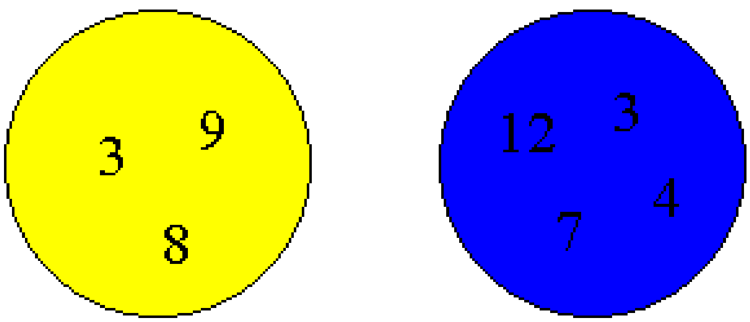
What is Fuzzy Logic?

Fuzzy Logic is a very powerful concept when applied to control systems. It allows control system designers to greatly reduce the number of rules necessary to deploy a desired control strategy. Additionally, it allows rules to be crafted in a very linguistic syntax. Finally, if properly designed, instabilities and hysteresis can be reduced or eliminated.

Classical Set Theory:

Fuzzy Logic is a mathematical technique that greatly enhances the capability of the classical set theory. If you recall from simple set theory, problems were defined by grouping items into sets. For example, we may have six numerical members which can be categorized into two different sets for some reason. For the purpose of this example let's assume the following arrangement:

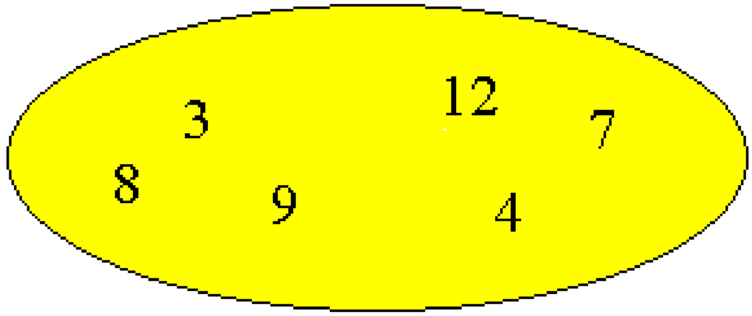


Set A **Set B**

Mathematically we write the fact that set A contains the numbers 3, 8 and 9 as {**A:** 3, 8, 9}. Likewise, set B can be written {**B:** 3, 4, 7, 12} since it contains the numbers 3, 4, 7, and 12.

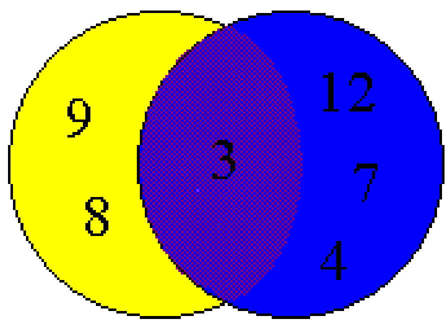
Sets can be combined and compared several different ways. We will look at two methods here, the *UNION* and the *INTERSECTION*.

The *UNION* of two sets is a list of all members contained in both sets. Should a member appear in both sets, the *UNION* operator reports only one instance of the member. In our example above, both sets contain the number three; however, the *UNION* of the sets A&B (written "$\mathbf{A \cup B}$") is { $\mathbf{A \cup B}$: 3, 4, 7, 8, 9, 12}.



Union: $A \cup B$

The *INTERSECTION* of two sets is a list of all members both sets have in common. In our example above, the *INTERSECTION* of the sets A&B (written "$A \cap B$") is { $A \cap B$: 3}.



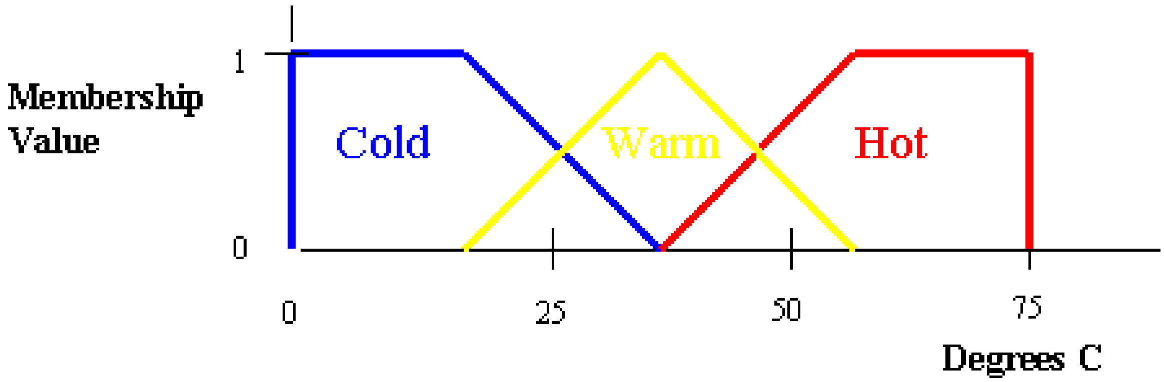
Intersection: $A \cap B$

Boolean Logic:

Classical set theory utilizes a type of logic known as Boolean logic. If you are familiar with computer programming you will no doubt recognize this term. In Boolean logic, conditions are evaluated to be either TRUE or FALSE. There are no intermediate values. In classical set theory members either have full membership or no membership in a given set or operation on sets. That is, the "degree of membership" is restricted to either TRUE or FALSE or 1 and

Fuzzy Logic and "Truth Values":

Fuzzy Logic is a mathematical technique that greatly enhances the capability of classical set theory by allowing the "degree of membership" or "truth value" to range over the interval of 0 to 1. Sets in fuzzy logic systems typically describe ranges of operations and are named using linguistic adjectives such as "slow", "medium" or "fast". The degree of membership describes how "slow" or how "fast" a particular value is. For example, we may have a temperature range (0-75 Deg. C) that is described using fuzzy sets as follows:



In the example above, temperature values from 0 to about 20 are defined as being "cold" without question. In the overlap region from about 20 to 37.5 the temperatures are described as being somewhat "cold" and somewhat "warm". As can be seen from the defining contours, the degree of "warmness" increases with increasing temperature while the degree of "coldness" decreases. We can continue the discussion in a similar fashion as we transition from "warm" to "hot". Finally, there is a region that is defined to be entirely "hot" from about 55 degrees to 75 degrees.

To understand how this concept of "degrees of membership" is useful let's back up and consider how a Boolean based system behaves. Boolean logic uses the *AND* and the *OR* operators to perform the *INTERSECTION* and the *UNION* functions respectively. We use "truth tables" to describe these functions:

AND Operator

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

OR Operator

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

If you have never seen a truth table, consider row one in the *AND* operator table. It says that if we consider two conditions together using the *AND* function and both of those conditions are false, then our conclusion, or output, is also false. For example, I may have a statement such as:

IF (Temperature is HOT) AND (Humidity is HIGH) THEN Cooler is ON

According to the first row in the table, if we have a situation where Temperature is not HOT and Humidity is not HIGH then our Cooler remains OFF. As we consider the other rows in the table we see that only when both Temperature is HOT and Humidity is HIGH does our Cooler turn ON.

Looking at the patterns in the *AND* table we notice that in each case the output is equal to the lower of the two inputs under consideration. Looking at the *OR* table we notice that the higher of the two input values is used to generate the output value. This same approach is used to resolve the *AND* and *OR* operators in fuzzy systems. Using our example above:

IF (Temperature is HOT) AND (Humidity is HIGH) THEN Cooler is ON

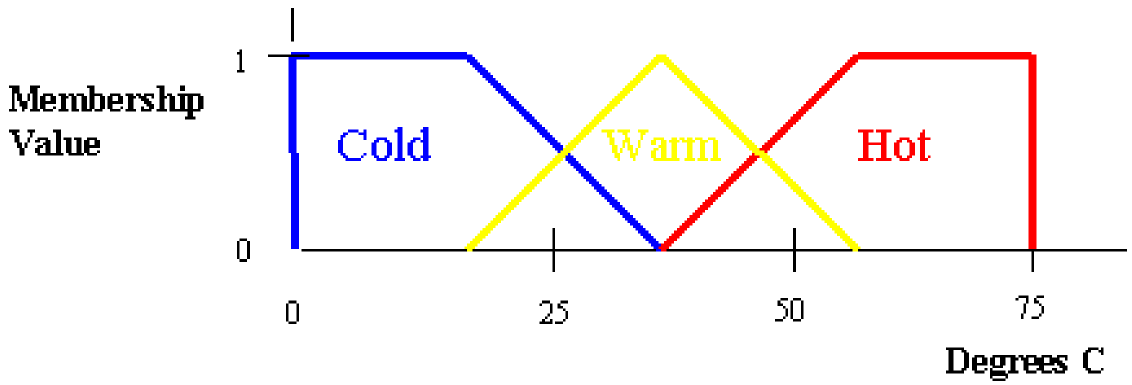
……and supposing that Temperature is HOT with a degree of membership equal to 0.3 and Humidity is HIGH with a degree of membership equal to 0.6, then Cooler has an ON value equal to 0.3. One of the values of fuzzy logic becomes very clear here when we recognize that we can use a single rule to provide control to a variable mode cooler. In our example here, the value of ON can be mapped such that it drives a variable speed drive or some other variable mode device. The behavior of the system can be modified and "tuned" by simply changing the geometries of the fuzzy sets that define Temperature and Humidity.

The Fuzzy Process:

So far we have learned that fuzzy logic allows conditions to take on values over the interval of 0 to 1. We have also learned that using the *AND* and the *OR* operator in a fuzzy system is very akin to the way we used them in Boolean logic, although we may not have recognized it.

In real-world applications we typically deal with "crisp" numbers. Fuzzy systems are very useful in that they provide a layer of abstraction similar to the spoken language when implementing a control strategy. As was the case in our earlier example, we can discuss the general behavior of a cooling system very easily. We know that when the temperature increases the cooling system needs to be ramped up as well. Likewise, as humidity increases we know that the cooling system is going to have to work harder to maintain the desired environment. However, gauges, meters and control devices don't understand such concepts as "cold" and "hot" or "slow" and "fast", so there is a mapping process that needs to take place to convert "crisp" values into fuzzy values and vice versa. We call these mapping operations "fuzzification" and "de-fuzzification".

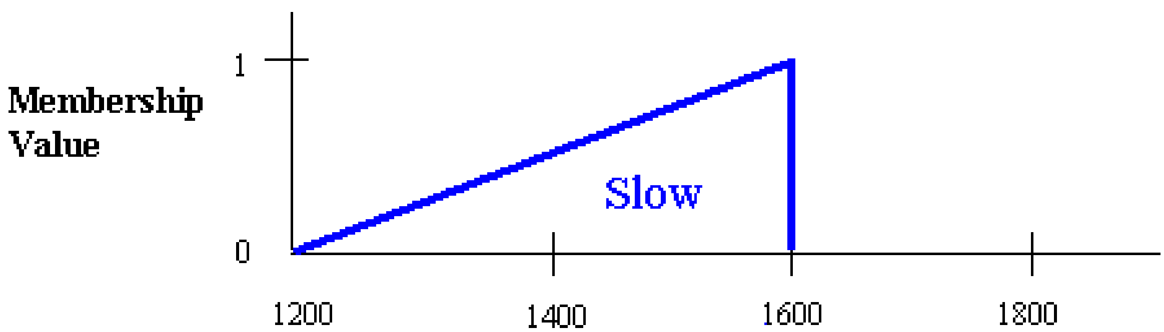
Fuzzification: Mapping input variables to their respective fuzzy sets is very straight forward. Revisiting our earlier example, suppose we have a range of temperatures over which we assign three fuzzy sets: cold, warm, and hot.



Again, from 0 to about 20 degrees The Temperature is cold with a truth value of 1. As we move into the transition zone, Temperature is still cold but with an ever decreasing truth value. At the same time, Temperature is also warm with a truth value that continues to increase until about 37.5 degrees. Between 37.5 and 55 degrees we transition again into decreasing values of warm and increasing values of hot. Finally, beyond 55 degrees Temperature is entirely hot (truth value is 1).

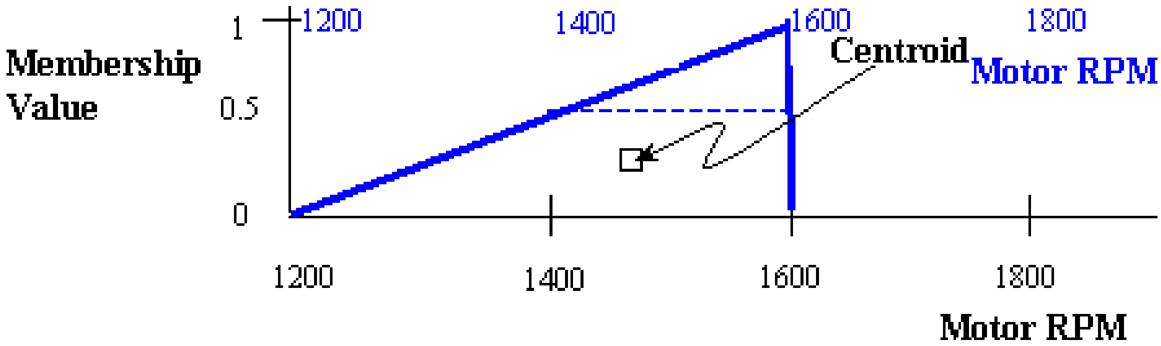
De-Fuzzification: Once a fuzzy result is produced, it typically needs to be broken down into a crisp value to be useful. This "de-fuzzification" process is a bit more involved than the "fuzzification" process but still easy to understand.

Suppose we are given a fuzzy result that is "slow" with a truth value of 0.5 and the following fuzzy function definition for "slow":



To resolve the given information into a crisp value we take the following steps:

- Define a region bounded by the fuzzy function with y-values less than the truth value.
- Calculate the centroid (center of area) for this region.
- The x-coordinate of the centroid is the crisp value from this fuzzy set.



In our example the crisp output is 1444 RPM.

Multiple Rules: We have been dealing with single rule scenarios to this point. What happens when multiple rules are introduced into a fuzzy system? Multiple rules can lead to multiple outputs so there needs to be a method of combining these multiple outputs to derive a singular crisp value when needed.

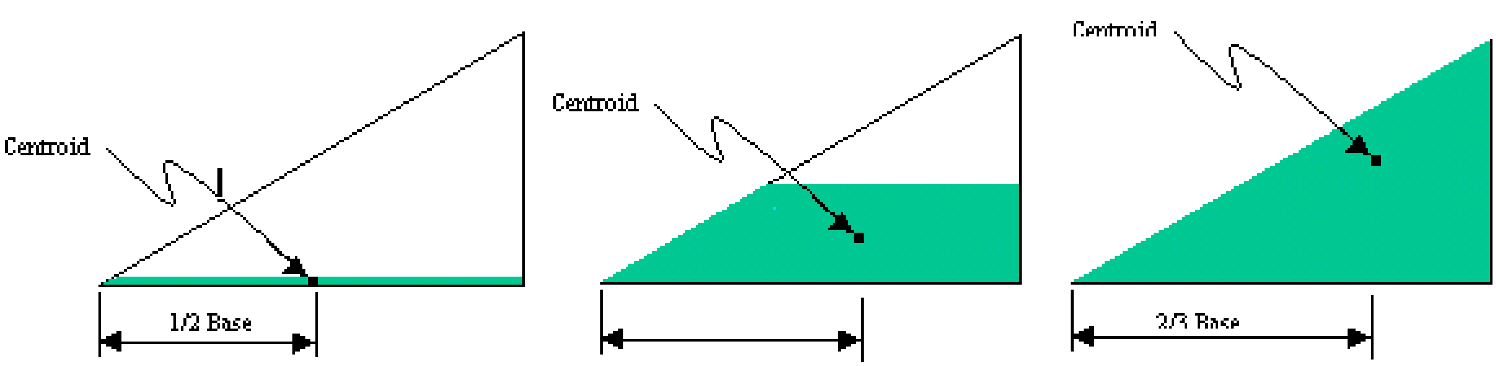
One method of combining multiple outputs is to use a weighted average of the various outputs. In the example above we not only calculated the crisp value but there was an area, or a weight, also associated with that value. That area, of course, is the region bounded by the fuzzy function and below the y constraint provided by the truth value.

If two or more solution "components" are created from the multiple rules, we can consolidate these various contributions to the final solution by taking a weighted average of the components. To see how this might work please download our "Fuzzy Logic Example".

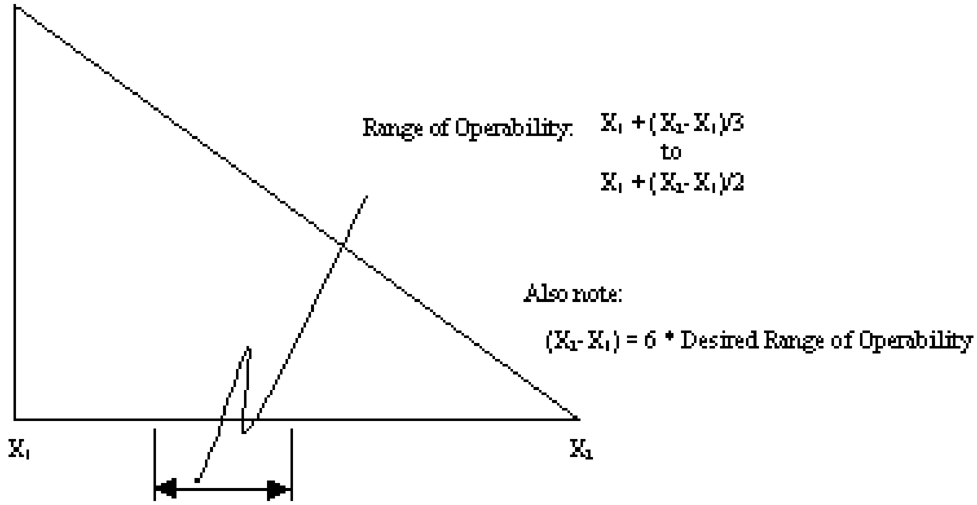
Understanding How Different Geometries Behave:

As you have no doubt gathered by now, fuzzy rule bases can indeed be much simpler than their crisp counterparts; however, this simplicity comes at the cost of understanding how the geometries of the fuzzy functions affect the control strategy. Input sets are very straight forward as the mapping is very intuitive. With very little practice the behavior of inputs sets becomes second nature. Outputs set, on the other hand, merit more discussion.

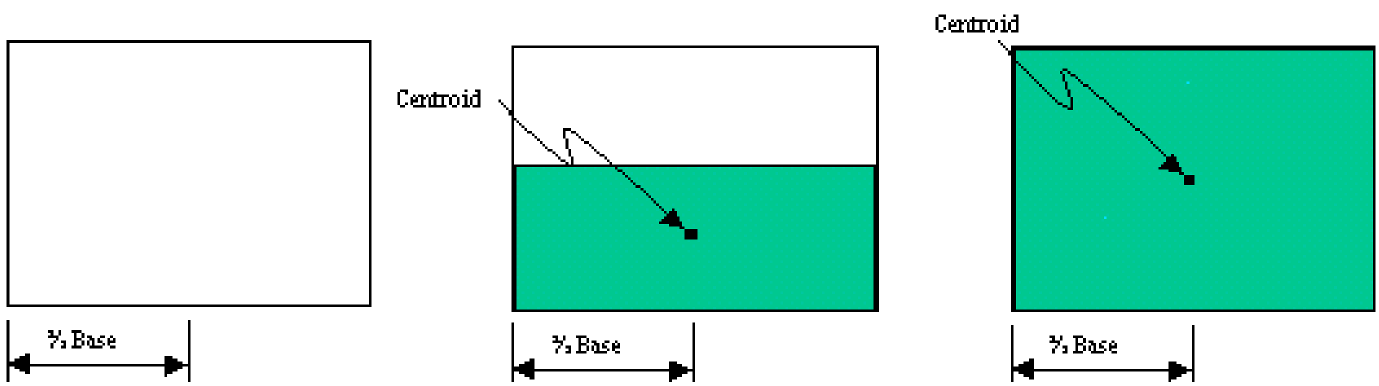
Notice how the response of a right triangle behaves as the truth value runs from 0 to 1:



The behavior of a right triangle used to define an output can be generalized as:

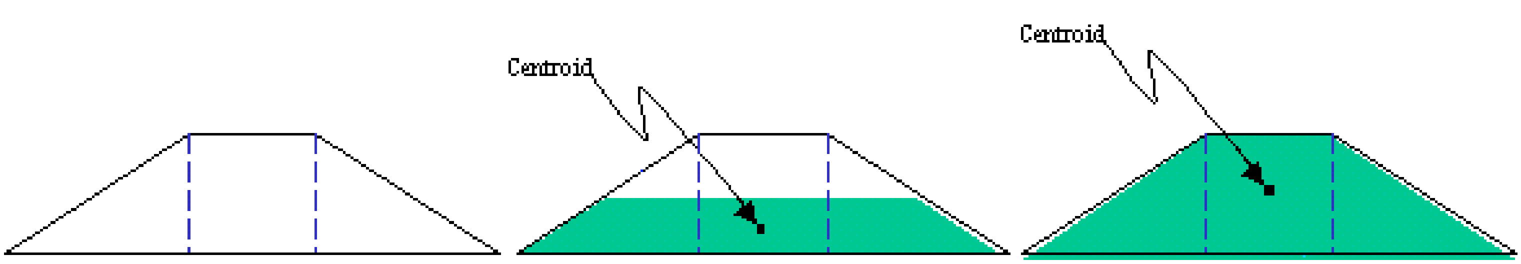


Let’s conduct a similar study for a rectangular contour:

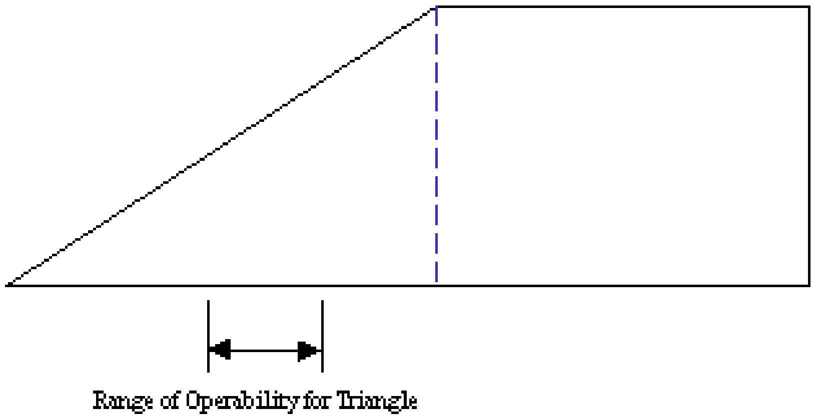


Notice that output is independent of width, height and truth value. Its value is always the value of the midpoint of the base. This shape obviously has little value in a fuzzy system.

This same behavior is exhibited with any geometry that has symmetry about a vertical axis:

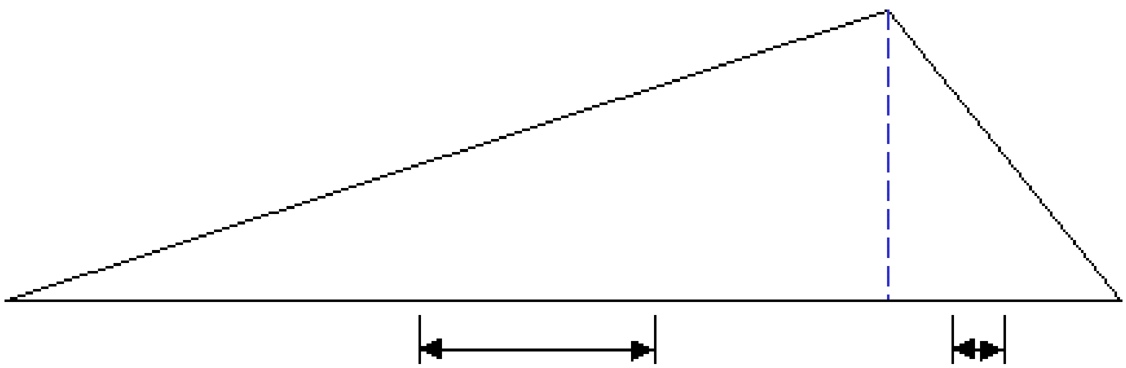


What happens when we combine a rectangular section with a right triangle?

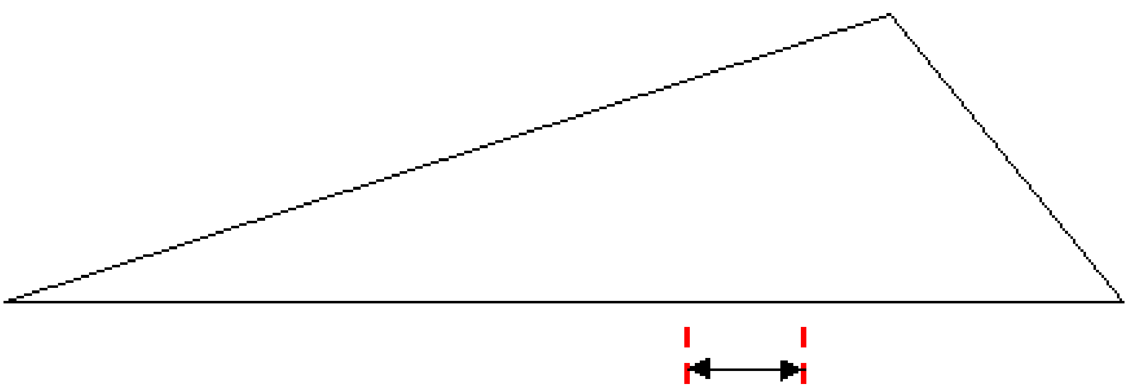


In this case, the addition of the rectangular portion serves only to shift the range of operability to the right. The simpler solution would place the triangle to the right by ½ width of the rectangular portion.

What happens when we combine two right triangles of different sizes?



In this case, there is a range of operability associated with each right triangle. The combined range of operability can be calculated by considering the average between the two 1/2 base values for each right triangle and the average between the two 2/3 base values for each right triangle. The region between these two averages is the combined range of operability.

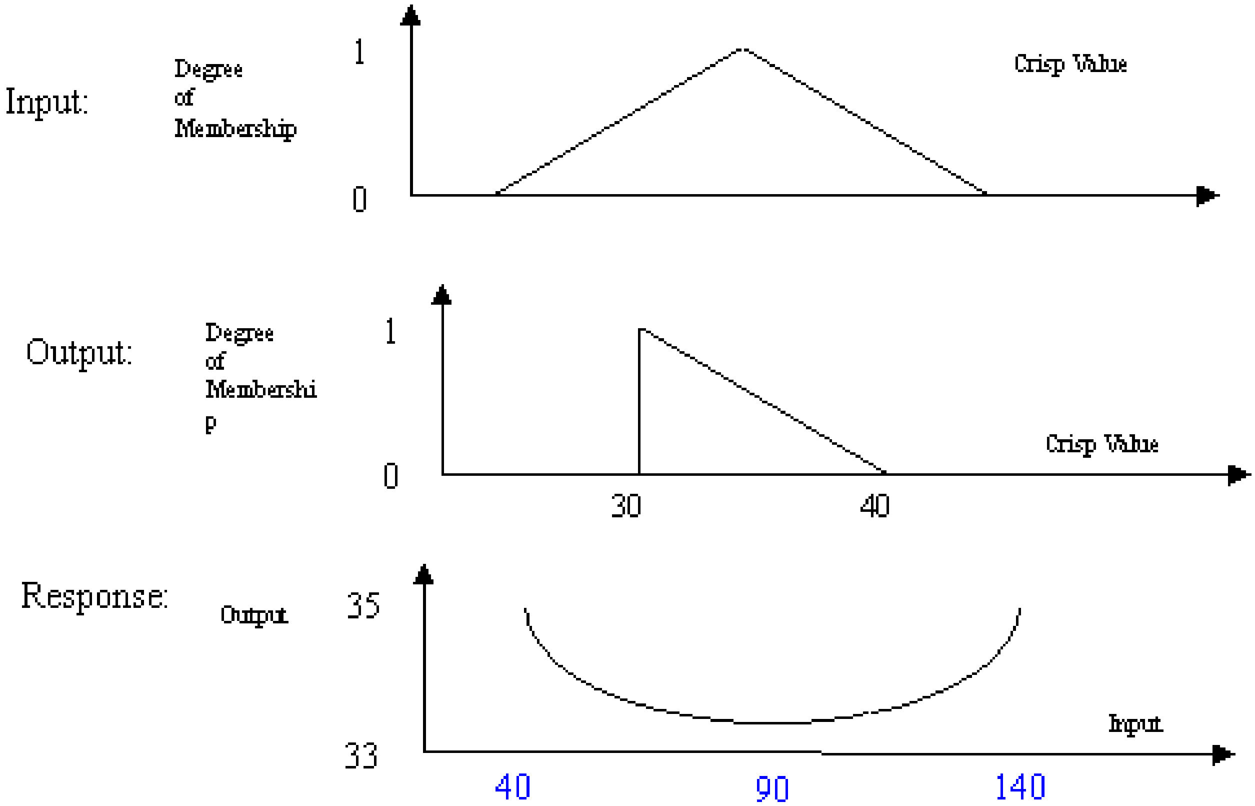


Again, the simpler implementation here would be to construct the appropriate right triangle to provide the same range of operation.

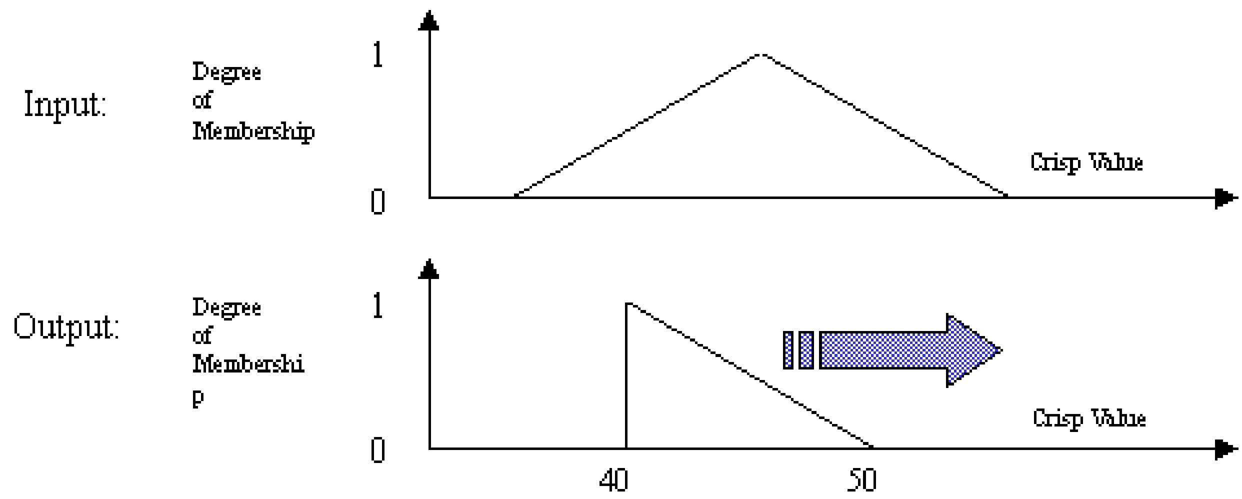
Exercises:

The following exercises are designed to emphasize the concepts discussed above. You can practice building fuzzy sets and fuzzy rules by downloading the "Fuzzy Learner Module" from this web site. This training module is limited in scope but does provide a medium on which the basics of fuzzy logic can be explored. To get a demo copy of the KnowledgeScape expert system, please contact your local KnowledgeScape representative or contact us here at KnowledgeScape Systems (see the Contact Us page of this site).

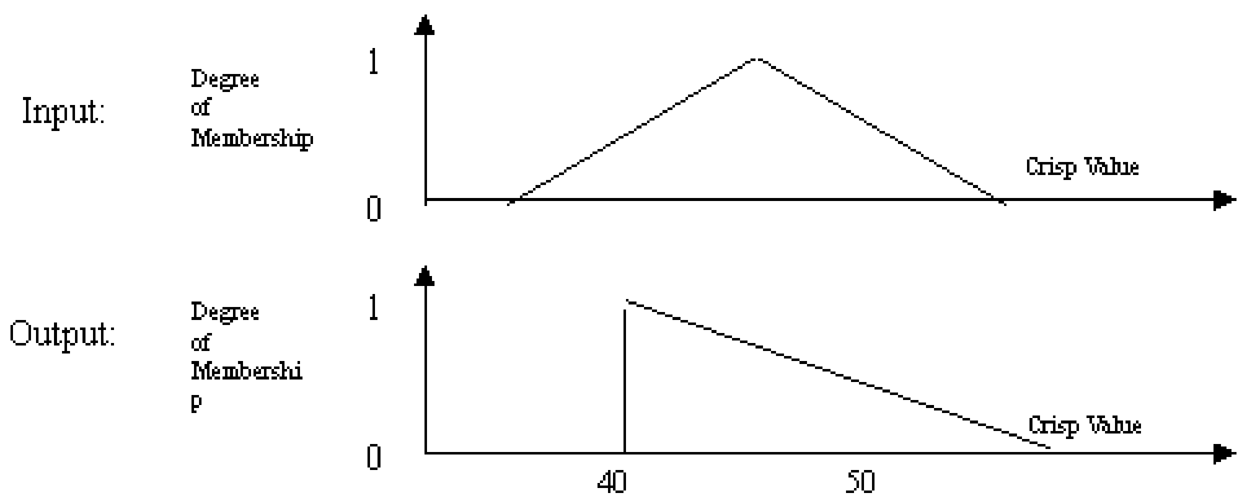
Exercise 1: Build the following fuzzy system and confirm that the response is as shown below:



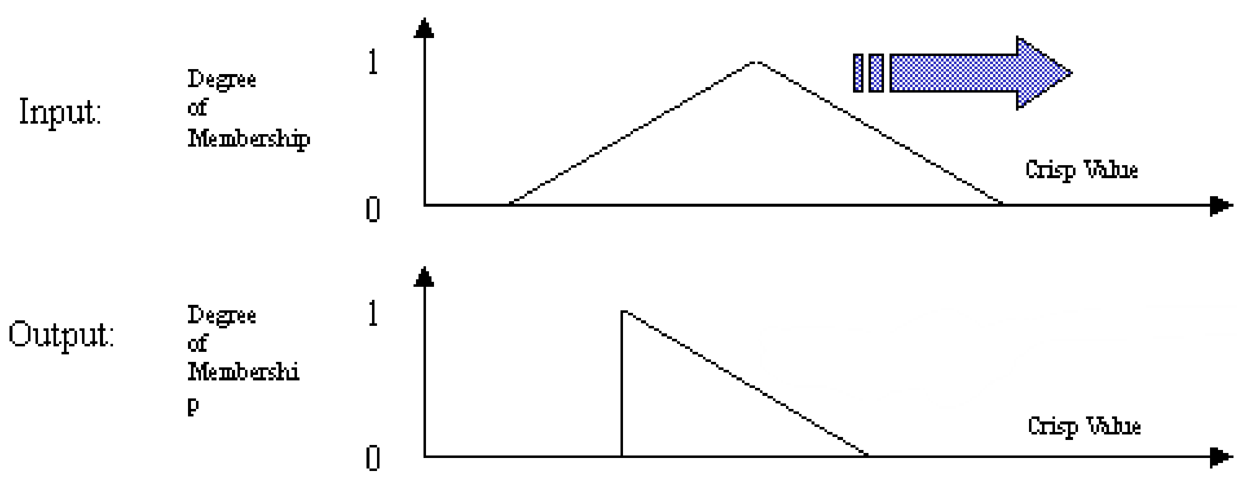
Exercise 2: What happens to the system if the output set is shifted to the right?



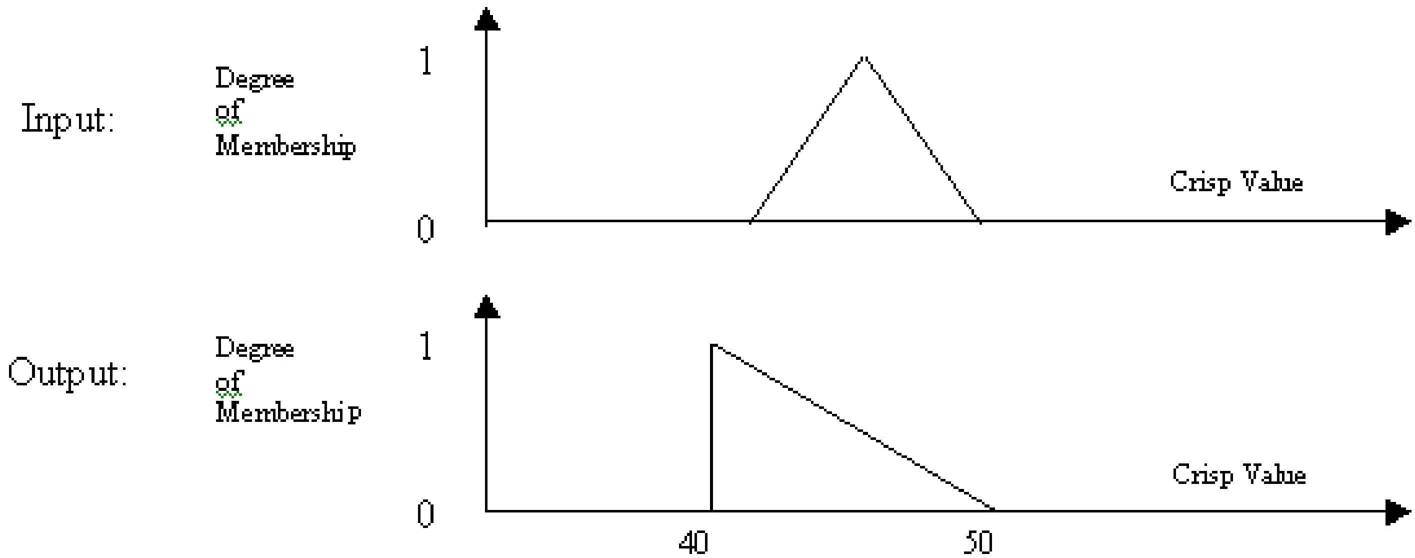
Exercise 3: What happens to the system if the output set is stretched to the right?



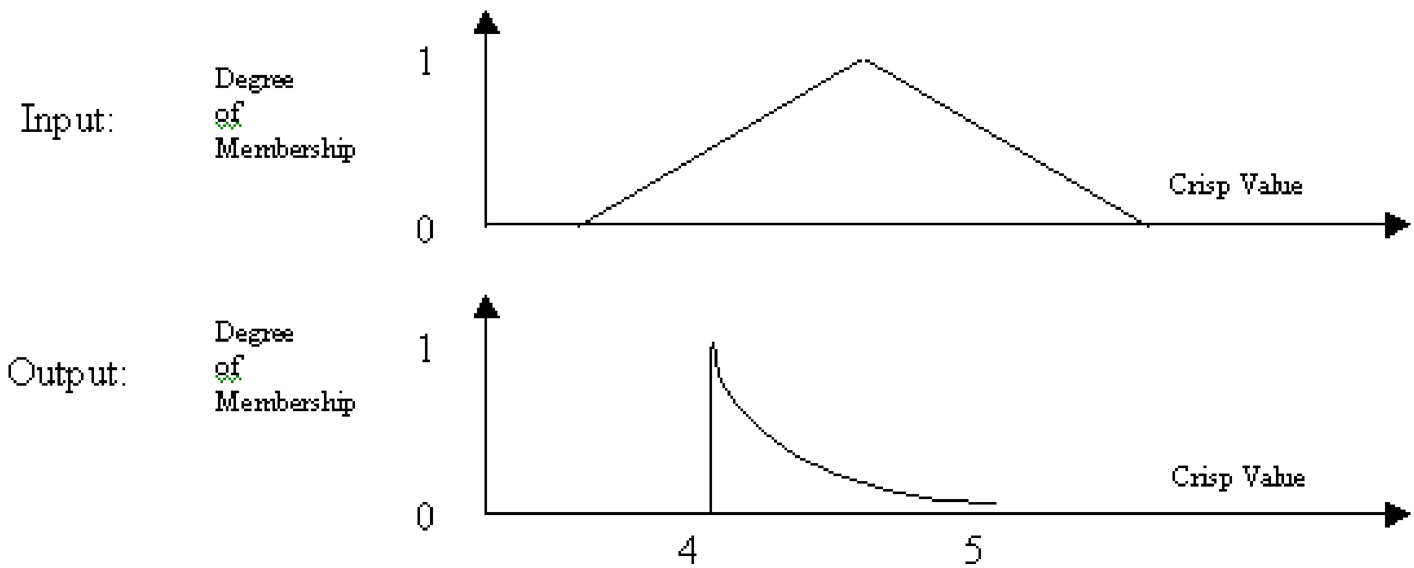
Exercise 4: What happens to the system if the input set is shifted to the right?



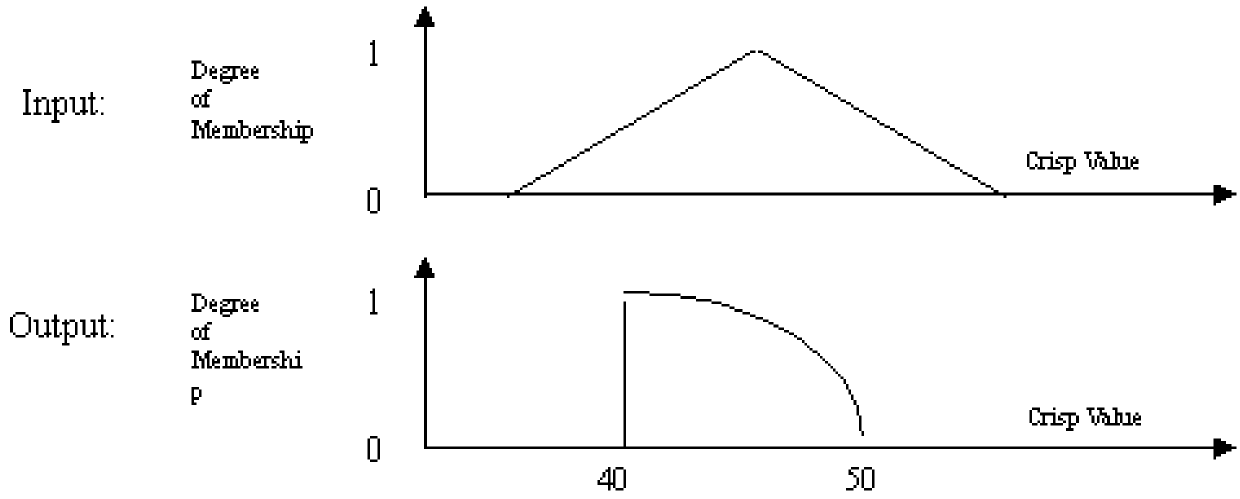
Exercise 5: What affect does compressing the input triangle have?



Exercise 6: What is the effect of contorting the hypotenuse of the output triangle?



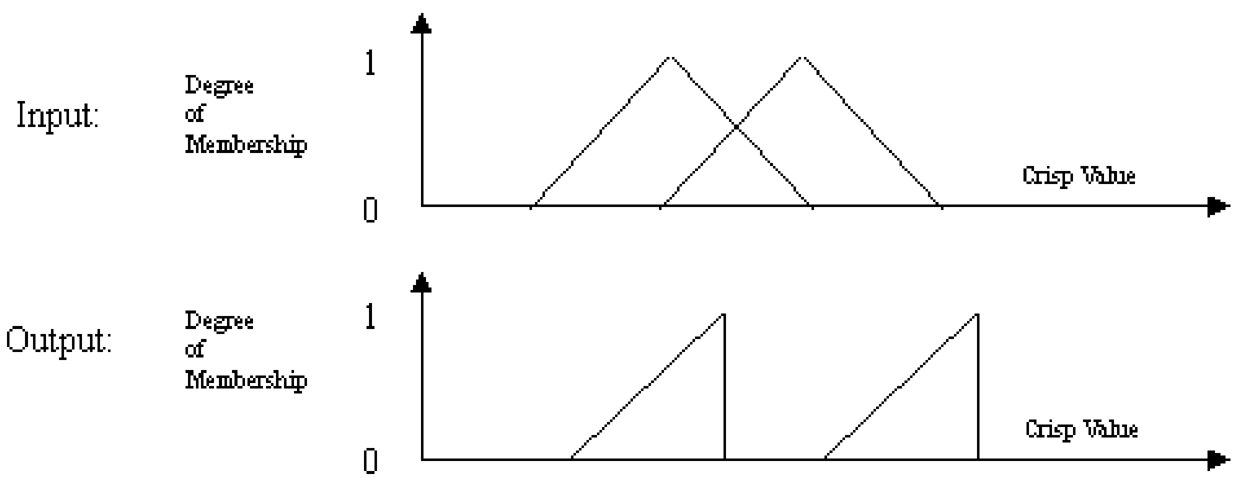
Exercise 7: What is the affect of contorting the hypotenuse of the output triangle?



Exercise 8: Qualitatively determine the response curve generated by applying the following rules to the fuzzy sets below:

-IF Input IS Input1 THEN Output is Output1

-IF Input IS Input2 THEN Output is Output2



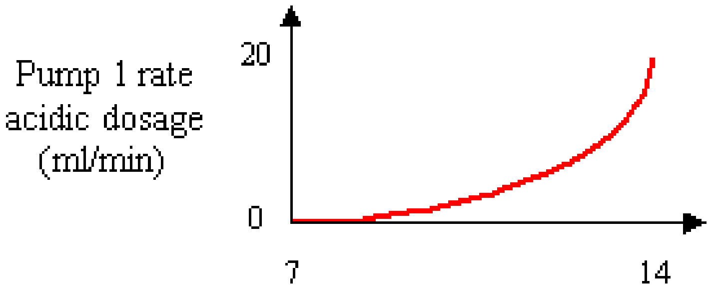
Exercise 9: Using KnowledgeScape, redefine the fuzzy membership functions in the cooling demo such that the entire motor operating range is utilized (0 to 3600 RPM).

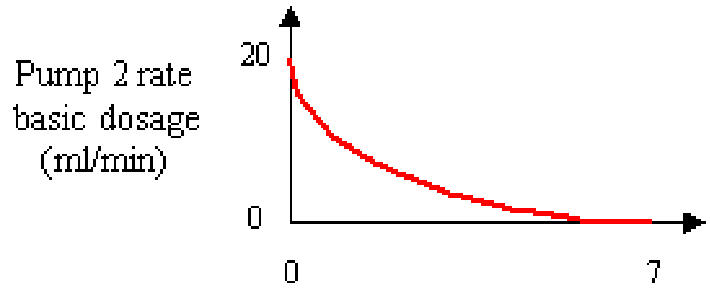
Exercise 10: Using KnowledgeScape, build a fuzzy control system that stabilizes the pH in a process using a pH probe capable of monitoring pH over a range of 0 to 14 and two variable speed dosage pumps. Both pumps operate over a range of 0 - 20 ml/min and one is used to inject a basic solution while the is used to feed an acidic solution. The control objective is to stabilize the pH to normal (7).

Use three fuzzy sets to describe the acidic end of the pH spectrum (0-7) and three fuzzy sets to describe the basic end of the spectrum (7-14).

A secondary objective is to minimize the control energy; viz., utilize the variable speed capability of the pumps so as to minimize the loading on the pump motors. Remember that pH scale is based on the log function such that as you move farther from normal (7.0) your acid/basic concentrations increase exponentially.

The response of the system should look something like this:





Design and implement a set of rules which will test the response of your system.

Exercise 11: Upgrade the metering pumps of your previous example to operate over a range of 0-40 ml/min.

Exercise 12: Using KnowledgeScape, build a fuzzy control system that controls the turbine speed of an air aspirator used in a biological reduction process. The system includes two inputs and a single output. The inputs are liquid level and dissolved oxygen concentration.

The tank side water depth can range from 5 to 14 feet but is typically operated between 10 and 11. The D.O. probe can detect D.O. levels from 0.0 mg/l to 100.0 mg/l. The turbine is driven by a VFD through a gearbox such that the operating range of the turbine is 0 to 60 RPM.

The control objective is to maintain D.O. levels at 60.0 mg/l ± 0.5 mg/l while an uncontrollable process variability causes the tank levels to fluctuate over the range of 10 to 11 feet. The strategy should be able to handle occasional upset conditions which cause the tank levels to rapidly fall outside this range into the tank limits of 5 to 14 feet SWD. Additionally, biological loading can change nearly instantaneously on occasion and can cause the D.O. level to spike dramatically upward or downward. The strategy should stabilize the D.O. level as quickly as possible.

Design and implement a set of rules which will test the response of your system. D.O. level is dependant on time, liquid level and turbine speed according to the following relationship (based on a 5 second sampling period):

DO_reading = DO_reading*.9 + (0.025/Tank_level)*(Turbine_rpm)*(Turbine_rpm)

RETURN TO TOP