A REVIEW ON FUZZY LOGIC – ITS APPLICATION AND METHODOLOGY

W.SARADA*

DR.P.V.KUMAR**

ABSTRACT

Fuzzy logic is an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based. Fuzzy logic seems closer to the way our brains work. We aggregate data and form a number of partial truths which we aggregate further into higher truths which in turn, exceeds certain thresholds and cause certain further results such as motor reaction. A similar kind of process is used in artificial computer neural network and expert systems. It may help to see fuzzy logic as the way reasoning really works and binary or Boolean logic is simply a special case of it. Fuzzy logic is a rule-based decision process that seeks to solve problems where the system is difficult to model and where ambiguity or vagueness is abundant between two extremes. The fuzzy rules relate the input variables to the output variables via the subsets. Given a set of fuzzy rules, the system can compensate quickly and efficiently. Though the Western world did not initially accept fuzzy logic and fuzzy ideas, today fuzzy logic is applied in many systems. In this paper, the steps of how to create a fuzzy system are described as well as the description of how the fuzzy system works.

Key words—Fuzzy logic, rules, membership functions, linguistic variables

* Assistant Professor, RBVRR Women's College, Narayanaguda, Hyderabad

** Professor, University College of Computer Science and Engineering, Osmania University, Hyderabad.

Introduction

The first historical connection to fuzzy logic can be seen in the thinking of Buddha, the founder of Buddhism around 500 B.C. He believed that the world was filled with contradictions and everything contained some of its opposite. Contrary to Buddha's thinking, the Greek philosopher Aristotle created binary logic through the Law of the Excluded Middle. Much of the Western world accepted his philosophy and it became the base of scientific thought. Still today, if something is proven to be logically true, it is considered scientifically correct.

Prior to Zadeh, a man named Max Black published a paper in 1937 called "Vagueness: An exercise in Logical Analysis" [1]. The idea that Black missed was the correlation between vagueness and functioning systems. Zadeh, on the other hand, saw this connection and began to develop his "fuzzy" ideas and fuzzy sets. Professor Zadeh reasoned that people do not require precise, numerical information input, and yet they are capable of highly adaptive control. If feedback controllers could be programmed to accept noisy, imprecise input, they would be much more effective and perhaps easier to implement. Unfortunately, U.S. manufacturers have not been so quick to embrace this technology while the Europeans and Japanese have been aggressively building real products around it.

The concept of Fuzzy Logic (FL) [2] was conceived by Lofty Zadeh, a professor at the University of California at Berkley, and presented not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership or non-membership. This approach to set theory was not applied to control systems until the 70's due to insufficient small-computer capability prior to that time. Fuzzy logic began in 1965 with a paper called "Fuzzy Sets" by a man named Lofty Zadeh. Zadeh is an Iranian immigrant and professor from UC Berkeley's electrical engineering and computer science department.

Meaning of fuzzy logic

FL is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems. It can be implemented in hardware, software, or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL's approach to control problems mimics how a person would make decisions, only much faster.

About Fuzzy Systems

Fuzzy systems are knowledge-based or rule-based systems. The heart of a fuzzy system is a knowledge base consisting of the so-called fuzzy IF-THEN rules. A fuzzy IF-THEN rule is an IF-THEN statement in which some words are characterized by continuous membership functions. For example, the following is a fuzzy IF-THEN rule:

IF the speed of a car is high, THEN apply less force to the accelerator (1.1)where the words "high" and "less" are characterized by the membership functions shown in Figs.l.1 and 1.2, respectively.' A fuzzy system is constructed from a collection of fuzzy IF-THEN rules. Let us consider one example.

Example 1.1: Suppose we want to design a controller to automatically control the speed of a car. Conceptually, there are two approaches to designing such a controller: the first approach is to use conventional control theory, for example, designing a PID controller; the second approach is to emulate a human driver that is, converting the rules used by human drivers into an automatic controller. We now consider the second approach. Roughly speaking, human drivers use the following three types of rules to drive a car in normal situations:

IF speed is low, THEN apply more force to the accelerator
IF speed is medium, THEN apply normal force to the accelerator
IF speed is high, THEN apply less force to the accelerator

where the words "low," "more," "medium," "normal," "high," and "less" are characterized by membership functions similar to those in Fig.l.1. Of course, more rules are needed in real situations. We can construct a fuzzy system based on these

IJMIE Volume 6, Issue 6 ISSN: 2249-0558

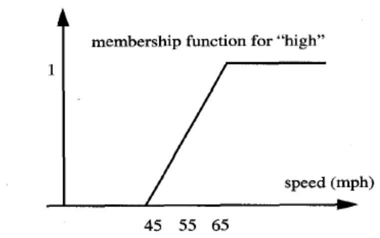

Figure 1.1. Membership function for "high," where the horizontal axis represents the speed of the car and the vertical axis represents the membership value for "high."

Figure 1.1.

Membership function for "high," where the horizontal axis represents the speed of the car and the vertical axis represents the membership value for "high." rules. Because the fuzzy system is used as a controller, it also is called a fuzzy controller.

Difference between FL and conventional controls

FL incorporates a simple, rule-based IF X AND Y THEN Z approach to a solving control problem rather than attempting to model a system mathematically. The FL model is empirically-based, relying on an operator's experience rather than their technical understanding of the system. For example, rather than dealing with temperature control in terms such as "SP =500F", "T <1000F", or "210C <TEMP <220C", terms like "IF (process is too cool) AND (process is getting colder) THEN (add heat to the process)" or "IF (process is too hot) AND (process is heating rapidly) THEN (cool the process quickly)" are used. These terms are imprecise and yet very descriptive of what must actually happen. Consider what you do in the shower if the temperature is too cold: you will make the water comfortable very quickly with little trouble. FL is capable of mimicking this type of behaviour but at very high rate.

Working of FL

FL requires some numerical parameters in order to operate such as what is considered significant error and significant rate-of-change-of-error, but exact values of these numbers are usually not critical unless very responsive performance is required in which case empirical tuning would determine them. For example, a simple temperature control system could use a single temperature feedback sensor whose data is subtracted from the command signal to compute "error" and then time-differentiated to yield the error slope or rate-of-change-of-error, hereafter called "error-dot". Error might have units of degs F and a small error considered to be 2F while a large error is 5F. The "error-dot" might then have units of degs/min with a small error-dot being 5F/min and a large one being 15F/min. These values don't have to be symmetrical and can be "tweaked" once the system is operating in order to optimize performance. Generally, FL is so forgiving that the system will probably work the first time without any tweaking.

FL was conceived as a better method for sorting and handling data but has proven to be an excellent choice for many control system applications since it mimics human control logic. It can be built into anything from small, hand-held products to large computerized process control systems. It uses an imprecise but very descriptive language to deal with input data more like a human operator. It is very robust and forgiving of operator and data input and often works when first implemented with little or no tuning.

Features of FL

There are many benefits to using fuzzy logic. Fuzzy logic is conceptually easy to understand and has a natural approach. Fuzzy logic is flexible and can be easily added to and adjusted. It is very tolerant of imprecise data and can model complex nonlinear functions with little complexity. It can also be mixed with conventional control techniques. There are three major components of a fuzzy system: fuzzy sets, fuzzy rules, and fuzzy numbers.

FL offers several unique features [3] that make it particularly a good choice for many control problems.

1) It is inherently robust since it does not require precise, noise-free inputs and can be programmed to fail safely if a feedback sensor quits or is destroyed. The output control is a smooth control function despite a wide range of input variations.

2) Since the FL controller processes user-defined rules governing the target control system, it can be modified and tweaked easily to improve or drastically alter system performance.

3) FL is not limited to a few feedback inputs and one or two control outputs, nor is it necessary to measure or compute rate-of-change parameters in order for it to be implemented. Any sensor data that provides some indication of a system's actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low.

4) Because of the rule-based operation, any reasonable number of inputs can be processed (1-8 or more) and numerous outputs (1-4 or more) generated, although defining the rule base

quickly becomes complex if too many inputs and outputs are chosen for a single implementation since rules defining their interrelations must also be defined. It would be better to break the control system into smaller chunks and use several smaller FL controllers distributed on the system, each with more limited responsibilities.

5) FL can control nonlinear systems that would be difficult or impossible to model mathematically. This opens doors for control systems that would normally be deemed unfeasible for automation.

Steps for creating Fuzzy System

1) Define the control objectives and criteria: What am I trying to control? What do I have to do to control the system? What kind of response do I need? What are the possible (probable) system failure modes?

2) Determine the input and output relationships and choose a minimum number of variables for input to the FL engine (typically error and rate-of-change-of-error).

3) Using the rule-based structure of FL [4], break the control problems down into a series of IF X AND Y THEN Z rules that define the desired system output response for given system input conditions. The number and complexity of rules depends on the number of input parameters that are to be processed and the number fuzzy variables associated with each parameter. If possible, use at least one variable and its time derivative. Although it is possible to use a single, instantaneous error parameter without knowing its rate of change, this cripples the system's ability to minimize overshoot for a step inputs.

4) Create FL membership functions that define the meaning (values) of Input/Output terms used in the rules.

5) Create the necessary pre- and post-processing FL routines if implementing in S/W, otherwise program the rules into the FL H/W engine.

6) Test the system, evaluate the results, tune the rules and membership functions, and retest until satisfactory results are obtained.

Linguistic Variables

In 1973, Professor Lofty Zadeh proposed the concept of linguistic or "fuzzy" variables. Think of them as linguistic objects or words, rather than numbers. The sensor input is a noun, e.g. "temperature", "displacement", "velocity", "flow", "pressure", etc. Since error is just the difference, it can be thought of the same way. The fuzzy variables themselves are adjectives that modify the variable (e.g. "large positive" error, "small positive" error, “zero" error, "small negative" error, and "large negative" error). As a minimum, one could simply have "positive", "zero", and "negative" variables for each of the parameters.

The Rule Matrix

The fuzzy parameters of error command-feedback) and error-dot (rate-of-change-of-error) were modified by the adjectives "negative", "zero", and "positive". To picture this, imagine the simplest practical implementation, a 3-by-3 matrix. [5]The columns represent "negative error", "zero error", and "positive error" inputs from left to right. The rows represent "negative", "zero", and "positive" "error-dot" input from top to bottom. This planar construct is called a rule matrix. It has two input conditions, "error" and "error-dot", and one output response conclusion (at the intersection of each row and column).The maximum number of possible rules is simply the product of the number of rows and columns, but definition of all of these rules may not be necessary since some input conditions may never occur in practical operation. The primary objective of this construct is to map out the universe of possible inputs while keeping the system sufficiently under control.

Membership Function

The membership function [6] is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. Once the functions are inferred, scaled, and combined, they are defuzzified into a crisp output which drives the system.

Conclusion

Fuzzy Logic provides a completely different, unorthodox way to approach a control problem. This method focuses on what the system should do rather than trying to understand how it works. One can concentrate on solving the problem rather than trying to model the system mathematically, if that is even possible. Once understood, this technology is not difficult to apply and the results are usually quite surprising and pleasing. All in all, fuzzy logic is another way to look at the world. It is another way of thinking and challenges our current scientific thought. It presents an easy and practical way to solve many problems.

References

[1] Kosko, Bart. 1993. "Fuzzy Thinking: The New Science of Fuzzy Logic." Hyperion. New York.

[2]"Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh", ed. R.R. Yager et al. (John Wiley, New York).

[3] "Fuzzy Sets" by Ivars Peterson (Science News, Vol. 144, July 24, 1993, pp. 55).

[4] "Fuzzy Logic Flowers in Japan" by D.G. Schartz & G.J. Klir (IEEE Spectrum, July 1992, pp. 32-35).

[5] "Fuzzy Logic Makes Guesswork of Computer Control" by Gail M. Robinson (Design News, Vol. 47, Nov. 28, 1991, pp. 21).

[6] "Neural Networks and Fuzzy Systems--A Dynamic Systems Approach to Machine Intelligence" by B. Kosko (Prentice-Hall, Englewood Cliffs, N.J., 1992).

[7] "Estimation of Fuzzy Membership from Histograms, Information Sciences" by B.B. Devi et al (Vol. 35, 1985, pp. 43-59).

[8] "The Coming Age of Fuzzy Logic" Proceedings of the 1989 IFSA Congress, J.C. Bezdek, ed. (University of Washington, Seattle, WA 1989).

W.Sarada is a research scholar at Rayalaseema University, Kurnool, Andhra Pradesh, India. She is working as an Assistant Professor in the Department of Computer Science at RBVRR Women's College. Her interests include Data Mining, Digital Image Proc**essing, Computer Networks and Software Engineering.**