Chapter 9

Fuzzy Logic and Fuzzy Neural Networks

Fuzzy logic provides a method for representing analog processes in a digital framework. Processes that are implemented through fuzzy logic are often not easily separated into discrete segments and may be difficult to model with conventional mathematical or rule-based paradigms that require hard boundaries or decisions, i.e., binary logic where elements are a member of a given set or they are not. Consequently, fuzzy logic is valuable where the boundaries between sets of values are not sharply defined or there is partial occurrence of an event. In fuzzy set theory, an element's membership in a set is a matter of degree. This chapter describes the concepts inherent in fuzzy set theory and applies them to the solution of the inverted pendulum problem and a Kalman filter problem. Fuzzy and artificial neural network concepts may be combined to form adaptive fuzzy neural systems where either the weights and/or the input signals are fuzzy sets. Fuzzy set theory may be extended to fuse information from multiple sensors as discussed in the concluding section.

9.1 Conditions under which fuzzy logic provides an appropriate solution

Lotfi Zadeh originally developed fuzzy set theory in 1965. Zadeh reasoned that the rigidity of conventional set theory made it impossible to account for vagueness, imprecision, and shades of gray that are commonplace in real-world events.[1,2] By establishing rules and fuzzy sets, fuzzy logic creates a control surface that allows designers to build a control system even when their understanding of the mathematical behavior of the system is incomplete. Fuzzy logic permits the incorporation of the concept of vagueness into decision theory. For example, an observer may say that a person is "short" without specifying their actual height as a number. One may postulate that a reasonable specification for an adult of short stature is anyone less than 5 feet. However, other observers may declare 5 feet-2 inches or 5 feet-3 inches the cutoff between average and short height. Other examples of vagueness abound. An object may be said to be "near" or "far" from the observer, or that an automobile is traveling "faster" than the speed limit. In these examples, there is a range of values that satisfies the subjective term in quote marks. A further contrast of conventional and fuzzy set theory is given in Section G.1 of Appendix G.

The conditions under which fuzzy logic is an appropriate method for providing optimum control are:

- One or more of the control variables are continuous;
- Deficiencies are present in the mathematical model of the process:
  - Model does not exist;
  - Model exists but is difficult to encode;
  - Model is too complex to be evaluated in real time;
  - Memory demands are too large;
- High ambient noise is of concern;
- Inexpensive sensors or low-precision microcontrollers must be used;
- An expert is available to specify rules that govern the system behavior and the fuzzy sets that represent the characteristics of each variable.

9.2 Illustration of fuzzy logic in an automobile antilock braking system

The application of fuzzy control may be illustrated by examining an automobile antilock braking system. Here control rules are established for variables such as the vehicle's speed, brake pressure, brake temperature, interval between brake applications, and the angle of the vehicle's lateral motion relative to its forward motion. These variables are all continuous. Accordingly, the descriptor that characterizes a variable within its range of values is subject to the interpretation of the designer (e.g., speed characterized as fast, slow; pressure as high, low; temperature as cold, hot; and interval as large, small).[3]

Expanded ranges of temperature states such as cold, cool, tepid, warm, and hot are needed to fully specify the temperature variable. Yet, the change from one state to another is not precisely defined. Thus, a temperature of 280°F may belong to the warm or hot state depending on the interpretation afforded by the designer. But at no point can an increase of one-tenth of a degree be said to change a "warm" condition to one that is "hot." Therefore, the concept of cold, hot, etc. is subject to different interpretations by different experts at different points in the domain of the variable.

Fuzzy logic permits control statements to be written to accommodate the imprecise states of the variable. In the case of brake temperature, a fuzzy rule could take the form: "IF brake temperature is warm AND speed is not very fast, THEN brake pressure is slightly decreased." The degree to which the temperature is considered "warm" and the speed "not very fast" controls the extent to which the brake pressure is relaxed. In this respect, one fuzzy rule can replace many conventional mathematical rules.

9.3 Basic elements of a fuzzy system

There are three basic elements in a fuzzy system, namely, fuzzy sets, membership functions, and production rules. Fuzzy sets consist of the "imprecise-labeled" groups of the input and output variables that characterize the fuzzy system. In the antilock brake system example, the temperature variable is grouped into sets of cold, cool, tepid, warm, and hot. Each set has an associated membership function that provides a graphical representation of its boundaries. A particular value of the variable has membership in the set between the limits of 0 and 1. Zero indicates that the variable is not a member of the set, while 1 indicates that the variable is completely a member of the set. A number between 0 and 1, which is provided by the person defining the fuzzy sets, expresses intermediate membership in a set. A variable may be a member of more than one set. In the antilock brake system, a given temperature may sometimes be a member of the warm set and at other times a member of the hot set. Thus, each member of a fuzzy set is defined by an ordered pair of numbers in which the first is the value of the variable and the second is the associated membership of the variable in one or more sets.

Bell-shaped curves were originally used to define membership functions. However, the complex calculations and similarity of results led to their replacement with triangular and trapezoidal functions in many applications. The lengths of the triangle and trapezoid bases, and consequently the slopes of their sides, serve as design parameters that are calibrated for satisfactory system performance. Using a heuristic model, Kosko shows that contiguous fuzzy sets should generally overlap by approximately 25 percent in area.[4] Too much overlap may blur the distinction between the fuzzy set values. Too little overlap produces systems that resemble bivalent control, causing excessive overshoot and undershoot. Section G.2 of Appendix G elaborates on the effects of fuzzy set widths on control.

Production rules represent human knowledge in the form of "IF-THEN" logical statements. In artificial intelligence applications, IF-THEN statements are an integral part of expert systems. However, expert systems rely on binary on-off logic and probability to develop the inferences used in the production rules. Fuzzy sets incorporate vagueness into the production rules since they represent less precise linguistic terms, e.g., short, not very fast, and warm. The production rules operate in parallel and influence the output of the control system to varying degrees. The logical processing through the use of fuzzy sets is known as fuzzy logic.

9.4 Fuzzy logic processing

The fuzzy logic computation process is outlined in Figure 9.1. The sequence of fuzzy logic processing may be divided into two broad functions—inference and

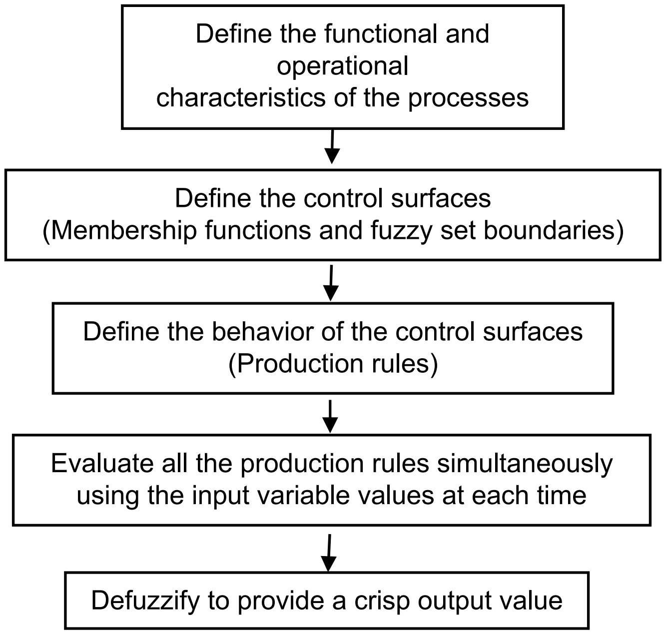

Figure 9.1 Fuzzy logic computation process.

defuzzification. Inference processing begins with the development of the production rules in the form of IF-THEN statements, also referred to as fuzzy associative memory. The antecedent or condition block of the rule begins with IF and the consequent or conclusion block begins with THEN. The value assigned to the consequent block is equal to the *logical product* of the activation values of the antecedent membership functions (i.e., fuzzy sets). The activation value is equal to the value of the membership function at which it is intersected by the input variable at the time instant being evaluated. If the antecedent block for a particular rule is a compound statement connected by AND, the logical product is the minimum value of the corresponding activation values of the antecedent membership functions. If the antecedent block for a particular rule is a compound statement connected by OR, the logical product is the maximum value of the activation values. All the production rules that apply to a process are evaluated simultaneously (i.e., as if linked by the OR conjunction), usually hundreds of times per second. When the logical product for the antecedents is zero, the value associated with the consequent membership function is also zero.

A defuzzification operation is performed to convert the fuzzy values, represented by the logical products and consequent membership functions, into a fixed and discrete output that can be used by the control system. Defuzzification may occur in several ways. Typically, a center of mass or fuzzy centroid computation is performed on the consequent fuzzy set. This is equivalent to finding the mode of the distribution if it is symmetric and unimodal. The fuzzy centroid incorporates all the information in the consequent fuzzy set. Two techniques are commonly used to calculate the fuzzy centroid. The first, correlation-minimum inference, clips the consequent fuzzy set at the value of the logical product as shown in Figure 9.2(a). An alternative approach uses correlation-product inference, which scales the consequent fuzzy set by multiplying it by the logical product value as illustrated in Figure 9.2(b). In this sense, correlation-product inferencing preserves more information than correlation-minimum inferencing.[4]

Still other methods are sometimes used for defuzzification. In the simplest approach, the maximum peak is selected as the defuzzified output value. It is applicable when the distribution formed by the logical product and consequent membership functions has a unique peak.[4,5] Illustrations of defuzzification methods are found in Section G.3 of Appendix G.

9.5 Fuzzy centroid calculation

Following the derivation given by Kosko, the fuzzy centroid c_k is[4]

$$c_k = \frac{\int y\, m_o(y)\, dy}{\int m_o(y)\, dy}, \tag{9-1}$$

where the limits of integration correspond to the entire universe of output parameter values,

y = output variable,

$m_o(y)$ = combined output fuzzy set formed by the simultaneous evaluation of all the production rules at time k

$$= \sum_{i=1}^{N} m_{o_i}(y), \tag{9-2}$$

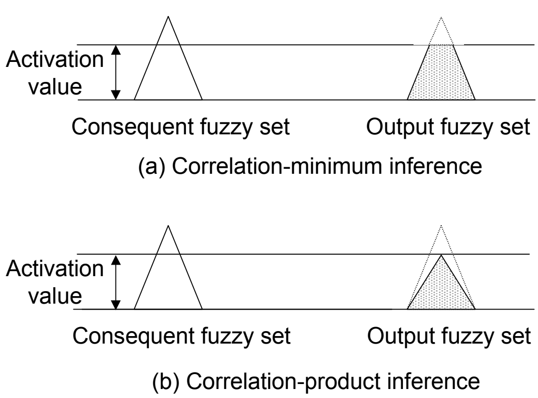

Figure 9.2 Shape of consequent membership functions for correlation-minimum and correlation-product inferencing.

N = number of production rules, and

o_i = output or consequent fuzzy set for i^{th} production rule.

If the universe of output parameter values can be expressed as p discrete values, Eq. (9-1) becomes

$$c_k = \frac{\sum_{j=1}^{p} y_j \, m_o(y_j)}{\sum_{j=1}^{p} m_o(y_j)} . \tag{9-3}$$

When the output fuzzy set is found using correlation-product inference, the global centroid c_k can be calculated from the local production rule centroids according to

$$c_k = \frac{\sum_{i=1}^{N} w_i \, c_i \, A_i}{\sum_{i=1}^{N} w_i \, A_i} , \tag{9-4}$$

where

w_i = activation value of the i^{th} production rule's consequent set L_i,

c_i = centroid of the i^{th} production rule's consequent set L_i

$$= \frac{\int y \, m_{L_i}(y) \, dy}{\int m_{L_i}(y) \, dy} , \tag{9-5}$$

A_i = area of the i^{th} production rule's consequent set L_i

$$= \int m_{L_i}(y) \, dy , \tag{9-6}$$

and L is the library of consequent sets.

Furthermore, when all the output fuzzy sets are symmetric and unimodal (e.g., triangles or trapezoids) and the number of library fuzzy sets is limited to seven, then the fuzzy centroid can be computed from only seven samples of the combined output fuzzy set o. In this case,

$$c_k = \frac{\sum_{j=1}^{7} y_j\, m_o(y_j)\, A_j}{\sum_{j=1}^{7} m_o(y_j)\, A_j}, \tag{9-7}$$

where A_j is the area of the j^{th} output fuzzy set and is equal to A_i as defined above. Thus, Eq. (9-7) provides a simpler but equivalent form of Eq. (9-1) if all the fuzzy sets are symmetric and unimodal and if correlation-product inference is used to form the output fuzzy sets o_i.

9.6 Balancing an inverted pendulum with fuzzy logic control

A control problem often used to illustrate the application of fuzzy logic is the balance of an inverted pendulum (equivalent to the balance of a stick on the palm of a hand) as depicted[4,6] in Figure 9.3.

9.6.1 Conventional mathematical solution

The mathematical model for a simple pendulum attached to a support driven horizontally with time is used to solve the problem with conventional control theory. The weight of the rod of length l is negligible compared to the weight of the mass m at the end of the rod in this model.

The x, y position and $\dot{x}$, $\dot{y}$ velocity coordinates of the mass m are expressed as

$$x, y = x_s + l\sin\theta, -l\cos\theta \tag{9-8}$$

and

$$\dot{x},\ \dot{y} = \dot{x}_s + l\dot{\theta}\cos\theta,\ l\dot{\theta}\sin\theta, \tag{9-9}$$

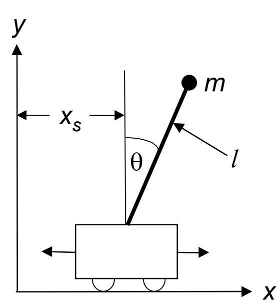

Figure 9.3 Model for balancing an inverted pendulum.

where θ is the angular displacement of the pendulum from equilibrium and a dot over a variable denotes differentiation with respect to time.

The equation of motion that describes the movement of the pendulum is found from the Lagrangian L of the system given by

$$L = T - V, \tag{9-10}$$

where T is the kinetic energy and V the potential energy of the pendulum as a function of time t.[7,8] Upon substituting the expressions for the kinetic and potential energy, the Lagrangian becomes

$$L = \frac{m}{2}(\dot{x}_s^2 + l^2\dot{\theta}^2 + 2l\,\dot{x}_s\,\dot{\theta}\cos\theta) + m\,g\,l\cos\theta, \tag{9-11}$$

where $\dot{\theta}$ is the rate of change of angular displacement and g is the acceleration due to gravity.

The equation of motion is expressed by Lagrange's equation as[9,10]

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0. \tag{9-12}$$

Substituting Eq. (9-11) into Eq. (9-12) gives

$$l\,\ddot{\theta} + \ddot{x}_s\cos\theta + g\sin\theta = 0. \tag{9-13}$$

The solution of Eq. (9-13) is not elementary because it involves an elliptic integral.[11] If θ is small ($|\theta| < 0.3$ rad), however, $\sin\theta$ and θ are nearly equal and Eq. (9-13) is closely approximated by the simpler equation

$$l\,\ddot{\theta} + \ddot{x}_s + g\,\theta = 0. \tag{9-14}$$

When $x_s = x_0\cos\omega t$, Eq. (9-14) becomes

$$\ddot{\theta} + \omega_0^2\,\theta = \frac{x_0}{l}\omega^2\cos\omega t, \tag{9-15}$$

where

$$\omega_0 = \sqrt{\frac{g}{l}}. \tag{9-16}$$

A particular solution of Eq. (9-15) obtained using the method of undetermined coefficients is[9]

$$\theta_p(t) = \frac{x_0\,\omega^2 \cos\omega t}{l\,(\omega_0^2 - \omega^2)} \text{ if } \omega_0 \neq \omega. \tag{9-17}$$

The general solution of Eq. (9-15) is then

$$\theta(t) = \frac{x_0\,\omega^2 \cos\omega t}{l\,(\omega_0^2 - \omega^2)} + A\cos\omega_0 t + B\sin\omega_0 t \text{ for } \omega_0 \neq \omega. \tag{9-18}$$

As long as $\omega \neq \omega_0$, the motion of the pendulum is bounded. Resonance (i.e., build up of large amplitude angular displacement) occurs if $\omega_0 = \omega$. At resonance, the equation of motion becomes

$$\theta(t) = \frac{x_0\,\omega_0\, t\sin\omega_0 t}{2l} + A\cos\omega_0 t + B\sin\omega_0 t\,. \tag{9-19}$$

The constants A and B are evaluated from boundary conditions imposed on θ and $\dot{\theta}$ at $t = 0$.

9.6.2 Fuzzy logic solution

Fuzzy logic generates an approximate solution that does not require knowledge of the mathematical equations that describe the motion of the pendulum or their solution. Instead, the seven production rules listed in Table 9.1 are applied.

Production rules describe how the states of the input variables are combined. In this example, the input variables are the angle θ the pendulum makes with the vertical and the instantaneous rate of change of the angle, now denoted by $\Delta\theta$. Both variables take on positive and negative values. The antecedent membership functions that correspond to each variable represent the ambiguous words in the antecedent block of the rules, such as “quickly,” “moderately,” “a little,” and “slowly.” These words are coded into labels displayed on the membership functions shown in Figure 9.4 by the terms “large,” “medium,” and “small.”

The seven labels consist of three ranges in the positive direction, three in the negative direction, and a zero. The membership functions for each variable overlap by approximately 25 percent in area to ensure a smooth system response when the input level is not clear or when the level changes constantly. The membership functions describe the degree to which θ and $\Delta\theta$ belong to their respective fuzzy sets. The numbers at the bases of the triangular membership functions are used later to identify the centroids of each fuzzy set.

Table 9.1 Production rules for balancing an inverted pendulum.

Rule	Antecedent Block	Consequent Block
1	IF the stick is inclined moderately to the left *and* is almost still	THEN move the hand moderately to the left quickly
2	IF the stick is inclined a little to the left *and* is falling slowly	THEN move the hand moderately to the left a little quickly
3	IF the stick is inclined a little to the left *and* is rising slowly	THEN don't move the hand much
4	IF the stick is inclined moderately to the right *and* is almost still	THEN move the hand moderately to the right quickly
5	IF the stick is inclined a little to the right *and* is falling slowly	THEN move the hand moderately to the right a little quickly
6	IF the stick is inclined a little to the right *and* is rising slowly	THEN don't move the hand much
7	IF the stick is almost not inclined *and* is almost still	THEN don't move the hand much

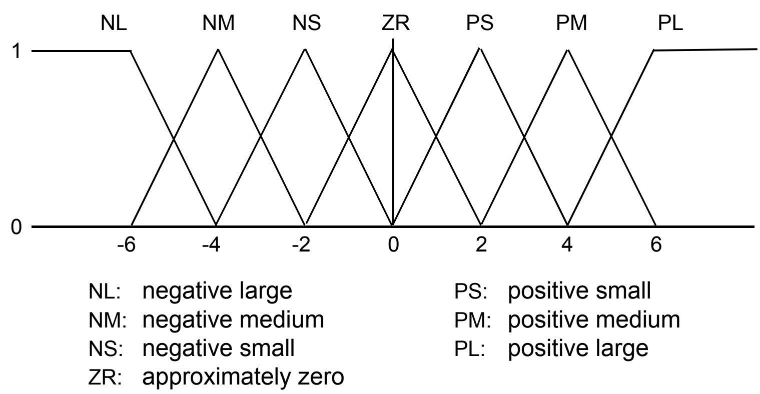

Figure 9.4 Triangle-shaped membership functions for the inverted pendulum example.

Consequent membership functions specify the motion of the pendulum base resulting from the θ and Δθ values input to the antecedent block. The minimum of the activation values of the antecedent membership functions is selected as the input to the consequent fuzzy sets since the antecedent conditions are linked by AND. Finally, the distribution formed by the simultaneous evaluation of all the production rules is defuzzified. In this example, a center of mass or fuzzy centroid calculation is used to compute the crisp value for the velocity of the base of the pendulum.

The fuzzy processing sequence for balancing the inverted pendulum is illustrated in Figure 9.5 for a single time instant. One input to the fuzzy controller is provided by a potentiometer that measures the angle θ. The second input represents Δθ as approximated by the difference between the present angle measurement and the previous angle measurement. The output of the control system is fed to a servomotor that moves the base of the pendulum at velocity Δ*v*. If the pendulum falls to the left, its base should move to the left and vice versa.

Examining the antecedent block for Production Rule 1 in Figure 9.5 shows that θ intercepts the membership function for "inclined moderately to the left" at 0.7 and Δθ crosses the membership function for "almost still" at 0.8. The logical product of these two values is 0.7, the minimum value of the two inputs. The value of 0.7 is next associated with the consequent block of Production Rule 1, "move moderately to the left quickly." Proceeding to Production Rule 2, we find that θ intercepts the membership function for "inclined a little to the left" at 0.3 and Δθ crosses the membership function for "falling slowly" at 0.2. The logical

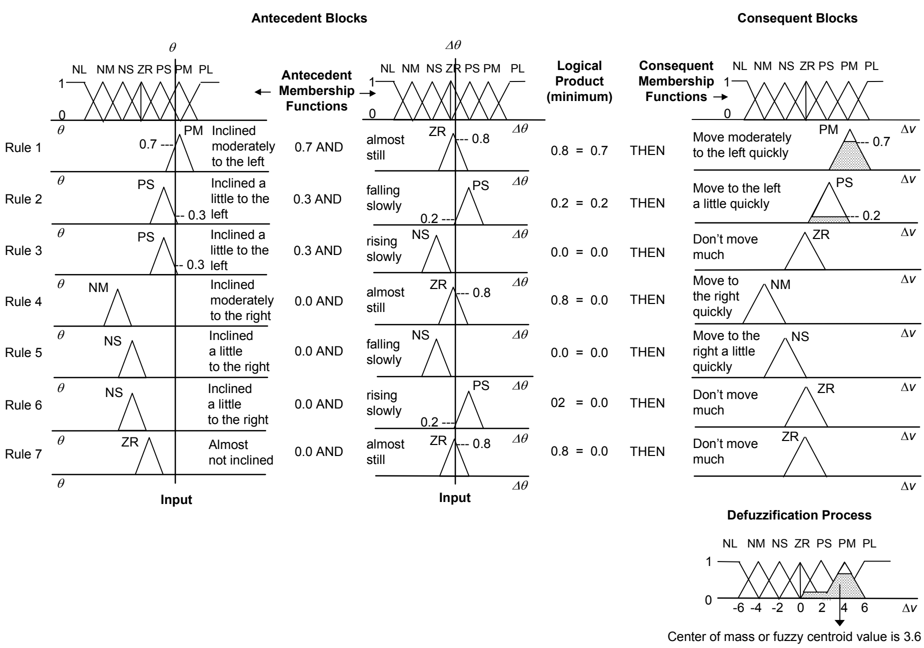

Figure 9.5 Fuzzy logic inferencing and defuzzification process for balancing an inverted pendulum. (G. Anderson, "Applying fuzzy logic in the real world," Reprinted with permission of Sensors magazine, Sept. 1992. Helmers Publishing, Inc. Copyright 1992.)

product value of 0.2 is then associated with the consequent block of Production Rule 2, "move to the left a little quickly." The logical products for the remaining production rules are zero since at least one of the antecedent membership functions is zero.

Defuzzification occurs once the simultaneous processing of all the rules is complete for the time sample. Defuzzification is performed by the center of mass calculation illustrated in the lower right corner of the figure for correlation-minimum inference. The defuzzified output controls the direction and speed of the movement required to balance the pendulum. In this case, the command instructs the servomotor to move the base of the pendulum to the left at a velocity equal to the center of mass value of 3.6.

The value of 3.6 was calculated using Eq. (9-3) and the entries in Table 9.2. The numerator in Eq. (9-3) is equal to the sum of the products of $y_j\ m_o(y_j)$, while the denominator is equal to the sum of $m_o(y_j)$ for j =1 to 7. Since the areas A_j of the consequent sets are equal, the sum of the products of w_j and c_j may be substituted for the numerator and the sum w_j for the denominator, where w_j is the activation value and c_j the centroid of the consequent of production rule j.

Table 9.2 Outputs for the inverted pendulum example.

j	**Consequent**	w_j	c_j	w_jc_j
1	PM	0.7	4	2.8
2	PS	0.2	2	0.4
3	ZR	0	0	0
4	NM	0	-4	0
5	NS	0	-2	0
6	ZR	0	0	0
7	ZR	0	0	0
Sum		**0.9**		**3.2**

Although the output from a fuzzy system is crisp, the solution is still approximate as it is subject to the vagaries of the rule set and the membership functions. Fuzzy logic control is considered robust because of its tolerance for imprecision. Fuzzy systems can operate with reasonable performance even when data are missing or membership functions are loosely defined.

9.7 Fuzzy logic applied to multitarget tracking

This example uses a fuzzy Kalman filter to correct the estimate of the target position and velocity state vector at time k+1 using parameters available at time k. The Kalman filter provides a state estimate that minimizes the mean squared error between the estimated and observed position and velocity states over the entire history of the measurements.[12,13] The discrete time implementation of the fuzzy Kalman filter reduces computation time as compared with the conventional Kalman filter implementation, especially for multidimensional, multitarget scenarios.

9.7.1 Conventional Kalman filter approach

A Kalman filter contains a state transition model and a measurement model. The *state transition model* predicts the target position and velocity coordinates of the state vector **X** at time k+1 based on information available at time k according to

$$\mathbf{X}_{k+1} = \mathbf{F}\mathbf{X}_k + \mathbf{J}\mathbf{w}_k, \tag{9-20}$$

where

$$\mathbf{X}_k^{\mathrm{T}} = [x_k \quad \dot{x}_k \quad y_k \quad \dot{y}_k]^{\mathrm{T}} \text{ (in two dimensions)},$$

T = transpose operation,

F = system or transition matrix,

J = input matrix, and

$\mathbf{w}_k$ = disturbance input vector or process noise that exists in the target motion model.

The Kalman filter extrapolates the state vector **X** and measurement error covariance matrix **P** to time k using data collected at time k–1. Accordingly, the state vector estimate $\hat{\mathbf{X}}_{k|k-1}$ provided by the Kalman filter is given by

$$\hat{\mathbf{X}}_{k|k-1} = \mathbf{F}\hat{\mathbf{X}}_{k-1|k-1} \tag{9-21}$$

and the covariance matrix $\mathbf{P}_{k|k-1}$ by

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^{\mathrm{T}} + \mathbf{J}\Sigma_k\mathbf{J}^{\mathrm{T}}, \tag{9-22}$$

where the notation $k|k$-1 indicates the estimated or extrapolated value at time k calculated with data gathered at time k–1, and

Σ_k = covariance matrix for the disturbance input vector $\mathbf{w}_k$.

The *measurement model* uses new information contained in the innovation vector to correct the extrapolated state estimate. The innovation vector $\tilde{\mathbf{Z}}_k$ is defined as the difference between the observed and extrapolated measurement vectors such that

$$\tilde{\mathbf{Z}}_k = \mathbf{Z}_k - \hat{\mathbf{Z}}_{k|k-1} = \mathbf{Z}_k - \mathbf{H}\hat{\mathbf{X}}_{k|k-1}, \tag{9-23}$$

where

$$\mathbf{Z}_k = \mathbf{H}\,\mathbf{X}_k + \mathbf{n}_k \tag{9-24}$$

$\mathbf{H}$ = output or observation matrix, and

$\mathbf{n}_k$ = measurement noise vector that generally contains a fixed, but unknown bias and a random component.

Finally, the extrapolated state vector and covariance matrix in Eqs. (9-21) and (9-22) are corrected as

$$\hat{\mathbf{X}}_{k|k} = \hat{\mathbf{X}}_{k|k-1} + \mathbf{G}_k \tilde{\mathbf{Z}}_k \tag{9-25}$$

and

$$\mathbf{P}_{k|k} = [(\mathbf{P}_{k|k-1})^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}_k^{-1}\mathbf{H}]^{-1}, \tag{9-26}$$

where

$\mathbf{G}_k$ = filter gain, a function of the measurement error covariance matrix,

$$= \mathbf{P}_{k|k-1}\;\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^{\mathrm{T}} + \mathbf{R}_k)^{-1} \tag{9-27}$$

and

$\mathbf{R}_k$ = covariance matrix of the noise vector $\mathbf{n}_k$.

The corrected covariance matrix may also be written in terms of the gain as[14]

$$\mathbf{P}_{k|k} = [\mathbf{I} - \mathbf{G}_k\mathbf{H}]\mathbf{P}_{k|k-1}, \tag{9-28}$$

where $\mathbf{I}$ is the identity matrix. The matrix inversion lemma may be used to convert the corrected covariance matrix into the form

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}^{\mathrm{T}}[\mathbf{H}^{\mathrm{T}}\mathbf{P}_{k|k-1}\mathbf{H} + \mathbf{R}_{\mathbf{k}}]^{-1}\mathbf{H}\mathbf{P}_{k|k-1}. \tag{9-29}$$

9.7.2 Fuzzy Kalman filter approach

Fuzzy logic reduces the time to perform complex matrix multiplications that are characteristic of higher order systems. This example treats the incomplete information case in which only the position variables are available for measurement. Fuzzy logic is used for data association and for updating the extrapolated state vector. Data are associated with a specific target by defining (1) a validation gate based on Euclidean distance and (2) a similarity measure based on object size and intensity. A fuzzy return processor is created to execute these functions. The output of this process is the average innovation vector used as the input to a fuzzy state correlator. The fuzzy state correlator updates the extrapolated state estimate of the position and velocity of the target at time k given information at time k–1.

The extrapolation equation for the state vector **X** that appears in the fuzzy Kalman filter is identical to Eq. (9-21). The approaches depart by using fuzzy logic to generate the correction vector $\mathbf{C}_k$ needed to update the state estimate according to

$$\hat{\mathbf{X}}_{k|k} = \hat{\mathbf{X}}_{k|k-1} + \mathbf{G}_k \mathbf{C}_k \,, \tag{9-30}$$

where $\mathbf{C}_k$ is the fuzzy equivalent of the innovation vector $\tilde{\mathbf{Z}}_k$.

Step 1: Fuzzy return processor. The function of the fuzzy return processor is to reduce the uncertainty in target identification caused by clutter, background noise, and image processing. In this example, the data used to identify and track the targets are produced by a sequence of forward-looking infrared (FLIR) images.[12] The passive FLIR sensor allows the position, but not the velocity, of the target to be measured. The fuzzy return processor produces two parameters that are used to associate the FLIR sensor data with a specific target. The first is based on a validation gate. The second is a similarity measure related to the rectangular size of the image and intensity of the pixels in the image.

Data validation is needed when multiple returns are received from the vicinity of the target at time k. Fuzzy validation imparts a degree of validity between 0 and 1 to each return. The validity $\beta_{\text{valid},i}$ for the i^{th} return is inversely related to the Euclidean norm of the innovation vector defined as

$$\left\| \tilde{\mathbf{Z}}_{k,i} \right\| = [(x_{k,i} - \hat{x}_k)^2 \;+\; (y_{k,i} - \hat{y}_k)^2]^{\frac{1}{2}}, \tag{9-31}$$

where

$\tilde{\mathbf{Z}}_{k,i}$ = innovation vector at time k for the i^{th} return

$$= \mathbf{Z}_{k,i} - \hat{\mathbf{Z}}_{k|k-1} \text{ [analogous to Eq. (9-23)]} \tag{9-32}$$

and the parameters in parentheses represent the observed and extrapolated values of x and y, respectively.

The fuzzy membership function for the validity is illustrated in Figure 9.6. The constants d_1 and d_2 are varied to optimize the performance of the filter as the number of clutter returns changes. The degree of validity is combined with the similarity measure to calculate an average innovation vector $\tilde{\mathbf{Z}}'_k$, which is used in the fuzzy state correlator.

The similarity measure is used to correlate new data with previously identified targets. The correlation is performed using size-difference and intensity-difference antecedent membership functions as shown in Figure 9.7.

An example of the production rules that determine if a return i falls within the size and intensity validation gate is

> IF (size_diff$_i$_is_small) AND (intensity_diff$_i$_is small), THEN (degree_of_ similarity$_i$_is_high).

The complete set of production rules needed to associate new data with targets is illustrated in Table 9.3.

Once the data have been associated with previously identified targets, a similarity membership function, such as that depicted in Figure 9.8, is used to find the consequent values. The result is defuzzified to find the weight $\beta_{\text{similar},i}$ through a center of mass calculation based on the activation value of the degree_of_similarity and the inferencing method applied to the consequent fuzzy sets.

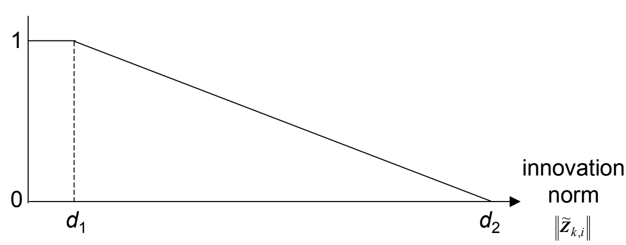

Figure 9.6 Validity membership function.

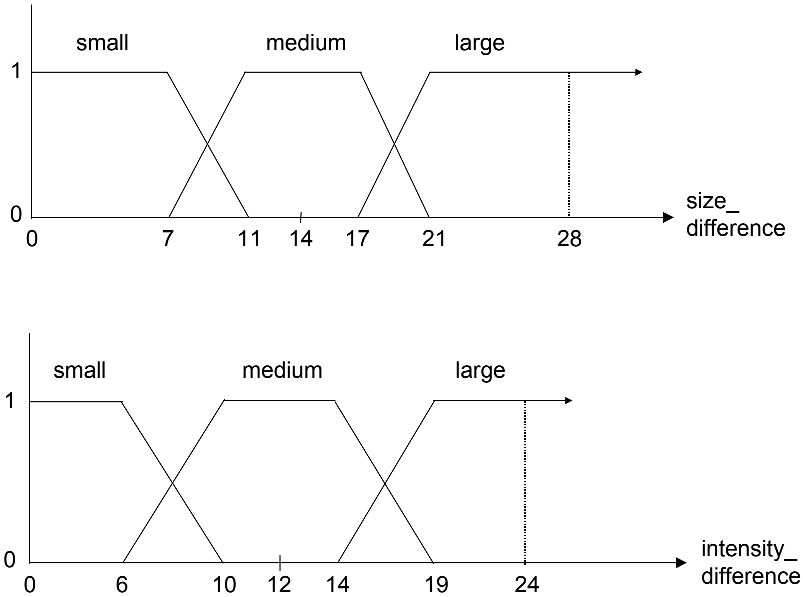

Figure 9.7 Size-difference and intensity-difference membership functions.

Table 9.3 Fuzzy associative memory rules for degree of similarity.

Intensity_diff	Size_diff		
	Small	**Medium**	**Large**
Small	High	High	Medium
Medium	High	Medium	Low
Large	Medium	Low	Low

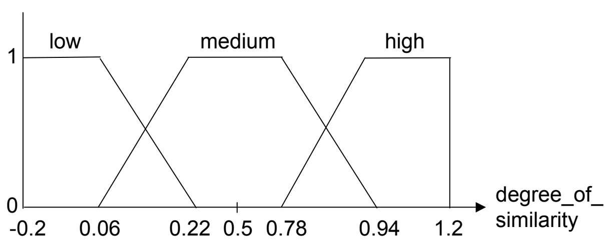

Figure 9.8 Similarity membership functions.

The weights $\beta_{valid,i}$ and $\beta_{similar,i}$ found for all $i = 1, \ldots, n$ returns are used to calculate a weighted average innovation vector as

$$\tilde{\mathbf{Z}}'_k = \begin{bmatrix} \tilde{x}'_k \\ \tilde{y}'_k \end{bmatrix} = \sum_{i=1}^{n} \beta_i \quad \tilde{\mathbf{Z}}_{k,i}\,, \tag{9-33}$$

where β_i, with values between 0 and 1, is the weight assigned to the i^{th} innovation vector. It represents the belief or confidence that the identified return is the target. The value β_i is calculated as a linear combination of $\beta_{valid,i}$ and $\beta_{similar,i}$ as

$$\beta_i = b_1\,\beta_{valid,i} + b_2\,\beta_{similar,i}, \tag{9-34}$$

where the constants b_1 and b_2 sum to unity. These constants are used to alter the return processor's performance by trading off the relative importance of validity and similarity. The weighted average innovation vector as found from Eq. (9-33) is used as the input to the fuzzy state correlator for the particular target of interest.

Step 2: Fuzzy state correlator. The fuzzy state correlator calculates the correction vector $\mathbf{C}_k$ that updates the state estimate for the position and velocity of the target at time k according to Eq. (9-30).

To find $\mathbf{C}_k$, the weighted average innovation vector is first separated into its x and y components, e_x and e_y. An error vector $\boldsymbol{e}_k$ is then defined as

$$\mathbf{e}_k = \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \tilde{\mathbf{Z}}'_k\,. \tag{9-35}$$

Since the x and y directions are independent, Horton and Jones[12] develop the fuzzy state correlator for the x direction and then generalize the result to include the y direction. The production rules that determine the fuzzy output of the correlator have two antecedents, the average x component of the innovation vector e_x and the differential error d_e_x. Assuming the current and previous values of the error vector, e_x and past_e_x, are available, allows d_e_x to be computed as

$$d\_e_x = (e_x - \text{past\_}e_x)/\text{timestep}. \tag{9-36}$$

The antecedent membership functions that define the fuzzy values for e_x and d_e_x are shown in Figure 9.9.

Using the values of e_x and d_e_x, the production rules for the fuzzy state correlator take the form

IF (e_x is large negative [LN]) AND (d_e_x is large positive [LP]), THEN ($C_{k,x}$ is zero [ZE]).

Table 9.4 summarizes the 49 rules needed to implement the fuzzy state correlator.

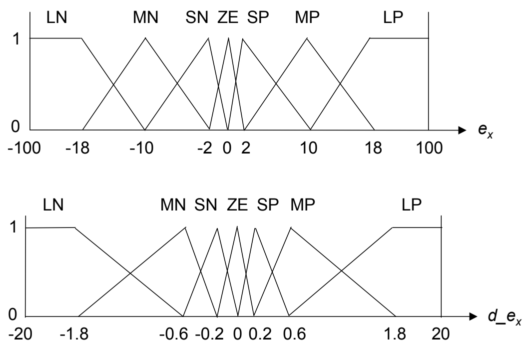

Figure 9.9 Innovation vector and the differential error antecedent membership functions.

Table 9.4 Fuzzy associative memory rules for the fuzzy state correlator.

d_e_x	e_x						
	LN	**MN**	**SN**	**ZE**	**SP**	**MP**	**LP**
Large negative (LN)	LN	LN	MN	MN	MN	SN	ZE
Medium negative (MN)	LN	MN	MN	MN	SN	ZE	SP
Small negative (SN)	MN	MN	MN	SN	ZE	SP	MP
Zero (ZE)	MN	MN	SN	ZE	SP	MP	MP
Small positive (SP)	MN	SN	ZE	SP	MP	MP	MP
Medium positive (MP)	SN	ZE	SP	MP	MP	MP	LP
Large positive (LP)	ZE	SP	MP	MP	MP	LP	LP

After tuning the output to reduce the average root least-square error (RLSE), Horton and Jones find the consequent membership functions to be those shown in Figure 9.10. In this example, the bases of the trapezoidal and triangular membership functions were scaled to provide the desired system response.

The defuzzified output is calculated from the center of mass or fuzzy centroid corresponding to the activation value of the correction vector $\mathbf{C}_k$ and the inferencing method applied to the consequent fuzzy sets. The performance of the fuzzy tracker was improved by adding a variable gain Γ to the defuzzified inputs and outputs of the system as shown in Figure 9.11 for the x direction. By proper choice of gains ($\Gamma_1 = 1, \Gamma_2 = 1, \Gamma_3 = 7$), the average RLSE error was reduced to approximately 1 from its value of 5 obtained when the gains were not optimized.

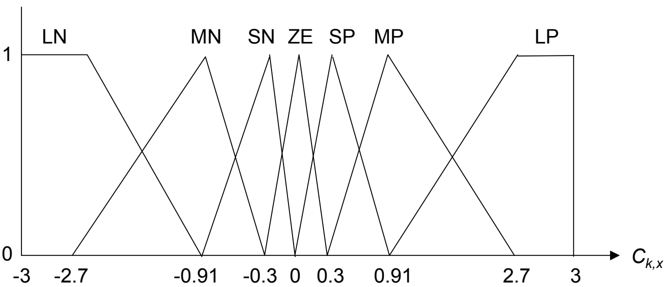

Figure 9.10 Correction vector consequent membership functions.

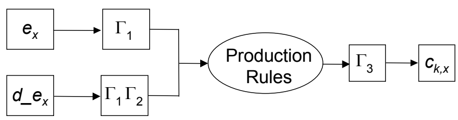

Figure 9.11 Improving performance of the fuzzy tracker by applying gains to the crisp inputs and outputs.

9.8 Fuzzy neural networks

Adaptive fuzzy neural systems use sample data and neural algorithms to define the fuzzy system at each time instant. Either the weights and/or the input signals are fuzzy sets. Thus, fuzzy neural networks may be characterized by

- Real number signals with fuzzy set weights;
- Fuzzy signals with real number weights;
- Both fuzzy signals and fuzzy weights.

An example of the first class of fuzzy neural network is the fuzzy neuron developed by Yamakawa et al.[15,16] As illustrated in Figure 9.12, the neuron contains real number inputs x_i ($i = 1, \dots, m$) and fixed fuzzy sets μ_{ik} ($k = 1, \dots, n$) that modify the real number weights w_{ik}. The network is trained with a heuristic learning algorithm that updates the weights with a formula similar to the backpropagation algorithm. A restriction is placed on the fuzzy sets μ_{ik} such that only two neighboring μ_{ik} can be nonzero for a given x_i.

Accordingly, if $\mu_{ik}(x_i)$ and $\mu_{i,k+1}(x_i)$ are nonzero in Figure 9.12, then

$$y_i = \mu_{ik}(x_i)w_{ik} + \mu_{i,k+1}(x_i)w_{i,k+1}. \tag{9-37}$$

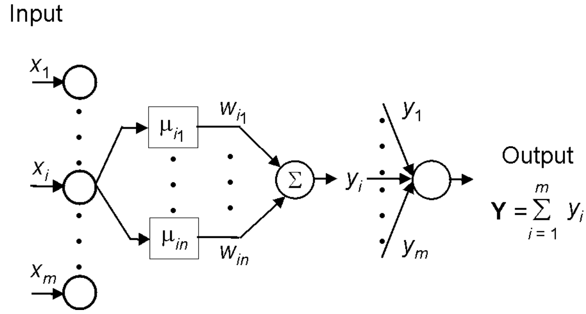

Figure 9.12 Yamakawa's fuzzy neuron.

The output **Y** of the neuron is equal to the sum of the y_i such that

$$\mathbf{Y} = y_1 + y_2 + \ldots + y_m. \tag{9-38}$$

Nakamura et al.[17] and Tokunaga et al.[18] developed another type of fuzzy neuron having the topology shown in Figure 9.13.

The learning algorithm in this case optimizes both the trapezoidal membership functions for fuzzy sets μ_i (i = 1, … , m) and the real number weights w_i. The output **Y** is equal to

$$\mathbf{Y} = \sum_{i=1}^{m} w_i \mu_i(x_i) / \sum_{i=1}^{m} w_i \,. \tag{9-39}$$

The second and third classes of fuzzy neural networks are similar in topology to multilayer feedforward networks. The second class of fuzzy neural networks contains a fuzzy input signal vector and a fuzzy output signal vector. Backpropagation and other training algorithms have been proposed for this class of network.[19,20,21] The third class of fuzzy neural networks contains fuzzy input

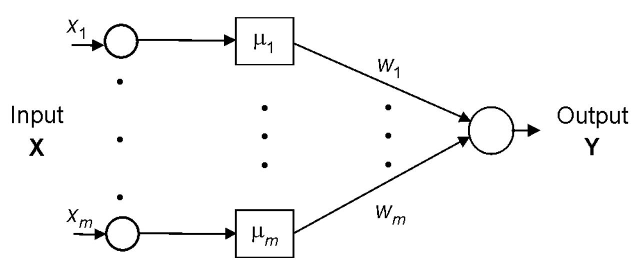

Figure 9.13 Nakamura's and Tokunaga's fuzzy neuron.

and output signal vectors and fuzzy weights that act on the signals entering each layer. Learning algorithms for the third class of fuzzy neural networks are discussed by Buckley and Hayashi.[15] They surmise that learning algorithms will probably be specialized procedures when operations other than multiplication and addition act on signals in this class of fuzzy neural networks.

9.9 Fusion of fuzzy-valued information from multiple sources

Yager considered the problem of aggregating information from multiple sources when their information is imprecise.[22] For example, object distances may be stated in terms of near, mid-range, and far by available sensors or human observers. Object size may be given in terms of small, medium, and large or object temperatures in terms of statements such as cold, warm, and hot. The imprecise information is combined using two knowledge structures. The first produces a combinability relationship, which allows inclusion of information about the appropriateness of aggregating different values from the observation space. The second is a fuzzy measure, which carries information about the confidence of using various subsets of data from the available sensors. By appropriately selecting the knowledge structures, different classes of fusion processes can be modeled. Yager demonstrates that if an idempotent fusion rule is assumed and if a combinability relation that only allows fusion of identical elements is used, the fusion of any fuzzy subsets is their intersection. A defuzzification method is described, which reduces to a center of area procedure when it is acceptable to fuse any values drawn from the observation space.

Denoeux discusses another approach to the incorporation of imprecise degrees of belief provided by multiple sensors to assist in decision making and pattern classification.[23] He adopts Smets transferable belief model described in Chapter 6 to represent and combine fuzzy-valued information using an evidence theory framework. To this end, the concept of belief mass is generalized such that the uncertainty attached to a belief is described in terms of a "crisp" interval-valued or a fuzzy-valued belief structure. An example of an interval-valued belief for a proposition is $m(a_1) = (0.38, 0.65)$, meaning that the information source ascribes a belief that ranges from 0.38 to 0.65 to proposition a_1. An example of a fuzzy-valued belief assignment for two subsets b_1 and b_2 belonging to possibility space $\Omega = \{1, \ldots, 10\}$is $m(b_1) = \{1, 2, 3, 4, 5\}$ and $m(b_2) = \{0.1/2, 0.5/3, 1/4, 0.5/5, 0.1/6\}$. The nomenclature that describes the fuzzy-valued assignments for subset b_2 is in the form of corresponding belief/value pairs, e.g., assign belief of 0.1 that the proportion has a value of 2. Subset b_1 is a crisp subset of Ω corresponding to the proposition "X is strictly smaller than 6", where X represents the unknown variable of interest. Subset b_2 is a fuzzy subset that corresponds to the fuzzy proposition "X is around 4."

9.10 Summary

Fuzzy logic, somewhat contrary to its name, is a well-defined discipline that finds application where the boundaries between sets of values are not sharply defined, where there is partial occurrence of an event, or where the specific mathematical equations that govern a process are not known. Fuzzy logic also is used to reduce the computation time that would otherwise be needed to control complex or multidimensional processes or where low-cost control process implementations are needed.

A fuzzy control system nonlinearly transforms exact or fuzzy state inputs into a fuzzy set output. Fuzzy systems contain membership functions and production rules or fuzzy associative memory. Membership functions define the boundaries of the fuzzy sets consisting of the input and output variables. The production rules operate in parallel and are activated to different degrees through the membership functions. Each rule represents ambiguous expert knowledge or learned input-output transformations. A rule can also summarize the behavior of a specific mathematical model. The output fuzzy set is usually defuzzified using a centroid calculation to generate an exact numerical output for the control system.

The balance of an inverted pendulum and track estimation with a Kalman filter were described to illustrate the wide applicability of fuzzy logic and contrast the fuzzy solution with the conventional mathematical solution. Adaptive fuzzy neural systems can also be constructed. These rely on sample data and neural algorithms to define the fuzzy system at each time instant.

The value of fuzzy logic to data fusion has appeal in identifying battlefield objects, describing the composition of enemy units, and interpreting enemy intent and operational objectives. Perhaps the most difficult aspect of its application is the definition of the membership functions that specify the influence of the input variables on the fuzzy system output.

One can envision multiple data source inputs to a fuzzy logic system whose goal is to detect and classify objects or potential threats. Each data source provides one or more inputs that have values, which are used to find the membership (i.e., activation value) of the input in one or more fuzzy sets. For example, fuzzy sets can consist of "not a member," "possibly a member," "likely a member," "most likely a member," and "positively a member" of some target or threat class. Each set has an associated membership function, which can be in the form of a graphical representation of its boundaries or a membership interval expressed as a belief structure. Membership functions may be triangles or trapezoids with equal or unequal positive and negative slopes to their sides. The lengths of the triangle and trapezoid bases and, hence, the slopes of their sides are determined by trial and error based on known correspondences between input information and output classification or action pairs that link to activation values of the input fuzzy sets. An expert is required to develop production rules that specify all the output actions of the system, in terms of fuzzy sets, for all combinations of the input fuzzy sets. Membership functions are defined for the output fuzzy sets using the trial and error process. The production rules are activated to different degrees through the logical product that defines membership in the output fuzzy sets.

Comparisons of the information needed to apply classical inference, Bayesian inference, Dempster-Shafer evidential theory, fuzzy logic, and other classification and identification data fusion algorithms to a target identification application are summarized in Chapter 11.

References

1. L.A. Zadeh, *Fuzzy Sets and Systems*, North-Holland Press, Amsterdam (1978).
2. L.A. Zadeh, "Fuzzy logic," *IEEE Computer*, 83-93 (Apr. 1988).
3. E. Cox, "Fuzzy fundamentals," *IEEE Spectrum*, 58-61 (Oct. 1992).
4. B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, Englewood Cliffs, NJ (1992).
5. N. Vadiee, "Fuzzy rule-based expert systems II," Chapter 5 in *Fuzzy Logic and Control: Software and Hardware Applications*, Editors: M. Jamshidi, N. Vadiee, and T. Ross, PTR Prentice Hall, Englewood Cliffs, NJ (1993).
6. G. Anderson, "Applying fuzzy logic in the real world," *Sensors*, 15-25 (Sep. 1992).
7. R.B. Lindsay, *Physical Mechanics,* 3rd Ed., D. Van Nostrand Company, Princeton, NJ (1961).
8. L. Yung-kuo, Ed., *Problems and Solutions on Mechanics*, #2029, World Scientific, River Edge, NJ (1994).
9. H. Margenau and G.M. Murphy, *The Mathematics of Physics and Chemistry*, D. Van Nostrand Comp., Princeton, NJ (1961).
10. J. Irving and N. Mullineux, *Mathematics in Physics and Engineering*, Academic Press, New York, NY (1959).
11. E.D. Rainville, *Elementary Differential Equations*, 2nd Ed., Macmillan, New York, NY (1959).
12. M.J. Horton and R.A. Jones, "Fuzzy logic extended rule set for multitarget tracking," *Acquisition, Tracking, and Pointing IX, Proc. SPIE* 2468, 106-117 (1995).
13. Y. Bar-Shalom, K.C. Chang, and H.A.P. Bloom, "Automatic track formation in clutter with a recursive algorithm," Chapter 2 in *Multitarget-Multisensor Tracking: Advanced Applications*, Editor: Y. Bar-Shalom, Artech House, Norwood, MA (1990).
14. M.P. Dana, *Introduction to Sensors and Multi-Target, Multi-Sensor Data Fusion Techniques for Detection, Identification, and Tracking, Part II*, Johns Hopkins University, Organizational Effectiveness Institute Short Course, Rockville, MD (1998).
15. J.J. Buckley and Y. Hayashi, "Fuzzy neural networks," Chapter 11 in *Fuzzy Sets, Neural Networks, and Soft Computing*, Editors: R.R. Yager and L.A. Zadeh, Van Nostrand Reinhold, New York, NY (1994).
16. T. Yamakawa, E. Uchino, T. Miki, and H. Kusanagi, "A neo fuzzy neuron and its application to system identification and prediction of the system behavior," *Proc. 2nd Int. Conf. Fuzzy Logic Neural Networks* (*IIZUKA '92*), Iizuka, Japan, 477-483 (July 17-22, 1992).
17. K. Nakamura, T. Fujimaki, R. Horikawa, and Y. Ageishi, "Fuzzy network production system," *Proc. 2nd Int. Conf. Fuzzy Logic Neural Networks* (*IIZUKA '92*), Iizuka, Japan, 127-130 (July 17-22, 1992).

18. M. Tokunaga, K. Kohno, Y. Hashizume, K. Hamatani, M. Watanabe, K. Nakamura, and Y. Ageishi, "Learning mechanism and an application of FFS-network reasoning system," *Proc. 2nd Int. Conf. Fuzzy Logic Neural Networks* (*IIZUKA '92*), Iizuka, Japan, 123-126 (July 17-22, 1992).
19. H. Ishibuchi, R. Fujioka, and H. Tanaka, "An architecture of neural networks for input vectors of fuzzy numbers," *Proc. IEEE Int. Conf. Fuzzy Syst.* (*FUZZ-IEEE '92*), San Diego, CA, 1293-1300 (Mar. 8-12, 1992).
20. H. Ishibuchi, H. Okada, and H. Tanaka, "Interpolation of fuzzy if-then rules by neural networks," *Proc. 2nd Int. Conf. Fuzzy Logic Neural Networks* (*IIZUKA '92*), Iizuka, Japan, 337-340 (July 17-22, 1992).
21. I. Requena and M. Delgado, "R-FN: A model of fuzzy neuron," *Proc. 2nd Int. Conf. Fuzzy Logic Neural Networks* (*IIZUKA '92*), Iizuka, Japan, 793-796 (July 17-22, 1992).
22. R.R. Yager, "A general approach to the fusion of imprecise information," *Int. J. Intel. Sys.*, Vol. 12, 1–29 (1997).
23. T. Denoeux, "Modeling vague beliefs using fuzzy-valued belief structures," *Fuzzy Sets and Systems*, 116(2), 167-199 (2000).