

# Lecture 35

## Fuzzy Logic Control (III)

# Outline

- FLC design procedure
  - Defuzzification
  - Fine-tuning fuzzy rules.
- Dog chases cat example implementation details

# FLC Design Procedures

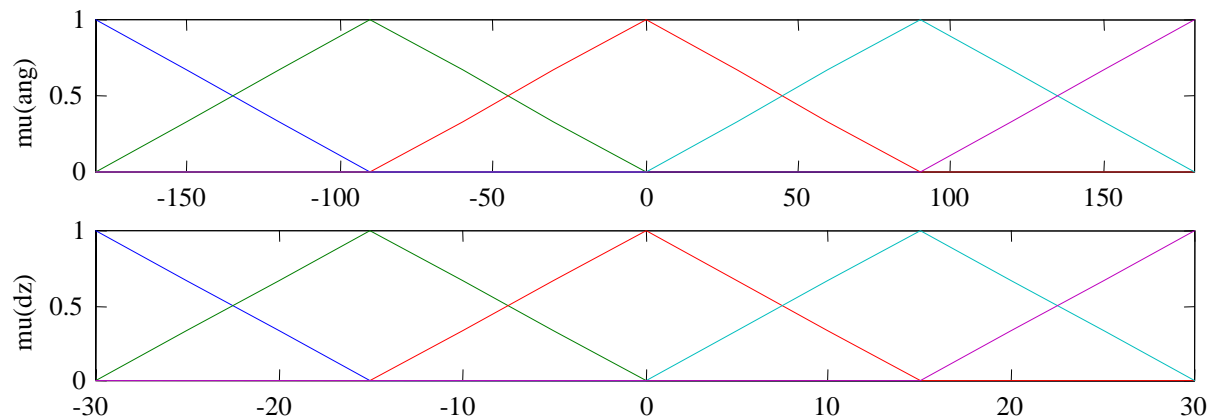
Step 4. Defuzzification

Step 5. Fine-tuning the control rules and performance evaluation

- Evaluate the quality of the control rules using testing data set, and iteratively refine the definition of the fuzzy sets, and the fuzzy control rules.
- Most time consuming, tedious, and difficult part.
- Evaluate the effectiveness of a fuzzy controller by comparing to (if available) existing base line algorithms, analyzing cost benefit trade-off, implementation issues, etc.
- FLC is only one of many alternatives! The value of FLC must be weighted against competing solutions.

# Implementation Details

- Fuzzification:
    - Use discrete support. The universe-of-discourse of support are sampled at uniform (or non-uniform) intervals.
    - Each fuzzy set (linguistic variable) is represented as a vector. Each element of the vector represents the values of the membership function at a particular point in the universe of discourse.
  - DCC example
    - Support vectors (both in degrees)
- sang: [-180 -150 -120 -90 -60 -30 0 30 60 90 120 150 180]
- sdz: [-30 -25 -20 -15 -10 -5 0 5 10 15 20 25 30]



# Representing Fuzzy Sets

```
M=[1 .67 .33 0 0 0 0 0 0 0 0 0 0; % LN
    0 .33 .67 1 .67 .33 0 0 0 0 0 0 0; % SN
    0 0 0 0 .33 .67 1 .67 .33 0 0 0 0; % ZO
    0 0 0 0 0 0 0 .33 .67 1 .67 .33 0; % SP
    0 0 0 0 0 0 0 0 0 0 .33 .67 1]; % LP
```

Each row is represent a different fuzzy variable

Each column of M is a sampling point over the universe of discourse of the support.

To fuzzify  $\text{ang}(t) = 20^\circ$ , first represent it on the support sang using interpolation:  $\alpha = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1/3 \ 2/3 \ 0 \ 0 \ 0 \ 0 \ 0]$

Next, determine the fuzzy representation (input fuzzy variable activation)of  $\text{ang}(t)$ :

$$a = \alpha M' = \begin{bmatrix} 0 & 0 & 7/9 & 2/9 & 0 \end{bmatrix}$$

LN   SN   ZO   SP   LP

# Representing Rules

	ang					&?		dz				weight
	LN	SN	ZO	SP	LP	LN	SN	ZO	SP	LP		
rule=[	1	0	0	0	0	1	1	0	0	0	0	1; % if ang is LN then dz is LN
	0	1	0	0	0	1	0	1	0	0	0	1; % if ang is SN then dz is SN
	0	0	1	0	0	1	0	0	1	0	0	1; % if ang is ZO then dz is ZO
	0	0	0	1	0	1	0	0	0	1	0	1; % if ang is SP then dz is SP
	0	0	0	0	1	1	0	0	0	0	1	1]; % if ang is LP then dz is LP

&? = 1 if there is only one input fuzzy variable (this case) or the second fuzzy variable is to be ignored for that rule.

Each row is a rule.

# Inference

- Calculate rule activation from input fuzzy variable activation

Activation = max(antecedent part of each rule \* fuzzy set activation)

	LN	SN	ZO	SP	LP	$\omega$
Rule#	0	0	7/9	2/9	0	$\omega$
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	7/9	0	0	7/9
4	0	0	0	2/9	0	2/9
5	0	0	0	0	0	0

# Inference

- Calculate output fuzzy set activation
  - Multiply each  $\omega$  (rule activation value) to the output variable portion of each corresponding rule. (assuming only one output variable)
  - Since multiple rules may be activated, find the maximum activation (fuzzy-OR) of each output fuzzy set. This gives the activation of individual output fuzzy set.
- $B'$  is found using either the max-product method, or the max-min method

# Simulation Result

