CIRCUIT CELLAR INK®
THE COMPUTER APPLICATIONS JOURNAL

Constantin von Altrock

Fuzzy Logic in Automotive Engineering

Automotive engineering is one area where fuzzy logic has made significant inroads. Constantin brings us up to date with the latest developments. He examines fuzzy logic's impact on ABS braking, engine control systems, transmissions, and antiskid steering.

Fuzzy logic is a powerful way to put engineering expertise into products in a short amount of time. It's highly beneficial in automotive engineering, where many system designs involve the experience of design engineers as well as test drivers.

Over the past years, fuzzy logic has become is a common design technology in Japan, Korea, Germany, Sweden, and France. The reasons are manifold.

First, control systems in cars are complex and involve multiple parameters.

Second, the optimization of most systems is based on engineering expertise rather than mathematical models. "Good handling," "Fahrvergnügen," and "riding comfort" are optimization goals that can't be defined mathematically.

Third, automotive engineering is competitive on an international scale. A technology that proves a competitive advantage is soon commonly used.

In this article, I point to case studies in antilock braking systems (ABS), engine control, and automatic gearbox control. I show how superior performance is achieved via fuzzy-logic and neural-fuzzy design techniques. I also discuss development methodologies, tools, and code speed/size requirements.

ABS WITH FUZZY LOGIC

In 1947, Boeing developed the first ABS for airplanes as a mechanical system. Today, ABS is standard equipment on most cars. A microcontroller and electronic sensors measure the speed of every wheel and control the fluid pressure for the brake cylinders.

Mathematical models for a car's braking system exist, but the interaction of the braking system with the road is far too complex to model adequately. Hence, today's ABS contains the engineering experience of years of testing in different roads and climates.

PRODUCING FUZZY ABS

Because fuzzy logic is an efficient way to put engineering knowledge into a technical solution, it's no surprise that many ABS applications are already on the market. Currently, Nissan and Mitsubishi ship cars with fuzzy ABS. Honda, Mazda, Hyundai, BMW, Bosch, Mercedes-Benz, and Peugeot are working on solutions as well.

ABS also benefits from fuzzy logic's high computational efficiency. During a control loop time of 2–5 ms, the controllers must fetch all sensor data, pre-

Road Condition	Optimum Slack (s)
Dry	0.2
Slippery or Wet	0.12
Ice or Snow	0.05

Table 1—*The slack value for maximum brake effect depends on the road condition.*

process it, compute the ABS algorithm, drive the bypass valves, and conduct the test routines. Any additional function thus has to be computationally efficient.

Most ABS systems use 16-bit controllers, which can compute a medium size fuzzy-logic system in about 0.5 ms, using only about 2-KB ROM space [1]. You can check out a comparison of computing times of fuzzy-logic systems on different microcontrollers [2].

BRAKING BASICS

There are different ways in which fuzzy logic is used in ABS design. The implementation of Nippondenso [3] that I present exhibits an intelligent combination of conventional techniques with fuzzy logic.

Let's first discuss some basics of the braking process. If a wheel rotates exactly as fast as it corresponds to the speed of the car, the wheel has no braking effect at all. If the wheel doesn't rotate at all, it is blocked.

The blocking situation has two disadvantages. First, a car with blocking wheels is hard to steer. Second, the brake effect is not optimal. The point of optimum brake effect is between these two extremes.

The speed difference between the car and the wheel during braking is called "slack." Its definition is:

$$s = \frac{V_{car} - V_{wheel}}{V_{car}}$$

where s is slack (between 0 [no braking] and 1 [blocking]), V_{car} is the car's velocity, and V_{wheel} is the wheel's velocity.

Figure 1 plots the relation between brake effect and slack for different road surfaces. For $s = 0$, the wheel's speed equals the car's. In the case of $s = 1$, the wheel blocks completely.

The curves show that the optimum brake effect lies between these two extremes. However, the point of maximum brake effect depends on the type of road. Table 1 lists typical values.

ROAD SURFACE

Conventional ABS controls the bypass valves of the brake fluid so the slack equals a set value. Most manufacturers program this set value to a slack of 0.1, which is a good compromise for all road conditions.

But, as Figure 1 and Table 1 show, this set value is not optimal for every road type. By knowing the road type, the braking effect can be enhanced further.

So, how do you determine what the road type is? Asking the driver to push a button on the dashboard before an emergency brake is not feasible.

Sensors provide one logical alternative. Many companies have evaluated different types of sensors and concluded that sensors which deliver good road-surface identification are too expensive or not sufficiently robust.

However, consider sitting in a car equipped with a standard ABS. After driving at a known speed, you could jam on the brake so the ABS starts to work.

Even if you didn't know what the road surface was like, you could make a good guess from the car's reaction. If a driver can estimate the road surface from the car's reaction, fuzzy logic can implement the same ideas into the ABS.

Nippondenso did exactly this. When the ABS first detects the wheel blocking, it starts to control the brake-fluid valves so each wheel rotates with a slack of 0.1.

The fuzzy-logic system then evaluates the reaction of the car to the braking and estimates current road surface. Considering this estimate, the ABS corrects the set value for the slack to achieve the best braking effect in the interval from $s = 0.05$ to $s = 0.2$.

The fuzzy-logic system only uses input data stemming from the existing sensors of the ABS. Such input variables are deceleration and speed of the car, deceleration and speed of the wheels, and hydraulic pressure of the brake fluid. These variables indirectly indicate the current operation point of the braking and its behavior over time.

Experiments show that a first prototype with just six fuzzy-logic rules improves performance significantly. On a test track alternating from snowy to wet roads, the fuzzy ABS detected the road-surface changes even during braking.

A FUZZY BRAKE?

Due to the high competition in this area, most manufacturers are reluctant to publish any details about the technologies they use. The cited application only shows results from an experimental fuzzy-logic system. The details about the final product aren't published.

Also, some car makers (especially in the U.S.) worry about the negative connotation of the word "fuzzy." Since it implies imprecision and inexactness, manufacturers are afraid that drivers may think a fuzzy ABS is inferior. Others are threatened by the possibility of a suit in which a clever lawyer suggests to a layman's jury that a fuzzy-logic ABS is something hazardous.

In Japan, where an appreciation for ambiguity lies in the culture, "fuzzy" doesn't have a negative connotation. By contrast, it's an advantage, as it enables intelligent systems. Hence, companies are proud of its use and promote it in their advertising.

In Germany, on the other hand, the concepts of fuzziness and engineering masterpiece do not fit together well in the public perception. Hence, most manufacturers

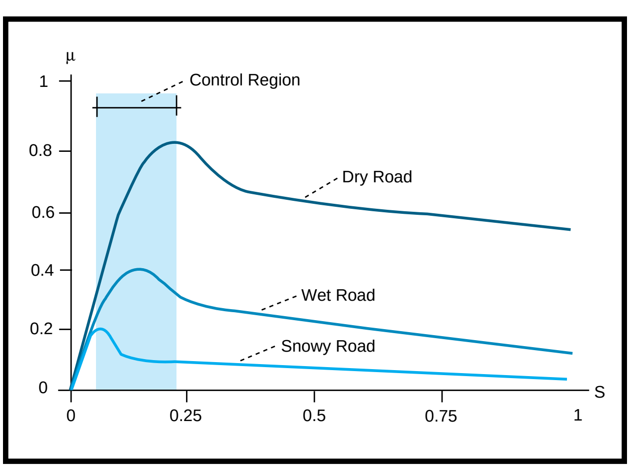

Figure 1—*This plot illustrates brake effect over the wheel slack* s *for dry, wet, and snowy road surfaces. (μ is the friction coefficient, or measure of brake effect.)*

using fuzzy logic in ABS hide the fact. After all, a fuzzy-logic system is only a segment of assembly code in a microcontroller. Once implemented, who can tell that this code contains fuzzy logic?

VERIFICATION AND STABILITY

When many publications about fuzzy logic appeared for the first time about five years ago, even reputed scientists and professors in the U.S. stated that fuzzy logic shouldn't be used for critical applications. They claimed that it produces inherently instable systems.

This attitude is truly shameful. They demonstrated only that they did not understand what fuzzy logic is about.

A fuzzy-logic system is a time invariant, deterministic, and nonlinear system—nothing fuzzy about that. Such systems are already known and applied in control engineering, and conventional stability theory covers them well [4].

In the case of a fuzzy ABS, stability isn't even an issue. Conventional ABS was considered stable for any slack set value in the interval from 0.05 to 0.2. Hence, a fuzzy-logic road-surface estimator that tunes this value to the optimum cannot make the ABS instable.

Let's move on to see how fuzzy logic easily implements human experience in an embedded engine-control system.

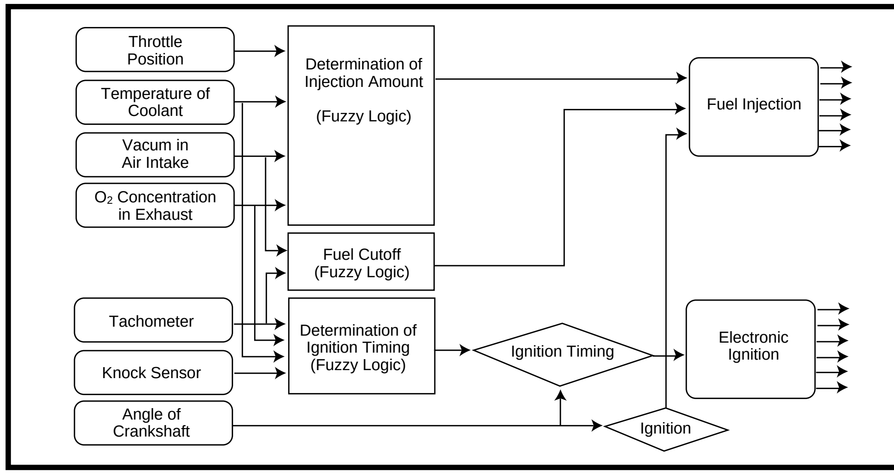

Figure 2—*As you can see, the engine controller of NOK Corporation contains three fuzzy-logic modules.*

ENGINE CONTROL

The control of car and truck engines is becoming increasingly more complex with more stringent emission standards and constant effort to gain higher fuel efficiency. Twenty years ago, control systems were mechanical (i.e., carburetor, distributor, and breaker contact). Now, microcontroller-based systems control fuel injection and ignition.

Since the control strategy for an engine depends strongly on the current operating point (e.g., revolutions, momentum, etc.), linear control models, (e.g., PID) are not suitable.

On the other hand, no mathematical model describing the complete behavior of an engine exists. Most engine controllers use a look-up table to represent the control strategy. The table is generated from the results of extensive testing and engineer's experience.

The generation of such a look-up table, however, is only suitable for three dimensions (two inputs and one output). Also, the generation and interpretation of such tables is difficult and considered a black art.

Although fuzzy logic can replace these look-up-tables, most manufacturers will not publish any details on a fuzzy-logic engine-control solution.

This secretiveness is due to the fact that the rules of the fuzzy-logic system make the entire engine-control knowledge of the company completely transparent. They are afraid competitors will learn too much about the solution by disassembling the fuzzy-logic rules.

Listing 1—*The current operation point of the engine is classified by the linguistic variable* `Situation`*. Each linguistic term denotes a typical operation point. Because each term is represented as a fuzzy-logic membership function, the linguistic variable can classify all other operation points, too.*

```
linguistic variable Situation {

Term 1: Start
Control strategy is that the cold engine runs smooth. Ignition is
timed early, and the mix is fat;

Term 2: Idle
Control ignition timing and fuel injection depending on engine
temperature to ensure that the engine runs smooth;

Term 3: Normal drive, low or medium load
Maximize fuel efficiency by meager mix, watch knocking;

Term 4: Normal drive, high load
Fat mix and early ignition to maximize performance. The only
constraint is the permitted emission maximum;

Term 5: Coasting
Fuel cut-off, depending on situation;

Term 6: Acceleration
Depending on load, fattening of the mix }
```

IDENTIFY DRIVING CONDITION

Nok and Nissan's case study [5] gives the benefits of fuzzy logic in engine control. Figure 2 depicts the components of this engine controller, which contains three fuzzy-logic modules.

The system first notes the engine's operational condition by the linguistic variable `Situation`. This variable has the linguistic terms in Listing 1.

The determination of `Situation` is a state estimation of the operation point. Because `Situation` is a linguistic variable, more than one term can be valid at the same time, so combinations of the operational points can be expressed as defined by the terms.

A possible value of `Situation` could be {0.8; 0; 1; 0; 0; 0.3}. Linguistically, this value represents the driving condition "engine started a short while ago,

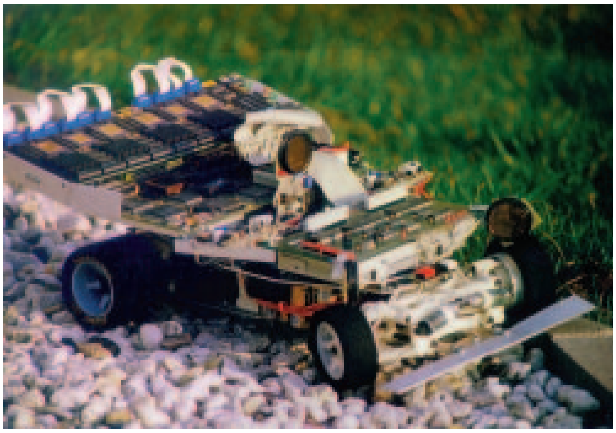

Photo 1—*This model car is used in high-speed driving experiments.*

normal drive condition at medium or low load, slightly accelerating." From this operation-point identification, the individual fuzzy-logic modules control injection, fuel cutoff, and ignition.

Like ABS, engine control needs a very short loop time. Some systems are as fast as 1 ms for an entire control loop. Some manufacturers design the system using fuzzy logic but then translate it into a look-up table for faster processing.

Although a look-up table computes faster, memory requirements may prohibit its use. A look-up table with two inputs and one output, all 8-bit resolution, already requires 64 KB of ROM.

Restricting the resolution of the input variables to 6 bits each, the look-up-table still requires 4 KB. A table with three inputs and one output, all inputs 6-bit resolution, requires ¼ MB.

Some engineers implemented a look-up table with a limited resolution and used an interpolation algorithm. However, the interpolation needs about as much computing time as the fuzzy-logic system itself [2].

Another published application of fuzzy logic in engine control is an idle control unit by Ford Motor Corp. [6].

Next, let's check out automatic-transmission control to show how fuzzy-logic systems can adapt their control strategy to drivers.

ADAPTIVE AUTOMATICS

When the first three-speed automatic transmissions appeared on the market about 30 years ago, the engine power of most cars was just sufficient to keep the car in pace with traffic. The necessity of getting maximum momentum from the engine determined the shift points for the gears.

Now, when most car engines can deliver much more power than necessary to keep the car in pace with traffic, automatic transmission systems have up to five speeds, and fuel efficiency has become an important issue, controlling shift points is much more complex.

Five speeds and higher engine power give the automatic-transmission system a much higher degree of freedom. Driving at 35 MPH, a three-speed automatic transmission has to select second gear. A five-speed transmission with a powerful engine can select second gear for maximum acceleration, third gear for normal driving condition, and fourth gear for minimal acceleration.

ACCELERATE OR SAVE FUEL

Unfortunately, the goal for the control strategy is in a dilemma. For maximum fuel efficiency, you want to select the next higher gear as early as possible. But for maximum performance, you switch to the next higher gear later.

If you have a standard shift, you choose your strategy depending on the traffic condition. An automatic gearbox has no understanding of the traffic condition or the driver's wishes.

However, intelligent control techniques can enhance automatic transmissions as it is based on experience and engineering knowledge rather than mathematical models. Fuzzy logic therefore proves to efficiently implement the technology.

In 1991, Nissan introduced fuzzy-logic–controlled automatic five-speed transmission systems [7, 8]. Honda followed in 1992 [9], and GM/Saturn in 1993.

The job for the fuzzy-logic system in these applications is similar:

- avoid "nervous" shifting back and forth on winding or hilly roads
- understand whether the driver wants economical or sporty performance
- avoid unnecessary overdrive, if switching to the next lower gear does not deliver more acceleration

Figure 3 shows a typical situation on a fast, winding road. With a standard shift, you'd leave it in fourth gear, but a five-speed automatic transmission switches between the fourth and fifth gears depending on the speed of the car.

The fuzzy-logic transmission controller evaluates more than just the current speed of the car. It also analyzes how the driver accelerates and brakes.

To detect a winding road, the fuzzy-logic controller looks at the number of accelerator pedal changes within a period. Figure 4 shows the definition of the linguistic variable `Accelerator pedal changes`. The variance of the accelerator pedal changes is input to the fuzzy-logic controller.

Some of the rules estimating the road and driving conditions from these input variables are:

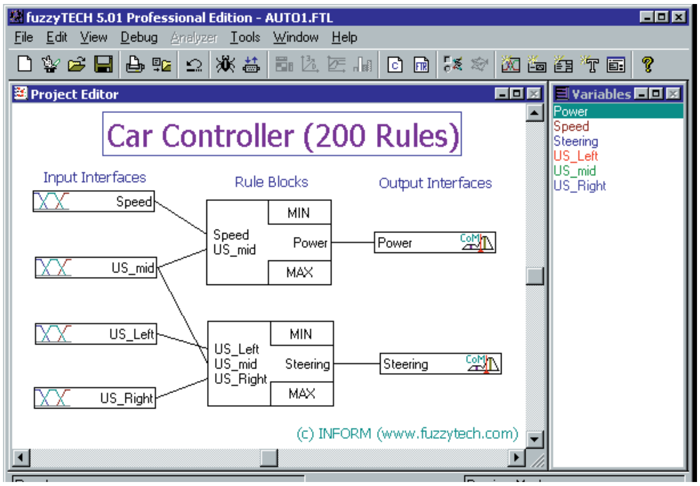

Photo 2—*The first version of the fuzzy-logic controller has 200 rules in two rule blocks. The four left boxes indicate input interfaces for sensors, the two right boxes indicate output interfaces for the actuators, and the two large boxes in the middle represent fuzzy-logic rule blocks.*

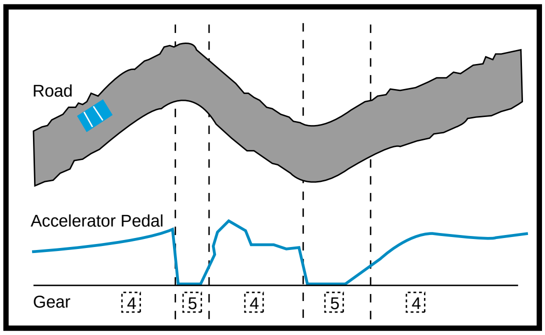

Figure 3—*A five-speed automatic transmission with fixed shift points always switches between fourth and fifth gear on a winding road. A driver with a shift gearbox would leave it in fourth gear.*

- many pedal changes within a period indicate a fast and winding road
- few pedal changes within a period indicate a freeway
- many pedal changes within a period and a high variance of pedal changes indicate a slow and winding road
- medium variance of pedal changes indicates a fast and winding road
- low variance of the pedal changes indicates a freeway

The interesting part of this application is that the fuzzy-logic controller uses the driver as the sensor. It interprets the driver's reaction to the road and driving conditions and adapts the car's performance accordingly.

This behavior could be used to define an intelligent control system. The technical system tries to understand whether the human is satisfied with its performance and adapts itself to suit the needs of the human using it.

"INTELLIGENT" TRANSMISSIONS

Another example of an automatic transmission system currently under development in Germany illustrates this possibility even better.

If drivers want to accelerate, but aren't satisfied with their cars' responce, they unconsciously push the pedal down even more wintin 1–1.5s. This scenario represents the subconscious reaction of most drivers to unsatisfactory acceleration.

Most drivers don't even realize that they like the car to accelerate faster. If an automatic transmission system is capable of detecting this, it can move the shift points higher to achieve more acceleration.

The opposite case is similar. If the automatic transmission detects that the driver accelerates carefully and takes the foot off the accelerator long before red lights, chances are that the driver wants high fuel efficiency.

WHY FUZZY LOGIC?

The question remains, why do you need fuzzy logic to implement these intelligent functions? My answer: while you can use other techniques to implement these control strategies, fuzzy logic is likely to be the most efficient.

Intelligent control strategies are built on experience and experiments rather than from mathematical models. Hence, a linguistic formulation is more efficient.

These strategies mostly involve a large number of inputs. Most of the inputs are only relevant for some specific condition. Using fuzzy logic, these inputs are only considered in the relevant rules, keeping even complex control-system designs transparent.

Another consideration is that intelligent control strategies implemented in mass-market products have to be implemented cost efficiently. In comparison to conventional solutions, fuzzy logic is often much more computational and code-space efficient.

Let's look now at how fuzzy logic enables the design of new functionality for automatic steering control.

ANTISKID STEERING

Active stability control systems in cars have a long history. First, ABS improved braking performance by reducing the amount of brake force applied by the driver to what the road surface can take. This system avoids skidding and sliding, resulting in shorter braking distances.

Second, traction-control systems, which do essentially the same thing as ABS, improve acceleration. By reducing engine power applied to the wheels to what the road can take, a traction-control system maximizes acceleration and minimizes tire wear.

After skid-controlled braking and acceleration, the next logical step is skid-controlled steering. An antiskid steering system (ASS) reduces the steering angle applied by the driver through the steering wheel to the amount the road can take. It optimizes the steering action and avoids sliding since a sliding car is very difficult to restabilize, especially for drivers not accustomed to such situations.

Though an ASS makes a lot of sense from a technical point of view, such a system is harder to market. For an ABS, you can prove that it never performs worse than a traditional braking system. For an ASS, this is hard to prove.

Also, it may be difficult to sell cars that "take over the steering" in emergency situations. Even ABS faced a long period of rejection by customers because they felt uneasy about a system "inhibiting" their brake action.

For these reasons, it may take a long time before ASS will be implemented in a production car. All results shown in this section stem from the research of a German car manufacturer [10]. Because this system is one of the most complex fuzzy-logic embedded systems ever developed, it effectively demonstrates the potential of the technology.

THE TEST VEHICLE

Real experiments were made on a modified Audi sedan and the 20″ model car shown in Photo 1. In the following discussion, I only present the results derived from the model-car experiments.

A midmounted 1-hp electric motor powers the car, rendering the power-to-weight ratio of a race car. This setup enables the researchers to perform skidding and sliding experiments in extreme situations at high speeds.

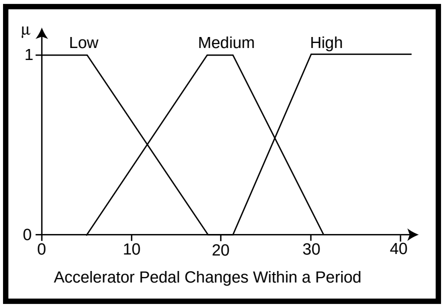

Figure 4—*Here, driving condition is classified using a linguistic variable. The variable linguistically interprets the amplitude of accelerator pedal changes within a certain period.*

On dry surface, the car reaches a velocity of 20 MPH in 3.5 s, with top speeds up to 50 MPH. The speed for most experiments ranges from 20 to 30 MPH. Each wheel features individual suspension and has a separate shock absorber. The car has disk brakes and a lockable differential [10].

The car's controller uses the motherboard of a notebook PC connected to an interface board driving the actuators and sensors. Actuators are power steering servo, disk brake servo, and pulse-width modulated motor control.

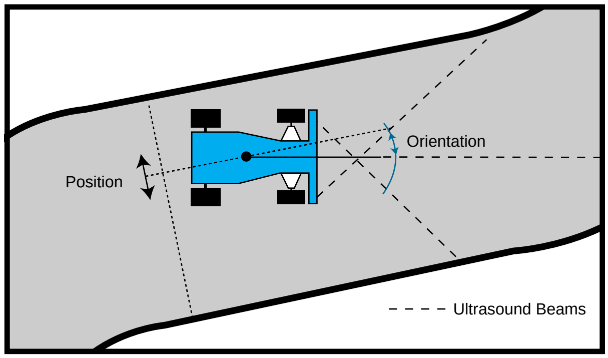

Figure 5—*Three ultrasound sensors guide the car in the track.*

Sensors are three ultrasound (US) distance sensors for tracking guidance (see Figure 5) and infrared (IR) reflex sensors in each wheel for speed. The control loop time—from reading in sensor signals to setting the values for the actuators—is 10 ms.

To measure the dynamic state of the car (e.g., skidding and sliding), IR sensors measure the individual speed of all four wheels. Evaluating wheel-speed differences, the fuzzy-logic system interprets the current situation.

Three fixed-mounted US sensors measure the distance to the next obstacle to the front, left, and right. This setup permits autonomous operation of the car. Low-cost sensors were intentionally used in this study—rather than CCD cameras and image-recognition techniques—to show that expensive sensors can be replaced by a fuzzy-logic control strategy.

Figure 6 shows a sample experiment involving the model car. The obstacle is placed right after the curve, so the US sensors of the car detect the obstacle too late.

To not hit the obstacle, the car has to decide for a very rapid turn. To optimize the steering effect, the anti-skid controller must reduce the desired steering angle to the maximum the road can take, avoiding both sliding and hitting the obstacle.

MODEL BASED VS. FUZZY LOGIC

In theory, you can build a mechanical model for a car and derive a mathematical model with differential equations to implement a model-based controller. In reality, the complexity of this approach is overwhelming, and the resulting controller would be difficult to tune.

Here is the point for fuzzy logic: race-car drivers can control a car in extreme situations very well without solving differential equations. Hence, there must be an alternative way for anti-skid steering control.

This alternative way is to represent the driving strategy in engineering heuristics. Although there are multiple ways of expressing engineering heuristics, fuzzy logic has proven very effective for the following reasons.

You can often formulate engineering heuristics in if-then causalities. In contrast to other methods of expressing if-then causalities (e.g., expert systems), the computation in a fuzzy-logic system is quantitative rather than symbolic.

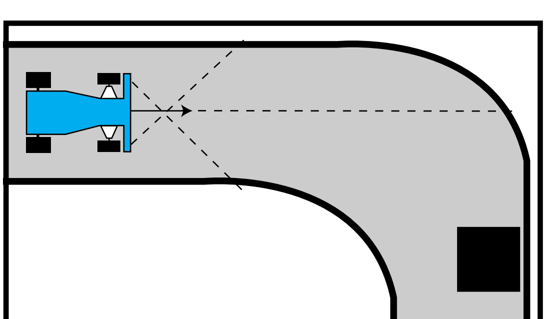
Figure 6—*In this example of an experiment, the car's ultrasound sensors will detect the obstacle placed right after the curve very late, making a rapid turn necessary.*

In a fuzzy-logic system, you use a few rules to express general situations, and then the fuzzy-logic algorithm deduces decisions for the real situations that occur. A conventional expert system needs a rule for each possible situation.

In a fuzzy-logic system, every element is self-explanatory. Linguistic variables are close to the human representation of continuous concepts. Fuzzy if-then rules combine these concepts much the same way humans do.

Fuzzy logic is nonlinear and multiparametric by nature. So, it can better cope with complex control problems that are also nonlinear and involve multiple parameters.

And finally, fuzzy logic can be efficiently implemented in embedded control applications. Even on a standard microcontroller, a fuzzy-logic system can outperform a comparable conventional solution both by code size and computing speed.

DESIGN AND IMPLEMENTATION

Photo 2 shows the first version of a fuzzy-logic controller for the car. The objective for this controller was autonomous guidance of the car in the track at slow speed, where no skidding and sliding yet occurs.

In Photo 2, the lower rule block uses the distances measured by the three US sensors to determine the steering angle. The upper rule block implements a simple speed control by using the distance to the next obstacle measured by the front US sensor and the speed of one front wheel only.

Due to the slow speeds, no skidding or sliding occurs. All wheel speeds are the same.

This first version of the fuzzy-logic controller contained about 200 rules and took only a few hours to implement.

The second version of the fuzzy-logic controller implements a more complex fuzzy system for dynamic stability control. It includes antilock braking as well as traction and antiskid steering control (see Photo 3).

This 600-rule fuzzy-logic controller has two stages of the fuzzy inference. The first stage, represented by the three left rule blocks, estimates the state variables of the car's dynamic situation from sensor data. The two lower rule blocks estimate skidding and sliding states from speed sensor signals, while the upper rule block estimates the car's position and orientation in the test track.

Note that the output of the left three rule blocks—the state variable estimation—is linguistic rather than numerical. An estimated state of the car can therefore be "the position is rather left, while the orientation is strongly to the right, and the car skids over the left front wheel."

The second stage, represented by the three right rule blocks, uses these estimations as inputs to determine the best control action for that driving situation. The upper rule block determines the steering angle, the middle one the engine power to be applied, and the lower one the brake force.

Such a two-stage control strategy is similar to the human behavior. It first analyzes the situation and then determines the action. It also allows for efficient optimization, since the total of 600 rule structures in six rule blocks can be designed and optimized independently.

The first version of the controller was only able to guide the car on autonomous cruise (see Photo 2). The second version also succeeded to dynamically stabilize the car's cruise via ABS, traction control, and ASS (see Photo 3).

However, this version required a much longer design time before the results were completely satisfactory. The second version also uses advanced fuzzy-logic technologies such as FAM inference [11] and the Gamma aggregational operator [12].

ONLINE DEVELOPMENT

The development of the fuzzy-logic system used the *fuzzy*TECH software product [11]. Given the graphical definition of the system structure (cf. Photos 2 and 3), the linguistic variables, and the rule bases, *fuzzy*TECH's compiler generates the system as C or assembly code.

This code was implemented on a PC board mounted on the car. Figure 7 shows how the running fuzzy-logic system was modified on-the-fly for optimization.

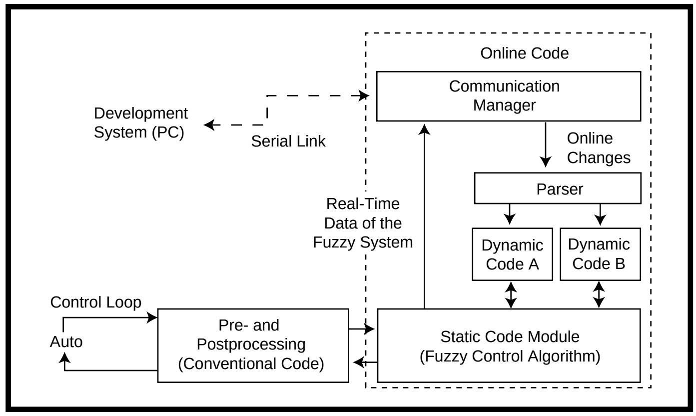

Figure 7—*The fuzzyTECH Online Edition features both visualization of running system and modifications on-the-fly.*

The fuzzy-logic code is separated into two segments. One contains all static parts—code that doesn't need to be modified for system alterations. The other segment contains all dynamic parts—the code containing membership functions of the linguistic variables, the inference structure, and the rules.

The dynamic segment is doubled, with only one of the segments active at the same time. In this situation, the parser, linked to the development PC via a communications manager, can modify the inactive code segment.

This technique enables modifications on the running system without halting or compiling. At the same time, the entire inference flow inside the fuzzy-logic controller is graphically visualized on the PC, since the communications manager also transfers all real-time data.

The ASS example demonstrates the applicability of fuzzy-logic technologies for a complex control problem found in the automotive industry. The system was a straightforward design based on experimental experience without a mathematical model of the process.

During optimization, the control strategy was easy to optimize due to the linguistic representation inherent in the fuzzy-logic system. Tests and verification were expedited due to the controller's transparency. And, the poor computational performance of early fuzzy-logic software solutions was overcome via a new generation of software-implementation tools [2, 13].

In the remainder of this article, I provide an overview of other automotive applications where fuzzy logic has been used successfully.

HVAC IN CARS

Fuzzy-logic design technologies are well-established in heating and air conditioning of residences and offices. Hence, it's no surprise that many car manufacturers also use fuzzy logic in their HVAC-system designs.

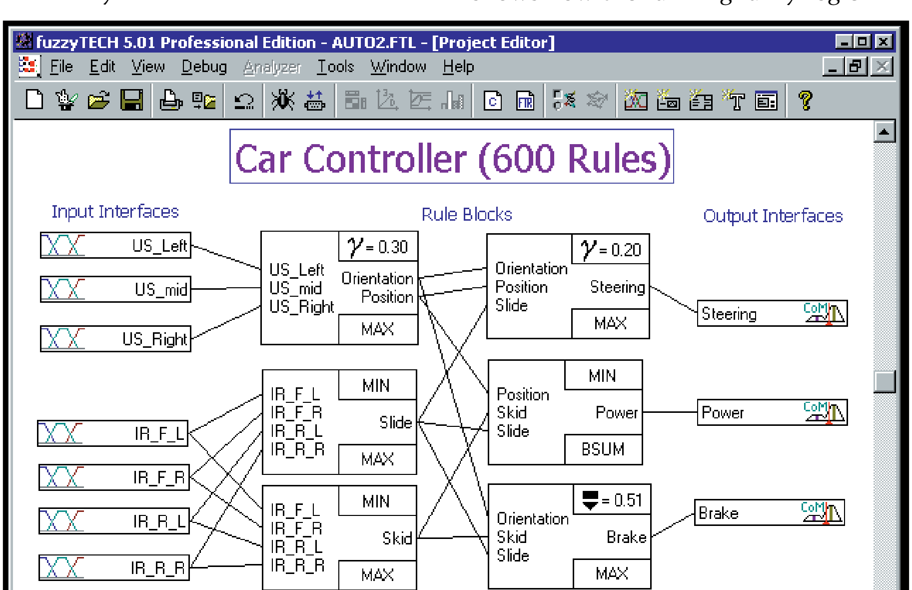

Photo 3—*Here's the second version of the fuzzy-logic controller. The controller uses advanced fuzzy-logic design technologies and contains a total of 600 rules.*

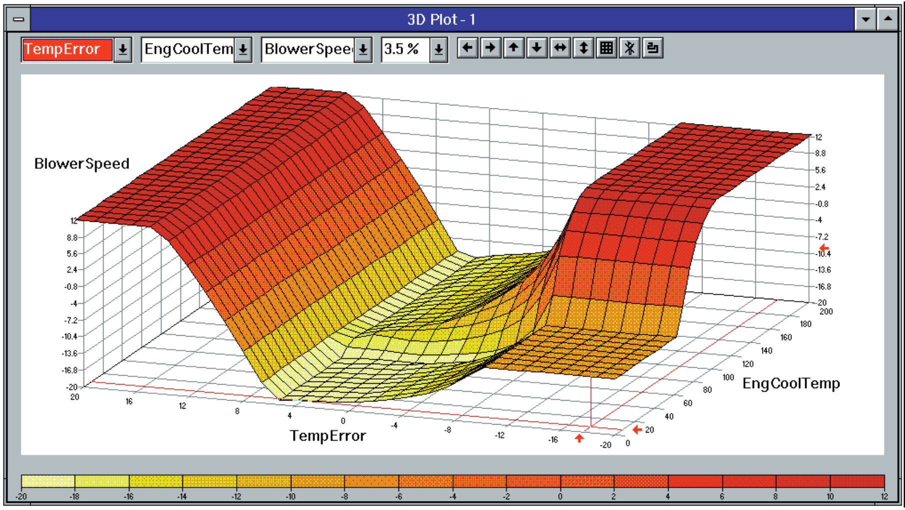

Photo 4—*This graph depicts the control surface of the air conditioner's blower-speed control. Blower speed is determined by temperature error and engine-coolant temperature.*

While most car manufacturers work on these systems, very few publish their efforts. The control approach in general and hence the use of fuzzy logic in the design differ significantly for each manufacturer. In this section, I use an example that Ford Motor Company developed in the U.S. [14].

The fundamental goal of HVAC in cars is to make vehicle occupants comfortable. Human comfort, however, is a complex reaction, involving physical, biological, and psychological responses to the given conditions. The performance criterion—comfort—is not some well-defined mathematical formula but a sometimes inconsistent and empirically determined goal.

Typical HVAC-system sensors measure cabin temperature, ambient temperature, sun heating load, humidity, and other factors. Typical actuators are variable speed blowers, means for varying air temperature, ducting, and doors to control the direction of airflow, and the ratio of fresh to recirculated air. This multiple-input, multiple-output control problem doesn't fall into any convenient category of traditional control theory.

Photo 4 shows the control surface of a part of a HVAC system—the blower-speed control. Blower speed depends on two input variables—the temperature error (i.e., the in-car temperature minus the set-point temperature) and the engine-coolant temperature.

Photo 5 shows the rule base. If the temperature error is zero, low blower speed is desired. If its too hot inside (i.e., positive temperature error), high blower speed is needed to cool the cabin.

If the error is negative, indicating that it's too cold inside, and the engine is cold, little blower speed is needed for defrost. If the error is negative but the engine is warm, high blower speed is needed to heat up the cabin.

OTHER APPLICATIONS

This section briefly introduces some other examples of fuzzy-logic control in automotive engineering. For details, refer to the papers cited.

Peugeot Citroën of France developed a fuzzy-logic system for an intelligent cruise control [15]. The system combines multiple functions for autonomous intelligent cruise control (i.e., following another vehicle, stop and go procedures, and emergency stop). The system uses three fuzzy-logic blocks with four inputs, one output, and 30 rules each.

Optimization and verification of the rule base used a Citroën XM sedan with automatic gearbox and ABS as test vehicle. The fuzzy-logic controller runs on an 8-bit microcontroller.

The car uses a speed sensor and a single-beam telemeter for the distance to the next car. The actuators command brake pressure and accelerator. Tests show the fuzzy-logic controller can handle the cruise under all the tested conditions.

Future regulations in the European Community (EC) require a speed control for limiting truck speeds on roads in Europe. Today's speed limiters use adaptive PID-type controller. However, the resulting truck behavior is unsatisfactory, compared to an experienced driver.

Therefore, a number of recent designs use fuzzy-logic control to achieve robust performance, even under the strong load changes of commercial trucks [16, 17].

A paper from Ford Electronics describes the design of a traction-control system for a radio-controlled model car [18]. The fact that Ford publishes model-car applications is symptomatic of the fear of many automotive manufacturers to admit that they use fuzzy logic as a design technique for "real" cars.

THE FUTURE IS FUZZY

Over the past five years, fuzzy logic has significantly influenced the design of automotive control systems. Since using fuzzy logic involves a paradigm shift in the design of a control system, five years is a short period. The move from analog to digital solutions has taken a much longer time.

The key reason for fuzzy logic's success in automotive engineering lies in the implications of its paradigm shift. Previously, engineers spent much time creating mathematical models of mechanical systems. More time went to

Photo 5—*The rule base for blower-speed control shows how the two variables of engine temperature and temperature error affect blower speed.*

Spreadsheet Rule Editor - BlowerControl

Matrix	IF		THEN	
Utilities	EngCoolTemp	TempError	DoS	BlowerSpeed
1		zero	1.00	low
2	high	negative	1.00	high
3		positive	1.00	high
4	low	negative	1.00	med_low
5				

real-world road tests that tuned the fudge factors of the control algorithms.

If they succeeded, they ended up with a control algorithm of mathematical formulas involving many experimental parameters. Modifying or later optimizing such a solution is very difficult because of its lack of transparency.

Fuzzy logic makes this design process faster, easier, and more transparent. It can implement control strategies using elements of everyday language. Everyone familiar with the control problem can read the fuzzy rules and understand what the system is doing and why.

It also works for control systems with many control parameters. Designers can build innovative control systems that would have been intractable using traditional design techniques.

The future for fuzzy logic in automotive engineering is bright. Semiconductor manufacturers are incorporating fuzzy-logic instruction sets in their controllers. Motorola just introduced the new 68HC12 family of 16-bit micros that integrate a complete instruction set for fuzzy logic at no extra cost.

Another new development is the upcoming IEC 1131-7 fuzzy-logic standard [19]. This international standard defines consistent fuzzy-logic development and documentation procedures.

With these two developments, designing with fuzzy logic becomes a much simpler task.

Constantin von Altrock began research on fuzzy logic with Hewlett-Packard in 1984. In 1989, he founded and still manages the Fuzzy Technologies Division of Inform Software Corp., a market leader in fuzzy-logic development tools and turn-key applications. You may reach Constantin at cva@inform-ac.com.

REFERENCES

[1] Intel, "Fuzzy Anti-Lock Braking System," developer.intel.com/design/MCS96/DESIGNEX/2351.htm, 1996.

[2] "Benchmark Suites for Fuzzy Logic," www.fuzzytech.com/e_dwnld.htm, 1997.

[3] N. Matsumoto et al., "Expert antiskid system," *IEEE IECON'87*, 810–816, 1987.

[4] C. von Altrock, *Fuzzy Logic and NeuroFuzzy Applications Explained*, Prentice Hall, Englewood Cliffs, NJ, 1995.

[5] H. Kawai et al., "Engine control system," *Proc. of the Int'l Conf. on Fuzzy Logic and Neural Networks*, Iizuka, Japan, 929–937, 1990.

[6] L. Feldkamp and G. Puskorius, "Trainable fuzzy and neural-fuzzy systems for idle-speed control," *2nd IEEE Int'l. Conf. on Fuzzy Systems*, 45–51, 1993.

[7] H. Takahashi, K. Ikeura, and T. Yamamori, "5-speed automatic transmission installed fuzzy reasoning," *IFES'91–Fuzzy Engineering toward Human Friendly Systems*, 1136–1137, 1991.

[8] H. Ikeda et al., "An intelligent automatic transmission control using a one-chip fuzzy inference engine," *Proc. of the Int'l. Fuzzy Systems and Intelligent Control Conf. in Louisville*, 44–50, 1992.

[9] P. Sakaguchi et al., "Application of fuzzy logic to shift scheduling method for automatic transmission," *2nd IEEE Int'l. Conf. on Fuzzy Systems*, 52–58, 1993.

[10] C. von Altrock, B. Krause, and H.-J. Zimmermann, "Advanced fuzzy logic control of a model car in extreme situations," *Fuzzy Sets and Systems*, **48:1**, 41–52, 1992.

[11] INFORM GmbH/Inform Software Corp., fuzzy*TECH and NeuroFuzzy Module 5.0 User's Manual*, Chicago, IL, 1997.

[12] H.-J. Zimmermann and U. Thole, "On the suitability of minimum and product operators for the intersection of fuzzy sets," *Fuzzy Sets and Systems*, **2**, 173–186, 1979.

[13] C. von Altrock and B. Krause, "On-Line-Development Tools for Fuzzy Knowledge-Base Systems of Higher Order," *2nd Int'l Conf. on Fuzzy Logic and Neural Networks Proceedings*, Iizuka, Japan, 1992.

[14] L.I. Davis et al., "Fuzzy Logic for Vehicle Climate Control," *3rd IEEE Int'l. Conf. on Fuzzy Systems*, 530–534, 1994.

[15] J.-P. Aurrand-Lions, M. des Saint Blancard, and P. Jarri, "Autonomous Intelligent Cruise Control with Fuzzy Logic," *EUFIT'93–1st Eur. Congress on Fuzzy and Intelligent Technologies*, Aachen, 1–7, 1993.

[16] V.M. Thurm, P. Schaefer, and W. Schielen, "Fuzzy Control of a Speed Limiter," ISATA Conf., 1993.

[17] www.fuzzytech.com/e_a_spe.htm.

[18] R. Russ, "Designing a Fuzzy Logic Traction Control System," *Proc. of the Embedded Systems Conf.*, **2**, 183–196, 1994.

[19] www.fuzzytech.com/e_iec.htm.

SOURCES

***fuzzy*TECH Development System**
Inform Software Corp.
2001 Midwest Rd.
Oak Brook, IL 60523
(630) 268-7550
Fax: (630) 268-7554
www.fuzzytech.com

Fuzzy-logic 68HC12 microcontroller
Motorola MCU Information
P.O. Box 13026
Austin, TX 78711
(512) 328-2268, x985
www.mcu.motsps.com
www.fuzzytech.com/motorola.htm