# Lecture 33
# Fuzzy Logic Control (I)

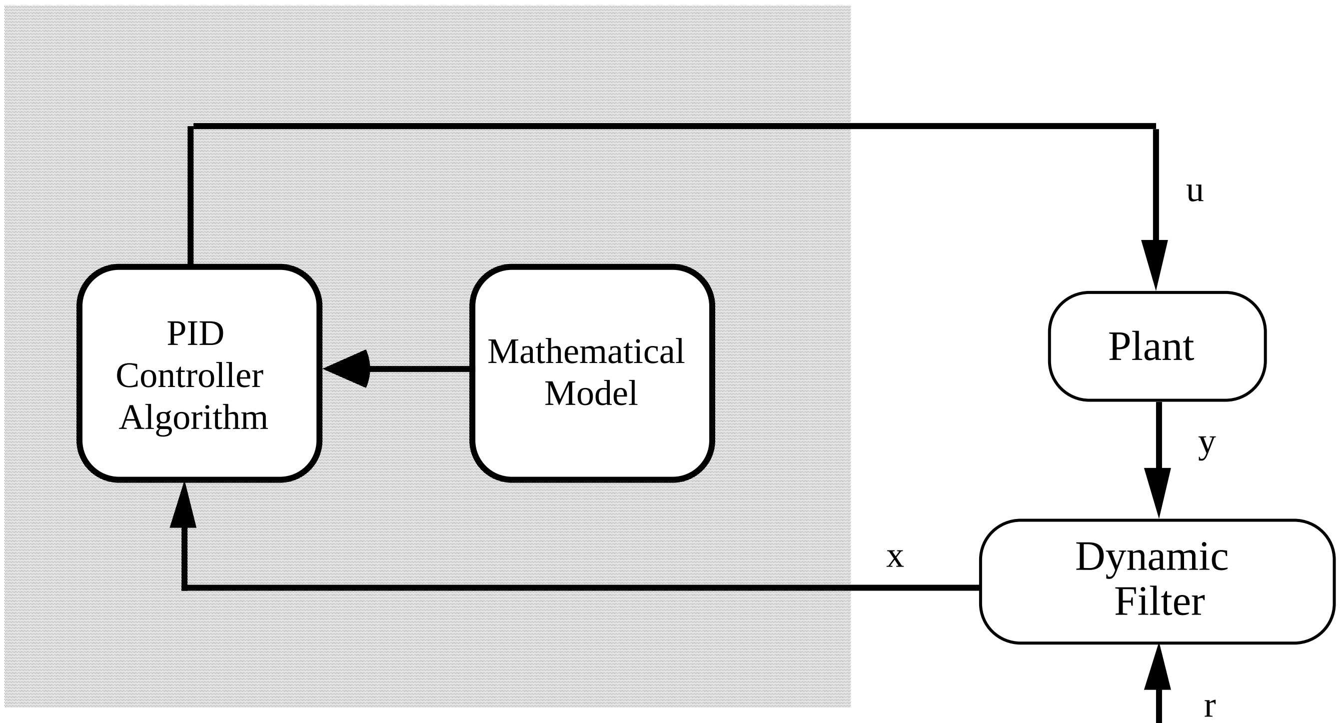
# Outline

- Overview
- Review of PID Control
- FLC Architecture
- Fuzzification

# Overview

- Fuzzy Logic Control (FLC) or sometimes known as Fuzzy Linguistic Control is a knowledge based control strategy that can be used
  - when either a *sufficient accurate* and yet not *unreasonably complex* model of the plant is unavailable, or
  - when a (single) precise measure of performance is not meaningful or practical.
- FLC design is based on *empirically* acquired knowledge regarding the operation of the process.
- This knowledge, cast into *linguistic*, or *rule-based* form, is the core of the FLC system.

# Conventional Controller




Conventional PID Controller Architecture

The dynamic filter compute all the system dynamics: x (state variables) consists of selected elements of e = r – y, de/dt, and ∫e dτ.

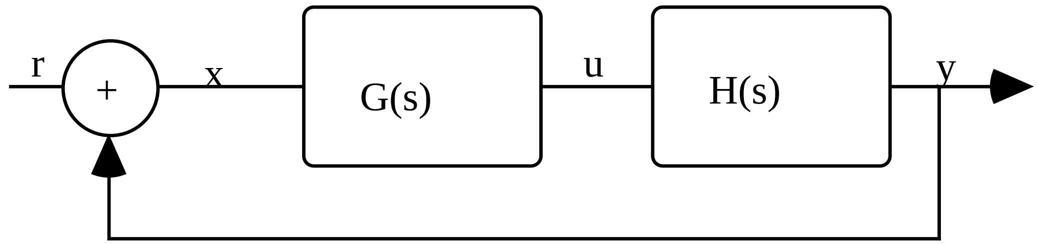
# PD & PI Control: A Review

- PD controller:
  - Use derivative term to reduce overshoot.
  - $K_p$ must be selected to maintain desired steady state error.
  - $K_d$ will be selected to minimize max. overshoot
- PI controller:
  - Use integral term to reduce steady state error by increasing the order of the system by 1.
  - The steady state error can often be reduced to 0.
- $K_p$ and $K_I$ are chosen to meet transient response specifications. Often they lead to longer rise and settling times.

# PID Control: A Review

- Controller Transfer Function:
  $G(s) = U(s) / x(s) = K_p + K_1/s + K_d s$
  $= (1+K_{d1}s)(K_{p2} + K_{I2} / s)$
  where $K_p = K_{p2} + K_{d1}K_{I2}$, $K_d = K_{d1}K_{p2}$, and $K_I = K_{I2}$.
- Select $K_{p2}$ and $K_{I2}$ in the PI portion to satisfy the rise time requirement of the system.
- Select $K_{d1}$ to meet damping requirements and thereby reduce the overshoot.
- Convert the values to $K_p$, $K_d$, and $K_I$.

# PID Control Example




Given the plant model: H(s) = 815265 / s(s+361.2)

Open loop transfer function with PI section only:

$G(s)H(s) = 815265\ (K_{p2} + K_{I2} / s\ ) / s(s+361.2)$

$= 815265 \bullet K_{p2}(\ s + K_{I2} / K_{p2}) / s^2(s+361.2)$

The system is stable for $0 < K_{I2}/K_{p2} < 361.2$ (Routh's test)

A heuristic is to choose $K_{I2}/K_{p2} = 10 << 361.2$.

# PID Example Cont'd

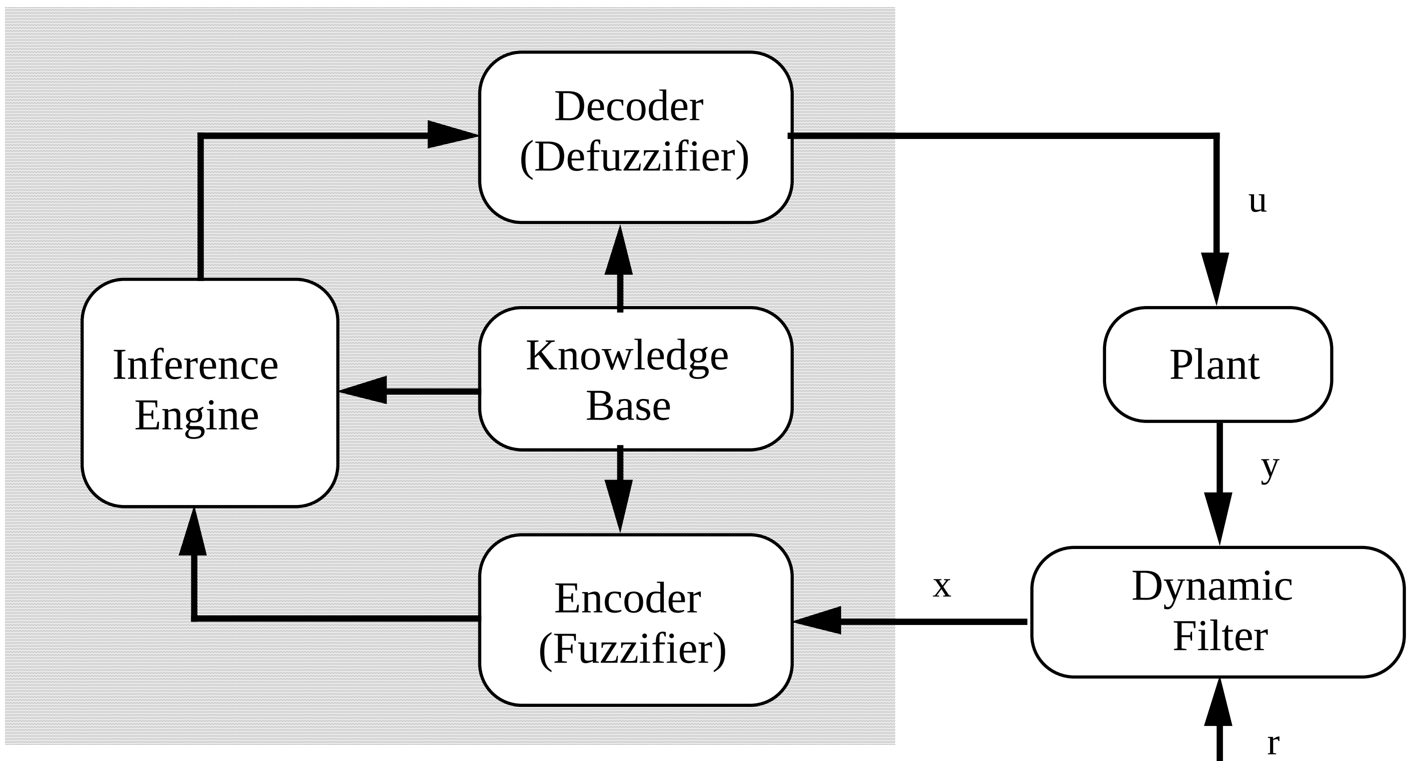
Then $G(s)H(s) \approx 815265 \bullet K_{p2} / s(s+361.2)$. The characteristic equation $1+G(s)H(s) = 0$ has a damping factor $\xi = 0.2$ if $K_{p2} = 1$.

Including the PD section, and choose $K_{I2} = 10$, $K_{p2} = 1$,

$$G(s)H(s) = 815265 \bullet (s+10)(1+K_{d1}s) / s^2(s+361.2)$$

Finally, $K_{d1} = 0.003544$ is selected to minimize overshoot through trial-and-error experimentation. Other values such as .001772 or .001 will also yield acceptable solution. A PD controller with $K_p = 1$ and $K_d = .001772$ works fine too!

# FLC Architecture



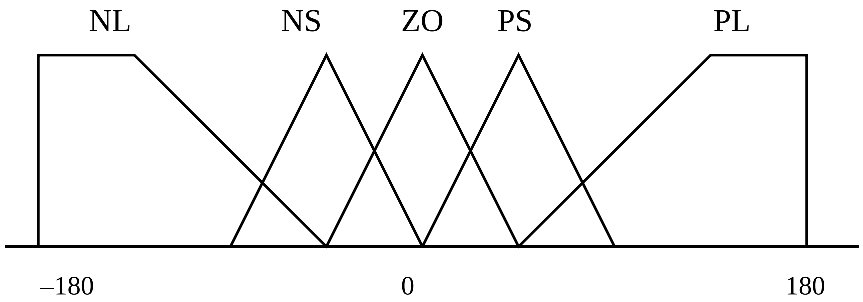

Fuzzy Logic Controller Architecture

The rule base (knowledge base) provides nonlinear transformations without any built-in dynamics.

# Fuzzification

Fuzzy quantization of the state variables. For example, the state variable "Angle" may be quantified into a set of linguistic variables, with *two* parameters, polarity and size:

NL – Negative, Large;
NS – Negative, Small;
ZO – Zero;
PS – Positive, Small;
PL – Positive, Large.

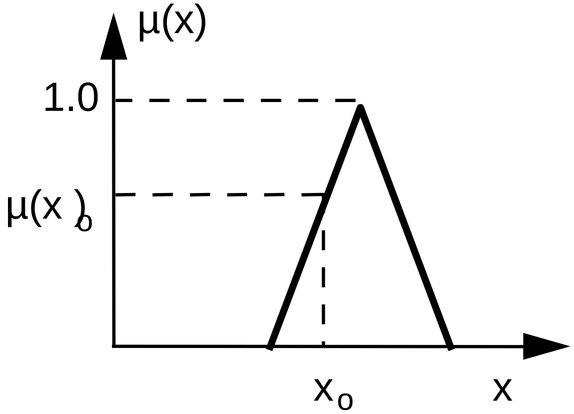



Fuzzification is the process to convert a crisp sensor reading (value of state variable) x = xo into the grade values of each of these linguistic variables. In particular, we have

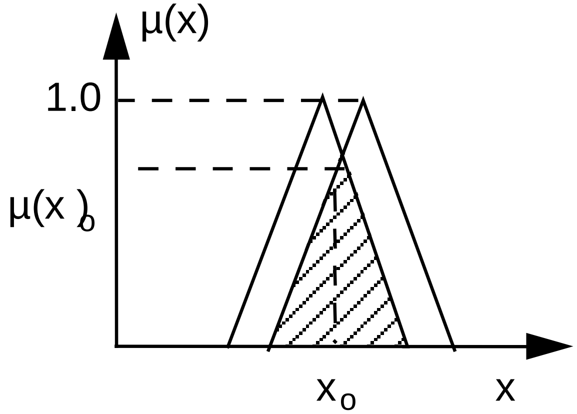
$$[\ \mu_{NL}(x_o),\ \mu_{NS}(x_o),\ \mu_{ZO}(x_o),\ \mu_{PS}(x_o),\ \mu_{PL}(x_o)\ ]$$

# Fuzzification

Sensor reading may also contain noise. In this case, we may model the noisy sensor reading as another fuzzy variable. In this case, the fuzzification amounts to take the intersection (minimum) of the sensor reading and each of the linguistic variables.




Crisp sensor reading




Fuzzy sensor reading