MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 31, 1994

3. REPORT TYPE AND DATES COVERED
Final Report

4. TITLE AND SUBTITLE
Motion Planning and Sensory Processing in a Dynamic Environment

5. FUNDING NUMBERS
DAAL03-89-D-0003

6. AUTHOR(S)
Ren C. Luo
David W. Hislop

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
College of Engineering
Department of Electrical and Computer Engineering
Box 7911
North Carolina State University
Raleigh, NC 27695-7911

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ARO 27567.3-MA

11. SUPPLEMENTARY NOTES
The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

19950202 076

13. ABSTRACT *(Maximum 200 words)*

The objective of this research is to develop a sensor controlled intelligent mobile robot system for operating in a dynamic environment. In contrast to approaches for operating in a static environment, the proposed method is designed explicitly for navigation in dynamic environments and is able to immediately modify its control strategies in response to unexpected changes in the environment. The research herein is divided into two major areas: motion planning and sensory processing. The motion planning research introduces the concept of traversability vectors which can be used to represent a dynamic environment, and uses a temporal reasoning scheme to plan the robot motion. The primary research in sensory processing being the control or integration of multiple sensors on the mobile robot so as to allow for their coordinated use for the detection and avoidance of objects and obstacles in the environment.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

MASTER COPY: PLEASE KEEP THIS "MEMORANDUM OF TRANSMITTAL" BLANK FOR REPRODUCTION PURPOSES. WHEN REPORTS GENERATE UNDER ARO SPONSORSHIP, FORWARD A COMPLETED COPY OF THIS FORM WITH EACH REPORT TO OUR OFFICE. THIS WILL ASSURE PROPER IDENTIFICATION.

MEMORANDUM OF TRANSMITTAL

U.S. Army Research Office
ATTN: SLCRO-IP-Library
P.O. Box 12211
Research Triangle Park, NC 27709-2211

Dear Library Technician:

[ ] Reprint (15 copies) [ ] Technical Report (50 copies)

[ ] Manuscript (1 copy) [X] Final Report (50 copies)

[ ] Thesis (1 copy)

[ ] MS [ ] PhD [ ] Other ________

TITLE: Motion Planning and Sensory Processing in a Dynamic Environment

is forwarded for your information.

SUBMITTED FOR PUBLICATION TO (applicable only if report is manuscript):

____

____

Sincerely,

Ren C. Luo
Professor
Department of Electrical and Computer Engineering
North Carolina State University, Box 7911
Raleigh, NC 27695-7911

DO NOT REMOVE THE LABEL BELOW
THIS IS FOR IDENTIFICATION PURPOSES

Motion Planning and Sensory Processing in a Dynamic Environment

A final report prepared by

Ren C. Luo

Center for Robotics and Intelligent Machines

North Carolina State University

Raleigh, NC 27695-7911

Table of Contents

Statement of the Problem Studied

The objective of this proposed research is to develop a sensor controlled intelligent mobile robot system for operating in a dynamic environment. In contrast to approaches for operating in a static environment, the proposed method is designed explicitly for navigation in dynamic environments and is able to immediately modify its control strategies in response to unexpected changes in the environment.

Summay of Most Important Results

The three central challenges to dynamic motion control of sensor-based autonomous mobile robots involve finding optimal obstacle-free routing, updating dead reckoning, adhering to a schedule and dealing with unexpected events such as previously unknown moving obstacles. If a robot can reliably meet all of these challenges simultaneously, it can be considered truly autonomous. The research presented presented in this paper uses t-vectors, C-space-time and fuzzy behavior fusion to improve the environment awareness and versatility of autonomous vehicles. It is believed that this research helps extend the scope of mobile robots closer to performing more sophisticated tasks in manufacturing, servicing, hazardous environments and battlefields.

1. Introduction

Making a mobile robot autonomous entails simultaneously solving three inherent problems: *dynamic global motion planning, self-localization and navigation*. Hence, the three fundamental levels of dynamic motion control are: (1) high-level dynamic motion planning for global routing; (2) mid-level self-referencing for position/orientation/schedule updating; (3) low-level reactive control with fuzzy behavior fusion. This paper discusses the research that has been conducted in each of these areas at the Center for Robotics and Intelligent Machines (CRIM) in simulation and on an actual mobile robot named MARGE (Mobile Autonomous Robot for Guidance Experiments).

One reason few if any robots can claim to be truly autonomous is that most research groups have focused only on certain aspects and ignored others. For example, most work done in motion planning to date has typically ignored the needs of self-referencing and vice versa. More specifically, global motion planning has usually been confined to *geometric* environments because they minimize computer storage and complexity. Self-referencing, on the other hand, has typically been confined to *cellular* environments because they presumably facilitate range predictability, sensor-based map construction and reactive/reflexive obstacle avoidance. Such environmental differences, alone, conjure up an incompatibility whenever motion planning and self-referencing are required to work in conjunction.

The research discussed in this paper has a number of immediate impacts on autonomous robot tasking through: (1) increased reliability of environment

traversal planning. The proposed levels of dynamic global motion planning and navigation will eliminate the need for dedicated, preplanned routes that constrain mobile robots whenever the environment unexpectedly changes. (2) Improved system efficiency. All levels will utilize the same tools (i.e. remain compatible with each other), require less data storage and have lower complexity than other models presently permit. (3) Enhanced multitasking capabilities. With fuzzy behavior fusion for reactive control, mobile robots will become capable of not only traversing densely cluttered environments but also *interacting* with objects or other robots in such environments.

1.1. Dynamic global motion planning

In dynamic global motion planning, the objective is to provide the best guess at the shortest, collision-free overall routing based on the information initially given to and ascertained by the robot. Such information typically includes obstacle configurations and their time-varying trajectories. The first step in solving the dynamic global motion planning problem is deciding how to represent the environment so that it will accurately reflect the time-dependent surroundings and provide the high-level motion planner with all necessary information. The second step is being able to detect path-obstacle collisions. The third step is being able to plan motion around these obstacles. Finally, the motion should be optimized in terms of distance and time. Keeping data storage, processing time and complexity to a minimum as well as remaining compatible with self-localization is important in every step.

1.2. Self-localization

In self-localizatoin, the objective is to monitor the position, orientation and schedule of the mobile robot. Referencing position and orientation relies on sensory devices and confidence levels based on the quantity and quality of the readings. The sheer variety of potential environmental changes is difficult to pinpoint. So, too, is the variety of potential resulting sensor data patterns. It is, therefore, important to represent the environment such that the density of reference beacons is high enough that all reliable readings are predictable. Another responsibility of the self-localizer in a dynamic environment is to track the robot's position in the time domain to ensure that the robot is not too far ahead of or behind schedule with respect to the planned avoidance of other moving obstacles. The first step in solving the self-localizing problem is, as with motion planning, representing the environment so that it will enable the self-referencer to

predict and, hence, correlate sensor data for position/orientation updating in real-time. The second step is for the robot to be capable of deciding how the surrounding environment might have changed (if at all) and if such changes warrant obstacle avoidance. The third step is to act as a conductor so that the robot will not risk colliding with the very moving obstacles it had planned to avoid.

1.3. Reactive Navigation

In navigation, the objective is to control the motion of the mobile robot by *reacting* to the immediate surroundings based on information from sensors, the self-localizer and the global motion planner. Speed and turning should consider and prioritize the following: node information from the global motion planner; the self-localizer's confidence in position, orientation and schedule adherence; and the sensed proximity of nearby objects. Solving the navigation problem does not require the environment to be structured. It does, however, require that the robot interpret noisy, incomplete sensor data and respond in real-time. A decentralized control architecture based on fuzzy agents is used to smoothly arbitrate among the sometimes contradictory types of motion requirements that occur when unmapped or moving obstacles are present. The first step in solving the navigation problem is to establish a set of fuzzy expert rules that govern the type of reaction the mobile robot should have to stereotyped situations. The second step is to implement a real-time control system that can infer from the expert rules an appropriate reaction that will maintain motion fluidity.

2. Environment Representation

2.1. Geometric versus Cellular

The first issue that must be addressed when solving the motion planning and self-localization problemis how to represent the environment. It was shown by Janét, et. al. [Jan 93, Jan 94] that given the choice between cellular and geometric environment representations, geometric models (polygons and/or polytopes) are more preferrable as they reflect finely detailed objects more precisely, are simpler to map (i.e. draw in CAD), facilitate feature inferencing for self-referencing and consume less memory in general.

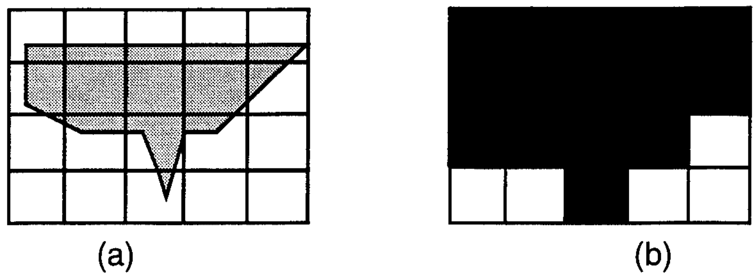


Figure 1. (a) Geometric and (b) cellular representation of same object.

2.2 Configuration-Space-Time

Configuration-space-time (C-space-time) will be used to model the robot as a compact point in space. This requires that the objects in the environment grow by a buffer of thickness t_b (robot radius plus a safety margin). The resulting *path polygon* prevents the robot from colliding with the objects and provides a set of potential paths for circumnavigation. There are two types of path polygon: the corresponding path polygon and the space-efficient path polygon. As the name implies, the corresponding path polygon has vertices that correspond to the actual polygon's vertices. Corresponding path polygons have fewer vertices than their space-efficient counterparts and hence require less memory. Despite the somewhat more complicated math, space-efficient path polygons consume only enough free space to prevent a robot-obstacle collision. As well, space-efficient path polygons provide smoothly curved trajectories for the robot to travel around edges.

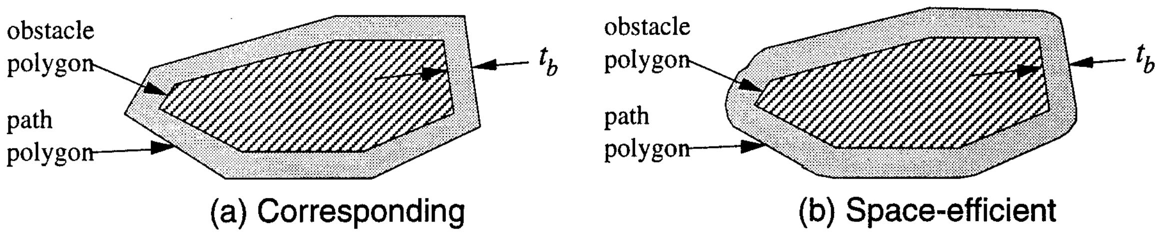


Figure 2. The corresponding path polygon and the space-efficient path polygon.

The corresponding path polygon has the tendency to force the robot into a start-stop-turn-start-stop-turn... cycle. However, there is a way to provide smooth trajectories even with the corresponding path polygons by using the same straight line approximations for the space-efficient path polygon on the optimal route path segments.

C-space-time buffers were shown to facilitate *geometric beacon* [Leo 91] detection in self-referencing [Jan 93]. Also, unlike any of the global motion

planning approaches mentioned later in this paper, C-space-time was proven capable of coping with marginally traversable regions [Jan 94]. That is, where tight passageways are normally closed to robot traffic through overlapping C-space-time buffering, local modifications can be automatically made to permit the cautious and deliberate flow of traffic. This is an attribute most motion planning approaches fail to provide.

2.3 Half-Planes and Traversability Vectors

Recently the role of half-planes and traversability-vectors (t-vectors) [Pan 90] was expanded to yield a method of collision and visibility detection that is faster, less volatile and more insightful than traditional algebraic methods [Jan 93, Jan 94]. The impacts of t-vectors (and C-space-time) were shown to be significant and extend equally into the realms of global motion planning and self-localization. The following summarizes some of the benefits and applications of half-planes and t-vectors. First, to detect a collision between a path segment and a static or moving *r*-sided polygon, t-vectors require $7r + 3$ steps while traditional algebraic methods require $12r + 2$ steps. Path segments can be potential route legs to motion planning or probes ("pseudo-paths") contained by a *sensor window* to detect in-range objects for self-referencing.

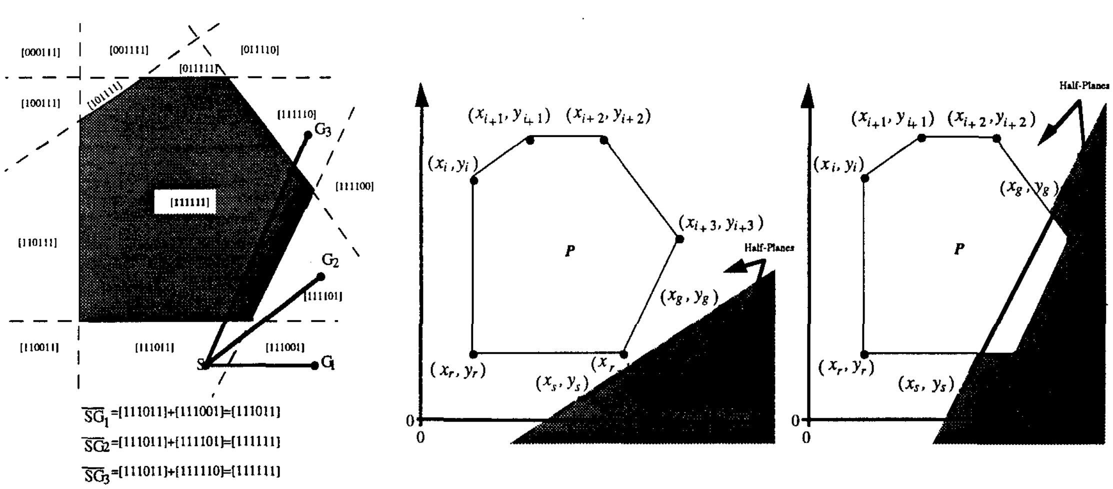
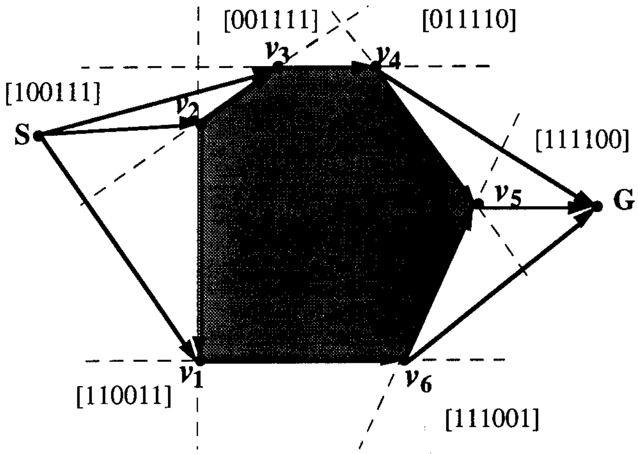


Figure 3. Two-step t-vector test detects path segment-polygon collision.

This reduction in complexity is the same for finding vertices and surfaces that are visible to an arbitrary point in space (e.g. an ultrasonic transducer). Being able to use t-vectors to eliminate vertices (edges) and surfaces (walls) that a sensor cannot see benefits self-referencing by reducing the search time needed to find the most reliable in-range feature. Since the manner in which a portion of surface falls inside a sensor cone is important to predicting echo intensity, t-

vectors further benefit self-referencing in defining which regions of the cone covered by the surface. T-vector vertex visibility has been shown to benefit global motion planning by identifying the optimal initial via points (also known as *farthest front vertices*) that yield the most efficient route around an obstacle. By default, any visible vertex that is not a farthest front vertex is considered *redundant* to keep in memory as a potential leg of robot motion (as standard V-graphs do) since it will simply never be used in optimized routes. Eliminating redundant path segments from static V-graphs significantly reduce the data size and complexity of motion planning. This will become clearer in the section on static global motion planning.



Vertex	S + v_i	FFV_S	G + v_i	FFV_G
v_1	[110111]	YES	[111111]	NO
v_2	[100111]	NO	[111111]	NO
v_3	[101111]	YES	[111111]	NO
v_4	[111111]	NO	[111110]	YES
v_5	[111111]	NO	[111100]	NO
v_6	[111111]	NO	[110101]	YES

Figure 4. T-vectors find visible vertices, surfaces and optimal via points (FFV_S and FFV_G).

3. Global Motion Planning

Compatibility issues aside, it can safely be said that most global motion planning approaches are too time consuming, complex and memory- and free space-hungry. Some cannot guarantee optimal path generation even for static maps. Others are so rigidly defined that there is little or no allowance for on-the-fly re-routing or marginally traversable regions. Previous motion planning approaches have traditionally assumed that the obstacles in the environment must be stationary with their position and orientations known exactly without further changes; otherwise the graph created for path search may not be consistent with the environment during the planning process.

Most approaches to solving global motion planning problems were addressed in the work done by Hwang and Ahuja [Hwa 92] and Pan [Pan 90]. For purposes of this paper the following approaches can be rejected based on their

surveys: The Potential Fields approach is too computationally expensive and limited in its planning capabilities. The World or Cell Decomposition approach to motion planning, much like the cellular approach to environment representation, fails in the way it approximates boundaries for self-referencing. Also, since the convexity of objects is not used to determine paths, shortest paths are not always generated. The Geometric Primitives approach is also frequently non-optimal and risks not finding safe paths when the workspace is cluttered with closely spaced objects. The Path-Network Learning approach requires the robot to make *enough* trips through the environment to be capable of finding optimal or near-optimal paths. Frequently, this does not occur because the robot travels previously learned paths rather than exploring new ones.

Of course, this list does not fully encompass all possible motion planning approaches. Several other approaches that were either not addressed by Pan or Hwang and Ahuja or simply deemed worthy of greater focus were discussed in the work done by Janét [Jan 94]. Considered in finer detail were the Visibility Graph (V-graph) [Loz 79, Loz 87, Oom 87], the Voronoi diagram [Aur 91, Rao 88a, Rao 8b, O'Du 85, Yap 84, Tak 89 Ram 85], the Route Map [Pan 90] and the Extended Tangency Graph (ETG) [Liu 90, Liu 91]. One advantage shared by all is the use of geometrically represented environments which makes them compatible with self-referencing. However, one extremely important disadvantage each shared is that marginaly traversable regions (like doorways) were not accomodated for. Another common disadvantage to these approaches, excludinging the Route Map, is that collision and visibility detection is done with the overly complex and potentially volatile algebraic method

There are numerous other disadvantages to using any of the aforementioned global motion planning approaches as they are prescribed in the literature. (For a more indepth analysis of each approach the reader is directed to [Jan 94], [Hwa 93] and [Pan 90].) Hence, it was felt that a hybrid approach should be developed that utilizes or improves the *exactness* of the V-graph, the safety margin of the Voronoi diagram, the applicability to dynamic environments of the Route Map and the space- and memory-efficiencies of the ETG. The result would be a global motion planner that minimized data storage requirements and algorithmic complexity to fulfill the high-level needs of dynamic motion control for an autonomous mobile robots.

3.1 Static Global Motion Planning with the EVG

A streamlined, appendable version of the V-graph called the Essential Visibility Graph (EVG) was developed from a geometrically represented environment, t-vectors and C-space-time to provide a static network of usable path segments for exact routing [Jan 94]. For an environment of P polygons and N vertices $\{N \geq 3P\}$, the EVG was shown to have a data size of $O\left(P^2 + N\right)$. This is a significant reduction from the $O\left(N^2\right)$ size of the V-graph and the $O(P \times N)$ size of the nearest rival the ETG. It was proven by Liu and Arimoto [Liu 90, Liu 91] that path networks free of redundancy require less memory storage, computation time and sorting time.

Since t-vectors were shown to be the more efficient way of detecting and eliminating redundant path segments from the EVG, at least a small reduction in complexity can also be expected. Using t-vectors to include only farthest-front-vertex pairs (non-redundant path segments) in the path segment network further reduces the complexity of the EVG to $O\left(k_c N^3\right)$ where $k_c \leq 2/3$. The more congested and, hence, realistic the environment, the lower the constant k_c. Hence, k_c is usually between 30% and 50% less complex than the $O\left(N^3\right)$ of standard V-graphs and the ETG. Reductions in path segment data storage of networks for realistic maps are usually between 40% to 60% of the original V-graph. Figure 5 pictoralizes these reductions. Figure 6 has applied EVG routing.

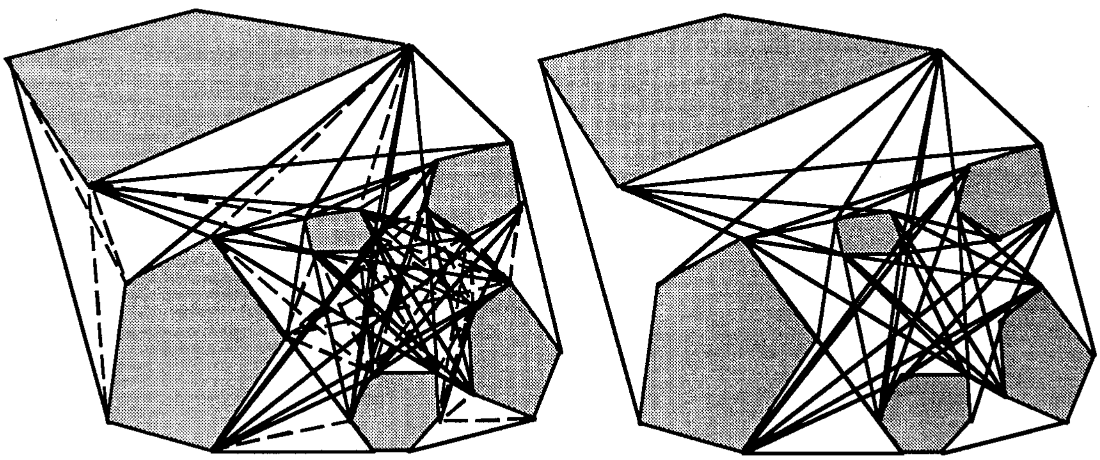


Figure 5. V-graph with redundant segments (left). Non-redundant EVG (right).

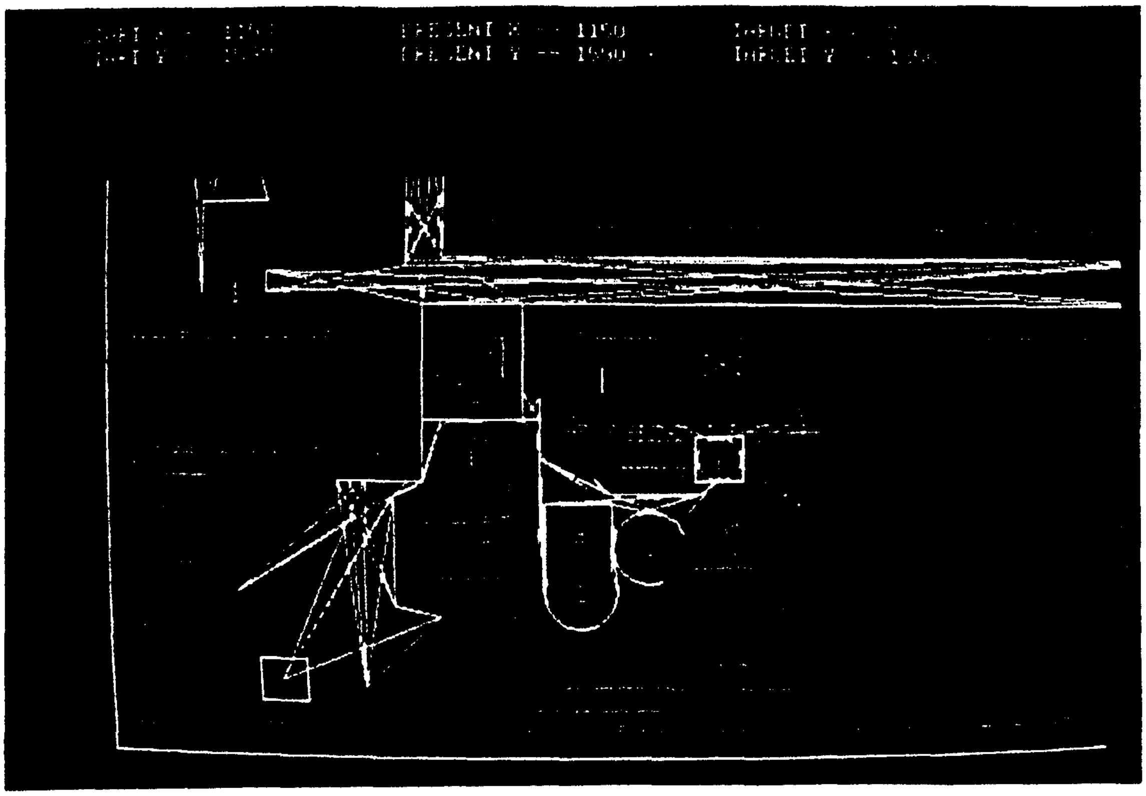


Figure 6. Routing from appended EVG network of CRIM environment.

3.2 Time Segmentation and Instance-Based Collision Detection

While it might be unrealistic to assume that one can accurately ascertain the trajectories of other moving objects, the fact remains that actual environments will indeed have them. Pan [Pan 90] showed that motion could be planned around moving objects provided the C-space-time point trajectory $m(\tau)$ and rotational angular velocity $d\theta/d\tau$ about the C-space-time point were known. Pan also assumed that at time $\tau = \tau_0$ the movable polygon has the model $\overline{A(\tau_0)}\overline{x} - \overline{c(\tau_0)} \leq 0$ and the initial angular displacement for rotation is $\theta(\tau_0)$. Again, whether it is better for a robot to try to incorporate estimated trajectories in motion planning or simply employ collision avoidance behavior is debatable. None-the-less, it is important to know that collisions *can* be detected between line segments and both static and dynamic objects. Specifically, when in self-referencing mode, the robot will be able to explain an otherwise unexpected close sensor reading when it recognizes that a moving object had just passed through its sonar cone. This will become more clear in the section on self-referencing.

Knowing a polygon's trajectory enables us to detect a collision between it

and a path segment (or polygon edge) at a point in time. Factoring in a robot's acceleration and velocity, collisions between a moving robot and a moving obstacle can also be detected provided the robot's location and orientation are known that point. To find out how long it will take for a robot to travel collision-free along a given path segment, the path must be segmentized in the time domain. Specifically, using the path/velocity decomposition method, we assume that the path a robot intends to travel along is given as shown in figure 7.

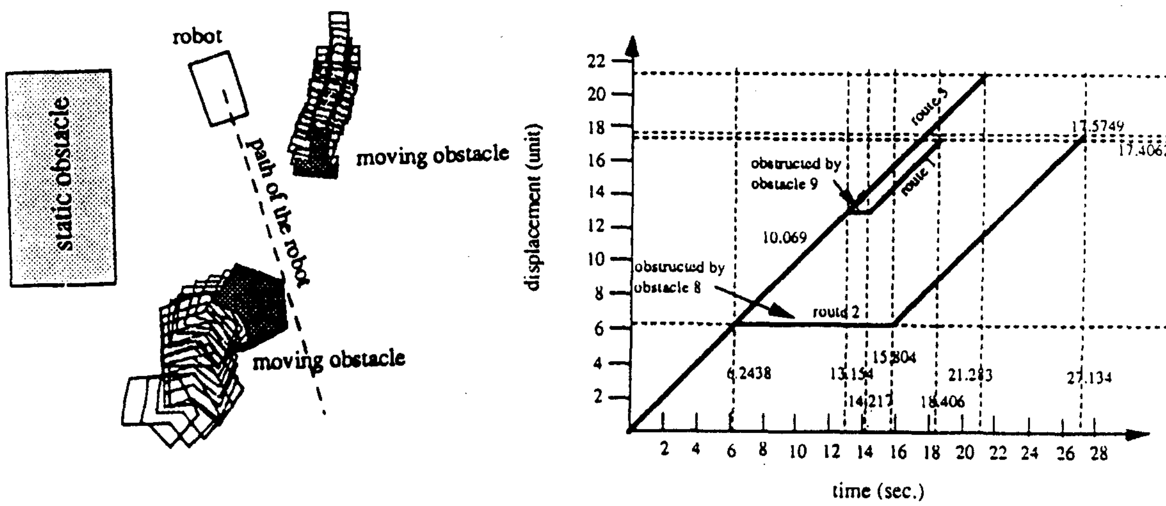


Figure 7. Illustration of the motion planning problem with moving obstacles and a displacement/time chart shows possible robot motion on segments.

The rest of the problem is to plan for the robot's velocity such that a safe motion is obtained. Interference detection between the robot and moving obstacles is performed at a sampling rate. The result is arranged into a grid representation. Each grid is defined by the coordinates (d,t) where d is the displacement along the path and t is time. When a collision is detected, the corresponding grid is labeled.

An intersection between the robot and obstacles is registered on the costraint map for velocity planning. Complicated obstacle motions can be dealt with by simply detecting collisions between polygons. Thus, velocity planning with the constraint map is independent of obstacle motions. The constraint map is naturally a discrete-time representation in which the accuracy depends greatly on the time resolution. If the task of collision-detection can be performed efficiently,

the grid representation can be more accurate by increasing the time-resolution.

3.3 Optimizing the Overall Route

With a static network and the known trajectories of other (known) moving obstacles the network needs to be sorted by Dijkstra's greedy algorithm so that the most efficient, overall route can be found [Baa 88, Hwa 91]. Weighting of the path segments will be based on the shortest time required for the robot to travel along that segment and prior connected segments.

4. Self-Localization

Conservative self-localization approaches require manually placed active or semi-active beacons which are neither cost-effective nor efficient. Bolder self-referencing approaches suffer in how they require user supplied environment information, predict sensor readings and sample the environment at low frequencies. G*eometric beacon* labels like "edge", "wall" and "corner" are commonly used to explain discrepancies between actual sensor data and predicted time-of-flight (TOF) measurements [Leo 91]. One problem with limiting the list of geometric beacons to just these three features is that the potential variance in data quality each can produce is substantial. Another problem is that beacon density depends on either user supplied labeling or complex algorithms to determine where one feature ends and the other begins. This problem is further compounded if cellular maps are used.

A mobile robot must be able to maintain a dynamic memory of obstacles by mapping and accepting into memory any previously unknown objects [Boz 91, Elf 87, Zel 91] and/or deleting from memory any removed objects [Eve 90, Leo 91b]. Perhaps of greater importance, the autonomous mobile robot must be capable of confirming and adjusting its position and orientation in real time [Eve 90, Wan 91]. To effectively close the loop on position and orientation control a robust world model must be initially be constructed such that it can be expediently searched and used to calculate range data from identified features of *geometric beacons.* Efficient data interpretation and position updating rely, of course, on the models used to generate acceptable sensor ranges and the algorithms used to compare those ranges with actual data and react accordingly [Cha 85, Kuc 89].

4.1 Self-Localization with the Extended Kalman Filter

Perhaps the most effective and reliable model to date is the Multiple

Hypothesis Tracking (MHT) technique which employs Extended Kalman filtering [Cox 91]. In short, the MHT is a stochastic, discrete-time approach that requires objects to be represented geometrically and feaures labelled. Not only can it filter data while comparing calculated and observed ranges, but it can also *build* maps. Some problems inherent to approaches like the MHT are that feature labels (1) need to be user-defined; (2) are usually sparsely placed; (3) do not encompass the variety of potential range readings a sensor can get. These problems can be accounted for, however, with the self-referencing approach discussed in the next subsection.

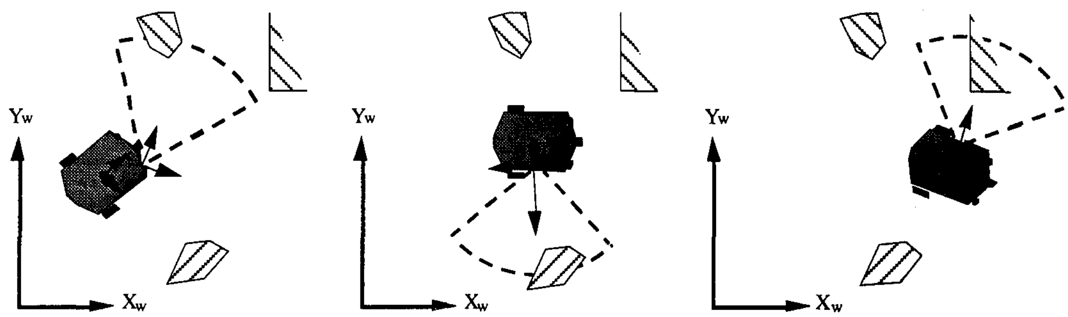


Figure 8. Using geometric beacons to reference position, orientation & schedule.

4.2 Self-Referencing

It was shown by Janét, et. al., [Jan 93] that *every* object in the environment could be used as a geometric beacon through the use of *sensor windows.* If feature labels were to be used, Janét, et. al., presented the necessary and simple inference rules. However, it was later determined that range readings were less dependent on feature type and more dependent on surface dimension, cone penetration and reflective angle [Jan 94]. It was also shown that range readings could be predicted more precisely using geometric representation. Finally, Janét, et. al., showed that traversability vectors and configuration-space-time expedited the search for geometric beacons. Figure 9 shows a simple example of planned motion around tables, carts, boxes and barrels in one of the rooms in the Center for Robotics and Intelligent Machines (CRIM). To test the sensor window method of self-referencing, sonar data was calculated at frequent intervals travelled by our mobile robot, MARGE (Mobile Autonomous Robot for Guidance Experiments). The plot of figure 10 reinforces the use of sensor windows.

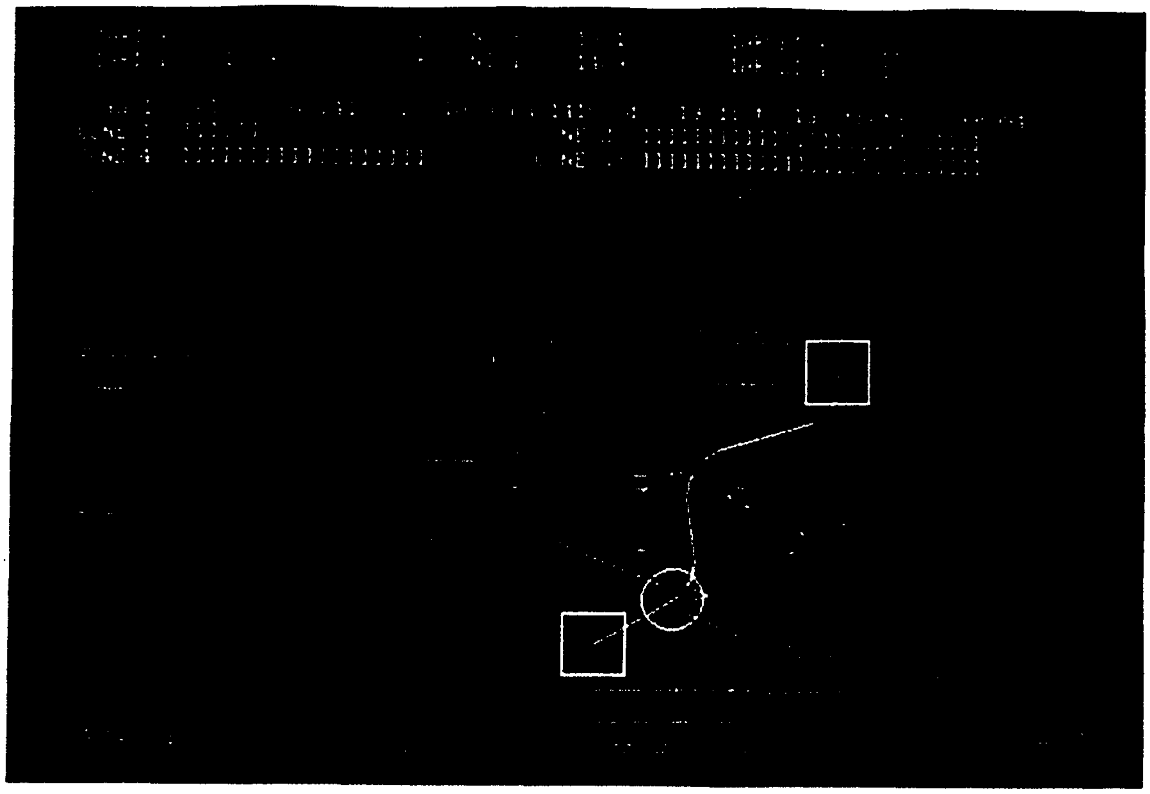


Figure 9. Path from the southwest corner of CRIM room to the northeast corner.

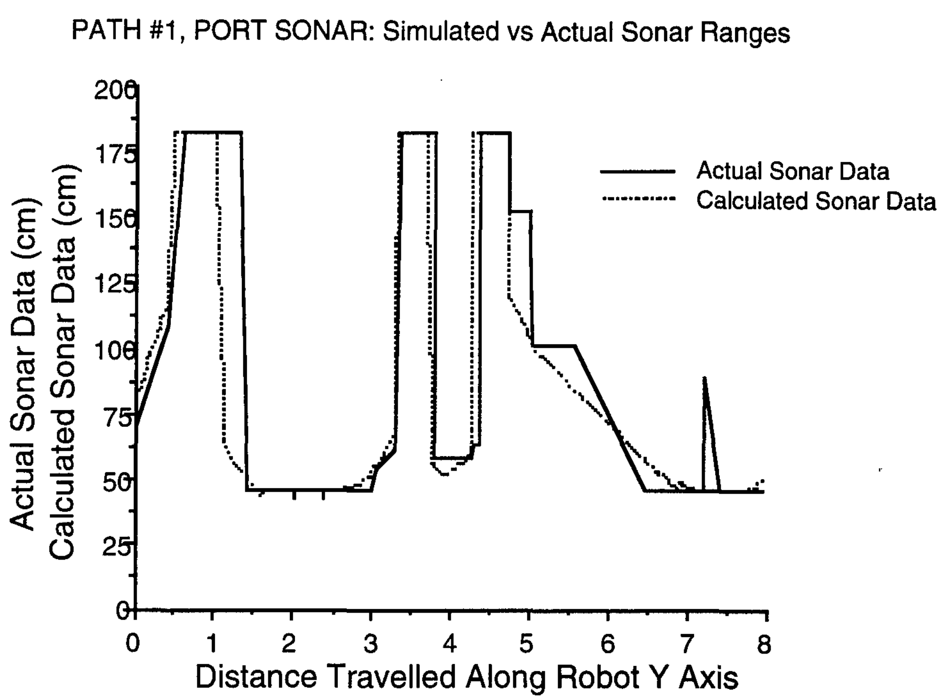


Figure 10. Plot shows correlation between calculated and sensed sonar data.

5. Fuzzy Behavior Fusion for Reactive Navigation

A recent trend in mobile robotics research has focused on real-time reactive

control actions with independent behaviors called "schemas." Each schema generates a potential field function for objectives such as goal seeking and obstacle avoidance. These vectors are fused by a higher-level supervisor, which assigns weights to the outputs of each behavior. The common feature of these reactive methods is the speed and reliability with which they perform simple tasks.

The lack of a precise model of a mobile robot's environment provides considerable incentive to the programmer to use fuzzy logic [Zad 65, Zad 88]. Fuzzy control has been explored for mobile robot guidance in simulations by Ishikawa [Ish 91], Song and Tai [Son 92] and Pin, et al. [Pin 92]. Successful hardware implementations have been realized by Sugeno, et al. [Sug 89], Pin, et al. [Pin 92] and Konolige, et al. [Kon 92]. S.R.I.'s robot, "Flakey," [Kon 92] uses fuzzy logic to define control behaviors for a variety of tasks. These behaviors are switched on and off by a supervisory controller.

6. Concluding Remarks

The proposed research solves the global motion planning/self-referencing/navigation problem in four steps: (1) environment representation; (2) acquisition of global robot motion from a high-level dynamic motion planner; (3) schedule tending and location of geometric beacons to be used as position and orientation references while the robot traverses the environment; and (4) goal-seeking and avoidance of unexpected obstacles from a low-level fuzzy logic based reactive navigator. The environment will be represented geometrically since such representation facilitates motion planning on both static and dynamic levels and provides better surface information to sensors. All objects will be convex polygons and represented by half-planes which can be time-varying when the environment is dynamic. Traversability-vector (t-vector) theory will be used to yeild a more efficient collision detection crucial to building an Essential Visibility Graph (EVG) and determining which objects are within range of robot sensors. T-vectors can also be time-varying and quickly indicate which portion of a polygon is visible to robot sensors. Configuration space time (C-space-time) will be used to ensure a collision free path around all objects, enhance the detectability of landmarks in the environment, reduce the number of time segments needed to detect moving obstacles and yield smooth turning trajectories.

On the highest level, a network of non-redundant path segments will

efficient collision detection crucial to building an Essential Visibility Graph (EVG) and determining which objects are within range of robot sensors. T-vectors can also be time-varying and quickly indicate which portion of a polygon is visible to robot sensors. Configuration space time (C-space-time) will be used to ensure a collision free path around all objects, enhance the detectability of landmarks in the environment, reduce the number of time segments needed to detect moving obstacles and yield smooth turning trajectories.

On the highest level, a network of non-redundant path segments will comprise the EVG which, when compared to other global motion planning methods, is more compatible with the chosen self-referencing method and has lower complexity and data storage requirements. Sensor windows will be used in the mid-level self-referencing to accurately reflect sensor boundaries and determine which geometric beacons are in range of mobile robot sensors. All tools utilized in constructing the EVG will also be employed by the sensor windows. Low-level motion control and (previously unmapped) obstacle avoidance will utilize reactive fuzzy behavior control architectures to ensure smooth movement and responsiveness to unexpected events.

While maximizing the motion-planning/self-referencing/navigation compatibility is of prime importance, measures will also be persistently taken to reduce complexity, data storage and processing time in general. Since real-time performance is essential, no reduction in any of these will be considered too trivial. Individually, motion planning performance will be judged by its ability to always generate optimal paths and maximize the robot's ability to get position and orientation information from objects in the environment. Self-referencing performance will be judged by its ability to accurately predict real sensor readings from objects within sonar range, confirm or modify its position/orientation and modify the change of environment whenever new obstacles are found or old ones removed. The rationale for this is as follows: if sensor readings are reliably predictable, regardless of object type, *every* object in the environment can be used as a landmark for self-referencing. Hence, there will be no need for expensive active or semi-active beacons (e.g., IR docking beacons, RF triangulation, in-floor guide wires or tracks and/or reflective path tape). Instead, the very objects that are considered *obstacles* in motion planning will be considered passive *geometric beacons* in self-referencing. Navigation will be judged by its ability to prioritize the numerous and potentially conflicting demands from the global motion planning

module, self-referencing module and on-board sensors to react in a fashion that guarantees safe and smooth motion control.

Task 3: Distributed Fuzzy Behaviors for Reactive Navigation

A distributed, heterogeneous network of fuzzy controllers has been developed for reactive behavior-based control of an autonomous mobile robot. This methodology allows our vehicle, MARGE, to perform realistic tasks in unstructured environments. Control actions for the robot are generated by a colony of independent agents that compete and cooperate to determine the emergent motion of the vehicle. Our multi-layer approach differs from other methods that perform the fuzzy inference mapping in one step. We have implemented and tested our system on a physical mobile robot with great success. MARGE used fuzzy behaviors to win first place in Event III of the 1993 AAAI Mobile Robot Competition in Washington, D.C.

Fuzzy Controller Design

Fuzzy sets, originally developed by Zadeh [Zad 65] allow data to be assigned a fractional degree of membership to a set. If a fuzzy set is named after an adjective, then its membership function can be defined to reflect the similarity between sampled data and the quality meant by the adjective. Fuzzy rules use such sets in order to trigger a control response that increases in strength in proporition to the similarity between the system's state and the adjectives used. The application of multiple fuzzy rules results in multiple output recommendations that must be combined. If each rule *i* prescribes an output value of $output_i$ with an antecedent certainty of $weight_i$, then the output of the controller is calculated as:

$$control\_output = \frac{\sum_{i=1}^{\#rules} weight_i \bullet output_i}{\sum_{i=1}^{\#rules} weight_i} \qquad (32)$$

Unlike traditional fuzzy logic schemes, our system defines the outputs of fuzzy rules as singleton values, rather than fuzzy sets. In effect, our controller design is only half fuzzy, with de-fuzzification accomplished directly by the simplified centroid calculation. This greatly increases the speed at which fuzzy rules can be processed without special hardware. It allows the system to scale well and still be used for real time control. Such a scheme could not be used for a

traditional knowledge-based or constraint-based fuzzy expert system, but it is more than adequate for the colony of behavioral agents in our reactive system.

The Fuzzy Control Bottleneck

Previous fuzzy mobile robots, described by Sugeno, et al [Sug '89], Konolige, et al [Kon 92], and Pin, et al [Pin 92], have demonstrated the utility of fuzzy control, but the problem of scalability has plagued most implementations. Most fuzzy control applications, such as a servo motor controller, involve a small number of inputs and outputs, where the mapping from sensors to actuators is accomplished in one step by a fuzzy rule base. This makes the complexity of the rule base simple enough for the engineer to define rules manually. Mobile robots, however, often feature many redundant sensors that provide different types of information. It is often desirable to incorporate this data, as well as memory of past experience, into the reactive control scheme.

Suppose we wish to design a controller for a system with N inputs, and each input i is to be described by M_i fuzzy sets. A different rule may be written for every intersection of set descriptions that describes the N inputs. This exhaustive method yields a rule set of the following size:

$$\# RULES = \prod_{i=1}^{N} M_i \tag{33}$$

Unfortunately, the number of fuzzy set evaluations in a rule base increases exponentially as more inputs are added to the controller. In order to keep the rule base manageable, other mobile robot implementations have reduced the input space by throwing away what might otherwise be useful sensor data [8], [9].

Distributed Fuzzy Agents

Rather than reducing the input space of the fuzzy control system by non-fuzzy means, we chose to develop a system that would process a large data space with many independent fuzzy controllers. In the spirit of distributed intelligence, multiple control recommendations are generated in parallel as independent agents. These behaviors themselves can be made up of smaller fuzzy behaviors, in a modular network. By feeding the output of one node into the input of another, the mapping to be performed at each stage much simpler. This is called *fuzzy pre-processing*, and is illustrated in Figure 20a. Not only can fuzzy

rules accept the outputs of other controllers as inputs, but the output assignment of a rule can be the output of another fuzzy node, rather than a fixed value. When such a rule fires, the output of another control node is added into the centroid calculation for the present node. This allows a control node to function as a *fuzzy multiplexer* by making smooth transitions between multiple recommendations according to qualitative rules. This effect is illustrated in Figure 20b.

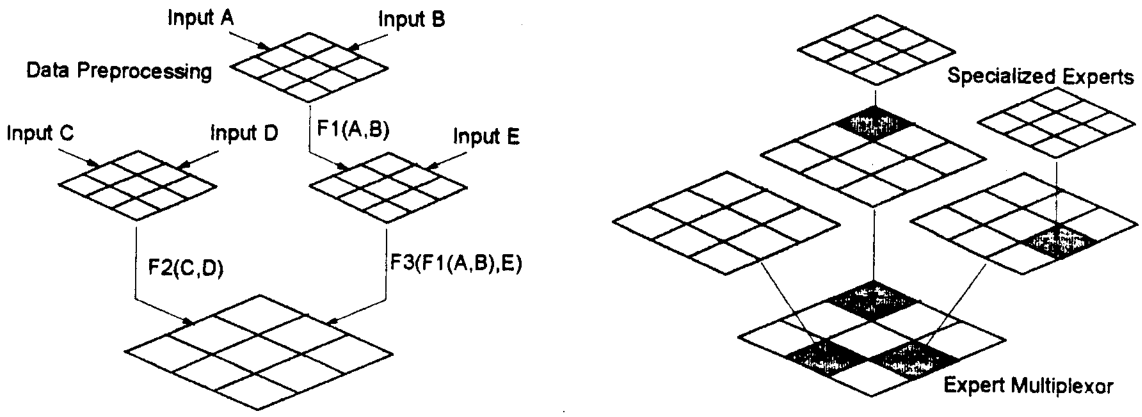


Figure 20. Fuzzy pre-processing and fuzzy multiplexing.

Note that such a multiplexing operation is possible with traditional fuzzy set theory. However, the computational overhead usually associated with a chain of fuzzy inferences (re-calculating the supporting membership at each step) makes real time implementation difficult. Since our method only passes a singleton value between nodes, we avoid this bottleneck.

Fuzzy Behaviors

In order to compute the robot's motor actions in a distributed manner, individual behaviors are developed using one or more fuzzy controllers. Each behavior makes a control recommendation based on its own limited input domain. Examples of behaviors for MARGE include goal seeking, obstacle avoidance, barrier following, and object docking. The obstacle avoidance behavior filters sonar data and suggests an appropriate steering and drive velocity given the presence of obstacles sensed by the vehicle's sonar. The goal-seeking behavior generates the proper control values to attain a goal location, and barrier following stabilizes the vehicle's motion with respect to straight walls. Object docking allows the robot to manipulate objects, which is usually difficult for autonomous systems in unmapped environments.

Fusion of Control Recommendations

Multiple behaviors eventually must be fused to result in the vehicle's motion. Many different schemes have been used for this in the literature, such as hierarchical switching [Bro 86, Kon 92] and weighted summation [Ish 91]. Our approach combines these techniques by using additional fuzzy controllers acting as fuzzy multiplexers to perform the fusion operation. Each fuzzy multiplexer may use sensor values, motivational state values, or the values of the sources themselves as inputs to its rule base. This allows behaviors to be smoothly blended. The networks shown in Figure 21 illustrate the behavior fusion scheme that allows MARGE to travel among obstacles in search of a goal location.

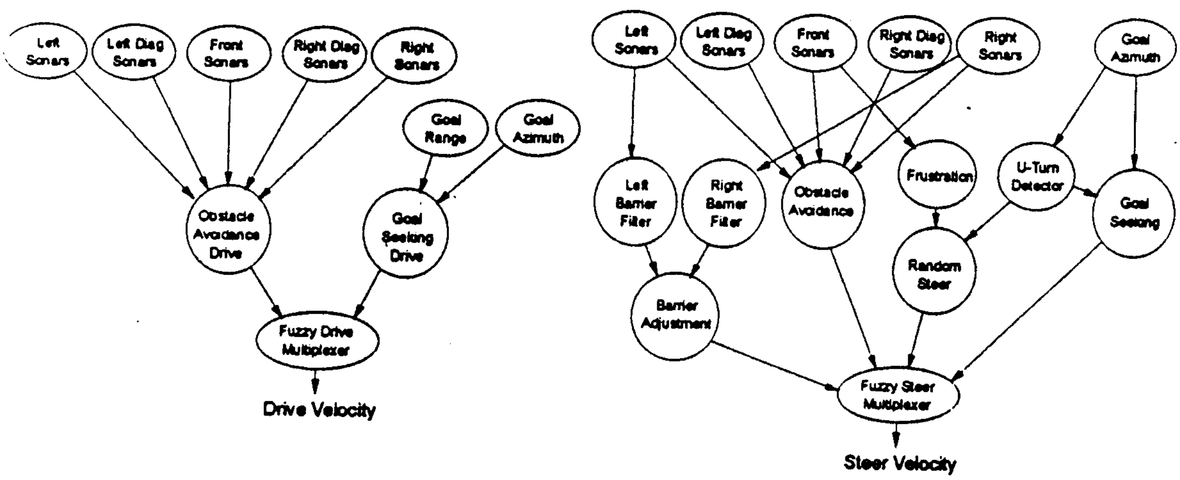


Figure 21. Hierarchy for fuzzy control of drive velocity and steering.

Sometimes, two behaviors may command opposite actions that would combine to form an ineffective control value, or a situation may arise where no clear choice is obvious. This can occur when a mobile robot meets an unexpected obstacle head on, for example. To recover from such situations, the control function must adapt by changing its control surface. One way to acheive this is by modellling internal motivation, such as frustration. On our robot, if a situation of indecision occurs and the robot cannont proceed, a random value is added in to the steer controller until the robot breaks out of its predicament. The gain of the random value increases with the "frustration" level, and the value changes to prevent endless loops in a local area.

Although the basic behaviors employed on MARGE are primitive, their

combined operation results in very powerful activity. For example, if a goal location exists on the other side of a barrier as depicted in Figure 22, the obstacle avoidance and goal seeking behaviors compete. This causes the robot to follow the barrier until it finds its way around it. Concave obstacles are easily escaped, and dynamic obstacles are dealt with in real time.

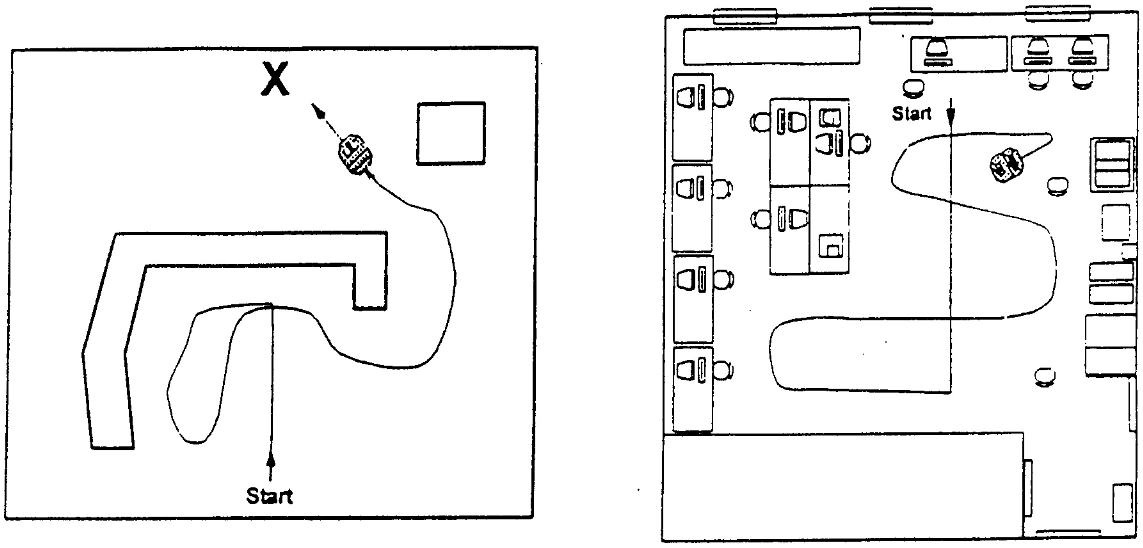


Figure 22. Goal seeking and obstacle avoidance behavior with no goal.

This behavior allows the robot to circumnavigate unmapped obstacles en route to a goal location. Even with no explicit goal, the robot may safely wander through its natural habitat, as shown in Figure 22. Note that this data is not the result of simulation, but was recorded by observing the actual motion of MARGE wandering through our laboratory.

Implementation in a Reactive Task

In order to test our system and compare it to others, MARGE was entered in the 1993 Mobile Robot Competition hosted by the American Association for Artificial Intelligence in Washington, D.C. The competition consisted of three events; MARGE won first place in the final event, "Office Rearrangement." This involved rapidly searching for and moving large boxes around obstacles into a prescribed pattern. The initial and final configurations of the competition arena are shown in Figure 23. Boxes were marked on one or two sides with 'X' and '+' signs. Such a task required a diversity of capabilities: visual target recognition, goal seeking, obstacle avoidance, position re-referencing, and environment manipulation. The fact that the objects and obstacles were randomly placed meant that building and maintaining a map would be difficult. Fortunately,

MARGE's reactive control architecture does not need a map to behave in a competent manner.

Our competition strategy was simple: A landmark recognition algorithm running on one processor was used to find signs and estimate their position in the room. These coordinates were then used as destinations for the goal seeking behavior. The robot wandered through the arena, avoided obstacles and traveled a serpentine path in order to cover a wider area with its cameras. When the landmark recognition program detected a sign, the fuzzy goal seeking behavior would drive to the box location. Once in range, the vehicle turned around to grab the boxes with its vacuum gripper, and towed them to the goal coordinates. MARGE was the only robot to reliably move boxes around obstacles or complete the task within the time limit. Details of this competition will are available in [ ].

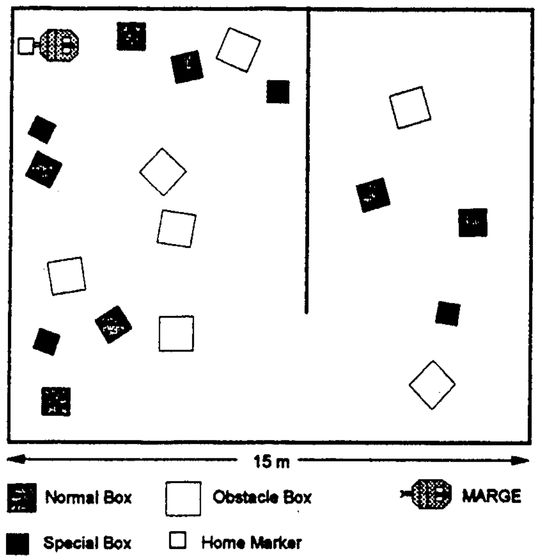


Figure 23. Initial layout of competition and completed box pattern.

List of Publications

(1) R.C. Luo and T.J. Pan, "An Intelligent Mobile Robot Path Planning System," SME Transaction on Robotics Research, Vol. 1, No. 1, 1990.

(2) T.J. Pan and R.C. Luo, "A Feasible Collison-Detection Algorithm for Mobile Robot Motion Planning with Moving Obstacles," IEEE IECON'91, Kobe, Japan, October 1991.

(3) M.G. Kay and R.C. Luo, "Camera Placement for Global Vision," 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC, July 1992.

(4) R.C. Luo, Harsh Potlapalli, and D.W. Hislop, "Translation and Shift Invariant Landmark Recognition Using Receptive Field Neural Networks," 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC, July 1992.

(5) R.C. Luo, Harsh Potlapalli and D.W. Hislop, "Translation and Scale Invariant Landmark Recognition Using Receptive Field Neural Networks," in 1992 IEEE Proceedings of International Robotics and Systems Conference, Raleigh, NC, July 1992.

(6) R.C. Luo, and Michael G. Kay, "Global Vision for the Control of Free-ranging AGV Systems", in Proceedings of the IEEE International Conference on Robotics and Automation, Atlanta, GA, May 1993.

(7) R.C. Luo, Harsh Potlapalli and D.W. Hislop, "Landmark Detection in Natural Scenes for Mobile Robot Navigation," in IEEE Transactions on Systems, Man and Cybernetics. *In Review* since 1 June 1993.

(8) R.C. Luo, Harsh Potlapalli and D.W. Hislop, "Landmark Recognition for Mobile Robots in Dynamic Environments Using Self-Organizing Neural Networks," in Proceedings of IEEE Robotics and Automation Conference, San Diego, CA, 1994. *In Review* since 1 September 1993.

(9) R.C.Luo, Harsh Potlapalli and D.W. Hislop, "Traffic Sign Recognition in Outdoor Environments Using Reconfigurable Neural Networks," in Proceedings of International Joint Conference on Neural Networks, Nagoya, Japan, 1993.

(10) Steven G. Goodridge and R.C. Luo, "Fuzzy Behavior Fusion for Reactive Control of an Autonomous Mobile Robot," in Proceedings of IEEE Robotics and Automation Conference, San Diego, CA, 1994.

(11) Steven G. Goodridge and R.C. Luo, "Fuzzy Behavior Fusion for Autonomous Mobile Robot Control," presented at the IEEE International Conference on Fuzzy Theory and Technology, Durham, NC, Oct. 16-18, 1993.

(12) J.A. Janet, R.C. Luo, et al., "Sonar Windows and Geometrically Representd Objects for Mobiel Robot Self-Referencing." *IEEE/RSJ International Conference on Intelligent Robotics and Systems*, 1993.

(13) J.A. Janet, Global Motion Planning and Self-Referencing for Autonomous Mobile Robots, MS Thesis. North Carolina State University, Raleigh, NC, 1994.

(14) J.A. Janet, S.G. Goodridge, Ren C. Luo, "Dynamic Motion Control of Sensor-Based Autonomous Mobile Robot." *APS International Conference on Mechatronics and Robotics*, April 1994, Aachen, Germany.

(15) J.A. Janet, R.C. Luo, M.G. Kay, "Autonomous Mobile Robot Motion Planning Enhanced with Extended Configuration-Space and Half-Planes." *IEEE/SICE/AEI International Conference on Industrial Electronics*, Sept. 1994, Bologna, Italy.

(16) J.A. Janet, R.C. Luo, M.G. Kay, "Traversability Vectors Make Autonomous Mobile Robot Motion Planning and Self-Referencing More Efficient." *IEEE/SICE/AEI International Conference on Intelligent Robotics and Systems*, Sept. 1994.

List of Participating Scientific Personnel

1.	Harsh Potlappali	Ph.D.
2.	Jason Janet	M.S.
3.	Steven G. Goodridge	M.S.

E. Bibliography

[Ark 90] Ronald C. Arkin, "Integrating Behavioral, Perceptual and World Knowledge in Reactive Navigation," *Robotics and Autonomous Systems* 6, pp. 105-122, 1990.

[Aur 91] Aurenhammer, F., "Voronoi Diagrams-A survey of fundamental geometric data structure." *ACM Computing Surveys* Vol. 23 no. 3, pp. 345-405, Sept 1991.

[Boz 91] Bozma, O. and R. Kuc, "Building a Sonar Map in a Specular Environment Using a Single Mobile Sensor." *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 13 no. 12, pp. 1260-1269, Dec 1991.

[Bro 86] Rodney A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, Vol. RA-2, No.1, pp 14-23, March 1986.

[Bro 91] Rodney A. Brooks, "Intelligence without representation," *Artificial Intelligence* 47, pp. 139-159, 1991.

[Cha 85] Chattergy, R., "Some Heuristics for the Navigation of a Robot." *Int'l Journal of Robotics Research* Vol. 4 no. 1, pp. 59-66, Spring 1985.

[Cox 91] Cox, I. J. and J. J. Leonard, "Probabilistic Data Association for Dynamic World Modeling: A Multiple Hypothesis Approach." in *Fifth International Conference on Advanced Robotics*, 1991, pp. 1287-1294.

[Cun 91] Glen B. Cunningham, "Integrating Fuzzy Logic Technology into Control Systems," technical document, Togai InfraLogic, Inc, Irvine, California, 1991.

[Elf 87] Elfes, A., "Sonar-Based Real-World Mapping and Navigation." *Journal of Robotics and Automation* Vol. 3 no. 3, pp. 249-265, June 1987.

[Eve 90] Everett, H. R.;G. A. Gilbreath, et al. (1990). Modeling the Environment of a Mobile Security Robot. Technical Document 1835. San Diego, CA, Naval Ocean Systems Center.

[Gat 93] Erann Gat, "On the Role of Stored Internal State in the Control of Autonomous Mobile Robots," *AI Magazine*, pp. 64-73, Spring 1993.

[Hwa 92] Hwang, Y. K. and N. Ahuja, "Gross Motion Planning-A Survey." *ACM Computing Surveys* Vol. 24 no. 3, pp. 219-291, September 1992.

[Ish 91] Shigeki Ishikawa, "A Method of Indoor Mobile Robot Navigation by Using Fuzzy Control," in Proceedings of IROS '91, the 1991 IEEE/RSJ International Workshop on Intelligent Robots and Systems, Nov 3-5, 1991, Osaka, Japan.

[Jan 93] Janet, J. A.;R. C. Luo , et al., “Sonar Windows and Geometrically Represented Objects for Mobile Robot Self-Referencing.” in *IEEE/RSJ International Conference on Intelligent Robotics and Systems*, 1993, pp. .

[Kon 92] Kurt Konolige, Karen Meyers, and Alessandro Saffiotti, "FLAKEY, an Autonomous Mobile Robot," technical document, Stanford Research Institute International, July 20, 1992.

[Kuc 89] Kuc, R. and B. Barshan, “Navigating Vehicles Through an Unstructured Envrironment With Sonar.” *Proc. 1989 IEEE Int'l Conference on Robotics and Automation* Vol. no. , pp. 1422-1426, 1989.

[Leo 91] Leonard, J. and H. Durrant-Whyte, “Mobile Robot Localization by Tracking Geometric Beacons.” *Transaction on Robotics and Automation* Vol. 7 no. 3, pp. 376-382, June 1991.

[Leo 91] Leonard, J. and H. Durrant-Whyte, “Simultaneous Map Building and Localization for an Autonomous Mobile Robot.” *IEEE/RSJ Int'l Workshop on Intelligent Robots and Systems IROS ‘91* Vol. no. , pp. 1442-1447, Nov 1991.

[Liu 90] Liu, Y. and S. Arimoto, “A Flexible Algorithm for Planning Local Shortest Path of Mobile Robots Based on Reachability.” in *International Workshop on Intelligent Robots and Systems*, 1990, pp. 749-756.

[Liu 91] Liu, Y.-H. and S. Arimoto, “Proposal of Tangent Graph and Extended Tangent Graph for Path Planning of Mobile Robots.” in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 312-317.

[Loz 87] Lozano-Perez, T., “Spatial Planning: A Configuration Space Approach.” *IEEE Transactions on Computers* Vol. C-32 no. 2, pp. 108-119, Feb. 1987.

[Loz 79] Lozano-Perez, T. and M. A. Wesley, “An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles.” *Communication* Vol. ACM 22 no. 10, pp. 560-570, October 1979.

[O'Du 85] O'Dulaing, C. and C. K. Yap, “A retraction method for planning the motion of a disc.” *Journal of Algorithms* Vol. 6 no. , pp. 104-111, 1985.

[Oom 87] Oommen, B. J.;S. S. Iyengar, et al., “Robot Navigation in Unknown Terrains Using Learned Visibility Graphs. Part I: The Disjoint Convex Obstacle Case.” *IEEE Journal of Robotics and Automation* Vol. RA-3 no. 6, pp. , Dec 1987.

[Pan 90a] Pan, T.-J. (1990). Robotic Motion Planning in a Dynamic Environment with Moving Obstacles. North Carolina State University.

[Pan 90b] Pan, T.-J. and R. C. Luo, “Motion Planning for Mobile Robots in a

Dynamic Environment with Moving Obstacles." in *IEEE Internat'l Conf. on Robotics and Automation*, 1990, pp. .

[Pin 92] F.G. Pin, H. Watanabe, J.R. Symon, and R.S. Pattay, "Using Custom-Designed VLSI Fuzzy Inferencing Chips for the Autonomous Navigation of a Mobile Robot" in Proc. of IROS 92, the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, July 7-10, 1992, Raleigh, North Carolina.

[Ram 85] Ramanathan, B. and V. S. Algar, "Algorithmic motion planning in robotics: coordinated motion of several disks amidst polygonal obstacles." in *Proc. of 1985 IEEE International Conference of Robotics and Automation*, 1985, pp. 514-522.

[Rao 88a] Rao, N. S. V. and e. al., "On terrain model acquisition by a point robot amidst polyhedral objects." *IEEE Journal of Robotics and Automation* Vol. 4 no. 4, pp. , 1988.

[Rao 88b] Rao, N. S. V.;N. Stoltzfus, et al., "A 'Retraction' Method for Terrain Model Acquisition." in *Proc. 1988 Int. Conf. Robotics and Automation*, 1988, pp. 1224-1229.

[Sna 91] Martin Snaith and Owen Holland, "An Investigation of Two Mediation Strategies for Behavioral Control in Animals and Animats," *From Animals to Animats,* MIT Press, 1991.

[Son 92] K.Y. Song and J. C. Tai, "Fuzzy Navigation of a Mobile Robot," in Proceedings of IROS 92, the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, July 7-10, 1992, Raleigh, North Carolina.

[Sug 89] M.Sugeno, T. Murofushi, T. Mori, T. Tatematasu, J. Tanaka, "Fuzzy Algorithmic Control of a Model Car by Oral Instructions," *Fuzzy Sets and Systems,* No. 32, pp. 207-219, September 11, 1989.

[Tin 51] N. Tinbergen, *The Study of Instinct,* Oxford University Press, 1951.

[Tak 89] Takahashi, O. and R. J. Schilling, "Motion Planning in a Plane Using Generalized Voronoi Diagrams." *IEEE Transactions on Robotics and Automation* Vol. 5 no. 2, pp. 143-150, April 1989.

[Wan 91] Wang, L. and W. Tsai, "Collision Avoidance by a Modified Least-Mean-Square-Error Classification Scheme for Indoor Autonomous Land Vehicle Navigation." *Journal of Robotic Systems* Vol. 8 no. 5, pp. 677-698, 1991.

[Yap 84] Yap, C. K. (1984). Coordinating the motion of several discs. Computer Science Department, New York University.

[Zad 65] L. A. Zadeh, "Fuzzy Sets," *Info. and Control,* No 8, pp. 338-353, June 1965.

[Zad 88] Lofti A. Zadeh, "Fuzzy Logic," *IEEE Computer,* pp. 83-93, April 1988.

[Zel 91] Zelinsky, A., “Mobile Robot Map Making Using Sonar.” *Journal of Robotic Systems* Vol. 8 no. 5, pp. 557-577, 1991.