

Introduction to fuzzy control

Educational Material

Author(s):

Geering, Hans P.

Publication date:

1998

Permanent link:

<https://doi.org/10.3929/ethz-a-004953512>

Rights / license:

In Copyright - Non-Commercial Use Permitted

Introduction to Fuzzy Control

Hans P. Geering

Abstract

In this report, some of the basic mathematical definitions and rules of fuzzy system theory are described inasmuch as they are relevant for fuzzy control. Two examples are covered in detail, viz., a fuzzy closed-loop halting control scheme for the forward motion of a mobile robot in an automatic factory and a dog chasing a cat using fuzzy control.

Issues of computational efficiency are discussed. And some recommendations to potential designers of fuzzy controllers are summarized.

After studying this report, the reader should be in a position to design simple fuzzy controllers and simulate the behaviour of the resulting fuzzy control system on a general purpose digital computer.

IMRT Press

© Measurement and Control Laboratory
Swiss Federal Institute of Technology (ETH)
ETH-Zentrum
CH-8092 Zurich, Switzerland

3rd ed., September, 1998

Contents

1	Fuzzy Sets	1
2	Fuzzification	2
3	Fuzzy Logic	2
4	Fuzzy Variables	3
4.1	Introducing Fuzzy Variables	3
4.2	Fuzzification Revised	4
4.3	Fuzzy Vectors	4
5	Fuzzy Rules	5
5.1	Fuzzy SISO-Rule	5
5.2	Fuzzy AND-Rules	5
5.3	Other Fuzzy Rules	6
6	Fuzzy Associative Memory	6
7	Defuzzification	7
8	Fuzzy Control Systems	8
8.1	Structure of a Fuzzy Control System	8
8.2	Example 1: Closed-loop halting control	10
8.3	Example 2: Dog chasing cat	15
9	Computational Issues	18
9.1	Efficient Defuzzification	18
9.2	Derivatives of the Control Function	20
9.3	Observations and Suggestions	21
	References	22

1 Fuzzy Sets

Definition: A fuzzy set s is an ordered pair (X, f) , where X is a vector space (usually the real line R) and f is a set membership function mapping X onto the interval $[0, 1]$ of the real line R , i.e., $f : X \rightarrow [0, 1]$.

In a fuzzy control problem, X is the signal space of a signal or a vector signal, respectively.

A set $S \subset X$ is associated with the fuzzy set $s = (X, f)$ in a natural way: $S = \text{cl}\{x \in X \mid f(x) > 0\}$ is the closure of the set in X where f attains positive values.

Notice that the set membership function f is normalized in the sense that the value $f(x) = 1$ is attained for at least one element $x \in S \subset X$. However, this normalization has mainly been introduced for practical and intuitive reasons. Mathematically speaking, this normalization is dispensable.

Usually, a fuzzy set is a constant construct, i.e., a time-invariant part of a fuzzy control system.

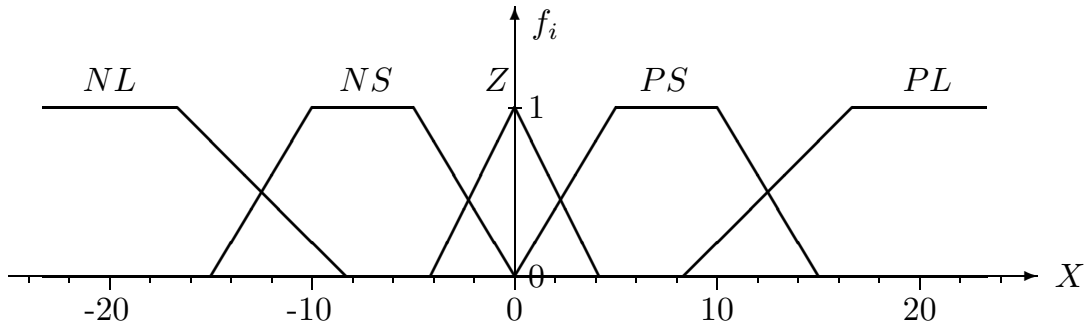


Figure 1: Fuzzy sets NL (negative large), NS (negative small), Z (zero), PS (positive small), and PL (positive large) covering the real line $X = R$.

The following two scalar characteristics of a fuzzy set will be useful later.

Definition: The weight w and the centroid c of a fuzzy set $s = (X, f)$ are defined as follows:

$$w = \int f(x) dx$$

and

$$c = \frac{\int x f(x) dx}{\int f(x) dx}$$

where all of the integrals are taken over the signal space X .

2 Fuzzification

Consider a signal space X covered by several fuzzy sets s_i , $i = 1, \dots, k$. The fuzzy question is: Given a vector $x \in X$, to which of the fuzzy sets s_i does x belong or, in which of the sets S_i associated with the fuzzy sets s_i does x lie?

In mathematical set theory, the answer for each of the sets S_i is a binary one. In fuzzy set theory, set membership is “by degree”.

Definition: Consider a fuzzy set $s = (X, f)$. An arbitrary element $x \in X$ belongs to the fuzzy set s with degree $d = f(x)$.

Hence, the answer to the fuzzy question is: x belongs to *each* of the fuzzy sets s_i to some degree, viz., to degrees $d_i = f_i(x)$, $i = 1, \dots, k$.

Examples: Consider the fuzzy sets NL , NS , Z , PS , and PL defined on the signal space $X = R$ which are displayed in Figure 1.

a) The element $x = -20$ is “negative large” to degree 1 and “negative small”, “zero”, “positive small”, and “positive large” to degrees 0.

b) The element $x = 2$ is “negative large” and “negative small” to degrees 0, “zero” to degree 0.52, “positive small” to degree 0.4, and “positive large” to degree 0.

3 Fuzzy Logic

Fuzzy logic defines the rules governing the operators intersection and union of fuzzy sets.

Consider two fuzzy sets $s_1 = (X, f_1)$ and $s_2 = (X, f_2)$ defined on the *same* signal space X and their associated sets $S_1 \subset X$ and $S_2 \subset X$, respectively.

Definition: An arbitrary element $x \in X$ belongs to the union $s_1 \cup s_2$ of the two fuzzy sets s_1 and s_2 with degree $d = \max(f_1(x), f_2(x))$.

Definition: An arbitrary element $x \in X$ belongs to the intersection $s_1 \cap s_2$ of the two fuzzy sets s_1 and s_2 with degree $d = \min(f_1(x), f_2(x))$.

Consequently, the union operator and the intersection operator yield the fuzzy sets $s_1 \cup s_2 = (X, \max(f_1, f_2))$ and $s_1 \cap s_2 = (X, \min(f_1, f_2))$, respectively. Notice that the intersection $s_1 \cap s_2$ is a degenerated fuzzy set in the sense that its set membership function $\min(f_1, f_2)$ does not map onto the interval $[0, 1]$ as requested by the definition of a fuzzy set. This detail is not pursued any further here because in fuzzy control, all calculations are done with fuzzy variables rather than with fuzzy sets.

4 Fuzzy Variables

4.1 Introducing Fuzzy Variables

Definition: A fuzzy variable v is an ordered pair (s, d) where s is a fuzzy set and $d \in [0, 1]$ a real bounded variable.

Fuzzy variables arise in the fuzzification operation in a natural way: For the variable $x \in X$, the real variable d is the degree of membership in the fuzzy set s . (Cf. Section 4.2)

In another interpretation of a fuzzy variable, the real variable d “modulates” the fuzzy set s : The scalar d and the set membership function $f : X \rightarrow [0, 1]$ of the fuzzy set s define a new function $g : X \rightarrow [0, 1]$. There are two modulation schemes:

- a) “linear modulation”: $g(x) = d \cdot f(x)$
- b) “modulation by clipping”: $g(x) = \min(f(x), d)$.

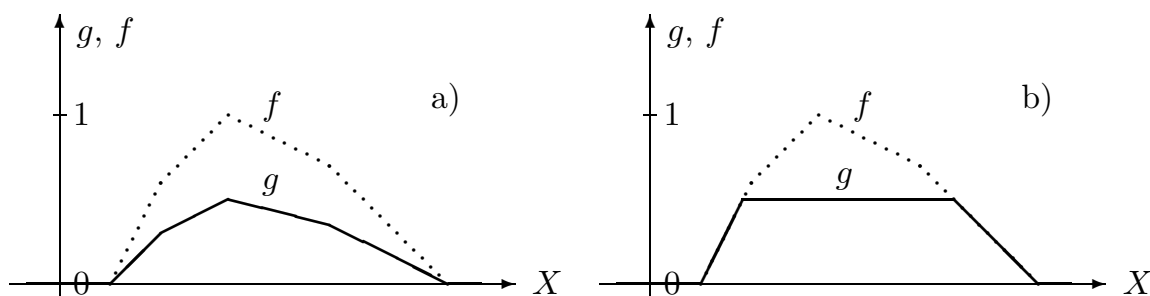


Figure 2: Example of a fuzzy variable: fuzzy set (X, f) modulated by $d = 0.5$: a) linear modulation, b) modulation by clipping.

The author prefers the linear modulation scheme because the function g obtained by linear modulation typically contains more detailed information about the structure of the fuzzy variable.

In Section 1, the weight w and the centroid c of a fuzzy set $s = (X, f)$ have been defined. Obviously, the weight and the centroid of a fuzzy variable v can be defined in an analogous way by replacing the set membership function f by the modulated function g in these formulae.

Notice that the linear modulation scheme results in a linear reduction of the weight of the fuzzy variable, $w_v = d \cdot w_s$, while the centroid remains unchanged, $c_v \equiv c_s$ for all $d \in (0, 1]$.

For calculations with a fuzzy variable, it is more practical to use the “modulated” function g than to keep the scalar d and the set membership function f of the underlying fuzzy set s apart. Furthermore, the restriction $g(x) \leq 1$ for all $x \in X$ can be dropped. This is practical when sums of fuzzy variables are calculated.

4.2 Fuzzification Revised

Consider a signal space X covered by the N fuzzy sets s_1, \dots, s_N . An arbitrary element $x \in X$ belongs to the fuzzy sets s_1, s_2, \dots, s_N to degrees $d_1 = f_1(x)$, $d_2 = f_2(x)$, \dots , and $d_N = f_N(x)$, respectively.

Using fuzzy variables leads to the following definition of the fuzzification operation.

Definition: The fuzzification operator \mathcal{F} maps an element $x \in X$ to the set of fuzzy variables $\{(s_1, f_1(x)), (s_2, f_2(x)), \dots, (s_N, f_N(x))\}$.

Examples: Reconsider the examples a) and b) of Section 2. With the above definition we can rewrite the results succinctly in the following way:

- a) $\mathcal{F} : -20 \mapsto \{(NL, 1), (NS, 0), (Z, 0), (PS, 0), (PL, 0)\}$ and
- b) $\mathcal{F} : 2 \mapsto \{(NL, 0), (NS, 0), (Z, 0.52), (PS, 0.4), (PL, 0)\}$.

4.3 Fuzzy Vectors

As the example in Section 4.2 shows, introducing vector notation in the range space of the fuzzification operator \mathcal{F} is efficient.

Again, consider a signal space X covered by the N fuzzy sets s_1, \dots, s_N . The fuzzification $\mathcal{F}(x)$ of an arbitrary element $x \in X$ can be represented by an N -vector in several equivalent ways:

$$\mathcal{F}(x) = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} (x) = \begin{bmatrix} (s_1, f_1(x)) \\ (s_2, f_2(x)) \\ \vdots \\ (s_N, f_N(x)) \end{bmatrix} \cong \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_N(x) \end{bmatrix} .$$

The relation operator “ \cong ” points out the fact that, in the last vector, the fuzzy sets s_i are not explicitly noted down but are implied by the indices i .

On the other side, every fuzzy n -vector can be represented by the corresponding modulated set membership functions g_i :

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} (s_1, d_1) \\ (s_2, d_2) \\ \vdots \\ (s_n, d_n) \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix} \cong \begin{bmatrix} (w_{g_1}, c_{g_1}) \\ (w_{g_2}, c_{g_2}) \\ \vdots \\ (w_{g_n}, c_{g_n}) \end{bmatrix} .$$

Here, the relation operator “ \cong ” points out the fact that the weight w_{g_i} and the centroid c_{g_i} do not completely characterize the fuzzy variable v_i .

This representation is useful at the outputs of the fuzzy rules describing a fuzzy controller.

5 Fuzzy Rules

Fuzzy rules are used in fuzzy control in order to define the map from the fuzzified input signals (error signals, measured signals, or command signals) of the fuzzy controller to its fuzzy output signals (control signals).

5.1 Fuzzy SISO-Rule

Consider a fuzzy set $s_e = (E, f_e)$ defined on the signal space E where the error signal e “lives” and a fuzzy set $s_u = (U, f_u)$ defined on the signal space U where the control signal u “lives”. (Usually, $E = R$ and $U = R$, hence the designation “SISO-rule”.)

Definition: The SISO-rule mapping the fuzzy input variable $v_e = (s_e, d_e)$ to the fuzzy output variable $v_u = (s_u, d_u)$ (of the fuzzy controller) is defined by $v_u = (s_u, d_e)$.

In the jargon of control engineering, this definition should be read as follows: If the value $e(t)$ of the error signal belongs to the fuzzy set s_e to degree d_e then the fuzzy set s_u of the control signal is fired to degree $d_u = d_e$, i.e., modulated by $d_u = d_e$.

In shorthand notation, the fuzzy SISO-rule is denoted by $s_e \Rightarrow s_u$, where the degree of firing $d_u = d_e$ is implied.

The value $u(t)$ of the control signal is obtained later by “defuzzification” after all of the fuzzy rules pertaining to the control signal have been processed.

5.2 Fuzzy AND-Rules

Consider two fuzzy sets $s_{e1} = (E_1, f_{e1})$ and $s_{e2} = (E_2, f_{e2})$ defined on the signal spaces E_1 and E_2 , respectively, where the error signals e_1 and e_2 “live” and a fuzzy set $s_u = (U, f_u)$ defined on the signal space U where the control signal u “lives”.

Definition: The AND-rule mapping the fuzzy input variables $v_{e1} = (s_{e1}, d_{e1})$ and $v_{e2} = (s_{e2}, d_{e2})$ to the fuzzy output variable $v_u = (s_u, d_u)$ is defined by $v_u = (s_u, \min(d_{e1}, d_{e2}))$.

In the jargon of control engineering, this definition should be read as follows: If the value $e_1(t)$ of the first error signal belongs to the fuzzy set s_{e1} to degree d_{e1} and the value $e_2(t)$ of the second error signal belongs to the fuzzy set s_{e2} to degree d_{e2} then the fuzzy set s_u of the control signal is fired to the smaller of the two degrees, i.e., $d_u = \min(d_{e1}, d_{e2})$.

In shorthand notation, the fuzzy AND-rule is denoted by $s_{e1} \cap s_{e2} \Rightarrow s_u$, where the degree of firing $d_u = \min(d_{e1}, d_{e2})$ is implied.

The value $u(t)$ of the control signal is obtained later by “defuzzification” after all of the fuzzy rules pertaining to the control signal have been processed.

It should be obvious how the definition of the fuzzy AND-rule can be extended to three or more fuzzy input variables.

5.3 Other Fuzzy Rules

In analogy to the fuzzy AND-rules, fuzzy OR-rules or more complicated logical combinations for fuzzy rules could be defined.

The author prefers to use fuzzy AND-rules exclusively because OR-ing several AND-rules together typically results in a weaker contribution to the overall fuzzy output variable(s) and the corresponding defuzzified control variable(s).

Therefore, in the remainder of this report “fuzzy rule” stands for “fuzzy AND-rule” or its SISO special case “fuzzy SISO-rule”.

6 Fuzzy Associative Memory

For a fuzzy controller, the collection of all of its fuzzy rules is called the fuzzy associative memory.

For every control cycle, each of the fuzzy rules is evaluated. This can be done by massively parallel processing. The output of each fuzzy rule is a fuzzy variable.

The output of the fuzzy associative memory is equal to the (vector) sum of all these fuzzy variables:

In the case of a scalar control signal $u(t)$, the signal space is the real line, $U = R$. Summing the fuzzy variables involves calculating the sum $g_u = \sum_j g_j$ of their modulated functions g_j . — Notice that the fuzzy variable v_u at the output of the fuzzy associative memory is represented exclusively by the (“modulated”) function g_u . (I.e., this fuzzy variable has no directly underlying fuzzy set which is modulated by some degree d_u to yield the function g_u .)

In the case of a vector control signal $u(t) \in R^m$, typically, the signal space of each of the components $u_i(t)$ is the real line, i.e., $U_i = R$ for $i = 1, \dots, m$. Summing the fuzzy variables involves calculating the m sums $g_{u_i} = \sum_j g_{ij}$ of the modulated functions g_{ij} for each index i , $i = 1, \dots, m$.

Of course, the summing operator \sum takes the pointwise sum of its argument functions. As mentioned in Section 4.1, the sum $g(u)$ may exceed 1 for some values of the argument u . This poses no problem (cf. Section 7). (Clipping $g(u)$ to the maximal value 1 would be counterproductive because the centroid of g would be shifted.)

7 Defuzzification

Defuzzification is the process of assigning a representative value to a fuzzy variable. Consider a fuzzy variable v_u on the signal space $U = R$ which is represented by the modulated function g_u .

Definition: The defuzzification operator \mathcal{D} maps the fuzzy variable v_u to the centroid u of the modulated function g_u ,

$$u = \mathcal{D}\{v_u\} = \mathcal{D}\{g_u\} = \frac{\int \alpha g_u(\alpha) d\alpha}{\int g_u(\alpha) d\alpha},$$

where both of the integrals are calculated over the signal space $U = R$. The defuzzification operator \mathcal{D} is understood to accept an arbitrary representation of the fuzzy variable v_u as its argument.

Upon conclusion of the fuzzy control algorithm, precise values $u_1(t), \dots, u_m(t)$ must be assigned to the components of the control vector. However, the fuzzy associative memory yields m fuzzy variables $v_{u_i}(t)$ represented by their sum functions $g_{u_i}(\cdot, t) = \sum_j g_{ij}(\cdot, t)$. Defuzzifying yields the control signals

$$u_i(t) = \mathcal{D}\{v_{u_i}(t)\} = \frac{\int \beta g_{u_i}(\beta, t) d\beta}{\int g_{u_i}(\beta, t) d\beta} \quad i = 1, \dots, m$$

For the sake of simplicity, one-dimensional signal spaces $U_i = R$ for $i = 1, \dots, m$ have been discussed. It would be rather straightforward to use signal spaces of higher dimensions, in particular $U = R^m$. The defuzzification of the m control signals $u_i(t)$ would then obviously involve the corresponding multiple integrals over the signal space $U = R^m$. — In practice, there is no advantage in using such a generalization.

8 Fuzzy Control Systems

8.1 Structure of a Fuzzy Control System

Sections 1, 2, and 4–7 describe all of the elements and operations needed in a fuzzy controller. Figure 3 shows the block diagram of a fuzzy control system implementing both feedback control and feed-forward control. All of the signals

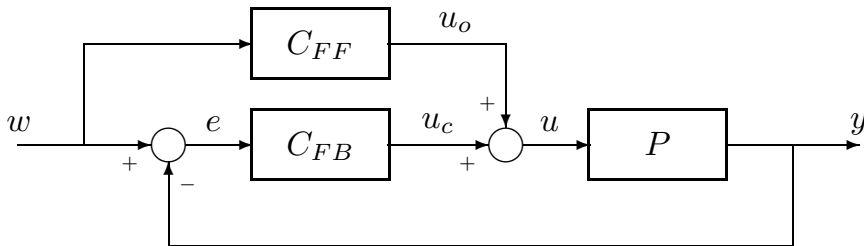


Figure 3: Fuzzy control system consisting of the plant P , the fuzzy feedback controller C_{FB} , and the fuzzy feed-forward controller C_{FF} .

in Figure 3 are precise (i.e., crisp or non-fuzzy) signals. The internal structures of the feed-forward controller and the feedback controller are identical.

Figure 4 depicts the major components of the fuzzy feedback controller C_{FB} , viz., the fuzzifier \mathcal{F} , the fuzzy associative memory FAM , and the defuzzifier \mathcal{D} .

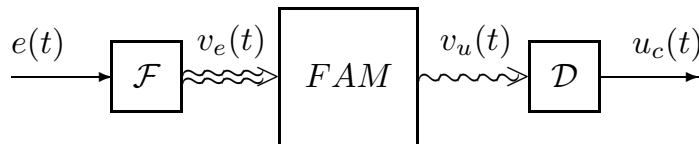


Figure 4: Fuzzy feedback controller.

The “wiggly” or “fuzzy” double arrow emphasizes the fact that even for a scalar error signal $e(t)$, the quantity $v_e(t) = \mathcal{F}\{e(t)\}$ is a fuzzy vector (cf. Sections 4.2 and 4.3). On the other hand, $v_u(t)$ is a fuzzy variable which is defuzzified to the scalar control signal $u_c(t)$.

Figure 5 shows a detailed block diagram of a fuzzy controller with two input signals and one output signal. The error signals e_1 and e_2 are fuzzified to the fuzzy N -vector v_{e_1} and the fuzzy M -vector v_{e_2} , respectively. The components of the fuzzy vectors, i.e., the $N + M$ fuzzy variables are shipped to the $N \cdot M$ fuzzy AND-rules. The fuzzy variables at the outputs of the AND-rules are summed up. This yields the fuzzy control variable v_{u_c} at the output of the fuzzy associative memory and the defuzzified control signal u_c . — Notice that, typically, one and the same fuzzy set of the control signal is fired by several of the fuzzy AND-rules.

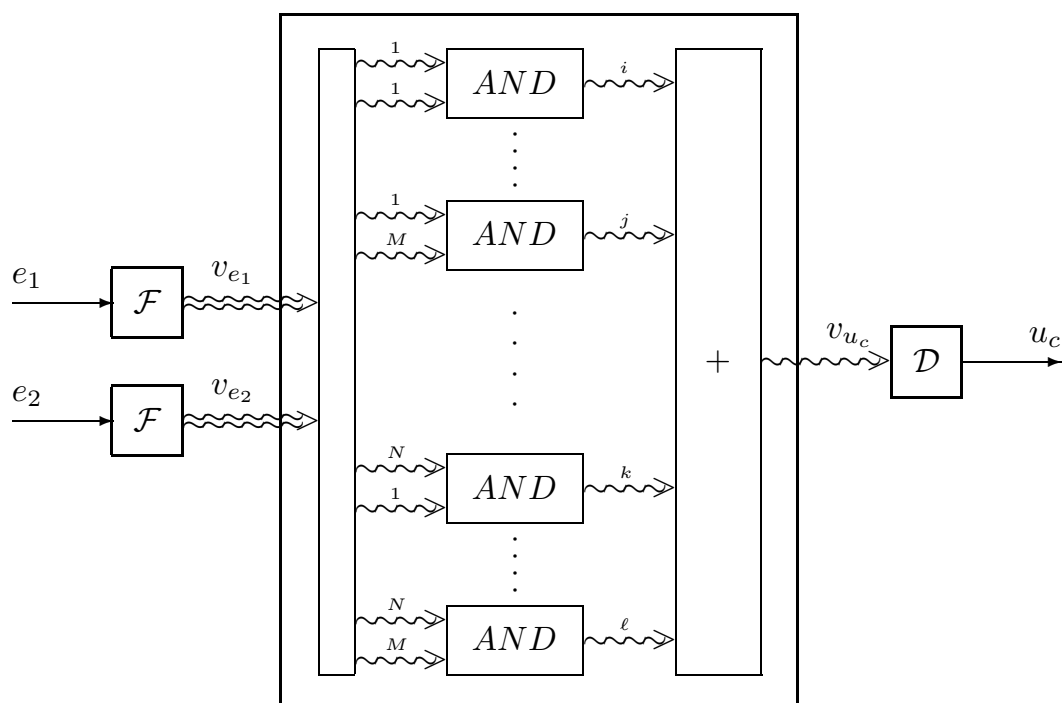


Figure 5: Detailed block diagram of a fuzzy feedback controller.

8.2 Example 1: Closed-loop halting control

In a fully automatic factory, parts, subassemblies, and the finished products are transported by several identical mobile robots. The navigation of the mobile robots about the factory is fully automated. In this example, the forward motion along the trajectories is considered only. Getting away from one station and cruising to the next station is under closed-loop speed control. The robot should stop at the next station within a very small tolerance for the position error. Therefore, closed-loop halting control must be implemented.

The problem of designing a fuzzy control scheme for halting control is investigated here.

The following information is available: The robot has a mass of anywhere between 150 and 450 kg, depending on the actual payload. The maximal cruising speed is 2 m/s. The positioning error at a station must be less than 1 mm. The maximal traction force for acceleration and deceleration is 500 N. The nominal deceleration in the approach to a station is $a = 1 \text{ m/s}^2$. There is no requirement for the smoothness of the acceleration. At every station, within a reach of 2.5 m, a precise measurement of the instantaneous distance to the stop is transmitted to the robot at a very high rate. Furthermore, the robot has a precise velocity sensor.

The following concept for a fuzzy halting controller is chosen:

The control law for the forward motion of the robot is switched from cruise control to fuzzy halting control as soon as the simultaneous measurements p and v of the position and the velocity, respectively, satisfy the inequality

$$|p| \leq \frac{v^2}{2a} .$$

At the switching time, a non-fuzzy signal generator for the setpoint variable $w_p(t)$ for the position, the setpoint variable $w_v(t)$ for the velocity, and the feed-forward force $u_o(t)$ is initialized as follows:

$$\begin{aligned} w_p(0) &= p \\ w_v(0) &= -\text{sign}(p)\sqrt{2a|p|} \\ u_o(0) &= \text{sign}(p)m_{\text{nom}}a = \pm 250 \text{ N} . \end{aligned}$$

While the robot is in the halting control mode, the signal generator furnishes the setpoint values $w_p(t)$ and $w_v(t)$ and the feed-forward control signal $u_o(t)$ according to the formulae

$$\begin{aligned} w_p(t) &= w_p(0) + w_v(0)t + \text{sign}(w_p(0))\frac{at^2}{2} \\ w_v(t) &= w_v(0) + \text{sign}(w_p(0))at \\ u_o(t) &\equiv u_o(0) , \end{aligned}$$

where t is the time elapsed since the initialization of the signal generator.

The error signals $e_p(t) = w_p(t) - p(t)$ and $e_v(t) = w_v(t) - v(t)$ are processed in a fuzzy feedback controller which produces the feedback force $u_c(t)$. The detailed block diagram of this fuzzy two-input one-output feedback controller corresponds to the one shown above in Figure 5. Finally, the total traction force is $u(t) = \text{sat}\{u_o(t) + u_c(t)\}$. (It is clipped to $+500$ N or -500 N, respectively, whenever the sum exceeds the limit of the available traction force.)

The fuzzy halting control law is deactivated as soon as the velocity vanishes for the first time, $v(t) = 0$. Simultaneously, the parking brakes are set.

The fuzzy sets covering the signal space $E_p = R$ where the position error $e_p(t)$ lives are shown in Figure 6. The fuzzy sets covering the signal space $E_v = R$ where the velocity error $e_v(t)$ lives are shown in Figure 7.

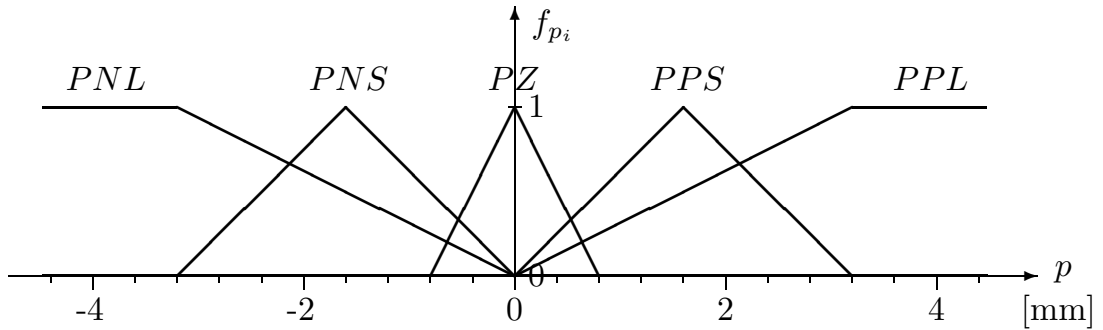


Figure 6: Fuzzy sets for the position error: *PNL* (negative large), *PNS* (negative small), *PZ* (zero), *PPS* (positive small), and *PPL* (positive large).

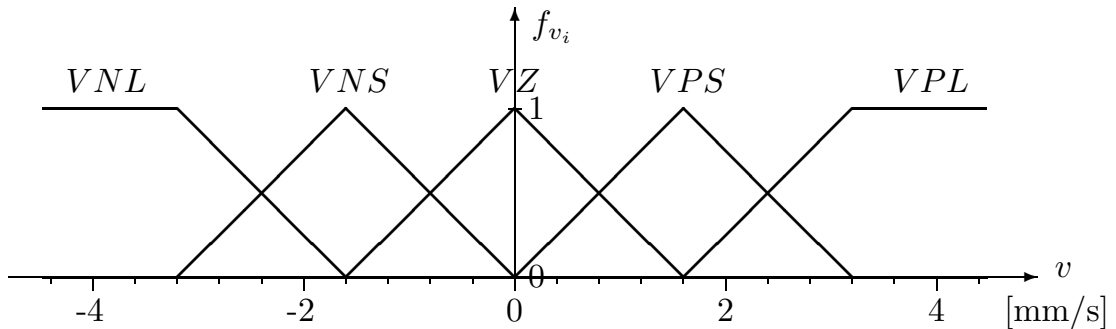


Figure 7: Fuzzy sets for the velocity error: *VNL* (negative large), *VNS* (negative small), *VZ* (zero), *VPS* (positive small), and *VPL* (positive large).

The fuzzy sets chosen for the feedback traction force $u_c(t)$ are depicted in Figure 8. Notice that the centroids of the fuzzy sets *UPX* and *UNX* are located at -500 N and $+500$ N, respectively, in order to give the fuzzy feedback controller full authority over the range $-500 \dots +500$ N. Furthermore, *no* fuzzy set *UZ* centered at 0 N is introduced in order to obtain “stiff” characteristics of the fuzzy feedback control law.

In Figure 9, the set of 20 fuzzy rules defining the fuzzy associative memory of the fuzzy feedback controller is displayed. The entries of this tabloid should

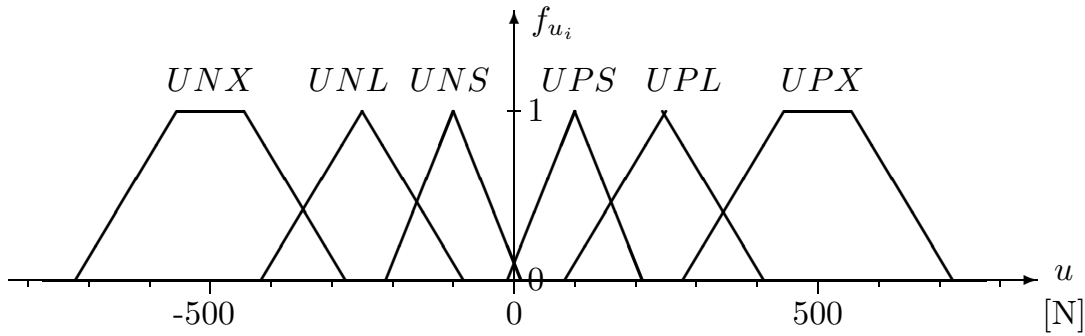


Figure 8: Fuzzy sets for the control of the traction force: UNX (negative extra large), UNL (negative large), UNS (negative small), UPS (positive small), UPL (positive large), and UPX (positive extra large).

be read as explained in the following example. Shorthand explanation: $PNL \cap VNL \Rightarrow UNX$. Longhand explanation: If $e_p(t)$ belongs to the fuzzy set PNL to degree d_1 and if $e_v(t)$ belongs to the fuzzy set VNL to degree d_2 then the fuzzy set UNX is fired to the smaller of the two degrees, i.e., to degree $d = \min(d_1, d_2)$.

v_v : velocity error	VPL		UPS	UPL	UPX	UPX
	VPS	UNS		UPS	UPL	UPX
	VZ	UNL	UNS		UPS	UPL
	VNS	UNX	UNL	UNS		UPS
	VNL	UNX	UNX	UNL	UNS	
		PNL	PNS	PZ	PPS	PPL
		v_p : position error				

Figure 9: Fuzzy associative memory for the fuzzy closed-loop servo controller containing twenty fuzzy rules.

In order to evaluate the effectiveness of the proposed fuzzy control scheme, the mobile robot is simulated. For the simulation, the “true” robot is modelled as follows:

$$\begin{aligned}\dot{p}(t) &= v(t) \\ p(0) &= -2 \text{ m} \\ \dot{v}(t) &= \frac{1}{m} \{u(t) - \text{sign}(v(t))\gamma mg\} \\ v(0) &= 2 \text{ m/s} ,\end{aligned}$$

where $m = 150 \dots 450$ kg is the true mass of the robot, $g = 9.81 \text{ m/s}^2$ the gravitational constant, and $\gamma = 0.01$ the coefficient of roll friction. For the simulations, digital control with a sampling and control rate of 500 Hz is assumed. This fairly high sampling rate is chosen in order to prevent mechanical resonances in the mobile robot.

In Figures 10 and 11, the trajectory of the robot is shown in the phase plane (p, v) in several scales for a true mass $m = 450$ kg and $m = 150$ kg, respectively. For the complete trajectory labelled “a)”, the units for p and v are m and m/s, respectively. For the increasingly enlarged final parts “b)”, “c)”, and “d)” of the trajectory, the units are dm and dm/s, cm and cm/s, and mm and mm/s, respectively.

As the Figures show, the heaviest robot ($m = 450$ kg) overshoots the station by less than 0.2 mm, whereas the lightest robot ($m = 150$ kg) stops less than 0.2 mm short of the station. Hence, the specifications are met.

This servo control example is deceptively simple because the plant under consideration essentially is a double integrator and because with a PD-controller or with the equivalent linear state feedback controller one cannot arrive at an unstable control system, provided the *signs* of the two control gains are chosen correctly. The only open question is whether the specifications for the precision of halting are met.

From the next example it can be inferred that asymptotic stability of a fuzzy control system is not necessarily obtained by choosing the fuzzy control scheme with straightforward commonsense logic. As a matter of fact, proving the asymptotic stability of a fuzzy control system (even of moderate complexity) can turn out to be very difficult.

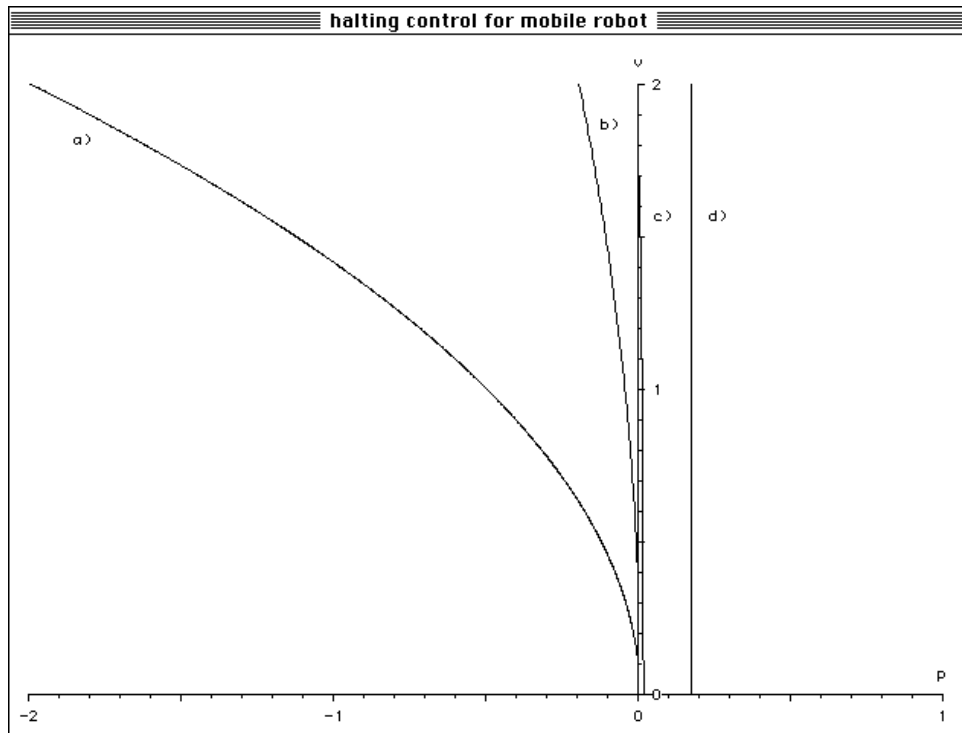


Figure 10: Trajectory of a robot with a mass of 450 kg in the phase plane (p, v) . Scales: a) m and m/s; b) dm and dm/s; c) cm and cm/s; d) mm and mm/s.

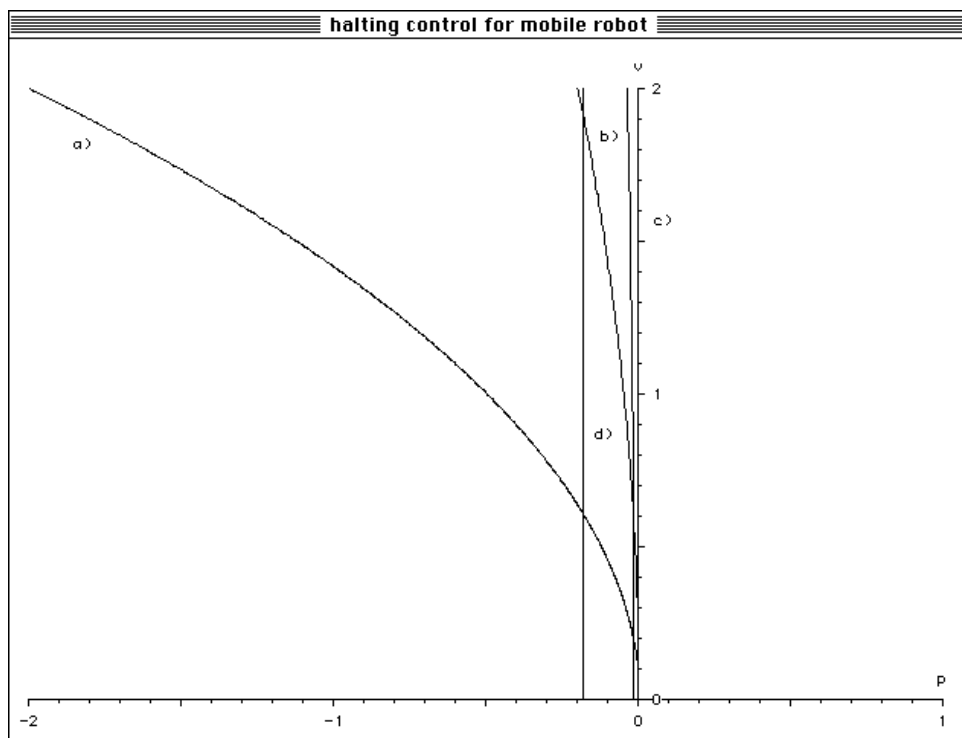


Figure 11: Trajectory of a robot with a mass of 150 kg in the phase plane (p, v) . Scales: a) m and m/s; b) dm and dm/s; c) cm and cm/s; d) mm and mm/s.

8.3 Example 2: Dog chasing cat

In this example, a dog chasing a cat is considered. Like most dogs, this dog is not smart enough to use proportional navigation. Rather, he tries to always run in the direction where he sees the cat. The dog would probably explain his scheme of navigation by the following qualitative rules for choosing the direction of his next leap: If the cat is straight ahead, I take the next leap in the same direction; if the cat is a little bit to the left (right), I turn a little bit to the left (right) for the next leap; if the cat is pretty much to the left (right), I turn quite a bit to the left (right) for the next leap.

Obviously, a fuzzy one-input one-output feedback controller for this scheme of navigation can be formulated with five fuzzy sets for the line of sight angle $\alpha(t_k)$, five fuzzy sets for the change of the dog's heading angle $\Delta\gamma(t_k)$ for the next leap, and five fuzzy SISO-rules.

Figure 12 shows the fuzzy sets covering the signal space $A = [-180^\circ, 180^\circ]$ where the line of sight angle $\alpha(t_k)$ lives. Figure 13 shows the fuzzy sets covering the signal space $U = R$ where the change of the dog's heading angle $\Delta\gamma(t_k)$ lives.

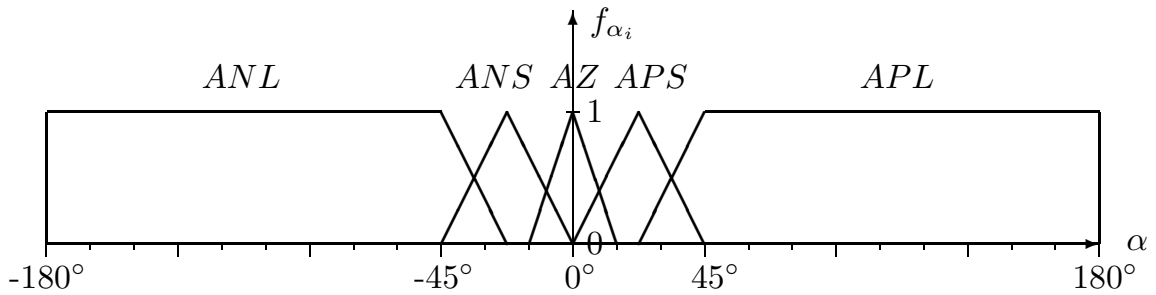


Figure 12: Fuzzy sets for the line of sight angle: *ANL* (negative large), *ANS* (negative small), *AZ* (zero), *APS* (positive small), and *APL* (positive large).

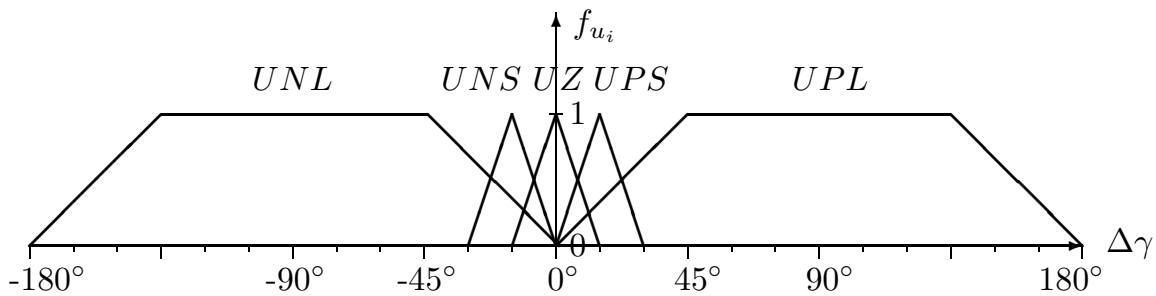


Figure 13: Fuzzy sets for the dog's change of heading angle: *UNL* (negative large), *UNS* (negative small), *UZ* (zero), *UPS* (positive small), and *UPL* (positive large).

The five fuzzy rules of the fuzzy associative memory of the fuzzy feedback controller can be written in shorthand as follows: $ANL \Rightarrow UNL$, $ANS \Rightarrow UNS$,

$AZ \Rightarrow UZ$, $APS \Rightarrow UPS$, and $APL \Rightarrow UPL$. This should be read as follows: If $\alpha(t_k)$ belongs to the fuzzy set ANL to degree d then the fuzzy set UNL for the change of heading angle should be fired to degree d , etc.

Since it is not quite clear what the dog means by “turning a little bit” or “turning pretty much”, a multiplicative doggy gain K is introduced. Furthermore, it is assumed that the dog cannot change his heading angle by more than 90° in either direction from one step to the next. This leads to the final control law

$$u(t_k) = \Delta\gamma(t_k) = \text{sat}\{K\mathcal{D}\{v_u(t_k)\}\} ,$$

where $v_u(t_k)$ is the fuzzy variable at the output of the fuzzy associative memory at time t_k (cf. Figure 4).

In order to evaluate the effectiveness of the dog’s fuzzy feedback control, a “dog chases cat” scenario is simulated.

In Figures 14 and 15, the cat runs along a rectangular course at constant speed, starting in the northwest corner. The dog starts running at a position close to the center of the rectangle. The dog’s speed exceeds the cat’s speed by 32.5%. A blood stain is left on the ground whenever the dog intercepts the cat with very high precision but the chase continues immediately.

A natural value for the doggy gain is $K = 0.5$ since there is no reason why the dog should turn by more than 45° when the cat is off by 45° . The simulation for this value of the doggy gain is shown in Figure 14. Obviously, the dog performs reasonably well. Of course, whenever the dog overshoots the cat, he has to manoeuvre with a high turning rate until the cat is again “in front” of him.

In Figure 15, the simulation is done with $K = 1$. Now, the dog is zig-zagging most of the time, changing his direction by $\pm 90^\circ$ from step to step. He only catches the cat because he luckily happens to cut the southeast corner in an efficient way. Essentially, the dog fails his mission with this high value of the doggy gain. Note however, that the critical value for the doggy gain depends on how much the dog’s speed exceeds the cat’s speed.

In [9], the fuzzy pursuit problem of this example has been extended to a fuzzy pursuit and evasion game, i.e., both the dog and the cat use fuzzy control schemes.

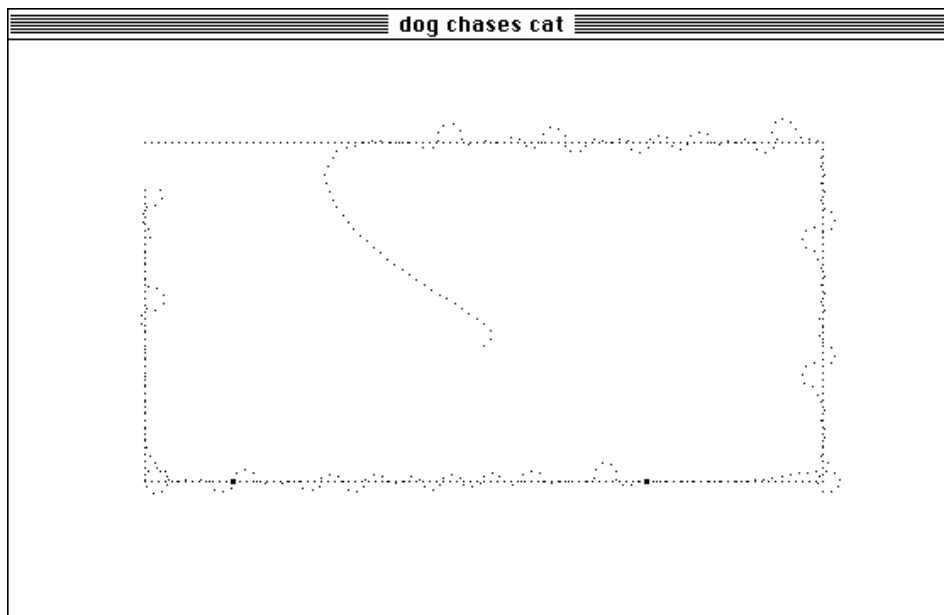


Figure 14: The dog chases the cat with gain $K = 0.5$.

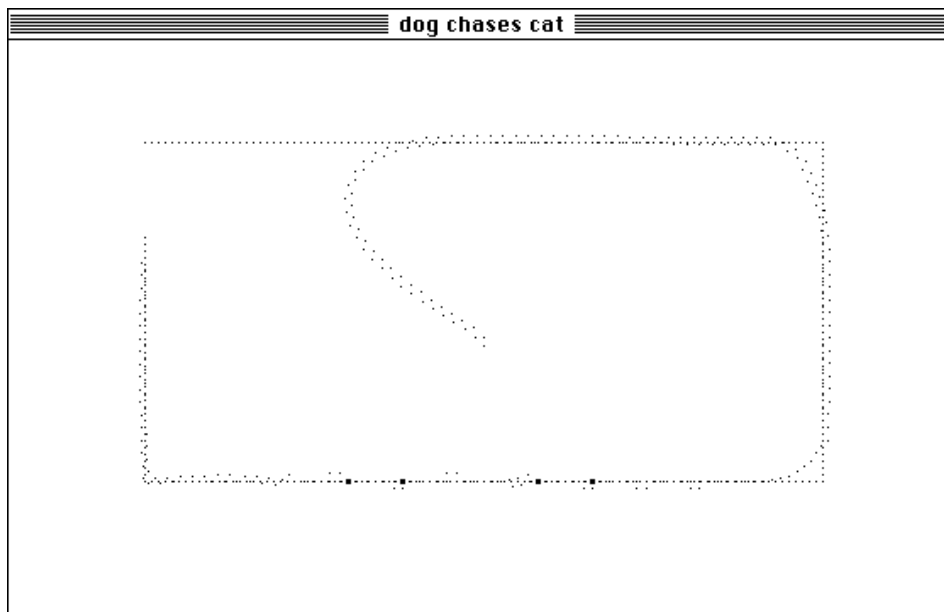


Figure 15: The dog chases the cat with gain $K = 1$.

9 Computational Issues

9.1 Efficient Defuzzification

Consider the task of simulating a fuzzy control system on a general purpose digital computer. For the simulation of the fuzzy (feedback) controller, Figures 4 and 5 suggest that the following sequence of operations should be executed at every sampling time:

- Fuzzify each of the error signals e_i to the corresponding fuzzy vector v_{e_i} : $v_{e_i} = \mathcal{F}(e_i)$.
- Evaluate all of the fuzzy rules defined for the fuzzy controller. Each of the fuzzy rules yields a fuzzy variable v_j which is represented by the modulated function $g_j(u)$.
- Calculate the sum of all of these fuzzy variables v_j in order to get the fuzzy control variable $v_u = \sum_j v_j$ which is represented by the function $g(u) = \sum_j g_j(u)$.
- Defuzzify the fuzzy control variable v_u by calculating its centroid u : $u = \mathcal{D}(v_u) = \frac{\int ug(u)du}{\int g(u)du}$.

Notice that for piecewise linear functions $g_j(u)$, the sum $g(u)$ is also a piecewise linear function and both of the integrals of the defuzzification operation can be calculated analytically. Therefore, using fuzzy sets with piecewise linear set membership functions only (such as “triangles”, “trapezoids”, or piecewise linear approximations of more sophisticated smooth functions) can lead to rather efficient program code.

In order to further reduce the run time of the simulation significantly, the result of the following lemma is needed.

Consider the modulated functions g_j , $j = 1, \dots, N$, and the corresponding function $g = \sum_{j=1}^N g_j$ of the fuzzy control signal v_u . Let w_{g_j} and c_{g_j} be the weights and the centroids, respectively, of the modulated functions g_j , i.e.,

$$w_{g_j} = \int g_j(u) du$$

and

$$c_{g_j} = \frac{\int ug_j(u) du}{\int g_j(u) du} = \frac{1}{w_{g_j}} \int ug_j(u) du .$$

Lemma: The defuzzified value u of the control variable can be obtained by evaluating the following purely algebraic equation:

$$u = \mathcal{D}(v_u) = \frac{\sum_{j=1}^N w_{g_j} c_{g_j}}{\sum_{j=1}^N w_{g_j}} .$$

Proof:

$$u = \mathcal{D}(v_u) = \frac{\int u g(u) du}{\int g(u) du} = \frac{\int u \sum_{j=1}^N g_j(u) du}{\int \sum_{j=1}^N g_j(u) du} = \frac{\sum_{j=1}^N \int u g_j(u) du}{\sum_{j=1}^N \int g_j(u) du} = \frac{\sum_{j=1}^N w_{g_j} c_{g_j}}{\sum_{j=1}^N w_{g_j}} .$$

Notice that the weight w_{g_j} and possibly the centroid c_{g_j} vary as the degree d_j of firing for the corresponding fuzzy set s_j varies.

However, if the linear modulation scheme is used, the variable weights w_{g_j} and centroids c_{g_j} can be expressed by the constant weights w_j and centroids c_j of the set membership functions f_j as follows:

$$w_{g_j} = d_j w_j \quad c_{g_j} \equiv c_j \quad \text{for } j = 1, \dots, N .$$

This observation leads to the following

Corollary: If the linear modulation scheme is used, the defuzzification operation simplifies to

$$u = \mathcal{D}(v_u) = \frac{\sum_{j=1}^N d_j w_j c_j}{\sum_{j=1}^N d_j w_j} .$$

Of course, the N products $w_j c_j$ need only be calculated once.

It now is obvious that in the Examples of Sections 8.2 and 8.3 there was no need to fully specify the set membership functions of the fuzzy sets for the control variables (cf. Figures 8 and 13, respectively). It would have sufficed to merely specify their weights and their centroids. This could have been done in the following equivalent way:

Example 1: Closed-loop halting control

fuzzy set s_i	centroid c_i	weight w_i
<i>UNX</i>	−500 N	277
<i>UNL</i>	−250 N	167
<i>UNS</i>	−100 N	111
<i>UPS</i>	100 N	111
<i>UPL</i>	250 N	167
<i>UPX</i>	500 N	277

Example 2: Dog chasing cat

fuzzy set s_i	centroid c_i	weight w_i
<i>UNL</i>	−90°	9
<i>UNS</i>	−15°	1
<i>UZ</i>	0°	1
<i>UPS</i>	15°	1
<i>UPL</i>	90°	9

9.2 Derivatives of the Control Function

If the linear modulation scheme is used, the simplified defuzzification formula of the above Lemma lends itself to calculating derivatives of the control function in a straightforward way.

In order to keep the notation fairly simple, consider a fuzzy feedback controller with the p input signals e_1, \dots, e_p , the single output signal u , and N fuzzy AND-rules. The partial derivative $\partial u/\partial e_i$ of the control function $u(e_1, \dots, e_p)$ with respect to the error signal e_i is

$$\frac{\partial u}{\partial e_i} = \frac{\left(\sum_{j=1}^N \frac{\partial d_j}{\partial e_i} w_j c_j \right) \left(\sum_{j=1}^N d_j w_j \right) - \left(\sum_{j=1}^N d_j w_j c_j \right) \left(\sum_{j=1}^N \frac{\partial d_j}{\partial e_i} w_j \right)}{\left(\sum_{j=1}^N d_j w_j \right)^2}.$$

Notice that j numbers the fuzzy rules rather than the fuzzy sets defined for the control signal. One and the same fuzzy set of the control signal may be fired by more than one fuzzy rule (cf. Figure 9).

Discontinuities of these partial derivatives are caused both by the “corners” of the (piecewise linear) set membership functions of the fuzzy sets covering the signal spaces where e_1, \dots, e_p live and the “corners” of the functions $d_j = \min(\dots)$ of the fuzzy AND-rules.

9.3 Observations and Suggestions

For the sake of simplicity, consider a fuzzy SISO feedback controller. From the formulae for the defuzzification operator and for the derivative of the control function, the following observations and suggestions ensue:

- If none of the fuzzy sets of the control signal u is fired to a strictly positive degree, the value of the defuzzified control signal is undefined. Therefore, the signal space E , where the error signal e lives, must be covered completely by its collection of fuzzy sets, and these fuzzy sets should overlap. In other words, every error $e \in E$ should belong to at least one fuzzy set to a strictly positive degree. Furthermore, in the associative memory, every fuzzy set of the error signal should fire (at least) one of the fuzzy sets of the control signal.
- If two neighbouring fuzzy sets of the error signal “touch” at e_1 but do not overlap, the control function is discontinuous at e_1 . For $e = e_1$, the result of the defuzzification operator is undefined. The value of the control signal must be defined separately. — In Example 1 (fuzzy halting control), the analogous situation occurs for $e_p = e_v = 0$. The obvious extra definition is $u_c(0, 0) = 0$.
- Assume that the linear modulation scheme is applied. If in some interval $[e_a, e_b] \subset E$ the error e belongs to exactly one fuzzy set to a strictly positive degree, the control function is constant on this interval, irrespective of the shape of this fuzzy set for the error signal. — In Example 2 (dog chasing cat), the dog will turn by $\Delta\gamma = \text{sat}\{K \cdot 15^\circ\}$ if the line of sight angle α is in the interval $[15^\circ, 22.5^\circ]$, or by $\Delta\gamma = \text{sat}\{K \cdot 90^\circ\}$ if the line of sight angle exceeds 45° . (Obviously, the dog could improve his performance significantly by choosing the centroids $+45^\circ$ and -45° for the fuzzy sets UPL and UNL , respectively, and the doggy gain $K = 1$.)
- Assume that the linear modulation scheme is applied. For the purpose of implementing a finished design of a fuzzy controller, only the weights w_i and the centroids c_i of the fuzzy sets for the control signals are needed in the defuzzification operation (cf. Section 9.1). — On the other hand, the complete specifications of the set membership functions of the fuzzy sets for the control signals are needed, if the fuzzy rules of the controller must be “learnt” by watching an expert performing the task at hand. This topic is beyond the scope of this report. The interested reader is referred to [1], and [8], and the references cited there.
- In Section 8, “static” fuzzy controllers are considered only. The reader should have no problem in extending these ideas to fuzzy controllers incorporating “dynamic compensation”. In the simplest case the input signals of the fuzzy dynamic compensator include the most recent error signals

$e_i(k)$ and the delayed error signals $e_i(k-1)$, $e_i(k-2)$, etc., and perhaps previously issued control signals $u_j(k-1)$, $u_j(k-2)$ etc.. In the latter case it is efficient to refuzzify the defuzzified control signals in order to obtain the required delayed fuzzy control variables. All or most of the processing is performed in the fuzzy part of the controller. In more sophisticated cases, non-fuzzy dynamic compensation (e.g., in the form of a full state observer) can be performed in a preprocessor to the fuzzy controller proper.

References

- [1] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1992.
- [2] H.-J. Zimmermann, *Fuzzy Set Theory and its Applications*, 2nd ed., Kluwer, Boston, MA, USA, 1991.
- [3] W. Pedrycz, *Fuzzy Control and Fuzzy Systems*, Electronic & Electrical Engineering Research Studies, Control Theory and Applications Series, vol. 3, Research Studies Press (Wiley), Taunton, Somerset, England, 1989.
- [4] D. J. Dubois, H. M. Prade, *Fuzzy Sets and Systems*, Academic Press, New York, NY, 1980.
- [5] L. A. Zadeh, *Fuzzy Sets and Applications: Selected Papers*, Wiley-Interscience, New York, NY, USA, 1987.
- [6] M. M. Gupta, T. Yamakawa, (eds.), *Fuzzy Computing, Hardware, and Applications*, North-Holland, Amsterdam, Netherlands, 1988.
- [7] M. Sugeno, (ed.), *Industrial Applications of Fuzzy Control*, North-Holland, Amsterdam, Netherlands, 1985.
- [8] C.-T. Lin, C. S. G. Lee, "Neural-Network-Based Fuzzy Logic Control and Decision Systems," *IEEE Transactions on Computers*, vol. 40(1991), pp. 1320–1336.
- [9] S. Ginsburg, R. Wimmer, H. P. Geering, "Clever Dog versus Smart Cat," *Proceedings of the International ICSC Symposium on Fuzzy Logic*, pp. A99–A104, Zurich, Switzerland, May, 1995.
- [10] H. Kiendl, *Fuzzy Control methodenorientiert*, Oldenbourg, Munich, Germany, 1997.