

# Fuzzy Logic: Principles and Applications

## ISS0023 Intelligent Control Systems

Sergei Astapov

Laboratory for Proactive Technologies  
Department of Computer Control  
Tallinn University of Technology, Estonia

# Self Introduction

- Engineer at the Laboratory for Proactive Technologies (ProLab)
- PhD student at the Department of Computer Control
- Research topics
  - Band-limited signal analysis
  - Signal processing and data mining algorithms
  - Classification and decision-making algorithms
- Room: U02-305
- E-mail: [sergei.astapov@ttu.ee](mailto:sergei.astapov@ttu.ee)

# Lecture Overview

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion

# Outline

## 1. Concepts of Fuzzy Logic

### 1.1 Introduction

### 1.2 Fuzzy Sets

### 1.3 Fuzzy Set Operations

### 1.4 Membership Functions

### 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

### 2.1 Introduction

### 2.2 Mamdani FIS

### 2.3 Mamdani Inference

### 2.4 Defuzzification

### 2.5 Mamdani FIS Editor

### 2.6 Takagi-Sugeno FIS

### 2.7 Sugeno FIS Editor

## 3. Applications

### 3.1 Fuzzy PID Controller

### 3.2 Fuzzy %Anything%

### 3.3 Adaptive FIS

### 3.4 Adaptive Control

### 3.5 Clustering and Classification

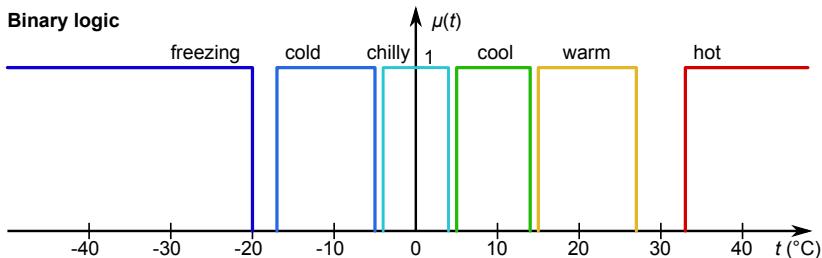
### 3.6 Discussion

# Why Go Fuzzy?

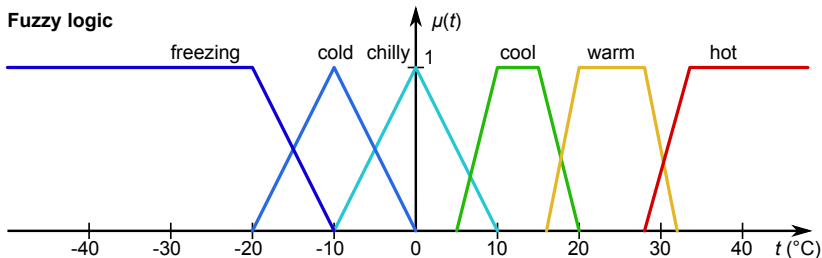
- Fuzzy logic models human expertise and knowledge in some task or application
- Consider conventional binary logic
  - Variables may take values of TRUE or FALSE (0 or 1)
- Try then to answer a simple question with binary logic
  - What do you consider warm temperature?
- How to answer?
  - You could try to give a value or interval of “warm” temperature
  - But then when does the temperature become cold or hot?

# The Fuzzy Way of Thinking

Binary logic



Fuzzy logic



# The Concept of Fuzzy Logic

- Fuzzy logic variables have a range of truthfulness from 0 to 1
- Fuzzy logic operates with **linguistic variables**, like “temperature” instead of  $t(^{\circ}\text{C})$
- Each variable has a specific number of **linguistic values**, like “hot” or “cold”
- Fuzzy inference is performed using **linguistic rules**, e.g.

IF temperature is cold THEN dress warm

- The linguistic values and their truth degree are quantified using **membership functions** (MF)

# A Little Bit of History

- 1964 : Lotfi A. Zadeh, UC Berkeley, introduced the paper on fuzzy sets
  - Idea of grade of membership
  - Imperfection and noise in the real world
  - Sharp criticism from academic community
- 1965–1975 : Zadeh continued to broaden the foundation of fuzzy set theory
  - Fuzzy multistage decision-making
  - Fuzzy similarity relations
  - Fuzzy restrictions, linguistic hedges
- 1970s : Research was mainly centered in Japan
- 1974 : E. H. Mamdani, UK, developed the first fuzzy logic controller
- 1977 : Dubois applied fuzzy sets in a comprehensive study of traffic conditions
- 1976–1987 : Industrial application of fuzzy logic in Japan and Europe
- 1987–Present : Widespread application



# Outline

## 1. Concepts of Fuzzy Logic

### 1.1 Introduction

### 1.2 Fuzzy Sets

### 1.3 Fuzzy Set Operations

### 1.4 Membership Functions

### 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

### 2.1 Introduction

### 2.2 Mamdani FIS

### 2.3 Mamdani Inference

### 2.4 Defuzzification

### 2.5 Mamdani FIS Editor

### 2.6 Takagi-Sugeno FIS

### 2.7 Sugeno FIS Editor

## 3. Applications

### 3.1 Fuzzy PID Controller

### 3.2 Fuzzy %Anything%

### 3.3 Adaptive FIS

### 3.4 Adaptive Control

### 3.5 Clustering and Classification

### 3.6 Discussion

# Conventional and Fuzzy Sets

Let  $X$  be a space of objects and  $x$  be a generic element of  $X$ . A classical set  $A$ ,  $A \subseteq X$ , is defined as a collection of elements  $x \in X$ , such that each element  $x$  can either belong or not belong to the set  $A$ .

The classical set thus can be characterized as  $A = \{x \mid x \in X\}$ .

By defining a **characteristic function** for each  $x$ , we can represent the classical set  $A$  by a set of ordered pairs  $(x, 0)$  or  $(x, 1)$ , which indicate  $x \notin A$  or  $x \in A$  respectively.

In a fuzzy set the characteristic function is allowed to have values of membership between 0 and 1.

# Fuzzy Set Definition

## Definition 1 (Fuzzy set)

If  $X$  is a collection of objects  $x$ , then a **fuzzy set**  $A$  in  $X$  is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) \mid x \in X\}, \quad (1)$$

where  $\mu_A(x)$  is called the **membership function** (MF) for the fuzzy set  $A$ .

In fuzzy set theory classical sets are referred to as **crisp** sets and the values as crisp values.

$X$  is usually referred to as the **universe of discourse**. It represents the range of values the fuzzy variables may take.

Universes of discourse may be either discrete or continuous.

# Linguistic Variables and Values

- Fuzzy sets usually carry names appealing in our daily linguistic usage
- The universe is called a **linguistic variable** and its sets are called **linguistic values**
- The universe of discourse  $X$  is partitioned into several fuzzy sets, with MFs covering  $X$  in a more or less uniform manner

## Example 1

Consider the universe  $X$  of linguistic variable “temperature”. The universe may be defined differently, depending on the application. We may set it from the lowest to the highest temperature a typical human being can live in, e.g.  $[-50, 50]$  °C.

We partition the universe into 6 fuzzy sets: “freezing”, “cold”, “chilly”, “cool”, “warm”, “hot”. These sets are characterized by MFs

$\mu_{freezing}(x)$ ,  $\mu_{cold}(x)$ ,  $\mu_{chilly}(x)$ ,  $\mu_{cool}(x)$ ,  $\mu_{warm}(x)$ ,  $\mu_{hot}(x)$ .

# Relevant Properties of Fuzzy Sets

## Definition 2 (Support)

The **support** of a fuzzy set  $A$  is the set of all points  $x \in X$ , such that  $\mu_A(x) > 0$ :

$$\text{support}(A) = \{x \mid \mu_A(x) > 0\}. \quad (2)$$

## Definition 3 (Core)

The **core** of a fuzzy set  $A$  is the set of all points  $x \in X$ , such that  $\mu_A(x) = 1$ :

$$\text{core}(A) = \{x \mid \mu_A(x) = 1\}. \quad (3)$$

## Definition 4 (Crossover points)

A **crossover point** of a fuzzy set  $A$  is a point  $x \in X$ , at which  $\mu_A(x) = 0.5$ :

$$\text{crossover}(A) = \{x \mid \mu_A(x) = 0.5\}. \quad (4)$$

# Relevant Properties of Fuzzy Sets Continued

## Definition 5 (Normality)

A fuzzy set  $A$  is **normal** if its core is nonempty, i.e. we can always find a point  $x \in X$ , such that  $\mu_A(x) = 1$ .

## Definition 6 (Fuzzy singleton)

A fuzzy set, the support of which is a single point in  $X$  with  $\mu_A(x) = 1$  is called a **fuzzy singleton**.

## Definition 7 (Symmetry)

A fuzzy set  $A$  is **symmetric** if its MF is symmetric around a certain point  $x = c$ , namely,  $\mu_A(c + x) = \mu_A(c - x)$ ,  $\forall x \in X$ .

# Relevant Properties of Fuzzy Sets Continued

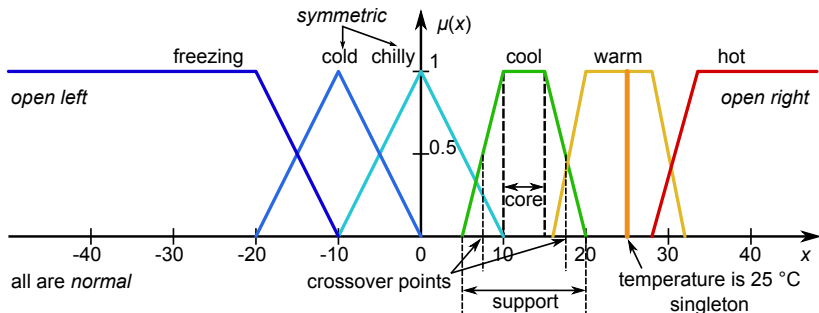
## Definition 8 (Open left, open right, closed sets)

A fuzzy set  $A$  is:

**open left** if  $\lim_{x \rightarrow -\infty} \mu_A(x) = 1, \lim_{x \rightarrow +\infty} \mu_A(x) = 0$ ;

**open right** if  $\lim_{x \rightarrow -\infty} \mu_A(x) = 0, \lim_{x \rightarrow +\infty} \mu_A(x) = 1$ ;

and **closed** if  $\lim_{x \rightarrow -\infty} \mu_A(x) = \lim_{x \rightarrow +\infty} \mu_A(x) = 0$ .



# Outline

## 1. Concepts of Fuzzy Logic

### 1.1 Introduction

### 1.2 Fuzzy Sets

### 1.3 Fuzzy Set Operations

### 1.4 Membership Functions

### 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

### 2.1 Introduction

### 2.2 Mamdani FIS

### 2.3 Mamdani Inference

### 2.4 Defuzzification

### 2.5 Mamdani FIS Editor

### 2.6 Takagi-Sugeno FIS

### 2.7 Sugeno FIS Editor

## 3. Applications

### 3.1 Fuzzy PID Controller

### 3.2 Fuzzy %Anything%

### 3.3 Adaptive FIS

### 3.4 Adaptive Control

### 3.5 Clustering and Classification

### 3.6 Discussion

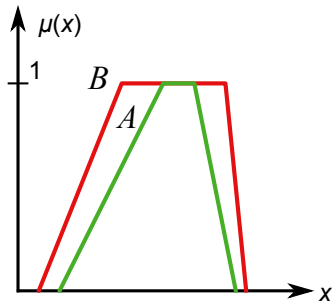


# Containment

## Definition 9 (Containment or subset)

Fuzzy set  $A$  is **contained** in fuzzy set  $B$  (or  $A$  is a **subset** of  $B$ ), iff  $\mu_A(x) \leq \mu_B(x)$  for all  $x$ :

$$A \subseteq B \iff \mu_A(x) \leq \mu_B(x). \quad (5)$$

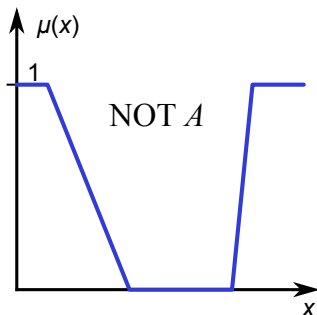
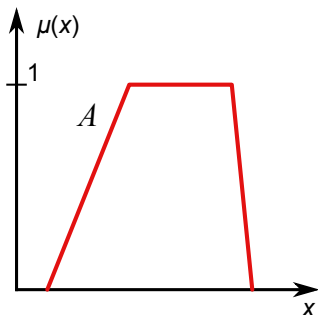


# Complement

## Definition 10 (Complement or negation)

The **complement** of a fuzzy set  $A$ , denoted by  $\overline{A}$  or  $\neg A$ , or NOT  $A$  is defined as

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x). \quad (6)$$

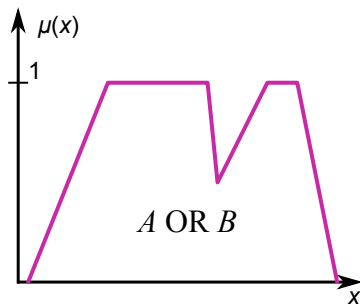
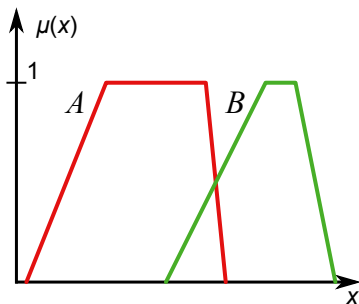


# Union

## Definition 11 (Union or disjunction)

The **union** of two fuzzy sets  $A$  and  $B$  is a fuzzy set  $C$ , written as  $C = A \cup B$  or  $C = A \text{ OR } B$ , the MF of which is related to those of  $A$  and  $B$  by

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x). \quad (7)$$

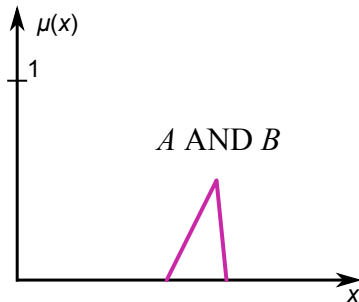
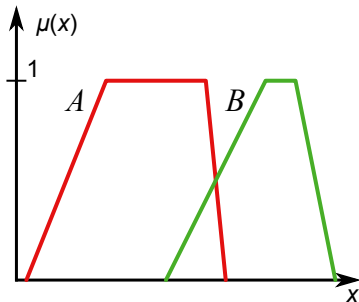


# Intersection

## Definition 12 (Intersection or conjunction)

The **intersection** of two fuzzy sets  $A$  and  $B$  is a fuzzy set  $C$ , written as  $C = A \cap B$  or  $C = A \text{ AND } B$ , the MF of which is related to those of  $A$  and  $B$  by

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x). \quad (8)$$



# Cartesian Product and Co-product

## Definition 13 (Cartesian product and co-product)

Let  $A$  and  $B$  be fuzzy sets in  $X$  and  $Y$ , respectively. The **Cartesian product** of  $A$  and  $B$ , denoted by  $A \times B$ , is a fuzzy set in the product space  $X \times Y$  with the membership function

$$\mu_{A \times B}(x, y) = \min(\mu_A(x), \mu_B(y)). \quad (9)$$

Similarly, the Cartesian co-product  $A + B$  is a fuzzy set with the membership function

$$\mu_{A+B}(x, y) = \max(\mu_A(x), \mu_B(y)). \quad (10)$$

# Outline

## 1. Concepts of Fuzzy Logic

1.1 Introduction

1.2 Fuzzy Sets

1.3 Fuzzy Set Operations

**1.4 Membership Functions**

1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

2.1 Introduction

2.2 Mamdani FIS

2.3 Mamdani Inference

2.4 Defuzzification

2.5 Mamdani FIS Editor

2.6 Takagi-Sugeno FIS

2.7 Sugeno FIS Editor

## 3. Applications

3.1 Fuzzy PID Controller

3.2 Fuzzy %Anything%

3.3 Adaptive FIS

3.4 Adaptive Control

3.5 Clustering and Classification

3.6 Discussion

# Preface

- A fuzzy set is completely characterized by its MF
- As the universe most often consists of real values,  $X \subseteq \mathbb{R}$ , it is convenient to define MFs as continuous functions
- For a single linguistic variable the MFs are one-dimensional
- Combining the universes of different linguistic variables, MFs of higher dimensions may be derived
- Here the most commonly applied MF types are presented

# Straight-Line MF: Triangular MF

## Definition 14 (Triangular MF)

A **triangular MF** is specified by three parameters  $\{a, b, c\}$  as follows:

$$\text{triangle}(x; a, b, c) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & c \leq x. \end{cases} \quad (11)$$

It may also be described by min and max as

$$\text{triangle}(x; a, b, c) = \max \left( \min \left( \frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right). \quad (12)$$

The parameters  $a$  and  $c$  locate the “feet” of the triangle and  $b$  — its peak.



# Straight-Line MF: Trapezoidal MF

## Definition 15 (Trapezoidal MF)

A **trapezoidal MF** is specified by four parameters  $\{a, b, c, d\}$  as follows:

$$\text{trapezoid}(x; a, b, c, d) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ 1, & b \leq x \leq c. \\ \frac{d-x}{d-c}, & c \leq x \leq d. \\ 0, & d \leq x. \end{cases} \quad (13)$$

An alternative expression using  $\min$  and  $\max$  is

$$\text{trapezoid}(x; a, b, c, d) = \max \left( \min \left( \frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right). \quad (14)$$

The parameters  $a$  and  $d$  locate the “feet” of the trapezoid and  $b$  and  $c$  — its “shoulders”.

# Smooth MF: Gaussian and Bell MF

## Definition 16 (Gaussian MF)

A **Gaussian MF** is specified by two parameters  $\{c, \sigma\}$  as follows:

$$\text{gaussian}(x; c, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}. \quad (15)$$

The parameter  $c$  represents the MF center and  $\sigma$  determines the MF width.

## Definition 17 (Generalized bell MF)

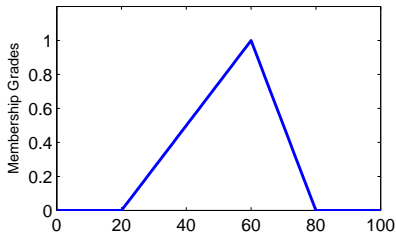
A **generalized bell MF** is specified by three parameters  $\{a, b, c\}$  as follows:

$$\text{bell}(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}}, \quad (16)$$

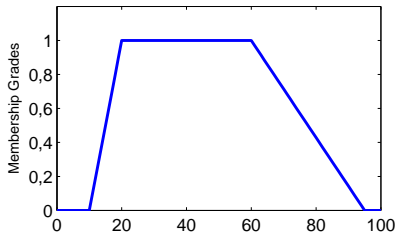
where  $b$  is usually positive (if  $b < 0$ , then the MF becomes an upside-down bell).

# MATLAB MF Examples

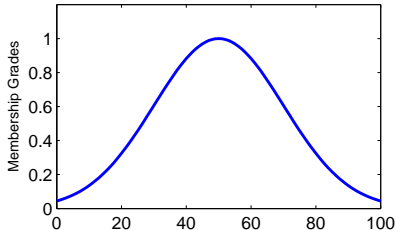
(a) Triangular MF: `trimf(x,[20,60,80])`



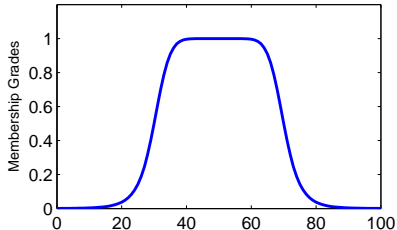
(b) Trapezoidal MF: `trapmf(x,[10,20,60,95])`



(c) Gaussian MF: `gaussmf(x,[20,50])`

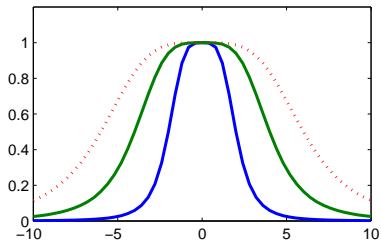


(d) Generalized Bell MF: `gbellmf(x,[20,4,50])`

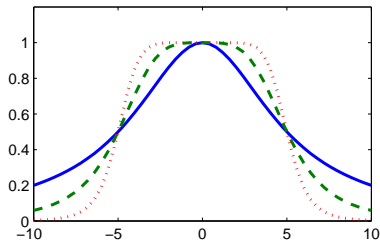


# Changing the Parameters of Bell MF

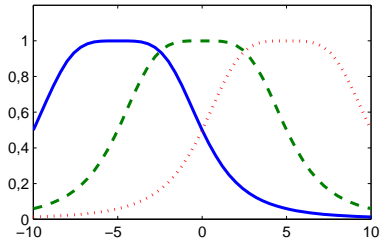
(a) Changing 'a'



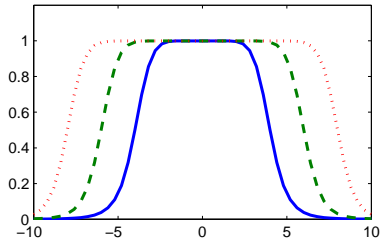
(b) Changing 'b'



(c) Changing 'c'



(d) Changing 'a' and 'b'



# Straight-line and Smooth MFs: Analysis

What are the advantages and drawbacks of straight-line and smooth MFs?

- Straight-line MFs
  - Simple formulas: computational efficiency
  - Zero points strictly defined:
    - Good, when boundary strictness is needed
    - Bad, when fuzzy sets cannot be adequately characterized by sudden drops to zero membership
  - Limitations due to linearity
  - Simple for manual tuning, unsuited for automated tuning
- Smooth MFs:
  - Non-linear: higher flexibility
  - Best for automated tuning (adaptive systems)
  - Less straight-forward: more problems during initial design

# Open Membership Functions

## Definition 18 (Sigmoidal MF)

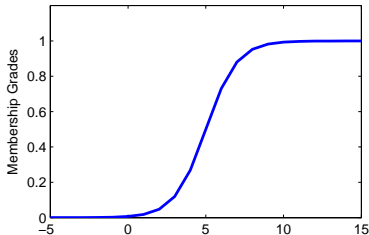
A **sigmoidal MF** is specified by two parameters  $\{a, c\}$  as follows:

$$\text{sig}(x; a, c) = \frac{1}{1 + e^{-a(x-c)}}, \quad (17)$$

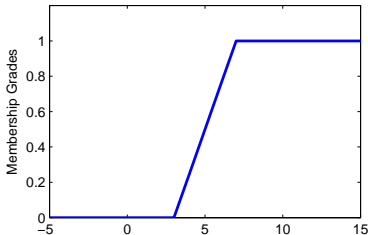
where  $a$  controls the slope of the crossover point  $c$ .

An open triangular MF is obtained by specifying  $\pm \text{inf}$  as a left or right “foot” parameter, e.g. `trimf(x, [3, 7, inf])`

(a) Sigmoidal MF: `sigmf(x, [1, 5])`



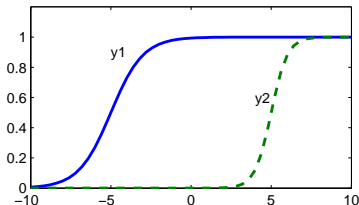
(b) Triangular MF: `trimf(x, [3, 7, inf])`



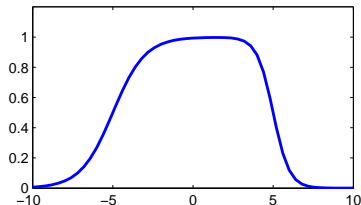
# Asymmetric Membership Functions

There are numerous ways to get asymmetric smooth MFs. One way is taking the difference  $|y_1 - y_2|$  and product  $y_1 y_2$  of sigmoid MFs:

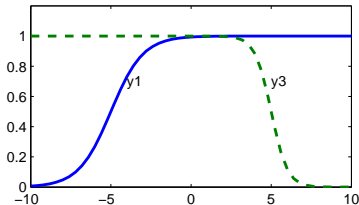
(a)  $y_1 = \text{sig}(x; 1, -5)$ ;  $y_2 = \text{sig}(x; 2, 5)$



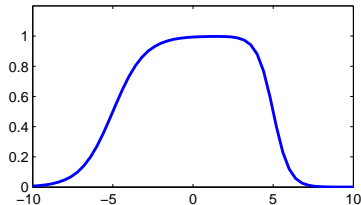
(b)  $|y_1 - y_2|$



(c)  $y_1 = \text{sig}(x; 1, -5)$ ;  $y_3 = \text{sig}(x; -2, 5)$



(d)  $y_1 y_3$



# Asymmetric MF: Left-Right MF

## Definition 19 (Left-right MF)

A **left-right MF** is specified by three parameters  $\{\alpha, \beta, c\}$  as

$$\text{LR}(x; \alpha, \beta, c) = \begin{cases} F_L\left(\frac{c-x}{\alpha}\right), & x \leq c, \\ F_R\left(\frac{x-c}{\beta}\right), & x \geq c, \end{cases} \quad (18)$$

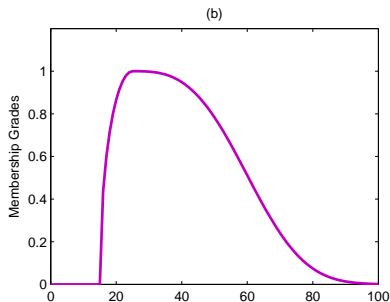
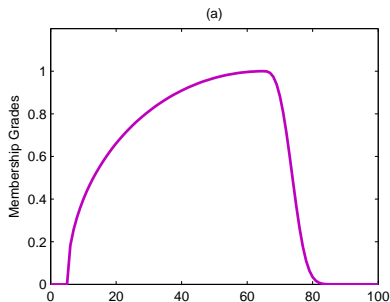
where  $F_L(x)$  and  $F_R(x)$  are monotonically decreasing functions defined on  $[0, \infty)$  with  $F_L(0) = F_R(0) = 1$  and  $\lim_{x \rightarrow \infty} F_L(x) = \lim_{x \rightarrow \infty} F_R(x) = 0$ .



# Asymmetric MF: Left-Right MF Example

## Example 2

Let  $F_L(x) = \sqrt{\max(0, 1 - x^2)}$ ,  $F_R = e^{-|x|^3}$ . Then applying (18) we can generate different curves, e.g. (a)  $\text{lr\_mf}(x, 60, 10, 65)$ ; and (b)  $\text{lr\_mf}(x, 10, 40, 25)$ ;



# Asymmetric MF: Two-Sided Gaussian MF

## Definition 20 (Two-sided Gaussian MF)

A **two-sided Gaussian MF** is defined by four parameters  $\{c_1, \sigma_1, c_2, \sigma_2\}$  as

$$\text{gaussian2}(x; c_1, \sigma_1, c_2, \sigma_2) = \begin{cases} \exp\left[-\frac{1}{2}\left(\frac{x-c_1}{\sigma_1}\right)^2\right], & x \leq c_1, \\ 1, & c_1 < x \leq c_2, \\ \exp\left[-\frac{1}{2}\left(\frac{x-c_2}{\sigma_2}\right)^2\right], & c_2 \leq x, \end{cases} \quad (19)$$

where  $c_1, \sigma_1$  are the parameters of the left-most curve and  $c_2, \sigma_2$  are the parameters of the right-most curve.

The two-sided Gaussian is essentially a mixture of two Gaussian functions defined by (15). It is computed in MATLAB using the `gauss2mf` function.

# Remarks on Membership Functions

- The presented MFs are only the most common ones
- For a full glossary of available MFs refer to the MATLAB Fuzzy Toolbox manual and other sources
- Be creative! Nobody forbids you from inventing your own MFs
- Non-normality and other properties of MFs can be achieved by mathematical manipulations on existing MFs or by defining one's own MFs
- Two-dimensional MFs are not discussed here, for further study please refer to literature

# Outline

## 1. Concepts of Fuzzy Logic

### 1.1 Introduction

### 1.2 Fuzzy Sets

### 1.3 Fuzzy Set Operations

### 1.4 Membership Functions

### 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

### 2.1 Introduction

### 2.2 Mamdani FIS

### 2.3 Mamdani Inference

### 2.4 Defuzzification

### 2.5 Mamdani FIS Editor

### 2.6 Takagi-Sugeno FIS

### 2.7 Sugeno FIS Editor

## 3. Applications

### 3.1 Fuzzy PID Controller

### 3.2 Fuzzy %Anything%

### 3.3 Adaptive FIS

### 3.4 Adaptive Control

### 3.5 Clustering and Classification

### 3.6 Discussion

# Fuzzy IF-THEN Rules

## Definition 21 (Fuzzy if-then rule)

A **fuzzy if-then rule**, also known as a **fuzzy rule**, **fuzzy implication**, or **fuzzy conditional statement**, assumes the form

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B, \quad (20)$$

where  $A$  and  $B$  are linguistic values defined by fuzzy sets on universes of discourse  $X$  and  $Y$ , respectively. The expression  $x$  is  $A$  is called the **antecedent** or **premise**, while  $y$  is  $B$  is called the **consequence** or **conclusion**.

Expression (20), which is abbreviated as  $A \rightarrow B$ , can be defined as a binary fuzzy relation  $R$  on the product space  $X \times Y$ :  $R = A \rightarrow B$ .  $R$  can be viewed as a fuzzy set of two-dimensional MF

$$\mu_R(x, y) = f(\mu_A(x), \mu_B(y)),$$

where the function  $f$  is called the fuzzy implication function, that transforms the membership degrees of  $x$  in  $A$  and  $y$  in  $B$  into those of  $(x, y)$  in  $A \rightarrow B$ .

# Multiple Input Multiple Output Rules

Let premise linguistic variables  $x_i$ ,  $i = 1, 2, \dots, n$  and consequence linguistic variables  $y_j$ ,  $j = 1, 2, \dots, m$  take on values of their universes of discourse  $X_i$  and  $Y_j$ , respectively. Let  $x_i$  be characterized by a set of linguistic values

$$A_i = \{A_i^k : k = 1, 2, \dots, N_i\},$$

and  $y_j$  be characterized by a set of linguistic values

$$B_j = \{B_j^l : l = 1, 2, \dots, M_j\}.$$

Then a MIMO rule with number of inputs  $n$  and number of outputs  $m$  can be written as

$$\begin{array}{l} \text{IF } x_1 \text{ is } A_1^p \text{ AND } x_2 \text{ is } A_2^q \text{ AND } \dots \text{ AND } x_n \text{ is } A_n^r \\ \text{THEN } y_1 \text{ is } B_1^s \text{ AND } y_2 \text{ is } B_2^u \text{ AND } \dots \text{ AND } y_m \text{ is } B_m^v. \end{array} \quad (21)$$

# Linguistic Operators

- A large number of operators may be applied to linguistic terms in fuzzy rules
  - Negation, e.g. “*not* warm”
  - Connectives: *and*, *or*, *either*, *neither*, etc.
  - Hedges: *too*, *very*, *more or less*, *quite*, *extremely*, etc.  
For example “*more or less* warm *but not too* warm”
- Here only *not*, *and*, *or* operators are discussed as they are most common and sufficient in the majority of applications
- In practice we will use only the *and* operator

# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion

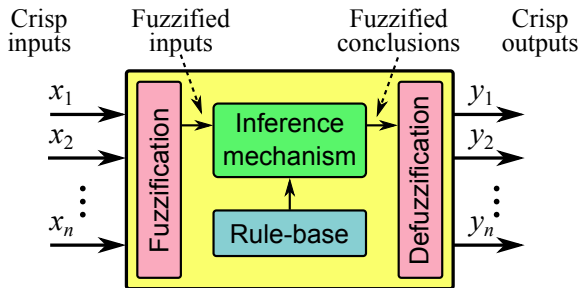


# Fuzzy Inference Systems

A **fuzzy inference system** (FIS) or, as it is also known in different application areas, **fuzzy expert system**, **fuzzy model**, **fuzzy associative memory** and **fuzzy logic controller** (FLC), is a computing framework based on the concepts of fuzzy theory, fuzzy if-then rules and fuzzy reasoning.

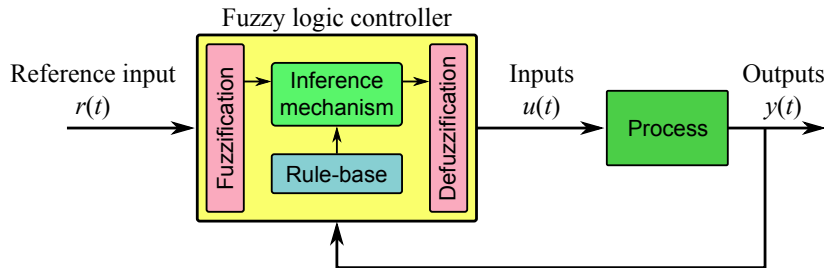
- FIS have many application areas
  - Automatic control and robotics
  - Classification and clustering
  - Pattern recognition
  - Decision analysis and expert systems

# Generic FIS Structure



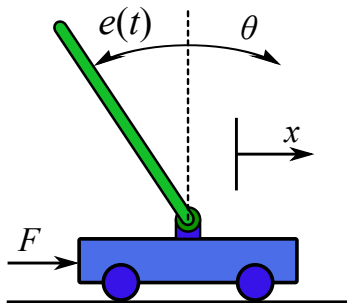
- Fuzzification: transformation of crisp values to fuzzy sets
- Rule-base: contains a selection of fuzzy rules
- Inference mechanism: performs a certain inference procedure upon the rules and derives a conclusion
- Defuzzification: transformation of output fuzzy sets to crisp values

# What Will Be Discussed



- We investigate two most common FIS types:
  - Mamdani and Takagi-Sugeno fuzzy models
- An example of a fuzzy control system is provided along the course of investigation

# Controlled Process: Inverted Pendulum

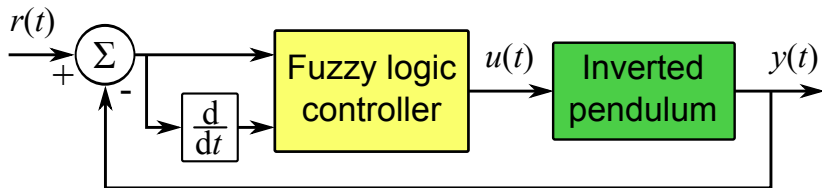


$r(t)$  — reference  $\theta$  angle

$u(t)$  — force (N)

$y(t)$  —  $\theta$  angle (rad)

$e(t) = r(t) - y(t)$



# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion

# Mamdani Type FIS

- Was proposed as the first attempt to control a steam engine and boiler combination by a set of linguistic control rules obtained from experienced human operators
- The most straight-forward cognitive approach to transferring knowledge into fuzzy models
- Design steps
  - Choose controller inputs and outputs (linguistic variables)
  - Assign linguistic values to every variable
  - Derive control rules for every possible scenario
  - Choose proper MF for every linguistic value
  - Specify the parameters of the inference mechanism
  - Test, observe behavior, tune

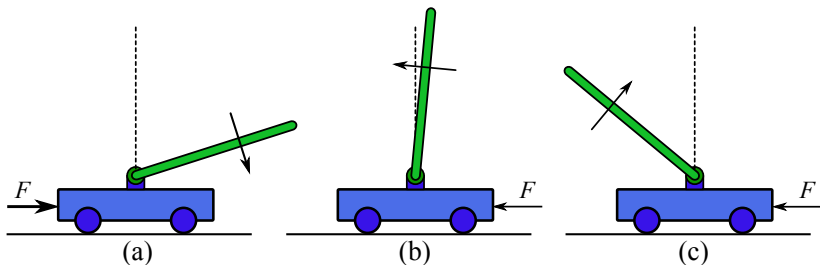
# FIS Linguistic Variables and Values

- For our inverted pendulum example we choose the following inputs and outputs:
  - “error” describes  $e(t) = r(t) - y(t)$
  - “change-in-error” describes  $\frac{d}{dt}e(t)$
  - “force” describes  $u(t)$
- The linguistic variables take on the following values:
  - “negative large” or “neglarge”, represented by “-2”
  - “negative small” or “negsmall”, represented by “-1”
  - “zero”, represented by “0”
  - “positive small” or “possmall”, represented by “1”
  - “positive large” or “poslarge”, represented by “2”

# Fuzzy Rules

Recall the general MIMO rule structure (15). Substituting mathematical characters with our assigned linguistic labels and values, we get rules of the following structure:

- (a) IF error is neglarge AND change-in-error is neglarge THEN force is poslarge
- (b) IF error is zero AND change-in-error is possmall THEN force is negsmall
- (c) IF error is poslarge AND change-in-error is negsmall THEN force is negsmall





# Fuzzy Rule-Base

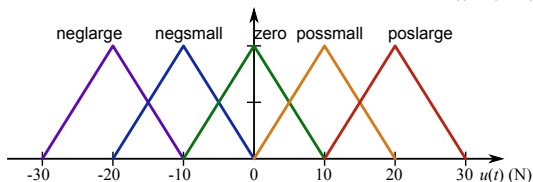
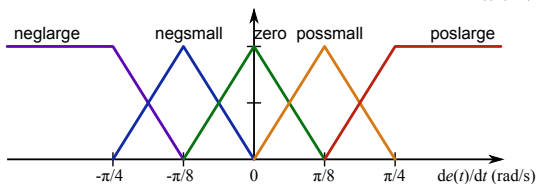
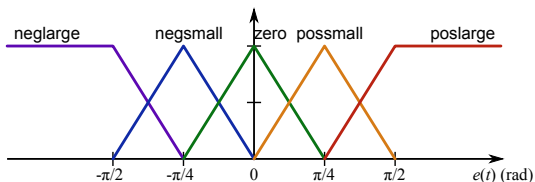
The number of rules for a MISO FIS is at most  $\prod_{i=1}^n N_i$ , where  $N_i$  is the number of linguistic values for the  $i$ -th linguistic premise variable. (All possible combinations of premise linguistic values.)

In our case the number of rules is equal to  $5 \cdot 5 = 25$ .

Continuing the logic of the previous three rule cases, we can derive the rule-base, presented as a table.

force		change-in-error				
		-2	-1	0	1	2
error	-2	2	2	2	1	0
	-1	2	2	1	0	-1
	0	2	1	0	-1	-2
	1	1	0	-1	-2	-2
	2	0	-1	-2	-2	-2

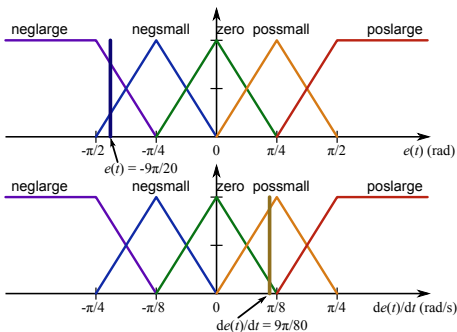
# Membership Functions



# Fuzzification

Singleton fuzzification: apply a fuzzy singleton  $\mu_{A_i}^{\text{fuz}}(x)$  to the premise variable universe, perform intersection.

This method is applied when measurement noise is not accounted for — the crisp input values are certain. In “Gaussian fuzzification” a Gaussian is used as a fuzzification function, which accounts for inconsistency in the input signal.



Crisp input  $e(t) = -9\pi/20$ :

$$\mu_{\widehat{\text{neglarge}}}(e) = \min(\mu_{\text{neglarge}}(e), \mu_1^{\text{fuz}}(e)) = \min(0.75, 1) = 0.75;$$

$$\mu_{\widehat{\text{negsmall}}}(e) = \min(\mu_{\text{negsmall}}(e), \mu_1^{\text{fuz}}(e)) = \min(0.25, 1) = 0.25; \text{ all other zero.}$$

Crisp input  $\dot{e}(t) = 9\pi/80$ :

$$\mu_{\widehat{\text{zero}}}(\dot{e}) = \min(\mu_{\text{zero}}(\dot{e}), \mu_2^{\text{fuz}}(\dot{e})) = \min(0.125, 1) = 0.125;$$

$$\mu_{\widehat{\text{possmall}}}(\dot{e}) = \min(\mu_{\text{possmall}}(\dot{e}), \mu_2^{\text{fuz}}(\dot{e})) = \min(0.875, 1) = 0.875; \text{ all other zero.}$$

# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference**
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion

# Mamdani Inference Mechanism Steps

- Calculate the **firing strength** for each rule in the rule-base
- Determine which **rules are on** using the firing strengths
- Determine **implied fuzzy sets** — perform **fuzzy implication**
- Determine **overall implied fuzzy set** — perform **fuzzy aggregation\***

\*Performed in case of applying specific types of defuzzification.  
If defuzzification uses implied fuzzy sets, the step is not performed.

# Firing Strength of a Premise

The firing strength of a rule is the degree of certainty that the rule premise holds for the given inputs. Its calculation depends on the linguistic operators used in the structure of a premise.

For any linguistic variables  $x_1$  and  $x_2$  the typical operators are the following:

Fuzzy complement (NOT):

- Defined in (6) as  $\mu_{\hat{A}_1^k}^-(x_1) = 1 - \mu_{\hat{A}_1^k}(x_1)$

Fuzzy union (OR):

- Defined in (7) as maximum  $\mu_{\hat{A}_1^k \cup \hat{A}_2^l}(x_1, x_2) = \max(\mu_{\hat{A}_1^k}(x_1), \mu_{\hat{A}_2^l}(x_2))$
- Alternative: algebraic sum  
$$\mu_{\hat{A}_1^k \cup \hat{A}_2^l}(x_1, x_2) = \mu_{\hat{A}_1^k}(x_1) + \mu_{\hat{A}_2^l}(x_2) - \mu_{\hat{A}_1^k}(x_1) \mu_{\hat{A}_2^l}(x_2)$$

Fuzzy intersection (AND):

- Defined in (8) as minimum  $\mu_{\hat{A}_1^k \cap \hat{A}_2^l}(x_1, x_2) = \min(\mu_{\hat{A}_1^k}(x_1), \mu_{\hat{A}_2^l}(x_2))$
- Alternative: algebraic product  $\mu_{\hat{A}_1^k \cap \hat{A}_2^l}(x_1, x_2) = \mu_{\hat{A}_1^k}(x_1) \mu_{\hat{A}_2^l}(x_2)$

# Firing Strength: More Complex Premises

In premises with more complex logic, the firing strength is calculated by partitioning the premise into simpler terms.

## Example 3

The premise

IF  $x_1$  is  $\hat{A}_1^2$  AND  $x_2$  is  $\hat{A}_2^1$  AND  $x_3$  is NOT  $\hat{A}_3^5$  OR  $x_4$  is  $\hat{A}_4^3$   
yields the firing strength

$$\mu_{\text{premise}}(x_1, x_2, x_3, x_4) = \max \left[ \min \left( \mu_{\hat{A}_1^2}(x_1), \mu_{\hat{A}_2^1}(x_2), 1 - \mu_{\hat{A}_3^5}(x_3) \right), \mu_{\hat{A}_4^3}(x_4) \right].$$

Also there exists an option to use a “rule certainty” weight. This way, for the  $i$ -th rule, the firing strength is multiplied by the weight  $w_i$ , which specifies how certain we are in this specific rule compared to other rules.

Keep in mind that there are more alternatives to AND and OR operations, you can also specify your custom ones.

# Which Rules Are On

- The rule is considered being “on” if its premise is non-zero:  
 $\mu_{premise}(x_1, x_2, \dots, x_n) > 0$
- An optional step, that reduces the number of computations
- Alternatively, perform fuzzy implication over the whole rule-base, but you will be doing a large number of operations over zero values

## Example 4

Consider a FIS with 3 inputs and 10 MFs per input. The number of rules is then at most  $10^3 = 1000$ . With the universes partitioned by so many rules, the number of “on” rules at any given time will be quite small. If for example 10 rules are on, then mark those rules and perform later steps with 10 sets of parameters, instead of using the whole rule-base and performing 100 times more computations, mainly with zeros.



# Implied Fuzzy Sets: Fuzzy Implication

The implied fuzzy set of an output  $y_j$  for a rule  $i$ , which has a consequent  $B_j^k$ , and a premise degree of membership equal to  $\mu_{premise(i)}(x_1, x_2, \dots, x_n)$ , is characterized by

$$\mu_{\hat{B}_j^k}(y_j) = \min \left( \mu_{premise(i)}(x_1, x_2, \dots, x_n), \mu_{B_j^k}(y_j) \right).$$

Alternatively the algebraic product can be defined as the implication operation:

$$\mu_{\hat{B}_j^k}(y_j) = \mu_{premise(i)}(x_1, x_2, \dots, x_n) \mu_{B_j^k}(y_j).$$

An implied fuzzy set is computed for every rule that is “on”.

# Overall Implied Fuzzy Set: Fuzzy Aggregation

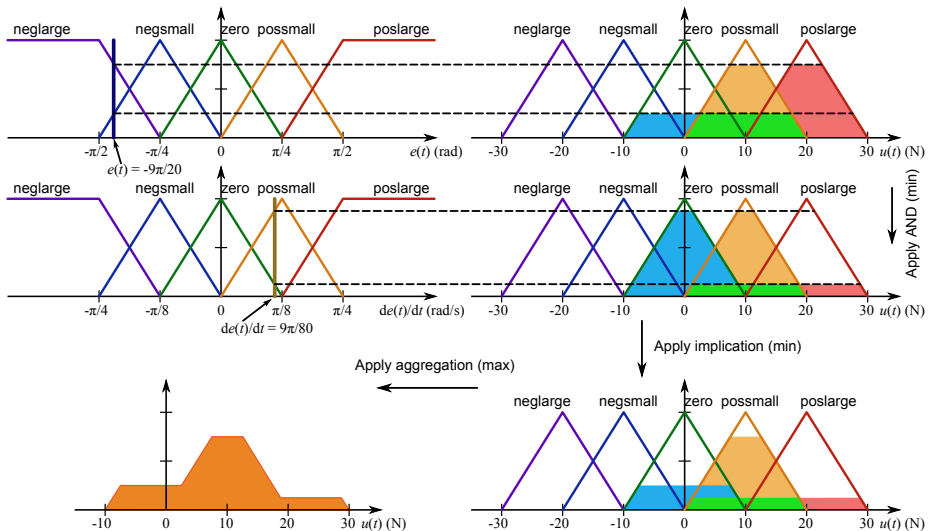
The overall implied fuzzy set  $\hat{B}_j$  of an output  $y_j$ , which incorporates the implied fuzzy sets  $\{\hat{B}_j^k, \hat{B}_j^l, \dots, \hat{B}_j^p\}$  is characterized by

$$\mu_{\hat{B}_j}(y_j) = \max \left( \mu_{\hat{B}_j^k}(y_j), \mu_{\hat{B}_j^l}(y_j), \dots, \mu_{\hat{B}_j^p}(y_j) \right).$$

Alternatively the algebraic sum can be defined as the aggregation operation:

$$\begin{aligned} \mu_{\hat{B}_j}(y_j) = & \mu_{\hat{B}_j^k}(y_j) + \mu_{\hat{B}_j^l}(y_j) + \dots + \mu_{\hat{B}_j^p}(y_j) - \\ & - \mu_{\hat{B}_j^k}(y_j) \mu_{\hat{B}_j^l}(y_j) \dots \mu_{\hat{B}_j^p}(y_j). \end{aligned}$$

# Mamdani Inference: Example



# Mamdani Inference: Example Computations

From the fuzzification stage we have established that we have four fuzzy values:

$$\mu_{\widehat{\text{neglarge}}}(e) = 0.75; \mu_{\widehat{\text{negsmall}}}(e) = 0.25; \mu_{\widehat{\text{zero}}}(\dot{e}) = 0.125; \\ \mu_{\widehat{\text{possmall}}}(\dot{e}) = 0.875. \text{ Thus the rules that are on are:}$$

IF error is neglarge AND change-in-error is zero THEN force is poslarge (red)

IF error is neglarge AND change-in-error is possmall THEN force is possmall (orange)

IF error is negsmall AND change-in-error is zero THEN force is possmall (green)

IF error is negsmall AND change-in-error is possmall THEN force is zero (blue)

Compute the firing strengths of the four rules using min for the AND operator:

$$\mu_{\text{premise}(1)}(e, \dot{e}) = \min(\mu_{\widehat{\text{neglarge}}}(e), \mu_{\widehat{\text{zero}}}(\dot{e})) = \min(0.75, 0.125) = 0.125;$$

$$\mu_{\text{premise}(2)}(e, \dot{e}) = \min(\mu_{\widehat{\text{neglarge}}}(e), \mu_{\widehat{\text{possmall}}}(\dot{e})) = \min(0.75, 0.875) = 0.75;$$

$$\mu_{\text{premise}(3)}(e, \dot{e}) = \min(\mu_{\widehat{\text{negsmall}}}(e), \mu_{\widehat{\text{zero}}}(\dot{e})) = \min(0.25, 0.125) = 0.125;$$

$$\mu_{\text{premise}(4)}(e, \dot{e}) = \min(\mu_{\widehat{\text{negsmall}}}(e), \mu_{\widehat{\text{possmall}}}(\dot{e})) = \min(0.25, 0.875) = 0.25.$$

# Mamdani Inference: Example Computations Continued

Implied fuzzy sets are derived from the rule premises using  $\min$  as:

$$\begin{aligned}\mu_{\widehat{poslarge}(1)}(u) &= \min(\mu_{premise(1)}(e, \dot{e}), \mu_{poslarge}(u)) = \\ &= \min(0.125, 1) = 0.125;\end{aligned}$$

$$\begin{aligned}\mu_{\widehat{possmall}(2)}(u) &= \min(\mu_{premise(2)}(e, \dot{e}), \mu_{possmall}(u)) = \\ &= \min(0.75, 1) = 0.75;\end{aligned}$$

$$\begin{aligned}\mu_{\widehat{possmall}(3)}(u) &= \min(\mu_{premise(3)}(e, \dot{e}), \mu_{possmall}(u)) = \\ &= \min(0.125, 1) = 0.125;\end{aligned}$$

$$\mu_{\widehat{zero}(4)}(u) = \min(\mu_{premise(4)}(e, \dot{e}), \mu_{zero}(u)) = \min(0.25, 1) = 0.25.$$

The overall implied fuzzy set is obtained by fuzzy aggregation using  $\max$  as:

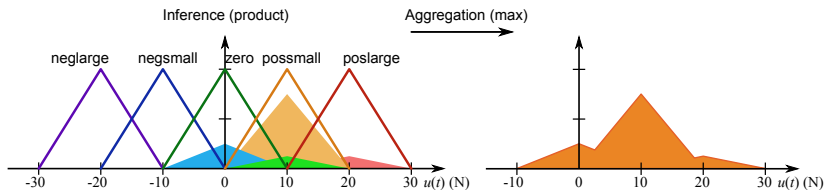
$$\begin{aligned}\mu_{\widehat{overall}}(u) &= \\ \max &\left( \mu_{\widehat{poslarge}(1)}(u), \mu_{\widehat{possmall}(2)}(u), \mu_{\widehat{possmall}(3)}(u), \mu_{\widehat{zero}(4)}(u) \right).\end{aligned}$$

# Mamdani Inference: Principle Justification

The choice of linguistic operator functions, fuzzy inference and fuzzy aggregation operations is based on the assertions that:

*We can be no more certain in our premises than we are certain in our data.*

*We can be no more certain in our conclusions than we are certain in our premises.*



# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification**
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion

# Defuzzification

The result of fuzzy inference is the implied fuzzy set (or sets). For the systems, where a crisp value is required from the FIS, the operation called defuzzification is applied to the implied sets.

A number of defuzzification strategies exist, and it is not hard to invent more, suiting your specific application.

Each provides a means to choose a crisp output  $y_j^{\text{crisp}}$  based on either the implied fuzzy sets or the overall implied fuzzy set.

Reviewed defuzzification methods:

- Center of gravity (COG)
- Center-average
- Maximum criterion: mean of maximum (MOM), smallest of maximum (SOM), largest of maximum (LOM)
- Center of area (COA)



# Defuzzification on IFS: Center of Gravity

## Definition 22 (Center of gravity)

In **Center of gravity** (COG) defuzzification the output  $y_j^{\text{crisp}}$  is computed using the center of area and area of each implied fuzzy set:

$$y_j^{\text{crisp}} = \frac{\sum_{i=1}^R b_i^j \int_{Y_j} \mu_{\hat{B}_j^i}(y_j) dy_j}{\sum_{i=1}^R \int_{Y_j} \mu_{\hat{B}_j^i}(y_j) dy_j}, \quad (22)$$

where  $R$  is the number of rules,  $b_i^j$  is the center of area of the MF of  $B_j^p$  associated with the implied fuzzy set  $\hat{B}_j^i$  for the  $i$ -th rule and

$$\int_{Y_j} \mu_{\hat{B}_j^i}(y_j) dy_j$$

denotes the area under  $\mu_{\hat{B}_j^i}(y_j)$ .

# Defuzzification on IFS: Center of Gravity

The COG is easy to compute if you have simple areas under implied fuzzy set MFs, e.g. triangles with tops chopped off while using triangle MFs and `min` for implication.

Notice though, that for this method to be reliable the fuzzy system must be defined such that

$$\sum_{i=1}^R \int_{Y_j} \mu_{\hat{B}_j^i}(y_j) dy_j \neq 0$$

for all  $x_i$ . This is achieved if for every possible combination of inputs the consequent fuzzy sets all have nonzero area.

Also areas must be computable, thus we cannot use **open** MFs for output fuzzy sets.

# Defuzzification on IFS: Center-Average

## Definition 23 (Center-average)

In **Center-average** defuzzification the output  $y_j^{\text{crisp}}$  is computed using the centers of each of the output MFs and the maximum certainty of each of the implied fuzzy sets:

$$y_j^{\text{crisp}} = \frac{\sum_{i=1}^R b_i^j \sup_{y_j} \left\{ \mu_{\hat{B}_j^i}(y_j) \right\}}{\sum_{i=1}^R \sup_{y_j} \left\{ \mu_{\hat{B}_j^i}(y_j) \right\}}, \quad (23)$$

where  $\sup_{y_j}$  denotes the supremum (i.e. the least upper bound) of the implied fuzzy set  $\mu_{\hat{B}_j^i}(y_j)$ .

# Defuzzification on IFS: Center-Average

The center-average is easy to compute if implied fuzzy set MFs have a single maximum, e.g. reduced triangles while using triangle MFs and product for implication — in this case

$$\sup_{y_j} \left\{ \mu_{\hat{B}_j^i}(y_j) \right\} = \max \left( \mu_{\hat{B}_j^i}(y_j) \right).$$

Notice though, that the fuzzy system must be defined such that

$$\sum_{i=1}^R \sup_{y_j} \left\{ \mu_{\hat{B}_j^i}(y_j) \right\} \neq 0$$

for all  $x_i$ . This is achieved as in the case of COG.

# Defuzzification on IFS: Center-Average

If using **normal** MFs for output fuzzy sets, then for many inference strategies we have

$$\sup_{y_j} \left\{ \mu_{\hat{B}_j^i}(y_j) \right\} = \mu_{\text{premise}(i)}(x_1, x_2, \dots, x_n),$$

which is the firing strength of rule  $i$ . The formula for defuzzification is then given by

$$y_j^{\text{crisp}} = \frac{\sum_{i=1}^R b_i^j \mu_{\text{premise}(i)}(x_1, x_2, \dots, x_n)}{\sum_{i=1}^R \mu_{\text{premise}(i)}(x_1, x_2, \dots, x_n)}, \quad (24)$$

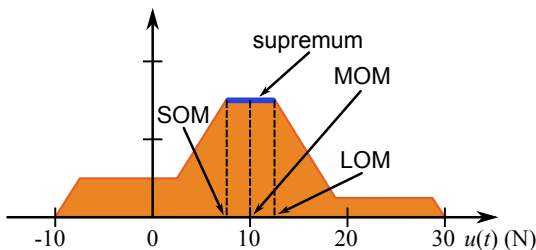
where  $\sum_{i=1}^R \mu_{\text{premise}(i)}(x_1, x_2, \dots, x_n) \neq 0, \forall x_i$  must be ensured. The shape of the output MFs does not matter, as bounds of supremum subsets can be defined using singletons.

# Defuzzification on The Overall IFS: Maximum Criterion

For the MOM, SOM and LOM defuzzification the crisp output is chosen as a point on the output universe  $Y_j$ , for which the overall implied fuzzy set  $\hat{B}_j$  reaches its maximum:

$$y_j^{\text{crisp}} \in \left\{ \arg \sup_{Y_j} \left\{ \mu_{\hat{B}_j}(y_j) \right\} \right\}.$$

MOM, SOM and LOM differ in the strategy of choosing the crisp value from this subset.



# Defuzzification on The Overall IFS: MOM

## Definition 24 (Mean of maximum)

Define a fuzzy set  $\hat{B}_j^* \subseteq Y_j$  with a MF defined as

$$\mu_{\hat{B}_j^*}(y_j) = \begin{cases} 1, & \mu_{\hat{B}_j}(y_j) = \sup_{Y_j} \{ \mu_{\hat{B}_j}(y_j) \}, \\ 0, & \text{otherwise.} \end{cases}$$

Then the crisp output of **mean of maximum (MOM)** defuzzification is defined as

$$y_j^{\text{crisp}} = \frac{\int_{Y_j} y_j \mu_{\hat{B}_j^*}(y_j) dy_j}{\int_{Y_j} \mu_{\hat{B}_j^*}(y_j) dy_j}, \quad (25)$$

where the fuzzy system must be defined so  $\int_{Y_j} \mu_{\hat{B}_j^*}(y_j) dy_j \neq 0, \forall x_i$ .

Notice that if  $\mu_{\hat{B}_j^*}(y_j) = 1$  lies in a single interval  $[y_j^{\text{left}}, y_j^{\text{right}}] \subseteq Y_j$ , then  $y_j^{\text{crisp}} = (y_j^{\text{left}} + y_j^{\text{right}}) / 2$ .

# Defuzzification on The Overall IFS: SOM and LOM

## Definition 25 (Smallest of maximum)

In **smallest of maximum** (SOM) defuzzification the output  $y_j^{\text{crisp}}$  is computed as the minimal argument of the output universe  $Y_j$ , for which the the overall implied fuzzy set  $\hat{B}_j$  reaches its maximum:

$$y_j^{\text{crisp}} = \min \left[ \arg \sup_{Y_j} \left\{ \mu_{\hat{B}_j} (y_j) \right\} \right]. \quad (26)$$

## Definition 26 (Largest of maximum)

In **largest of maximum** (LOM) defuzzification the output  $y_j^{\text{crisp}}$  is computed as the maximal argument of the output universe  $Y_j$ , for which the the overall implied fuzzy set  $\hat{B}_j$  reaches its maximum:

$$y_j^{\text{crisp}} = \max \left[ \arg \sup_{Y_j} \left\{ \mu_{\hat{B}_j} (y_j) \right\} \right]. \quad (27)$$



# Defuzzification on The Overall IFS: Center of Area

## Definition 27 (Center of area)

*In **center of area** (COA) defuzzification the output  $y_j^{\text{crisp}}$  is computed over the area of the MF of the overall implied fuzzy set  $\hat{B}_j$  as*

$$y_j^{\text{crisp}} = \frac{\int_{Y_j} y_j \mu_{\hat{B}_j}(y_j) dy_j}{\int_{Y_j} \mu_{\hat{B}_j}(y_j) dy_j}, \quad (28)$$

*where the fuzzy system must be defined so  $\int_{Y_j} \mu_{\hat{B}_j}(y_j) dy_j \neq 0, \forall x_i$ .*

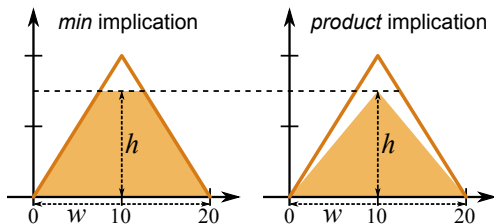
Computationally expensive: overlapping implied fuzzy sets may result in a overall implied fuzzy set with a sophisticated shape. Computing the area of such shapes in real-time is not an easy task.

# Defuzzification: Example

For our symmetrical triangular MFs the area and center of area of the implied fuzzy sets are easily calculated. If a symmetric triangle has a height 1 and base width  $w$  :

- The area of a triangle with the top “chopped off” at height  $h$  is equal to  $w \left( h - \frac{h^2}{2} \right)$
- The area of a triangle with height  $h$  is equal to  $\frac{1}{2}wh$

Here  $w$  is the support length of  $\hat{B}_j^i$  and  $h$  is  $\mu_{premise(i)}(x_1, x_2, \dots, x_n)$ .



# Defuzzification: COG Example

For implication defined by min:

$$\begin{aligned}
 u^{\text{crisp}} &= \frac{b_{pl} \int_U \mu_{\widehat{pl}(1)}(u) du + b_{ps} \int_U \mu_{\widehat{ps}(2)}(u) du + b_{ps} \int_U \mu_{\widehat{ps}(3)}(u) du + b_z \int_U \mu_{\widehat{z}(4)}(u) du}{\int_U \mu_{\widehat{pl}(1)}(u) du + \int_U \mu_{\widehat{ps}(2)}(u) du + \int_U \mu_{\widehat{ps}(3)}(u) du + \int_U \mu_{\widehat{z}(4)}(u) du} = \\
 &= \frac{(20)(1.1719) + (10)(4.6875) + (10)(1.1719) + (0)(2.1875)}{1.1719 + 4.6875 + 1.1719 + 2.1875} = \frac{82.032}{9.2188} = 8.90
 \end{aligned}$$

For implication defined by product:

$$\begin{aligned}
 u^{\text{crisp}} &= \frac{b_{pl} \int_U \mu_{\widehat{pl}(1)}(u) du + b_{ps} \int_U \mu_{\widehat{ps}(2)}(u) du + b_{ps} \int_U \mu_{\widehat{ps}(3)}(u) du + b_z \int_U \mu_{\widehat{z}(4)}(u) du}{\int_U \mu_{\widehat{pl}(1)}(u) du + \int_U \mu_{\widehat{ps}(2)}(u) du + \int_U \mu_{\widehat{ps}(3)}(u) du + \int_U \mu_{\widehat{z}(4)}(u) du} = \\
 &= \frac{(20)(0.625) + (10)(3.75) + (10)(0.625) + (0)(1.25)}{0.625 + 3.75 + 0.625 + 1.25} = \frac{56.25}{6.25} = 9.0
 \end{aligned}$$

# Defuzzification: Center-Average Example

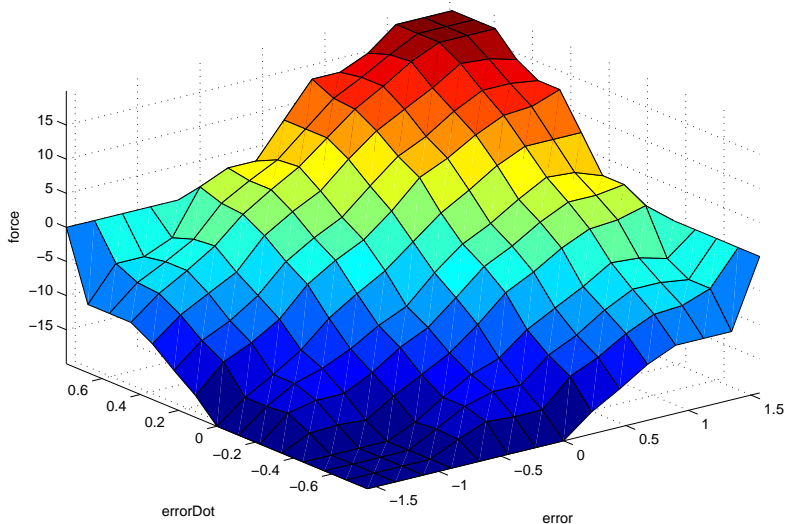
For implication defined by product:

$$\begin{aligned} u^{\text{crisp}} &= \\ &= \frac{b_{pl} \sup_u \{ \mu_{\widehat{pl}(1)}(u) \} + b_{ps} \sup_u \{ \mu_{\widehat{ps}(2)}(u) \} + b_{ps} \sup_u \{ \mu_{\widehat{ps}(3)}(u) \} + b_z \sup_u \{ \mu_{\widehat{z}(4)}(u) \}}{\sup_u \{ \mu_{\widehat{pl}(1)}(u) \} + \sup_u \{ \mu_{\widehat{ps}(2)}(u) \} + \sup_u \{ \mu_{\widehat{ps}(3)}(u) \} + \sup_u \{ \mu_{\widehat{z}(4)}(u) \}} \\ &= \frac{(20)(0.125) + (10)(0.75) + (10)(0.125) + (0)(0.25)}{0.125 + 0.75 + 0.125 + 0.25} = \frac{11.25}{1.25} = 9.0 \end{aligned}$$

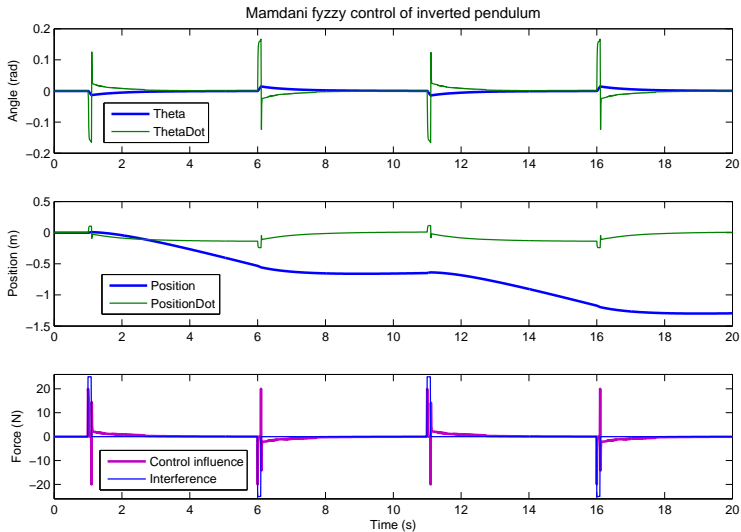
The supremum of a reduced triangular MF is its single peak which is equal to  $\mu_{\text{premise}(i)}(x_1, x_2, \dots, x_n)$ .

# FIS Input-Output Curve

Portrays the dependency of FIS output on its inputs. MATLAB command: `gensurf`



# Mamdani Fuzzy Control of Inverted Pendulum



# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

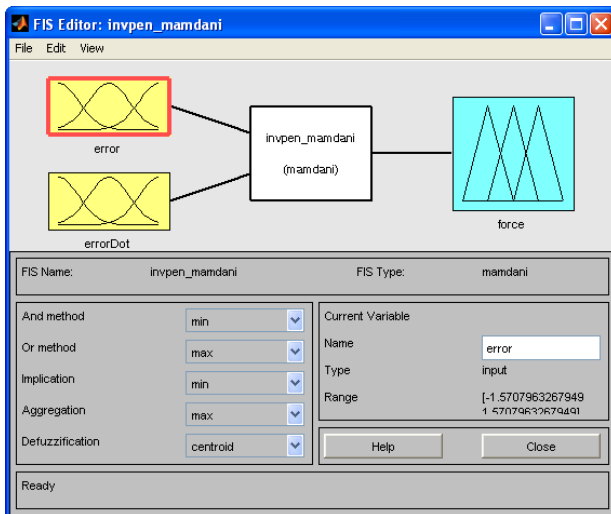
## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor**
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

## 3. Applications

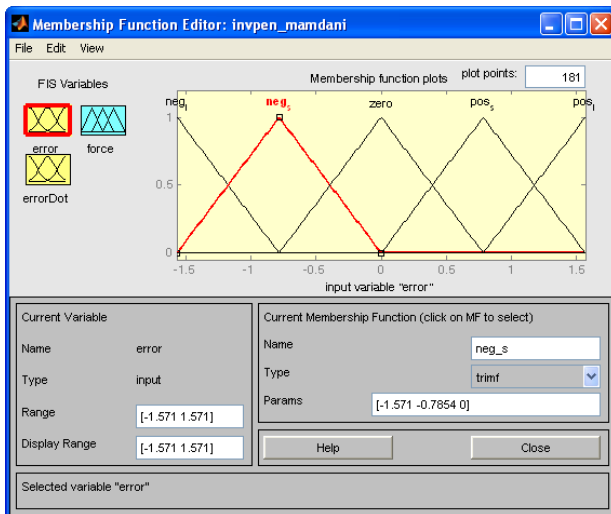
- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion

# FIS Design in MATLAB: Editor Main

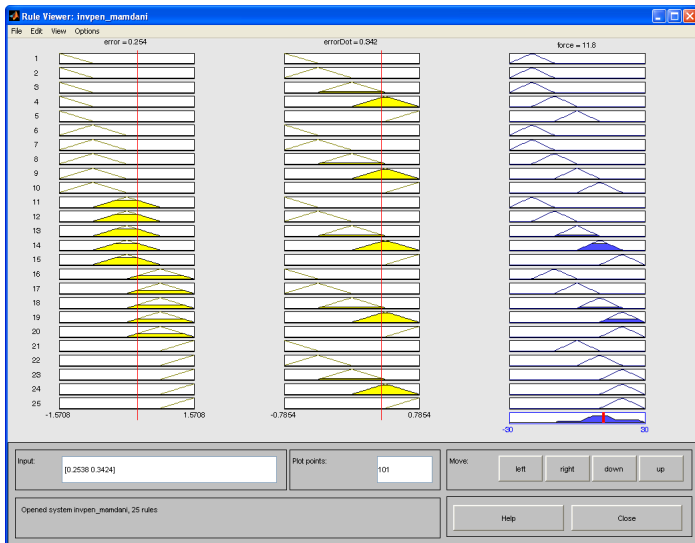




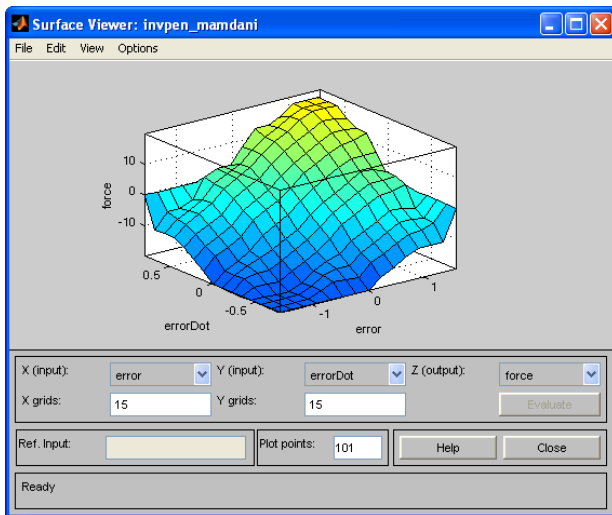
# FIS Design in MATLAB: MF Editor



# FIS Design in MATLAB: Rule Viewer



# FIS Design in MATLAB: Input-Output Curve



# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion

# Takagi-Sugeno FIS

Takagi-Sugeno or simply Sugeno-type FIS has a different way of computing the consequence and defuzzification.  
A general Sugeno rule has a form

IF  $x_1$  is  $A_1^k$  AND  $x_2$  is  $A_2^l$  AND ... AND  $x_n$  is  $A_n^p$  THEN  $z_i = f_i(\cdot)$ .

Here  $z = f(\cdot)$  may be any function (even another mapping, like neural network, or another FIS)

Usually  $z_i = f_i(x_1, x_2, \dots, x_n)$  is used. If this function is a first order polynomial, i.e.

$$z_i = a_n x_1 + a_{n-1} x_2 + \dots + a_1 x_n + a_0,$$

the inference system is called a **first-order** Sugeno FIS. When  $f$  is a constant, the system is called a **zero-order** Sugeno FIS.

# Sugeno Inference Principles

The premises  $\mu_{\text{premise}(i)}(x_1, x_2, \dots, x_n)$  are computed as in the Mamdani FIS, incorporating fuzzification and linguistic operators.

The defuzzification is usually performed using weighted average:

$$y^{\text{crisp}} = \frac{\sum_{i=1}^R z_i \mu_{\text{premise}(i)}(x_1, x_2, \dots, x_n)}{\sum_{i=1}^R \mu_{\text{premise}(i)}(x_1, x_2, \dots, x_n)}, \quad (29)$$

where the fuzzy system is defined so that

$$\sum_{i=1}^R \mu_{\text{premise}(i)}(x_1, x_2, \dots, x_n) \neq 0, \forall x_i.$$

Thus the Sugeno FIS can be used as a general mapper for a wide variety of applications.

# Sugeno FIS Example

We will not define the whole rule-base of the inverted pendulum controller for the Sugeno FIS. Lets specify  $z_i = f_i(e, \dot{e})$  for the rules that are on in our example:

$$z_1 = -5e + 4\dot{e} + 3 \quad \text{for the red rule;}$$

$$z_2 = -4e + 2\dot{e} + 2 \quad \text{for the orange rule;}$$

$$z_3 = -2e + 1\dot{e} + 1 \quad \text{for the green rule;}$$

$$z_4 = -0.5e + 0.5\dot{e} + 0 \quad \text{for the blue rule.}$$

For the values  $e = -\frac{9}{20}\pi$  and  $\dot{e} = \frac{9}{80}\pi$  the functions take on values

$$z_1 = -5 \left( -\frac{9}{20}\pi \right) + 4 \left( \frac{9}{80}\pi \right) + 3 = 11.482$$

$$z_2 = -4 \left( -\frac{9}{20}\pi \right) + 2 \left( \frac{9}{80}\pi \right) + 2 = 8.362$$

$$z_3 = -2 \left( -\frac{9}{20}\pi \right) + 1 \left( \frac{9}{80}\pi \right) + 1 = 4.181$$

$$z_4 = -0.5 \left( -\frac{9}{20}\pi \right) + 0.5 \left( \frac{9}{80}\pi \right) + 0 = 0.884$$

# Sugeno FIS Example

Then the FIS crisp output will be

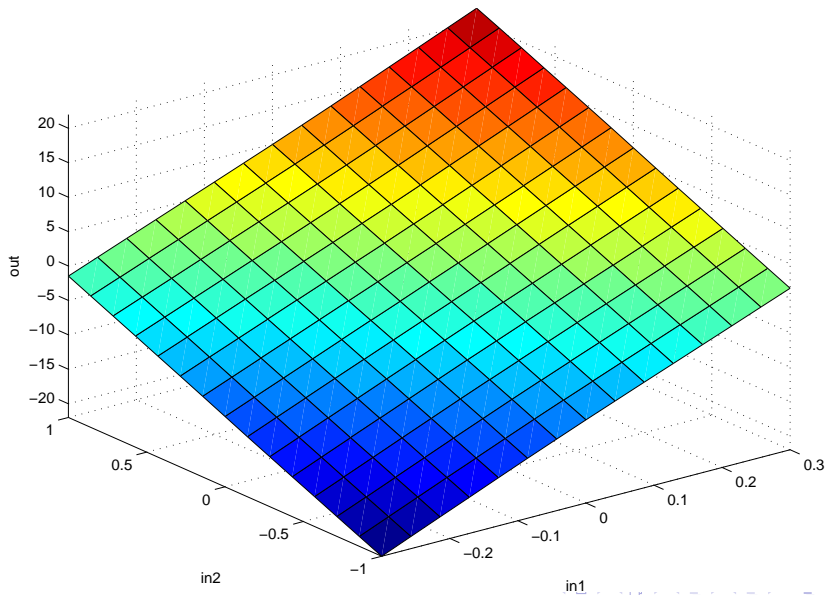
$$y^{\text{crisp}} = \frac{\sum_{i=1}^4 z_i \mu_{\text{premise}(i)}(e, \dot{e})}{\sum_{i=1}^4 \mu_{\text{premise}(i)}(e, \dot{e})} =$$

$$\frac{11.482 \cdot 0.125 + 8.362 \cdot 0.75 + 4.181 \cdot 0.125 + 0.884 \cdot 0.25}{0.125 + 0.75 + 0.125 + 0.25} = 9.03$$

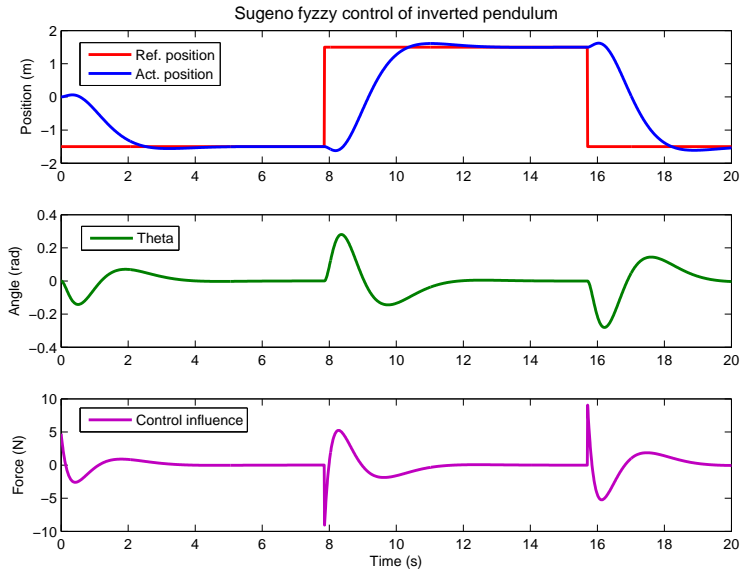
Notice, that no implication and aggregation is used. This simplifies the inference process a lot.



# Sugeno Input-Output Curve



# Sugeno Fuzzy Control of Inverted Pendulum



# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

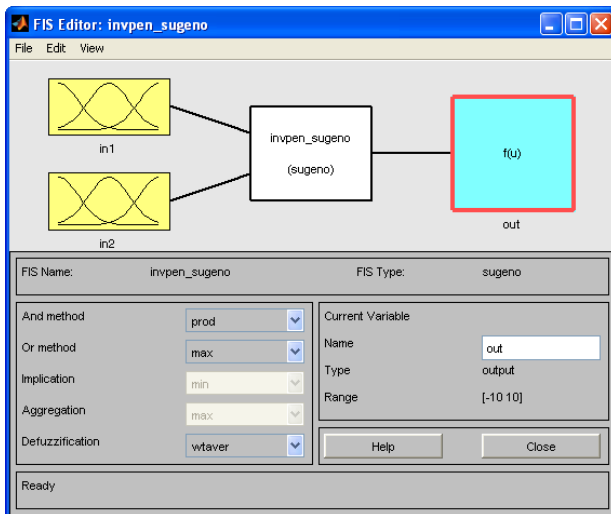
## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

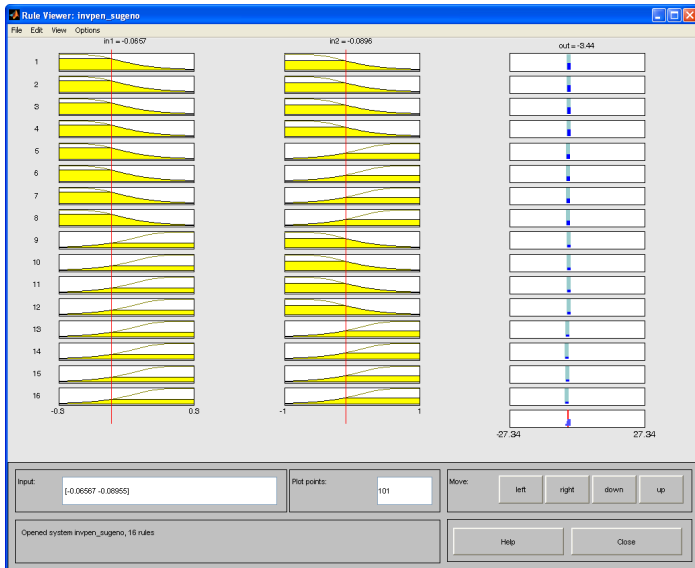
## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion

# FIS Design in MATLAB: Sugeno FIS Editor

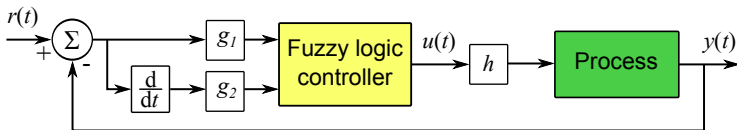


# FIS Design in MATLAB: Sugeno Rules



# FIS Tuning

As was mentioned, the testing and tuning is the last step of FIS development. If testing fails, the FIS has to be tuned or even redesigned.



External FIS tuning is performed via input and output scaling gains. The gain values may be either constant or functions of some sort, e.g. bell or Gaussian functions.

Internal tuning is performed by reviewing the membership functions and the rule-base. Trying out different inference and defuzzification operations is also a good practice.

The MATLAB FIS editor is a good tool for debugging. There you can observe the reaction of your rule firing strengths, input-output curves, etc. to the changes you make.

# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion

# Fuzzy PID Controller

The term Fuzzy PID controller can be understood in two ways:

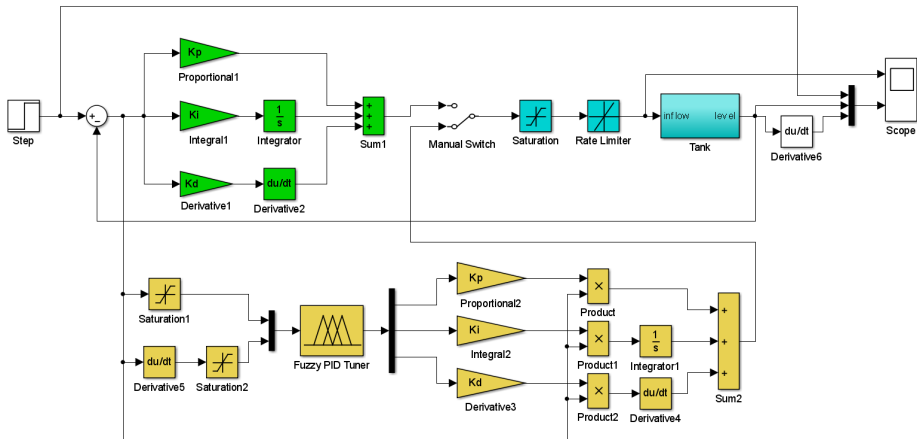
- A fuzzy FIS, which has the inputs  $e(t)$ ,  $\frac{d}{dt}e(t)$ ,  $\int e(t)dt$
- A crisp PID controller, the  $K_P$ ,  $K_I$  and  $K_D$  coefficients of which are tuned by a fuzzy expert system

A tunable PID controller allows to:

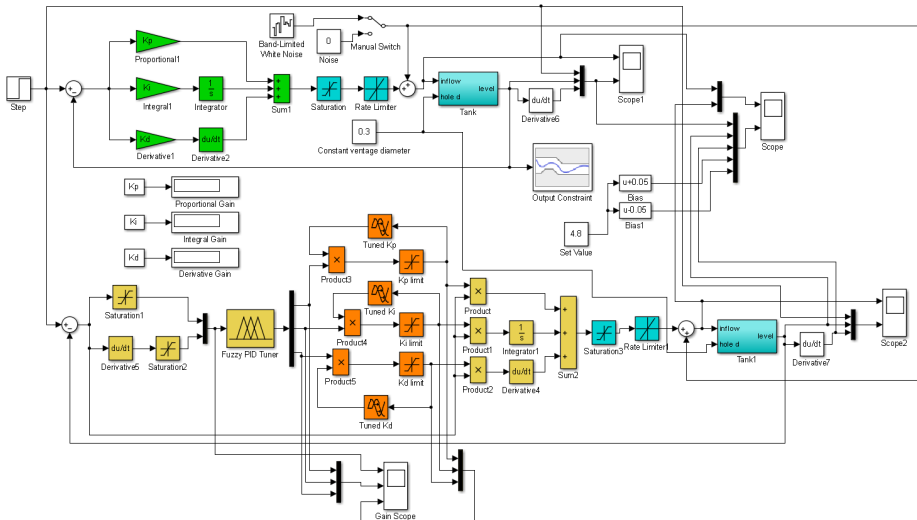
- Increase the robustness of the typical PID controller
- Increase its dynamic range
- Account for different scenarios of system operation



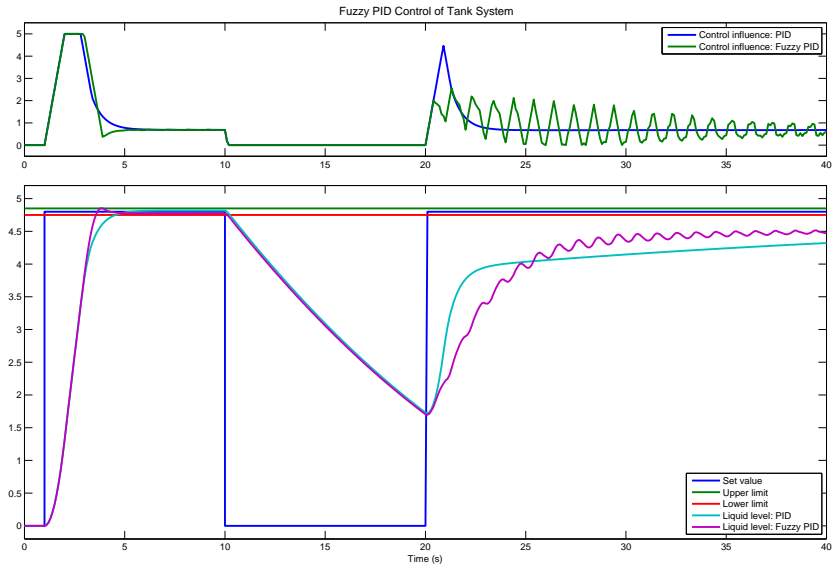
# Fuzzy PID Controller Example 1



# Fuzzy PID Controller Example 2



# Fuzzy PID Control Simulation



# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion

# Making Anything Fuzzy

- If you have some time variant system with parameters that cannot be statically specified...
- If you cannot describe parameter variation mathematically but you intuitively know how they should be changed...
- Introduce a fuzzy expert or control system to do it!
- We have seen it in the fuzzy PID example
- It is generally applicable to any linear or nonlinear dynamic system model

# Fuzzy Predictor

Having a discrete time series  $\theta_1, \theta_2, \dots, \theta_k$ , the task of a prediction algorithm is to determine the next values of the given time series  $\hat{\theta}_{k+1}, \hat{\theta}_{k+2}, \dots, \hat{\theta}_{k+l}$ .

The time series possesses certain dynamical properties  $\theta_{k+1} = \theta_k + \omega_k$ , where  $\omega_k$  is the system perturbation of unknown distribution.

The observed value may be affected by external interference  $\tilde{\theta}_k = \theta_k + \nu_k$ , where  $\nu_k$  is referred to as observation noise.

The Kalman (exponential average) predictor is given by the recurrence

$$\hat{\theta}_{k+1} = \alpha \hat{\theta}_k + (1 - \alpha) \tilde{\theta}_k,$$

where  $\hat{\theta}_{k+1}$  is the predicted value of the time series,  $\hat{\theta}_k$  is the last known predicted value,  $\tilde{\theta}_k$  is the last observed value and  $\alpha \in [0, 1]$  is the weight parameter.

# Fuzzy Predictor Continued

Basic logic tells us that:

- If the system is steady,  $\tilde{\theta}_k$  influences prediction more and  $\alpha \rightarrow 0$ .
- On the other hand, if the system is not steady or noisy,  $\tilde{\theta}_k$  is less reliable than  $\hat{\theta}_k$  and  $\alpha \rightarrow 1$ .

Specify the FIS input as *error*  $e_k = \left| \hat{\theta}_k - \tilde{\theta}_k \right|$ , then develop rules, e.g.

IF error is small THEN  $\alpha$  is large

IF error is medium THEN  $\alpha$  is medium

IF error is large THEN  $\alpha$  is small

And, well, you know the rest.

P.S. Think, how introducing the *change-in-error* into the FIS will improve the situation.

# General Linear Dynamic Model

The linear discrete-time dynamic system model takes the form

$$\begin{aligned}\mathbf{x}_k &= \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{q}_{k-1} \\ \mathbf{y}_k &= \mathbf{H}_{k-1}\mathbf{x}_k + \mathbf{r}_{k-1}\end{aligned},$$

where  $\mathbf{x}_k$  is the system state vector at time step  $k$ ,  $\mathbf{y}_k$  is the measurement vector at  $k$ ,  $\mathbf{A}_{k-1}$  is the transition matrix of the dynamic model,  $\mathbf{H}_{k-1}$  is the measurement matrix,  $\mathbf{q}_{k-1} \sim \mathcal{N}(0, \mathbf{Q}_{k-1})$  is the process noise with covariance  $\mathbf{Q}_{k-1}$  and  $\mathbf{r}_{k-1} \sim \mathcal{N}(0, \mathbf{R}_{k-1})$  is the measurement noise with covariance  $\mathbf{R}_{k-1}$ .

- For the majority of applications it is assumed that the noise has fixed variance and a normal distribution
- What to do, if noise is time variant and has varying distribution?
- One solution is to develop a fuzzy system, which will estimate noise parameters and tune the controller, filter, etc. online



# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

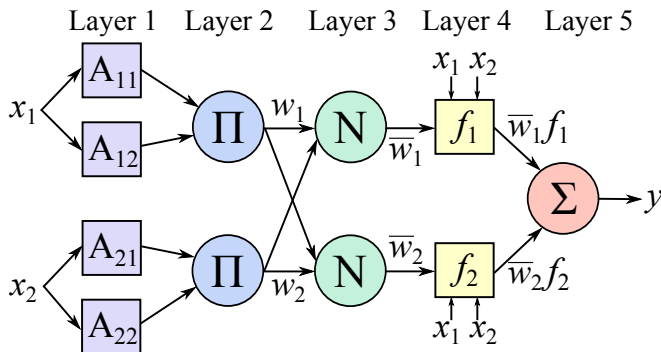
- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion

# Adaptive Neuro-Fuzzy Inference System

Adaptive Neuro-Fuzzy Inference System (ANFIS) is a representation of the Sugeno FIS in a form of a feed-forward neural network.



# ANFIS Architecture

The first-order Sugeno system  $r$ -th rule takes the form

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_1^k \text{ AND } x_2 \text{ is } A_2^l \text{ AND } \dots \text{ AND } x_n \text{ is } A_n^p \\ &\text{THEN } f_r = p_{n,r}x_1 + p_{n-1,r}x_2 + \dots + p_{1,r}x_n + p_{0,r} \end{aligned}$$

Lets take a system with two inputs  $x_1$ ,  $x_2$ , one output  $y$ , and two rules:

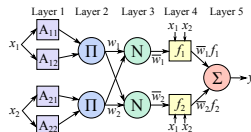
Rule1 : IF  $x_1$  is  $A_1^1$  AND  $x_2$  is  $A_2^1$  THEN  $f_1 = p_{2,1}x_1 + p_{1,1}x_2 + p_{0,1}$

Rule2 : IF  $x_1$  is  $A_1^2$  AND  $x_2$  is  $A_2^2$  THEN  $f_2 = p_{2,2}x_1 + p_{1,2}x_2 + p_{0,2}$

**Layer 1:** Every  $i^*$ -th node is an adaptive node with a function

$$O_{1,i^*} = \mu_{A_i^k}(x_i), \quad i^* = i \times k : i = 1, 2; k = 1, 2.$$

The parameters of the node's MF are called **premise parameters**.



# ANFIS Architecture Continued

**Layer 2:** Every  $i^*$ -th node is a fixed node, which calculates the firing strengths for each rule:

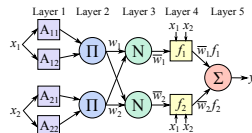
$$O_{2,i^*} = w_{i^*} = \mu_{A_1^k}(x_1) \mu_{A_2^k}(x_2), \quad i^* = k = 1, 2.$$

Besides product, other operations for the linguistic AND may be used.

**Layer 3:** Every  $i^*$ -th node is a fixed node, which computes the ratio of the  $i^*$ -th firing strength to the sum of all  $R$  rules firing strengths:

$$O_{3,i^*} = \bar{w}_{i^*} = \frac{w_{i^*}}{\sum_{r=1}^R w_r}, \quad i^* = 1, 2.$$

The outputs of this layer are called **normalized firing strengths**.



# ANFIS Architecture Continued

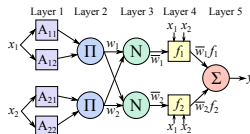
**Layer 4:** Every  $i^*$ -th node is an adaptive node with a function

$$O_{4,i^*} = \bar{w}_r f_r = \bar{w}_r (p_{2,r}x_1 + p_{1,r}x_2 + p_{0,r}), \quad i^* = r = 1, 2.$$

The parameters  $\{p_{2,r}, p_{1,r}, p_{0,r}\}$  are called **consequent parameters**.

**Layer 5:** The single node is a fixed node, which computes the overall output as a summation of all incoming values:

$$O_{5,1} = y = \sum_{r=1}^R \bar{w}_r f_r = \frac{\sum_{r=1}^R w_r f_r}{\sum_{r=1}^R w_r}.$$



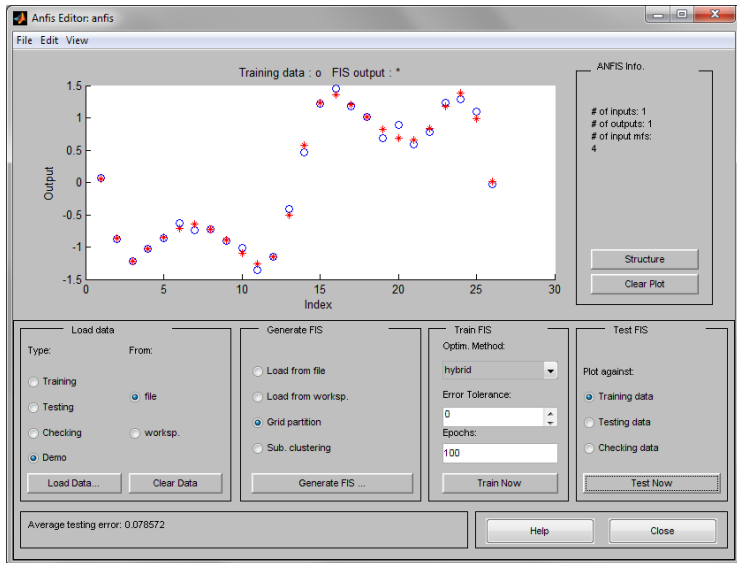
# ANFIS Training

ANFIS is trained by a hybrid learning algorithm. Each iteration makes two passes:

- During the forward pass node outputs go forward until layer 4 and the consequent parameters are identified by the least-squares method
- In the backward pass the error signals (i.e. reference minus layer 4 output) propagate backward and the premise parameters are updated by gradient descent

	Forward pass	Backward pass
Premise parameters	Fixed	Gradient descent
Consequent parameters	Least-squares estimator	Fixed
Signals	Node outputs	Error signals

# MATLAB ANFIS Editor: anfisedit



# Adaptive FIS Applications

The advantage of adaptive fuzzy systems compared to, e.g. Artificial Neural Networks (ANN) is that they are gray box as opposed to ANN, which are black box systems.

The application range is no less than of ANN:

- Nonlinear system identification
- Adaptive control (process control, inverse kinematics, etc.)
- Adaptive machine scheduling
- Clustering, classification and pattern recognition
- Adaptive expert systems, predictors
- Adaptive noise cancellation
- And others!



# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

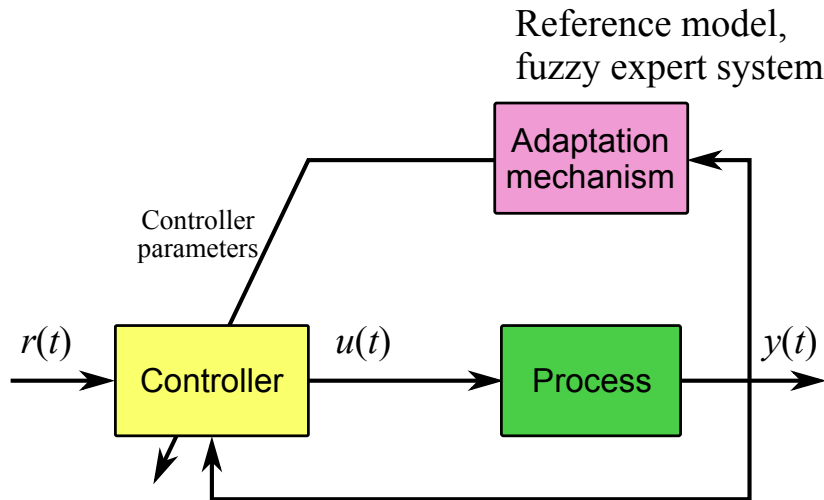
## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

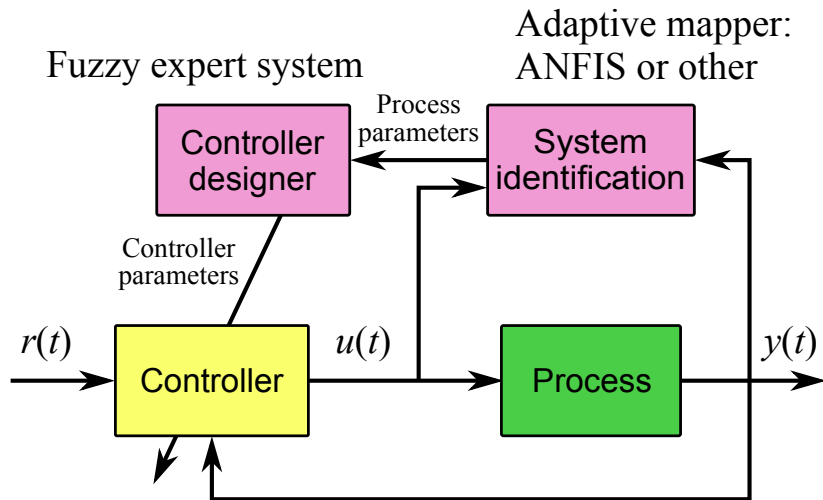
## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control**
- 3.5 Clustering and Classification
- 3.6 Discussion

# Direct Adaptive Control



# Indirect Adaptive Control



# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion

# Fuzzy Clustering and Classification

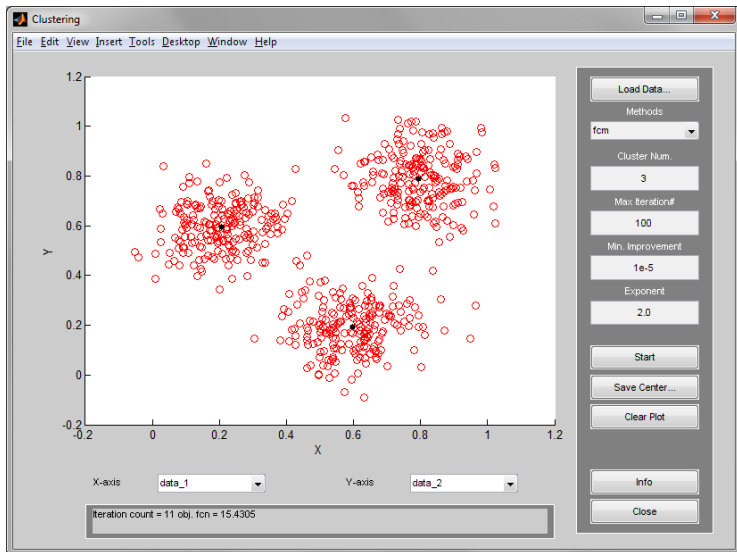
Fuzzy clustering and fuzzy classification differ from conventional crisp clustering and classification approaches in that:

- In crisp clustering each element of a dataset has a degree of belonging 1 to its assigned cluster and 0 to all other clusters
- In fuzzy clustering each element of a dataset has a degree of belonging ranging from 0 to 1 to each of the clusters
- In crisp classification a classified pattern belongs to one of the pre-specified classes with certainty 1 and with certainty 0 to all other classes
- In fuzzy classification a classified pattern belongs to each of the pre-specified classes with certainty ranging from 0 to 1

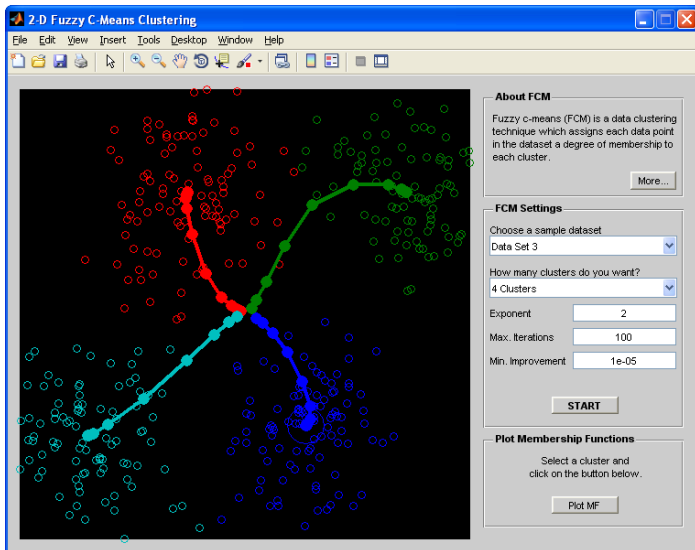
The most common fuzzy clustering algorithm is Fuzzy C-means clustering.

Fuzzy classification is performed applying any of the adaptive FIS structures, e.g. ANFIS, ARIC, GARIC, NNDFR, NEFCLASS, etc.

# Fuzzy C-Means: MATLAB GUI findcluster



# Fuzzy Clustering Demo: fcmdemo



# Outline

## 1. Concepts of Fuzzy Logic

- 1.1 Introduction
- 1.2 Fuzzy Sets
- 1.3 Fuzzy Set Operations
- 1.4 Membership Functions
- 1.5 Fuzzy Rules

## 2. Fuzzy Inference Systems

- 2.1 Introduction
- 2.2 Mamdani FIS
- 2.3 Mamdani Inference
- 2.4 Defuzzification
- 2.5 Mamdani FIS Editor
- 2.6 Takagi-Sugeno FIS
- 2.7 Sugeno FIS Editor

## 3. Applications

- 3.1 Fuzzy PID Controller
- 3.2 Fuzzy %Anything%
- 3.3 Adaptive FIS
- 3.4 Adaptive Control
- 3.5 Clustering and Classification
- 3.6 Discussion



# Mamdani vs Sugeno FIS

## Advantages of the Mamdani Method

- It is intuitive
- It has widespread acceptance
- It is well suited for human input

## Advantages of the Sugeno Method

- It is computationally more efficient
- It works well with linear techniques (e.g. PID control)
- It works well with optimization and adaptive techniques
- It has guaranteed continuity of the output surface
- It is well suited for mathematical analysis

# Criticism of Mamdani Fuzzy Control

Fuzzy control methods are “parasitic:” they simply implement trivial interpolations of control strategies obtained by other means

- 99% of fuzzy feedback control applications deal with essentially 1st or 2nd-order, overdamped, SISO systems

Attempt to emulate or duplicate human control behavior?

- Human is a very poor controller for complex, multi-variable, marginally stable dynamic plants
- Very hard to generate multidimensional if-then rule tables
- No guarantees of closed-loop stability, stability-robustness and of performance in presence of uncertainty
- Cannot generate “differential equation” controller rules

M. Athans, Crisp Control Is Always Better Than Fuzzy Feedback Control, EUFIT '99 debate with prof. L.A. Zadeh

# Criticism of Sugeno Fuzzy Control

- Approach developed to overcome criticism regarding closed-loop stability guarantees
- Design full-state feedback controllers for each linear model (using crisp control methods) and “interpolate” using membership functions
- Given that a state space model is necessary, why bother to introduce fuzzy ideas when conventional crisp control methods can deal with the design problem directly?
- Current methodology does not address stability-robustness and performance-robustness issues
- Current methodology does not address output feedback requiring dynamic compensator designs

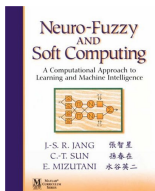
# Look on the Bright Side

- The answers to some of the critical claims can be found in Passino, Chapter 8
- Although there are many unsolved problems with fuzzy control, everyone may try and decide for himself, whether the methodology suits him or not
- Fuzzy systems are useful in many other fields of intelligent computer systems besides process control
- It is good to have this tool in your pocket
- If you cannot express your view in equations, but you can verbally — go fuzzy!

# Useful Literature



K. M. Passino and S. Yurkovich, *Fuzzy Control*.  
Addison Wesley Longman, Menlo Park, CA, 1998



J.-S. R. Jang, C.-T. Sun and E. Mizutani,  
*Neuro-fuzzy and Soft Computing: A Computational  
Approach to Learning and Machine Intelligence*.  
Prentice Hall, 1997