Lecture 2: Fuzzy Control II

Jens Kober  Robert Babuška

Knowledge-Based Control Systems (SC42050)
Cognitive Robotics
3mE, Delft University of Technology, The Netherlands
07-04-2018

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

If \( x \) is \( A_i \) 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

\[
y = f(x)
\]
Linear Mapping with a Singleton Model

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

- Triangular partition:

\[ \begin{array}{c}
A_1 \\
A_2 \\
A_3 \\
A_4 \\
A_5 \\
\end{array}
\]

- 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

\[ y = \frac{\sum_{i=1}^{K} \mu_{A_i}(x) y_i}{\sum_{j=1}^{K} \mu_{A_j}(x)} = \frac{\sum_{i=1}^{K} \mu_{A_i}(x)(a_i^T x + b_i)}{\sum_{j=1}^{K} \mu_{A_j}(x)} \]

\[ y = \left( \sum_{i=1}^{K} \gamma_i(x) a_i^T \right) x + \sum_{i=1}^{K} \gamma_i(x) b_i \]

linear in parameters \( a_i \) and \( b_i \), pseudo-linear in \( x \) (LPV)
TS Model is a Polytopic System

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

- **Data**
  - static
  - dynamic

- **Model**

- **User**
  - accuracy
  - interpretation
  - mathematical form
  - reliability (extrapolation)

- **Prior knowledge**
  - partial models
  - stability
  - gain, nonlinearity
  - settling time

Building Fuzzy Models

Knowledge-based approach:
- expert knowledge → 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

**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

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

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

\[ R_i: \text{If } x \text{ is } A_i \text{ then } y = b_i \]

- Given \( A_i \) and a set of input–output data:
  \[ \{(x_k, y_k) \mid k = 1, 2, \ldots, N}\]
- Estimate optimal consequent parameters \( b_i \).
Least-Squares Estimation of Singletons

1. Compute the membership degrees $\mu_{A_i}(x_k)$
2. Normalize
   $$\gamma_{ki} = \frac{\mu_{A_i}(x_k)}{\sum_{j=1}^{K} \mu_{A_j}(x_k)}$$
   (Output: $y_k = \sum_{i=1}^{K} \gamma_{ki} b_i$, in a matrix form: $y = \Gamma b$)
3. Least-squares estimate: $b = \left[\Gamma^T \Gamma\right]^{-1} \Gamma^T y$

Least-Square Estimation of TS Consequents

$$\begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix}, \quad \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad \Gamma_i = \begin{bmatrix} \gamma_{i1} & 0 & \cdots & 0 \\ 0 & \gamma_{i2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_{iN} \end{bmatrix}$$

$$\theta_i = \begin{bmatrix} a_i^T \\ b_i \end{bmatrix}^T, \quad X_e = \begin{bmatrix} X & 1 \end{bmatrix}$$

Antecedent Membership Functions

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

Fuzzy partition matrix:
\[ U = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{c1} & \mu_{c2} & \cdots & \mu_{cN} \end{bmatrix} \]

Cluster centers (means):
\[ V = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \]

Fuzzy Clustering Problem

Given the data:
\[ z_k = [z_{1k}, z_{2k}, \ldots, z_{nk}]^T \in \mathbb{R}^n, \quad k = 1, \ldots, N \]

Find:
the fuzzy partition matrix:
\[ U = \begin{bmatrix} \mu_{11} & \mu_{1k} & \cdots & \mu_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{c1} & \mu_{ck} & \cdots & \mu_{cN} \end{bmatrix} \]

and the cluster centers:
\[ V = \{v_1, v_2, \ldots, v_c\}, \quad v_i \in \mathbb{R}^n \]

Fuzzy Clustering: an Optimization Approach

Objective function (least-squares criterion):
\[ J(Z; V, U, A) = \sum_{i=1}^{c} \sum_{j=1}^{N} \mu_{ij}^m d_i^2(z_j, v_i) \]

subject to constraints:
\[ 0 \leq \mu_{ij} \leq 1, \quad i = 1, \ldots, c, \quad j = 1, \ldots, N \quad \text{membership degree} \]
\[ 0 < \sum_{j=1}^{N} \mu_{ij} < N, \quad i = 1, \ldots, c \quad \text{no cluster empty} \]
\[ \sum_{i=1}^{c} \mu_{ij} = 1, \quad j = 1, \ldots, N \quad \text{total membership} \]

Fuzzy c-Means Algorithm

Repeat:
1. Compute cluster prototypes (means): \[ v_i = \frac{\sum_{k=1}^{N} \mu_{ik}^m z_k}{\sum_{k=1}^{N} \mu_{ik}^m} \]
2. Calculate distances: \[ d_{ik} = (z_k - v_i)^T (z_k - v_i) \]
3. Update partition matrix: \[ \mu_{ik} = \frac{1}{\sum_{j=1}^{c} (d_{ik}/d_{ij})^{1/(m-1)}} \]
until \( \| \Delta U \| < \epsilon \)
Distance Measures

- Euclidean norm:
  \[ d^2(z_j, v_i) = (z_j - v_i)^T(z_j - v_i) \]
- Inner-product norm:
  \[ d^2_A(z_j, v_i) = (z_j - v_i)^T A_i(z_j - v_i) \]
- Many other possibilities …

Fuzzy Clustering – Demo

1. Fuzzy c-means

Extraction of Rules by Fuzzy Clustering

- Takagi–Sugeno model
  Rule-based description:
  If \( x \) is \( A \), then \( y = a x + b \).
  If \( x \) is \( A \), then \( y = a x + b \).
  etc...

- Inverse Takagi–Sugeno model
  Rule-based description:
  If \( y \) is \( B \), then \( x = a y + b \).
  If \( y \) is \( B \), then \( x = a y + b \).
  etc...

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

Example of Operator Knowledge

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

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

\[ u = P e \]

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

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

![DC Motor Model Diagram]

Proportional Controller

![Proportional Controller Diagram]

Linear Control

![Linear Control Graph]

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

<table>
<thead>
<tr>
<th>Error Level</th>
<th>Membership Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Big</td>
<td>Triangular</td>
</tr>
<tr>
<td>Zero</td>
<td>Triangular</td>
</tr>
<tr>
<td>Negative Big</td>
<td>Triangular</td>
</tr>
</tbody>
</table>

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;

Fuzzy Control

<table>
<thead>
<tr>
<th>Time [s]</th>
<th>Shaft Angle [rad]</th>
<th>Control Input [V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-0.15</td>
<td>0.05</td>
</tr>
<tr>
<td>5</td>
<td>-0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>10</td>
<td>-0.05</td>
<td>0.15</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>20</td>
<td>0.05</td>
<td>0.10</td>
</tr>
<tr>
<td>25</td>
<td>0.10</td>
<td>0.15</td>
</tr>
<tr>
<td>30</td>
<td>0.15</td>
<td>0.05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time [s]</th>
<th>Shaft Angle [rad]</th>
<th>Control Input [V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.15</td>
<td>0.05</td>
</tr>
<tr>
<td>5</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>10</td>
<td>0.05</td>
<td>0.15</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>20</td>
<td>0.05</td>
<td>0.10</td>
</tr>
<tr>
<td>25</td>
<td>0.10</td>
<td>0.15</td>
</tr>
<tr>
<td>30</td>
<td>0.15</td>
<td>0.05</td>
</tr>
</tbody>
</table>
Input–Output Mapping of the Controller

Another Solution: Sliding Mode Control

Experimental Results

Membership Functions
**Fuzzy PD Controller: Rule Table**

<table>
<thead>
<tr>
<th>error rate</th>
<th>NB</th>
<th>ZE</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>NB</td>
<td>NB</td>
<td>ZE</td>
</tr>
<tr>
<td></td>
<td>ZE</td>
<td>NB</td>
<td>PB</td>
</tr>
<tr>
<td></td>
<td>PB</td>
<td>ZE</td>
<td>PB</td>
</tr>
</tbody>
</table>

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

**Supervisory Fuzzy Control**

![Diagram of Fuzzy Supervisor and Classical Controller]

- Fuzzy Supervisor
- Classical controller
- Process
- u
- y
- External signals

**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

Reference

Position controller

Angle controller

Inverted pendulum

Experimental Results

Takagi–Sugeno Control

Takagi–Sugeno PD controller:

\[ R_1 : \text{If } r \text{ is Low then } u_L = P_L e + D_L \dot{e} \]

\[ R_2 : \text{If } r \text{ is High then } u_H = P_H e + D_H \dot{e} \]

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

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

\[ = P(0) e + D(0) \dot{e}, \quad P(0) \in \text{conv}(P_L, P_H), \ldots \]
Takagi–Sugeno Control is Gain Scheduling

TS Control: Input–Output Mapping

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

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**

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

**TS (proportional) control rules:**

- If \(y\) is Small then \(u(k) = P_{\text{Small}} \cdot e(k)\)
- If \(y\) is Medium then \(u(k) = P_{\text{Medium}} \cdot e(k)\)
- If \(y\) is Large then \(u(k) = P_{\text{Large}} \cdot e(k)\)

**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