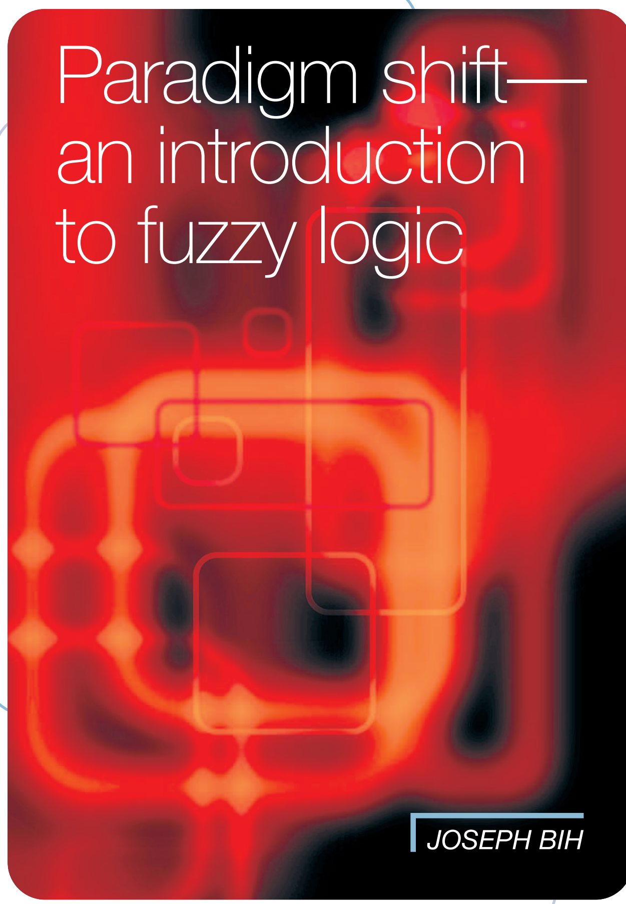
Paradigm shift—an introduction to fuzzy logic

JOSEPH BIH



FUZZY LOGIC SUGGESTS inaccuracy and imprecision. Webster's dictionary defines the word fuzzy as "not clear, distinct, or precise; blurred." In a broad sense, fuzzy logic refers to fuzzy sets, which are sets with blurred boundaries, and, in a narrow sense, fuzzy logic is a logical system that aims to formalize approximate reasoning.

Fuzzy logic is an approach to computer science that mimics the way a human brain thinks and solves problems. The idea of fuzzy logic is to approximate human decision making using natural language terms instead of quantitative terms. It is formally defined as

> a form of knowledge representation suitable for notions that cannot be defined precisely, but which depend upon their contexts. It enables computerized devices to reason more like humans.

Fuzzy-logic technology has created a paradigm shift evident through many scientific and industrial applications.

So, did it begin with fuzzy?

Interestingly, fuzzy science started in the questioning minds of philosophers. Confused and inquisitive, from Buddha, to Aristotle, to Plato, these ancient philosophers were constantly searching for a "rule of law" beyond true or false.

Observing that computer logic was incapable of representing subjective ideas such as "very hot" or "very cold," in 1965, Zadeh published his ideas of fuzzy set theory that made it possible for computers to distinguish between different shades of gray, similar to the process of human reasoning. He described fuzzy mathematics, devising precise rules for combining vague expressions such as "somewhat fast," "very hot," and "usually wrong," which are particularly useful for controlling robots, machine tools, and various electronic systems.

Prof. Terano was inspired by Zadeh's work and introduced fuzzy logic to the Japanese scientific community in 1972. As Bart Kosko, a Zadeh protégé and a professor of electrical engineering at the University of Southern California said: "Fuzziness begins where Western logic ends." Japan embraced the technology and adapted it to physical control systems.

In 1980, F.L. Smidth & Co. of Copenhagen began marketing the first commercial fuzzy expert system: a computer program that controlled the fuel-intake rate and gas flow of a rotating kiln used to make cement. From Hitachi's subway system, to Nissan's fuzzy auto transmission and antiskid braking systems, to Yamaichi Security's fuzzy stock-market investment program for signaling shifts in market sentiment, to Matsushita's fuzzy automobile-traffic controller, Japan has been taking the lead in fuzzy-logic research and development and transforming the technology into industrial applications. Some say that the Japanese cultural environment plays a significant role by embracing fuzzy logic. For these companies, fuzzy logic is a paradigm to introduce human subjectivity into objective science and a method to model and use human knowledge and senses as they are, without complicating abstraction.

Since NASA pioneered fuzzy-controller experiments that could help astronauts pilot the space shuttle in earth orbit, there is growing interest at such aerospace firms as Rockwell and Boeing. "The only barrier remaining to wider use of fuzzy logic," says Kosko, "is the philosophical resistance of the West."

Why fuzzy logic?

Fuzzy logic comes in when conventional logic fails. Fuzzy logic can deal with virtually any proposition expressed in natural language. For example, the proposition, "It is very unlikely that the price of gold will significantly increase in the near future," which is beyond the classical first-order predicate logic, is perfectly manageable by fuzzy logic. The meanings of propositions like this can be determined, for example, by a method known as test-score semantics.

An important concept in fuzzy logic lies in the concept of linguistic variables: variables whose values are words or sentences in natural language. In general, any relation between two linguistic variables can be expressed in terms of fuzzy if–then rules. Such rules, when properly elicited from experts, form the knowledge base of fuzzy controllers or fuzzy expert systems.

Once the meanings of relevant propositions are determined, fuzzy logic provides us with approximate reasoning in linguistic terms that are available in natural language. The approximate reasoning may involve deductive inferences as well as interpolative inferences, as shown in the following example:

> Old coins are usually rare collectibles.
> Rare collectibles are expensive.
> Old coins are usually expensive.

Fuzzy logic versus neural networks

The idea of fuzzy logic is to approximate human decision-making using natural-language terms instead of quantitative terms. Fuzzy logic is similar to neural networks, and one can create behavioral systems with both methodologies. A good example is the use of fuzzy logic for automatic control: a set of rules or a table is constructed that specifies how an effect is to be achieved, provided input and the current system state. Using fuzzy arithmetic, one uses a model and makes a subset of the system components fuzzy so that fuzzy arithmetic must be used when executing the model. In a broad sense, fuzzy logic refers to fuzzy sets, which are sets with blurred boundaries, and, in a narrow sense, fuzzy logic is a logical system that aims to formalize approximate reasoning.

Antecedent Block | Consequent Block

1. IF Cmd-Temp=N AND d(Cmd-Temp)/dt=N THEN Output=C
2. IF Cmd-Temp=Z AND d(Cmd-Temp)/dt=N THEN Output=H
3. IF Cmd-Temp=P AND d(Cmd-Temp)/dt=N THEN Output=H
4. IF Cmd-Temp=N AND d(Cmd-Temp)/dt=Z THEN Output=C
5. IF Cmd-Temp=Z AND d(Cmd-Temp)/dt=Z THEN Output=NC
6. IF Cmd-Temp=P AND d(Cmd-Temp)/dt=Z THEN Output=H
7. IF Cmd-Temp=N AND d(Cmd-Temp)/dt=P THEN Output=C
8. IF Cmd-Temp=Z AND d(Cmd-Temp)/dt=P THEN Output=C
9. IF Cmd-Temp=P AND d(Cmd-Temp)/dt=P THEN Output=H

Error -(Cmd-Temp)

Error-dot - (d(cmd-Temp)/dt)	N	Z	P
N	1 C	2 H	3 H
Z	4 C	5 NC	6 H
P	7 C	8 C	9 H

Fig. 1 Rule structure and rule matrix

For example, a temperature control system has three settings: cold, moderate, and hot (see Fig. 1). The first step is to develop a matrix. Because there are three conditions, the matrix will be 3×3. There will also be quite a few variables. These include *N* for negative, *P* for positive, and *Z* for zero. Each represents the possible input error and its derivative, the direction of temperature change; in other words: it is hot, getting hotter, or cold, getting older, or moderate. The variables inside the matrix represent the responses to changing conditions. *C* represents a "cool" response, *H* represents a "heat" response, and *NC* represents a "no change" response. Together, these variables provide nine rules to conditions and their responses depending on varying situations. This is how fuzzy logic "makes decisions." If the temperature is hot (*N*) and getting hotter (*N*), then the response should be to turn the cooling feature on the temperature control system (*N*). This can be implemented very easily by computer hardware, software, or a combination of the two. While this is a very simple example, more practical applications would make up very large matrices and a more complex set of rules. It is key to note, though, that conditional rules can easily be stated linguistically and that conditional statements can use the AND, OR, or NOT operators.

Neural networks, however, are a different paradigm for computing. Neural networks process information in a similar way that the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example and cannot be programmed to perform a specific task. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques.

Neural networks are a form of multiprocessor computer system, with
- simple processing elements
- a high degree of interconnection
- simple scalar messages
- adaptive interaction between elements.

A biological neuron may have as many as 10,000 different inputs, and may send its output (the presence or absence of a short-duration spike) to many other neurons. Neurons are wired up in a three-dimensional (3-D) pattern. Real brains, however, are orders of magnitude more complex than any artificial neural network so far considered.

A simple, signle-unit adaptive network is shown in Fig. 2. The network has two inputs and one output. All are binary. The output is 1 if $W_0 * I_0 + W_1 * I_1 + W_b > 0$, and 0 if $W_0 * I_0 + W_1 * I_1 + W_b \leq 0$.

We want it to learn simple OR: output a 1 if either I_0 or I_1 is 1.

Conventional computers use an algorithmic approach; i.e., the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known, the computer cannot solve the problem. This restricts the problem-solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do. This is where neural networks come in. Neural-network systems help when formulating an algorithmic solution is extremely difficult, lots of examples of the behavior that are required, or there is a need to pick out the structure from existing data.

Fuzzy logic in control systems—case studies

Fuzzy logic in design methodology and for nonlinear control systems

Fuzzy logic is a paradigm for an alternative design methodology that can be applied in developing both linear and nonlinear systems for embedded control. Using fuzzy logic, designers can realize lower development costs, superior features, and better end-product performance. Furthermore, products can be brought to market faster and more cost effectively.

Simpler and faster design methodology

To appreciate why a fuzzy-based design methodology is very attractive in embedded control applications, let us examine a typical design flow. Figure 3 illustrates a sequence of design steps required to develop a controller using a conventional and a fuzzy approach.

Using the conventional approach, the first step is to understand the physical system and its control requirements. Based on this understanding, the second step is to develop a model that includes the plant, sensors, and actuators. The

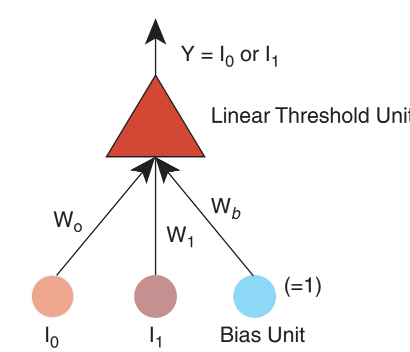


Fig. 2 A simple, single-unit adaptive network

third step is to use linear-control theory in order to determine a simplified version of the controller, such as the parameters of a proportional-integral-derivative (PID) controller. The fourth step is to develop an algorithm for the simplified controller. The last step is to simulate the design including the effects of nonlinearity, noise, and parameter variations. If the performance is not satisfactory, we need to modify our system modeling, re-design the controller, rewrite the algorithm, and retry.

Fuzzy-logic approach reduces the design process to three steps, starting with understanding and characterizing the system behavior through knowledge and experience. The second step is to directly design the control algorithm using fuzzy rules that describe the principles of the controller's regulation in terms of the relationship between its inputs and outputs. The last step is to simulate and debug the design. The fact that one only needs to modify some fuzzy rules and retry the process satisfies the performance requirements.

Though the two design methodologies are similar, fuzzy-based methodology substantially simplifies the design loop, resulting in significantly reduced development time, simpler design, and faster time to market.

Fuzzy-logic design methodology simplifies the steps, especially during the debugging and tuning cycle, in which the system can be changed by simply modifying rules rather than redesigning the controller. The fuzzy rule-based feature focuses more on the application instead of programming, therefore substantially reducing the overall development cycle.

Commercial applications in embedded control require a significant development effort, a majority of which is spent on the software portion of the project. Due to its simplicity, the description of a fuzzy controller is not only is transportable across design teams, but also provides a superior medium to preserve, maintain, and upgrade intellectual property.

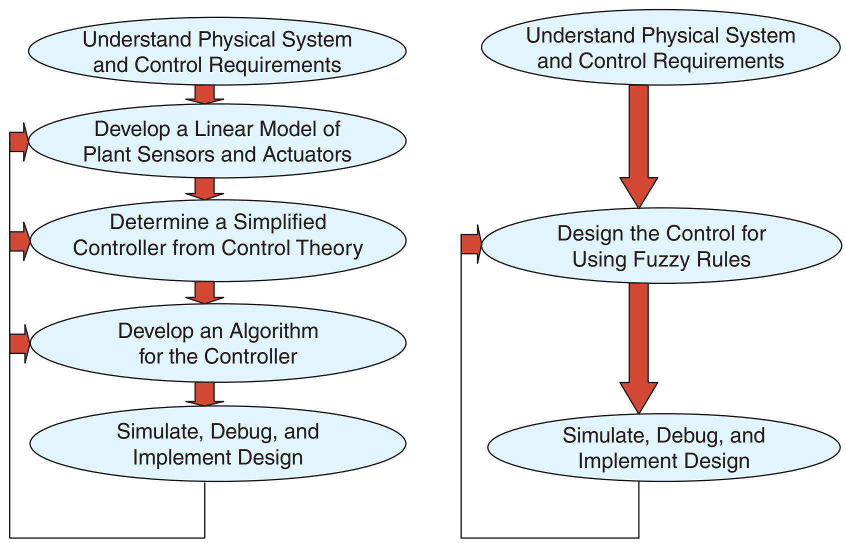


Fig. 3 Conventional and fuzzy design

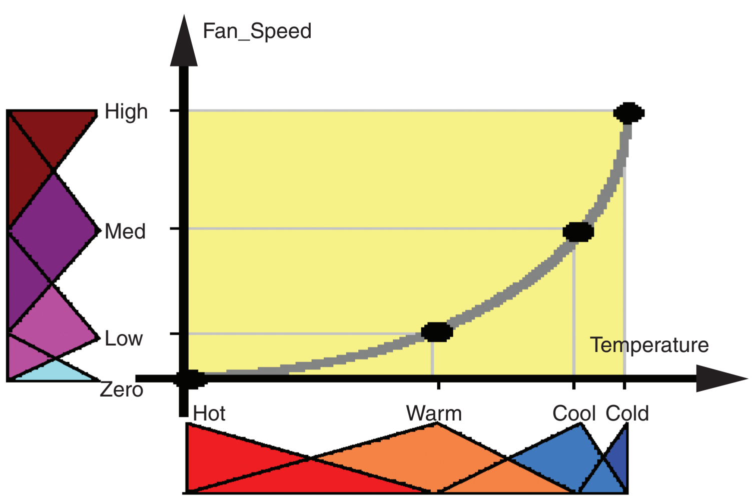


Fig. 4 Rules and membership function approximating a nonlinear function

A better alternative solution to nonlinear control

Most real-life physical systems are actually nonlinear systems. Conventional design approaches use different approximation methods to handle nonlinearity: linear, piecewise linear, and lookup table approximations to trade off factors of complexity, cost, and system performance.

Fuzzy logic rules and membership functions approximate any continuous function to a degree of precision used as in Fig. 4, which shows an approximate desired control curve for temperature controller using four rules (or points). More rules can be added to increase the accuracy of the approximation, which yields improved control performance. Rules are much simpler to implement and much easier to debug and tune than piecewise linear or lookup table techniques. The desired control curve for the temperature controller can be approximated using four points (or four rules) as in the following.

IF temperature IS *cold* **THEN** force IS *high*.

IF temperature IS *cool* **THEN** force IS *medium*.

IF temperature IS *warm* **THEN** force IS *low*.

IF temperature IS *hot* **THEN** force IS *zero*

The fuzzy arithmetic interpolates the shape of the nonlinear function. The combined memory required for the labels and fuzzy inference is substantially less than a lookup table, especially for multiple input systems. As a result, processing speed can be improved as well.

Most control applications have multiple inputs and require modeling and tuning of a large number of parameters which makes implementation time consuming. Fuzzy rules can help simplify implementation by combining multiple inputs into single if-then statements while still handling nonlinearity as the following shows:

IF temperature IS cold AND humidity IS high THEN fan_spd IS high.

IF temperature IS cool AND humidity IS high THEN fan_spd IS medium.

IF temperature IS warm AND humidity IS high THEN fan_spd IS low.

IF temperature IS hot AND humidity IS high THEN fan_spd IS zero.

IF temperature IS cold AND humidity IS med THEN fan_spd IS medium.

IF temperature IS cool AND humidity IS med THEN fan_spd IS low.

IF temperature IS warm AND humidity IS med THEN fan_spd IS zero.

IF temperature IS hot AND humidity IS med THEN fan_spd IS zero.

IF temperature IS cold AND humidity IS low THEN fan_spd IS medium.

IF temperature IS cool AND humidity IS low THEN fan_spd IS low.

IF temperature IS warm AND humidity IS low THEN fan_spd IS zero.

IF temperature IS hot AND humidity IS low THEN fan_spd IS zero.

Fuzzy logic is used to describe the output as a function of two or more inputs linked with operators such as AND. It requires significantly less entries than a lookup table depending upon the number of labels for each input variable. Rules are much easier to develop and simpler to debug and tune than a lookup table, as in Fig. 5.

The lookup table for the two-input temperature controller requires 64 Kb of memory, while the fuzzy approach is accomplished with less than 0.5 Kb of memory for labels and the object code combined. This difference in memory savings implies a cheaper hardware implementation. Conventional techniques in most real life applications would require complex mathematical analysis and modeling, floating point algorithms, and complex branching. This typically yields a substantial size of object code, which requires a high end DSP chip to run. The fuzzy-logic approach uses a simple, rule-based approach that offers significant cost savings, both in memory and processor class.

Fuzzy antilock brake system solution

Many electronic control systems in the automotive industry, such as automatic transmissions, engine control, and antilock brake systems (ABS) realize superior characteristics through the use of fuzzy-logic-based control rather than traditional control algorithms.

ABS is implemented in automobiles to ensure optimal vehicle control and minimal stopping distances during hard or emergency braking. The number of cars equipped with ABS is on the rise. ABS is now accepted as an essential contribution to vehicle safety. The methods of control utilized by ABS are responsible for system performance.

Fuzzy ABS

ABS systems were introduced to the commercial vehicle market in the early 1970s to improve vehicle braking irrespective of road and weather conditions. Electronic control units (ECUs), wheel speed sensors, and brake modulators are major components of an ABS module. Wheel-speed sensors transmit pulses to the ECU with a frequency proportional to wheel speed. The ECU then processes this information and regulates the brake accordingly. The ECU and control algorithm are at least partially responsible for how well the ABS system performs.

Since ABS systems are nonlinear and dynamic in nature, they are prime candidates for fuzzy-logic control. For most driving surfaces, as vehicle braking force is applied to the wheel system, the longitudinal relationship of friction between vehicle and driving surface rapidly increases. Increasing brake force in a decreasing frictional environment often results in full wheel lockup. It has been both mathematically and empirically proven a sliding wheel produces less friction a moving wheel. Inputs to the Intel Fuzzy ABS are derived from wheel speed. Acceleration and slip for each wheel may be calculated by combining the signals from each wheel. These signals are then processed in the Intel Fuzzy ABS system to achieve the desired control. Unlike earlier 8-b microcontroller architectures with limited math capability, Intel Fuzzy ABS example utilizes a high performance, low cost, 16-b 8XC196Kx architecture to take advantage of improved math execution timing.

modelBUILDER

Unlike a conventional ABS system, performance of the Intel Fuzzy ABS System can be optimized with less detailed knowledge of the internal system dynamics. This is achieved by the process used to refine the rule base and in the initial development of the system using Inform Software Corporation *fuzzy*TECH(R) 3.0 MCU-96 software tuned for the Intel Architecture with optimized code output and the associated Real Time Cross Debugger. Rapid development is attained by the software tool set combined with a linguistic approach to control implemented in the Intel Fuzzy ABS solution. A cornerstone of this rapid development is the Intel fuzzy logic modeling software kit called *fuzzy*BUILDER.

*fuzzy*TECH MCU-96 contains a fully graphical computer-aided software engineering (CASE) tool to support all design steps for fuzzy system engineering and a simulation and optimization tool for fuzzy systems. It is specifically optimized for the MCS 96 architecture. This tool displays system performance and can be interfaced to conventional simulators to obtain performance data; a code generator which generates complete C-code for the fuzzy system. The C-code calls optimized assembly routines on the target controller for fast performance.

Conventional ABS control algorithms, however, must account for nonlinearity in brake torque because of the temperature variation and dynamics of

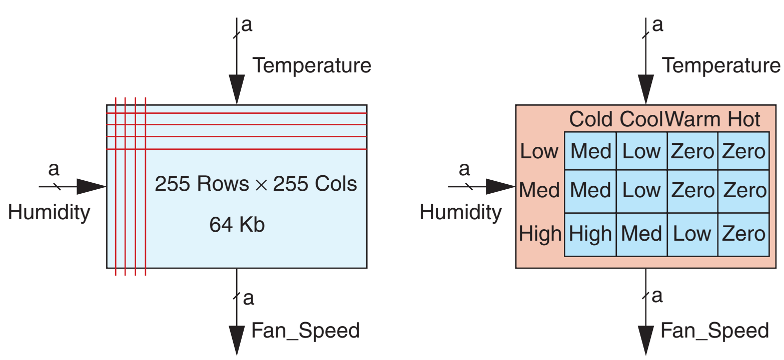


Fig. 5 Lookup table versus rules and membership functions

brake-fluid viscosity. Also, external disturbances, such as changes in frictional coefficient and road surface, must be accounted for, not to mention the influences of tire wear and system components aging. These influential factors increase system complexity, in turn effecting mathematical models used to describe systems. As the model becomes increasingly complex, equations required to control ABS also become increasingly complicated. Due to the highly dynamic nature of ABS, many assumptions and initial conditions are used to make control achievable. Once control is achieved, the system is implemented in-vehicle and tested. The system is then modified to attain the desired control status.

However, due to the nature of fuzzy logic, influential dynamic factors are accounted for in a rule-based description of ABS. This type of "intelligent" control allows for faster development of system code.

The inputs to the Intel Fuzzy ABS are represented in Fig. 6 and consist of:

1) *The brake:* This block represents the brake pedal deflection/assertion. This information is acquired in a digital or analog format.

2) *The 4 WD:* This indicates if the vehicle is in the four-wheel-drive mode.

3) *The ignition:* This input registers if the ignition key is in place and if the engine is running or not.

4) *Feedback:* This block represents the set of inputs concerning the state of the ABS system.

5) *Wheel speed:* In a typical application, this will represent a set of four input signals that convey the information concerning the speed of each wheel. This information is used to derive all necessary information for the control algorithm.

fuzzyTECH 3.0

The proposed system shown in Fig. 6 has two types of outputs: the pulsewidth modulated (PWM) signals to control ABS braking, and an error lamp signal to indicate a malfunction if one exists.

Significant applications of fuzzy logic technology

Fuzzy logic has also seen its tremendous power used in business, finance, and management. FuzzyTECH for business is one of the new products with standard software already available for business, finance, and management in accurate forecasting and analysis. The models' applied fuzzy logic generates more accurate forecasting in the stock market. Pricing models for new products and fuzzy, zero-based budgeting are typical examples of fuzzy-logic technology applied for decision-making processes. One of the most important areas is the growing interest in fuzzy database models. The extensive and surging research interest in fuzzy relational database technology is now and will have an enormous impact on database technology. Fuzzy queries, a further approach from traditional database queries yet combined with the database technology, will provide very powerful new technology in database management systems.

Fuzzy-logic defense applications include Aerospace-Missile A framework, which can be used as a guideline to design a Tagagi-Sugeno (T-S) fuzzy controller using a T-S fuzzy model. A T-S fuzzy controller can be designed using a T-S fuzzy model by using the antecedent part of the T-S fuzzy model as that of the T-S fuzzy controller. The synthesis of parallel-distributed compensation (PDC) type T-S fuzzy control systems can, in a systematic and analytic way, find proper feedback gains that simultaneously guarantee stability and system's performance. Therefore they have very broad use in fuzzy gain scheduling. Fuzzy logic and target-maneuver detection of the noise characteristics of a monopulse radar seeker are influenced by the motion of the target. When the relative rate of rotation between the radar and the target is low, the received noise is roughly Gaussian. If the target rotates rapidly, as in banking as a prelude to a turn, the noise statistics change. A set of fuzzy inference systems was developed that monitored the noise statistics and provided an indication of when the target maneuvered.

Hybrid modeling (HM) and land-vehicle HM is a methodology which fuses conventional mathematical techniques and intelligent techniques such as fuzzy logic, neural-networks, and genetic algorithms. HM can result in models without the cost of increased dimensionality when compared to models derived by applying strictly intelligent-based methods. In particular, HM is applied with a particular emphasis on the application of fuzzy logic in land vehicles. The force characteristics of electrically propelled land vehicles are fundamental in understanding and predicting the vehicles' handling and performance properties. The fuzzy-logic methodology has been applied to predict the tyre forces by combining a well-balanced HM system representation that can provide a high-fidelity model.

Expert systems have been the most obvious recipients of the benefits of fuzzy logic, since their domain is often inherently fuzzy. Examples of expert systems with fuzzy logic central to their control are decision-support systems, financial planners, and diagnostic systems.

The future paradigm shift

Fuzzy logic, by exploring uncertainty and unpredictability, continues to shape the world in which we live.

(continued on page 21)

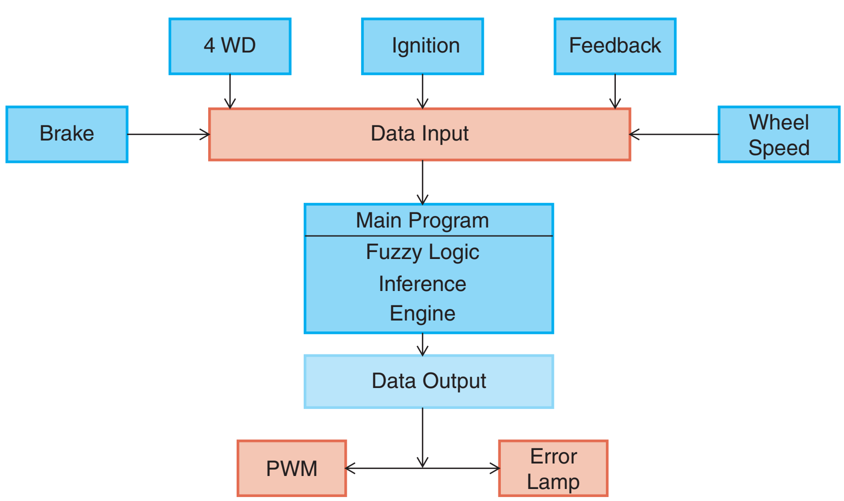


Fig. 6 ABS block diagram

Further afield, applications such as injecting biocompatible MR fluids directly into the bloodstream, where they could control the flow of blood to cancerous tumors, are also being envisioned. Indeed, MR fluids might someday flow in the veins of robots to animate hands and limbs as naturally as those of humans or provide active hand grips that conform to the shape of each individual hand or fingers. Other future applications include creating magneto-liquid-mirror telescopes that bend and deform to cancel out the twinkling of starlight, enabling astronomers to make better observations, and shock absorbers for payloads in spacecraft.

But however they end up being used—whether in cars, buildings, or mobile phones—it appears that smart fluids have a solid future ahead of them.

Acknowledgments

Inspiration for the figures came from Philips Corporation and *The Economist*.

Read more about it

- mrfluids.com [Online]. Available: http://www.mrfluids.com
- Advanced Fluid Systems Web site [Online]. Available: http://www.a-f-s.com
- Technology Catalysts Web site [Online]. Available: http://www.technology-catalysts. com/C&A_Materials/elector.htm

About the author

Guruprasad "Guru" Madhavan is an associate editor of *IEEE Potentials* from the Thomas J. Watson School of Engineering and Applied Science and School of Management at the State University of New York in Binghamton. E-mail: guru@binghamton.edu.

Paradigm shift — an introduction to fuzzy logic

(continued from page 10)

As the foundation of neural networks and artificial intelligence, the use of fuzzy logic can only expand and our understandings of it develop. It could facilitate human-to-machine speech, accounting for differing accents, tone, pronunciations, and, eventually, even different languages. Fuzzy logic could also be used in systems that are able to analyze literary works and discuss important concepts of those works.

Fuzzy systems will play role in a digital and connected world, automating decisions, intelligently analyzing large amounts of data, and learning from their mistakes.

Though fuzzy logic has huge potential, we will only get the best of it when it is integrated with computing, artificial intelligence, neural networks, and related areas.

Read more about it

- "Fuzzy logic: An overview of the latest control methodology," Texas Instruments, 1993. Available: http://focus.ti.com/lit/an/spra028/spra028.pdf
- P. Elmer-DeWitt "Time for some fuzzy thinking," *Time*, p. 79, Sept. 25 1989.
- J.G. Klir, "Japanese advances in fuzzy theory and applications," *SIB*, vol. 16, no. 3, pp. 65–74..
- G. Bojadziev and M. Bojadziev, *Fuzzy Logic for Business, Finance, and Management*. Singapore: World Scientific, 1997
- C. von Altrock, *Fuzzy Logic & Neurofuzzy Applications in Business and Finance*. Englewood Cliffs, NJ: Prentice Hall, 1997.
- T. Hageman, "Omron's Fuzzy Technology Business Promotion Center," *Comp. Sci*., May 1991.
- D. Elting, M. Fennich, R. Kowalczyk, and B. Hellenthal, Fuzzy anti-lock brake system solution [Online]. Available: http://www.intel.com/design/mcs96/designex/2351.htm
- Why use fuzzy logic? [Online]. Available: http://www.aptronix.com/fide/whyfuzzy.htm#figur
- FIDE Power Design Tool [Online]. Available: http://www.aptronix.com/fide/
- L. Smith (Apr. 2003). An introduction to neural networks [Online]. Available: http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html
- P.A. Fishwick (1994). Computer simulation: Growth through extension [Online]. Available: http://www.cis.ufl.edu/~fishwick/paper/paper.html

About the author

Joseph Bih obtained his master's degree in business computer information systems in May 2001 from the University of North Texas. He has been teaching at Jarvis Christian College and Tyler Junior College since then. With MCSE, OCP, and other IT certifications, he is actively involved in the IT industry. He is also an IEEE Member and an ACM member. His research interest areas include network security and management, system development, and new technology applications. E-mail: jjmbih@ieee.org.