Managing Uncertainty in the Real World

2. Fuzzy Systems

Satish Kumar

In Part 1 of this article we understood the idea of a fuzzy set. We also reviewed simple operations such as fuzzy set intersection, union, and complementation, and related these to geometry. In Part 2, we extend the idea of fuzzy sets to explain how we may develop systems that reason with fuzzy sets using fuzzy logic. The idea of extending fuzzy logic to fuzzy engineering and fuzzy function approximation is explored with a working example. It is hoped that readers will be provided with enough motivation to appreciate the use of fuzzy set theory for real world applications.

Satish Kumar is a Reader in the Department of Physics and Computer Science and Applications, Dayalbagh Educational Institute. His current research interests are in the areas of fuzzy-neural systems, knowledge representation in connectionist architectures and some theoretical aspects of neural networks.

Part 1. Fuzzy Sets, *Resonance*, Vol.4, No.2, pp.37–47, 1999.

Fuzzy Rules for Approximate Reasoning

We now turn our attention to the development of a logical framework that allows us to design reasoning systems based on fuzzy set theory. We reason on an every day basis with imprecise concepts – without even realizing it. Consider the following statements which we consider everyday.

- If the weather conditions remain favourable we usually complete harvesting in about a months time.
- If the traffic density is high then keep the green light on longer.
- It usually takes about half an hour to reach the station in normal traffic and weather conditions.

Examples such as these abound and come naturally to mind. Reasoning with vague concepts is natural. And, we now know that each concept can be modelled by a fuzzy set, given the inherent elasticity in the meaning of everyday concepts. With this idea one can develop conditional statements which could form the basis of a reasoning framework. *Fuzzy logic is reasoning with fuzzy sets.*

Fuzzy logic is reasoning with fuzzy sets.

As an example, consider the problem of controlling the pressure and temperature within a boiler. We present two fuzzy rules that could form a part of the rule base of a fuzzy controller:

*R*1: If (temperature is BELOW-AVERAGE) and (pressure is BELOW-AVERAGE) then heater-power is MEDIUM-HIGH and valve-opening is MEDIUM-LOW.

*R*2: If (temperature is LOW) and (pressure is LOW) then heater-power is HIGH and valve-opening is SMALL.

In these rules, multiple antecedent conditions are combined together using a conjunction or logical "AND" operator. Notice that:

1. Linguistic variables such as temperature and pressure, can take on values such as HIGH, LOW, YOUNG, OLD. These values need to be defined at a superficial level and assigned membership functions over a specific UOD at a deeper level;

2. In general, fuzzy rules are expressions of propositions of the canonical form:
   If X_1 is R_1 and X_2 is $R_2 \cdots$ and X_n is R_n then Y_1 is S_1 and Y_2 is $S_2 \ldots$ and Y_m is S_m.

Approximate models that use fuzzy rules such as these, are different from classical, imprecise or uncertain (probabilistic) models. To see this difference clearly we re-frame one of the above rules in three different forms.

Classical rule: If (temperature = 80°C) and (pressure = 1 ATM) then (heater-power is 50W) and (valve-opening is 0.5mm)

Imprecise rule: If (temperature ≥ 75°C) and (pressure ≤ 1 ATM) then (heater-power ≤ 60W) and (valve-opening ≤ 1 mm)

Uncertain rule: If (temperature ≥ 75°C) and (pressure ≤ 1 ATM) then prob(heater-power = 50W) = 0.9 and prob(valve-opening = 0.5mm) = 0.9.

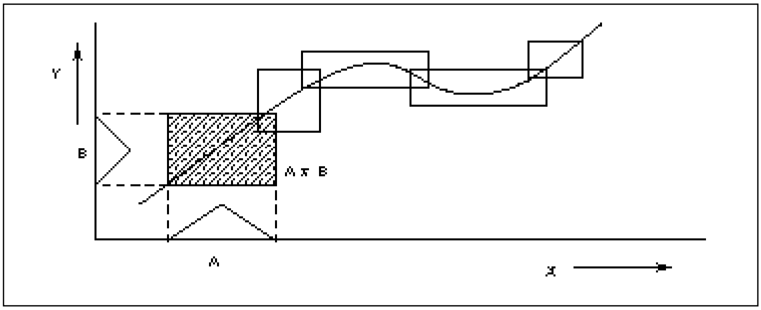


Figure 1. Fuzzy rules form patches in a geometric view. A well designed fuzzy system will attempt to 'cover' the function that governs the behaviour of a system. Such behaviour is almost always characterised by input-output data, rather than by a closed form analytical expression. Fuzzy rules are derived from such data sets typically using a neural network or fuzzy clustering algorithm.

In their simplest form, fuzzy rules take the canonical form:

If (ANTECEDENT) then CONSEQUENT

or: If X is A then Y is B, where A and B are constraints on variables X and Y. Fuzzy if-then rules define joint possibility distributions and patch the space $X \times Y$ as shown in *Figure* 1.

The shaded area in *Figure* 1 represents the cartesian product $A \times B$, derived by projections from the antecedent and consequent fuzzy sets of a rule: If X is A then Y is B. This forms a patch in the $X \times Y$ space. A set of fuzzy rules developed appropriately can be made to patch the space in a specific fashion – for example to cover a function as exemplified by a set of input-output data points. Once again, the geometrical picture is important.

An important issue that one must deal with when developing an approximate reasoning system is to derive a rule base that accurately 'covers' or patches the function that underlies the application system. For this purpose one often needs to collect numeric data from system measurements of inputs and outputs. Given such a data set, one can use a clustering algorithm that finds an optimal cover of the function. A fuzzy rule base can then be derived from cluster information [1].

Rule Composition and Defuzzification

Assuming that we are given a set of fuzzy rules, one needs to devise an interpolation mechanism that is capable of generating a numeric output from a numeric set of inputs after considering all the rules in the rule base. Designing this is straightforward.

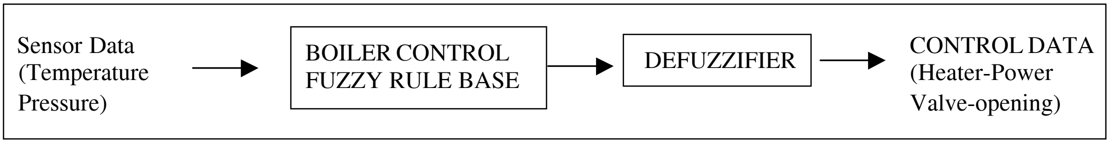


Figure 2. Typical flow of data in a fuzzy control system. Sensors provide a current status of the system. This status is input to the fuzzy controller rule base which determines the extent to which different rules in the rule base fire. Finally, a defuzzification mechanism generates a crisp numeric output which is fed back to the external control system for action.

What one needs to really understand is that given a set of real valued numeric inputs, to what extent does each rule fire? Returning to the boiler control example, given a set of sensor readings of temperature and pressure how do we generate a value of the extent of valve-opening and heater power as dictated by the rule base?

Crisp inputs need to be composed in a systematic way with individual rules of the rule base and a crisp output generated using a well defined procedure. This process is called defuzzification. *Figure* 2 portrays this process in block diagram form.

There are numerous variants of the defuzzification procedure. Here we describe only one of them – *centroidal defuzzification*. The procedure is easy to understand graphically. Consider our two rules in the boiler control rule base. We will go through the defuzzification for one consequent, heater-power. The case for the second output can be handled similarly. *Figure* 3 shows the various fuzzy sets defined on each of the universes of discourse – temperature, pressure, and heater-power. Rules *R*1 and *R*2 of our boiler control example are also portrayed graphically in *Figure* 3. In general a compositional rule for inference involves the following procedure:

1. Compute memberships of current inputs in the relevant antecedent fuzzy sets of a rule;

2. Since the antecedents are in conjunctive form, the AND operation is replaced by a minimum, i.e., compute the minimum of the memberships found in step 1 above;

3. Scale or clip the consequent fuzzy set of the rule by the minimum value found in step 2 since this gives the smallest degree to which the rule must fire;

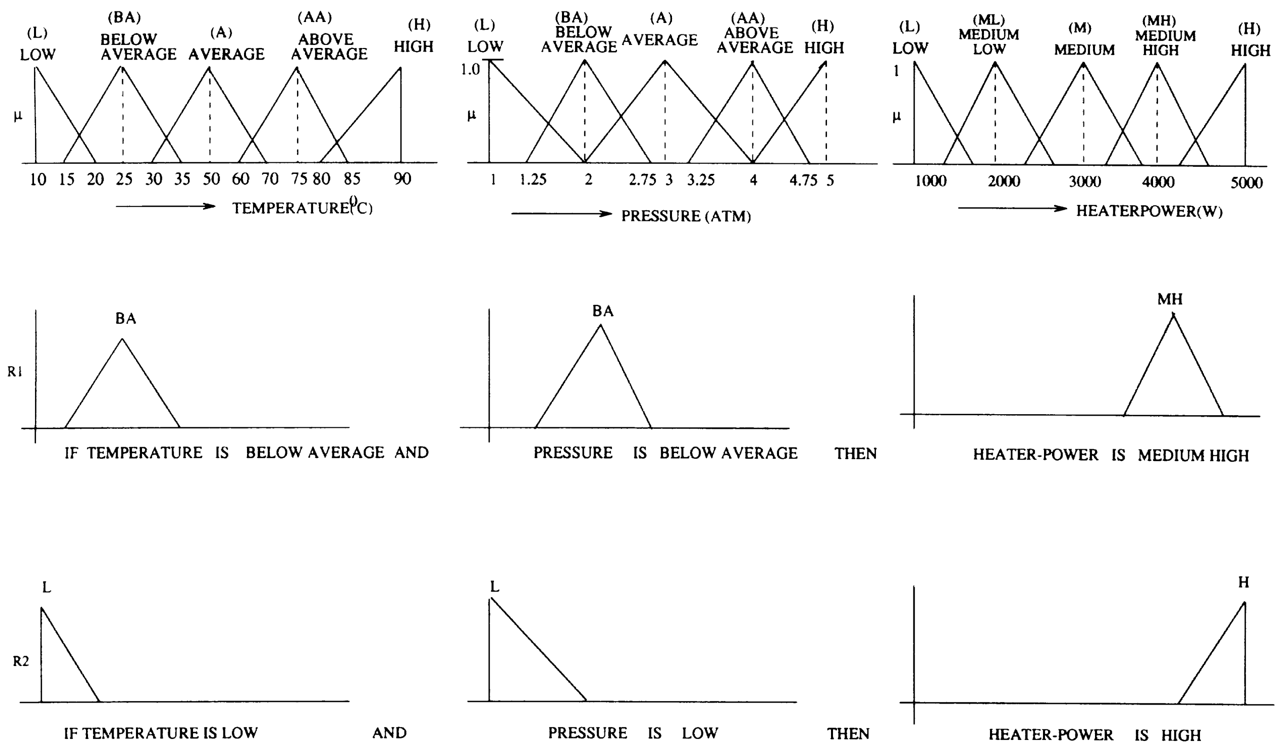


Figure 3. Fuzzy sets defined on the UODs of temperature, pressure and heater-power. Each of the sets overlap approximately to an extent of 25%. Fuzzy rules combine antecedent labels using the conjunction operator and map the combination to an output UOD fuzzy set. Two rules are depicted graphically in the lower part of the figure.

4. Repeat steps (1) – (3) for each rule in the rule base;

5. Superpose the scaled consequent fuzzy sets and compute the centroid of the composite set formed by such a superposition.

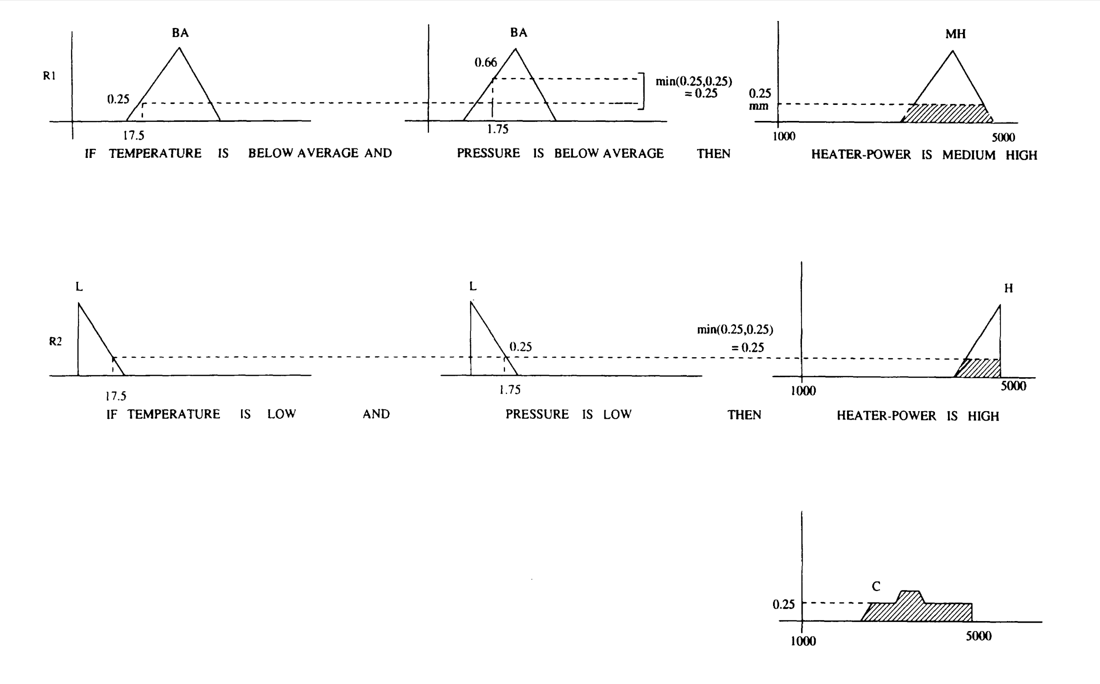
Figure 4 clarifies this procedure graphically for the two rules *R*1 and *R*2 of our boiler control example. In the example shown, we assume that the sensors in the boiler control system read a temperature of 17.5°C and pressure of 1.75ATM. We compute the centroid as follows:

*R*1: μ^T_{BA} (17.5) = 0.25; μP_{BA}(1.75) = 0.66; therefore min(0.25,0.66) = 0.25

*R*2: μ^T_L (17.5) = 0.25; μP_L(1.75) = 0.25; therefore min(0.25,0.25) = 0.25.

In this example both rules fire to the same degree for an instantaneous sensor input of (17.5, 1.75). The output sets are

Figure 4. The figure shows the defuzzification procedure graphically for a sensor reading of (17.5°C, 1.75ATM). Rule 1 antecedents (BA and BA) fire to degrees 0.25 and 0.66 respectively. The minimum of these (0.25) is used to clip the consequent set (MH). Rule 2 antecedents fire to degrees 0.25 each, and this value clips the consequent set (H) to 0.25. The two clipped consequents are superposed using addition of memberships. The defuzzified output is the centroid of the superposed area.



clipped to 25% peak memberships as shown by the shaded areas. The resultant clipped sets are added together. Typically the defuzzified output is calculated from the composite output set, *C*, using a center of area method:

$$Z_c = \frac{\Sigma_i \mu_c(Z_i).Z_i}{\Sigma_i \mu_i(Z_i)}$$

In our example, the defuzzified centroid is around 4100 W.

Fuzzy engineering deals primarily with the development of fuzzy systems for the broad task of function approximation.

Fuzzy Engineering

Fuzzy engineering deals primarily with the development of fuzzy systems for the broad task of function approximation. It seeks to model complex non-linear systems with the mathematics of fuzzy logic. Given a set of sensor based readings of a set of inputs and outputs of a system, what underlying function governs the input-output relationship of the system in question? Can we approximate such a function with a practically sufficient level of accuracy?

As it turns out, fuzzy systems are capable of function approximation provided one is able to develop an appropriate set of rules based on a set of linguistic labels and attendant membership functions defined over the UOD's in question. The development of a suitable rule base is a tricky task and often one needs to resort to neural network technologies [2] to be able to search out membership functions. Deciding on the number of labels and the resulting rule base is easier than defining a set of membership functions which we associate with defined linguistic. For both these tasks, one usually develops a first cut design using heuristics or clustering, and subsequently fine tunes the membership functions using a supervised neural network learning algorithm [3]. In this section our purpose is only to exemplify the possibility of employing a fuzzy system for function approximation and to demonstrate the behaviour of an exemplary system. We will attempt to approximate the well known bell shaped function: $f = e^{-(x^2+y^2)}$. Let us assume that the input data is constrained to the interval [–3, 3] along both the *X* and *Y* axes.

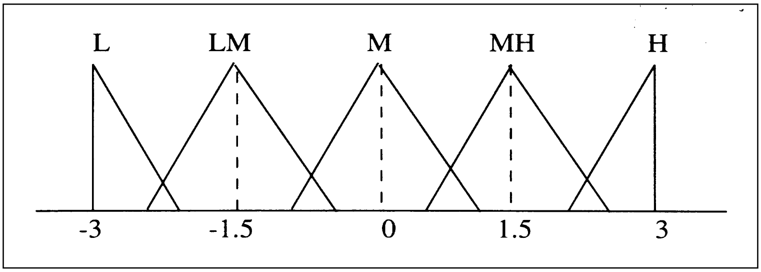


Figure 5. Fuzzy sets defined on the interval [–3, 3] to partition the X and Y axes into five regions each. The five fuzzy sets on X and Y can combine to form 25 rules.

Below, we spell out the basic ideas in developing such an approximator.

Definition of Fuzzy Sets on Input and Output UOD's

We first need to define fuzzy sets over each UOD, i.e., on the interval [–3, 3] for both *X* and *Y* axes. For this example, we define 5 sets on each axis as shown in *Figure* 5. These five sets and their associated labels give us a rough division of [–3, 3] into five regions reflecting an increasing magnitude of each of the inputs *X* and *Y*.

Notice that the fuzzy sets defined in both *X* and *Y* axes are identical. For simplicity, we have defined them symmetrically although in general they need not be so. The sets have been labelled LOW(L), LOW-MEDIUM(LM), MEDIUM(M), MEDIUM-HIGH(MH), and HIGH(H).

The range of outputs for this example is in the interval [0,1] (for the function $f = e^{-(x^2+y^2)}$). We define 4 sets as shown in *Figure* 6. These sets are LOW(L), LOW-MEDIUM(LM), MEDIUM-HIGH (MH), and HIGH(H).

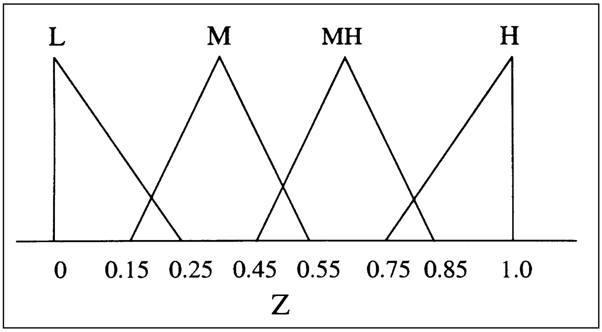


Figure 6. Four fuzzy sets defined on the output UOD. Here the magnitude of the bell shaped function that can vary from 0 to 1 is roughly divided into four regions LOW, LOW-MEDIUM, MEDIUM-HIGH, and HIGH.

Table 1. Rule matrix structure for the example.

M					
MH					
M					
LM					
L					
	L	LM	M	MH	H

Development of the Rule Matrix

Since there are five labels defined on each of the inputs *X* and *Y*, we can have 25 combinations of antecedent conjunctions. For example, If *X* is L AND *Y* is H then *Z* is ...

These combinations index into the rule matrix of *Table* 1. The issue at hand that needs to be decided is: to which output label does a specific input label combination get associated?

We will develop this association (rule base) heuristically in this example. To do this, observe the 2-dimensional bell shaped function (see *Figure* 8) from above, looking towards the origin on the *X*-*Y* plane. Notice that the function is symmetric about the origin, and it increases uniformly towards the origin. Looking from above we notice that if we were to draw a contour plot we could have three concentric bands reflecting an increase in magnitude moving from the periphery towards the centre. These are labelled regions of low magnitude, low-medium magnitude, medium-high magnitude, and high magnitude. Now all that one needs to do is to pick up consequent labels from *Figure* 6 and assign them to the combinations of *Table* 1 to reflect this contour

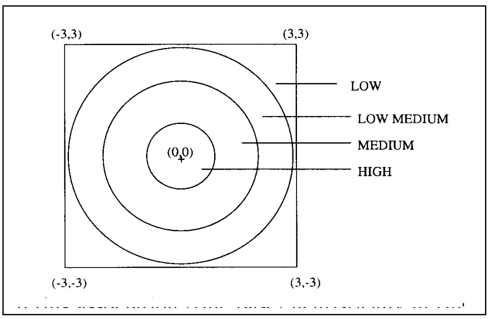


Figure 7. A contour plot of the bell shaped function looking from above towards the origin. Concentric regions reflect increasing magnitude towards the center. This picture gives us an intuitive feel of how the rule base consequents should be spread out over the matrix. This picture is reflected in Table 2.

Table 2. The fuzzy rule base derived intuitively. L's at the periphery enclose ML's and MH's which in turn enclose an H at the centre.

M	L	L	ML	L	L
MH	L	ML	MH	ML	L
M	ML	MH	H	MH	ML
LM	L	ML	MH	ML	L
L	L	L	ML	L	L
	L	LM	M	MH	H

map! The result of this is shown in *Table* 2. Notice the L's on the periphery, and the H in the centre. This is the rule base of our fuzzy system.

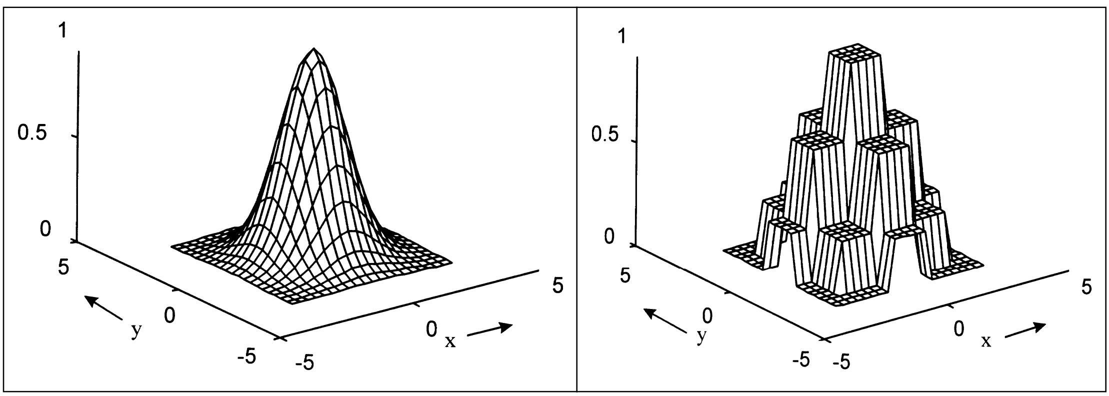
Some examples of rules read off this matrix are:

If *X* is *L* and *Y* is H then *Z* is L.
If *X* is M and *Y* is M then *Z* is H.

Simulating the Approximation

We can generate an output estimate of $f(\cdot)$ using our hand-crafted fuzzy system using the compositional rules of inference of centroidal defuzzification described in the previous section. We choose 30 uniformly spaced points in the interval [–3,3]. Each pair of points (amounting to 900 pairs) is input to the fuzzy system and for each input pair an output is generated. *Figure* 8 shows the original function, and *Figure* 9 its fuzzy approximation using our fuzzy system. Notice the similarity in the shapes of both functions, and the stepped nature of the fuzzy system response. Fuzzy systems generate stepped outputs. Given the

Figure 8 (left). Bell shaped function to be approximated.
Figure 9 (right). Approximated function using a simple fuzzy system.



simplicity of the design, the approximation is good.

Applications and Conclusions

Fuzzy applications use fuzzy systems in their micro-chip controllers. The most famous application is the Sendai Subway Controller from Hitachi Corporation. On a route of 13.6 kms with 16 stations, the fuzzy controller adjusts the braking and the acceleration to give passengers a smoother ride with a better power efficiency than conventional controllers. Fuzzy washing machines use pulsing optical sensor data to decide wash times and cycles, helping prevent damage to clothes. Canon developed one of the first cameras and camcorders that use fuzzy systems for auto-focus and to cancel hand-jitter. Michio Sugeno's unmanned helicopter control responds to voice commands like 'up', 'land', and 'hover' using a fuzzy voice controller. And the list of applications goes on. For whatever the probability theorists say, fuzzy systems are here to stay – because they work!

This two part article introduces the subject of fuzzy set theory based on a geometrical interpretation. It also extends these ideas to discuss the basis for fuzzy logic. Through simple examples a fuzzy system has been designed to show the way such systems behave. I would like to end this article with two quotes – one by probably the greatest physicist ever known and the other by a profound philosopher.

> *"So far as the laws of mathematics refer to reality, they are not certain. And so far as they are certain, they do not refer to reality."*
>
> *Albert Einstein, Geometry and Experience*

> *"All traditional logic habitually assumes that precise symbols are being employed. It is therefore not applicable to this terrestrial life, but only to an imagined celestial one."*
>
> *Bertrand Russell, Vagueness*

These great men had realised early on that there was a void in the existing mathematical framework when it came to handling real world issues. Fuzzy set theory, fuzzy systems and fuzzy engineering are now helping fill this void.

Suggested Reading

[1] B A Kosko, *Fuzzy Thinking*, Hyperion Press, 1993.

[2] B A Kosko, *Fuzzy Engineering*, Prentice Hall, Englewood Cliffs, NJ, 1997.

[3] G Klir and B Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Englewood Cliffs, NJ, 1995.

Address for Correspondence
Satish Kumar
Department of Physics and Computer Science
Dayalbagh Educational Institute
Dayalbagh
Agra 282005, India.