

自律移動ロボットの知的行動決定アルゴリズムに関する研究

前田, 陽一郎 / MAEDA, Yoichiro

(開始ページ / Start Page)

1

(終了ページ / End Page)

65

(発行年 / Year)

1994-03-24

(学位授与番号 / Degree Number)

32675乙第77号

(学位授与年月日 / Date of Granted)

1994-03-24

(学位名 / Degree Name)

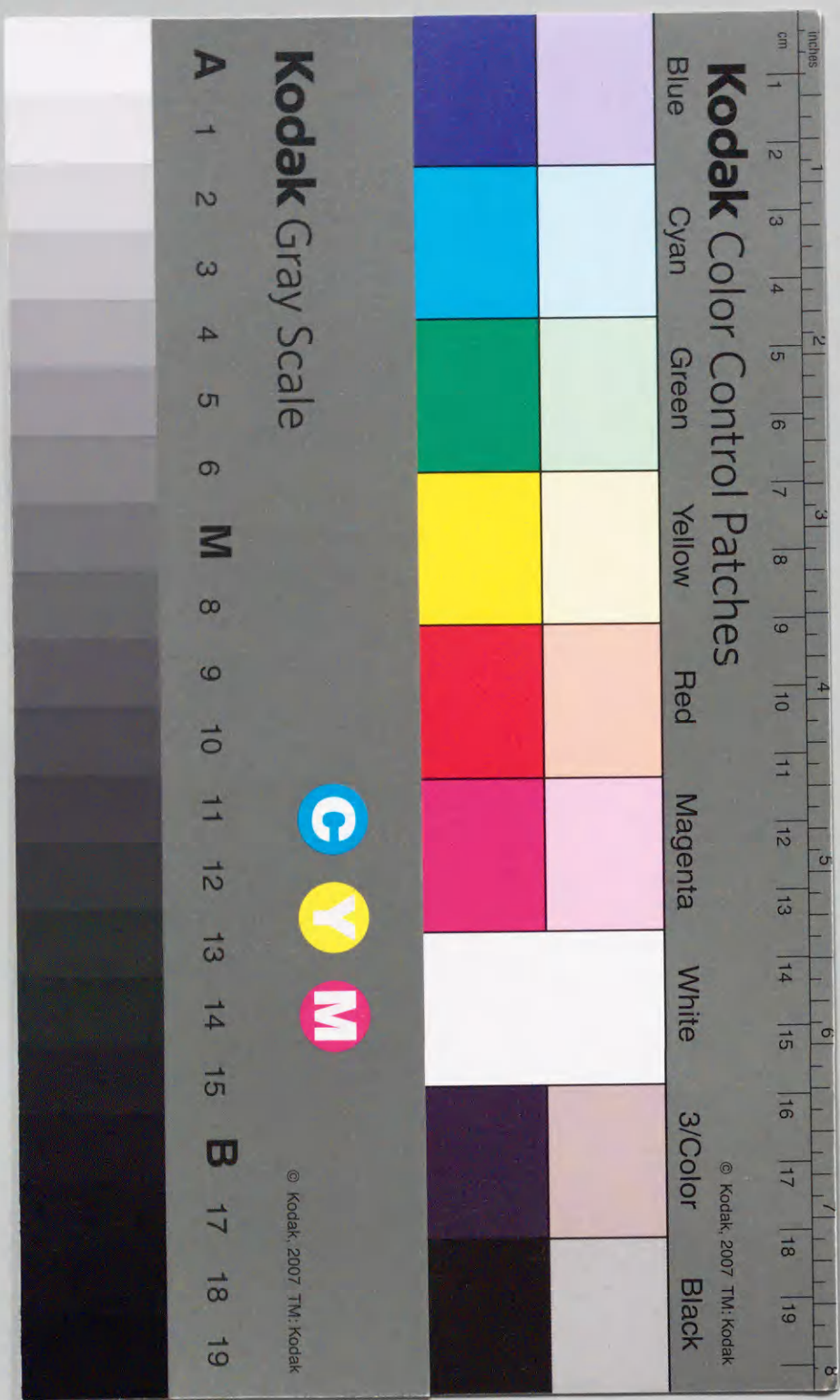
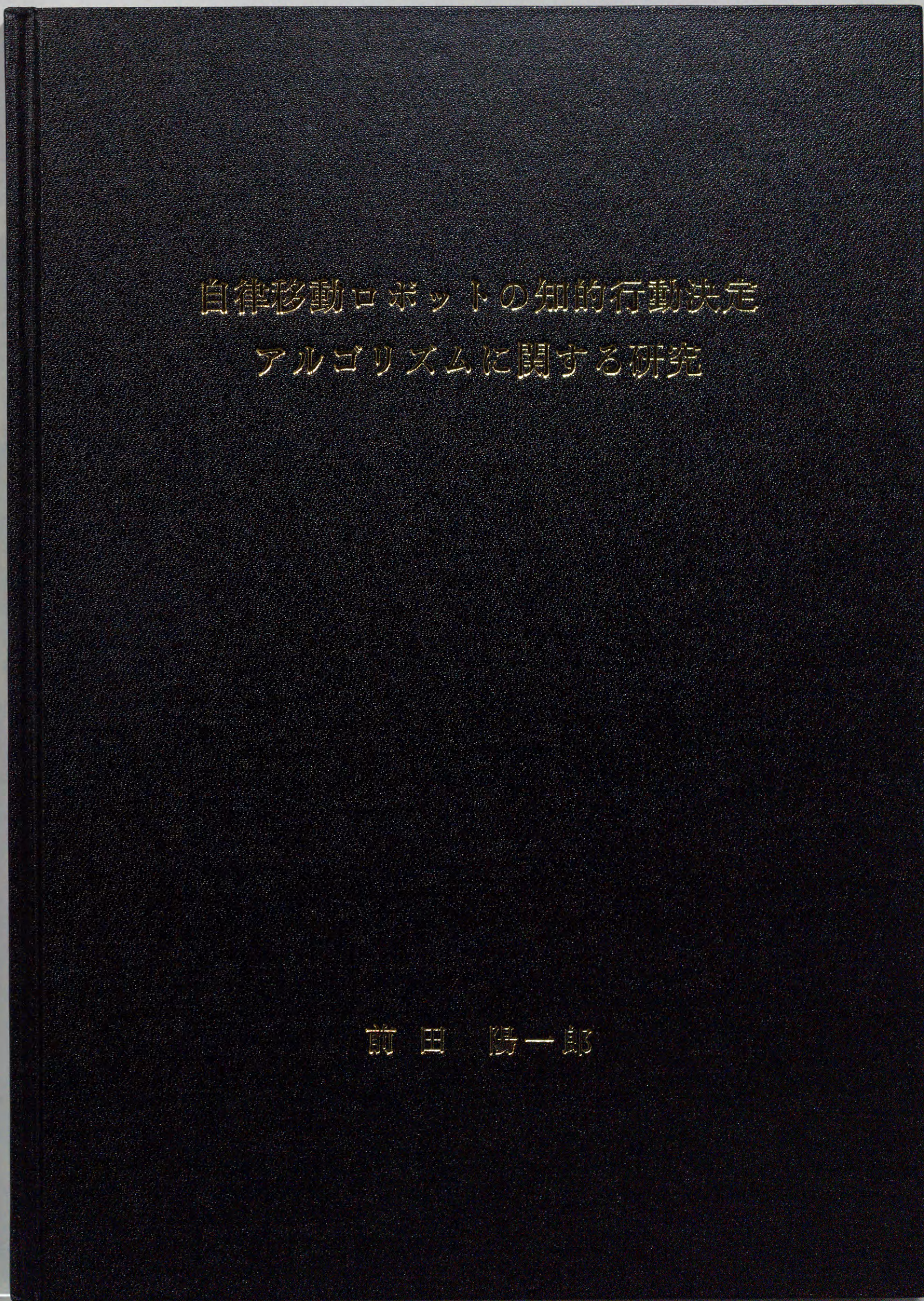
博士(工学)

(学位授与機関 / Degree Grantor)

法政大学 (Hosei University)

(URL)

<https://doi.org/10.11501/3073509>



①

法政大学 博士論文

自律移動ロボットの知的行動決定
アルゴリズムに関する研究

平成6年1月

前田 陽一郎

Abstract

Intelligent Behavior-Decision Algorithm for Autonomous Mobile Robots

Yoichiro MAEDA

Abstract

The use of fuzzy logic in control applications in recent years has been remarkable. It is no exaggeration to say that fuzzy logic has become recognized as the best model available for representing the knowledge of an experienced or skilled professional. In particular, the practical significance of what is called "fuzzy control", a typical fuzzy reasoning method proposed by E.H.Mamdani, has already been recognized in the field of machine control. Among the many features of fuzzy logic based knowledge representation, the most advantageous include knowledge structures with the ability to represent human knowledge and excellent rule maintainability due to the availability of compact knowledge representations.

On the other hand, in the field of robotics, new research approaches oriented to the behavior of creatures, such as research for the hierarchical control mechanism and the modeling of insect motion, have been increasing. This constitutes a new approach which differs from conventional robot control in that it deals with realizations of the macro level and the hard-to-express function called "behavior". However, the biggest task involved in making robots highly intelligent is the simulation of the complex judgement mechanism used by humans, and such things as reasoning and common sense.

When an autonomous mobile robot moves around a complex environment, the ambiguous state recognition is one of the most important problems for describing "behavior". The macro expression of the state and behavior is indispensable in attempting to represent "human-like" behavior-decision abilities in the intelligent robot. Accordingly, the realization of a high-order intelligent robot with autonomous behavior-decision abilities requires an intelligent information processing and a model construction method. It is also important to have a methodology capable of describing complicated motion sequences macroscopically and flexibly. Therefore, I have attempted to construct the behavior-decision algorithm for a mobile robot simulating the flexible circumstantial judgement and adaptable ability of human beings who always decide precisely in recognizing the ambiguous states of environments and purposes of behavior when moving.

This research contains themes mainly concerning the algorithms for planning, obstacle avoidance and behavior-decision functions, and the inference shell and the locomotive mechanism for support tools to include almost all basic functions necessary for an intelligent mobile robot. This paper presents the following five research approaches performed for the final purpose of producing a high-order behavior-decision ability in an autonomous mobile robot by using fuzzy logic to capture various ambiguities existing in a robot:

Abstract

- 1) high-order locomotive mechanism,
- 2) obstacle avoidance control algorithm,
- 3) path planning algorithm,
- 4) behavior-decision algorithm,
- 5) fuzzy inference shell for intelligent behavior-decision algorithm.

In the first approach, a prototype of the multifunctional robot vehicle (MRV) has been developed. This robot has various locomotive modes of moving in all directions on the ground by 4 steering wheels, and striding over obstacles, crossing over ditches and climbing stairs by 4 rotational crawler legs. In this paper, I will discuss the sequence of motions for striding an obstacle such as a pipe and analyze the static stability of this motion. This robot's ability to adapt to the ground has been confirmed by various experiments.

The second approach presents a method for performing obstacle avoidance control for mobile robots in an environment with moving obstacles. This method consists of recognizing the degree of danger and making a decision concerning avoidance actions by using a strategy map. The static and dynamic degree of danger between a robot and an obstacle are decided by fuzzy reasoning. The map describing this control strategy is constructed by the acquisition of the skill of an operator by using the CMAC learning algorithm. Simulation results show the use of this method provided that the robot does not come to a deadlock as with the Artificial Potential Field method.

The third approach presents the heuristic path search algorithm based on fuzzy reasoning and the intelligent path planning method using an ambiguous map. In the former method, I proposed the search algorithm with high efficiency in comparison with alfa-beta pruning by checking the effect of the search based on fuzzy reasoning for problems of the complex Grid-map path planning. In the simulations using a 3-dimensional Grid-map, it was proved that this heuristic path search algorithm can obtain an optimal solution with search times of about 1/5 for the perfect search, about 1/3 for beta pruning. In the latter method, I can treat the map with ambiguous information and generate the path as a sequence of subgoals. Simulations of the path planning confirm the ability of the effective path generation from results of the path planning for various situations.

In the fourth approach, I proposed a construction method for the behavior-decision algorithm using fuzzy algorithms capable of expressing sequence flow which includes a mixture of both crisp and fuzzy processing. Furthermore, I also proposed a method of tuning algorithms for giving robots the autonomous ability to judge purposes of action like humans. Results from computer simulations supposed a simple indoor environment and locomotive experiments using a real autonomous mobile robot were also shown to prove the effectiveness of this method. Furthermore, I have attempted to evaluate the fuzziness in a fuzzy algorithm by modelling and analysis based on Petri Nets. From results of these simulations, the change in ambiguity in an example of the behavior-decision fuzzy algorithm were confirmed.

In the final approach, I developed a fuzzy inference shell FRASH for use in realtime intelligent control applications. This shell possesses the following various functions: a fast inference engine, online tuning libraries, an offline editor and an inference display for the fuzzy inference, fuzzy

Abstract

frame-type data structures and a fuzzy algorithm compiler. Benchmark tests and simple simulation proved that the fuzzy inference speed is enough fast for the realtime machine control and the function of the fuzzy frame and the fuzzy algorithm are useful for describing control algorithms including macro decision making.

Of all the research stated above, I have attempted to add the human-like judgement for ambiguous states to behavior-decision abilities of a robot. By applying the algorithms proposed in this paper, I have included the "human-like" functions in behavior-decision rules which are important to an autonomous mobile robot and also express simple rules which are easy to understand for humans by describing macro rules. And finally, I believe that the application of fuzzy logic to intelligent robots gives possibilities of applying fuzzy information processing techniques to research in the high-order intelligence and suggests a new direction for realizing a human-friendly robot.

要 旨

自律移動ロボットの知的行動決定アルゴリズムに関する研究

要 旨

近年、ファジィ理論の制御への応用はめざましく、ファジィ理論が経験者や熟練者のノウハウを記述するために最適な知識表現形式であると認識されていると言っても過言ではない。中でもいわゆる「ファジィ制御」と呼ばれるMamdaniに代表されるファジィ推論方式は制御分野における実用的意義がすでに認められている。ファジィ理論を用いた知識表現手法にはいくつかの特長があるが、他と比べて最も大きな利点は人間のノウハウを記述しやすい知識構造を有しておりかつ知識を集約表現できるためルールメンテナンス性が極めて良いことであろう。

一方ロボット分野を見た場合、階層的制御メカニズムや昆虫の行動モデルを提唱した研究などの生物の行動を指向した新しい知能ロボットの研究アプローチが盛んになっている。これらはいずれも"行動 (behavior)" というマクロなレベルでかつ表現が困難な機能の実現を対象としている点で従来のロボット制御とは異なった新たな研究方向であると考えられる。しかしながら、今までのところロボットの高度知能化においては、人間の複雑な判断機構、理性・常識などの模擬が最も大きな課題として残されているのが現状である。

複雑な環境中で行動する自律移動ロボットの場合、あいまい状態認識が"行動"を表現する上で重要な問題の一つとなる。また知能ロボットにおいて"人間らしい"行動決定能力を表現するには、状態および行動のマクロ表現が不可欠である。従って、知能ロボットの自律行動決定機能の高度化を考えた場合、知的情報処理手法やモデル構築手法はもとより、複雑な動作シーケンスをマクロにかつ柔軟に記述できる方法論が最も重要になってくる。そこで筆者らは、移動ロボットを対象として移動する際、環境や行動目的に対して比較的あいまいな状況を常に認識しながら的確な判断を行っている人間の柔軟な状況判断・適応能力を模擬した行動決定アルゴリズムを構築することを試みた。

本研究では、知能移動ロボットに必要とされる基本的な機能を包含するため、プランニング、障害物回避、行動決定に関するアルゴリズムを中心として、これを支援するツールとして推論シェル、移動機構の各テーマを主に対象としている。本論文ではロボットに内在するさまざまなあいまい性に着目し、ファジィ理論の適用により自律移動ロボットの行動決定機能の高度化を行うこと最終目標として実施した以下の5つの研究アプローチについて述べる。

- 1) 高度移動メカニズム
- 2) 障害物回避制御アルゴリズム
- 3) 経路決定アルゴリズム
- 4) 行動決定アルゴリズム
- 5) 知的行動決定アルゴリズム構築用ファジィシェル

高度移動メカニズムの研究では、多機能移動ロボット (MRV) の試作機を開発した。MRVはステアリング可能な4つの車輪による全方向移動と、4つのクローラ脚による障害物またぎ越え、溝越えおよび階段昇降などのさまざまな移動モードをもっている。本論文では、パイプなどの障害物またぎ越えの動作シーケンスおよび動作の静的安定解析を行った。このロボットの対地適応能力はさまざまな実験により確認された。

障害物回避制御アルゴリズムの研究では、本論文では移動物体を有する環境における移動ロボットの障害物回避制御に対し、危険度認識と戦略マップを用いた回避行動決定から構成される手法について述べる。ロ

要 旨

ロボットと障害物間の静的および動的危険度はファジィ推論によって決定される。また制御戦略を記述したマップは、CMAC学習アルゴリズムを用いてあらかじめオペレータの操作特性をスキル獲得させることにより構築している。シミュレーション結果より、本方式を用いると人工ポテンシャル法の場合ようなデッドロック現象を回避できることが確認された。

経路決定アルゴリズムの研究では、ファジィ推論に基づくヒューリスティック経路探索アルゴリズムとあいまいな地図を用いた知的経路計画手法について述べる。前者の手法では、複雑なグリッドマップの経路探索問題において、ファジィ推論を用いて探索有効度をチェックすることにより、従来の枝刈り法に比べ探索効率が大幅に高い探索アルゴリズムを提案した。立体グリッドマップを用いたシミュレーションにより、本経路探索アルゴリズムは総当たりりに比べ約1/5、 β カットに比べても約1/3程度の探索時間で最適解を得ることが可能であることがわかった。後者の手法では、あいまい情報を有するマップを扱うことができ、サブゴール点列による経路生成が可能である。シミュレーションにより、さまざまな状況に対する経路計画の結果から効率的な経路生成能力が確認できた。

行動決定アルゴリズムの研究では、クリスプ量およびファジィ量が混在するようなシーケンスフローが表現可能なファジィアルゴリズムの概念を用いた行動決定アルゴリズムの構築手法を提案した。また人間に近い目的自律判断機能をもたせるためのアルゴリズムチューニング方法についても提案した。本方式の簡単な屋内環境を想定した計算機シミュレーションおよび実際の自律移動ロボットを用いた移動実験を行ったところ、本方式の有効性が示された。さらにベトリネットに基づくモデル化と解析によりファジィアルゴリズムにおけるあいまい性を評価することを試みた。シミュレーション結果から、行動決定ファジィアルゴリズムの例題におけるあいまい性の遷移状態が確認できた。

知的行動決定アルゴリズム構築用ファジィシェルの研究では、リアルタイム知的制御のためのファジィ推論シェルFRASHを開発した。このシェルは高速な推論エンジン、オンラインチューニングライブラリ、オフラインエディタ機能、推論状況表示機能、ファジィフレーム表現、ファジィアルゴリズムコンパイラなど数々の機能を有している。ベンチマークテストと簡単なシミュレーションによりファジィ推論機能はリアルタイムのメカトロ制御には十分高速であること、ファジィフレームおよびファジィアルゴリズム機能はマクロ意思決定を含む制御アルゴリズムの記述に有効であることを検証した。

本論文における以上の研究では、主にロボットにおける行動決定機能を対象として人間を模擬したあいまい状況判断能力を付加することを試みた。本論文で提案したアルゴリズムを用いることにより、自律移動ロボットにとって重要な行動決定ルールに「人間らしい」機能をもたせることができるだけでなく、マクロ記述により人間に可視性のよいシンプルなルール記述が可能となる。さらに知能ロボットへのファジィ理論のさまざまな適用を通して、知能化研究へのファジィ情報処理技術の応用可能性を探るとともに、ヒューマンフレンドリロボットの実現に向けて一つの方向性を示唆することができたと考える。

目 次

目 次

第1章 序 論.....	1
第2章 高度移動メカニズムに関する研究.....	6
2.1 緒 言.....	6
2.2 多機能移動ロボットMRVの開発.....	7
2.2.1 車輪・クローラ脚複合形多機能移動機構.....	7
2.2.2 多機能移動モード.....	8
2.2.3 適応動作シーケンス.....	9
2.2.4 障害物またぎ越え動作解析.....	13
2.2.5 実証評価実験.....	15
2.3 結 言.....	17
第3章 障害物回避制御アルゴリズムに関する研究.....	18
3.1 ファジィ障害物回避制御アルゴリズム.....	18
3.1.1 緒 言.....	18
3.1.2 問題設定.....	18
3.1.3 ファジィ障害物回避アルゴリズム.....	19
3.1.4 静的危険度の決定.....	20
3.1.5 動的危険度の決定.....	21
3.1.6 回避ベクトルおよび操舵ベクトルの決定.....	22
3.1.7 障害物回避シミュレーション.....	23
3.1.8 結 言.....	25
3.2 CMACニューラルネットワークを用いた障害物回避戦略の獲得方法.....	29
3.2.1 緒 言.....	29
3.2.2 CMAC学習アルゴリズム.....	29
3.2.3 オフラインにおける制御戦略の学習プロセス.....	30
3.2.4 シミュレーション.....	32
3.2.5 考 察.....	33
3.2.6 結 言.....	34

目次

第4章 経路決定アルゴリズムに関する研究.....	35
4.1 ファジィ推論を適用したグリッドマップ経路探索アルゴリズム.....	35
4.1.1 緒言.....	35
4.1.2 探索経路のモデル化.....	35
4.1.3 ヒューリスティック経路探索アルゴリズム.....	36
4.1.4 ファジィ推論の探索アルゴリズムへの適用.....	38
4.1.5 進退しきい値 λ の自動修正.....	40
4.1.6 シミュレーション結果.....	40
4.1.7 結言.....	43
4.2 あいまい評価に基づく知的経路計画手法.....	44
4.2.1 緒言.....	44
4.2.2 経路計画におけるあいまい性.....	44
4.2.3 経路計画アルゴリズム.....	47
4.2.4 領域分割.....	48
4.2.5 経路評価.....	50
4.2.6 サブゴール生成.....	52
4.2.7 有効性検証シミュレーション.....	55
4.2.8 結言.....	59
第5章 行動決定アルゴリズムに関する研究.....	60
5.1 自律移動ロボットの行動決定ファジィアルゴリズム.....	60
5.1.1 緒言.....	60
5.1.2 行動決定ファジィアルゴリズムの構築.....	60
5.1.3 修正ファジィアルゴリズムの提案.....	61
5.1.4 あいまい概念およびあいまい状況の記述.....	62
5.1.5 行動目的自律判断機能の記述.....	64
5.1.6 行動決定ファジィアルゴリズムの記述.....	66
5.1.7 有効性検証シミュレーション.....	66
5.1.8 実証評価実験.....	74
5.1.9 結言.....	76

目次

5.2 ペトリネットを利用したファジィアルゴリズムのあいまい性評価.....	77
5.2.1 緒言.....	77
5.2.2 ファジィアルゴリズムの問題点.....	77
5.2.3 ファジィ分岐処理によるファジィアルゴリズムの分類.....	77
5.2.4 ファジィアルゴリズムのペトリネット表現.....	79
5.2.5 ファジィペトリネット (FPN) の状態方程式.....	80
5.2.6 ファジィアルゴリズムのFPN表現例.....	81
5.2.7 あいまい性評価シミュレーション.....	84
5.2.8 結言.....	87
第6章 知的行動決定アルゴリズム構築用ファジィシェルに関する研究.....	88
6.1 緒言.....	88
6.2 知的制御用ファジィシェルFRASHの開発.....	88
6.2.1 基本機能.....	88
6.2.2 基本構成.....	89
6.2.3 ルールデータ構造.....	91
6.2.4 ファジィ推論機能.....	91
6.2.5 ファジィフレーム機能.....	92
6.2.6 ファジィアルゴリズム機能.....	93
6.2.7 推論シミュレーション.....	96
6.3 結言.....	99
第7章 結論.....	100
謝辞.....	104
参考文献.....	105
研究業績.....	110
著者略歴.....	112

第1章 序論

近年、メカトロ機械の高性能化、高機能化が進んでおり、これを受けて人類の生活が飛躍的に向上した。しかしながら、これまでに開発されてきた機械のほとんどは人間のルーチンワークの負荷を軽減するのが主目的であった。これに対し、従来の機械の限界を打ち破り、より高度化を図るために知能をもたせた機械（インテリジェント・マシーン）の実現を試みる研究が最近急速に発展してきている。現在、機械の知能化において最大の問題とされているのが「人間にやさしい」「人間にとって扱い易い」機能の実現方法である。人間が接し、人間がコントロールする以上、機械にこのようなヒューマンフレンドリ機能が必要なのは当然で、むしろ従来人間のほうが機械に合わせ、機械を理解しようと努めていたのが不自然であったと考えるべきであろう。

インテリジェント・マシンの最も進んだ代表的な例として、知能ロボットがあげられる。ロボットにおける知能化研究はその内容も多岐にわたっており、もはや主流となっていると言っても過言ではない。しかしながら、ロボット研究の歴史を見た場合に、現時点で最も研究が遅れており、かつ重要なテーマと考えられているのは、やはり究極目標である「人間らしい」ロボットの開発である。特に知能ロボットの研究においては、人間の複雑な判断機構、理性・常識などのメタ知識やマクロ判断にあたる部分の模擬はほとんど実現されておらず、最も大きな課題として残されているのが現状である。このようないわゆる「人間らしい」機能をロボットにもたせるためには、従来の知的情報処理手法やモデル構築手法はもとより、複雑な動作シーケンスをマクロにかつ柔軟に記述できる方法論が最も重要になってくる。

これに対する一つの方向性として、C.ISIKらの階層的制御メカニズム[1]やR.A.Brooksらの昆虫の行動モデル[2]を提唱した研究などの"行動 (behavior)"というカテゴリーを指向した新しい研究アプローチが最近盛んになっている[3]。これらの研究は、従来のロボット制御系の構築といった内面的でミクロな研究アプローチをとらず、ロボット自身が動作というかたちで発現する"行動"に着目し、それを表現するモデルを構築するという全く新しい外面的でマクロな研究アプローチであると言える。しかしながら、ロボット分野においてこのようなマクロ"行動"を模擬する研究はまだ始まったばかりであり、今後さらに活発になるものと予想される。

一方、知的情報処理のための知識表現手法においても、人工知能 (AI)、ファジィ、ニューラルネットなどさまざまな分野で要素技術が研究されており、それぞれの特長を利用した形での応用研究も進んでいる。これらの知識表現ツールの研究の発展は、制御の分野におけるオペレータ知識の記述・獲得、システム構造の表現、意思決定機能の実現などに強力な支えとなっている。中でもいわゆる「ファジィ制御」と呼ばれる Mamdani に代表されるファジィ推論方式は制御分野における実用的意義がすでに認められている[4]。このファジィ制御の出現は、従来明示的な記述が困難であったオペレータのノウハウや経験則をうまく制御ルールの中に盛り込む一つの手法が確立されたという点で意義深い。

ファジィ理論を用いた知識表現手法にはいくつかの特長があるが、他と比べて最も大きな利点は、人間のノウハウを記述しやすい知識構造を有しておりかつ知識を集約表現できるためルールメンテナンス性が極めて良いことであろう。前述のようなヒューマンフレンドリロボットを実現しようとした場合、ファジィ理論は現在最も有力な方法論の一つであると考えられる。しかしながら、人間が行なっているような意思決定レベルのマクロ判断を実現しようとした場合、従来のファジィ制御で用いられている Min-Max 重心合成法を用いた推論方法だけでは記述能力が必ずしも十分とは言えない。例えば、L.A.Zadeh がもともとファジィ理論の基本概念として提唱したファジィアルゴリズム[5]のような本来の特質であるあいまい情報処理手法としてのファジィ理論の制御への応用をこれからは指向する必要があると考えられる。

1. 序論

以上のようなことから本研究では、知能ロボットの中でも特に"行動"を伴う動作を特徴とする移動ロボットを対象として環境や行動目的に対して比較的あいまいな状況を常に認識しながら的確な判断を行っている人間の柔軟な状況判断・適応能力を模擬した行動決定アルゴリズムを構築することを試みる。しかしながら、前述のようなヒューマンフレンドリロボットを構築する場合、これらのあいまい性を何らかの形で模擬する必要がある。例えば、樫木らは自律ロボットの走行制御[6]において、経路計画に対して定性モデルを用いることによりあいまい性を取り扱っている。これに対し本研究においては、知能化研究へのファジィ情報処理技術の応用可能性を探るため、特に行動決定に関連するあいまい性に着目し、主にファジィ理論を適用することにより従来手法では実現が困難であった人間を模擬した状況判断能力をロボットに付加することを目標とする。

知能移動ロボットに必要とされる基本的な機能を図1に示す。ロボットは、外界環境に対するセンシング機能、オペレータである人間に対するコミュニケーション機能により外界からのコマンドや情報を入力する。これらの情報を基にデータやルール等の知識ベースを参照しながら、プランニング、ナビゲーション、ガイダンス、移動制御のアルゴリズムが実行され、移動機構により行動を行う。この際、自律的に動作する「人間らしい」ロボットを想定した場合、「いまどういう状況か?」「どのような経路で目標まで移動すればよいのか?」「どういう戦略でいま行動すべきか?」「どうやって相手を回避するか?」といったようなよりマクロな判断がロボット自身でなされる必要がある。このような「行動決定」の機能は自律的な知能移動ロボットには不可欠で、図の各部の基本機能を知的に支援する重要な役割を果たす。

本論文では、ロボットの判断決定機能にあたるプランニング、ナビゲーション、ガイダンス、移動制御に関連する行動決定アルゴリズムとこれをサポートする基本機能として知識ベース、移動機構の各テーマを主に対象としている。さらに、このような自律移動ロボットの基本機能に着目した場合、前述のようにさまざまなレベルのあいまい性が含まれており、これをどのように扱うかが常に問題となる。本論文ではこのような行動決定に関連するあいまい性を積極的に取り扱い、その表現方法やアルゴリズムを提案する。

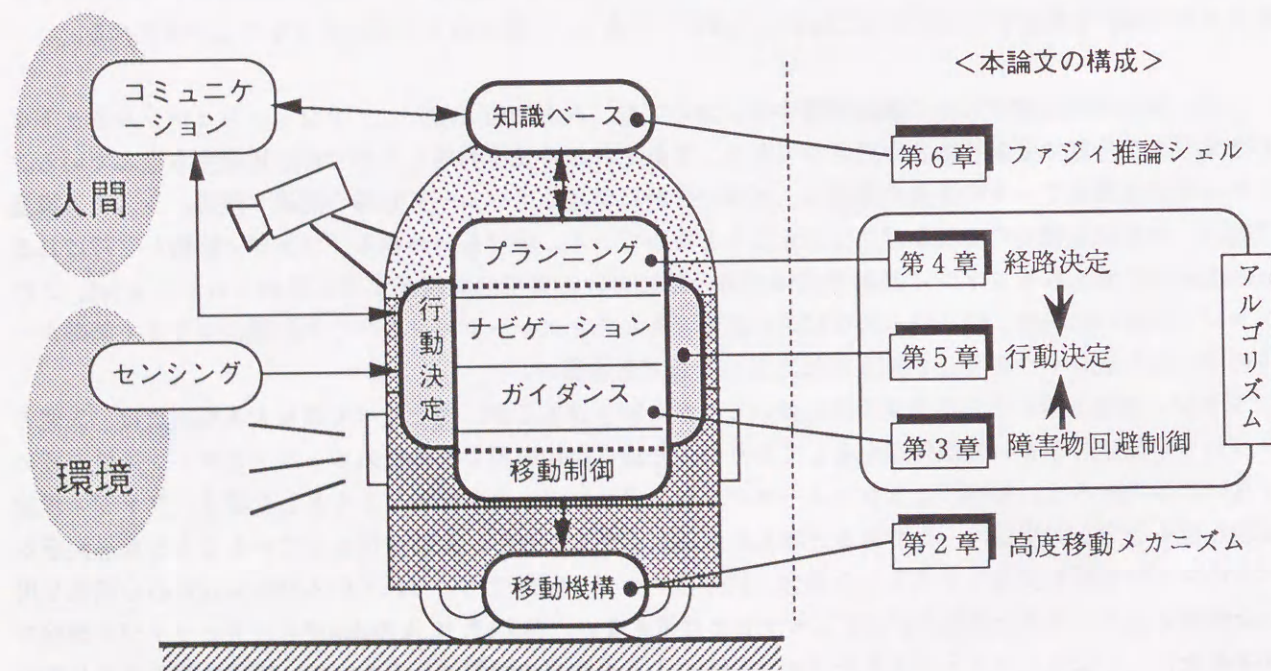


図1 知能移動ロボットに必要とされる基本機能および本論文の構成

1. 序論

筆者らはロボットに内在するさまざまなあいまい性に着目し、ファジィ理論の適用により自律移動ロボットの行動決定機能の高度化を行うこと最終目標とし、主として以下の5つのアプローチ（本論文では各章に対応）により研究を進めてきた。

- 1) 第2章：高度移動メカニズムに関する研究
- 2) 第3章：障害物回避制御アルゴリズムに関する研究
- 3) 第4章：経路決定アルゴリズムに関する研究
- 4) 第5章：行動決定アルゴリズムに関する研究
- 5) 第6章：知的行動決定アルゴリズム構築用ファジィシェルに関する研究

前述の移動ロボットの研究テーマから見ると、第2章は移動機構、第3章はガイダンス、移動制御、第4章はプランニング、第5章は行動決定、ナビゲーション、第6章は知識ベースの研究課題にそれぞれ相当している。図1には、知能移動ロボットの基本機能とともに本論文の各章における研究との関連性を示している。第3章の障害物回避制御アルゴリズムはオンラインにおける移動ロボットの移動戦略を、第4章の経路決定アルゴリズムはオフラインにおける移動計画をそれぞれ策定するための方法論について論じている。第5章の行動決定アルゴリズムはそれらを統合するかたちである行動目的に従った自律移動を実現するための知識表現手法について述べており、本論文の知能移動ロボットにおいてはいわゆる意思決定の中心となる部分である。これらのアルゴリズムを具体的に実現するためのハードウェアおよびソフトウェアの構築方法として、第2章では複雑な移動機能を実現する移動メカニズムの設計手法を、第6章では知的アルゴリズムを記述するためのファジィ知識表現を基本機能とするエキスパートツールの構築手法について論じる。以下では、各章における研究内容についてその概要を述べる。

第2章の高度移動メカニズムの研究では、移動ロボット研究の第1段階として複雑かつ多彩な移動機能の実現のためのメカニズム設計について述べる。移動ロボットには大別して、車輪式、クローラ式、脚式があるが、それぞれに一長一短があるために従来のものは適用範囲が限られていた。これに対し、カールスルーエ原子力研究所（ドイツ）の旋回可能クローラロボット[7]や岩本らの形状可変形クローラ走行車[8]、中山らのステアリング可能な半月形クローラ車[9]など、3つの移動方式の変形タイプともいべきユニークな移動機構が開発されている。しかしながらこれらの移動ロボットも上記3方式の長所を完全に満たしているわけではなく、全方向移動・階段昇降・障害物またぎ越えのすべての動作を迅速にこなす高度な多機能移動ロボットはいまだに存在しないのが現状である。そこで筆者らは車輪とクローラと脚を複合した独自の移動機構により、上記のような移動機能をすべて実現する多機能移動ロボットMRVを開発した。本論文では、この提案した移動機構の駆動メカニズムについて述べ、障害物またぎ越え、溝越え、階段昇降の3動作に着目し、本機の対障害適応性について述べる。尚、本章で述べる高度移動メカニズムは後述の知的行動決定アルゴリズムの研究における移動性能を高めるための基盤技術に位置づけられるものである。

第3章の障害物回避制御アルゴリズムの研究では、移動ロボットにとって最も基本的な機能である障害物回避を取り上げ、人間の回避性能に近い柔軟なアルゴリズムの提案を行っている。移動ロボットの障害物回避制御方式には代表的なものに、1) コンフィギュレーション空間法 (Lozano-Perezら[10])、2) 人工ポテンシャル法 (Khatibら[11])、3) ヒューリスティック手法 (武野, 柿倉ら[12]) などがある。これらの手法の中

1. 序論

でも比較的問題の定式化が容易で演算量が少なく、かつ動的障害物にも対応可能であると考えられるのはポテンシャル法である。ところが、この方法は目標点に向かうための負のポテンシャルと障害物からの反発を与える正のポテンシャルが平衡し、動作が停留と呼ばれるデッドロック状態に陥る場合が存在する。これに対し、人間は周囲の物体のポテンシャルを感じつつ反発しながら回避しているわけではなく、移動時に物体が近づいたとき、潜在的に危険を察知する能力をもっており、大まかな状況判断を瞬時にやっつける。このような見地から本研究では、動的な障害物が存在する環境中で走行する移動ロボットに対し、ファジィ推論を用いることにより、人間と似たフレキシブルな状況判断能力をもたせ、知的な回避性能を実現することを試みる。本手法ではまず主観的、直感的な状況把握などの粗い認識を危険度という尺度でファジィ推論により求め、その結果にもとづきプロダクションルールによる回避戦略を表現したディシジョンテーブルで行動決定を行なうという階層的決定方式をとっている。またここで用いられている回避戦略にあたるルールはあらかじめ人間が与えたものであり、シミュレーション結果に基づいて人間がチューニングを行っている。これを自動化するため、さらに本論文ではCMACと呼ばれるニューラルネット学習アルゴリズムを用いることにより、人間の回避操作から自動的にディシジョンテーブルの制御戦略をスキル獲得することを試みる。

第4章の経路決定アルゴリズムの研究では、移動ロボットにおいて最も上位のレベルとされている経路計画に関する知的アルゴリズムの構築を行っている。経路計画は前述の障害物回避等と比較してより大局的な見地によるグローバルな意思決定が必要とされるところが特徴である。経路計画に関する研究は一般に大別して、a) 自由空間の分割方法、b) 自由空間を通る経路の探索方法、c) 得られた経路に対する評価方法、の3つに分かれる。a) はロボットが動作できる空間（自由空間）の効率的な表現方法が主な課題であり、これにはコンフィギュレーション空間法（Lozano-Perezら[10]）、ハイブリッドセル空間法（登尾ら[13]）などがある。b) は自由空間に分割された後にグリッドからグリッドを辿りながら目標ノードへ到達するグラフ探索問題と考えられ、効率的な探索アルゴリズムが主な課題となる。探索の効率化手法には、縦型探索に枝刈り法、横型探索にDP法、ヒューリスティック探索にA*アルゴリズムなどが知られている。c) は複数の経路群に対して最終的な経路を決定するための評価アルゴリズムが中心課題である。これには、当然あいまい性を含んだ判断が必要な場合も存在する。これらの中で特にb)、c) は大局的な意思決定が影響するため、人間の高度な決定アルゴリズムを参考にすることが可能である。本論文ではまずb) に関する研究として、経路情報が格子（グリッド）状ネットワークで与えられているような複雑な経路探索問題を対象として、ファジィ推論を適用したヒューリスティック探索アルゴリズムの一手法を提案する。本手法は、従来の枝刈り探索に比べ探索効率が高く、比較的最適解を得やすい特徴がある。さらにc) に関する研究として、人間の持つ概念に近いあいまいさを含めた経路評価アルゴリズムについて述べる。ここでは、地図情報に不確かさが存在する、サブゴールにあいまいさを含む、経路の評価関数に人間の主観的な評価を含む、などの条件のもとでの知的経路計画アルゴリズムを提案している。

第5章の行動決定アルゴリズムの研究では、知能移動ロボット研究の最終段階としてあいまいな状況を常に認識しながら的確なマクロ判断を行っているような人間の柔軟な状況判断・適応能力を模擬した行動決定アルゴリズムを構築する。通常AIプロダクションルールによる表現方法を用いた場合、シーケンスフローの記述には向いているが、あいまい状態を取り扱う一般的な表現方法がない。一方、ファジィルールを用いた場合はあいまい状態は表現し得るが、分岐を伴うようなシーケンスルールの表現には向いていない。そこで本研究では、Zadehにより提案されたあいまいな状態を記述できかつシーケンスフローが表現可能なファジィアルゴリズム[5]の概念を用いて、人間が通常行っているのに近い行動決定アルゴリズムを記述する方法

1. 序論

を提案する。このファジィアルゴリズムを用いると、アルゴリズムシーケンスにおけるファジィ分岐が表現できるが、多段で用いるとあいまいさが爆発するという実用上の問題も同時にかかえている。室伏ら[14]はこの問題に対し、分岐の判断に固定スレシヨールドを用いてファジィアルゴリズムを表現したが、本来のファジィ分岐の特徴を失っており、クリスプな分岐と事実上変わらなくなっている。そこで本方式では上位の目的自律判断機能によるスーパーバイザリなスレシヨールド制御を行うことによりアルゴリズムチューニング機能を有する修正ファジィアルゴリズムを提案している。さらに本研究ではファジィベトリネットの考え方を導入し、ファジィアルゴリズムのフロー全体のベトリネット表現を行い、これを用いてファジィ真理値トークンの伝搬によりシステムのあいまい性の推移状況を捉えることを試みる。

第6章の知的行動決定アルゴリズム構築用ファジィシェルの研究では、前述の行動決定アルゴリズム記述のための実用的な汎用エキスパートツールの開発を行った。従来のファジィ推論をサポートしているエキスパートシステムとしては、安信らのES/KERNEL/W[15]や山崎らの多段型ファジィ推論シェル[16]などがあるが、一般にユーザインタフェース機能は比較的高いものが多いが、ユーザプログラムとのリンクが困難であったり、推論スピードが遅くリアルタイム制御に不向きであるなどの問題点があった。そこで筆者らはリアルタイム知的制御用ファジィシェルFRASHを開発した。このシェルはユーザプログラム主導を基本思想とするライブラリ形式のファジィ推論エンジンおよびそのチューニング機能、オフラインエディタ機能および推論状況表示機能、ファジィフレーム表現およびファジィアルゴリズムコンパイラなど数々の特長を有しており、ファジィ情報処理を含む幅広い知識表現に対し強力な記述能力をもつ。

第2章 高度移動メカニズムに関する研究

2.1 緒言

ここでは知的行動決定アルゴリズムの研究に先立ち、自律移動ロボットの移動性能を向上するためのハードウェア要素技術開発の位置づけとして行った高度移動メカニズムの研究について述べる。

現在、移動ロボットはさまざまな分野に適用されているが、なかでも作業が危険であるため人間の行なっている特殊かつ多様な作業を代行するための高度な移動メカニズムの開発が不可欠である特殊作業ロボットは、一般に複雑な保守点検作業を自動化し無人化することを目的としているため、従来の産業用ロボットでは考えられないようなフレキシブルな動作や不特定作業に対応できる汎用性が必要となる。

移動方式には軌道式と無軌道式があるが、保守点検用移動ロボットの場合フレキシビリティが必要なため無軌道式に限ると、1)車輪式、2)クローラ式、3)脚式、の3方式に大別できる。車輪式はステアリングを駆使することにより平地を全方向に高速移動できる反面、階段昇降や障害物越えなどは一般に不可能である。クローラ式は不整地走行や階段昇降などの路面に影響を受けない移動に向いているが、障害物を乗り越える際に柔軟な物体を損傷する可能性がある。また脚式は障害物を非接触でまたぎ越えることが可能であり、動物のような多様な移動機能を実現できる可能性を有しているが、走行速度が遅く制御が複雑になるなどの問題がある。以上のように各移動方式にはそれぞれ一長一短があるため、これら3方式の移動ロボットはいずれも適用場所や範囲が限られていた。

さらに新しいタイプの複合型移動ロボットも数多く開発されている。例えば、車輪変形タイプとしては、4輪独立ステアリングが可能な全方向移動ロボット[1]など、クローラ変形タイプとしては、4基の巡回可能なクローラをもつ原子力点検ロボット[2]やフレキシブルな乗り越え動作が可能な形状可変形クローラ走行車[3]、ステアリングを可能とした半月形クローラ車[4]など、脚変形タイプとしては、脚の先端に駆動可能な車輪をもうけた5足ロボット[5]や膝関節および車輪をもつ4足ロボット[6]などがある。しかしながらこれらの移動ロボットも上記3方式の長所を完全に満たしているわけではなく、全方向移動・階段昇降・障害物またぎ越えのすべての動作を迅速にこなす高度な多機能移動ロボットはいまだに存在しないのが現状である。

筆者らは車輪とクローラと脚を複合した独自の移動機構により、上記のような多様な移動機能をすべて実現する多機能移動ロボット (Multifunctional Robot Vehicle: MRV) を開発した[7][8][9][10]。本機はステア

リング可能な4つの車輪とその同軸上に駆動スプロケットをもつ巡回動作可能な4基のクローラ脚をもっており従来にない多機能動作が可能である。本研究で提案する機構を用いると少ないアクチュエータで複雑な多自由度動作が実現でき、エネルギー効率のよい自立型移動ロボットが構築できる。

本論文では、MRVの基本メカニズムと多機能移動モード、適応動作シーケンスについて述べると共に、本機の最大の特徴である障害物またぎ越え時の静的安定性や障害物形状についての動作解析を行う。最後に、実験によりこの多機能移動ロボットの動作シーケンスや動作解析に対する検証を行なったので、これについても報告する。

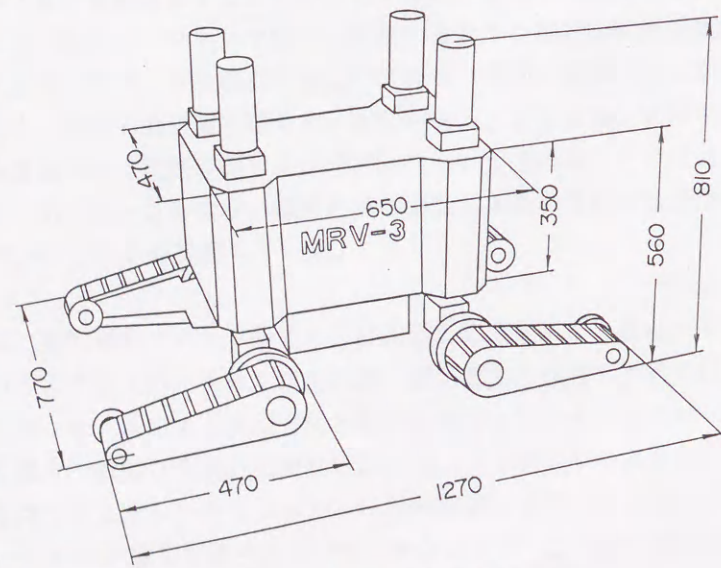


図2.2.1 MRVの外形寸法

2.2 多機能移動ロボットMRVの開発

2.2.1 車輪・クローラ脚複合形

多機能移動機構

試作したMRVの外形および各部寸法を図2.2.1に示す。本機は4つの同じ脚部モジュールをもっており、それぞれは巡回可能なクローラとその巡回軸と同軸内側に車輪を有している。これらにより、MRVは走行、脚巡回、ステアリングの3自由度を各脚部にもっている。このロボットはプラント内の各種作業を想定して開発されており、実際に発電プラントを調査して外形寸法や重量、移動機能等を決定した。

車輪とクローラと脚の機能を複合したMRVの移動機構を図2.2.2に示す。図では楕円状の履帯は長軸方向を鉛直上方に立てているが、この部分は駆動軸を中心に旋回することができ、鉛直下方に立てることもできる。従って、この部分を1関節の脚と考えると以下ではクローラ脚と呼ぶことにする。

図2.2.2において3重構造の伝達軸の中心軸Aを駆動すると、傘歯車を通して車輪および履帯駆動用スプロケットが同時に回転する。また中空軸Bを駆動すると、傘歯車を介してクローラ脚を支える側板が回転する。さらに軸Cを駆動すると、駆動軸の方向が変わり、移動方向が変化する。但し、この場合は軸Bを固定すると、クローラ脚が同時に回転するので、クローラ脚の方向を一定に保つために軸Bも同時に駆動する必要がある。この図のようにクローラ脚が鉛直上方にあるときには、車輪のみが接地するように車輪外径は履帯屈曲部の外径より大きくしている。

以上のように、この移動機構は3自由度を有しているが、3個の電磁クラッチを用いて1台のモータで駆動する機構を考案しモータ台数を減らしている。図2.2.3はこの駆動機構を示す。車輪および履帯駆動の際にはクラッチAをつなぎクラッチBおよびCを切るとモータはギヤAを介して軸Aを駆動する。クローラ脚巡回時にはクラッチBをつなぎクラッチAおよびCを切れば、モータはギヤA、Bを介して軸Bを駆動する。さらにステアリング時にはク

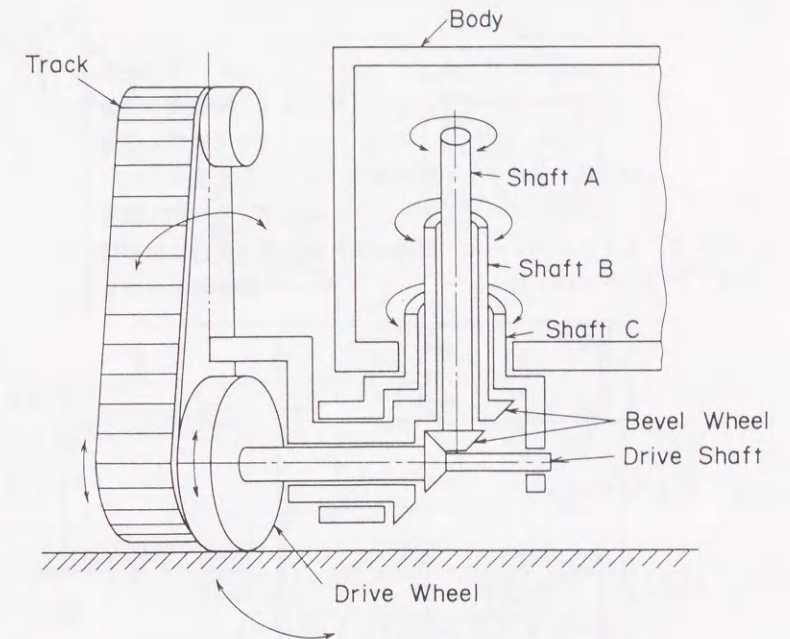


図2.2.2 多機能移動機構

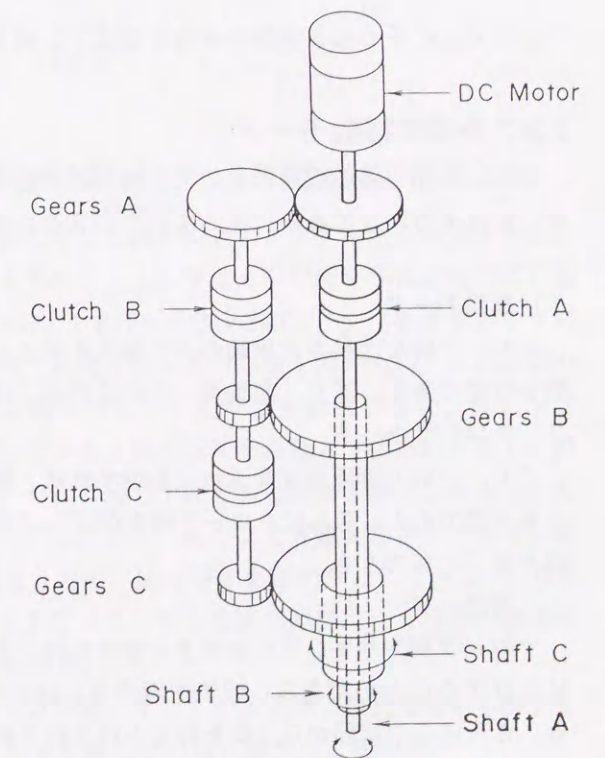


図2.2.3 駆動メカニズム

2.2 多機能移動ロボットMRVの開発

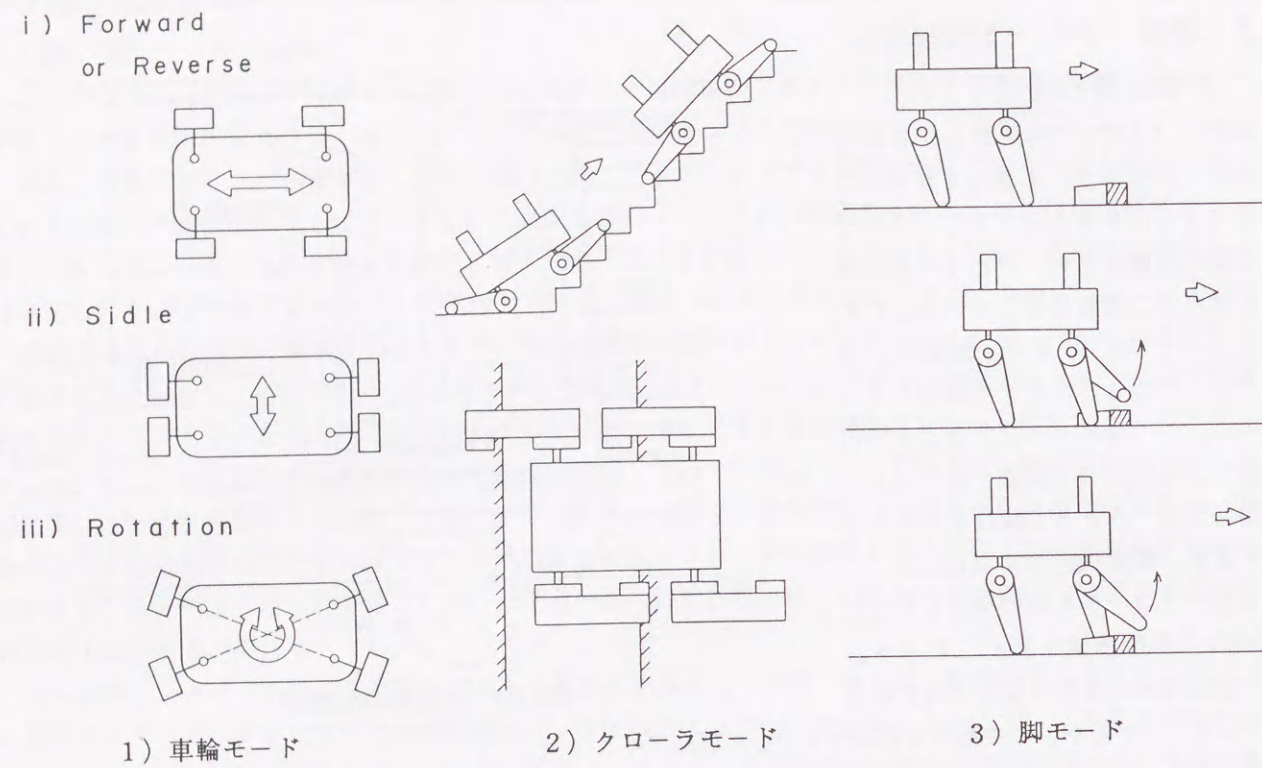


図2.2.4 多機能移動モード

ラッチBとCをつなぎクラッチAを切ると、軸Bと軸Cが同時に駆動される。

2.2.2 多機能移動モード

図2.2.2に示す移動機構において、楕円状の履帯をロボット本体に対して鉛直上方に立てるか、平行にするか、鉛直下方に立てるかによって以下の3つの移動モードに変化させることができる(図2.2.4参照)。

1) 車輪モード

クローラ脚を上昇させ車輪のみで接地させると任意の方向に自在にステアリングが切れるため、全方向移動が可能である。即ち、前後進、その場回転、横走行等の多様な移動が容易にできる。

2) クローラモード

クローラ脚の旋回角度を適当に制御すれば、階段をスムーズに昇降したり、不整地をトレース走行することも可能である。さらにクローラ脚を伸ばして全長を長くすることにより、ホイールベースより長い溝を乗り越えることもできる。

3) 脚モード

クローラ脚を水平以下に旋回させ本体を持ち上げることにより、立脚状態になり、履帯を回転させて前後に走行することができる。履帯先端内側には小車輪が設けられているので立脚時にも滑らかな走行が可能である。さらにこの状態から1脚を持ち上げ3脚で静的安定を保ちながら移動することにより、パイプなどの障害物を非接触でまたぎ越えることも可能である。

2.2 多機能移動ロボットMRVの開発

MRVにはあらかじめ上記のような各適応動作パターンがいくつもプログラムされており、これらのモード切り替えにより柔軟な対障害適応機能を実現している。次節ではこれらの動作のうち代表的な3つの動作を取り上げ、その動作シーケンスを紹介する。

2.2.3 適応動作シーケンス

MRVは自在な脚旋回動作が可能のため、さまざまな障害に対するフレキシブルな移動性能を有する。以下にその代表的な3動作、即ち、障害物またぎ越え、溝越え、階段昇降について適応的な回避動作シーケンスをシミュレーションを中心に紹介する。尚、試作したMRVの諸元を表2.2.1に、本章で用いる $\theta(1)$ から $\theta(4)$ の脚旋回角度の定義を図2.2.5に示す。[鉛直上方をホームポジション($\theta=0$)とする]

(a) 障害物またぎ越え

図2.2.6a)は静的安定性を保ちながら障害物を擬似的歩行動作によりまたぎ越える様子を示す。転倒しないためには脚接地点を結ぶ多角形内にロボット本体の重心投影点が常に存在するように動作パターンを決定する必要がある。この図に示す

ように4本のクローラ脚を平行四辺形状に傾斜させ、対象物に対して斜めにアタックすると、本体を水平に保ったまま安定した非接触またぎ越えが可能である。まずステップ[1]では4本の脚を本体が後方にくるように鉛直下向きに対し 20° 前方に傾ける[$\theta(1)=\theta(4)=160^\circ$, $\theta(2)=\theta(3)=200^\circ$]。(後述するように 20° という値は最適値ではないが、シミュレーションにおいては簡単のためこの値を用いた。)この状態のときには本体の重心投影点がやや後方にあるため、前の2脚のどちらを上げてても転倒しない。ステップ[2]では、障害物に近い方の前脚を持ち上げた後、接地している3脚の履帯を駆動して前方に移動し、持ち上げている脚を障害物の向こう側へ送る。ステップ[3]では、持ち上げた方の前脚を接地させるとともに、もう一方の前脚を持ち上げて前方へ移動する。このようにして前の2脚のまたぎ越えが終了すると、次のステップ[4]および[5]では後脚で同様の動作を行なうため、脚の傾きを逆向きに行っている[$\theta(1)=\theta(4)=200^\circ$, $\theta(2)=\theta(3)=160^\circ$]。以後ステップ[6]および[7]でも前脚と同様にして後脚の障害物のまたぎ越えを行なう。終了時にはステップ[8]の状態になる。

次にまたぎ越え可能な障害物の最大幅を求める。図2.2.7a)は移動ロボットが障害物に接近している様子を上から見た図を示している。この図において、またぎ越え可能な障害物の幅が最大になるのは、

$$\overline{AB} = \overline{BC} = \overline{CD} = \sigma m \quad (2.2.1)$$

表2.2.1 MRVの寸法諸元

ホイールベース	: $l = 50$ (cm)
左右の脚間隔(トレッド)	: $w = 66$ (cm)
駆動スプロケットと 従動スプロケットの軸間隔	: $f = 34.5$ (cm)
従動スプロケット半径	: $r = 5$ (cm)
脚旋回中心から脚先端までの距離	: $m = f + r = 39.5$ (cm)
クローラ脚の幅	: $g = 8$ (cm)

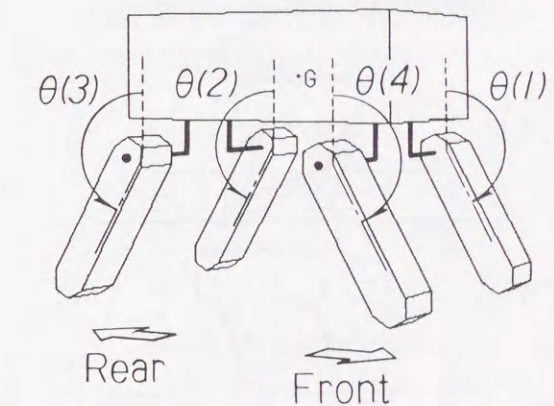


図2.2.5 クローラ脚旋回角度の定義

2.2 多機能移動ロボットMRVの開発

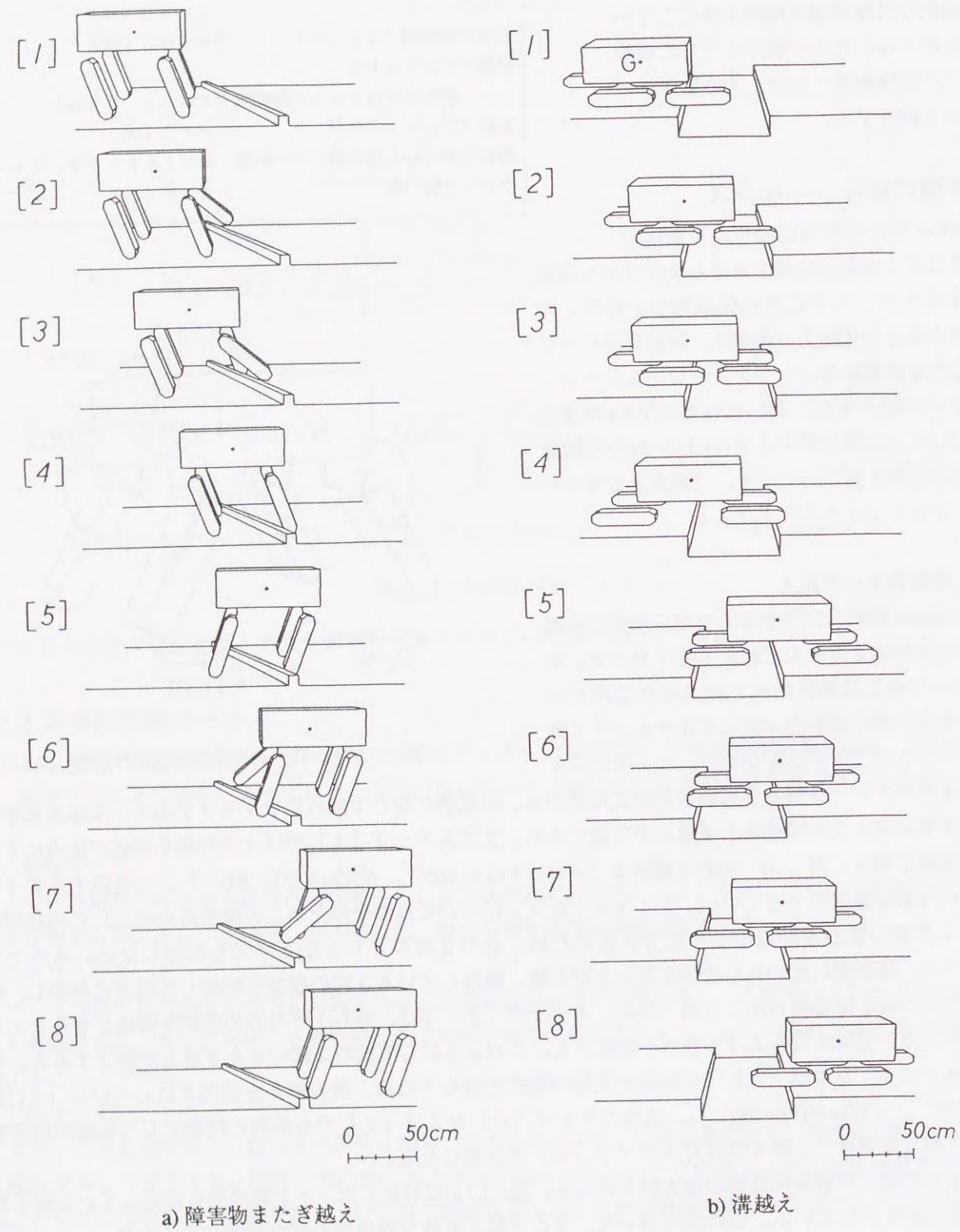


図2.2.6 適応動作シーケンス

2.2 多機能移動ロボットMRVの開発

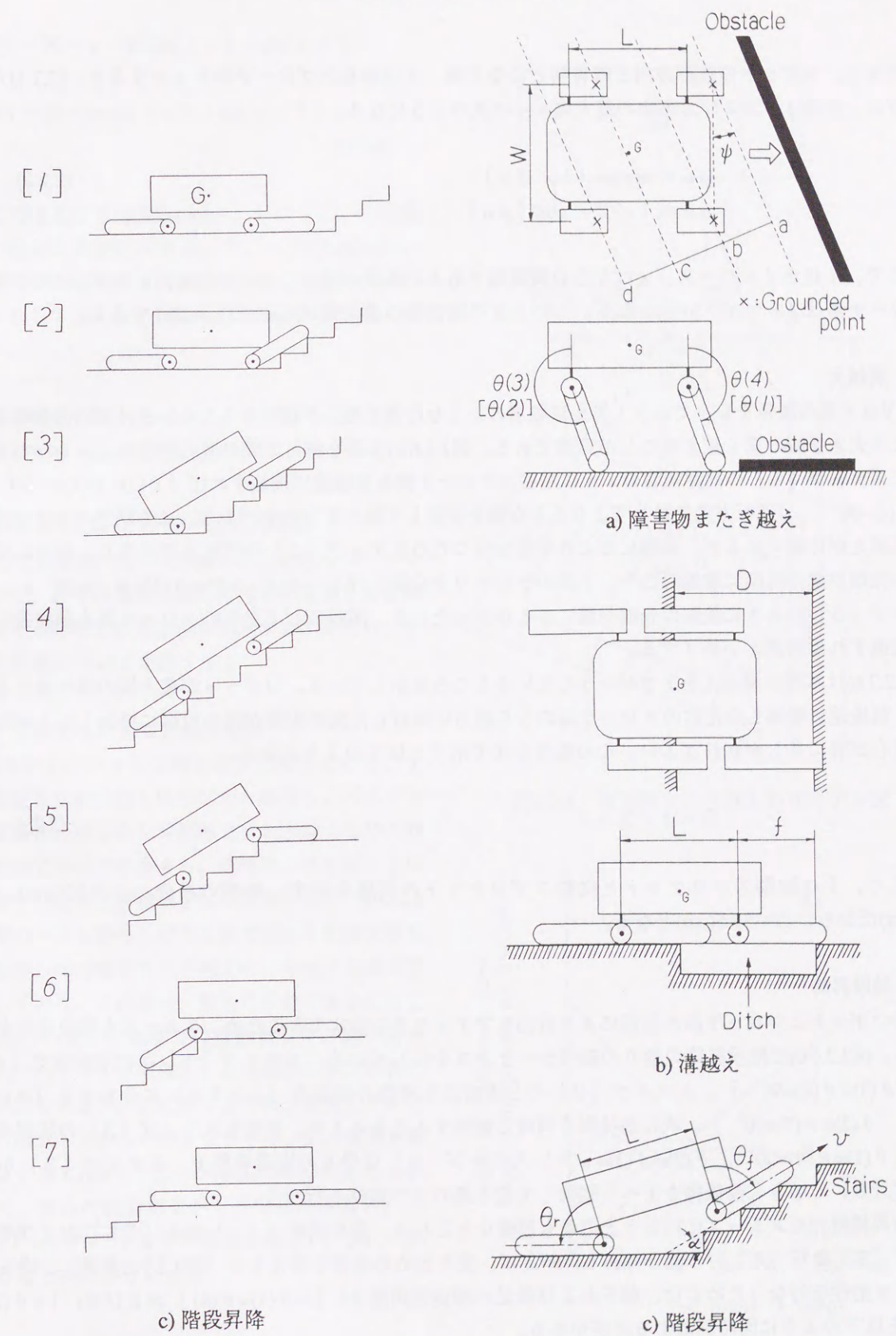


図2.2.6 適応動作シーケンス

図2.2.7 アプローチ動作のモデル図

2.2 多機能移動ロボットMRVの開発

の時である。ロボットの進行方向と障害物とのなす角、すなわちアプローチ角を ϕ とすると、(2.2.1)式の時のアプローチ角 ϕ_m および障害物の最大幅 σ_m は次のようになる。

$$\phi_m = \arctan (1 / 2 w) \quad (2.2.2)$$

$$\sigma_m = 1 / 2 \cdot \cos (\phi_m) \quad (2.2.3)$$

ここで、 l はホイールベース、 w は左右の脚間隔である。本機の場合、 $l = 50(\text{cm})$ 、 $w = 66(\text{cm})$ だから最適アプローチ角は $\phi_m = 20.7(\text{deg})$ となる。このときの障害物の最大幅は $\sigma_m = 23.4(\text{cm})$ である。

(b) 溝越え

MRVは4基の履帯をもっており、例えば戦車のような片側1基しか履帯をもたないタイプの無限軌道車に比べより大きな幅の溝を越えることが可能である。図2.2.6b)は溝を越える際の適応動作シミュレーションを示す。ステップ[1]の初期状態のように左右のクローラ脚を前後逆に接地すれば $[\theta(1)=\theta(3)=-90^\circ, \theta(2)=\theta(4)=90^\circ]$ 、進行方向に対してより大きな溝を安定して越えることができる。この状態のままでも溝越えが可能であるが、実際にはより平衡を保つためステップ[2]の位置まで前進し、前方に伸ばした脚の先端が溝の対岸に着地したら、4基のクローラを全部広げる $[\theta(1)=\theta(2)=\theta(3)=\theta(4)=90^\circ]$ 。次にステップ[5]のように後脚の先端が溝にさしかかったとき、同様にして左右のクローラ脚を前後逆に接地し、前進すれば溝越えが終了する。

図2.2.7b)は本機が溝越えをアプローチしているところを示している。ロボットが最大幅の溝を越えるためには、前後逆に接地した左右のクローラ脚のうち前方に伸ばした脚の先端が溝の対岸に接地したと同時に、本体重心が溝にさしかかればよい。この条件を式で示すと以下のようになる。

$$D = l / 2 + f \quad (2.2.4)$$

ここで、 f は駆動スプロケットと従動スプロケットの間隔を示す。本機の場合、 $l = 50(\text{cm})$ 、 $f = 34.5(\text{cm})$ だから、 $D = 59.5(\text{cm})$ となる。

(c) 階段昇降

このロボットはクローラ脚の旋回により自由なアタック角を設定できるため、スムーズな階段走行が可能である。図2.2.6c)は階段昇降の登りの動作シーケンスを示している。ステップ[1]は初期状態で $[\theta(1)=\theta(2)=\theta(3)=\theta(4)=90^\circ]$ 、ステップ[2]のとき前足を階段の傾斜角 (α とする) に合わせる $[\theta(1)=\theta(4)=\alpha, \theta(2)=\theta(3)=90^\circ]$ 。次に前後脚を同時に制御することにより、本体をステップ[3]の状態まで傾ける $[\theta(1)=\theta(4)=90^\circ, \theta(2)=\theta(3)=\alpha]$ 。ステップ[4]は登りの定常状態で、ステップ[5]からステップ[6]の場合も前後脚をすべて制御して登り終わりの動作を行なう。

階段昇降動作をスムーズに行なう上で最も困難なところは、登り始め ([2] から [3]) および登り終わり ([5] から [6]) の部分である。例えば、登り始めの場合を考えると (図2.2.7c)参照)、滑らかで連続的な動作を行なうためには、前足および後足の脚旋回角度 θ_f $[\theta(1)=\theta(4)]$ および θ_r $[\theta(2)=\theta(3)]$ を以下のように同時制御する必要がある。

2.2 多機能移動ロボットMRVの開発

$$\theta_f = 90 - \alpha + \arcsin (v t \cdot \sin \alpha / l) \quad (2.2.5)$$

$$\theta_r = 90 - \arcsin (v t \cdot \sin \alpha / l) \quad (2.2.6)$$

ここで v は登坂速度(m/min)、 t は [2] の状態からの経過時間(min)である。今、 $v = 5.5(\text{m/min})$ 、 $\alpha = 30^\circ$ と仮定すると以下ようになる。これをパターンとして与えると自動的な昇降動作が可能となる。

$$\theta_f = 60 + \arcsin 5 t \quad (2.2.7)$$

$$\theta_r = 90 - \arcsin 5 t \quad (2.2.8)$$

2.2.4 障害物またぎ越え動作解析

ここでは前節で述べたMRVの適応動作シーケンスの中で、最も特徴的な動作である障害物またぎ越えにおける対障害物適応性を明らかにするため、その動作性能について考察する。

(a) 3脚支持による立脚安定性

MRVが4脚のうち1脚を交互に持ち上げて、3脚で安定な立脚状態を保ちながら移動し、パイプなどの直線状の障害物を非接触でまたぎ越える時の静的安定性について考察する。前脚の一つを持ち上げて3脚で安定な立脚状態を保つ方法として、図2.2.8は前脚の一つを持ち上げて3脚で安定な立脚状態を保ちながら障害物をまたぎ越えている様子を模式図で示している。この場合、静的に安定であるためには、MRVの重心の地面への投影点Gが3脚の接地点 P_1, P_2, P_3 を頂点とする三角形の中に存在しなければならない。図に示すように、接地している各脚の鉛直下方からの傾きを θ とし、脚の重量を無視すると共にMRVの重心が本体の中心にあると仮定して、重心の投影点Gから三角形の各辺 $P_1 P_2, P_2 P_3, P_3 P_1$ までの最短距離 $\delta_1, \delta_2, \delta_3$ を求めると次のようになる。

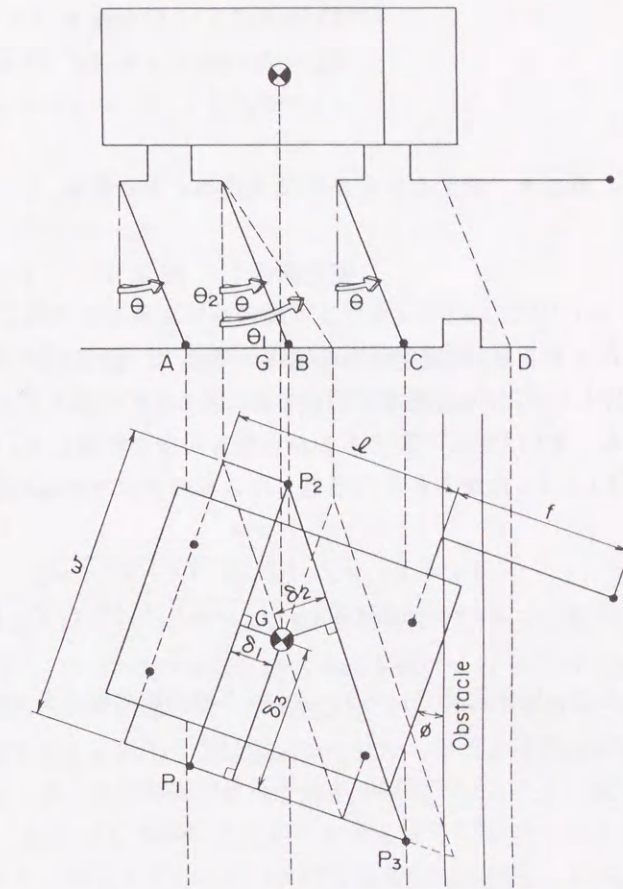


図2.2.8 障害物またぎ越え動作モデル図

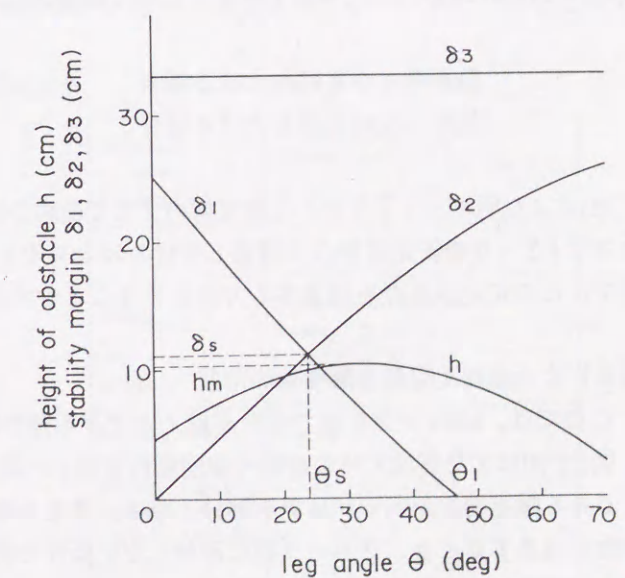


図2.2.9 脚傾斜角に対する安定余裕と障害物高さ

2.2 多機能移動ロボットMRVの開発

$$\delta_1 = l/2 - f \cdot \sin \theta \quad (2.2.9)$$

$$\delta_2 = f \cdot \sin \theta \cdot w / \sqrt{l^2 + w^2} \quad (2.2.10)$$

$$\delta_3 = w/2 \quad (2.2.11)$$

これより、静的な安定条件は次式のようになる。

$$\text{安定条件: } \delta_i \geq 0 \quad (i = 1, 2, 3) \quad (2.2.12)$$

従って、 δ_i は安定余裕を意味する値で、安定限界は $\delta_i = 0$ ($i = 1, 2, 3$)で表わされる。表2.2.1に示す試作したMRVの各部寸法の値を(2.2.9)~(2.2.11)式に代入し、 θ に対する δ_1 、 δ_2 、 δ_3 を求めると図2.2.9を得る。 δ_3 は常に一定であるから最も安定な状態は $\delta_1 = \delta_2$ ($= \delta_s$)の時である。この安定余裕 δ_s および $\delta_1 = \delta_2$ になるときの接地した3本のクローラ脚の傾斜角 θ_s は、次式で与えられる。

$$\delta_s = l \cdot w / \{ 2 (\sqrt{l^2 + w^2} + w) \} \quad (2.2.13)$$

$$\theta_s = \arcsin [l \sqrt{l^2 + w^2} / \{ 2 f (\sqrt{l^2 + w^2} + w) \}] \quad (2.2.14)$$

また、安定限界 $\delta_1 = 0$ 、 $\delta_2 = 0$ での脚の傾斜角 θ_1 、 θ_2 は次のようになる。

$$\theta_1 = \arcsin (l/2f) \text{ (deg)} \quad (2.2.15)$$

$$\theta_2 = 0 \text{ (deg)} \quad (2.2.16)$$

試作したMRVの場合、 $\theta_s = 23.8$ (deg)、 $\delta_s = 11.1$ (cm)、 $\theta_1 = 46.4$ (deg)を得る。

以上は前脚の一つを持ち上げる時の議論であるが、後脚の一つを持ち上げる時にも同様に考えることができる。これらの結果から、MRVが3脚支持で安定な立脚状態を保つための安定条件として次の条件を得る。

$$\text{前脚の一つを持ち上げる場合: } \theta_2 (=0) < \theta < \theta_1 \quad (2.2.17)$$

$$\text{後脚の一つを持ち上げる場合: } -\theta_1 < \theta < \theta_2 (=0) \quad (2.2.18)$$

(5)式より明らかなように、3脚支持のまま前脚の持ち上げから後脚の持ち上げへ移行しようとする時、途中で $\theta_2 = 0$ の安定限界点を通過しなければならないこととなる。この場合は、一旦4脚とも接地した状態で、この安定限界点を通過する方法をとることができる。

(b) またぎ越え可能な障害物の形状

ここでは、MRVが非接触でまたぎ越えができる障害物の断面形状を求める。

図2.2.10はまたぎ越え時の前脚の旋回動作を示した図である。図において、ロボットが非接触でまたぎ越えできる障害物の断面形状は斜線部分となる。すなわち、またぎ越え可能な障害物の形状は脚と脚の間を障害物が通過するとき、クローラ脚に接触しない条件で決まる。簡単のためにクローラ脚の駆動スプロケット側の大車輪は脚の小車輪と同じ半径であると仮定すると、この三角形で示される斜線部分の頂点の高さ h は次式を解くことによって得られる。

2.2 多機能移動ロボットMRVの開発

$$\alpha h^2 - (2 m c_1 \cdot \alpha + \beta) h + (m^2 c_1^2 \cdot \alpha + m c_1 \cdot \beta + \gamma) = 0 \quad (2.2.19)$$

但し、 $s_1 = \sin \theta$, $c_1 = \cos \theta$, $s_2 = \sin \phi$, $c_2 = \cos \phi$
 $\alpha = \{ 1 + (s_1/c_1)^2 \} \cdot c_2^2$
 $\beta = 2 (w s_2 - r c_2 / c_1 - g s_2 / 2) s_1 c_2 / c_1$
 $\gamma = (w s_2 - r c_2 / c_1 - g s_2 / 2)^2 - m^2 c_2^2$

アプローチ角 ϕ を上式の ϕ_m にとると、(2.2.19)式は脚の傾斜角 θ のみの関数となる。試作したMRVについて θ に対する h を求めると図2.2.9となる。図よりまたぎ越え可能な障害物の高さ h の最大値は $h_m = 10.3$ (cm)であり、この時の脚の傾斜角は $\theta = 32$ (deg)である。また $\theta = \theta_s = 23.8$ (deg)では $h = 9.9$ (cm)となる。

2.2.5 実証評価実験

(a) 試作モデル

すでに述べた多機能移動機構の実証評価のため、筆者らはMRV-1からMRV-3のプロトタイプを設計製作した。ここでは、最も改良の加えられているMRV-3を中心に試作機の特徴を述べる。本機は4台のDCサーボモータと12個の歯型電磁クラッチをもっており、4脚のそれぞれに走行、脚旋回、ステアリングの3自由度計12自由度を有している。表2.2.2にMRV-3の基本性能を示す。外形寸法は幅800mm程度の階段や通路の移動が可能となるように決めている。ホイールベースは溝越え等の動作時にクローラ脚が1回転できる長さに設定している。また、クローラ脚の長さはプラント内の一般階段をスムーズに走行できるように設計されている。本体底部にはアルミハニカムシャーシを用いており、初期モデルに比べ大幅な重量軽減を図っている。

試作モデルは有線と無線の両制御が可能で、バッテリーおよびコントローラを搭載した無線制御の場合のシステム構成を図2.2.11に示す。ロボット上のCPUはintel8086/87で、各種入出力ボードはすべてマルチバス(Multibus)で接続されている。制御コンソール上で操作者が入力した指令値はワイヤレスモデムを通して

表2.2.2 MRV-3の基本性能

Dimension	770W x 650L x 810H (Home position)
Actuator	4 DC servo motors
D.O.F.	12
Speed	Plain : 20 m/min Stairs : 5 m/min
Weight	110 kg (without battery)
Battery weight	40 kg
Battery capacity	10 AH (DC 72 V)
Climbing capacity	Max. 45°
Carrying capacity	40 kg (On the 30° inclination)

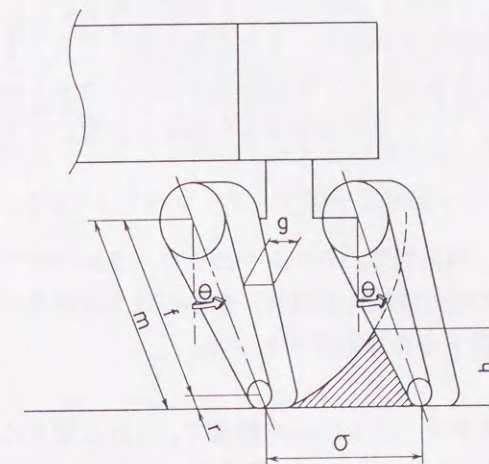


図2.2.10 またぎ越え可能な障害物の断面形状

2.2 多機能移動ロボットMRVの開発

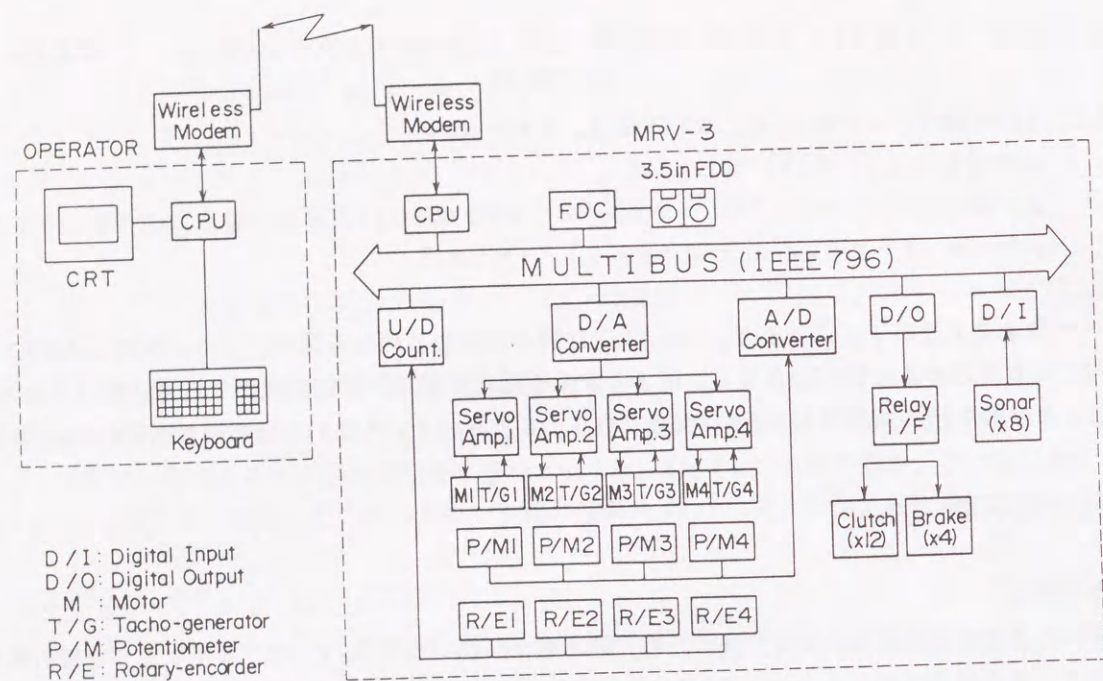


図2.2.11 MRV-3のシステム構成

FM通信によりロボット側に送信される。電磁クラッチはマイクロリレー搭載のインタフェースボードを通してロボット上のCPUよりON/OFFされる。サーボアンプはPWM方式で、サイズをコンパクトにするともにパワーの高効率化を図っている。バッテリーはパワー系の72Vと制御系の24Vの2系統を持っている。

(b) 実験結果

すでに述べたMRVの適応動作シーケンスおよび動作性能を試作機を用いた実験により検証した。実験の状況を図2.2.12に示す。

1) 障害物またぎ越え

前章で得られた結果をもとに、乗り越え可能な障害物として直径4cmのパイプを選び、地上から3cm浮かせてまたぎ越え動作の実験を行なった。尚、立脚に要した時間は約10秒で、またぎ越え動作の完了までに要した時間は約2分であった。計算どおり障害物との間隔には余裕があり、本体を水平に保ったままのまたぎ越え動作がすでに述べたシーケンスで実現できることがわかった。またロボットの動きは非常にスムーズで安定した動作であった。

2) 溝越え

ホイールベースより広い59cm幅の溝越えを行なったところ、適応動作パターンへのクローラ脚の折りたたみの時間を含めても約20秒で通過できた。但し、通過可能最大幅の溝越えの場合、重心の地上投影点が溝にさしかかるあたりの状態は若干不安定となるため、低速で移動するのは困難であった。

3) 階段昇降

実験に使用したのは斜度30°、7段、ステップ幅25cm、ステップ高さ15cmの階段で、1回の登りに要した時間は約1分であった。尚、同じ階段で積載重量の測定をおこなったところ、最大約40kgまでの積載登坂が可能であった。登坂時の最大スピードは約5m/minである。

2.3 結 言

2.3 結 言

ステアリング可能な車輪による全方向移動と、クローラ脚による階段昇降、障害物のまたぎ越え、および幅の広い溝の乗り越えなど、多様な移動機能を持つ新しいタイプの車輪・クローラ・脚複合型多機能移動ロボットを提案し、その試作モデルMRVを開発した。このロボットは発電所をはじめ各種プラントや工場での保守点検作業の自動化に必要な移動ロボットとして有効であると考えられる。

本論文では、MRVの基本メカニズムと多機能移動モード、適応動作シーケンスについて述べると共に、3脚で安定な立脚状態を保ちながらパイプなどの障害物を非接触でまたぎ越える時の静的安定性および障害物の大きさについて考察を加えた。さらにこの多機能移動ロボットを実際に試作し実験により、機構としての動作チェック、動作シーケンスの実証、動作解析に対する検証を行ない、その有効性を確認することができた。

本論文で提案した機構を用いると、少ないアクチュエータで複雑な多自由度動作が実現でき、エネルギー効率のよい自立型移動ロボットが構築できる。今後、知能移動ロボットがさらに複雑な特殊作業分野に応用されることは十分予想され、ここで提案したような移動機構がさまざまな用途に利用できるようになると考えられる。

本章で開発した高度移動メカニズムは後述の知的行動決定アルゴリズムの研究における移動性能を高めるための基盤技術に位置づけられるものである。移動メカニズムに関する研究についての記述は本論文ではこの章のみで、次章以降では本論文の主題である知的アルゴリズム関連の研究を中心に述べる。尚、本研究は新しい高度移動メカニズム開発への試みを主眼としてなされたため、メカニズムとしてはかなり複雑な機構で構成されている。そのため2次元平面の自在な移動機能のみを必要とする後述の研究（主に第5章）においては、MRVとは異なったより単純な構造をもつステアリング可能な車輪式の自立移動ロボットを用いている。

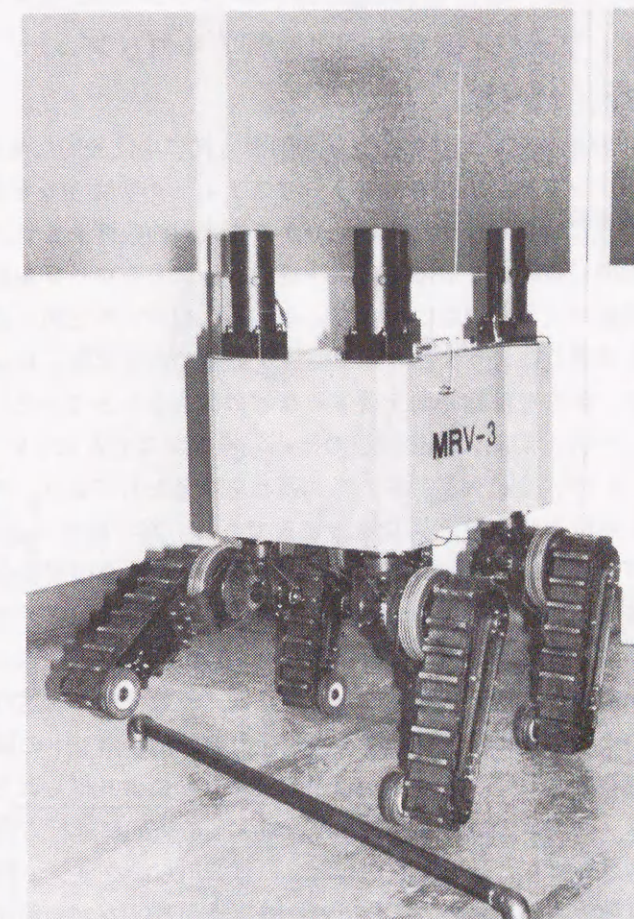


図2.2.12 移動実験の様子

第3章 障害物回避制御アルゴリズムに関する研究

3.1 ファジィ障害物回避制御アルゴリズム

3.1.1 緒言

移動ロボットの障害物回避制御方式には代表的なものに、1) コンフィギュレーション空間法[1]、2) 人工ポテンシャル法[2]、3) ヒューリスティック手法[3]などがある。1)はグローバルな経路計画に向いているが、処理に時間がかかり、いくつかの拘束制約条件をもつ。2)はシンプルで高速なアルゴリズムでリアルタイム制御には向いているが、近傍情報のみによるローカルな回避しか期待できず、停留と呼ばれるデッドロック現象が大きな問題となっている。3)はAIルールを用いた柔軟な知的制御によりグローバルとローカルの両方を考慮したような総合的な回避が可能である反面、ヒューリスティックルールを発見するのが困難であったり、多大な推論時間を要するなどの欠点をもっている。

一方、船舶の衝突回避のための操舵システムについてもいくつか報告されている(安西ら[4]、長谷川ら[5])が、主に大型船舶の動的遅れを対象としており、それほどシステムの高速度性が要求されない点がロボットの障害物回避と異なる点である。特に船舶の場合、知的制御を衝突回避に応用した例がまだ少ないが、これは操舵制御の大半を人間に頼らざるを得ず完全自動化が困難であることと一般に航法ルールが海洋法で定まっている場合が多いため回避に選択の余地があまりないことなどが考えられる。

移動障害物が存在する環境において、人間は潜在的に危険を察知する能力を有しており、適切でかつ大まかな状況判断を瞬時にやっつける。最近では、前述の3)の手法にファジィ推論を適用することにより、このような人間の状況判断・決定能力を移動ロボットの障害物回避機能にも付加する試みがいくつかなされている。例えば、樫木ら[6]は超音波センサにより得られた前方の障害物回避方向と目標通過方向の各メンバーシップ関数のMIN合成をとり、最大値の方向に進行方向をとる移動ロボットの走行制御を提案している。これ以外にも、最近発表されたものに石川らのファジィ制御による走行誘導方式[7]や野村らのファジィ推論の自動チューニング手法を用いた障害物回避[8]などの例があるが、いずれの方式もルール前件部の変数に障害物の位置や方位などのパラメータを用いて直接回避方向を決定しているため、推論ルール数が極めて多くなるとともに人間の経験則が入りにくい構成となっている。

そこで本研究ではこの人間の行動決定を模擬する目的で移動障害物が存在する環境中においても人間に近い感覚で回避アルゴリズムが記述でき、高速で効率的な判断が可能な障害物回避手法を提案する[9][10][11]。これはロボットのセンシング情報から一旦ファジィ推論により静的および動的危険度を認識し、これらにより回避戦略を記述したディシジョンテーブルを用いて回避ベクトルを決定するものである。本方式は、

- (1) ファジィ推論を用いることによりあいまいな状況に基づく判断ルールを明確に記述できる、
- (2) 静的および動的「危険度」という2つの中間変数のみを基準値として状況判別を可能とした、
- (3) 行動決定のプロセスを「あいまい状況判断+手続き的戦略決定」の階層構造で表現した、

などの特徴をもつ。最後に、さまざまな条件で行った障害物回避シミュレーションについても報告する。

3.1.2 問題設定

図3.1.1は移動障害物を回避しながら目標点に向かっている移動ロボットの状態モデルを示している。 r, o, t の添字はそれぞれロボット、障害物、目標点を示し、 P は絶対位置ベクトル、 V は絶対速度ベクトル、 S, U はロボット座標系から見た障害物の相対位置および相対速度ベクトル、 O, D, M はそれぞれロボットの障害物回避ベクトル、目標到達ベクトル、操舵ベクトルを示している。初期状態においては P_r, P_o, P_t および V_r, V_o, V_t の初期値はそれぞれ与えられるものとする。

本研究では前提条件として以下のような仮定を設定する。

- 1) 移動ロボットは1度に複数個の障害物に遭遇しない。
- 2) 移動ロボットはステアリングのみを制御する。
- 3) 移動ロボットは前方180°の視野で障害物までの大まかな距離及び方位を検知できる。
- 4) 移動ロボットは与えられた指令値どおりに動くものとする。
- 5) 移動ロボット及び障害物は質点とし、ダイナミクスを考慮しない。
- 6) 目標点は固定とする。($V_t = 0$)
- 7) 障害物は連続的な運動をする。

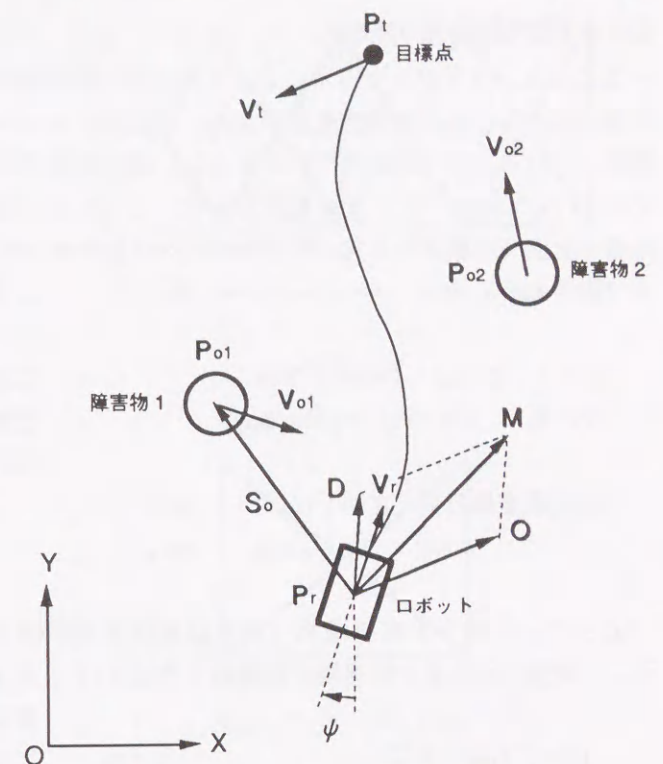


図3.1.1 移動ロボットの障害物回避モデル

3.1.3 ファジィ障害物回避アルゴリズム

本方式では障害物回避の処理プロセスを直感的な状況判断を行なうファジィ推論による粗い危険度認識とその結果に基づいてプロダクションルールにより回避方向を定める行動決定の2つのフェーズで構成されると考える。

本方式の障害物回避アルゴリズムを図3.1.2に示す。状態入力としてロボット、障害物、目標点の現在位置が与えられると、まずこれらを相対位置 (S_o, S_t) および相対速度 (U_o, U_t) に換算して、ロボット座標系から見た極座標表示に変換する。次に現在の周囲状況をファジィ推論で大まかに状況認識する。即ち位置の極座標表示を用いて静的危険度 α 、速度の極座標表示を用いて動的危険度 β を決定する。ここで α は人間が進む方向の目前にある物ほど危険を感じることを示すパラメータで β は人間に向かって速いスピードで近づいてくる物体に危険を感じることを示すパラメータと考えることができる。

さらにこの α, β より回避方向をディシジョンテーブル (プロダクションルール) を用いて決定する。人間の場合の認識および判断はこの場合のファジィ推論およびディシジョンテーブルの部分に相当する。ここで得られた回避ベクトル O と目標点に対する到達ベクトル D が求めれば、 O と D を合成することにより操舵ベクトル M が決定できる。これを用いて指令操舵角が得られ移動ロボットに出力される。

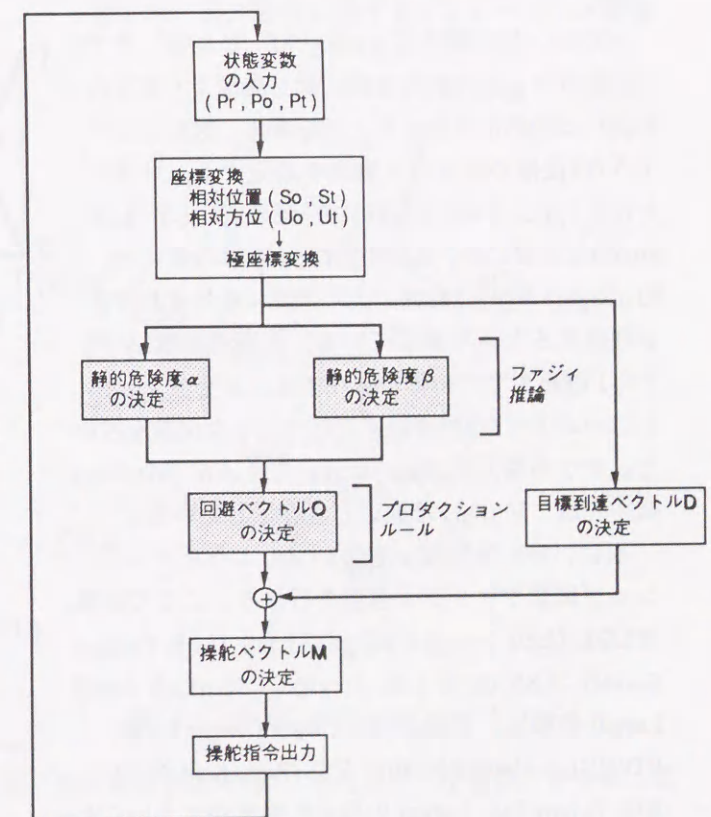


図3.1.2 障害物回避アルゴリズム

3.1 ファジィ障害物回避制御アルゴリズム

3.1.4 静的危険度の決定

ここでは、ファジィルールによりロボットと障害物の静的な位置関係から生じる危険性を表現する。移動ロボットの絶対座標系における方位（進行方向）を ψ とし、絶対速度ベクトルを $V_r=[V_{rx}, V_{ry}]^T$ ($[]^T$ は転置を示す)とすると、 ψ およびロボット直交座標系から見た障害物の相対位置ベクトル S_o は次式で表される。

$$\psi = \pi/2 - \tan^{-1}(V_{ry} / V_{rx}) \quad (3.1.1)$$

$$S_o = R \cdot (P_o - P_r) = [S_{ox}, S_{oy}]^T \quad (3.1.2)$$

$$\text{座標系変換行列 } R = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix}$$

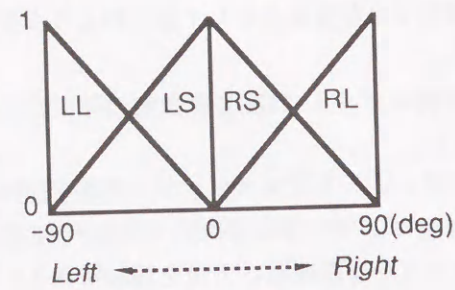
よって、ロボット極座標系（角度は進行方向が 0° 、右が正、左が負）から見た障害物の距離 L_s と方位 θ_s は、

$$L_s = \sqrt{S_{ox}^2 + S_{oy}^2} \quad (3.1.3)$$

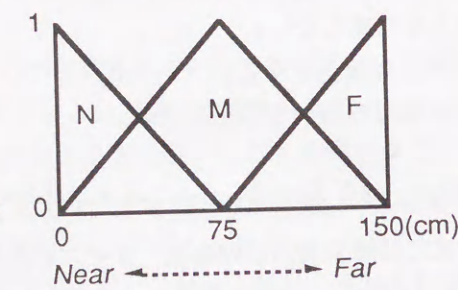
$$\theta_s = \pi/2 - \tan^{-1}(S_{oy} / S_{ox}) \quad (3.1.4)$$

となる。極座標方位 θ_s は -90° から 90° まで（移動ロボットの進行方向に対し真横より前方の範囲）に物体が存在するときを考え、図3.1.3 a)のような4段階のファジィ表現を設定する。ロボットに対しLL (Left Large)は左側に大、LS (Left Small)は左側に小、RS (Right Small)は右側に小、RL (Right Large)は右側に大の角度にそれぞれ物体が存在するときを表している。極座標距離 L_s は0から150cmまでに物体が存在するときを考え、図3.1.3 b)のような3段階のメンバーシップ関数を設けた。すなわち、N (Near)は近いとき、M (Medium)は中間域、F (Far)は遠いときを示している。

次に、静的危険度 α を図3.1.4のようなメンバーシップ関数でファジィ表現を行なう。ここで左側はLDL (Left Danger Large)、LDS (Left Danger Small)、LSS (Left Safe Small)、LSL (Left Small Large)の順に、右側はRDL (Right Danger Large)、RDS (Right Danger Small)、RSS (Right Safe Small)、RSL (Right Safe Large)の順に危険度が大きいことを表す。メンバーシップ関数は横軸の負側がロ



a) 方位



b) 距離

図3.1.3 相対位置に関するメンバーシップ関数

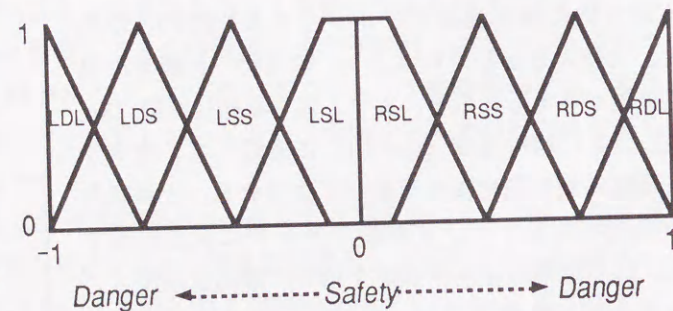


図3.1.4 危険度のメンバーシップ関数

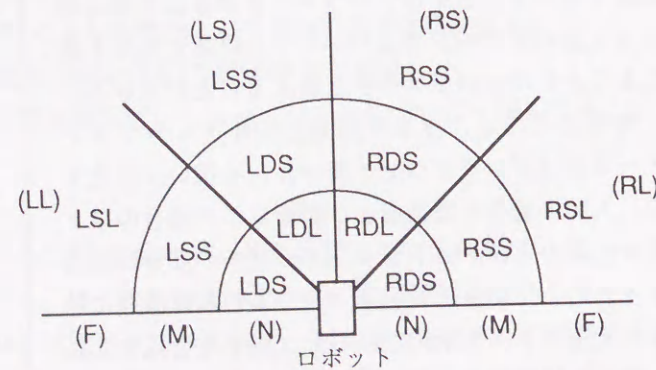


図3.1.5 静的危険度決定ファジィルール

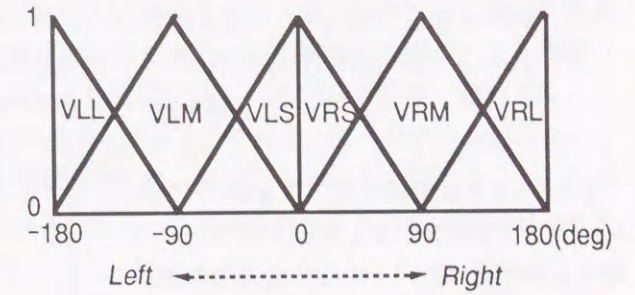
3.1 ファジィ障害物回避制御アルゴリズム

ボット左側の危険度、正側がロボット右側の危険度を示し、0付近が安全、-1と1付近が危険であることを表わす。

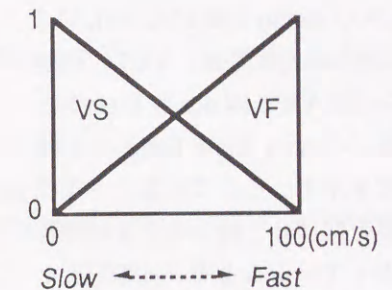
以上のメンバーシップ関数を用いて図3.1.5の図に従い、方位および距離のファジィ状態に対し静的危険度のファジィ認識を行なう。前件部では図3.1.3 a), b)より方位および距離のファジィ表現を行ない、これに対し後件部では図3.1.4より静的危険度のファジィ表現の1つを対応させる。例えば、「If {方位がやや右 (RS) で距離が近く (N) に障害物が存在する} then {右が大変危険 (RDL)}」のように判断する。これをERICを用いて実際にルールで記述すると次のようになる。

If {obst.dire is RS; obst.dist is N;}
then {fset (obst.alfa, RDL);}

同様に、図3.1.5に示したような計12の状態をファジィルールで記述し、それらをもとにファジィ推論 (MIN-MAX重心合成)を行なうと方位および距離による適合度で重みづけられた静的危険度 α が決定される。



a) 速度方位



b) 速度の大きさ

図3.1.6 相対速度に関するメンバーシップ関数

3.1.5 動的危険度の決定

ここではロボットと障害物との相対速度から生ずる危険性をファジィルールで表現する。前節の座標系変換行列 R を用いると、ロボット直交座標系から見た相対速度ベクトル U_o は次のようになる。

$$U_o = R \cdot (V_o - V_r) = [U_{ox}, U_{oy}]^T \quad (3.1.5)$$

さらに、これを障害物の方向をy軸方向とするロボット直交座標系から見た相対速度ベクトル U_o' に変換する。

$$U_o' = R_s \cdot U_o = [U_{ox}', U_{oy}']^T \quad (3.1.6)$$

$$\text{座標系変換行列 } R_s = \begin{bmatrix} \cos \theta_s & -\sin \theta_s \\ \sin \theta_s & \cos \theta_s \end{bmatrix}$$

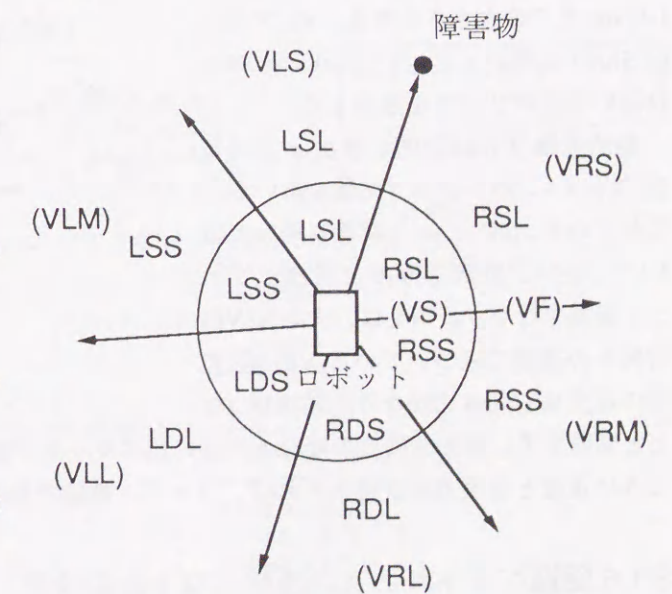


図3.1.7 動的危険度決定ファジィルール

従って、障害物の方向を向いたロボット極座標系（角度は障害物方向 0° 、右が正、左が負）から見た障害物の速度 W_u および速度方位 ϕ_u は、次式のようなになる。

3.1 ファジィ障害物回避制御アルゴリズム

$$Wu = \sqrt{Uox'^2 + Uoy'^2} \quad (3.1.7)$$

$$\phi u = \pi / 2 - \tan^{-1} (Uoy' / Uox') \quad (3.1.8)$$

図3.1.6 a)は極座標速度方位 ϕu のメンバーシップ関数を示す。移動ロボットに対する障害物の方向を中心に全方向の速度を考慮するため、速度方位は -180° から 180° までを考え、6段階のファジィ表現を設定した。図でVLL (Velocity Left Large), VLM (Velocity Left Medium), VLS (Velocity Left Small)は左側、VRS (Velocity Right Small), VRM (Velocity Right Medium), VRL (Velocity Right Large)は右側の速度方位を示す。ここで注意すべきことは、速度方位は 0° 付近のときは障害物が遠ざかっていることを、 -180° または 180° 付近は障害物が近づいていることを示すということである。

次に極座標速度 Wu のメンバーシップ関数を図3.1.6 b)に示す。速度は0から100cm/sまでの大きさを考え、VS (Velocity Slow) は遅いとき、VF (Velocity Fast) は速いときの2段階を設定した。

動的危険度 β は静的危険度 α と同じ図3.1.4のメンバーシップ関数を用いる。これらのメンバーシップ関数を用いて図3.1.7に従い、静的危険度と同様にファジィ認識を行なう。例えば、大きな(VF)右向きで近づいている(VRL)障害物の速度は右向きにかなり危険(RDL)などと判断する。静的危険度の場合と同様に12のルールで動的危険度のファジィルールが記述できる。以上のように速度と速度方位が得られれば、ファジィ推論で動的危険度 β が求まる。

3.1.6 回避ベクトルおよび操舵ベクトルの決定

すでに得られた静的危険度 α 、動的危険度 β により障害物に対する回避ベクトルをディシジョンテーブルを用いて決定する。横軸を α 、縦軸を β として、回避方向を定めたテーブルを図3.1.8に示す。このディシジョンテーブルの中の数字は図3.1.9に示したような回避方向を表しており、1+, 2+, ..., 5+は進行方向に対し右にそれぞれ $15^\circ, 30^\circ, \dots, 75^\circ$ の方向、1-, 2-, ..., 5-は同様に左方向のステアリングで、0はステアリングを切らずにそのまま走行する。

		α Danger ← Safety → Danger												
		-1.0	-0.8	-0.6	-0.4	-0.2	0	0.2	0.4	0.6	0.8	1.0		
Danger	-1.0	3+	2+	1+	0	0	0	2-	3-	4+	5+			
	-0.8	2+	1+	0	0	0	0	1-	1-	3+	4+			
	-0.6	1+	0	0	0	0	0	0	0	1-	2-			
	-0.4	0	0	0	0	0	0	0	0	0	1-			
	-0.2	0	0	0	0	0	0	0	0	0	0			
	0	0	0	0	0	0	0	0	0	0	0			
Safety	0.2	1+	0	0	0	0	0	0	0	0	0			
	0.4	2+	1+	0	0	0	0	0	0	0	1-			
	0.6	4-	3-	1+	1+	0	0	0	0	1-	2-			
	0.8	5-	4-	3+	2+	0	0	0	1-	2-	3-			
	1.0													

図3.1.8 回避ベクトル決定のためのディシジョンテーブル

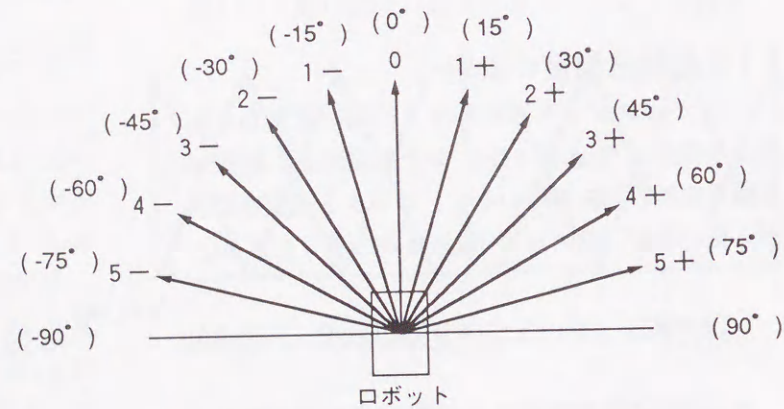


図3.1.9 回避方向ベクトル

3.1 ファジィ障害物回避制御アルゴリズム

図3.1.8で、例えば左下隅の0以外の部分は静的危険度が負(左に障害物が存在するとき)で、動的危険度が正(右方向の相対速度で接近しているとき)の回避方向をしめしているが、障害物の位置があまり危険でない ($\alpha > -0.6$) か、接近速度が危険でない ($\beta < 0.6$) ときは右に回避し、障害物の位置、相対速度ともに危険 ($\alpha \leq -0.6, \beta \geq 0.6$) なときは左に回避することを指示している。このようにこのディシジョンテーブルには人間の行動パターンを模擬する上で非常に特徴的な回避戦略を盛り込んでいる。以上のようにして決定された回避方向はロボット直交座標系における単位ベクトルとして与えられ、これが障害物回避ベクトル O となる。

また目標点に対してはロボットが目標方向を向くように次のような単位ベクトルを与える。

$$D = (Pt - Pr) / |Pt - Pr| \quad (3.1.9)$$

目標到達ベクトル D と障害物回避ベクトル O により、次式で操舵(ステアリング)ベクトルを決定する。

$$M = (D + O) / |D + O| \quad (3.1.10)$$

この(3.1.10)式で得られた操舵ベクトル M の方向にロボットのステアリングが設定され、次の移動が行なわれる。

3.1.7 障害物回避シミュレーション

シミュレーションにおけるファジィルールおよびプロダクションルールの作成にあたっては、制御用知識ベース構築ツールERIC (Extended Rule-based system for Intelligent Control) [10]を使用している。

(a) 障害物回避ルール

図3.1.10に示したのがERICで作成した障害物回避のルール構成である。ERICはルールを階層構造に構築でき、ワーキングメモリをフレーム形式で記述できる特徴を持っており、知識を整理した形でもつことができる。ここに示したものはプロダクションルールをいくつか内包しているルール群で、()内は各ルール群に含まれるルールの数を示す。ルール群の数は29、全ルール数は112である。また、ここには示さなかったが、ワーキングメモリはobst (障害物)、targ (目標物)、robo (ロボット)、control (制

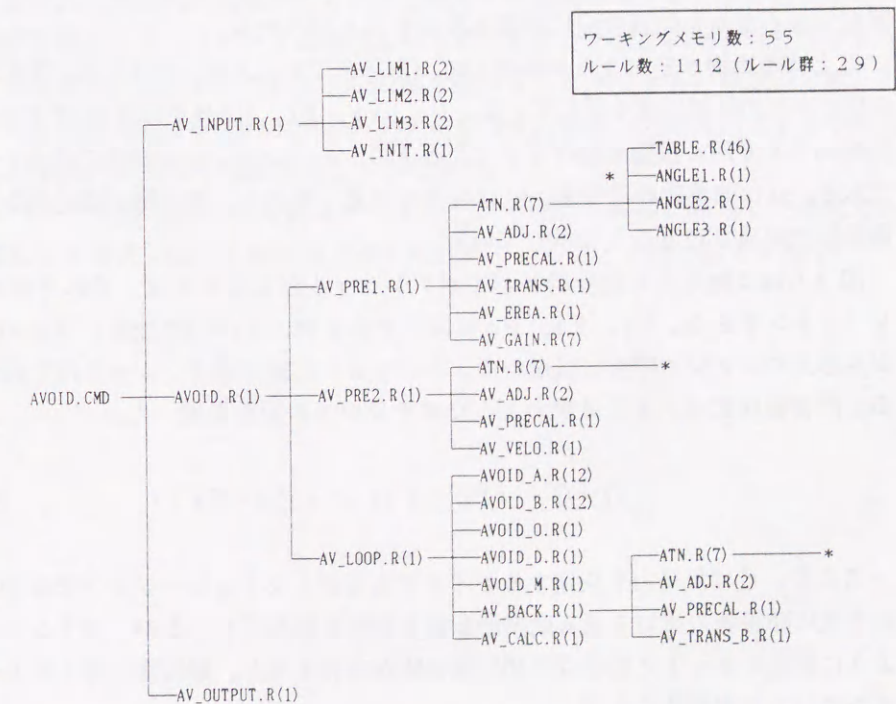


図3.1.10 障害物回避のルール構成

3.1 ファジィ障害物回避制御アルゴリズム

御変数) の計4つのタイプがあり、合計55のワーキングメモリ要素をもっている。

図においてAVOID.CMDのコマンドファイルはAV_INPUT.R, AVOID.R, AV_OUTPUT.Rの3つのルール群をもっている。このうちAV_INPUT.R, AV_OUTPUT.Rはそれぞれ入力用、出力用のルール群である。メインはAVOID.Rで、この中でAV_PRE1.R, AV_PRE2.Rは移動ロボットに対する障害物の位置および速度の座標変換用ルール、AV_LOOP.Rは回避アルゴリズムを記述したルールを示す。さらに、AVOID_A.R, AVOID_B.Rが危険度 α および β を決定するファジィルール、AVOID_O.R, AVOID_D.R, AVOID_M.Rはそれぞれ障害物回避ベクトル、目標到達ベクトル、操舵ベクトルを求めるルールである。

(b) シミュレーション結果

ERICにリンクさせたFORTRANのグラフィックルーチンで推論サイクルごとにワークステーション上にアニメーション表示させたシミュレーション結果を図3.1.11~図3.1.16に示す。図中大きな○は障害物、小さな○は目標点、△は移動ロボット(それぞれ質点と考へ、その重心座標を位置とする)で、矢印は移動方向を示している。また障害物と移動ロボットで黒色部分は同一時刻の位置を示している。尚、目標点は固定で、シミュレーション結果の長さの単位はすべてcmである。

図3.1.11はロボットの左前方から障害物が接近してくる場合の回避シミュレーションで、ロボットの初期位置 $P_r = [1,1]^T$ 、初期速度 $V_r = [30,30]^T$ 、目標点の位置 $P_t = [200,200]^T$ 、障害物の初期位置 $P_o = [100,200]^T$ 、サンプリングタイム $S_T = 0.5\text{sec}$ のときの回避性能を示しており、a)が障害物接近が遅いとき($V_o = [0,-30]^T$)、b)は速いとき($V_o = [0,-50]^T$)である。これより障害物が左前方にあって、余裕のあるときは障害物の手前を、接近速度が速く危険なときは障害物の後ろを回避していることがわかる。

図3.1.12は P_r , V_r , P_t , P_o は同じで、 $V_o = [0,-30]^T$ のとき、サンプリングタイム(推論周期)をa) $S_T = 0.1\text{sec}$ (短い)、b) $S_T = 1.0\text{sec}$ (長い)として、シミュレーションした結果である。推論周期が小さくてもほぼ同じ回避性能を示すが、周期が大きくなるとやや異った回避動作をとる。このため、推論周期が大きくなると回避性能に影響を及ぼすことがわかる。

以上は障害物が定ベクトル移動している場合であったが、図3.1.13は障害物の速度ベクトルが変化する場合についての回避性能を示している。 P_r , V_r , P_t , P_o はやはり図3.1.11と同じで、障害物を周期0.27Hz、振幅40でX方向に振動的動作をさせたときのシミュレーション結果である。a)はY方向速度-30で、b)は-40である。a)は障害物の手前を、b)は後方を回避しており、障害物が変動的な動きをしたときにも、柔軟な回避動作が実現されることを示している。

図3.1.14は他方式の代表例としてポテンシャル法をとりあげ、それぞれの回避性能の比較をしたシミュレーションである。 P_r , P_t , P_o は同じであるが、 $V_r = [25,25]^T$ 、 $V_o = [10,-15]^T$ 、 $S_T = 0.3\text{sec}$ とし、a)が本方式のファジィ障害物回避シミュレーション、b)がポテンシャル法を用いた回避シミュレーションである。障害物回避ベクトルは次のようなポテンシャル関数を用いた。

$$O = G \cdot (P_o - P_r) / |P_o - P_r|^2 \quad (3.1.11)$$

ここで、Gは反発力を決定するゲインであるが、シミュレーションでは50に設定した。ファジィを用いた本方式は障害物の後方を通る効率的な回避動作を行なっているが、ポテンシャルを用いた場合、この条件のように移動ロボットと目標点の間に障害物が存在すると、障害物に押されるかたちで移動ロボットが大きく弧を描いて回避動作をとる。

図3.1.15ではさらに $V_r = [30,30]^T$ 、 $P_o = [180,180]^T$ 、 $V_o = [-15,-15]^T$ という設定で、障害物が正面から

3.1 ファジィ障害物回避制御アルゴリズム

接近してくる場合についてシミュレーションを行なった。但し、この例では図3.1.3 a)のような方位のメンバーシップ関数だと正面からの接近($\theta_s = 0^\circ$)に対応できないため、正面方位のメンバーシップ関数を追加で定義し、このときの静的危険度を右が危険とした。a)はファジィ障害物回避であるが、この追加ルールにより近づいてきた障害物に対して右が危険と判断し、左に操舵するルールが働いて回避が成功したことがわかる。b)はポテンシャル回避の停留問題の典型的な例で、正面からくる障害物がロボットに接近すると(3.1.11)式の分母が小さくなり、大きな反発力がロボットに加わる。その結果ロボットは真後に押し戻され脱出できなくなる。ここではたまたま数値演算の誤差からかろうじて脱出できた例を示した。これらの結果よりポテンシャル回避に比べファジィ障害物回避がよりフレキシブルな回避性能を有することがわかる。

図3.1.16は移動ロボットの速度を変化させた場合の回避性能を示す。すべての条件は図3.1.11、図3.1.15と同じに設定し、 V_r だけを障害物とロボットとの距離に比例して変化させた。a)は左前方から、b)は正面から障害物が接近した場合で、いずれも速度変化がない場合よりも大きな回避コースをとらないためより効率的な回避ができることがわかった。

3.1.8 結 言

動的障害物を有する環境において、ファジィ推論を用いて状態の危険度を認識しながら、人間と同じようなフレキシブルな知的回避ができるファジィ障害物回避アルゴリズムを提案した。本方式では、「危険度」というあいまい尺度を導入することにより、状態空間を2つのパラメータによるクラスタリングで扱うことが可能になり、行動決定アルゴリズムの記述性が良くなったと考えられる。また直感的状況判断による粗い危険度認識と回避方向を定める操作量決定の2つのフェーズで階層的にアルゴリズムを構成したことにより、行動決定メカニズムを明示的に表現できるようになり、種々の戦略を包含することが容易になった。

この方法を用いると障害物がかなり複雑な動きをした場合にもフレキシブルに対応できるだけでなく、ルールの変更などによる回避性能の変化も容易に実現できる。また従来の代表的なポテンシャル法の場合、目標物とロボットの間に障害物が存在すると、ロボットは停留というデッドロック状態に陥る可能性があったが、本方式ではメンバーシップ関数の設定によりこのような問題を避けることができる。本報告で示した以外にもさまざまなシミュレーションを通して多くのケーススタディを行った結果、人間と同じような柔軟な回避行動が確認され、提案した手法の有効性を検証することができた。

最後に、本方式は自律移動ロボットの障害物回避問題においてファジィ理論を最初に応用した研究であり、現在では他の同様の研究者にも本方式に関する発表論文が引用されていることを付け加えておく。

3.1 ファジィ障害物回避制御アルゴリズム

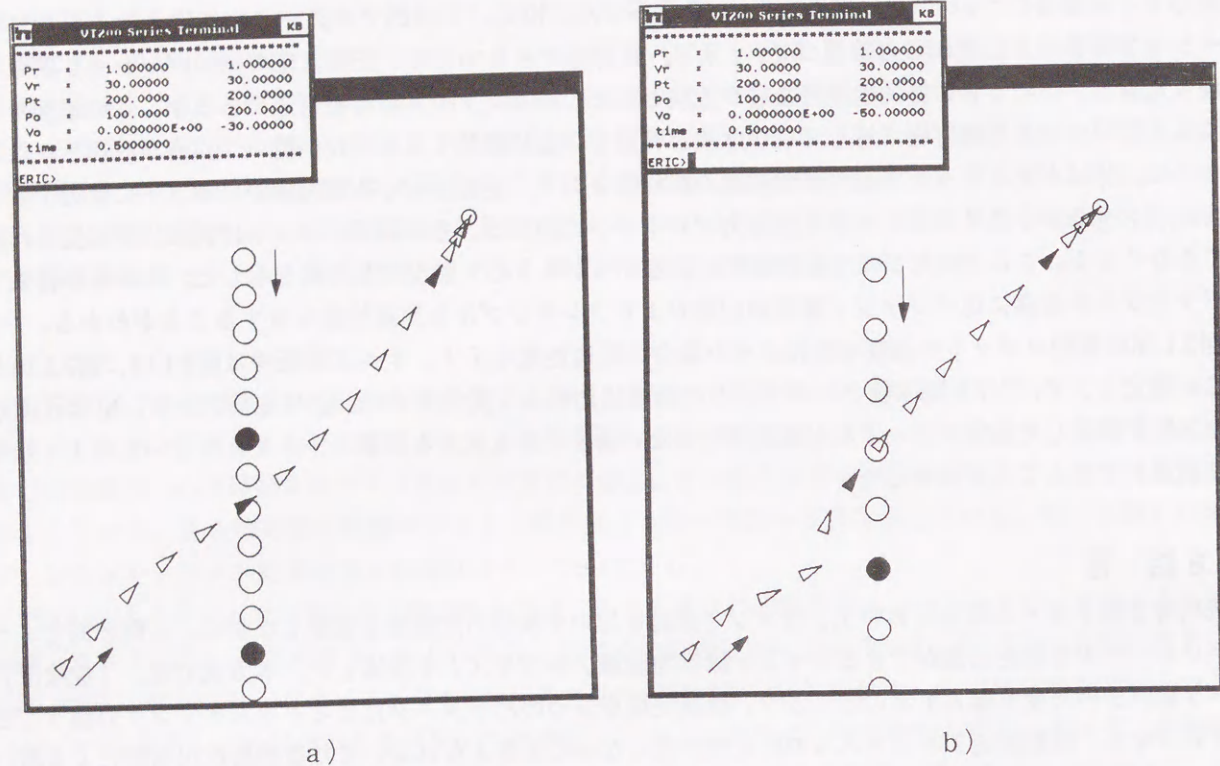


図3.1.11 障害物回避シミュレーション結果 (1)

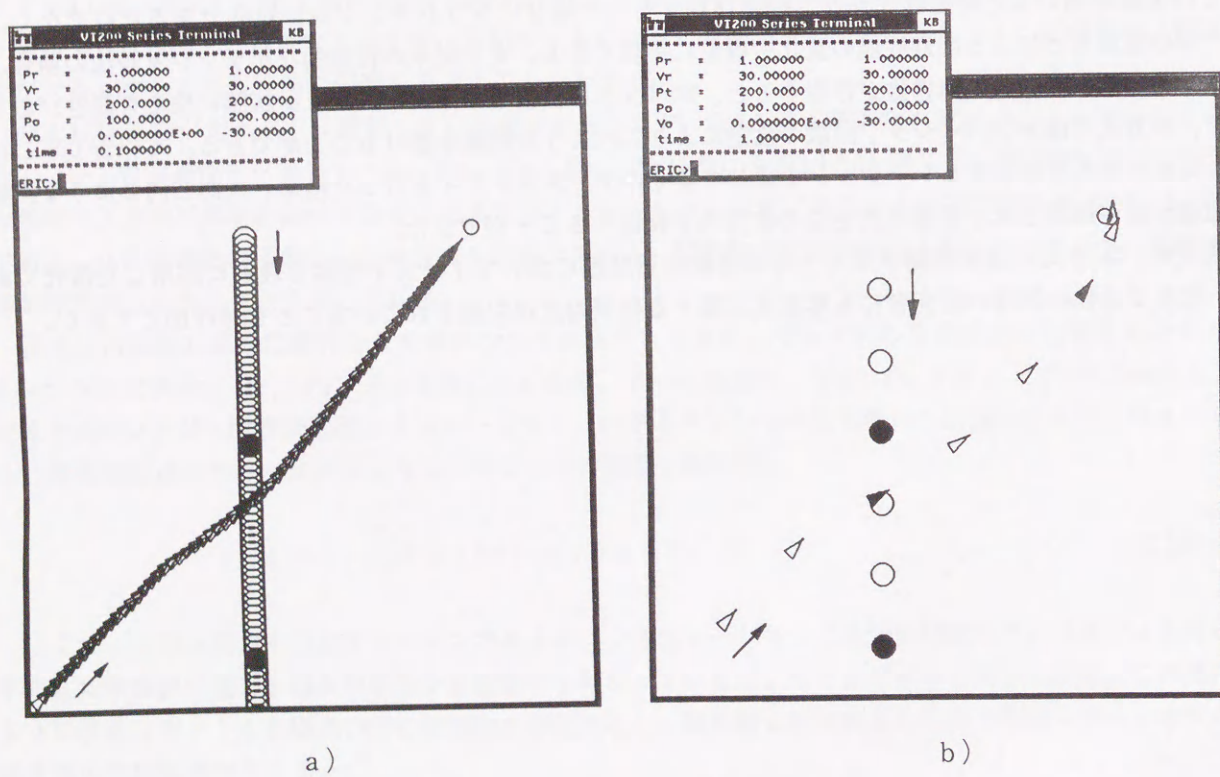


図3.1.12 障害物回避シミュレーション結果 (2)

3.1 ファジィ障害物回避制御アルゴリズム

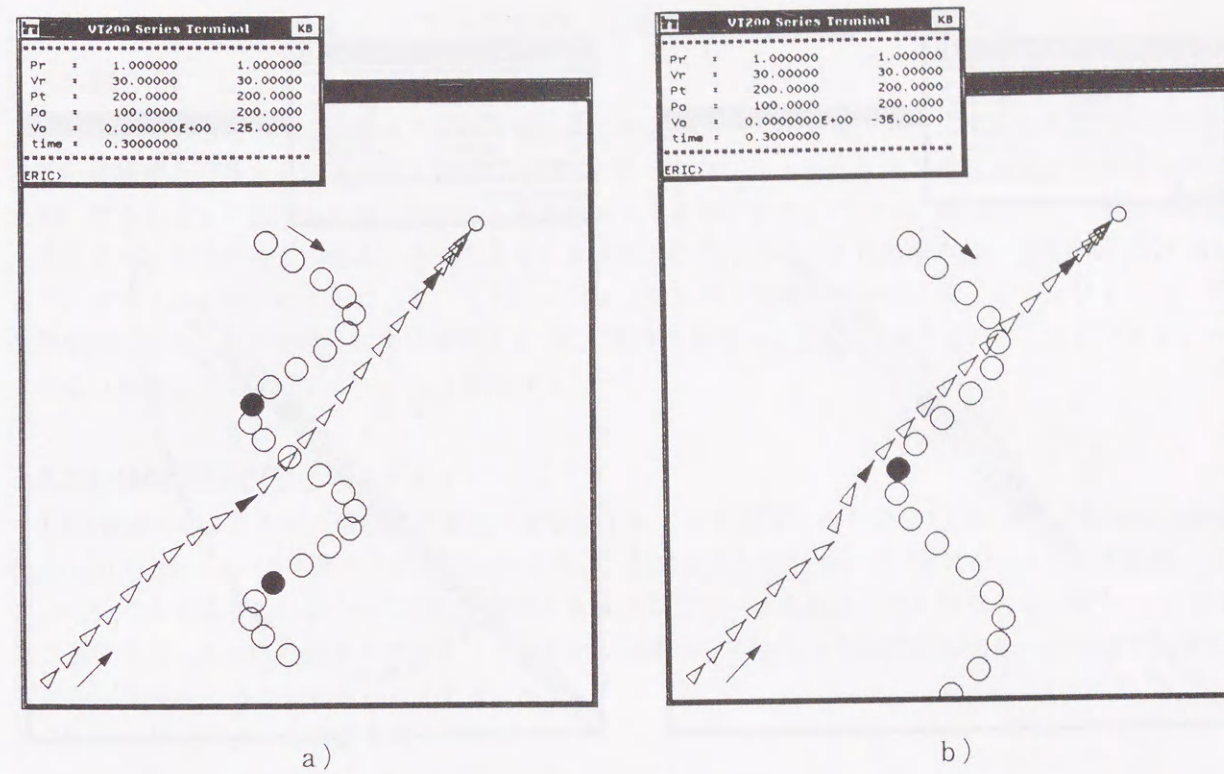


図3.1.13 障害物回避シミュレーション結果 (3)

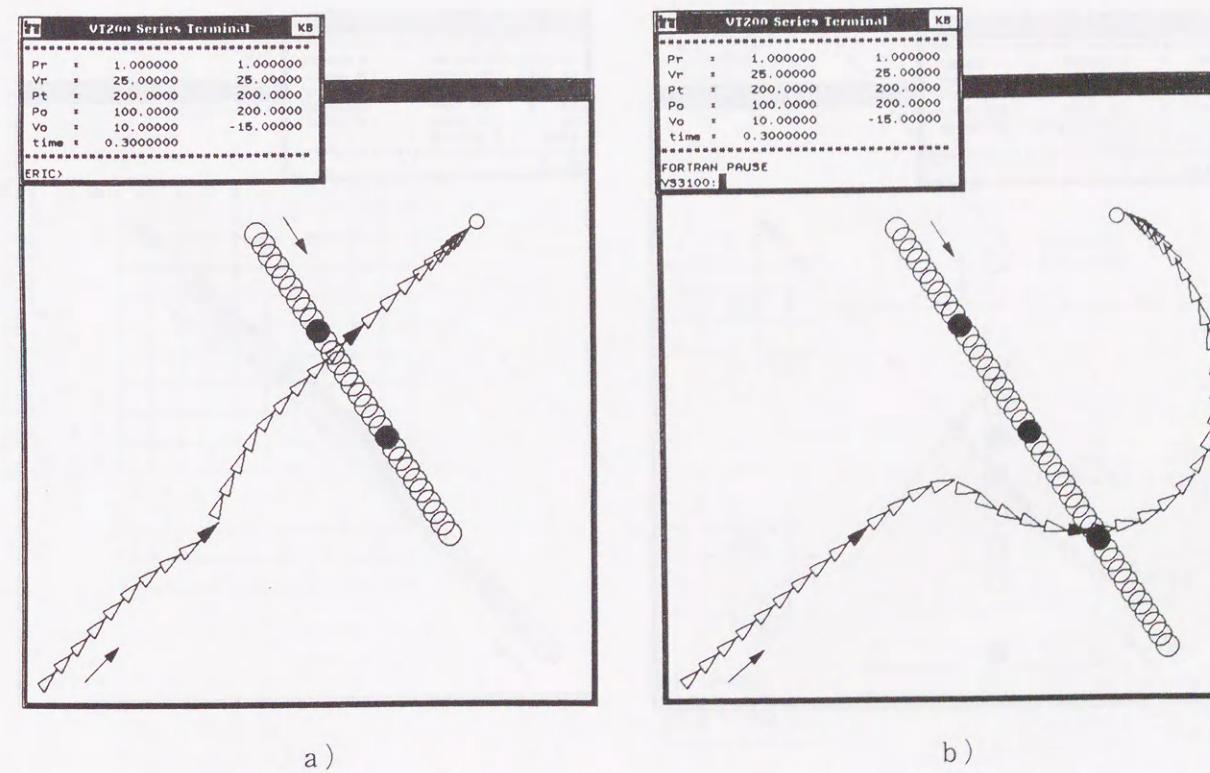


図3.1.14 障害物回避シミュレーション結果 (4)

3.1 ファジィ障害物回避制御アルゴリズム

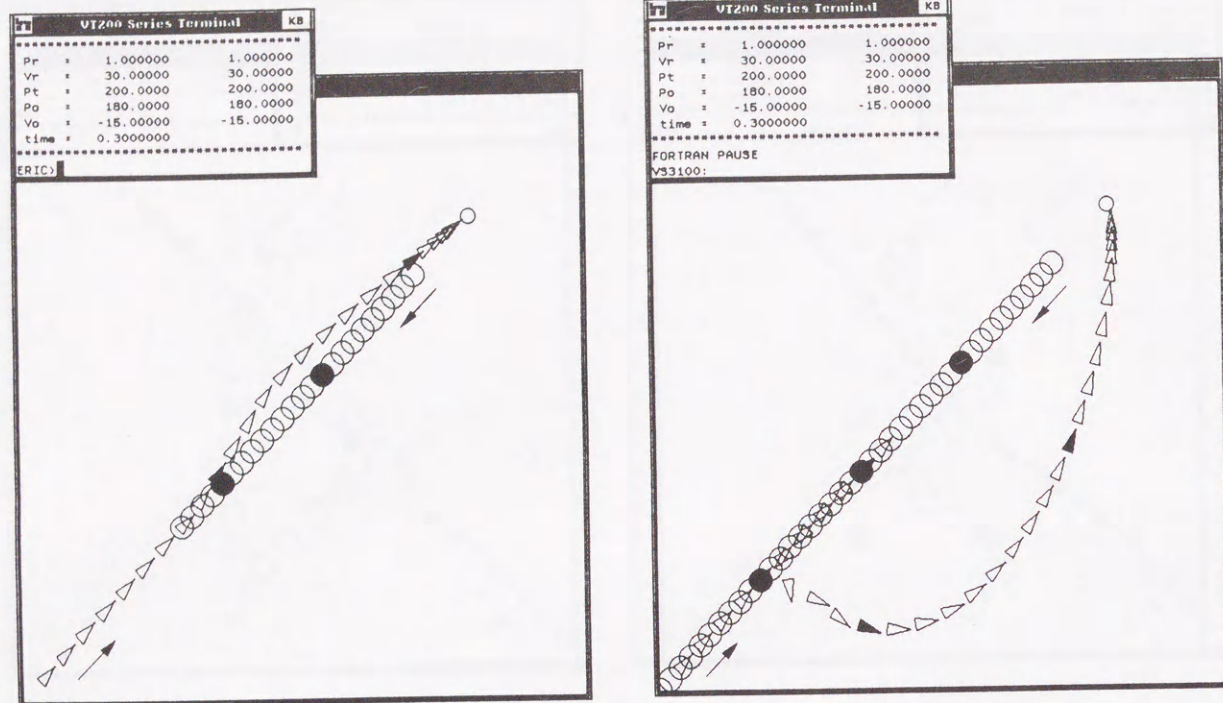


図3.1.15 障害物回避シミュレーション結果 (5)

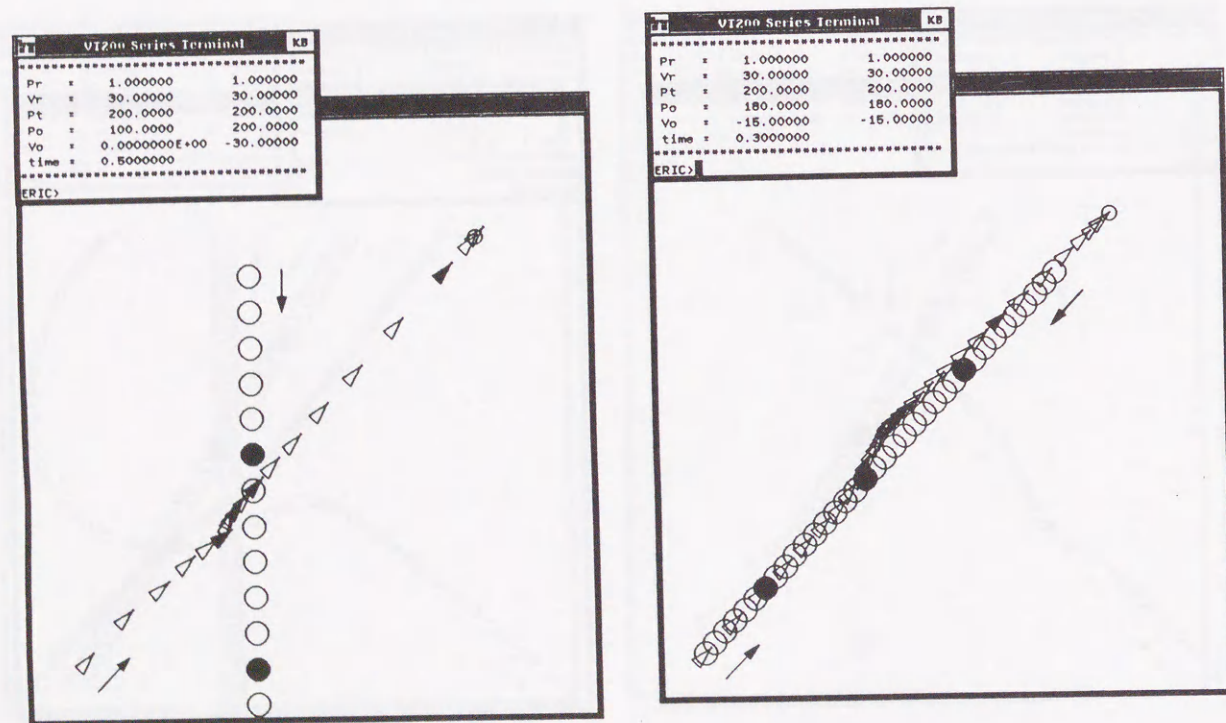


図3.1.16 障害物回避シミュレーション結果 (6)

3.2 CMACニューラルネットワークを用いた障害物回避戦略の獲得方法

3.2 CMACニューラルネットワークを用いた障害物回避戦略の獲得方法

3.2.1 緒言

前述の方式の回避戦略にあたるルールはあらかじめ人間が与える必要があり、妥当な決定ルールの作成には試行錯誤のため多大な時間を要する。この修正プロセスはシミュレーション結果のフィードバックにより自動化することも可能であるが、実際に人間にロボットを操作させ、うまく回避したケースを制御戦略として学習させる方法が考えられる。そこでさらに本研究では汎化作用があり効率の良い学習が可能なCMAC学習アルゴリズムを用いることにより、シミュレータ上で人間に回避操作を行わせ、ディシジョンテーブルの制御戦略にあたる操作特性をスキル獲得する方式の提案を行なった[1][2]。本方式の有効性をシミュレーションにより確認したのでこれについても報告する。

3.2.2 CMAC学習アルゴリズム

J.S.Albusらにより小脳の情報処理機構の数学モデルとして提案されたCMAC(Cerebellar Model Arithmetic Computer)[3]はニューラルネットワークの一手法で、複雑な入力空間マップの要素をいくつかの細胞(セル)に分担保持させることにより、効率良く学習できるアルゴリズムを提供したものである。田中ら[4]はCMACを冗長マニピュレータの逆キネマティクス問題に適用することにより制御応用にも十分高速で効率的な学習アルゴリズムであることを示している。

CMACは以下のような特長をもっている。

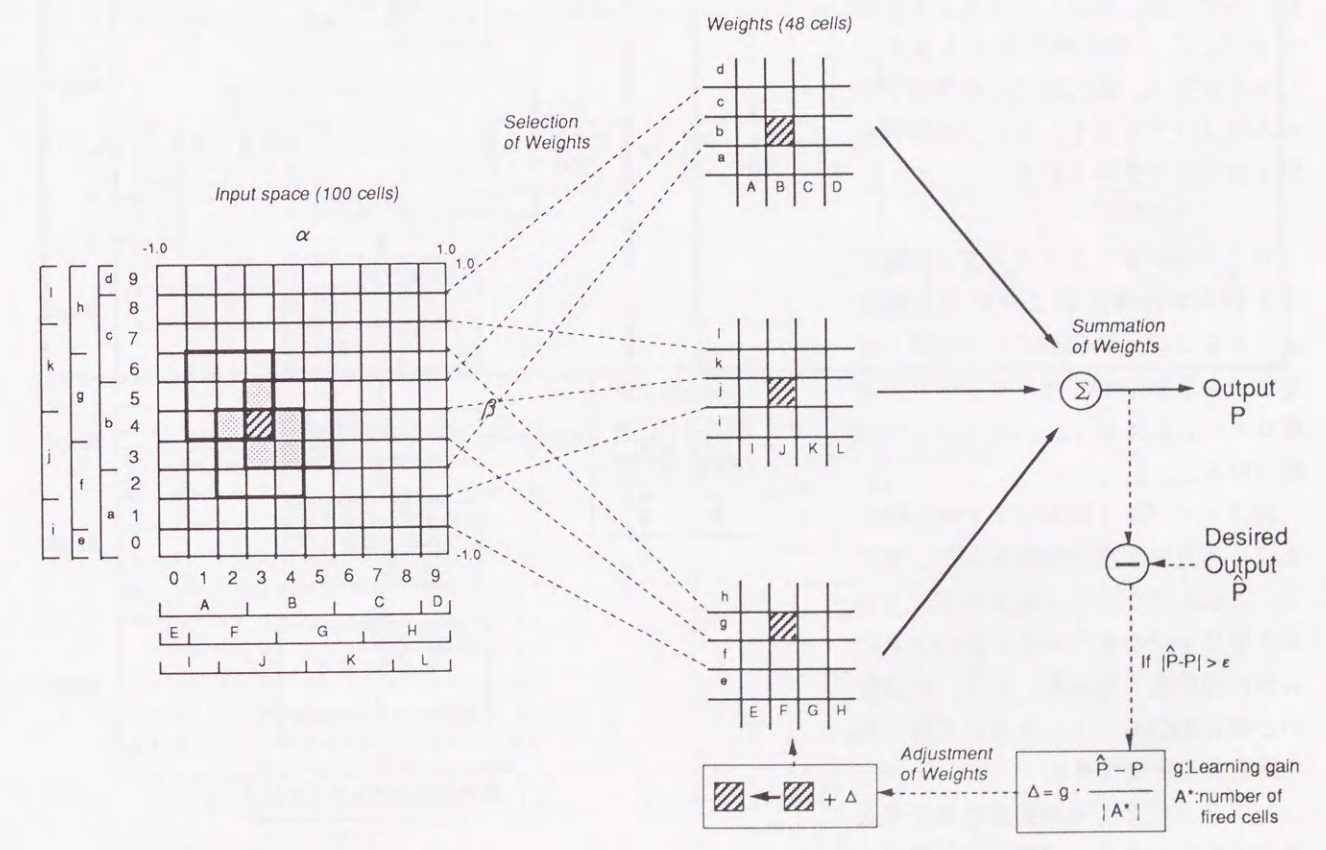


図3.2.1 CMAC学習アルゴリズム

3.2 CMACニューラルネットワークを用いた障害物回避戦略の獲得方法

- 1) 強力な汎化作用をもつ。
- 2) 従来のテーブルマッピングよりはるかに少ない記憶容量で足りる。
- 3) 学習速度が比較的速い。
- 4) 入力が多次元のマッピングに向いている。

図3.2.1はCMAC学習アルゴリズムの基本的なメカニズムを図示したものである。この例では本方式のディシジョンテーブルの学習を意識して、入力空間は100要素(10×10)としている。入力空間マップの3つの太い四角形が斜線要素に対する分散記憶を示しており、図中央の3つのセルの斜線部分の荷重(Weight)に対応する。これらの荷重合成をとり、教師データ(本研究ではオペレータの操作データに相当)と比較して重みの修正値を計算し、これにより荷重調整(修正学習)がなされる。尚、この例では入力空間要素100に対し、記憶空間要素48と1/2以下のメモリで表現できる。

3.2.3 オフラインにおける制御戦略の学習プロセス

本報告では、オフラインで仮想的に人間にロボットの操作をさせ、CMAC学習アルゴリズムを用いて前節で述べた制御戦略を記述したディシジョンテーブルを自動生成することを考える。学習の後、完成したマップを戦略マップとして障害物回避シミュレーションを行う。図3.2.2はこの学習プロセスのフローを示す。マップの学習は以下の手順で進められる。

1) ステップ0: まずファジィ推論により得られた静的および動的危険度 α 、 β を2次元のCMAC入力空間と仮定する。最初CMACセルのすべての荷重はゼロにしておく。(本ステップは初回のみ)

2) ステップ1: ロボットと障害物および目標点の2次元状態を人間に表示し、これを見ながら人間がロボットのステアリングを操作する。このときの人間の操作量(回避角)を \hat{P} 、危険度から現在のCMACセルを用いて得られた回避角を P とする。

3) ステップ2: セル要素の修正量 Δ を次式で算出する。(CMAC学習アルゴリズム)

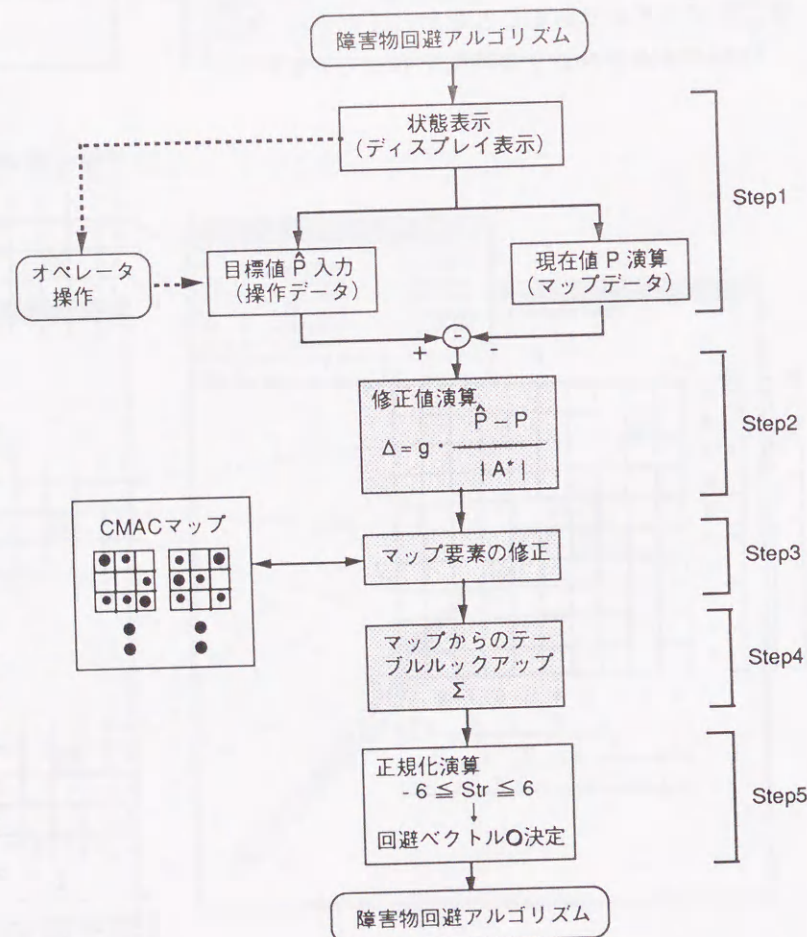


図3.2.2 CMACによる戦略学習アルゴリズム

3.2 CMACニューラルネットワークを用いた障害物回避戦略の獲得方法

$$\Delta = g \cdot (\hat{P} - P) / |A^*| \quad (3.2.1)$$

- 但し、 g は学習の利得係数、 $|A^*|$ は発火細胞の数。 $|\hat{P} - P| \leq \epsilon$ ならば学習を完了する。
- 4) ステップ3: 修正量 Δ を発火に関係したセルの要素すべてに加え、CMACセルを修正する。
 - 5) ステップ4: 修正された各セルからテーブルルックアップにより入力空間マップの該当要素を求める。
 - 6) ステップ5: 得られた要素の値をステアリング角度に正規化する。この結果を回避ベクトル O の値としてメインフローに返す。

シミュレーションにおいては上記ステップ1~5をディシジョンテーブルによる回避ベクトル決定部のかわりに挿入し、オペレータの操作学習を何回か繰り返す。当然のことながら、学習終了後、出来上がったCMACマップはそのままオペレータの操作特性が盛り込まれた戦略マップを示すディシジョンテーブルとして利用できる。このようにある戦略をあらかじめマップの形で学習しておく、これを置き換えることによ

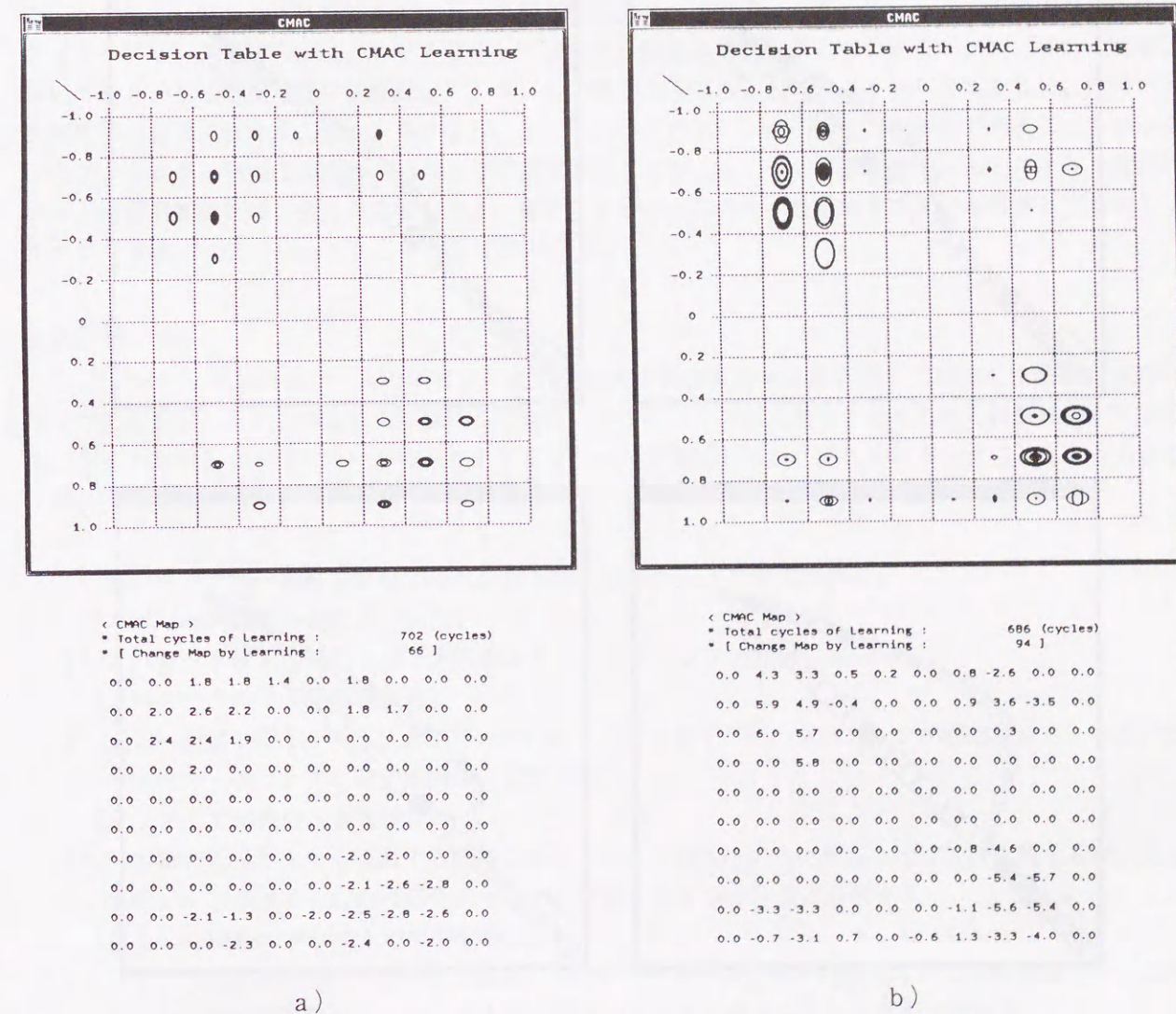


図3.2.3 操作学習の結果得られたディシジョンテーブル (戦略マップ)

3.2 CMACニューラルネットワークを用いた障害物回避戦略の獲得方法

り、容易にロボットの回避制御戦略を変えることができ、場合によってはこのような戦略をいくつか用意しておきトップダウンで戦略切り替えを行うことなども可能となる。

3.2.4 シミュレーション

まずはじめに前節で述べた戦略マップのCMAC学習アルゴリズムを用いて、あらかじめオペレータの制御戦略を学習させた結果について述べる。図3.2.3は学習の結果得られたディシジョンテーブル(戦略マップ)で、横軸は静的危険度 α 、縦軸は動的危険度 β を示している。いずれの軸も中央付近は安全、両端近くは危険な状態と考えることができる。図は、さまざまな方向から出現する移動障害物に対し、同一オペレータがある一定の制御戦略を念頭に置きながらリアルタイムで回避操作をしたときに得られたマップである。

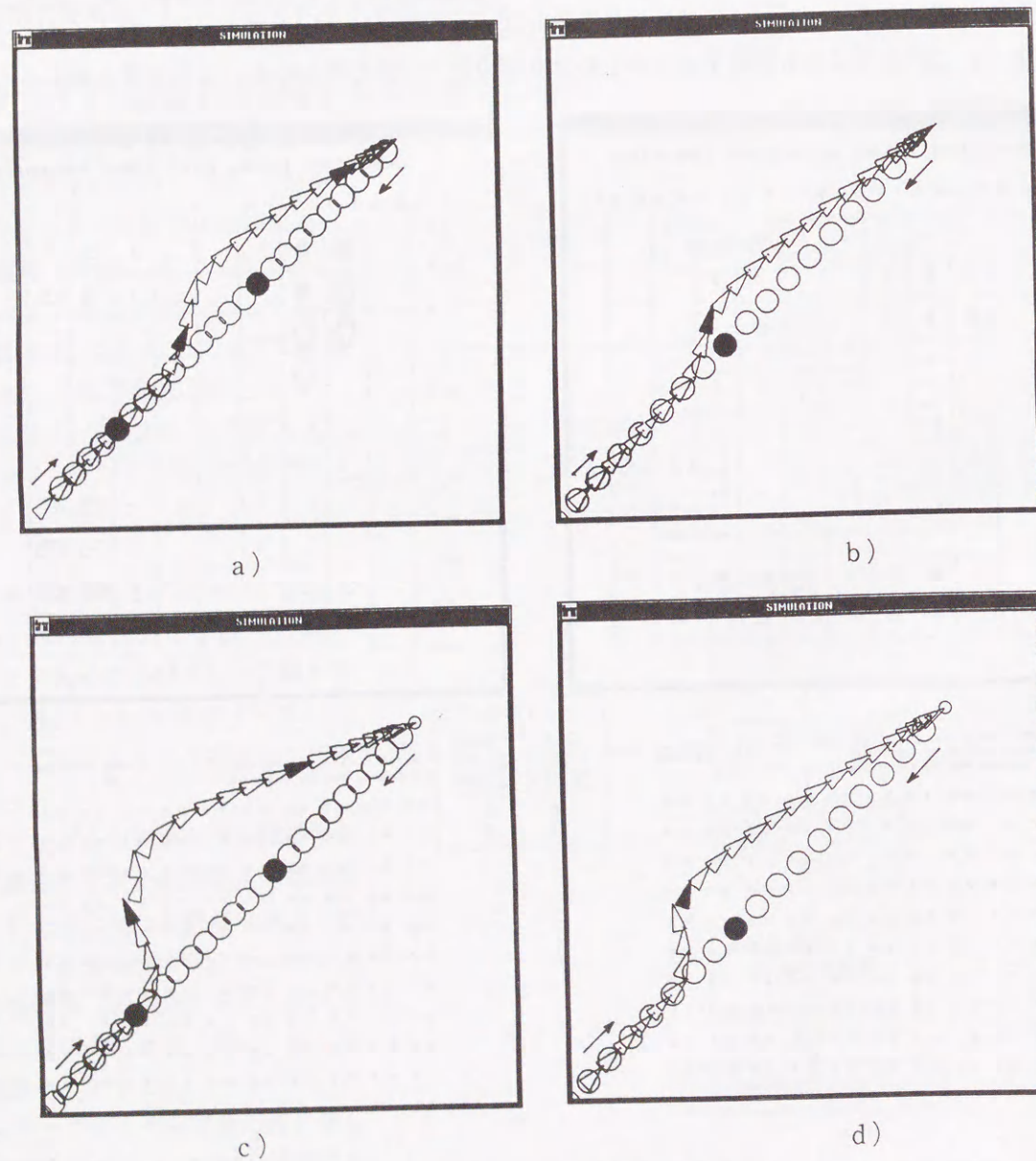


図3.2.4 得られた戦略マップによる障害物回避シミュレーション

3.2 CMACニューラルネットワークを用いた障害物回避戦略の獲得方法

a) は障害物の遠くから余裕をもってゆっくりと回避(あまり大きく操舵しない)する戦略[戦略1]をもった場合に得られたマップ、b) は障害物のかなり近くで突然大きく回避する戦略[戦略2]の場合に得られたマップを示している。図中の数字はその上のマップに楕円(長径が大きさ)で表現した回避角の大きさを示しており、正は右回避(縦楕円)、負は左回避(横楕円)を示している。1.0が 15° に対応しており、図で最大はほぼ5.0で 75° に対応している。学習の利得係数 g は0.7、入力空間が α 、 β それぞれ10段階で3つずつに分散記憶セルを設けたので、発火細胞の数 $|A^*|$ は3である。

a) は比較的小きな楕円(回避角小)が均等に広がっている(安全な領域ですでに回避している)のに対し、b) は大きな楕円(回避角大)が偏在している(危険な状態になるまで回避しない)ことがわかる。試行回数はいずれも約700回程度であるが、実際にマップを修正した有効学習はa)が66回、b)が94回で、比較的少ない回数で学習が収束していることがわかる。

ここで得られた戦略マップをもととのディシジョンテーブルと置き換えて、障害物回避シミュレーションを行わせたときの回避結果を図3.2.4に示す。大きな丸は障害物、小さな丸は目標点、三角形はロボットを表している。矢印は移動方向で、黒く塗りつぶしたところは3秒ごとの同一時刻における位置を示している。a)、b) はそれぞれ戦略1のもとでの障害物のスピードが遅いときと速いときの回避例で、あまり大きくステアリングを切らずに、しかもももってゆっくり回避していることがわかる。c)、d) は戦略2を用いており、初期位置・移動速度はa)、b)と同様の条件である。これより明らかなように、回避が比較的障害物の直前で行われており、かつa)、b)に比べステアリングの切り方が極端であることもわかる。

このようにある戦略をあらかじめマップの形で学習しておく、これを置き換えることにより、容易にロボットの回避制御戦略を変えることができ、場合によってはこのような戦略をいくつか用意しておきトップダウンで戦略切り替えを行うことなども可能となる。

3.2.5 考察

シミュレーション例では一人のオペレータが両極端な制御戦略を念頭に置いて操作した場合に得られたマップのみを示したが、異なる人が操作した場合のオペレータの個人差などももちろん特徴識別が可能である。シミュレーションはこのほかにもさまざまなケースで実施したが、明らかになったことをまとめると以下ようになる。

- 1) 同一人物が同一戦略で試行すると、比較的一定のマップが学習される。
[オペレータ固有の戦略の不変性]
- 2) オペレータにより得意な回避方向があり、ステアリングの傾向に差がでる。
[オペレータの個人差の検証可]
- 3) 正しい回避を行うとマップ上でだいたい左上が正、右下が負のステアリングが出現する。右上、左下のコーナーはステアリング方向が一定でない。
[マップ上での戦略評価可能]
- 4) 障害物の近傍で大きく回避する戦略の場合、マップのコーナーに大きな操作量が出現するが、遠方からゆっくり回避する戦略の場合、マップに似た大きさの操作量が分散する。
[異なった戦略のマップ上での特徴識別]
- 5) マップの左と右の端の列は学習の結果、ステアリングがほとんど出現しないことがわかった。これは正しい回避をする限り、このような危険な状態が発生し得ないことを意味している。
[特異(危険)状態の戦略データが獲得困難]

3.2 CMACニューラルネットワークを用いた障害物回避戦略の獲得方法

- 6) 学習完了マップを用いた自律動作はほぼその人の特性どおりに動作することが確認されたが、場合によっては学習時とは異なった逆回避や接触を行うときがあり、学習ゲインをより小さくして試行回数をもっと増やす必要がある。学習ゲイン $g = 1.0$ (単発学習) の場合、ミス操作や不定戦略をすべて学習してしまうため、 $g \leq 0.7$ 程度が適当であるものと考えられる。

[学習精度の向上]

3.2.6 結 言

本報告では、前節の障害物回避制御アルゴリズムの中で制御戦略を記述したディジションテーブルをオフラインで自動生成するため、汎化作用があり、効率の良い学習が可能であるCMAC学習アルゴリズムを適用した。この手法を用いると、実際にオペレータにロボットを操作させたときの回避特性・制御戦略等のスキルを容易に獲得できる。

また2次元平面内における移動ロボットの障害物回避問題に対して計算機シミュレーションを行い、CMAC学習を用いることによりオフラインでオペレータの実際の操作を学習させたところ、人間の操作特性(回避戦略)を表すディジションテーブルを得ることができた。この結果、オペレータにロボットを操作させたときの制御戦略等のスキルを容易に獲得でき、その戦略によりオペレータの個人差を反映した自律制御が可能ながシミュレーションにより明らかとなった。ここで得られた戦略マップはそのままディジションテーブルとなり、オペレータのノウハウを学習によってマップの形式で獲得することができる。この方法を用いると、例えば、さまざまな異なるオペレータの操作戦略をあらかじめ獲得しておき実際の移動時に戦略を切り替えることにより、ロボットの障害物回避における特徴を微妙に変化させるようなことも可能となる。

4.1 ファジィ推論を適用したグリッドマップ経路探索アルゴリズム

第4章 経路決定アルゴリズムに関する研究

4.1 ファジィ推論を適用したグリッドマップ経路探索アルゴリズム

4.1.1 緒 言

一般に経路探索問題は、主に障害物の補集合を自由空間とするエリア内で最適経路を生成する自由空間経路探索と経路情報が格子(グリッド)状ネットワークで与えられているようなグリッドマップ経路探索がある。前者の自由空間経路探索は例えば、障害物のある広い部屋の中など移動可能な部分が経路ではなく空間でしかわかっていない場合の最適経路探索を行なう方法で、コンフィギュレーション空間法(Lozano-Perezら[1])、ハイブリッドセル表現法(例えば、登尾らのオクトツリー[2]、クウォドツリー[3][4])などが代表的である。これらは、自由空間をVisual graphに変換するためVグラフ法とも呼ばれるが、セルと呼ばれるノードに分割した後は組み合わせ探索になるためグリッドマップ経路探索に帰着する。

一方、グリッドマップ経路探索(例えば、Chengら[5]、藤井ら[6])は、建屋内の通路網などの経路情報があらかじめ既知である複雑な経路環境から最適な経路の組み合わせを探索することを対象としている。そのため、探索経路が複雑な場合、組み合わせの爆発が起こり、実行時間が膨大になることが知られている。これに対処する方法として通常 α カットや β カットなどの枝刈り法が用いられるが、大幅な改善は期待できない。そこでさらに探索を効率化するためA*アルゴリズム[7]などの一般的なヒューリスティックアルゴリズムが用いられる。しかしながら、ヒューリスティック手法は探索を効率化できる反面、最適解が得られる保証がないため、効率的でかついかに正確な解が得られるかがヒューリスティック探索にとって重要な問題である。さらにすでに述べたヒューリスティック手法はほとんどAI的な知的アルゴリズムを用いて構築されているが、このような方法の場合、人間の探索のためのヒューリスティクスを盛り込むことは困難である。

本報告ではスタートおよびゴールを与えられた複雑なグリッドマップにおける最適経路探索問題にファジィ推論を適用したヒューリスティック探索アルゴリズムの一手法を提案する[8]。本手法によると、従来の枝刈り探索に比べ探索効率が高く、比較的最適解を得やすい人間の直感に基づいたヒューリスティック探索が実現できる。また立体経路で本手法の探索シミュレーションを行なったので、これについても報告する。

4.1.2 探索経路のモデル化

本研究で対象とした探索経路は出発点および目標点の間の移動可能区域がグリッドマップで表現されているものとし、ゴールに対し後戻りするような経路を含む冗長探索は除いて考える。また各ノード間の経路に関するコスト情報(距離、必要エネルギー等)はあらかじめ与えられており、探索時の評価はこのコストを積算し、最小の経路の組み合わせを最適経路と決定する。

グリッドマップで与えられる経路は一般にグリッド数(格子の縦横の網目数)が大きくなると組み合わせ経路数が飛躍的に増大する。 $n \times n$ の2次元正方マップの場合、組み合わせルート数は、 $2(n^2 - n + 1)$ となる。例えば、 2×2 のグリッドマップの場合6通り、 3×3 で14通り、 5×5 になると42通りの経路が存在する。さらに、ビル内通路などの上下方向の移動も含めた3次元立体マップを考えると、例えば $2 \times 2 \times 2$ の場合でも82通りとなり、グリッド数が増えると組み合わせ経路数が爆発的に増加する。

探索経路の一例として、図4.1.1のような 5×5 のグリッドマップを考える。P1~P35は分岐点(ノード：ここではポイントと呼ぶ)、R1~R45は経路(ブランチ：ここではルートと呼ぶ)を示している。経路は一般性を考慮して完全な格子状ではなく、ところどころに適当な抜けをつくっている。左下が出发点、右上が目標点で、冗長探索を防ぐため左下から右上に対し戻す方向の探索は禁止するものとする。

また探索を効率的に行なうため、各経路には探索レベルを割り当てる。すなわちスタートからゴールに至

4.1 ファジィ推論を適用したグリッドマップ経路探索アルゴリズム

4.1.4 ファジィ推論の探索アルゴリズムへの適用

枝刈りを行っても複雑な探索問題の場合まだまだ探索時間がかかるため、さらに探索効率を上げる方法としてさまざまなヒューリスティックアルゴリズムが提案されている。A*アルゴリズムなどがその一例であるが、これらは探索効率を上げる反面、最適解が得られない可能性がある。さらにA*アルゴリズムの場合には現在地点からゴールまでのコスト情報が必要であるが、このような情報なしに最適解を求める場合には適用できない。そこで本研究ではヒューリスティック探索の一例として、ファジィ推論を経路探索に適用し、より人間に近い感覚で探索アルゴリズムを表現することを試みる。前節の図4.1.3で述べたように基本的な探索アルゴリズムはGOルーチンとBACKルーチンで構成されているが、GOルーチン内の探索続行を決定するReturnルーチンでさらにファジィによる今後の探索価値（これ以上探索を続ける価値があるかどうか）を判断させる。すなわち、探索を続けていく途中で、過去の評価値を常に見ながらこのまま現在の探索を続けていくべきかどうかの判断を各探索レベルごとに行なう。実際には現在の探索レベル (flvl=Level) と現在の評価値を探索レベルで割った値 (fev = ev/Level) を前件部のファジィ変数とし、どれだけ探索を続行すべきかという探索価値 (fgo) を後件部のファジィ変数としてファジィ推論を行なう。

ファジィ推論のメンバーシップ関数を図4.1.4に示す。現在の探索レベルflvlはレベル1を除くレベル2~10を3段階で表現し、LS (低いレベル)、LM (中位のレベル)、LL (高いレベル) と設定した。平均評価値fevは0~8までを2段階に分け、ES (小さい)、EL (大きい) とした。これに対し、探索有効度は0~1を6段階で表し、DBL (大きく戻る)、DBM (中位戻る)、DBS (少し戻る)、DGS (少し進む)、DGM (中位進む)、DGL (大きく進む) とした。このように、ファジィルールは全部で6つで記述できる。例えば、rule 1は「If (flvl is LS) and (fev is ES) then (fgo is DGL)」で、これは探索が低いレベルで、かつ平均評価値が低いときは大きく進むことを指示している。以上のルールにより、まず前件部の適合度をmin演算で求め、これらによる後件部の適合度を各ルールごとに出した後、それらルールごとの推論結果の重みつき平均として、全体の推論結果FGO*を求める。

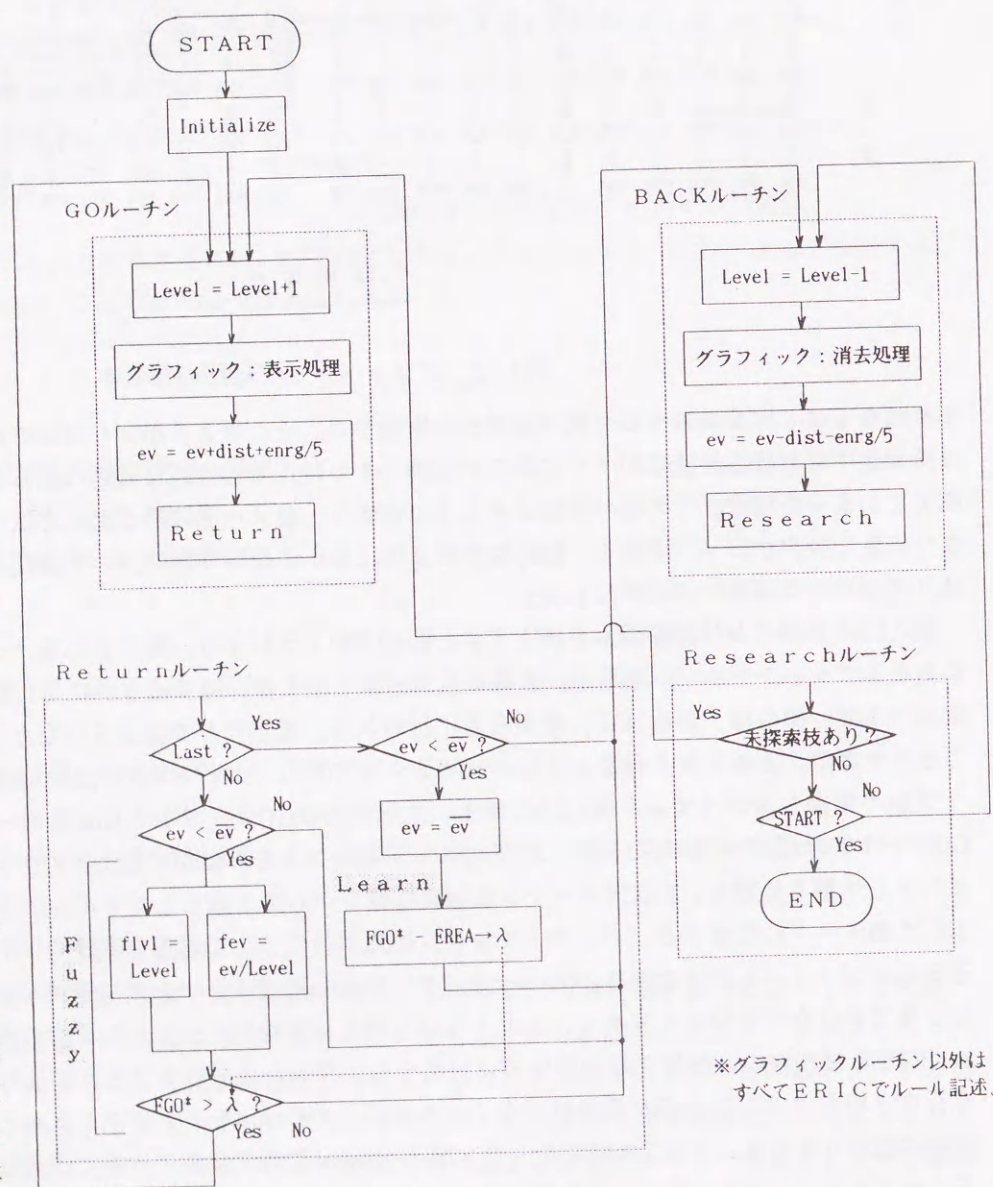


図4.1.3 ヒューリスティック探索アルゴリズム

4.1 ファジィ推論を適用したグリッドマップ経路探索アルゴリズム

ファジィ推論で得られたFGO*値に対して、実際に探索を先に進めるか、打ち切って戻るかの判断は進退しきい値λ (既知とする) により決定する。すなわち、FGO* > λならば前へ進む、FGO* < λならば後へ戻る、と定義する。このλはどの程度の探索有効度までなら探索続行を許すか、ということを示す値で、探索特性に大きな影響を与える。

図4.1.3に示すように、以上のファジィ推論による探索有効度のチェックをGOルーチン内のReturn処理ルーチンで1つ探索ルートを進進するごとに毎回行なう。これにより、βカットが行なわれない場合に対しても、探索価値の低い無駄な探索をカットでき、さらに効率の良い探索が可能となる。但し、本方式はあくまでヒューリスティックアルゴリズムであるため必ず最適解が得られるわけではないが、進退しきい値λの選び方

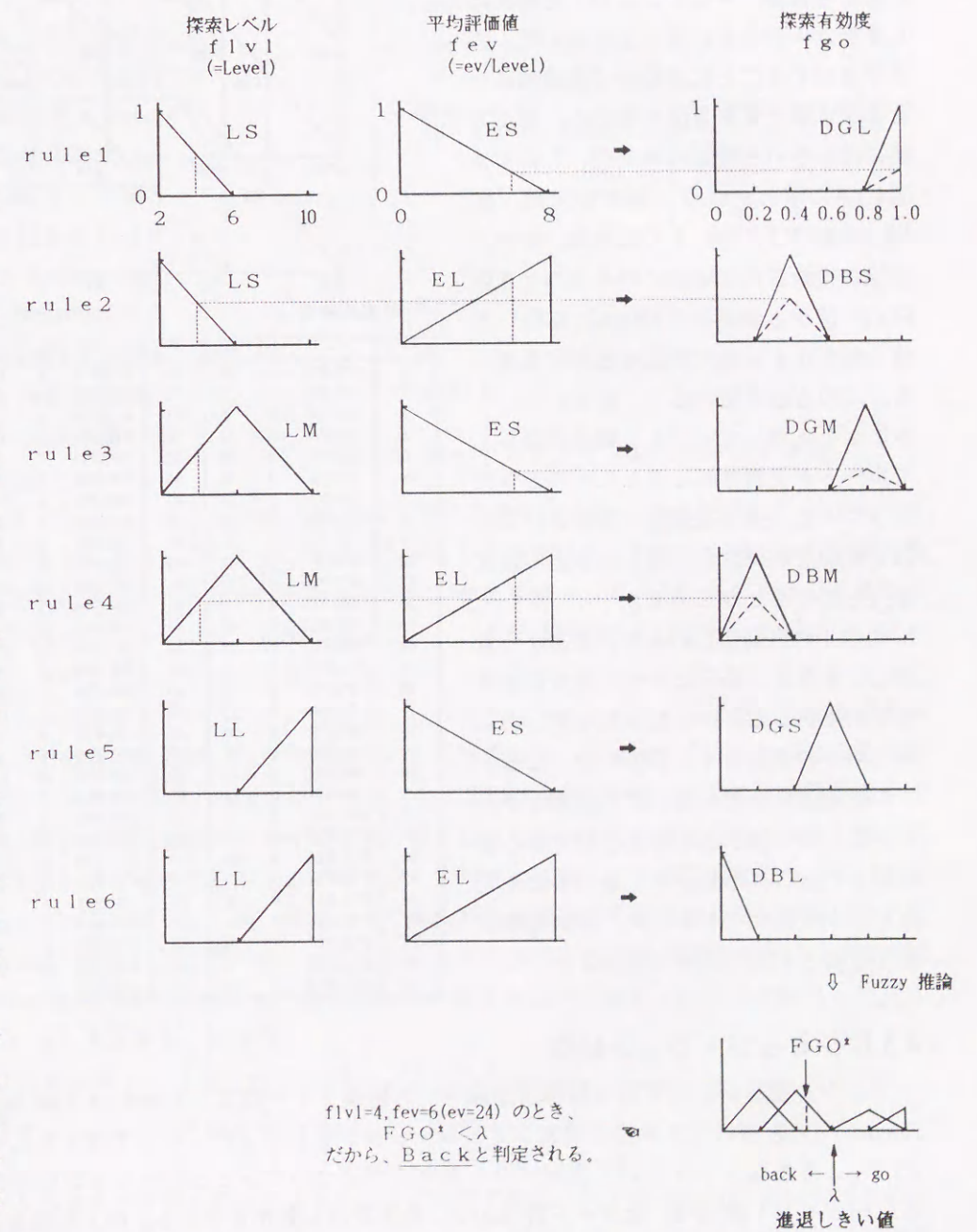


図4.1.4 メンバーシップ関数と探索ルール

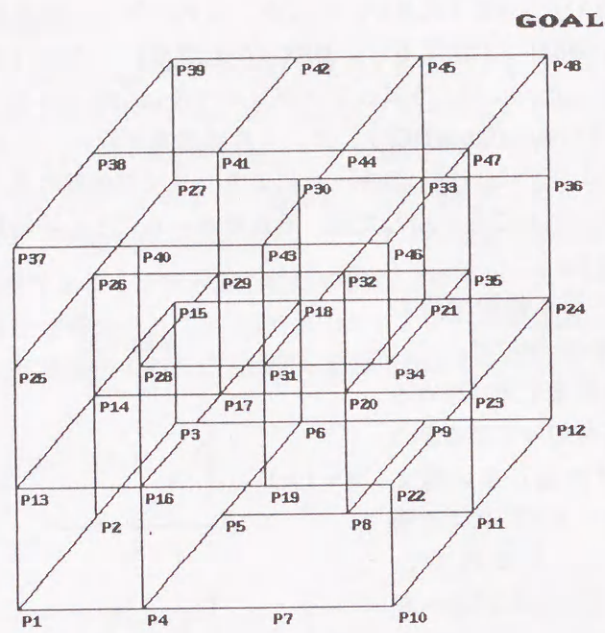
4.1 ファジィ推論を適用したグリッドマップ経路探索アルゴリズム

しだいでかなり正解率（最適解が得られる確率）を上げることは可能である。

4.1.5 進退しきい値λの自動修正

すでに述べたように、λの選定が探索特性に大きな影響を及ぼすが、一般に探索開始時には最適なλの値は不明であると考えられる。そこで、このλを最初は0または十分小さい値にしておいて、探索が進行するごとに自動的に最適値になるように修正する方法を考える。修正方法にはいろいろ考えられるが、ここでは図4.1.3に示したように探索が末端（図4.1.1の例ではP35）まで到達し、かつ評価値が更新された場合にのみそのときのFGO*値から余裕分（EREA）を引いた値を新たなλとして更新する方法を用いる。この方法を用いると、最初に全く小さいλを入力していても、最適経路としてルートが更新されるごとにそのときに持っていた（探索有効度-余裕分）より低い有効度の探索が出現したらその探索を打ち切ることができる。

このλの自動修正により、最初から最適なλを与えた場合に比べて多少は探索効率が下がることが予想されるが、あらかじめλの選定をする必要がなくなる。λの初期値は0でも良いが、通常λがとれると思われる値がある程度予想できる場合はその値より十分小さくかつ0より大きい値からスタートすれば、探索効率が若干改善されると考えられる。



ルート	ノード	dist	enrg	ルート	ノード	dist	enrg	ルート	ノード	dist	enrg
1	P1-P2	2	16	31	P17-P18	8	19	61	P31-P34	5	2
2	P2-P3	4	2	32	P16-P19	10	5	62	P32-P35	7	8
3	P1-P4	6	8	33	P17-P20	3	11	63	P33-P36	9	14
4	P3-P6	8	14	34	P18-P21	5	17	64	P34-P35	1	20
5	P4-P5	1	20	35	P20-P21	7	3	65	P25-P37	3	6
6	P5-P6	3	6	36	P19-P22	9	9	66	P27-P39	6	12
7	P4-P7	5	13	37	P20-P23	1	15	67	P28-P40	8	18
8	P5-P8	7	19	38	P21-P24	4	1	68	P29-P41	10	4
9	P6-P9	9	5	39	P22-P23	6	8	69	P31-P43	2	10
10	P8-P9	2	11	40	P23-P24	8	14	70	P33-P45	5	16
11	P7-P10	4	17	41	P13-P25	10	20	71	P34-P46	7	2
12	P8-P11	6	3	42	P14-P26	3	6	72	P35-P47	9	9
13	P9-P12	8	9	43	P16-P28	5	12	73	P36-P48	1	15
14	P10-P11	10	15	44	P17-P29	7	18	74	P37-P38	3	1
15	P11-P12	3	1	45	P18-P30	9	4	75	P38-P39	6	7
16	P1-P13	5	7	46	P19-P31	1	10	76	P37-P40	8	13
17	P2-P14	7	13	47	P20-P32	4	16	77	P38-P41	10	19
18	P3-P15	9	20	48	P21-P33	6	2	78	P39-P42	2	5
19	P4-P16	2	6	49	P23-P35	8	8	79	P41-P42	4	11
20	P6-P18	4	12	50	P24-P36	10	15	80	P40-P43	7	17
21	P8-P20	6	18	51	P25-P26	2	1	81	P41-P44	9	3
22	P10-P22	8	4	52	P26-P27	5	7	82	P42-P45	1	10
23	P11-P23	10	10	53	P26-P29	7	13	83	P43-P44	3	16
24	P12-P24	3	16	54	P27-P30	9	19	84	P44-P45	6	2
25	P13-P14	5	2	55	P28-P29	1	5	85	P43-P46	8	8
26	P14-P15	7	8	56	P28-P31	4	11	86	P44-P47	10	14
27	P13-P16	9	14	57	P29-P32	6	17	87	P45-P48	2	20
28	P14-P17	1	20	58	P30-P33	8	3	88	P46-P47	4	6
29	P15-P18	4	7	59	P31-P32	10	9	89	P47-P48	7	12
30	P16-P17	6	13	60	P32-P33	2	15				

図4.1.5 3次元グリッドマップ経路

4.1.6 シミュレーション結果

ファジィ推論が記述可能な制御用知識ベース構築ツールERIC (Extended Rule-based system for Intelligent Control) [9]を用いて本手法の探索シミュレーションを行なった。ここではさらに複雑な経路を想定し、図4.1.5のような2×3×3の3次元グリッドマップについてシミュレーションを行なった結果について述べる。全ポイント数は48、全ルート数は89で、最大レベル数が8である。尚、距離およびエネルギーの値は図4.1.1の例と同様に1~10および1~20の整数の中からランダムに選んだ。図4.1.1と同じところどころに

4.1 ファジィ推論を適用したグリッドマップ経路探索アルゴリズム

ルートの抜けをつくり、経路の規則性になるべく少なくしている。またワーキングメモリおよびルールは図4.1.6に示すような構成をとっている。WMタイプは4つで、()内に示したWM要素数は合計149である。ルールは図に示したのがルール群で21、()内が実際のルール数で合計49である。

この例を用いて、スタートからゴールまでの経路探索を、総当たり、枝刈り(β_cut)、ファジィ探索(β_cut+Fuzzy)、ファジィ+自動修正探索(β_cut+Fuzzy+Learn)で行なった場合の探索時間および処理回数を表4.1.1に示す。これよりあきらかなように、総当たりでは約5分かかった探索が枝刈りをして半分にならなかったが、ファジィ探索を行なうと約1分と非常に高速になっている。尚、ファジィ推論の際の進退しきい値λをここでは0.3とおいた。

さらにこのλを自動修正させると、λ=0、EREA=0.15のとき約1分50秒かかっているが、λ=0.3からスタートさせると修正しない場合に比べてごくわずかではあるが良くなっている。これらのことは、処理回数を見ても顕著に現われており、探索ルート数が総当たりと比べ、かなり減少していることがわかる。またβカット数はファジィ探索を行なうことにより、大幅に減っている。これは枝刈りが行なわれる前にファジィ推論によりかなりの枝刈りが行なわれていることを示している。

次に、本方式がどの程度探索効率を上げられるかという、いわゆるヒューリスティック度に当たる探索時間の減少率と、どれくらいの確率で最適解が得られるかという正解率について検討する。表4.1.2にこの結果を示すが、ファジィとファジィ+自動修正の場合についてλおよびEREAの値を変化させて探索時間減少率と正解率を求めた。但し、結果が経路の組み合わせにできる限り依存しないように探索経路のコスト値(距離、エネルギー)は乱数を用いて30通りの組み合わせをつくりすべての場合に対して探索を行なった結果を平均した。30種類の経路パターンに対し、正解(総当たりで得られた最適経路の組み合わせ)が得られた場合をi=1, 2, ..., nr (0 ≤ nr ≤ 30)、正解が得られたときのβカットのみの探索およびファジィ探索の探索時間をti、ti* (i=1, 2, ..., nr)とすると、探索時間減少率ηおよび正解率αは以下で算出した。

$$\begin{aligned} \text{探索時間減少率 } \eta &: (\sum (t_i - t_i^*) / t_i) / nr \\ \text{正解率 } \alpha &: nr / 30 \end{aligned}$$

ηは大きいほどヒューリスティック度が高いと考えられ1が上限である。例えばη=0.6ならば30通りの経路パターンに対してβカットのみのときの探索時間に比べ、ファジィ探索を行なったときの探索時間が平均40%で実行されたことを示す。表4.1.2よりλの自動修正なしのファジィ探索の場合、λ=0.3ぐらいがヒュー

ワーキングメモリ (WMタイプ: 4, WM要素: 149)

```
FPLAN.T
point (49)
route (90)
stuck (9)
control (1)
```

ルール (ルール群: 21, ルール数: 49)

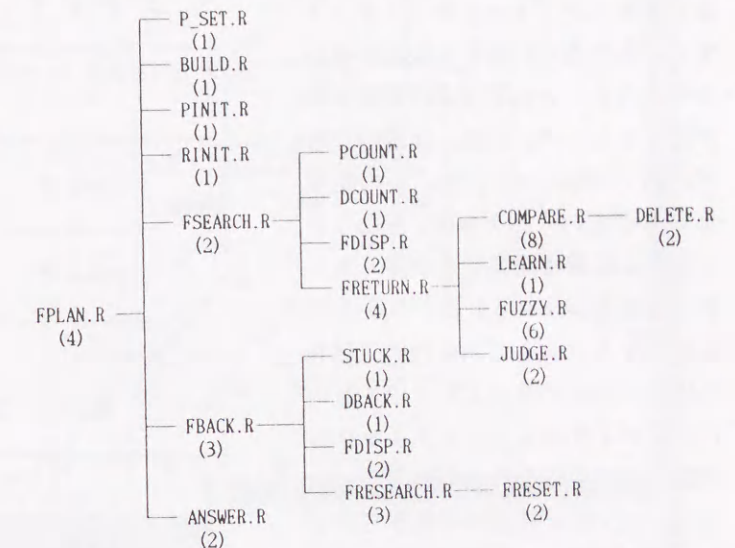


図4.1.6 ワーキングメモリおよびルール構成

4.1 ファジィ推論を適用したグリッドマップ経路探索アルゴリズム

リスティク度と正解率を考慮して最も最適な値であることがわかる。この場合9割の正解率である。

さらに、 λ を自動修正させた場合はEREAの選び方により微妙に結果が変わるが、 $\lambda=0$ からスタートすると当然最初は無駄な探索を強いられるため、あまり探索時間減少率が高くなる。以後、比較的正解率が高いEREA=0.15で、 λ の初期値を上げていくと $\lambda=0.3$ あたりから急激に探索時間減少率が高くなるが、正解率は $\lambda=0.4$ あたりから急に悪くなる。よって、自動修正付きの場合もある程度は λ を上げておいたほうがよりヒューリスティックな探索を実現できることがわかる。シミュレーション結果からあきらかなように $\lambda=0.3$ で修正をスタートしたとき、 β カットに比べ約70%の無駄な探索をファジィでカットしている。以上により、 λ が不明であっても0もしくは適当な小さい値に初期値をセットしておけば、自動修正により最適な λ を自動的に選び、ファジィによる枝刈りを行なうことができることがわかる。

最後に、図4.1.7にシミュレーション実行時の表示画面の様子を示す。これはファジィ+自動修正探索の例であるが、メインウィンドウ上部に λ の修正した結果を表示している。さらに画面下部のサブウィンドウには時々刻々のファジィ推論により得られた合成メンバーシップ関数およびその推論結果を表示している。探索ルートはその時点での最適経路が太線で残るようにし、決定ルートが更新されたことがわかるようにしている。

4.1.7 結 言

複雑なグリッドマップの経路探索問題において、ファジィ推論を用いて探索有効度をチェックすることにより、従来の枝刈り法に比べ探索効率が大幅に高い探索アルゴリズムを提案した。さらにこの探索特性を大

表4.1.1 探索時間および処理回数の比較

探索方式	探索時間 (分:秒)	探索ルート数	β カット数	ファジィ推論回数	自動修正回数
総当たり	4:59	719	—	—	—
β カット	2:43	441	137	—	—
ファジィ $\lambda=0.3$	0:55	145	10	107 go: 71 back: 36	—
ファジィ + 自動修正 $\lambda=0.0$ EREA=0.15	1:49	290	50	192 go: 171 back: 21	11
$\lambda=0.3$ EREA=0.15	0:54	141	6	106 go: 69 back: 37	6

表4.1.2 探索時間減少率および正解率の比較

ファジィ	λ の自動修正	パラメータ		探索時間減少率	正解率
		λ	EREA	$\frac{\sum (t_i - t_i^*)}{t_i} / n_r$	$n_r / 30$
あり	なし	0.25	—	6.775 / 30 = 0.226	30 / 30 = 1.00
		0.3	—	17.405 / 27 = 0.645	27 / 30 = 0.90
		0.35	—	9.362 / 12 = 0.780	12 / 30 = 0.40
	あり	0.0	0.1	5.842 / 17 = 0.344	17 / 30 = 0.57
		0.0	0.12	6.232 / 20 = 0.312	20 / 30 = 0.67
		0.0	0.15	8.124 / 29 = 0.280	29 / 30 = 0.97
		0.2	0.15	8.967 / 29 = 0.309	29 / 30 = 0.97
		0.3	0.15	17.702 / 26 = 0.681	26 / 30 = 0.87
		0.4	0.15	8.789 / 11 = 0.799	11 / 30 = 0.37
		0.5	0.15	0	0 / 30 = 0
なし	総当たり	—	—	30 / 30 = 1.00	
	β カット	—	—	30 / 30 = 1.00	

4.1 ファジィ推論を適用したグリッドマップ経路探索アルゴリズム

きく左右する進退しきい値の自動修正方法についても検討を加えた。このヒューリスティック探索アルゴリズムの有効性を検証するため、立体グリッドマップを用いて経路探索シミュレーションを行ない良好な結果を得た。本方式を用いた場合、総当たりと比べ約1/5、 β カットに比べても約1/3程度の探索時間で最適解を得ることが可能であることがわかった。本方式を用いると複雑な経路を対象とした経路計画問題において、領域分割後のノード探索などの単純繰り返し評価の知的な効率化が図れるものと期待できる。

また最近、このような大規模探索問題に対してニューラルネット (NN) [10]や遺伝的アルゴリズム (GA) [11]を用いた研究がいくつか発表されている。NNを用いた手法は、本来関数近似を得意とするためアナログ的な問題、すなわち緻密な経路または軌道を自動生成するには向いているが、離散的なノードをもつ一般的な経路探索に対しては収束が遅く、さらに初期値の選び方しだいではローカルミニマムに陥りやすい傾向がある。これに対し、GAを用いた手法の場合、離散的な経路探索問題に関する限りNN探索に比べて比較的収束速度も速く (演算コストはNNの約数分の一以下) かつ正解 (最適解) を得る可能性も高いが、最適解近傍での収束は一般に困難で、またランダム探索であるため得られる解に再現性がない。これらはいずれも全探索 (総当たり) とヒューリスティック探索の中間に位置するもので、大まかに言ってNNは個の学習としての局所探索、GAは種の適応としての大域サンプリングを得意としている[12]。しかしながら、巡回セールスマン問題 (TSP) やジョブショップ・スケジューリング問題 (JSSP) など複雑かつ大規模な探索問題に対して、ある特定の条件のもとで解探索を工夫することにより、従来の枝刈りなどの探索に比べてやや効率的 (探索時間はせいぜい数分の一程度) であることが報告されているのが現状である。本研究の探索手法は、これらの手法とは全く異なっており正解こそ得る確率は低いが、ファジィルールを用いて評価規範を構築しているため、探索自体に人間のヒューリスティクスを入れることができる。さらに実際の探索におけるノードの評価回数や探索のための評価戦略などもルール記述によって変更することができ、探索評価にある種の判断やアプリアリな知識を含めて効率化を図ろうとするようなケースにおいては本手法は有効であると考えられる。

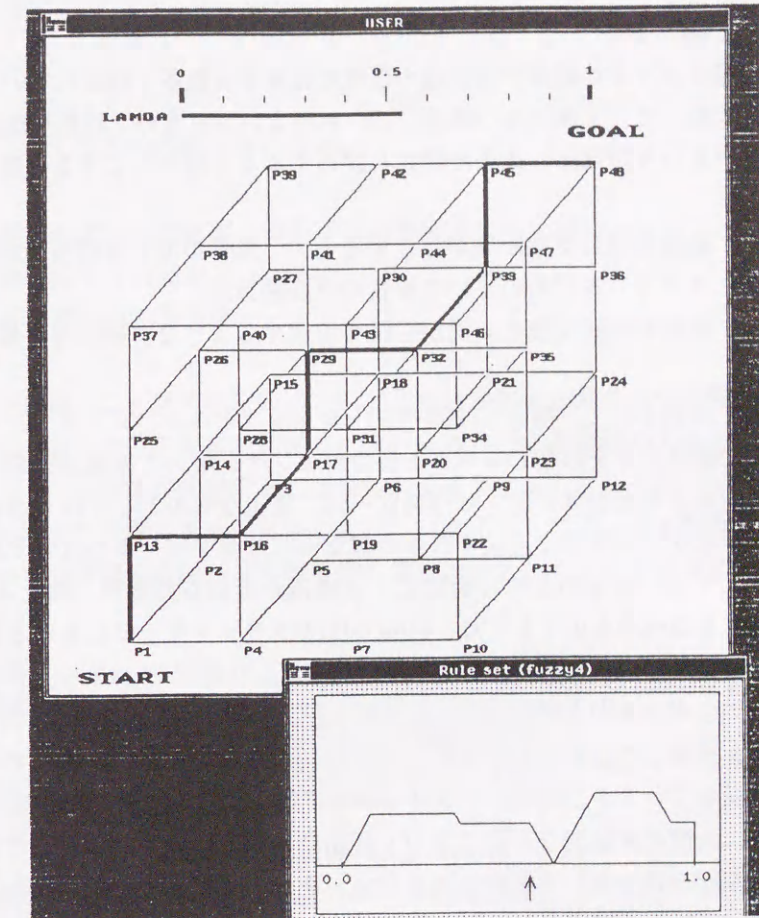


図4.1.7 シミュレーションの実行画面

4.2 あいまい評価に基づく知的経路計画手法

4.2.1 緒言

移動ロボットの経路計画問題の研究は従来から数多く行われているが、地図などの環境情報や計画された経路に関して、人間の持つ概念に近いあいまいさを含めて検討を加えたものは少ない。特にあいまいさの取り扱いという観点から従来の研究を眺めてみると以下のような問題を取り扱った研究が比較的少ない。

- 1) 地図情報に不確かさが存在するという条件のもとで行った経路計画
- 2) サブゴールにあいまいさを含めた経路計画
- 3) 経路の評価関数に人間が言語で表現するような主観的な評価を対象とした経路計画

1) の例として、油田ら[1]はベクトルマップとセル(グリッド)マップの両方を用いてロボットが自主的に未知環境の中を移動しロボット自身のセンサを用いて環境地図を作成していく方法を提案しているが、地図に記載された物体に関する情報は一般に確実なものとして取り扱われている。2) では、通常得られた経路はロボットが厳密に追従すべき曲線ないしはサブゴールの点列である場合が一般的である。3) の経路計画としては、従来のものは数式による表現が可能な経路長、消費エネルギーなどの物理量がほとんどである。これ以外のものとしては、Elfesら[2]のグリッドマップにおける各格子点に障害物の存在する確率などをもちこたせることにより不確実な環境情報を表現した研究やLevittら[3]の目印となる特徴物をランドマークとしこれらのロボットからの見え方を定性的に記述することにより人間の目標到達に近いナビゲーションアルゴリズムを提案した研究などがある。一方、王[4]らはファジィ推論をマニピュレータの軌道計画に応用した研究を発表しているが、単純にアクチュエータ制御にファジィ推論を適用した数値処理的手法である。以上のように、人間の主観的な評価に基づく経路計画の研究はまだほとんどなされていないのが現状である。

人間が地図を用いて経路計画を行い、それに基づいて移動する場合を考えると、少なくとも地図を図形的に正確な情報源として取り扱ってはならず、計画された経路も厳密に追従すべきものとして取り扱ってはならない。これは人間の環境認識能力の高さと移動時の動的な計画能力の高さによるところが大きいと考えられる。また経路計画の結果得られた経路も、未知の環境や障害物の存在を前提とするならば、ロボットが厳密に追従すべきものとしたのでは実際の走行に際して過度の拘束条件を与えることになる。さらに、経路計画の際の評価関数に関しても、最短距離経路や最小消費エネルギー経路などの物理量を最適にする経路が要求される場合は少なく、実際はそれらの物理量を属性として持つ抽象的な評価を用いる場合が多い。

本研究では人間が行っているような、抽象的な評価がある許容範囲内に入る経路を見出す過程をファジィ手法を用いて模擬し、この手法のアルゴリズムを示すことを目的としている[5]。すなわち、数値的な情報に基づく経路計画の規範をファジィ手法を用いてシンボルで表現された知識として記述することにより、数理計画的な手法では導出することが困難な解を導き出すことが可能になると考えられる。本論文では、提案した経路計画手法のアルゴリズムについて述べるとともに、有効性検証のため行なったシミュレーション結果についても報告する。

4.2.2 経路計画におけるあいまい性

一般に経路計画問題には以下の3つのあいまい性が含まれている。本方式では、これらのあいまい性をすべて考慮した手法の提案を行なう。以下ではこれらのあいまい性について本研究における考え方を述べる。

a) 地図情報の不確かさ

- b) 評価関数のあいまいさ
- c) 計画された経路のあいまいさ

(a) 地図情報の不確かさ

地図情報の不確かさには以下のようないくつかの種類が存在する。

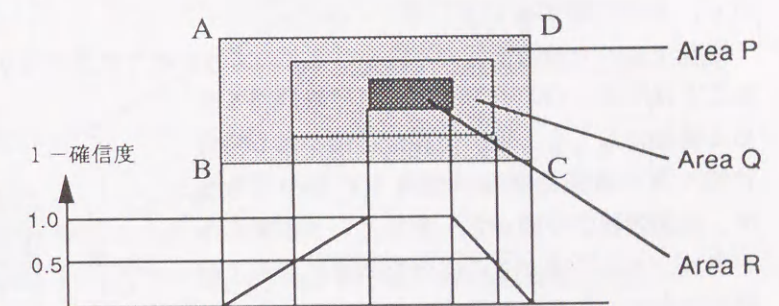
- 1) 物体が存在することには確信があるが、その位置については確信が持てない場合
- 2) 物体が存在すること自体に確信が持てない場合
- 3) 未知の環境や障害物の存在する可能性がある場合

このうち3) は経路計画の段階では取り扱うことはできない。1) と2) は地図に記載された情報に対する確信度が1でない場合であるが、両者は表している意味が異なる。1) は物体の存在領域の大きさがあいまいであることを表しているのに対し、2) は物体が存在すること自体が不確かであることを表している。言い替えると1) はタイプ1のファジィ集合で表現できるのに対し、2) はタイプ2のファジィ集合でないことと表現できないものである。本研究では地図情報の不確かさを取り扱う第一段階として1) を対象とする。

確信度と拘束の強さの間には負の相関がある。確信度が高いということは拘束が緩いということを表し、逆に確信度が低いということは拘束が厳しいということを表す。これを経路計画に用いる地図に関していえば、ある点の存在する可能性のある領域の面積の大小の差となって現れる。すなわち、確信度の高い点の近傍には大きな領域が現れ、確信度の小さな点の近傍には小さな領域が現れる。

また、点ではなく物体の場合には、それぞれの点の位置に対する確信度が異なっても、物体の幾何学的形状から定まる相互の位置関係の拘束が存在するために、異なる確信度に対するそれぞれの形状は相似形になると考えられ、たとえば図1のような領域が地図上の物体の周囲に現れる。この図ではそれぞれの領域は次のようなものである。

- ・領域P(確信度1.0の領域)：この領域内に存在することは絶対に間違いない。
- ・領域Q(確信度0.5の領域)：たぶんこの領域内に存在すると思われる。



四角形ABCD
P: 確信度=1.0
Q: 確信度=0.5
R: 確信度=0.2

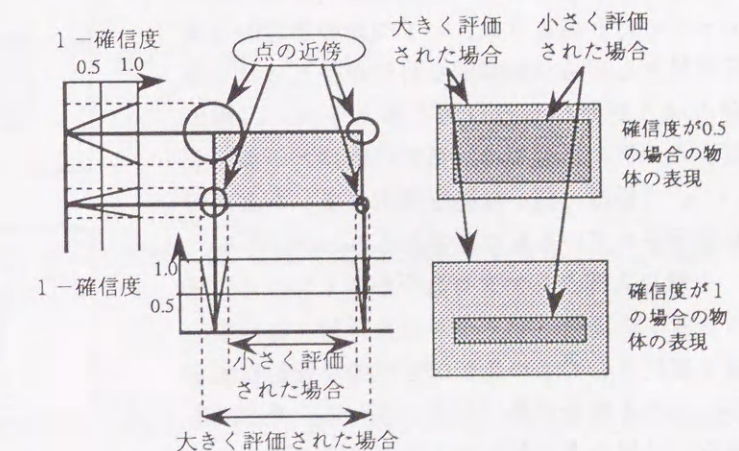


図4.2.1 位置情報があいまいな地図

4.2 あいまい評価に基づく知的経路計画手法

・領域R（確信度0.2の領域）：あえて限定するならばこの領域の内部に存在する。

この時、地図上の領域の包含関係は $P > Q > R$ となる。また、絶対に確実な情報（確信度が1の地図）では $P = Q = R$ となる。図4.2.1において、物体の各頂点におかれた円は頂点その時の確信度の値で存在するといえる領域を表している。この領域の大きさは確信度の値が大きくなるにつれて大きくなる。すなわち、ある1つの確信度において、物体の各頂点はその存在する領域の大きさが変化するため、それらで代表される物体は実際よりも大きく評価される場合と、小さく評価される場合の2通りの評価を受ける。

もしも物体の大きい側の評価を経路計画に用いるならば、物体との干渉の可能性のより小さな経路を得ることができるであろうし、逆に小さい側の評価を経路計画に用いるならば、より長さの短い経路を得ることができるであろう。移動ロボットにおける経路計画問題を考えると、ロボットが周囲の環境や物体と干渉しないという条件は、いかなる目的や評価関数が与えられても常に満たすべき条件であると考えられる。そこで、本研究では物体の大きさは与えられた確信度において、大きい側の評価を用いることにする。

尚、図4.2.2に示すように異なる確信度で地図を描いた場合、経路の数が新たに増える場合が考えられる。よって、ある確信度で確定した地図を用いて経路計画を行い、その後確信度を変化させて同様の処理を行う事も重要になると考えられる。

(b) 評価関数のあいまいさ

前節で述べた地図情報の不確かさをどのように取り扱うべきかは実際には評価量が与えられないと決定することはできない。たとえば最短距離経路を求める場合にしても、地図情報が正確である場合に限り真の最短距離経路が得られるのであって、地図情報が不確かさを含むという前提のもとでは、その不確かさの度合を考慮に入れて経路を決定する必要がある。

また、本研究で対象としているような人間を模擬したロボットの場合、最短距離経路などの物理的な量を最適にする経路が要求される場合よりも、「これを急いで部屋の向こう側の端にもっていけ」のように、いくつかの物理的な量を属性としても抽象的な目的が与えられる場合のほうがむしろ自然と考えられる。例えば、この例の場合では「室内の物体と干渉しない」、「距離の短い経路を選択する」などが暗に指示されているものと考えられることができる。

一般に人間からマクロ命令が与えられる知能ロボットの場合、いくつかの物理的なあいまい量を属性としても比較的定性的な行動目的が与えられる場合が多いと考えられる。そこで本研究では異なる目的による計画経路の違いを比較するため、「注意して走行する」と「急いで

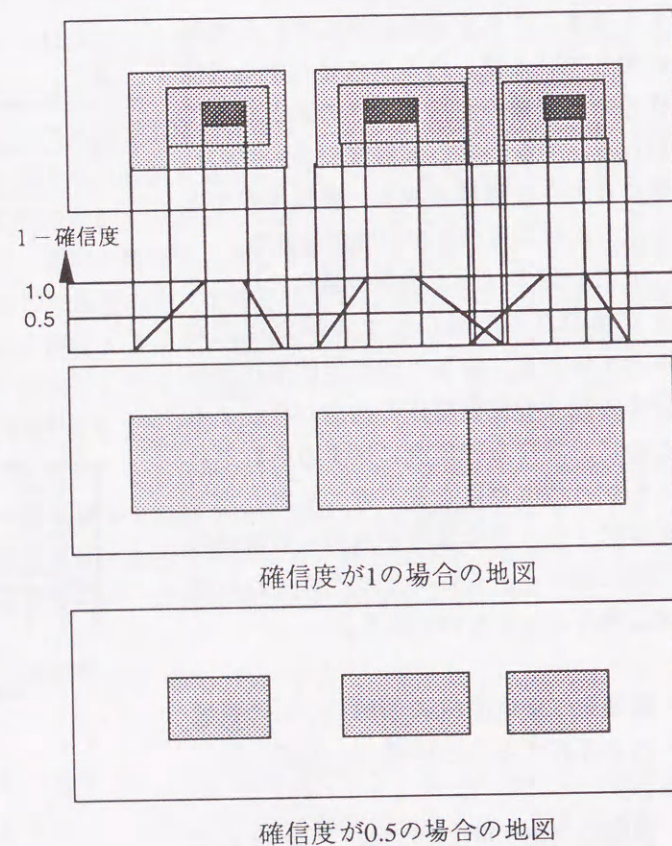


図4.2.2 確信度による地図の相違

4.2 あいまい評価に基づく知的経路計画手法

走行する」を例として取り上げ経路評価を行なってみた。移動ロボットに対する指示として、例えば「注意して」と「急いで」の場合、次のような経路選択が適当であると考えられる。

- 「注意して」
 - ・できるだけ危険度の小さな経路を選ぶ。
 - ・できるだけ確信度の高い地図を用いて経路を決定する。
- 「急いで」
 - ・できるだけ短い経路を選ぶ。
 - ・できるだけ速度を大きく保てるように経路を選ぶ。

上にあげた行動目的には「危険度」、「地図の確信度」、「経路の長さ」、「方向転換の数」、「方向転換の大きさ」等の因子が含まれている。

(c) 計画された経路のあいまいさ

本研究ではロボットの自律的な障害物回避機能を利用するため、サブゴールを「あいまい」に表現し、ある程度の許容範囲を包含した形で目標値をロボットに与えるものとする。図4.2.3に示すように「あいまい」サブゴールとは地図上の点で、その近傍にその点への近さを決定するための所属度メンバーシップ関数が定義されている点を表わす。センサ情報から得られた現在位置の情報はセンサの精度に依存するファジィ数と考え、このファジィ数とサブゴールの所属度メンバーシップ関数とのマッチングの度合をもってサブゴールに到達したかどうかの判断基準とする。

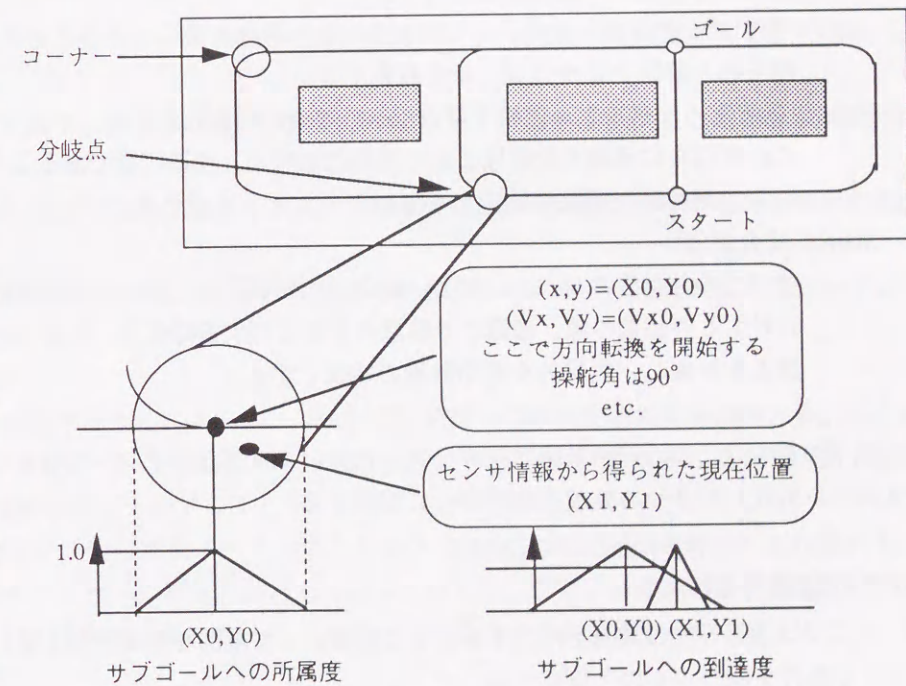


図4.2.3 あいまいなサブゴール

4.2.3 経路計画アルゴリズム

ここでは、本研究で提案する経路計画手法について説明する。前節で述べたあいまい性をすべて考慮した知的経路計画アルゴリズムを以下の手順で実行する。

Step1：あいまいな地図の作成

地図上に物体を置き、それぞれの物体に対し図4.2.1のように確信度のレベルに対応した存在領域を定義する。地図上の物体の表現は多角形の内部と辺上とする。

4.2 あいまい評価に基づく知的経路計画手法

- Step2: 確信度による地図のスライス
経路計画の目的に応じて適切と判断される確信度を設定し、確信度に応じた大きさの領域を地図上に表示する。
- Step3: 地図の領域分割
探索のために通過可能領域を分割する。
- Step4: 位相情報の抽出
分割された通過可能領域に対し、それらの接続関係を表現するリストを作成する。
- Step5: 通過可能領域列の抽出
リストを探索し、スタート地点からゴール地点に至る通過可能領域の列をすべて求め、これを詳細探索領域とする。
- Step6: 経路生成
詳細探索領域内で経路生成を行う。
- Step7: サブゴール生成
得られた経路上にサブゴールを設定する。
- Step8: 確信度のレベルを一段階下げ、Step1～Step7を繰り返す。
この時新たに通過可能領域として現れた領域は、通過可能であることに対する確信度が低い領域である。最終的な経路を決定する際にこのことを考慮する。
- Step9: 総合評価
すべての確信度のレベルに対し、経路計画が終了した後、全経路を再評価する。すなわち、各経路に対する評価量の値、通過する領域の存在に対する確信度、生成されたサブゴールの制約としての強さを評価し、最終的な経路計画の結果とする。

以下の節では、この経路計画アルゴリズムにおいて重要なポイントとなる「領域分割」「経路評価」「サブゴール生成」の3つの処理内容を中心に説明する。

4.2.4 領域分割

ここでは通過可能領域列を抽出するための準備として地図を領域分割する。このときに用いる地図は次のような条件を満たすものと仮定する。

- (1) 物体の形状はすべて多角形で表現する。
- (2) 物体の表現として確信度最大（確信度=1）と確信度最小（確信度>0）の多角形領域を与える。
- (3) 中間の確信度の領域は線形補間により求めることができる。

経路計画に用いられている代表的な地図の領域分割方法には以下のようなものがある。

- ・V-グラフ法 [6]
- ・接線グラフ法 [7]
- ・ボロノイグラフ法 [8]
- ・特定方向にのみ接線をひく方法 [9]
- ・もっとも近い障害物の頂点、環境境界に向かって直線をひく方法 [10]

4.2 あいまい評価に基づく知的経路計画手法

V-グラフ法、接線グラフ法は地図を分割した境界線がそのまま経路の一部となるもので、経路長を評価量とした場合には特に有効であると思われる。これに対し、ボロノイグラフ法、特定方向にのみ接線をひく方法は出発点から目標点に至る自由領域の列を発見してから詳細な経路計画を行う方法である。この場合、経路上のサブゴールは地図を分割した境界線上に得られる。

本研究では経路長の他にロボットの特性や危険度などを含めて評価することを考えているため、分割境界上にサブゴールを発生させる方法であるMAKLINK法[10]を採用する。この方法はもともと高速経路計画手法として発表されたものである。本研究でも処理の高速性に着目し、この方法を採用した。

MAKLINK法の地図の分割アルゴリズムを以下に示す。前提として、移動環境の境界および環境内に存在する障害物は2次元凸多角形で表されるものと仮定する。このとき以下の手順に従って地図の分割および分割された領域どうしの隣接関係を記述したグラフを作ることができる。

- 1) 障害物の各頂点から他の頂点あるいは環境境界の最短点をリンクの接続点の候補として選び、近い点からリストを作成する。
- 2) リストの先頭から候補を選び、障害物の境界と交差しないかをチェックし、交差した場合には次の候補を選ぶ。
- 3) リンクによって分割された自由空間の内角が π 以下である場合にはその頂点からのリンクの生成を終える。
- 4) リンクによって分割された自由空間の内角が π を越える場合、さらに次の接続点の候補について2)以下の手順を繰り返す。

すべての頂点からリンクを生成することによって、自由空間は複数の凸多角形領域に分割される。各凸多角形領域は、障害物境界あるいはリンクで囲まれており、すべての内角が π 以下であることから、同一領域内のあらゆるリンク上の2点は障害物と干渉することなく結ぶことができる。

次に図4.2.4に示したようなリンクの中点をノードとするグラフ（MAKLINK-Graphと呼ぶ）を作成する。文献[10]ではMAKLINK-Graphをロボットの移動経路として扱っているが、ここでは領域の接続関係を記述したものと扱う。この時、MAKLINK-Graphのノードとエッジは次のような意味をもつと考える。

[ノード]

- ・地図上においては領域境界上の1点
- ・ロボットの操作上においては方向転換（属性）方向転換が「急である」
「緩やかである」

[エッジ]

- ・地図上においては分割された領域の1つ
- ・地図上においてはノード間をむすぶ最短距離経路
- ・ロボットの操作上においては領域内部の通過

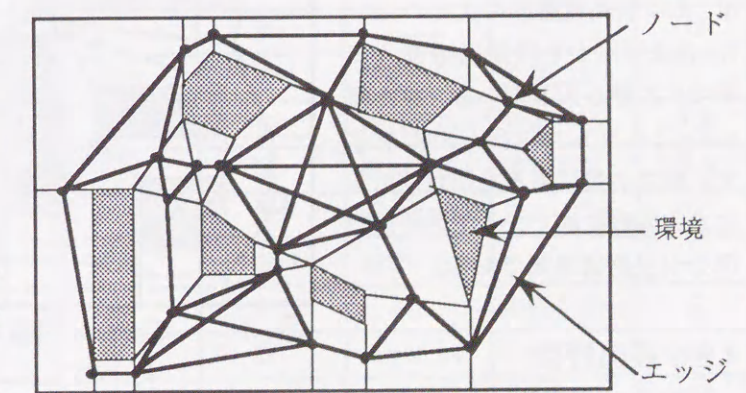


図4.2.4 MAKLINK-Graphを用いた経路生成の例

(属性) 領域が「狭い」「広い」

最短距離経路が環境に「接近している」「適当に離れている」「離れ過ぎている」

さらにこれらの情報をもとに、ロボットの移動にとってより本質的と考えられる2つの部分「狭路通過部」と「方向転換部」を抽出することができる。以下にこれらの定義を示す(図4.2.5参照)。

<狭路通過部>

向かい合う2辺が環境の境界または物体の辺で構成される4辺形領域。ロボットと物体との干渉はこの領域の内部でのみおこる。上で述べた「領域が広い」あるいは「狭い」という属性はこの狭路通過部の属性に置き換えることができる。

<方向転換部>

環境あるいは物体の頂点の近傍で、その頂点をはさむ2つの辺がともに「狭路通過部」に含まれる部分。上でのべた「方向転換が急である」あるいは「緩やかである」という属性はこの方向転換部の属性に置き換えることができる。

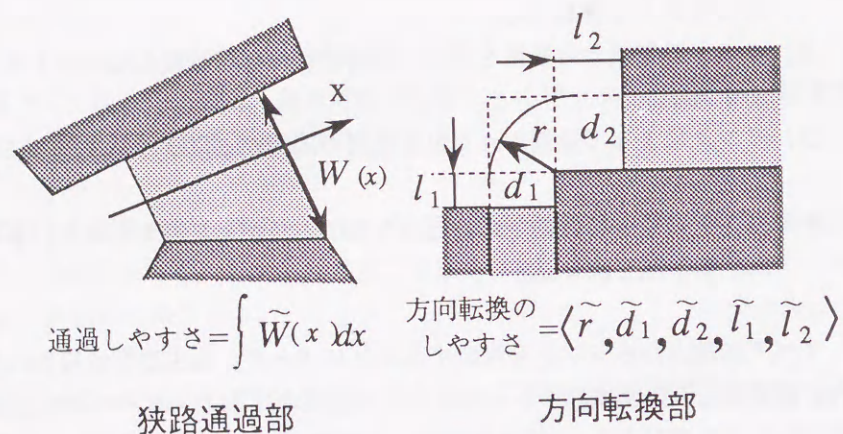


図4.2.5 狭路通過部と方向転換部の定義

この「狭路通過部」と「方向転換部」は、図4.2.5に示したように幾何学的な形状とロボットの特性を用いることにより、「通過のしやすさ」および「方向転換のしやすさ」を定義することが可能である。ここでwは狭路通過部の幅、rはロボットの最小回転半径、d1、d2は狭路通過部の入口の幅、l1、l2は頂点から狭路通過部入口までの距離を示している。また図4.2.6にこれらを抽出した例を示す。物体の角にある丸印が抽出された方向転換部で、うすい灰色の部分に狭路通過部である。

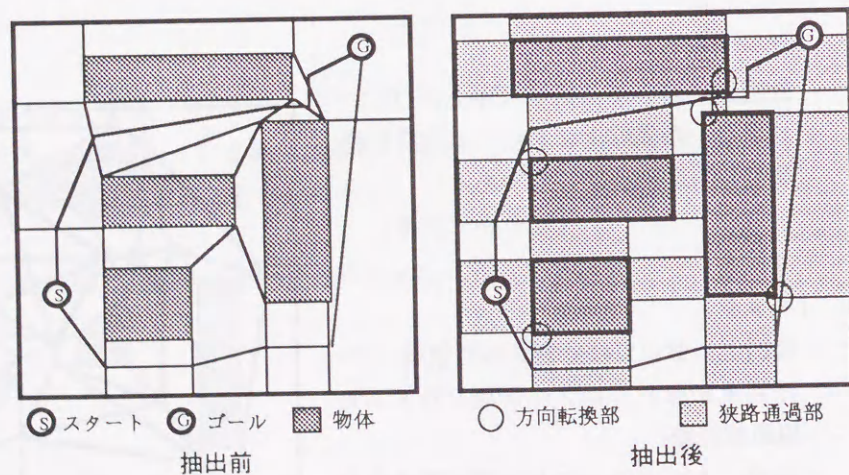


図4.2.6 抽出された狭路通過部と方向転換部

4.2.5 経路評価

本経路計画手法では、以下の点を基本前提として経路評価を行

なっている。

- ・急な方向転換は好ましくない。
- ・狭い領域を通過することはできるだけ避けた方がよい。
- ・環境に過度に接近することは好ましくない。
- ・環境から過度に離れた経路は冗長である。

まず、抽出された狭路通過部と方向転換部の経路評価を行なうために、評価量のそれぞれの因子ごとに設定されたメンバーシップ関数と実際の値との適合度を評価する。それぞれのメンバーシップ関数はロボットの代表長さなどの特性値で正規化された台集合の上で記述されているものとする。

通過可能領域列の抽出は、ノードグラフ上でスタート地点を含むエッジからゴール地点を含むエッジを探索することにより達成される。そのとき探索結果に含まれる狭路通過部の「通過のしやすさ」と方向転換部の「方向転換のしやすさ」および「おおよその経路の長さ」を評価し、探索結果に順位づけを行なう。

まず、狭路通過部と方向転換部をそれぞれ独立に評価する。狭路通過部に対しては図4.2.5に示した式を用いて「通過のしやすさ」を評価する。方向転換部はその定義により、2つの狭路通過部と1つの方向転換で構成されているため、「狭路通過」-「方向転換」-「狭路通過」の組合せとして評価すると考える。方向転換部に関しては表4.2.1に示す評価規則を用いる。すなわち、まず方向転換部の幾何学的特性量とメンバーシップ関数との適合度を求め、次にその適合度を用いて表4.2.1の評価規則により方向転換部の「方向転換のしやすさ」を評価する。ここでは概略の評価値を得る目的であるため、この表の評価値には大まかな数値を用いている。

次に、個別の評価を総合評価にまとめるための重み係数を決定する。この重み係数は経路計画に用いた地図の確信度に応じて変化させる。これにより、用いた情報のあいまいさに応じて経路計画の方法あるいは評価の方法を修正できるようになる。例えば、得られた評価結果に確信度の値を乗ずることにより、評価結果を修正する方法を用いた場合、地図の確信度が低くなるにつれ評価結果も低くなることになる。

さらにここでは、評価結果の有意差に対するしきい値を考える。これはここで得られた評価が大まかなものであり、実際に移動する際にロボットに最大の自由度を与えとともに、別の経路の選択を可能にするために複数の領域列を

表4.2.1 方向転換部の評価

r	d1	d2	l1	l2	評価
適当に広い	—	—	—	—	0
非常に広い	—	—	—	—	0
狭い	狭い	—	—	—	0
狭い	非常に広い	狭い	—	—	0
狭い	狭い	適当に広い	—	—	0
狭い	適当に広い	狭い	—	—	0
狭い	狭い	狭い	非常に広い	非常に広い	1
狭い	狭い	狭い	適当に広い	非常に広い	1.5
狭い	狭い	狭い	非常に広い	適当に広い	1.5
狭い	狭い	狭い	適当に広い	適当に広い	2
狭い	狭い	狭い	非常に広い	狭い	2
狭い	狭い	狭い	狭い	非常に広い	2
狭い	狭い	狭い	適当に広い	狭い	2.5
狭い	狭い	狭い	狭い	適当に広い	2.5
狭い	狭い	狭い	狭い	狭い	3

抽出したいためである。

以上の準備をした上で、MAKLINK-Graphの探索を行い、抽出された領域列を評価し、実際の経路計画を行う領域列に対して順位づけをおこなう。順位づけの際の探索はプログラムがコンパクトであることとアルゴリズムの簡単なことから深さ優先の縦型探索を行うことにした。この手順を図4.2.7に示す。

4.2.6 サブゴール生成

前節で順位づけられたスタートからゴールに至る領域列に対し、具体的な経路生成を行う。この段階は通常の移動ロボットの経路計画と同じ問題である。しかも、すでに実際の経路を生成するための領域列が抽出されているため、問題は単連結な空間の内部に環境や物体と干渉しないという条件のもとで、できるだけ経路長が短く、方向転換の数が少なく、ロボットにとって操舵可能な範囲に収まる経路を生成するという問題に帰着する。

この段階ではロボット自身の環境認識能力を考慮に入れる必要がある。ここではロボットは、環境あるいは物体と自分自身との距離を計測でき、かつ与えられた目標地点に向かいながら障害物を回避できるものと仮定している。

このとき以下に示す方針で実際の経路を生成することにする。

- 1) できるだけ少ないサブゴールを生成する。
サブゴールはある意味ではロボットの移動にとっては拘束条件とみなされ、多くのサブゴールを与えることはロボットの障害物回避の際の自由度を奪ってしまうことになる。
- 2) サブゴールはロボットが操舵を必要とする箇所に生成する。
実際のロボットの経路は線分と円弧で構成さ

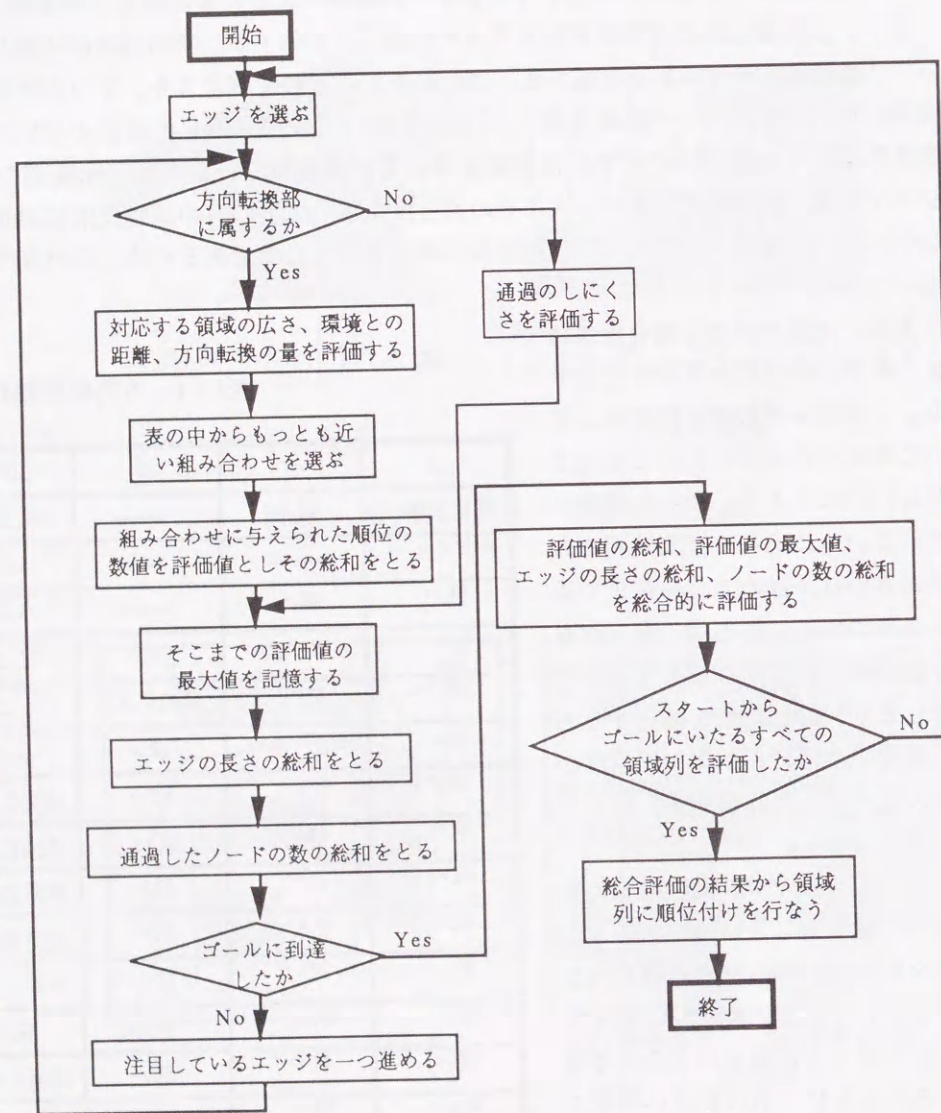


図4.2.7 領域列評価の手順

れ、その場合には操舵を行う位置とその大きさが経路を一意的に決定する。

3) 経路生成には幾何学的な条件に基づいて2つの方法を併用する

すなわち、発見的な方法と最適経路法である。いわゆる「巧みな」走行が必要とされるのは狭い領域から出てくる場合と狭い領域に入ろうとする場合に限られ、それ以外の場合では経路の差がほとんど生じない。このような場合には最短距離経路が与えられた目的にもっとも適した経路となる。

そこで本研究では、まず「狭い」と評価された領域に対してあらかじめ知識として与えられた走行パターン（経路パターン）をマッチングすることによって経路を生成し、残りの領域に対しては最短距離経路を生成するという2段階で実際の経路を生成することにする。「狭い」と評価される領域はさらに2つに分類することができる。すなわち、方向転換部に属さない場合と属する場合である。前者では経路は単純に狭路通過部の中線として生成され、中線の両端の点がサブゴールとなる。ここではロボットに位置および方向の自由度がほとんど許されないため、サブゴールはこれらを限定した拘束の厳しいサブゴールとなる。

後者の場合、特に図4.2.8のパターンaに示すように、方向転換部に属する「狭路通過部」のいずれか一方が狭くない場合、経路は方向転換部の近傍でなめらかに接続されれば十分である。従って、「巧みな」走行あるいは経路が必要とされるのは図4.2.8のパターンbからdに示すように、方向転換部を構成する2つの狭路通過部がともに「狭い」場合のみである。この場合には方向転換部の幾何学的形状に基づいて、あらかじめ与えた方向転換パターンをマッチングさせる。

経路計画の最終的な出力はロボットに対するサブゴールの列であって経路全体ではない。そこで、本研究では、方向転換のパターンとしては適合度が最大のものを用いることとし、中間的な適合度のものに対しては実際の経路を生成する場合に適合度を考慮してパラメータを修正して用いることにする。方向転換部に対してある方向転換パターンが決定されたのち、実際の経路生成を行う。経路の生成は決定された方向転換パターンの特徴量である円弧部の半径および、円弧どうし、あるいは円弧と線分の接触位置を変化させることで行う。

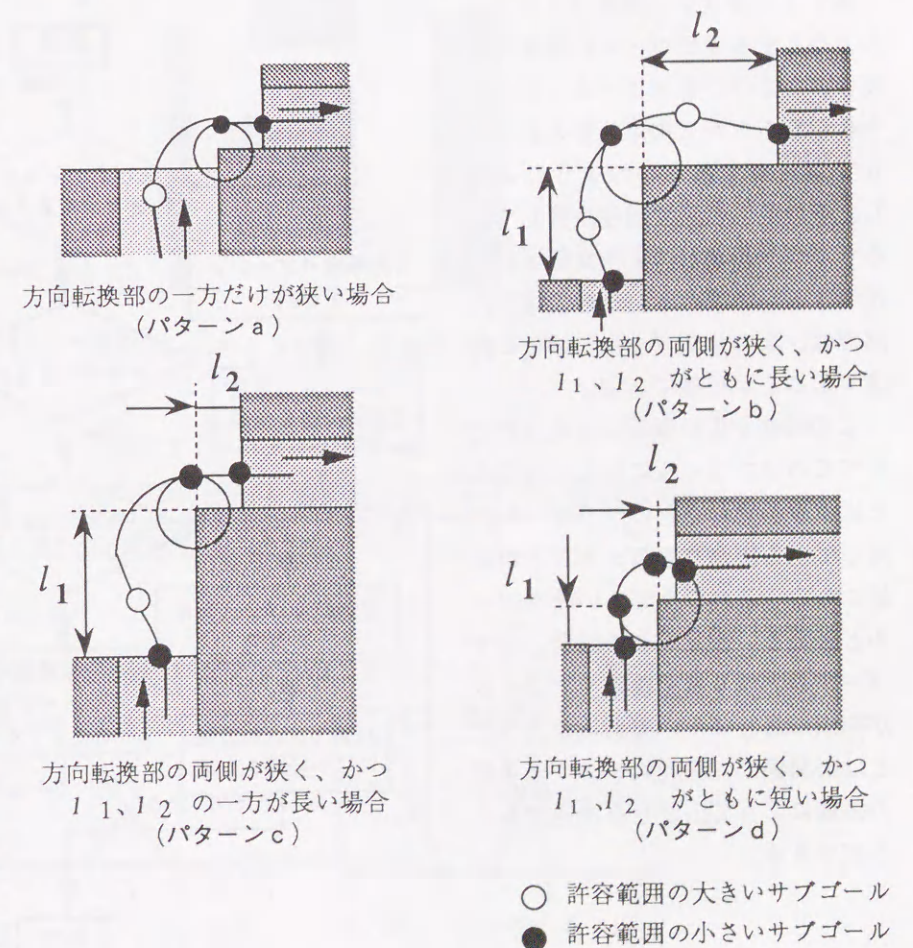


図4.2.8 方向転換のパターン

4.2 あいまい評価に基づく知的経路計画手法

以上で述べた経路およびサブゴール生成の手順を図4.2.9および図4.2.10に示す。

ここで生成されたサブゴールのうち、「狭い」領域に生成されたサブゴールはその領域を通過するためにロボットの位置および方向が満たすべき条件を記述したものである。これらの条件が満たされない場合には、その領域を通過することが不可能になるため、このようなサブゴールをファジィ化（あいまいサブゴール）することはできない。あいまいサブゴールは基本的に「広い」領域に生成されたサブゴールを対象とする。

あいまいサブゴールはファジィ化しようとするサブゴールの前後に生成された2つのサブゴールを考え、それらをあたかも拘束条件の厳しいサブゴールであるかのようにみなし、その条件の下で現在注目しているサブゴールがどのような領域に存在できるかを求める。この領域の内部では、次のサブゴールに厳密に到達することが可能である。

この操作を広い領域に生成されたすべてのサブゴールに対して行うことにより、それぞれのサブゴールに対してある広がりを与えることが可能になる。これをあいまいサブゴールと考える。例えばこの場合、サブゴールの広がりには前後のサブゴールが拘束の厳しいものであるかどうかとは無関係に、単純にサブゴール間の距離によって広がりを見決定することができる。

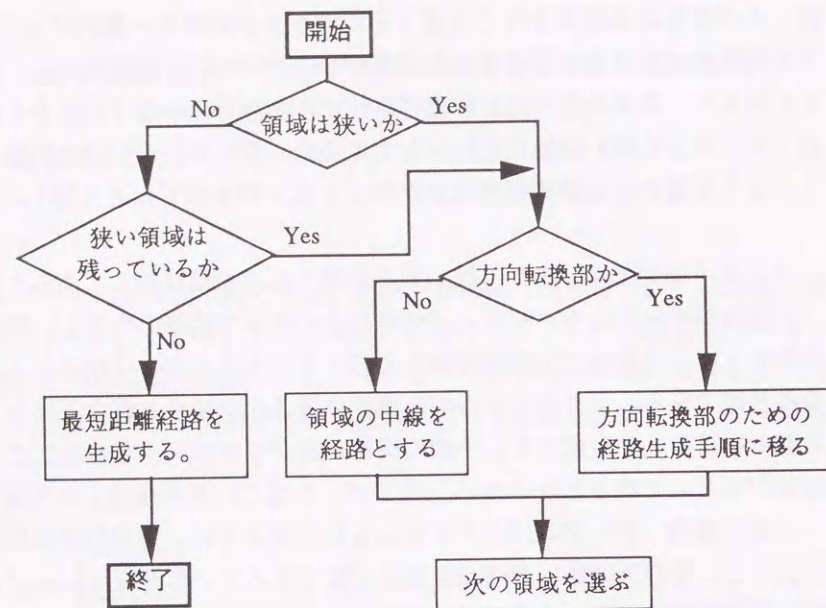


図4.2.9 サブゴール生成の手順

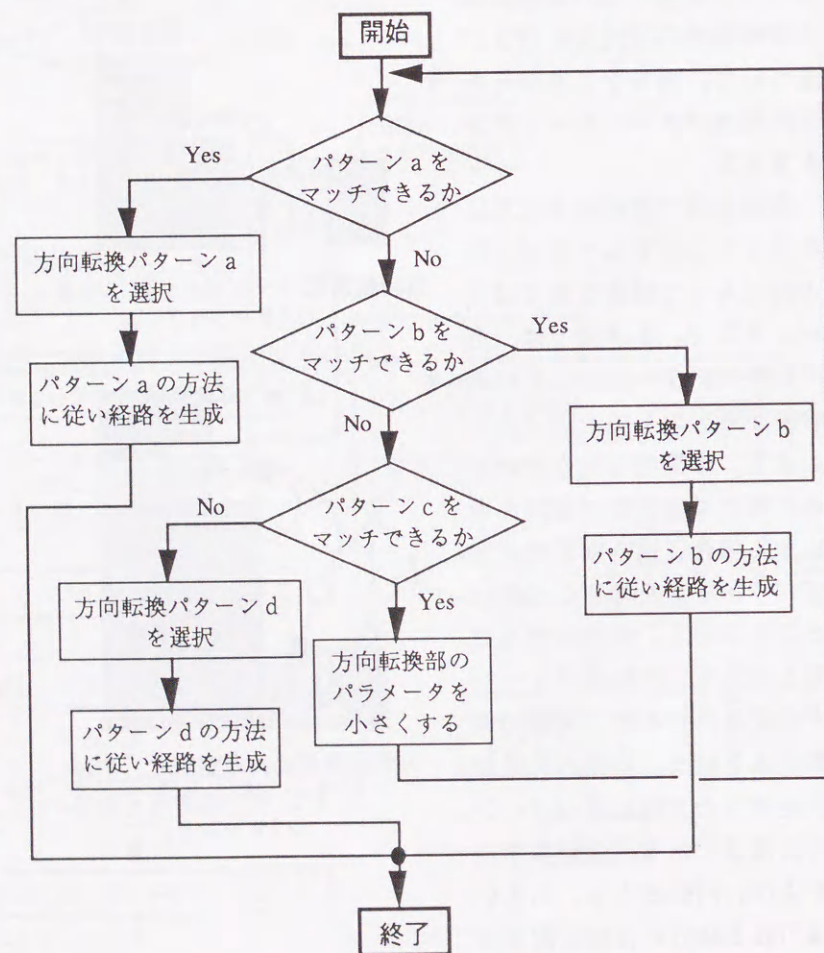


図4.2.10 方向転換部におけるサブゴール生成手順

4.2 あいまい評価に基づく知的経路計画手法

4.2.7 有効性検証シミュレーション

ここでは、前項で述べた手法に基づいて作成したプログラムを用いて、本手法の有効性を示すために行った計算機シミュレーションについて報告する。ここでは主に経路生成の処理アルゴリズムの妥当性を確認する目的でシミュレーションを行った。よって、地図の確信度は1であると仮定し、サブゴールに対する広がり設定は本シミュレーションでは扱わなかった。

図4.2.11にこのシミュレーションで用いた地図（ケース1）を示す。この図において白丸がスタート、黒丸がゴールを示している。また、表4.2.2にロボットの特性として用いる値、図4.2.12にメンバーシップ関数、表4.2.3、表4.2.4に総合評価のための重み係数を示す。すでに述べたように、メンバーシップ関数はロボットの特性値で正規化されている。移動に対しては「注意して走行する」および「急いで走行する」の2つの目的を与えた。

これらをもとに、「注意して走行する」という目的を与えた場合に生成されたサブゴールの例を図4.2.13に示す。図中の（a）がもっともよい評価をうけた経路であり、以下（b）、（c）、（d）の順に評価が

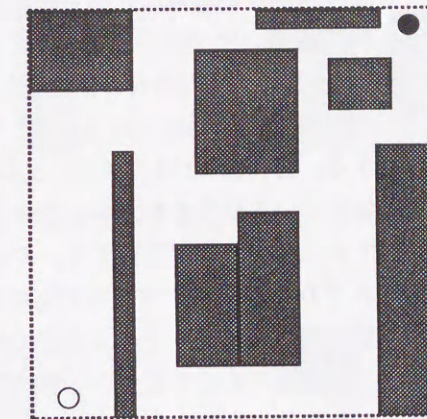


図4.2.11 シミュレーションで用いた地図（ケース1）

表4.2.2 ロボットの特性値

ロボットの幅	ロボットの長さ	最大操舵角(deg)	安全操舵角(deg)
10.00	10.00	45.00	20.00

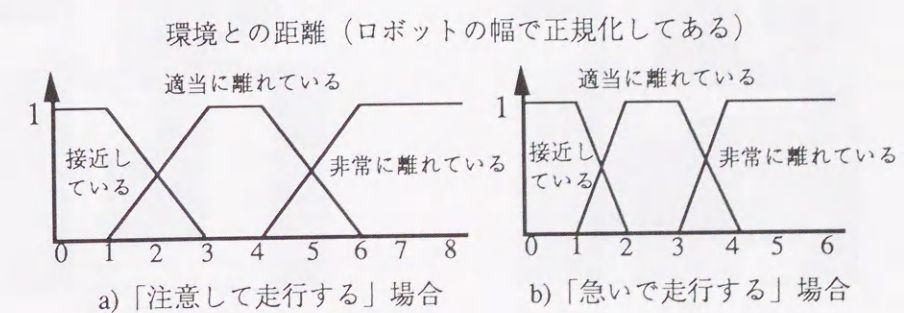


図4.2.12 メンバーシップ関数の例

4.2 あいまい評価に基づく知的経路計画手法

表4.2.3 「注意して走行する」場合の重み係数

「狭い」に対する最大適合度	狭路通過部の通過しにくさ	方向転換しにくいに対する最大適合度	方向転換のしにくさ	経路の長さ	方向転換の数
100.00	100.00	0.00	100.00	1.00	50.00

表4.2.4 「急いで走行する」場合の重み係数

「狭い」に対する最大適合度	狭路通過部の通過しにくさ	方向転換しにくいに対する最大適合度	方向転換のしにくさ	経路の長さ	方向転換の数
100.00	10.00	0.00	100.00	10.00	50.00

悪くなる。図中、うすい灰色で示したエリアが狭路通過部で、環境のコーナーにかかれた丸印は方向転換部の抽出された部分を示す。また、大きな矢印は狭い領域に生成された拘束の厳しいサブゴールを表し、小さい矢印はそれ以外の領域に生成された拘束の緩いサブゴールを表している。

図4.2.14は与える目的を「急いで走行する」に変えた場合に生成されたサブゴールの例である。図4.2.13と図4.2.14を比較することにより、「急いで走行する」場合には「注意して走行する」場合に比べ、明らかに長さの短い経路を選択していることがわかる。特に図4.2.13の(d)と図4.2.14の(a)からわかるように、与えられた目的に応じて経路の評価が逆転している様子が見られる。このように、与えられた目的によって評価値が変化し、その結果「注意して走行する」、「急いで走行する」の2つの目的に対して経路が適切に評価され順位づけられていること、またそれぞれに対しサブゴールが適切に生成されていることがわかる。

図4.2.15および図4.2.16は異なった地図を用いた場合(ケース2)の例である。この場合には先の例ほど際立った差は見られないが、この場合も「注意して走行する」、「急いで走行する」の2つの目的に対して適切に評価され順位づけられていることがわかる。また、それぞれに対しサブゴールが適切に生成されていることがわかる。

以上の2つの例題により、本研究で提案した手法はロボットの与えられた目的、「注意して走行する」、「急いで走行する」の2つに対し、与えられた地図内の領域を適切に評価し、経路生成およびサブゴール生成を行えることを確認した。

4.2 あいまい評価に基づく知的経路計画手法

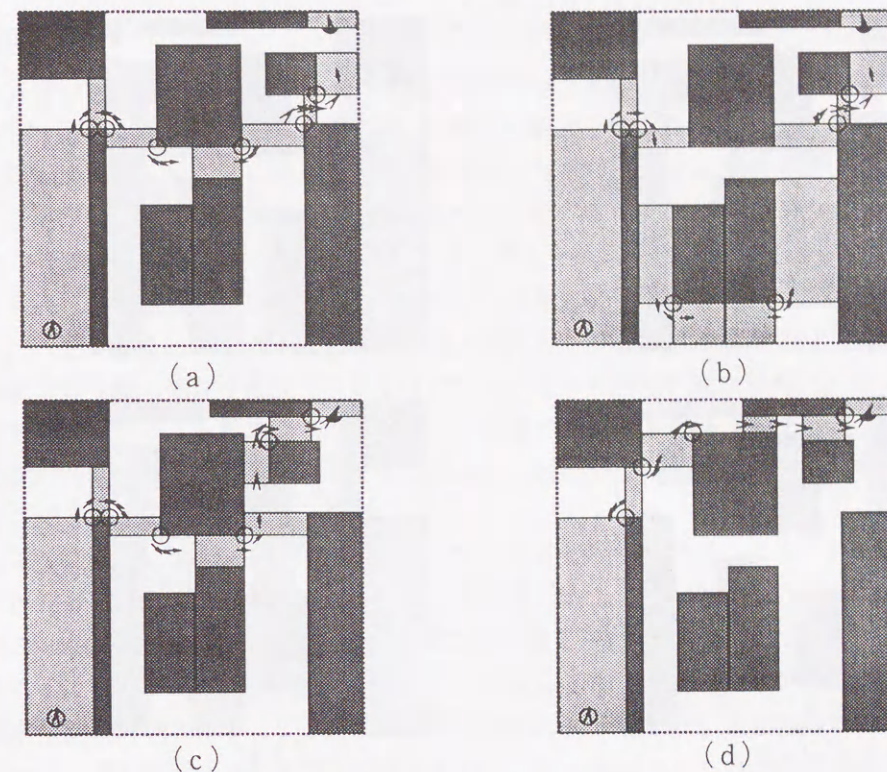


図4.2.13 「注意して走行する」場合の生成サブゴール(ケース1)

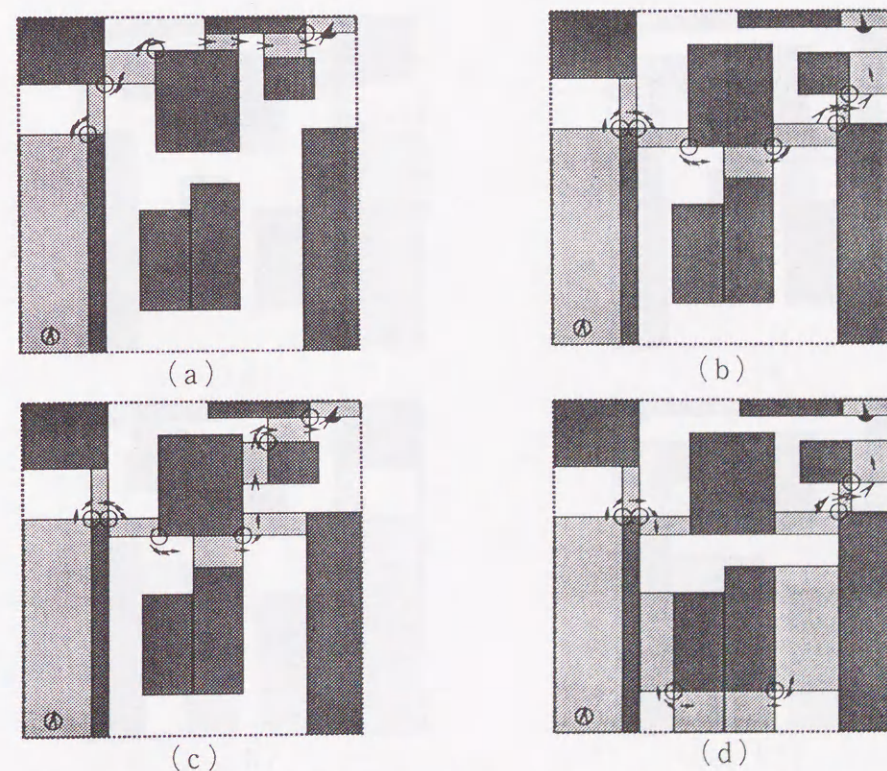


図4.2.14 「急いで走行する」場合の生成サブゴール(ケース1)

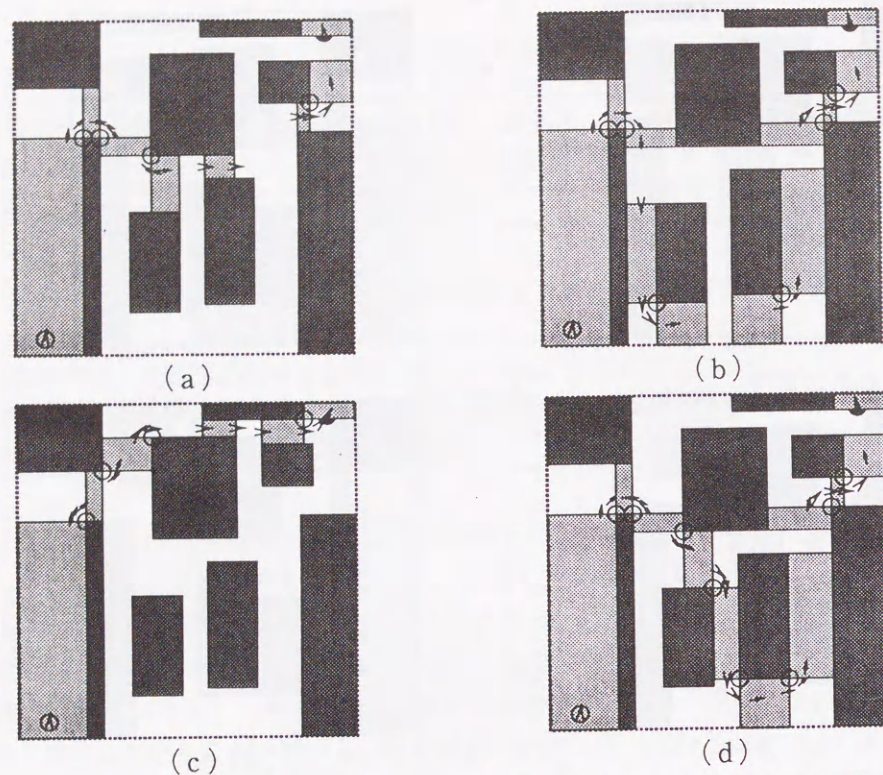


図4.2.15 「注意して走行する」場合の生成サブゴール (ケース2)

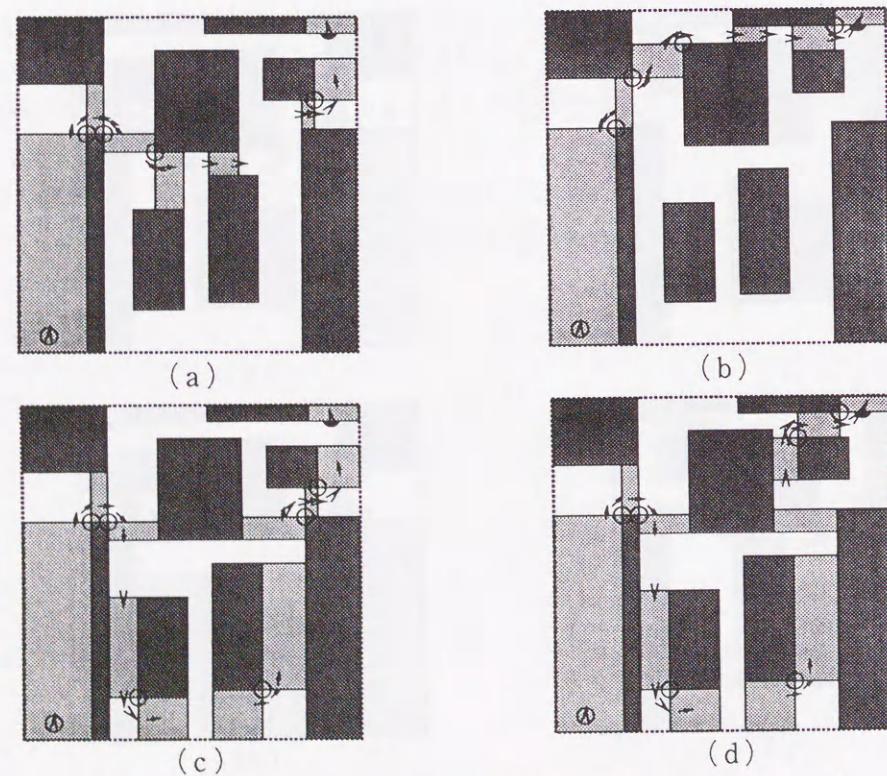


図4.2.16 「急いで走行する」場合の生成サブゴール (ケース2)

4.2.8 結 言

本研究では人間が行っているような、地図情報があいまいであっても未知環境や障害物に対する適応性の高い経路計画を行える手法をファジィ手法を用いて実現することを試みた。まず地図情報のあいまいさについて検討し、物体が存在することに對する確信度をもちいてあいまいな地図を定義した。ここではあいまいなサブゴールを地図上のファジィ集合の代表点として考えた。さらにこれらを用いて実際人間が行っているような階層的な経路決定を模擬する経路計画アルゴリズムを提案した。本方式においては、スタートからゴールに至る通過可能領域列を発見する際に、ロボット自身の特性による制約や通過する領域の広さや領域を移る際の方向転換の量などの組合せで決まる評価をあたえ、詳細な経路計画を行う前に優先順位をつけることで詳細計画を行う際の処理を低減できるようにした。さらに通過可能な領域列に対して、人間から与えられる概念的な目標とロボット自身の特性を考慮して経路計画を行う際に、領域の広さや環境との距離などをロボット自身の特性から定まる量を基準として表現し、経路計画の処理過程をロボットの特性に依存しないかたちで記述できるようにした。

本研究で提案した手法に対して経路計画シミュレーションを行った結果、与えられた目的に応じて地図内の領域を適切に評価し、経路生成およびサブゴール生成を行えることが確認でき、本手法の有効性が検証された。本論文では次節のシミュレーションおよび移動実験で用いられたものと同様の例題に対してシミュレーションを行ったが、実際には環境内に物体がより多く存在する場合、領域分割を行うと当然ノードおよびエッジの数が飛躍的に増大する。よってこのような場合には、環境内に物体の数が増えないような確信度マップの生成やノードの統合/消去を行う必要がある。しかしながら、本方式は本来オフラインで経路計画を行うことを前提としているため通常の屋内環境程度の経路であれば十分適用可能で、特に経路が明確でなくあいまいな評価や目的が与えられている問題には向いていると考えられる。

第5章 行動決定アルゴリズムに関する研究

5.1 自律移動ロボットの行動決定ファジリアルゴリズム

5.1.1 緒言

近年、知能ロボットの自律行動決定機能の高度化を目指した新しい研究アプローチとして、階層的な制御メカニズム[1][2]や生物的な行動モデル[3][4]を提唱した研究がいくつか発表されている。これらはいずれもロボットの行動 (behavior) というマクロなレベルでかつ表現が困難な機能の実現を対象としている点で従来のロボット制御とは異なった新たな研究方向であると考えられる。しかしながら、今までのところロボットの高度知能化においては、人間の複雑な判断機構、理性・常識などの模倣が最も大きな課題として残されているのが現状である。このようないわゆる「人間らしい」機能をロボットにもたせるためには、人間に似たマクロ判断機構をもち、高度な自律制御が可能な知的機械システムの制御系の構築手法を確立することが必要となる。

このヒューマンフレンドリロボットの実現を考えた場合、知的情報処理手法やモデル構築手法はもとより、複雑な動作シーケンスをマクロにかつ柔軟に記述できる方法論が最も重要になってくる。しかしながら例えばAIプロダクションルールによる表現方法を用いた場合、このようなシーケンスフローの記述が可能である反面、あいまい状態を容易に取り扱うことができない。一方、ファジールールによる表現方法を用いた場合はあいまい状態は表現し得るが、分岐を伴うシーケンシャルな多段の推論を表現するには不向きである。

そこで本研究では、移動ロボットを対象としてオンラインで移動する際、変化する前方の環境に対して比較的あいまいな状況を常に認識しながら的確な判断を行っているような、人間の柔軟な状況判断・適応能力を模擬した行動決定アルゴリズムを構築することを試みる。ここでは、あいまいな状態を記述でき、かつシーケンスフローが表現可能なファジリアルゴリズムの概念を用いて、人間が通常行っているのに近い行動決定アルゴリズムを記述する方法を提案する[5][6][7]。また人間は行動目的を自律的に判断し状況に応じた行動決定を行っていると考えられるが、このような目的自律判断機能によるスーパーバイザリなファジリアルゴリズムチューニング方法についても述べる。尚、本手法はシミュレーションおよびロボットを用いた実証実験でもその有効性を検証しているので、これについても報告する[8][9][10][11]。

5.1.2 行動決定ファジリアルゴリズムの構築

ファジリアルゴリズム (Fuzzy Algorithm) [12][13]は意思決定システムの近似解析のための一手法としてL.A.Zadehにより提唱された概念である。これはクリスプ量およびファジィ量

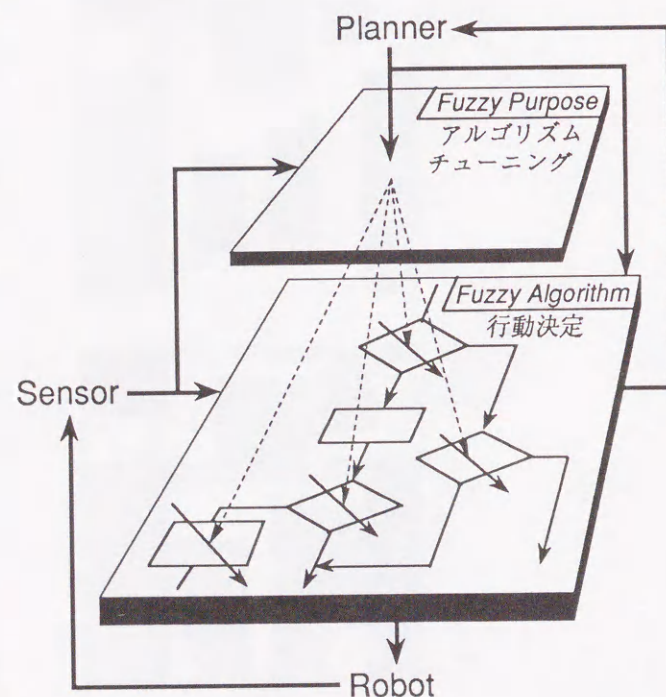


図5.1.1 行動決定ファジリアルゴリズムの概念図

が混在するようなシーケンスフローが表現可能な記述手法で、ファジィ理論の基本的概念となったものである。以下ではこのファジリアルゴリズムを用いることにより自律移動ロボットの行動決定において人間に近いマクロ判断機能を表現することを考える。

図5.1.1に本方式の概念図を示す。あらかじめ移動ロボットのあいまい行動決定のシーケンスフローはファジリアルゴリズムにより記述されている。ロボットが動作中はSensorの外界情報およびPlannerの目標情報により得られたあいまい行動目的 (Fuzzy Purpose) に従ってファジリアルゴリズムの分岐および処理を上位制御することによりオンラインでアルゴリズムチューニングが行われる。これにより、マクロな行動決定シーケンス、あいまいな状況認識、柔軟な環境適応能力、等を有するフレキシブルなスーパーバイザリ制御システムを構築できる。以下では本方式で用いたファジリアルゴリズムおよびシステム構築に関するルール記述について述べる。

5.1.3 修正ファジリアルゴリズムの提案

Zadeh教授によるとファジリアルゴリズムは「ファジィ指示 (Fuzzy Instructions) の順序集合」で、定義・生成・関係・決定の4つの表現能力をもっているとしている。これらのアルゴリズム記述はいずれも我々の日常のあいまい思考のスタイルに極めて近いマクロな表現能力をもっている。

しかしながら、ファジリアルゴリズムをそのままロボットの行動決定アルゴリズムの表現に用いるには実用上の問題がある。ここではこの問題点および提案した修正法について述べる。

(a) ファジリアルゴリズムについて

ファジリアルゴリズムを用いると、一般に知られているIF...THEN...で表現されたファジールールを用いるファジィ制御と異なり、IF...THEN...ELSE...のファジィ分岐を許しているため、クリスプ量とファジィ量が混在するような極めて自然な表現形式でシーケンスアルゴリズムを記述できる特長がある。ファジィ制御で一般に用いられているMIN-MAX重心合成に基づくファジィ推論などは、通常のシーケンスプログラム中ではある一つの処理ブロックにしか相当しないのに対して、ファジリアルゴリズムでは分岐処理にまでファジィ数を導入できるため柔軟にアルゴリズムフローを記述することができる。

ところで、本来のファジリアルゴリズムではファジィ分岐後の処理を並列発火し、それぞれ前件部グレードの重み (あいまいさ) を継承する形で後件部を実行することが提案されている。しかしながら、この方法で実際のアルゴリズムを実行すると並列処理が増大し、やがてアルゴリズムフローが爆発するため、収束が保証されている処理量減少規則が必要となる。さらに逐次制御アルゴリズムなどの各推論サイクルごとに一つの出力を求めなければならない場合には、どうしても出力段階では処理を収束させる必要がある。

室伏ら[14]はこのファジリアルゴリズムを用いて模型自動車のファジィ制御を実現しており、その言語指示に対する有効性を簡単な実験により実証している。この論文ではファジィ分岐におけるルールの前件部グレード値に対しスレシヨールドを設け、これを越えるかどうかで分岐を決定しており、アルゴリズムの処理を爆発しないようにしている。しかしながら、スレシヨールド固定のため完全にクリスプな分岐処理と同等になってしまっており、ファジリアルゴリズム本来の特長が生かされていない。

(b) 修正ファジリアルゴリズム

前述のようなファジリアルゴリズムの問題点を考慮し、本研究ではファジィ分岐のルール前件部にスレシヨールド値を設けこれを自律的に判断した行動目的によりチューニングする方法 (以後、修正ファ

5.1 自律移動ロボットの行動決定ファジアルゴリズム

ジアルゴリズムと呼ぶ)を提案する。この手法の概念図を図5.1.2に示す。各ファジ分岐にはそれぞれスレシヨールド値(図では μ_1, μ_2 など)が定義されており、IF...THEN...ELSE...の形式で分岐表現がなされている。

分岐が通常のクリस्पで表現されている場合、前件部が成立(Yes)すればTHEN部が、不成立(No)ならばELSE部が一方だけ実行(単一発火)される。しかしながらファジ分岐の場合、前件部にある入力を与えられ、そのマッチ度(成立グレード値)が m であったと仮定すると、マッチ度 m でTHEN部が、マッチ度 $(1-m)$ でELSE部がともに実行(並列発火)される。ところが、この方法で実際のアルゴリズムを実行すると前節で述べたように並列処理が増大し、やがてアルゴリズムフローが爆発する。

これに対し、提案した修正ファジアルゴリズムではファジ分岐のスレシヨールド値を μ_1 と仮定すると、 m が μ_1 より大きいときはTHEN部が m のマッチ度で、 m が μ_1 より小さいときはELSE部が $(1-m)$ のマッチ度でどちらか一方が実行(単一発火)されるものとする。このため、アルゴリズムの分岐処理が爆発することはない。しかも、アルゴリズムチューニングの重要なパラメータであるスレシヨールド値をアルゴリズムの上位レベルで制御(本研究では自律移動ロボットを対象としているため行動目的に従って制御)しているため、オンラインで状況に応じてアルゴリズムのフロー自体を変化させることができる。

さらにこれを本来のファジアルゴリズムに近づけるための手法としては、例えば分岐の際それぞれのグレード値が近いような場合は並列処理を行い、離れた場合はグレード値が大きいほうの処理のみを選択するか、あるステップ数を限定してその間だけ並列分岐を行うなどの処理方法が考えられる。

5.1.4 あいまい概念およびあいまい状況の記述

前述の修正ファジアルゴリズムを用いてメインフローを記述する前に、ここでは行動決定のアルゴリズム記述に必要なあいまい概念およびあいまい状況を定義する。

(a) あいまい概念

まず初めに状況記述で用いるあいまい概念の一例として、障害物(Obstacle)・壁(Wall)・前方空間(Free_space)・ロボット(Robot)に関する属性をファジ変数とクリस्प変数が混在したフレーム形

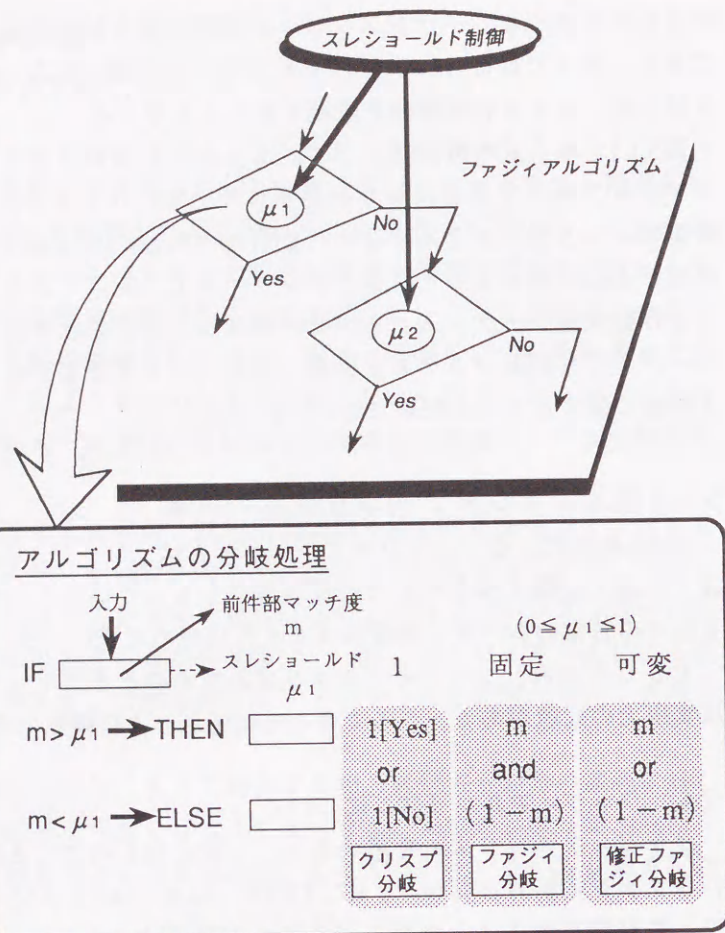


図5.1.2 修正ファジアルゴリズム

5.1 自律移動ロボットの行動決定ファジアルゴリズム

式で定義したものを図5.1.3に示す。ここでの各属性集合は言語指示の場合のLinguistic Variableに相当し、人間に近い感覚で表現されているものとする。

カッコ内に略称で示したラベルは、斜体がファジラベルでその他はクリस्पなラベルである。上記の表現は属性スロット値がファジラベルで定義できるフレーム(以後、ファジフレームと呼ぶ)により記述される。

(b) あいまい状況

移動の場合、最も問題になると考えられるのが走行時において前方が通過できる(Passable)のか、障害物が回避できる(Avoidable)のかなどといった行動決定において必要とされる比較的あいまいな状況に対する認識である。このような状況認識を明確に記述するのは容易ではないが、ここでは前述の修正ファジアルゴリズムを適用し、あいまい状況の近似的表現を行うことを考える。

図5.1.4に前述のあいまい概念を用いて"PASSABLE"と"AVOIDABLE"を表現した例を示す。ここで、PASSABLE, IMPASSABLE, AVOIDABLE, UNAVOIDABLEはあいまい状況を表現するファジ数を示す。この例では、例えばPASSABLEを定義する場合、IMPASSABLEの定義を行ってから残りの概念をすべてPASSABLEとしている。ある概念を定義しようとする場合、概念自身の集合とその補集合のうち規定しやすい部分のどちらかから定義すればよい。

ルールの前件部にはファジ変数とクリस्प変数の両方が混在しており、比較的人間が行っているあいまい状況判断に近い定義ができると考えられる。この例の場合、後件部は同じであるので例えばPASSABLEに関しては、上から順にルー

Fuzzy concept Obstacle (OB)	
Kind	human (HM)/animal (AN)/not_creature (NC)
Char	very careful (VC)/careful (C)/normal (NO)
Dist	close (CL)/far (FR)/out of range (OR)
Dire	S_forward (SF)/forward (FW)/side (SD)
Speed	very fast (VF)/fast (F)/slow (SL)
S_dir	approach (AP)/not_approach (NA)
Fuzzy concept Wall (WA)	
Dist	close (CL)/far (FR)/out of range (OR)
Dire	S_forward (SF)/forward (FW)/side (SD)
Fuzzy concept Free-space (FS)	
Width	very narrow (VN)/narrow (N)/wide (W) /very wide (VW)/out of scan (OS)
Dist	close (CL)/far (FR)/out of range (OR)
Fuzzy concept Robot (RO)	
Wagon	with (WI)/without (WO)
Speed	very fast (VF)/fast (F)/slow (SL)
Time	long (LO)/litttle long (LL)/short (SH)
Energy	empty (EM)/litttle empty (LE)/full (FU)

図5.1.3 あいまい概念の記述例

Fuzzy state PASSABLE	\rightarrow Threshold value ζ ($0 \leq \zeta \leq 1$)
IF (FS.Width is VN)	
THEN (Forward is IMPASSABLE); stop.	
IF (FS.Width is N and FS.Dist is CL)	
THEN (Forward is IMPASSABLE); stop.	
IF ((FS.Width is N and FS.Dist is FR) and RO.Wagon is WI)	
THEN (Forward is IMPASSABLE); stop.	
IF ((FS.Width is W and FS.Dist is CL) and RO.Wagon is WI)	
THEN (Forward is IMPASSABLE); stop.	
Forward is PASSABLE; stop.	

Fuzzy state AVOIDABLE	\rightarrow Threshold value μ ($0 \leq \mu \leq 1$)
IF (OB.Speed is VF and OB.S_dir is AP)	
THEN (Obstacle is UNAVOIDABLE); stop.	
IF (OB.Dist is CL and OB.Dire is SF	
and OB.Speed is F and OB.S_dir is AP)	
THEN (Obstacle is UNAVOIDABLE); stop.	
IF (OB.Dist is CL and (OB.Char is VC or OB.Kind is HM))	
THEN (Obstacle is UNAVOIDABLE); stop.	
IF (OB.Dist is CL and OB.Dire is SF and OB.Char is C)	
THEN (Obstacle is UNAVOIDABLE); stop.	
Obstacle is AVOIDABLE; stop.	

図5.1.4 あいまい状況の記述例

5.1 自律移動ロボットの行動決定ファジィアルゴリズム

ル前件部のMINグレード値を求め、どれかがスレシヨールド ξ を越えた場合、その時点でその時のあいまい状況IMPASSABLEが成立し、どのルールも越えなかった場合、PASSABLEが成立することになる。尚、後件部のPASSABLE, IMPASSABLE等はファジィラベルで定義するため、その状況の成立度合も得られる。

また上記のスレシヨールド値をコントロールすることにより、あいまい状況判断を微妙に変化させることが可能である。さらにあいまい状況定義ルールの中で用いられているあいまい概念自体も変化させれば間接的に状況判断をコントロールすることもできる。例えば、ファジィフレームの属性中のメンバーシップ関数を変化させること (e.g.危険な場合、Obstacleのvery careful, carefulをそれぞれnormal側にスライドさせ、常にcarefulと感じさせるようにするなど) が考えられる。

5.1.5 行動目的自律判断機能の記述

ここではロボットがある行動目的を自律的に判断し、その結果をファジィアルゴリズムに反映させる方法について述べる。この部分はアルゴリズム全体をそのときの行動目的に応じて微妙にチューニングするところで、行動決定アルゴリズムの上位に位置づけられるものである。このように階層構造を構成することにより目的評価機能をスーパーバイザとして、よりマクロな判断に基づくアルゴリズムの統括コントロール手法を実現することを考える。本手法ではこの目的判断部の構成として人間の状況適応能力を参考に「戦略評価」と「戦術決定」の2段階決定方式をとっている。

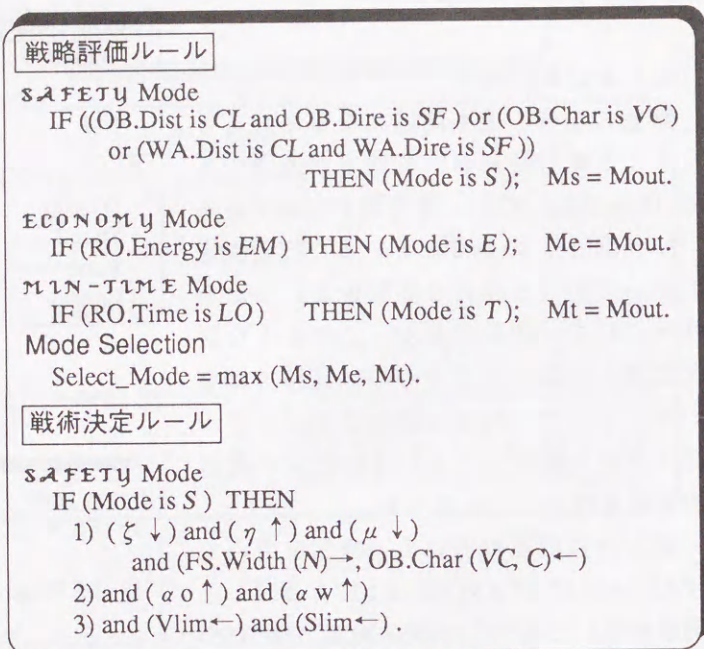


図5.1.5 戦略評価および戦術決定ルールの記述例

(a) 戦略評価

ロボットが現在得られる情報から判断して自分自身が今どういう移動モードで行動すべきかの評価を行う。この部分では明確な定義が困難なあいまい行動目的 (ここではFUZZY_PURPOSEと呼ぶ) をファジィアルゴリズムを用いて表現することを考える。たとえば、「安全性重視 (SAFETY)」、「省エネ (ECONOMY)」、「最短時間 (MIN-TIME)」の3つの移動モードを考えた場合、図5.1.5 (上段) のようになる。

S,E,Tはそれぞれ「安全性重視」、「省エネ」、「最短時間」の3つの移動モードと判断される度合いを示すファジィ数を表す。このアルゴリズムでは行動目的の多数決評価をとるため、各ルールのモード成立度をそれぞれMs, Me, Mtとして、これらの最大値をとるモードをそのときの最大目的として一つ選択する。

(b) 戦術決定

前述の戦略評価に基づきアルゴリズム中でこの戦略を反映させるための戦術を決定する。ここでは評価された戦略ごとにアルゴリズム中で主に以下の3点を修正する機能を提供するものとする。

5.1 自律移動ロボットの行動決定ファジィアルゴリズム

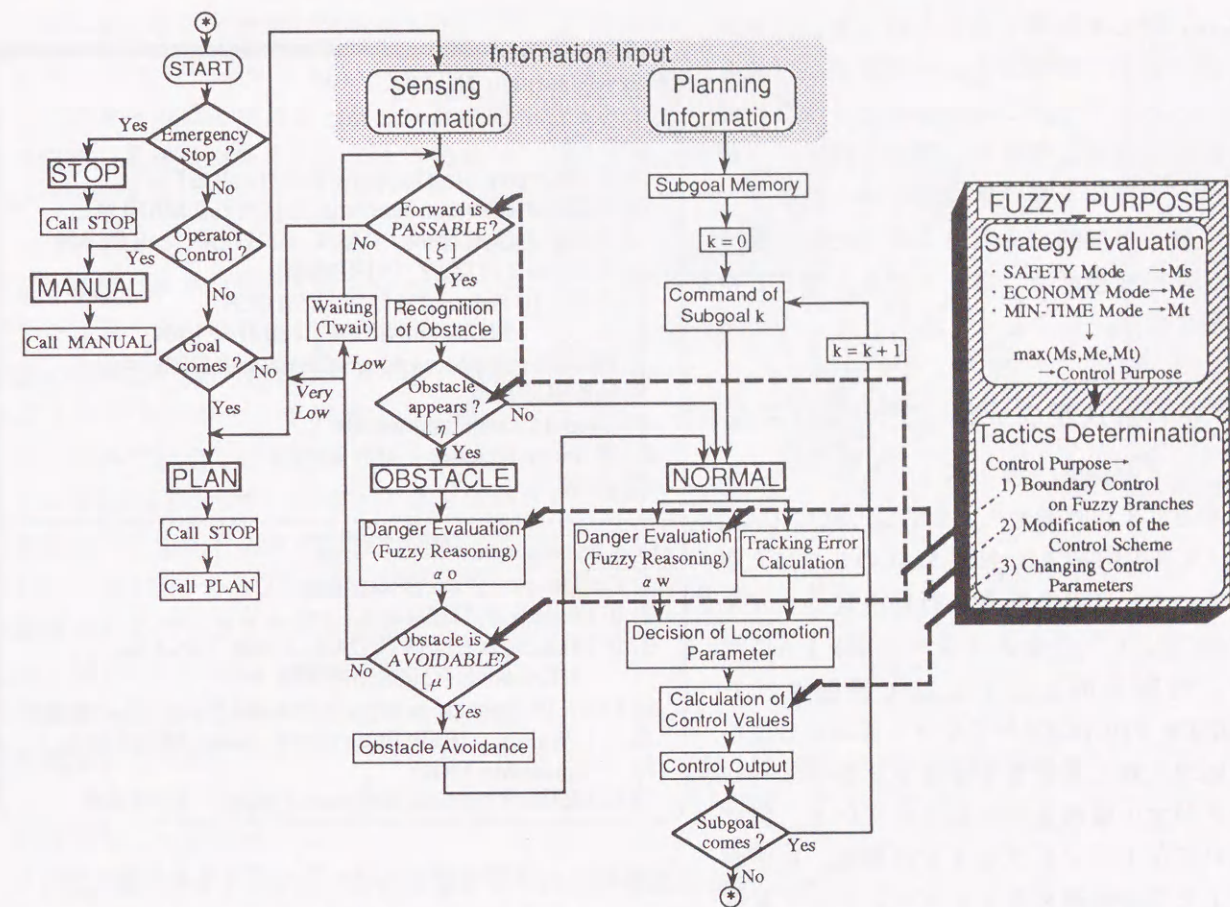


図5.1.6 目的自律判断機能を有する行動決定ファジィアルゴリズム

- 1) アルゴリズムの分岐判断制御
 - ・ファジィ分岐のスレシヨールド値の修正
 - ・あいまい概念の属性パラメータの修正
- 2) 行動方策の修正
 - ・経路追従/障害物回避の重みの修正
 - ・危険度評価のスレシヨールド値の修正
- 3) 制御条件の変更
 - ・速度、操舵メンバーシップ台集合の修正

これらの機能を先ほどの戦略ごとに規定するルールを記述する。図5.1.5 (下段) ではそれらのうち、SAFETY Modeを例にとり戦術決定ルールを示す。

ここで、前件部は一つのファジィ数で表されており、戦略評価の出力グレード値に従って後件部のアルゴリズム修正操作がなされる。上記のルールでは、 \uparrow ・ \downarrow の矢印がファジィアルゴリズム中の前件部スレシヨールド値の上下操作を、 \leftarrow ・ \rightarrow はメンバーシップ関数自体を左右にスライドさせる概念定義の修正操作を示す。

以上のような戦術が推論サイクルごとに一つの戦略に基づいて決定される。これを行動決定アルゴリ

5.1 自律移動ロボットの行動決定ファジアルゴリズム

ズムに対して適用することにより、アルゴリズムのオンラインチューニングが可能となる。本方式ではここで述べたような行動目的を自律的に判断し、それに伴ってアルゴリズム自体を変化させるという柔軟性・適応性を付加的にもたせているため、高度な状況判断機能をロボットに与えることが可能になる。

5.1.6 行動決定ファジアルゴリズムの記述

本方式では行動モードをSTOP, MANUAL, PLAN, OBSTACLE, NORMALの5つに大別している。これらを含む行動決定ファジアルゴリズムの全体フローを図5.1.6に示す。行動目的として上位で評価されたFUZZY_PURPOSEがアルゴリズムの分岐判断処理に対して行なうさまざまなチューニングが太い破線矢印で示されている。図において左上がアルゴリズムの開始、右下が終了でこの処理を各サンプリングタイム毎に逐次繰り返す。図の上方にある入力部はセンシング部およびプランニング部[15]からの入力情報を取り込む部分で、サブゴール情報はオフラインであらかじめ目標値メモリに入力されていると仮定しアルゴリズム実行中にサブゴールに到達するたびに順次与えられる。

アルゴリズムフロー全体を統括制御するメタルールはファジアルゴリズムを用いて、例えば図5.1.7(上段)のように記述される。ここでIMPASSABLEは5.1.4節で定義したあいまい状況を示す。後件部の太字は分岐先のファジサブアルゴリズムを表しており、FUZZY_PURPOSEは前節の目的判断部の処理ルーチンを示す。

さらにサブアルゴリズムOBSTACLEについて、目的判断部において「安全重視」モードがすでに選択されているものと仮定した場合の記述例を図5.1.7(下段)に示す。ここで斜体字はファジ数、太字はファジサブアルゴリズム、その他はクリスプ情報を表す。また"Obstacle is UNAVOIDABLE with Very Low"とは障害物に対する回避可能性がVery LowのグレードでUNAVOIDABLEであること示している。尚、ステップ5)では回避不能の判断後、障害物の様子を見るためしばらくウェイトイングしたあとメインアルゴリズムMETAのステップ6)へ戻ってリトライしていることを表している。尚、図5.1.6における危険度認識、障害物回避方策等についてはここでは割愛する[16][17]。

5.1.7 有効性検証シミュレーション

提案した手法のあいまい状況の記述性およびファジアルゴリズムチューニング機能を確認するため、2次元の屋内環境を想定した計算機シミュレーションを実施した。以下ではまず移動ロボットシ

```

Fuzzy-Algorithm META
1) Input Subgoals (k=1,...,m).
2) k = 0.
3) IF Emergent Stop happens THEN call STOP;
4) ELSE IF Operator interrupts THEN call MANUAL;
5) ELSE IF Goal comes THEN call STOP; call PLAN;
6) ELSE call FUZZY_PURPOSE;
   IF Forward is IMPASSABLE
   THEN call STOP; call REPLAN;
7) ELSE IF Obstacle appears THEN call OBSTACLE.
8) call NORMAL.
9) Output Control commands.
10) IF Subgoal comes THEN k = k + 1.
11) go to 3).
    
```

```

Fuzzy-Subalgorithm OBSTACLE
1) Decide  $\alpha_o$  [Fuzzy Reasoning].
2) IF Mode is S THEN  $\alpha_o \uparrow$  and  $\mu \downarrow$ .
3) IF Obstacle is UNAVOIDABLE with Very Low
   THEN call STOP; call REPLAN;
4) ELSE IF Obstacle is UNAVOIDABLE with Low THEN
   Waiting; IF (Waiting time  $\leq$  Twait) THEN go to 5);
5) go to META 6);
6) ELSE Call Obstacle_Avoidance_Rule. RETURN.
    
```

図5.1.7 行動決定ファジアルゴリズムの記述例

5.1 自律移動ロボットの行動決定ファジアルゴリズム

ミュレータについて簡単に述べたあと、シミュレーションにおいて用いられたパラメータ、ルールおよびスレーショルド制御の説明、最後にシミュレーション結果について報告する。

(a) 移動ロボットシミュレータ
シミュレーションに先立ち、ロボットの行動決定および動作を確認するため、移動ロボットシミュレータを構築した。本シミュレータで仮定した移動ロボットの状態モデルを図5.1.8に示す。ロボットは8個の超音波センサを進行方向に設置しており、各15°ずつ120°のエリアの前方物体までの距離を計測できる。これは実験システムと全く同じハードウェア構成を想定したものである。シミュレータでは以下の入力情報を仮定している。

- 1) プランニング情報
(P_{sx}, P_{sy}, δ)_k: サブゴールの位置および収束許容範囲(k=1,...,m)
- 2) センシング情報
(P_{rx}, P_{ry}, θ_r): ロボットの現在位置および姿勢
d_i (i=0,...,7): ロボットから壁または障害物までの距離
θ_o: ロボット座標系における障害物の方位

ここでプランニング情報に関してはあらかじめシミュレータに与えられるが、センシング情報はオンラインでロボットのセンサ系より取り込まれる。これらの情報よりロボットはサブゴール、障害物、壁、狭路等の状態を認識する。

(b) シミュレーション条件

表5.1.1はこのシミュレーションで用いた各種パラメータおよびロボット、センサ(ソナー)、障害物の特性を示す。またシミュレーションにおいては、図5.1.9に示されたような比較的簡単なルールを用いた。あいまい状況PASSABLEに関する最初のルールは

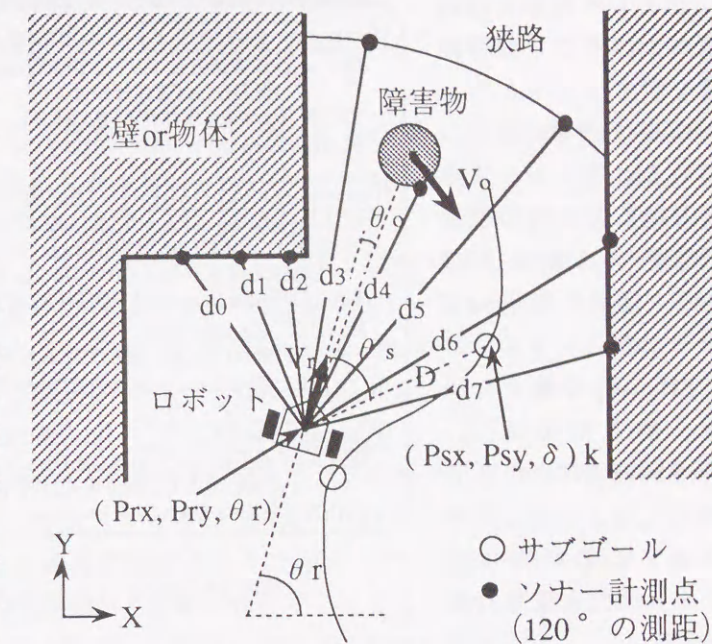


図5.1.8 移動ロボットの状態モデル

表5.1.1 シミュレーション条件

演算サンプリングタイム : 0.1 sec	ロボット -> 速度/操舵制御が可能な自律移動ロボット
表示サンプリングタイム : 0.3 sec	● 速度: 10cm/s
移動環境 : 4m * 4m room	● 大きさ: 幅約 30 cm
サブゴールノード数 : 12 nodes	ソナー -> ロボット中心から前方8方向の測距センサ
スレーショルド値 : TH_imp0 = 0.9 TH_una0 = 0.8 G_imp = 0.02 G_una = 0.05	● レンジ: 前方120° (15° × 8)
	● 計測限界: 2 m以内
	障害物 -> 円柱形移動物体
	● 速度: 40 cm/s
	● 大きさ: 直径20 cm
	● 移動時刻: 130 sec ~ 165 sec

5.1 自律移動ロボットの行動決定ファジアルゴリズム

ロボットの前方空間の幅およびそこまでの距離を示すファジラベルに基づいて記述されている。この規則の出力は前方空間の通過不能度 ($0 \leq IM \leq 1$) である。通過不能度IMが規則のスレシールドTH_impを越えた場合、前方空間はIMPASSABLEと判断され、このシミュレーションにおいてはロボットは引き返す行動をとる。

あいまい状況AVOIDABLEに関する次の規則では、ロボットが出くわした障害物に対する回避不能度 ($0 \leq UN \leq 1$) を定義している。もしUNがスレシールドTH_unaを越えたら障害物はUNAVOIDABLEと判断され、ロボットはIMPASSABLEと同様にその場で引き返す。さらに、UNがTH_una/2以上TH_una以下の場合、ロボットはウェイトン動作（しばらくの間静止する動作）をとるように規則記述がなされている。3つ目の規則は行動目的SAFETYを定義するものでここでは目的評価のため、前方危険度を認識させている。ロボットはこの評価ルールに従って前方危険度 ($0 \leq DA \leq 1$) を認識する。これらのルールに対して、ファジ分岐がIMPASSABLEやUNAVOIDABLEのスレシールド値に従って制御される。例えば、IMPASSABLEのスレシールドTH_impは次式に従い各サンプリング毎に決定される。

$$TH_imp = TH_imp + G_imp \cdot ((1 - DA^2) \cdot TH_imp0 - TH_imp) \quad (5.1.1)$$

ここで、TH_impおよびTH_imp0はそれぞれIMPASSABLEの現在およびデフォルトのスレシールド値、G_impはスレシールドの制御ゲインである。またロボットの移動速度は最初の規則の出力値IMに従い以下のように制御される。

$$Vr = (1 - IM^2) \cdot Vr0 \quad (5.1.2)$$

この式で、Vr0はロボットの初期速度を示す。このシミュレーションにおいてはロボットの現在速度Vrは常にVr0以下になるように設定している。この制御操作は5.1.5節の戦術決定における1)と3)のアルゴリズムチューニング方針に相当する。

(c) シミュレーション結果

図5.1.10から図5.1.12にロボットの軌跡を描いたシミュレーション結果を示す。環境中における静止物体（黒い多角形）と移動障害物（大きい円）はロボット（五角形）から発射される仮想超音波センサに

Fuzzy state	出力値	スレシールド	行動
Fuzzy state PASSABLE	(出力値 $0 \leq IM \leq 1$)		
IF P = [(FS.Width is VN) or (FS.Width is N and FS.Dist is CL)] is high			
THEN (Forward is IMPASSABLE)			
ELSE (Forward is PASSABLE)			
Fuzzy state AVOIDABLE	(出力値 $0 \leq UN \leq 1$)		
IF A = [(OB.Speed is VF and OB.S_dire is AP) or (OB.Dist is CL and OB.Dire is SF and OB.Speed is F and OB.S_dire is AP)] is high			
THEN (Obstacle is UNAVOIDABLE)			
ELSE IF A is little high			
THEN (Obstacle is LITTLE UNAVOIDABLE)			
ELSE (Obstacle is AVOIDABLE)			
Fuzzy purpose SAFETY	(出力値 $0 \leq DA \leq 1$)		
IF S = [(OB.Dist is CL and OB.Dire is SF) or (WA.Dist is CL and WA.Dire is SF)] is high			
THEN (Mode is DANGER)			

図5.1.9 シミュレーションで用いたファジアルゴリズム

5.1 自律移動ロボットの行動決定ファジアルゴリズム

より前方8方向（15° おき、前方120°）の測距を行い粗い距離を検知している。ロボットは最初サブゴール（小さい円）情報のみを与えられ、環境形状等の詳細情報はオンラインで移動中にセンシングにより獲得するものとする。

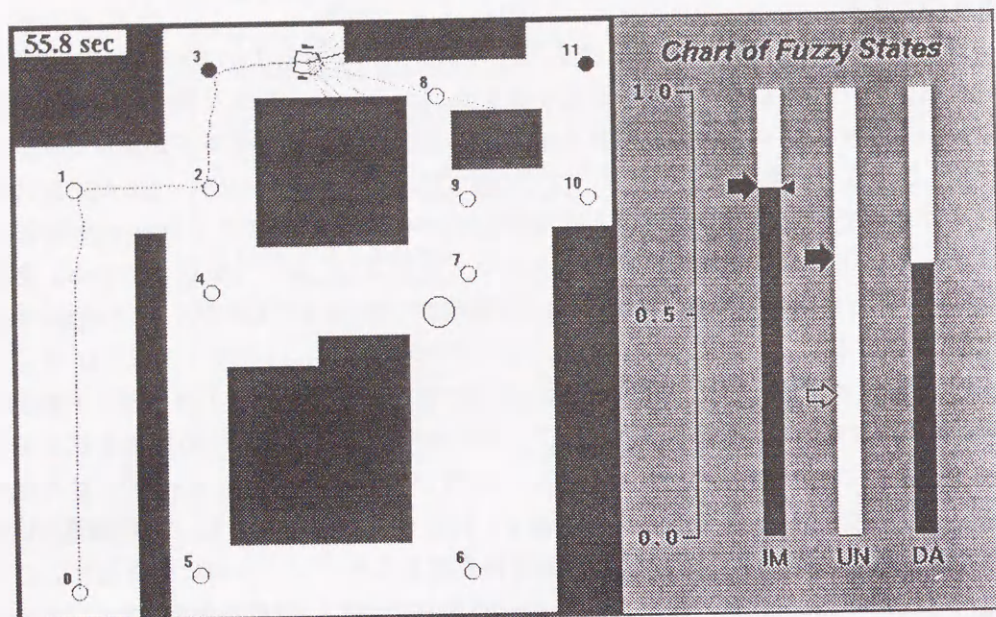
シミュレーションにおいては、ロボットは左下のスタート#0から右上のゴール#11へ移動する途中で中間サブゴール#3をクリアするという問題設定を考えた。シミュレーション時刻はマップ左上に示されている。シミュレーションウィンドウの右側はファジ状況、スレシールドおよびファジ目的の状態を示している。このうち左側の2本の棒グラフはIMPASSABLE (IM)とUNAVOIDABLE (UN)の値を示す。このグラフにおいて、黒の矢印はそのときのTH_impおよびTH_unaのスレシールドを、UNにおける白い矢印はウェイトンモードを決定するスレシールドTH_una/2を示している。右側のグラフはファジ目的SAFETYにおけるDANGER (DA)の評価値で、IMおよびUNのスレシールドはこの評価値DAに従って制御される。

図5.1.10はスレシールドTH_impとTH_unaを共にDAに従い制御した場合のロボットの移動軌跡を示している。状態a)では、ロボットは#3から#11への最短コースであるノード#8への移動を試みるが、引き返す動作をとっている。これは前方が狭かったため危険度が少し上がり、これに伴いやや下がったスレシールド値を通過不能度IMが越えてIMPASSABLEと判断されたことによる。次に状態b)において、ロボットは中央のやや狭い通路に入るが、前方に障害物を発見しサブゴール#4に引き返す。これは危険度DAが大きくなったためにスレシールドTH_unaが一時的に下がり、回避不能度UNがこれを越えたためである。状態c)では、ロボットが近づいてくる移動障害物に出くわすためにかなりスレシールド値が下がる現象が見られるが、前方空間が比較的広いために回避不能度UNが黒と白の矢印の間にある程度留まっている。このためロボットは障害物が通過する間、しばらく停止して様子を見た後、回避行動をとる。

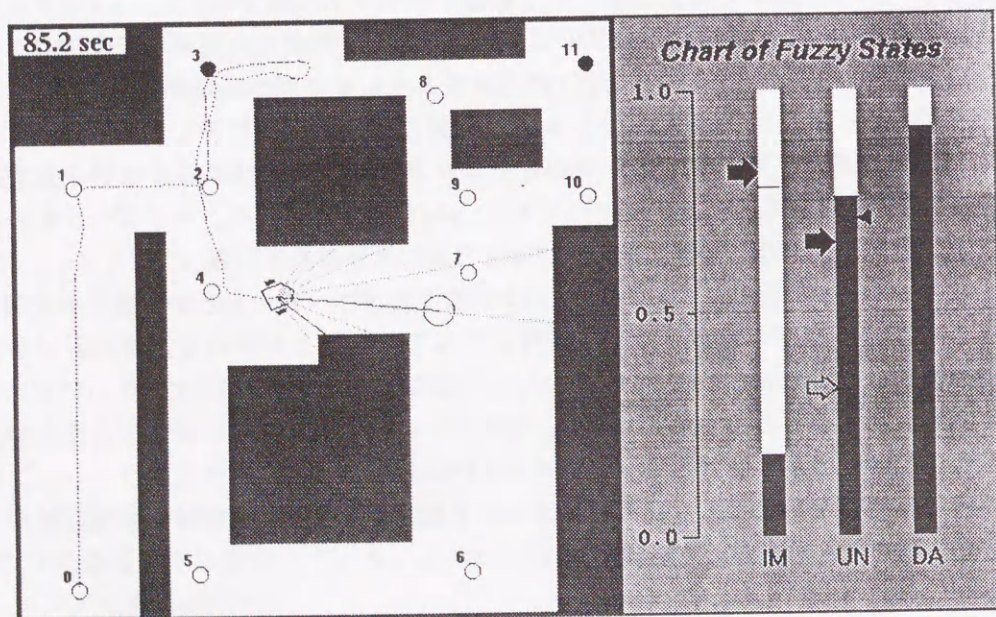
次に、スレシールド制御を全く行わなかった場合のロボットの動きを図5.1.11に示す。この場合、各スレシールドはデフォルト値のまま変化せずに一定となる。そのため通過不能度IMはスレシールド値TH_impを越えにくく、図中b)において示されているようにサブゴール#3から#8への狭い通路を強行で通過しようと試みる。また図5.1.12はスレシールド値TH_impのみを制御した場合のシミュレーション結果を示したものである。図において、ロボットはサブゴール#3と#8の間では引き返す行動をとるが、回避不能度UNが制御されないスレシールドTH_unaを越えないため、サブゴール#7の近くに存在する障害物のそばをぎりぎり通り過ぎるという行動をとることが確認される。

これらのシミュレーション結果より、ロボットは狭路にアプローチするときの通過不能度や障害物に出くわしたときの回避不能度の認識をかなりの確に行っていることがわかる。特に、グラフ中のスレシールド値の変化はIMPASSABLEおよびUNAVOIDABLEの認識がSAFETYの目的判断により妥当なチューニングが行われていることを示している。今回のシミュレーションでは前方空間や障害物の属性値により得られた危険度のみに従いスレシールドを制御したが、他の属性値やプランニング部からの上位レベルの行動目的なども考慮したスレシールド制御を行うとより複雑な状況認識やさらに人間的な行動決定が実現できるものと考えられる。

5.1 自律移動ロボットの行動決定ファジィアルゴリズム



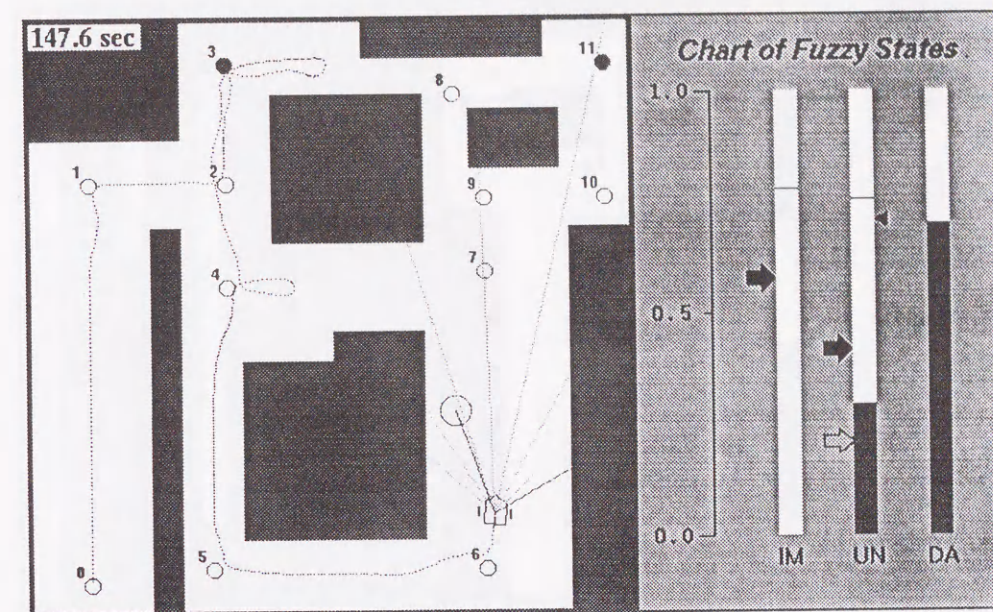
a)



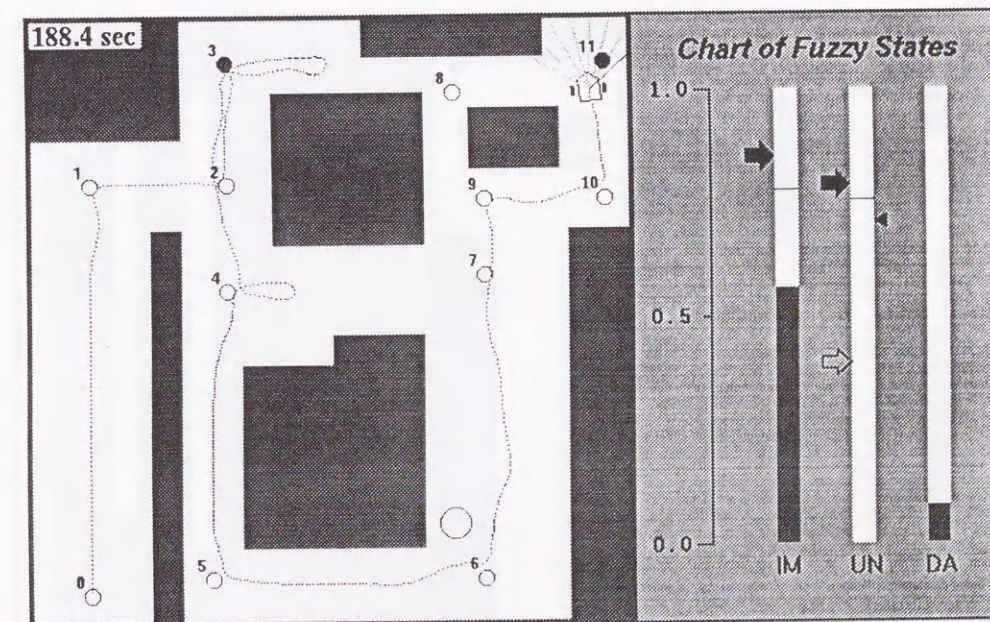
b)

図5.1.10 シミュレーション結果 (1) [TH_imp : controlled, TH_una : unctrolled]

5.1 自律移動ロボットの行動決定ファジィアルゴリズム



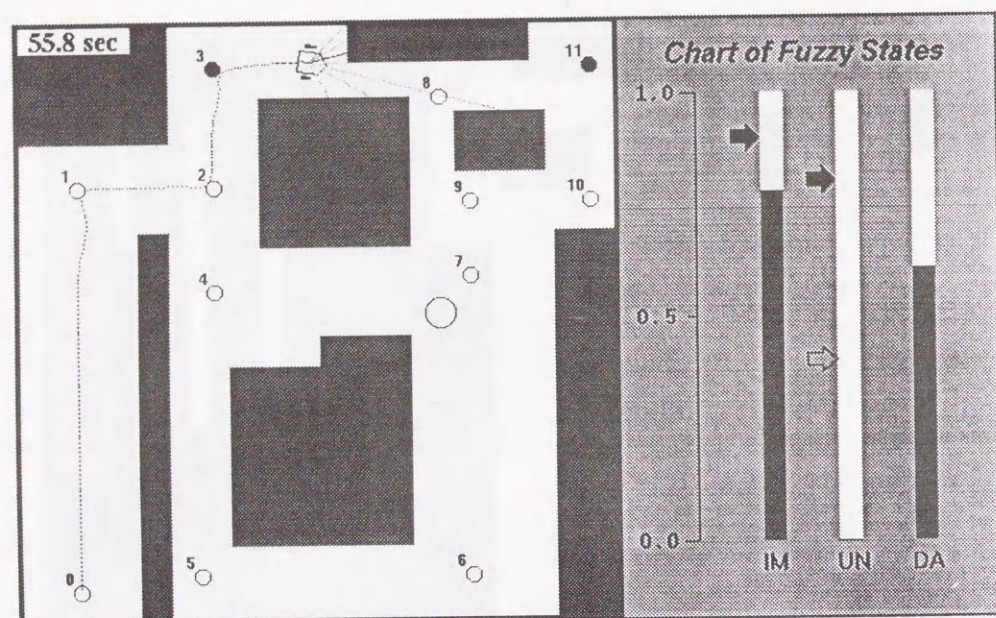
c)



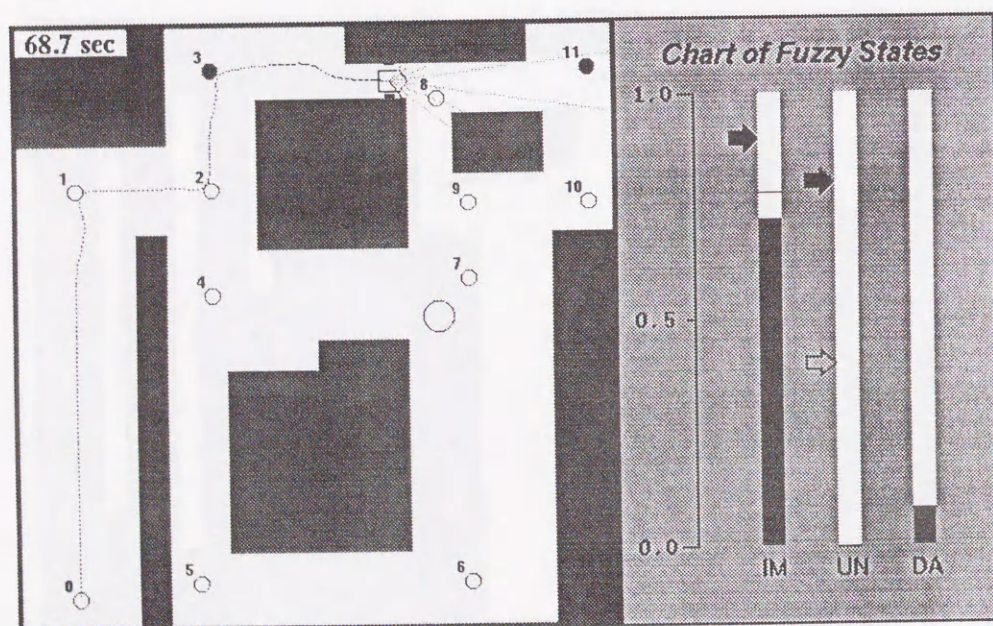
d)

図5.1.10 シミュレーション結果 (1) [TH_imp : controlled, TH_una : unctrolled]

5.1 自律移動ロボットの行動決定ファジィアルゴリズム



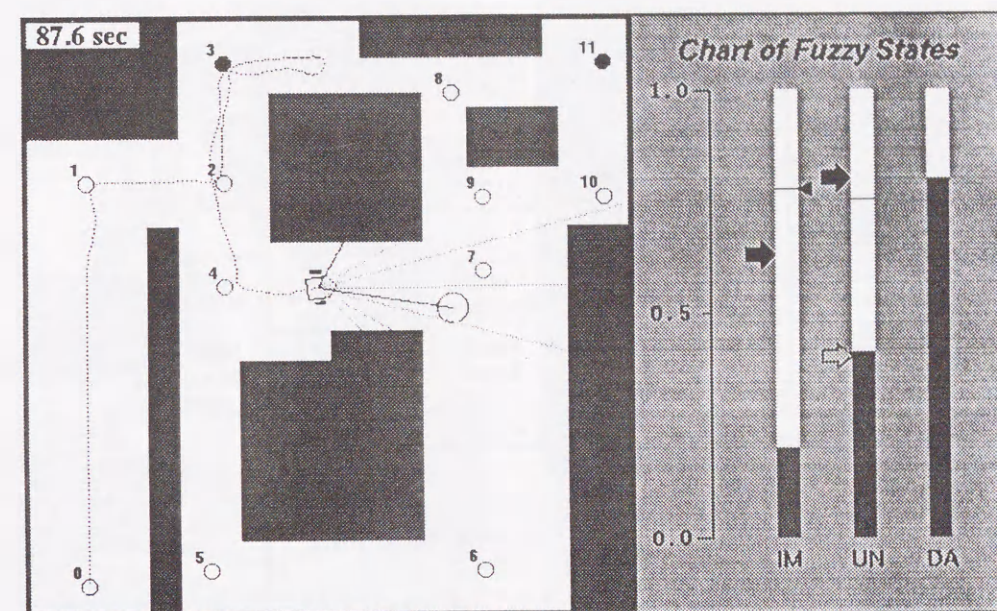
a)



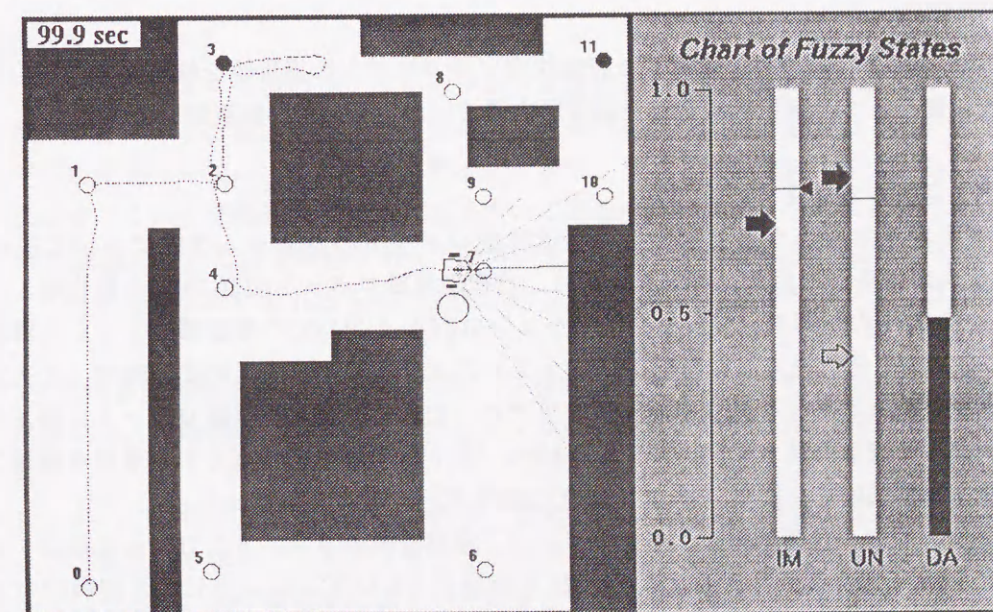
b)

図5.1.11 シミュレーション結果 (2) [TH_imp : uncontrolled, TH_una : uncontrolled]

5.1 自律移動ロボットの行動決定ファジィアルゴリズム



a)



b)

図5.1.12 シミュレーション結果 (3) [TH_imp : controlled, TH_una : uncontrolled]

5.1 自律移動ロボットの行動決定ファジアルゴリズム

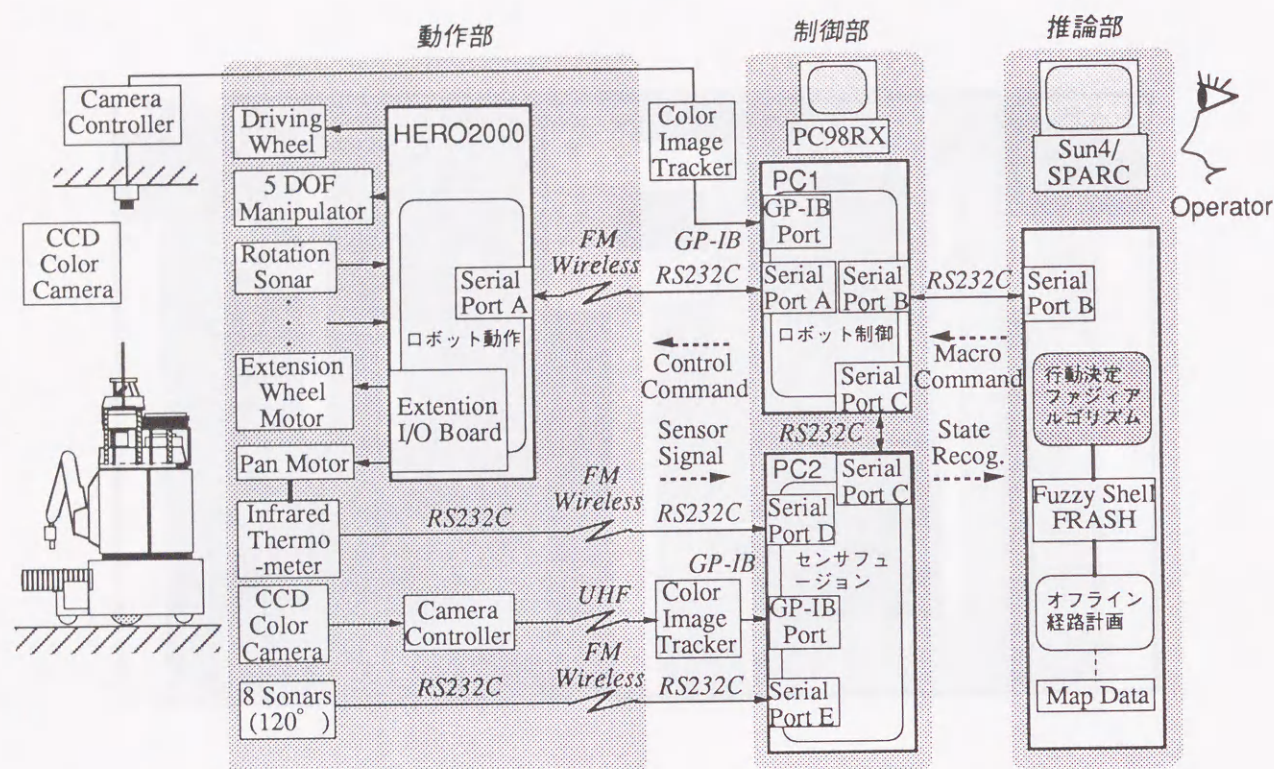


図5.1.13 自律移動ロボットのシステム構成図

5.1.8 実証評価実験

シミュレーション結果に基づき、提案した行動決定アルゴリズムの実用性を検証するため自律型移動ロボットを実際に製作して、部屋を想定した模擬屋内環境において自律移動実験を行った。

(a) 実証評価システムの構成

本節では、実証評価実験のために開発した自律移動ロボットのシステム構成について述べる。図5.1.13に本システムの構成図を示す。本システムは、自律型移動ロボットHERO2000の動作部、パーソナルコンピュータPC98RXの制御部、ワークステーションSUN4/SPARCの推論部の3つから構成される[18]。尚、ここでは本研究で提案した行動決定ファジアルゴリズムの検証が主目的で、2次元平面の操舵/速度制御が可能な車輪型移動ロボットで十分であったため、自律型移動ロボットにはメカニズムが複雑で自重が大きく俊敏な動きに向いていないMRV(第2章参照)ではなくより単純な構造であるステアリング可能な車輪式の自立移動ロボットHERO2000を用いて実験を行なった。

動作部は内部センサおよび搭載センサのデータ入力、移動制御のコマンド出力のみを担当する。制御部はロボットとワイヤレスモデムを通じてRS232C無線通信でBASICコマンドにより制御指令やセンサデータを送受信する。また推論部とはRS232C有線通信により推論結果の受信、制御結果およびセンシングデータの送信を行う。尚、もう一台のセンサフュージョン用コントローラはイメージトラッカ、放射温度計、超音波センサなどのデータを入力し、前方の障害物の生物度を判断するものである[19]。推論部は全体を統括する推論レベルの処理を制御用ファジィシェルFRASH[20][21][22](第6章参照)により

5.1 自律移動ロボットの行動決定ファジアルゴリズム

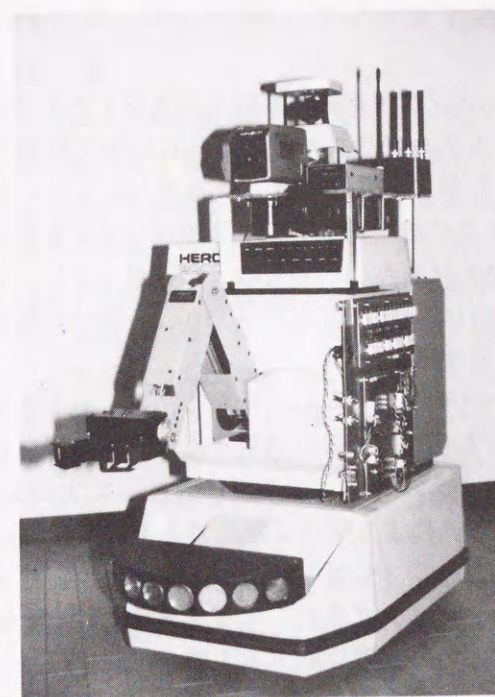


図5.1.14 自律移動ロボットの外観



図5.1.15 実験におけるロボットの移動軌跡

行い、制御部に指令を発する。オペレータはオンラインの推論・制御結果の観測をシミュレーションで用いた移動ロボットシミュレータを通して行うことができるようになっている。

(b) 実験結果

上記システムを用いて、前述のシミュレーションとほぼ同じ環境(4m×4m)を実験室内に実際に製作し、自律移動による行動決定ファジアルゴリズムの有効性検証実験を行った。実験では、ロボットの上部中心に発光体を設け、この軌跡をイメージトラッカを用いて斜め上方からの映像としてモニタ上に表示させ、これを位置計測データとして用いるとともに移動軌跡として撮影する方法をとった。図5.1.14に今回実験に用いた自律移動ロボットの外観を示す。ロボット上部に取付けられたセンサ類は主にセンサフュージョンに用いられるものである。前方環境認識用の超音波センサはロボット下部に設置している。

図5.1.15に屋内移動実験を行った結果のロボット移動軌跡を示す。この例はスレショルド制御を完全に行った場合で、行動決定アルゴリズムのシミュレーション(図5.1.10)とほぼ同等の条件で実施されたものである。

実験では、屋内空間の大きさ等の制限のため、斜め方向の画像から換算することによりロボットの中心位置および移動方位を求め、これにより自己位置同定を行った。そのため奥行き方向の分解能が左右方向に比べて低くなり、若干移動時にふらつく現象が見られたが、少々の環境レイアウトの変化や繰り返し実験に対してもほぼ安定した移動特性が得られた。移動軌跡より明らかなように、シミュレーション結果とほぼ同じマクロ状況判断能力が確認され、本方式の実システムにおける有効性が示された。

5.1.9 結 言

クリスプ量およびファジィ量が混在するようなシーケンスフローが表現可能なファジアルゴリズムの概念を用いた自律移動ロボットの知的行動決定に関するアルゴリズム構築方法について研究を行った。本研究ではファジアルゴリズムにおけるルール前件部に上位で制御されるスレシールドを設け、これによりオンラインでアルゴリズムを適応的に変化させる修正ファジアルゴリズムを提案し、自律移動ロボットの「人間らしい」行動決定ルールの記述を試みた。本方式は、ファジィフレームによるあいまい概念および修正ファジアルゴリズムによるあいまい状況の定義ができ、自律的に判断した行動目的に応じた柔軟なアルゴリズムチューニング機能を有する。

本方式を用いると、あいまいな行動決定を連続して（シーケンシャルに）行なう必要があるような知能ロボットの多段意思決定プロセスが記述できる。これにより人間に近い感覚であいまい処理を含むロボットの行動決定ルールを柔軟に記述できるようになると考えられる。ここでは例として"PASSABLE"や"AVOIDABLE"などのあいまい状況をファジアルゴリズムにより簡単に記述できることを示した。またこのあいまい状況判断はスレシールドを基準とするファジィ分岐で表現しているため、ロボットの行動目的/環境情報に応じたアルゴリズムに自動的にチューニングされ、ロボット自体にある種の状況適応能力を付加する機能も提供する。本研究では"行動"という問題に対する人間のマクロ判断・決定機能の一表現手法を示したが、ヒューマンフレンドリロボットの実現を目指した場合、今後このような研究アプローチが不可欠になるものと予想される。

また本方式の計算機シミュレーションを行った結果、マクロ状況判断機能の有効性を検証することができた。さらに実際の自律移動ロボットを用いて同等の屋内移動実験を行ったところ、シミュレーションで得られた結果とはほぼ同じ移動特性および状況判断能力が確認され、本方式の基本的な有効性が示された。

5.2 ペトリネットを利用したファジアルゴリズムのあいまい性評価

5.2.1 緒 言

人間に近いマクロ行動決定機能をロボットにおいて実現するため、筆者らはファジアルゴリズムにスレシールド制御による実用的な制限を加えることにより、アルゴリズムチューニング機能を付加した修正ファジアルゴリズムを前節ですでに提案している[1][2]。この修正ファジアルゴリズムは、基本的に単一発火であるため本来の並列発火であるファジアルゴリズムに比べてアルゴリズムの爆発という問題はないが、真理値が伝搬されるため多段で推論を繰り返すことにより徐々にあいまいさが減少する傾向がある。これを積極的に防止するには、スレシールド制御方策または真理値伝搬アルゴリズムを変更することが考えられる。これ以外の方法としては、ファジアルゴリズムそのものの真理値状態を把握（あいまい性評価）することによりアルゴリズム全体のあいまいさの管理を行ない、推論におけるあいまいさの爆発（または消滅）を事前に予測することなどが考えられる。

本論文では、ファジィペトリネット (FPN) [3]の考え方を導入し、ファジアルゴリズムの適用において最も問題となる多段発火時のあいまいさの爆発を管理する方法について検討を加える。この方法ではファジアルゴリズムのフロー全体のペトリネット表現を行い、これを用いてファジィ真理値トークンの伝搬（マーキング挙動）によりシステムのあいまい性の推移状況を捉えるを試みる[4]。さらに、上述の行動決定ファジアルゴリズムを例題として、ここで提案する評価方法に従ったシミュレーションも行ったのでこれについても報告する。

5.2.2 ファジアルゴリズムの問題点

ファジアルゴリズム (Fuzzy Algorithm) [5]は前節で述べたように意思決定システムの近似解析のため一手法としてL.A.Zadehにより提唱された概念で、クリスプ量およびファジィ量が混在するようなシーケンスフローが表現可能な記述手法で、ファジィ理論の基本的概念となったものである。ファジアルゴリズムを用いると、あいまいさを含んだ行動をファジィラベルを含む極めて自然な表現形式でアルゴリズム記述ができる。また一般に知られているファジィ制御のようにIF...THEN...で表現された複数のファジィルールを用いるのではなく、IF...THEN...ELSE...のような分岐処理（ファジィ分岐）を許しているため、決定プロセスにあいまいさを含んだシーケンスアルゴリズムを記述することが可能である。

ファジアルゴリズムにおける分岐処理は、判断（前件部）はファジィ処理、分岐（後件部）は並列発火の形式をとっており、判断における悩みや複数の可能性検討など人間のあいまい意思決定に近い記述が可能となる。しかしながら、本来のファジアルゴリズムでは分岐後の処理が並列発火でそれぞれ前件部グレードの重み（あいまいさ）を継承する形で後件部を実行することが提案されているため、実際のアルゴリズムを実行すると並列処理が増大し、やがてアルゴリズムフローが爆発するという重要な問題をかかえている。

本論文では、ファジアルゴリズムにおいて常に問題となるあいまいさの管理を行なう実用的な手法を提案することにより、推論におけるあいまいさの爆発に対する一つの解決策を与えることを目的としている。

5.2.3 ファジィ分岐処理によるファジアルゴリズムの分類

Zadehの論文ではファジアルゴリズムにおける実用上の問題にまでは言及されていないが、この実現方法に対していくつかの解釈がなされている。上記b),c)のファジィ分岐の具体的な処理方法を提案した例を以下に示す。

5.2 ペトリネットを利用したファジリアルゴリズムのあいまい性評価

<単一発火ファジィ分岐>

- 1) Dubois、水本らの方法[6][7]
 - a. 最大グレード実行法
 - b. 確率的実行法
 - c. 非決定的実行法
- 2) 室伏らの方法[8]→スレシヨールド固定
- 3) 前田らの方法 →スレシヨールド制御

<並列発火ファジィ分岐>

- 4) Novakらの方法[9]
 - a. 完全実行法
 - b. 頭切り実行法
- 5) 北島らの方法[10]→あいまいさの管理

1)のDubois、水本らの方法は、それぞれ、a. 最も高いグレードをもつ分岐先を選択、b. グレード値に比例した確率でランダムに分岐先を選択、c. 与えられたスレシヨールド α 以上のグレードをもつ分岐先から選択することを示唆しているが、いずれも分岐先のうち一つのみを選択実行するもので、並列実行の考え方は述べられていない。

実用面では、2)の室伏らがファジリアルゴリズムを模型自動車の制御に適用している。ここではファジィ分岐におけるルールの前件部グレード値に対しスレシヨールドを設け、これを越えるかどうかで分岐を決定しており、アルゴリズム処理の爆発を抑えている。制御においてはこの方法が比較の実用的であると考えられるが、スレシヨールドを固定しているためクリスピーな分岐に近づいている。

さらに3)の筆者らはこれを発展させ、ルール前件部のスレシヨールド値を自律的に判断した目的に応じてアルゴリズムをチューニングする方法（修正ファジリアルゴリズム：MFA）をすでに提案している。例えば、ファジィ分岐の場合、前件部にある入力を与えられ、そのマッチ度（成立グレード値）が m 、スレシヨールド値が μ であったと仮定すると、マッチ度 m でTHEN部が、マッチ度 $(1-m)$ でELSE部がともに実行（並列発火）される。これに対し、提案した修正ファジリアルゴリズムではTHEN部とELSE部のどちらか一方が実行（単一発火）されるため、アルゴリズムの分岐処理が爆発することはない。

4)のNovakはファジリアルゴリズムをIFQ THEN RT ELSE REの形式で表現し、a. RT、REの両方をそのまま実行し、以後実行結果のファジィ集合を用いる、b. RT、REのファジィ集合を αT 、 αE で頭切りしたもの的合成して実行結果とするものを提唱しており、よりファジィ推論の考え方に近づいている。しかしながら、基本的に並列発火であるため、実用上の問題も残されている。5)の北島らは、扱うモデルに関する知識をあらかじめファジィルールで表現しておき、これを用いてファジリアルゴリズムにおけるファジィ集合演算を行うもので、基本的には並列発火であるが、あいまい性の増大をある程度減少はできるものの爆発は避けられない。

これら以外にも、例えば、

- A) 一定値以下のグレードになるとその分岐を行わない。
- B) 一定回数の分岐のみ並列発火を認める。

5.2 ペトリネットを利用したファジリアルゴリズムのあいまい性評価

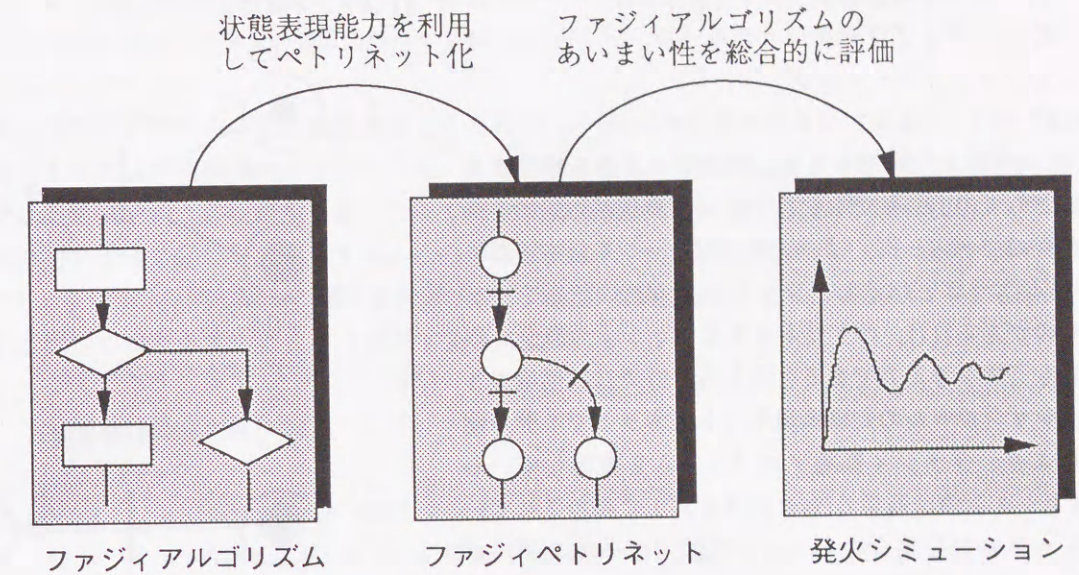


図5.2.1 ペトリネットによるファジリアルゴリズムのあいまい性評価手法

C) アルゴリズム全体として、枝刈りにより動作プロセスを制限する。

D) 分岐のグレードに大きな差が認められるときは単一発火を、あまり差がないとき（グレード値が似ているとき）には並列発火を行う。

などさまざまな分岐実行法が考えられる。このようなファジィ分岐の実行制御を考える際には、分岐に関するファジィ集合のグレードのみならず、アルゴリズム全体としてのあいまい性の状態把握および評価が重要となってくる。そこで以下では、ペトリネットの状態表現能力を利用して、ファジリアルゴリズムのあいまい性を総合的に評価する手法について述べる。

5.2.4 ファジリアルゴリズムのペトリネット表現

前節ですでに述べたようにファジリアルゴリズムを用いる場合、アルゴリズム中のあいまい性の変化の大局的な把握が問題となる。しかしながら、ファジリアルゴリズムそのもののシステム出力に基づいたあいまい性評価を行った場合、あいまいさによるアルゴリズムの爆発が必ずしも予測できない。

そこで本研究では、アルゴリズムに内在するあいまい性をグローバルに捉え、その傾向分析によりある程度の爆発予測ができる方法論の構築を目的とする。図5.2.1に本研究で提案するペトリネットを用いたファジリアルゴリズムのあいまい性評価手法の考え方を示す。本手法では、ファジリアルゴリズムを一旦、ペトリネットに変換する。ここではペトリネットの状態表現能力および状態遷移解析能力を利用し、ファジィペトリネットによるアルゴリズム記述を行い、トークン状態の遷移行列を求め、さらにできあがったFPNの遷移行列を用いて発火シミュレーションを行いそのトークン状態をみることにより、アルゴリズム全体のあいまい性の推移を把握することができる。

最近、ペトリネットにファジィの概念を適用したアプローチ（ファジィペトリネット）がいくつか発表されているが、代表的な例として、

5.2 ペトリネットを利用したファジリアルゴリズムのあいまい性評価

- 1) ファジィ真理値トークン
- 2) ファジィプロダクションルール
- 3) ファジィマーキング

などがある[3] (図5.2.2参照)。1) はプレース中を移動するトークン値として[0,1]の数値を扱うファジィ真理値の伝搬をモデル化したものである。例えば、トランジションをMIN演算、プレースをMAX演算と想定してファジィ真理値の伝搬をモデル化したものがある[11]。ペトリネットにファジィの概念を取り入れた最も自然な考え方であると言える。2) では確信度付きのプロダクションルールの知識表現をペトリネットでモデル化することにより、ファジィ推論アルゴリズムを表現している[12]。3) はトークン値は通常のPNと同様0/1で表わされているが、プレースの部分集合のトークン配置(マーキング)がファジィ集合となるもので、マーキングの分布の不確かさを表現できる。

本研究ではアルゴリズム中のあいまい性の変化を評価する目的であるため、プレース中を移動するトークン値として[0,1]の数値を扱う1)のファジィ真理値トークンと2)のプロダクションルール表現の考え方を採用している。例えば、修正ファジリアルゴリズムの場合、IF...THEN...ELSE...のファジィ分岐は図の4)のように表わされ、トランジションにあたるには発火を決定するスレシヨールドが書かれる。

5.2.5 ファジィペトリネット (FPN) の状態方程式

まずFPNにおけるファジィ真理値をもつマーキングの遷移を状態方程式により表現する。プレース数 $|P|=n$ 、トランジション数 $|T|=m$ のPNにおける接続行列 $A (= [a_{ij}], i=1,2,\dots,n, j=1,2,\dots,m)$ は、

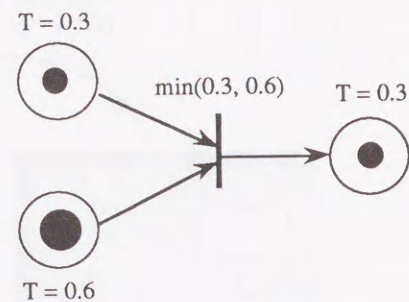
$$A = A^+ - A^- \tag{5.2.1}$$

A^+ : トランジションからプレースへの接続行列 ($n \times m$)
 A^- : プレースからトランジションへの接続行列 ($n \times m$)

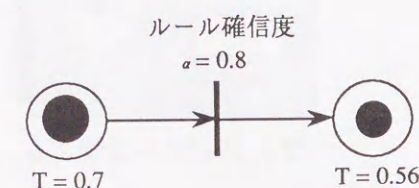
と表わせる。このとき、ある発火系列 σ により、マーキング M_0 が M へ遷移したとすると ($M_0 [\sigma > M]$)、次の状態方程式が成り立つ。

$$M = M_0 + A \cdot \Sigma \tag{5.2.2}$$

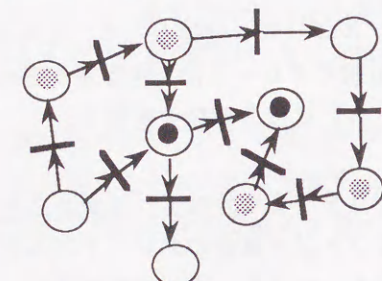
- 1) ファジィ真理値トークン



- 2) ファジィプロダクションルール



- 3) ファジィマーキング



- 4) ファジィ分岐 (MFAの場合)

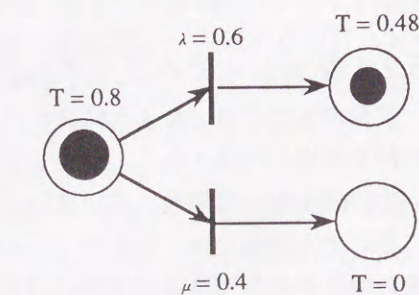


図5.2.2 ファジィペトリネットの種類

5.2 ペトリネットを利用したファジリアルゴリズムのあいまい性評価

M_0, M : 初期、目標マーキング (各プレース中のトークン数を要素とするベクトル、 $n \times 1$)
 Σ : σ 中のトランジションの発火回数ベクトル ($m \times 1$)

通常のPNでは、トークン数が整数値をとるため M_0, M は非負整数ベクトルであるが、FPNではあいまいさを表現するため、非負実数ベクトルとなる。そこで(5.2.2)式より接続行列 A または発火回数ベクトル Σ のいずれかは実数値を扱う必要が生じる。ここでは接続関係はそのまま単に整数値で表現し、トランジションの発火が[0,1]の数値で起こると仮定する。この場合、従来のように発火回数ベクトルにすべての発火(回数)を集約し、一度に M_0 から M への遷移を計算することができなくなる。そこで本FPNでは、次式に従って発火を必ず1回づつ繰り返すことにより演算するものとする。

$$M_n = M_{n-1} + A \cdot \Omega \tag{5.2.3}$$

M_n, M_{n-1} : $n, n-1$ 回目のトークン状態 ($n \times 1$)
 Ω : M_{n-1} から M_n へのトランジションの状態変化ベクトル ($m \times 1$)

今、状態変化ベクトル Ω の各要素は前回のトークン状態(マーキング)に各トランジションにおける発火度合を乗じたもので表わすことができ、

$$\Omega = B \cdot M_{n-1} \tag{5.2.4}$$

B : 各トランジションの発火度合行列 ($m \times n$)

となる。これを(5.2.3)式に代入すると、

$$M_n = (I + A \cdot B) \cdot M_{n-1} \tag{5.2.5}$$

$$= T \cdot M_{n-1} \tag{5.2.6}$$

I : 単位行列 ($n \times n$)
 $T = I + A \cdot B$: 状態遷移行列 ($n \times n$)

となり、 T をマーキングベクトルに毎回乗じることにより、順次次のトークン状態が計算される。これを逐次繰り返すとシステムのあいまい性の遷移状態が把握できる。ファジリアルゴリズムにおいて、一般に B は分岐を選択する場合(単一発火)においては変化しない。本方式ではトランジションの発火順序を考慮することにより、逐次演算でシーケンシャルアルゴリズムにおけるあいまいさの伝搬を評価している。

5.2.6 ファジリアルゴリズムのFPN表現例

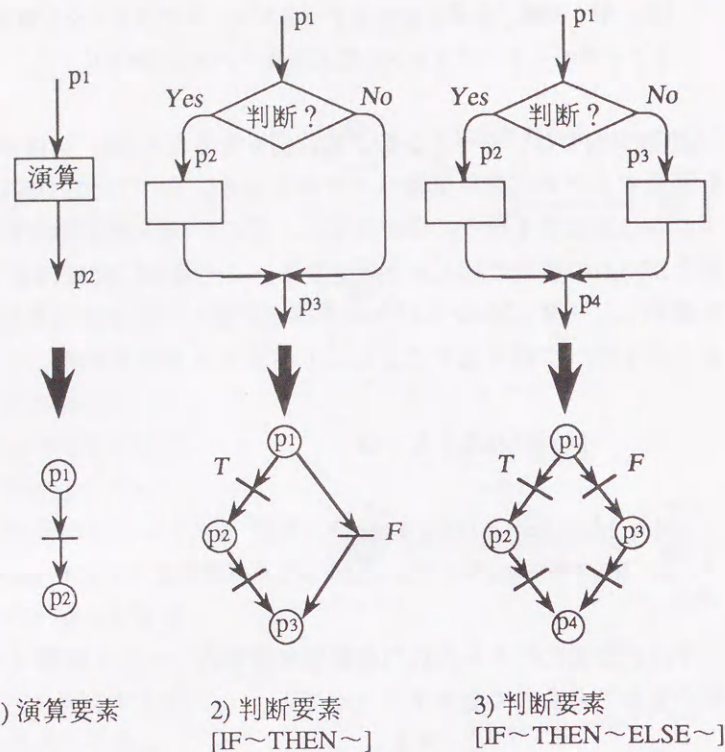
通常のシーケンシャルプログラムは一般に図5.2.3のようなペトリネットのトランジションに変換することができる。1) は演算要素で、入出力はプレース、演算ブロックはトランジションに対応する。2), 3) は分岐に対応する判断要素で、一般にはYes/Noのいずれかの分岐が単一発火となるので、これをペトリネット化した場合、True(T)/False(F)の競合ネットとなりいずれか一方のトランジションが競合解消条件に従って発火す

5.2 ペトリネットを利用したファジアルゴリズムのあいまい性評価

る。しかしながら、3) の判断要素においてファジィ分岐を行うケースがファジアルゴリズムに相当するが、並列発火の場合にはこの競合解消条件は不要となる。但し、スレシヨールドを用いて単一発火させる場合には、競合ネットとなり、スレシヨールドの条件比較が分岐の両方のトランジションで行われる。

ここでは具体的なファジアルゴリズムの例として、筆者らが以前、修正ファジアルゴリズムを自律移動ロボットのあいまい状況に直面した場合の行動決定に適用したものを取り上げる。図5.2.4に行動決定ファジアルゴリズムのフロー図を示す。

図においてアルゴリズムの処理を各サンプリング毎に逐次繰り返す。グレーの部分にはファジィ分岐を示している。このアルゴリズムの上位レベルに行動目的自律判断機能に相当する部分（図では省



1) 演算要素 2) 判断要素 [IF~THEN~] 3) 判断要素 [IF~THEN~ELSE~]

図5.2.3 フローチャートのペトリネット化

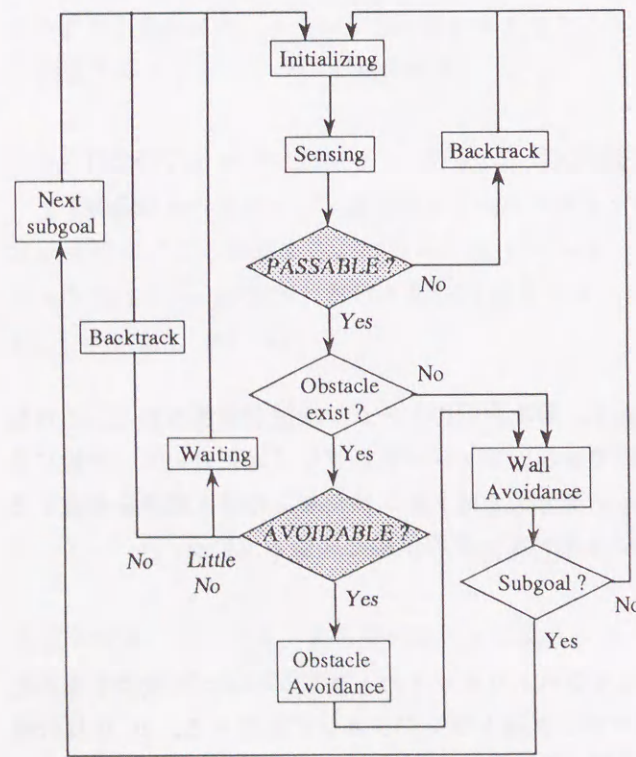


図5.2.4 行動決定ファジアルゴリズム

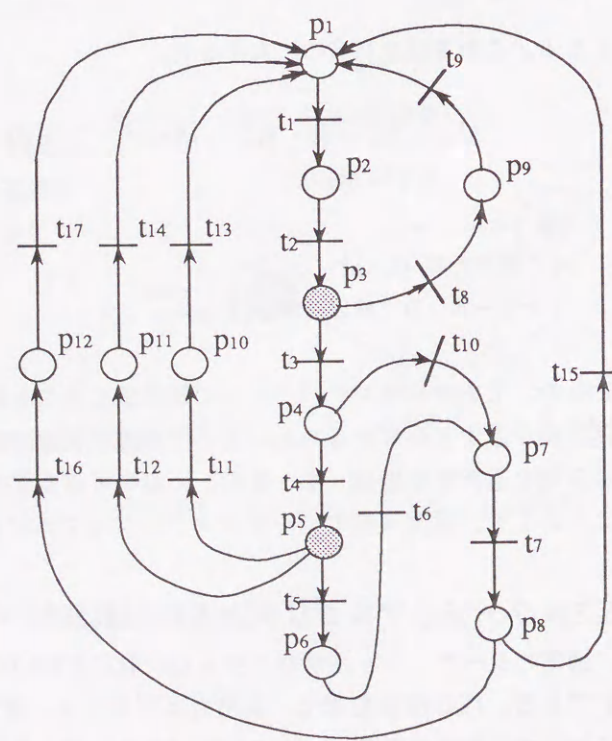


図5.2.5 図5.2.4のペトリネット表現

5.2 ペトリネットを利用したファジアルゴリズムのあいまい性評価

略)があり、ここからファジィ分岐のスレシヨールド制御等さまざまなアルゴリズムチューニングが行なわれる。あいまい状況PASSABLEのルール出力はスレシヨールドλを越えた場合、前方空間はIMPASSABLEと判断され、ロボットは引き返す行動をとる。あいまい状況AVOIDABLEに関するルール出力はスレシヨールドμを越えたら障害物はUNAVOIDABLEと判断され、ロボットはその場で引き返す。μ/2以上μ以下の場合ロボットはウェイトニング動作をとる。

これをペトリネット化したものを図5.2.5に示す。すべての演算要素および判断要素はトランジションに変換され、それ以外の状態をプレースとして構成する。p1,...,p12の12個のプレースとt1,...,t17の17個のトランジションにより定義される(5.2.1)式の接続行列Aは以下ようになる。

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

次に、 $t_3 : t_8 = \lambda : (1-\lambda)$ 、 $t_5 : t_{11} : t_{12} = \mu : 0 : (1-\mu)$ 、 $t_4 : t_{10} = 1 : 0$ 、 $t_{15} : t_{16} = 1 : 0$ [簡単のため、分岐を減少] のグレードで発火が起こるものと仮定すると、発火度合行列Bと状態遷移行列Tは、

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-\lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-\mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = I + A \cdot B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-\lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-\lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

となる。このアルゴリズムでは各トランジションの入出力プレースがただ一つであるようなコンカレント動作のない「状態機械」となっているため、発火度合行列Bの各列には当然入力プレースが2つ以上は現れない。

5.2.7 あいまい性評価シミュレーション

前節までの行動決定ファジアルゴリズムのペトリネット表現を用いて、あいまい性評価のための計算機シミュレーションを行った。初期マーキングは $M_0 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$ として、アルゴリズムの流れを左右している重要なプレース (p1, p3, p4, p5, p6, p9, p11) のみに着目してこれらにおけるあいまい度 (グレード値) のトレンドをグラフにした (図5.2.6a)~d)参照)。ファジィ分岐におけるルール前件部のマッチ度 (成立グレード値) は実際の対象システムから得られるため、このシミュレーションにおいては r (p3の成立グレード値) $= j/jend$, s (p5の成立グレード値) $= 1 - j/jend$ [j : 発火回数, $jend$: 全発火回数] と単純な変化のみを想定した。

さまざまな条件における種々のシミュレーションが考えられるが、ここでは以下の4つのケースについて評価を行った。

- ケース1: 単一発火 (スレシヨールド固定)
- ケース2: 並列発火
- ケース3: 単一発火 (スレシヨールド制御)
- ケース4: 単一/並列発火 (グレード差選択)

ケース1では、固定スレシヨールドを $\lambda = 0.7$, $\mu = 0.5$ とした。例えば、p3の成立グレード値 $r > \lambda$ のとき、(p3のトークン値) $\times r$ でトランジション t_3 が発火する。周期的にグレードが変化しているのはネットのループを巡回するトークンがまとまって流れているためである。このグレードの山が徐々に下がっていくのはあいまい度が次第にアルゴリズム全体に均等に分散していく模様を示している。この周期的なグレードの変化は発火30回目あたりでなくなっており、p3が減少すると同時にp4とp5が上がっている。これはp3とp5の成立グレード値を適当に単調増加、減少関数としたために25回目あたりからp3のランチ (t_8) からp5のランチ (t_{12}) へとアルゴリズムの流れが変わっていることを示している。

ケース2では、分岐において両方とも並列に発火させた場合の結果で、当然のことながらあいまい度がいはやく分散しており、トークン状態も周期的ではあるが安定して推移している。これは並列発火のためにトークンの分散が早く、システム全体にあいまい性が浸透している状態 (爆発) と考えられる。

ケース3では、デフォルトのスレシヨールド値を中心に λ をSIN関数で、 μ をCOS関数でそれぞれ仮想的に振動制御した場合の単一発火の様子を示している。ケース1と似ているが、スレシヨールド制御を行っている分だけシステムが変動的に挙動していることを示しており、若干グレード値も大きめに変化している。これはスレシヨールド制御を行うことにより、あいまい性の拡散を並列発火よりも大幅に、単一発火と比べてもさらに少し減少させる効果があるということがわかる。

ケース4では、単一発火と並列発火の混在型の一例として、5.2.3節のD)の分岐実行法を用いてシミュレーションを行った。ここでは成立グレード値が $\lambda \pm 0.2$, $\mu \pm 0.2$ 以内のときはそれぞれ並列発火を、それ以外のときは分岐のうち大きなグレードをもつもののみを単一発火させてみた。この分岐制御は我々人間が日常行っている方法にかなり近いと考えられるが、発火の必要性の低いものと発火すべきものをうまく選別分岐するため、これらが分かれて収束している様子が見られる。

以上のように本手法を用いることにより、ファジアルゴリズムにおけるあいまい性の状態遷移をある程度把握することができた。これにより、どのようなアルゴリズムもペトリネット化ができれば、シミュレーションによりシステムのあいまいさの爆発を事前にチェックすることが可能となる。

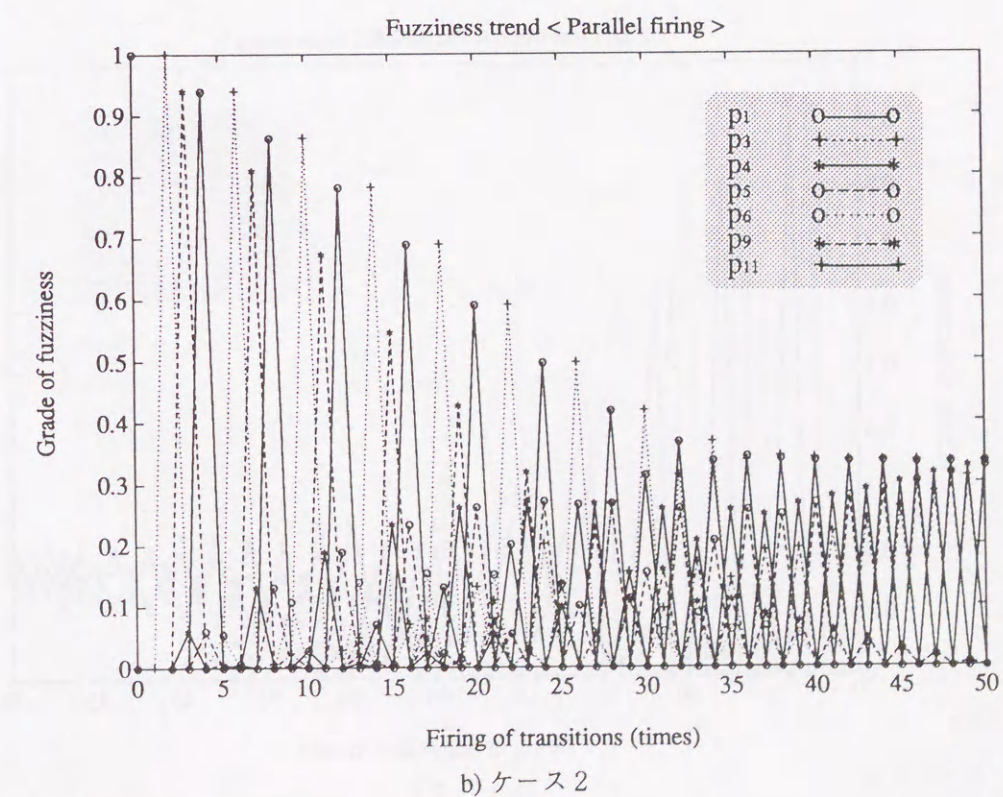
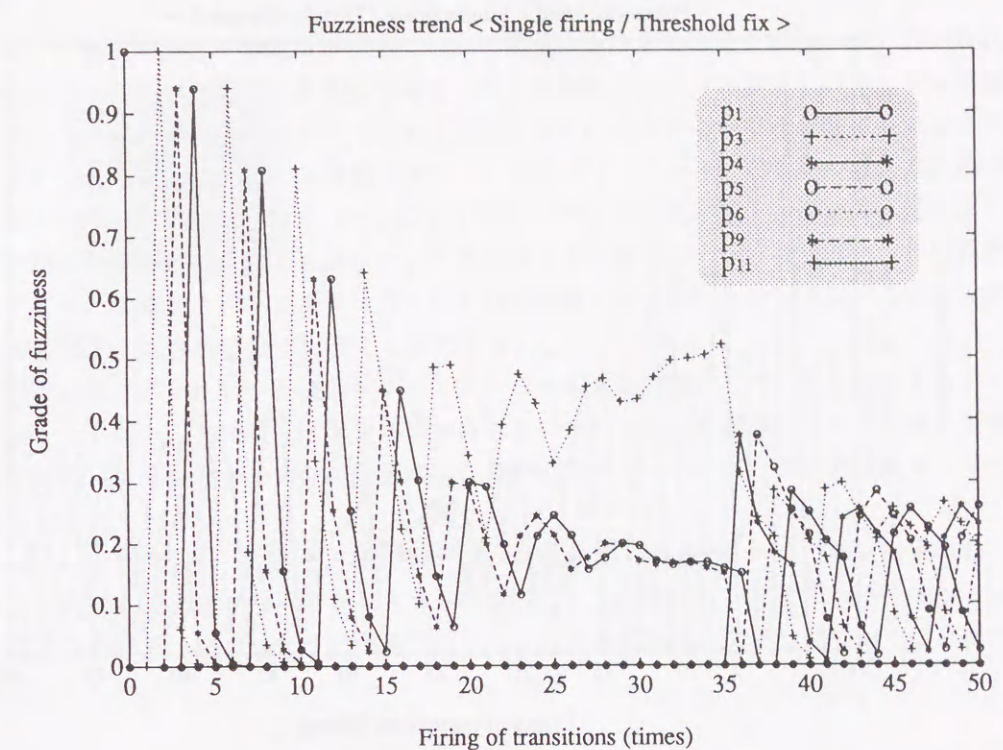
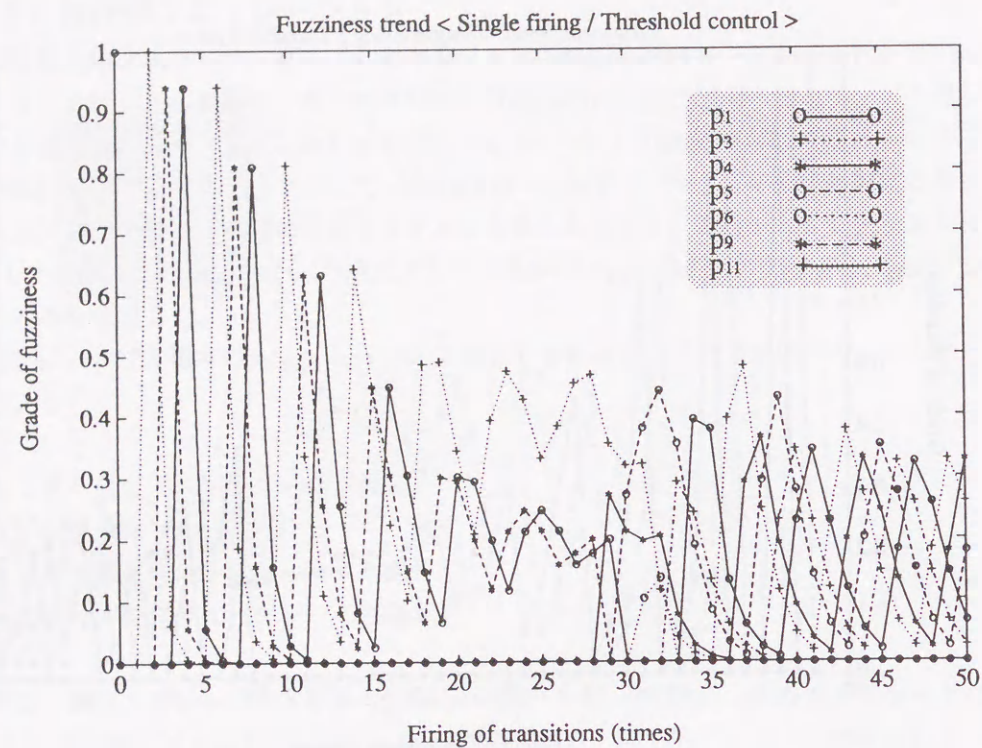
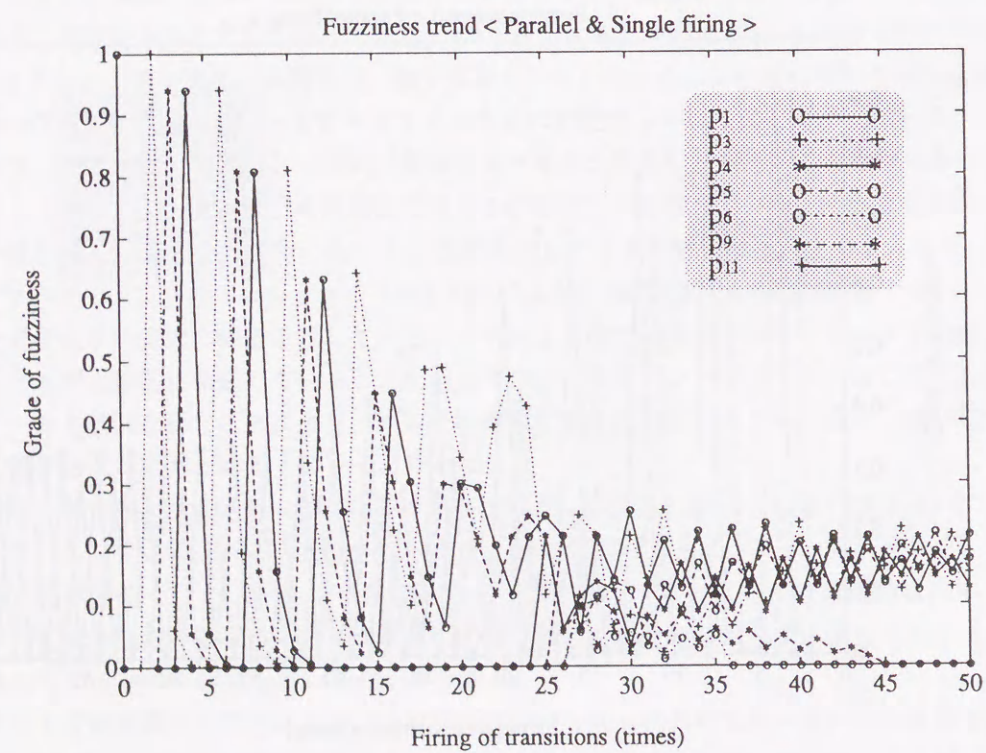


図5.2.6 シミュレーション結果 (1)

5.2 ペトリネットを利用したファジアルゴリズムのあいまい性評価



c) ケース 3



d) ケース 4

図5.2.6 シミュレーション結果 (2)

5.2 ペトリネットを利用したファジアルゴリズムのあいまい性評価

5.2.8 結 言

「人間らしい」行動決定機能を知能ロボットにおいて実現することを考えた場合、あいまいな状況認識を含む行動シーケンスを表現できることが重要である。そこで筆者らはクリस्पおよびファジ演算が混在するようなシーケンスフローが表現可能なファジアルゴリズムの概念を用いて、人間に近いマクロ行動決定アルゴリズムをすでに提案している。本研究では、ファジィペトリネット (FPN) の考え方にに基づき、ファジアルゴリズムのフロー全体の表現を行い、これを用いてファジィ真理値トークンの伝搬によりシステムのあいまい性の推移状況を捉えることを試みた。本手法はペトリネットの状態表現能力、状態遷移解析能力を用いることにより、従来ファジアルゴリズムにおいて困難であったシーケンスルールの多段発火におけるあいまい性の客観的・総合的評価を行うことができる。

また前節の自律移動ロボットの行動決定ファジアルゴリズムを例題として、本手法によるあいまい性評価シミュレーションを行い、アルゴリズム全体におけるあいまいさの遷移状況がかなり正確に把握できることがわかった。これにより、あいまいさの爆発が起こらないファジアルゴリズムの設計やシステム挙動解析が可能になると考えられる。

さらに本手法は、IF~THEN~ELSE~の2分岐から一般Case文 (多分岐) への拡張、発火時刻を考慮した時間ペトリネットの導入によるより正確なシステム挙動解析、あいまい性を有するペトリネットへのネットインバリエントを用いたデッドロックや可到達性などのシステム解析へと発展させていくことが可能であると考えられる。

第6章 知的行動決定アルゴリズム構築用ファジィシェルに関する研究

6.1 緒言

ファジィ推論を伴うプログラミングにおいては、通常メンバーシップ関数のデータ共用管理の煩雑さ、常に同じ推論エンジンをサブルーチン記述する冗長性、などが問題になる。このような場合、ファジィ推論機能を有するソフトウェアツール（以後、ファジィシェルと呼ぶ）を用いることにより、プログラミング効率を上げることが可能になる。このため最近では、安信らのルールベース型推論とファジィ推論を融合したもの[1]、山崎らの黑板モデルを用いた多段型のもの[2]、W.Silerらの並列帰納推論が可能なもの[3]、三好、小山らのさまざまなあいまいさの取扱いが可能なもの[4]、山本らのオブジェクト指向の概念を取り入れたもの[5]などいくつかのファジィシェルが発表されている。しかしながら一般に従来のファジィ推論をサポートしているエキスパートシステムはマルチウィンドウによるルールエディタ等のユーザインタフェース機能は比較的高いが、シェル上でルール起動するためユーザプログラムとのリンクが困難であったり、シェルシステムが大規模なため推論スピードが遅くリアルタイム制御に不向きであるなどの問題点があった。

そこで筆者らは前述のさまざまな知的行動決定アルゴリズムを記述するためのエキスパートツールとして、汎用的にも利用可能なリアルタイム知的制御用のユーザプログラム主導型ファジィシェルFRASH (Fuzzy Real-time Advanced SHell) を開発した[6][7][8]。まずライブラリによるリアルタイムチューニングが可能なファジィ推論機能をVer1.0 (FRASH1) で実現した。その後、データおよびルール表現機能を強化するため、ファジィ数をスロット値として定義できるフレーム型データ構造のエディタ機能およびファジィアルゴリズムのマクロコードが記述可能なC言語のソースコードジェネレータをもつコンパイラ機能を追加したVer2.0 (FRASH2) を開発した。本論文では、FRASHの基本構成並びにファジィ推論、ファジィフレーム、ファジィアルゴリズムの各機能について述べ、最後にこれらの機能検証のため車の追尾/追越制御を想定した簡単な推論シミュレーションを行なったのでこれについても報告する。

6.2 知的制御用ファジィシェルFRASHの開発

6.2.1 基本機能

FRASHはもともと筆者らがロボット等の比較的高速なメカトロ系リアルタイム制御用として開発したファジィシェルである[9]。よって、オートチューニング、高速ファジィ推論、複雑なファジィデータ構造等に柔軟に対応できるように設計されている。FRASHはSUN4/UNIX-C上でOpen-Windowsをベースに開発されており、ライブラリを核とするユーザプログラム主導を基本思想としている。

FRASHは主に以下のような機能をもっている。

FRASH1 (基本機能)

- 高速推論が可能なライブラリ形式のファジィ推論エンジン
- リアルタイムチューニング可能なライブラリ形式のオンラインルールエディタ
- マルチウィンドウによるユーザ・フレンドリなオフラインルールエディタ
- ルールデバッグが容易なリアルタイム推論状況表示機能

FRASH2 (追加機能)

- ファジィ数をスロット値として定義できるフレーム型データ構造
- ファジィアルゴリズムのマクロコードが記述可能なソースコードジェネレータ

従来の推論シェル起動タイプの場合、ユーザがすでに作成しているプログラムの処理をシェルに移植してすべてシェル内で開発作業を行なう必要があった。これに対し、上記のa), b)はこのシェルの大きな特長で、ユーザプログラムからライブラリコールによりファジィ推論やルールチューニングができる仕様となっている。これにより、開発が容易になるだけでなく、シェルの機能をライブラリ単位で分割管理するため推論が高速になる。さらにc), d)のマルチウィンドウ機能により、ルールおよびメンバーシップ関数入力時の煩雑さの軽減やルールデバッグ時の作業性向上も同時に実現している。

さらにFRASH2に関しては、e)においてデータの属性に基づく階層記述をサポートしており、スロット値において通常のクリस्प数だけでなく、ファジィ数も扱えるようなフレーム型データ構造（以後、ファジィフレームと呼ぶ）を導入し、データ表現能力および知識のメンテナンス性をさらに高めた。ルール記述に関してもf)のC言語のソースコードジェネレータにより、限定されたファジィアルゴリズム (Fuzzy Algorithm) [10][11]の機能をマクロコード記述でサポートしている（以後、ファジィアルゴリズムコンパイラと呼ぶ）。この機能により従来のファジィ制御における推論記述のみならず、意思決定レベルを含む幅広い制御アルゴリズムの表現に対応できるようになった。

6.2.2 基本構成

FRASHの基本構成を図6.2.1に示す。本システムは、以下の4つの主要プロセスから成っている。

- 1) スケジューラ
- 2) エディタプロセス (ルール/フレームエディタ)
- 3) 推論プロセス (推論エンジン/オンラインエディタ)
- 4) 推論状況表示プロセス

シェルを起動するとまずスケジューラのモードに入り、ここから推論、エディタ、推論状況表示の各プロセスを選択でき、それぞれのプロセスを単独にまたは同時に動作させることが可能である。

エディタプロセスはユーザがオフラインで推論に関するすべてのデータ（ルール群、ルール、前件部変数、後件部変数、ファジィレベル、メンバーシップ）をマウスおよびグラフィックス入力できる。推論プロセス中のオンラインエディタは推論中に上記のすべてのデータがリアルタイム修正可能なチューニング用ライブラリを提供する。推論エンジンはMIN-

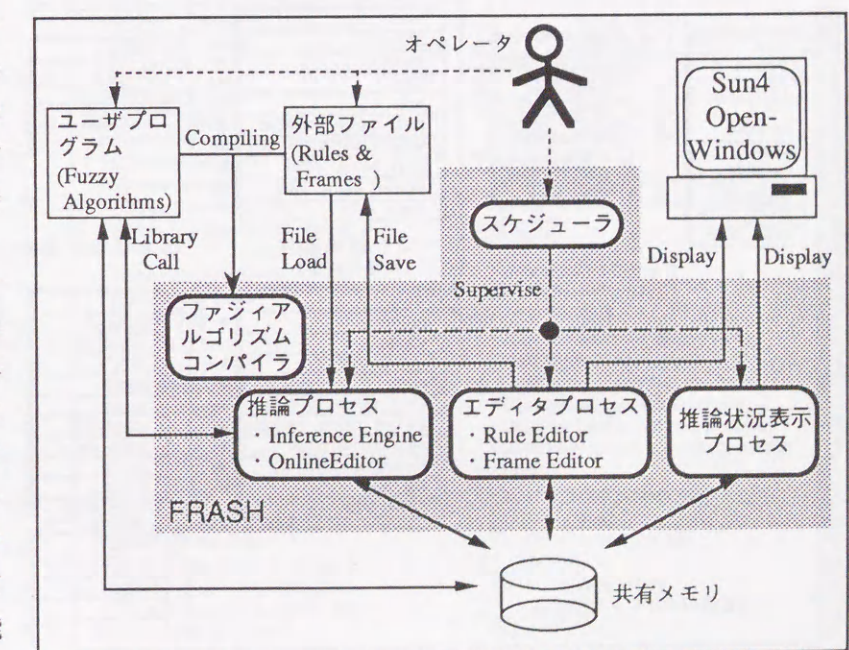


図6.2.1 FRASHの基本構成

MAX重心法 (Mamdani法)、簡易型ファジィ推論法 (Singleton法)、高木・菅野モデルに基づく推論法 (T-S Model法) などの代表的な推論方法を標準サポートしている。入力クリスプ、ファジィの両方が取り扱い可能である。

ユーザはまずはじめにエディタプロセスに入り、マルチウィンドウおよびグラフィック入力機能によりルール等を生成する。これを外部ファイルにセーブした場合、ユーザプログラムからルールに関するすべてのデータがロードできる。もちろんユーザプログラム内でメンバーシップ関数やルール等を直接定義するライブラリも用意されており、どちらを用いてもよい。さらにユーザは自己のプログラム内でライブラリコール (C言語ライブラリ) を記述することにより、推論操作 (推論エンジン)、オンラインチューニング操作 (オンラインエディタ) を実現できる。シェルの構成の中では推論プロセスの機能のみがすべてライブラリ形式で提供されており、ユーザプログラムからコールできるようになっている。

3つのプロセスはともに共有メモリを参照して情報を入出力している。オンラインでユーザが推論状況表示プロセスをセレクトした場合、この内部データのうちメンバーシップ関数 (合成メンバーシップ関数も含む)、ファジィラベル、ルール、などの推論中のリアルタイム値をディスプレイに表示することができる。ユーザはこの機能を利用して、推論結果をモニタしながらオンラインでルールのデバッグを効率よく行なうことができる。このプロセスは推論状況をウィンドウまたはファイルにテキスト形式

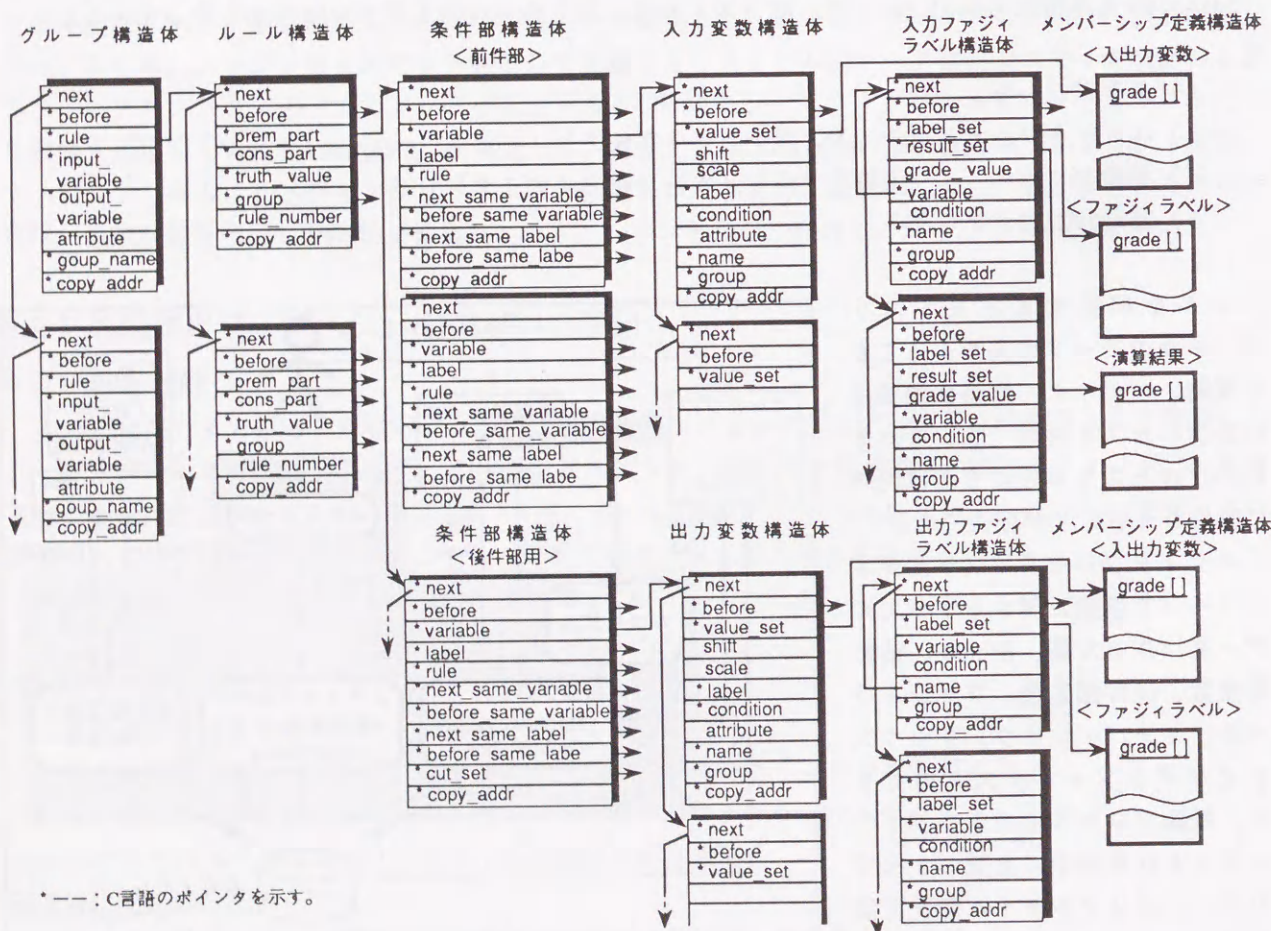


図6.2.2 FRASH1のルールデータ構造 (Mamdani法/ファジィ入力)

で出力するトレース機能、およびルール、入力変数、出力変数の内容をグラフィカルに表示するモニタ機能からなる。尚、共有メモリ中のすべてのデータは基本的に読み書き可能なため、ユーザプログラムの中で自在に加工、表示を行なうことができる。

6.2.3 ルールデータ構造

FRASH1はUNIX-Cで記述されており、すべてのデータを構造体形式で保有しており、それらの関係はポインタにより管理されている。このデータ構造により、高速でかつメモリ効率の良いリアルタイムシェルを実現している。図6.2.2にMamdani法/ファジィ入力の場合のファジィ推論ルールの構造体を示した。ルールデータ構造は推論方式により異なる構成が用意されている。メンバーシップのデータについては構造体の内容を、入出力変数およびファジィラベルについては構造体のポインタを変更すれば容易に修正が可能である。

FRASH2ではこのルールデータ構造の他にフレームデータ構造をもっている (6.2.5節参照)。この場合のルール構造は、フレーム構造で定義されるスロット値やラベルに大きく依存しているので、ルール構造を定義した後にフレーム内のオブジェクトを生成/削除する場合は注意を要する。

6.2.4 ファジィ推論機能

FRASH1は図6.2.3に示すようにファジィ推論に関する約80の基本ライブラリをもっている。ファジィ推論エンジンもこのライブラリの中に機能として組み込まれている。通常ファジィルールはこれらのライブラリを組み合わせて用いることにより自在に記述、修正、チューニング、推論、デファジファイ等を行なうことができる。これに加えてFRASH2ではフレーム構造の生成・変更等の機能を提供するファジィフレームの編集機能に関するライブラリが新たに追加されている。シェルを使用して記述されたユーザプログラムは全

<p>領域初期化関数</p> <ul style="list-style-type: none"> FzAllocCell (cell_max) FzAllocGrade (grade_max) FzAllocName (name_max) <p>推論演算関数</p> <ul style="list-style-type: none"> FzCalcCutSet (rule, calc) FzCalcItem (label) FzCalcLabel (variable, label) FzCalcTruth (rule) FzCalcVariable (group) <p>データ複写関数</p> <ul style="list-style-type: none"> FzCopyBufferCS (cons_part, buffer) FzCopyBufferLR (label, buffer) FzCopyBufferLS (label, buffer) FzCopyBufferVS (variable, buffer) FzCopyLabelResult (buffer, label) FzCopyLabelSet (buffer, label) FzCopyValueSet (buffer, variable) FzCopyCutSet (buffer, cons_part) FzCopyGrade (before, after) <p>構造体複写関数</p> <ul style="list-style-type: none"> FzCopyGroup (old_group, group_name) FzCopyRule (old_rule, group, rule_number) FzCopyPremPart (old_cond, rule) FzCopyConsPart (old_cond, rule) FzCopyVariable (old_variable, group, name) FzCopyLabel (old_label, variable, name) FzCopyItem (old_item, label) <p>構造体初期化関数</p> <ul style="list-style-type: none"> FzCreateGroup (group_name, attribute) FzCreateRule (group, rule_no) FzCreatePremPart (rule) FzCreateConsPart (rule) FzCreateVariable (group, name, shift, scale, flag) FzCreateLabel (variable, name) FzCreateItem (label) <p>構造体削除関数</p> <ul style="list-style-type: none"> FzDeleteGroup (group) FzDeleteRule (rule) FzDeletePremPart (prem_part) FzDeleteConsPart (cons_part) FzDeleteVariable (variable) FzDeleteLabel (label) FzDeleteItem (item) <p>構造体破壊関数</p> <ul style="list-style-type: none"> FzDestroyGroup (group) FzDestroyRule (rule) FzDestroyPremPart (prem_part) FzDestroyConsPart (cons_part) FzDestroyVariable (variable, flag) FzDestroyLabel (label) 	<p>構造体返却関数</p> <ul style="list-style-type: none"> FzFreeCell (cellp) FzFreeGrade (gradep) FzFreeName (namep) <p>構造体確保関数</p> <ul style="list-style-type: none"> FzGetCell () FzGetGrade () FzGetName () <p>構造体検索関数</p> <ul style="list-style-type: none"> FzQueryGroup (group_name) FzQueryRule (group, rule_number) FzQueryPremPart (rule, variable, label) FzQueryConsPart (rule, variable, label) FzQueryVariable (group, name, flag) FzQueryLabel (variable, name) <p>推論実行関数</p> <ul style="list-style-type: none"> FzInference (group_name, calc) <p>初期設定関数</p> <ul style="list-style-type: none"> FzInit (cell_max, grade_max, name_max, membersize, argc, argv) FzInitC (cell_max, grade_max, name_max) <p>メンバーシップデータ変換関数</p> <ul style="list-style-type: none"> FzDigitize (n, x0, y0, x1, y1, shift, scale, grade) <p>メンバーシップ領域初期化関数</p> <ul style="list-style-type: none"> FzNullGrade (grade) <p>ファイル関連関数</p> <ul style="list-style-type: none"> FzRuleLoad (file_name) FzRuleSave (file_name) <p>ルール作成関数</p> <ul style="list-style-type: none"> FzRuleSetIf (variable, label, prem) FzRuleSetThen (variable, label, cons) <p>同期関数</p> <ul style="list-style-type: none"> FzAllowShmWhite () FzSendLaunchSignal () FzSendPremSignal () <p>データセット関数</p> <ul style="list-style-type: none"> FzSetItem (variable, factor, item) FzSetPoints (Gdata, label) <p>カット関数</p> <ul style="list-style-type: none"> FzCut (grade, cut_value, cutset) FzMultiply (grade, cut_value, cutset) FzSiftDown (grade, cut_value, cutset) <p>推論関数</p> <ul style="list-style-type: none"> FzComposition (group, calc) FzEasy (group) FzTakagiSugeno (group) <p>重心演算関数</p> <ul style="list-style-type: none"> FzGc (variable, value) <p>エラー出力関数</p> <ul style="list-style-type: none"> FzPerror (string)
--	--

図6.2.3 FRASH1のライブラリ関数

6.2 知的制御用ファジィシェルFRASHの開発

てこのライブラリをリンクしなくてはならない。

FRASH2ではファジィ推論を行なうユーザープログラムは、

- 1) ファジィ推論ライブラリを初期化する
- 2) フレーム構造を作る
- 3) ルール構造を作る
- 4) 無限ループ内で推論ルール群を発火させる

という一定の形式で記述する。またファジィ推論を実行するには、データ入力、ルール群の発火、データ出力の3つを無限ループ内で実行させる。

6.2.5 ファジィフレーム機能

FRASH2では新たに、属性のスロット値としてファジィ数を扱うことができるファジィフレームの表現機能が付加されたため、データ構造の階層化やあいまい知識表現が可能となった。ファジィフレーム機能はファジィ推論およびファジィアルゴリズムで共用することができる。FRASH2のフレームによるデータ表現は、一般的なフレームと同様、クラスの階層表現、インスタンスに対する属性スロットの継承機能等を持つ。

(a) ファジィフレーム構造

FRASH2に於けるフレーム構造は大別すると、クラス、スロット（属性に相当する）、インスタンス、スロットバリュー、ラベルの5つの構成要素からなる（図6.2.4参照）。クラスはインスタンス群を定義し、クラスのもつスロットをテンプレートとしてインスタンスが定義される。またクラス同士は階層的に定義することができ、子クラス内のインスタンスはその上位にあたる親クラスで定義されたスロットを全て継承する。スロットはクラス内の各インスタンス間で共通なスロットバリューの属性、推論属性、ラベル等を定義する。

ファジィフレームのクラスにはスタティッククラスとダイナミッククラスの2種類がある。スタティッククラスはクラス宣言のデフォルト

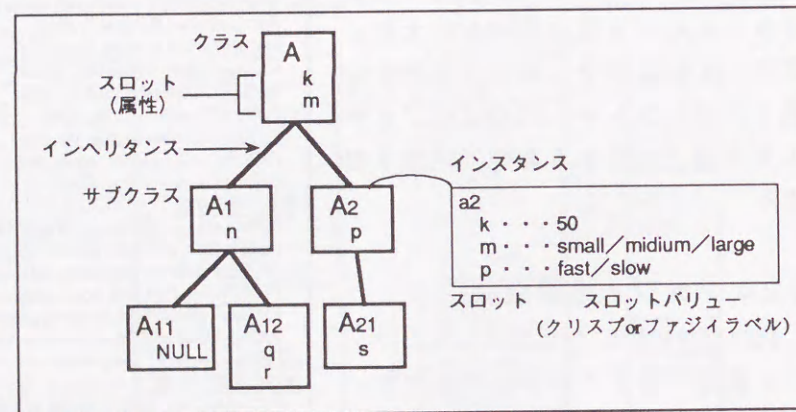


図6.2.4 ファジィフレームの構造

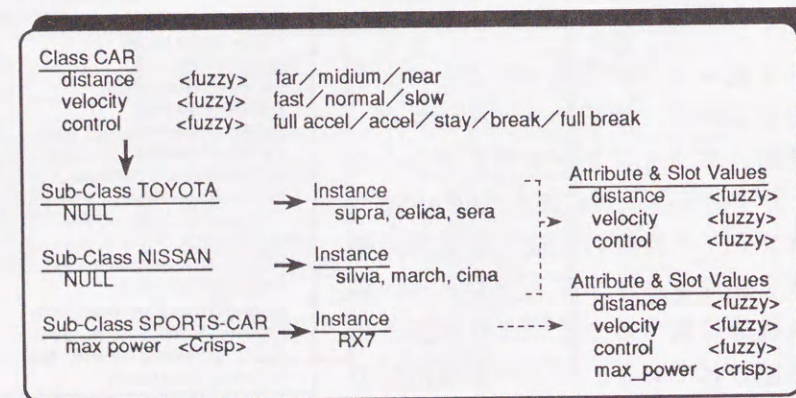


図6.2.5 ファジィフレームの記述例

6.2 知的制御用ファジィシェルFRASHの開発

で、推論処理中にスロットの変更を許さない代わりに、比較的高速に動作する。ダイナミッククラスは推論処理中にファジィスロットの追加、削除、名称変更を行なうことができる。

(b) ファジィフレームエディタ

このファジィフレームでデータを作成する場合、ユーザはまずクラスおよびサブクラスの属性（スロット）定義を行なう（図6.2.5参照）。クラスにおける属性について、ユーザはクリスプまたはファジィのどちらかが選択できる。クラスのスロット定義を終えると、各クラスに対してインスタンスを定義することができ、それぞれのスロットに対してはスロットバリュー（具体的なメンバーシップで定義されたファジィラベルや数値に相当）が与えられる。FRASH2のファジィフレームエディタではファジィデータの効率的利用を考慮して、各スロットに対して推論属性やラベル属性が独立して定義できるようになっている。またインスタンスがもつスロットバリューは、クラスで定義されているスロットのファジィラベル（または数値）を自動的に継承するが、それを削除することも、新たに追加することも可能である。

FRASH2ではスロットの属性がファジィ数である場合、メンバーシップ関数はスロット自身をクリックすることにより、グラフィック関数エディタのウィンドウが現われ、ここで自在な形状をマウスのみで容易に作成できるようになっている。尚、FRASHにおける全プロセスは共有メモリを用いているため、エディタで修正されたこのフレームデータは同時にオンラインで他のプロセスにおいて有効となる。

6.2.6 ファジィアルゴリズム機能

人間に近いマクロ判断や意思決定アルゴリズムが柔軟に記述できる方法論という立場に立った場合、従来のファジィ制御で用いられているMIN-MAX重心合成に基づく推論方法だけではアルゴリズム記述能力が十分ではない。そこでクリスプ演算およびファジィ演算が混在するようなシーケンスフローが表現可能なファジィアルゴリズムを用いると、このような意思決定レベルの制御則の表現には有効であることが知られている。FRASH2ではこのファジィアルゴリズムの機能をサポートしており、知識表現能力が大幅に拡大した。

(a) 修正ファジィアルゴリズム

ファジィアルゴリズムを用いると、例えば、" IF x is small and y is large THEN z is very small ELSE z is not small. " のようなファジィ変数を含む極めて自然な表現形式でアルゴリズムを記述できる。このように、一般に知られているIF...THEN...で表現された複数のファジィルールを用いて推論を行うファジィ制御と異なり、IF...THEN...ELSE...のファジィ分岐を許しているため、柔軟にシーケンスアルゴリズムを記述することができる。

本来のファジィアルゴリズムでは分岐後の処理を並列発火（ファジィ分岐）し、それぞれ前件部グレードの重み（あいまいさ）を継承する形で後件部を実行することが提案されている。ところが、この方法で実際のアルゴリズムを実行すると並列処理が増大し、やがてアルゴリズムフローが爆発するという問題がある。

一般にファジィアルゴリズムにおける分岐処理には以下の3つのレベルが考えられる。

6.2 知的制御用ファジィシェルFRASHの開発

- a) 分岐判断 (前件部) はクリスプ演算で、
分岐実行 (後件部) は単一発火
- b) 分岐判断 (前件部) はファジィ演算で、
分岐実行 (後件部) は単一発火
- c) 分岐判断 (前件部) はファジィ演算で、
分岐実行 (後件部) は並列発火

通常のAIプロダクションルールでは、並列発火のELSE分岐を許していないので、a)のみを取り扱えるだけであるが、ファジィアルゴリズムの場合は、原理的にb)およびc)のレベルを包含しており、判断における悩みや複数の可能性検討など人間の意思決定に近い記述が可能となる。しかしながら、c)のレベルのような本来のファジィアルゴリズムを表現しようとすると、分岐が無数に増加する上、ルール中にGOTO文やCALL文などを記述すると一般にアルゴリズムが爆発する。

そこで筆者らはb)とc)の中間レベルの一手法として、前件部の成立グレード値にスレシヨールドを設け、この値によりファジィ分岐をコントロールするファジィアルゴリズムの簡易的な実現方法 (修正ファジィアルゴリズムと呼ぶ) を提案した[12][13][14]。図6.2.6はこの制御方式の概念図を示している。前件部がクリスプ判断のみで表現されている場合、後件部はマッチ度1で実行される (THEN部は"yes"、ELSE部は"no")。しかしながら、前件部にファジィ判断を含んでいる場合には、THEN部は前件部のマッチ度がスレシヨールド値 (図では μ_1) を越えたときのみ実行され、その他のときはELSE部が実行される。本方式ではスレシヨールド値自身は上位レベルの制御などにより可変で、そのときの状況によりアルゴリズムが微妙にチューニングされるような知識表現が可能となる。

さらにこの方法では、前件部のマッチ度がmである場合には、THEN部は成立度合mで、ELSE部は成立度合(1-m)でどちらかが選択実行されるため、比較的妥当な発火を行なうシーケンスルールを記述できる。この場合、連続意思決定において多段で推論を繰り返した場合マッチ度が急激に減少するケースが考えられるが、FRASH2ではこれを防止するための後件部の成立度合決定方法もユーザが指定できるようになっている。

FRASH2ではこの修正ファジィアルゴリズムに限定したファジィアルゴリズムマクロ記述を実現するため、C言語のソースコードジェネレータを搭載する方法を採用した。

(b) ファジィアルゴリズムのマクロ記述

ファジィアルゴリズムを記述する文法は、C言語の拡張セットになっており、C言語のソース中にファ

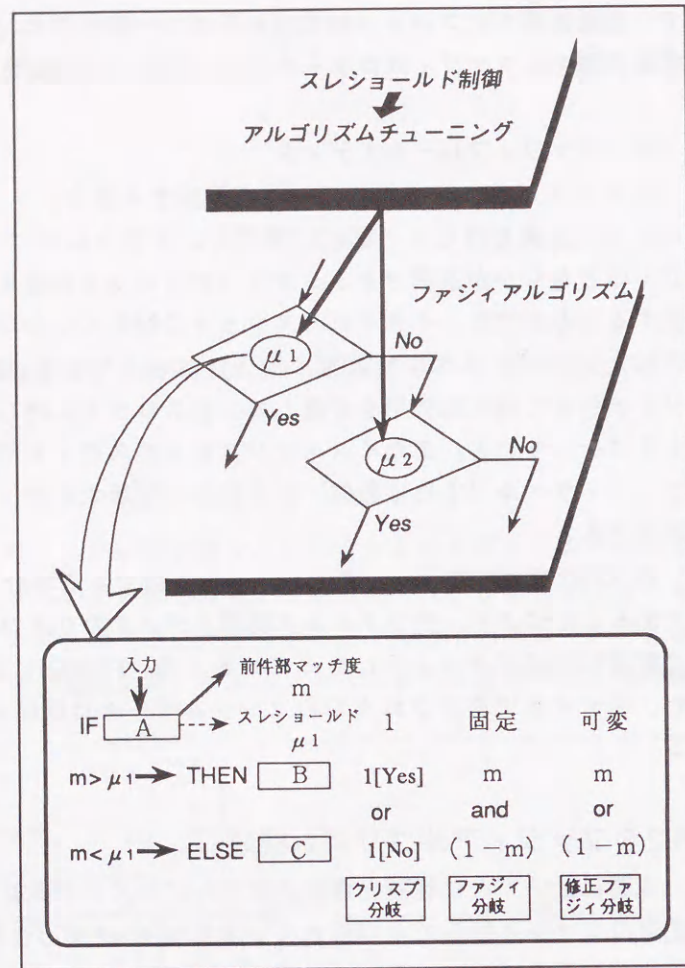


図6.2.6 修正ファジィアルゴリズムの概念図

6.2 知的制御用ファジィシェルFRASHの開発

ジアルゴリズムのフレーム記述、ルール記述を挿入する形になっている。これ以外の部分は通常のC言語であると理解し、コンパイラは無視する。ファジィアルゴリズムソースのマクロ記述例を図6.2.7に示す。アルゴリズム中で用いられるフレームデータ、変数、ファジィラベルはすべてオフラインエディタによりあらかじめ入力される。ソースコードジェネレータによりすべてのデータが自動的に定義されるのでアルゴリズム中では一切データに関する宣言は行なう必要がない。またローカルにアルゴリズムで用いられオンラインで修正する可能性のある動的変数 (ダイナミッククラス) はマクロコードの最初 ("DECLARE begin" から "DECLARE end" の間) に定義しておく。

ファジィアルゴリズムの場合、通常のファジィ推論と異なり、ルール中にファジィ演算とクリスプ演算が混在するが、これらについては [] で節ごとの記述範囲を示すことにより、識別できるようにしている。ファジィアルゴリズムのルールは "RULE begin" と "RULE end" の間にマクロ記述をする。ルール前件部は、スレシヨールド指定、マッチ度演算関数指定、条件文記述の3つのフィールドからなる。図6.2.8にこのうち条件文記述フィールドの書式を示す。スレシヨールド指定フィールドには、そのルールで使用するスレシヨールド値をfloat値で記述する。指定がない場合は前件部マッチ度のデフォルト値Thresholdの値が使用される。これをユーザプログラム中で変動させることにより微妙なアルゴリズムの分岐制御が可能となる。マッチ度演算関数指定フィールド (図6.2.7では

```
#include "Fza.h"
#include "TestData.fzr"

DECLARE begin
    Dynamic Class SPORTS_CAR:CAR { None distance; Crisp max_power; };
    Declare SPORTS_CAR RX7;
DECLARE end

int get_value ();

main (argc, argv)
    int argc;
    char* argv [];
{
    FzFrameClass* SPORTS_CAR;

    FZA_INIT

    SPORTS_CAR = FzDefineClass (FzQueryClass ("CAR"), "SPORTS_CAR", NULL);
    FzUnDefineSlot (SPORTS_CAR, "distance");
    FzInstantiate (SPORTS_CAR, "RX7");

    FZA_INIT_INSTANCE

    RULE begin
        Crisp i = 0.9;
        Threshold = 0.68;

    START:
        IF (, [[silvia.distance] is near] and [[RX7.max_power] > [silvia.distance] * 2 + 5])
        { [[silvia.control] is stay};
          i = [RX7.velocity];
          get_value ();
        ELSE
        { IF (0.5, [[sera.distance] is near] and [[supra.velocity] > [sera.distance] * 2 + 5])
          { [[supra.control] is "full break"];
            IF (, [[march.distance] is far]) [[cima.control] is accel];
          ELSE
          { IF (, MATCH_FUNC_2, [[celica.velocity] is fast])
            { i = i * 1.8;
              goto START; } }
          }
        }
    RULE end
}
```

図6.2.7 ファジィアルゴリズムのマクロ記述例

```
ルール前件部書式
IF (<スレシヨールド指定>, <マッチ度演算関数指定>, <条件文記述>) THEN ...

条件文記述フィールド書式
スロットバリュウ名      := [<インスタンス名>.<スロット名>]
ファジィ演算子         := [is]
クリスプ演算子         := [+ | - | * | /]
クリスプ比較子         := [= | < | > | <= | >= | !=]
クリスプアトム         := [<スロットバリュウ名> | <浮動小数点値>]
クリスプ演算式         := [<クリスプアトム> [<クリスプ演算子> <クリスプアトム>] ]
条件文接続子           := [and | or]
ファジィ条件文         := [<スロットバリュウ名> <ファジィ演算子> <ラベル名>]
クリスプ条件文         := [<クリスプ演算式> <クリスプ比較子> <クリスプ演算式>]
条件文                 := [<ファジィ条件文> | <クリスプ条件文>]
条件文記述フィールド := [<条件文> [<条件文接続子> <条件文>] ]
```

図6.2.8 ファジィアルゴリズムの前件部書式

"MATCH_FUNC_2"など)では、分岐の際のルールマッチ度の演算方法を設定する関数を指定する。これによりファジィ分岐における分岐後のルールマッチ度の決定方法を変えることができる。

ルール後件部はルールの実行文であり、C言語ソースコード、ファジィ演算式、そしてルール文を記述することができる。これにより、CALL文・GOTO文による外部分岐、ルールの階層記述、ファジィ推論 (FRASH1の機能) の実行などが後件部で許されるため、かなり複雑なアルゴリズムを柔軟に記述することが可能となる。

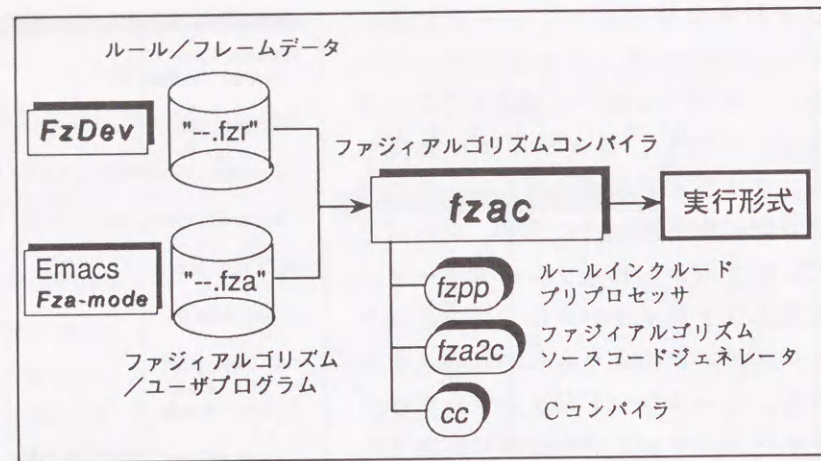


図6.2.9 FRASH2ファジィアルゴリズムコンパイラ

(c) ファジィアルゴリズムコンパイラ

図6.2.9はFRASH2のファジィアルゴリズムコンパイラの構成を示している。FRASH2のユーザーインターフェイスは、ファジィ推論用フレームデータやルールを作成するGUIプログラムFzDev、ファジィアルゴリズムソース編集用に機能拡張されたGNU Emacs用ファジィアルゴリズム編集モード Fza-mode、そしてファジィアルゴリズムソースをC言語に変換するファジィアルゴリズムコンパイラFzacからなる。これらのモジュールはGNU Emacsを核として有機的に統合されている。

ルールおよびフレームデータファイル ("---.fzr") はFzDevのオフラインエディタ (オンラインライブラリでも可) で作成され、ファジィアルゴリズムを含むユーザプログラム ("---.fza") はEmacsエディタのFza-modeモードで作成される。このFza-modeは、標準装備のc-modeに拡張機能を付加して、ファジィアルゴリズムソース編集用に改良されたものである。このモードを使用することにより、Emacsの自動インデント機能がファジィアルゴリズムソース編集中でも使用できるようになっている。また、FzDevやFzacをファジィアルゴリズム編集中に呼び出すこともできる。

これらのファイルはファジィアルゴリズムコンパイラFzacによってリンクされる。Fzacは、IF...THEN...ELSE...形式で記述するファジィアルゴリズムルールの文法に基づいたソースファイルをC言語に変換する。このコンパイラは、ルールデータファイルからフレームの宣言文を生成するプリプロセッサfzpp、ファジィアルゴリズムのソースコードを生成するための変換作業を自動的にこなすためのフロントエンドfza2c、実際の変換を行なうファジィアルゴリズムコンパイラccの3つのコマンドにより構成されている。これによりファジィアルゴリズムマクロコードを含むユーザプログラムがオフラインで作成されたルール/フレームデータとともに実行形式に変換される。

6.2.7 推論シミュレーション

FRASH2のエディタウィンドウの画面を図6.2.10に示す。図において、右下がトップウィンドウ、右上がフレームエディタウィンドウ、左下がルールエディタウィンドウで、左上がEMACSによるファジィアルゴリズムの編集画面である。ユーザはまずトップウィンドウでルールおよびフレームのエディタウィンドウを選択できる。

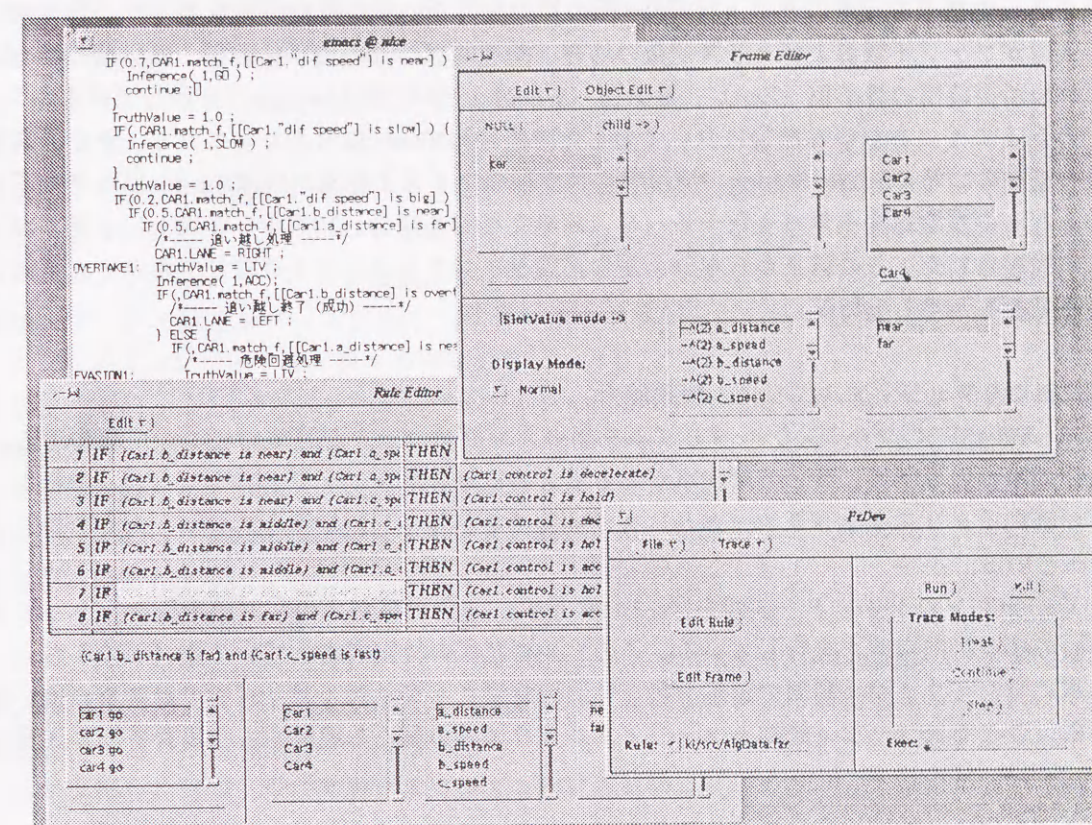


図6.2.10 FRASH2のエディタウィンドウ

以下では、FRASH1およびFRASH2を用いて行なわれた推論シミュレーションについて報告する。

(a) FRASH1のファジィ推論ベンチマークテスト

ここではおもにFRASH1のリアルタイム性を確認するために行なった推論速度のベンチマークテストの結果について述べる。シミュレーションは、簡単のため自動車の車間距離制御を例にとり、9つのファジィルールを用いてファジィ推論を行なった。このシミュレーションは2台の車のうち前方の車を一定速度で走行させ、後方の車をファジィ制御 (入力相対車間距離と相対速度、出力はアクセル開度) で前方の車に追従させるものである。表6.2.1にベン

表6.2.1 FRASH1の推論ベンチマークテストの結果

推論法	入力型	論理関数	Best (msec)		Worst (msec)		Average (msec)		推論速度(rule/sec)	
			SUN4	SS2	SUN4	SS2	SUN4	SS2	SUN4	SS2
Mamdani	Fuzzy	論理積	9.6	5.8	10.7	6.4	9.8	5.9	917.805	1514.642
		代数積	9.1	5.6	10.2	6.1	9.3	5.7	965.976	1573.702
		限界積	10.5	6.1	11.3	6.6	10.6	6.3	847.059	1433.578
	Crisp	論理積	6.2	3.7	7.2	4.3	6.4	3.8	1414.427	2343.140
		代数積	5.7	3.6	6.6	4.2	5.9	3.7	1529.572	2410.930
		限界積	6.9	4.1	7.7	4.6	7.0	4.2	1289.953	2144.900
Singleton	Fuzzy	3.6	2.1	4.6	2.5	3.8	2.1	2387.901	4221.388	
	Crisp	*	*	0.8	0.5	0.1	0.1	72580.645	128571.429	
T-S Model	Crisp	0.1	0.1	0.7	0.4	0.2	0.1	18913.043	76271.186	

*は計測限界以下 / SUN4: SUN4/370GX / SS2: Sparc Station 2

6.2 知的制御用ファジィシェルFRASHの開発

チマークテストの結果を示す。テストは10,000回の推論を行ない平均推論時間を出した。各推論方式に対して、入力ファジィ数およびクリップ数の場合、論理関数が論理積、代数積、限界積の場合についてそれぞれの所要時間の最小値 (Best)、最大値 (Worst)、平均値 (Average) を示している。

表よりわかるようにあらゆる推論において要した時間は約10msec以内となっており、かなり高速な推論を実現している。特に Sparc Station 2 使用時には今回のファジィ推論がほぼ数msec以内で完了しており、1秒あたりの推論速度換算でも数千ルール (簡略化推論の場合、10万ルール以上) の処理が可能であることが検証された。この結果から本シェルの推論ライブラリはメカトロのリアルタイム制御には十分使用可能であると考えられる。

(b) FRASH2のファジィアルゴリズム表現能力

ここでは、FRASH2のファジィアルゴリズムを用いたマクロコード記述能力について報告する。この例では前述の車の追尾制御において対向車がある場合の追越判断をファジィアルゴリズムで行なった。尚、各車の属性はクリップなパラメータとファジィラベルが混在したようなスロットをもつフレーム記述を行なっている。

シミュレーションのウィンドウ画面を図6.2.11に示す。4台の車が8の字無限走行を行っており、各2台づつの車が対向車線を走行している。このアルゴリズムで制御されている車 (このうちの1台) は前方の車に対してファジィ制御で追尾を行ない、同時にファジィアルゴリズムで対向車の追い越し動作を行なう。ここではファジィ制御とファジィアルゴリズムの異なる戦略をもつ場合をシミュレートし

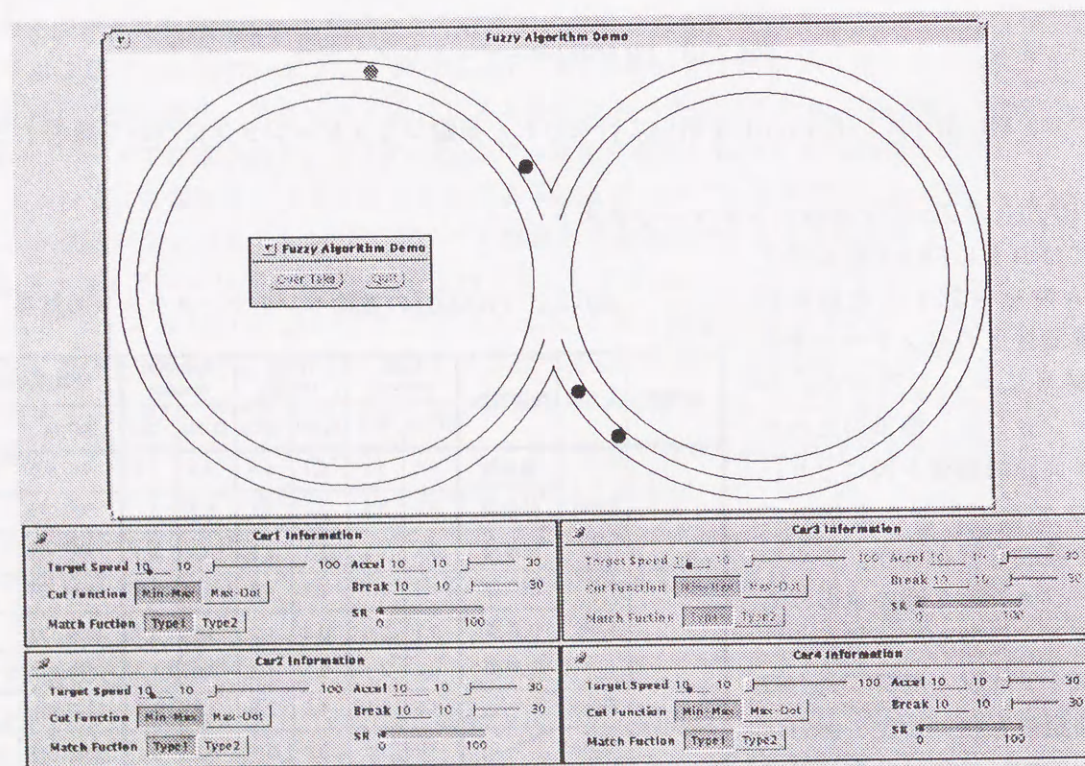


図6.2.11 追尾/追越制御のシミュレーション画面

6.3 結 言

たが、制御性能は極めて良好であった。これにより、FRASH2がマクロ意思決定を含んだような複雑な問題に対しても有効かつ効果的であることが示された。

6.3 結 言

本研究では、筆者らの開発したユーザプログラム主導型の知的制御用ファジィシェルの基本構成並びに表現機能について述べた。このシェルはライブラリ形式の高速な推論エンジン、推論におけるすべてのデータのオンラインチューニング機能、マルチウィンドウによるオフラインエディタ機能、推論状況表示機能、ファジィ数をスロット値として定義可能なファジィフレーム表現、マクロ記述によるファジィアルゴリズムコンパイラなど数々の特長を有している。

このシェルを用いると、従来のシェルではファジィ推論のみであったルール表現がファジィアルゴリズム機能により、AIプロダクションルールに近い多段のシーケンスアルゴリズムも記述可能となり、ファジィ情報処理としての幅広い利用が期待できる。またファジィフレーム機能によりデータの属性値の記述が困難であった従来手法に対し、あいまい属性をもつ階層的なデータ構造による効率的かつ強力な知識表現が可能となる。

またシェルの基本機能を確認するため、車の追従/追越制御を例題として簡単なシミュレーションを行なった。これにより、Sparc Station 2で9つのファジィルールに対してMamdaniのファジィ推論で数千ルール/秒、簡略化ファジィ推論で10万ルール/秒以上というファジィ推論機能の高速性、マクロ行動決定問題に対するファジィフレームおよびファジィアルゴリズムの記述性を検証することができた。尚、ここで開発されたシェルは知的アルゴリズムが記述可能な汎用のファジィシェルとしてあらゆるインテリジェント制御に幅広く利用できるとともに、今後のこの分野の研究者に対し一つの有益なシェルとなることを期待したい。

第7章 結論

本研究では、主として知能ロボットの"行動"を伴う動作に着目し、環境や目的に対して比較的あいまいな状況を常に認識しながら的確な判断を行っている人間の柔軟な状況判断・適応能力を模擬した行動決定アルゴリズムを構築することを試みた。本論文では、自律移動ロボットを対象とした行動決定アルゴリズムにおいて、次の5項目の新しい研究成果を示した。

1) 高度移動メカニズムに関する研究

全方向移動、階段昇降、障害物のまたぎ越え、および幅の広い溝の乗り越えなど多様な移動機能を持つ新しいタイプの車輪・クローラ・脚複合型多機能移動ロボットを提案し、その試作モデルMRVを開発した。このロボットはさまざまな分野での保守点検作業用移動ロボットとして有効であると考えられる。本論文で提案した機構は1軸で3自由度(車輪回転1、履帯旋回1、クローラ脚旋回1)、計12自由度を有しており、少ないアクチュエータで複雑な多機能動作を実現するエネルギー効率のよい自立型移動ロボットが構築できる。

本論文では、MRVの基本メカニズムと多機能移動モード、適応動作シーケンスについて述べると共に、

1) 3脚で安定な立脚状態を保ちながらパイプなどの障害物を非接触でまたぎ越えときの適応動作シーケンスにおける静的安定性とその安定余裕

2) 約10cmの直径をもつ障害物がまたぎ越え可能であること

などが明らかになった。さらにこの多機能移動ロボットを実際に試作し実験により、機構としての動作チェック、動作シーケンスの実証、動作解析に対する検証を行い、その有効性を確認することができた。

2) 障害物回避制御アルゴリズムに関する研究

移動障害物を有する環境において、人間に似た柔軟な知的回避行動ができる自律移動ロボットのためのファジィ障害物回避アルゴリズムを提案した。本方式は以下のような特徴をもっている。

1) ファジィ推論を用いることにより、あいまいな状況に基づく判断ルールを明確に記述できる。

2) 「静的危険度」および「動的危険度」という2つの中間変数のみを基準値として用いることにより状況判別を可能とした。

3) 行動決定のプロセスを「あいまい状況判断(危険度認識) + 手続き的戦略決定(操作量決定)」の階層構造で表現することにより、種々の戦略を包含しやすい構成とした。

本報告で示した以外にもさまざまなシミュレーションを通して多くのケーススタディを行った結果、従来の代表的なポテンシャル法の場合のようにロボットが停留というデッドロック状態に陥ることなく人間と同じような柔軟な回避行動をとることが確認され、提案した手法の有効性を検証することができた。

さらに本報告では、制御戦略を記述したディシジョンテーブルをオフラインで自動生成するため、汎化作用があり、効率の良い学習が可能であるCMAC学習アルゴリズムを適用した。この手法を用いると、実際にオペレータにロボットを操作させたときの回避特性・制御戦略等のスキルを容易に獲得できる。2次元平面内における移動ロボットの障害物回避問題に対して計算機シミュレーションを行い、オフラインでオペレータの実際の操作をCMAC学習させたところ、人間の操作特性(回避戦略)を表すディシジョンテーブルを戦略マップの形式で獲得することができた。この方法を用いると、例えば、さまざまな異なるオペレータの操作戦略をあらかじめ獲得しておき実際の移動時に戦略を切り替えることにより、ロボットの障害物回避における特徴を微妙に変化させるようなことも可能となる。

3) 経路決定アルゴリズムに関する研究

複雑なグリッドマップの経路探索問題において、ファジィ推論を用いて探索有効度をチェックすることにより、従来の枝刈り法に比べ探索効率が大幅に高い探索アルゴリズムを提案した。さらにこの探索特性を大きく左右する進退しきい値の自動修正方法についても検討を加えた。このヒューリスティック探索アルゴリズムの有効性を検証するため、立体グリッドマップを用いて経路探索シミュレーションを行ったところ、本方式は、総当たり比に約1/5、 β カットに比べても約1/3程度の探索時間で最適解を得ることが可能であることがわかった。本方式を用いると複雑な経路を対象とした経路計画問題において、領域分割後のノード探索などの単純繰り返し評価の知的な効率化が図れるものと期待できる。

さらに移動ロボットの経路計画問題において、人間が行っているような地図情報があいまいであっても未知環境や障害物に対する適応性の高い経路計画を行える手法をファジィ手法を用いて実現することを試みた。まず地図情報のあいまいさについて検討し、物体が存在することに対する確信度であいまいな地図を定義し、これらを用いて実際人間が行っているような階層的な経路決定を模擬する経路計画アルゴリズムを提案した。また本手法に対して経路計画シミュレーションを行った結果、「注意して」「急いで」などの与えられた行動目的に応じて評価順に異なる経路生成が可能であることが確認できた。本方式は本来オフラインで経路計画を行うことを前提としているため通常の屋内環境程度の経路であれば十分適用可能で、特に経路が明確でなくあいまいな評価や目的が与えられている問題に適していると考えられる。

4) 行動決定アルゴリズムに関する研究

クリップ量およびファジィ量が混在するようなシーケンスフローが表現可能なファジィアルゴリズムの概念を用いた自律移動ロボットの知的行動決定に関するアルゴリズム構築方法について研究を行った。本研究ではファジィアルゴリズムにおけるルール前件部に上位で制御されるスレシヨールドを設け、これによりオンラインでアルゴリズムを適応的に変化させる修正ファジィアルゴリズム(MFA)を提案し、自律移動ロボットの「人間らしい」行動決定ルールの記述を試みた。本方式は、ファジィフレームによるあいまい概念および修正ファジィアルゴリズムによるあいまい状況の定義ができ、自律的に判断した行動目的に応じた柔軟なアルゴリズムチューニング機能を有する。本方式の計算機シミュレーションを行った結果、「PASS-ABLE」や「AVOIDABLE」などのあいまい状況のマクロ判断機能の有効性を検証することができた。また実際の自律移動ロボットを用いて同等の屋内移動実験を行ったところ、シミュレーションで得られた結果とほぼ同じ移動特性および状況判断能力が確認され、本方式の基本的な有効性が示された。

さらに、ファジィベトリネット(FPN)の考え方にに基づき、ファジィアルゴリズムのフロー全体の表現を行い、これを用いてファジィ真理値トークンの伝搬によりシステムのあいまい性の推移状況を捉えることを試みた。本手法はベトリネットの状態表現能力、状態遷移解析能力を用いることにより、従来ファジィアルゴリズムにおいて困難であったシーケンスルールの多段発火におけるあいまい性の客観的・総合的評価を行うことができる。また前述の自律移動ロボットの行動決定ファジィアルゴリズムを例題として、本手法によるあいまい性評価シミュレーションを行い、アルゴリズム全体におけるあいまいさの遷移状況がかなり正確に把握できることがわかった。これにより、あいまいさの爆発が起こらないファジィアルゴリズムの設計やシステム挙動解析が可能になると考えられる。

5) 知的行動決定アルゴリズム構築用ファジィシェルに関する研究

前述のさまざまな知的行動決定アルゴリズムの記述を行うため開発したユーザプログラム主導型の知的制御用ファジィシェルの基本構成並びに表現機能について述べた。このシェルは、1) ライブラリ形式

7. 結論

の高速推論エンジン、2) オンラインチューニング機能、3) オフラインエディタ機能、4) 推論状況表示機能、5) ファジィフレーム表現、6) ファジィアルゴリズムコンパイラ、など数々の特長を有している。このシェルを用いると、従来のシェルではファジィ推論のみであったルール表現がファジィアルゴリズム機能により、AIプロダクションルールに近い多段のシーケンスアルゴリズムも記述可能となり、ファジィ情報処理としての幅広い利用が期待できる。またファジィフレーム機能によりデータの属性値の記述が困難であった従来手法に対し、あいまい属性をもつ階層的なデータ構造による効率的かつ強力な知識表現が可能となる。

またシェルの基本機能を確認するため、車の追従/追越制御を例題として簡単なシミュレーションを行った。これにより、Sparc Station 2で9つのファジィルールに対してMamdaniのファジィ推論で数千ルール/秒、簡略化ファジィ推論で10万ルール/秒以上というファジィ推論機能の高速性とマクロ行動決定問題に対するファジィフレームおよびファジィアルゴリズムの記述性を検証することができた。

本論文における以上の研究では、特に自律移動ロボットの行動決定に関連するあいまい性に着目し、主にファジィ理論を適用することにより従来手法では実現が困難であった人間を模擬した状況判断能力をロボットに付加することを試みた。本論文で提案したアルゴリズムを用いることにより、自律移動ロボットにとって重要な行動決定ルールに「人間らしい」機能をもたせることができるだけでなく、マクロ記述により人間にとってわかりやすいシンプルなルール記述が可能となる。

以上の研究を通じて得られた成果を総括すると、対人親和性のよい知能ロボットにおける行動決定アルゴリズムの構築には以下のような3つの条件を含んでいる必要があることが明らかとなった。

a) 知識表現におけるあいまい性の包含

すでに述べたようにさまざまな決定レベルの知識表現、特に状況認識においてあいまい性が必然的に含まれており、自律移動ロボットの基本機能を構築しようとした場合、これらのあいまい性を定性的にマクロに捉え何らかのかたちで包含することが不可欠である。ヒューマンフレンドリロボットを構築する場合、これまでに対象としてきた行動決定アルゴリズムだけではなく、人間とのコミュニケーションや外界情報のセンサフュージョンなどさまざまなあいまい性を含む他の機能が数多く存在する。ファジィ理論をはじめとする表現手法を用いることにより、このようなさまざまなあいまい性を積極的に明示的に取り扱うことにより、人間にとって可視性のよいアルゴリズムが実現できると考えられる。

b) シーケンスアルゴリズムによる決定プロセスの表現

ロボットによって発現されるある一連の"行動"を表現しようとした場合、シーケンシャルな手順(アルゴリズム)を記述する必要性が生じる。このようなアルゴリズムの蓄積により高度な決定システムを構築できる。特に、このシーケンスアルゴリズムの分岐判断の中に上述のあいまい性が含まれる場合には、本論文で述べたようなファジィアルゴリズムの適用が有効である。これにより状態表現だけでなく、決定プロセス自身にもあいまいさを許容した柔軟な構造をもたせることができる。さらにファジィベトリネットの適用により、このような把握が困難なあいまい性を含むアルゴリズム全体の挙動を総合的に評価できることも明らかとなった。

c) 階層決定構造によるシステム構成

Brooksの"Subsumption Architecture"に代表されるシステムの階層構造は、昆虫ロボットなどの比較的反射的な行動をとるような下等動物の行動決定をうまく模擬できると言われている。これは多層並列処理による反射的決定構造をとっており、高度な推論などを必要としない。しかしながら、言語を扱うような高度な推論

7. 結論

を行う人間を模擬しようとする、やはり階層的意思決定構造が不可欠となる。特に、下位は局所的な状態認識/状況判断機能、上位は大局的な戦略評価/行動決定機能で構成されているような階層決定構造は、人間の意思決定プロセスに非常によく合致している。知能ロボットの行動決定においてもこのような形態をとることにより、複雑かつ高度な決定システムが構築できることが本研究を通じて認められた。

以上の研究において知能ロボットへのファジィ理論のさまざまな適用を通して、知能化研究へのファジィ情報処理技術の応用可能性を探ることができた。本研究で得られた成果はいずれもロボット研究において従来にない新たな局面を展開するものであるが、中でも障害物回避制御アルゴリズムおよび行動決定アルゴリズムに関する研究はヒューマンフレンドリロボットの実現に向けて一つの方向性を示唆することができたと考える。知能ロボットによりよい対人親和性をもたせるためには、「人間にやさしい」ヒューマンインタフェース技術はもちろん、ここで述べたようなロボット自身に「人間らしい」行動を実現させるためのアルゴリズムを提供する技術もこれからは重要になるものと考えられる。本研究で得られた知見をもとに、今後はより人間に近い判断が可能な知能ロボットを実現するためのさらに高度なアルゴリズム構築手法の研究へと発展させていく考えである。

謝 辞

謝 辞

本論文をまとめるにあたり、懇切丁寧なる御指導と御鞭撻を賜りました法政大学工学部 廣田薫教授に深甚な謝意を申し上げます。

本研究の方向性指示および機会を与えていただいた国際ファジィ工学研究所 (LIFE) 寺野寿郎所長に深く感謝致します。また本研究をまとめる過程で適切な御指導および御教示をいただきました法政大学工学部 渡辺嘉二郎教授、同 小林尚登教授、京都大学工学部 榎木哲夫助手、LIFE 高木友博次長 (現在、松下電器 (株))、LIFE 山中猛室長 (現在、(株)クマリフト) に心から謝意を申し上げます。

さらに本研究に暖かい激励とLIFE出向の機会を与えていただいた三菱電機 (株) 名古屋製作所開発部萩原史朗部長、同開発本部 渡辺光人技師長、同産業システム研究所 丸山寿一主管技師長に感謝致します。尚、本研究を遂行するにあたりLIFEでの有益なディスカッションおよび実験等に協力を頂いた田鍋実主任研究員 (現在、(株)NKK)、湯田盛和主任研究員 (現在、山武ハネウェル (株))、稲垣泰弘主任研究員、法政大学工学部 橋本智己氏に感謝致します。

最後に、本研究の推進にあたり、有益な御指導およびアドバイスをいただいた三菱電機 (株) 産業システム研究所制御システム開発部 竹垣盛一グループリーダー、同FAシステム開発部 森田温グループリーダー、同制御製作所原子力部原子力情報通信システム技術グループ 津谷定廣主幹、ならびにLIFE主任研究員の方々に厚く感謝致します。

参考文献

参考文献

1. 序 論

- [1] C.ISIK and A.M.Meystel, "Pilot Level of a Hierarchical Controller for an Unmanned Mobile Robot", IEEE Jour. of Robotics and Automation, Vol.4, No.3, pp.241-255 (1988)
- [2] P.W.Cudhea and R.A.Brooks, "Coordinating Multiple Goals for a Mobile Robot", Intelligent Autonomous Systems (North-Holland), pp.168-174 (1986)
- [3] 日本機械学会講習会テキスト"ロボメカアドバンスシリーズ～アドバンスロボットビヘービア～"(1993)
- [4] E.H.Mamdani, "ヨーロッパにおけるファジィ技術の現状(1)", 日本ファジィ学会誌, Vol.5, No.3, pp.465-473 (1993)
- [5] L.A.Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Process", IEEE Trans. on Systems, Man, and Cybernetics, Vol.SMC-3, No.1, pp.28-44 (1973)
- [6] 榎木, 片井他, "視野情報に基づく自走ロボットのopportunistic path-planningとファジィ走行制御", 第12回 知能システムシンポジウム, pp.63-68 (1990)
- [7] G.W.Kohler, M.Selig and M.Salaske, "Manipulator Vehicles of the nuclear Emergency Brigade in the Federal Republic of Germany", Proc. of 24th Conf. on Remote System Technology, pp.196-218 (1976)
- [8] 岩本, 山本, 他, "地形変化に応じながら走行する形状可変形クローラ走行車の機構と制御", 日本ロボット学会誌, vol.2, no.3, pp.24-32 (1984)
- [9] 竹原, 中山, 他, "原子力点検用ロボットに用いる半月形クローラ車の開発", 日本ロボット学会誌, vol.2, no.3, pp.62-67 (1984)
- [10] T.Lozano-Perez, "Automatic Planning of Manipulator Transfer Movements", IEEE Trans. SMC, Vol.11-10, pp.681-698 (1981)
- [11] O.Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", The International Journal of Robotics Research, Vol.5, No.1, pp.90-98 (1986)
- [12] 武野, 柿倉, "移動ロボットの移動障害物に対する衝突回避問題について", 日本ロボット学会誌, Vol.4, No.5, pp.33-37 (1986)
- [13] 登尾, 野田, 有本, "オクトツリーを用いた経路決定のための各種アルゴリズムの提案", 知識工学と人工知能研究会, 49-9, pp.65-71 (1985)
- [14] 室伏, 菅野, "模型自動車のファジィ制御", 日本ロボット学会誌, Vol.6, No.6, pp.536-541 (1988)
- [15] 染谷, 安信, "エキスパートシステム構築ツールES/KERNEL/Wのファジィ推論", 日本ファジィ学会誌, Vol.2, No.2, pp.13-20 (1990)
- [16] 山崎, 野村, 他, "多段型ファジィ推論シエルの構築", 第6回ファジィシステムシンポジウム, pp.445-448 (1990)

2.2 多機能移動ロボットMRVの開発

- [1] E.Nakano, N.Koyachi, "An Advanced Mechanism of the Omni-Directional Vehicle (ODV) and its Application to the Working Wheelchair for the Disabled", Proc. '83 ICAR, pp.277-284 (1983)
- [2] G.W.Kohler, M.Selig and M.Salaske, "Manipulator Vehicles of the nuclear Emergency Brigade in the Federal Republic of Germany", Proc. of 24th Conf. on Remote System Technology, pp.196-218 (1976)
- [3] 岩本, 山本, 他, "地形変化に応じながら走行する形状可変形クローラ走行車の機構と制御", 日本ロボット学会誌, vol.2, no.3, pp.24-32 (1984)

参考文献

- [4] 竹原, 中山, 他, "原子力点検用ロボットに用いる半月形クローラ車の開発", 日本ロボット学会誌, vol.2, no.3, pp.62-67 (1984)
- [5] N.Ozaki, M.Suzuki, and Y.Ichikawa, "Tele-Operated Mobile Robot for Remote Maintenance in Nuclear Facilities", Proc.'83 ICAR, pp.67-74 (1983)
- [6] 広瀬, 梅谷, "4足歩行機械システムの開発", 第1回知能移動ロボットシンポジウム, pp.101-106 (1982)
- [7] →研究業績 C-1
- [8] →研究業績 S-2
- [9] →研究業績 C-2
- [10] →研究業績 P-2

3.1 ファジィ障害物回避制御アルゴリズム

- [1] T.Lozano-Perez, "Automatic Planning of Manipulator Transfer Movements", IEEE Trans. SMC, Vol.11-10, pp.681-698 (1981)
- [2] O.Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", The International Journal of Robotics Research, Vol.5, No.1, pp.90-98 (1986)
- [3] 武野, 柿倉, "移動ロボットの移動障害物に対する衝突回避問題について", 日本ロボット学会誌, Vol.4, No.5, pp.33-37 (1986)
- [4] Y.Anzai, "Cognitive Control of Real-Time Event-Driven Systems", Cognitive Science 8, pp.221-254 (1984)
- [5] 長谷川, "船舶の衝突回避システム", オートメーション, Vol.33, No.6, pp.42-49 (1988)
- [6] 樫木, 片井他, "視野情報に基づく自走ロボットのopportunistic path-planningとファジィ走行制御", 第12回知能システムシンポジウム, pp.63-68 (1990)
- [7] 石川, "ファジィ制御を用いた自律型移動ロボットの誘導方式の検討", 日本ロボット学会誌, Vol.9, No.2, pp.149-161 (1991)
- [8] 野村, 林, 若見, "デルタルールによるファジィ推論の自動チューニング手法と障害物回避への応用", 日本ファジィ学会誌, Vol.4, No.2, pp.379-388 (1992)
- [9] →研究業績 P-1
- [10] →研究業績 C-3
- [11] →研究業績 C-4
- [12] 大井, 石岡, 竹垣, "プラント制御用エキスパートシェルERIC", 情報処理学会第36回全国大会, pp.1593-1594 (1988)

3.2 CMACニューラルネットワークを用いた障害物回避戦略の獲得方法

- [1] →研究業績 S-6
- [2] →研究業績 C-5
- [3] J.S.Albus (小杉, 他訳), "ロボティクス" [啓学出版] (1984)
- [4] 田中, 清水, 土屋, "神経回路網による冗長マニピュレータの逆キネマティクス問題の解法", 計測自動制御学会論文集, 27, pp.815-821 (1991)

4.1 ファジィ推論を適用したグリッドマップ経路探索アルゴリズム

- [1] T.Lozano-Perez, "Automatic Planning of Manipulator Transfer Movements", IEEE Trans. SMC, Vol.11-10,

参考文献

- pp.681-698 (1981)
- [2] 登尾, 野田, 有本, "オクトツリーを用いた経路決定のための各種アルゴリズムの提案", 知識工学と人工知能研究会, 49-9, pp.65-71 (1985)
- [3] 登尾, 浪花, 有本, "クワッドツリーを用いた移動ロボットの高速経路決定アルゴリズム", 第5回日本ロボット学会学術講演会, pp.145-148 (1987)
- [4] S.Kambhampati and L.S.Davis, "Multiresolution Path Planning for Mobile Robots", IEEE Jour. R&A, Vol.RA-2, No.3, pp.135-145 (1986)
- [5] P.Y.Cheng, "Cost-Appraising Techniques for Route Planning", SPIE Vol.852 Mobile Robots, pp.240-247 (1987)
- [6] S.Fujii et al., "Routing Control of Automated Guided Vehicles in FMS", USA-JAPAN Symp. on Flexible Automation, pp.629-636 (1988)
- [7] N.Nilsson, "A Mobile Automation: An Application of Artificial Intelligence Techniques", Proc. Int. Joint Conf. Artificial Intelligence, pp.509-520 (1969)
- [8] →研究業績 S-5
- [9] 大井, 石岡, 竹垣, "プラント制御用エキスパートシェルERIC", 情報処理学会第36回全国大会, pp.1593-1594 (1988)
- [10] 堤, "ニューロコンピューティングによるロボットの軌道生成", 日本ロボット学会, Vol.9, No.2, pp.217-223 (1991)
- [11] 馬場, 久保田, "遺伝アルゴリズムを用いたロボットマニピュレータの軌道生成及び障害物回避", Vol.11, No.2, pp.299-302 (1993)
- [12] 北野, "遺伝的アルゴリズム", 産業図書

4.2 あいまい評価に基づく知的経路計画手法

- [1] J.Iijima, S.Asaka and S.Yuta, "Searching Unknown Environment by a Mobile Robot Using Range Sensor--An Algorithm and Experiment", IROS'89, pp.46-53 (1989)
- [2] A.Elfes, "Using Occupancy Grid for Mobile Robot Perception and Navigation", IEEE Computer, pp.46-57 (1989)
- [3] T.S.Levitt and D.T.Lawton, "Qualitative Navigation for Mobile Robots", Artificial Intelligence 44, pp.305-360 (1990)
- [4] 王, 土谷, "ファジィ軌道計画とそのロボットマニピュレータ経路制御への応用", 日本ロボット学会誌, Vol.9, No.3, pp.304-313 (1991)
- [5] →研究業績 C-6
- [6] L. Perez and A.Wesley, "An algorithm for planning collision-free path among polyhedral obstacles", Commun. ACM, Vol.22, pp.560-570 (1979)
- [7] 劉, 有本, "一般的な作業環境における移動ロボットの経路計画に利用できる接線グラフの提案", 第8回日本ロボット学会学術講演会, Vol.2, pp.541-544 (1990)
- [8] P. Kampmann and K. Schmidt, "Multilevel motion planning for mobile robots based on a topologically structured world model", Intelligent Autonomous System 2, Vol.1, pp.241-252 (1989)
- [9] E. Jungert and D. Holmes, "A hierarchical knowledge structure for heuristic path planning", Intelligent Autonomous System 2, Vol.1, pp.230-240 (1989)
- [10] 浅間, マキ, "自律型移動ロボットのための自由空間構造化手法と高速経路計画アルゴリズム", 第8回日本ロボット学会学術講演会, Vol.3, pp.881-882 (1990)

参考文献

5.1 自律移動ロボットの行動決定ファジリアルゴリズム

- [1] 榎木, 片井他, "視野情報に基づく自走ロボットのopportunistic path-planningとファジィ走行制御", 第12回 知能システムシンポジウム, pp.63-68 (1990)
- [2] C.ISIK and A.M.Meystel, "Pilot Level of a Hierarchical Controller for an Unmanned Mobile Robot", IEEE Jour. of Robotics and Automation, Vol.4, No.3, pp.241-255 (1988)
- [3] P.W.Cudhea and R.A.Brooks, "Coordinating Multiple Goals for a Mobile Robot", Intelligent Autonomous Systems (North-Holland), pp.168-174 (1986)
- [4] M.Zimmermann, "Intelligent Control for Insect-Like Locomotion", SICICI'92, pp.346-351 (1992)
- [5] →研究業績 S-9)
- [6] →研究業績 S-10)
- [7] →研究業績 C-7)
- [8] →研究業績 C-10)
- [9] →研究業績 S-14)
- [10] →研究業績 P-3)
- [11] →研究業績 P-4)
- [12] E.S.Santos, "Fuzzy Algorithm", INFORMATION AND CONTROL 17, pp.326-339 (1970)
- [13] L.A.Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Process", IEEE Trans. on Systems, Man, and Cybernetics, Vol.SMC-3, No.1, pp.28-44 (1973)
- [14] 室伏, 菅野, "模型自動車のファジィ制御", 日本ロボット学会誌, Vol.6, No.6, pp.536-541 (1988)
- [15] →研究業績 C-6)
- [16] →研究業績 C-3)
- [17] →研究業績 S-6)
- [18] →研究業績 S-13)
- [19] →研究業績 S-11)
- [20] →研究業績 C-9)
- [21] →研究業績 S-12)
- [22] →研究業績 P-5)

5.2 ペトリネットを利用したファジリアルゴリズムのあいまい性評価

- [1] →研究業績 C-10)
- [2] →研究業績 P-4)
- [3] 椎塚, "ファジィペトリネット", 日本ファジィ学会誌, Vol.4, No.6, pp.1069-1085 (1992)
- [4] →研究業績 S-16)
- [5] L.A.Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Process", IEEE Trans. on Systems, Man, and Cybernetics, Vol.SMC-3, No.1, pp.28-44 (1973)
- [6] D.Dubois, H.Prade, Fuzzy Sets and Systems: Theory and Applications, Academic Press (1980)
- [7] 水本, ファジィ理論とその応用, サイエンス社.
- [8] 室伏, 菅野, "模型自動車のファジィ制御", 日本ロボット学会誌, Vol.6, No.6, pp.536-541 (1988)
- [9] A.Novak, Fuzzy Sets and Their Applications, Adam Hilger
- [10] 北島, 菅野, "ファジリアルゴリズムにおける知識を用いたあいまいさの管理方法", 第8回ファジィシステムシンポジウム, pp.349-352 (1992)

参考文献

- [11] C.G.Looney, "Fuzzy Petri Nets for Rule-Based Decisionmaking", IEEE Trans. SMC, Vol.18-1, pp.178-183 (1988)
- [12] S.M.Chen, J.S.Ke and J.F.Chang, "Knowledge Representation Using Fuzzy Petri Nets", IEEE Trans. on Knowledge and Data Engineering, Vol.2, No.3, pp.311-319 (1990)

6.2 知的制御用ファジィシェルFRASHの開発

- [1] 染谷, 安信, "エキスパートシステム構築ツールES/KERNEL/Wのファジィ推論", 日本ファジィ学会誌, Vol.2, No.2, pp.13-20 (1990)
- [2] 山崎, 野村他, "多段型ファジィ推論シェルの構築", 第6回ファジィシステムシンポジウム, pp.445-448 (1990)
- [3] W.Siler著, 廣田薫監修, "ファジィ・エキスパートシステム理論と実戦", 電気書院 (1990)
- [4] T.Miyoshi, H.Koyama, S.Fukami and T.Takagi, "Overview of LIFE Fuzzy Expert System Shell", Sino-Japan Joint Meeting on Fuzzy Sets and Systems, C2-10 (1990)
- [5] S.Yamamoto, Y.Inoue and S.Yasunobu "Object-Oriented Approaches for Fuzzy Information Processing", Proc. of IFES'91, pp.375-386 (1991)
- [6] →研究業績 C-9)
- [7] →研究業績 S-12)
- [8] →研究業績 P-5)
- [8] →研究業績 S-13)
- [9] E.S.Santos, "Fuzzy Algorithm", INFORMATION AND CONTROL 17, pp.326-339 (1970)
- [10] L.A.Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Process", IEEE Trans. on SMC, Vol.SMC-3, No.1, pp.28-44 (1973)
- [11] →研究業績 C-10)
- [12] →研究業績 S-14)
- [13] →研究業績 P-4)

その他の参考文献

- [1] 廣田 (編著), "ファジィシステム", SICE出版委員会
- [2] 廣田, "あいまい制御知能ロボット", マグロウヒル
- [3] 寺野, 浅井, 菅野, "ファジィシステム入門", オーム社
- [4] 岩井, 片井, 榎木, 他, "知識システム工学", SICE出版委員会
- [5] 竹垣, 石岡, "知的制御システム", 海文堂
- [6] 北村, 片井, "画像と制御", 朝倉書店
- [7] 椎塚, "事例ペトリネット", コロナ社
- [8] 離散事象システム研究専門委員会編, "ペトリネットとその応用", SICE出版委員会

研究業績

研究業績

学術誌掲載論文

- P-1) 前田, 竹垣, "ファジィ推論を用いた移動ロボットの動的障害物回避制御", 日本ロボット学会誌, Vol.6, No.6, pp.50-54 (1988)
- P-2) 前田, 津谷, 萩原, "多機能移動ロボットの障害物またぎ越え動作解析", 日本ロボット学会誌, Vol.7, No.2, pp.13-19 (1989)
- P-3) Y.Maeda, M.Tanabe, T.Takagi, "Behavior-Decision Fuzzy Algorithm for Autonomous Mobile Robots", Information Sciences 71, pp.145-168 (1993)
- P-4) 前田, "あいまい状況適応能力を有する自律移動ロボットの行動決定ファジィアルゴリズム", 日本ファジィ学会誌, Vol.5, No.1, pp.116-128 (1993)
- P-5) 前田, "ファジィアルゴリズムコンパイラを有する知的制御用ファジィシェルFRASH", 日本ファジィ学会誌, Vol.5, No.3, pp.626-638 (1993)
- P-6) 前田, "ペトリネット・マーキングに基づくファジィアルゴリズムのあいまい状態遷移評価", 日本ファジィ学会誌 [投稿中]

国際会議発表論文

- C-1) Y.Maeda, S.Tsutani and S.Hagihara, "Prototype of Multifunctional Robot Vehicle", ICAR, pp.421-428 (1985)
- C-2) Y.Maeda, S.Tsutani and S.Hagihara, "Adaptable Motion of Multifunctional Robot Vehicle", Japan-USA Symp. on Flexible Automation, pp.365-372 (1986)
- C-3) Y.Maeda, "Fuzzy Obstacle Avoidance Method for a Mobile Robot Based on the Degree of Danger", Proc. of NAFIPS'90, Vol.1, pp.169-172 (1990)
- C-4) Y.Maeda, "Collision Avoidance Control among Moving Obstacles for a Mobile Robot on the Fuzzy Reasoning", Ro.Man.Sy'90, pp.55(abstract) (1990)
- C-5) Y.Maeda, T.Yamanaka, "Fuzzy Obstacle Avoidance Control with the Strategy Map Obtained by Operation Learning", Joint Hungarian-Japanese Symposium on Fuzzy Systems and Applications, pp.110-113 (1991)
- C-6) M.Tanabe, Y.Maeda, M.Yuta and T.Takagi, "Path Planning Method for Mobile Robot Using Fuzzy Inference Under Vague Information of Environment", IFES'91, pp.758-769 (1991)
- C-7) Y.Maeda, M.Tanabe, M.Yuta and T.Takagi, "Control Purpose Oriented Behavior-Decision Fuzzy Algorithm with Tuning Function of Fuzzy Branch", IFES'91, pp.694-705 (1991)
- C-8) Y.Maeda, M.Tanabe, M.Yuta, "Demonstration of Macro Behavior-Decision on Intelligent Autonomous Mobile Robot", IFES'91, pp.1110-1111 (1991)
- C-9) Y.Maeda, J.Murakami and T.Takagi, "FRASH - A Fuzzy Real-time Auto-tuning Shell with Expressive Function of Fuzzy Algorithms", SICIC'92, pp.1363-1368 (1992)
- C-10) Y.Maeda, M.Tanabe, M.Yuta and T.Takagi, "Hierarchical Control for Autonomous Mobile Robots with Behavior-Decision Fuzzy Algorithms", IEEE International Conference on Robotics and Automation, pp.117-122 (1992)
- C-11) Y.Maeda, M.Tanabe, M.Yuta and T.Takagi, "Intelligent Behavior-Decision Method based on Fuzzy Algorithm for Autonomous Mobile Robots", Ro.Man.Sy'92 (1992)

研究業績

口頭発表論文

- S-1) 前田, 津谷, 萩原, "車輪とクローラ脚を有する多機能移動機構", 第2回フレキシブル・オートメーション・シンポジウム, pp.89-90 (1985)
- S-2) 前田, 津谷, 萩原, "多機能移動ロボットの開発", 第9回ビークル・オートメーション・シンポジウム, pp.51-52 (1986)
- S-3) 前田, 樋口, 津谷, 萩原, "送電線巡視点検用鉄塔乗り越え移動機構の検討", 第5回日本ロボット学会学術講演会, pp.517-518 (1987)
- S-4) 前田, 竹垣, "ファジィ制御による移動ロボットの障害物回避", 計測自動制御学会関西支部シンポジウム「あいまい情報処理と知的システム制御」, pp.51-56 (1988)
- S-5) 前田, "ファジィ推論を用いたグリッドマップヒューリスティック経路探索の一手法", 第7回日本ロボット学会学術講演会, pp.513-516 (1989)
- S-6) 前田, 山中, "CMAC学習に基づくオペレータの制御戦略を有するファジィ障害物回避制御", 第6回ファジィシステムシンポジウム, pp.531-534 (1990)
- S-7) 小山, 前田, 深海, "ファジィエキスパートシステムによる障害物回避問題の実現", 電気学会システム・制御研究会, SC-90-24, pp.41-50 (1990)
- S-8) 小山, 前田, 深海, 高木, "ファジィプロダクションシステムによる移動ロボットの障害物回避問題の検討", 日本ロボット学会誌, Vol.9, No.1, pp.75-78 (1991)
- S-9) 前田, 田鍋, 湯田, 高木, "知能移動ロボットの目的対応型行動決定ファジィアルゴリズム", 第1回ロボットシンポジウム, pp.311-316 (1991)
- S-10) 前田, 田鍋, 高木, "目的判断に基づく知能ロボットの行動決定ファジィアルゴリズム", 第7回ファジィシステムシンポジウム, pp.119-122 (1991)
- S-11) 湯田, 前田, 田鍋, "フレーム型ファジィ知識表現を用いた階層型センサフュージョンによる生物認識", 第7回ファジィシステムシンポジウム, pp.131-134 (1991)
- S-12) 前田, "ファジィアルゴリズムが記述可能な制御用ファジィシェルFRASH", 第15回知能システムシンポジウム, pp.163-168 (1992)
- S-13) 前田, 田鍋, 湯田, 高木, "あいまい状況判断が可能な自律移動ロボットシステムの一構成法", 第8回ファジィシステムシンポジウム, pp.321-324 (1992)
- S-14) 前田, "修正ファジィアルゴリズムを用いた自律移動ロボットの行動決定", 第2回インテリジェントシステム(FAN)シンポジウム, pp.19-24 (1992)
- S-15) 前田, "ファジィ・ベースド・ビヘービア", 日本機械学会講習会「アドバンスドロボットビヘービア」, pp.31-36 (1993)
- S-16) 前田, "ペトリネットを利用したファジィアルゴリズムのあいまい性評価", 第9回ファジィシステムシンポジウム, pp.501-504 (1993)

工業所有権一覧

工業所有権一覧

筆者が本研究に関連して出願した特許のリストを以下に示しておく。

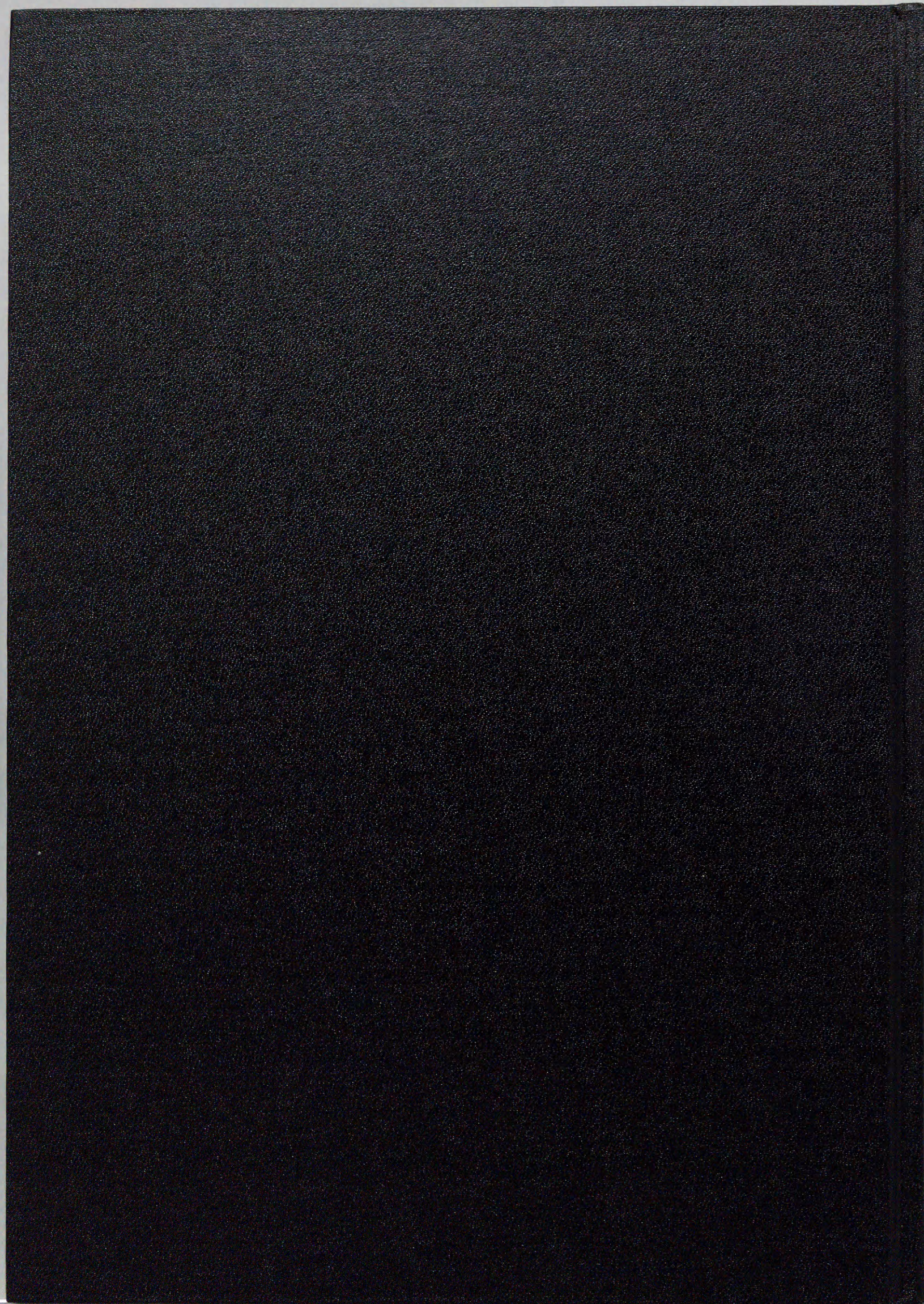
<国内特許>

出願番号 (出願日)	公開番号 (公開日)	発明名称	発明者
●59190880 (1984/9/10)	61067678 (1986/4/7)	移動装置	津谷定廣、前田陽一郎
●59190881 (1984/9/10)	61075069 (1986/4/17)	移動装置	津谷定廣、前田陽一郎
●59249781 (1984/11/28)	61129376 (1986/6/17)	クローラ式移動装置	前田陽一郎
●60114643 (1985/5/28)	61271176 (1986/12/1)	移動機械	萩原史朗、津谷定廣、前田陽一郎
●61055463 (1986/3/11)	62210512 (1987/9/16)	移動体の誘導制御方式	前田陽一郎、津谷定廣
●61054978 (1986/3/14)	62212709 (1987/9/18)	移動体の誘導制御方式	前田陽一郎、津谷定廣
●61099342 (1986/5/1)	62258870 (1987/11/11)	移動機械	津谷定廣、前田陽一郎
●62093512 (1987/4/15)	63258275 (1988/10/25)	移動機械	前田陽一郎、津谷定廣
●62059150 (1987/3/12)	63222982 (1988/9/16)	移動機械	前田陽一郎
●01008351 (1989/1/17)	02188809 (1990/7/24)	移動体の障害物回避制御装置	前田陽一郎

<海外特許>

出願番号 (出願日)	出願国	登録番号 (登録日)	発明名称	発明者
●854880 (1986/4/23)	アメリカ	4702331 (1987/10/27)	移動機械	萩原史朗、津谷定廣、前田陽一郎

注) 本特許はすでに成立/登録済み。



0005.jp2	要旨
0006.jp2	目次
0008.jp2	第1章 序論 / p1
0011.jp2	第2章 高度移動メカニズムに関する研究 / p6
0011.jp2	2.1 緒言 / p6
0011.jp2	2.2 多機能移動ロボットMRVの開発 / p7
0016.jp2	2.3 結言 / p17
0017.jp2	第3章 障害物回避制御アルゴリズムに関する研究 / p18
0017.jp2	3.1 ファジィ障害物回避制御アルゴリズム / p18
0022.jp2	3.2 CMACニューラルネットワークを用いた障害物回避戦略の獲得方法 / p29
0025.jp2	第4章 経路決定アルゴリズムに関する研究 / p35
0025.jp2	4.1 ファジィ推論を適用したグリッドマップ経路探索アルゴリズム / p35
0030.jp2	4.2 あいまい評価に基づく知的経路計画手法 / p44
0038.jp2	第5章 行動決定アルゴリズムに関する研究 / p60
0038.jp2	5.1 自律移動ロボットの行動決定ファジィアルゴリズム / p60
0046.jp2	5.2 ペトリネットを利用したファジィアルゴリズムのあいまい性評価 / p77
0052.jp2	第6章 知的行動決定アルゴリズム構築用ファジィシエルに関する研究 / p88
0052.jp2	6.1 緒言 / p88
0052.jp2	6.2 知的制御用ファジィシエルFRASHの開発 / p88
0057.jp2	6.3 結言 / p99
0058.jp2	第7章 結論 / p100
0060.jp2	謝辞 / p104
0060.jp2	参考文献 / p105
0063.jp2	研究業績 / p110
0064.jp2	著者略歴 / p112