Outline

1. Singleton and Takagi–Sugeno fuzzy system.
2. Knowledge based fuzzy modeling.
3. Data-driven construction.
4. Direct fuzzy control.
5. Supervisory fuzzy control.

Singleton Fuzzy Model

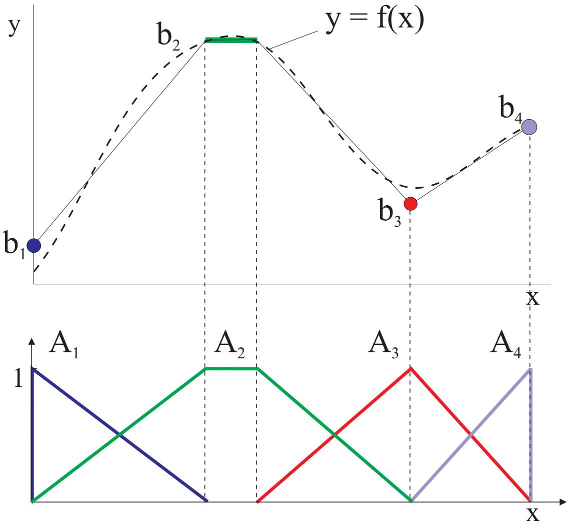
$$\textbf{If } x \textbf{ is } A_i \textbf{ then } y = b_i$$

Inference/defuzzification:

$$y = \frac{\sum_{i=1}^{K} \mu_{A_i}(x) b_i}{\sum_{i=1}^{K} \mu_{A_i}(x)}$$

- well-understood approximation properties
- straightforward parameter estimation

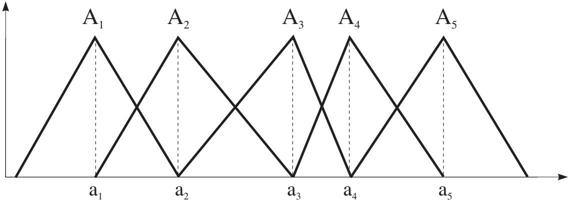
Piece-wise Linear Approximation



Linear Mapping with a Singleton Model

$$y = \boldsymbol{k}^T \boldsymbol{x} + q = \sum_{j=1}^{p} k_j x_j + q$$

- Triangular partition:



- Consequent singletons are equal to:

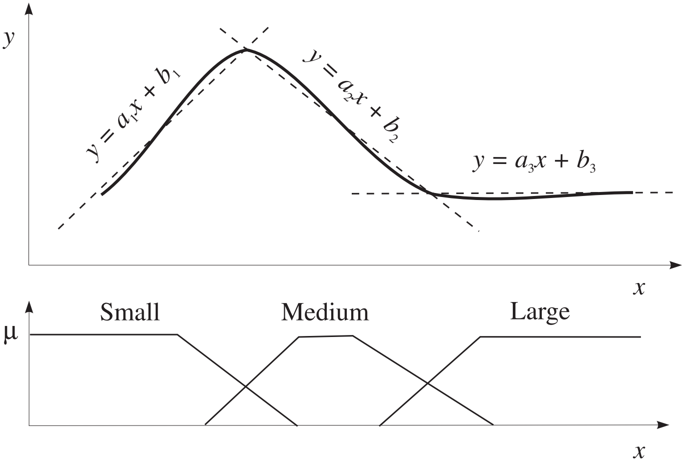
$$b_i = \sum_{j=1}^{p} k_j a_{i,j} + q$$

Takagi–Sugeno (TS) Fuzzy Model

If x **is** A_i **then** $y_i = a_i x + b_i$

$$y = \frac{\sum_{i=1}^{K} \mu_{A_i}(x) y_i}{\sum_{i=1}^{K} \mu_{A_i}(x)} = \frac{\sum_{i=1}^{K} \mu_{A_i}(x)(a_i x + b_i)}{\sum_{i=1}^{K} \mu_{A_i}(x)}$$

Input-Output Mapping of the TS Model



Consequents are approximate local linear models of the system.

TS Model is a Quasi-Linear System

$$y = \frac{\sum_{i=1}^{K} \mu_{A_i}(\boldsymbol{x}) y_i}{\sum_{j=1}^{K} \mu_{A_j}(\boldsymbol{x})} = \frac{\sum_{i=1}^{K} \mu_{A_i}(\boldsymbol{x})(\boldsymbol{a}_i^T \boldsymbol{x} + b_i)}{\sum_{j=1}^{K} \mu_{A_j}(\boldsymbol{x})}$$

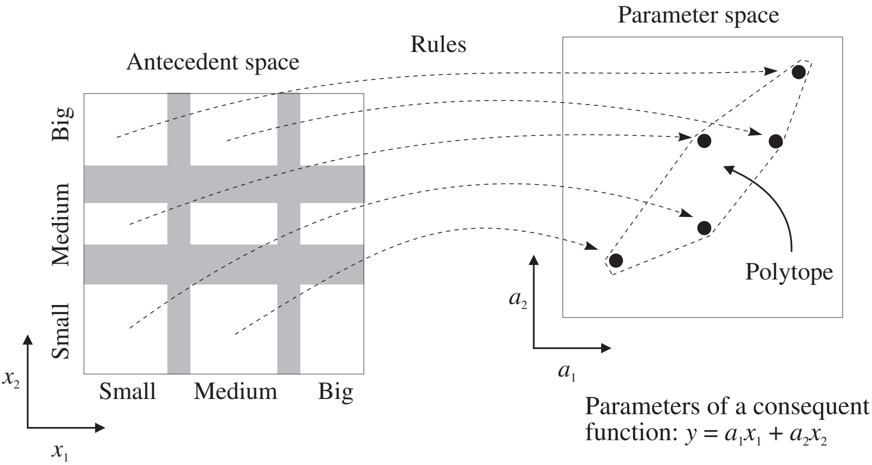
TS Model is a Quasi-Linear System

$$y = \frac{\sum_{i=1}^{K} \mu_{A_i}(\boldsymbol{x}) y_i}{\sum_{j=1}^{K} \mu_{A_j}(\boldsymbol{x})} = \frac{\sum_{i=1}^{K} \mu_{A_i}(\boldsymbol{x})(\boldsymbol{a}_i^T \boldsymbol{x} + b_i)}{\sum_{j=1}^{K} \mu_{A_j}(\boldsymbol{x})}$$

$$y = \underbrace{\left(\sum_{i=1}^{K} \gamma_i(\boldsymbol{x}) \boldsymbol{a}_i^T \right)}_{\boldsymbol{a}(\boldsymbol{x})^T} \boldsymbol{x} + \underbrace{\sum_{i=1}^{K} \gamma_i(\boldsymbol{x}) b_i}_{b(\boldsymbol{x})}$$

linear in parameters a_i and b_i, pseudo-linear in $\boldsymbol{x}$ (LPV)

TS Model is a Polytopic System



Construction of Fuzzy Models

Modeling Paradigms

- Mechanistic (white-box, physical)
- Qualitative (naive physics, knowledge-based)
- Data-driven (black-box, inductive)

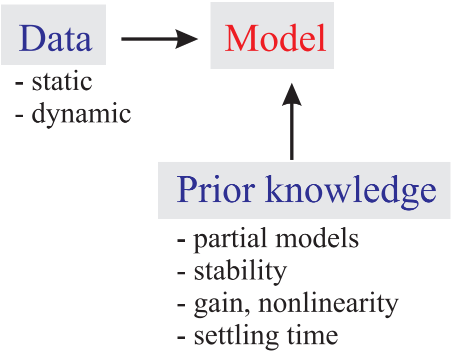
Often combination of different approaches semi-mechanistic, gray-box modeling.

Parameterization of nonlinear models

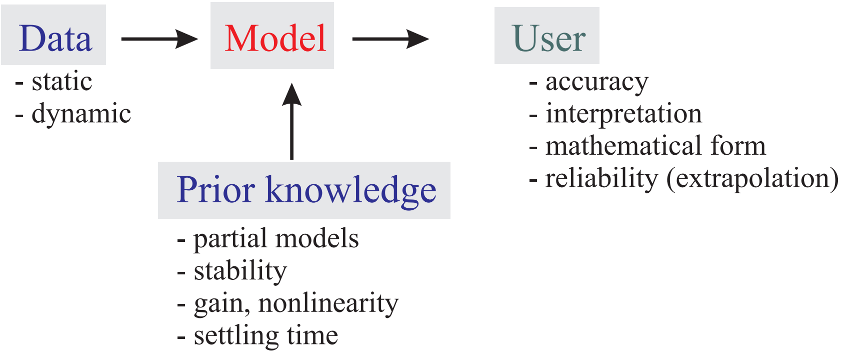
- polynomials, splines
- look-up tables
- fuzzy systems
- neural networks
- (neuro-)fuzzy systems
- radial basis function networks
- wavelet networks
- ...

Modeling of Complex Systems

Modeling of Complex Systems



Modeling of Complex Systems



Building Fuzzy Models

Knowledge-based approach:

- expert knowledge $\rightarrow$ rules & membership functions
- fuzzy model of human operator
- linguistic interpretation

Building Fuzzy Models

Knowledge-based approach:

- expert knowledge $\rightarrow$ rules & membership functions
- fuzzy model of human operator
- linguistic interpretation

Data-driven approach:

- nonlinear mapping, universal approximation
- extract rules & membership functions from data

Knowledge-Based Modeling

- Problems where little or no data are available.
- Similar to expert systems.
- Presence of both quantitative and qualitative variables or parameters.

Typical applications: fuzzy control and decision support, but also modeling of poorly understood processes

Wear Prediction for a Trencher

Trencher T-850 (Vermeer)

Chain Detail

Goal: Given the terrain properties, predict bit wear and production rate of trencher.

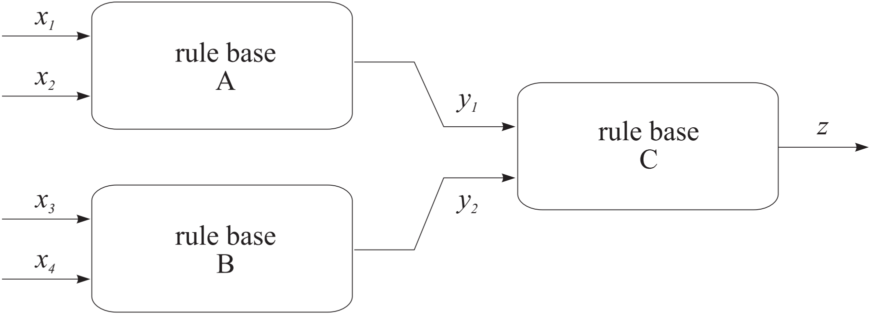
Why Knowledge-Based Modeling?

- Interaction between tool and environment is complex, dynamic and highly nonlinear, rigorous mathematical models are not available.
- Little data (15 data points) to develop statistical regression models.
- Input data are a mixture of numerical measurements (rock strength, joint spacing, trench dimensions) and qualitative information (joint orientation).
- Precise numerical output not needed, qualitative assessment is sufficient.

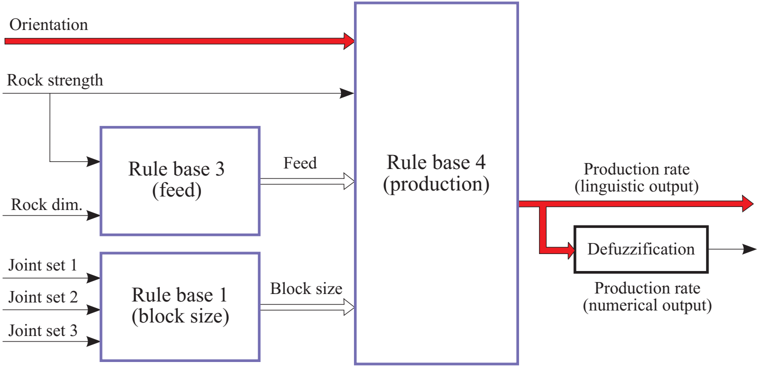
Dimensionality Problem: Hierarchical Structure

Assume 5 membership functions for each input

625 rules in a flat rule base vs. 75 rules in a hierarchical one

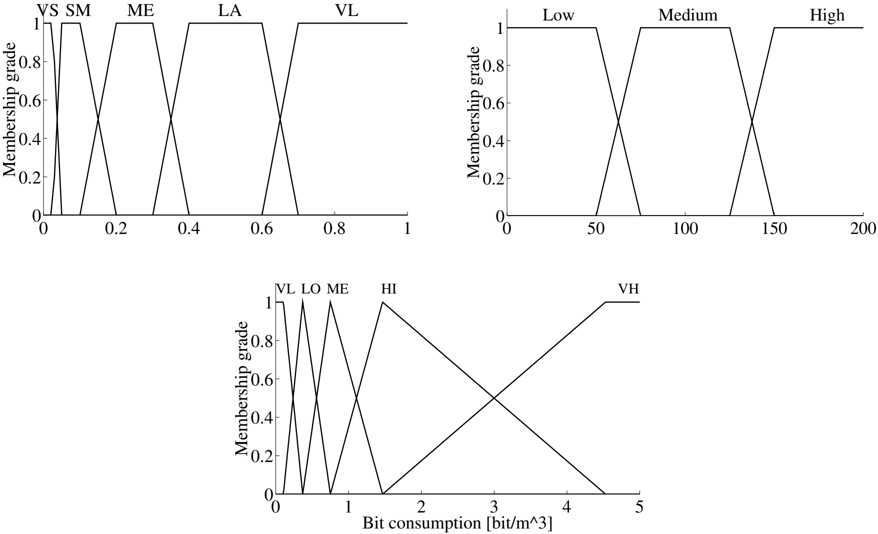


Trencher: Fuzzy Rule Bases




```
If TRENCH-DIM is SMALL  and STRENGTH is LOW    Then FEED is VERY-HIGH;
If TRENCH-DIM is SMALL  and STRENGTH is MEDIUM Then FEED is HIGH;
....
If JOINT-SP is EXTRA-LARGE and FEED is VERY-HIGH Then PROD is VERY-HIGH
```

Example of Membership Functions



Output: Prediction of Production Rate

data no.	measured value	predicted linguistic value(s)			
1	2.07	VERY-LOW	1.00		
2	5.56	HIGH	1.00		
3	23.60	VERY-HIGH	0.50		
4	11.90	HIGH	0.40	VERY-HIGH	0.60
5	7.71	MEDIUM	1.00		
6	7.17	LOW	0.72		
7	8.05	MEDIUM	0.80		
8	7.39	LOW	1.00		
9	4.58	LOW	0.50		
10	8.74	MEDIUM	1.00		
11	134.84	EXTREMELY-HIGH	1.00		

Data-Driven Construction

Structure and Parameters

Structure:

- Input and output variables. For dynamic systems also the representation of the dynamics.
- Number of membership functions per variable, type of membership functions, number of rules.

Parameters:

- Consequent parameters (least squares).
- Antecedent membership functions (various methods).

Least-Squares Estimation of Singletons

$$R_i: \quad \textbf{If } \boldsymbol{x} \textbf{ is } A_i \textbf{ then } y = b_i$$

- Given A_i and a set of input–output data:

$$\{(\boldsymbol{x}_k, y_k) \mid k = 1, 2, \ldots, N\}$$

- Estimate optimal consequent parameters b_i.

Least-Squares Estimation of Singletons

1. Compute the membership degrees $\mu_{A_i}(\mathbf{x}_k)$
2. Normalize

$$\gamma_{ki} = \mu_{A_i}(\mathbf{x}_k) / \sum_{j=1}^{K} \mu_{A_j}(\mathbf{x}_k)$$

(Output: $y_k = \sum_{i=1}^{K} \gamma_{ki} b_i$, in a matrix form: $\mathbf{y} = \mathbf{\Gamma} \mathbf{b}$)

3. Least-squares estimate: $\mathbf{b} = \left[\mathbf{\Gamma}^T \mathbf{\Gamma}\right]^{-1} \mathbf{\Gamma}^T \mathbf{y}$

Least-Square Estimation of TS Consequents

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_N^T \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad \boldsymbol{\Gamma}_i = \begin{bmatrix} \gamma_{i1} & 0 & \cdots & 0 \\ 0 & \gamma_{i2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_{iN} \end{bmatrix}$$

$$\boldsymbol{\theta}_i = \begin{bmatrix} \boldsymbol{a}_i^T & b_i \end{bmatrix}^T, \quad \boldsymbol{X}_e = [\boldsymbol{X} \quad \mathbf{1}]$$

Least-Square Estimation of TS Consequents

- Global LS: $\boldsymbol{\theta}' = \left[(\boldsymbol{X}')^T \boldsymbol{X}' \right]^{-1} (\boldsymbol{X}')^T \boldsymbol{y}$

with $\boldsymbol{X}' = [\boldsymbol{\Gamma}_1 \boldsymbol{X}_e \quad \boldsymbol{\Gamma}_2 \boldsymbol{X}_e \quad \dots \quad \boldsymbol{\Gamma}_c \boldsymbol{X}_e]$

and $\boldsymbol{\theta}' = \left[\boldsymbol{\theta}_1^T \quad \boldsymbol{\theta}_2^T \quad \dots \quad \boldsymbol{\theta}_c^T \right]^T$

Least-Square Estimation of TS Consequents

- Global LS: $\boldsymbol{\theta}' = \left[(\boldsymbol{X}')^T \boldsymbol{X}'\right]^{-1} (\boldsymbol{X}')^T \boldsymbol{y}$

  with $\boldsymbol{X}' = [\boldsymbol{\Gamma}_1 \boldsymbol{X}_e \quad \boldsymbol{\Gamma}_2 \boldsymbol{X}_e \quad \ldots \quad \boldsymbol{\Gamma}_c \boldsymbol{X}_e]$

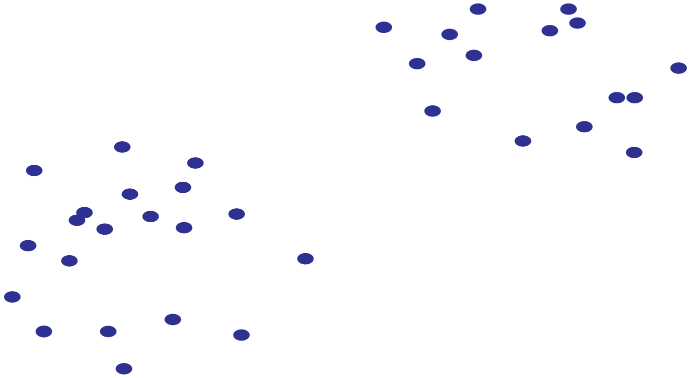
  and $\boldsymbol{\theta}' = \left[\boldsymbol{\theta}_1^T \quad \boldsymbol{\theta}_2^T \quad \ldots \quad \boldsymbol{\theta}_c^T\right]^T$

- Local LS: $\boldsymbol{\theta}_i = \left[\boldsymbol{X}_e^T \boldsymbol{\Gamma}_i \boldsymbol{X}_e\right]^{-1} \boldsymbol{X}_e^T \boldsymbol{\Gamma}_i \boldsymbol{y}$

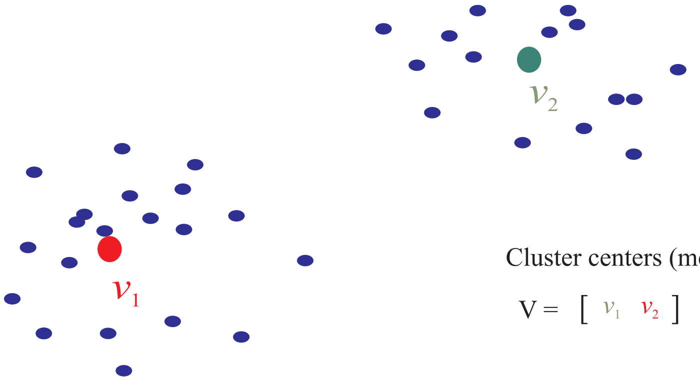
Antecedent Membership Functions

- templates (grid partitioning),
- nonlinear optimization (neuro-fuzzy methods),
- tree-construction
- product space fuzzy clustering

Fuzzy Clustering: Data



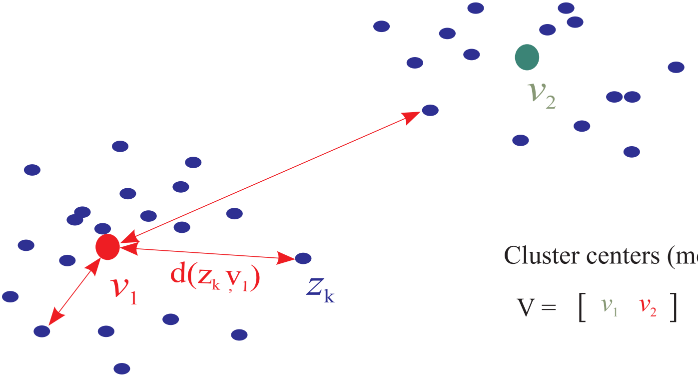
Fuzzy Clustering: Prototypes



Cluster centers (means):

$$V = [\ v_1 \quad v_2\ ]$$

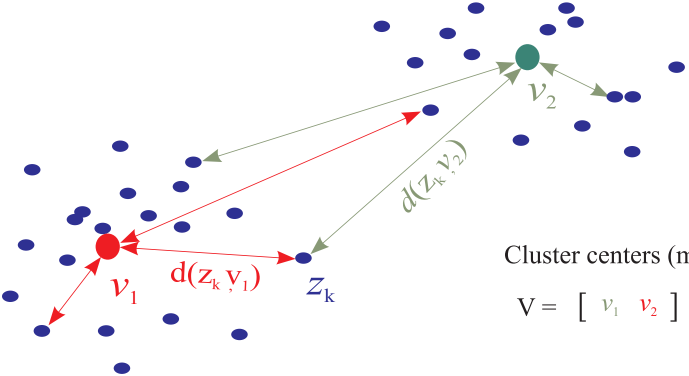
Fuzzy Clustering: Distance



Cluster centers (means):

$V = [\ v_1 \quad v_2\ ]$

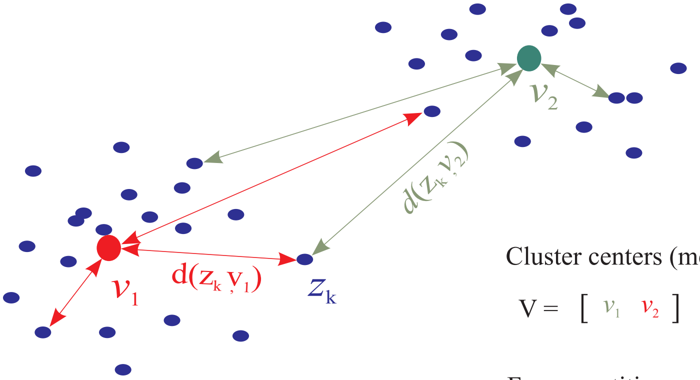
Fuzzy Clustering: Distance



Cluster centers (means):

$V = [\ v_1 \quad v_2\ ]$

Fuzzy Clustering: Partition Matrix



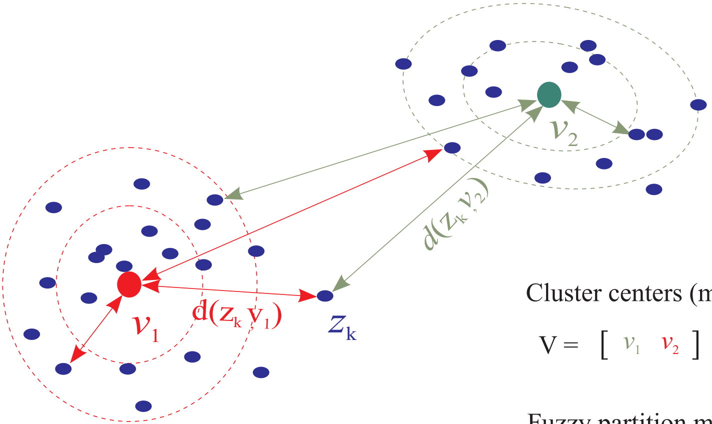
Cluster centers (means):

$$V = [\, v_1 \quad v_2 \,]$$

Fuzzy partition matrix:

$$U = \begin{bmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2N} \end{bmatrix}$$

Fuzzy Clustering: Shapes



Cluster centers (means):

$$V = [\; v_1 \quad v_2 \;]$$

Fuzzy partition matrix:

$$U = \begin{bmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2N} \end{bmatrix}$$

Fuzzy Clustering Problem

Given the data:

$$\mathbf{z}_k = [z_{1k}, z_{2k}, \ldots, z_{nk}]^T \in \mathbb{R}^n, \quad k = 1, \ldots, N$$

Find:
the fuzzy partition matrix:

$$\mathbf{U} = \begin{bmatrix} \mu_{11} & \ldots & \mu_{1k} & \ldots & \mu_{1N} \\ \vdots & \ldots & \vdots & \ldots & \vdots \\ \mu_{c1} & \ldots & \mu_{ck} & \ldots & \mu_{cN} \end{bmatrix}$$

and the cluster centers:

$$\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c\}, \quad \mathbf{v}_i \in \mathbb{R}^n$$

Fuzzy Clustering: an Optimization Approach

Objective function (least-squares criterion):

$$J(Z; V, U, A) = \sum_{i=1}^{c} \sum_{j=1}^{N} \mu_{i,j}^{m} d_{A_i}^{2}(\mathbf{z}_j, \mathbf{v}_i)$$

subject to constraints:

$0 \leq \mu_{i,j} \leq 1,$	$i = 1, \ldots, c, \; j = 1, \ldots, N$	membership degree
$0 < \sum_{j=1}^{N} \mu_{i,j} < N,$	$i = 1, \ldots, c$	no cluster empty
$\sum_{i=1}^{c} \mu_{i,j} = 1,$	$j = 1, \ldots, N$	total membership

Fuzzy c-Means Algorithm

Repeat:

1. Compute cluster prototypes (means): $v_i = \frac{\sum_{k=1}^{N} \mu_{i,k}^{m} \mathbf{z}_k}{\sum_{k=1}^{N} \mu_{i,k}^{m}}$

Fuzzy c-Means Algorithm

Repeat:

1. Compute cluster prototypes (means): $\boldsymbol{v}_i = \frac{\sum_{k=1}^{N} \mu_{i,k}^m \boldsymbol{z}_k}{\sum_{k=1}^{N} \mu_{i,k}^m}$
2. Calculate distances: $d_{ik} = (\boldsymbol{z}_k - \boldsymbol{v}_i)^T (\boldsymbol{z}_k - \boldsymbol{v}_i)$

Fuzzy c-Means Algorithm

Repeat:

1. Compute cluster prototypes (means): $\mathbf{v}_i = \frac{\sum_{k=1}^{N} \mu_{i,k}^m \mathbf{z}_k}{\sum_{k=1}^{N} \mu_{i,k}^m}$
2. Calculate distances: $d_{ik} = (\mathbf{z}_k - \mathbf{v}_i)^T (\mathbf{z}_k - \mathbf{v}_i)$
3. Update partition matrix: $\mu_{ik} = \frac{1}{\sum_{j=1}^{c} (d_{ik}/d_{jk})^{1/(m-1)}}$

until $\|\Delta \mathbf{U}\| < \epsilon$

Distance Measures

- Euclidean norm:

$$d^2(\boldsymbol{z}_j, \boldsymbol{v}_i) = (\boldsymbol{z}_j - \boldsymbol{v}_i)^T(\boldsymbol{z}_j - \boldsymbol{v}_i)$$

- Inner-product norm:

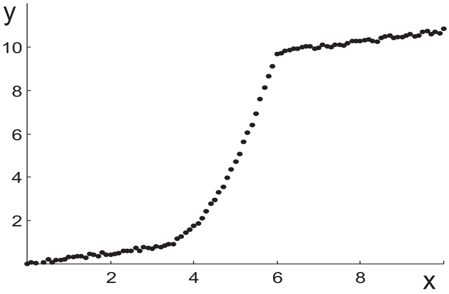
$$d^2_{A_i}(\boldsymbol{z}_j, \boldsymbol{v}_i) = (\boldsymbol{z}_j - \boldsymbol{v}_i)^T A_i(\boldsymbol{z}_j - \boldsymbol{v}_i)$$

- Many other possibilities ...

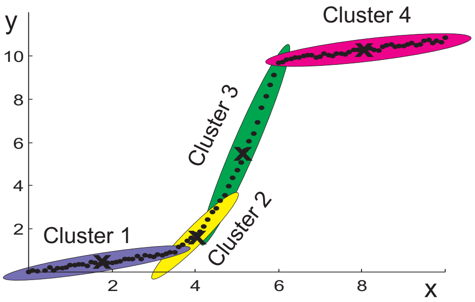
Fuzzy Clustering – Demo

1. Fuzzy c-means

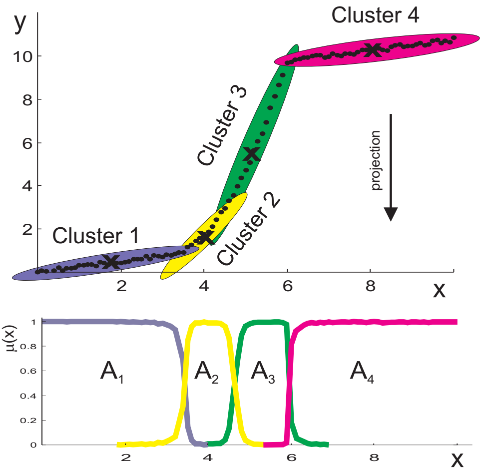
Extraction of Rules by Fuzzy Clustering



Extraction of Rules by Fuzzy Clustering



Extraction of Rules by Fuzzy Clustering



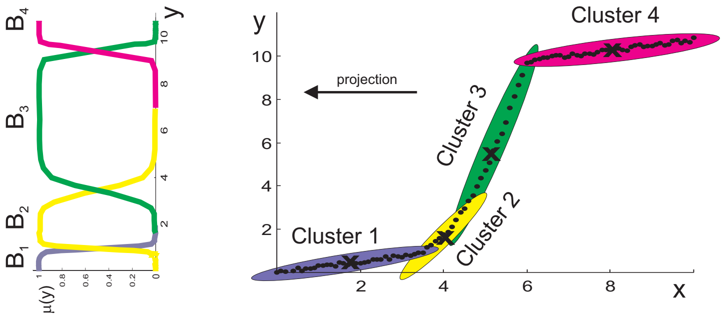
Takagi-Sugeno model

Rule-based description:

If x is A_1 then y = $a_1x + b_1$
If x is A_2 then y = $a_2x + b_2$

etc...

Extraction of Rules by Fuzzy Clustering



Rule-based description:

If y is B_1 then $x = a_1 y + b_1$
If y is B_2 then $x = a_2 y + b_2$

etc...

Inverse Takagi-Sugeno model

Rule Extraction – Demo

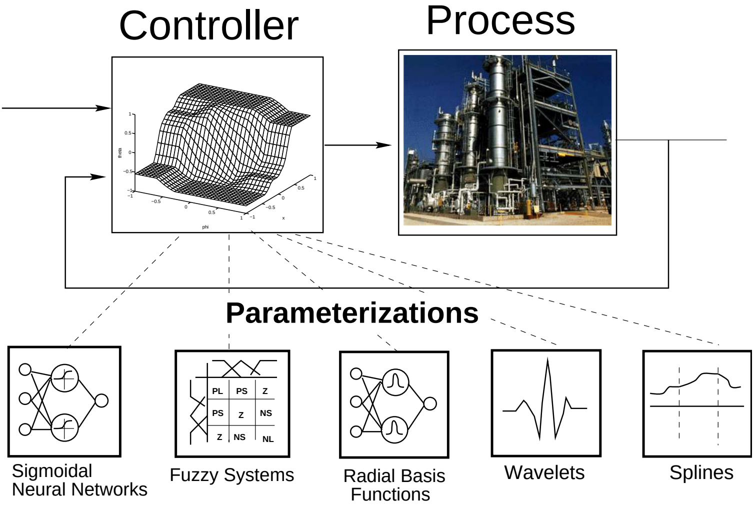
- Extraction of Takagi–Sugeno rules

Fuzzy Control

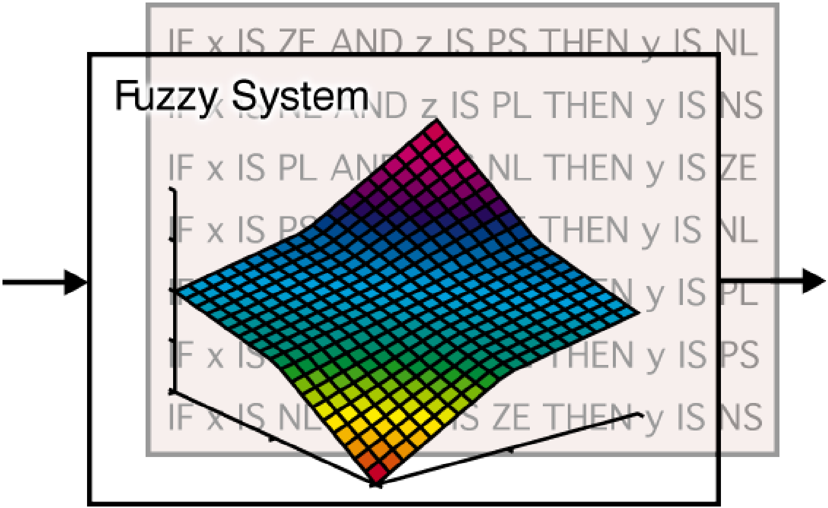
Fuzzy Control: Background

- controller designed by using If–Then rules instead of mathematical formulas (knowledge-based control),
- early motivation: mimic experienced operators,
- fuzzy reasoning: interpolation between discrete outputs,
- currently: also controllers designed on the basis of a fuzzy model (model-based fuzzy control),
- a fuzzy controller represents a *nonlinear* mapping (but completely deterministic!).

Parameterization of Nonlinear Controllers



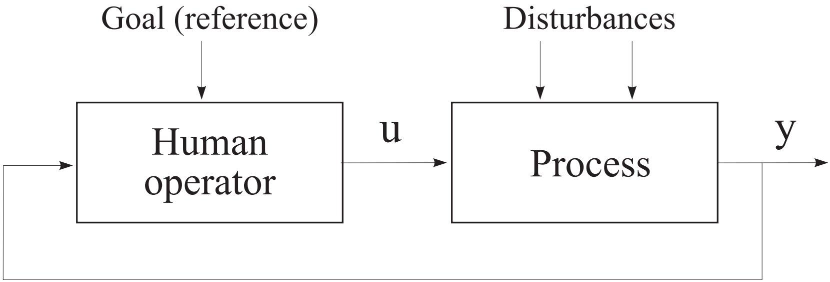
Fuzzy System is a Nonlinear Mapping



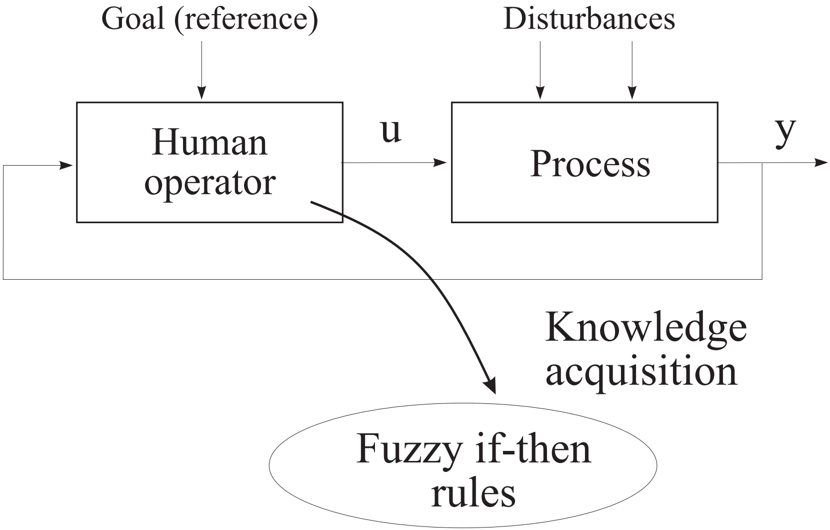
Basic Fuzzy Control Schemes

- Direct (low-level, Mamdani) fuzzy control
- Fuzzy supervisory (high-level, Takagi–Sugeno) control
- Fuzzy model-based control

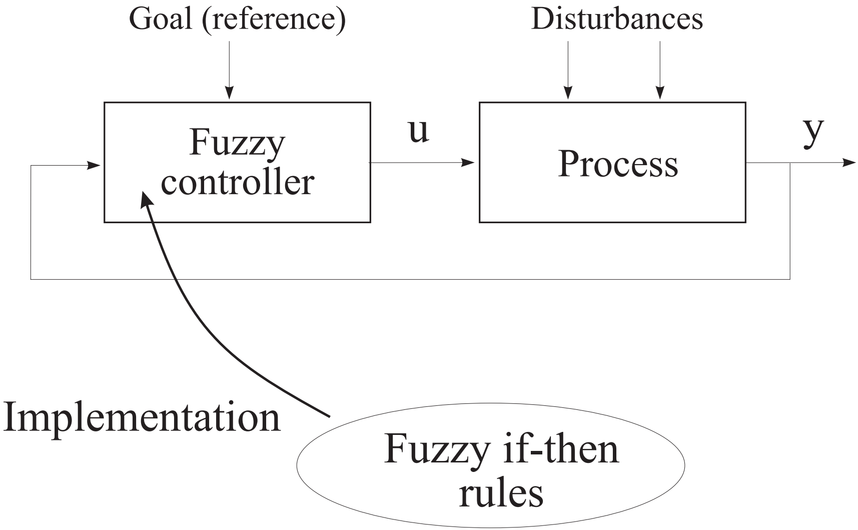
Process Controlled by Operators



Knowledge Acquisition



Direct Fuzzy Control

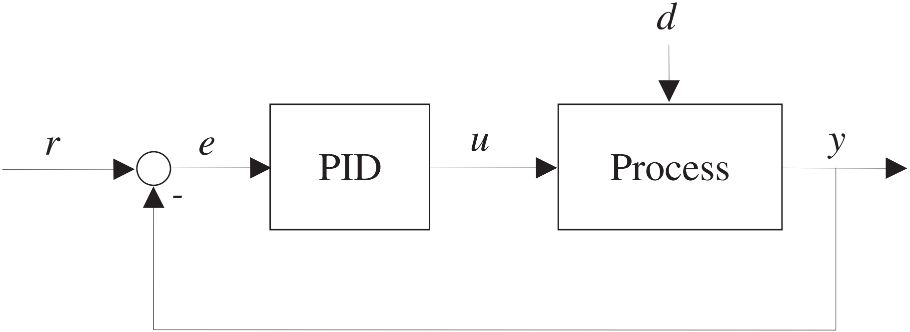


Example of Operator Knowledge

Case	Condition	Action to be taken	Reason
11	BZ OK OX low BE OK	a. Decrease fuel rate slightly	To raise percentage of oxygen
12	BZ OK OX low BE high	a. Reduce fuel rate b. Reduce fan speed	To increase percentage of oxygen for action b To lower back-end temperature and maintain burning zone temperature
13	BZ OK OX OK BE low	a. Increase fan speed b. Increase fuel rate	To raise back-end temperature To maintain burning zone temperature

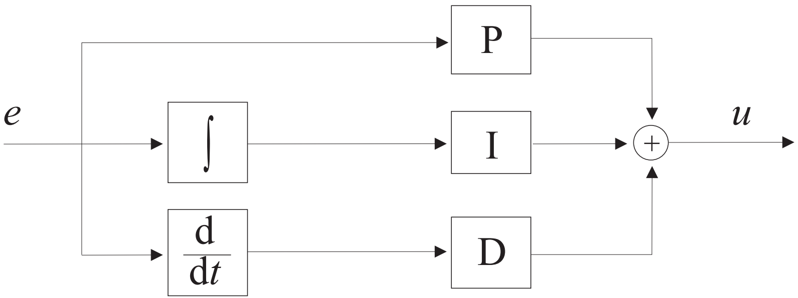
Extract from Peray's textbook for kiln operators (Oestergaard, 1999)

FLC Analogue to PID Control



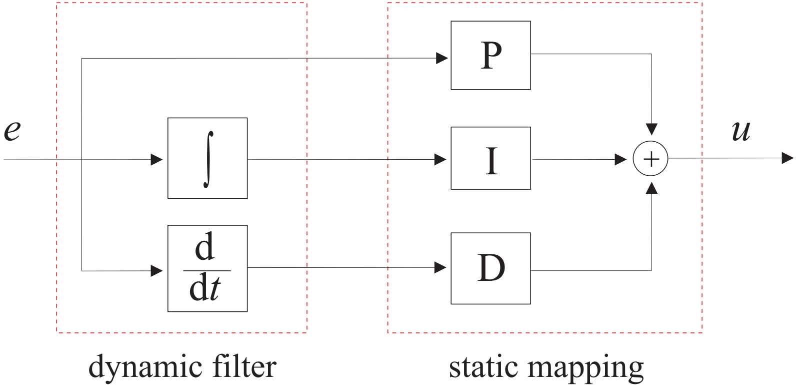
PID Control: Internal View

$$u(t) = Pe(t) + I\int_0^t e(\tau)d\tau + D\frac{de(t)}{dt}$$



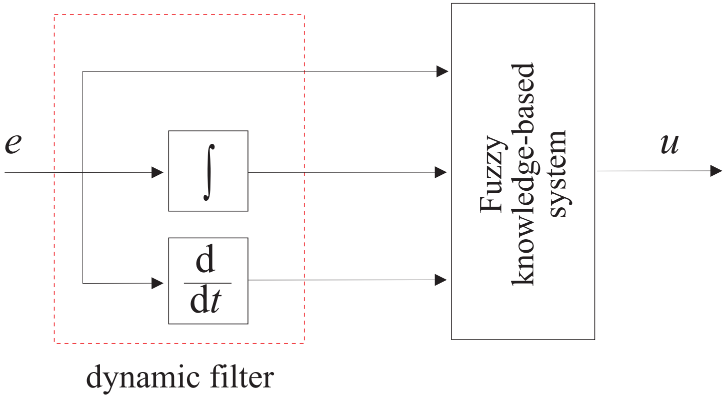
PID Control: Internal View

$$u(t) = Pe(t) + I\int_0^t e(\tau)d\tau + D\frac{de(t)}{dt}$$

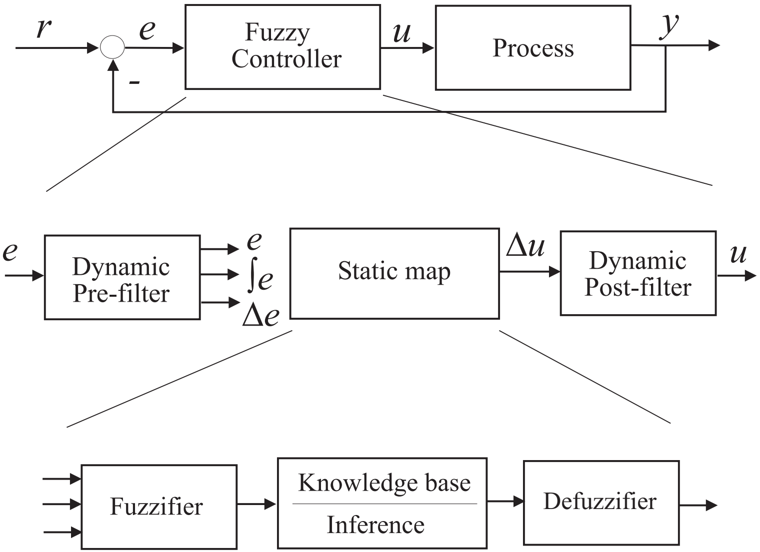


Fuzzy PID Control

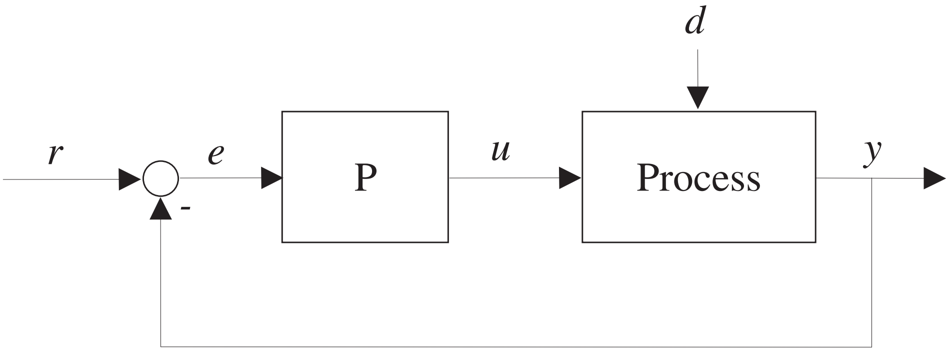
$$u(t) = f\left(e(t), \int_0^t e(\tau)d\tau, \frac{de(t)}{dt}\right)$$



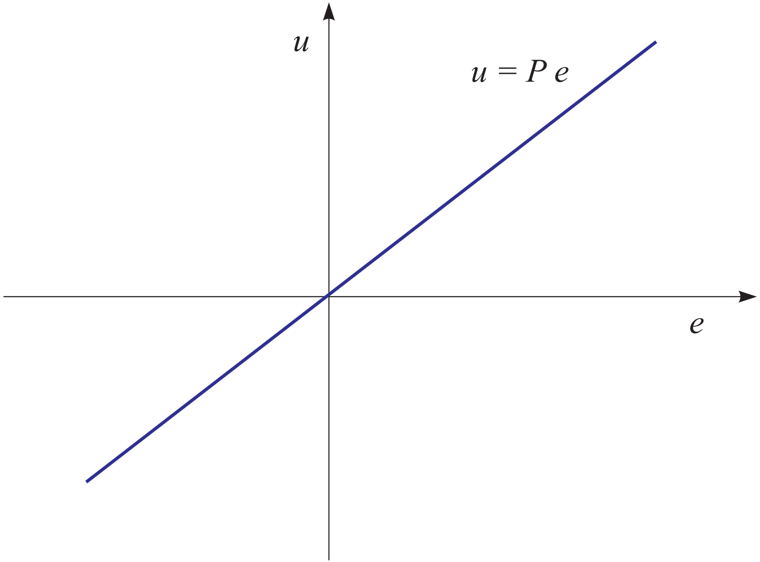
Fuzzy PID Control



Example: Proportional Control



Controller's Input–Output Mapping



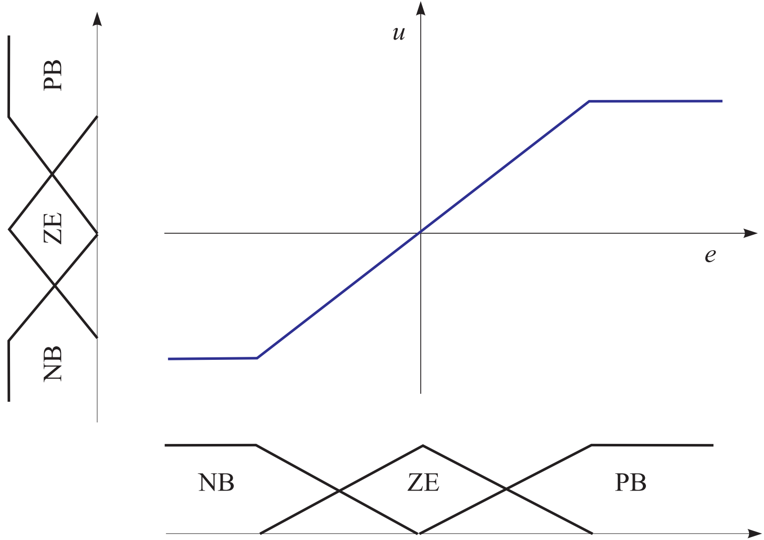
Fuzzy Proportional Control: Rules

If error is Negative Big **then** control input is Negative Big

If error is Positive Big **then** control input is Positive Big

If error is Zero **then** control input is Zero

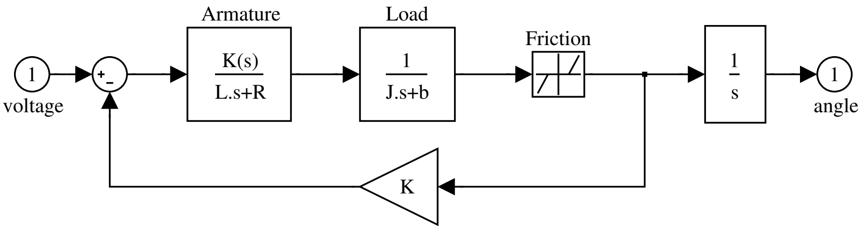
Controller's Input–Output Mapping



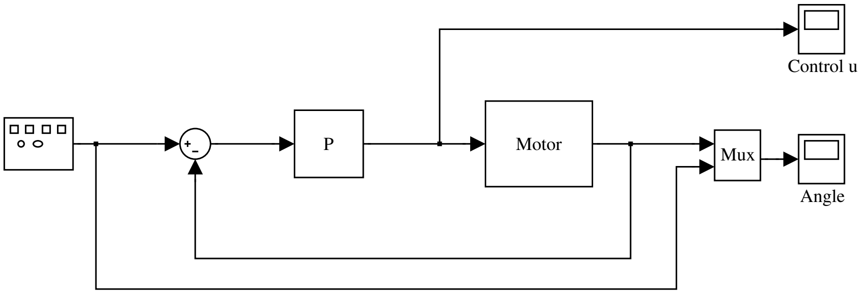
Example: Friction Compensation

1. DC motor with static friction.
2. Fuzzy rules to represent “normal” proportional control.
3. Additional rules to prevent undesirable states.

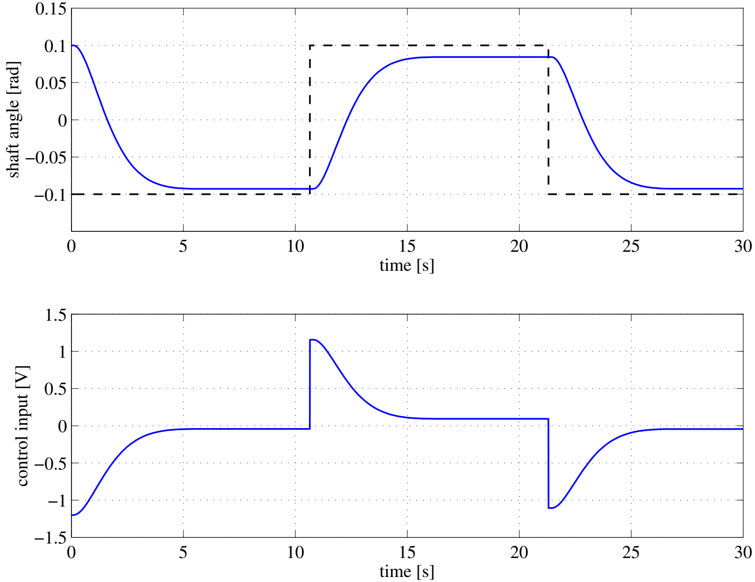
DC Motor: Model



Proportional Controller



Linear Control



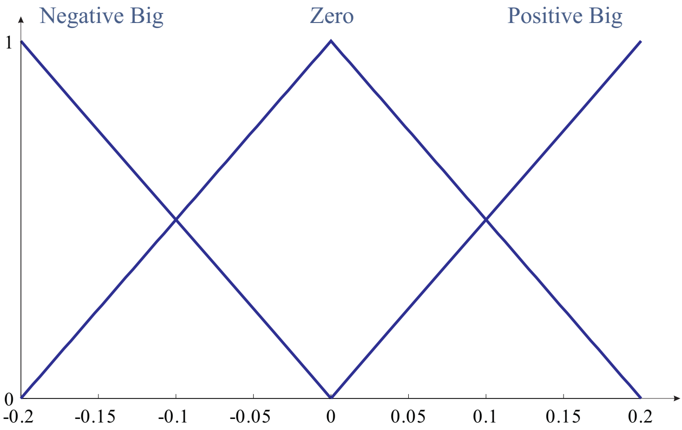
Fuzzy Control Rule Base

If error is Positive Big **then** control input is Positive Big;

If error is Negative Big **then** control input is Negative Big;

If error is Zero **then** control input is Zero;

Membership Functions for Error



Additional Rules

If error is Positive Big then control input is Positive Big;

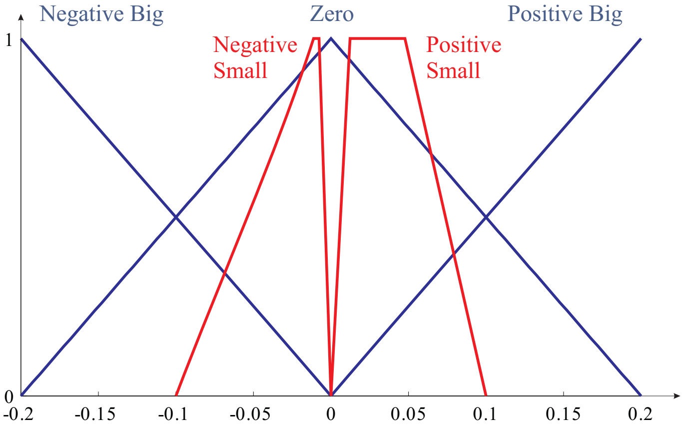
If error is Negative Big then control input is Negative Big;

If error is Zero then control input is Zero;

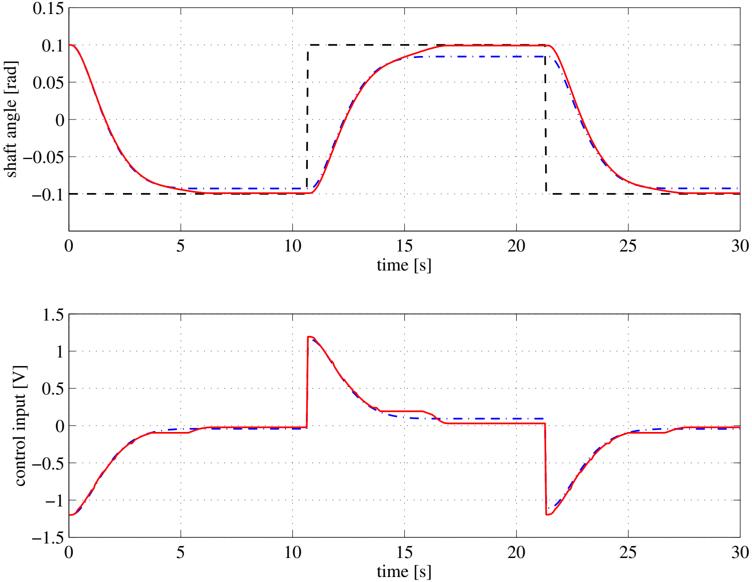
If error is Negative Small **then** control input is not Negative Small;

If error is Positive Small **then** control input is not Positive Small;

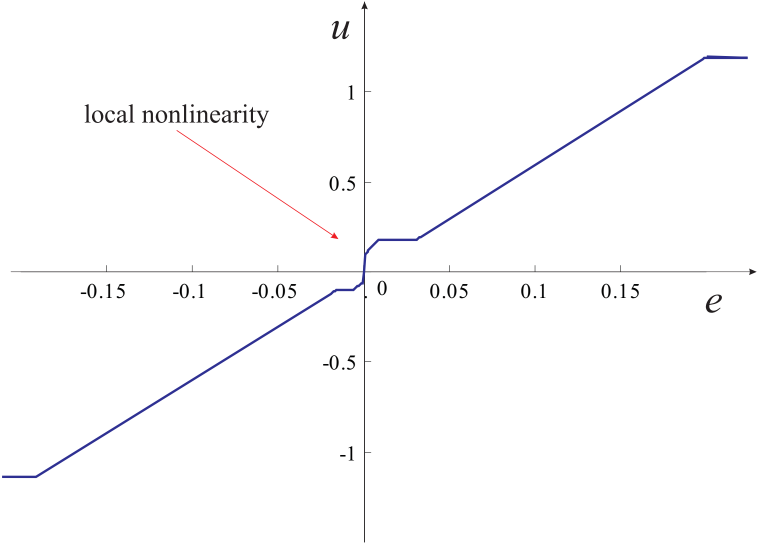
Membership Functions for Error



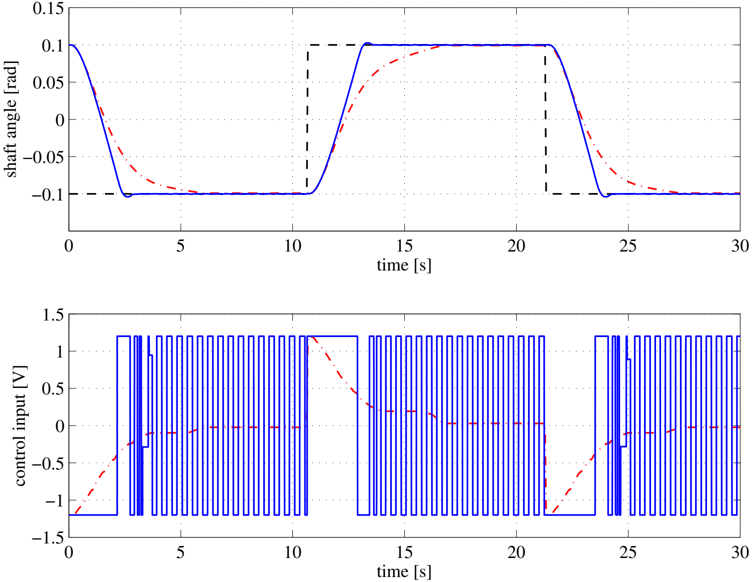
Fuzzy Control



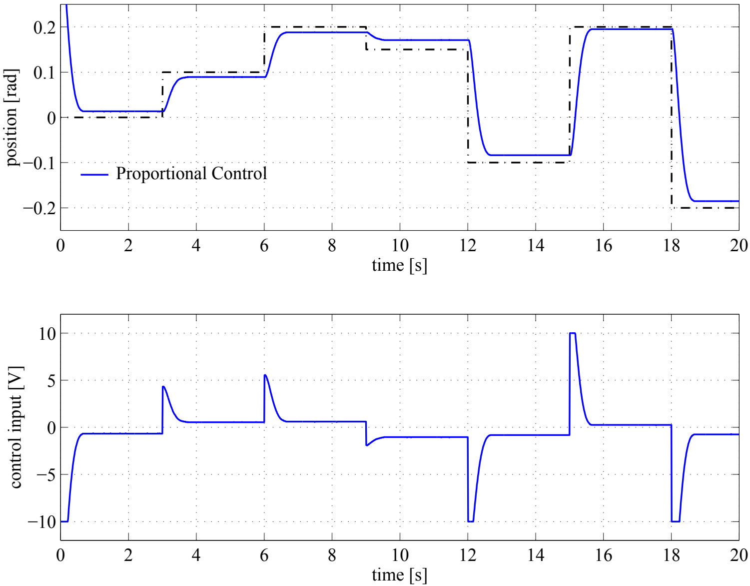
Input–Output Mapping of the Controller



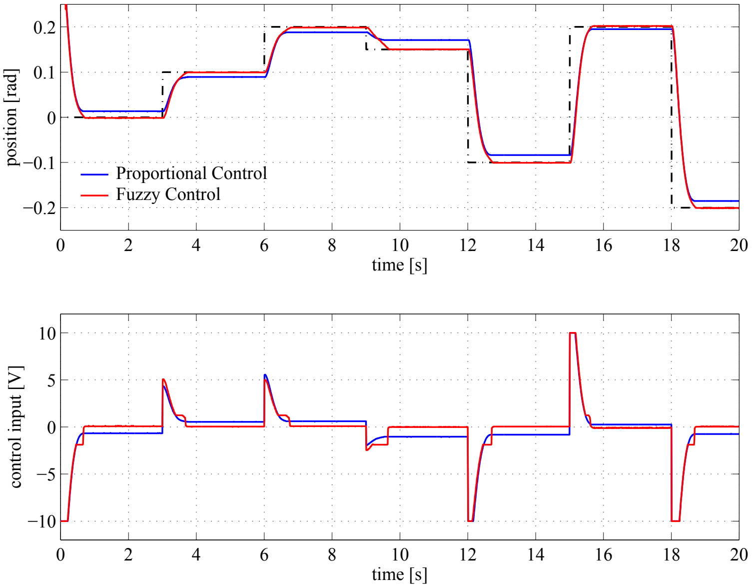
Another Solution: Sliding Mode Control



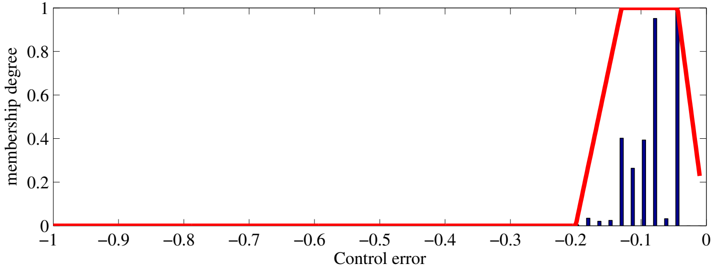
Experimental Results

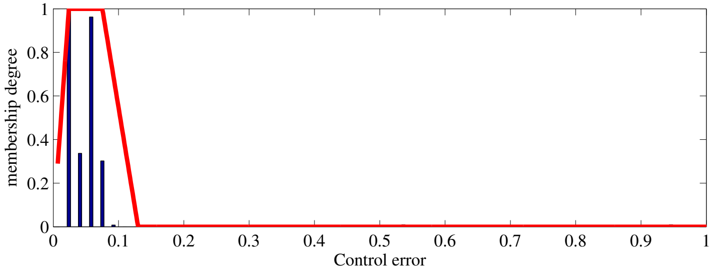


Experimental Results



Membership Functions





Fuzzy PD Controller: Rule Table

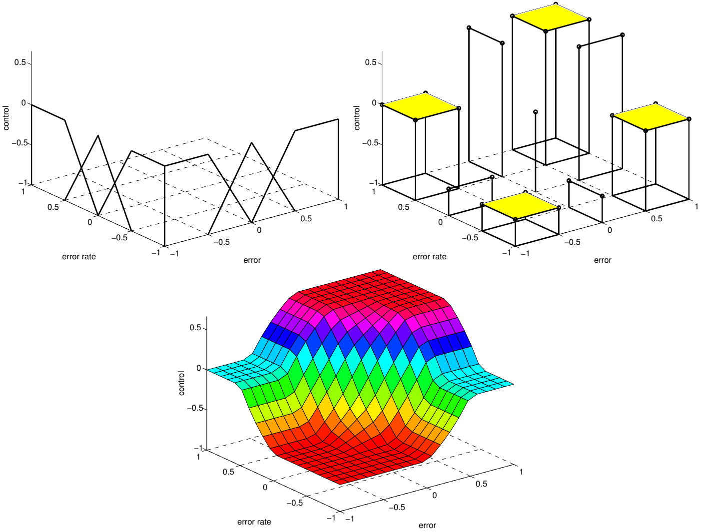
error rate

error

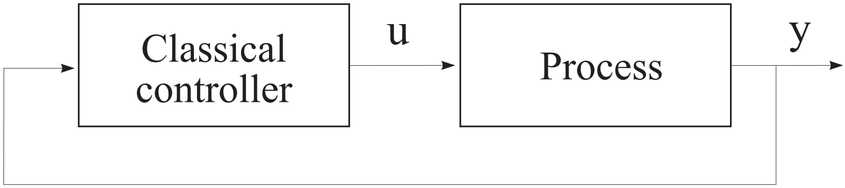
	NB	ZE	PB
NB	NB	NB	ZE
ZE	NB	ZE	PB
PB	ZE	PB	PB

R_{12}: If *error* is NB and *error rate* is ZE then *control* is NB

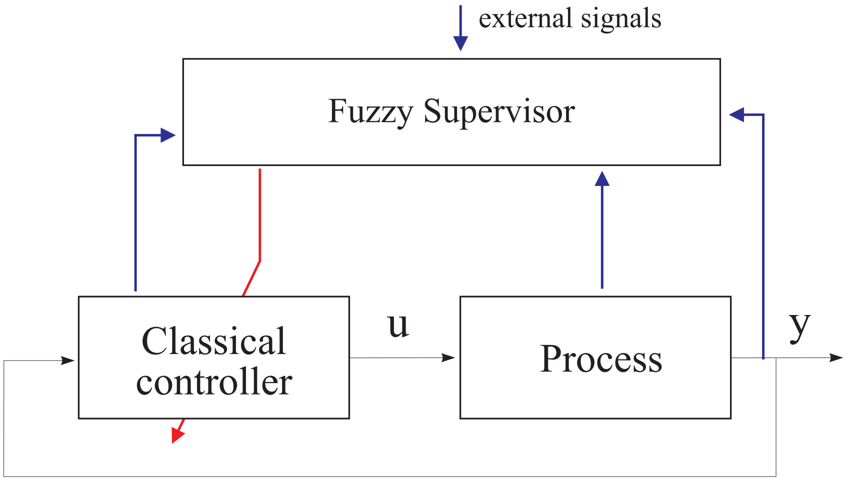
Fuzzy PD Controller – cont'd



Supervisory Fuzzy Control



Supervisory Fuzzy Control

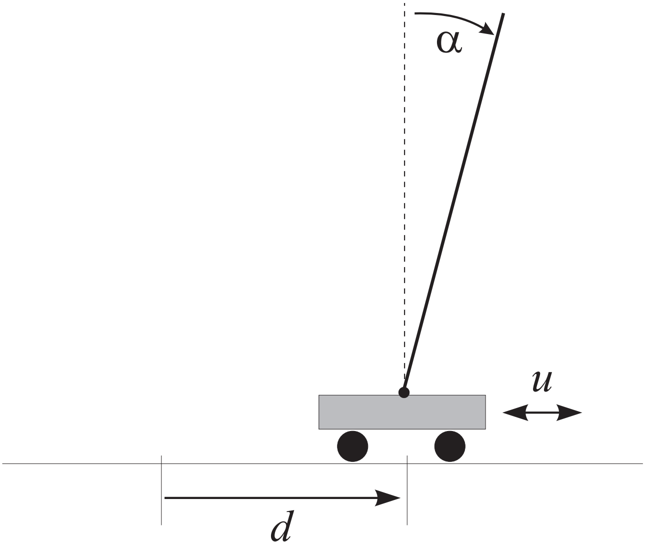


Supervisory Control Rules: Example

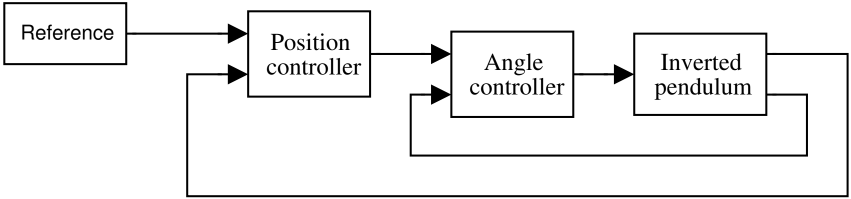
If	process output is *High*
then	reduce proportional gain *Slightly* **and**
	increase derivative gain *Moderately*.

(Supervised PD controller)

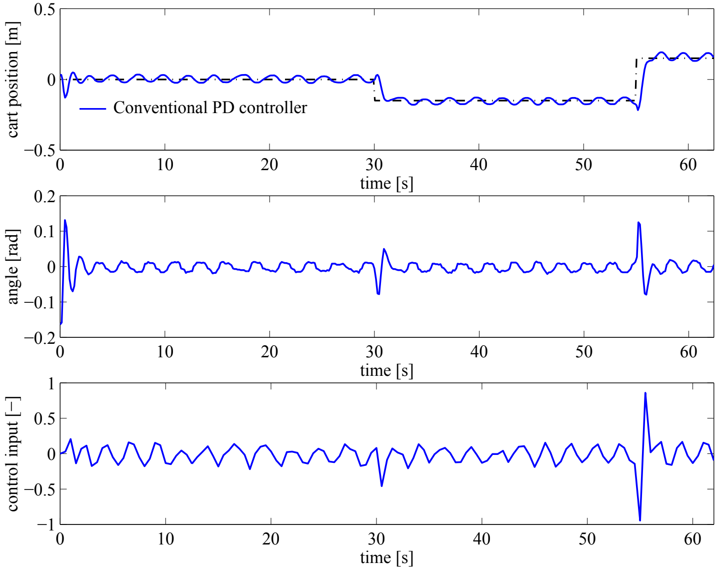
Example: Inverted Pendulum



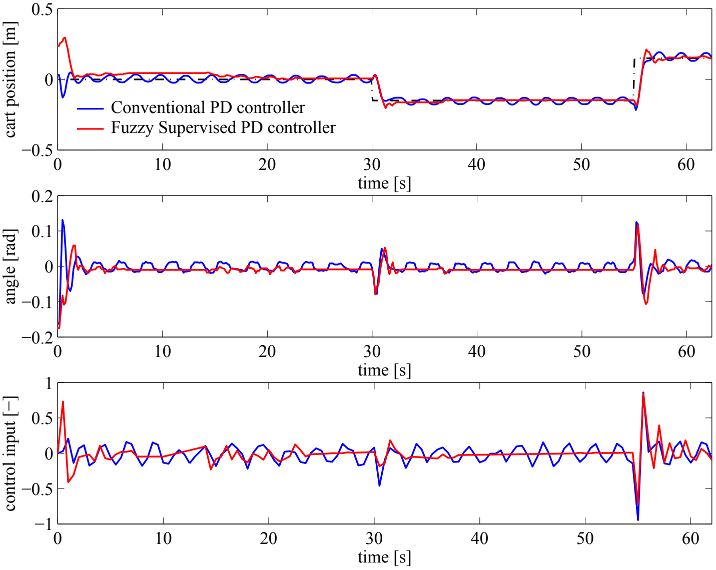
Cascade Control Scheme



Experimental Results



Experimental Results



Takagi–Sugeno Control

Takagi–Sugeno PD controller:

$$R_1 \quad : \quad \textbf{If } r \textbf{ is } \text{Low} \textbf{ then } u_{\mathrm{L}} = P_{\mathrm{L}} e + D_{\mathrm{L}} \dot{e}$$

$$R_2 \quad : \quad \textbf{If } r \textbf{ is } \text{High} \textbf{ then } u_{\mathrm{H}} = P_{\mathrm{H}} e + D_{\mathrm{H}} \dot{e}$$

$$u = \frac{\mu_{\mathrm{L}}(r)\, u_{\mathrm{L}} + \mu_{\mathrm{H}}(r)\, u_{\mathrm{H}}}{\mu_{\mathrm{L}}(r) + \mu_{\mathrm{H}}(r)} = \gamma_{\mathrm{L}}(r)\, u_{\mathrm{L}} + \gamma_{\mathrm{H}}(r)\, u_{\mathrm{H}}$$

Takagi–Sugeno Control

Takagi–Sugeno PD controller:

$$
\begin{aligned}
R_1 &: \quad \textbf{If } r \textbf{ is } \text{Low} \textbf{ then } u_\text{L} = P_\text{L} e + D_\text{L} \dot{e} \\
R_2 &: \quad \textbf{If } r \textbf{ is } \text{High} \textbf{ then } u_\text{H} = P_\text{H} e + D_\text{H} \dot{e}
\end{aligned}
$$

$$
\begin{aligned}
u &= \frac{\mu_\text{L}(r)\, u_\text{L} + \mu_\text{H}(r)\, u_\text{H}}{\mu_\text{L}(r) + \mu_\text{H}(r)} = \gamma_\text{L}(r)\, u_\text{L} + \gamma_\text{H}(r)\, u_\text{H} \\
&= \{\gamma_\text{L}(r) P_\text{L} + \gamma_\text{H}(r) P_\text{H}\}\, e + \{\gamma_\text{L}(r) D_\text{L} + \gamma_\text{H}(r) D_\text{H}\}\, \dot{e}
\end{aligned}
$$

Takagi–Sugeno Control

Takagi–Sugeno PD controller:

$$R_1 \quad : \quad \textbf{If } r \textbf{ is } \text{Low} \textbf{ then } u_L = P_L e + D_L \dot{e}$$
$$R_2 \quad : \quad \textbf{If } r \textbf{ is } \text{High} \textbf{ then } u_H = P_H e + D_H \dot{e}$$

$$u = \frac{\mu_L(r)\, u_L + \mu_H(r)\, u_H}{\mu_L(r) + \mu_H(r)} = \gamma_L(r)\, u_L + \gamma_H(r)\, u_H$$

$$= \{\gamma_L(r) P_L + \gamma_H(r) P_H\}\, e + \{\gamma_L(r) D_L + \gamma_H(r) D_H\}\, \dot{e}$$

$$= P(r) e + D(r) \dot{e},$$

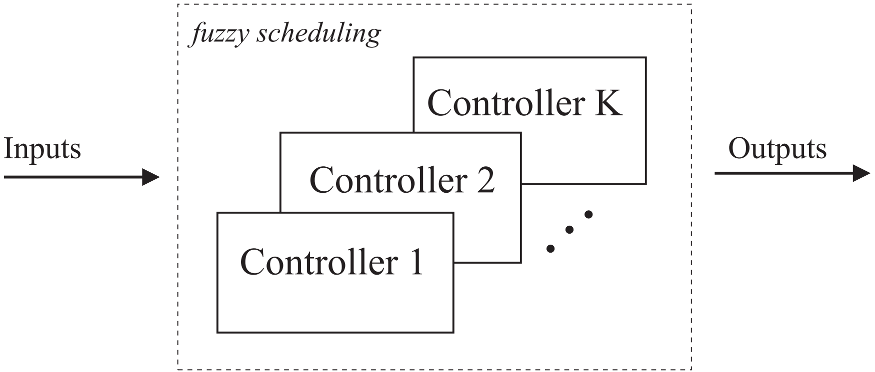
Takagi–Sugeno Control

Takagi–Sugeno PD controller:

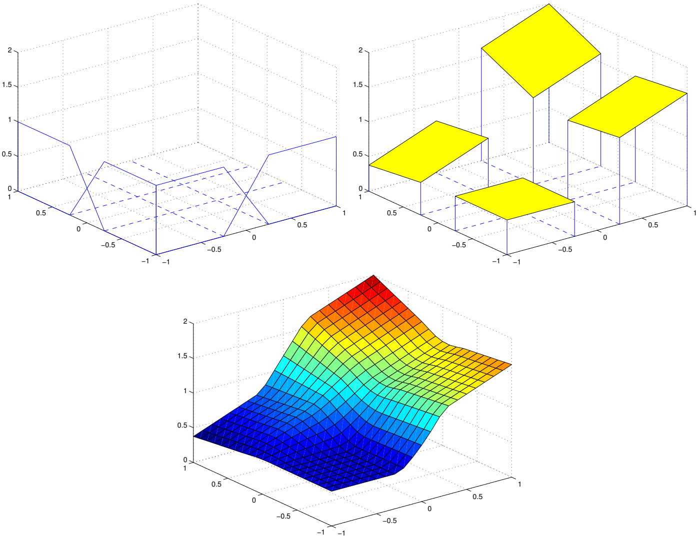
$$
\begin{aligned}
R_1 &: \quad \textbf{If } r \textbf{ is } \text{Low} \textbf{ then } u_L = P_L e + D_L \dot{e} \\
R_2 &: \quad \textbf{If } r \textbf{ is } \text{High} \textbf{ then } u_H = P_H e + D_H \dot{e}
\end{aligned}
$$

$$
\begin{aligned}
u &= \frac{\mu_L(r)\, u_L + \mu_H(r)\, u_H}{\mu_L(r) + \mu_H(r)} = \gamma_L(r)\, u_L + \gamma_H(r)\, u_H \\
&= \{\gamma_L(r) P_L + \gamma_H(r) P_H\}\, e + \{\gamma_L(r) D_L + \gamma_H(r) D_H\}\, \dot{e} \\
&= P(r)e + D(r)\dot{e}, \qquad P(r) \in \text{conv}(P_L, P_H), \ \ldots
\end{aligned}
$$

Takagi–Sugeno Control is Gain Scheduling



TS Control: Input–Output Mapping



TS Control: Example

1. Strongly nonlinear process (output-dependent gain).
2. Fuzzy supervisor to adjust the gain of a proportional controller.
3. Comparison with linear (fixed-gain) proportional control.

TS Control: Example

Nonlinear process:

$$\frac{d^3y(t)}{dt^3} + \frac{d^2y(t)}{dt^2} + \frac{dy(t)}{dt} = y^2(t)u(t)$$

Problems with linear control:

- stability and performance depend on process output
- re-tuning the controller does not help
- nonlinear control is the only solution

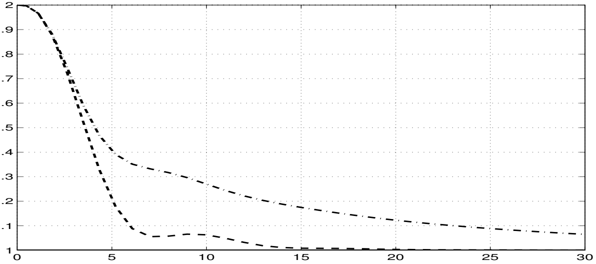
TS Control: Example

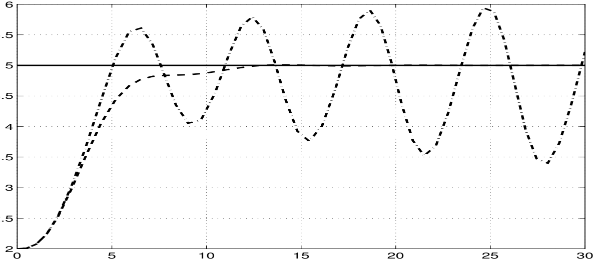
Goal: Design a controller to stabilize the process for a wide range of operating points ($y > 0$):

TS (proportional) control rules:

$$
\begin{array}{ll}
\textbf{If } y \text{ is Small} & \textbf{then } u(k) = P_{\text{Small}} \cdot e(k) \\
\textbf{If } y \text{ is Medium} & \textbf{then } u(k) = P_{\text{Medium}} \cdot e(k) \\
\textbf{If } y \text{ is Large} & \textbf{then } u(k) = P_{\text{Large}} \cdot e(k)
\end{array}
$$

Comparison of Performance





Typical Applications

- Tune parameters of low-level controllers (auto-tuning).
- Improve performance of classical control (response-assisted PID).
- Adaptation, gain scheduling (aircraft control).

Typical Applications

- Tune parameters of low-level controllers (auto-tuning).
- Improve performance of classical control (response-assisted PID).
- Adaptation, gain scheduling (aircraft control).

+ Enhancement of classical controllers.
+ Interface between low-level and high-level control.
– Ad hoc approach, difficult analysis.

Fuzzy Control: Design Steps

control engineering approaches + heuristic knowledge

1. Determine inputs and outputs.
2. Define membership functions.
3. Design rule base.
4. Test (completeness, stability, performance).
5. Fine-tune the controller.

Parameters in a Fuzzy Controller