) \quad (6.5)$$

この時、超広角カメラ画像上の極座標点  $(r, \phi)$  は、超広角カメラ画像上のデカルト

座標点  $(u_p, v_p)$  により式 (6.6) で表される。ただし、ここで  $\arctan 2(y, x)$  は、値域を  $[-\pi, \pi)$  に拡張した  $\arctan(\frac{y}{x})$  とする。

$$(r, \phi) = (\sqrt{u_p^2 + v_p^2}, \arctan 2(v_p, u_p)) \quad (6.6)$$

式 (6.4), (6.5) から、ピンホールモデル画像のデカルト座標点  $(U_p, V_p)$  を超広角カメラ画像の極座標点  $(r, \phi)$  を用いて表すと、式 (6.7) のようになる。

$$\begin{aligned} (U_p, V_p) &= (R \cos(\Phi), R \sin(\Phi)) \\ &= (a_1 \tan(a_2 r) \cos(\phi), a_1 \tan(a_2 r) \sin(\phi)) \end{aligned} \quad (6.7)$$

## 6.5 画像の走査方法 (扇型走査)

本章で扱っているレーザレンジセンサは、床面と平行な扇状光を対象物に当て、その反射光をカメラで撮像し、得られた画像を走査して、輝度の高い画素を探し、その画素を扇状光が対象物にあたっている部分とみなして、その位置からレーザ光が対象物にあたっている位置を求めることにより、対象物までの距離を求めている。画像を走査する一般的な方法は、画像全部の画素を例えば左端上部から順番に走査する方法である。しかし、一例として図 6.7 のような状況において撮像されたロボット前方を向いた超広角カメラからの画像図 6.8 をみると、一般的な左端上部からの行順次の走査ではうまくゆかないことがわかる。図 6.8 をみるとわかるように明るい歪んだ台形状の部分 (壁にレーザ扇状光が直接あたっている部分) の両端に角 (つの) 状の線が見えている。これはたまたま図 6.7 で示した壁が光沢のある壁であったため反射光の虚像がみえているものである。この反射光の虚像は、ロッカーなどの金属性の光沢のある壁の場合や、床面がピカピカに磨かれている場合などにおきやすい。このような状況で一般的な走査を行なうと、この虚像も十分に明るいと、これが対象物体に直接扇状光があたっている部分と区別できず、結果として物体が無いところに、あたかも物があるかのように、計算されてしまう問題が発生する (図 6.11)。

そこで、この問題を解決するために以下のような画像走査方法を考案した。ただし、以下は前方のカメラから得られた画像の走査方法について例示するが、左右方向のカメラから得られた画像の走査方法もこれに準じる。また、ロボットの真上から見て右

方向を  $0^\circ$ 、前方を  $90^\circ$ 、左方向を  $180^\circ$  方向と定義する。

(1) 図 6.9 のようにロボットの真上から見て、 $0^\circ$  方向に対してロボットの中心から放射状に遠ざかるようにレーザ平面上の点を移動させ、その点に対応する画像上のピクセルの位置を式 (6.1), (6.2), (6.6), (6.7) を逆に解き求める。そして、レーザ平面上で放射状に伸ばした点に対応する画像上のすべての画素の座標位置を配列にテーブルとして記憶しておく (図 6.10)。

(2) さらに、 $0^\circ$  方向から  $180^\circ$  方向まで  $1^\circ$  ごとに (1) と同様に繰り返し、その配列に記憶する。ただし、上記 (1) とこのステップはレーザ投光器とカメラの配置を決定した後に行なう。

(3) 以上 (1)(2) は走査のための準備である。得られた画像からロボット周囲の壁や対象物までの距離を計算するたびに、上記の (1), (2) で用意した配列に格納された座標の順番に超広角カメラ画像を走査し、輝度値の急峻なピークを探し、そのピーク群の最大値の位置から、物体までの距離を計算する。

このような画像の走査法を、ここでは扇型走査と呼ぶ。扇型走査は、細いレーザビームを動径としてロボットの中心から床面に平行な平面内を回転させて走査するものと同等なものになっており、この走査によってひとつの動径が対象物にあたるがあるとすれば高々 1 回のみであることが保証される。実際、図 6.12 は、図 6.8 にロボット中心から扇状光を含む平面内で右斜め  $60^\circ$  方向に対する走査線を黒線で重ねた図である。この走査線上の輝度値は図 6.13 のようになり、急峻なピークが 2 つ表れる。扇型走査では、1 本の走査線上にピークは 1 つ以下であることが保証され、かつ、反射光の虚像は直接反射光より輝度が低いため、虚像を無視することができる。このようにして得られたピークから計算された対象物の位置座標値は図 6.14 のようになり、反射光の虚像の影響の無いほぼ正確なものとなった。

## 6.6 実機 (山彦) による環境の測定例

実際にこのセンサを図 6.15 のような廊下環境で使用した。図 6.16 のような三つのカメラから得られた画像をもとに計算した対象物の位置データを図 6.17 に示す。この図は、3 つのカメラ画像それぞれから、レーザ扇状光の対象物による反射位置を上で述べた扇型走査方法によって求めたものをレーザ扇状平面に平行な、ロボットに固定された座標系上にプロットしたものである。実際の壁や障害物の位置も図中に示した。

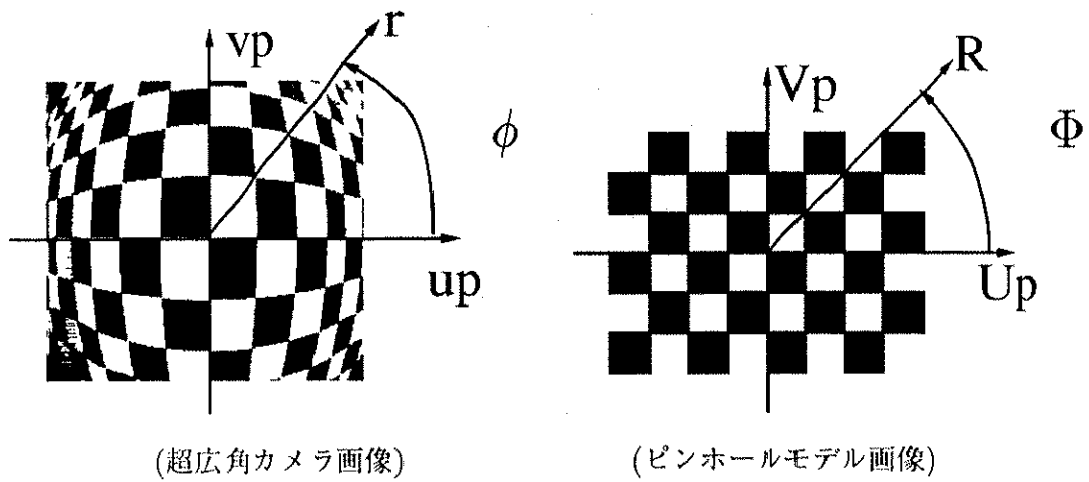


図 6.6: 超広角カメラで市松模様を撮像した場合の例

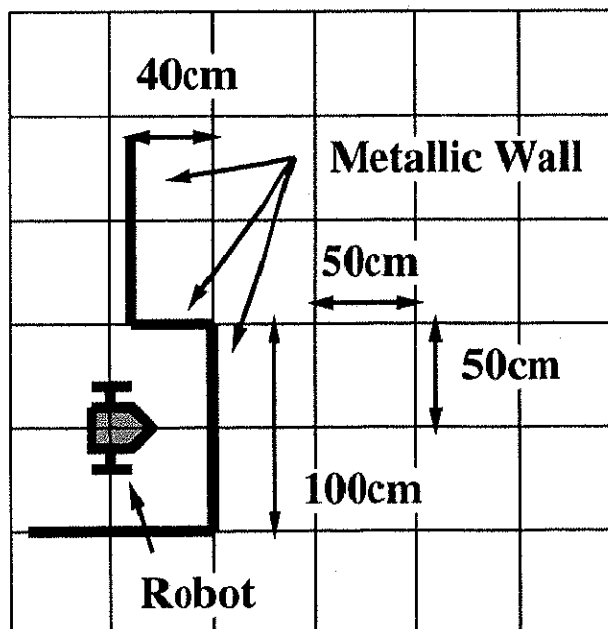


図 6.7: 光沢のある壁の配置

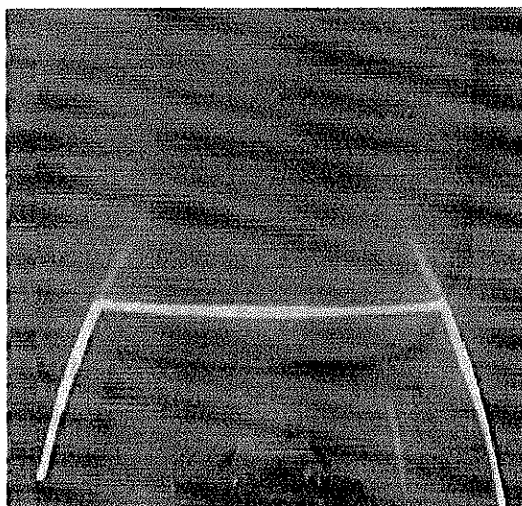


図 6.8: 前方の超広角カメラ画像

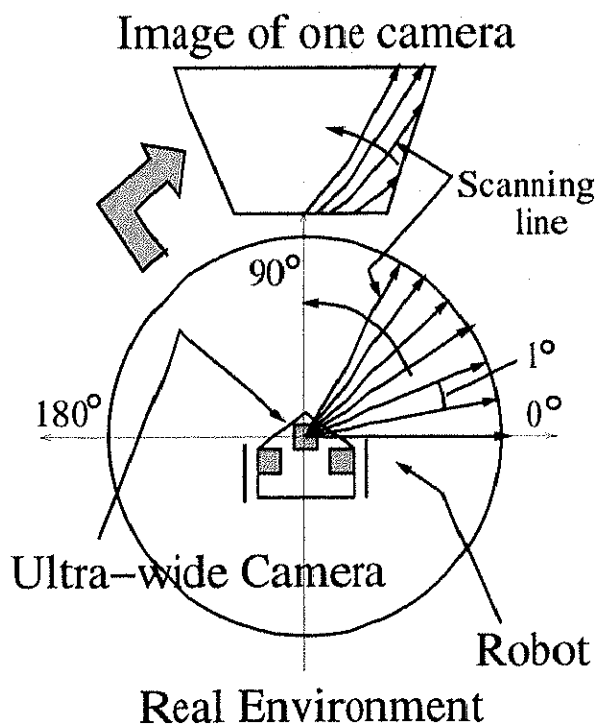


図 6.9: 画像の走査方法

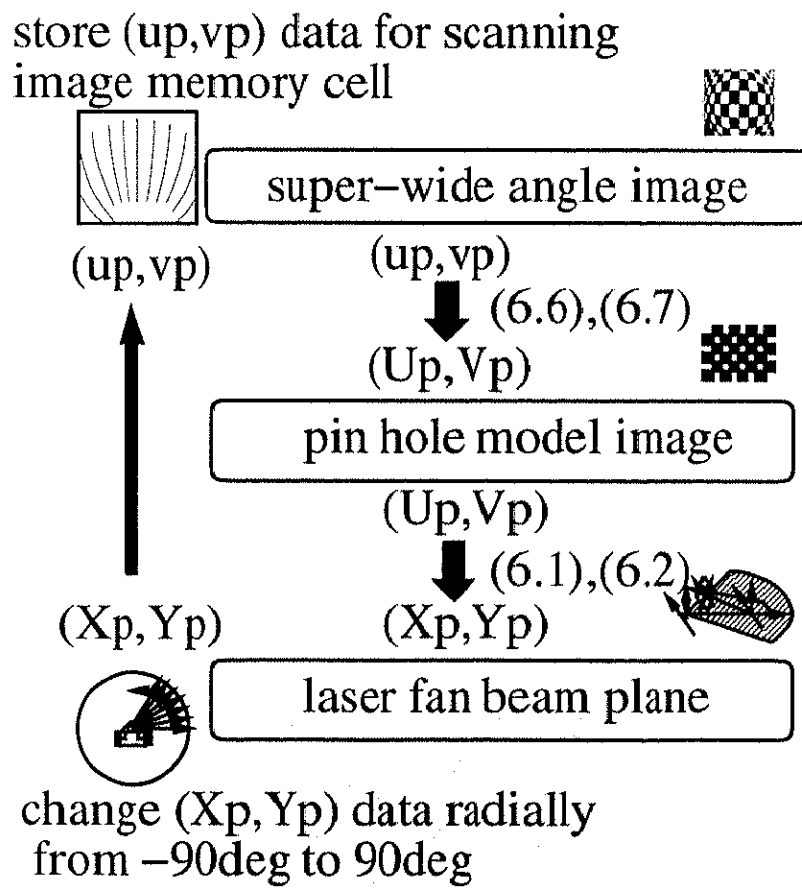


図 6.10: 画像の走査方法と距離計算処理



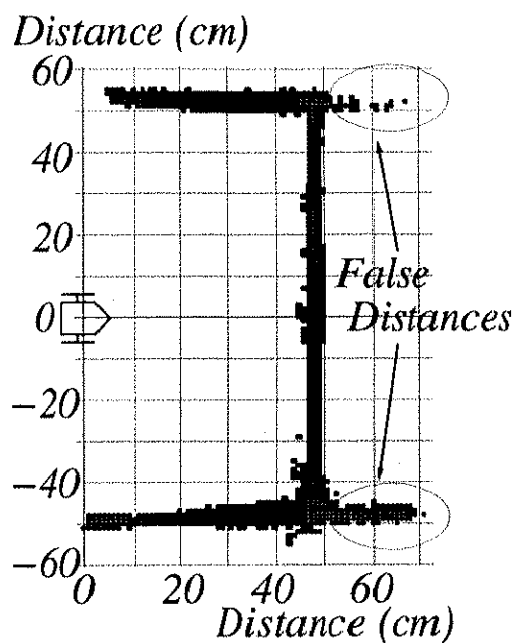


図 6.11: 画像全体の走査により得られた対象物表面のロボット上に固定された座標系からみた座標位置

Scanning line

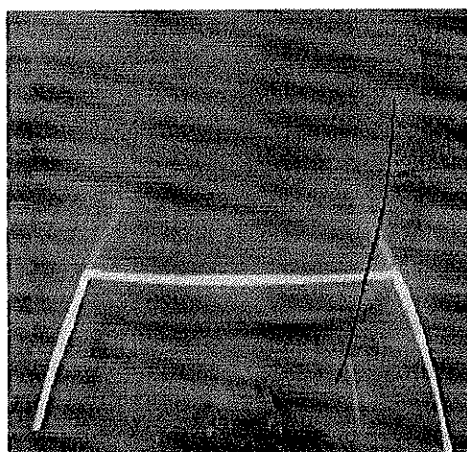


図 6.12: 前方の超広角カメラ画像 (黒線: 右斜め 60° 方向のスキャンライン)

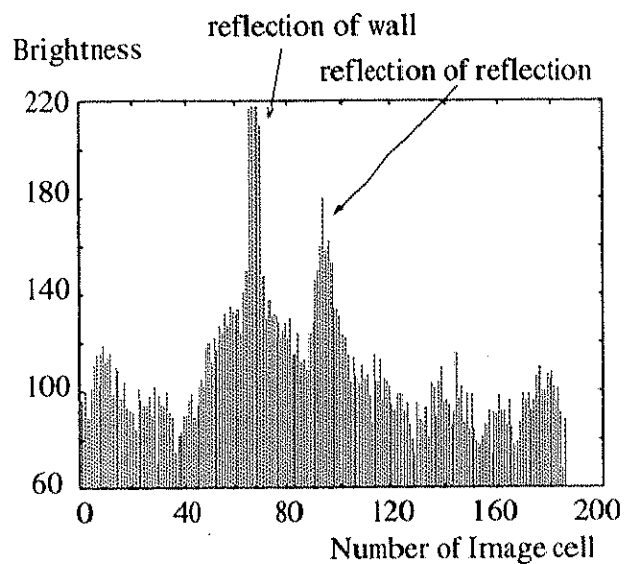


図 6.13: 右斜め 60° 方向のスキャンライン上のピクセル輝度値

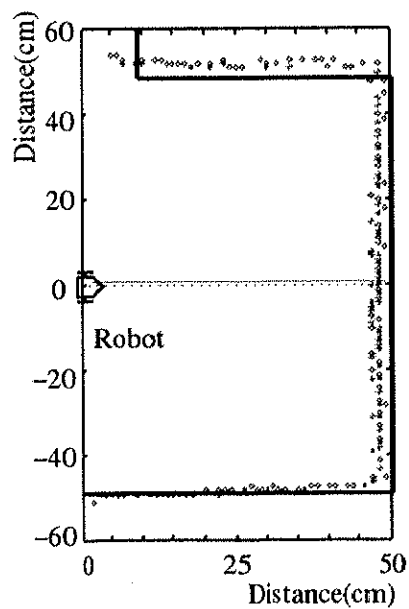


図 6.14: 扇型スキャン方法による距離測定結果

ただし、ロボットは図の右方が前方になるように置かれている。この結果から、ほぼ正確に対象物の位置データが得られていることが分かる。

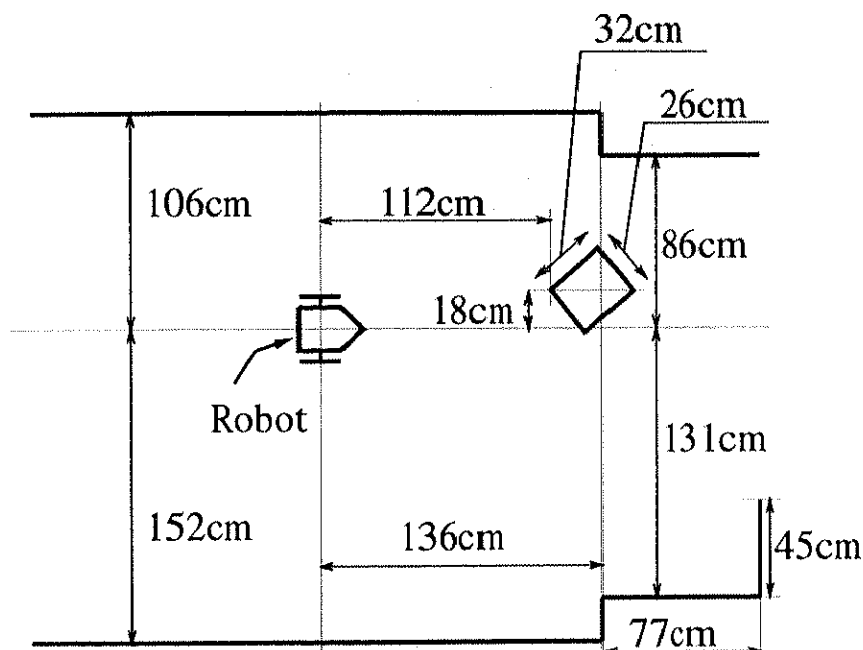


図 6.15: 実験環境(廊下)

位置の測定精度は毎回のキャリブレーションの程度に大きく左右されるが、ロボットから約 1m の距離にある対象物を測定する場合の位置測定誤差は距離誤差に換算しておよそ ±5cm、対象物が約 2m の距離にある場合で距離誤差にしておよそ ±10cm である。環境にもよるが、測定可能範囲はおよそ 3 ~ 4m である。処理時間は、現在のところ 0° から 180° を 1° 毎に距離を測定するのにカメラ 1 台あたりおよそ 600msec かかる。

## 6.7 6 章のまとめ

本章では、超広角カメラと扇状光レーザを用いたレーザレンジセンサの開発について述べた。センサの構造、センサによる対象物の位置計算方法、画像走査方法などについて検討を行い実装した。実際にこのセンサを廊下で使い、その実験結果から測距データがほぼ正確に得られていることがわかった。

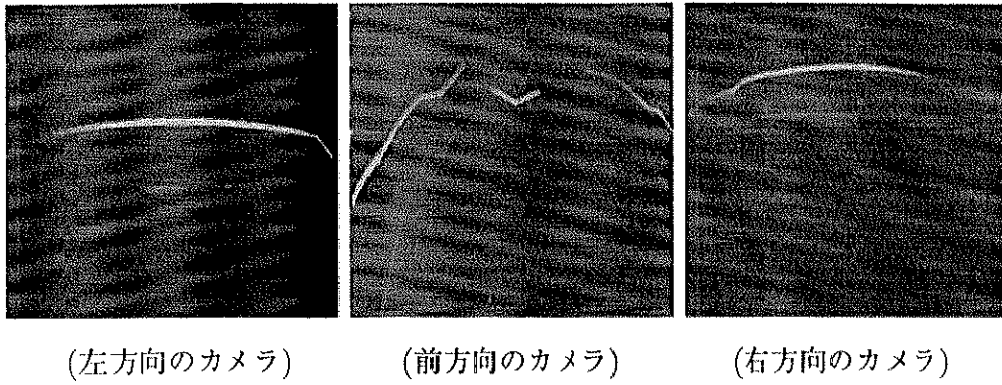


図 6.16: 得られた 3 方向の超広角カメラ画像

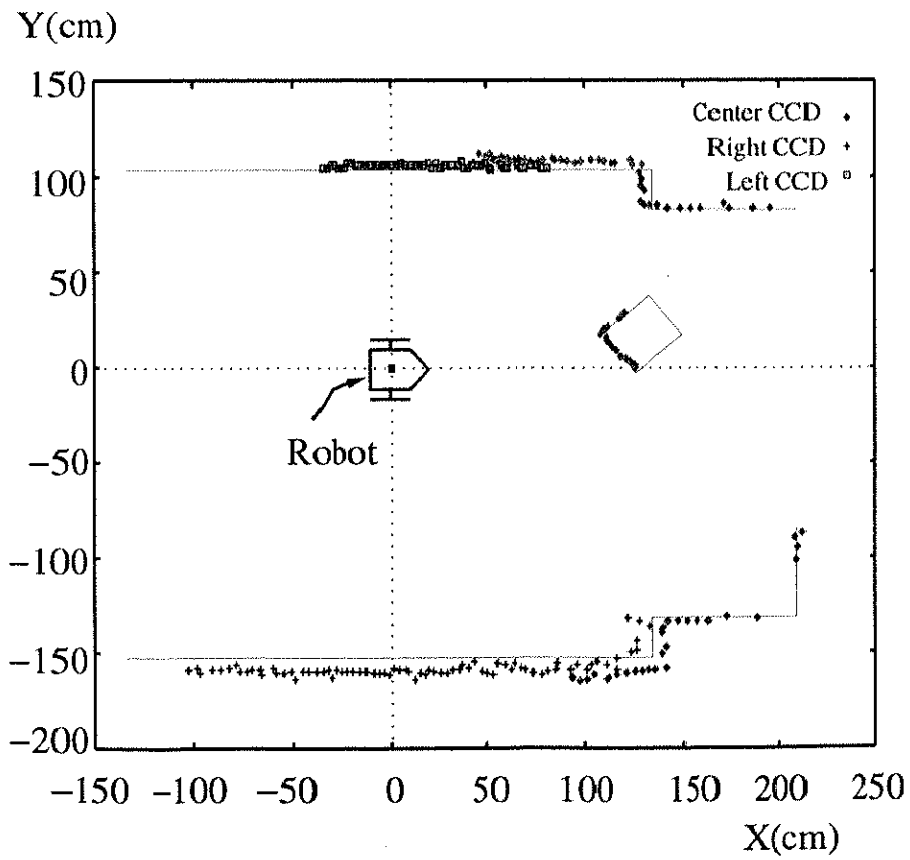


図 6.17: 得られた距離データ

超広角カメラから得られた画像に対する走査方法として扇型走査法を考案したことにより、ロッカーなどの光沢のある壁で、反射光の虚像があっても、これを無視することができるようになり、対象物までの位置情報をほぼ正確に得られるようになった。扇型走査処理をすることによって、1回の処理速度は600msecかかってしまうが、これは実際には満足できる処理速度である。1mの対象物までの位置を測定した時の計測誤差は、距離誤差にして約±5cmであり、これは使う上で満足できる距離誤差である。このような小型の移動ロボットに搭載でき、広い角度(260°)にわたって、実用的な処理時間(600msec)で距離を測定できるレンジセンサを開発することができた。

## 第 7 章

# レーザレンジセンサによる廊下長手方向 走行

### 7.1 はじめに

本章では、筑波大学第3学群棟内の廊下(図7.1)を実験環境とし、その廊下の壁に沿って長手方向へ移動ロボットを自律的に走行させることを取り扱った。この移動ロボットには第6章で述べたレーザレンジセンサを搭載し、センサベーストビヘーピアに基づいて廊下の長手方向へ走行させることが目的である。すなわち、廊下の長手方向に延びる壁をレーザレンジセンサによりみつけ、その壁の方向に走行させ、壁に近づきすぎれば離れ、離れすぎれば壁に近づくようにして、結果的に壁とある間隔を保って走行させることを実現する。このような走行は、もし、単純な大きく長い形状の壁が単に続くだけの環境であれば比較的容易であると考えられる。しかし実験環境として設定した実際の廊下における走行はそれほど容易ではない。なぜなら、実際の廊下環境は、図7.2のように壁の形状に凹凸があったり、ロッカーや箱等の障害物があるためである。そのため、廊下環境で移動ロボットが壁沿い走行中に、障害物に遭遇しても、衝突を避けて廊下の長手方向に沿って走行できなければならない。このような実際の廊下環境において、移動ロボットを壁沿いに廊下長手方向へ走行させることを実現することが本章の目的である。



図 7.1: 環境 (筑波大学 第三学群)

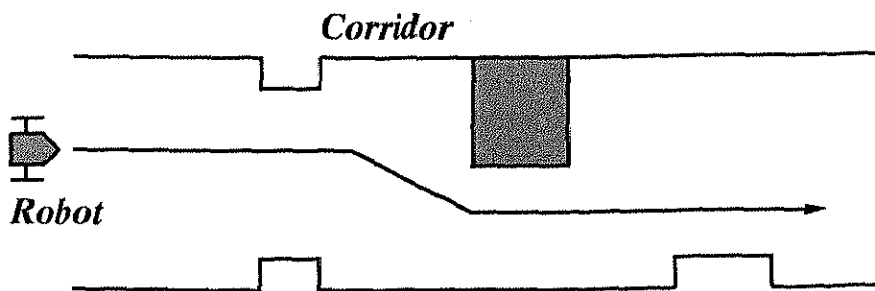


図 7.2: 箱などの障害物のある廊下環境

## 7.2 壁沿いに廊下長手方向へ走行する手法

移動ロボットを壁沿いに廊下長手方向へ走行させるためには、まずロボットの移動方向を決定するために廊下の長手方向を検出する必要がある。また、廊下の長手方向に沿う走行中、ロボットの進行方向に存在する障害物を検出し、その障害物への衝突を避け、回避を行なうことも必要である。そこで、本章で述べる廊下の長手に沿う走行のためのアルゴリズムは大略次のようなものとした。すなわち、まず、廊下の壁から一定の距離離れて、壁との相対角度がほとんど 0 になるように移動ロボットの走行経路を決める。同時に前方を監視し、いま決めた経路上にも壁 (障害物) があり、このまま進行すると衝突を起こす時には、それに衝突しないような経路をあらためて決める、というようなものとした。その際壁の傾きなどの情報が得やすいように、またこのような経路の計画がしやすいように、まずレーザレンジセンサから得られた距離データに線分をあてはめ、その線分に基づいて経路の計画を行なう。

### 7.2.1 距離点列への線分あてはめ

第 6 章で示したようにレーザレンジセンサから得られるデータは、扇状のレーザ光による平面内において、ロボットの中心から  $1^\circ$  毎の方向毎の対象物までの距離の値あるいは、その平面を張るロボットに固定された座標系に関する対象物の座標値の点列である。いまここでは、この点列を距離点列と呼ぶ。壁の傾きなどの情報を得、次節で述べる廊下長手方向走行アルゴリズムでこれを用いるために、センサから得られた距離点列へ線分あてはめを行う。以下に距離点列への線分あてはめ方法について示す。ただし、以下の処理は各 TV カメラの画像処理装置毎に行なう。

#### 1. (点列の分割)

それぞれ隣り合う点の間の距離がしきい値  $T_1$  以下だった場合同じ点列の集合とし、あるしきい値を越えたら、そこで点列の集合を分割する (図 7.3)。

#### 2. (再帰的な線分の決定)

分割された点列に対し、それぞれ点列の重心を計算する。重心を通る線分のうち、各点列と線分との距離の自乗が最小になるような傾きを求める。重心と傾きから、その点列にあてはめる 1 本の線分が決定される。あてはめられた線分



とそれぞれの点列の間の距離の最大値があるしきい値を越えた場合、その最大値の点で線分を分割し、折れ線にする。この処理を再帰的に行う (図 7.4)[51]。

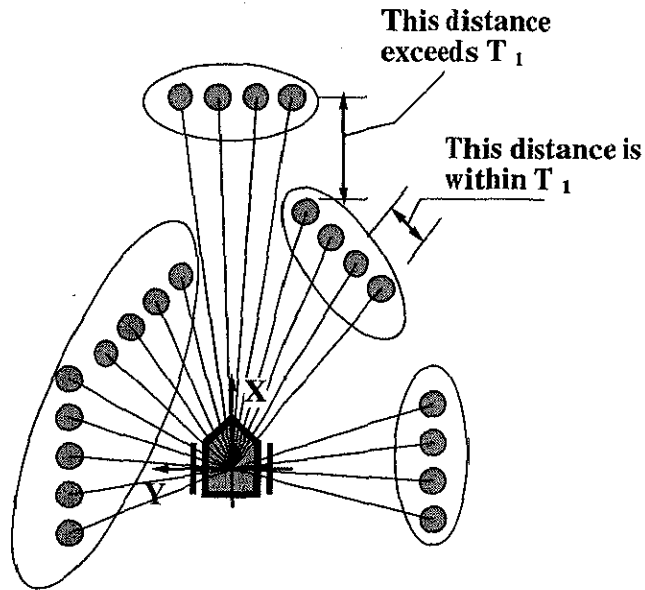


図 7.3: 線分あてはめ (点列の分割)

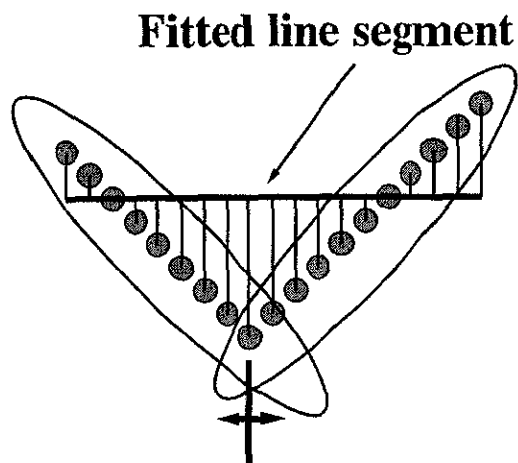


図 7.4: 線分あてはめ (再帰的な線分の決定)

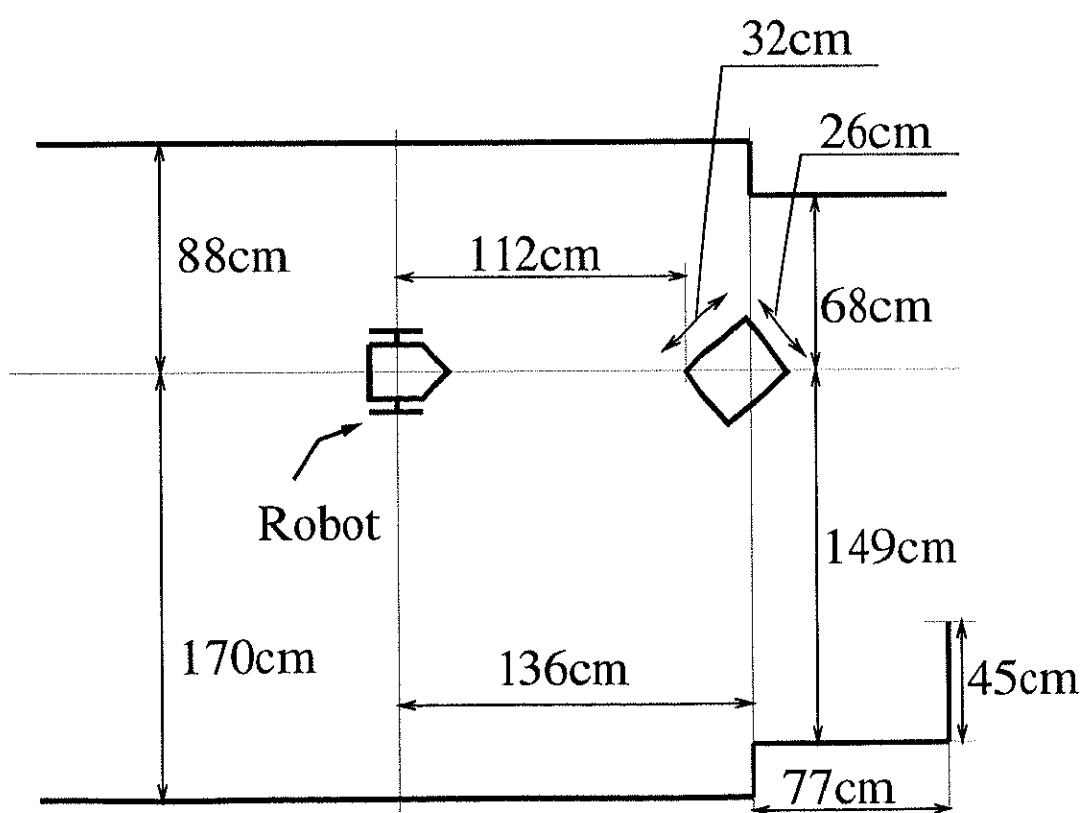


図 7.5: 測距環境

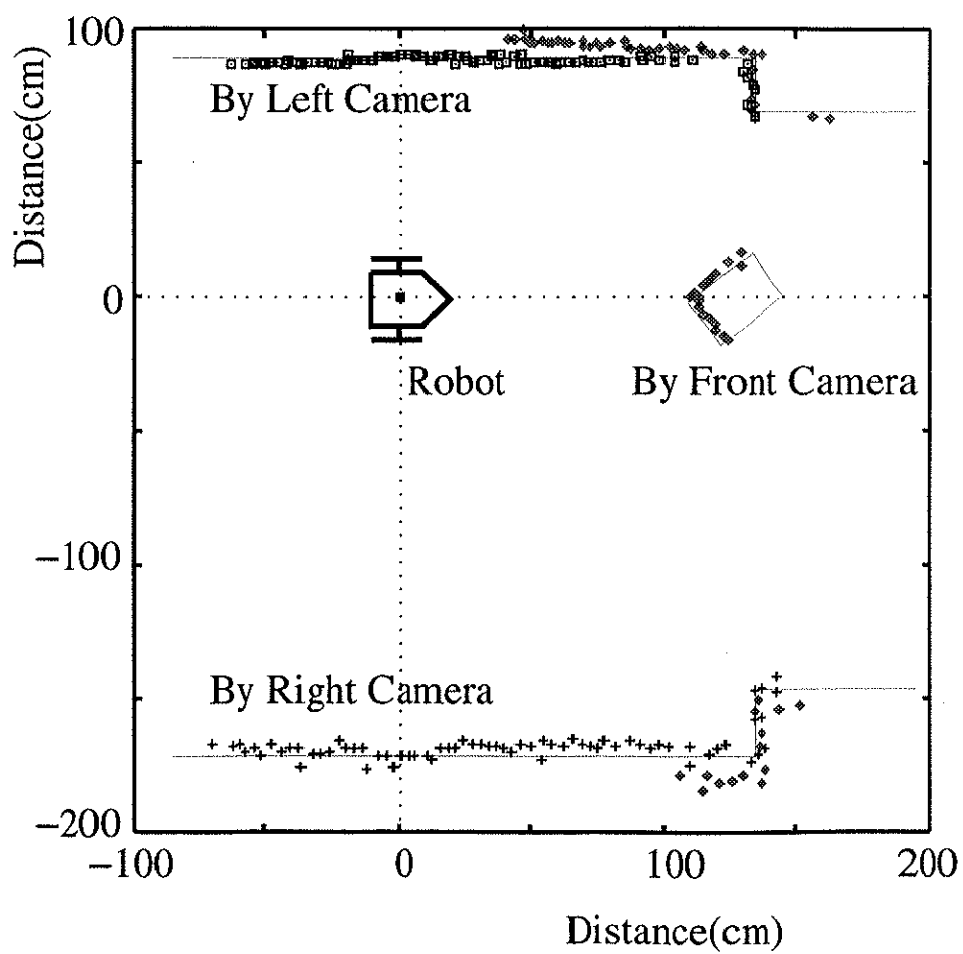


図 7.6: 実環境測距データ

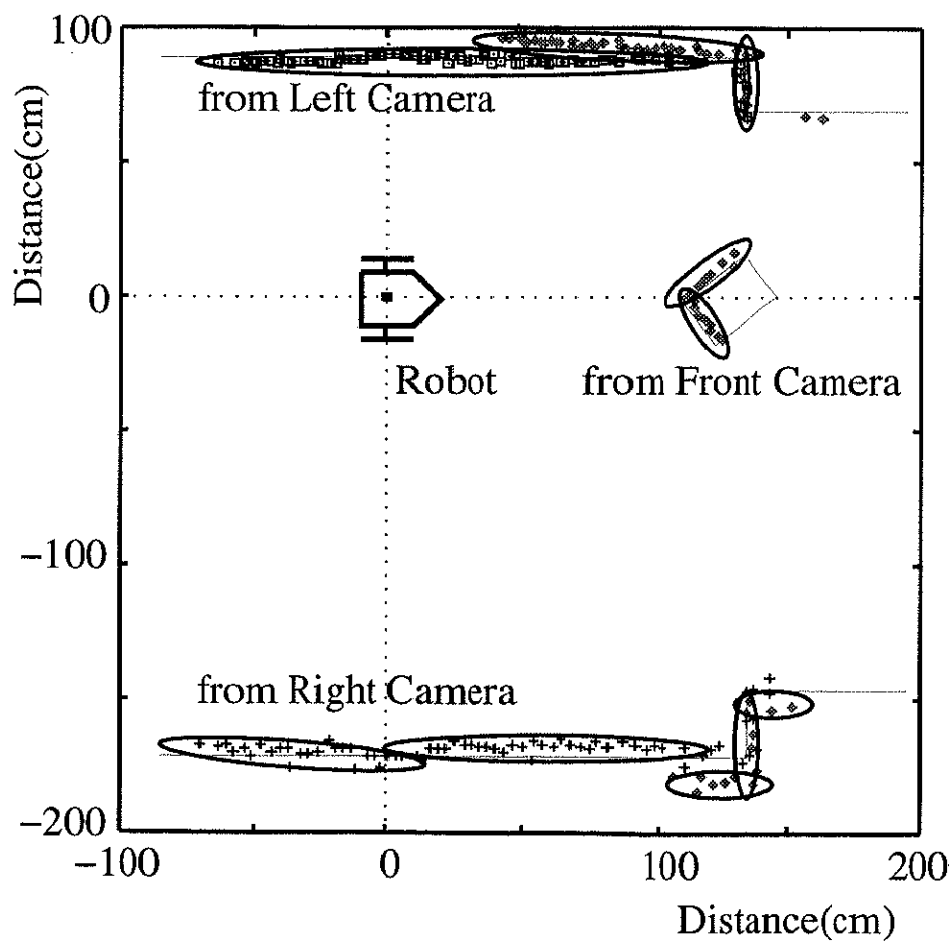


図 7.7: 実環境測距データの分割結果

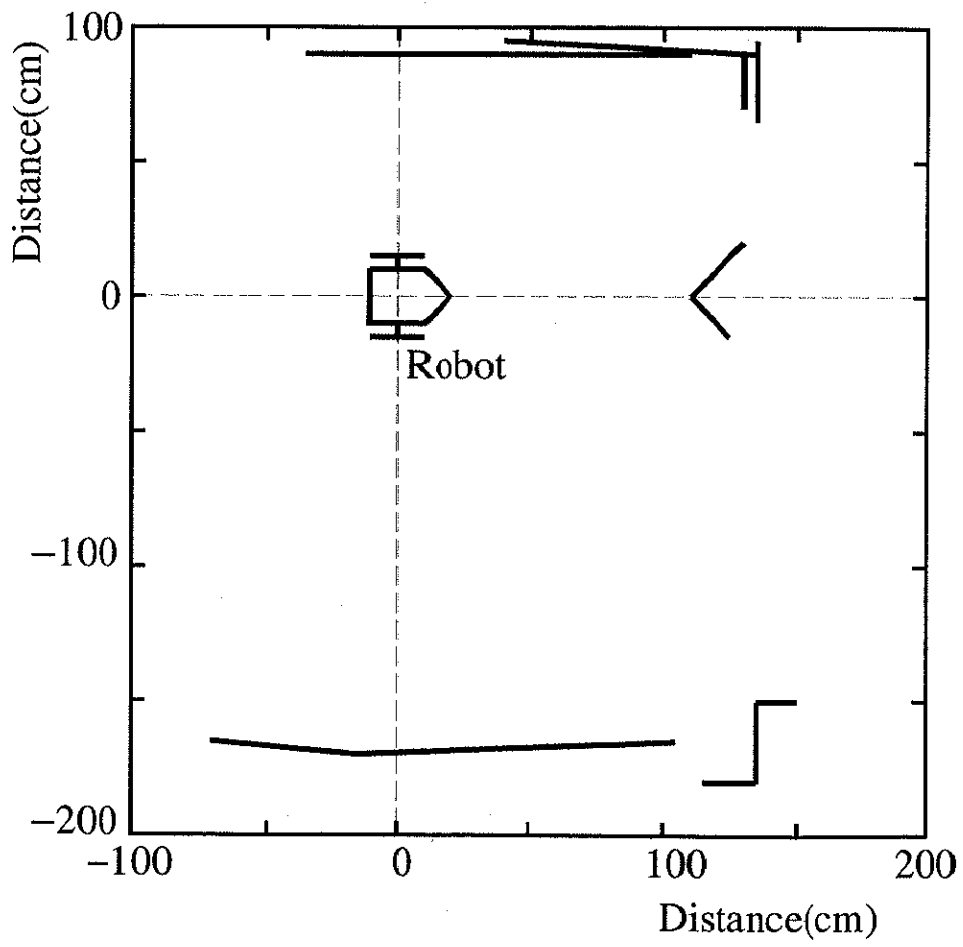


図 7.8: 実環境測距データへの線分あてはめ結果

先に述べた線分あてはめ方法を、図7.5のような廊下環境における測距データに実際に適用した。図7.6のように得られた距離点列は、最終的に図7.7のように分割された。また、最終的にあてはめられた線分は図7.8のようになった。図7.8から、ほぼ期待通りに、壁の平らな部分の距離点列に長い線分があてはめられることがわかった。

### 7.2.2 走行アルゴリズムの概要

第6章において述べたレーザレンジセンサは、ロボットの周囲  $260^\circ$  (前方から左右に  $\pm 130^\circ$ ) 程度の角度範囲、およびおよそ 3m 以内の範囲においてロボットを取り巻く壁や障害物などの表面までの距離を測定できた。また前小節で示したように直線状に距離点列が並んでいる部分に対して線分をあてはめることができた。このようにして得られた情報をもとに、ロボットが廊下の長手方向へ、廊下の壁に沿って自律的な走行ができるようなアルゴリズムを開発した。

このアルゴリズムにおいては、ロボットの初期位置は廊下の壁と壁のほぼ中央、初期方位は廊下の長手方向とほぼ平行であると仮定した。また障害物の形状は、幾何学的に簡単な形状であると仮定した。以下に、移動ロボットを廊下の壁沿いに長手方向へ走行させるアルゴリズムの概要を示す。いま、得られるセンサ情報は、ロボット前方から  $\pm 130^\circ$  の範囲の半径  $R$  以内の対象物までの距離点列である。ここでは右壁から一定の距離  $D$  離れて走行する場合について示す。(左壁に沿う場合は、左右が逆になる以外は同様とする。) 実際に用いたレーザレンジセンサでは、 $R=3\text{m}$  程度であり、ロボットが壁から  $D=50\text{cm}$  程度の距離を維持しながら走行することを想定している。ちなみに、ここで想定した走行環境としての廊下の幅は、一般的な建物の廊下の幅の  $1.5 \sim 3\text{m}$  程度である。したがって、走行中は、左右に壁があれば、レーザレンジセンサは両側の壁とも、検出できることを想定している。また、アルゴリズム中にあらわれる距離定数  $L$  は、サブゴールを与える時に使用する定数であり、 $R$  以内で与える。ここでは、 $L=1\text{m}$  を想定している。また以下では、 $X$ - $Y$  座標系をロボットに固定したものとし、ロボットの中心を原点としロボットの前方(進行方向)を向くように  $X$  軸を定める。

#### Step1: 線分あてはめ

ロボット前方から  $\pm 130^\circ$  の範囲の 3m 以内の対象物までの距離を測定し、その距離点列に線分をあてはめ、Step2へ進む(図7.9)。

**Step2: 直線 A の決定**

各線分のうち、X 軸より端点がすべて右側にある線分のうち、X 軸にほぼ平行な線分の中で一番長い線分をさがす (図 7.9)。

もし、その線分が予め与えられたスレッシュホールドを越える長さであれば、その線分と平行に、その線分から距離  $D$  離れた x 軸よりの直線を引き直線 A とする。そしてその後、Step3 へ進む。

もし、そうでなければ、X 軸を直線 A とし、Step3 へ進む。

**Step3: サブゴール P の仮決定**

移動ロボットの中心から直線 A に直交するように直線を引き、その直線と直線 A とが交わった交点を求める。そして、その交点から進行方向に対し距離  $L$  離れた点をサブゴール P とする (図 7.9)。そして、Step4 へ進む。

**Step4: 衝突チェック**

得られた各距離線分を (移動ロボットの幅の半分+安全なマージン) だけ拡大した長方形をつくり、各長方形の辺と先の直線 A の長さ区間  $L$  との交差チェックを行う (図 7.10)。

もし、交差していれば、Step5 へ進む。

もし、交差していなければ、Step6 へ進む。

**Step5: 通り抜けられる隙間探しとサブゴール P の再設定**

移動ロボットが通り抜けられる隙間を探すために、おのおのの線分の隣り合う線分間の距離の最小値を、Step4 で交差した線分から順番に左右に交互に計算する (図 7.11)。

もし、その距離がロボットの幅よりも大きければ、その隙間の直線 A に近い側の線分を拡大した長方形の隙間側の頂点に改めてサブゴール P を設定する (図 7.12)。そして、Step6 へ進む。

**Step6: サブゴール P まで直線追従**

移動ロボットを、直線 A に沿ってサブゴール P まで第 3 章で述べた直線追従コマンドを用いて、直線追従させる。そして、Step7 へ進む。

**Step7: 停止と回転**

移動ロボットを、サブゴール P で停止させ、移動ロボットの前方が、直線 A と平行になるように回転させる。そして、Step1 へ進む。

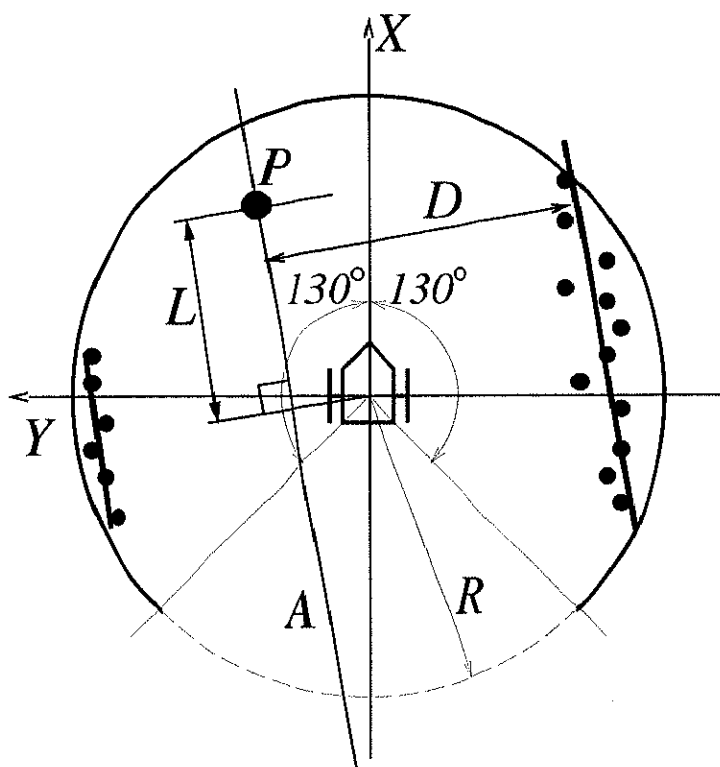


図 7.9: 距離測定と線分あてはめ (Step1)、直線 A の決定 (Step2)、および、サブゴール P の仮決定 (Step3)

**7.3 実機実験**

上述の、壁沿いに廊下の長手方向へ走行する手法を用いて図 7.13 のような環境を走行させた。図 7.13 では、移動ロボットがまっすぐ走行したら、衝突するように四角い箱を置いておいた。図 7.14 に、この実験走行時の移動ロボットのオドメトリによる自己位置を廊下平面上にプロットして示す。また、図 7.15 に、実験風景をスナップショット



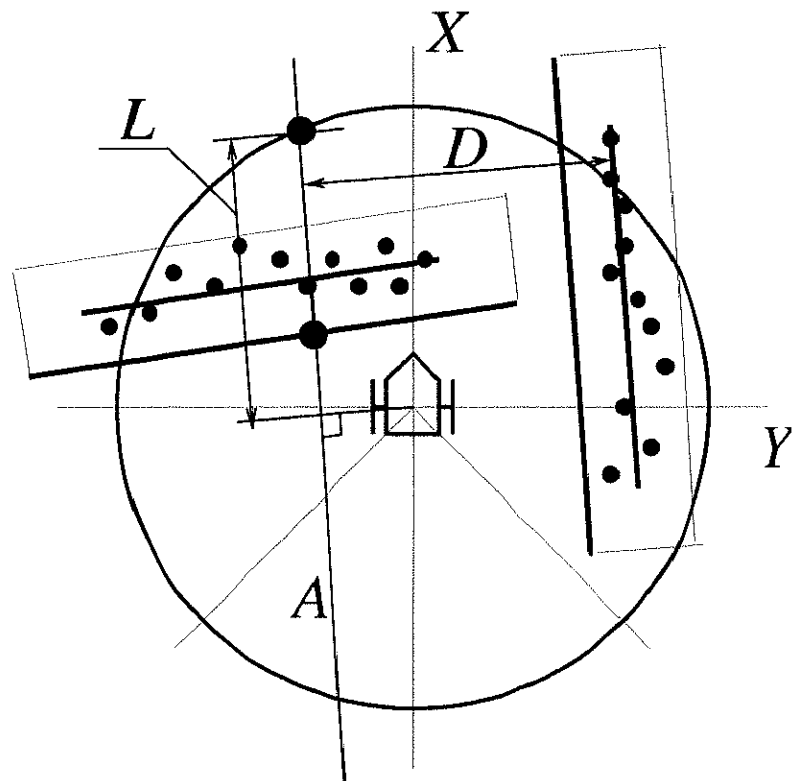


図 7.10: 衝突チェック (Step4)

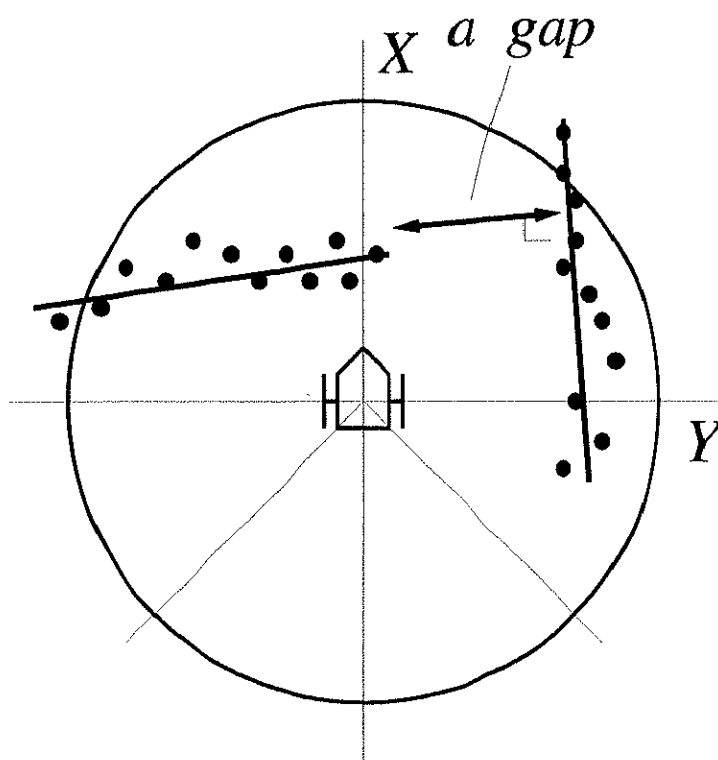


図 7.11: 通り抜けられる隙間探し (Step5)

*Width of the robot + safety margin*

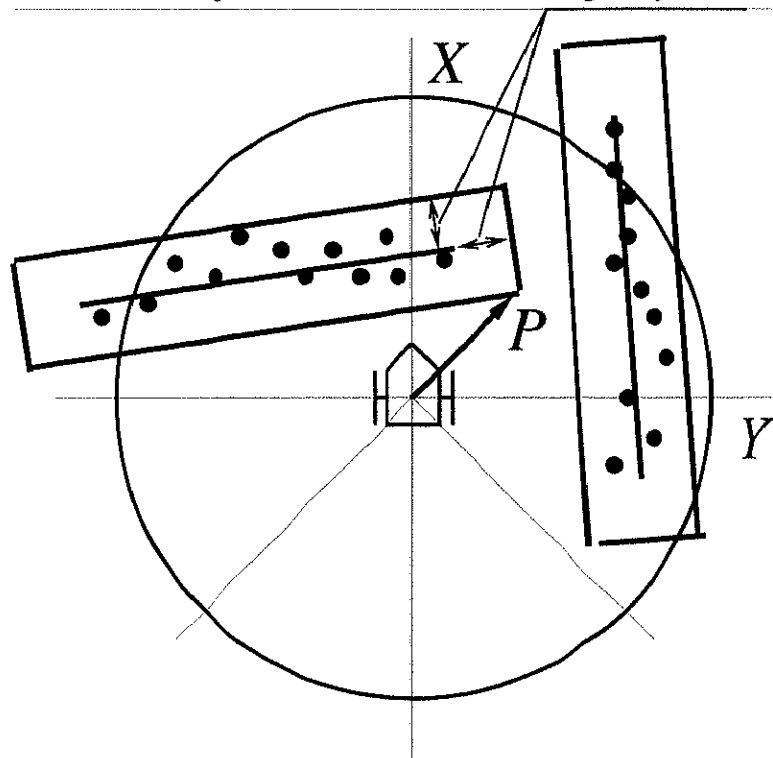


図 7.12: サブゴール P の再設定 (Step5)、および、サブゴール P まで直線追従 (Step6)

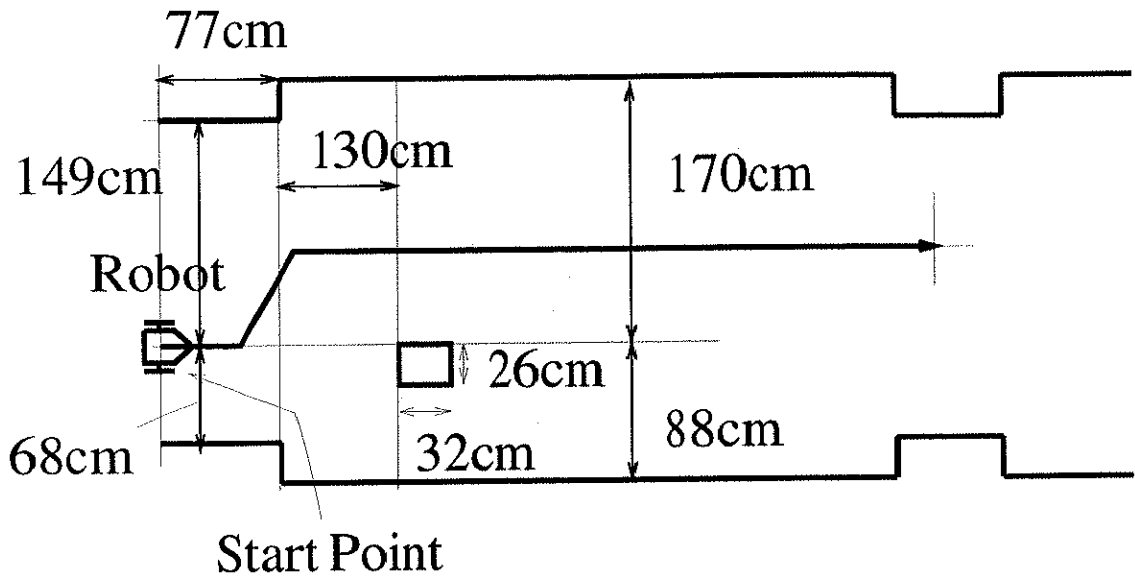


図 7.13: 実験環境と走行経路

トで示す。この結果、移動ロボットは障害となる箱を避け、予期した通り衝突せずに廊下の壁に沿って走行できていることが確認された。

#### 7.4 7章のまとめ

距離データに線分当てはめをすることにより、移動ロボットを壁沿いに廊下長手方向へ走行させる手法を開発した。そして、実機実験により、移動ロボットがほぼ期待

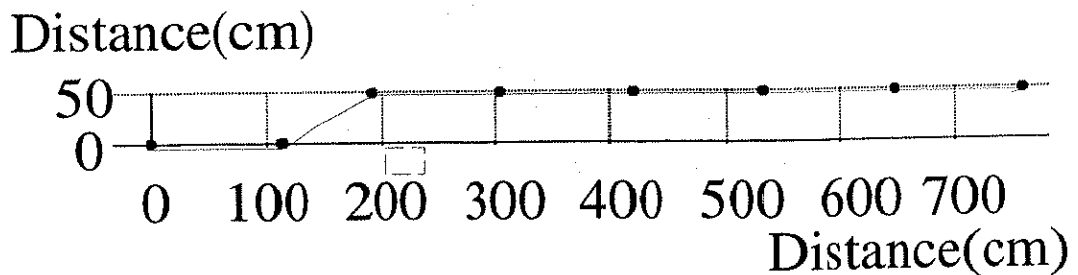


図 7.14: 実験時のオドメトリによるロボットの位置(・印)



図 7.15: 実験時のスナップショット

通りに走行することを確認した。廊下環境に障害物などがある場合でも、それを避けて壁沿い走行するような移動ロボットの壁沿い走行を実現することができた。

## 第 8 章

# 言語的な行動シーケンスの指示によるナビゲーション

### 8.1 はじめに

本章は比較的広範囲にわたる廊下環境において移動ロボットに走行すべき道順を行動シーケンスとして言語で与えることによってナビゲーションさせる問題を取り扱う。実際に移動ロボットを走行させる実験環境としては、第 7 章と同様に筑波大学の第 3 学群の廊下とした。この廊下は、実際には図 8.1 のような数百メートル四方に広がりを持ち、いくつかの廊下が交差点で直角に接続された環境になっている。廊下の幅は 1.5 ~ 3m 程度であって一様ではなく、場所により異なっている。また個々の廊下は第 7 章で述べたように、ロッカーや柱等の箱状の物体があり、単純な形状の壁ばかりではない環境である。ただし廊下のドアはすべて閉じているものとした。このように、走行環境が広範囲にわたり、また壁の形状も単純でない場合、柱のひとつひとつまでの形状をはかり、詳細にこの環境に関する地図を人手で作ろうとすると多くの手間がかかる。したがって、このような詳細な環境地図はなるべく使わず、むしろ、ユーザが知っている壁の有無情報と、壁が無い場合(交差点)の右左折の指示を与える方が、比較的広範囲にわたる移動ロボットのナビゲーションには都合がよい。すなわち、「右壁に沿え、もし、右壁が無くなったら、右に曲がれ」のようなとるべき行動と監視すべき事象を順にシーケンスとして記述した言語指示によりナビゲーションが行える方が便利である。そこで、このように人が人に道を教えるような言語的な指示方法によ

り、移動ロボットのナビゲーションを実現するのが本章での目的である。

これを実現する方法として、第 6 章で述べたレーザレンジセンサを移動ロボット上に搭載し、(A) 第 7 章で述べた廊下の長手方向を走行するアルゴリズムを用い、さらに、(B) 壁の有無の検出により交差点の候補を認識することにより、上に述べた言語的な行動指示シーケンスを実際に実行できる枠組を移動ロボット上に構築してナビゲーションを行うようにした。本章では、この行動指示シーケンスを与えるために設計した言語仕様について述べ、壁の有無検出による廊下交差点の認識方法を示し、最後にこれらの言語的指示によるナビゲーションの実験例を示す。

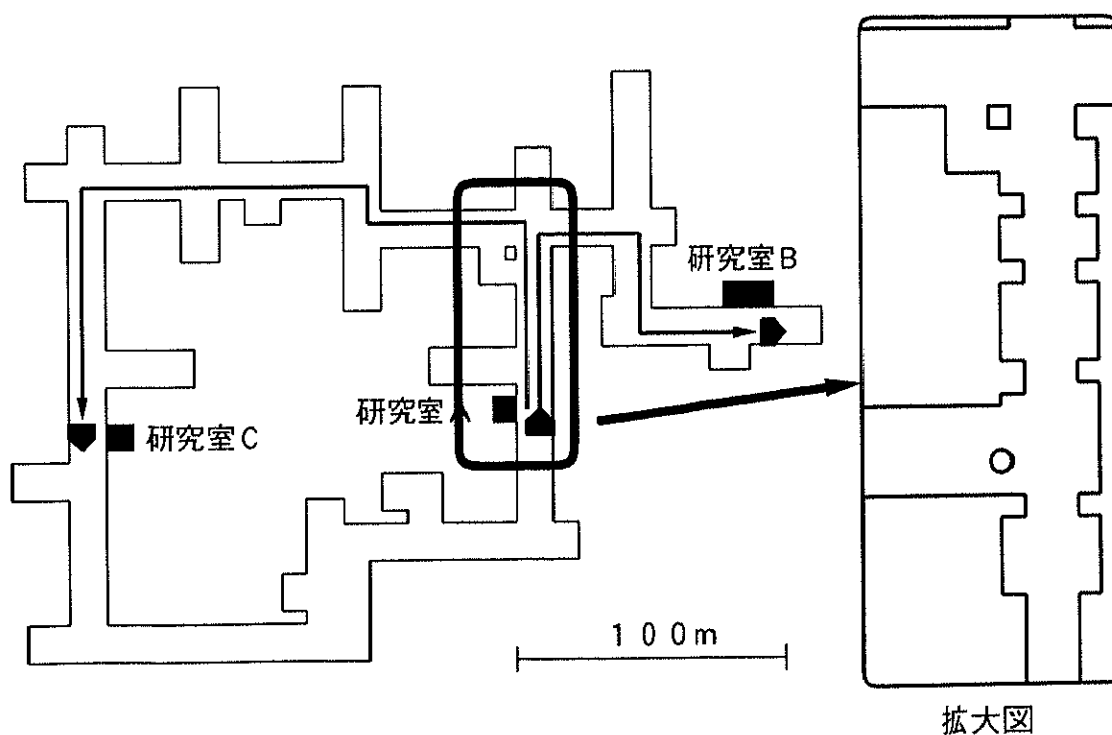


図 8.1: 想定環境 (筑波大学 第三学群)



## 8.2 行動シーケンスによってナビゲーション経路を指示する言語の様

例えば、図 8.1 のような廊下環境において、研究室 A 前の移動ロボットの現在位置から研究室 B の前まで行くナビゲーション経路を指示するためには、「廊下に沿って進め。(交差点で) 右側に壁がなくなったら右折し、廊下に沿って進め。右側に壁がなくなったら右折し、廊下に沿って進め。左側に壁がなくなったら左折し、廊下に沿って進め。Xm 進んだら停止せよ。」というような定型的な記述ができればよい。すなわち、「廊下に沿って進め」「XX 側に壁がなくなったら YYY し」というような記述単位のシーケンスが定義できるようにすればよい。ここでは、この言語の処理系を計算機上に実装するための便宜上英語的な記述によることにし、以下のような記述を順に繰り返すことによりナビゲーション経路を指示できるようにした。

Keep\_on Job.

When Condition [and Condition...] [at Number],

Exec Action.

- Job に与えられる命令
  - ・ going\_along\_corridor : 廊下長手方向に沿って走行し続けよ
- Condition に与えられる条件
  - ・ no\_right\_wall : 右壁がない
  - ・ right\_wall : 右壁がある
  - ・ no\_left\_wall : 左壁がない
  - ・ left\_wall : 左壁がある
  - ・ no\_front\_wall : 前方に壁がない
  - ・ front\_wall : 前方に壁がある

- ・ for Dist m : 直前の Job で走行し始めた点から、およそ 距離 Dist m 走行した場所
- Number に与えられる条件
  - ・ 1st: 1 番目の (default)
  - ・ 2nd: 2 番目の
  - ・ 3rd: 3 番目の
- Action に与えられる命令
  - ・ stop : 停止せよ
  - ・ turn\_right : 右に 90° 曲がり直進せよ。
  - ・ turn\_left : 左に 90° 曲がり直進せよ。
  - ・ go\_straight : 直進せよ。

まず、廊下の長手に沿って進み、交差点があったら左に曲がるべき場合を例にとり、ここで提案する言語的なナビゲーション経路指示の具体例を示す。

左に曲がることのできる交差点の種類は、廊下と廊下は直角に交差するものとしていたので表 8.1, 8.2 に示す 4 種に限られる。経路の指示は、基本的には、左壁がないところを左に曲がれという記述になる。すなわち、When 以下について “When no\_left\_wall” とのみ記述すれば (表 8.1, 8.2 の [ ] で囲まれた部分を省略しても) この 4 種の交差点のいずれかにさしかかった時、移動ロボットは左折できる。しかし、この交差点の特徴をより明示的に示せば、そこが本当に曲がるべき交差点であるかどうかを確認でき、誤って指示とは別の交差点で左折してしまうケースを減らすことができる。

例えば、表 8.1 の最上段の場合には、十字路の特徴である右壁がなく、左壁がなく、前方に壁がない場所という記述をすることにより、明示的に十字路で左に曲がれということを示すことができる。また、十字路までのおおよその距離 (for 20m) や、このシチュエーションの現れる順番 (at 1st) も付加的に与えることができるようになっている。

以下、T 字路の場合の 2 ケースと L 字路の場合について、表 8.1 下段、表 8.2 の右側にその明示例を記した。左方向に曲がる例について示したが、右方向に曲がる場合も

表 8.1: 言語的な経路指示方法の具体例 (その 1)

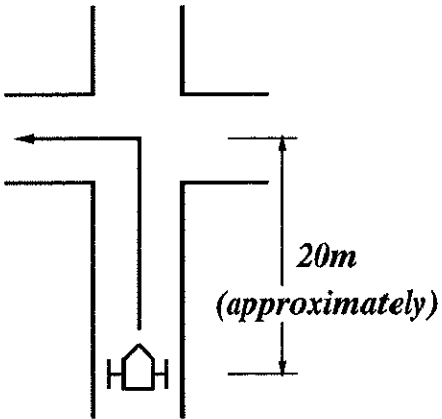
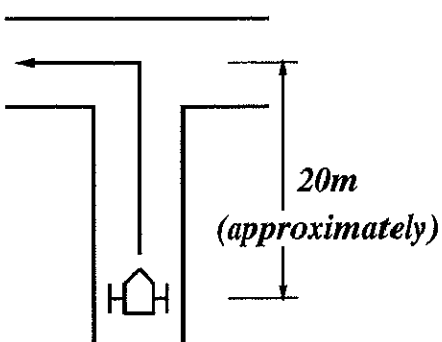
Situation	Instruction Example
	<p>Keep_on going_along_corridor.</p> <p>When no_left_wall                      [and for 20 m]                      [and no_right_wall]                      [and no_front_wall]                      [at 1st],                      Exec turn_left.</p>
	<p>Keep_on going_along_corridor.</p> <p>When no_left_wall                      [and for 20 m]                      [and no_right_wall]                      [and front_wall]                      [at 1st],                      Exec turn_left.</p>

表 8.2: 言語的な経路指示方法の具体例 (その 2)

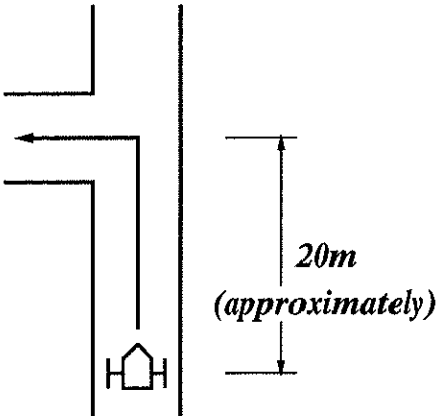
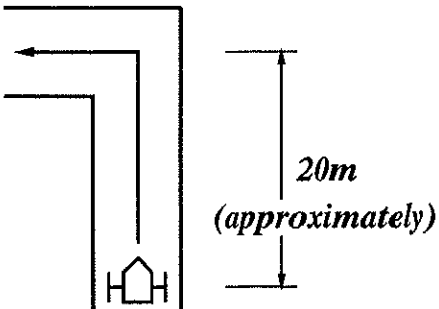
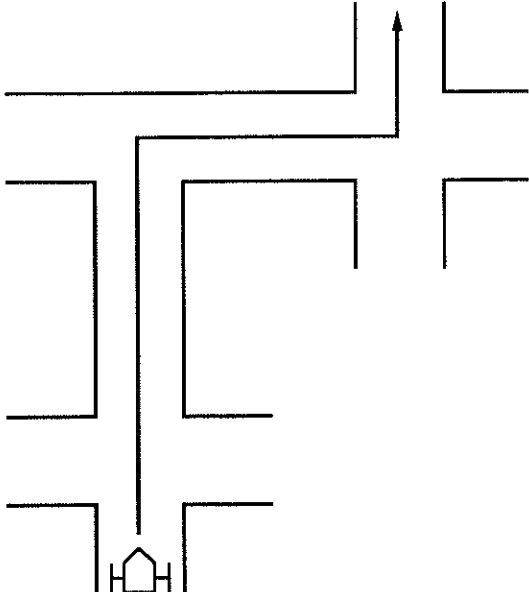
Situation	Instruction Example
	<p>Keep_on going_along_corridor.                      When no_left_wall                      [and for 20 m]                      [and right_wall]                      [and no_front_wall]                      [at 1st],                      Exec turn_left.</p>
	<p>Keep_on going_along_corridor.                      When no_left_wall                      [and for 20 m]                      [and right_wall]                      [and front_wall]                      [at 1st],                      Exec turn_left.</p>

表 8.3: 言語的な経路指示方法の具体例 (その 3)

Situation	Instruction Example
	<p>Keep.on going_along_corridor.                  When no_right_wall                  and front_wall                  and no_left_wall,                  Exec turn_right.</p> <p>Keep.on going_along_corridor.                  When no_left_wall                  and no_front_wall                  and no_right_wall,                  Exec turn_left.</p>
	<p>Keep.on going_along_corridor.                  When no_right_wall                  and no_left_wall                  at 2nd,                  Exec turn_right.</p> <p>Keep.on going_along_corridor.                  When no_left_wall                  and no_front_wall                  and no_right_wall,                  Exec turn_left.</p>

同様の記述になる。このように、明示的に条件を示すか、必須の条件のみを記述して他の条件を省略するかは、この指示を与えるユーザに任せることにする。

表 8.3 の例は、経路指示の記述を組み合わせた応用例である。いま、表 8.3 左側にあるような走行環境を考える。この場合、少なくとも次の 2 通りの経路指示の記述方法が考えられる。

(1) 「廊下の長手方向に進め。もし、左壁がなく、右壁がなく、前方の壁があることが生じたら、右折し、直進し、廊下の長手方向に進め。もし、左壁がなく、右壁がなく、前方の壁がないことが生じたら、左折し、直進せよ。」または、

(2) 「廊下の長手方向に進め。もし、2 度目に左壁がなく、右壁がない状況が現れたら、右折し、直進し、廊下の長手方向に進め。もし、左壁がなく、右壁がなく、前方の壁がないことが生じたら、左折し、直進せよ。」である。これに対応する記述を表 8.3 の右側に記した。本章で提案するナビゲーション経路指示の与え方は、このような記述をどちらでも許している。

### 8.3 ナビゲーション経路を指示する言語の実装とナビゲーションの実現

前節で定義したナビゲーション経路を指示する言語は、図 8.2 のように、構文解析と字句解析を行うプリプロセッサを用いて、筆者らの研究グループで用いている移動ロボット行動記述言語 Robol/0 [52] に変換し、それをコンパイル後、移動ロボット上のマスターモジュールに転送することにより移動ロボットを走行させる。

移動ロボット上には、第 6 章で述べたレーザレンジセンサを搭載する。また、前節で定義した “Keep\_on going\_along\_corridor” に対応する行動は、第 7 章で述べた廊下長手方向走行方法に展開されるようにした。また、同じ前節で定義した構文の “When” 以下の節で壁の有無を調べなければならないが、これについては次のようにした。

廊下の長手方向に走行するアルゴリズム (7.2.2 節) の Step 1 において、その時のセンシングによるレーザレンジセンサの左右の距離データから、図 8.3 のような移動ロボットの左右の空間の距離点列の有無を調べ、距離点列がその空間になれば壁の無い通れる場所 (交差点) とし、あれば壁のある場所 (交差点でない) と判断する。

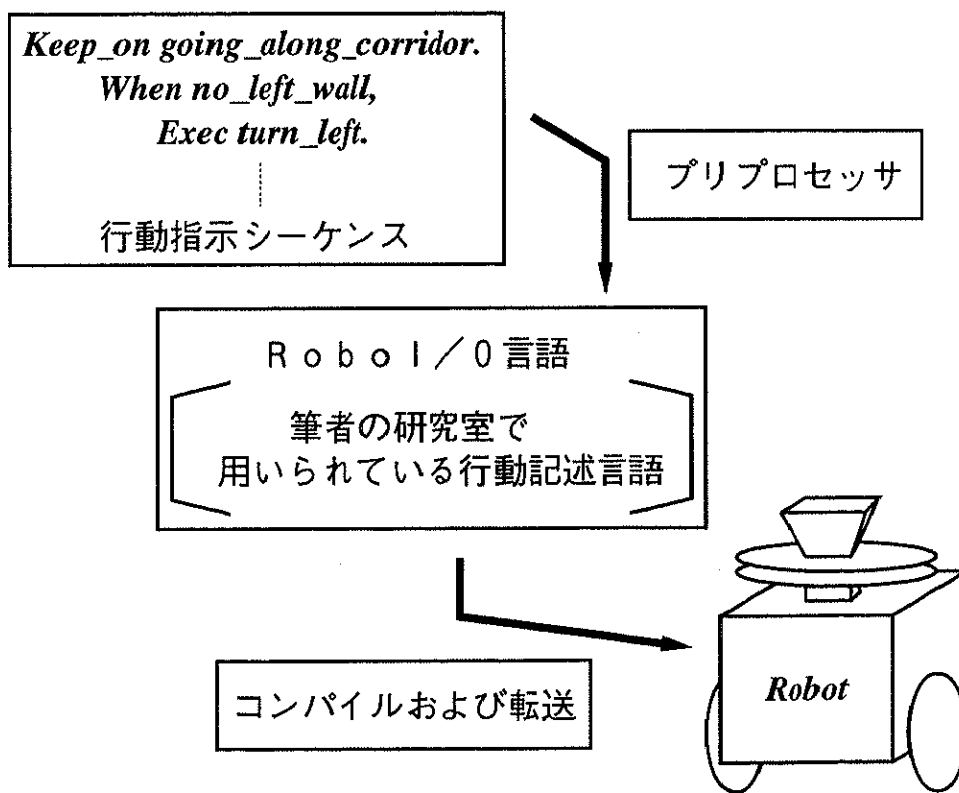


図 8.2: 言語実装

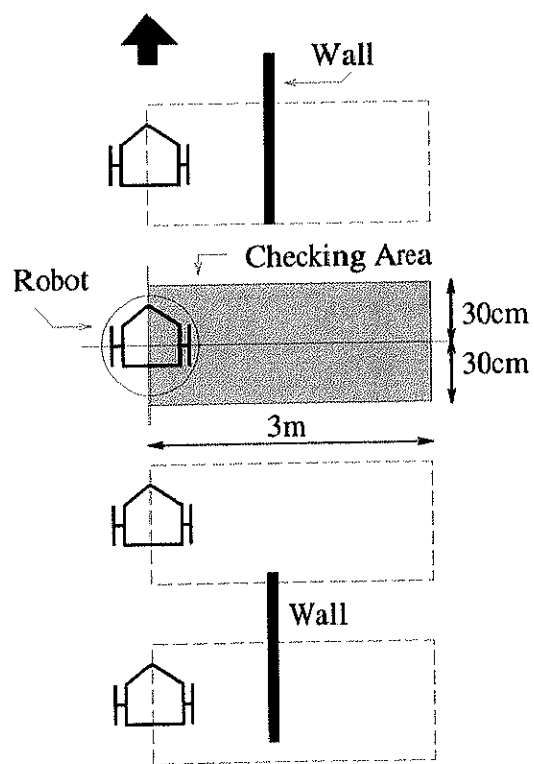


図 8.3: 壁の有無検出



## 8.4 言語的指示による実環境におけるナビゲーションの実験

図 8.4に示す実際の廊下環境において、移動ロボットを走行させる実験を行った。図 8.4の研究室 A 前をスタート地点とし、研究室 B の前に移動ロボットを行かせたい場合、図 8.5のようなナビゲーション経路指示を移動ロボットに与えれば目的の位置に到達できるはずである。

このような指示により移動ロボットにナビゲーションさせ、ほぼ期待通りに移動ロボットが走行したことを確かめた。その時の、図 8.4の記号に対応した地点の実験時スナップショットを図 8.6に示す。

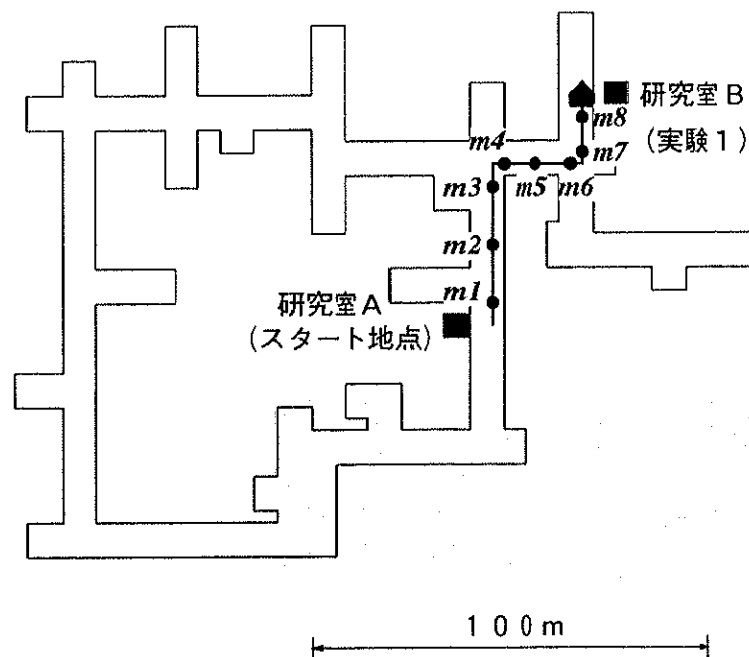


図 8.4: 実験結果 (軌跡)

## 8.5 8章のまとめ

本章では「2つめの交差点を右に曲がって、3つめの交差点を左に曲がって、10m行ったところに目的地があります。」というような人が人に道を指示するような行動指示シーケンスにより、移動ロボットのナビゲーションを実現することを扱った。

```
Keep_on going_along_corridor.  
  When no_right_wall  
    Exec turn_right.  
Keep_on going_along_corridor.  
  When no_left_wall  
    Exec turn_left.  
Keep_on going_along_corridor.  
  When for 5 m,  
    Exec stop.
```

図 8.5: 言語的指示例

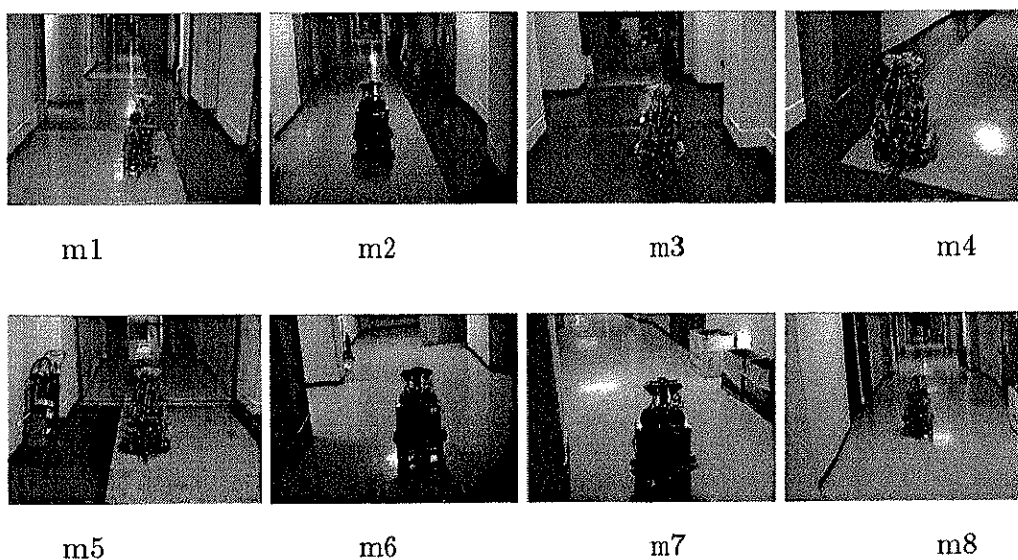


図 8.6: 実験 1(目的地: 研究室 B) のスナップショット

廊下の長手に沿う行動と、壁が無い部分(交差点)の検出およびそれが検出された場合にとるべき行動を組にして定義し、これを順にシーケンスとして記述することでナビゲーションのための走行経路を指示する形式言語を定義した。また、この言語の実装を行ない、想定した環境のなかで実際にこの言語の記述に従ったナビゲーションが行なえることを実験的に確認した。この実現のため、第 6 章で述べたレーザレンジセンサを活用し、このセンサから得られる情報をもとに、第 7 章で述べた壁沿いに廊下の長手方向に向かって走行する機能を用いた、また壁の有無の検出を行なってナビゲーションを実現した。

## 第 9 章

### まとめと結論

本研究では、「詳細な環境モデルを用いない移動ロボットの距離センサ情報に基づく屋内ナビゲーション」をテーマとした。本研究は、ロボット周囲の環境、すなわち、ロボット周囲に存在する壁や対象物の形状を十分に知ることができる外界センサをロボット上に搭載し、壁や対象物の周囲に沿うという定型的な行動をセンサベーストビヘービアに基づいて実現すれば、詳細な環境モデルを事前に用意しなくても、移動ロボットの屋内ナビゲーションを実現できる、という立場に立ったものである。この点において従来の移動ロボットの屋内ナビゲーション研究のひとつの大きな流れである、詳細な環境モデルから走行経路を位置座標に基づいて計画し、常に自己の推定位置と照合しながら高い環境認識能力のある外界センサを用意せずとも走行する方法(位置ベース走行)とは、対極にあるアプローチとなっている。

本研究では、外界センサとして超音波距離センサを 12 個円環上に配置したソナーリング、および床面に平行な扇状のレーザ光を投射し周囲の物体にその光があたった部分を TV カメラで検出することにより三角測量の原理を用いて物体までの距離あるいは位置を求めるレーザレンジセンサをそれぞれ開発・製作した。本論文では、これらの 2 つのセンサを独立に使い、本研究のテーマに基づく壁沿い走行アルゴリズムを提案し走行実験例を示した。また、この壁沿い走行のアルゴリズムを応用した屋内ナビゲーション法を提案して走行実験を行なった。具体的にはまずソナーリングを搭載した移動ロボットにより、

- (1) ソナーリングで計測された距離情報に基づいて未知形状の壁あるいは対象物の周囲に沿う走行(第 4 章)、

(2) ソナーリングによる壁沿い走行と位置ベース走行を適宜切り換えながら行なう移動ロボットのナビゲーション(第5章)

を取り扱った。

また、レーザレンジセンサに関するものでは、まず、

(3) 超広角TVカメラと扇状光を用いた移動ロボット用レーザレンジセンサの構造および画像処理手法、距離・位置計算法(第6章)

(4) レーザレンジセンサで計測された環境に関する情報を用いて、廊下の長手方向に沿って走行させるアルゴリズムの提案と実験(第7章)、

(5) 移動ロボットの走行経路を行動指示として言語で与える屋内廊下環境におけるナビゲーション(第8章)を取り扱った。以下、上記の5つのテーマについてそれぞれ総括する。

## 9.1 各テーマの総括

### [ソナーリングを用いた壁沿い走行]

第4章において述べたソナーリングにより計測された距離情報に基づくセンサベーストビヘービアによる壁沿い走行アルゴリズムでは、ソナーリングが対象物を検出することができる限りは、対象物の周囲あるいは廊下等の壁面に沿って移動できることを示した。ここで示したアルゴリズムは、対象物の形状を特に想定せずに設計を行なったものである。しかし、実際に様々な対象物においてこの壁沿い走行アルゴリズムを用いて実験をしてみると、超音波距離センサの測距範囲に対象物があっても、これを検出できないことがあることがわかった。これは、アルゴリズムのせいではなく、本質的な超音波の鏡面反射特性によるものであり、対象物の反射面の向きによっては超音波の反射がセンサに戻らず、対象物を検出できないためであると考えられる。このため対象物の形状によってはそれが検出できない場合があるので、対象物の形状をまったくの未知として、対象物に衝突する可能性のほとんどない壁沿い走行アルゴリズムを工夫することは困難であることを指摘した。

一般に、ソナーリングを使えばどんな形状の対象物でも、あたりまえに壁沿い走行できると考えられていたが、超音波の鏡面反射の特性により、対象物の形状によっては本質的に壁沿い走行がうまくいかないことがあることを示しことが成果である。

### [ソナーリングによる壁沿い走行と位置ベース走行を切り換える屋内ナビゲーション]

第 5 章では、移動ロボットに対して事前に走行すべき軌跡を与える位置ベース走行が適する環境と、ジグザグな壁が連続し、走行すべき軌跡をいちいち与えると煩雑になり、むしろ壁沿い走行させる方が都合の良い環境があることを指摘した。第 4 章で提案した壁沿い走行アルゴリズムを移動ロボットの行動を指示するプログラムからひとつの関数で呼び出すことができるように実装し、経路の途中にジグザグな通り道があり、第 4 章の壁沿い走行アルゴリズムが適用できる環境の場合にはこの関数を発行するだけで複雑な経路の指定を避けることができた。また、移動ロボットがナビゲーションをしている時に、移動ロボットの進路軌道上に予期しない障害物などがあつた場合も同様に一時的に移動ロボットに壁沿い走行をさせて障害物回避をさせることが実現できた。また、壁沿い走行から位置ベース走行への切り換えができるためには、壁沿い走行中も移動ロボットのオドメトリにより自己位置を把握し、その切り換え点に到達したか否かを監視する必要がある。このとき、壁沿い走行が約 10m 程度続くと、オドメトリの累積誤差が増大しすぎて、スムーズに位置ベース走行に切り換えることができないことがあつた。そのため、この切り換え時に誤差を修正する方法を開発した。本手法は、1 回の壁沿い走行の距離が 10m 程度以下であれば、支障なく、壁沿い走行から位置ベース走行に切り換えることができる。

ここでは、位置（座標値）ベース走行と、センサベースな壁づたい走行を切り換えながら自律移動ロボットのナビゲーションを実現し、それが実現できる技術を開発したことが成果である。

### [扇状光と超広角 TV カメラを用いたレーザレンジセンサの開発]

第 6 章では、超広角 TV カメラと扇状光レーザを用いたレーザレンジセンサの開発について示した。また、そのセンサの構造、対象物までの距離および位置の計算方法、画像メモリセルの探索方法などについて検討を行ない実装した。実際にこのセンサを廊下で用い、その実験結果からほぼ正確な計測結果が得られることが示された。なお開発したレーザレンジセンサは床面上 23cm の平面内のロボット前方より  $\pm 130^\circ$  の範囲でロボットより 3m 程度までに存在する対象物を検出することができた。このレンジセンサによる計測の 1 回に要する処理速度は 600msec であるが、これは使用上は満足できる処理速度であった。1m の対象物までの距離を測定した時の距離誤差は、

約± 5cm であり、これも使用上満足できる距離誤差である。

ここでは、このような小型の移動ロボットに搭載でき、広い角度(260°)にわたって、実用的な処理時間(600msec)で距離を測定できるレンジセンサを開発したことが成果である。

#### [レーザレンジセンサによる廊下長手方向に沿う走行]

第 7 章では、ロボットの走行環境を建物内の廊下とし、第 6 章で示したレーザレンジセンサによって計測された主に壁面の距離点列に線分をあてはめ、この線分群のうち、ロボットの横にある線分を壁であるとしこれに平行に進むことを基本とする壁沿い走行のアルゴリズムを設計した。また、進行方向に壁あるいは障害物がある場合にはそれを回避するようにした。そして、このアルゴリズムを移動ロボットに実装し、実環境の筑波大学の廊下環境で、障害物などを避けて、壁沿い走行するような移動ロボットの壁沿い走行を実現した。ここでは、廊下環境に障害物などがある場合でも、それを避けて壁沿い走行するような移動ロボットの壁沿い走行を実現し、それが実現できる技術を開発したことが成果である。

#### [言語的な行動指示によるナビゲーション]

第 8 章では、移動ロボットの走行経路を「廊下の長手方向に沿う」「左壁がなかったら左に曲がれ」などの行動指示として与える言語を定義し、実装を行なった。この言語によって指示される廊下の長手方向に沿う走行は第 7 章で扱った走行アルゴリズムを用い、また壁の有無の検出にも第 6 章で扱ったレーザレンジセンサを用いた。そして、定義された言語仕様に従って走行経路を指示し実際に筑波大学の廊下環境を走行させる実験を行ない、目的位置まで走行させることが実現できた。前提としている環境は、幅が 1.5 ~ 3m のほぼ直交するような廊下環境であった。この環境の前提は、特殊な構造の建物を除いて、一般の大きな建物に対して、多くの場合あてはまり、筑波大学以外の建物でも適用できると考えられる。定義した言語仕様は、前提としていた筑波大学の第 3 学群の廊下のような、幅が 1.5 ~ 3m のほぼ直交する廊下環境においては、使えるものであることがわかった。ここでは、人が人に道を指示するような行動指示方法を使って移動ロボットの屋内ナビゲーションを実現し、それが実現できる技術を開発したことが成果である。

## 9.2 本研究の産業や社会への応用について

本研究によって得られた成果として、特に、

(a) ロボットをとりまく周囲の環境の様子を十分に把握できるセンサを用意することによって、廊下の壁に沿うなどの定型的な行動をセンサベーストビヘービアによって実現し、走行環境に関する詳細なモデルを持たなくとも移動ロボットのナビゲーションが可能となるパラダイムを示したこと。

(b) ソナーリングより得られる情報に基づいてセンサベーストビヘービアを行なう場合のセンシング能力の限界を示し、超音波による測距原理をよく考慮する必要があることを示したこと。

(c) 広い角度範囲にわたり、計測を行なうレーザレンジセンサを扇状光の投射器と超広角 TV カメラを用いて設計・製作し、特に超広角 TV カメラを利用する場合の画像処理手法についても知見を得たこと

の3点をここで挙げるができる。

(a) に記したパラダイムは移動ロボットの屋内ナビゲーションに関するものとしてはこれまであまり注目されなかったものである。移動ロボットが、病院や図書館の内部において書籍や物品の搬送に用いられるケースも増加しており、また最近では巨大な TV 局やオフィスビル内での警備に移動ロボットが使われるケースもある。これらの環境では、特に移動ロボットの行動範囲が広範囲で長距離にわたる場合には、走行経路の指示のための負荷も軽減され、(a) で示したパラダイムが有効であると考えられる。本論文で示した実施・実験例はこのようなパラダイムに基づくひとつの具体例であり、今後このようなパラダイムに基づくナビゲーションを具体的な環境に適用する場合、十分に参考となる内容を含むものと信じている。

また、Denning 社の移動ロボットや NOMAD と呼ばれるロボットなど商業的に販売されている移動ロボットにもソナーリングが装備されているケースが多い。しかし、上記 (b) の成果は、ソナーリングの環境センサ能力を過信してはならないことを示している。ソナーリングを装備した移動ロボットを利用して特にセンサベーストビヘービアによる行動記述を行なおうとする場合には、センサの近くに対象物があっても対象物を見失うケースがあることに注意しなければならない。このことはソナーリングを利用して移動ロボットの行動プログラムを記述するプログラマすべてに対して有用



な情報である。

さらに (c) で示したレーザレンジセンサは、TV カメラとレーザ投光器によりコンパクトで安価に構成でき、暗い照明の中を夜間走行するような警備用の移動ロボットなどの、方向分解能の高いレンジセンサとして十分に応用できる。

### 9.3 結論

以上、本研究では屋内を移動するロボットのための詳細な環境モデルを用いない屋内ナビゲーションに関するテーマを取り扱った。当然ながら、本研究において示したシステムによって屋内を走行させる移動ロボットのナビゲーション問題をすべて解決するものではない。しかし、前節の (a) に記したようなパラダイムをロボットの屋内環境走行について導入し、目的としたナビゲーションを実現し、それが実現できる技術を開発し、具体的な実施・実験例を示したことは、大きな成果である。よって、本研究では、移動ロボットの詳細な環境モデルを用いない移動ロボットの距離センサ情報に基づく屋内ナビゲーションに関する研究に多くの知見を与え貢献を行なった。

## 謝辞

本論文は、筆者が筑波大学大学院工学研究科電子情報工学専攻に在学中、工学博士 油田信一 筑波大学電子情報工学系教授 と工学博士 坪内孝司 筑波大学電子情報工学系助教授 の御指導のもとで行なわれた研究をまとめたものである。

油田教授には、研究の進め方、論文の書き方、学会発表の方法など様々なことをわかりやすく熱心に御指導いただいた。また、油田教授には他大学や研究所の様々なロボットの研究室に見学に連れて行っていただいたり、著名なロボット研究者の講演を筑波大学で聞かせていただいた。このことは、本論文をまとめるのに大変有益であった。

坪内助教授には、博士課程に進学してから主に、研究の進め方、論文の書き方、学会発表の方法などの御指導をいただいた。また、坪内助教授には本論文をまとめるにあたり、年末年始の休みを返上して熱心な御指導をいただいた。

工学博士 太田道男 筑波大学構造工学系教授、工学博士 宮本定明 筑波大学電子情報工学系教授、そして、工学博士 津川定之 機械技術研究所物理情報部知識工学研究室長 (筑波大学連携大学院教授) には本論文をまとめるにあたり、有意義な御指導、御助言をいただいた。

博士(工学) 大矢 晃久 筑波大学電子情報工学系講師には、本論文の発表練習において、夜遅くまで議論に参加していただき、貴重な御指導、および御助言をいただいた。

博士(工学) 前山 祥一 筑波大学電子情報工学系助手には、在学中、ロボットシステムのハードウェアやソフトウェア、および研究の進め方に関して様々な御助言、御指導をいただいた。

Prof. Arthur C. Sanderson (Rensselaer Polytechnic Institute)、Prof. Phillip McKerrow (Wollongong University)、Prof. Suparek Premvuti (Thammasat Uni-

versity)、Dr. Chatila Raja (L.A.A.S. in France)、Dr. Alexander Zelinsky (Australian National University)、Dr. Markus Rude (筑波大学 研究員)、そして、Dr. Maki Khalil Habib らには、折に触れ貴重な御助言をいただいた。

博士(工学) 飯島 純一 氏には、年3回のシンポジウムを通して、研究に対する貴重な御助言をいただいた。

またこの研究は、筆者の研究室の研究プロジェクトが長年にわたって積み重ねてきた研究成果があっではじめてこの研究が可能になったものである。超音波距離センサ系の開発に携わった宗 敏鎬氏(研究室 OB)、渡辺 豊氏(現 日立建機)、大野 孝之氏(現 日立建機)、画像処理系の開発に携わった石川 忠明氏(現 日立製作所 機械研究所)、広瀬 哲也氏(現 NEC)、豊嶋 依里氏(現 日立製作所)、走行制御系の開発に携わった工学博士 飯田 重喜氏(現 防衛庁)、上田 暁彦氏(現 SONY) オペレーティングシステムの開発に携わった博士(工学) 木元 克美氏(現 慶應義塾大学助手)、宮井 岳志氏(現 SONY)、そして、ここに書ききれない大勢のプロジェクト関係者に、直接的または間接的にお力をいただいた。

小柳 栄次 横須賀工業高校 機械科 教諭(筑波大学大学院博士課程)には、ロボットに搭載するレーザレンジセンサの製作に並々ならぬ御尽力をいただいた。

また、学会や酒の席などにおいて、大阪大学の 細田 耕 先生、東北大学の 近野 敦先生、大阪大学の 鈴木 昭二 先生、京都大学の 栗栖 正充 先生、中央大学の 梅田 和昇 先生、東京大学の 相山 康道 先生、広島大学の 原田 研介 先生、奈良先端大学の 松本 吉央 先生、大分県立芸術文化短期大学の 吉岡 孝 先生、CMU の 永谷 圭司 さん、電総研の 山崎 信行 さん、比留川 博久 さん、Gordon Cheng さん、機械研の 加藤 晋 さん、Wollongong 大学 博士課程の David Jung さん、東京大学 の 平原 清隆 さん、そして、筑波大学 博士課程の 渡嘉敷 浩樹 さんらとは研究に有益なディスカッションをさせていただいた。

博士(工学) 春日 智恵 芝浦工業大学 電気工学科主任教授 は、芝浦工業大学において非常勤講師の機会を与えて下さった。100人程度の学生の前で毎週講義を行なうことは、筆者にとってとてもよい経験になった。

矢田 晃子 さん(筑波大学 博士課程)には、本論文を提出する際に、御尽力いただいた。

筑波大学知能ロボット研究室のメンバーには、研究上で貴重な御助言をいただいた

り、研究活動以外でも素晴らしい時間を共有させていただいた。

日本育英会には、学費および生活費を3年間にわたって貸与していただいた。

そして、最後に、両親や家族には、長年にわたり経済的精神的に並々ならぬ御支援をいただいた。

以上の方々および、ここに書ききれないこれまでにお世話になった全ての方々に対して、心から深く感謝の意を表す。

## 参考文献

- [1] James L. Crowley: "Navigation for an Intelligent Mobile Robot", IEEE Journal of Robotics and Automation, Vol.RA-1, No.1, pp. 31-41, March 1985.
- [2] Ingemar J. Cox: "Blanche — An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle", IEEE Transactions on Robotics and Automation, Vol. 7, No. 2, pp.193-204, April 1991.
- [3] T. Skewis, J. Evans, V. Lumelsky, B. Krishnamurthy, B. Barrows: "Motion Planning for a Hospital Transport Robot", Proceedings of International Conference on Robotics and Automation (ICRA '91) pp. 58-63, April 1991.
- [4] Shin'ichi Yuta and Keiji Nagatani: "Realizing a Navigation Function on Small-sized Autonomous Robot in In-door Environment", Robots for Australian Industries, pp. 237-244, 1995.
- [5] Javier Gonzalez, Anibal Ollero and Antono Reina: "Map Building for a Mobile Robot equipped with a 2D Laser Rangefinder", Proceedings of International Conference on Robotics and Automation (ICRA '94) pp. 1904-1909, 1994.
- [6] Shoichi. Maeyama, Akihisa. Ohya and Shin'ichi Yuta: "Outdoor landmark map generation through human assisted route teaching for mobile robot", Proceedings of International Conference on Intelligent Robots and Systems, pp. 957-962, 1996.
- [7] S. Tsugawa, et al.: "An automobile with artificial intelligence," Proc. of 6th IJCAI, 1979.

- 
- [8] K-D. Kuhnert and V. Greafe: "Vision Systems for Autonomous Mobility," Proc. of International Workshop on Intelligent Robots and Systems, pp. 477-482, 1988.
- [9] Rodney A. Brooks: "A Robust Layered Control System For A Mobile Robot", IEEE Journal of Robotics and Automation, Vol. RA-2, No. 1, pp. 14-23, March 1986.
- [10] Ronald C. Arkin: "Behavior-Based Robotics", The MIT Press, 1998.
- [11] 西川 和弘、森英雄: "簡潔な経路情報を基にした移動ロボットの誘導法", 日本ロボット学会誌, Vol. 12, No. 8, pp.1183-1191, 1994.
- [12] V. Lumelsky and A. A. Stepanov: "Path-planning Strategies for a point mobile Automation Moving amidst Unknown Obstacles of arbitrary Shape," *Algorithmica*, Vol. 2, pp. 403-430, 1987.
- [13] Hiroshi Noborio, Takashi Yoshioka and Shoji Tominaga: "On the sensor-based navigation by changing a direction to follow an encountered obstacle", Proceedings of International Conference on Intelligent Robot and Systems (IROS 1997), pp. 510-517. 1997.
- [14] 柴田 史久、芦田 昌也、角所 考、北橋 忠宏: 「移動ロボットによる言語的経路指示理解のための記号情報とパターン情報の対応付け」, 人工知能学会誌, Vol. 13, No. 2, pp. 221-230, March 1998.
- [15] P.van.Turenout,G.Honderd:"Following a Wall with a Mobile Robot using Ultrasonic Sensors" ,Proc. of the 1992 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'92),pp.1451-1456(1992).
- [16] P.van.Turenout,G.Honderd:"Wall-following Control of a Mobile Robot" ,Proc. of the 1992 IEEE Int. Conf. on Robotics and Automation(RA'92), pp.280-285(1992).

- [17] 松本 吉央、池田 和徳、稲邑 哲也、稲葉 雅幸、井上 博允: 「画像列に基づくナビゲーションのための自律的な地図の獲得」, ロボティクス・メカトロニクス講演会'98 講演論文集, 1BI3-2, June 1998.
- [18] Charles E. Thorpe: "Vision and Navigation The Carnegie Mellon Navlab", Kluwer Academic Publishers.
- [19] 飯田重喜、斎藤雄一、油田信一、飯島純一: "スリット光を用いたレーザレンジファインダを有する自律型移動ロボットとその障害物回避走行", 第5回知能移動ロボットシンポジウム, pp.121-126, 1990.
- [20] Johann Borenstein and Liqiang Feng: "Measurement and Correction of Systematic Odometry Errors in Mobile Robots", IEEE Transactions on Robotics and Automation, Vol. 12, No. 6, pp.869-880, December 1996.
- [21] 渡辺 豊, 油田 信一: "移動ロボットの自己位置推定におけるセンサデータフュージョン", 日本機械学会 ロボット・メカトロニクス講演会予稿集, pp. 99-100, 1989.
- [22] T. Tsubouchi and S. Yuta: "Map Assisted Vision System of mobile robots for reckoning in a building environment," Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1978-1984, 1987.
- [23] 飯田 重喜," 車輪型自律移動ロボットの走行制御システムに関する研究", 筑波大学博士論文,1991
- [24] Shigeki IIDA , Shin'ichi YUTA: "Vehicle Command System and Trajectory Control for Autonomous Mobile Robots" ,Proc. of the 1992 IEEE/RSJ Int. Workshop on Intelligent Robots and Systems(IROS'91), pp.212-217(1991).
- [25] 木元 克美, 油田 信一: "移動ロボットのための超音波センサシミュレータ", 日本ロボット学会誌, Vol.13 No.2 pp.137-144, 1995.
- [26] K. Kimoto and S. Yuta, "A simulator for programming the behavior of an autonomous sensor based mobile robot", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 1431-1438, 1992.

- [27] Yoshinobu Ando and Shin'ichi Yuta : "Following a Wall by an Autonomous Mobile Robot with a Sonar-Ring", Proc. of 1995 IEEE Int. Conf. on Robotics and Automation (ICRA'95), pp. 2599-2606.
- [28] 安藤 吉伸, 木元 克美, 油田 信一: "自律移動ロボットのセンサベースな行動のプログラミングのためのシミュレータ - 移動する障害物への対応-", 知能移動ロボットシンポジウム'94, pp. 142-147, 1994.
- [29] Yoshinobu Ando, Takashi Tsubouchi and Shin'ichi Yuta : "An Algorithm and its Limitation of Sonar-Based Wall Following for an Autonomous Mobile Robot and Experiments in Real Environment", Proc. of 1995 National Conf. of the Australian Robot Association (RAI'95), pp. 245-252.
- [30] 安藤 吉伸, 油田 信一: "ソナーリングを用いた自律移動ロボットのウォールフォロ잉", 日本ロボット学会学術講演会予稿集, pp. 1091-1092, 1994.
- [31] 安藤 吉伸 坪内 孝司 油田 信一: 「レーザレンジセンサを用いた移動ロボットの壁沿い走行」, ロボティクス・メカトロニクス講演会'98 講演論文集, June 1998.
- [32] 安藤 吉伸 坪内 孝司 油田 信一: 「超広角カメラと扇状光による移動ロボット用レーザ距離センサの開発と線分あてはめを用いた壁沿い走行」, 第3回ロボティクスシンポジア 予稿集, pp.253-260, May 1998.
- [33] Scott A. Walter: "The Sonar Ring: Obstacle for a Mobile Robot", Proc. of the 1987 IEEE Int. Conf. on Robotics and Automation(RA'87), pp.1574-1579, 1987.
- [34] Phillip John McKerrow: "Control of Ultrasonic Mapping by a Mobile Robot with an Expert System", Proc. Robots for Competitive Industries, Brisbane, ARA, pp.300-309, 1993.
- [35] Erann Gat and Greg Dorais: "Robot Navigation by Conditional Sequencing", Proc. of 1994 IEEE Int. Conf. on Robotics and Automation (ICRA'94), pp. 1293-1299, 1994.



- [36] Eran Gat: "Robust Low-computation Sensor-driven Control for Task-Directed Navigation", Proc. of 1991 IEEE Int. Conf. on Robotics and Automation (ICRA'91), pp. 2484-2489, 1991.
- [37] T. Skewis and V. Lumelsky: "Experiments with a Mobile Robot Operating in a Cluttered Unknown Environment", Proc. of the 1992 IEEE Int. Conf. on Robotics and Automation (RA'92), pp. 1482-1487, 1992.
- [38] H. Noborio and T. Yoshioka, "An on-line and deadlock-free path-planning algorithm based on world topology", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 1425-1430, 1993.
- [39] Keiji Nagatani and Shin'ichi Yuta: "Path and Sensing Point Planning for Mobile Robot Navigation to Minimize the Risk of Collision", IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 93), pp. 2199-2203, 1993.
- [40] Rodney A. Brooks: "A Robot that Walks; Emergent Behavior from a Carefully Evolved Network", 1989 IEEE Int. Conf. on Robotics and Automation (RA89), pp. 692-694, 1989.
- [41] 油田 信一, 飯島 純一: "自律移動ロボットのコントローラのためのマルチプロセッサアーキテクチャ", 第5回知能移動ロボットシンポジウム, 1990.
- [42] Takashi Tsubouchi, Kazuyuki Hiraoka, Tomohide Naniwa and Suguru Arimoto, "A Mobile Robot Navigation Scheme for an Environment with Multiple Moving Obstacles", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1791-1798, 1992.
- [43] 石川 繁樹, 浅香 俊一: "移動障害物を含むような動的変化を伴う走行環境における自律移動ロボットの走行誘導方式", 日本ロボット学会誌 Vol.11, No.6, pp. 856-867, 1993.
- [44] Tai-Jee Pan and Ren C. Luo, "Motion Planning for Robots in a Dynamic Environment with Moving Obstacles", IEEE International Conference on Robotics and Automation, pp. 578-583, 1990.

- 
- [45] Shoichi Maeyama, Akihisa Ohya and Shin'ichi Yuta: "Positioning by Tree Detection Sensor and Dead Reckoning for Outdoor Navigation of a Mobile Robot", Proc. of 1994 IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI'94), pp. 653-660.
- [46] Shoichi Maeyama, Nobuyuki Ishikawa and Shin'ichi Yuta: "Rule based filtering and fusion of odometry and gyroscope for a fail safe dead reckoning system of a mobile robot", Proc. of Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems, pp 541-548, 1996.
- [47] T.Sekimoto, T.Tsubouchi and S.Yuta: "A remote operation using visual images and a pointing device for mobile robot control," Proc. of the 2nd International Conference on Mechatronics and Machine Vision in Practice, pp.260-263, 1995.
- [48] Gerhard Weis, Christopher Wetzler, Ewald von Puttkamer: "Keeping Track of Position and Orientation of Moving Indoor Systems by Correlation of Range-Finder Scans", International Conference on Intelligent Robots and Systems (IROS'94), pp.595-601, 1994.
- [49] Yoshio Matsumoto, Masayuki Inaba and Hirochika Inoue: "Visual Navigation using View-Sequenced Route Representation", Proc. of Int. Conf. on Robotics and Automation, pp 83-88, 1996.
- [50] J.Vandorpe, H.Van Brussel, H.Xu: "Exact Dynamic Map Building for a Mobile Robot using Geometrical Primitives Produced by a 2D Range Finder" , International Conference on Robotics and Automation (ICRA'96), pp.901-908, 1996.
- [51] Richard O. Duda and Peter E. Hart: "Pattern Classification and Scene Analysis", Wiley, New York, 1973.
- [52] Shooji Suzuki, Shin'ichi Yuta and Jun'ichi Iijima: "A Consideration on the Programming Method of Sensor-Driven Robot Behavior Based on the Action

---

Mode Representation", Proc. of The 20th International Symposium on Industrial Robots(ISIR), pp.127-134, October 1989.

## 研究業績

### 1. 査読のある論文誌

- (a) Yoshinobu ANDO, Takashi TSUBOUCHI and Shin'ichi YUTA: "A Reactive Wall Following Algorithm and its Behavior of an Autonomous Mobile Robot with Sonar-Ring", Journal of Robotics and Mechatronics, Vol.8, No.1, pp.33-39, 1996.
- (b) Yoshinobu ANDO, Takashi TSUBOUCHI and Shin'ichi YUTA: "Development of Ultra Wide Angle Laser Range Sensor and Navigation of a Mobile Robot in a Corridor Environment", Journal of Robotics and Mechatronics, (採録決定).

### 2. 国際会議

- (a) Yoshinobu ANDO and Shin'ichi YUTA: "Following a Wall by an Autonomous Mobile Robot with a Sonar-Ring", Proceedings 1995 IEEE International Conference on Robotics and Automation (ICRA '95), Volume 3, pp2599-pp2606, May 1995
- (b) Yoshinobu ANDO, Takashi TSUBOUCHI and Shin'ichi YUTA: "An Algorithm and its Limitation of Sonar-Based Wall Following for an Autonomous Mobile Robot and Experiments in Real Environment", Robots for Australian Industries, Proceedings of the 1995 National Conference of the Australian Robot Association, pp.245-252, July 1995.
- (c) Yoshinobu ANDO, Takashi TSUBOUCHI and Shin'ichi YUTA: "A Mobile Robot Navigation Using Both Wall Following Control and Position

Based Control”, Video Proceedings of (IECON '96), 1996.

- (d) Takashi Tsubouchi, Yoshinobu Ando and Shin'ichi Yuta: "A mobile robot navigation by means of range data from ultra wide angle laser sensor and simple linguistic behavior instructions", Proceedings of International Conference on Experimental Robotics (ISER '99), March 1999.(accepted).

### 3. 国内会議

- (a) 安藤 吉伸、木元 克美、油田 信一：「自律移動ロボットのセンサーベーストな行動のプログラミングのためのシミュレータ — 移動する障害物への対応 —」, "A simulator for programing the behavior of an autonomous sensor-based mobile robot — Dealing with a moving obstacle —", 知能移動ロボットシンポジウム'94 予稿集, pp.142-147, 1994.
- (b) 安藤 吉伸、油田 信一：「ソナーリングを用いた自律移動ロボットのウォール・フォローイング」、"Following a wall with an Autonomous Mobile Robot using a Sonar Ring", 1994 第12回日本ロボット学会学術講演会予稿集, No.3, pp1091-pp1092, 1994,11.
- (c) 安藤 吉伸、坪内孝司、油田 信一：「ウォールフォローイングと位置情報に基づく動作を併用した移動ロボットのナビゲーション」, "A Mobile Robot Navigation Using both Wall Following and Position Based Control", 日本ロボット学会第5回ロボットシンポジウム予稿集, pp.87-pp.92, 1995 5.
- (d) 安藤 吉伸、坪内 孝司、油田 信一：「超音波センサベーストウォールフォローイングアルゴリズムとその限界」 An algorithm and its limitation of Sonar-based wall following by an autonomous mobile robot, 1995 第13回日本ロボット学会学術講演会予稿集, No.2, pp585-pp586, 1995,11.
- (e) 安藤 吉伸、坪内 孝司、油田 信一：「自立移動ロボットによる非数量的かつ反射的な手法を用いる実環境中の自律ナビゲーション」 An Autonomous Mobile Robot Navigation using Reactive Wall Following Algorithm in Real Environment, 1996 ロボティクス・メカトロニクス講演会'96 予稿集, Vol.A, pp263-pp266, 1996,6.

- (f) 安藤 吉伸、坪内 孝司、油田信一：「自立移動ロボットによる非数量的かつ反射的な手法を用いる実環境中の自律ナビゲーション」  
An Autonomous Mobile Robot Navigation using Reactive Wall Following Algorithm in Real Environment, 1996 ロボティクス・メカトロニクス講演会'96 ビデオプロシーディングス, 1996,6.
- (g) 安藤 吉伸、坪内 孝司、油田信一：「超広角カメラと扇状光を用いた移動ロボット用光学式距離センサの開発」, The development of optical range sensor using ultra-wide angle camera and fan beam laser, 1997 第15回日本ロボット学会学術講演会予稿集, No.2, pp593-pp594, 1997,9.
- (h) 安藤 吉伸、坪内 孝司、油田信一：「超広角カメラと扇状光による移動ロボット用レーザ距離センサの開発と線分あてはめを用いた壁沿い走行」, Development of Laser Range Finder Made of Ultra-wide-lens Cameras and Fan Beam Lasers for Mobile Robot, and Following the Wall by Fitting Line Segments to Sensing Data, 第3回ロボティクスシンポジウム予稿集, pp253-pp260, 1998,5.
- (i) 安藤 吉伸 坪内 孝司 油田 信一: 「レーザレンジセンサを用いた移動ロボットの壁沿い走行」, Wall Following by Autonomous Mobile Robot with Laser Range Sensor, ロボティクス・メカトロニクス講演会'98 講演論文集, 1998,6.
- (j) 安藤 吉伸、坪内 孝司、油田信一：「移動ロボットの言語的な行動指示の列挙による広範囲屋内ナビゲーション」, Mobile Robot Navigation using Linguistic Instruction in Wide Area Indoor Environment, 1998 第16回日本ロボット学会学術講演会予稿集, pp1599-pp1600, 1998,9.

筑波大学附属図書館



1 00990 12361 8

本学関係