p 22

Real-Time Fuzzy Inference Based Robot Path Planning

Final Report

JOHNSON GRANT
IN-61-CR

(NASA-CR-188598) REAL-TIME FUZZY INFERENCE BASED ROBOT PATH PLANNING Final Report (Houston Univ.) 17 p CSCL 09B

N91-25631

Unclas
G3/61 0020001

Peter J. Pacini
Jon S. Teichrow

Togai InfraLogic, Inc.

August 17, 1990

Cooperative Agreement NCC 9-16
Research Activity No. AI.13

NASA Johnson Space Center
Information Systems Directorate
Information Technology Division

ricis

Research Institute for Computing and Information Systems
University of Houston - Clear Lake

T·E·C·H·N·I·C·A·L R·E·P·O·R·T

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information systems in 1986 to encourage NASA Johnson Space Center and local industry to actively support research in the computing and information sciences. As part of this endeavor, UH-Clear Lake proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a three-year cooperative agreement with UH-Clear Lake beginning in May, 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The mission of RICIS is to conduct, coordinate and disseminate research on computing and information systems among researchers, sponsors and users from UH-Clear Lake, NASA/JSC, and other research organizations. Within UH-Clear Lake, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business, Education, Human Sciences and Humanities, and Natural and Applied Sciences.

Other research organizations are involved via the "gateway" concept. UH-Clear Lake establishes relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research.

A major role of RICIS is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. Working jointly with NASA/JSC, RICIS advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research, and integrates technical results into the cooperative goals of UH-Clear Lake and NASA/JSC.

Real-Time Fuzzy Inference Based Robot Path Planning

Final Report

Peter J. Pacini
Jon S. Teichrow

Togai InfraLogic, Inc.

August 17, 1990

Cooperative Agreement NCC 9-16
Research Activity No. AI.13

NASA Johnson Space Center
Information Systems Directorate
Information Technology Division

Research Institute for Computing and Information Systems
University of Houston - Clear Lake

T·E·C·H·N·I·C·A·L R·E·P·O·R·T

Preface

This research was conducted under auspices of the Research Institute for Computing and Information Systems by Peter J. Pacini and Jon S. Teichrow of Togai InfraLogic, Inc. Dr. Terry Feagin served as RICIS research coordinator.

Funding has been provided by Information Technology Division, Information Systems Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA technical monitor for this activity was Robert N. Lea, of the Software Technology Branch, Information Technology Division, Information Systems Directorate, NASA/JSC.

Real-Time Fuzzy Inference Based Robot Path Planning

Project # AI•13
Subcontract # 46

by
Peter J. Pacini • Jon S. Teichrow

Togai InfraLogic, Inc.
30 Corporate Park, Suite 107
Irvine, CA 92714
(714) 975-8522

August 17, 1990

Project Summary

This project addresses the problem of adaptive trajectory generation for a robot arm. Conventional trajectory generation involves computing a path in real time to minimize a performance measure such as expended energy. This method can be computationally intensive, and it may yield poor results if the trajectory is weakly constrained. Typically some implicit constraints are known, but cannot be encoded analytically.

The alternative approach used in this study is to formulate domain-specific knowledge, including implicit and ill-defined constraints, in terms of fuzzy rules. These rules utilize linguistic terms to relate input variables to output variables. Since the fuzzy rulebase is determined off-line, only high-level, computationally light processing is required in real time.

Potential applications for adaptive trajectory generation include missile guidance and various sophisticated robot control tasks, such as automotive assembly, high speed electrical parts insertion, stepper alignment, and motion control for high speed parcel transfer systems

Table of Contents

Table of Figures

1.0 Introduction

The intent of this Phase I effort was to develop an efficient, adaptive alternative to conventional trajectory generation. Instead of solving analytically for the optimal path at run time, we encode domain-specific knowledge and constraints off-line in terms of linguistic fuzzy rules. This new approach significantly reduces run time computational requirements. The software simulation accompanying this report graphically demonstrates the feasibility of such an approach.

This report is organized as follows: Section 2 is a high-level description of the robot arm trajectory simulation. Section 3 describes the user interface, including the various windows on the simulation screen, available user commands, and puck trajectory options. Section 4 describes the three classical guidance algorithms, and a default algorithm, used to control the robot arm. Section 5 discusses the fuzzy expert system which rates the algorithms' effectiveness at each sampling interval. Section 6 briefly describes the catch mode, which is invoked at the time of interception. Section 7 presents the conclusion of the Phase I effort and covers proposed Phase II tasks.

2.0 Simulation Overview

In the simulation accompanying this report, a robot arm moving along an adaptively generated trajectory intercepts and grapples a puck. The user first positions the arm by selecting one of four setup commands. Then he selects attributes of the puck trajectory from a menu. When the puck is released, the robot arm moves to intercept it.

At each sampling interval, a fuzzy expert system evaluates three classical guidance algorithms, rating their effectiveness for the current situation. The most effective algorithm then determines the arm's new heading. The user sees which algorithm is executing at all times, because a list of guidance methods is displayed with the current method highlighted in red.

After the arm intercepts and catches the puck, the simulation freezes until the user presses a key. Then the puck disappears, and the arm remains idle until the next user command. At this point the user can select the Trails command to see the arm and puck trajectories. All of the available user commands are described in section 3.2, as well as on a run-time help menu.

3.0 User Interface

The user interface consists of a command buttons window on the simulation screen and a puck trajectory menu, which appears when the Pucks command is executed.

3.1 Simulation Screen

The simulation screen contains six windows, as shown in Figure 3.1.

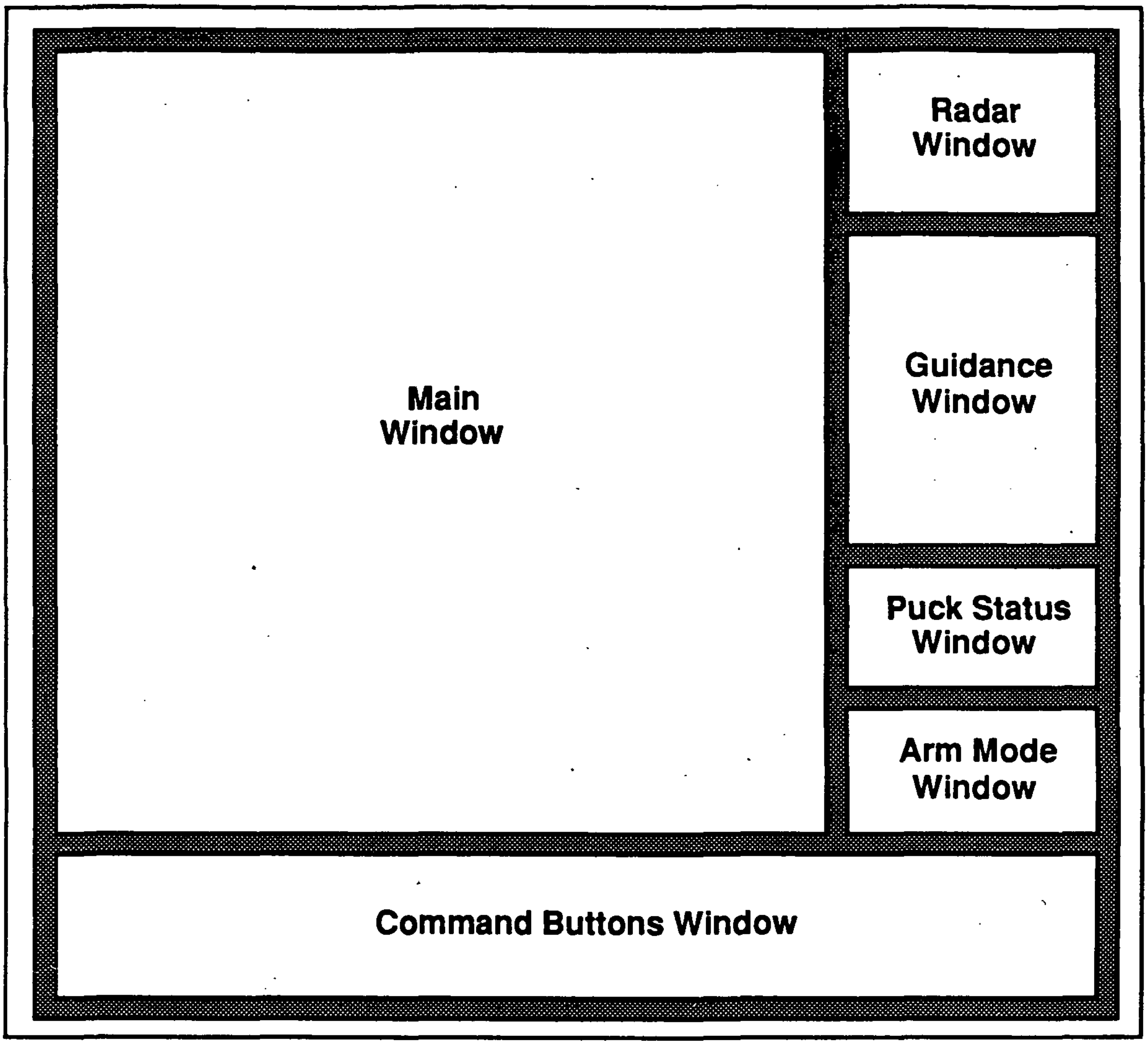

Figure 3.1 - The six windows on the simulation screen.

1) Main Window - The robot arm appears with its base at the bottom center of the window. The effector at the end of the arm always points in the direction of movement and remains fully extended until a puck comes within catching range. When the arm heading changes suddenly, the effector will turn toward the new heading at its maximum turn rate of 5 degrees per second. The current arm heading and speed are printed in red just to the right of the effector.

   When a puck is launched, it appears as a green ball with a red border. The puck's heading and speed are printed in cyan immediately to its right. All arm and puck headings are in the range -180 to 180 degrees, where 0 degrees is horizontal. A set of axes with 0 degrees labeled remains in the lower right corner of the window at all times.

2) Radar Window - The radar window is a scaled down version of the main window with a few additional features. The maximum range of the robot arm is indicated by a red arc. Pucks are visible as soon as they are launched, even before they appear in the main window. Also, the arm and puck trails are visible whenever the arm is active.

3) Guidance Window - The guidance window displays the four guidance algorithms. The active algorithm is highlighted in red, while the others are printed in yellow. When the puck is not in use, Arm Test Demo is highlighted.

4) Puck Status Window - The puck status window shows whether or not a puck has been launched.

5) Arm Mode Window - The arm mode window displays one of eight possible operational modes. Four are seek modes, including Seek Left Base, Home, Right Base, and User X,Y. The other four are Exercise, Intercept, Catch, and Arm Idle.

6) Command Buttons Window - The command buttons window displays the ten user commands as rectangular buttons. A red border marks the currently selected button.

3.2 User Commands

Figure 3.2 shows the command buttons window. Ten user commands are provided. They are selected with the arrow keys and activated by pressing ENTER.

Go Left	Go Home	Go Right	Goto XY	Exercise
Pucks	Trails	Pause	Help	Quit

Figure 3.2 - Command buttons window.

1) Go Left - Moves the arm to its left base position and points the effector at 45 degrees.

2) Go Home - Extends the arm vertically and points the effector at 90 degrees.

3) Go Right - Moves the arm to its right base position and points the effector at 135 degrees.

4) Goto XY - Moves the arm to a user-specified location. Red crosshairs indicate the arm's destination. The arrow keys position the crosshairs, and the ENTER key starts the arm toward its new destination.

5) Exercise - Continually moves the arm to random destinations.

6) Pucks - Displays a puck trajectory menu. The user selects puck speed, path and origin from the available options. ENTER launches the puck. ESCape returns to the simulation screen without releasing a puck.

7) Trails - Shows the puck trail in yellow and the arm trail in green. The trails appear in both the main window and the radar window.

8) Pause - Pauses the simulation until another key is pressed.

9) Help - Displays a help menu with a list of user commands and a description of the four guidance algorithms.

10) Quit - Ends the simulation.

3.3 Puck Trajectory Menu

A puck trajectory menu appears when the Pucks command is executed. The TAB key selects the current field, while the up and down arrow keys select an option within the current field.

The first field specifies the puck speed. Slow, medium and fast speeds are 40, 60 and 90 mm/s, respectively. Variable speed pucks begin at medium speed, and for every iteration a random number between -10 and 10 is added to the previous puck speed. The second field specifies an arced or straight path. The third field determines the puck's starting position. Three locations on either side of the screen are possible.

4.0 Guidance Algorithms

While in intercept mode the robot arm is controlled by one of four guidance algorithms.

1) Constant Bearing - The arm speed and puck velocity vector are known. Assuming no change in these parameters, the sine law determines the arm heading of a straight intercept course.

2) Proportional - The change in arm heading is proportional to the change in the line-of-sight angle.

3) Optimal - The change in arm heading is proportional to the change in distance times the change in the line-of-sight angle.

4) Line of Sight - The arm heading equals the line-of-sight angle.

At each sampling interval, a fuzzy expert system determines an effectiveness rating between -1.0 (not at all effective) and 1.0 (extremely effective) for each of the first three algorithms. The most effective algorithm is then used to update the arm heading. If all three algorithms are rated below a threshold of -.65, then Line of Sight is executed to orient the arm.

5.0 Fuzzy Algorithm Selection

Figure 5.1 shows the variables needed to assess each guidance algorithm's effectiveness.

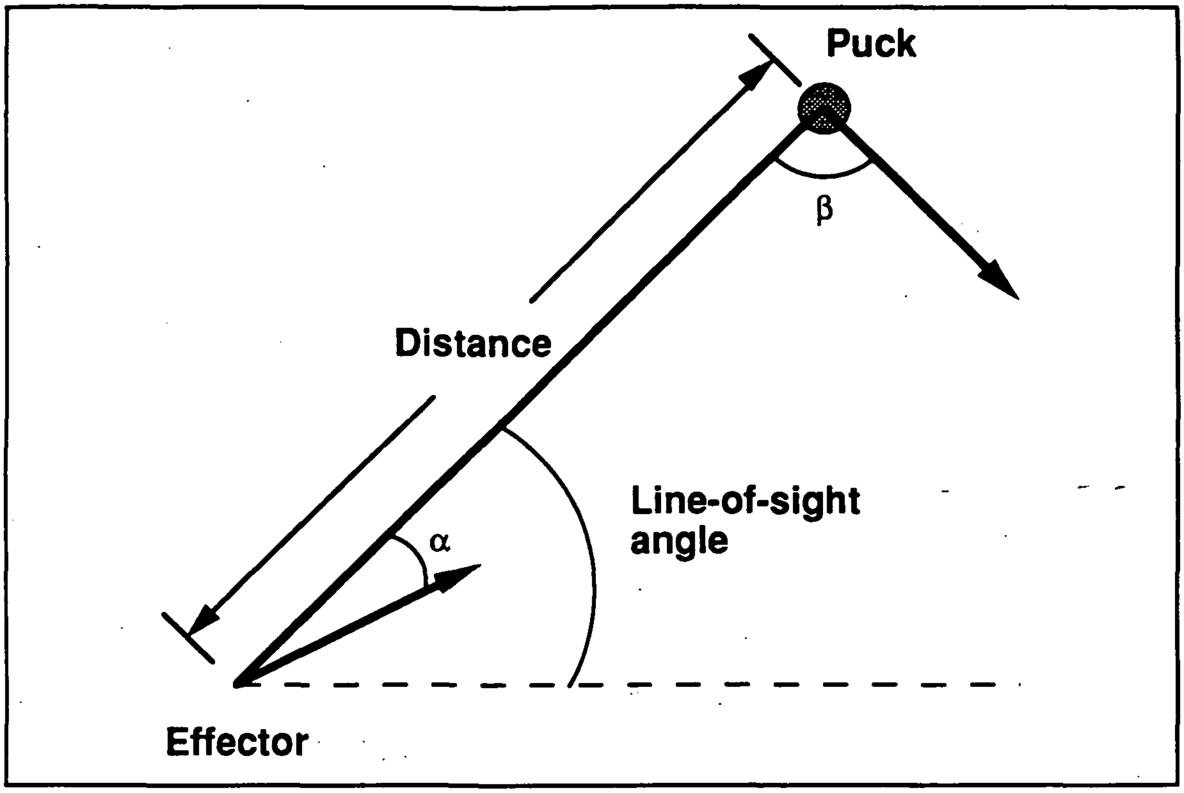

Figure 5.1 - Variable definitions for the fuzzy expert system.

Alpha is the angle formed by the line of sight and the arm's velocity vector. Beta is the angle formed by the line of sight and the puck's velocity vector. Only the magnitude of these angles is considered. The third input variable is the distance between the effector and the puck.

5.1 Fuzzy System Overview

The fuzzy expert system mentioned in section 4.0 was developed using Togai InfraLogic's CASE tool, TILShell. Figure 5.2 shows a high-level description of the system.

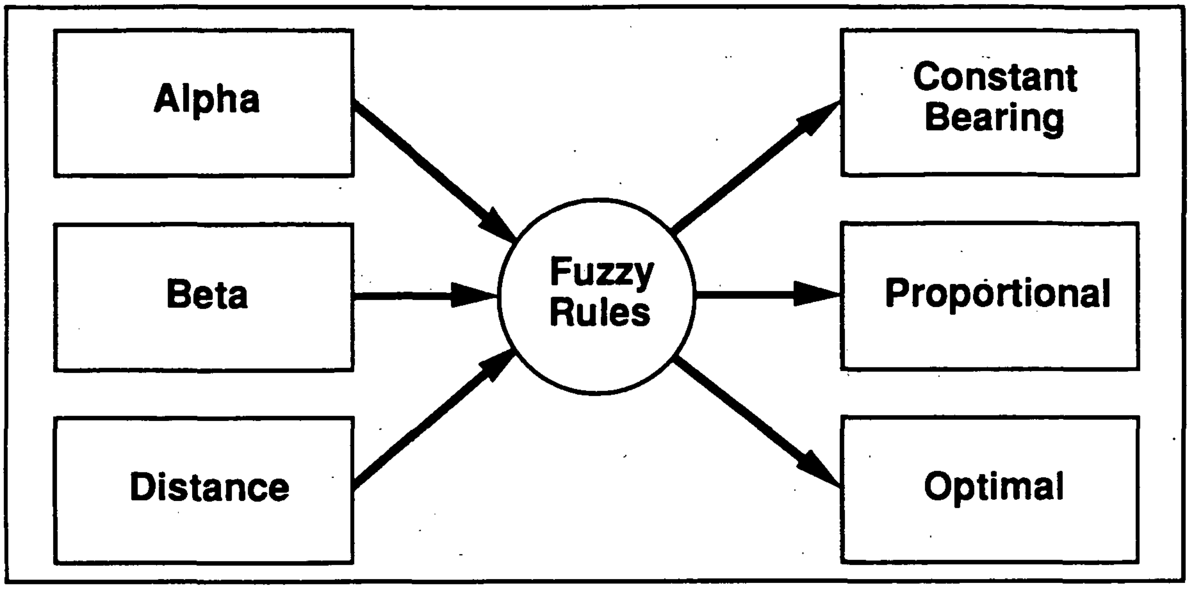

Figure 5.2 - High-level description of the fuzzy expert system.

Inputs appear on the left side of the screen, with connections leading to the fuzzy rulebase in the center. The outputs on the right are the three guidance algorithms to be evaluated.

5.2 Input Membership Functions

Figures 5.3 through 5.5 show the input membership functions (or fuzzy sets) for Alpha, Beta, and Distance. For each input variable, four membership functions are defined: Very Small (VS), Small (SML), Medium (MED), and Big.

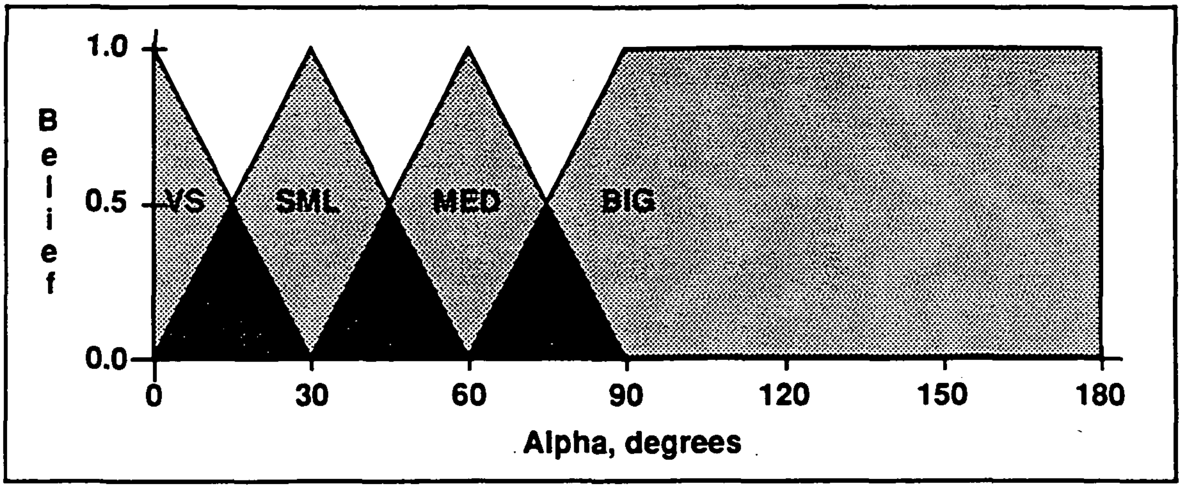

Figure 5.3 - Alpha membership functions.

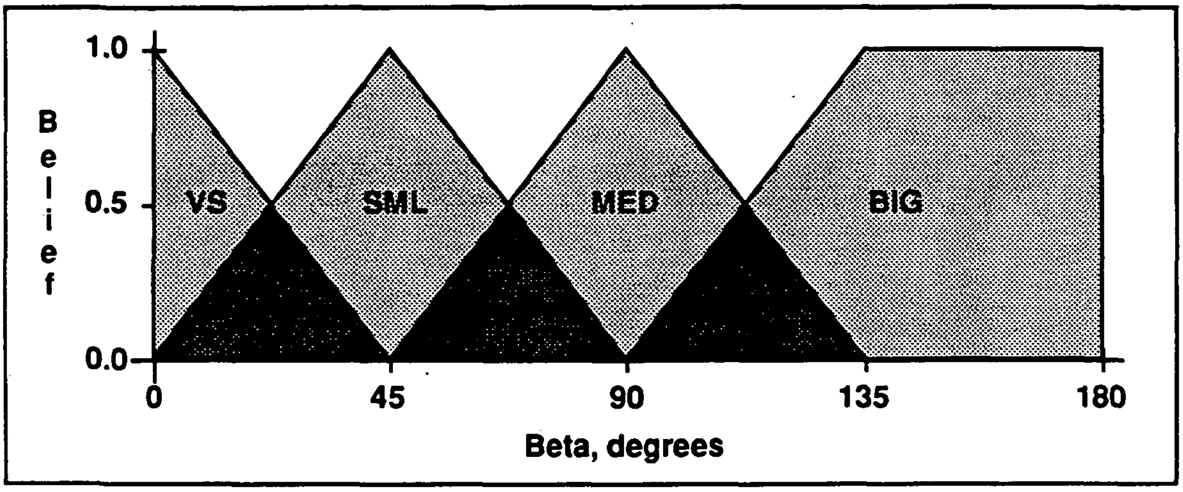

Figure 5.4 - Beta membership functions.

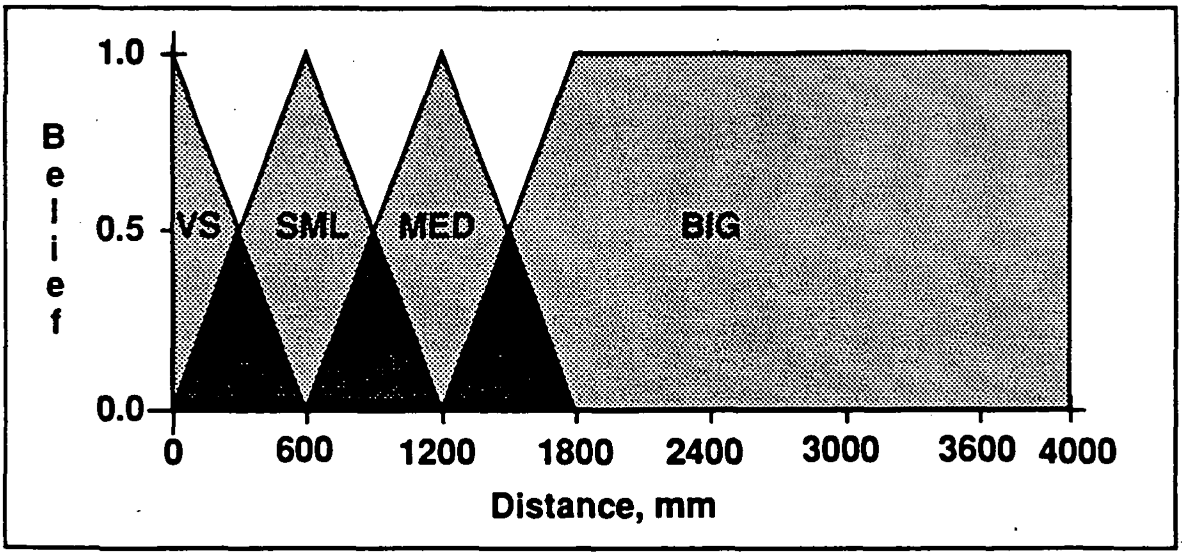

Figure 5.5 - Distance membership functions.

Consider a particular input value, say Alpha = 40 degrees. The belief level for each Alpha membership function at the point Alpha = 40 is the degree to which Alpha is a member of that fuzzy set. For example, when Alpha = 40 its degree of membership in all four fuzzy sets can be described as follows.

(Alpha is VS) to degree 0.00

(Alpha is SML) to degree 0.67

(Alpha is MED) to degree 0.33

(Alpha is BIG) to degree 0.00

Notice in Figure 5.3 that Alpha values greater than 90 degrees are considered Big to degree 1.0. As long as the user sets up the arm to the correct side before releasing a puck, Alpha will rarely exceed 90 degrees. Hence, the membership functions can be compressed toward smaller values to give finer control.

5.3 Output Membership Functions

All three outputs are described by the same set of membership functions, as shown in Figure 5.6.

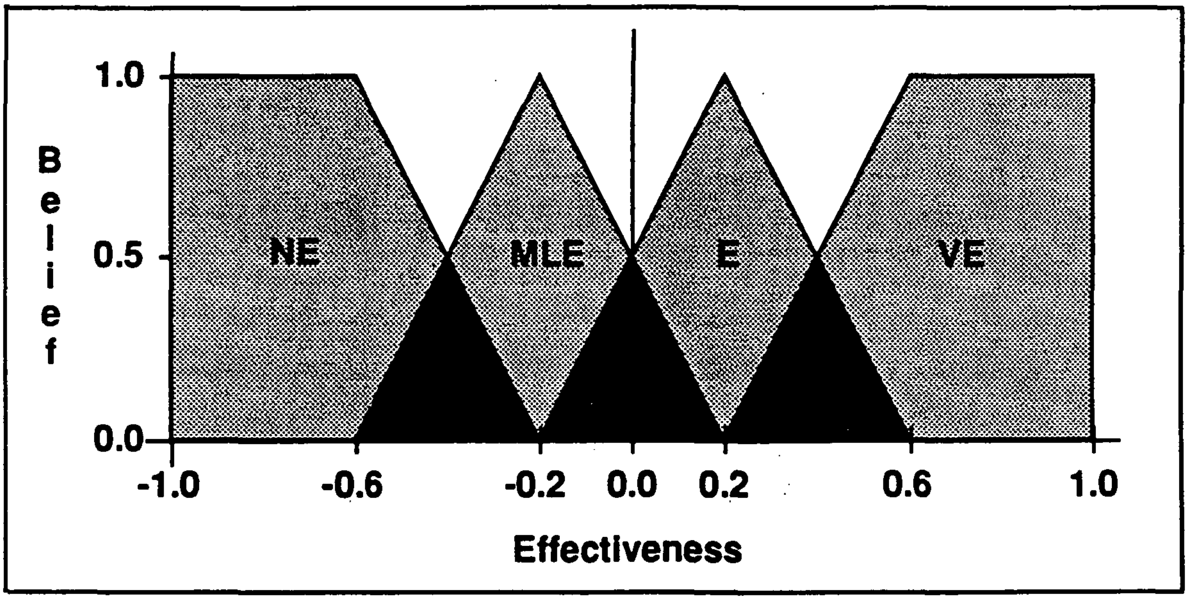

Figure 5.6 - Output membership functions.

From left to right, the membership functions are Not Effective (NE), More or Less Effective (MLE), Effective (E), and Very Effective (VE).

5.4 Fuzzy Rules

Each rule in the fuzzy rulebase is of the form

```
IF (Alpha is SML) AND (Beta is BIG) AND (Distance is SML)
                  THEN Constant Bearing is NE,
                        Optimal is MLE,
                   and  Proportional is E.
```

This rule has three outputs, each of which is a fuzzy set. All three output fuzzy sets are scaled by the belief level of the premise, also known as the output activation level. The belief level of each antecedent is determined by the input membership functions, as described in section 5.3. Since the antecedents are combined with AND, the output activation level is simply the minimum of the three antecedent belief levels.

For each of the three output variables -- Constant Bearing, Optimal, and Proportional -- the scaled output fuzzy sets of all the rules are summed to form a combined output fuzzy set. The centroid of this combined output fuzzy set is the effectiveness rating for that particular algorithm.

6.0 Catch Mode

As the arm and puck approach an interception point, the arm enters Catch mode. During the next six time intervals, the effector closes, and the arm slows down from its maximum speed of 60 mm/s to approximately zero at the point of contact. During this slow down phase, the fuzzy expert system still determines the most effective guidance algorithm, although the arm speed is controlled by a different equation.

After the slow down phase, the arm and puck both move along the puck's path, slowing down over four time intervals. When they have stopped, the simulation freezes until any key is pressed. This apparent catching motion was included strictly for aesthetic appeal. The fuzzy expert system does not operate after contact is made with the puck.

7.0 Conclusion and Phase II Plans

The robot arm trajectory simulation clearly demonstrates that domain-specific knowledge can be encoded off-line as a set of linguistic fuzzy rules. Using only these rules and membership functions to define the linguistic terms, the fuzzy expert system can quickly decide which guidance algorithm is most effective under the current circumstances.

Togai InfraLogic currently markets a chip called the FC110 designed for efficient processing of fuzzy rules. For the fuzzy system used in this simulation -- 3 inputs, 3 outputs, and 38 rules -- the FC110 operating at 20 MHz can return crisp effectiveness outputs in 315 μs. Since the FC110 is a general purpose microprocessor, a Single Board Fuzzy Computer (SBFC) utilizing this chip could control a robot arm in real time with no need for additional computer power.

In Phase II of this project, an industrial robot will be controlled by the methods described in this report. For the physical implementation, system dynamics omitted from the simulation model must be included. Hence, the guidance algorithms will be more complicated, and the fuzzy rules determining their effectiveness will have to be modified. When the rulebase is complete, it will be downloaded to an EPROM on the SBFC. The SBFC, along with the required sensors and interface, are all the hardware necessary to control the robot arm in real time. The completed system, with some modifications, will be applied to control a space-based robot arm.