

Control Inteligente

Diseño de Controladores Fuzzy con Matlab L5

Luis Moreno, Santiago Garrido, Dorin Copaci

Dpto. Ing. de Sistemas y Automática
Universidad Carlos III
Madrid

Oct 2019



Table of contents

Control borroso con Matlab.

Introducción al control borroso con Matlab

- ▶ En Matlab se dispone del Fuzzy Logic Toolbox que incluye un conjunto de funciones que permiten diseñare implementar sistemas y controladores fuzzy. Entre las funciones que incluyen están las necesarias para la: fuzzificación, defuzzificación, y la inferencia fuzzy.
- ▶ Pueden desarrollarse como funciones o bien por medio de una Graphical User Interface (GUI). El bloque 'fuzzy logic controller' de Simulink block.
- ▶ Este bloque permite:
 - ▶ Crear y editar el fuzzy inference system (FIS).
 - ▶ Integrar el sistema fuzzy en simulaciones con Simulink.
 - ▶ Crear programas stand-alone en C que llaman a los sistemas fuzzy construidos en Matlab.

Fuzzy Logic Toolbox

- ▶ El Toolbox incluye tres categorías de herramientas:
 1. Llamadas a funciones (Command line functions).
 2. Herramientas gráficas o interactivas.
 3. Bloques en Simulink.

Fuzzy Logic Toolbox

Command Line FIS Functions

- ▶ **Funciones:**
 - ▶ **addmf.** Add membership function to FIS.
 - ▶ **addrule.** Add rule to FIS.
 - ▶ **addvar.** Add variable to FIS.
 - ▶ **defuzz.** Defuzzify membership function.
 - ▶ **evalfis.** Perform fuzzy inference calculation.
 - ▶ **evalmf.** Generic membership function evaluation.
 - ▶ **gensurf.** Generate FIS output surface.
 - ▶ **getfis.** Get fuzzy system properties.
 - ▶ **mf2mf.** Translate parameters between functions.
 - ▶ **mfstrtch.** Stretch membership function.
 - ▶ **newfis.** Create new FIS. `parsrule` & `parse` fuzzy rules.

Fuzzy Logic Toolbox

Command Line FIS Functions

- ▶ Funciones (cont.):
 - ▶ **plotfis**. Display FIS input-output diagram.
 - ▶ **plotmf**. Display all membership functions for one variable.
 - ▶ **readfis**. Load FIS from disk.
 - ▶ **rmmf**. Remove membership function from FIS.
 - ▶ **rmvar**. Remove variable from FIS.
 - ▶ **setfis**. Set fuzzy system properties.
 - ▶ **showfis**. Display annotated FIS.
 - ▶ **showrule**. Display FIS rules
 - ▶ **writefis**. Save FIS to disk.

Fuzzy Logic Toolbox

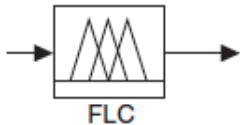
Graphical User Interface Editors (GUI tools)

- ▶ Editores en la herramienta gráfica: (GUI tools)
 - ▶ **fuzzy**. Basic FIS editor.
 - ▶ **mfedit**. Membership function editor.
 - ▶ **ruleedit**. Rule editor and parser.
 - ▶ **ruleview**. Rule viewer and fuzzy inference diagram.
 - ▶ **surfview**. Output surface viewer.

Fuzzy Logic Toolbox

Simulink Blocks

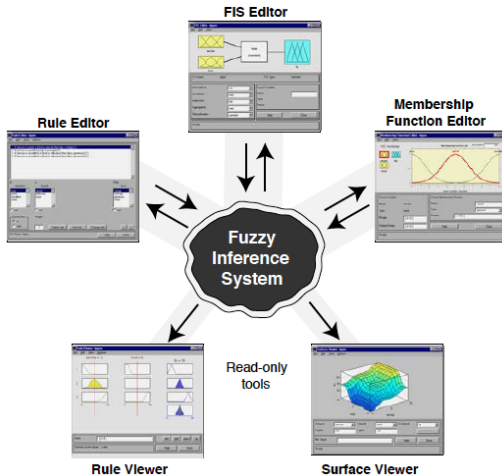
- ▶ Bloque de Simulink: Fuzzy logic controller.



Fuzzy Logic Toolbox

Simulink

- Estructura del ANFIS Editor GUI.

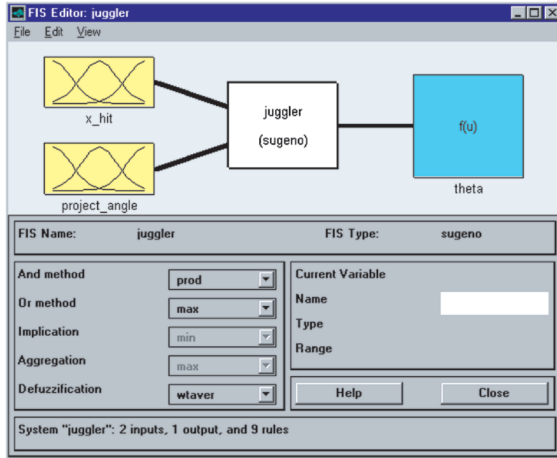


Fuente: Matlab fuzzy logic toolbox.

Fuzzy Logic Toolbox

Simulink Blocks

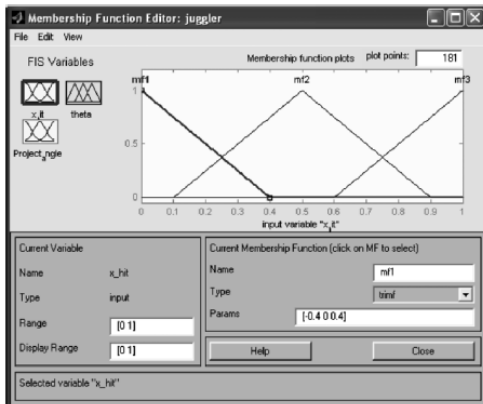
- ▶ Basic FIS editor.



Fuzzy Logic Toolbox

Simulink Blocks

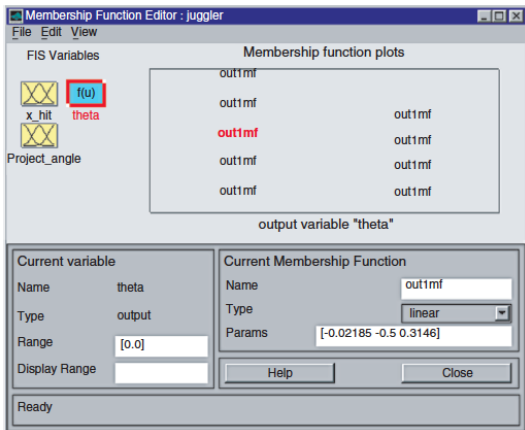
- ▶ Membership function editor (input 1)



Fuzzy Logic Toolbox

Simulink Blocks

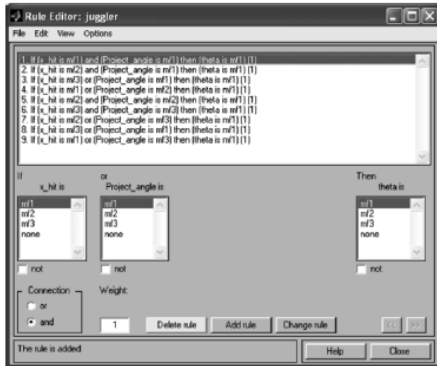
- Membership function editor (output 1, in Sugeno style)



Fuzzy Logic Toolbox

Simulink Blocks

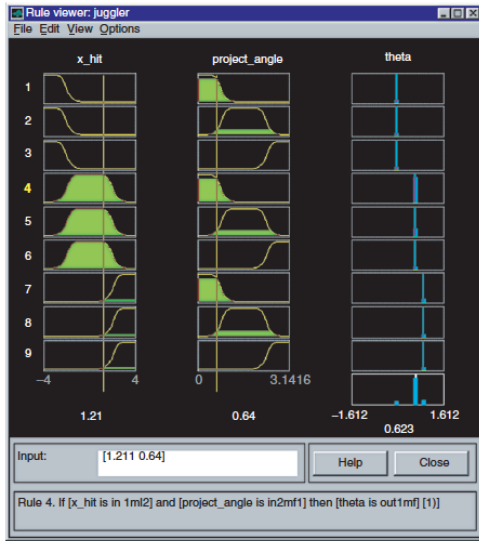
- ▶ Rule editor (sample with nine rule in verbose format)



Fuzzy Logic Toolbox

Simulink Blocks

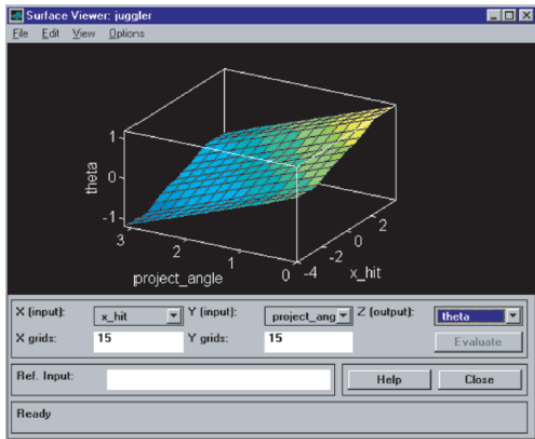
- ▶ Rule viewer (in Sugeno style)



Fuzzy Logic Toolbox

Simulink Blocks

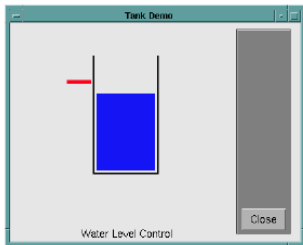
- ▶ Surface viewer (input 1 and input 2 versus output 1)



Ejemplo simple

Control de nivel de un depósito de agua

- ▶ Ejemplo en el Toolbox de Matlab: water level control.
- ▶ Para ejecutarlo teclear sltank en la ventana de control de Matlab.
- ▶ El control del sistema se realiza abriendo y cerrando la válvula que controla el flujo de entrada de agua ya que el flujo de salida no es controlable y depende del del diámetro del orificio de salida (constante en este ejemplo) y de la presión en el tanque (que varia con el nivel de agua).



Ejemplo simple

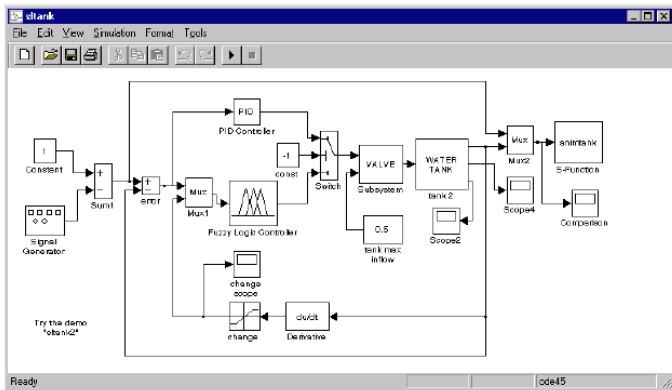
Control de nivel de un depósito de agua

- ▶ Este sistema es no lineal ya que el flujo de descarga de agua depende no linealmente de la altura de agua contenida en el depósito.
- ▶ La entrada al controlador es el error (error = nivel de agua deseado - nivel de agua actual),
- ▶ y la salida es la tasa de variación de la apertura de la válvula.
- ▶ Las reglas de control pueden ser las siguientes:
 1. IF (level is ok) THEN (valve is no change) (1)
 2. IF (level is low) THEN (valve is open fast) (1)
 3. IF (level is high) THEN (valve is close fast) (1)

Ejemplo simple

Control de nivel de un depósito de agua

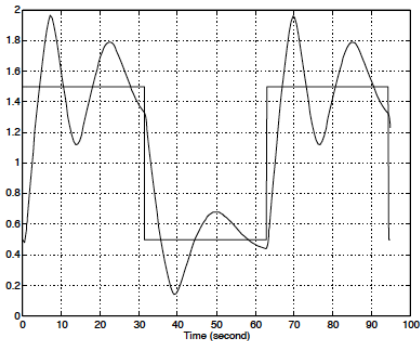
- ▶ El diagrama del bloque Simulink para el sistema ejemplo sltank es el siguiente (el fichero tank.fis es cargado en la estructura FIS que contiene la definición del tanque).



Ejemplo simple

Control de nivel de un depósito de agua

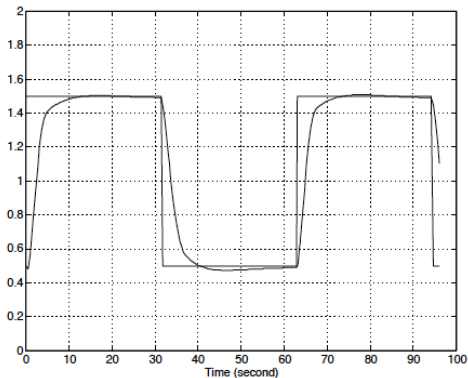
- ▶ Resultado de la simulación de las primeras reglas de control.
- ▶ Claramente insuficiente.



Ejemplo simple

Control de nivel de un depósito de agua

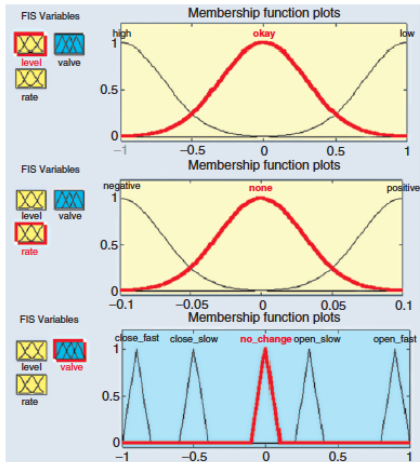
- ▶ Si añadimos otra entrada a las reglas del controlador que indique la velocidad de variación del nivel y dos nuevas reglas:
 - 4 If (level is good) and (rate is negative), then (valve is close slow) (1).
 - 5 If (level is good) and (rate is positive), then (valve is open slow) (1).
- ▶ El resultado mejora notablemente.



Ejemplo simple

Control de nivel de un depósito de agua

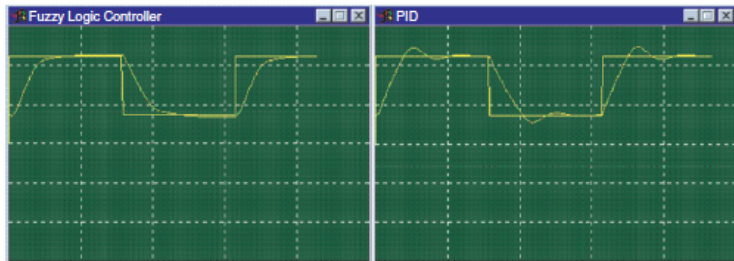
- Funciones de pertenencia.



Ejemplo simple

Control de nivel de un depósito de agua

- ▶ Comparación con un PID convencional



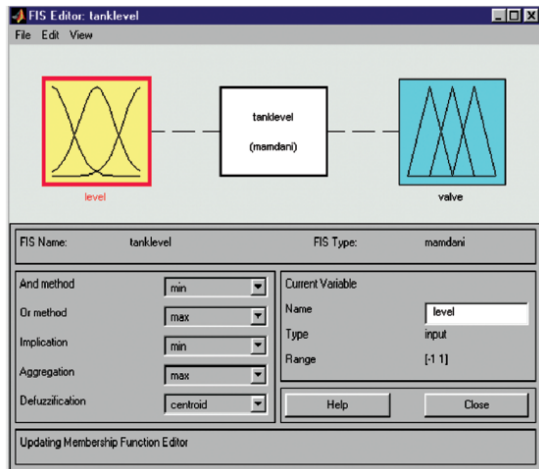
Casos prácticos

- ▶ Ejemplo 1.
 - ▶ Control de temperatura de un reactor donde el error y el cambio de error son entradas al controlador.
 - ▶ La temperatura del reactor es controlada por medio de la temperatura del baño que rodea al reactor, es decir la temperatura se controla mediante el control del flujo de líquido refrigerante que entra en el reactor (baño) .

Ejemplo 1

Solución

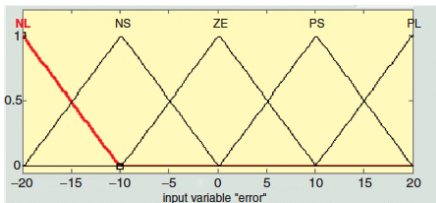
- ▶ Paso 1. Open the FIS editor and edit the membership function for the input and output as shown in the figure. Select the Mamdani or Sugeno method, fuzzification method, and defuzzification method.



Ejemplo 1

Solución

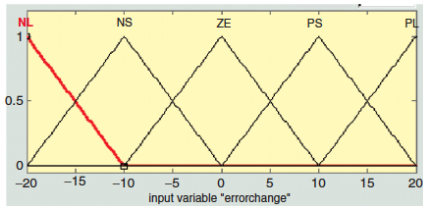
- ▶ In the FIS editor (choose either Mamdani or Sugeno model), we choose triangular membership function, and we can set the linguistic variables.
- ▶ Membership Function for error is



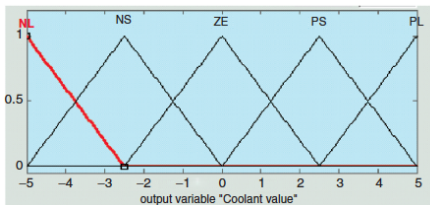
Ejemplo 1

Solución

- ▶ Membership Function for change in error is



- ▶ Membership Function for valve position is



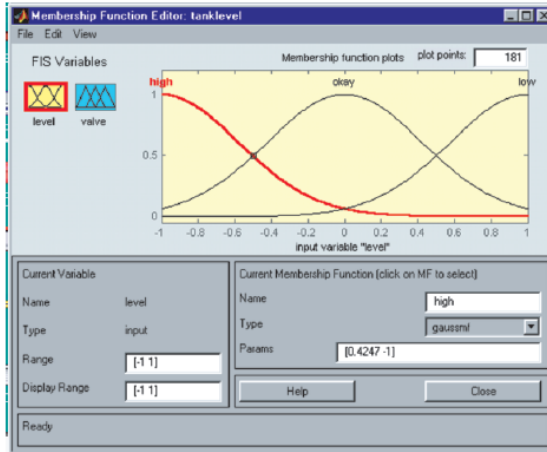
Ejemplo 1

Solución

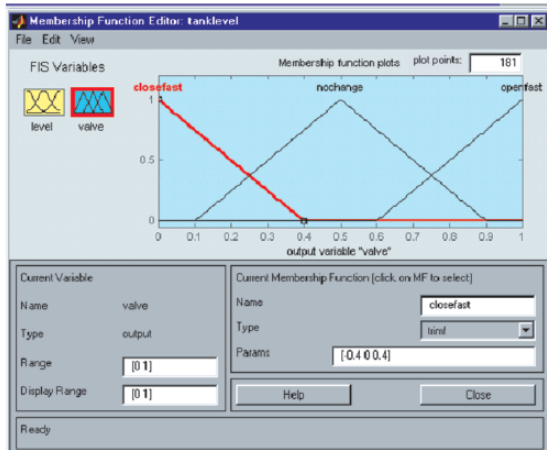
- ▶ The connective rules are formed and based on these rules the fuzzy associative memory table is formed.

e	δc				
	NL	NS	NE	PS	PL
PL	ZE	PS	PL	PL	PL
PS	NS	ZE	PS	PL	PL
ZE	NL	NS	ZE	PS	PL
NS	NL	NL	NS	ZE	PS
NL	NL	NL	NL	NS	ZE

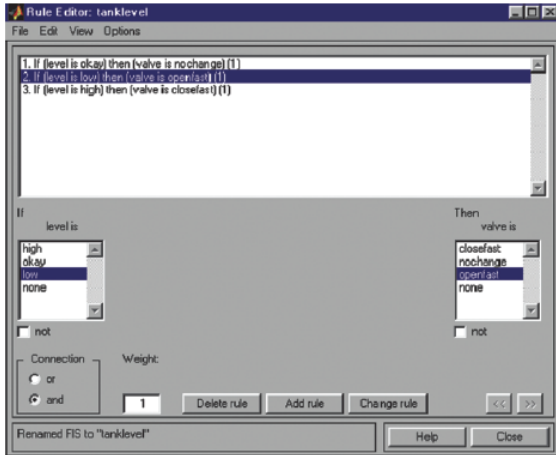
- ▶ Paso 2. Click the input variable and edit the membership function as shown below.



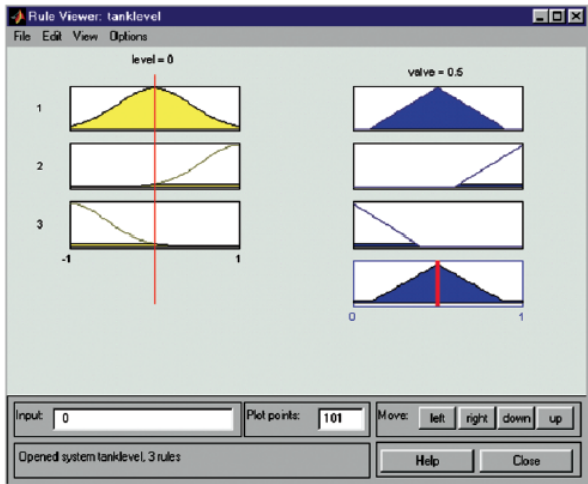
- ▶ Paso 3. Click the output variable and edit the member ship function as shown below.



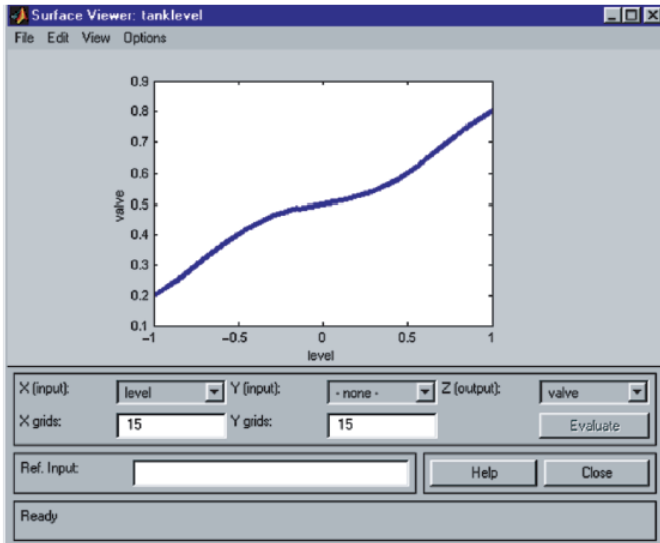
- ▶ Paso 4. Edit the rule base by clicking the rule base from view menu.



- ▶ Paso 5. To view the rule viewer go to the view and click rule viewer.



- ▶ Paso 6. To view the surface viewer go to the view and click surface viewer.



- ▶ Paso 7. Save the file to the workspace as tanklevel and go to the command window and enter the command tanklevel.

```
>> tanklevel
tanklevel =
name: 'tanklevel'
type: 'mamdani'
andMethod: 'min'
orMethod: 'max'
defuzzMethod: 'centroid'
impMethod: 'min'
aggMethod: 'max'
input: [1x1 struct]
output: [1x1 struct]
rule: [1x3 struct]
```

Aproximación de una función

Identificación

- ▶ Let $y = -2x + x^2$.
 - ▶ Form a fuzzy system, which approximates function f , when $x \in [-10, 10]$.
Repeat the same by adding random, normally distributed noise with zero mean and unit variance.
 - ▶ Simulate the output when the input is $\sin(t)$.
Observe what happens to the signal shape at the output.

Identificación

Método 1

- ▶ This is achieved by writing a Matlab program

```
% Generate input-output data and plot it
x=[-10:.5:10]'; y=-2*x-x.*x;
% Plot of parabola
plot(x,y)
grid
xlabel('x');ylabel('output');title('Nonlinear characteristics')
% Store data in appropriate form for genfis1 and anfis and plot it
data=[x y];
trndata=data(1:2:size(x),:);
chkdata=data(2:2:size(x),:);
% Plot of training and checking data generated from parabolic equation
plot(trndata(:,1),trndata(:,2),'o',chkdata(:,1),chkdata(:,2),'x')
xlabel('x');ylabel('output');title('Measurement data'); grid
%Initialize the fuzzy system with command genfis1. Use 5 bellshaped
membership functions.
nu=5; mftype='gbellmf'; fismat=genfis1(trndata, nu, mftype);
%The initial membership functions produced by genfis1 are plotted
plotmf(fismat,'input',1)
xlabel('x');ylabel('output');title('Initial membership functions');
grid
```

Identificación

Método 1

► Matlab program (cont.)

```
% Apply anfis-command to find the best FIS system - max number of
iterations = 100
numep=100;
[parab, trnerr,ss,parabcheck,chkerr]=anfis(trndata,fismat,numep,[],chkdata);
%Evaluate the output of FIS system using input x
anfi=evalfis(x,parab);
% Plot of trained fuzzy system using trained data
plot(trndata(:,1),trndata(:,2),'o',chkdata(:,1),chkdata(:,2),'x',x,anfi,'-')
grid
xlabel('x');ylabel('output');title('Goodness of fit')
```

Identificación

Método 1

► Salida:

iterations = 100

ANFIS info:

Number of nodes: 24

Number of linear parameters: 10

Number of nonlinear parameters: 15

Total number of parameters: 25

Number of training data pairs: 21

Number of checking data pairs: 20

Number of fuzzy rules: 5

Start training ANFIS.

1	1.11035	1.11725
---	---------	---------

2	1.1055	1.11205
---	--------	---------

3	1.10065	1.10685
---	---------	---------

4	1.09578	1.10165
---	---------	---------

5	1.09091	1.09644
---	---------	---------

Step size increases to 0.011000 after epoch 5.

6	1.08602	1.09123
---	---------	---------

7	1.08063	1.08548
---	---------	---------

8	1.07523	1.07974
---	---------	---------

9	1.06982	1.07399
---	---------	---------

Identificación

Método 1

► cont.:

Step size increases to 0.012100 after epoch 9.

10	1.0644	1.06823
----	--------	---------

11	1.05842	1.06189
----	---------	---------

12	1.05242	1.05555
----	---------	---------

13	1.04642	1.0492
----	---------	--------

Step size increases to 0.013310 after epoch 13.

.

.

.

.

.

.

86	0.021108	0.0264632
----	----------	-----------

87	0.0107584	0.0173793
----	-----------	-----------

88	0.0173351	0.0221364
----	-----------	-----------

89	0.00977897	0.0162064
----	------------	-----------

90	0.0165189	0.0217102
----	-----------	-----------

91	0.00934733	0.0162595
----	------------	-----------

Step size decreases to 0.066602 after epoch 91.

92	0.0160967	0.0218316
----	-----------	-----------

93	0.00734716	0.0157201
----	------------	-----------

94	0.0155897	0.0219794
----	-----------	-----------

95	0.00732559	0.016182
----	------------	----------

Identificación

Método 1

► cont.:

Step size decreases to 0.059942 after epoch 95.

96	0.015227	0.0222669
----	----------	-----------

97	0.0061041	0.0163292
----	-----------	-----------

98	0.0147314	0.022553
----	-----------	----------

99	0.00645142	0.0170149
----	------------	-----------

Step size decreases to 0.053948 after epoch 99.

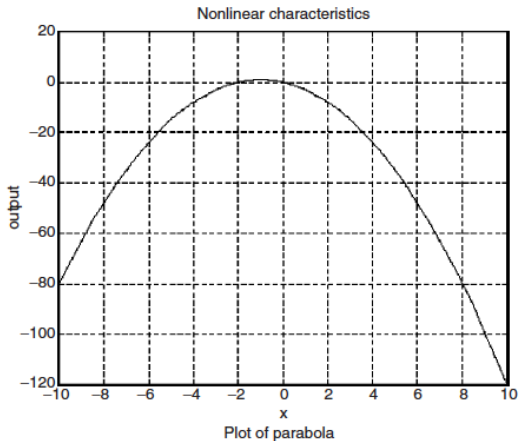
100	0.0144049	0.0229312
-----	-----------	-----------

Designated epoch number reached → ANFIS training completed at epoch 100.

Identificación

Método 1

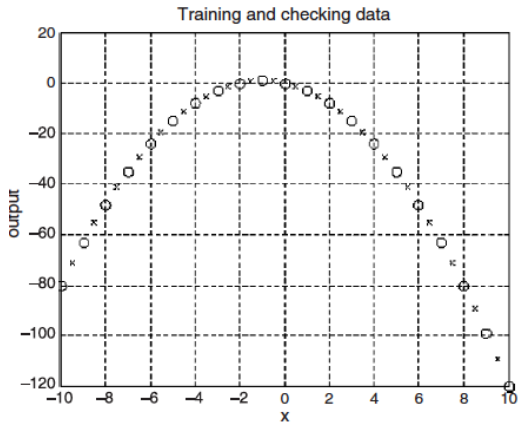
► Curva real



Identificación

Método 1

- ▶ Curva muestreada, datos para identificación y verificación.

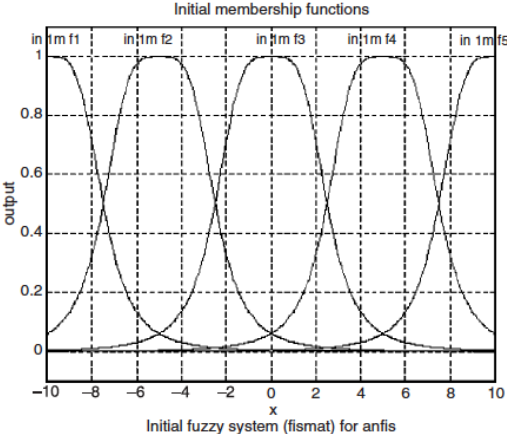


Training data (o) and checking data (x) generated from the parabolic equation

Identificación

Método 1

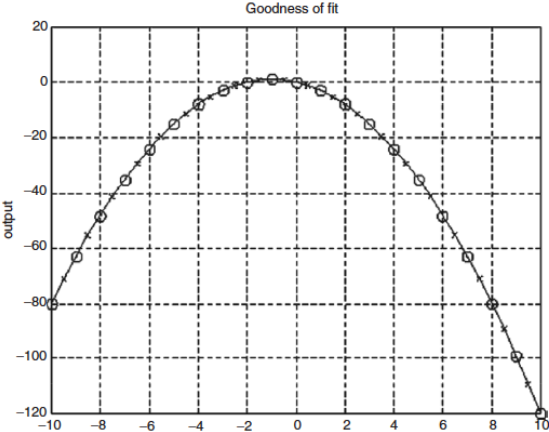
- ▶ Funciones de pertenencia iniciales para el sistema borroso.



Identificación

Método 1

- ▶ Ajuste despues de entrenar el sistema borroso.

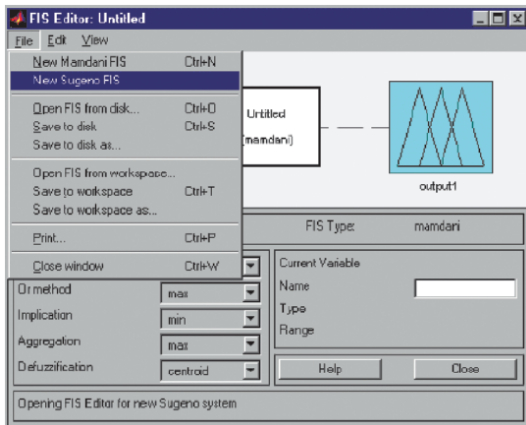


Identificación

Método 2

Using a fuzzy logic toolbox graphical user interface (GUI) can perform the same problem.

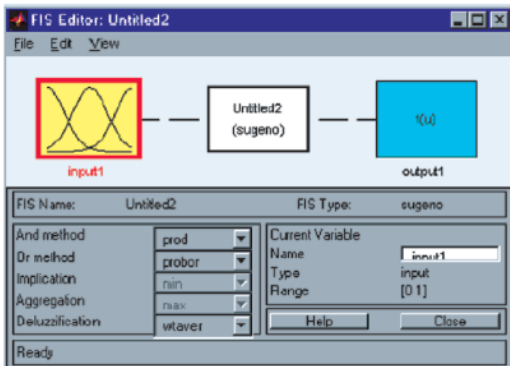
- 1 Open fuzzy toolbox GUI, choose new Sugeno system



Identificación

Método 2

- 2 Generate a new Sugeno type fuzzy system.



Identificación

Método 2

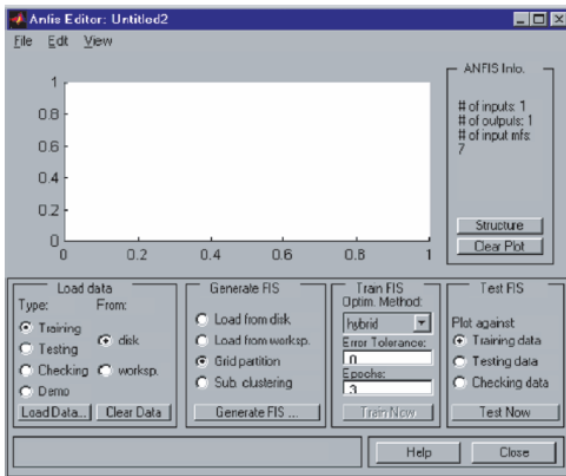
3 Display a new Sugeno type system.



Identificación

Método 2

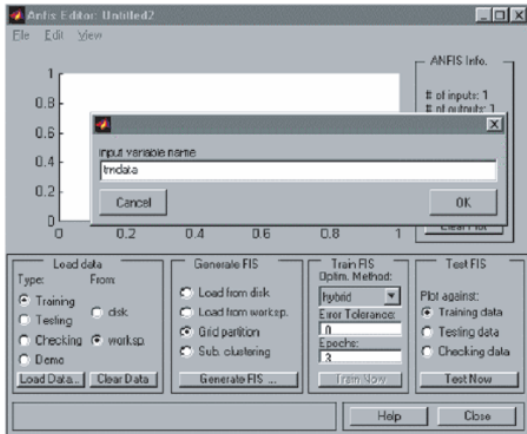
4 Generating anfis display.



Identificación

Método 2

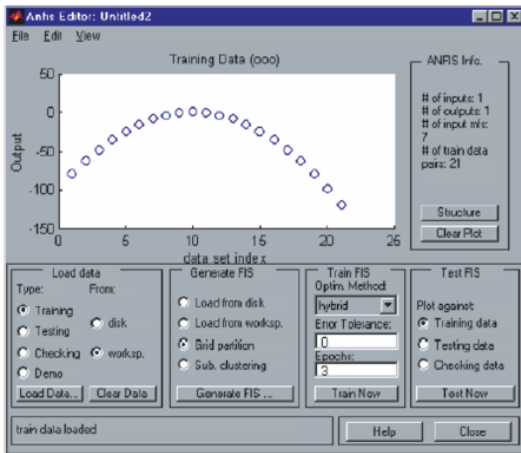
5 Anfis editor display.



Identificación

Método 2

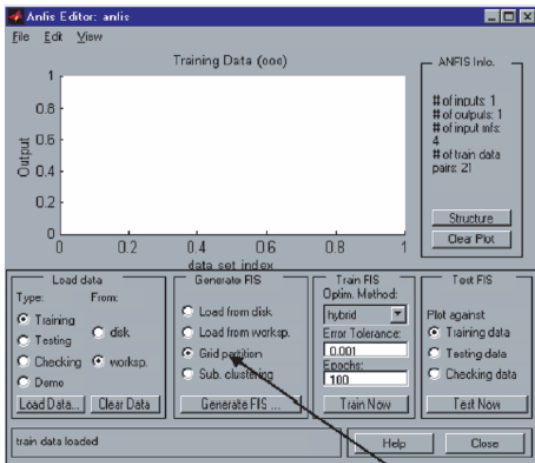
6 Load training data.



Identificación

Método 2

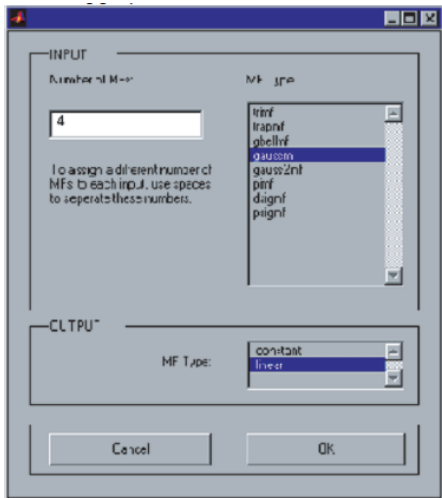
- 7 Plot of training data. The x-axis indicates numbering of data points rather than absolute values.



Identificación

Método 2

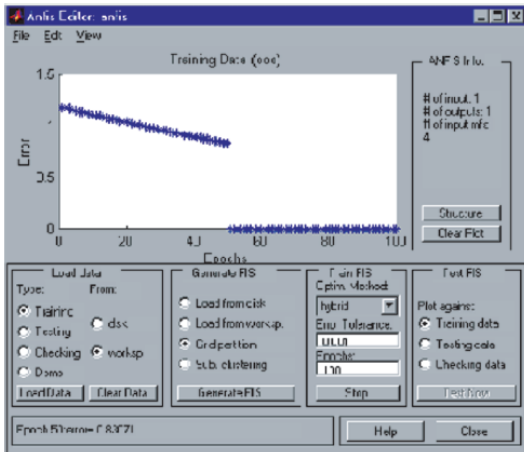
8 Generating initial FIS matrix using grid partition.



Identificación

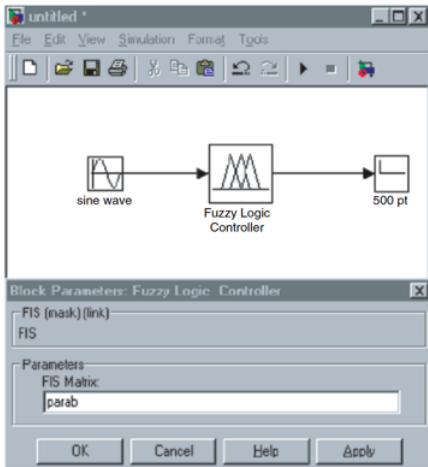
Método 2

- 9 Default membership function type (gaussmf) and their number (4).
Training when error tolerance is chosen to be 0.001 and number of epochs is limited to 100.



Identificación

Método 2

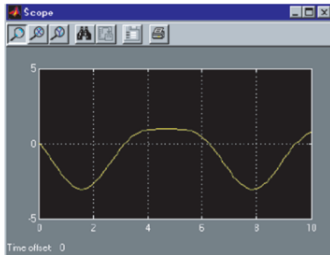


Entire system

The graphical output can be viewed through the Scope block

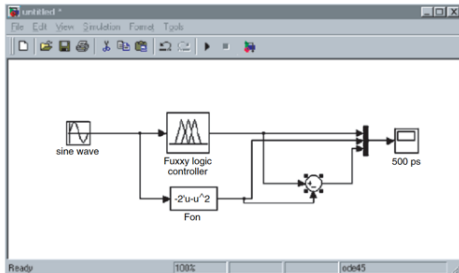
Identificación

Método 2



Output of the Simulink system

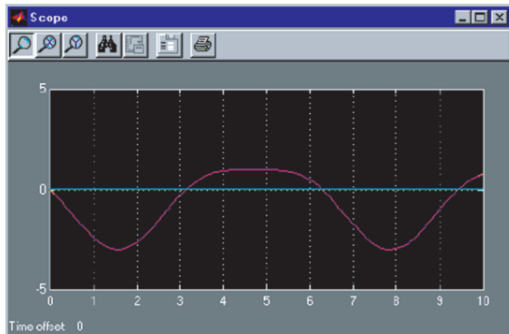
Comparison between the actual functions $-2x-x^2$ and the fuzzy system approximation is shown below.



Simulink blocks for the comparison of the actual function and the fuzzy system approximation

Identificación

Método 2



Comparison Output Responses

Combining the *Fuzzy system* with Simulink is an important feature from the user's point of view. Once the fuzzy system has been determined, it can be used in Simulink to simulate dynamical systems. This provides the user a very powerful tool to investigate behavior of complex systems. Computation by hand is tedious and in practice impossible without a computer, but it is at this point when fuzzy systems become really interesting and exciting.

Casos de estudio

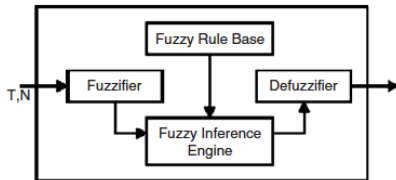
Biomedicina: Fuzzy Logic-Based Anesthetic Depth Control

▶ Introducción

- ▶ En la mayoría de las operaciones quirúrgicas en hospitales, para anestesiarse a los pacientes, se usan técnicas manuales.
- ▶ Los sistemas manuales funcionan normalmente en forma ON o OFF.
- ▶ Diferentes estudios han mostrado que los sistemas fuzzy pueden usarse para controlar procesos complejos.
- ▶ El beneficio potencial de usar un sistema de control fuzzy durante la anestesia a los pacientes estriba en la mejora de la seguridad y del confort, ya que permite la atención de los anestesiadores se pueda concentrar en otras variables fisiológicas que tienen que mantener bajo control, haciendo un uso óptimo del agente anestésico, y disminuyendo el coste de las operaciones quirúrgicas.

Biomedicina: Fuzzy Logic-Based Anesthetic Depth Control

- ▶ Estructura del sistema:



- ▶ Entradas al control fuzzy: T y N representan la presión sanguínea (mmHg) y la frecuencia del pulso (pm^{-1}), obtenidos de los pacientes durante la anestesia.
- ▶ La salida: Anesthesia Output (AO) representa la salida del sistema fuzzy.

Biomedicina: Fuzzy Logic-Based Anesthetic Depth Control

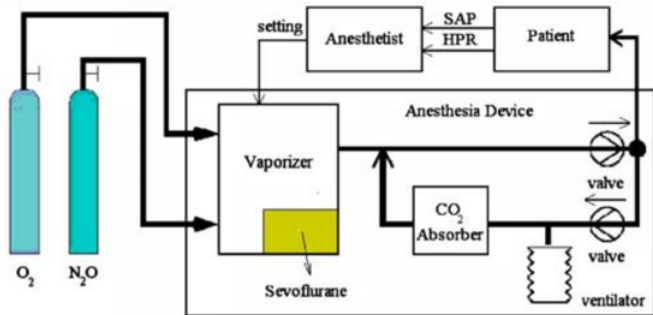


Fig. 1 The measurement system (flow and signal diagram during anesthesia)

Biomedicina: Fuzzy Logic-Based Anesthetic Depth Control

- ▶ Los valores mínimos y máximos (presión sanguínea sistólica y frecuencia del pulso) se han obtenido de medidas realizadas cada 10 minutos en operaciones quirúrgicas a 27 pacientes (12 hombres y 13 mujeres, de edades comprendidas entre 3 y 77 años).

Variable	Minimum value	Maximum value
Blood Pressure (mmHg)	60	220
Pulse Rate (p m^{-1})	40	150
Anesthesia Ratio (%u)	0	4

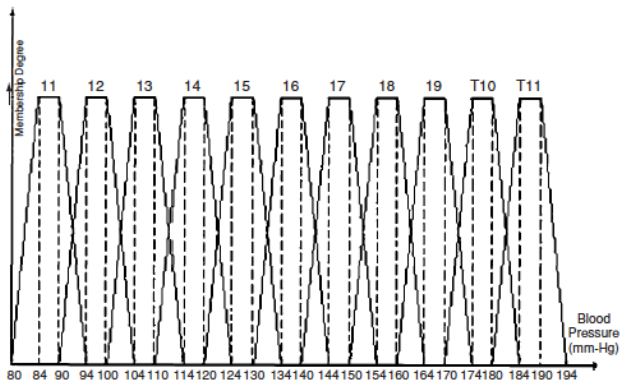
Biomedicina: Fuzzy Logic-Based Anesthetic Depth Control

- ▶ Los valores de las funciones de pertenencia definidas:

Linguistic variables	Very low	Low	Normal	High	Very high
Blood pressure (mmHg)	< 80	90	100–140	160–170	> 190
Pulse rate (p m) ⁻¹	< 50	60	70–90	95–110	> 120
Anesthesia ratio (%u)	0	0.5–0.8	1–2.5	3–3.6	4

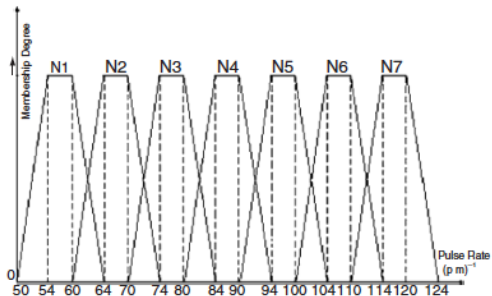
Biomedicina: Fuzzy Logic-Based Anesthetic Depth Control

- ▶ Funciones de pertenencia para la entrada de presión sanguínea:



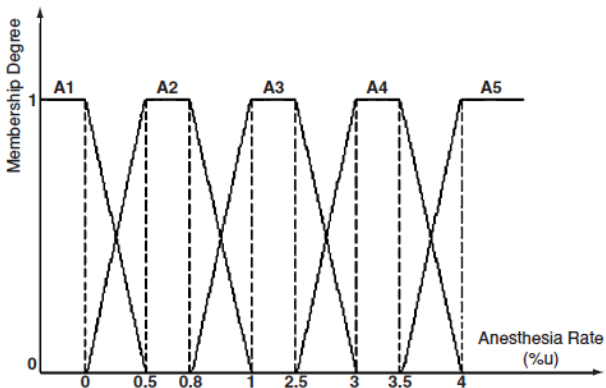
Biomedicina: Fuzzy Logic-Based Anesthetic Depth Control

- Funciones de pertenencia para la entrada de frecuencia cardiaca:



Biomedicina: Fuzzy Logic-Based Anesthetic Depth Control

- Funciones de pertenencia para la salida (Tasa de anestesia) :



Biomedicina: Fuzzy Logic-Based Anesthetic Depth Control

- ▶ Base de reglas (S indica situación no válida) :

	N1	N2	N3	N4	N5	N6	N7
T1	A1	A1	A2	A2	A2	S	S
T2	A2	A2	A3	A3	A3	A4	A4
T3	A2	A3	A3	A3	A3	A4	A4
T4	A2	A3	A3	A3	A3	A4	A4
T5	A2	A3	A3	A3	A3	A4	A4
T6	A2	A3	A3	A3	A3	A4	A4
T7	A2	A3	A3	A3	A3	A4	A4
T8	S	A4	A4	A4	A4	A5	A5
T9	S	A4	A4	A4	A4	A5	A5
T10	S	A4	A4	A4	A4	A5	A5
T11	S	A5	A5	A5	A5	A5	A5

Biomedicina: Fuzzy Logic-Based Anesthetic Depth Control

► Situaciones no válidas :

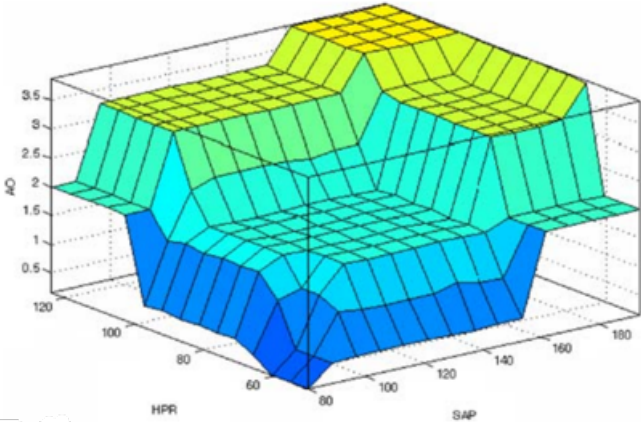
Blood pressure	Pulse rate	Anesthesia rate
High	Very_low	Invalid condition
Very_high	Very_low	Invalid condition
High	Low	Invalid condition
Very_high	Low	Invalid condition
Very_low	High	Invalid condition
Low	High	Invalid condition
Very_low	Very_high	Invalid condition
Low	Very_high	Invalid condition

Biomedicina: Fuzzy Logic-Based Anesthetic Depth Control

- ▶ Defuzzificación (COG):

$$AO = \frac{\sum_{x=a}^b \mu_A(x)x}{\sum_{x=a}^b \mu_A(x)} \quad (1)$$

Biomedicina: Fuzzy Logic-Based Anesthetic Depth Control



Surface view of the fuzzy logic system

▶ Fin L5