EE 5322: Intelligent Control Systems

Fuzzy Logic

Prepared by Venu Madhav

Introduction

The concept of Fuzzy Logic (FL) was conceived by Lotfi Zadeh, a professor at the University of California at Berkley, in the mid-1960's. Fuzzy Logic (FL) is a non-linear problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems. It can be implemented in hardware, software, or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL's approach to control problems mimics how a person would make decisions, only much faster.

FL is different from conventional control methods as it incorporates a simple, rule-based IF X AND Y THEN Z approach to a solving control problem rather than attempting to model a system mathematically. The FL model is empirically-based, relying on an operator's experience rather than their technical understanding of the system. For example, rather than dealing with temperature control in terms such as "Temp =500F", "Temp <1000F", or "210C <Temp <220C", terms like "IF (process is too cool) AND (process is getting colder) THEN (add heat to the process)" or "IF (process is too hot) AND (process is heating rapidly) THEN (cool the process quickly)" are used. These terms are imprecise and yet very descriptive of what must actually happen. Consider what you do in the shower if the temperature is too cold: you will make the water comfortable very quickly with little trouble. FL is capable of mimicking this type of behavior but at a very high rate.

Fuzzy Logic Architecture

The block diagram of a fuzzy controller is shown in Figure 1. The fuzzy controller is composed of the following four elements:

1) A ***Rule-Base*** (a set of If-Then rules), which contains a fuzzy logic quantification of the expert's linguistic description of how to achieve good control.

2) An ***Inference Mechanism*** (also called an "inference engine" or "fuzzy inference" module), which emulates the expert's decision making in interpreting and applying knowledge about how best to control the plant.

3) A ***Fuzzification*** interface, which converts controller inputs into information that the inference mechanism can easily use to activate and apply rules.

4) A ***Defuzzification*** interface, which converts the conclusions of the inference mechanism into actual inputs for the process.

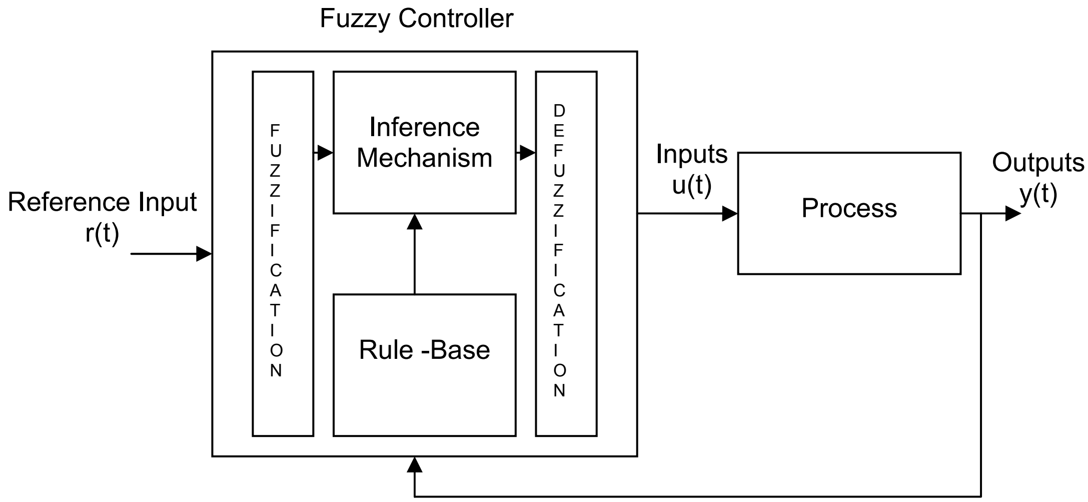


Fig 1. Block diagram of Fuzzy Controller

Terms and Definitions

The linguistic description provided by the expert can generally be broken into several parts. There will be "**linguistic variables**" that describe each of the time-varying controller inputs and outputs. Linguistic variables assume "**linguistic values**". That is, they can be described by the following values: "Neg. High, Neg. Low, Zero, Pos. Low, Pos. High" etc. Note that we are using "Neg. Low" as an abbreviation for "Negative small in size" and so on for other values. Such abbreviations keep the linguistic descriptions short yet precise. The linguistic variables and values provide a language for the expert to express his or her ideas about the control decision making process in the context of the framework established by our choice of fuzzy controller inputs and outputs.

Next, we will use the above linguistic quantification to specify a set of rules (a rule-base) that captures the expert's knowledge about how to control the plant. In particular for an inverted pendulum shown in Figure 6, we have the following rules:

- **If** angle (θ) is Neg. High and angular velocity (w) is Neg. High **Then** force on platform (F) is Neg. High.
- **If** angle (θ) is Zero and angular velocity (w) is Pos. Low **Then** force on platform (F) is Pos. Low.
- **If** angle (θ) is Pos. High and angular velocity (w) is Neg. Low **Then** force on platform (F) is Pos. Low

The convention that is followed here is, a particular linguistic variable is considered positive if it is acting towards the right and negative if it is acting towards the left.

Each of the three rules listed above is a "**linguistic rule**" since it is formed solely from linguistic variables and values. Since linguistic values are not precise representations of the underlying quantities that they describe, linguistic rules are not precise either. They are simply abstract ideas about how to achieve good control that could mean somewhat different things to different people.

The general form of the linguistic rules listed above is:

If premise **Then** consequent

As you can see from the rules listed above, the premises (sometimes called "antecedents") are associated with the fuzzy controller inputs and are on the left-hand-side of the rules. The consequents (sometimes called "actions") are associated with the controller outputs and are on the right-hand-side of the rules. The number of fuzzy controller inputs and outputs places an upper limit on the number of elements in the premises and consequents. Note that there does not need to be a premise (consequent) term for each input (output) in each rule, although often there is.

Using the above approach we can write down rules for the pendulum problem for all possible cases. With two inputs and five linguistic values for each of these, there are a maximum of $5\times5=25$ possible rules. A convenient way to list all possible rules for the case where there are not too many inputs to the fuzzy controller is to use a tabular representation as shown in Table 1.

Membership Functions

We quantify the meaning of the linguistic values using "membership functions". Consider for example Figure 2. This is a plot of a function μ versus *e(t)* (consider it to be the angle in radians that the inverted pendulum makes with the vertical on the platform) that takes on special meaning. The function μ quantifies the *certainty* that *e(t)* can be classified linguistically as "Pos. Low".

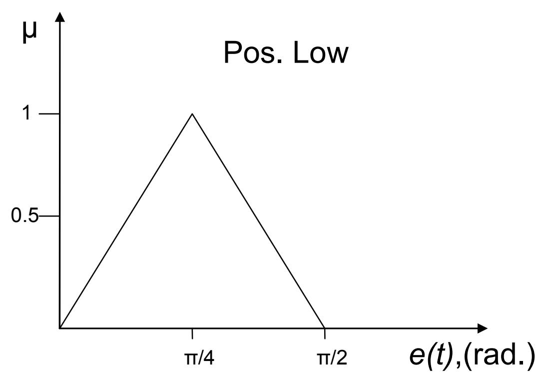


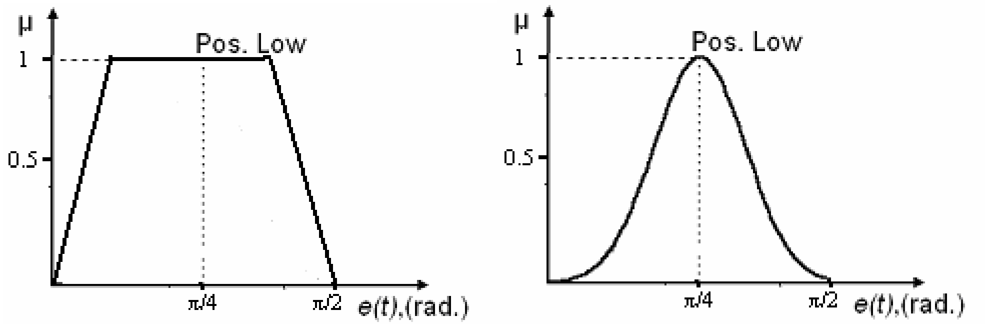
Fig 2. Membership function for linguistic value "Pos. Low"

To understand the way that a membership function (MF) works, it is best to perform a case analysis where we show how to interpret it for various values of *e(t)*:

- If $e(t)= -\pi/2$ then $\mu(-\pi/2)=0$, indicating that we are certain that $e(t)= -\pi/2$ is *not* "Pos. Low".
- If $e(t)= \pi/8$ then $\mu(\pi/8)=0.5$, indicating that we are halfway certain that $e(t)=\pi/8$ is "Pos. Low" (we are halfway certain since it could be "zero" with some degree of certainty - this value is in a "gray area" in terms of linguistic interpretation).
- If $e(t)= \pi/4$ then $\mu(\pi/4)=1$, indicating that we are absolutely certain that $e(t)=\pi/4$ is what me mean by "Pos. Low".
- If $e(t)= \pi$ then $\mu(\pi)=0$, indicating that we are certain that $e(t)= \pi$ is *not* "Pos. Low"(actually it is "Pos. High").

The membership function is not a probability density function, and there is no underlying probability space. By "certainty" we mean "degree of truth". The membership function does not quantify random behavior; it simply makes more accurate (less fuzzy) the meaning of linguistic descriptions.

It is important to recognize that the membership function in Figure 2 is only one possible definition of the meaning of “Pos. Low”; you could use bell shaped function, a trapezoid (a), a Gaussian(b), or many others.



(a) Trapezoidal **(b)** Gaussian

The set of values that is described by μ as being “Pos. Low” is called a “**fuzzy set**”. In Figure 2 while the vertical axis represents certainty, the horizontal axis is called the “**Universe of Discourse**” for the input *e(t)* since it provides the range of values of *e(t)* that can be quantified with linguistics and fuzzy sets. In summary, depending on the application and the designer (expert), many different choices of membership functions are possible.

Fuzzification is the process of obtaining a value of an input variable (e.g. *e(t)*) and finding the numeric values of the membership function(s) that are defined for that variable. For example if $e(t) = \pi/4$, the fuzzification process amounts to finding the values of the input membership functions for this. In this case:

$\mu(e(t)) = 1$

This information is then used in the fuzzy inference process using the rule-base.

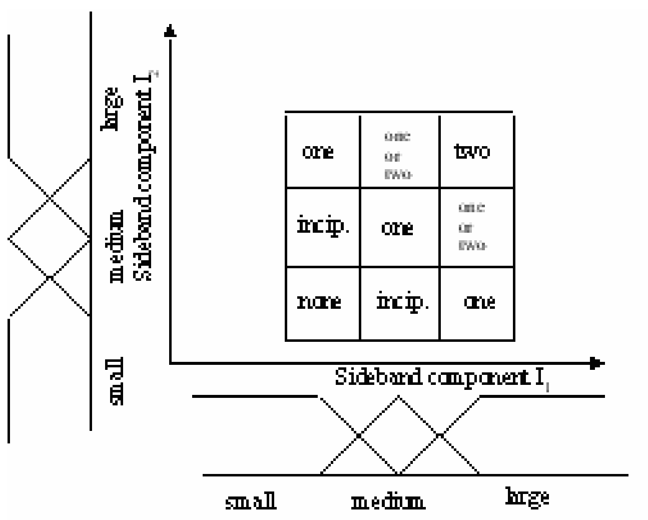


Fig 3. FL rule-base to diagnose broken bars in motor drives using sideband components of vibration signature FFT [Filippetti 2000].
Number of broken bars =none, one, two.
Incip. = Incipient fault

For example consider Figure 3 which shows the essential FL rule-base and it incorporates some information on incipient failures. Based on the approximate values of the sideband components I_1 and I_2 we can determine the state of the system. Suppose I_1 is "medium" and I_2 is "small" then we can infer that that the system has incurred an incipient fault. This means that the FL system can contain prognostic information.

Fuzzy Set Operations

Consider two fuzzy sets A and B with the membership functions μ_A and μ_B.

- *Union* of the two fuzzy sets is defined as the maximum of the two individual membership functions. This is called the *maximum* criterion.

$$\mu_{AuB} = \max\,(\mu_A\,,\, \mu_B)$$

- *Intersection* of the two fuzzy is defined as the minimum of the two individual membership functions. This is called the *minimum* criterion.

$$\mu_{A\cap B} = \min\,(\mu_A\,,\, \mu_B)$$

Intersection of two fuzzy can also be defined as the product of the two individual membership functions. This is called the *product* criterion.

$$\mu_{A\cap B} = \mu_A \times \mu_B$$

- *Complement* of a fuzzy set is defined as the negation of the specified membership function. This is called the *negation* criterion.

$$\mu_{\hat{A}} = 1 - \mu_A$$

The **inference** process is used for determining the extent to which each rule is relevant to the current situation and drawing conclusions using the current inputs and the information in the rule-base. The inference process includes the fuzzy set operations.
Graphical representation of the product implication rule with triangular and gaussian membership functions is shown in Figures 4 and 5.

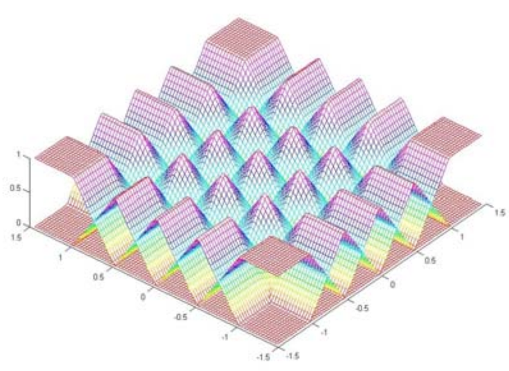


Fig 4 FL system with triangular MFs

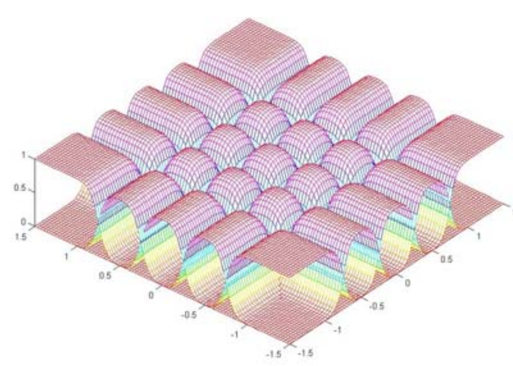


Fig 5. FL system with Gaussian MFs

Defuzzification

A number of defuzzification methods exist where each method provides a means to choose a single output based on the inference strategy employed. The most commonly used defuzzification strategy used is the "Centroid Defuzzification" which is given by the equation:

$$f(x) = \frac{\sum_{i=1}^{N} z^i \prod_{j=1}^{n} \mu_{ij}(x_j)}{\sum_{i=1}^{N} \prod_{j=1}^{n} \mu_{ij}(x_j)} \qquad (1)$$

where the control representative values are z^i and the 1-D membership functions are $\mu_{ij}(.)$. x_j are the components of the *n*-vector *x*.

In summary:

- The *Universe of Discourse* is the range of all possible values for an input to a fuzzy system.
- A *Fuzzy Set* is any set that allows its members to have different grades of membership (membership function) in the interval [0, 1].
- The *Support* of a fuzzy set F is the crisp set of all points in the Universe of Discourse U such that the membership function of F is non-zero.
- The *Crossover point* of a fuzzy set is the element in U at which its membership function is 0.5.
- A *Fuzzy singleton* is a fuzzy set whose support is a single point in U with a membership function of one.

Example

Fuzzy control system design is based on empirical methods, basically a methodical approach to trial-and-error. The general process is as follows:

1. Identify the inputs and their ranges and name them.
2. Identify the outputs and their ranges and name them.
3. Create the degree of fuzzy membership function for each input and output.
4. Construct the rule base that the system will operate under.
5. Decide how the action will be executed by assigning strengths to the rules.
6. Combine the rules and defuzzify the output.

As a simple example on how fuzzy controls are constructed, consider the following classic situation: the inverted pendulum. Here, the problem is to balance a pole on a mobile platform that can move in only two directions, to the left or to the right. The angle (θ) between the platform and the pendulum and the angular velocity (w) of this angle are chosen as the inputs of the system. The Force (F) to be applied on the platform to balance the pendulum is chosen as the corresponding output.

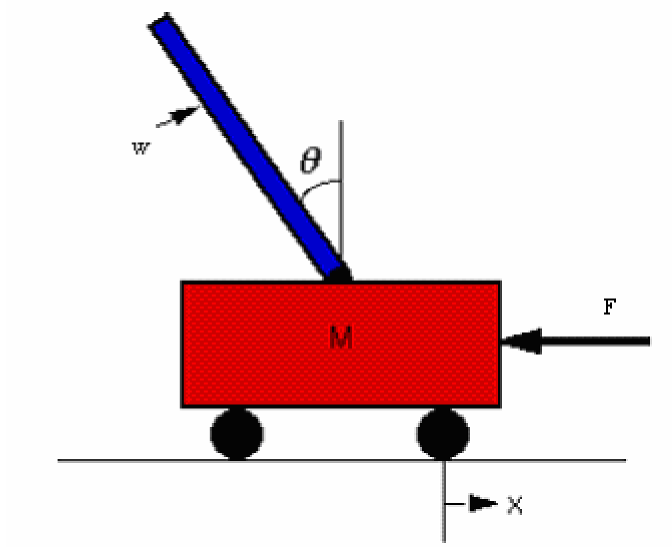
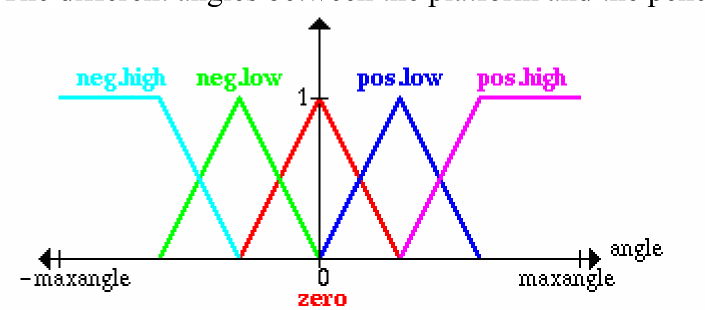


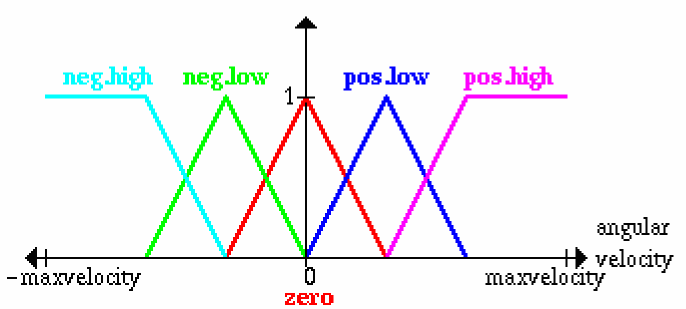
Fig 6. Inverted pendulum

STEP 1: First of all, the different levels of inputs and output (large Force, small Force etc.) of the platform are defined by specifying the membership functions for the fuzzy sets. The graphs of the functions are shown below:

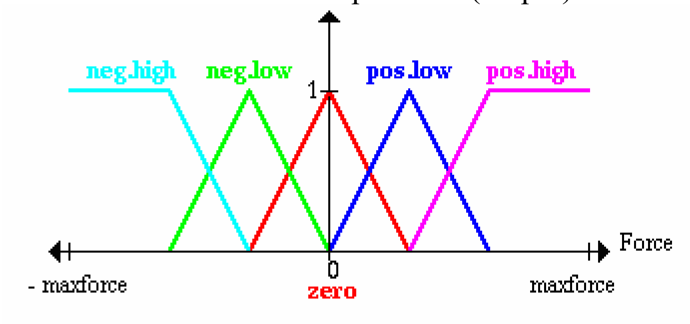
The different angles between the platform and the pendulum:



The angular velocities at specific angles:



The different forces on the platform (output):



STEP 2: The next step is to define the fuzzy rules. The fuzzy rules are merely a series of If-Then statements as mentioned above. These statements are usually derived by an expert to achieve optimum results. Some examples of these rules are:
i) If angle is zero and angular velocity is zero then force is also zero.
ii) If angle is zero and angular velocity is low then the force shall be low.

The full set of rules is summarized in the table below.

ANGLE

ANGULAR VELOCITY

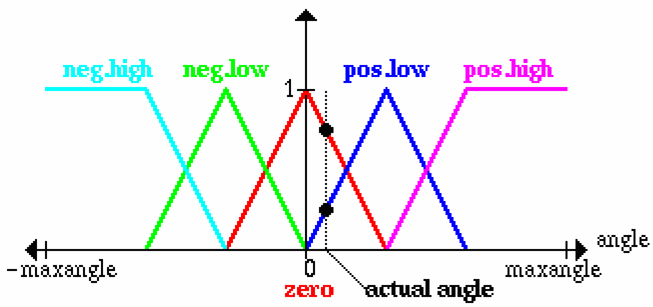
	Neg. High	**Neg. Low**	**Zero**	**Pos. Low**	**Pos. High**
Neg. High	Neg. High	Neg. High	Neg. High	Neg. Low	Zero
Neg. Low	Neg. High	Neg. High	Neg. Low	Zero	Pos. Low
Zero	Neg. High	Neg. Low	Zero	Pos. Low	Pos. High
Pos. Low	Neg. Low	Zero	Pos. Low	Pos. High	Pos. High
Pos. High	Zero	Pos. Low	Pos. High	Pos. High	Pos. High

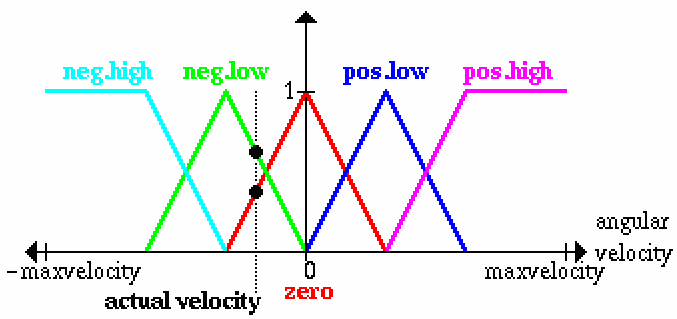
Table 1

Let us allocate values for the speed variables as follows:
Neg. High=-2; Neg. Low=-1; Zero=0; Pos. Low=1; Pos. High=2 (2)

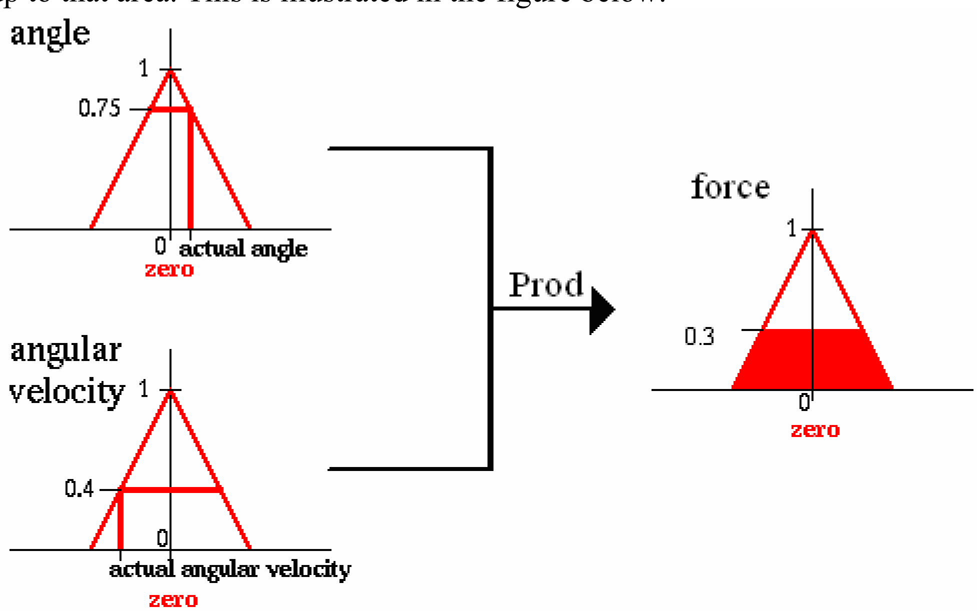
Notice the diagonal of zeros and viewing the body of the table as a matrix we see that it has certain symmetry to it. This symmetry that emerges when the rules are tabulated is no accident and is actually a representation of abstract knowledge about how to control the pendulum; it arises due to symmetry in the system dynamics.

An application of these rules is shown using specific values for angle and angular velocities. The membership values used for this example are 0.75 and 0.25 for zero and positive-low angles, and 0.4 and 0.6 for zero and negative-low angular velocities. These points are on the graphs below.

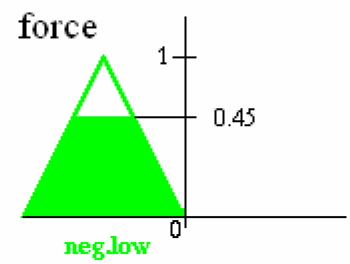


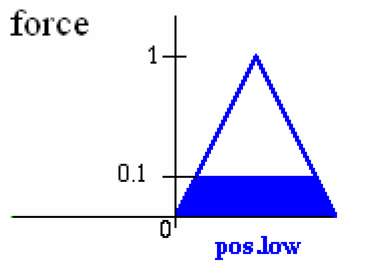


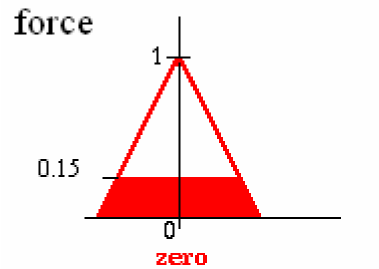
Consider the rule "if angle is zero and angular velocity is zero, the force is zero". The actual value belongs to the fuzzy set zero to a degree of 0.75 for "angle" and 0.4 for "angular velocity". Since this is an AND operation, the *product* criterion is used , and the fuzzy set zero of the variable "force" is cut at $0.75 \times 0.4 = 0.3$ and the patches are shaded up to that area. This is illustrated in the figure below.



Similarly, the *product* criterion is used for the other three rules. The following figures show the result *patches* yielded by the rule "if angle is zero and angular velocity is negative low, the force is negative low" , "if angle is positive low and angular velocity is zero, then force is positive low", and "if angle is positive low and angular velocity is negative low, the force is zero".

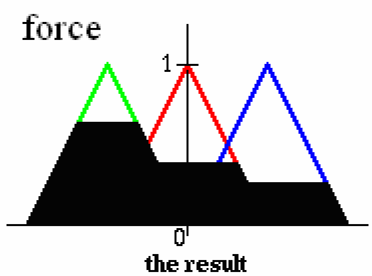






Note: The graphs are not drawn to scale.

The four results overlap and are reduced to the following figure:



STEP 3: The result of the fuzzy controller as of now is a fuzzy set (of force). In order to choose an appropriate representative value as the final output (crisp values), defuzzification must be done. There are numerous defuzzification methods, but the most common one used is the "Centroid Defuzzification" shown in eq(1).

$$f(x) = \frac{\sum_{i=1}^{N} z^i \prod_{j=1}^{n} \mu_{ij}(x_j)}{\sum_{i=1}^{N} \prod_{j=1}^{n} \mu_{ij}(x_j)}$$

From the above formula,

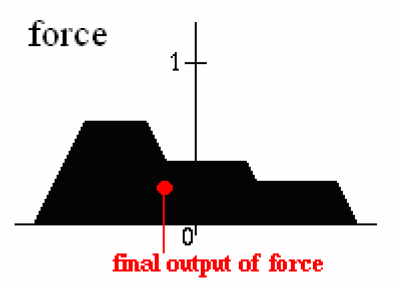
$$\text{force} = \frac{\text{"Zero"} \times (\mu_{33}) + \text{"Neg. Low"} \times (\mu_{23}) + \text{"Pos. Low"} \times (\mu_{34}) + \text{"Zero"} \times (\mu_{24})}{(\mu_{33} + \mu_{23} + \mu_{34} + \mu_{24})}$$

where $\mu_{33} = 0.75 \times 0.4 = 0.3$
$\mu_{23} = 0.75 \times 0.6 = 0.45$
$\mu_{34} = 0.4 \times 0.25 = 0.1$
$\mu_{24} = 0.6 \times 0.25 = 0.15$

Substituting the values from the table and eq(2) we get,

$$\text{force} = \frac{(0 \times 0.3) + (-1 \times 0.45) + (1 \times 0.1) \times (0 \times 0.15)}{(0.3 + 0.45 + 0.1 + 0.15)}$$

= - 0.35 N (assuming SI units)
Thus the resultant force is applied such that the platform moves towards the left.



Relation between Fuzzy logic and Neural Networks

Many researchers focused on combining neural networks and fuzzy logic systems, such as neuro-fuzzy systems, or fuzzy neural networks. While fuzzy logic uses approximate human reasoning in knowledge-based systems, the neural networks aim at pattern recognition, optimization and decision making. A combination of these two technological innovations delivers the best results.

It has been proved that rectangular wave activation function neural networks can represent nonlinear neural networks. Based on this result, we can prove that fuzzy logic systems and neural networks are equivalent, essentially under some restriction. First of all, we introduce interpolation representation of fuzzy logic systems. The antecedents of inference of a fuzzy logic system are the base functions of interpolation and the consequents of inference only relate to their peak values but not to the shape of the membership functions.

The interpolation representation of the FL system can be given by the following expression:

$$z = F(x, y) = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{ij}(x, y) z_{ij}$$

Comparing this with eq(1) we can see that R_{ij} is a function of the membership functions.

Similarly it is well known that the relation between the inputs and outputs of the neural network is expressed as follows:

$$z = \varphi\left(\sum_{i=1}^{n} w_i x_i - \theta \right)$$

where θ is the threshold and w and x represent the weights and the inputs respectively.
Pay attention to the simple fact that Σ plays a role in the synthesizing of n-dimensional Euclidean space and φ plays a role in activating signals. "φ" is called a signal activator or an activation function.

Now we create a feed-forward neural network shown in Fig. 7, where h_{ij} and h are the neurons (regarded as functions of several variables), holding

$h_{ij}(u,v) = uv(i = 1,2,...,n, j = 1,2,...,m)$

i.e., considered as hyperbolic functions, and h is given as Σ; that is

$$h(u_{11}, u_{12}, ..., u_{mn}) = \sum (u_{11}, u_{12}, ..., u_{mn})$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} u_{ij}$$

Thus the output of the network is as follows:

$$z = f(x, y)$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{m} A_i(x) B_j(y) z_{ij}$$

Here $A_i(x)$ and $B_j(y)$ can be considered equivalent to the individual membership functions of the FL system. Thus,

$$f(x, y) = F(x, y)$$

Hence a FL system is approximately equivalent to the feed-forward neural network.

We can notice that the following similarities exist between the two systems:

1) Output characteristics of the hidden neurons of the NN and the inference strategy employed on the membership functions are similar.

2) Multiply-add operations of neurons are equivalent to MAX-MIN or PRODUCT operations of the fuzzy sets.

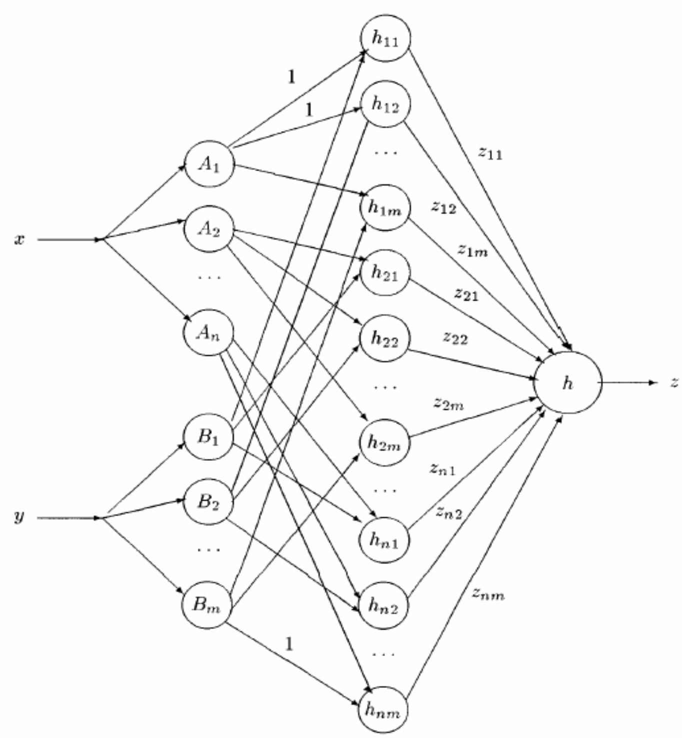


Fig. 7. Three layer feedforward neural network with two inputs and one output.

Application Areas of FL

Fuzzy systems have been used in a wide variety of applications in engineering, science, business, medicine, psychology, and other fields. For instance, in engineering some potential application areas include the following:

- *Aircraft/ spacecraft*: Flight control, engine control, avionic systems, failure diagnosis, navigation and satellite control.
- *Automated highway systems*: Automatic steering, braking and throttle control for vehicles.
- *Autonomous vehicles*: Navigation of ground and underwater vehicles.
- *Manufacturing Systems*: Scheduling and deposition process control.
- *Power industry*: Motor control, power control/ distribution and load estimation.
- *Robotics*: Position control and path planning.