(to appear in ***Robotica***, 1994)

Fuzzy Logic Based Collision Avoidance For a Mobile Robot

Angelo Martinez, Eddie Tunstel[1], and Mo Jamshidi
CAD Laboratory for Intelligent and Robotic Systems
Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, NM 87131 USA

Abstract: Navigation and collision avoidance are major areas of research in mobile robotics that involve varying degrees of uncertainty. In general, the problem consists of achieving sensor based motion control of a mobile robot among obstacles in structured and/or unstructured environments with collision-free motion as the priority. A fuzzy logic based intelligent control strategy has been developed here to computationally implement the approximate reasoning necessary for handling the uncertainty inherent in the collision avoidance problem. The fuzzy controller was tested on a mobile robot system in an indoor environment and found to perform satisfactorily despite having crude sensors and minimal sensory feedback.

1. INTRODUCTION

An aim of intelligent robotics research is to develop mobile robots capable of navigating autonomously in unstructured and/or unexplored environments. This development requires intelligent control strategies capable of overcoming the uncertainties presented by the real world. To ensure the safety of the robot system it is necessary that it be able to navigate without colliding with obstructions in its environment. A number of methodologies have been proposed for achieving collision avoidance. Some of the more popular methods are based on edge-detection, certainty grids, potential fields, or combinations of these [1-6]. In edge-detection the goal is to detect the visible edges of an obstacle; the lines connecting these edges represent the obstacle boundaries. This information can be used to decide on an appropriate direction to take for avoiding collision. One of the disadvantages of this approach is the necessity for the robot to stop and execute the algorithm that determines the existence of edges. Since many of these systems employ ultrasonic sensors, other disadvantages related to the shortcomings of these sensors detract from the system's ability to detect edges accurately. Typical problems associated with ultrasonic sensors are specular reflections, external interference from nearby sources, and the poor directionality characteristic of sonic waves. The certainty grid approach, as used in collision avoidance, consists of representing the robot's environment as a two-dimensional grid of cells. To each cell is associated a probability measure of the existence of an obstacle in that cell. The certainty grid is eventually used for planning obstacle-free paths. This technique is often used in the process of aquiring a local scan of the robot's immediate surrounding area, thus requiring that the robot stop periodically. Finally, potential field methods have also been used for mobile robot collision avoidance by many researchers following the seminal paper by Khatib [3]. The method consists of a superposition of attractive and repulsive forces resulting in an optimal direction of travel. Obstacles in the robot's environment exert repulsive forces on the robot and the goal state applies an attractive force. The success of the potential field method is dependent upon predefined environmental models; it thus requires accurate descriptions of obstacle locations in static environments.

We have employed fuzzy logic to facilitate the collision-free navigation task for a mobile robot. A fuzzy logic based system has the advantage that it allows the intuitive nature of obstacle

---

[1] On Leave from NASA Jet Propulsion Laboratory, Pasadena, CA.

avoidance to be easily modeled using linguistic terminology. In addition, with the proposed methodology we can relax some of the constraints associated with the aforementioned methods. The computational load of fuzzy inference systems, for instance, is considerably lighter than those of edge-detection, certainty grids and potential fields. Moreover, as demonstrated in this paper, accurate (expensive) sensors and detailed models of the environment are sufficient but not necessary for obstacle avoidance. The successful management of uncertain and/or unmodeled data is a proven attribute of fuzzy logic based inference engines. In this work, our objective is to demonstrate the feasibility of this approach for mobile robot sensor-based obstacle avoidance in unknown, semi-structured environments using a minimal hardware system.

Fuzzy logic based controllers are expert control systems that smoothly interpolate between rules. Rules fire to continuous degrees and the multiple resultant actions are combined into an interpolated result. Processing of uncertain information and saving of energy using common-sense rules and natural language statements are the basis for fuzzy logic control. The use of sensor data in practical control systems involves several tasks that are usually done by a human in the loop, e.g. an astronaut adjusting the position of a satellite or putting it in the proper orbit, an operator remotely teleoperating a robotic manipulator or vehicle, etc. All such tasks must be performed based on evaluation of the sensor data according to a set of rules/heuristics that the human expert has learned from experience or training. Often, if not most of the time, these rules are not crisp (based on binary logic), i.e. some common-sense or judgemental-type decisions are needed. The class of such problems can be addressed by a set of fuzzy variables and rules which, if done properly, can make expert decisions [7]. As pointed out by Lee [8], fuzzy logic controllers provide a means of transforming the linguistic control strategy based on expert knowledge into an automatic control strategy. Fuzzy control appears to be very useful for handling problems that are too complex for analysis using conventional quantitative techniques *or* when the available sources of information provide qualitative, approximate, or uncertain data. The control of mobile robots falls into this class of problems, and in the remainder of the paper we discuss the hardware implementation of a fuzzy logic controller for mobile robot collision avoidance. In Section 2 we present our approach to the collision avoidance problem. This is followed by a description of the mobile robot and accompanying hardware system in Sections 3 and 4. The fuzzy controller and supporting software are discussed in Section 5. A brief discussion of some experimental results of actual trial runs in an indoor environment is provided in Section 6. Finally, concluding remarks and recommendations are given.

2. APPROACH

It was decided that a PC-based control scheme be used to process sensory information and provide control action decisions. Obstacle range data is transmitted via tether from the mobile system to the PC which in turn transmits vehicle drive and steering commands. The fuzzy logic controller consists of a rule base that fuzzifies range values and produces defuzzified control actions (see Figure 1). The control actions are crisp values that represent robot velocity and steering directions.

Any information about the robot's environment is provided solely by a pair of ultrasonic transducers and two short-range infrared proximity sensors. That is, no feedback of kinematic or dynamic information (such as position and velocity) is employed. The use of such an impoverished sensor suite allows us to examine the power of fuzzy inferencing when constrained by minimal observability. From a practical point of view, this constraint is realistic relative to most autonomous mobile systems with limited computational resources. It has been demonstrated that such systems exhibit satisfactory performance when utilizing reactive and/or behavioral control strategies [9, 10].

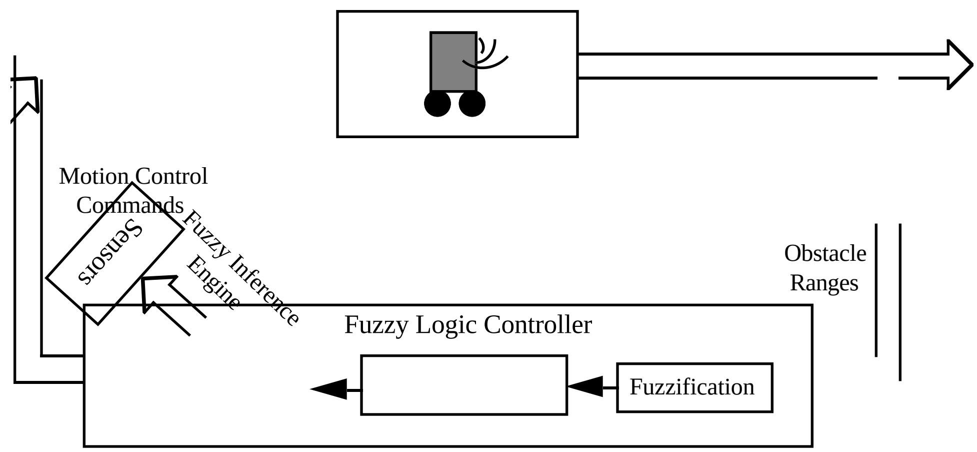


Figure 1 General fuzzy control architecture

Reactive control can be described as a sense-act cycle wherein each motion control decision is based only on current sensory input. That is, there is a tight coupling between the robot's sensors and actuators. This permits intelligent decisions to be made in real-time, thus allowing for uninterrupted robot motion. In intelligent systems, the reactive control strategy dictates faster control cycles than more conventional (sense-*plan*-act) systems since sensors are sampled at a higher rate. This near continuous sampling of sensor readings reduces the undesirable effects of inherently noisy sensor data. Behavior control, which makes reactive control possible, involves decomposing the motion control system into a set of special-purpose routines (behaviors) that achieve simple but distinct tasks when subject to particular stimuli. Clever combinations of these simple behaviors results in the emergence of more intelligent behavior suitable for dealing with more complex situations. Fuzzy logic control and reactive/behavior control share common advantages for designing intelligent systems. Namely, they both require short development times and incorporate flexibility in their design. Individual fuzzy rules (and behaviors) can be formulated independently, and additional rules (and behaviors) can easily be added to the control system if necessary. The advantages to be gained from combining the desirable characteristics of fuzzy logic and behavior-based control have recently been recognized by several researchers [11,12].

2.1 *Collision Avoidance*

For collision avoidance we have developed a reactive control system using fuzzy logic. Our collision avoidance system implements an intelligent reactive control strategy in that it senses the immediate environment and reacts (with the aid of approximate reasoning) to current sensory data. For example, if the robot is in close proximity to an obstacle ahead of it, the fuzzy logic controller will determine the best direction to proceed (at that instant) in order to avoid colliding with the obstacle. The robot will then immediately "react" according to the directive issued by the fuzzy controller. Such reactive behavior allows a mobile robot to avoid static and dynamic obstacles without requiring predefined world models. Using this sense-act cycle we avoid some of the computational bottlenecks associated with the planning phase of the sense-plan-act cycle traditionally employed in artificial intelligence systems. In general, collision avoidance is a fundamental requirement while traversing a prespecified path en route to some goal location. In fact, the majority of the work cited in the references of this paper concentrates on either the problem of concurrent path following and collision avoidance, or collision avoidance for future goal-directed path planning. In this paper, we do not address path planning or goal-directed navigation. The absence of a goal allows us to concentrate fully on the fundamental task of collision avoidance while circumventing (for the time being) the problem of collision avoidance *subject to* the parametric constraints of a particular path plan. The general algorithm consists of a sense-act cycle where the robot senses nearby obstacles and reacts in an appropriate manner to achieve the goal of avoiding impact. We focus mainly on the low level task of safe traversal throughout a given unknown environment. With this capability the vehicle can later be programmed for goal-directed navigation tasks simply by the addition of higher level path planning heuristics.

The sense-act cycle begins with the robot continuously surveying its immediate frontal and lateral area for the presence of an obstacle. The robot moves in a forward direction while surveying until an obstacle is detected within an adjustable fuzzy range threshold. At this point the steering direction and drive speed are modified by the fuzzy logic controller such that the robot veers away from the potential impact. The range of the robot's steering angle permits rotation in place (in either direction) when the steering reaches a mechanical limit. As for the drive speed, we are currently using a total of seven different speeds with a maximum of about 2.5-3.0 feet/second. We should point out here that the robot operating speed is proportional to the sensor sampling rate which is, in turn, a function of the number of sensors being used. For situations where many sensors are necessary, moderate to fast speeds can still be achieved by realizing the fuzzy controller at the "chip level". This suggests the use of VLSI implementations of fuzzy logic controllers as proposed in [11] and [13]. At any instant in the control cycle the fuzzy controller considers the current state of the world and decides the appropriate combination of steering and drive commands necessary to avoid collisions. In the following section we describe the mobile robot used as a testbed for implementing this approach. The software implementation of this fuzzy control strategy is discussed in Section 5.

3. MOBILE ROBOT SYSTEM

Our system uses the Hero-1 educational mobile robot manufactured by Heath Co. [14]. The Hero-1 is a tricycle-type vehicle with the front wheel used for both driving and steering. A stepper motor is used for the steering and a simpler DC motor is used for translational motion. In addition, the Hero-1 is equipped with an onboard sonar ranging system. The Hero-1 is battery-powered and can be programmed using an 8-bit Motorola 6808 microprocessor and associated circuitry mounted onboard. The robot can operate as a stand-alone unit under programmed control using only its onboard microprocessor. In our implementation the 6808 is used exclusively for providing sonar and proximity data while the PC performs associated control calculations. The 6808 allows simple I/O interfacing between the robot and an external PC. The available I/O operations enable acquisition of sonar range data directly from the microprocessor CPU. Similarly, the I/O capability facilitates the reception of drive and steering commands (from the PC) by the mobile robot.

The ranging system transmits short ultrasonic pulses (32kHz) and measures the time until the first echo of the pulse returns. The time is proportional to the distance the sound has traveled, thus giving a measure of obstacle distance relative to the robot. Empirical observations showed that the range of reliable obstacle detection is approximately 0.88 - 6.67 feet. The Hero-1 represents range readings as dimensionless 8-bit hexadecimal numbers. It was determined that the readings can be converted to decimal units of feet by down-scaling the decimal representation of the hexadecimal number by a factor of 36_{10}. For example, if the current range reading is 73_{16} (115_{10}), then the corresponding reading in feet is $115_{10}/36_{10}$ = 3.2 ft. (or 1m).

4. SYSTEM HARDWARE REALIZATION

Although the Hero-1 can operate as a stand-alone system, it was necessary to construct and attach a tethered interface between the robot and PC in order to implement PC based control. To augment the onboard sonar ranging system we constructed and mounted two infrared (IR) proximity detectors on either side of the robot. The IR detectors are comprised of an IR LED-phototransistor pair with associated interface circuits. Our initial IR detectors exhibited an effective range of about 4-6 inches. We were able to increase this range to 8-11 inches by using a comparator circuit which serves to magnify the IR sensing capability. The comparator chip establishes a distinct threshold in the range of continuous voltages output by the detector beyond which an obstacle is considered present or not present. The output of this network consists of two binary values indicating the existence of obstacles on either side of the Hero-1. Pertaining to the ultrasonic ranger, the robot presents frontal range data in the form of an 8-bit binary number. Hence, a small amount of 6808 assembler code was written to enable the presentation of range data to the robot's I/O port. Thus, the robot is equipped with sensing straight ahead of it and to its sides. In our collision avoidance system we only utilize forward motion of the robot. Hence, rear sensing is not necessary. Figure 2 summarizes the implementation of the sense-portion of the fuzzy reactive control strategy discussed in Section 2.

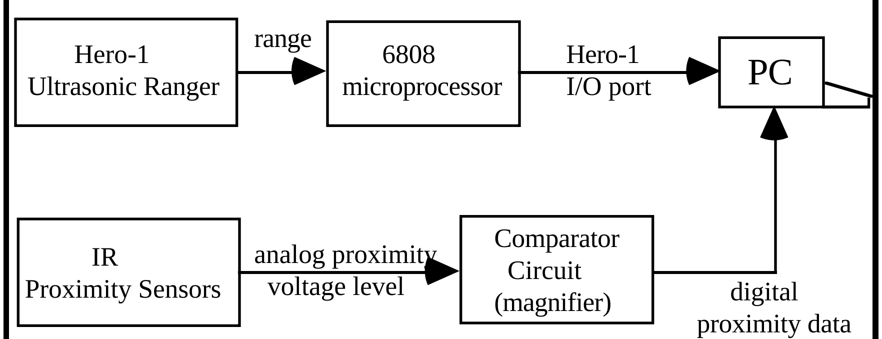


Figure 2 Block Diagram of Sensor Implementation

The application of motor control commands required the consideration of steering and drive commands as separate entities. Figure 3 illustrates this division and presents a diagram of the actuation hardware necessary to establish motion control. The digital drive command from the PC is first transformed into an analog voltage level through the use of a digital-to-analog converter IC. This voltage level is then used as the control input for a pulse-width-modulation circuit. Modulation of this input results in the creation of a square-wave signal whose duty-cycle is controlled by the PC drive command. Finally, the drive signal is passed through a power transistor circuit which provides a high-current voltage source capable of powering the DC drive motor.

The fact that the robot's front wheel is steered with a four-phase stepper motor required that a circuit be constructed to generate the necessary phasing. To this end, a sequencing network utilizing logic ICs was developed which accepts the binary steering command as input and provides an ensemble of square-waves as an output. These square-waves contain the proper phase information to accurately control the steering motor direction. As in the case of the drive motor, the last stage of the steering control network is comprised of power transistors to supply sufficient current to the motor. Schematics for motor control and other robot control circuitry are readily available in many sources. Consult references [15-17] for good examples of these.

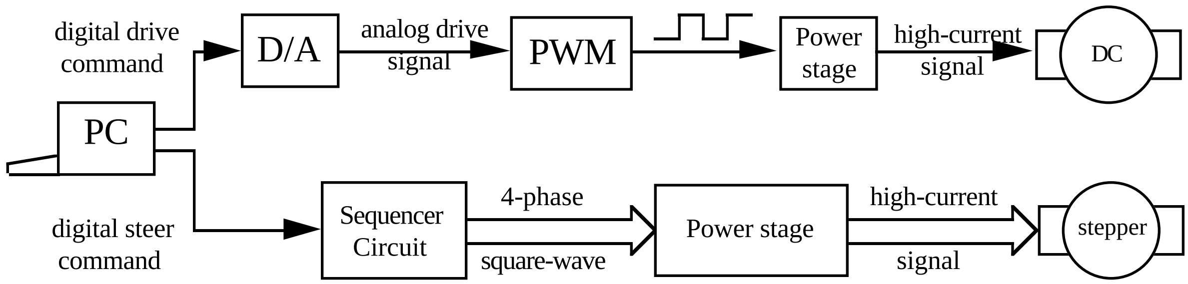


Figure 3 Block Diagram of Actuator Implementation

5. FUZZY CONTROLLER SOFTWARE

The software development for a fuzzy logic control system is facilitated by the use of available tools dedicated to fuzzy rule base generation. For our purposes a commercial software package compatible with C code was desired so that any necessary pre/post-processing (written in C) could be easily linked to the overall software system. We chose the TIL Fuzzy-C [18] development tool due to the ease with which a fuzzy rule base can be generated and compiled into C code. Fuzzy-C facilitates writing, testing, and debugging fuzzy logic expert systems. Rule bases can be constructed using the TIL Fuzzy Programming Language and portable ANSI C subroutines can be generated and called from user-supplied driver routines.

A fuzzy controller typically takes the form of a set of IF-THEN rules whose antecedents and consequents are membership functions. Consequents from different rules are numerically combined (typically union via MAX) and are then collapsed to yield a single real number output. Fuzzy sets may be represented by a mathematical formulation often known as the membership function. This function gives a degree or grade of membership within the set. The membership function of a fuzzy set A , denoted by $\mu_{A(X)}$ maps the elements of the universe X into a numerical value within the range [0,1], i.e.

$$\mu_{A(X)}: X \text{ ------------> } [0,1] .$$

Note that a membership function is a so-called *possibility* function and not a probability function. In control system applications membership values are actually measure of degree of causality in an input-output mapping. Fuzzy logic gave a new definition to the causality in dynamic systems. Within this framework, a membership value of zero corresponds to a value which is definitely not an element of the fuzzy set, while a value of one corresponds to the case where the element is definitely a member of the set [7]. Implementation of a fuzzy controller requires assigning membership functions for both inputs and outputs. Our fuzzy logic controller is a three input, two output system. The inputs are frontal obstacle range, and left and right obstacle proximities; the outputs are the steering direction and vehicle speed. Membership functions and their corresponding linguistic labels are shown in Figure 4 where:

CL = "clear"	NC = "not clear"	Z = "zero"
VNEAR = "very near"	VFAR = "very far"	VVS = "very very slow"
VS = "very slow"	S = "slow"	F = "fast"
M = "medium"	VF = "very fast"	VVF = "very very fast".

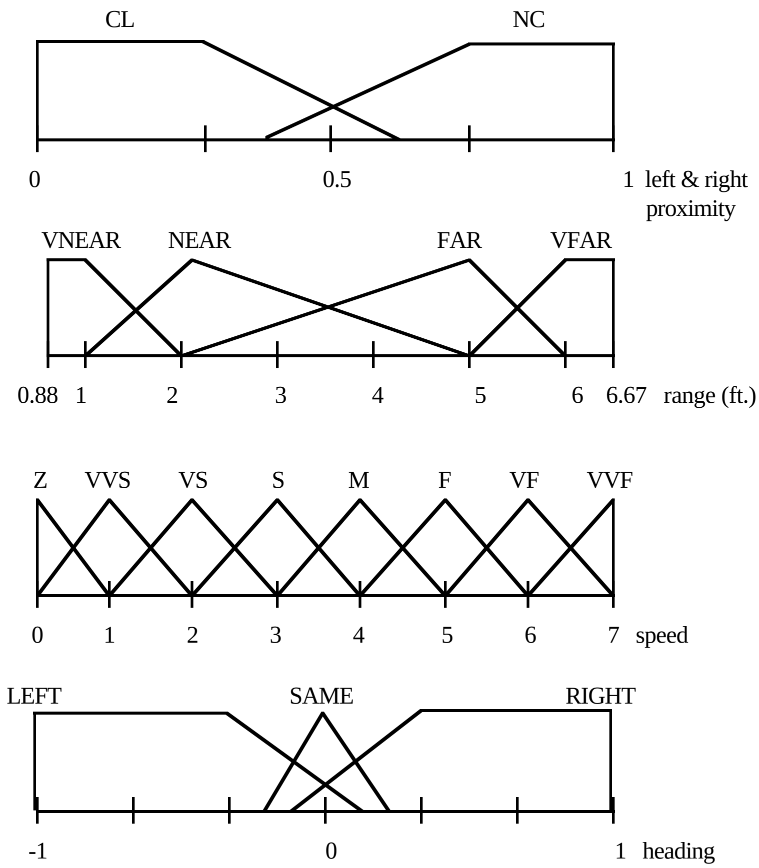


Figure 4 Membership functions for collision avoidance

The fuzzy rule base for our system consists of a set of IF-THEN rules that map fuzzy range variables into fuzzy heading and speed commands. These commands are defuzzified using the max-product defuzzification method. For more details on fuzzification and defuzzification processes, see [7,18]. Typical IF-THEN rules in our collision avoidance system are

IF range is NEAR and left-prox is CL and right-prox is NC
THEN heading is LEFT and speed is SLOW

IF range is FAR and left-prox is NC and right-prox is NC
THEN heading is SAME and speed is FAST

Using a set of rules such as these, a finite number of rules can be derived in the form of natural language statements as if a human operator was performing the controlling task. We have used a total of 16 rules of the above form in our initial design. Through experimentation and tuning of the membership functions, we found that this number of rules was sufficient to encompass all realistic combinations of inputs and outputs.

5.1 *Distributed Processing*

The software system is implemented as a distributed processing system in that the PC and the 6808 microprocessor each contribute to the overall control system by performing asynchronous tasks. The processors also communicate with each other according to an established handshaking protocol. The protocol was developed specifically for the current application; it ensures that all data required by the processors are made available at the right times during the control cycle. Realization of this distributed processing system requires that both the PC and the robot monitor their respective I/O ports for the appropriate handshake signals. Periodic observation of these ports is achieved through C language subroutines and 6808 assembler code for the PC and Hero-1, respectively. Proper implementation of this strategy permits accurate management of two-way data transmission. Thus, continuous communication between the PC and the robot is maintained at all times.

The software-hardware interface is achieved by digital data transmission across the PC parallel port. The parallel port provides eight digital output lines and five digital input lines. Thus, sensory input from the robot (sonar range and infrared proximities) is represented with a 5-bit resolution. Output to the robot is represented as an 8-bit control byte which packages motion control bits as shown below.

c7	c6	c5	c4	c3	c2	c1	c0

The control bits, c0 through c7, are defined as follows:

c0 => send sensor data	c1-c3 => drive speed (binary 0-7)
c4 => drive motor on/off	c5 => not used
c6 => steering direction	c7 => steering on/off

The digital sensory data is *pre*-processed in software prior to input to the fuzzy controller while the control outputs of the fuzzy controller are *post*-processed for transmission to the robot. This pre- and post-processing is necessary since the fuzzy controller accepts and outputs floating point numbers.

6. EXPERIMENTS

To assess the performance of the fuzzy collision avoidance system experiments were conducted using the Hero-1 in an indoor laboratory and hallway environment. An illustration of a typical scene is shown in Figure 5. The obstacles consisted of empty boxes and office furniture.

As performance metrics we observed the smoothness of the robot's path. We consider a smooth path to consist of minimal oscillatory motion when traversing straight corridors, and the least curvature during right-angle turns. Figure 6 illustrates this idea. Four types of experiments were run to test the mobile collision avoidance system in situations likely to be encountered indoors. These experiments included: (i) traversal though straight and narrow passageways, i.e. corridors with widths that are less than twice the robot's transverse diameter, (ii) head-on approach towards obstacles, (iii) partially blocked hallways with free space on one side of the obstacle, and (iv) dead-end situations. Each of these scenarios is depicted in Figure 7(a)-(d) respectively. In experiment (i) the robot successfully traversed the narrow corridor along a fairly smooth path. It has been determined [1] that in this situation, robots controlled using potential fields may tend to oscillate due to repulsive forces exerted simultaneously from opposite sides of the robot. In our experiments we did not encounter such oscillatory behavior. When approaching obstacles in experiment (ii) the robot exhibited a very smooth deceleration before stopping short of colliding with static obstacles. As might be expected, a more abrupt halt was observed when dynamic obstacles (e.g. humans) were encountered. This response can possibly be improved through tuning of appropriate membership functions and/or fuzzy rules. By tuning we mean modifying the shapes/locations of the membership functions (Figure 4) and modifying the composition of fuzzy rule antecedents and consequents.

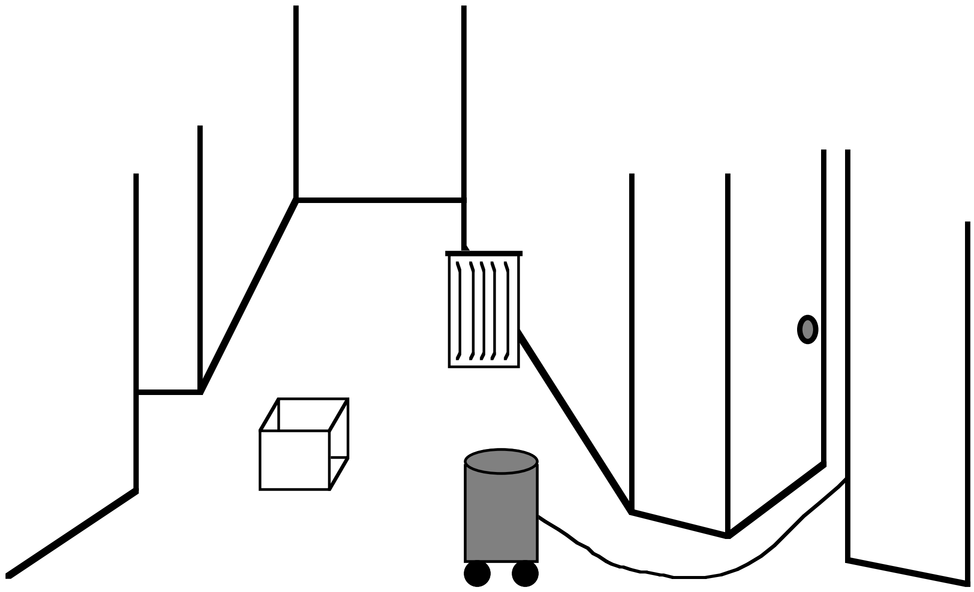
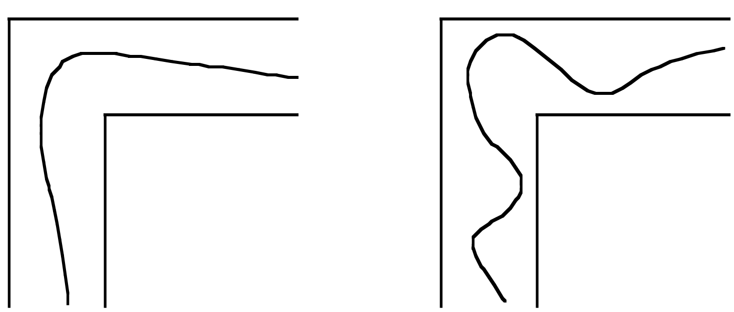


Figure 5 Hallway Environment

.



(a) Smooth path

(b) Irregular path

Figure 6 Path examples

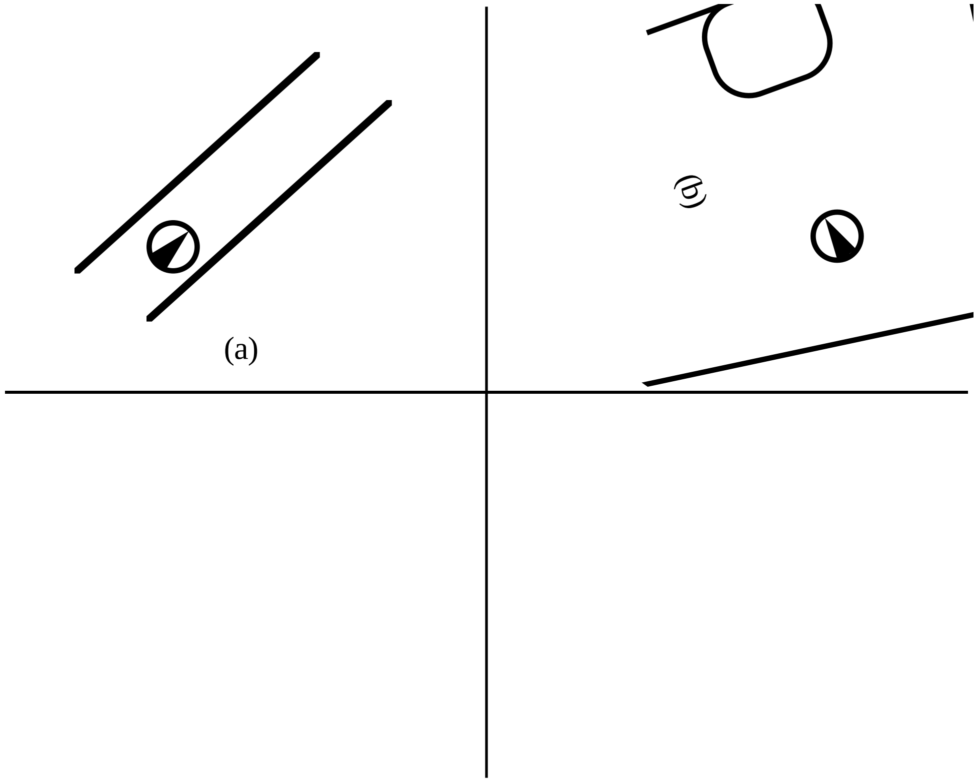


Figure 7 Four Experimental Situations for Collision Avoidance

In partially blocked hallways, experiment (iii), the robot always avoided collision but its behavior effected a tendency to wander about in the neighborhood of the obstacle. This can be attributed to the fact that the robot was not provided with a particular goal, perhaps on the other side of the obstacle. Experiment (iv) resulted in similar behavior. The robot could successfully escape from dead-end situations, but with no *particular* place to go it would sometimes wander into the same dead-end again.

7. CONCLUSION

Trial runs of the robot through the laboratory and hallways demonstrated that the robot was capable of rudimentary collision avoidance limited by the lack of steering angle feedback. The robot wandered about reactively using the sonar and IR proximity data to successfully avoid impact with obstacles. However, it was observed that the reflective characteristics of obstacles influenced the quality of range and proximity data operated on by the fuzzy controller. In addition, the motion control actions dictated by the fuzzy controller were not always carried out successfully due to intermittent stalling problems with the steering and drive motors.

Further research on this topic should include an investigation of utilizing position and velocity feedback. Such feedback would permit straight line drive capability and much smoother path traversal. The sensor arrangement and quantity should also be improved. The implementation of overlapping sonar and/or IR sensors would maximize the coverage of the robot's immediate environment. Better imaging capability would then lead to improved flexibility in the fuzzy rule base. That is, the availability of additional information facilitates the development of higher level algorithms for goal-directed behavior and path planning. In addition, to eliminate the problems with motor control the circuitry could be modified to deliver more power and torque.

Fuzzy logic control proved to be a satisfactory control strategy for the mobile robot collision avoidance problem. It exhibited intelligent behavior in the face of the uncertainty presented by previously unknown environments. Having investigated the hardware implementation of a fuzzy logic based collision avoidance system, it was discovered that numerous problems not typically accounted for in simulation studies are likely to arise. Two examples of these are, (1) possible failure of drive motors resulting in the interruption of robot motion, and (2) variations in the reflective characteristics of real-world objects that might influence sensor readings. Simulations rarely impose such realistic uncertainties and random variabilities of the actual world. As a result the utility of simulation studies is dependent on how well the simulation model approximates the real world. Overall, this project has served as groundwork for further meaningful investigation into the intelligent control of mobile robots.

REFERENCES

[1] Koren, Y. and Borenstein, J., "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation", *1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, April 1991, pp 1398-1404.

[2] Borenstein, J. and Koren, Y., "The Vector Field Histogram--Fast Obstacle Avoidance for Mobile Robots", *IEEE Transactions on Robotics and Automation*, Vol 7, No 3, June 1991, pp 278-288.

[3] Khatib, O. "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", *The International Journal of Robotics Research*, Vol 5, No 1, 1986, pp. 90-98.

[4] Kuc, R. and Barshan, B., "Navigating Vehicles Through an Unstructured Environment With Sonar", *1989 IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, May 1989, pp 1422-1426.

[5] Moravec. H.P., "Sensor Fusion in Certainty Grids for Mobile Robots", *AI Magazine*, Summer 1988, pp 61-74.

[6] Holenstein, A.A., and Badreddin, E., "Collision Avoidance in a Behavior-Based Mobile Robot Design", *1991 IEEE International Conference on Robotics and Automation*, Sacremento, CA, April 1991, pp 898-903.

[7] Jamshidi M., Vadiee N. and Ross T. (Eds) Fuzzy Logic and Control -- Software and Hardware Applications. Prentice-Hall, Englewood Cliffs, NJ (1993).

[8] Lee, C.C., "Fuzzy Logic in Control Systems: Fuzzy Logic Controller -- Part I", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol 20, No 2, March/April, 1990, pp 404-418.

[9] Gat, E. "Robust Low-Computation Sensor-Driven Control for Task-Directed Navigation", *1991 IEEE International Conference on Robotics and Automation*, Sacremento, CA, April 1991, pp 2484-2489.

[10] Tunstel, E., "Autonomous Robot Mapping With Behavior-based Navigation", *JPL Internal Document* , JPL D-10458, December 1992.

[11] Pin, F.G, Watanabe, H., Symon, J. and Pattay, R.S., "Using Custom-Designed VLSI Fuzzy Inferencing Chips for the Autonomous Navigation of a Mobile Robot ", *1992 IEEE /RSJ International Conference on Intelligent Robots and Systems*, Raleigh, NC, July 1992, pp 790-795.

[12] Goodridge, S. and Luo, R.C., "Fuzzy Behavior Fusion for Autonomous Mobile Robot Control", to appear in *The Second Fuzzy Theory and Technology Conference*, Durham, NC, October 1993.

[13] Yamakawa, T., "A Fuzzy Inference Engine in Nonlinear Analog Mode and its Application to a Fuzzy Logic Control", *IEEE Transactions on Neural Networks*, Vol 4, No 3, May 1993, pp 496-522.

[14] *HERO Robot Model ET-18 Technical Manual*, Heath Company, Benton Harbor, MI, 1983.

[15] Robillard, Mark J., Advanced Robot Systems, Howard W. Sams & Co, Indianapolis, IN, 1984.

[16] McComb, Gordon, The Robot Builder's Bonanza, TAB Books, Blue Ridge Summit, PA, 1987.

[17] Jones, J.L. and Flynn, A.M., Mobile Robots: Inspiration to Implementation, A K Peters, Wellesley, MA, 1993.

[18] Hill, G., Horstkotte, E. and Teichrow, J., *Fuzzy-C Development System User's Manual*, Release 2.1, Togai InfraLogic Inc., Irvine, CA, June 1989.