EE4305 Fuzzy/Neural Systems for Intelligent Robotics

PART II: FUZZY SYSTEMS

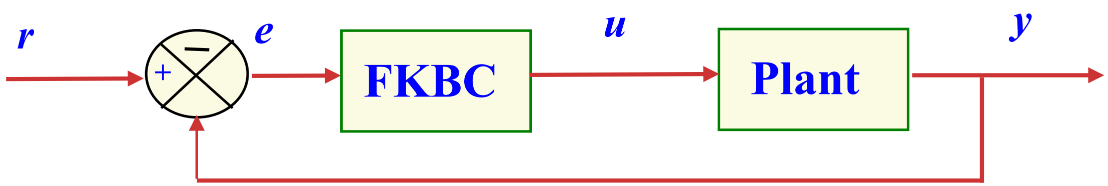
Chapter 8: Fuzzy Knowledge Based Control

Topics to be Covered…

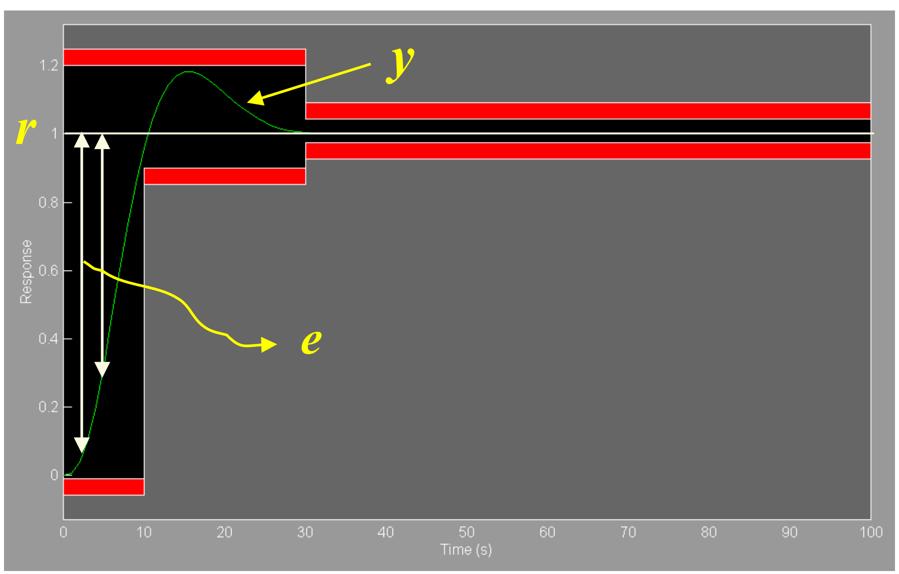
- Fuzzy sets and crisp sets
- Fuzzy operations, fuzzy relations, fuzzy compositions
- Extension principle, fuzzy numbers
- Approximate reasoning, fuzzy inference
- Multi-rule Fuzzy Inference
- Fuzzy knowledge based control (FKBC)

Fuzzy Knowledge Based Control (FKBC)

- Overview



The operator’s knowledge can be incorporated as knowledge-base or rule-base in FKBC



Fuzzy controllers, contrary to classical controllers, are capable of utilizing knowledge elicited from human operators.

This is crucial in control problems for which it is difficult or even impossible to construct precise mathematical models, or for which the acquired models are difficult or expensive to use.

It has been observed that experienced human operators are general able to perform well without knowing the mathematical models.

In many cases, the human experts can do much better than robots without using the mathematical equations!

But how to code the expert knowledge?

The expert knowledge is difficult to express in precise terms.

An imprecise linguistic description of the manner of control can usually be articulated by the expert operator with relative ease.

This linguistic description consists of a set of control rules that make use of fuzzy propositions.

A typical form of these rules:

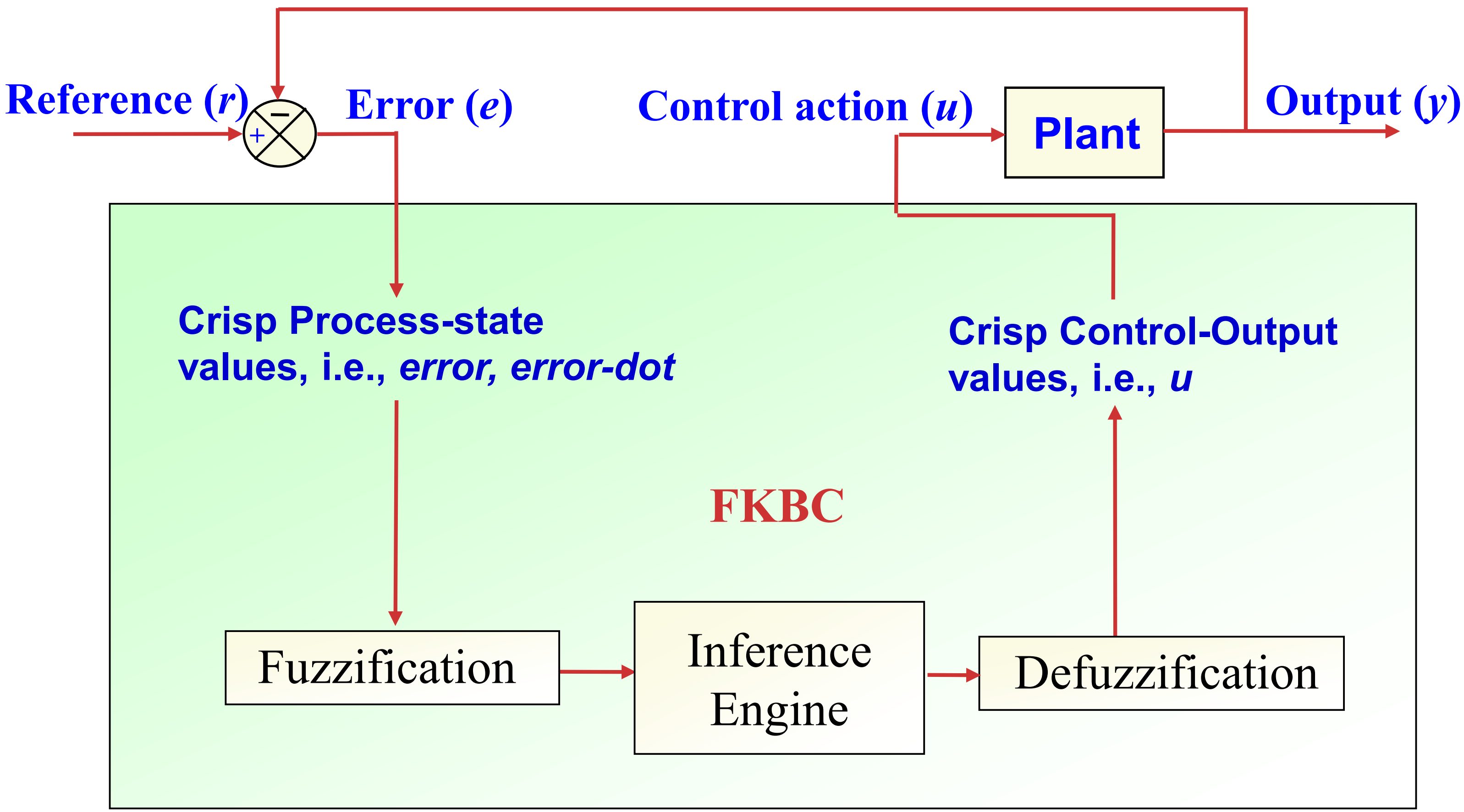
IF the temperature is very high

AND the pressure is slightly low

THEN the heat change should be slightly negative

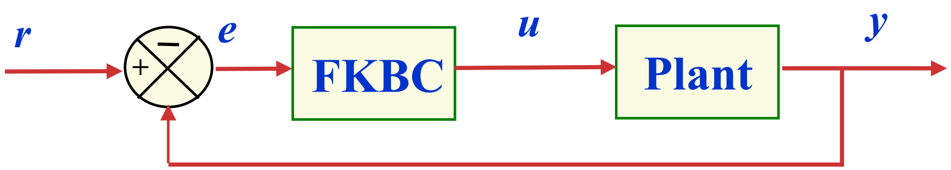
The vague terms *very high*, *slightly low*, and *slightly negative* can be conveniently represented by fuzzy sets!

- There are three main computational steps in FKBC: *Fuzzification, Inference, and Defuzzification*



Procedures in Designing a FKBC (5 steps)

1. Identify the variables (e.g., inputs, outputs) of the plant.



The objective is to keep the output at some reference constant r.

In many applications, both the error signal, e, and its derivative, $\dot{e}$, are measured. Using values of e and $\dot{e}$, the controller produces values of control input u.

The classical PID controller:

$$u = K_p e + K_D \dot{e} + K_I \int e(t)dt$$

We are going to use expert knowledge to design the fuzzy controller!

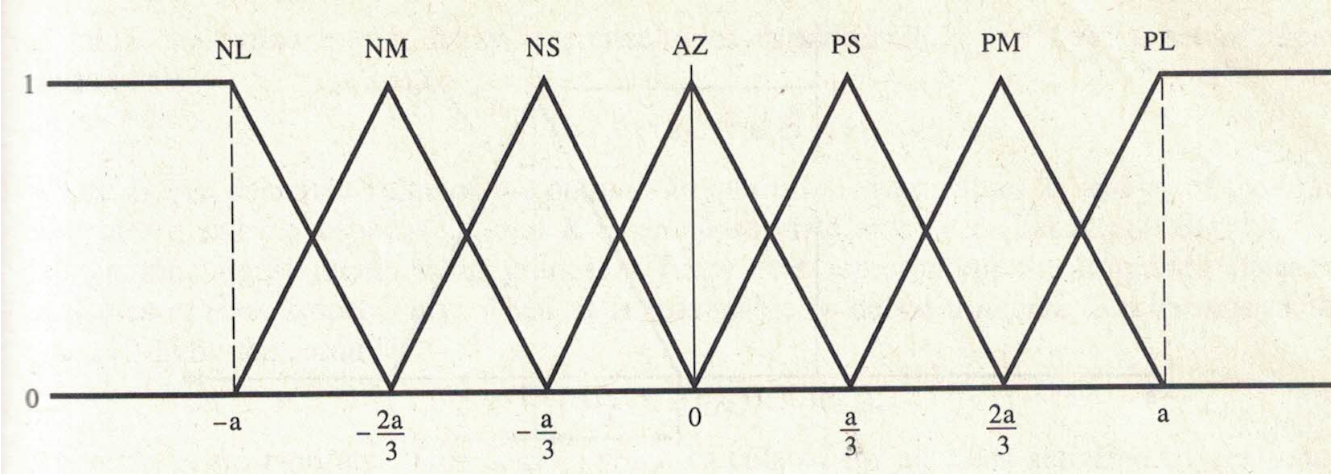
After identify relevant input and output variables of the controller and ranges of their values, we have to select meaningful linguistic states for each variable and express them by appropriate fuzzy sets.

In most cases, these fuzzy sets are fuzzy numbers which represent linguistic labels such as approximate zero, positive small, negative small, positive medium and so on.

The following seven linguistic states are commonly used in fuzzy controller.

NL—*negative large*
NM—*negative medium*
NS—*negative small*
AZ—*approximately zero*
PL—*positive large*
PM—*positive medium*
PS—*positive small*

A typical example of representing a variable in the range of [-a, a] with seven linguistic states



Other shapes of the membership functions might be used for different applications.

The results are not sensitive to the shape of membership functions. So the above triangular type is the most popular ones.

2. The expert knowledge related to the given control problem is formulated in terms of set of fuzzy inference rules.

In the example with variables e, $\dot{e}$, and u, the inference rules have the canonical form

If e=A and $\dot{e}$=B, then u=C

where A, B, and C are fuzzy numbers chosen from the set of fuzzy numbers that represent the linguistic states NL(NB), NM, NS, AZ(ZO), PS, PM, and PL(PB).

Since each input variable has seven linguistic states, the total number of possible non-conflicting fuzzy inference rules is 7x7=49.

In practice, a small subset of all possible fuzzy inference rules is often sufficient to obtain acceptable performance of the fuzzy controller.

An example of fuzzy rule base, which can conveniently be represented in a matrix form.

<table>
<tr><td colspan="2"></td><td colspan="7">ė</td></tr>
<tr><td></td><td>v</td><td>NL</td><td>NM</td><td>NS</td><td>AZ</td><td>PS</td><td>PM</td><td>PL</td></tr>
<tr><td rowspan="7">e</td><td>NL</td><td colspan="4" rowspan="2">PL</td><td rowspan="2">PM</td><td colspan="2" rowspan="2">AZ</td></tr>
<tr><td>NM</td></tr>
<tr><td>NS</td><td colspan="2" rowspan="3">PM</td><td>PM</td><td>PS</td><td>AZ</td><td colspan="2" rowspan="3">NM</td></tr>
<tr><td>AZ</td><td>PS</td><td>AZ</td><td>NS</td></tr>
<tr><td>PS</td><td>AZ</td><td>NS</td><td>NM</td></tr>
<tr><td>PM</td><td colspan="2" rowspan="2">AZ</td><td rowspan="2">NM</td><td colspan="4" rowspan="2">NL</td></tr>
<tr><td>PL</td></tr>
</table>

The fuzzy inference rules are obviously the most important part to achieve good performance.

How to design the fuzzy rules?

There are two principle ways in which relevant inference rules can be determined.

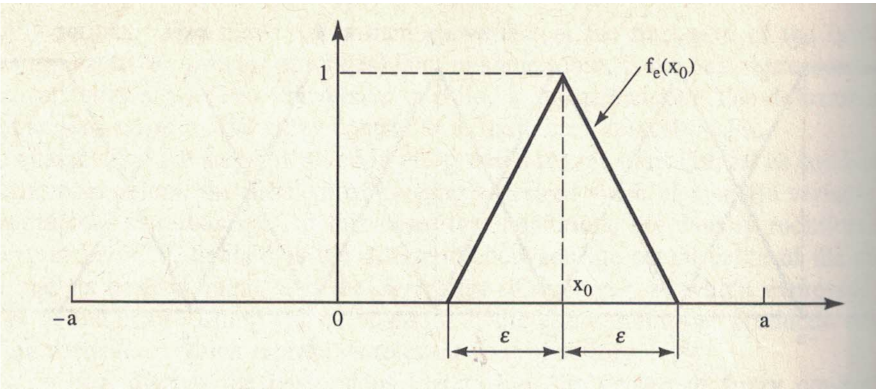
One way is to elicit them from experienced human operators.

The other way is to obtain them from empirical data by suitable learning methods, usually with the help of neural networks.

3. **Fuzzification**:

In this step, a fuzzification function is introduced for each input variable to express the associated measurement uncertainty.

A simple fuzzification method is shown below. A crisp measurement is changed into a fuzzy number.

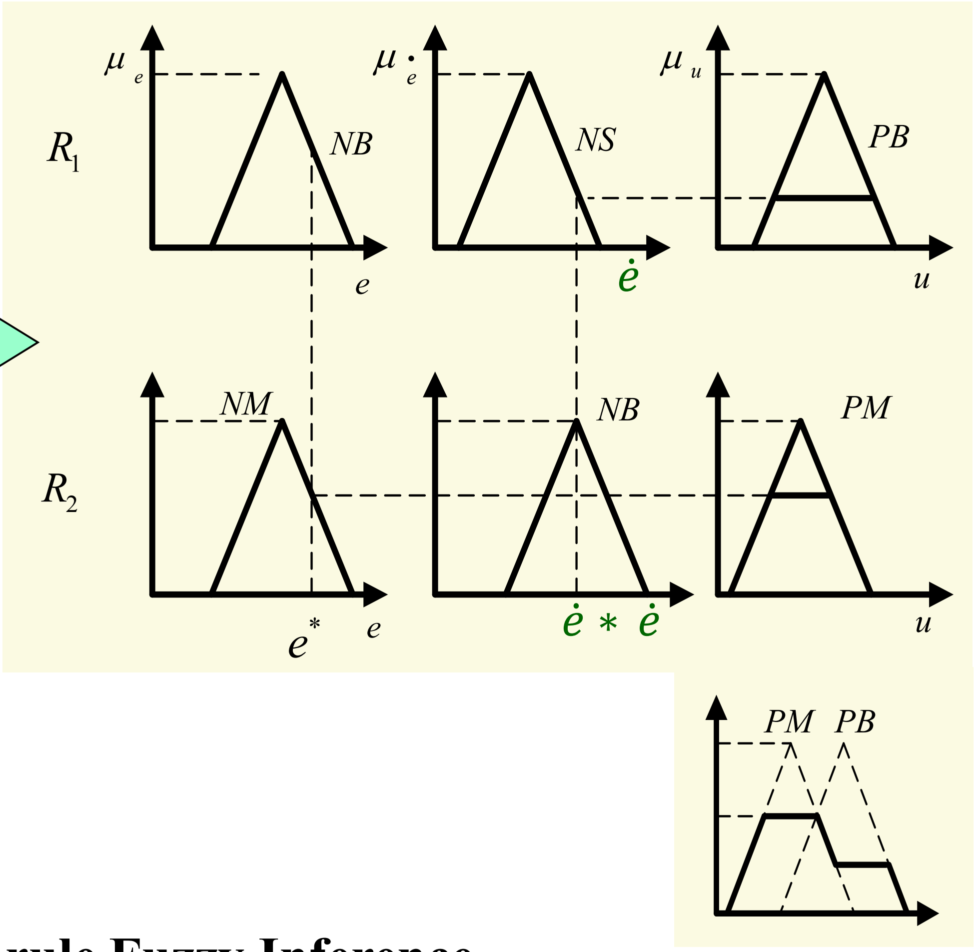


In many control problems, the crisp measurement numbers are not fuzzified. In this case, the crisp number is interpreted as a special fuzzy number where the support is only one point (ε=0 in above example) with full membership.

4. **Fuzzy inference**: Use fuzzy approximate reasoning to infer the output contributed from each rule. Aggregate the fuzzy outputs recommended by each rule.

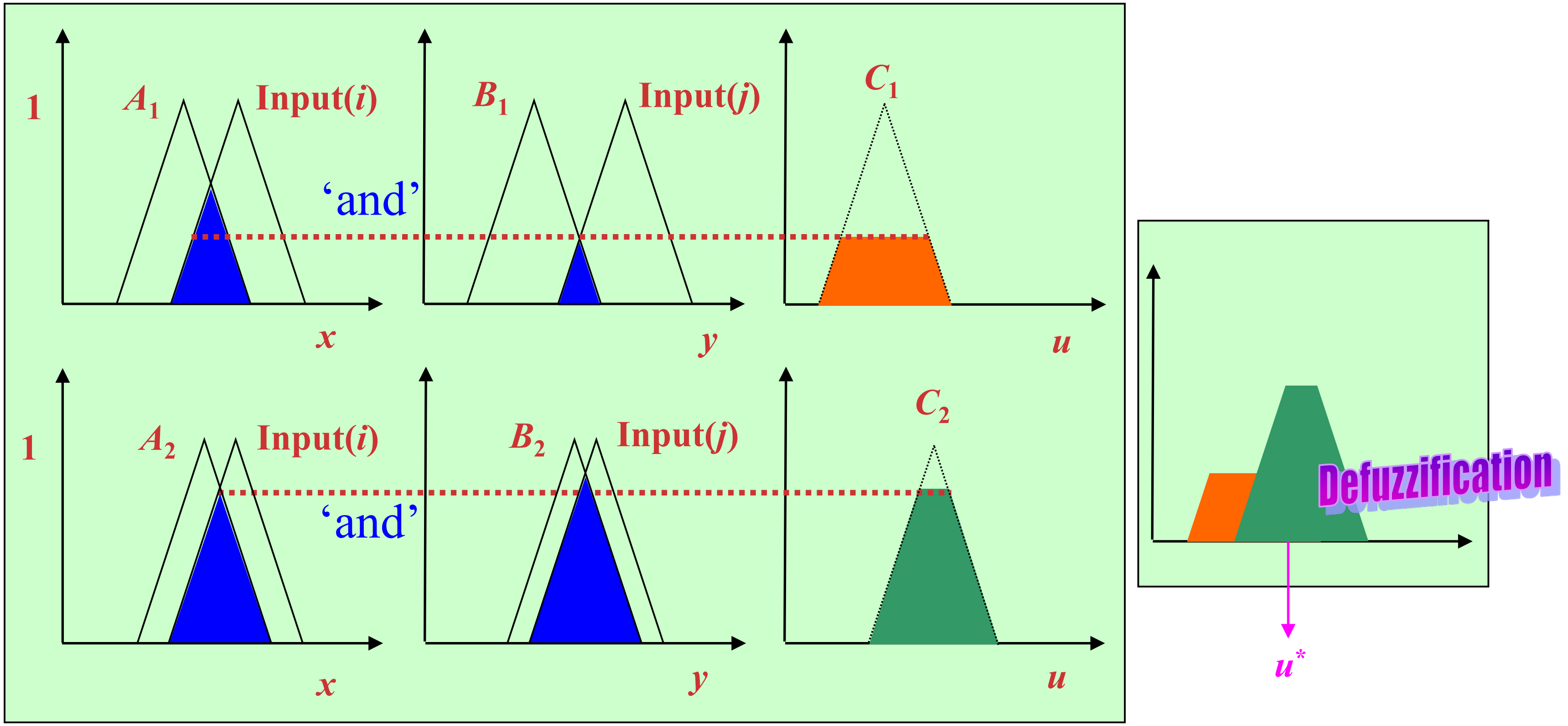
Rule statement with ‘and’

e.g., if e is A and $\dot{e}$ is B then u is C

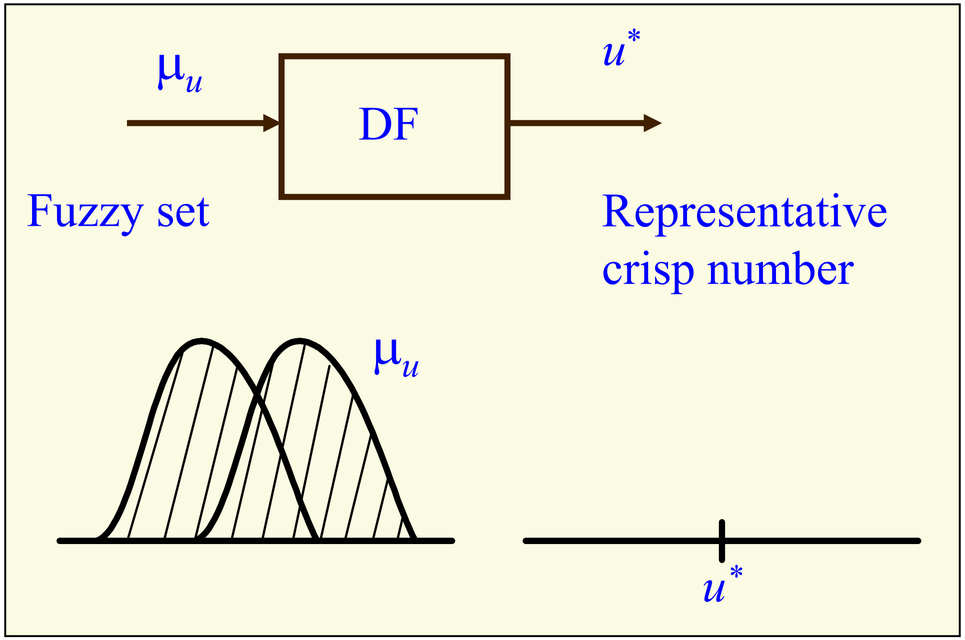


Follow the procedure for **Multi-rule Fuzzy Inference** detailed in chapter 7.

- Fuzzy inference with fuzzy inputs:



5. Defuzzification: The result of the fuzzy inference is a fuzzy set μ_u. Defuzzification is to obtain a scalar value of u^* from μ_u.

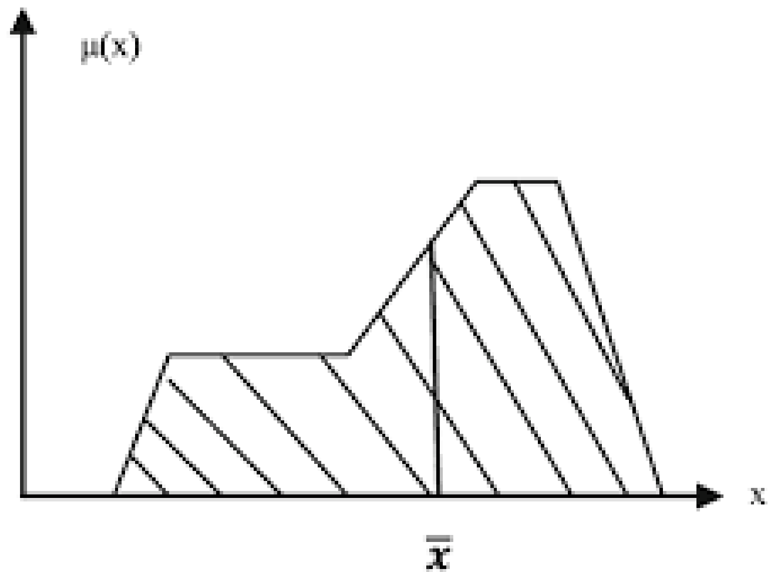


There are many defuzzification methods:

- Center-of-Area/Gravity
- Center-of-Sums
- ….
- ….

Center-of-Area/Gravity (Centroid) Method

Given a fuzzy set , the defuzzified value is the value for which the area under the graph of membership function is divided into two equal subareas.



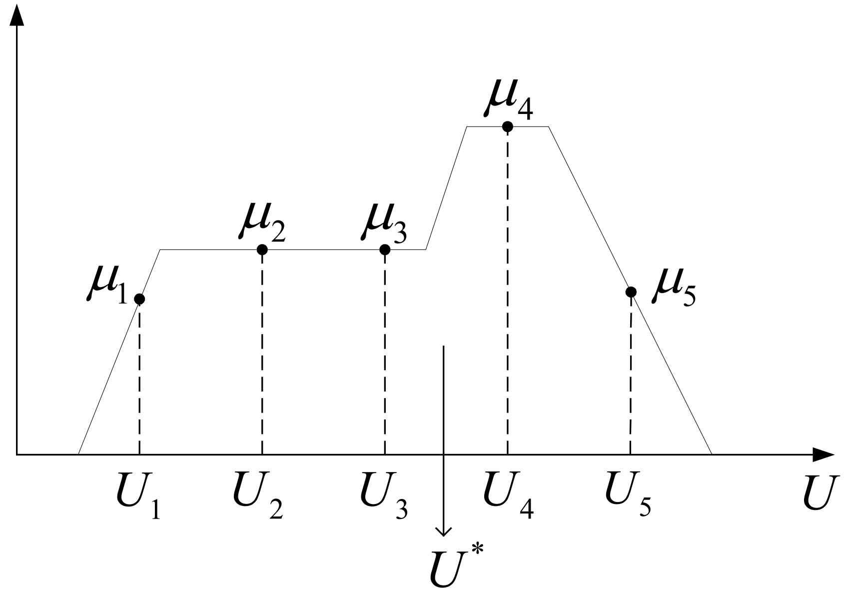
For continuous variable, the center of area is calculated by

$$x^* = \frac{\int \mu_A(x)x dx}{\int \mu_A(x) dx}$$

- Center-of-Area/Gravity:

Discrete: $\mathcal{U} = \{u_1, \ldots, u_m\}$

$$u^* = \frac{\sum_{i=1}^{m} u_i \cdot \mu_U(u_i)}{\sum_{i=1}^{m} \mu_U(u_i)}$$



This method determines the center of the area below the combined membership function.

$$U^* = \frac{U_1 \cdot \mu_1 + U_2 \cdot \mu_2 + U_3 \cdot \mu_3 + U_4 \cdot \mu_4 + U_5 \cdot \mu_5}{\mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5}$$

♦ Center-of-Area/Gravity:

Discrete: $\mathcal{U} = \{u_1, \ldots, u_m\}$

$$u^* = \frac{\sum_{i=1}^{m} u_i \cdot \mu_U(u_i)}{\sum_{i=1}^{m} \mu_U(u_i)} \quad \Longrightarrow \quad u^* = \sum_{i=1}^{m} w_i u_i$$

$$w_i = \frac{\mu_U(u_i)}{\sum_{i=1}^{m} \mu_U(u_i)} \qquad \sum_{i=1}^{m} w_i = 1$$

So the defuzzified value is the weighted average of all the input values, where the weights are normalized memberships.

Example: Let the common normalized domain $U = \{1, 2, \ldots, 8\}$ and $\mu_U = 0/1 + 0.2/2 + 0.5/3 + 0.8/4 + 1/5 + 0.5/6 + 0.2/7 + 0/8$

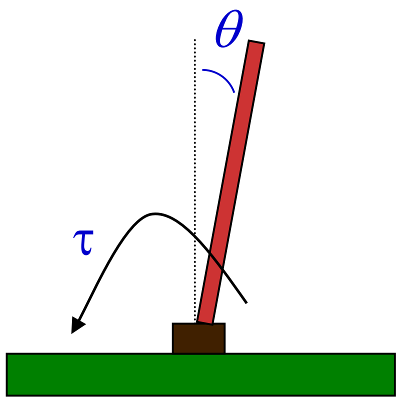
Then, u^*(Center-of-Area) =

$$\frac{0 \cdot 1 + 0.2 \cdot 2 + 0.5 \cdot 3 + 0.8 \cdot 4 + 1 \cdot 5 + 0.5 \cdot 6 + 0.2 \cdot 7 + 0 \cdot 8}{0 + 0.2 + 0.5 + 0.8 + 1 + 0.5 + 0.2 + 0} = 4.53$$

A Fuzzy Control Problem – Inverted Pendulum

- Design and analyse a fuzzy controller for a simplified version of the inverted pendulum.
- The differential equation describing the system is given as

$$-ml^2 \frac{d^2\theta}{dt^2} + (mgl)\sin(\theta) = \tau = u(t) \tag{1}$$



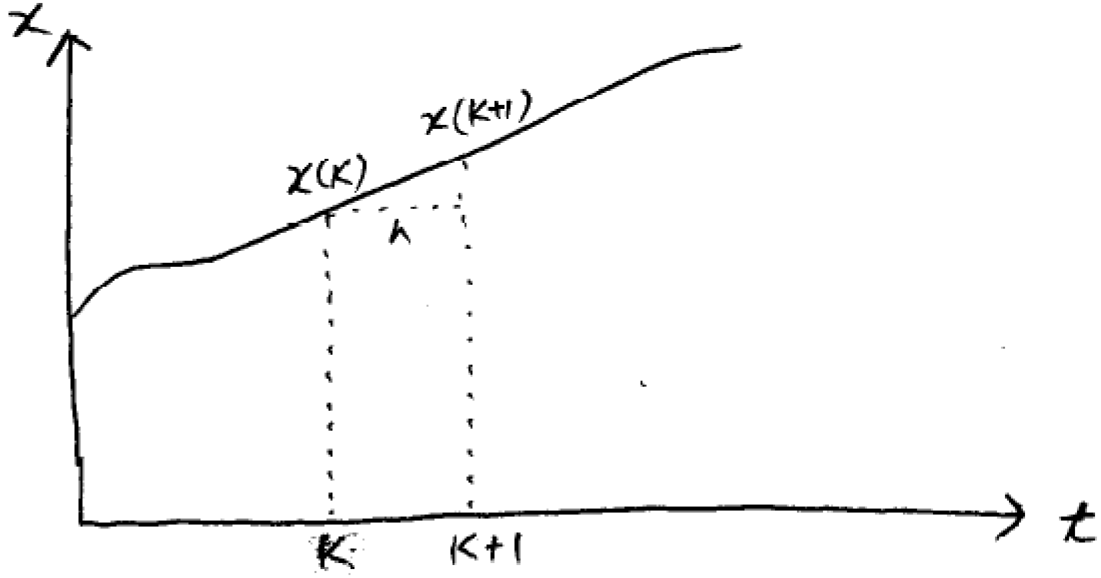
m = mass of the pole located at the tip point of the pendulum

l = length of the pendulum

θ = deviation angle from vertical in the clockwise direction

$\tau = u(t)$ = torque applied to the pole

t = time; g = gravitational acceleration constant

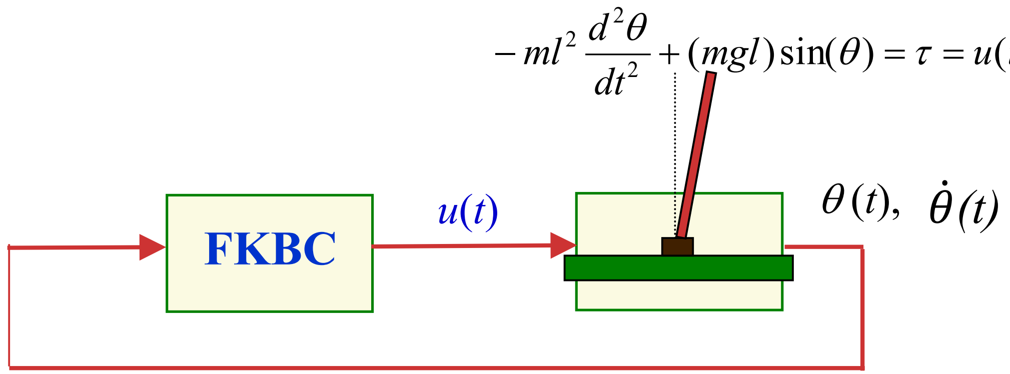


$$\frac{dx}{dt} = \frac{x(K+1) - x(K)}{h}$$

assume $h=1$ (step size and discrete), then

$$\frac{dx}{dt} = x(K+1) - x(K)$$

- Let us consider a simple FKBC system:



- For simplicity and easy calculation, assuming $x_1 = \theta$ and $x_2 = d\theta/dt$, and $\sin(\theta) = \theta,\ l = g,\ ml^2 = 1)$, $h = 1$

[2]

$$\frac{dx_1}{dt} = \frac{x_1(k+1) - x_1(k)}{h} = x_2$$

$$\Rightarrow x_1(k+1) - x_1(k) = x_2(k)$$

$$\Rightarrow \boxed{x_1(k+1) = x_1(k) + x_2(k)}$$

k is iteration index

from [1] & [2]

$$\frac{dx_2}{dt} = (g/l)x_1 - (1/ml^2)u(t) = x_1 - u(t)$$

$$\Rightarrow x_2(k+1) - x_2(k) = x_1(k) - u(k)$$

$$\Rightarrow \boxed{x_2(k+1) = x_1(k) + x_2(k) - u(k)}$$

- Therefore,

inputs of FKBC

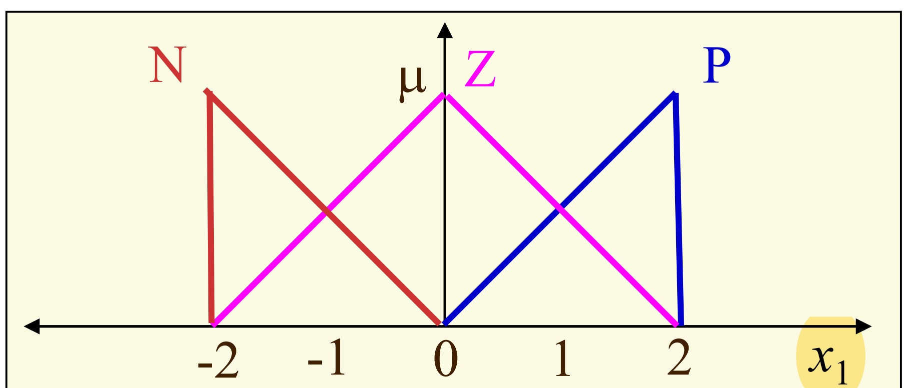
output of FKBC

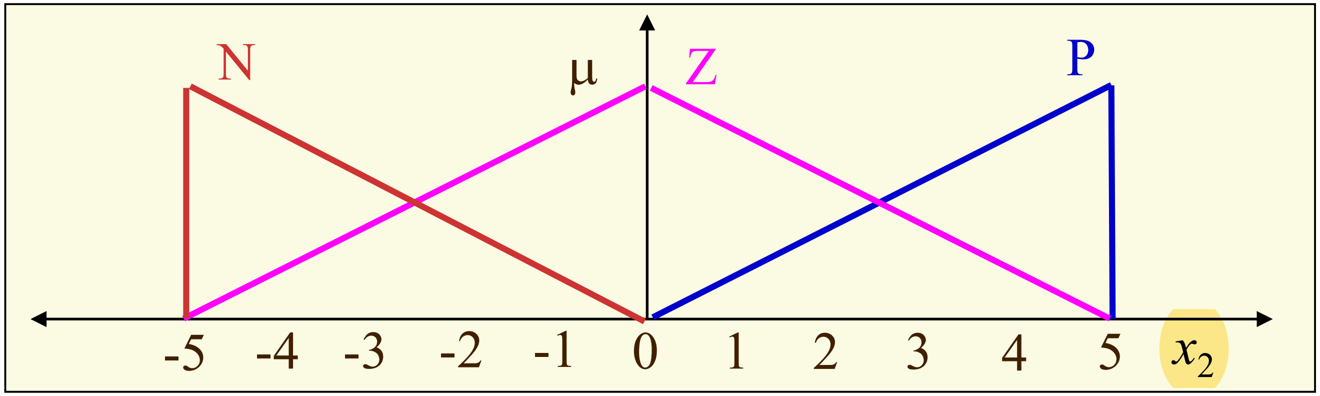
$$x_1(k+1) = x_1(k) + x_2(k)$$
$$x_2(k+1) = x_1(k) + x_2(k) - u(k)$$

- Assume that the universe of discourse for the two variables are:

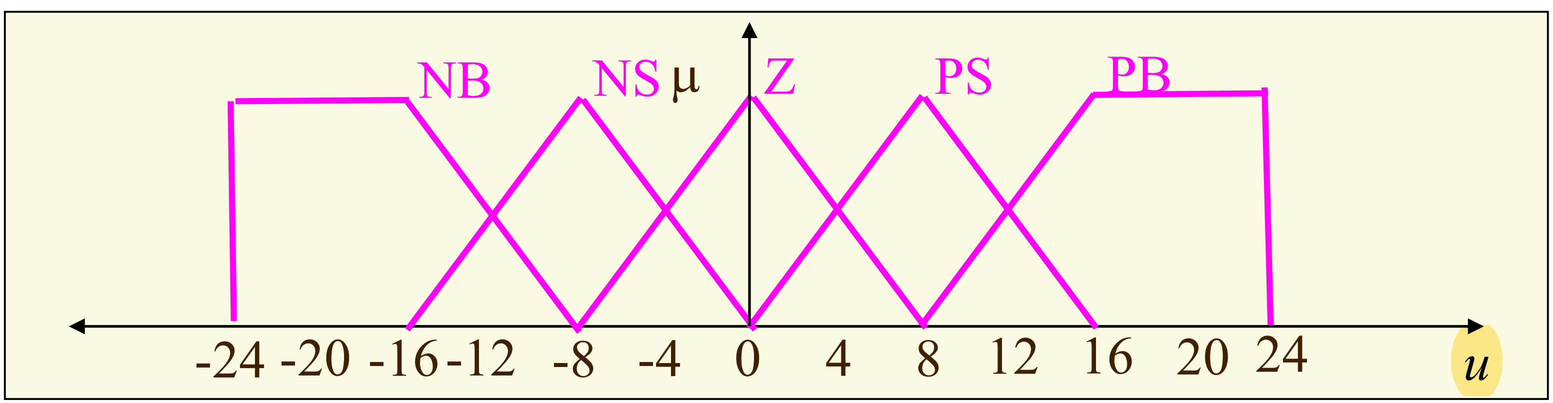
$$-2 \le x_1 \le 2 \text{ and } -5 \le x_2 \le 5$$

- Step 1: Construct three membership functions for x_1 and x_2 on its universe, positive (P), zero(Z) and negative (N)



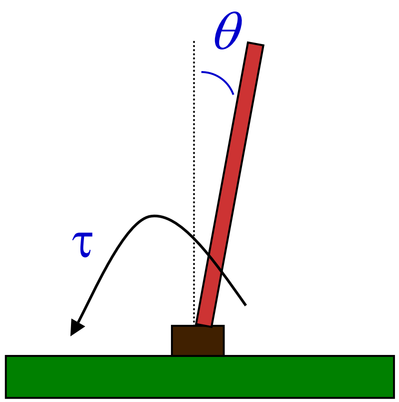


♦ Step 2: To partition the control space (output), we construct five membership functions for $u(k)$ on its universe [-24, 24].



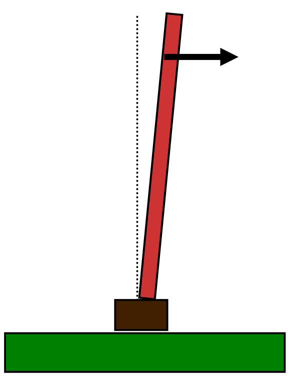
♦ Step 3: Construct 9 rules in a 3 × 3 Fuzzy rule-based table, which involves θ and $\dot{\theta}$ in order to stabilize the inverted pendulum system. The entries are control actions, $u(k)$.

x_1 \ x_2	P	Z	N
P	?	?	?
Z	?	?	?
N	?	?	?



How to formulate the rules?

If you cannot get an expert to help you, you can put yourself in the shoe of the operator!



If the error is positive (P) and the speed is positive (P), what would you do?

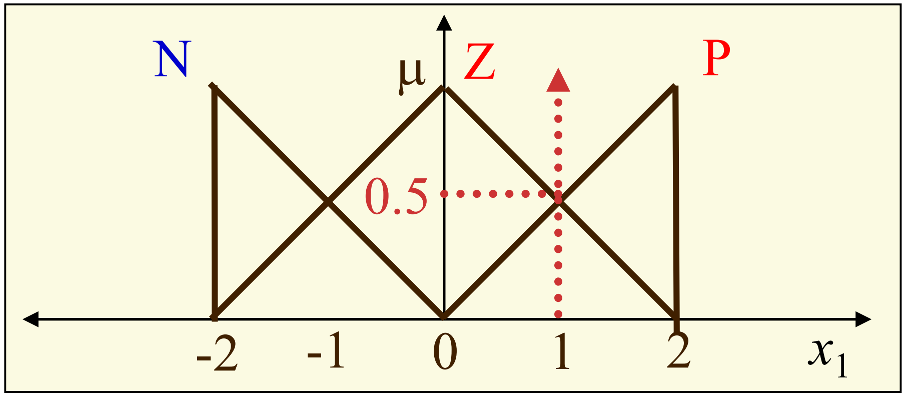
Apply a positive big (PB) torque to stop it from falling down!

♦ Step 3: Construct 9 rules in a 3 × 3 Fuzzy rule-based table, which involves θ and $\dot{\theta}$ in order to stabilize the inverted pendulum system. The entries are control actions, $u(k)$.

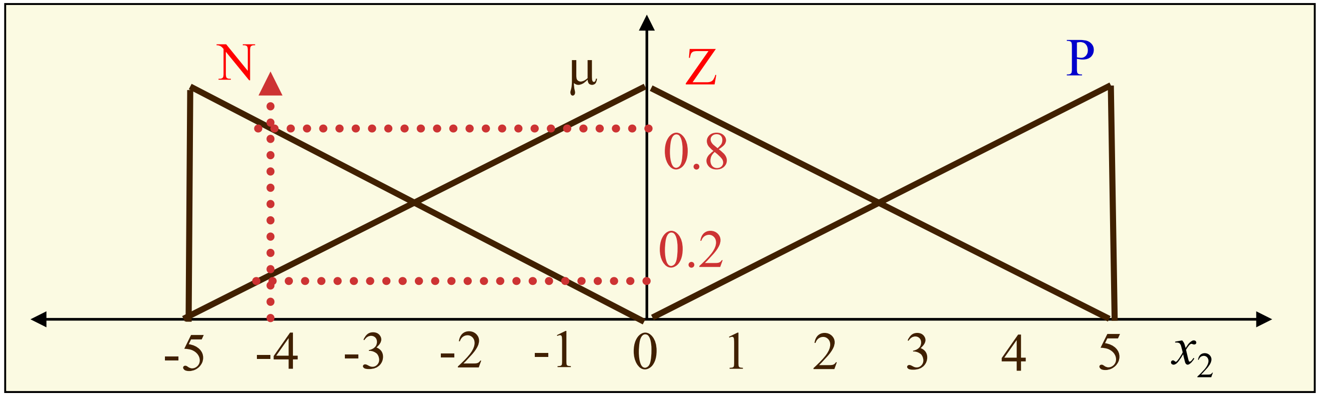
x_1 \ x_2	P	Z	N
P	PB	P	Z
Z	P	Z	N
N	Z	N	NB

By default, Table form rules mean the connective is "and"

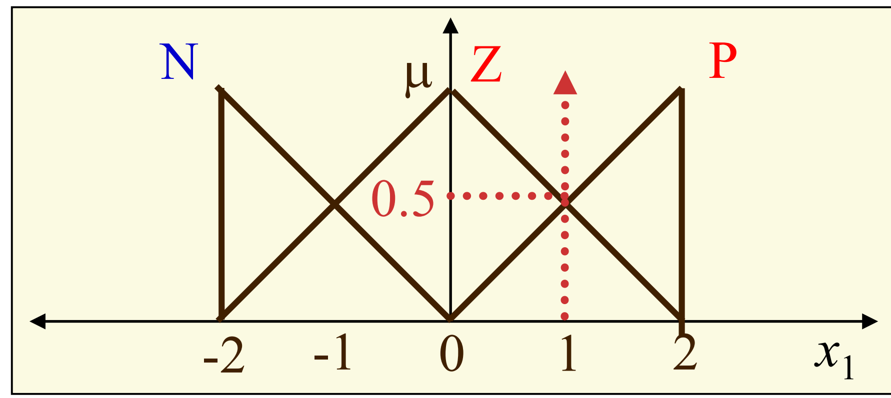
Given that the initial conditions are $x_1(0) = 1$ and $x_2(0) = -4$,



$x_1(0) = 1$



$x_2(0) = -4$,

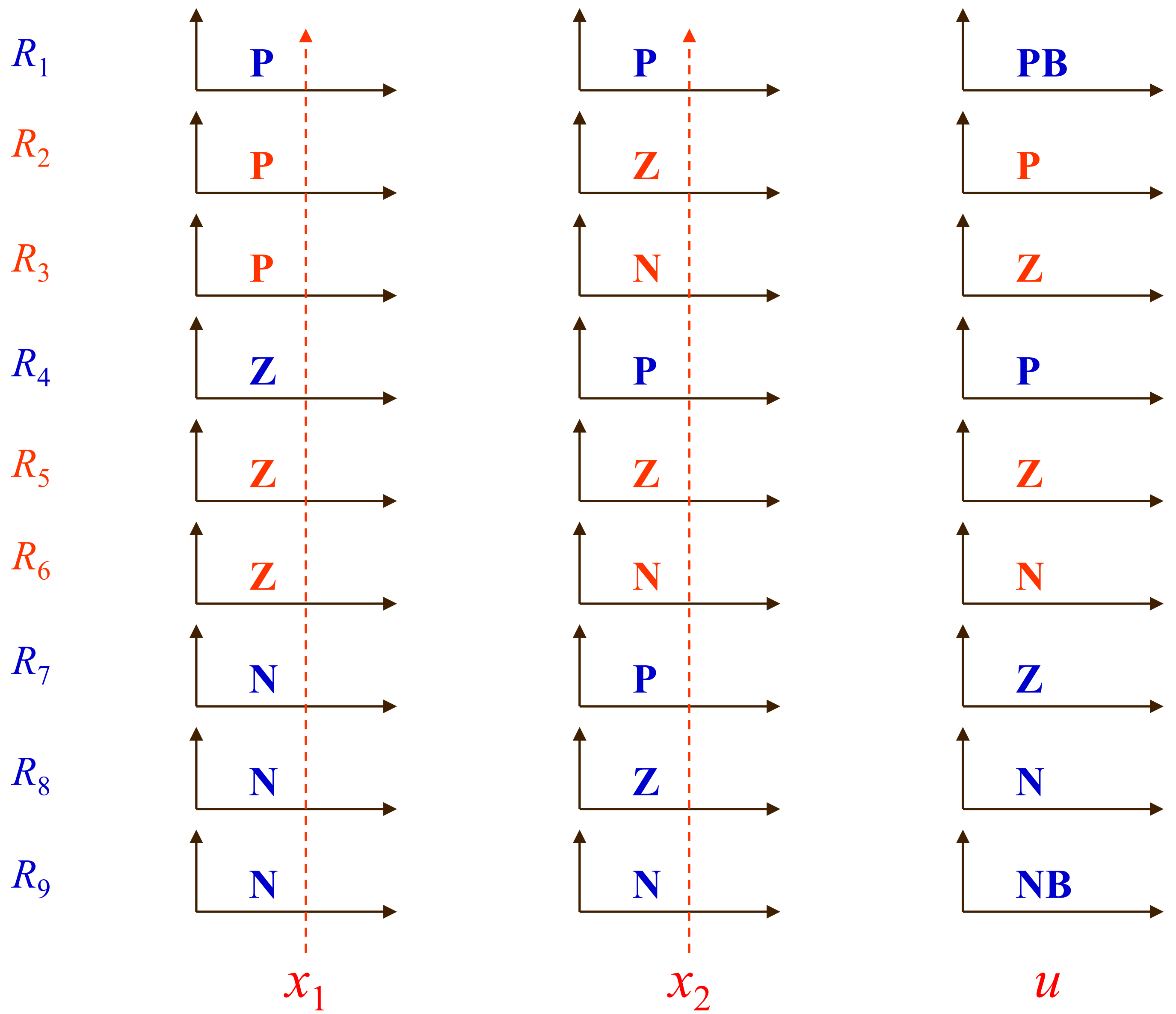


$x_1(0) = 1$

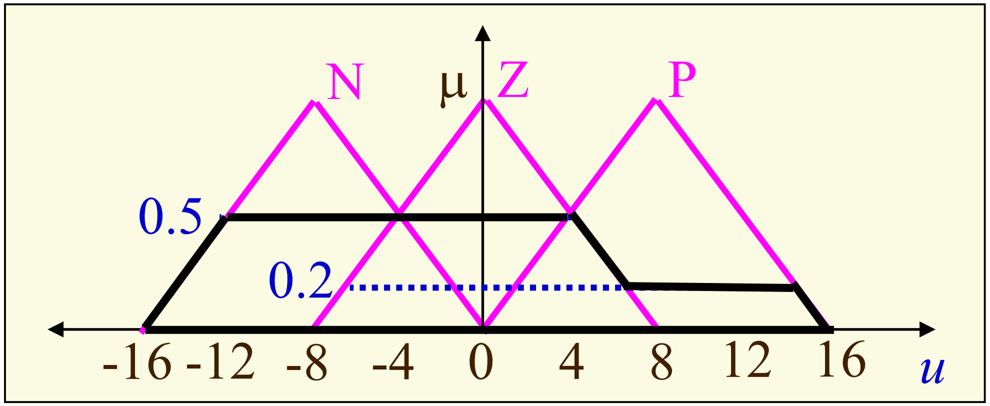
♦ Step 4: We simulate the system for 4 cycles, i.e., $k = 0, 1, 2, 3$

If $(x_1 = \text{P})$ and $(x_2 = \text{Z})$ then $(u = \text{P})$ min(0.5, 0.2) = 0.2 (P)
If $(x_1 = \text{P})$ and $(x_2 = \text{N})$ then $(u = \text{Z})$ min(0.5, 0.8) = 0.5 (Z)
If $(x_1 = \text{Z})$ and $(x_2 = \text{Z})$ then $(u = \text{Z})$ min(0.5, 0.2) = 0.2 (Z)
If $(x_1 = \text{Z})$ and $(x_2 = \text{N})$ then $(u = \text{N})$ min(0.5, 0.8) = 0.5 (N)

x_1 \ x_2	P	Z	N
P	PB	P	Z
Z	P	Z	N
N	Z	N	NB



R_1
P
P
PB
R_2
P
Z
P
R_3
P
N
Z
R_4
Z
P
P
R_5
Z
Z
Z
R_6
Z
N
N
R_7
N
P
Z
R_8
N
Z
N
R_9
N
N
NB
x_1
x_2
u



$$u^* = \frac{0.25(-14) + 0.5(-12-10-8-6-4-2+0) + 0.5(2+4) + 0.25(6) + 0.2(8+10+12+14)}{5.8}$$

$$= \frac{-11.2}{5.8} \approx -2 \quad \text{(Center-of-area)}$$

The 1[st] simulation cycle is completed. Find the states for next iteration:

$$x_1(1) = x_1(0) + x_2(0) = 1 - 4 = -3$$

$$x_2(1) = x_1(0) + x_2(0) - u(0) = 1 - 4 - (-2) = -1$$

$$x_1(k+1) = x_1(k) + x_2(k)$$

$$x_2(k+1) = x_1(k) + x_2(k) - u(k)$$

For second cycle,

If $(x_1 = \text{N})$ and $(x_2 = \text{N})$ then $(u = \text{NB})$ $\min(1, 0.2) = 0.2$ (NB)

If $(x_1 = \text{N})$ and $(x_2 = \text{Z})$ then $(u = \text{N})$ $\min(1, 0.8) = 0.8$ (N)

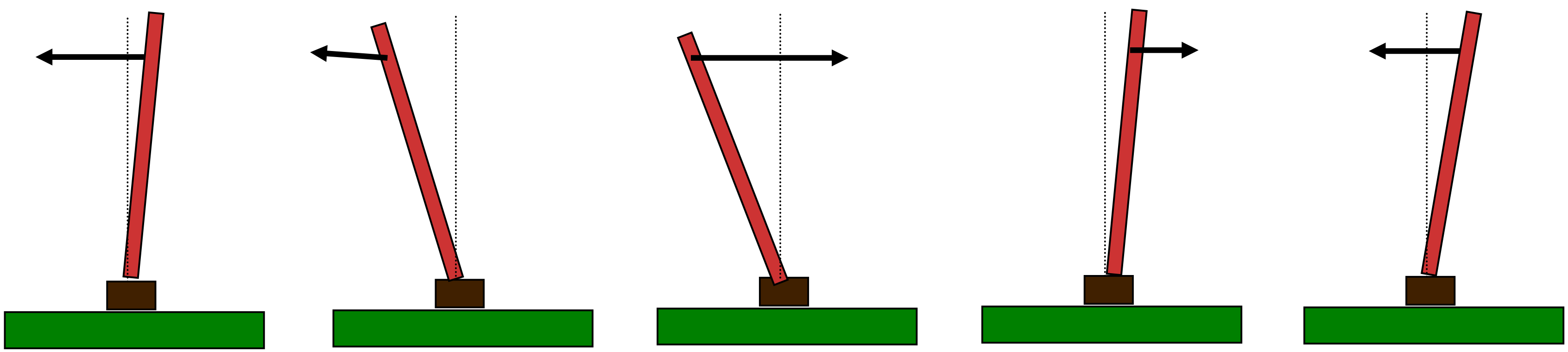
$x_1 \backslash x_2$	P	Z	N
P	PB	P	Z
Z	P	Z	N
N	Z	N	NB

The defuzzified value is u = -9.6, and $x_1(2) = -4$, $x_2(2) = 5.6$

Proceed with the calculation, we get

$$x_1(3) = 1.6,\ \ x_2(3) = 1.6$$

$$x_1(4) = 3.2,\ \ x_2(4) = -2.08$$



$x_1(0) = 1$	$x_1(1) = -3$	$x_1(2) = -4$	$x_1(3) = 1.6$	$x_1(4) = 3$
$x_2(0) = -4$	$x_2(1) = -1$	$x_2(2) = 5.6$	$x_2(3) = 1.6$	$x_2(4) = -2.08$