CSG2341
Intelligent Systems

Fuzzy Case Study and Fuzzy Control

Fuzzy Expert System Case Study

- A service centre keeps spare parts and repairs failed ones.
- A customer brings a failed item and receives a spare of the same type.
- Failed parts are repaired, placed on the shelf, and thus become spares.
- The objective here is to advise a manager of the service centre on certain decision policies to keep the customers satisfied.

***Step 1*: Specify the problem and define linguistic variables**

There are four main linguistic variables: average waiting time (mean delay) *m*, repair utilisation factor of the service centre ρ, number of servers *s*, and initial number of spare parts *n*.

***Step 2*: Determine fuzzy sets**

Fuzzy sets can have a variety of shapes. However, a triangle or a trapezoid can often provide an adequate representation of the expert knowledge, and at the same time, significantly simplifies the process of computation.

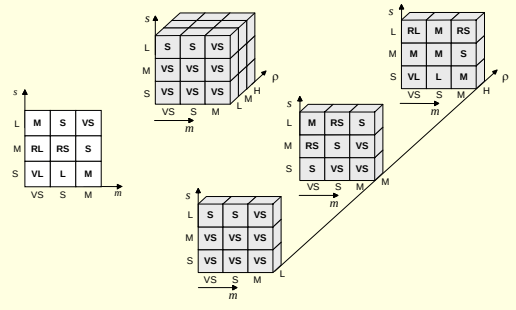
Let's code this much up….

Linguistic Variable: *Mean Delay, m*		
Linguistic Value	Notation	Numerical Range (normalised)
Very Short	VS	[0, 0.3]
Short	S	[0.1, 0.5]
Medium	M	[0.4, 0.7]
Linguistic Variable: *Number of Servers, s*		
Linguistic Value	Notation	Numerical Range (normalised)
Small	S	[0, 0.35]
Medium	M	[0.30, 0.70]
Large	L	[0.60, 1]
Linguistic Variable: *Repair Utilisation Factor, ρ*		
Linguistic Value	Notation	Numerical Range
Low	L	[0, 0.6]
Medium	M	[0.4, 0.8]
High	H	[0.6, 1]
Linguistic Variable: *Number of Spares, n*		
Linguistic Value	Notation	Numerical Range (normalised)
Very Small	VS	[0, 0.30]
Small	S	[0, 0.40]
Rather Small	RS	[0.25, 0.45]
Medium	M	[0.30, 0.70]
Rather Large	RL	[0.55, 0.75]
Large	L	[0.60, 1]
Very Large	VL	[0.70, 1]

***Step* 3: Elicit and construct fuzzy rules**

- To accomplish this task, we might ask the expert to describe how the problem can be solved using the fuzzy linguistic variables defined previously.
- Required knowledge also can be collected from other sources such as books, computer databases, flow diagrams and observed human behaviour.

FAM and 3D FAM



***Step* 4: Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system**

- Let's add these to our program…

Step 5: Evaluate and tune the system

- How??

Tuning fuzzy systems

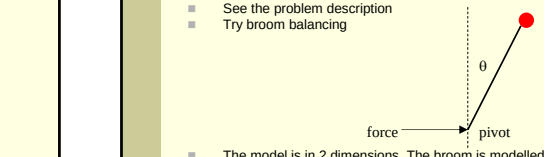
1. Review model input and output variables, and if required redefine their ranges.
2. Review the fuzzy sets, and if required define additional sets on the universe of discourse. The use of wide fuzzy sets may cause the fuzzy system to perform roughly.
3. Provide sufficient overlap between neighbouring sets. It is suggested that triangle-to-triangle and trapezoid-to-triangle fuzzy sets should overlap between 25% to 50% of their bases.

Tuning fuzzy systems

4. Review the existing rules, and if required add new rules to the rule base.
5. Examine the rule base for opportunities to write hedge rules to capture the pathological behaviour of the system.
6. Adjust the rule execution weights. Most fuzzy logic tools allow control of the importance of rules by changing a weight multiplier.
7. Revise shapes of the fuzzy sets. In most cases, fuzzy systems are highly tolerant of a shape approximation.

Another case study – inverted pendulum

- See the problem description
- Try broom balancing



- The model is in 2 dimensions. The broom is modelled as a massless rigid pole 1m long, with a 1kg mass attached at the top. The pole is supported from below at the pivot, and a force can be applied to the right or left at that point. The angle between the pole and the vertical (in radians) is called θ (theta). If the pole leans to the right, theta is positive.
- In order to keep the pole balanced, the force must be varied taking into account theta, as well as the rate of change or theta (i.e. the pole may be leaning to the right but rotating toward the left). A maximum force of about 1.5 newtons should be enough to keep the pole balanced.

Step 1: Find the variables

For a **control** application, we have two types of variables

- Inputs (sensors): ?
- Outputs (affectors): ?

Step 1: Find the variables

Possible solution:

- Inputs (sensors): theta and rate of change of theta
- Outputs (affectors): force

Step 2: Determine fuzzy sets

For each input variable, we need some linguistic values

- Change in theta: ?
- Theta: ?

- (For a Mandani-style system, we also need linguistic values for output variables)

Step 2: Determine fuzzy sets

Possible solution

- Change in theta: fastLeft, slowLeft, still, slowRight, fastRight
- Theta: wayLeft, left, slightlyLeft, straight, slightlyRight, right, wayRight

- Code this much up…

Step 3: Fuzzy rules

For each combination of linguistic values, what should the control action be? (i.e. what should be the value of the output variables)

Step 3: Fuzzy rules

Let's code this up…

Step 4: Code it up – done as we go.
Step 5: Evaluation and tuning.

Let's try it out and see if we can improve it…