1

Introduction

Fuzzy controllers appear in consumer products – such as washing machines, video cameras, cars – and in industry, for controlling cement kilns, underground trains, and robots. A *fuzzy controller* is an *automatic controller*, a self-acting or self-regulating mechanism that controls an object in accordance with a desired behaviour. The object can be, for instance, a robot set to follow a certain path. A fuzzy controller acts or regulates by means of rules in a more or less natural language, based on the distinguishing feature: fuzzy logic. The rules are invented by plant operators or design engineers, and fuzzy control is thus a branch of intelligent control.

1.1 What Is Fuzzy Control?

Traditionally, computers make rigid *yes* or *no* decisions, by means of decision rules based on two-valued logic: *true–false*, *yes–no*, or 1 – 0. An example is an air conditioner with thermostat control that recognizes just two states: above the desired temperature or below the desired temperature. *Fuzzy logic,* on the other hand, allows a graduation from *true* to *false*. A fuzzy air conditioner may recognize 'warm' and 'cold' room temperatures. The rules behind this are less precise, for instance:

If the room temperature is warm and slightly increasing, then increase the cooling.

Many classes or *sets* have *fuzzy* rather than sharp boundaries, and this is the mathematical basis of fuzzy logic; the set of 'warm' temperature measurements is one example of a fuzzy set.

The core of a fuzzy controller is a collection of *verbal* or *linguistic* rules of the *if–then* form. Several variables may occur in each rule, both on the *if*-side and the *then*-side. Reflecting expert opinions, the rules can bring the reasoning used by computers closer to that of human beings.

In the example of the fuzzy air conditioner, the controller works on the basis of a temperature measurement. The room temperature is just a number, and more information

Foundations of Fuzzy Control Jan Jantzen

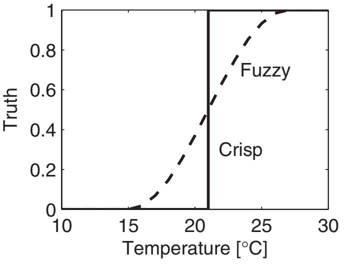

Figure 1.1: A warm room. The crisp air conditioner considers any temperature above 21 °C warm. The fuzzy air conditioner considers gradually warmer temperatures. (figwarm.m)

is necessary to decide whether the room is warm. Therefore the designer must incorporate a human's perception of warm room temperatures. A straightforward implementation is to evaluate beforehand all possible temperature measurements. For example, on a scale from 0 to 1, *warm* corresponds to 1 and *not warm* corresponds to 0:

Measurements (°C):	...	15	17	19	21	23	25	27	...
		|	|	|	|	|	|	|	
Grade:	...	0	0.1	0.3	0.5	0.7	0.9	1	...

The example uses *discrete* temperature measurements, whereas Figure 1.1 shows the same idea graphically in the form of a *continuous* mapping of temperature measurements to truth-values. The mapping is arbitrary, that is, based on preference and not mathematical reason.

1.2 Why Fuzzy Control?

If PID control (proportional-integral-derivative control) is inadequate – for example, in the case of higher-order plants, systems with a long deadtime, or systems with oscillatory modes (Åström and Hägglund 1995) – fuzzy control is an option. But first, let us consider why one would *not* use a fuzzy controller:

- The PID controller is well understood, easy to implement – both in its digital and analog forms – and it is widely used. By contrast, the fuzzy controller requires some knowledge of fuzzy logic. It also involves building arbitrary membership functions.
- The fuzzy controller is generally nonlinear. It does not have a simple equation like the PID, and it is more difficult to analyse mathematically; approximations are required, and it follows that stability is more difficult to guarantee.
- The fuzzy controller has more tuning parameters than the PID controller. Furthermore, it is difficult to trace the data flow during execution, which makes error correction more difficult.

On the other hand, fuzzy controllers are used in industry with success. There are several possible reasons:

- Since the control strategy consists of *if–then* rules, it is easy for a plant operator to read. The rules can be built from a vocabulary containing everyday words such as ‘high’, ‘low’, and ‘increasing’. Plant operators can embed their experience directly.

- The fuzzy controller accommodates many inputs and many outputs. Variables can be combined in an *if–then* rule with the connectives *and* and *or*. Rules are executed in parallel, implying a recommended action from each. The recommendations may be in conflict, but the controller resolves conflicts.

Fuzzy logic enables non-specialists to design control systems, and this may be the main reason for its success.

1.3 Controller Design

Established design methods such as pole placement, optimal control, and frequency response shaping only apply to linear systems, whereas fuzzy control is generally nonlinear. Since our knowledge of the behaviour of nonlinear systems is limited, compared with the situation in the linear domain, this book is based on a design procedure founded on linear control:

1. Design a PID controller
2. Replace it with a linear fuzzy controller
3. Make it nonlinear
4. Fine-tune it.

The idea is to exploit the design methods within PID control and carry them forward to fuzzy control. The design procedure is feasible, only because it is possible to build a linear fuzzy controller that functions exactly as any PID controller does. The following example introduces the design procedure.

1.4 Introductory Example: Stopping a Car

Consider this problem: Model the behaviour of a human driver stopping a car at a red stop light by using the brake pedal. Figure 1.2 defines the symbols.

Model the car first. Disregarding engine dynamics, skidding, slip, and friction – other than the frictional forces in the brake pads – the force F causes an acceleration a according to Newton’s second law of motion $F = ma$. Acceleration is the derivative of velocity $\dot{y}$, which is, in turn, the derivative of the position y. Thus $a = \ddot{y}$, and we can write the differential equation that models the motion of the car as

$$F = m\ddot{y} \Leftrightarrow \ddot{y} = \frac{F}{m} \tag{1.1}$$

For a Volkswagen Caddy Van (diesel, 2-L engine) the mass, without load and including the driver, is approximately 1500 kg. Assume that the stop light changes to red when the car

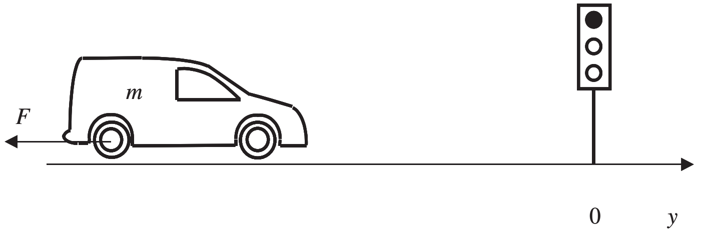

Figure 1.2: Stopping a car. Position y is positive towards the right, with zero at the stop light. The brakes act with a negative force F on the mass m.

is 25 m (82 ft) away at a speed of 10 m/s (36 km/h or 22.7 mph). We have thus identified the following constants:

$$m = 1500$$
$$y(0) = -25$$
$$\dot{y}(0) = 10$$

Once the speed is zero, the car will not move anymore. The variable F is thus negative or zero, since the brake is our only means of control. According to specifications, the distance at which the brakes have to be applied when the car is at a speed of from 80 km/h (49.7 mph.) to bring the speed to zero is $y = 27.3$ m. As all the kinetic energy is converted to work, we have, on the average,

$$\frac{1}{2}m\dot{y}^2 = Fy$$

and thus

$$\begin{aligned} F &= \frac{1}{y}\frac{1}{2}m\dot{y}^2 \\ &= \frac{1}{27.3}\frac{1}{2}1500\left(\frac{80\,000}{3600}\right)^2 \\ &\approx 13\,600 \end{aligned}$$

We shall therefore assume that the automatic brake system limits the magnitude of the brake force to 13 600 N (newton). The control signal is thus subject to the constraints

$$-13\,600 \le F \le 0$$

We can now simulate the system, and the objective is to study the behaviour related to various control strategies.

Step 1: Design a PID controller

Our first choice is to model the driver as a proportional controller,

$$F = K_p e \tag{1.2}$$

where K_p is the proportional gain, that can be adjusted. To interpret, the driver presses the brake pedal hard when the distance is large, relaxes the pressure as the distance decreases, and finally releases the pedal when arriving at the stop light. The error $e \geq 0$ is the position error between the stop line Ref and the current position $y \leq 0$,

$$e = Ref - y \tag{1.3}$$

The force F arises not from the engine, but from an opposite friction force in the brakes. Therefore K_p must be negative. The closed-loop system equations are obtained by inserting Equation (1.2) into Equation (1.1):

$$\ddot{y} = \frac{K_p e}{m} = \frac{K_p}{m} Ref - \frac{K_p}{m} y = -\frac{K_p}{m} y \tag{1.4}$$

since $Ref = 0$.

It turns out that only a particular controller setting ($K_p = -240$) will work such that the car stops at the stop light after about 10 s. A smaller magnitude of K_p stops the car after the light (overshoot), and a larger magnitude of K_p stops the car short of the light. The solution to Equation (1.4) shows why:

$$y(t) = \left(-\frac{5}{\sqrt{\frac{|K_p|}{m}}} - \frac{25}{2} \right) \exp^{-\sqrt{\frac{|K_p|}{m}} t} + \left(\frac{5}{\sqrt{\frac{|K_p|}{m}}} - \frac{25}{2} \right) \exp^{\sqrt{\frac{|K_p|}{m}} t}$$

Setting $K_p = -240$ suppresses the second exponential term, but if K_p is slightly different, the second exponential will diverge to plus or minus infinity.

Because of the long stopping time, and the highly sensitive solution, we conclude that proportional control is unrealistic.

Our second choice is to model the driver as a proportional-derivative (PD) controller, since it is likely that the driver takes the velocity into account. Figure 1.3 shows a Simulink (trademark of The MathWorks, Inc.) model, which includes the constraint on the brake force, and the model is therefore nonlinear. The controller is

$$F = K_p \left(e + T_d \dot{e} \right) \tag{1.5}$$

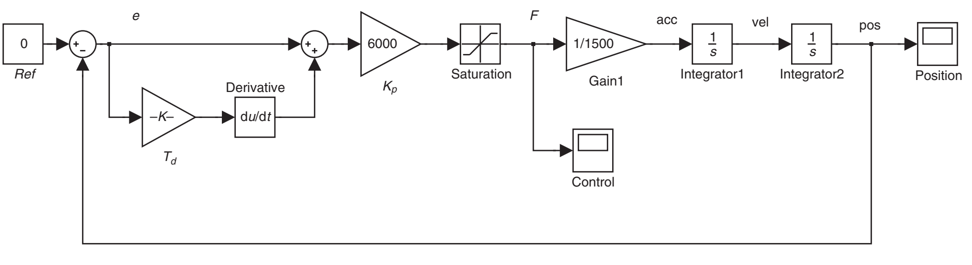

Figure 1.3: Simulink block diagram. A PD controller brakes the car from initial conditions $y(0) = -25$, $\dot{y}(0) = 10$. (figcarpd.mdl)

where T_d is the derivative gain, which can be adjusted. To interpret, the derivative action calls for extra brake force when the velocity is high. The closed-loop system equations are obtained by inserting Equation (1.5) into Equation (1.1):

$$\ddot{y} = \frac{K_p\,(e + T_d\dot{e})}{m} = -\frac{K_p T_d}{m}\dot{y} - \frac{K_p}{m}y \tag{1.6}$$

There will be a steady state solution, since in steady state the system is at rest, that is, $\ddot{y} = \dot{y} = 0$, and insertion yields the solution $y = 0$; this is in accordance with the problem specification. The transfer function of the closed-loop system is

$$\frac{y(s)}{Ref} = \frac{\frac{K_p}{m}}{s^2 + \frac{K_p}{m}T_d s + \frac{K_p}{m}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1.7}$$

The last expression is a general second-order system with natural frequency ω_n – the frequency of oscillation of the system without damping – and damping ratio ζ. It is useful, because we are looking for a solution without overshoot, which is as fast as possible; this corresponds to the case $\zeta = 1$, which yields a critically damped response. From Equation (1.7) we derive

$$\zeta = \frac{1}{2}\sqrt{\frac{K_p}{m}}T_d$$

Applying $\zeta = 1$ gives us an optimal tuning relationship

$$T_d = \frac{2}{\sqrt{\frac{K_p}{m}}}$$

It ensures that the response has no overshoot and there is a horizontal tangent at $y = 0$; consequently the velocity will be zero when the car arrives at the stop light.

Figure 1.4 shows the response with

$$K_p = 6000$$
$$T_d = 1$$

The figure also shows a fuzzy controller response for comparison, but we shall refer to that later. During the first 1.5 s, the control signal is zero, because the proportional action is positive and larger in magnitude than the derivative action, which is negative. Since the resultant action is positive, the saturation limits the signal to zero (the car moves owing to the initial velocity of 10 m/s). At time $t = 1.5$, the derivative action takes over and starts to brake the car. To interpret, the controller waits 1.5 s until it kicks in, applies less than maximum force about half a second later, and then relaxes the brake gently. It takes about 5 s to stop the car.

Step 2: Replace it with a linear fuzzy controller

A fuzzy controller consists of *if–then* rules describing the action to be taken in various situations. We will consider the situations where the distance to the stop light is long or

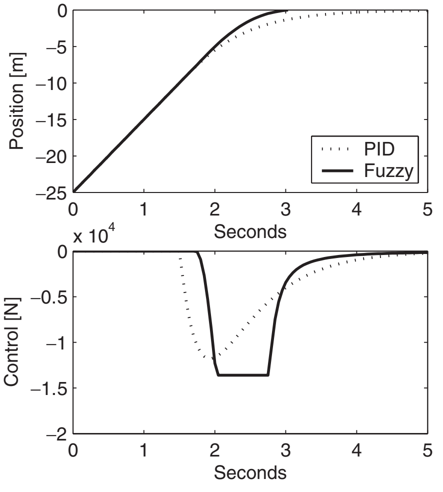

Figure 1.4: Stopping a car. Comparison between a PID controller and a fuzzy controller. (figstopcar.m)

short and situations where the car is approaching fast or slowly. A complete rule base of all possible combinations contains four rules:

$$\text{if the distance is long and the approach is fast, then brake zero;} \tag{1.8}$$

$$\text{if the distance is long and the approach is slow, then brake zero;} \tag{1.9}$$

$$\text{if the distance is short and the approach is fast, then brake hard;} \tag{1.10}$$

$$\text{if the distance is short and the approach is slow, then brake zero.} \tag{1.11}$$

The linguistic terms must be specified precisely for a computer to execute the rules. Figure 1.5 shows how to implement 'long', as in 'distance is long', as a fuzzy *membership function*, shaped like the letter 's'. The x-axis is the *universe,* the interval [0,100] % of the full range of 25 m. The y-axis is the *membership grade*, that is, how compatible a distance measurement is with our perception of the term 'long'. For instance, a distance of 25 m (100 %) has membership 1, because the distance is definitely long, while half that distance is long to a degree of just 0.5. Note that the x-axis corresponds to the previously defined error e, scaled onto a standard range of percentages relative to the maximum distance.

The term '*fast*', as in 'approach is fast', is another membership function. The x-axis is again percentages of the full range (10 m/s), but the numbers are negative to emphasize that the speed is decreasing rather than increasing. The x-axis corresponds to the previously defined time derivative $\dot{e}$ scaled onto the universe; when the speed decreases, $\dot{e}$ is negative. The -100 % corresponds to the maximum speed of 10 m/s.

Similarly, the membership function for 'short' is just a reflection of the membership function 'long', and the membership function 'slow' is a reflection of 'fast'.

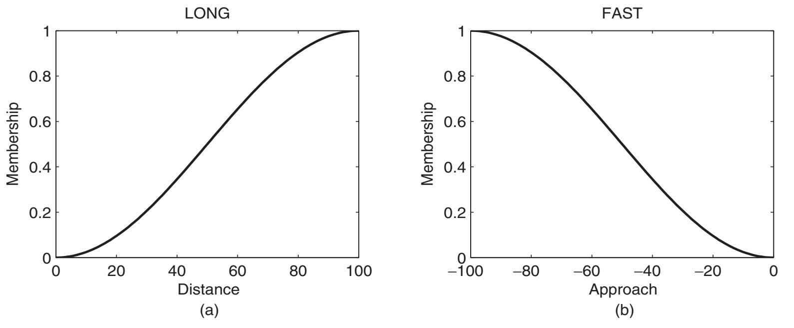

Figure 1.5: Fuzzy membership functions. The (a) specifies a LONG distance, and the (b) specifies a FAST approach. The curves correspond to the first rule (1.8). (figmfcar.m)

Turning to the *then*-side of the rules, the term '*zero*' is brake force 0, a constant. The term '*hard*' is the full brake force of -100 %, which is also a constant.

The following chapters will show how to design a *linear* fuzzy controller, with a performance that is exactly the same as the PD controller in the previous step. It is a design aid, because the PD controller, with its tuning, settles many design choices for the fuzzy controller. One requirement is that the membership functions should be linear.

At the end of this step, we have a fuzzy controller (not shown), with a response exactly as the dotted PD response in Figure 1.4.

Step 3: Make it nonlinear

Stepping into the nonlinear domain usually entails a trial and error design procedure, but the following chapters provide methods such that at least some *analysis* is possible.

The nonlinear fuzzy controller uses the nonlinear membership functions in Figure 1.5. The response is close to the PD controller response, and is therefore not shown.

Step 4: Fine-tune it

After adjusting one tuning factor (input gain on the error, GE, increased from 10 to 13), the response is as in Figure 1.4. The response is better than in PD control. The lower plot with the control signals shows what happens: The fuzzy controller waits longer before it kicks in, and then it uses all the available brake force. Thereafter it releases the brake quicker than the PD controller.

The example shows that good knowledge of the plant to be controlled is beneficial; to analyse stability, a mathematical model is even necessary. But more importantly, the PD controller design step gave us a tuning (K_p and T_d) that we could transfer to the fuzzy controller. Thereby, the PD controller constitutes a reference for the assessment of the performance of the fuzzy controller.

1.5 Summary

It is quite difficult to design a fuzzy controller, because it is in general nonlinear, and nonlinear systems are more or less unpredictable. Instead we propose to stay as long as possible in the linear domain, reflected in the proposed design procedure.

The idea is to start from PID control and design a linear fuzzy controller that is equivalent to the pre-designed PID controller. At this point, all the results from linear control theory can be applied, including tuning methods and stability calculations. In the next phase, the fuzzy controller is made nonlinear.

The design procedure has limited scope in the sense that it requires a PID controller be built to control the plant. As a compensation for the limited scope, the design procedure provides reliability: it guarantees that the fuzzy controller performs at least as well as its pre-designed PID controller. There is a possibility, but no guarantee, that it will perform better.

Some of the following chapters provide tools for analysing the nonlinear fuzzy controller, in particular, phase plane analysis and describing functions. Yet, trial and error is still a characteristic of the design procedure.

1.6 Notes and references

In the mid-1960s, Lotfi A. Zadeh (born in 1921 in Azerbaijan) of the University of California at Berkeley, USA, invented the theory of fuzzy sets. He argued that, more often than not, the classes of objects encountered in the real physical world have imprecisely defined criteria for membership (Zadeh 1965). For example, the 'class of numbers that are much greater than 1', or the 'class of tall human beings' have ill-defined boundaries. Yet, such imprecisely defined classes play an important role in human reasoning and communication.

Ebrahim (Abe) H. Mamdani, a control engineer at Queen Mary College in London (now Emeritus Professor at Imperial College), was attempting to develop an adaptive system that could learn to control an industrial process (Figure 1.6). He used a steam engine as a laboratory model, and with his colleagues set up a program that would teach the computer to control the steam engine by monitoring a human operator. At this point, Mamdani's

Figure 1.6: E.H. Mamdani

research student, Seto Assilian, tried to apply fuzzy logic. He created a set of simple rules in fuzzy terms, and Mamdani and Assilian then studied ways to use fuzzy rules of thumb directly in automating process controls. A few years later, Mamdani and Procyk managed to develop a linguistic self-organizing controller (Procyk and Mamdani 1979). It was an adaptive controller that was able to learn how to control a wide variety of processes, nonlinear and multi-variable, in a relatively short time. It was called '*self-organizing*' because at that point in time the meaning of the words 'adaptive' and 'learning' had not yet been agreed upon.

The work of the pioneers led to a growing literature in fuzzy control and wide-ranging applications, as Table 1.1 illustrates.

In Japan, Michio Sugeno (1989) developed a self-learning fuzzy controller. Twenty control rules determined the motion of a model car. Each rule recommends a specific change in direction, based on the car's distance from the walls of a corridor. The controller drives the car through angled corridors, after a learning session where a 'driving instructor' pulls it through the route a few times. Self-learning controllers that derive their own rules automatically are interesting because they could reduce the effort needed for translating human expertise into a rule base.

The first industrial application was in 1978, where a fuzzy controller was operating in closed loop on a rotary cement kiln in Denmark. Fuzzy control then became a commercial product of the Danish cement company F.L. Smidth & Co. The fuzzy control research program in Denmark was initiated in 1974 (Larsen 1981).

Table 1.1: Milestones in early fuzzy history.

Year	Event	Reference
1965	First article on fuzzy sets	Zadeh (1965)
1972	A rationale for fuzzy control	Zadeh (1972)
1973	Linguistic approach	Zadeh (1973)
1974	Fuzzy-logic controller	Assilian and Mamdani (1974)
1976	Warm-water plant	Kickert and van Nauta Lemke (1976)
1977	Table-based controller	Mamdani (1977)
1977	Heat exchanger	Østergaard (1977)
1977	Self-organizing controller	Procyk and Mamdani (1979)
1980	Fuzzy conditional inference	Fukami *et al.* (1980)
1980	Cement kiln controller	Holmblad and Østergaard (1982)
1983	Train operation	Yasunobu *et al.* (1983)
1984	Parking control of a model car	Sugeno *et al.* (1989)
1985	Fuzzy chip	Togai and Watanabe (1985)
1986	Fuzzy controller hardware system	Yamakawa and Miki (1986)
1987	Sendai subway in operation	Yasunobu *et al.* (1983)
1989	Fuzzy home appliances sold in Japan	
1989	The LIFE project is started in Japan	
1990	Rule learning by neural nets	Kosko (1992)
1990	Hierarchical controller	Østergaard (1990), (1996)

The Laboratory for International Fuzzy Engineering (*LIFE*), Yokohama, was set up by the Japanese Ministry of International Trade and Industry in 1989. It had a 6-year budget of 5 billion yen and a research staff of around 30. LIFE conducted basic research with universities and member Japanese companies and subsidiaries of U.S. and European companies, including Matsushita, Hitachi, Omron, and VW, about 50 companies in all. The research program was trimmed to five major projects: image understanding, fuzzy associative memory, fuzzy computing, intelligent interface, and the intelligent robot. They were all carried out with two themes in view: a navigation system for the blind and home computing.

A European network of excellence called *ERUDIT*[1] was initiated in 1995 with support from the European Commission. ERUDIT, which lasted 6 years, was an open network for uncertainty modelling and fuzzy technology, aimed at putting European industry at the leading edge. The network was followed by another network EUNITE[2] with a broader scope: smart adaptive systems. And this was in turn followed by a coordinated action NISIS[3] with an even broader scope: nature-inspired systems.

For further information

Beginners may start with two articles in Institute of Electrical and Electronics Engineers (IEEE) Spectrum (Zadeh 1984, Self 1990) and then move on to the more advanced textbook by Zimmermann (1993); half of it is dedicated to fuzzy set theory and the other half to applications.

The terms ‘rule base’ and ‘inference engine’ are loans from the field of expert systems, and Lee (1990) uses these to give a wide survey of the whole area of fuzzy control. The article lists 150 references. The book by Ross (1995) is oriented towards applications in engineering and technology with many calculated examples.

A major reference on fuzzy control is the book by Driankov *et al.* (1996). It is explicitly targeted at the control engineering community, in particular, engineers in industry, and university students. Chapter 3 gives more specific references related to fuzzy control.

Industrial applications are described in a special issue of the journal *Fuzzy Sets and Systems*, for instance, the fuzzy car by Sugeno *et al.* (1989) and an arc welding robot by Murakami *et al.* (1989). There are more early applications in the classical book by Sugeno (1985). Ten years later, Constantin von Altrock (1995) described more than 30 case studies from companies that employed fuzzy and neuro-fuzzy methods. The FL Smidth controller is described in detail in Holmblad and Østergaard (1982).

There are more than 10 journals related to fuzzy sets. Two of the major journals are *Fuzzy Sets and Systems* and *International Journal of Approximate Reasoning*, both published by Elsevier, and a third one is *Journal of Intelligent and Fuzzy Systems*, published by IOS Press, Netherlands. The Institute of Electrical and Electronics Engineers, *IEEE,* started a journal in 1992 called *IEEE Transactions on Fuzzy Systems*. The *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems* is published four times per year by World Scientific Publishing Co. It is a forum for research on imprecise, vague, uncertain, and incomplete knowledge. Other journals that occasionally have fuzzy control articles are

[1] http://www.erudit.de
[2] http://www.eunite.org
[3] http://www.nisis.de

Automatica, the control section of *IEE Proceedings*, *IEEE Transactions on Systems Man and Cybernetics*, *IEEE Transactions on Computers, Control Engineering Practice,* and the *International Journal of Man-Machine Studies*.

There is an active newsgroup called `comp.ai.fuzzy`[4]. It supplies useful news, conference announcements, and discussions.

There are two major professional organizations. The International Fuzzy Systems Association (IFSA) is a worldwide organization dedicated to fuzzy sets. IFSA publishes the *International Journal of Fuzzy Sets and Systems* (24 issues per year), holds international conferences, establishes chapters, and sponsors activities. The other organization is the North American Fuzzy Information Processing Society, NAFIPS[5], with roughly the same purpose.

[4]http://groups.google.com/group/comp.ai.fuzzy
[5]http://nafips.ece.ualberta.ca/