FUZZY LOGIC-IV
Fuzzy Logic Control
MODULE 10

Dr. R.B.Ghongade,
VIIT, Pune-411048

Agenda

- Why Fuzzy Logic Control?
- Typical fuzzy control systems
- Classical and Fuzzy PID controller
- Architecture of a Mamdani type Fuzzy Control System

Why Fuzzy Logic Control?

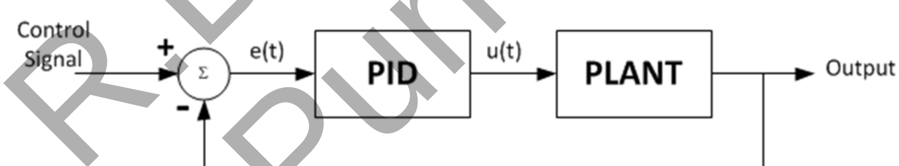
- Fuzzy logic is a technique to embody human like thinking into a control system
- A fuzzy controller can be designed to emulate human deductive thinking i.e., the process , humans use to infer conclusions from what they know
- Traditional control approach requires formal modeling of the physical reality
- There are many systems where humans operate the plant based on their experience
- Better the experience, better is the control mechanism
- We can utilize such principle of human way of decision making into designing control systems

- Fuzzy control incorporates ambiguous human logic into computer programs
- It suits control problems that
  - cannot be easily represented by mathematical models
  - have weak model(physics of systems is poorly understood)
  - have parameter variation problem
  - unavailable or incomplete data
  - are very complex
  - we have good qualitative understanding of plant or process operation
- Because of its unconventional approach, design of such controllers leads to faster development/implementation cycles

Typical Fuzzy Control Systems

- Two popular typical fuzzy control systems are Mamdani type and Tagaki-Sugeno (T-S) type
- Mamdani Type:
  - Employ fuzzy sets in the consequent part of the rule
  - Incremental control action is described in the consequent part of each rule
- T-S Type:
  - Employ function of the input fuzzy linguistic variables as the consequent of the rules
  - A fuzzy dynamic model is expressed in the form of local rules

A Classical PID Controller



- Objective: Output should follow the command signal
- PID Control Law:

$$u(t) = K\left[e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}\right]$$

- Error is multiplied by K, error is integrated over time with a time constant T_i and error is differentiated
- In discrete domain $t = k\Delta T$ is the sampling time interval and $k$is the sampling instant

$$\boldsymbol{u(k) = K\left[e(k) + \frac{1}{T_i}[e(k) + e(k-1) + \cdots]\Delta T + T_d\frac{e(k) - e(k-1)}{\Delta T}\right]}$$

- Consider a simpler PI controller

$$u(t) = K\left[e(t) + \frac{1}{T_i}\int_0^t e(t)dt\right]$$

- Its discrete form is

$$\boldsymbol{u(k) = K\left[e(k) + \frac{1}{T_i}[e(k) + e(k-1) + \cdots]\Delta T\right]}$$

- But

$$\boldsymbol{u(k-1) = K\left[e(k-1) + \frac{1}{T_i}[e(k-1) + e(k-2) + \cdots]\Delta T\right]}$$

- Giving

$$\boldsymbol{u(k) - u(k-1) = K\left[e(k) - e(k-1) + \frac{1}{T_i}[e(k)]\Delta T\right]}$$

- Thus we can write

$$\boldsymbol{u(k) = u(k-1) + \Delta u(k)}$$

Where $\boldsymbol{\Delta u(k) = f(e, \Delta e)}$

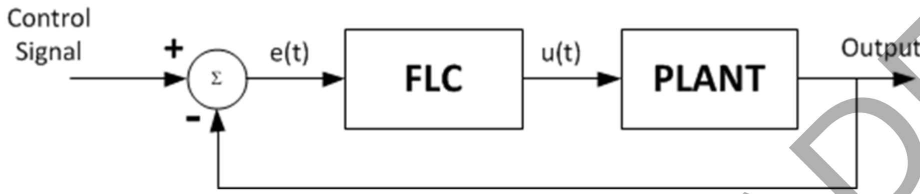
- Hence present control action is equal to previous control action +incremental control action
- This is the fundamental concept which led to the development of Fuzzy Logic Controller(FLC)
- It can be shown that:

$$u(k) = u(k-1) + \Delta u(k)$$

Where $\Delta u(k) = f[e(k), \Delta e(k), \cdots]$

- Means incremental control action is some function of error, change in error and higher order terms

Fuzzy PID Controller (Mamdani Type)



- Similar to classical PID controller a fuzzy PID controller can have the following structure:

$$u(k) = u(k-1) + \Delta u(k)$$

- However , $\Delta u(k)$ is obtained using fuzzy rule base that provides incremental control action , which is a function of two fuzzy variables , $\boldsymbol{e}$ and $\Delta \boldsymbol{e}$
- Classical PID controller is linear
- Fuzzy PID controller is non-linear
- Our goal to compute what is the incremental control action in FLC of Mamdani Type

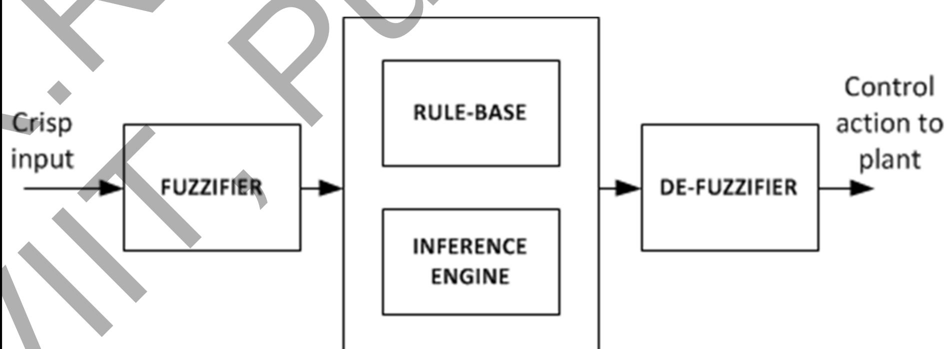
Example of a Rule-base

- Temperature control system employing a fuzzy proportional controller :$u(k) = Ke(k)$
- RULE 1: IF error is POSITIVE-HIGH, THEN keep the heater on for a LONGER duration
- RULE 2: IF error is POSITIVE-LOW, THEN keep the heater on for a SHORTER duration
- Antecedent part has only ONE fuzzy variable since $u(k) = Ke(k)$
- We do not have incremental control action but a DIRECT control action

- Temperature control system employing a fuzzy PI controller :$u(k) = u(k-1) + \Delta u(k)$
- RULE 1: IF error is POSITIVE-HIGH and change in error is POSITIVE-HIGH, THEN keep the heater on for a LONGER duration
- RULE 2: IF error is POSITIVE-LOW and change in error is POSITIVE-LOW, THEN keep the heater on for a SHORTER duration

$$u(k) = u(k-1) + \underbrace{\Delta u(k)}_{f(e,\Delta e)}$$

Architecture of a FLC: Mamdani Type



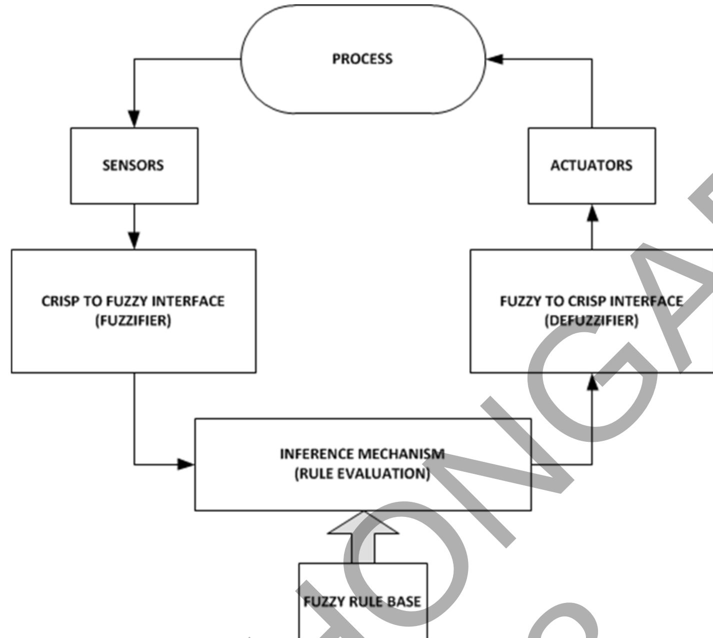
Operation explained

- Let 40°C crisp value be fuzzified into linguistic values $\{COLD, MODERATE, HOT, \cdots\}$
- Each linguistic value is associated with a specific membership function
- Once the data is fuzzified , it goes to the rule-base and then using the inference mechanism, a fuzzy inference is derived
- After we get the fuzzy inference about the control action we place a de-fuzzifier which converts the fuzzy control action to a crisp control action

Principal Design Parameters of a FLC

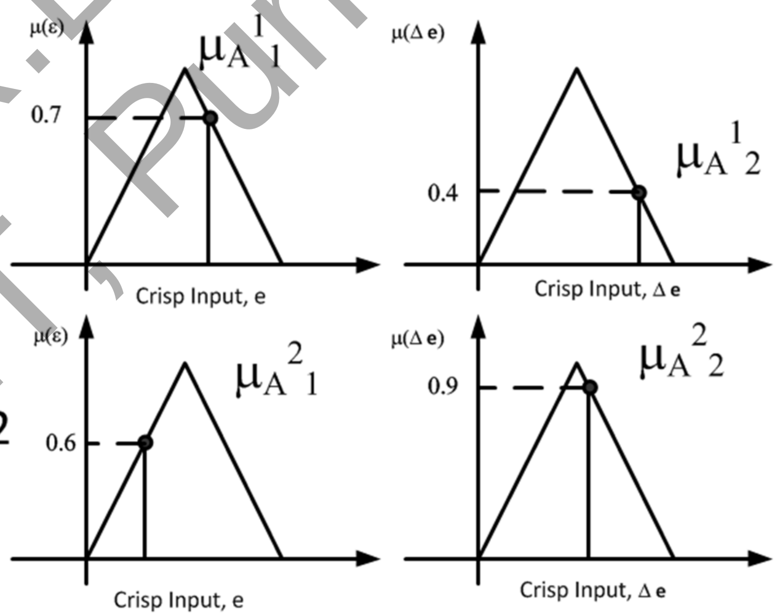
- Fuzzification strategies and the interpretation of a fuzzification operator(fuzzifier)
- Database:
  - Discretization/normalization of universe of discourse
  - Fuzzy partitioning of input and output spaces
  - Completeness
  - Choice of membership function of a primary fuzzy set
- Rule-base
  - Choice of process state(input) variables and control(output) variables
  - Source and derivation of fuzzy control rules
  - Types of fuzzy control rules
  - Completeness of fuzzy control rules
- Fuzzy Inference mechanism
- De-fuzzyfication operator (defuzzifier)

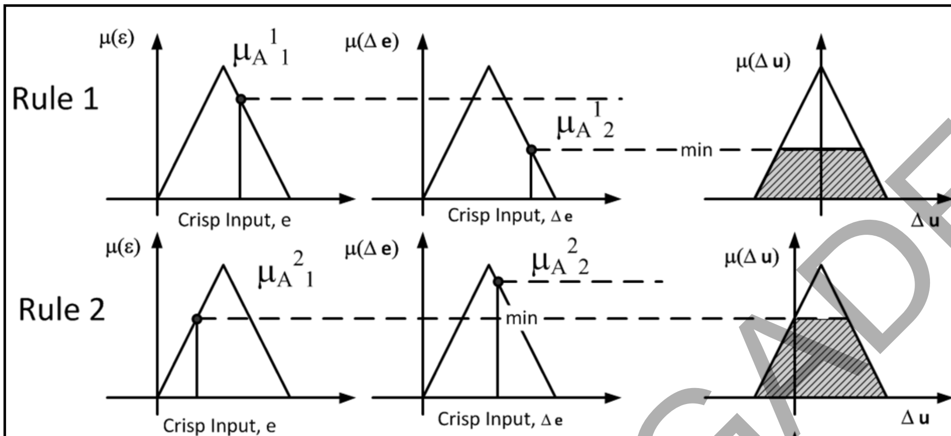
Complete Architecture of FLC

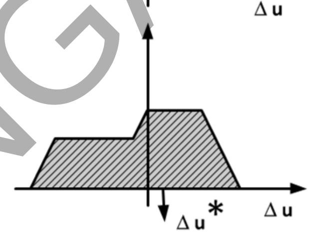


Fuzzification(graphical)

Rule 2







$$\Delta u^{*} = \frac{\int \mu(\Delta u)\Delta u d(\Delta u)}{\int \mu(\Delta u) d(\Delta u)}$$

TRADITIONAL vs FUZZY LOGIC CONTROLLERS (FLC)

- Traditional control method requires accurate, often complex system of equations
- Fuzzy logic proponents claim fuzzy systems are much closer to human control process
- Neural networks, fuzzy logic, and evolutionary systems share common thread
  - Do not rely on mathematical systems descriptions
- Interestingly, biological control systems do not rely on complex mathematical models
- Traditional camp treats these new approaches with skepticism for lack of mathematical rigor

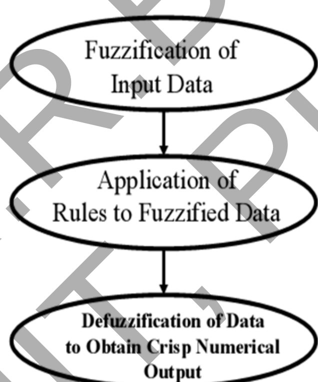
FUZZY CONTROL

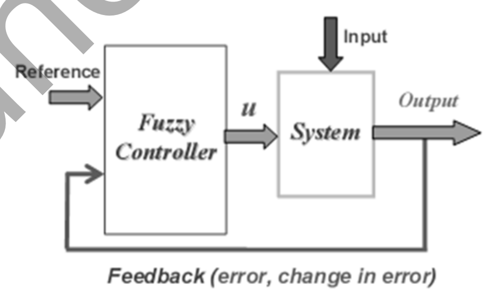
- Fuzzy controllers are the most important applications of fuzzy theory.
- They work rather different than conventional controllers:
  - expert knowledge **is used instead of differential equations to describe a system.**
  - **this knowledge can be expressed in a very natural way using linguistic variables, which are described by fuzzy sets.**

FUZZY LOGIC CONTROLLERS (FLC)

- **Although application of fuzzy logic to industrial problems has often produced results superior to classical control,** the design procedures are limited by the heuristic rules of the system.
- **This** implicit assumption limits **the application of fuzzy logic controller.**
  - **moreover, the majority of FLCs to date have been** static **and based upon** knowledge derived from imprecise heuristic knowledge of experienced operators.

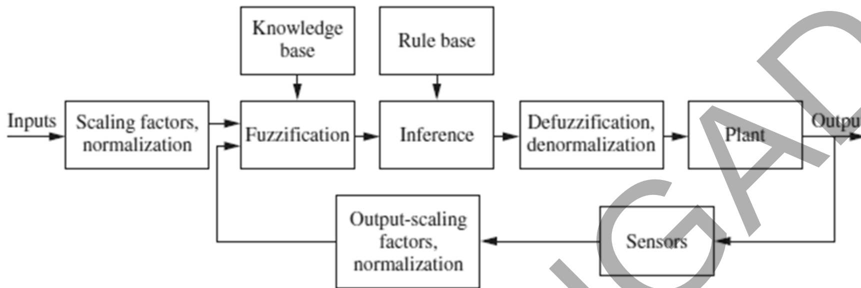
- The fuzzy logic-based approach to solving problems in control has been found to excel in those systems which are very complex, highly nonlinear and with parameter uncertainty.
- We may view a fuzzy logic controller as a real time expert system that employs fuzzy logic to analyze input to output performance.
- Indeed, they provide a means of converting a linguistic control strategy derived from expert knowledge into automatic control strategies and give us a means of interrogating the control system evolution and system performance.





- Inputs should be sufficient to describe current status of system
- Need appropriate relationships between inputs and outputs
- Represent output control values in suitable form for system control

ARCHITECTURE OF FLC



DESCRIPTION OF FLC COMPONENTS

A **Fuzzy Logic Controller** usually consists of:

- A **fuzzification unit** which maps measured inputs of crisp value into fuzzy linguistic values to be used by a fuzzy reasoning mechanism.
- A **knowledge base (KB)** which is the collection of expert control knowledge required to achieve the control objective.
- A **fuzzy reasoning mechanism** that performs various fuzzy logic operations to infer the control action for the given fuzzy inputs.
- A **defuzzification unit** which converts the inferred fuzzy control action into the required crisp control values to be entered into the system process.

FUZZY LOGIC CONTROLLER

- Control inputs,
- Fuzzy sets (membership functions) of inputs,
- Rules,
- Control outputs,
- Fuzzy sets (membership functions) of outputs,
- Fuzzy reasoning,
- Defuzzification.

- Control inputs
  - ***error, change in error, process input and output, setpoint***
- Fuzzy sets (membership functions) of inputs
  - ***number of sets, type of membership functions, location of membership functions***
- Rules
- Control outputs
  - ***control, change in control, control parameters, setpoint***

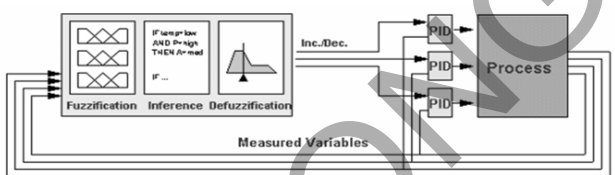
- Fuzzy sets (membership functions) of outputs
  - *number of sets, location of membership functions*
- Fuzzy reasoning
  - *max-min, sum-product, etc.*
- Defuzzification
  - *center of gravity, max, etc.*

Steps in designing a simple fuzzy control system

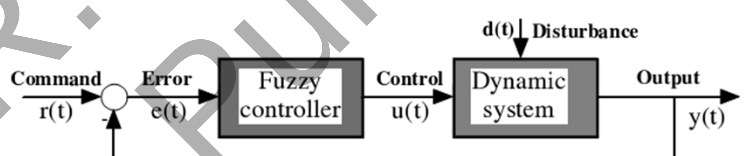
1. Identify the variables (inputs, states, and outputs) of the plant
2. Partition the universe of discourse or the interval spanned by each variable into a number of fuzzy subsets, assigning each a linguistic label (subsets include all the elements in the universe)
3. Assign or determine a membership function for each fuzzy subset.
4. Assign the fuzzy relationships between the inputs' or states' fuzzy subsets on the one hand and the outputs' fuzzy subsets on the other hand, thus forming the rule-base
5. Choose appropriate scaling factors for the input and output variables in order to normalize the variables to the [0, 1] or the [−1, 1] interval
6. Fuzzify the inputs to the controller
7. Use fuzzy approximate reasoning to infer the output contributed from each rule
8. Aggregate the fuzzy outputs recommended by each rule
9. Apply defuzzification to form a crisp output

FLC AND PID CONTROLLERS

- Fuzzy logic lets engineers design supervisory multi-variable controllers from operator experience and experimental results rather than from mathematical models.
- Each single process variable is kept constant by a PID controller, while the set values for the PID controller come from the fuzzy logic system. This arrangement is typical for cases like control of several temperature zones of an oven or control of oxygen concentrations in different zones of a wastewater basin. In other cases, it could be reasonable to develop the complete closed loop control solution in a fuzzy system.

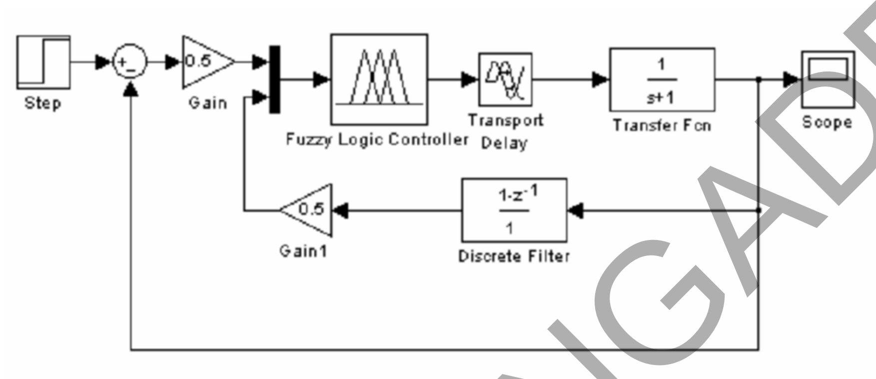


FUZZY LOGIC CONTROLLER (MAMDANI)



de\e	NL	NS	ZE	PS	PL
NL	PL	PL	PS	ZE	NS
NS	PL	PS	PS	ZE	NS
ZE	PL	PS	ZE	NS	NL
PS	PS	ZE	NS	NS	NL
PL	ZE	NS	NS	NL	NL

MATLAB SCREENSHOT OF FLC - SIMULINK BLOCK



ADVANTAGES OF FLC

- No need for mathematical model
- Less sensitive to system fluctuations.
- Design objectives difficult to express mathematically can be incorporated in a fuzzy controller by linguistic rules.
- Implementation is simple and straight forward.

APPLICATIONS OF FLC

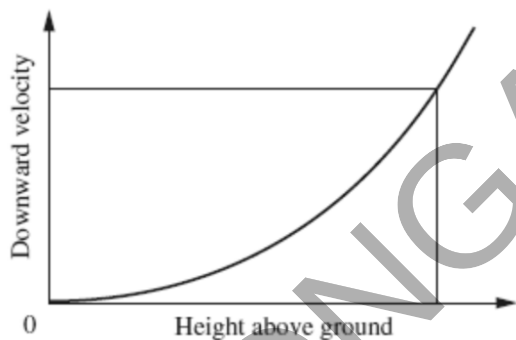
- Ride smoothness control
- Camcorder auto-focus and jiggle control
- Braking systems
- Copier quality control
- Rice cooker temperature control
- High performance drives
- Air-conditioning systems

1. traffic control;
2. steam engine;
3. aircraft flight control;
4. missile control;
5. adaptive control;
6. liquid-level control;
7. helicopter model;
8. automobile speed controller;
9. braking system controller;
10. process control (includes cement kiln control);
11. robotic control;
12. elevator (auto lift) control;
13. automatic tuning control;
14. cooling plant control;
15. water treatment;
16. boiler control;
17. nuclear reactor control;
18. power systems control;

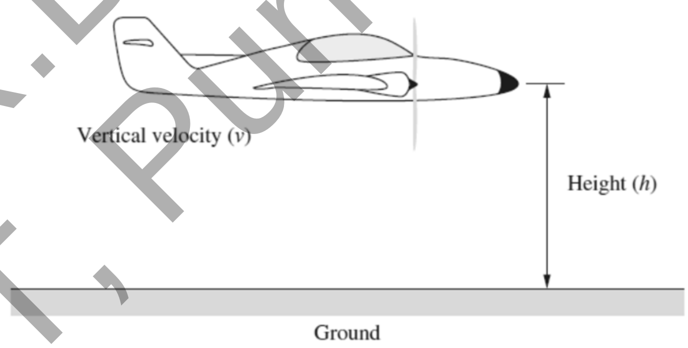
Example of FLC

- Aircraft Landing problem
- Necessary to simulate the final descent approach
- When aircraft lands:
  - downward velocity α square of height
- Thus, at higher altitudes, a large downward velocity is desired
- As the height (altitude) diminishes, the desired downward velocity gets smaller and smaller
- In the limit, as the height becomes vanishingly small, the downward velocity also goes to zero
- The aircraft will descend from altitude promptly but will touch down very gently to avoid damage.

The desired profile of downward velocity versus altitude



Aircraft landing control problem

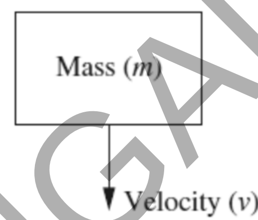


- Two state variables for this simulation will be the height above ground, ***h***, and the vertical velocity of the aircraft, ***v***
- The control output will be a force that, when applied to the aircraft, will alter its height, ***h***, and velocity, ***v***

The differential control equations

- Mass *m* moving with velocity *v* has momentum:
$$p = mv$$
- If no external forces are applied, the mass will continue in the same direction at the same velocity, *v*
- If a force *f* is applied over a time interval Δt, a change in velocity will be:
$$v = f * \Delta t \quad /m$$
- If we let Δt = 1.0 (s) and *m* = 1.0 (lb s^2/ ft), we obtain:
$$v = f \text{ (lb), or}$$

"the change in velocity is proportional to the applied force"



Simple momentum model

- In difference notation, we get:
$$v_{i+1} = v_i + f_i$$
$$h_{i+1} = h_i + v_i \cdot \Delta t$$

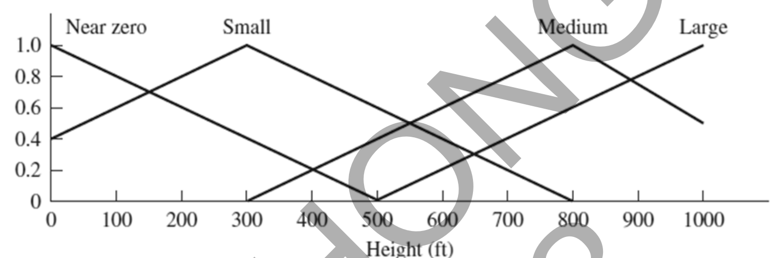
where v_{i+1} is the new velocity, v_i is the old velocity, h_{i+1} is the new height, and h_i is the old height

- These two "control equations" define the new value of the state variables v and h in response to control input and the previous state variable values

Construct membership functions for the height, *h*, the vertical velocity, *v*, and the control force, *f*

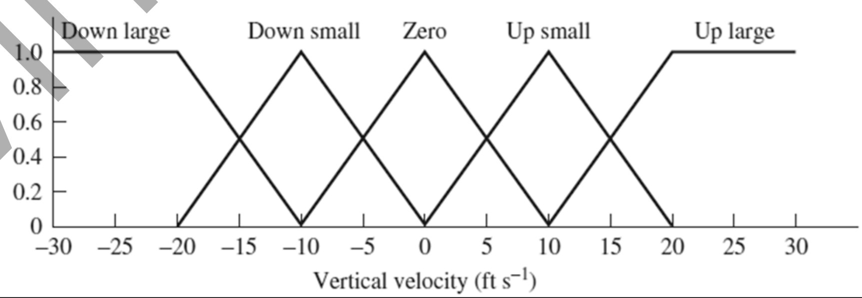
- *Step 1 a*. Define membership functions for state variable $\boldsymbol{h}$

Height (ft)											
	0	**100**	**200**	**300**	**400**	**500**	**600**	**700**	**800**	**900**	**1000**
Large (L)	**0**	**0**	**0**	**0**	**0**	**0**	**0.2**	**0.4**	**0.6**	**0.8**	**1**
Medium (M)	**0**	**0**	**0**	**0**	**0.2**	**0.4**	**0.6**	**0.8**	**1**	**0.8**	**0.6**
Small (S)	**0.4**	**0.6**	**0.8**	**1**	**0.8**	**0.6**	**0.4**	**0.2**	**0**	**0**	**0**
Near zero (NZ)	**1**	**0.8**	**0.6**	**0.4**	**0.2**	**0**	**0**	**0**	**0**	**0**	**0**



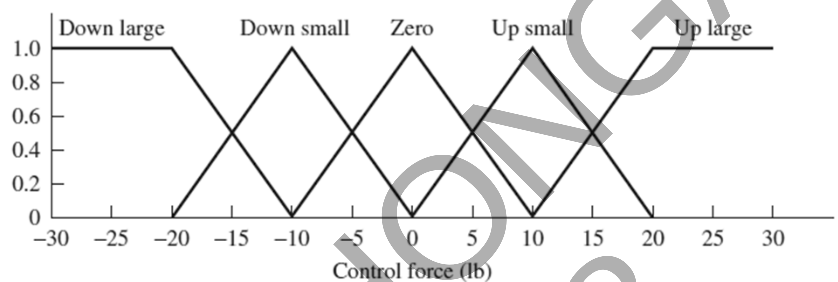
- *Step 1 b*. Define membership functions for state variable $\boldsymbol{v}$

Vertical velocity (ft/s)													
	−30	**−25**	**−20**	**−15**	**−10**	**−5**	**0**	**5**	**10**	**15**	**20**	**25**	**30**
Up large (UL)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0.5**	**1**	**1**	**1**
Up small (US)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0.5**	**1**	**0.5**	**0**	**0**	**0**
Zero (Z)	**0**	**0**	**0**	**0**	**0**	**0.5**	**1**	**0.5**	**0**	**0**	**0**	**0**	**0**
Down small (DS)	**0**	**0**	**0**	**0.5**	**1**	**0.5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Down large (DL)	**1**	**1**	**1**	**0.5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**



- *Step 2*. Define membership functions for control output ***f***

	Output force (lb)												
	−30	**−25**	**−20**	**−15**	**−10**	**−5**	**0**	**5**	**10**	**15**	**20**	**25**	**30**
Up large (UL)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0.5**	**1**	**1**	**1**
Up small (US)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0.5**	**1**	**0.5**	**0**	**0**	**0**
Zero (Z)	**0**	**0**	**0**	**0**	**0**	**0.5**	**1**	**0.5**	**0**	**0**	**0**	**0**	**0**
Down small (DS)	**0**	**0**	**0**	**0.5**	**1**	**0.5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Down large (DL)	**1**	**1**	**1**	**0.5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**



- *Step 3.* Define the rules and summarize them in an **Fuzzy Associative Memory** (FAM) table (The values in the FAM table, of course, are the control outputs)

	Velocity				
Height	**DL**	**DS**	**Zero**	**US**	**UL**
L	Z	DS	DL	DL	DL
M	US	Z	DS	DL	DL
S	UL	US	Z	DS	DL
NZ	UL	UL	Z	DS	DS

- *Step 4* . Define the initial conditions, and conduct a simulation for four cycles. Since the task at hand is to control the aircraft's vertical descent during approach and landing, we will start with the aircraft at an altitude of 1000 feet, with a downward velocity of −20 ft s−1.

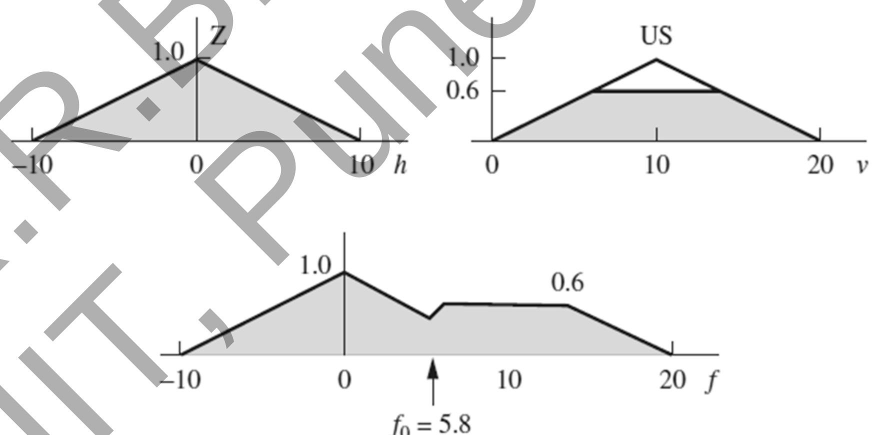
- We will use the following equations to update the state variables for each cycle:

$$v_{i+1} = v_i + f_i$$
$$h_{i+1} = h_i + v_i$$

CYCLE #0

- Initial height, h_0 : 1000 ft
- Initial velocity, v_0 : −20 ft s−1
- Control f_0 : to be computed
- Height h fires **L** at 1.0 and **M** at 0.6
- Velocity v fires only **DL** at 1.0

Height		Velocity		Output
L (1.0)	AND	DL (1.0)	⇒	Z (1.0)
M (0.6)	AND	DL (1.0)	⇒	US (0.6)

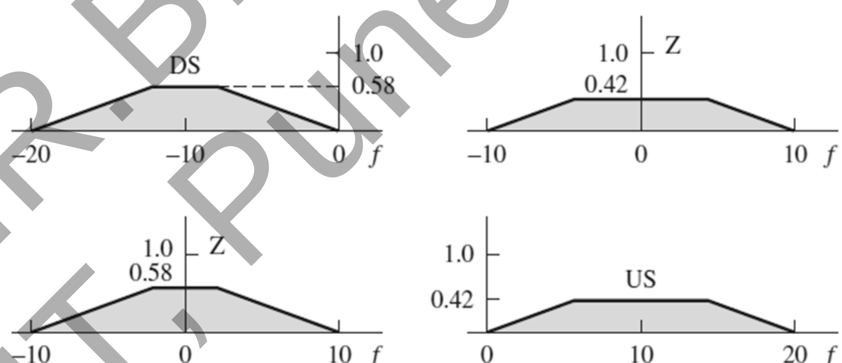


- We defuzzify using the centroid method and get f_0 = 5.8 lb. This is the output force computed from the initial conditions

CYCLE #1

- h_1 =h_0+v_0=1000+(-20)=980 ft
- v_1 =v_0+f_0= (-20)+(5.8)=-14.2 ft/s
- Height h_1 fires **L** at 0.96 and **M** at 0.64
- Velocity v_1 fires **DS** at 0.58 and **DL** at 0.42

Height		Velocity		Output
L (0.96)	AND	DS (0.58)	⇒	DS (0.58)
L (0.96)	AND	DL (0.42)	⇒	Z (0.42)
M (0.64)	AND	DS (0.58)	⇒	Z (0.58)
M (0.64)	AND	DL (0.42)	⇒	US (0.42)

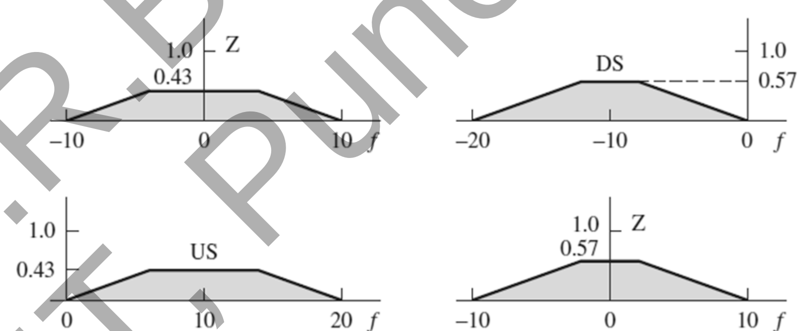


- We find the centroid to be f_1 = −0.5 lb

CYCLE #2

- $h_2 = h_1 + v_1$=980+(-14.2)=965.8 ft
- $v_2 = v_1 + f_1$= (-14.2)+(-0.5)=-14.7 ft/s
- Height h_2 fires **L** at 0.93 and **M** at 0.67
- Velocity v_2 fires **DS** at 0.57 and **DL** at 0.43

Height		Velocity		Output
L (0.93)	AND	DS (0.57)	⇒	DS (0.57)
L (0.93)	AND	DL (0.43)	⇒	Z (0.43)
M (0.67)	AND	DS (0.57)	⇒	Z (0.43)
M (0.67)	AND	DL (0.43)	⇒	US (0.57)

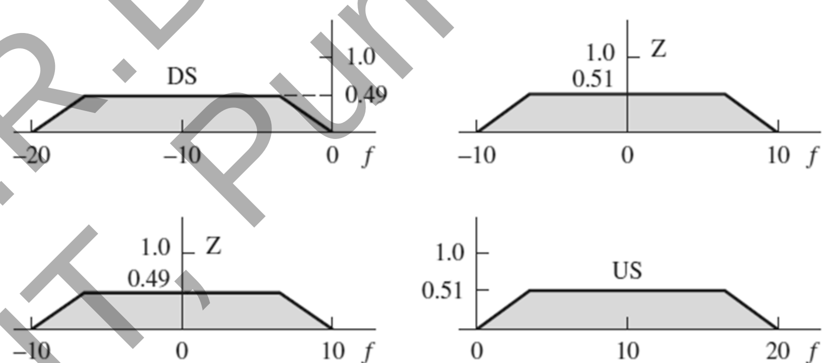


- We find the centroid to be f_2 = −0.4 lb

CYCLE #3

- h_3 =h_2+v_2=965.8+(-14.7)=951.1 ft
- v_3 =v_2+f_2= (-14.7)+(-0.4)=-15.1 ft/s
- Height $h_3$fires **L** at 0.9 and **M** at 0.7
- Velocity $v_3$fires **DS** at 0.49 and **DL** at 0.51

Height		Velocity		Output
L (0.9)	AND	**DS (0.49)**	⇒	**DS (0.49)**
L (0.9)	AND	**DL (0.51)**	⇒	**Z (0.51)**
M (0.7)	AND	**DS (0.49)**	⇒	**Z (0.49)**
M (0.7)	AND	**DL (0.51)**	⇒	**US (0.51)**



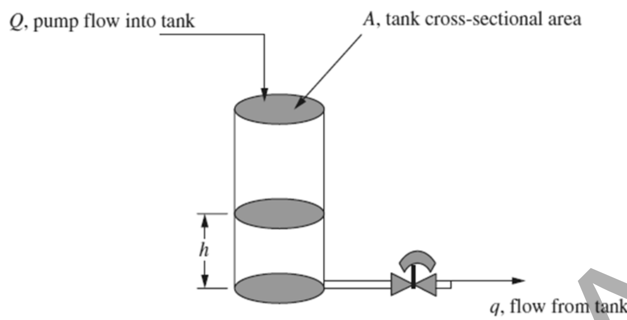
- We find the centroid to be f_3 = 0.3 lb

- Final values for the state variables to finish the simulation
- h_4 =h_3+v_3=951.1+(-15.1)=936.0 ft
- v_4 =v_3+f_3= (-15.1)+(0.3)=-14.8 ft/s

	Cycle 0	Cycle 1	Cycle 2	Cycle 3	Cycle 4
Height (ft)	1000.0	980.0	965.8	951.1	936.0
Velocity (ft s^{-1})	−20	−14.2	−14.7	−15.1	−14.8
Control force	5.8	−0.5	−0.4	0.3	

Example of FLC

- Liquid Level Control
- Design a controller that can be used to move the level set-point from, say, 4 feet to 6 feet, the set-point-tracking problem
- Suppose that the tank is 10 feet tall and the tank is empty.
- We want to fill the tank to a level of 5 feet, so we make the current set-point, *w*, equal to 5
- The idea is to fill the tank to the desired set-point as quickly and smoothly as possible
- We want to minimize the amount of overshoot, or the time that the tank has a level greater than the set-point value before it finally settles down



- The current level at any time, ***t*** , is designated as ***h***
- Liquid flows out of the tank through an open valve
- This flow is designated by the letter ***q***
- Liquid flows into the tank by means of a pump
- The pump flow, ***Q***, can be regulated by the controller
- The tank cross-sectional area is designated by the letter ***A***

- Equation that describes the mass balance for the liquid in the tank as a function of time is :

$$A\frac{dh}{dt} = Q - q$$

- Flow out of the tank, q, through the outlet pipe and the valve is described as:

$$q = \Phi A_p\sqrt{2gh}$$

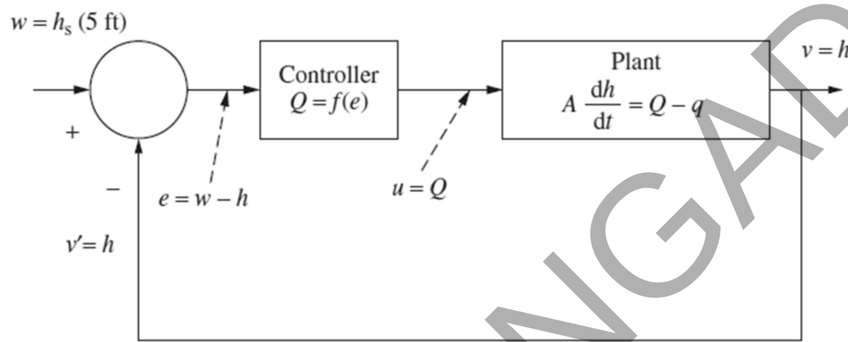
where

Φ= friction coefficient for flow through both the small exit pipe and the valve,

A_p=cross-sectional area of the small exit pipe

g =the gravitational constant, equal to 32.2 ft s−2 (US units)

Block flow diagram



Classical PID control

- The PID control algorithm is described as:

$$u = K_P e + K_I \int_0^T e\,dt + K_D \frac{de}{dt}$$

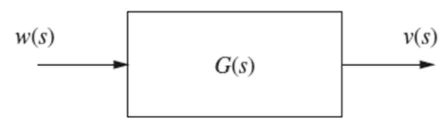
- K_P, K_I, K_D are proportional, integral, and derivative control constants, respectively ,are specific to the system in question and are usually picked to optimize the controller performance and ensure that the *system remains stable for all possible control actions*
- If we use a PID controller, which is linear, or any other linear controller with a linear plant, then the system is called a *linear system*
- We can use Laplace transforms to convert the linear equations in the
- blocks to the Laplace domain
- The blocks can then be combined to form a single transfer function for the entire system
- In the real world, many systems are at least slightly nonlinear and we can linearize them by expanding the non linear function in truncated Taylor series:

$$\sum_{n=0}^{\infty} \frac{f^n(a)}{n!}(x-a)^n = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^k(a)}{k!}(x-a)^k + \cdots$$

- The truncated Taylor's series for linearizing about a steady state value, in this case our set-point, is given as:

$$\sqrt{h} = \sqrt{h_s} + \frac{1}{2\sqrt{h_s}}(h - h_s)....(1)$$

- If we choose h_s = 5 ft, we can linearize the radical term over some of the control range

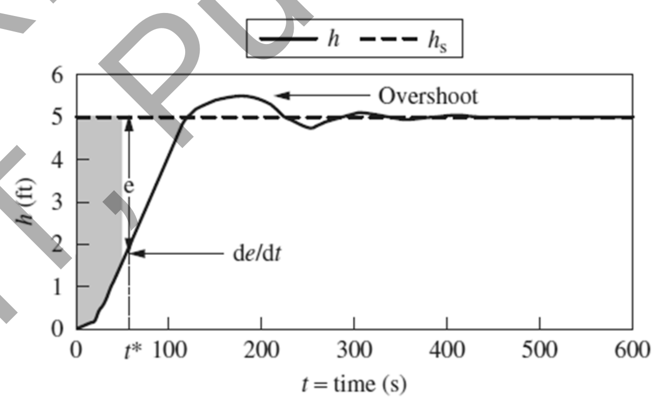


Overall system block flow diagram or transfer function for the Laplace domain

Approximate linearization.

h (ft)	$\sqrt{h}$	Equation (1)
10	3.162	3.354
9	3.0	3.130
8	2.828	2.907
7	2.646	2.683
6	2.449	2.460
5	2.236	2.236
4	2.0	2.012
3	1.732	1.789
2	1.414	1.565
1	1.0	1.342
0	0	1.118

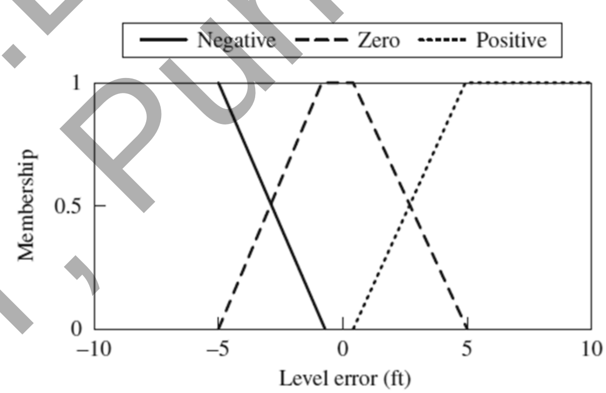
Time-level PID response for the tank-filling problem



Fuzzy Control for Tank Filling Problem

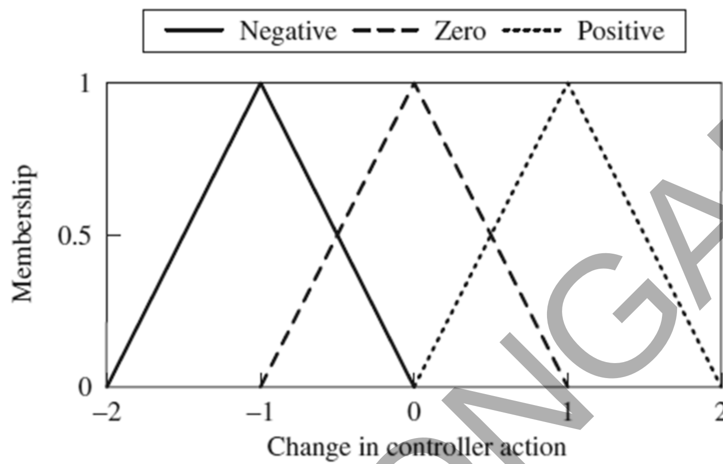
- A fuzzy control system connects input membership functions, functions representing the input to the controller, ***e***, to output membership functions that represent the control action, ***u***
- A simple fuzzy control system designed for our tank-level set-point-tracking problem consists of three rules:
  1. If the ***Level Error*** is ***Positive*** Then the ***Change in Control Action*** is ***Positive***
  2. If the ***Level Error*** is ***Zero*** Then the ***Change in Control Action*** is ***Zero***
  3. If the ***Level Error*** is ***Negative*** Then the ***Change in Control Action*** is ***Negative***

The input membership functions



- Notice the “dead band” or “dead zone” in the membership function ***Zero*** between ±3 Inches
- This is optional and is a feature commonly used with on–off controllers
- It is easy to implement with a fuzzy controller and is useful if the control engineer wishes to minimize control response to small transient-level changes
- This step can save wear and tear on equipment

The output membership functions

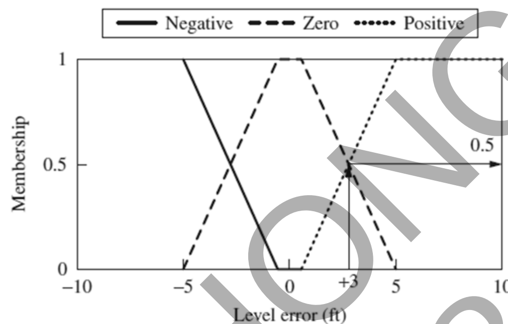


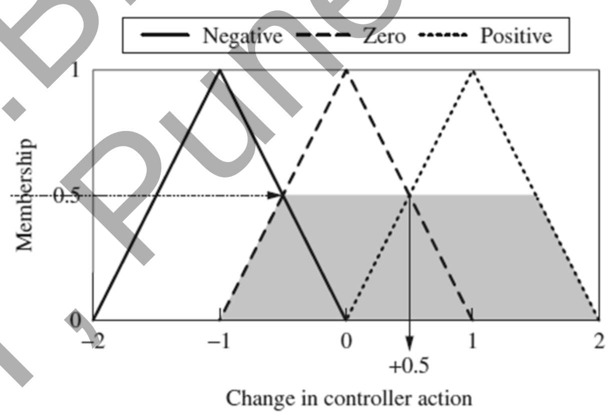
- The de-fuzzified output value from the controller is a fractional representing the required pump output for the desired level change
- It is defined by the following expression:

  Change in Controller Action = Δ*u* = ($Q_i - Q_{sp}$)/*Range*

Where,
  - The term Q_{sp} represents the pump output (gallons per minute) required to maintain the set-point level
  - The term Q_i is the new pump output requested by the controller
  - If Δ*u*>0, then the *Range* is defined as $Q_{max} - Q_{sp}$, where Q_{max} is the maximum pump output
  - If Δ*u* < 0, then the *Range* is defined as Q_{sp}
  - The term Q_{sp} must be calculated using a steady state mass balance for the tank or it must be estimated in some fashion
- The ranges of the fuzzy output sets ***Positive*** and ***Negative*** are +2.0 to 0.0 and −2.0 to 0.0, respectively
- Since the ***Change in Controller Action*** is a fraction between either
- 0.0 and 1.0 or 0.0 and −1.0, it is clear that we will never obtain a control action outside the range of −1.0 to 1.0
- Our de-fuzzification technique will require that we include numbers up to 2.0 in the fuzzy set or membership function ***Positive*** and numbers down to −2.0 in the fuzzy set ***Negative***

- Suppose that we decide to change our set-point level from 5 feet to 8 feet
- The error is defined as the set-point level, 8 feet, minus the current level, 5 feet, or +3 feet
- The three rules are fired, producing the following results:
  1. ***Positive*** error is 0.5
  2. ***Zero*** error is 0.5
  3. ***Negative*** error is 0.0





- The centroid of the “clipped” membership functions, the shaded area is +0.5

- Since u is greater than 0.0,

$$Q_i = (Q_{max} - Q_{sp})\Delta u + Q_{sp}$$

or

$$Q_i = 0.5(Q_{max} + Q_{sp}), \text{ since } \Delta u = 0.5$$

- This says that the new pump output, Q_i, should be adjusted to be halfway between the current or set-point output and the maximum pump output
- After an appropriate time interval, corresponding to a predetermined sample rate, the same procedure will be repeated until the set-point level, 8 feet, is achieved!

SUMMARY

- Easy to set up a working fuzzy controller even for difficult-to-control processes (but previously controlled by operator).
- Easy to understand the rule base in simple cases.
- No need to use FLC in straightforward linear cases, if implementation reasons would not require it.
- Complications will arise when rule base increases.