N91-27106

FUZZY LOGIC CONTROL SYSTEM TO PROVIDE AUTONOMOUS COLLISION AVOIDANCE FOR MARS ROVER VEHICLE

Final Report

NASA/ASEE Summer Faculty Fellowship Program--1990

Johnson Space Center

Prepared By:	Michael G. Murphy, Ph.D.
Academic Rank:	Professor
University & Department:	University of Houston - Downtown Department of Applied Math. Sciences Houston, Texas 77002

NASA/JSC

Directorate:	Information Systems
Division:	Information Technology
Branch:	Software Technology
JSC Colleague:	Robert N. Lea, Ph.D.
Date Submitted:	August 10, 1990
Contract Number:	NGT-44-005-803

ABSTRACT

NASA is currently involved with planning unmanned missions to Mars to investigate the terrain and process soil samples in advance of a manned mission. A key issue involved in unmanned surface exploration on Mars is that of supporting autonomous maneuvering since radio communication involves lengthy delays. It is anticipated that specific target locations will be designated for sample gathering. In maneuvering autonomously from a starting position to a target position, the rover will need to avoid a variety of obstacles such as boulders or troughs that may block the shortest path to the target. The physical integrity of the rover needs to be maintained while minimizing the time and distance required to attain the target position.

Fuzzy logic lends itself well to building reliable control systems that function in the presence of uncertainty or ambiguity. Systems already developed by Robert N. Lea of NASA/Johnson Space Center and others test out well for shuttle proximity operations and for Mars rover trajectory control in the absence of obstacles. One of the tools investigated for developing a fuzzy logic obstacle avoidance control system is the Togai InfraLogic TILShell fuzzy logic expert system shell that provides a software development tool to express fuzzy logic rules as well as crisp rules and procedural code segments, then generates standard C code for implementation. This shell has been successfully used in the proximity operations and rover trajectory systems. Fuzzy logic control systems have been underutilized in American application areas but have been very successfully employed in a variety of commercial and industrial applications in Japan. An additional technology that shows promise of applicability to a variety of control system environments (such as the one addressed here) is that of learning and adaptation based on neuronlike elements.

This report addresses three major issues;

1. the nature of fuzzy logic control systems and software tools to implement them;
2. collision avoidance in the presence of fuzzy parameters; and,
3. techniques for adaptation in fuzzy logic control systems.

In addition to current work on these issues, possible directions for future work are given.

INTRODUCTION

The Space Exploration Initiative of the United States includes plans for unmanned missions by NASA to Mars to investigate the terrain and to process soil samples in advance of a manned mission. Path planning is a crucial element in providing for this exploration. More specifically, surface exploration of Mars involves an environment where uncharted obstacles are likely to interfere with maneuvering of a rover vehicle and where radio transmissions to and from Earth involve lengthy delays, hence the need for autonomy of operation of the vehicle. In moving autonomously from a starting position to a target position, the rover will need to avoid a variety of obstacles such as boulders or troughs that may block the shortest path to the target. The problem is to maintain the physical integrity of the rover while minimizing the time and distance required to attain the target position.

Although the problem of autonomous path planning including collision avoidance is not new, the research into the use of fuzzy logic in the decision and control process is new. The theory of fuzzy sets and fuzzy logic has matured since 1965 when it was introduced by Professor L. A. Zadeh of the University of California, Berkeley (1). The book by Klir and Folger (2) gives a good treatment of the fundamentals of this field. What has been lacking until the last few years are significant applications of this body of knowledge to practical endeavors in business and industry. This lack has been partially addressed by Japan, where a variety of applications involving fuzzy logic have been developed. One of the most important types of application has been fuzzy logic control systems (3,4). Togai Infralogic, Inc., of California has recently marketed a fuzzy logic expert system shell to facilitate the production of software for fuzzy logic applications(5). Robert N. Lea of NASA/JSC and others have been working on fuzzy logic applications to space vehicle rendezvous and proximity operations (6,7,8). Since autonomous path planning with collision avoidance is structurally a control problem and since fuzzy parameters are likely to be present in the Mars rover maneuvering environment, a fuzzy logic control system would be a promising avenue of investigation.

FUZZY LOGIC CONTROL SYSTEMS AND SOFTWARE TOOLS

The purpose of control systems is to achieve and then maintain a desired state of a process by monitoring the output and state of the process then taking responsive control actions. An example would be monitoring the temperature and pressure of a chemical process then raising, maintaining, or lowering the temperature to achieve or keep an optimal state for the process. Since temperature and pressure are likely to change during operation, this monitoring needs to be continued at regular time intervals. It is also desirable that the control be automated so that operator intervention is minimized (but permitted).

Traditional automatic controllers usually require a sufficiently precise mathematical model of the process. For many complex control environments, it is very difficult if not impossible to construct a satisfactory model due to issues such as nonlinearity, time variance, or poor quality of available measurements.

In cases such as these, a human operator may be able to effect superior control over traditional automatic controllers. An expert human operator consciously or unconsciously uses heuristic control rules or "rules of thumb" based on experience to control the process. When articulated by the operator, these rules usually involve linguistic descriptions with vague or ambiguous concepts. An example could be "if the speed of a conveyor belt is very fast, then the likelihood of an accident is high" with the terms "very", "fast", and "high" being defined only approximately. Terms such as "fast" in this context are considered values of the linguistic variable "speed", where the degree of membership μ could be given by means of a graph, such as the one in Figure 1 below, where the horizontal axis takes on possible values for speed, and the vertical axis gives the degree (0 to 1) that a particular speed is a member of the linguistic value above it. Of course, the membership function could be of any form and not necessarily piecewise linear. The term "very" would usually be interpreted as the square of the membership of "fast" in our context.

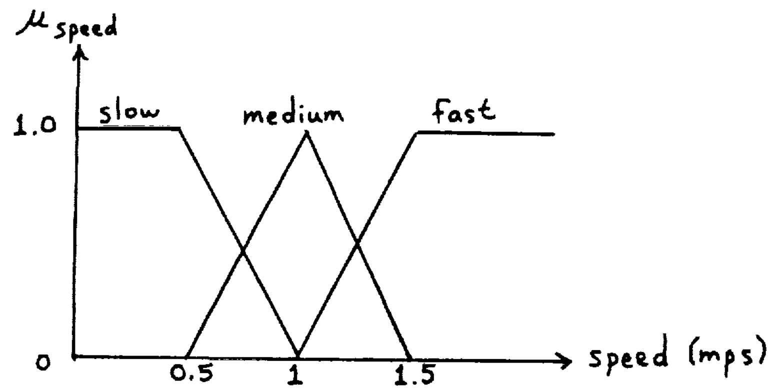


Figure 1. - Graphical representation of linguistic values of "speed" for conveyor.

A rule-based fuzzy logic controller consists of a finite set of fuzzy control rules which are processed by applying what is called approximate reasoning. A typical fuzzy logic control rule is of the form:

If x is A and y is B, then z is C.

Figure 2 below indicates how the memberships of x in A and y in B yields the membership level of z in C.

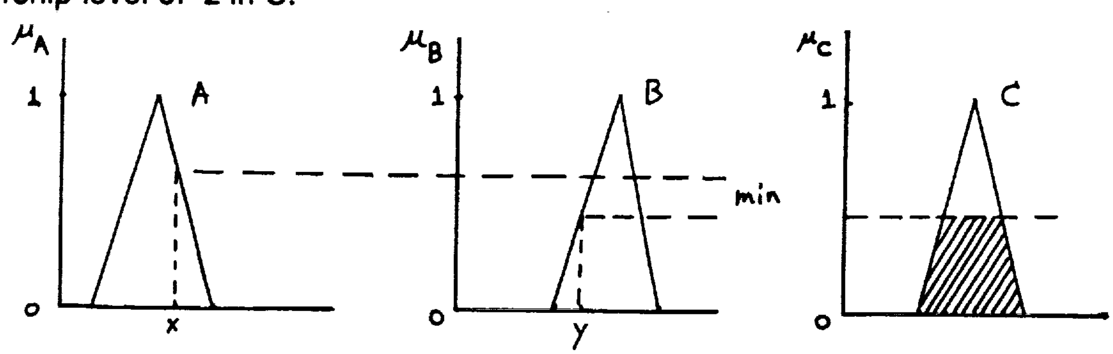


Figure 2. - Graphical representation of approximate reasoning for one rule.

One of the ways used to combine the recommended control actions (in membership form) for several rules is to determine the center of area for all of the actions, thus giving a crisp control action as indicated in Figure 3 below. This is sometimes called conflict resolution or inferencing.

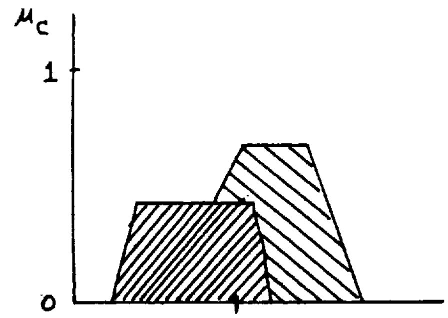


Figure 3. - Conflict resolution (inferencing) for more than one control rule.

Figure 4 below gives the architecture for a typical fuzzy logic control system. Simple systems of this type are usually called fuzzy logic controllers.

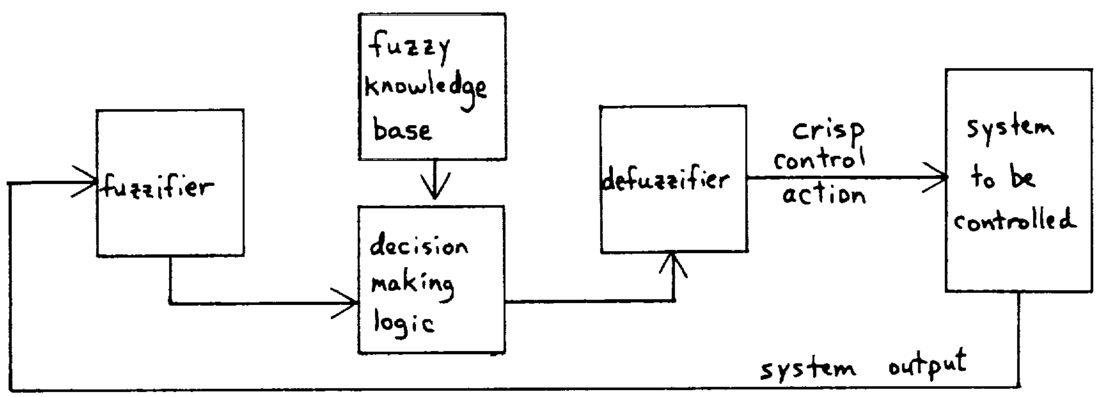


Figure 4. - Basic architecture for fuzzy logic control.

In addition to straightforward programming from scratch and analog fuzzy logic chips, there are now a limited number of software tools to facilitate the development of fuzzy logic expert systems, with a fuzzy logic control system as an example of such. One of the most readily available and most conveniently used tools is the fuzzy logic expert system shell called TILShell from Togai InfraLogic Inc. (5). Technically, TILShell is a graphical editor that creates a database for Togai InfraLogic Inc.'s Fuzzy-C Compiler, which then generates C language code with the data structures and inference engine required to implement the system. This code may then be compiled and linked to other application code. Additional software/hardware hybrid tools are being developed by Apt Instruments Inc. and should be available by the early part of 1991.

Fuzzy logic control systems have been implemented effectively for a variety of applications in Japan. A video teleconference on July 24, 1990, linked the close of the International Conference on Fuzzy Logic and Neural Networks (IIZUKA'90 at the Kyusu Institute of Technology in Japan) with interested parties at the Johnson Space Center, and it was made clear that this is very likely the dawn of an era of vigorous and expanded growth in fuzzy logic applications.

COLLISION AVOIDANCE UNDER CONDITIONS OF UNCERTAINTY

This section represents an original approach to collision avoidance under conditions of uncertainty. The first case to be considered for an avoidance system is a fuzzy obstacle that is not moving. The objective is to have the vehicle maneuver from a source position to a destination position (for sample gathering, etc.) in the shortest amount of time and avoid collision with the obstacle. Figure 5 below gives the scenario for this problem.

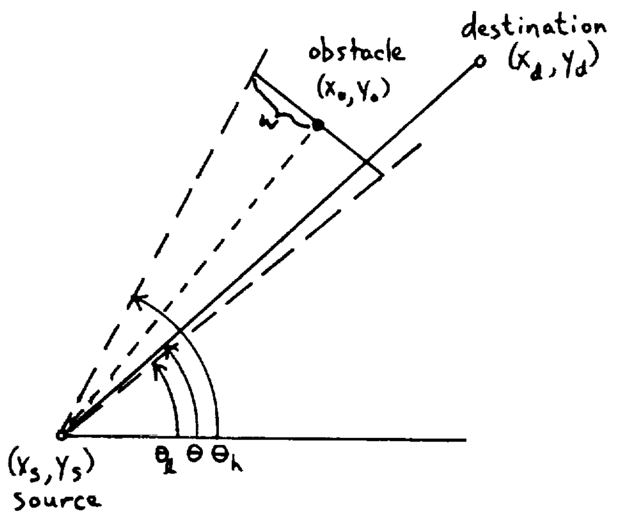


Table of symbols for diagram:

(x_s, y_s) = coordinates of source
(x_d, y_d) = coordinates of destination
(x_o, y_o) = coordinates of obstacle "center"
w = fuzzy "radius"
θ = angle of desired trajectory to destination
θ_l = lower angle for fuzzy avoidance
θ_h = higher angle for fuzzy avoidance
d = distance covered in 1 time unit

Figure 5. - Avoidance of a static fuzzy obstacle.

The symbol "w" is used to denote the fuzzy "radius" of an obstacle; that is, the radius about the obstacle "center" for which the membership in "is an obstacle" is above some threshhold (e.g., 0.1). Both the fuzzy "radius" and the fuzzy "center" for the obstacle are assumed to be available from current sensor data. This represents a significant relaxation of requirements for object identification and object extent determination. In general, the objective is to take fuzzy information regarding a potential obstacle, determine whether it is an obstacle, take evasive action if necessary, then continue until the destination is reached. The algorithm for collision avoidance in the case of this static fuzzy obstacle is:

1. Receive (x_o, y_o, w) from sensor processing.
2. Check for obstruction:

   if $\theta_l < \theta < \theta_h$

   then change θ to θ_l if $|\theta - \theta_l| < |\theta - \theta_h|$,

   otherwise change θ to θ_h

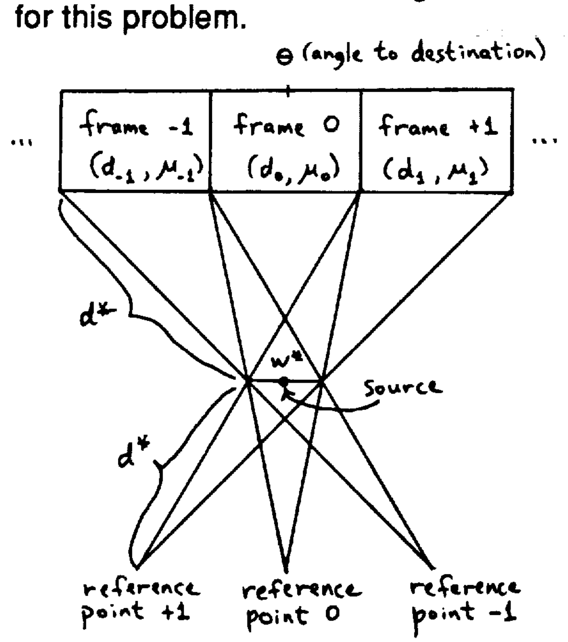
   else no change in θ.
3. Advance vehicle one time unit :

   new x_s = old $x_s + d \cos\theta$

   new y_s = old $y_s + d \sin\theta$.
4. Repeat steps 1., 2., 3. until destination is reached.

Once the vehicle is sufficiently close to the destination, it may be necessary to enter a new control environment for positioning, etc., so step 4. above may actually cease when some critical minimum distance to the destination is reached.

The second case to be considered for an avoidance system is an amorphous fuzzy obstacle system that is not moving. The term "amorphous" in this setting means that there may be more than one obstacle or perhaps even a mass of obstacles that have not been separated or may not be separable. Once again this represents a simplification of the requirements for sensor processing while providing a satisfactory environment for collision avoidance. The objective again is to have the vehicle maneuver from a source position to a destination position (for sample gathering, etc.) in the shortest amount of time and avoid collision or damage to the vehicle. Figure 6 below gives the scenario for this problem.



collision envelope

within critical distance = d^*

within critical sector = $[\theta-\Delta\theta, \theta+\Delta\theta]$

within critical height = $[\varepsilon, h]$

w^* = clearance width for safe passage

d_i = closest distance ($\le d^*$) from source to a "significant" feature in frame i

μ_i = membership of frame i in "is an obstacle"

Figure 6. - Avoidance of a static amorphous fuzzy obstacle system.

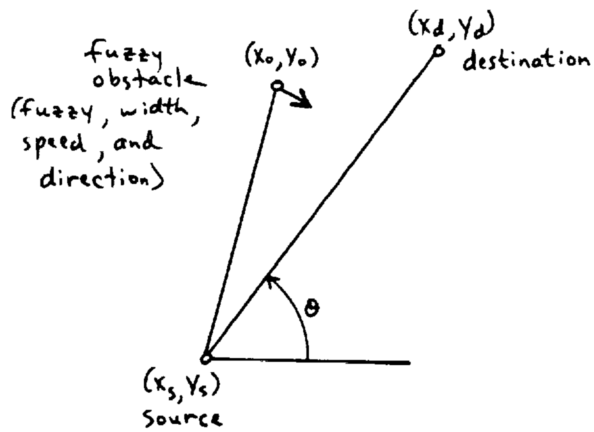
The idea in this avoidance process is to provide a corridor for safe passage of the vehicle. If θ is the angle from the source to the destination, then the angular interval $[\theta-\Delta\theta,\theta+\Delta\theta]$ is broken up into an odd number of angular sectors of width at least $2w^*$. The 0th sector is centered about θ at a distance of d^* from the source on the way to the destination. The first sector to the right of center is labelled 1, then the second sector to the right is labelled 2, and so forth. The first sector to the left of center is labelled -1, then the second sector to the left is labelled -2, and so forth. The sensor data is processed for each sector by scanning the region from the source to the sector boundary as indicated in Figure 6. The reference point is moved back from the source position to a position d^* units behind the source along a line extended from the center of the sector through the center of the vehicle. This ensures that the entire sector path is evaluated for safe passage. Each sector may be considered a frame of reference which is scanned with sensors and assigned a distance and membership in "is an obstacle" which is a generalized form of obstacle detection. In essence, what happens is that the closest distance to a "significant" obstacle is determined; significant would mean above a certain threshhold of membership. The exact form of sensor processing is not addressed here. A simplified version of the algorithm for collision avoidance in the case of a static amorphous obstacle system is:

1. Receive (d_i,μ_i) for each frame from sensor processing.
2. Select path if possible:

   if μ_0 < threshhold
   then proceed in direction θ
   else pick i with smallest |i| and μ_i < threshhold and proceed in direction of center of frame i
   else expand angular interval by $\Delta\theta$ on both sides and check for safe corridor there until one is found.
3. Repeat steps 1., 2. until destination is reached.

Once the distance to the destination is sufficiently close (e.g., < d^*), it is likely that a proximity positioning subsystem would take over for the final approach. One rather natural adjustment to the algorithm would be to scale the membership value of each frame j with a "significant" obstacle by d_i/d_j where i is the frame with the closest "significant" obstacle. Another feature to consider adding is that of speed control (e.g., speed up for safer passage such as the 0th sector with a low membership value and slow down for riskier passage such as a higher numbered sector with a membership value near the threshhold). A simplified form of this obstacle system and algorithm has been implemented by Larry Walters of Lincom Corporation using the TILShell software tool. Further work is needed to make a verifiable autonomous avoidance system for the case of a static amorphous fuzzy obstacle system.

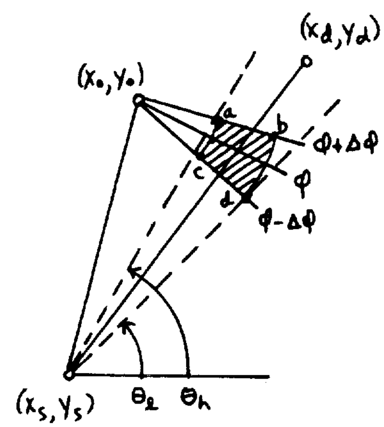
The third case considered is a single fuzzy obstacle with fuzzy movement. Work to date on this case is limited and represents a preliminary framework upon which to build in the future. The situation is that an obstacle with an estimated "center" and fuzzy "width", fuzzy magnitude of speed, and fuzzy direction may cross the path of the vehicle. The problem is to generate the range of positions that the obstacle may take and then determine if the vehicle is likely to be endangered; if so, evasive action must be taken. Figure 7 gives the scenario for the problem. This is a much more complicated problem than that of a single fuzzy obstacle, and it may be that the amorphous static approach may provide the basis of an approach for this situation as well.



fuzzy speed: $[m - \Delta m, m + \Delta m]$
fuzzy direction: $[\phi - \Delta\phi, \phi + \Delta\phi]$
fuzzy radius: w
time increment: Δt

calculate positions a,b,c,d:

$(x_a, y_a) = (x_o + (m - \Delta m)\Delta t \cos(\phi + \Delta\phi),$
$y_o + (m - \Delta m)\Delta t \sin(\phi + \Delta\phi))$
etc.



$d^* = \min\{$ distance from (x_s, y_s) to (x_a, y_a), ..., distance from (x_s, y_s) to $(x_d, y_d)\}$

ThetaSet = {angle from (x_s, y_s) to (x_a, y_a), ..., angle from (x_s, y_s) to (x_d, y_d)}

θ_l = min of ThetaSet

θ_h = max of Theta Set

$\Delta\theta$ = adjustment for fuzzy width (w)

check for $\theta_l - \Delta\theta < \theta < \theta_h + \Delta\theta$, which subject to time issues indicates the presence of an obstacle

Figure 7. - The situation for avoidance of a moving fuzzy obstacle.

ADAPTATION IN FUZZY LOGIC CONTROL SYSTEMS

The issue of developing adaptive fuzzy logic control systems is important and promises to permit the building of control systems that operate under conditions of uncertainty or ambiguity while projecting robustness of operation as conditions change and the likelihood of graceful degradation of operation under loss of control functionality. Adaptive in this setting means developing the system adaptively from training sets of typical data or even permitting "on the fly" modification of the system to reflect changing environments as they occur. The book by Widrow and Stearns (9) provides a framework for adaptive systems in general while the book by Kosko (10) addresses environments closer to those in this report including fuzzy logic systems and adaptive techniques associated with neural networks. Adaptive rule-based control systems that use neuronlike elements in a crisp environment have been addressed by Barto, Sutton, and Anderson (11) and then extended to systems with fuzzy logic rules by Lee and Berenji (12, 13, 14, 15).

The architecture for one type of trainable adaptive fuzzy logic control system is given in Figure 8. It makes use of coupled neurons that learn based on system performance, thus updating the rule-base for enhanced performance. The flow of calculations is given in Figure 9. A good reference for this approach is the paper by Lee (13).

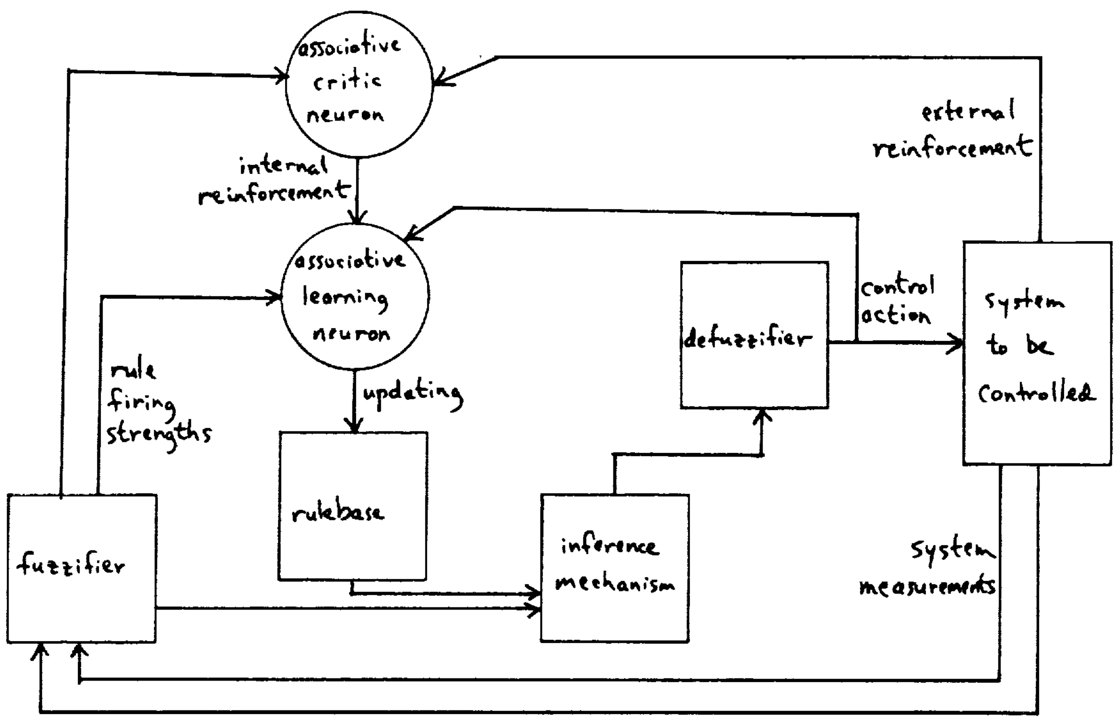


Figure 8. - Architecture for adaptive fuzzy logic control system.

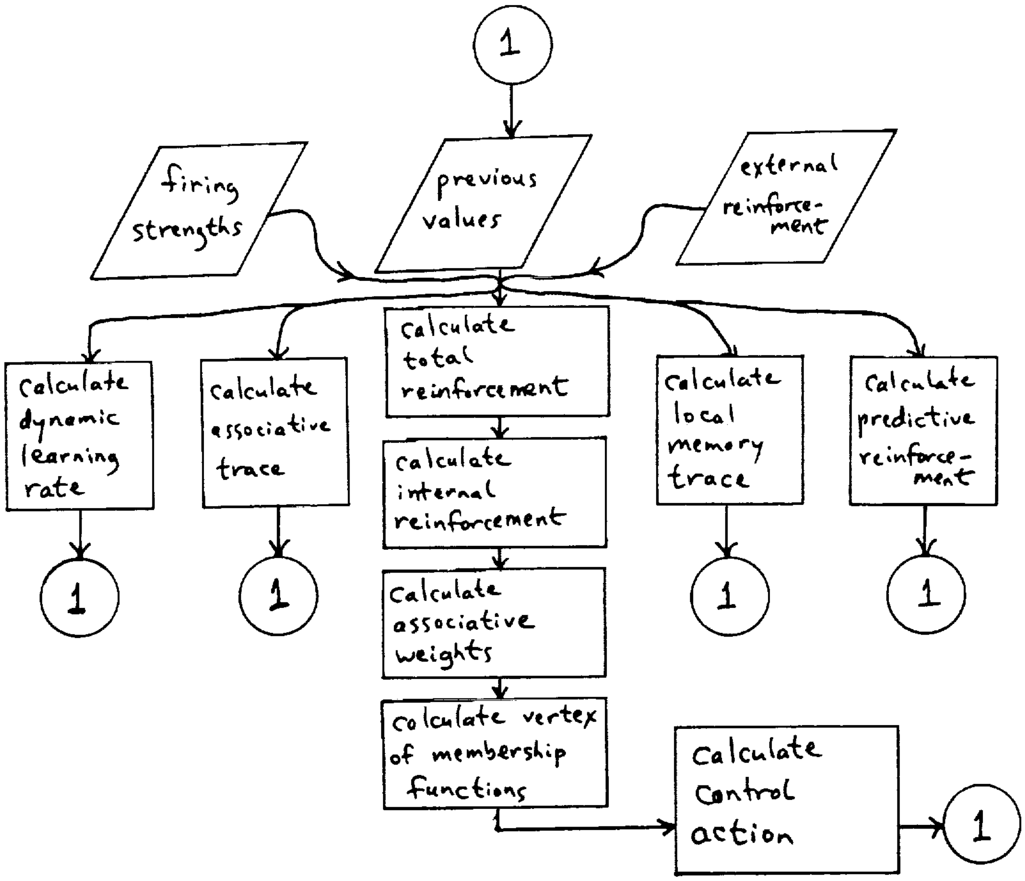


Figure 9. - Flow of calculations for adaptive fuzzy logic control system.

CONCLUSION AND FUTURE DIRECTIONS

Based on the literature identified this summer and additional topics explored from scratch, it is clear that the area of fuzzy logic control is one of significant opportunity for applications. In particular, it is likely to work well under conditions of vagueness or ambiguity and in settings where autonomy is important since fuzzy logic control supports approximate reasoning and model simplification. The specific area of collision avoidance for a Mars rover is a natural setting for the application of these techniques. Further work is needed to explore fuzzy logic collision avoidance in more general situations than those explored in this report. Examples would be fuzzy obstacles with fuzzy movement, collision avoidance in three-space (e.g., involving the space shuttle or other non-surface space vehicles), and developing a working adaptive mechanism for collision avoidance.

REFERENCES

1. Zadeh, L. A.: Fuzzy sets. *Information and Control*, vol. 8, pp. 338-353.

2. Klir, George J.; and Folger, Tina A.: Fuzzy Sets, Uncertainty, and Information. Prentice-Hall, 1988.

3. Lee, Chuen Chien: 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.

4. Lee, Chuen Chien: Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part II. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, 1990, pp. 419-435.

5. TILShell User's Manual, Release 1.1. Togai InfraLogic Inc., Jan., 1990.

6. Lea, Robert N.: Automated Space Vehicle Control for Rendezvous Proximity Operations. *Telematics and Informatics*, vol. 5, no. 3, 1988, pp. 179-185.

7. Lea, Robert N.: Applications of Fuzzy Sets to Data Monitoring and Process Control. *Proceedings of the ISA'88 International Conference and Exhibit*, October, 1988, pp. 1495-1502.

8. Lea, Robert N.; Togai, Masaki; Teichrow, Jon; and Jani, Yashvant: Fuzzy Logic Approach to Combined Translational and Rotational Control of a Spacecraft in Proximity of the Space Station. *Proceedings of the Third International Fuzzy Systems Association Congress*, August, 1989.

9. Widrow, Bernard; and Stearns, Samuel D.: Adaptive Signal Processing. Prentice-Hall, 1985.

10. Kosko, Bart: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall, 1990.

11. Barto, Andrew G; Sutton, Richard S.; and Anderson, Charles W.: Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, no. 5, 1983, pp. 834-846.

12. Lee, Chuen-Chien; and Berenji, Hamid R.: An Intelligent Controller based on Approximate Reasoning and Reinforcement Learning. *Proceedings of the IEEE International Symposium on Intelligent Control 1989*, pp. 200-205.

13. Lee, Chuen-Chien: Intelligent Control Based on Fuzzy Logic and Neural Net Theory. *Proceedings of the 1990 Workshop on Fuzzy Logic and Neural Networks* , NASA/JSC.

14. Lee, Chuen-Chien: A Self-Learning Rule-Based Controller Employing Approximate Reasoning and Neural Network Concepts. *International Journal of Intelligent Systems*, vol. 5, 1990.

15. Lee, Chuen-Chien: Modeling behavioral substrates of associative learning and memory: adaptive neural models. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, 1990.