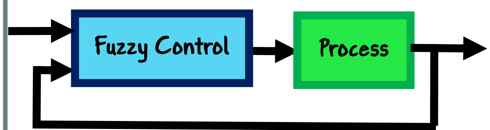
Intelligent Control



Fuzzy Control Systems

By:
Barmak Baigzadehnoe
b.baigzadeh@uok.ac.ir

Smart/Micro Grids Research Center, University of Kurdistan

Content

smgrc

Definition of Fuzzy Control

- A ***fuzzy controller*** is a controller that includes a nonlinear mapping based on fuzzy if-then rules. In fact, a controller in which fuzzy logic is used in anyway, is known as a ***fuzzy controller***.

- If fuzzy systems are directly used as controllers or controllers are designed based on the fuzzy model, then the resulting controllers are called ***fuzzy controllers***.

- ***Fuzzy control*** is a methodology of intelligent control that mimics human thinking and reacting by using a multivalent fuzzy logic and elements of artificial intelligence.

smgrc

Why Fuzzy Control?

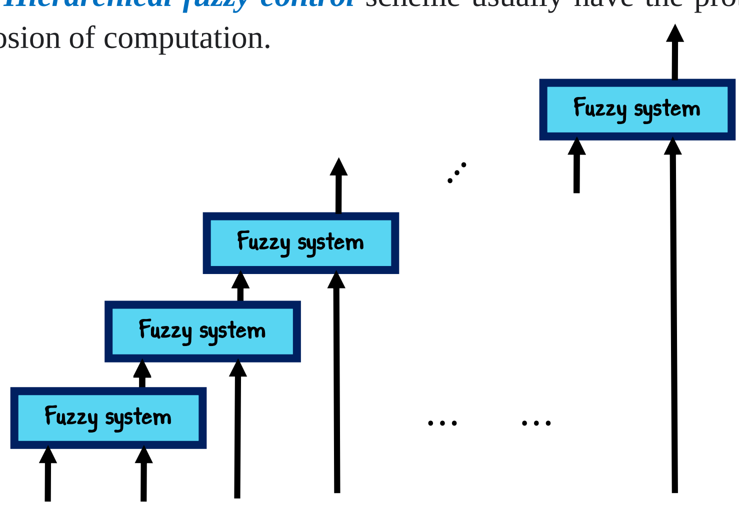
- Efficiently incorporates human expert information
- Does not require a mathematical model of system
- Produces nonlinear controllers
- Easy and inexpensive to design
- Easy to understand

Type of Fuzzy Control

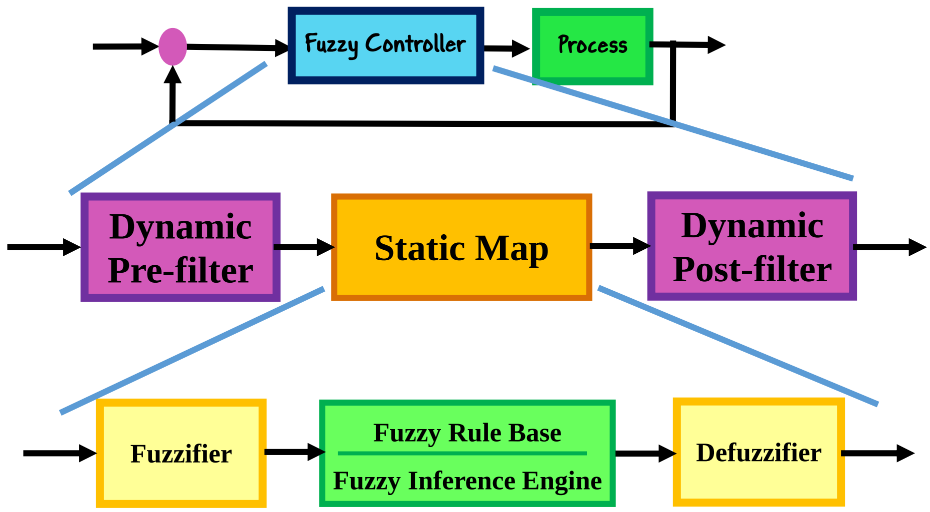
- ***Types of fuzzy systems:*** Mamdani/ Takagi-Sugeno-Kang
- ***Methods of applying:*** direct/ indirect
- ***Learning ability:*** supervised learning/ unsupervised learning
  auxiliary learning/ without learning ability
- ***Adaptation property:*** adaptive/non-adaptive
- ***Structure:*** hierarchical/ non-hierarchical

Type of Fuzzy Control

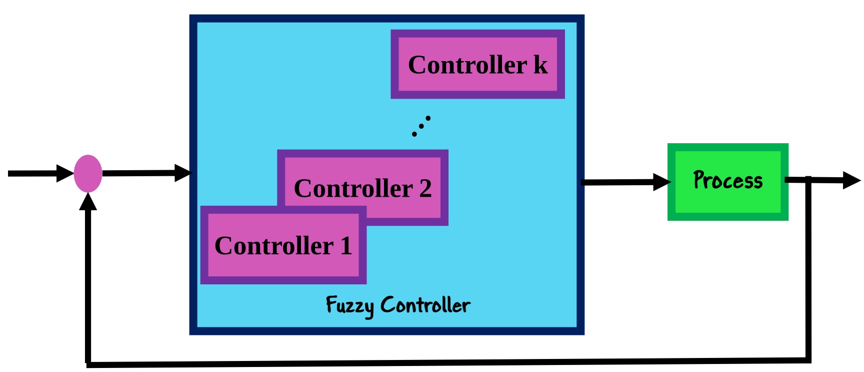
Note: ***Hierarchical fuzzy control*** scheme usually have the problem of explosion of computation.



Mamdani Fuzzy Control



TSK Fuzzy Control



Stability of Fuzzy Control

Note: Although the rigorous mathematical framework of control systems ***stability*** theory in some way opposes the vagueness of fuzzy controller properties, ***stability*** remains a key issue in fuzzy controller design. The main criticism of fuzzy control is related to its lack of precise ***stability*** analysis. That is why efforts have been put into the investigation of various techniques that have a potential to solve the ***stability*** issue in fuzzy controlled systems.

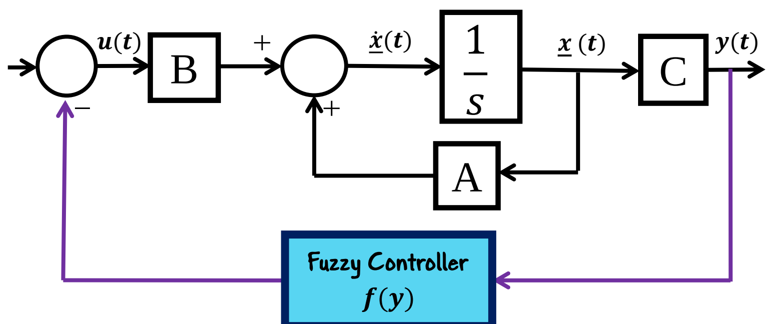
Content

Fuzzy Control system

Consider the following ***LTI-SISO system*** with the fuzzy control scheme:

$$system: \begin{cases} \dot{\underline{x}}(t) = A\underline{x}(t) + Bu(t) \\ y(t) = C\underline{x}(t) \end{cases}$$

$$Controller: \{u(t) = -f(y)$$



Fuzzy Control system

Proposition: Consider the following closed-loop control system

$$system: \begin{cases} \dot{\underline{x}}(t) = A\underline{x}(t) - Bf(y) \\ y(t) = C\underline{x}(t) \end{cases}$$

and suppose that

- all eigenvalues of A lie in the open left half of the complex plane
- the system is controllable and observable
- the transfer function of the system is strictly positive real.

If the nonlinear function f satisfies $f(0) = 0$ and

$$yf(y) \geq 0 \qquad for \ \forall y \in R$$

then the equilibrium point x = 0 of the closed-loop system is globally exponentially stable.

smgrc

Fuzzy Control system

Theorem: Consider the fuzzy system from the $2N+1$ fuzzy IF-THEN rules of the following form:

$$IF\ y\ is\ A^l\ \ THEN\ u\ is\ B^l$$

In which $l = 1,2,\ldots,2N+1$ and the centers $\bar{y}^l$ of fuzzy sets B^l are chosen such that

$$\bar{y}^l \begin{cases} \leq 0 & for \quad l = 1,2,\ldots N \\ = 0 & for \quad l = N+1 \\ \geq 0 & for \quad l = N=2, N+3, \ldots 2N+1 \end{cases}$$

If product inference engine, singleton fuzzifier, and center average defuzzifier are employed to construct fuzzy system; that is, the designed fuzzy system is

$$u = -f(y) = -\left(\sum_{l=1}^{2N+1} \bar{y}^l \mu_{A^l}(y)\right) \Big/ \left(\sum_{l=1}^{2N+1} \mu_{A^l}(y)\right)$$

, then the nonlinear function f satisfies $f(0) = 0$ and $yf(y) \geq 0$. $(for\ \ \forall y \in R)$.

smgrc

Optimal Fuzzy Control Scheme

Consider the following LTI SISO system:

$$\dot{\underline{x}}\,(t) = A\underline{x}\,(t) + B\underline{u}(t)$$

where $x \in R^n$ and $u \in R^m$, and that the performance criterion is the quadratic function:

$$J = \underline{x}^T\,(T)M\underline{x}(T) + \int_0^T \left[\underline{x}^T\,(t)Q\underline{x}(t) + \underline{u}^T\,(t)R\underline{u}(t)\right]dt$$

where the matrices $\mathrm{M} \in R^{n\times n}$, $\mathrm{Q} \in R^{n\times n}$and $\mathrm{R} \in R^{m\times m}$ symmetric and positive definite.

smgc

Optimal Fuzzy Control Scheme

Consider the following LTI SISO system:

$$\dot{\underline{x}}(t) = A\underline{x}(t) + B\underline{u}(t)$$

where $x \in R^n$ and $u \in R^m$, and that the performance criterion is the quadratic function:

$$J = \underline{x}^T(T)M\underline{x}(T) + \int_0^T \left[\underline{x}^T(t)Q\underline{x}(t) + \underline{u}^T(t)R\underline{u}(t)\right]dt$$

where the matrices $\mathrm{M} \in R^{n\times n}$, $\mathrm{Q} \in R^{n\times n}$ and $\mathrm{R} \in R^{m\times m}$ symmetric and positive definite.

smgc

Optimal Fuzzy Control Scheme

Now, assume that the controller $\underline{u}(t) = [u_1, u_2, .., u_m]^T$ is a fuzzy system in the following form:

$$u_j = -f_j(\underline{x}) = \frac{\sum_{l_1=1}^{2N_1+1} \cdots \sum_{l_n=1}^{2N_n+1} \bar{y}^{l_1\cdots l_n}\left[\prod_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)\right]}{\sum_{l_1=1}^{2N_1+1} \cdots \sum_{l_n=1}^{2N_n+1}\left[\prod_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)\right]}$$

Define the fuzzy basis functions $b(t) = [b_1, b_2, .., b_N]^T$

$$b_l = \left(\prod_{i=1}^{n} \mu_{A_i^{l_i}}(x)\right) \Big/ \left(\sum_{l_1=1}^{2N_1+1} \dots \sum_{l_n=1}^{2N_n+1} \left(\prod_{i=1}^{n} \mu_{A_i^{l_i}}(x)\right)\right)$$

where $l = 1,2, \dots, N$ and $N = \prod_{i=1}^{n} (2N_i + 1)$.

Optimal Fuzzy Control Scheme

Also, define the parameter matrix $\Theta \in R^{m\times N}$ as:

$$\Theta = \begin{bmatrix} \Theta_1^T \\ \Theta_2^T \\ \vdots \\ \Theta_m^T \end{bmatrix}$$

Where $\Theta_j^T \in R^{1\times N}$ consists of the N parameters $\bar{y}^{l_1 \cdots l_n}$. Using these notations, we can rewrite the fuzzy controller as:

$$\underline{u}(t) = \Theta b(x)$$

By using ***Pontryagin Minimum Principle***, we can determine the optimal value of Θ.

Optimal Fuzzy Control Scheme

Also, define the parameter matrix $\Theta \in R^{m\times N}$ as:

$$\Theta = \begin{bmatrix} \Theta_1^T \\ \Theta_2^T \\ \vdots \\ \Theta_m^T \end{bmatrix}$$

where $\Theta_j^T \in R^{1\times N}$ consists of the N parameters $\bar{y}^{l_1 \cdots l_n}$. Using these notations, we can rewrite the fuzzy controller as:

$$\underline{u}(t) = \Theta b(x)$$

By using ***Pontryagin Minimum Principle***, we can determine the optimal value of Θ.

Optimal Fuzzy Control Scheme

Let $\underline{x}^*$and P^* $(t \in [0,T])$ be the (offline) solution of :

$$\begin{cases} \dot{\underline{x}} = A\underline{x} + 2\left(\alpha^2(\underline{x}) - \alpha(\underline{x})\right)BR^{-1}B^T P \\ \dot{P} = -2Q\underline{x} - A^T P - \left(2\alpha(\underline{x}) - 1\right)\dfrac{\partial\alpha(\underline{x})}{\partial \underline{x}} P^T BR^{-1}B^T P \end{cases}$$

where $\alpha(\underline{x}) = 0.5b^T(\underline{x})\left(b(\underline{x})b^T(\underline{x})\right)^{-1} b(\underline{x})$ and the initial conditions are $\underline{x}(0) = \underline{x}_0$ and $P(t) = 2M\underline{x}(T)$. Then, the optimal fuzzy controller parameters are:

$$\boldsymbol{\Theta}^* = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}^*(\boldsymbol{t})\boldsymbol{b}^T\left(\underline{\boldsymbol{x}}^*(\boldsymbol{t})\right)\left(\boldsymbol{b}\left(\underline{\boldsymbol{x}}^*(\boldsymbol{t})\right)\boldsymbol{b}^T\left(\underline{\boldsymbol{x}}^*(\boldsymbol{t})\right)\right)^{-1}$$

Content

- Introduction
- Fuzzy Control of Linear systems
- Fuzzy Supervisory Control
- Design Example

smgrc

Introduction

Most controllers in operation today have been developed using conventional control methods. There are, however, many situations where these controllers are not properly tuned and there is heuristic knowledge available on how to tune them while they are in operation. There is then the opportunity to utilize ***fuzzy control*** methods as the ***supervisor*** that tunes or coordinates the application of conventional controllers.

The ***supervisor*** can use any available data from the control system to characterize the system's current behavior so that it knows how to change the controller and ultimately achieve the desired specifications.

smgrc

Introduction

The type of ***heuristic knowledge*** that is used in a ***supervisor*** may take one of the following two forms:

- Information from a ***human control system operator*** who observes the behavior of an existing control system (often a conventional control system) and knows how this controller should be tuned under various operating conditions.

- Information gathered by a ***control engineer*** who knows that under different operating conditions controller parameters should be tuned according to certain rules.

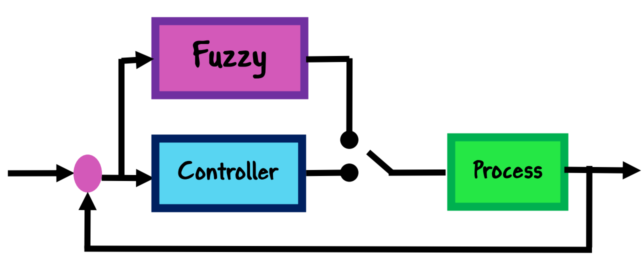
Types of Fuzzy Supervisory Control

There are four types of fuzzy supervisory control:

- ***Fuzzy replaces controller***
- ***Fuzzy replaces operator***
- ***Fuzzy adjusts control parameters***
- ***Fuzzy adds to controller***

Fuzzy Replaced Controller

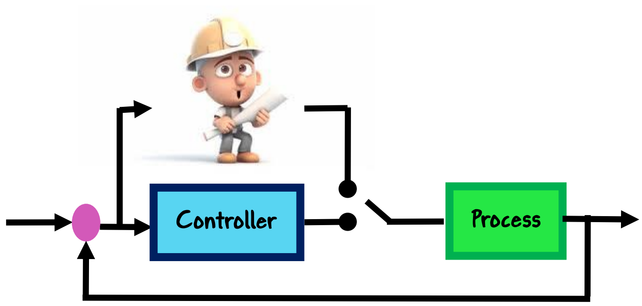
In this configuration, the operator may select between a high-level control strategy and conventional control loops.



smgc

Fuzzy Replaced Operator

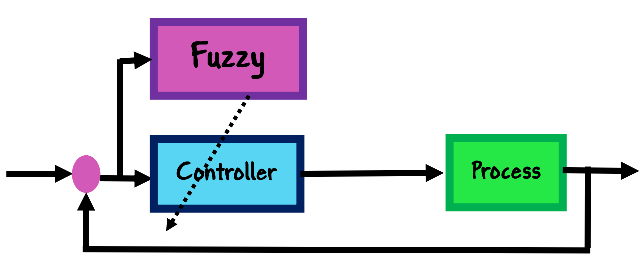
This configuration represents the original high level control idea, where manual control carried out by a human operator is replaced by automatic control



smgc

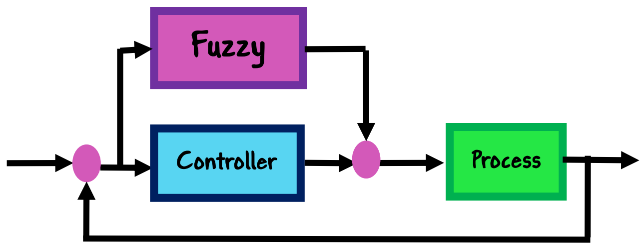
Fuzzy Adjusts Control Parameters

In this configuration, the high-level strategy adjusts the parameters of the conventional control loops.



Fuzzy adds to Controller

Normally, control systems based on controllers are capable of controlling the process when the operation is steady and close to normal conditions. However, if sudden changes occur, or if the process enters abnormal states, then this configuration may be applied to bring the process back to normal operation as fast as possible.



Fuzzy Systems

In order to design fuzzy supervisor, Mamdani fuzzy control with singleton fuzzifier, product inference engine and center average defuzzifier is usually employed. That is, the following formula:

$$Parameter:\ p = f(\underline{x}) = \frac{\sum_{l=1}^{M} \bar{p}^{l} \prod_{i=1}^{n} \mu_{A_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{A_i^l}(x_i)}$$

This parameter is usually the gain of controllers!

smgc

Algorithm

$$Parameter: \; p = f(\underline{x}) = \frac{\sum_{l=1}^{M} \bar{p}^{l} \prod_{i=1}^{n} \mu_{A_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{A_i^l}(x_i)}$$

Choose Inputs $x_1, x_2, \dots, x_n$ and Output p

↓

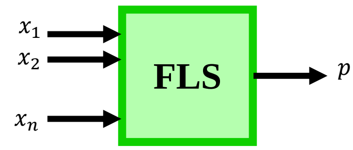
Obtain Inputs Bounds $[x_{i,min}, x_{i,max}]$

↓

Select Membership Functions $\mu_{A_i^l}(x_i)$
(types and numbers)

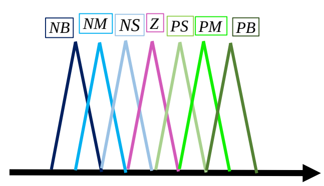
↓

Construct Rules $\bar{p}^{\,l}$



smgc

Fuzzy Supervisory



$$p' = \frac{p - p_{\min}}{p_{\max} - p_{\min}}$$

p or p'		$\dot{e}$						
		NB	NM	NS	Z	PS	PM	PB
e	NB							
	NM							
	NS							
	Z							
	PS							
	PM							
	PB							

Fuzzy Supervisory

$$Parameter:\ p = f(\underline{x}) = \frac{\sum_{l=1}^{M} \bar{p}^{l} \prod_{i=1}^{n} \mu_{A_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{A_i^l}(x_i)}$$

Calculate Inputs x_i

↓

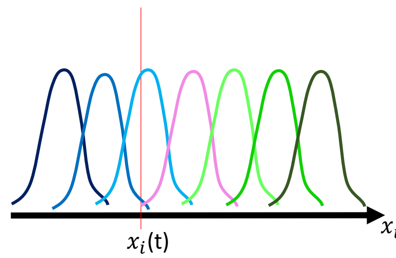
Obtain MF Values $\mu_{A_i^1}(x_i), \mu_{A_i^2}(x_i), \dots$

↓

Obtain $\bar{y}^{\,l}$ Based on Rules

↓

Calculate Output



Fuzzy Supervisory

$$Parameter:\ p = f(\underline{x}) = \frac{\sum_{l=1}^{M} \bar{p}^{l} \prod_{i=1}^{n} \mu_{A_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{A_i^l}(x_i)}$$

Number of Inputs: n
Number of Rules: M=$N_1 \times N_2 \times \cdots N_n$

$$p = \frac{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2}\cdots\sum_{i_n=1}^{N_n} \bar{p}^{i_1,i_2,\dots,i_n} \mu_{A_1^{i_1}}(x_1(t))\mu_{x_2^{i_2}}(x_2(t))\cdots\mu_{x_n^{i_2}}(x_n(t))}{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2}\cdots\sum_{i_n}^{N_n} \mu_{A_1^{i_1}}(x_1(t))\mu_{x_2^{i_2}}(x_2(t))\cdots\mu_{x_n^{i_2}}(x_n(t))}$$

Number of Inputs: 2
Number of Rules: M=$N_1 \times N_2$

$$p = \frac{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2} \bar{p}^{i_1,i_2} \mu_{A_1^{i_1}}(e(t))\mu_{A_2^{i_2}}(\dot{e}(t))}{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2} \mu_{A_1^{i_1}}(e(t))\mu_{A_2^{i_2}}(\dot{e}(t))}$$

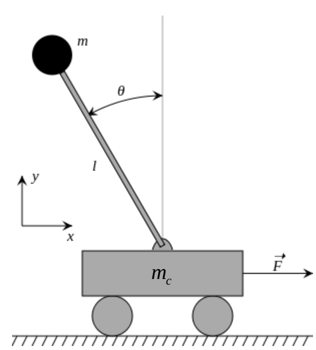
Content

- Introduction
- Fuzzy Control of Linear systems
- Fuzzy Supervisory Control
- Design Example

Inverted Pendulum System

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{g\sin(x_1) - \dfrac{mlx_2^2\cos(x_1)\sin(x_1)}{m_c+m}}{l\left(\dfrac{4}{3} - \dfrac{m(\cos(x_1))^2}{m_c+m}\right)} + \dfrac{\dfrac{\cos(x_1)}{m_c+m}}{l\left(\dfrac{4}{3} - \dfrac{m(\cos(x_1))^2}{m_c+m}\right)}u \end{cases}$$

Objective:
Tracking step type reference



smgrc

Inverted Pendulum System

Open-Loop System Simulation

↓

Choose Control Structure and Design

↓

Parameters Tuning (FLS Construction)

↓

Closed-Loop System Simulation

smgrc

Inverted Pendulum System

Open-Loop System Simulation

Set Sample Time

↓

Define Variables and Parameters and Initialization

↓

Create a Loop and Calculate Variables at Each Time

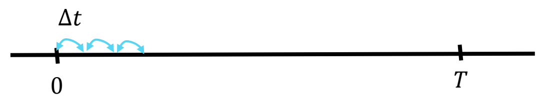
↓

Plot Results

smgrc

Inverted Pendulum System

```
T=5;
dt=0.001;
sample=T/dt+1;   Time
```




```
g=9.8;
mc=1;
m=0.1;
l=0.5      Define Parameters
```

```
x1=zeros(1,sample);
x2=zeros(1,sample);
%*********
u=zeros(1,sample);
%*********
x1(1,1)=0.3;
x2(1,1)=0.3;                Define Variables
```


smgrc

Inverted Pendulum System

```
for t=2:sample                                                   Loop
x1(1,t)=(x2(1,t-1))*dt+x1(1,t-1);

x2(1,t)=(1/(l*(4/3-m*(cos(x1(1,t-1))^2)/(mc+m))))
*(((cos(x1(1,t-1)))/(mc+m))*u(1,t)+g*sin(x1(1,t-1))-
(m*l*((x2(1,t-1))^2)*cos(x1(1,t-1))*sin(x1(1,t-1)))/(mc+m))
*dt+x2(1,t-1);
end
```

$$\dot{x}(t)\left(=\lim_{\Delta t\to 0}\frac{x(t)-x(t-\Delta t)}{\Delta t}\right)=f(x)$$

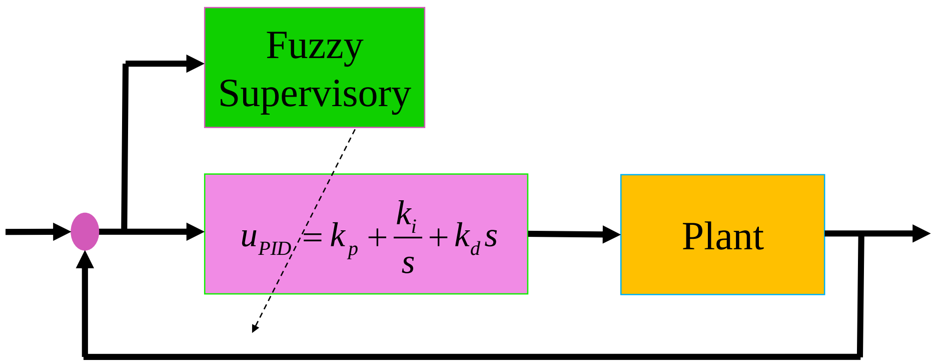
$$\Rightarrow \boxed{x(t)=\left(f(x(t-\Delta t))\right)\Delta t+x(t-\Delta t)}$$

$$\begin{cases}\dot{x}_1=x_2\\ \dot{x}_2=\dfrac{g\sin(x_1)-\dfrac{mlx_2^2\cos(x_1)\sin(x_1)}{m_c+m}}{l\left(\dfrac{4}{3}-\dfrac{m(\cos(x_1))^2}{m_c+m}\right)}+\dfrac{\dfrac{\cos(x_1)}{m_c+m}}{l\left(\dfrac{4}{3}-\dfrac{m(\cos(x_1))^2}{m_c+m}\right)}u\end{cases}$$

smgc

Control Structure

Choose Control Structure and Design



$$u_{PID} = 200 + \frac{10}{3s} + 2s$$

smgc

Fuzzy Supervisory

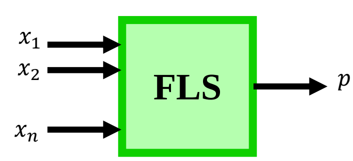
FLS Construction (Parameters Tuning)

Choose Inputs $x_1, x_2, \ldots, x_n$ and Output p

Obtain Inputs Bounds $[x_{i,min}, x_{i,max}]$

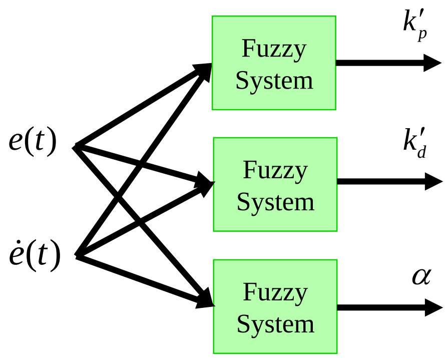
Select Membership Functions $\mu_{A_i^l}(x_i)$
(types and numbers)

Construct Rules $\overline{p}^{\,l}$



Tuning Parameters

Choose Inputs
$x_1, x_2, \dots, x_n$
and Output p



$$p = \frac{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2} \bar{p}^{i_1,i_2} \mu_{A_1^{i_1}}(e(t)) \mu_{A_2^{i_2}}(\dot{e}(t))}{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2} \mu_{A_1^{i_1}}(e(t)) \mu_{A_2^{i_2}}(\dot{e}(t))}$$

Tuning Parameters

Choose Inputs
$x_1, x_2, \dots, x_n$
and Output p

$$k'_p = \frac{k_p - k_{p,\min}}{k_{p,\max} - k_{p,\min}}$$

$$k'_d = \frac{k_d - k_{d,\min}}{k_{d,\max} - k_{d,\min}}$$

$$\alpha = \frac{T_i}{T_d}$$

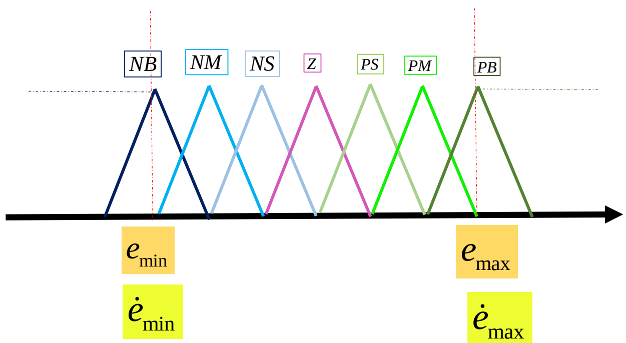
Fuzzy System Output

$$K[e(t) + \frac{1}{T_i}\int_{\tau=0}^{t} e(\tau)d\tau + T_d \frac{d}{dt}e(t)]$$

Input Membership Functions

Obtain Inputs Bounds $[x_{i,min}, x_{i,max}]$

Select Membership Functions $\mu_{A_i^l}(x_i)$

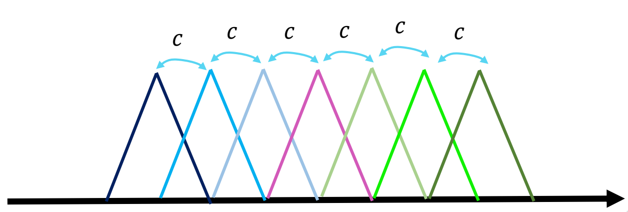


Input Membership Functions

```
e_max=0.3;
e_min=-0.3;
%*********
%*********MF e
ai=e_min;
af=e_max;
ne=7;
eff=ai:0.001:af;
c=(af-(ai))/(ne-1);
for i=1:ne
    mu(i,:)=trimf(eff,[ai+(i-2)*c ai+(i-1)*c ai+i*c]);
end
```

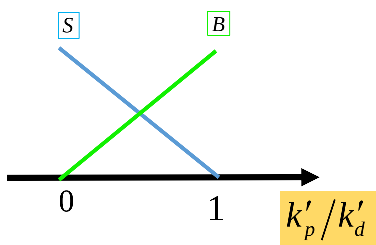
Obtain Inputs Bounds $[x_{i,min}, x_{i,max}]$

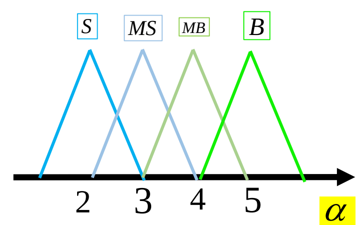
Select Membership Functions $\mu_{A_i^l}(x_i)$



Output Membership Functions

Construct Rules $\overline{p}^{\,l}$



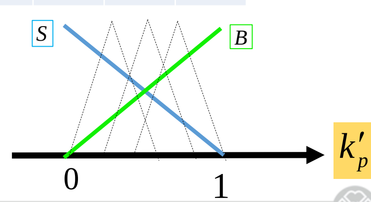


Fuzzy Rules

Construct Rules $\overline{p}^{\,l}$

k'_p		$\dot{e}$						
		NB	NM	NS	Z	PS	PM	PB
e	NB	**B**	**B**	**B**	**B**	**B**	**B**	**B**
	NM	**S**	**B**	**B**	**B**	**B**	**B**	**S**
	NS	**S**	**S**	**B**	**B**	**B**	**S**	**S**
	Z	**S**	**S**	**S**	**B**	**S**	**S**	**S**
	PS	**S**	**S**	**B**	**B**	**B**	**S**	**S**
	PM	**S**	**B**	**B**	**B**	**B**	**B**	**S**
	PB	**B**	**B**	**B**	**B**	**B**	**B**	**B**

$$k'_p(t) = \frac{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2} \overline{k'_p}^{\,i_1,i_2} \mu_{A_1^{i_1}}(e(t))\,\mu_{A_2^{i_2}}(\dot{e}(t))}{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2} \mu_{A_1^{i_1}}(e(t))\,\mu_{A_2^{i_2}}(\dot{e}(t))}$$



smgc

Fuzzy Rules

Construct Rules $\overline{p}^{\,l}$

```
R_kpf=[1 1 1 1 1 1 1;
       0 1 1 1 1 1 0;
       0 0 1 1 1 0 0;
       0 0 0 1 0 0 0;
       0 0 1 1 1 0 0;
       0 1 1 1 1 1 0;
       1 1 1 1 1 1 1];
```

k_p'		$\dot{e}$						
		NB	NM	NS	Z	PS	PM	PB
e	NB	**B**	**B**	**B**	**B**	**B**	**B**	**B**
	NM	**S**	**B**	**B**	**B**	**B**	**B**	**S**
	NS	**S**	**S**	**B**	**B**	**B**	**S**	**S**
	Z	**S**	**S**	**S**	**B**	**S**	**S**	**S**
	PS	**S**	**S**	**B**	**B**	**B**	**S**	**S**
	PM	**S**	**B**	**B**	**B**	**B**	**B**	**S**
	PB	**B**	**B**	**B**	**B**	**B**	**B**	**B**

smgc

Fuzzy Rules

Construct Rules $\overline{p}^{\,l}$

k_d'		$\dot{e}$						
		NB	NM	NS	Z	PS	PM	PB
e	NB	**S**	**S**	**S**	**S**	**S**	**S**	**S**
	NM	**B**	**B**	**S**	**S**	**S**	**B**	**B**
	NS	**B**	**B**	**B**	**S**	**B**	**B**	**B**
	Z	**B**	**B**	**B**	**B**	**B**	**B**	**B**
	PS	**B**	**B**	**B**	**S**	**B**	**B**	**B**
	PM	**B**	**B**	**S**	**S**	**S**	**B**	**B**
	PB	**S**	**S**	**S**	**S**	**S**	**S**	**S**

smgc

Fuzzy Rules

Construct Rules $\overline{p}^{\,l}$

α		$\dot{e}$						
		NB	NM	NS	Z	PS	PM	PB
e	NB	**S**	**S**	**S**	**S**	**S**	**S**	**S**
	NM	**MS**	**MS**	**S**	**S**	**S**	**MS**	**MS**
	NS	**MB**	**MS**	**MS**	**S**	**MS**	**MS**	**MB**
	Z	**B**	**MB**	**MS**	**MS**	**MS**	**MB**	**B**
	PS	**MB**	**MS**	**MS**	**S**	**MS**	**MS**	**MB**
	PM	**MS**	**MS**	**S**	**S**	**S**	**MS**	**MS**
	PB	**S**	**S**	**S**	**S**	**S**	**S**	**S**

smgc

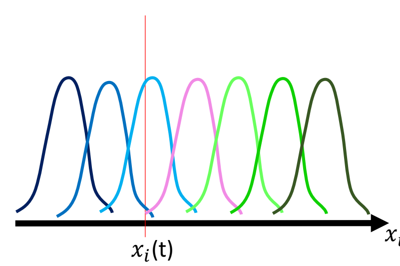
Fuzzy Supervisory

Calculate Inputs x_i

Obtain MF Values $\mu_{A_i^1}(x_i), \mu_{A_i^2}(x_i), \ldots$

Obtain $\overline{y}^{\,l}$ Based on Rules

Calculate Output



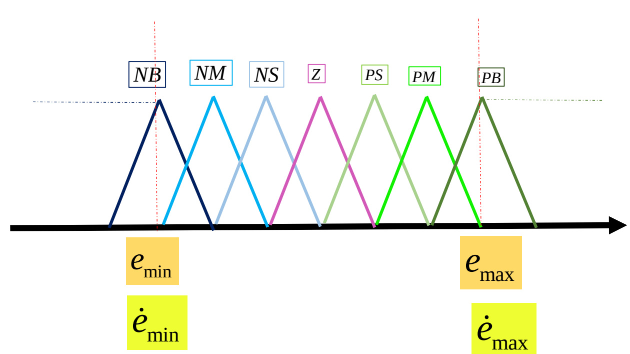
$e(t)$ & $\dot{e}(t)$ Membership Functions Values

Calculate Inputs
$e(t), \dot{e}(t)$

```
ef(1,t)=x1r(1,t-1)-x1f(1,t-1);
edf(1,t)=-x2f(1,t-1);
eif(1,t)=ef(1,t)*dt+eif(1,t-1);
```


$e(t)$ & $\dot{e}(t)$ Membership Functions Values

Obtain MF Values
$\mu_{A_e^1}(e(t)), \mu_{A_e^2}(e(t)), \ldots, \mu_{A_e^7}(e(t))$
$\mu_{A_{\dot{e}}^1}(\dot{e}(t)), \mu_{A_{\dot{e}}^2}(e(\dot{t})), \ldots, \mu_{A_{\dot{e}}^7}(\dot{e}(t))$



$e(t)$ & $\dot{e}(t)$ Membership Functions Values

Obtain MF Values
$\mu_{A_e^1}(e(t)), \mu_{A_e^2}(e(t)),\ldots, \mu_{A_e^7}(e(t))$
$\mu_{A_{\dot{e}}^1}(\dot{e}(t)), \mu_{A_{\dot{e}}^2}(\dot{e}(t)), \ldots, \mu_{A_{\dot{e}}^7}(\dot{e}(t))$

```
ee=ef(1,t-1);
eed=edf(1,t-1);
%********
if ee>e_max
    ee=e_max;
end
if ee<e_min
    ee=e_min;
end
```


$e(t)$ & $\dot{e}(t)$ Membership Functions Values

Obtain MF Values
$\mu_{A_e^1}(e(t)), \mu_{A_e^2}(e(t)),\ldots, \mu_{A_e^7}(e(t))$
$\mu_{A_{\dot{e}}^1}(\dot{e}(t)), \mu_{A_{\dot{e}}^2}(\dot{e}(t)), \ldots, \mu_{A_{\dot{e}}^7}(\dot{e}(t))$

```
finder_e=ee*ones(1,length(eff));
[mme,nee]=min(abs(ee-eff));

finder_ed=eed*ones(1,length(edff));
[mmed,need]=min(abs(eed-edff));
```

smgrc

Calculate FLS Outputs Values

Calculate Output

accd=0; accn_kpf=0; accn_kdf=0; accn_alphaf=0

```
for i=1:ne
  for j=1:ned
   accd=accd+mu(i,nee)*mud(j,need);
      %********kp
   accn_kpf=accn_kpf+R_kpf(i,j)*mu(i,nee)*mud(j,need);
      %*******kd
   accn_kdf=accn_kdf+R_kdf(i,j)*mu(i,nee)*mud(j,need);
      %*******alpha
   accn_alphaf=accn_alphaf+R_alphaf(i,j)*mu(i,nee)*mud(j,need);


  end
end
```

$$p = \frac{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2} \bar{p}^{i_1,i_2} \mu_{A_1^{i_1}}(e(t))\mu_{A_2^{i_2}}(\dot{e}(t))}{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2} \mu_{A_1^{i_1}}(e(t))\mu_{A_2^{i_2}}(\dot{e}(t))}$$

smgrc

Calculate FLS Outputs Values

Calculate Output

```
kpf(1,t)=(accn_kpf/accd)*(kp_max-kp_min)+kp_min;

kdf(1,t)=(accn_kdf/accd)*(kd_max-kd_min)+kd_min;

alphaf(1,t)=(accn_alphaf/accd);
```

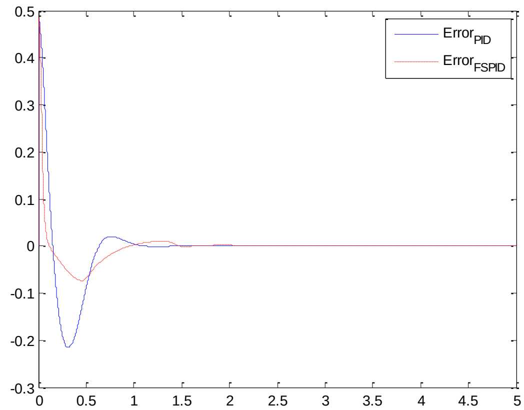
$$p = \frac{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2} \bar{p}^{i_1,i_2} \mu_{A_1^{i_1}}(e(t))\mu_{A_2^{i_2}}(\dot{e}(t))}{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2} \mu_{A_1^{i_1}}(e(t))\mu_{A_2^{i_2}}(\dot{e}(t))}$$

$$k'_p = \frac{k_p - k_{p,\min}}{k_{p,\max} - k_{p,\min}}$$

$$k'_d = \frac{k_d - k_{d,\min}}{k_{d,\max} - k_{d,\min}}$$

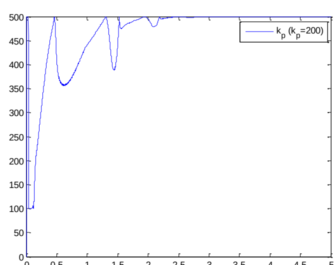
smgc

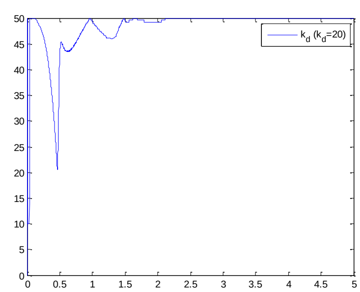
PID vs FSPID

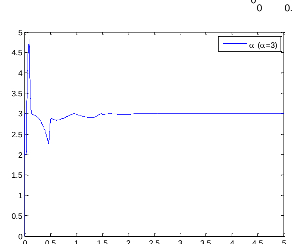


smgc

PID vs FSPID

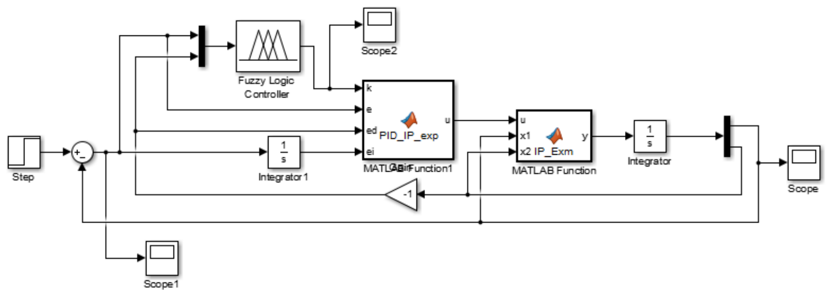






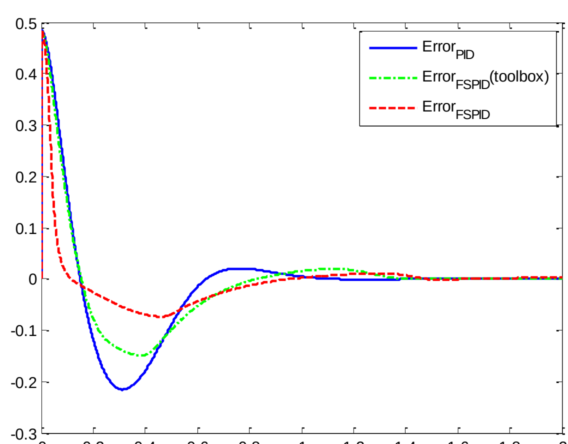
smgc

Fuzzy Supervisory with Toolbox



smgc

Personal Code vs. Toolbox



Thanks