

# Modelli Fuzzy

---



*Un modello fuzzy consiste in un insieme di regole deduttive che legano gli ingressi e le uscite basandosi su predicati e connessioni "vaghi" (fuzzy) di tipo IF - THEN*

**IF {Ingressi} THEN {Uscite}**

# Tipi di Modelli Fuzzy

---

1) **Modelli Linguistici (Mamdani)**: Si basano sulle classiche regole di inferenza fuzzy fra antecedente e conseguente, ambedue descritti con dei fuzzy sets che rappresentano dei concetti linguistici

IF {Antecedente} THEN {Consequente}

---

2) **Modelli di Sugeno**: gli antecedenti fuzzy (linguistici) implicano dei conseguenti funzionali (deterministici)

IF {Antecedente} THEN { $y = f(y,u)$ }

# Tipi di modelli dinamici Fuzzy

---

Mamdani: Modello fuzzy come insieme di **regole inferenziali**

Modelli  
linguistici  
(Mamdani)

*IF  $u_k$  is  $B_0$  AND  $u_{k-1}$  is  $B_1$  ... AND  $u_{k-n}$  is  $B_n$   
AND  $y_{k-1}$  is  $A_1$  AND ... AND  $y_{k-n}$  is  $A_n$   
THEN  $y_k$  is  $A_0$*

---

Sugeno: Modello fuzzy come insieme di **relazioni funzionali**

Modelli  
di  
Sugeno

*IF  $u_k$  is  $B_0$  AND  $u_{k-1}$  is  $B_1$  ... AND  $u_{k-n}$  is  $B_n$   
AND  $y_{k-1}$  is  $A_1$  AND ... AND  $y_{k-n}$  is  $A_n$   
THEN  $y_k = f(y_{k-1}, y_{k-2}, \dots, y_{k-n}, u_{k-1}, u_{k-2}, \dots, u_{k-n})$*

# Modelli à la Mamdani

- ☞ Si compongono di un insieme di regole fuzzy che hanno lo scopo di riprodurre il comportamento osservato
- ☞ L'inclusione fra gli antecedenti di ingressi ed uscite passati è limitata solo dalla causalità (*non si possono usare elementi futuri...*)
- ☞ Esempio: si voglia modellare un sistema dinamico alle differenze con la seguente struttura

$$y_t = f( y_{t-1}, u_{t-1}, u_{t-2} )$$

- ☞ Si può utilizzare una struttura fuzzy, ad es. con due regole, del tipo

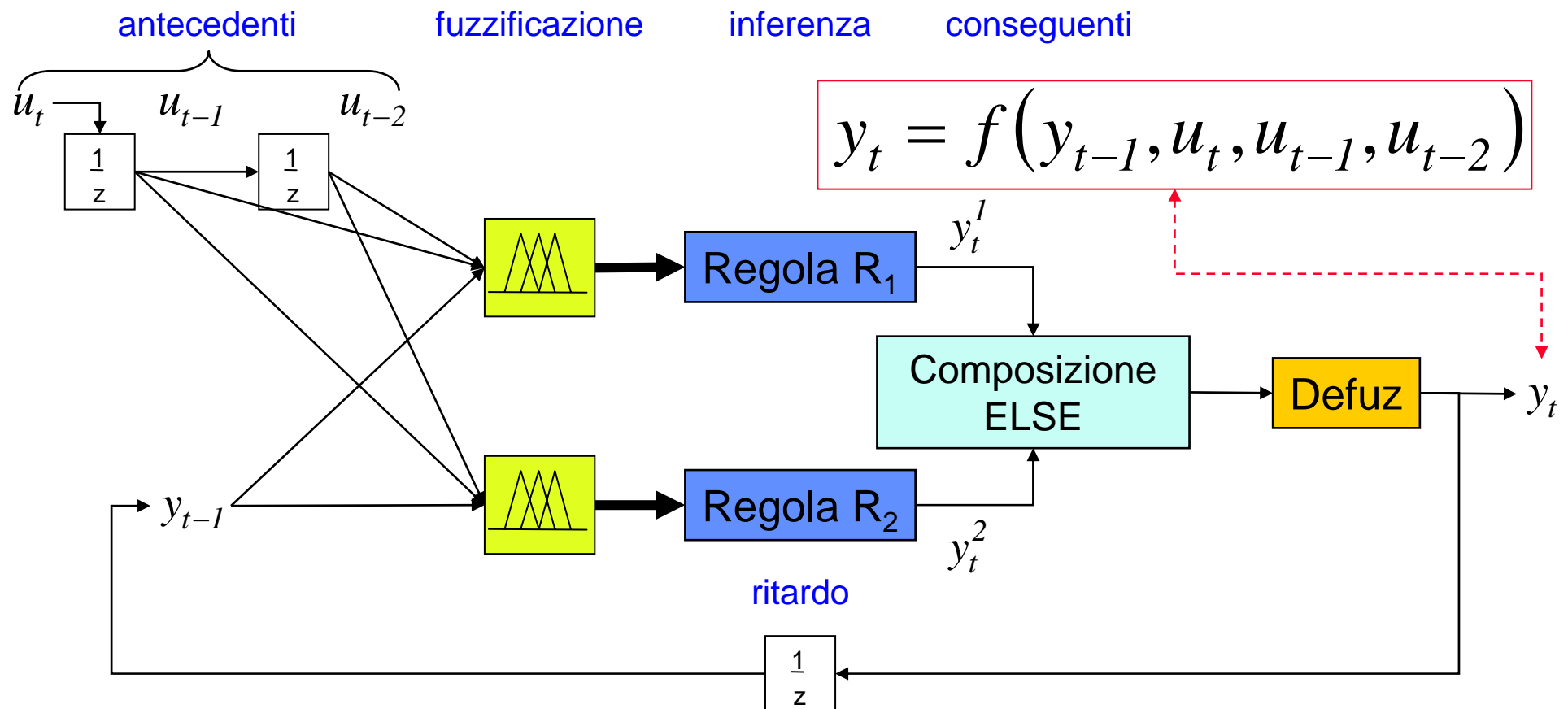
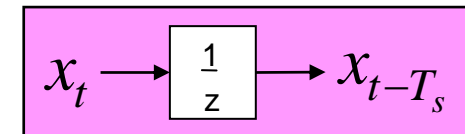
$R_1 : IF y_{t-1} \text{ is } A_{11} \text{ AND } u_{t-1} \text{ is } B_{11} \text{ AND } u_{t-2} \text{ is } B_{12} \text{ THEN } y_t \text{ is } C_1$   
 $ELSE$

$R_2 : IF y_{t-1} \text{ is } A_{21} \text{ AND } u_{t-1} \text{ is } B_{21} \text{ AND } u_{t-2} \text{ is } B_{22} \text{ THEN } y_t \text{ is } C_2$

- ☞ Il successo del modello consisterà nell'abilità di includere nelle regole tutti i comportamenti del sistema e scegliere in modo efficace i rappresentatori A, B e C, basandosi sui dati osservati.

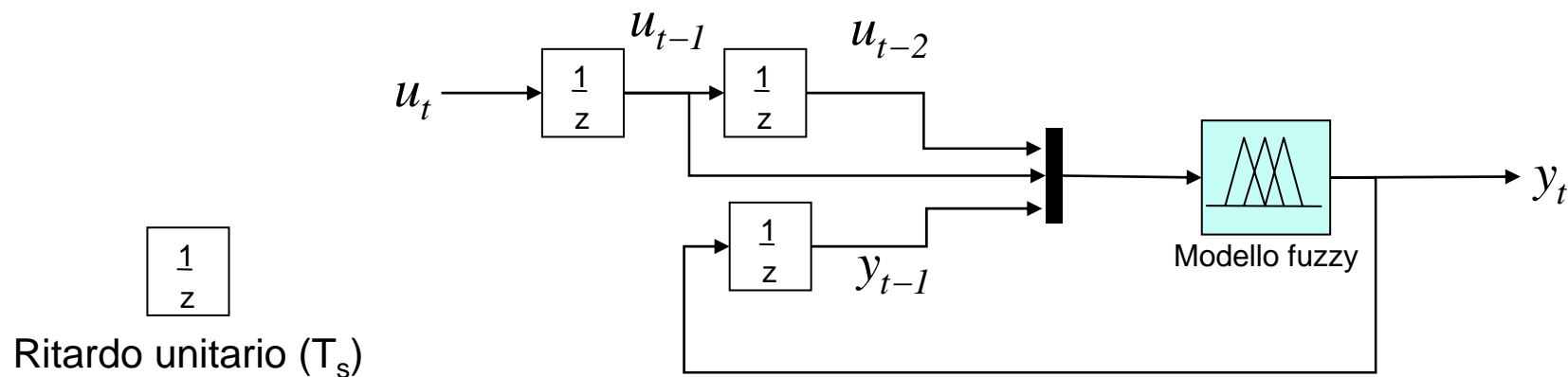
# Struttura del modello fuzzy a regole (Mamdani)

L'uso di ritardi unitari (pari a  $T_s$ ) è indispensabile per creare gli antecedenti passati, sia per gli ingressi che per le uscite

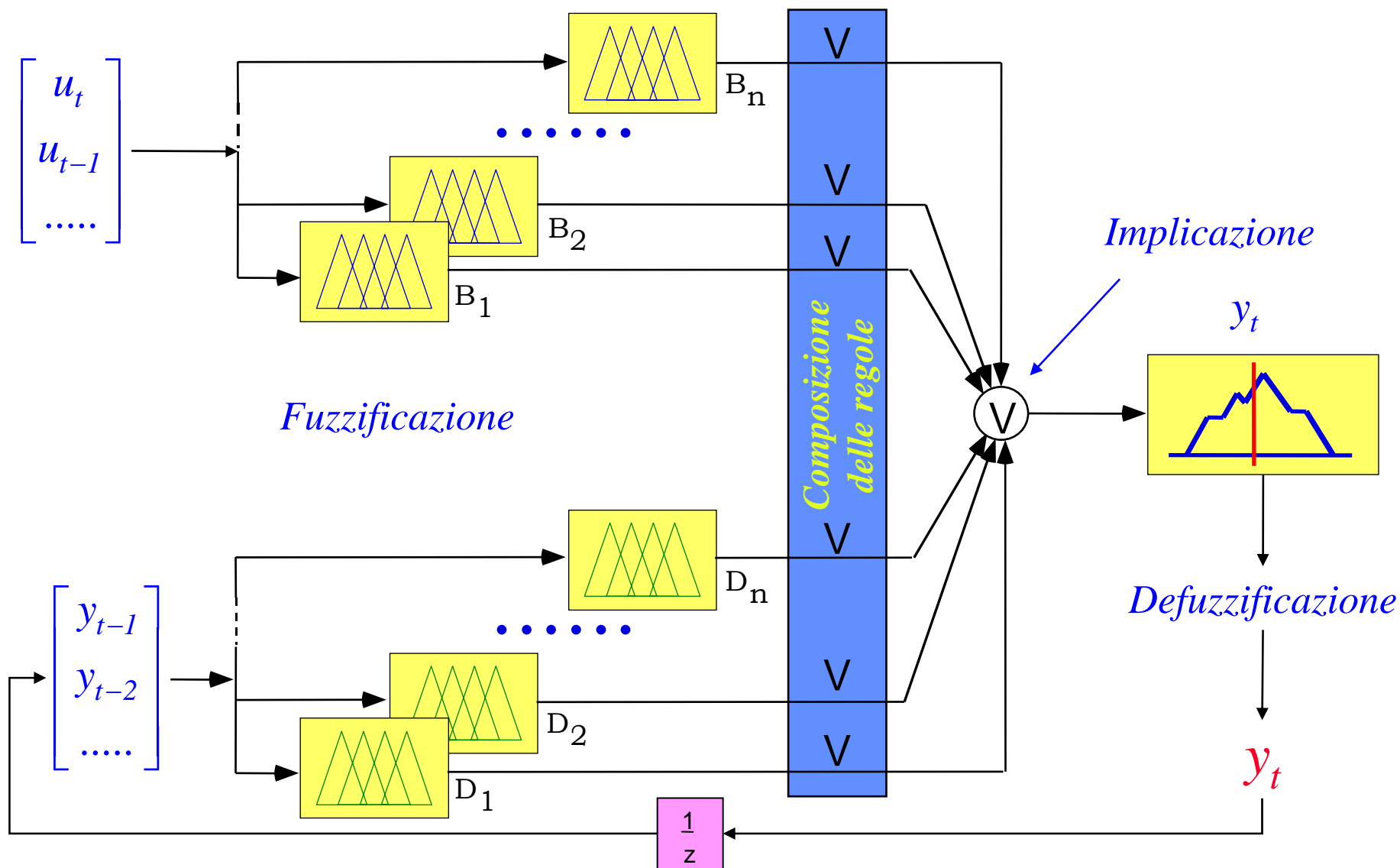


## Struttura ricorsiva del modello

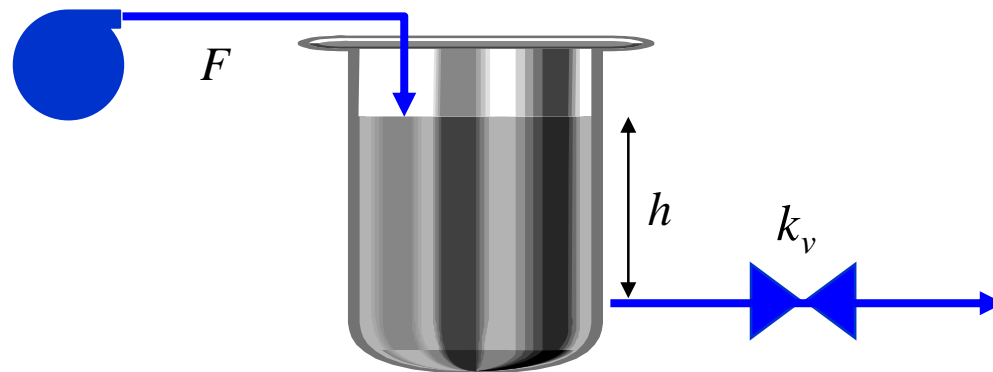
- Il calcolo dell'uscita attuale  $y_t$  utilizza uscite passate ( $y_{t-1}, y_{t-2}, \dots$ ) ed ingressi passati ( $u_{t-1}, u_{t-2}, \dots$ ). Perciò
  - Nel modello dovranno essere previsti dei ritardi unitari (di un passo) per conservare memoria degli elementi passati
  - Eventuali errori di modello nei passi precedenti si rifletteranno nel valore attuale (propagazione dell'errore)
- Dato che il modello fuzzy è naturalmente tempo-discreto, si dovrà scegliere con cura l'intervallo di tempo  $T_s$  fra campioni successivi dell'ingresso e dell'uscita
  - Un  $T_s$  troppo piccolo non farebbe risaltare le variazioni, rendendo inefficaci le regole
  - Un  $T_s$  troppo lungo farebbe perdere risoluzione al modello, non osservando variazioni rapide ma significative.



# Struttura generale del modello dinamico fuzzy di Mamdani

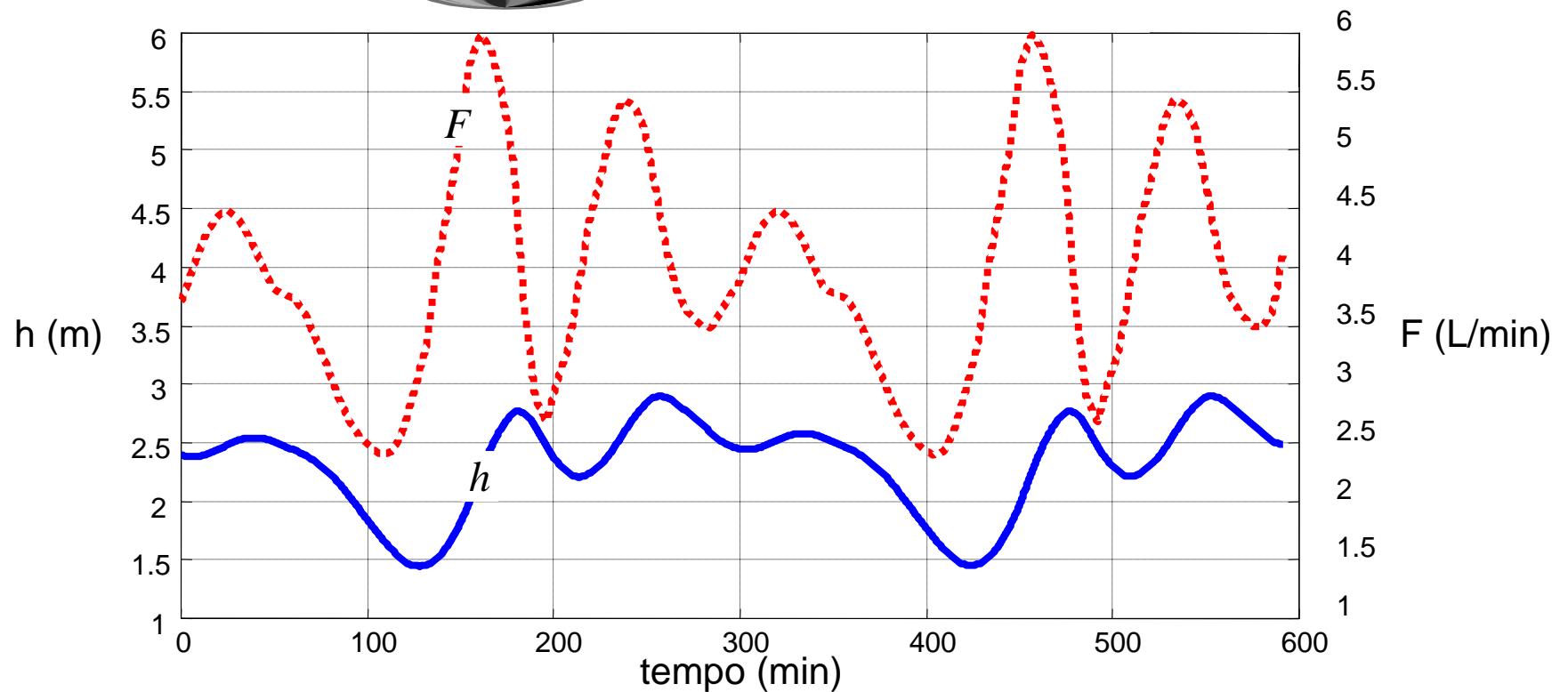


# Esempio: serbatoio con deflusso in pressione



Modello deterministico

$$A \frac{dh}{dt} = F - k_v \sqrt{h}$$



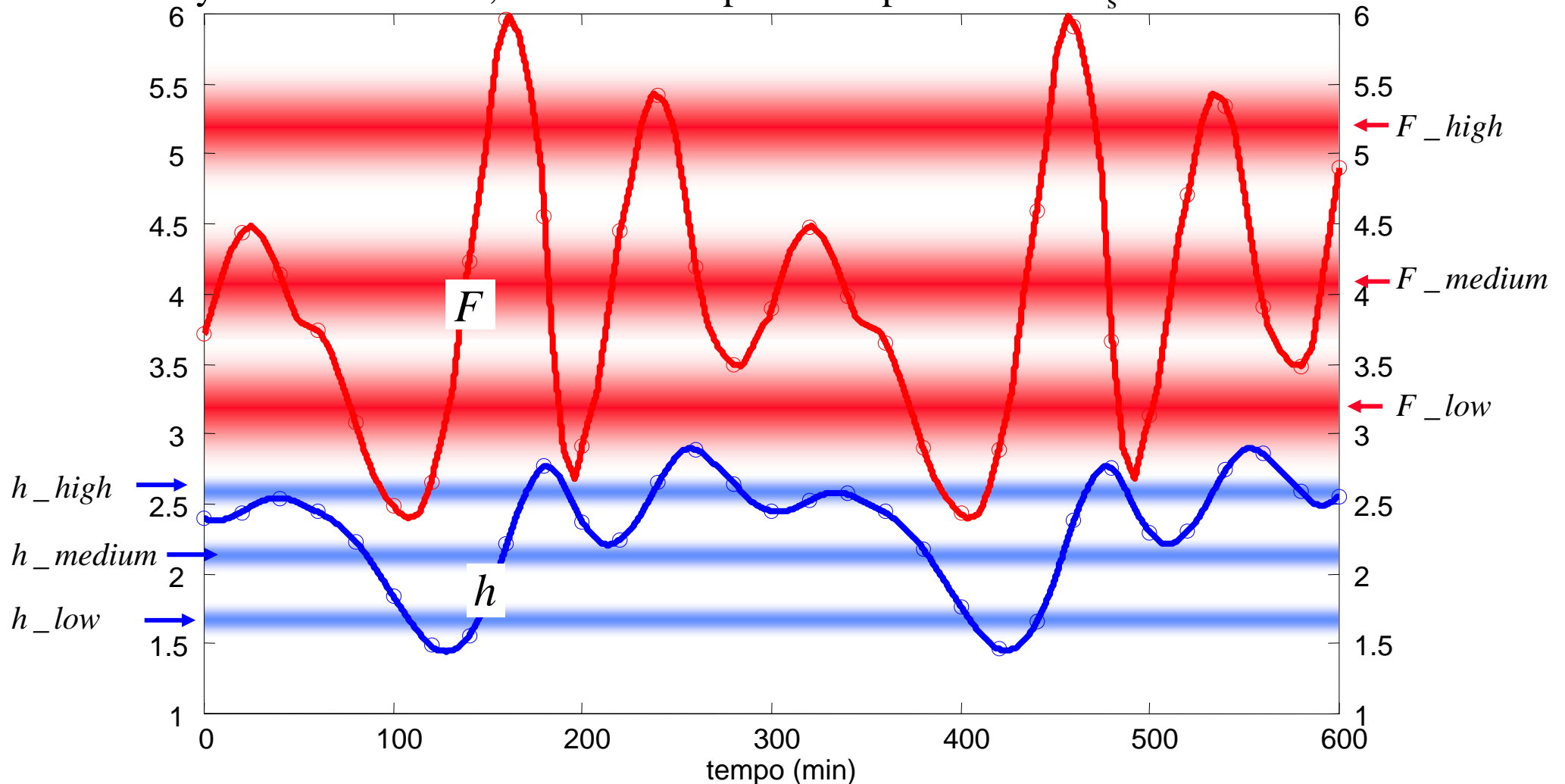
# Operazioni per la costruzione di un modello fuzzy a regole

---

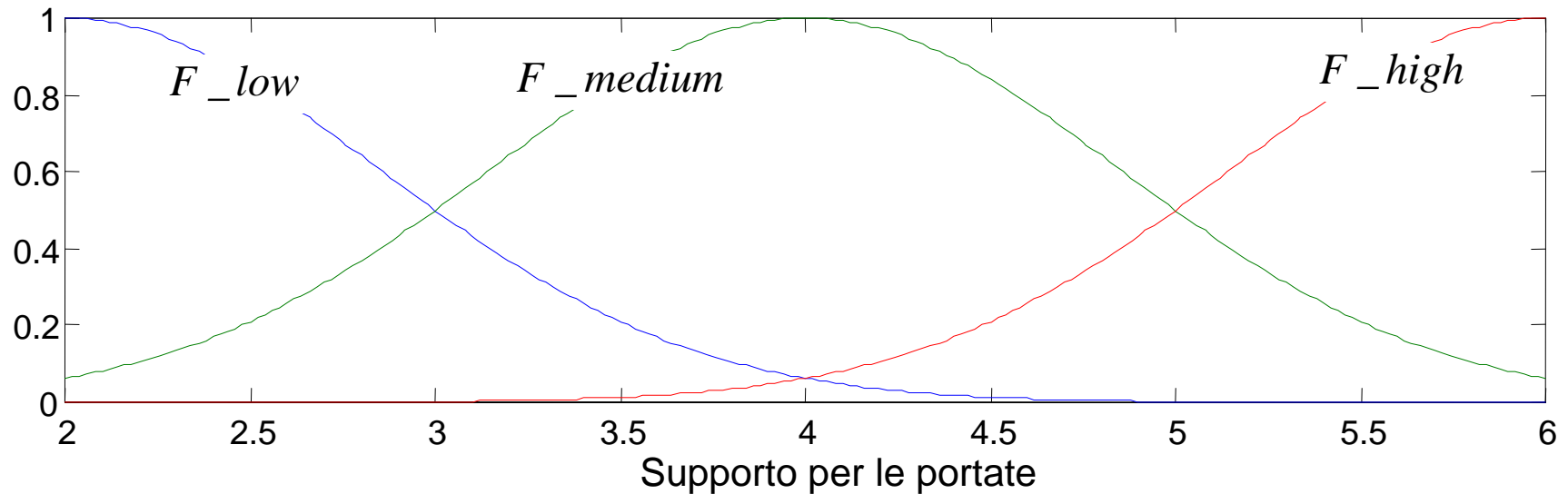
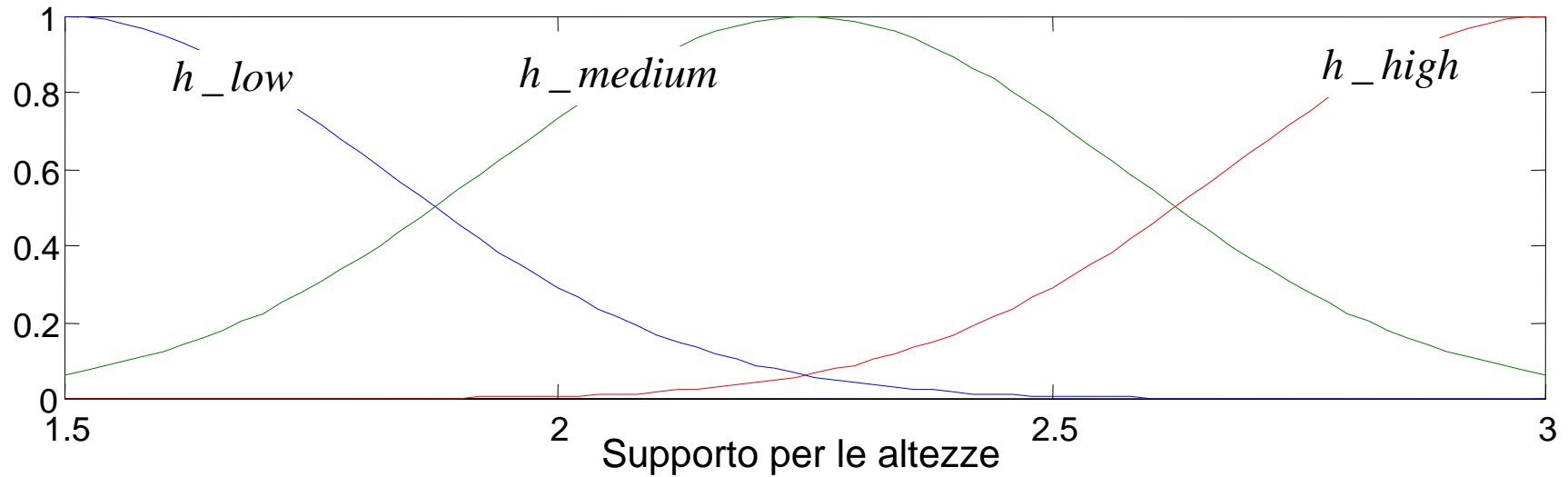
- ☞ Definire le fasce di valori “significativi” per le variabili per costruire i rappresentatori fuzzy
  - ⇒ Numero di rappresentatori
  - ⇒ Forma dei rappresentatori
- ☞ Definire matematicamente le funzioni di appartenenza
  - ⇒ Espressione matematica (triangolare, gaussiano, etc.)
  - ⇒ Parametri che ne determinano la forma rispetto ai dati
- ☞ Definire la struttura del modello e le regole di inferenza
  - ⇒ Per semplicità conviene scegliere un modello ingresso/uscita tempo-discreto del tipo
$$y_t = f( y_{t-1}, \dots, y_{t-n}, u_{t-k}, \dots, u_{t-n} )$$
con  $k$  ritardo ingresso-uscita
  - ⇒ In generale le regole possibili saranno  $n \times n-k$ , perciò è bene tenere basso l'ordine del modello.
  - ⇒ Ad es. 
$$y_t = f( y_{t-1}, y_{t-2}, u_{t-1}, u_{t-2} )$$
- ☞ Comporre le regole sulla base del comportamento osservato

# Operazioni preliminari alla modellazione

Individuazione di fasce di valori “significativi” per costruire i rappresentatori fuzzy delle le variabili, scelto un tempo di campionamento  $T_s = 20$  min.



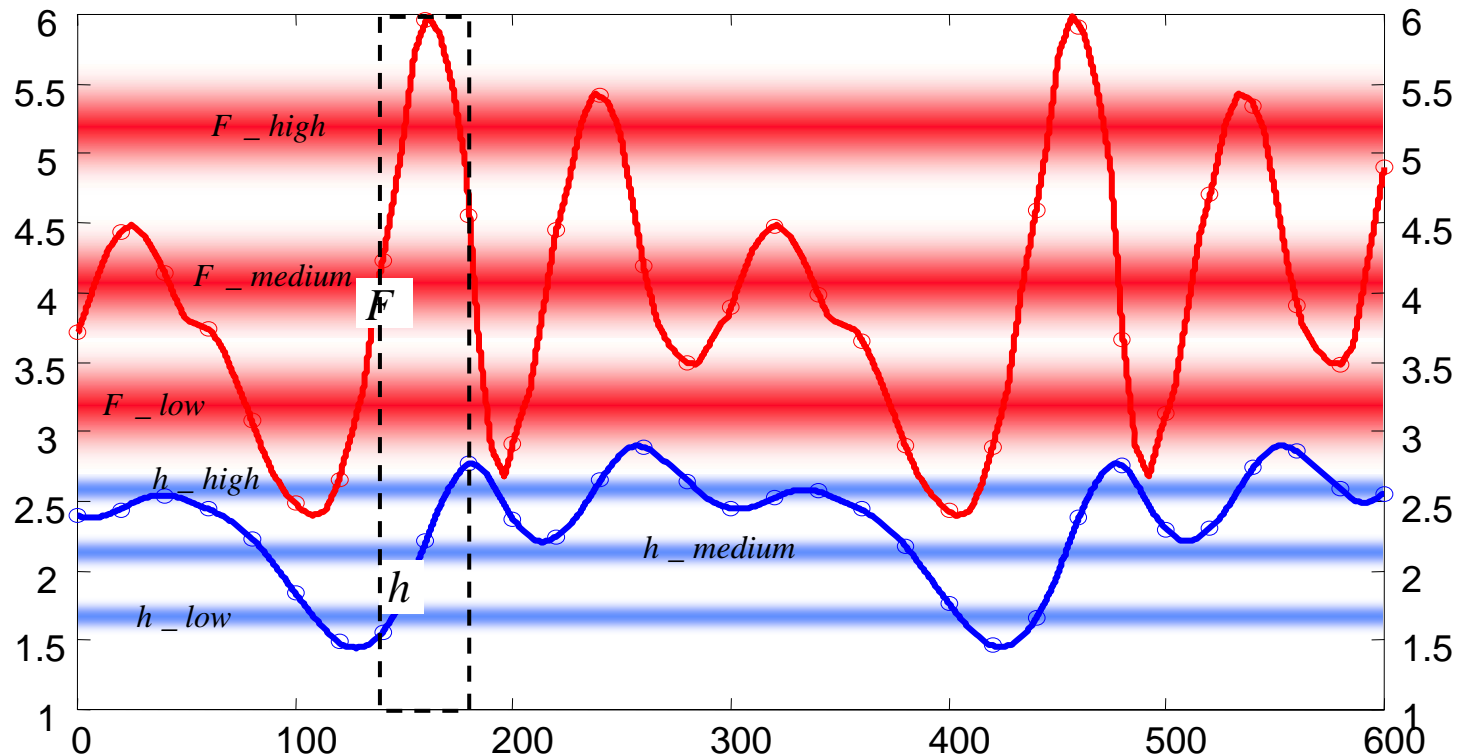
# Definizione delle funzioni di appartenenza



# Definizione delle regole

Usando una finestra mobile che copre l'intervallo di tempo pari alle variabili incluse nel modello, si cerca di inferire le regole di variazione. Ad es. nel rettangolo tratteggiato:

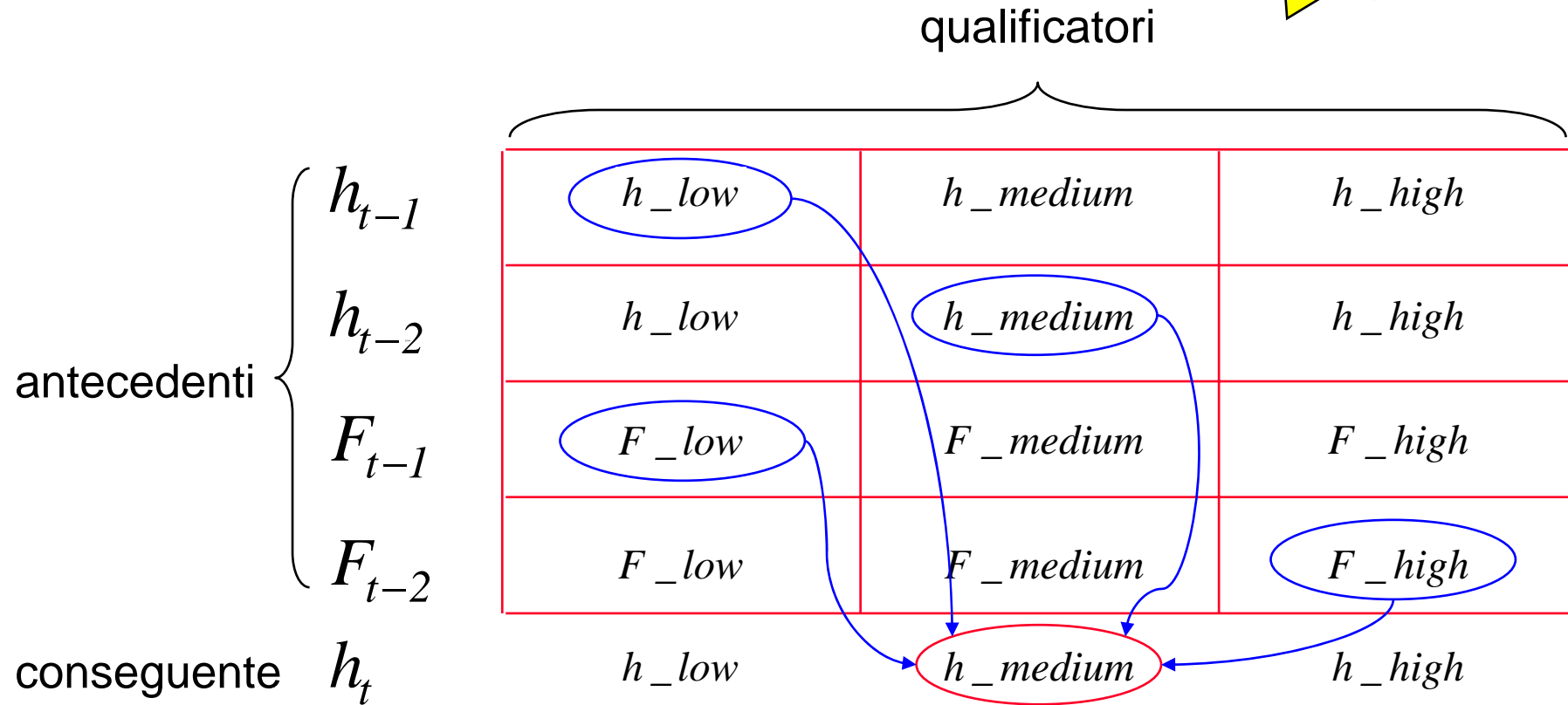
*IF  $h_{t-1}$  is  $h\_medium$  AND  $h_{t-2}$  is  $h\_low$  AND  $F_{t-1}$  is  $F\_high$  AND  $F_{t-2}$  is  $F\_medium$   
THEN  $h_t$  is  $h\_high$*



# Combinazioni delle regole

Numero totale di regole possibili  $(N_{ant} + N_{cons}) \times N_{qualif}$

Il numero di regole può rapidamente diventare



# Regole per il sistema Serbatoio

Numero totale di regole possibili:  $N_{variabili} \times N_{qualificatori}$   
In questo caso  $5 \times 3 = 15$ . Ne sono state utilizzate solo 8 !

*IF* ( $h_{t-1}$  is  $h\_high$ ) *AND* ( $h_{t-2}$  is  $h\_low$ ) *AND* ( $F_{t-1}$  is  $F\_high$ ) *AND* ( $F_{t-2}$  is  $F\_medium$ ) *THEN* ( $h_t$  is  $h\_high$ )

*IF* ( $h_{t-1}$  is  $h\_medium$ ) *AND* ( $h_{t-2}$  is  $h\_medium$ ) *AND* ( $F_{t-1}$  is  $F\_high$ ) *AND* ( $F_{t-2}$  is  $F\_low$ ) *THEN* ( $h_t$  is  $h\_high$ )

*IF* ( $h_{t-1}$  is  $h\_low$ ) *AND* ( $h_{t-2}$  is  $h\_low$ ) *AND* ( $F_{t-1}$  is  $F\_high$ ) *AND* ( $F_{t-2}$  is  $F\_low$ ) *THEN* ( $h_t$  is  $h\_low$ )

*IF* ( $h_{t-1}$  is  $h\_medium$ ) *AND* ( $h_{t-2}$  is  $h\_low$ ) *AND* ( $F_{t-1}$  is  $F\_high$ ) *AND* ( $F_{t-2}$  is  $F\_medium$ ) *THEN* ( $h_t$  is  $h\_high$ )

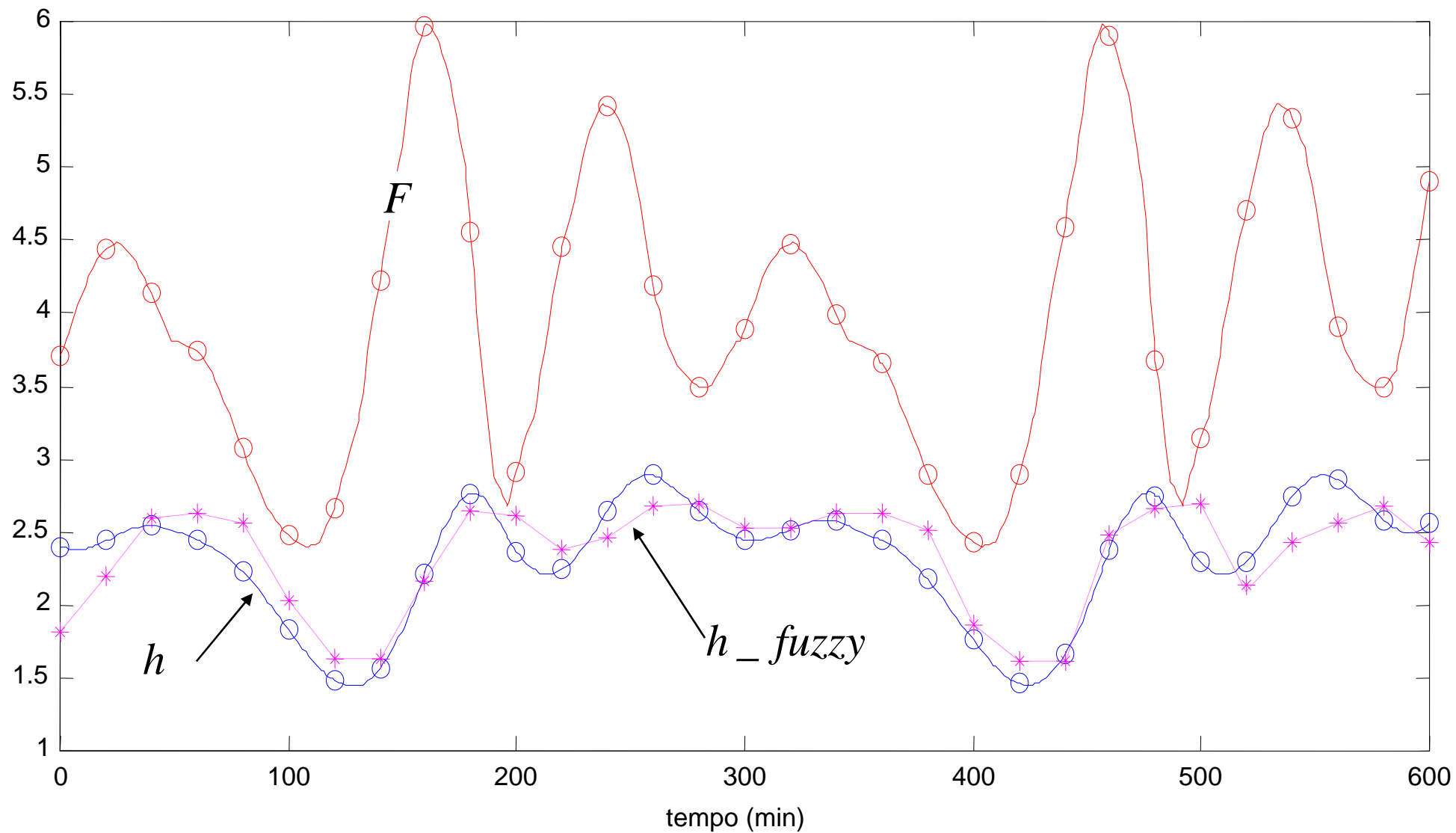
*IF* ( $h_{t-1}$  is  $h\_medium$ ) *AND* ( $h_{t-2}$  is  $h\_medium$ ) *AND* ( $F_{t-1}$  is  $F\_medium$ ) *AND* ( $F_{t-2}$  is  $F\_low$ ) *THEN* ( $h_t$  is  $h\_low$ )

*IF* ( $h_{t-1}$  is  $h\_high$ ) *AND* ( $h_{t-2}$  is  $h\_high$ ) *AND* ( $F_{t-1}$  is  $F\_medium$ ) *AND* ( $F_{t-2}$  is  $F\_high$ ) *THEN* ( $h_t$  is  $h\_high$ )

*IF* ( $h_{t-1}$  is  $h\_high$ ) *AND* ( $h_{t-2}$  is  $h\_high$ ) *AND* ( $F_{t-1}$  is  $F\_high$ ) *AND* ( $F_{t-2}$  is  $F\_medium$ ) *THEN* ( $h_t$  is  $h\_high$ )

*IF* ( $h_{t-1}$  is  $h\_medium$ ) *AND* ( $h_{t-2}$  is  $h\_high$ ) *AND* ( $F_{t-1}$  is  $F\_low$ ) *AND* ( $F_{t-2}$  is  $F\_medium$ ) *THEN* ( $h_t$  is  $h\_low$ )

# Paragone fra sistema e modello fuzzy



# Il sistema inferenziale di Matlab Fuzzy Toolbox

---



# FIS: Fuzzy Inference System

The screenshot shows the FIS Editor window for a system named 'tipper'. The interface includes a menu bar (File, Edit, View), a workspace with icons for input variables 'speed' and 'load', a central rule editor box, and an output variable icon 'tip'. Below the workspace is a configuration panel with dropdown menus for inference methods and a section for defining variables.

**These menu items allow you to save, open, or edit a fuzzy system using any of the five basic GUI tools.**

**Double-click on an input variable icon to open the Membership Function Editor.**

**Double-click on the system diagram to open the Rule Editor.**

**The name of the system is displayed here. It can be changed using one of the Save as... menu options.**

**Double-click on the icon for the output variable, tip, to open the Membership Function Editor.**

**These pop-up menus are used to adjust the fuzzy inference functions, such as the defuzzification method.**

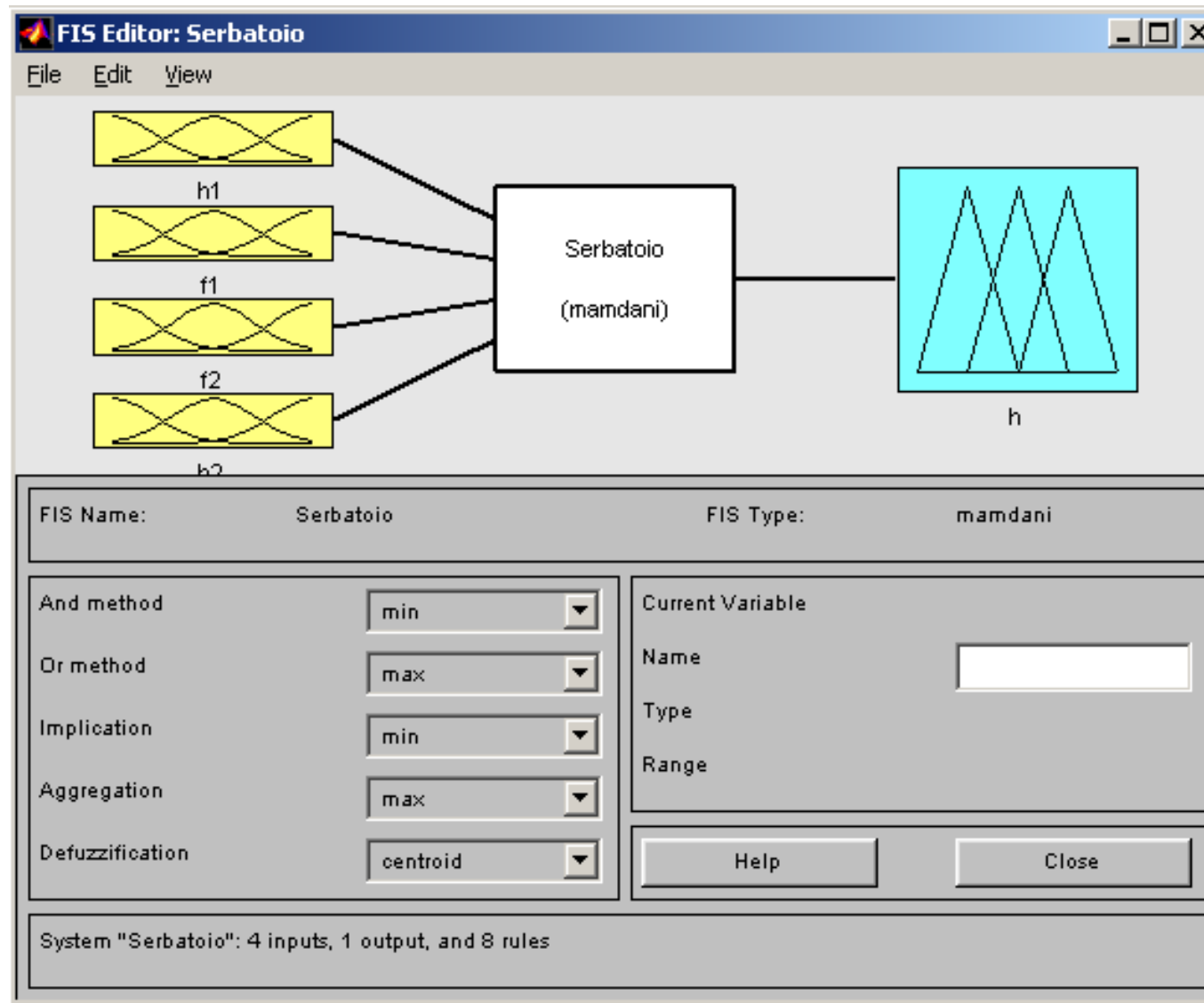
**This edit field is used to name and edit the names of the input and output variable.**

**This status line describes the most recent operation.**

FIS Name:	tipper	FIS Type:	mamdani
And method:	min	Current Variable:	
Or method:	max	Name:	
Implication:	min	Type:	
Aggregator:	max	Range:	
Defuzzification:	centroid	Help	Close

System "tipper" (input: 1 output: 1 and 1 rules)

# Matlab Fuzzy Toolbox: Fuzzy Inference System Editor



# Membership Function Editor

These menu items allow you to save, open, or edit a fuzzy system using any of the five basic GUI tools.

This is the "Variable Palette" area. Click on a variable here to make it current and edit its membership functions.

This graph field displays all the membership functions of the current variable.

Click on a line to select it and you can change any of its attributes, including name, type and numerical parameters. Drag your mouse to move or change the shape of a selected membership function.

These text fields display the name and type of the current variable.

This edit field lets you set the range of the current variable.

This edit field lets you set the display range of the current plot.

This status line describes the most recent operation.

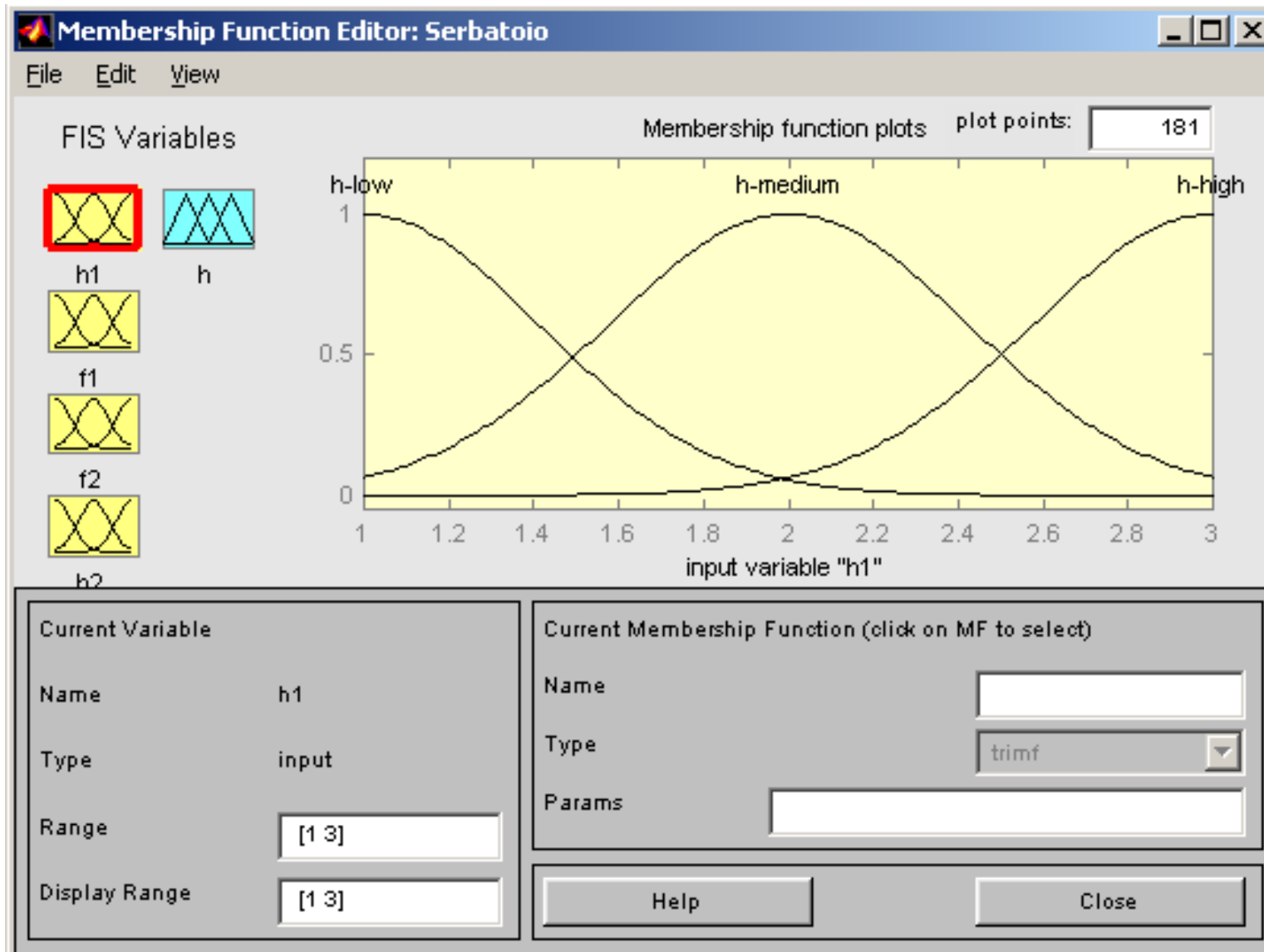
The screenshot shows the Membership Function Editor window. At the top left is a menu bar with 'File', 'Edit', and 'View'. Below it is the 'FIS Variables' palette containing icons for 'service', 'type', and 'load'. The main area is a graph titled 'membership function: good' with a 'plot points' field set to '181'. The graph shows three curves: 'poor' (green), 'good' (red), and 'good/ant' (black). Below the graph are two panels: 'Current Variable' and 'Current Membership Function (Click on MF to select)'. The 'Current Variable' panel has fields for Name (service), Type (type), Range ([0 1]), and Display Range ([0 1]). The 'Current Membership Function' panel has fields for Name (good), Type (gaussian), and Param ([1.5 6]). At the bottom is a 'Ready' status bar.

This edit field lets you change the name of the current membership function.

This pop-up menu lets you change the type of the current membership function.

This edit field lets you change the numerical parameters for the current membership function.

# Matlab Fuzzy Toolbox: Editor dei rappresentanti



# Rule Editor

The screenshot shows the 'Rule Editor: tipper' window. At the top, a menu bar contains 'File', 'Edit', 'View', and 'Options'. Below the menu is a list of three rules:

1. If (service is good) or (food is rancid) then (tip is cheap) (1)
2. If (service is good) then (tip is average) (1)
3. If (service is excellent) or (food is delicious) then (tip is generous) (1)

The main editing area is divided into 'If' and 'Then' sections. The 'If' section has two columns: 'service is' and 'food is'. The 'Then' section has one column: 'tip is'. Each column contains a list of fuzzy membership functions: 'poor', 'good', 'excellent', 'none' for service; 'rancid', 'delicious', 'none' for food; and 'cheap', 'average', 'generous', 'none' for tip. Below these lists are checkboxes for 'not' and radio buttons for 'Connection' (set to 'or'). A 'Weight' field is set to '1'. At the bottom, there are buttons for 'Delete rule', 'Add rule', 'Change rule', 'Help', and 'Close'. A status bar at the very bottom shows 'FIS Name: tipper'.

Callout boxes provide the following explanations:

- Input or output selection menus.** Points to the fuzzy membership function lists.
- The menu items allow you to save, open, or edit a fuzzy system using any of the five basic GUI tools.** Points to the File, Edit, View, and Options menu bar.
- The rules are entered automatically using the GUI tools.** Points to the list of rules.
- Link input statements in rules.** Points to the 'or' connection radio button.
- This status line describes the most recent operation.** Points to the 'FIS Name: tipper' status bar.
- Negate input or output statements in rules.** Points to the 'not' checkboxes.
- Create or edit rules with the GUI buttons and choices from the input or output selection menus.** Points to the 'Add rule', 'Change rule', and 'Delete rule' buttons.
- The Help button gives some information about how the Rule Editor works, and the Close button closes the window.** Points to the 'Help' and 'Close' buttons.

# Rule Viewer

The screenshot shows the 'Rule Viewer' window with a menu bar (File, Edit, View, Options), a status bar at the top, and a main plotting area. The plotting area is divided into three columns: 'Sensor - S', 'Temperature - T', and 'Speed - E'. Each column contains three rows of plots, one for each rule. The first two columns have yellow plots, while the third has blue plots. A vertical red line is present in each plot, representing the defuzzified value. At the bottom, there is an input field for 'Input' (set to '15'), a 'Fuzz output' field (set to '70'), and several control buttons: 'Move' (left, right, down, up), 'Help', and 'Close'. A status line at the bottom left reads 'Opened system (open, 3 rules)'.

The menu items allow you to save, open, or edit a fuzzy system using any of the five basic GUI tools.

Each column of plots (yellow) shows how the input variable is used in the rules. The input values are shown here at the top.

This column of plots (blue) shows how the output variable is used in the rules.

Each row of plots represents one rule (here there are 3). Click on a rule to display it in the status bar.

Slide this line to change your input values, and generate a new output response.

This edit field allows you to set the input explicitly.

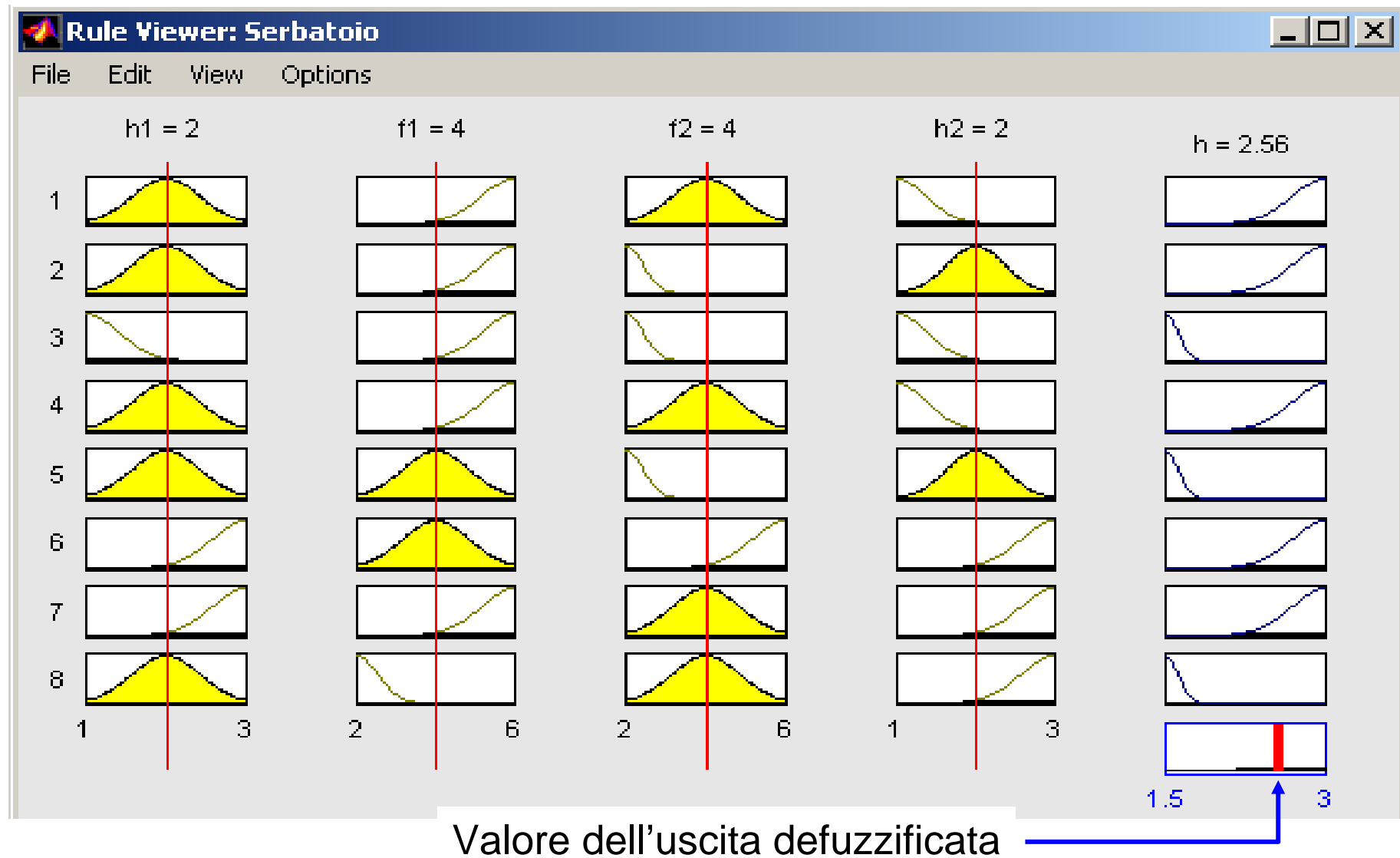
This status line describes the most recent operation.

This line provides a defuzzified value.

The bottom-right plot shows how the output of each rule is combined to make an aggregate output and then defuzzified.

Shift the plots left, right, up, or down with these buttons.

# Matlab Fuzzy Toolbox: visualizzazione delle regole



# Surface Viewer

The menu items allow you to save, open, or edit a fuzzy system using any of the five basic GUI tools.

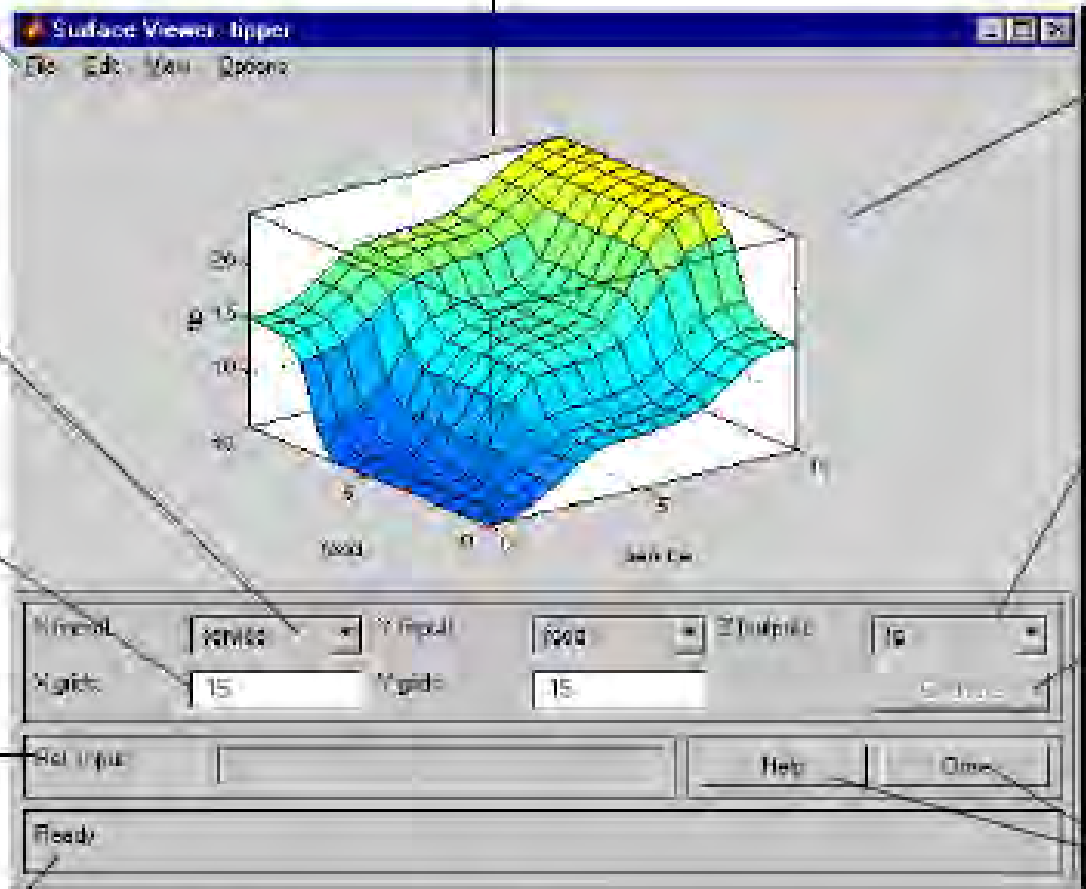
These pop-up menus let you specify the one or two displayed input variables.

These edit fields let you determine how densely to grid the input space.

This edit field lets you set the input explicitly for inputs not specified in the surface plot.

This status line describes the most recent operation.

Use mouse to rotate the axes.



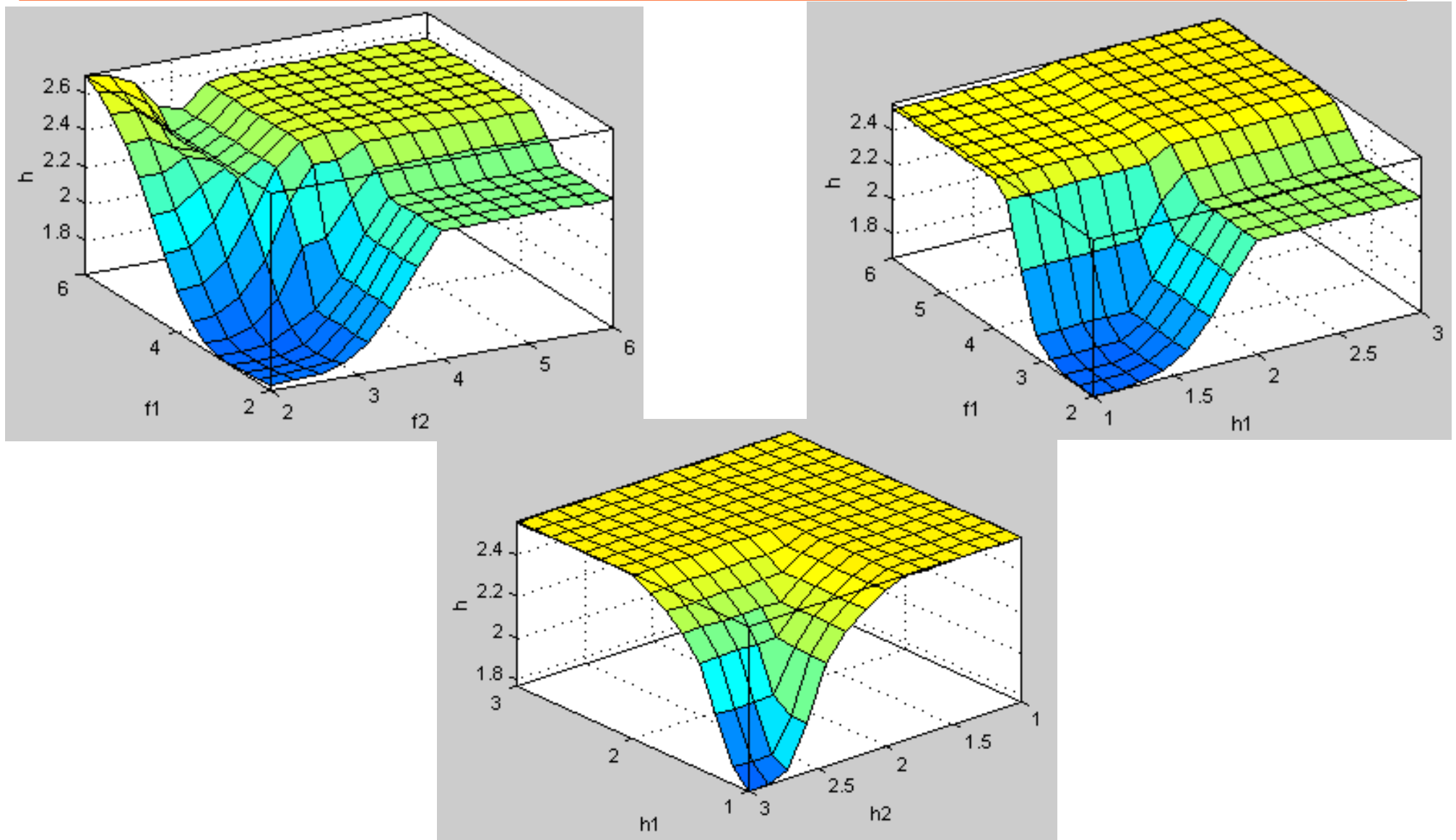
This plot shows the output surface for any output of the system versus any one or two inputs to the system

This pop-up menu lets you specify the displayed output variable.

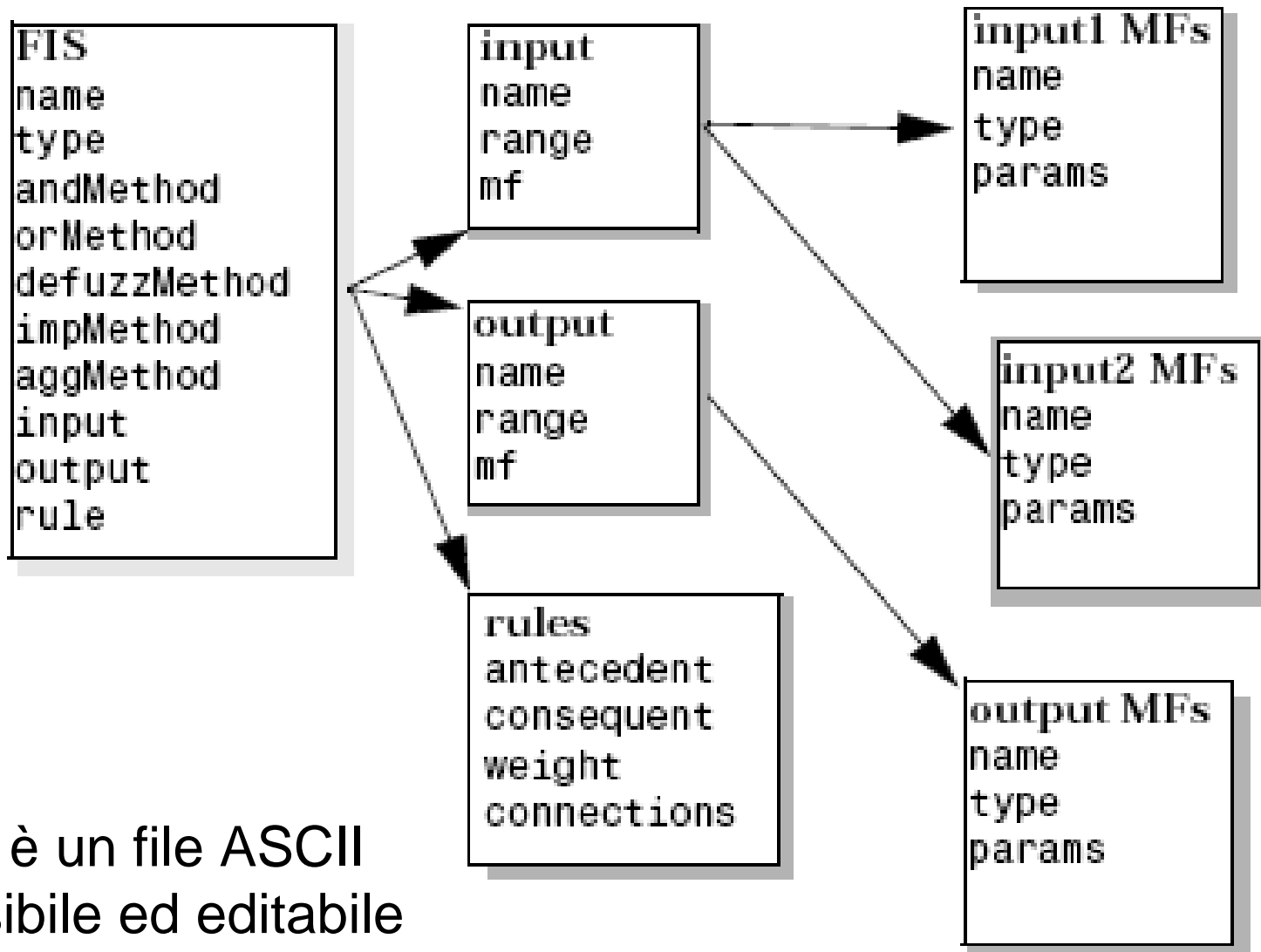
Push this button when you're ready to calculate and plot.

The Help button gives some information about how the Surface Viewer works, and the Close button closes the window.

# Matlab Fuzzy Toolbox: viste della superficie di controllo

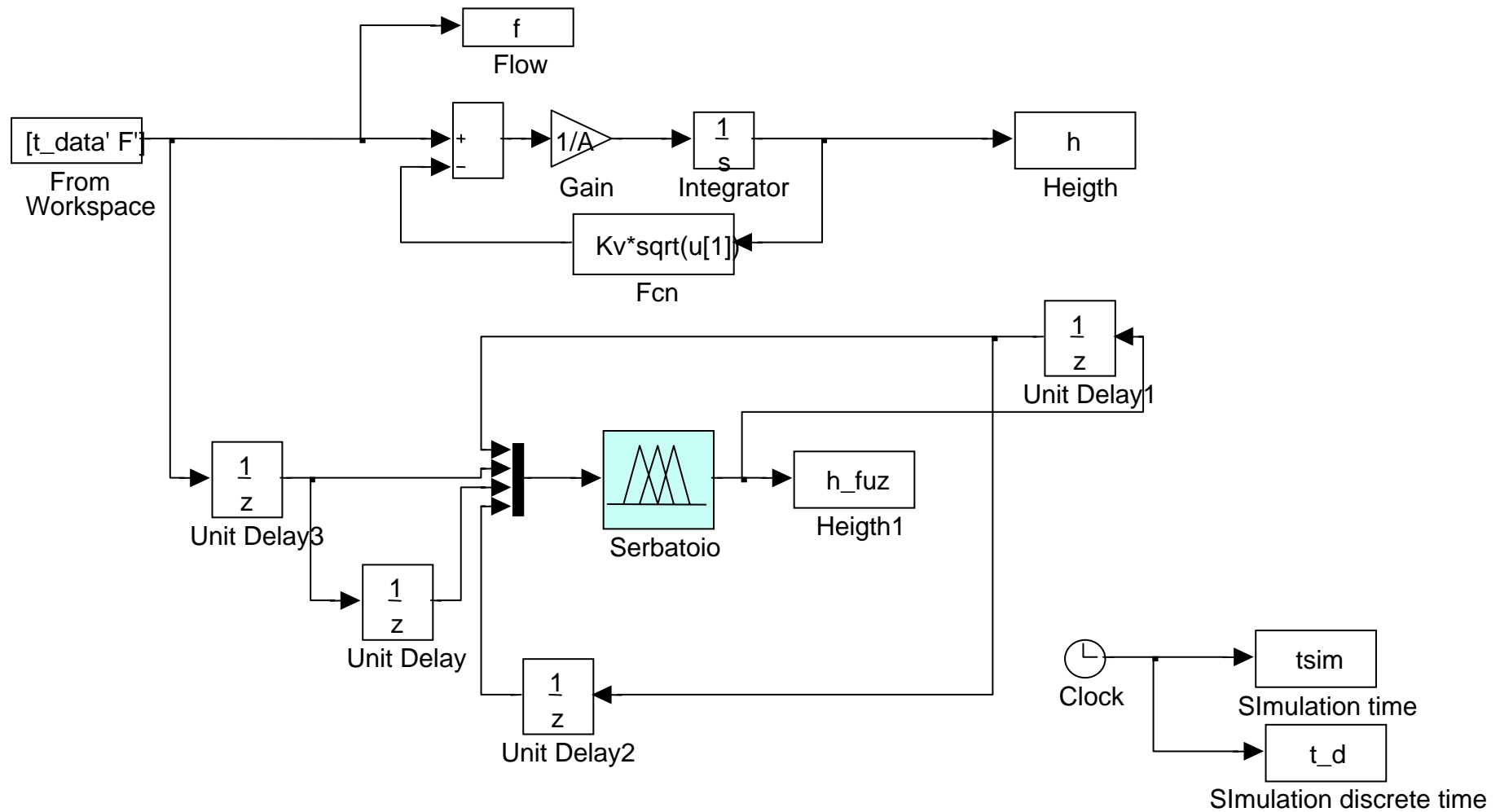


# Struttura della FIS



La FIS è un file ASCII  
accessibile ed editabile

# Inserimento della FIS in un diagramma Simulink








# Valutazione dell'esercizio

---

## **Vantaggi**

- ⇒ Non è necessario conoscere la struttura del sistema né il suo funzionamento interno
- ⇒ Il modello è stato ricavato dall'osservazione diretta dei dati

## **Svantaggi**

- ⇒ Nessuna guida nella definizione delle caratteristiche del modello
  -  Ordine
  -  Numero e forma dei qualificatori
  -  Numero e contenuto delle regole
- ⇒ Il modello ha validità limitata agli intervalli dei dati utilizzati per la sua definizione
  -  Es. se arrivasse un ingresso di portata  $F = 10$  ?
  -  Il modello non potrebbe applicarvi nessuna regola perché questo valore è esterno agli intervalli entro cui sono definite le sue variabili.

## **Domanda:** esistono metodi più sistematici?

- ⇒ **Risposta:** Sì

# Modelli di Sugeno

☞ Nella generica regola il conseguente è un *singleton*

$$IF \ x \text{ is } A_i \ THEN \ y = b_i$$

☞ Ciò implica che ogni regola produce un'uscita pari al valore del singleton ( $b_i$ ) moltiplicato per il grado di verità dell'antecedente  $\mu_i$  perciò

$$IF \ x \text{ is } A_i \ THEN \ y = b_i \Rightarrow \mu_i b_i$$

☞ La composizione delle regole e la defuzzificazione diventano un'unica operazione che di fatto è la media pesata dei singleton mediante i rispettivi gradi di verità

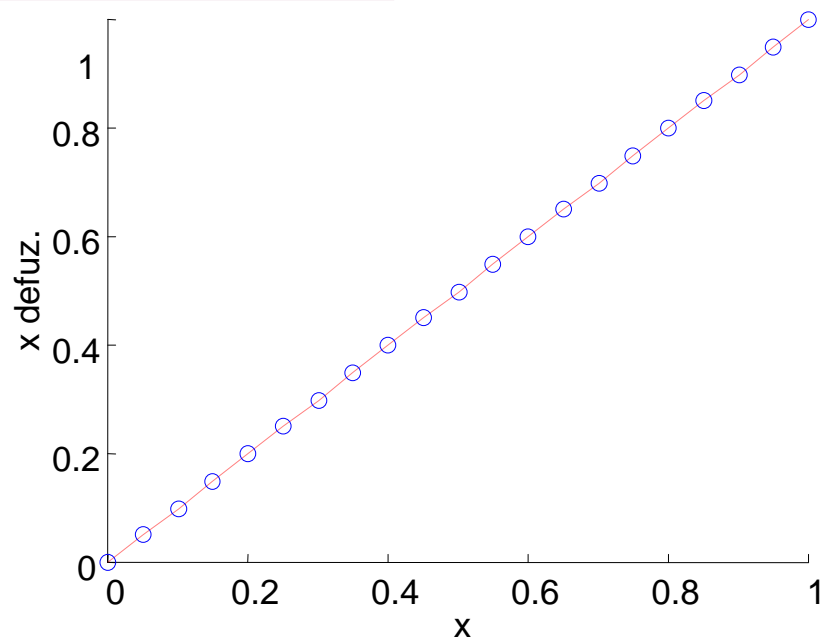
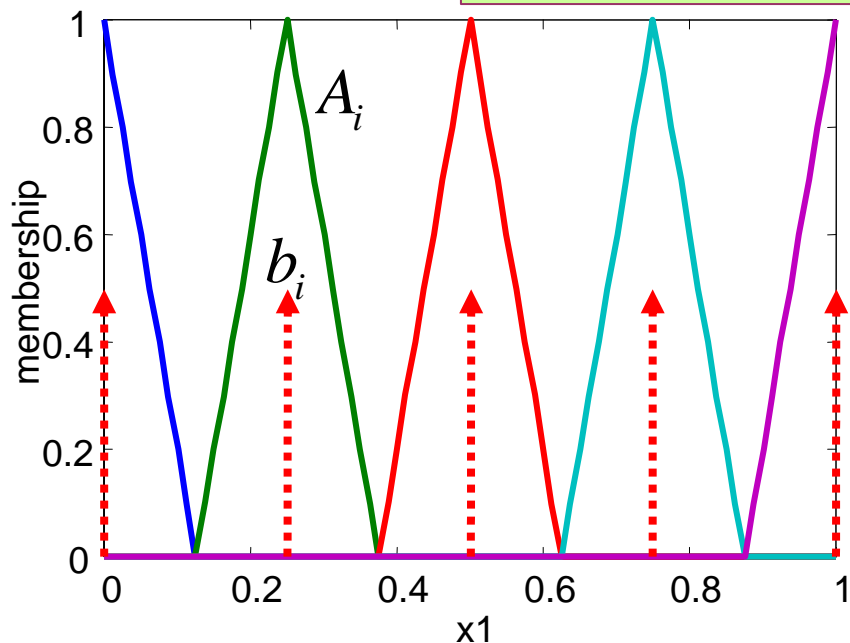
$$y_{defuz} = \frac{\sum_{i=1}^N b_i \mu_i}{\sum_{i=1}^N \mu_i}$$

☞ **Estensione:**  
al posto dei singleton si possono usare delle *funzioni*, aumentando le capacità approssimanti del modello

$$y_{defuz} = \frac{\sum_{i=1}^N \Phi_i(x) \mu_i}{\sum_{i=1}^N \mu_i}$$

# Errore di defuzzificazione in Sugeno

$R_i : \text{if } x \text{ is } A_i \text{ then } y = b_i$



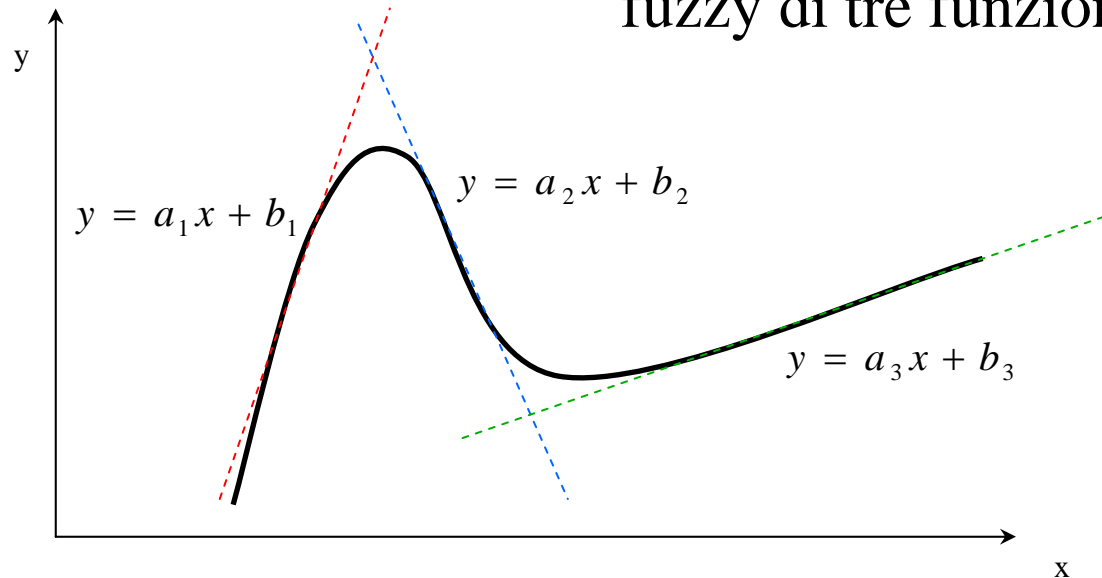
Anche utilizzando qualificatori non ottimali (intersezione al 50%) si ottiene ugualmente una defuzzificazione senza errore dovuto alla fuzzificazione

***Defuzzificazione: media dei conseguenti pesata con i gradi di attivazione***

$$y = y_{defuz} = \frac{\sum_{i=1}^M b_i \times \mu_i}{\sum_{i=1}^M \mu_i}$$

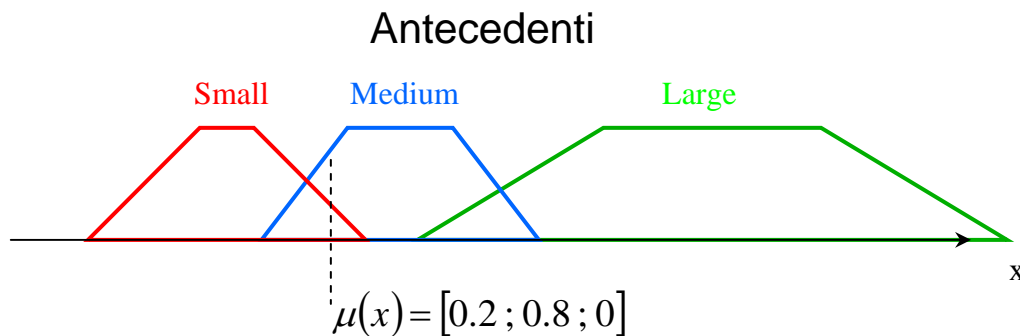
# Approssimazione di funzione con un modello di Sugeno

Modello di Sugeno come linearizzazione locale  
 La funzione viene approssimata mediante combinazione fuzzy di tre funzioni lineari



$$\mu_A(x): \mathbb{R} \rightarrow [0,1]$$

$$A \in \{Small, Medium, Large\}$$



## Regole

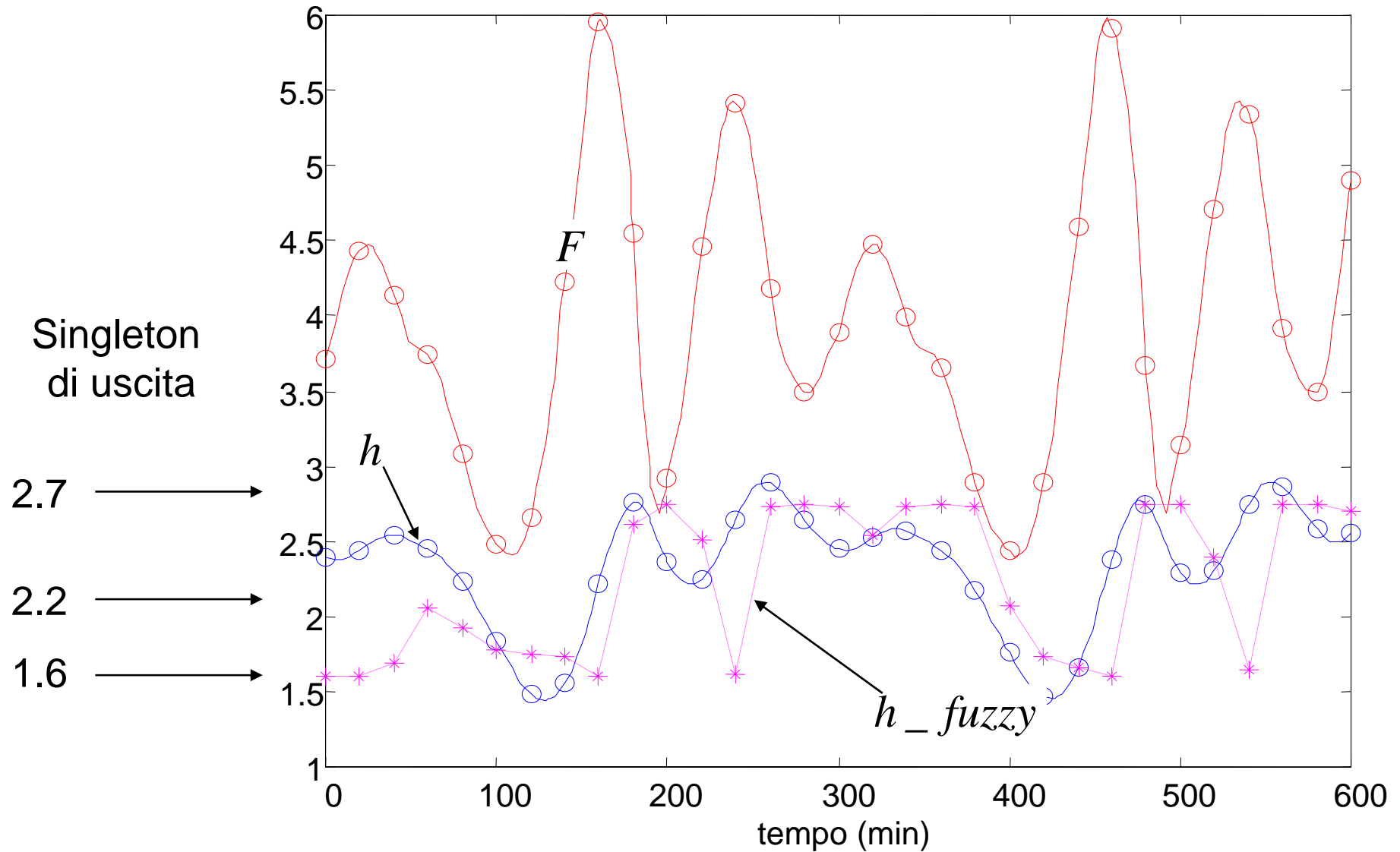
- IF** *x is Small* **THEN**  $y = a_1x + b_1$
- IF** *x is Medium* **THEN**  $y = a_2x + b_2$
- IF** *x is Large* **THEN**  $y = a_3x + b_3$

# Modellazione del serbatoio con Sugeno

---

1. If (h1 is h-medium) and (h2 is h-low) and (f1 is f-high) and (f2 is f-medium) then (h is h-high)
2. If (h1 is h-medium) and (h2 is h-medium) and (f1 is f-high) and (f2 is f-low) then (h is h-high)
3. If (h1 is h-low) and (h2 is h-low) and (f1 is f-medium) and (f2 is f-low) then (h is h-low)
4. If (h1 is h-medium) and (h2 is h-low) and (f1 is f-high) and (f2 is f-medium) then (h is h-high)
5. If (h1 is h-medium) and (h2 is h-medium) and (f1 is f-medium) and (f2 is f-low) then (h is h-medium)
6. If (h1 is h-high) and (h2 is h-high) and (f1 is f-medium) and (f2 is f-high) then (h is h-high)
7. If (h1 is h-high) and (h2 is h-high) and (f1 is f-high) and (f2 is f-medium) then (h is h-high)
8. If (h1 is h-medium) and (h2 is h-high) and (f1 is f-low) and (f2 is f-medium) then (h is h-low)
9. If (h1 is h-medium) and (h2 is h-medium) and (f1 is f-medium) and (f2 is f-medium) then (h is h-medium)
10. If (h1 is h-low) and (h2 is h-low) and (f1 is f-low) and (f2 is f-low) then (h is h-low)
11. If (h1 is h-high) and (h2 is h-high) and (f1 is f-high) and (f2 is f-high) then (h is h-high)
12. If (h1 is h-medium) and (h2 is h-medium) and (f1 is f-medium) and (f2 is f-low) then (h is h-medium)
13. If (h1 is h-high) and (h2 is h-medium) and (f1 is f-medium) and (f2 is f-high) then (h is h-medium)

# Modellazione Sugeno del serbatoio



# Valutazione del modello di Sugeno

---

☞ Apparentemente le prestazioni sono peggiori, ma.....

⇒ Non si è sfruttata la flessibilità dei conseguenti

✎ Si poteva usare dei conseguenti più complessi (es. combinazioni lineari degli antecedenti o funzioni nonlineari)

✎ Si poteva calibrare tali funzioni con un criterio di minimizzazione degli errori rispetto ai dati

⇒ L'insieme delle regole potrebbe non essere completo né appropriato

☞ In realtà l'approccio di Sugeno è oggi l'unico utilizzato, ma va accoppiato a metodi di calibrazione molto potenti

⇒ Stima a minimi quadrati dei conseguenti

⇒ Clusterizzazione degli antecedenti

# Esempio di Fuzzy Learning: ANFIS

---

☞ ANFIS = Adaptive Neuro-Fuzzy Inference System

☞ Struttura di ANFIS: Sugeno con conseguenti lineari

$R_i$  : *If  $x$  is  $A_i$  and  $y$  is  $B_i$  then  $f_i = p_i x + q_i y + r_i$*

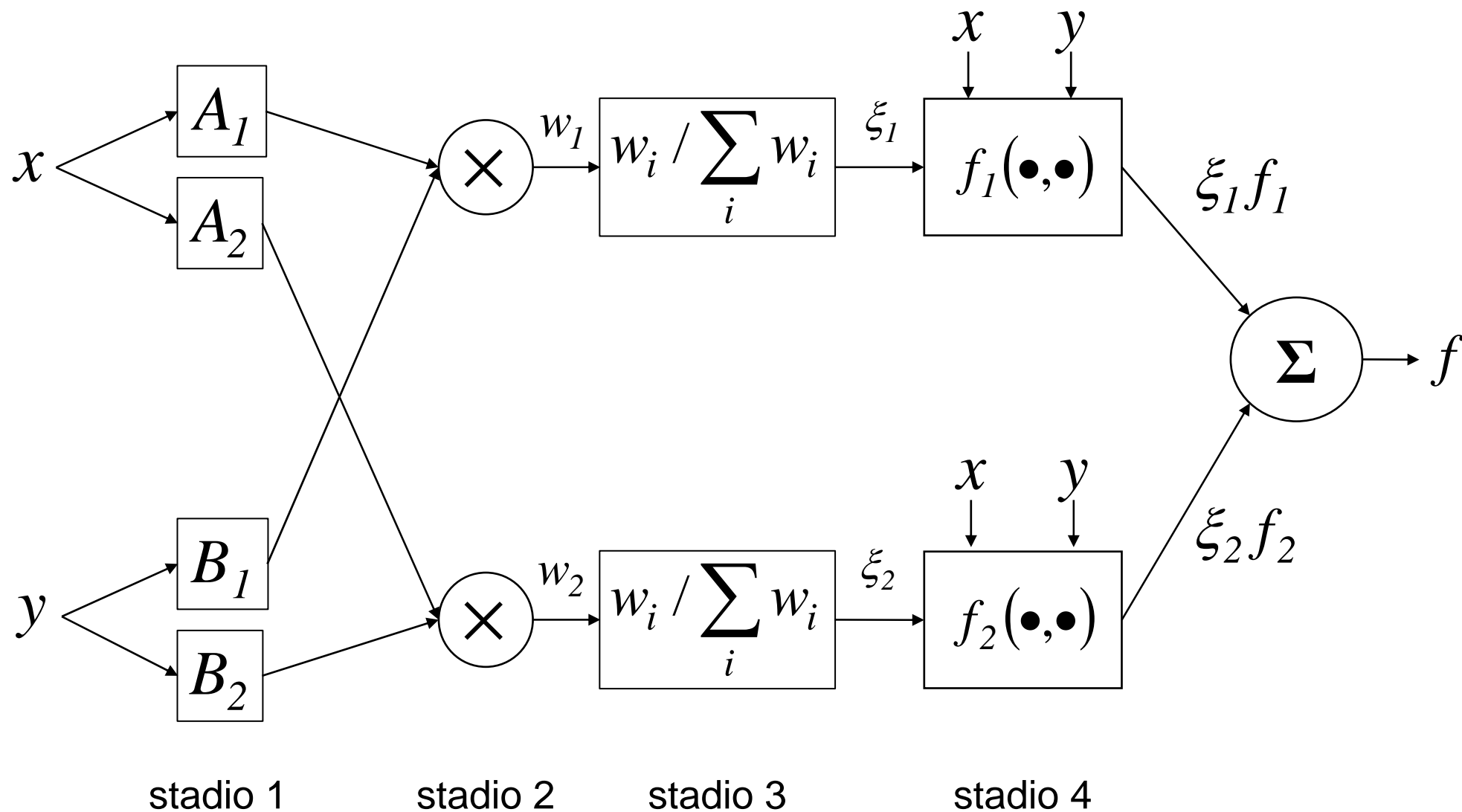
☞ Valutando le regole con una T-norma prodotto la membership del conseguente  $i$ -esimo è

$$w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y)$$

☞ La funzione complessiva è data da

$$f = \frac{\sum_i w_i f_i}{\sum_i w_i} = \sum_i \xi_i f_i \quad \text{con} \quad \xi_i = \frac{w_i}{\sum_i w_i}$$

## Struttura di ANFIS a due regole (esempio)



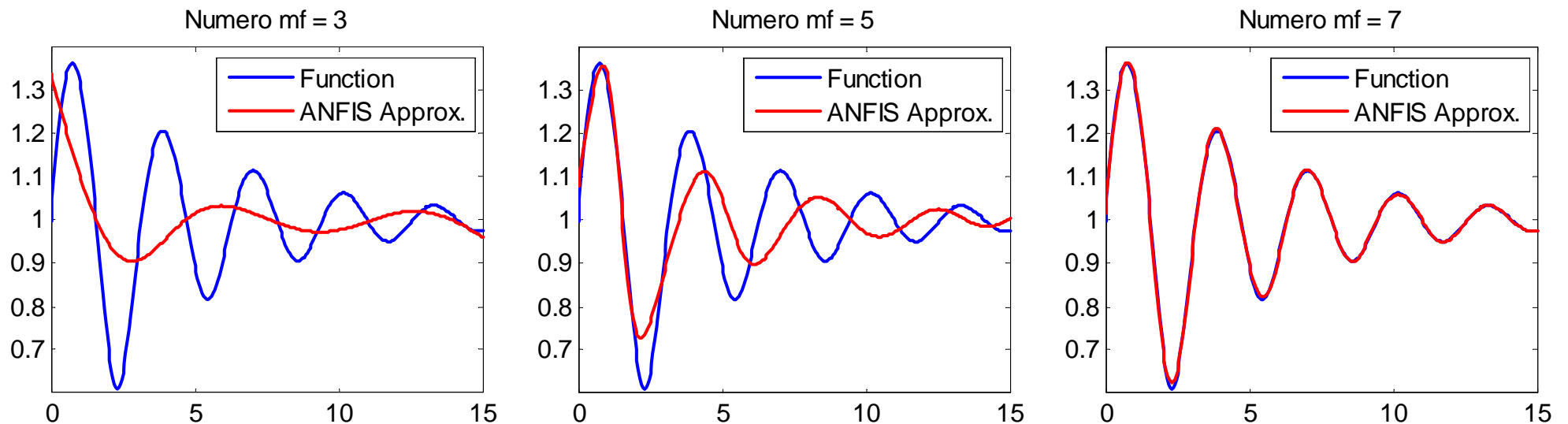
# Un esempio di ANFIS: approssimazione di funzione

Supponiamo di voler approssimare la funzione

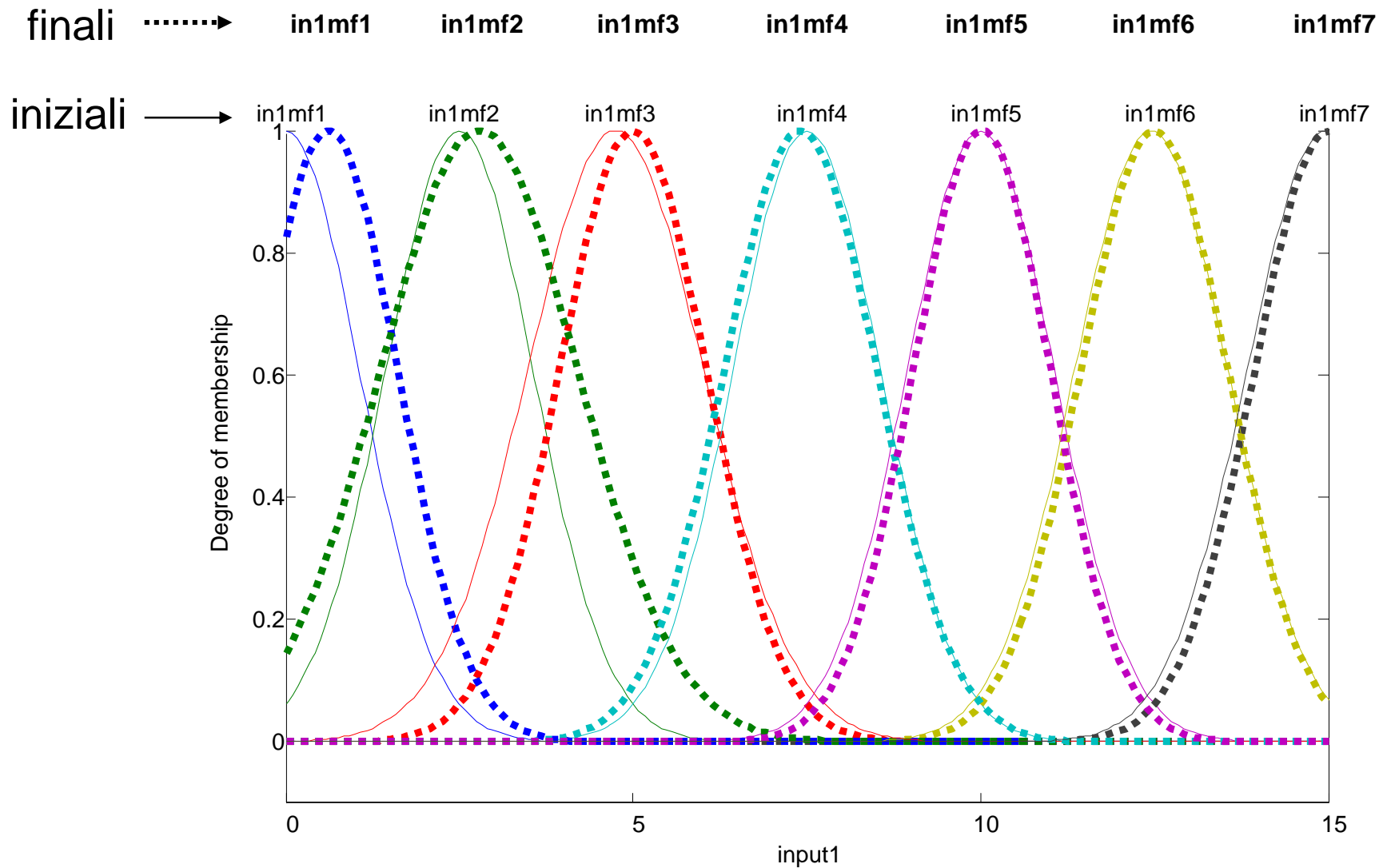
$$y = \sqrt{\left(1 + \sin(2x) \cdot e^{-\frac{x}{5}}\right)}$$

Vediamo come ANFIS approssima la funzione mediante funzioni lineari

Il grado di approssimazione varia molto con il numero di membership functions utilizzate

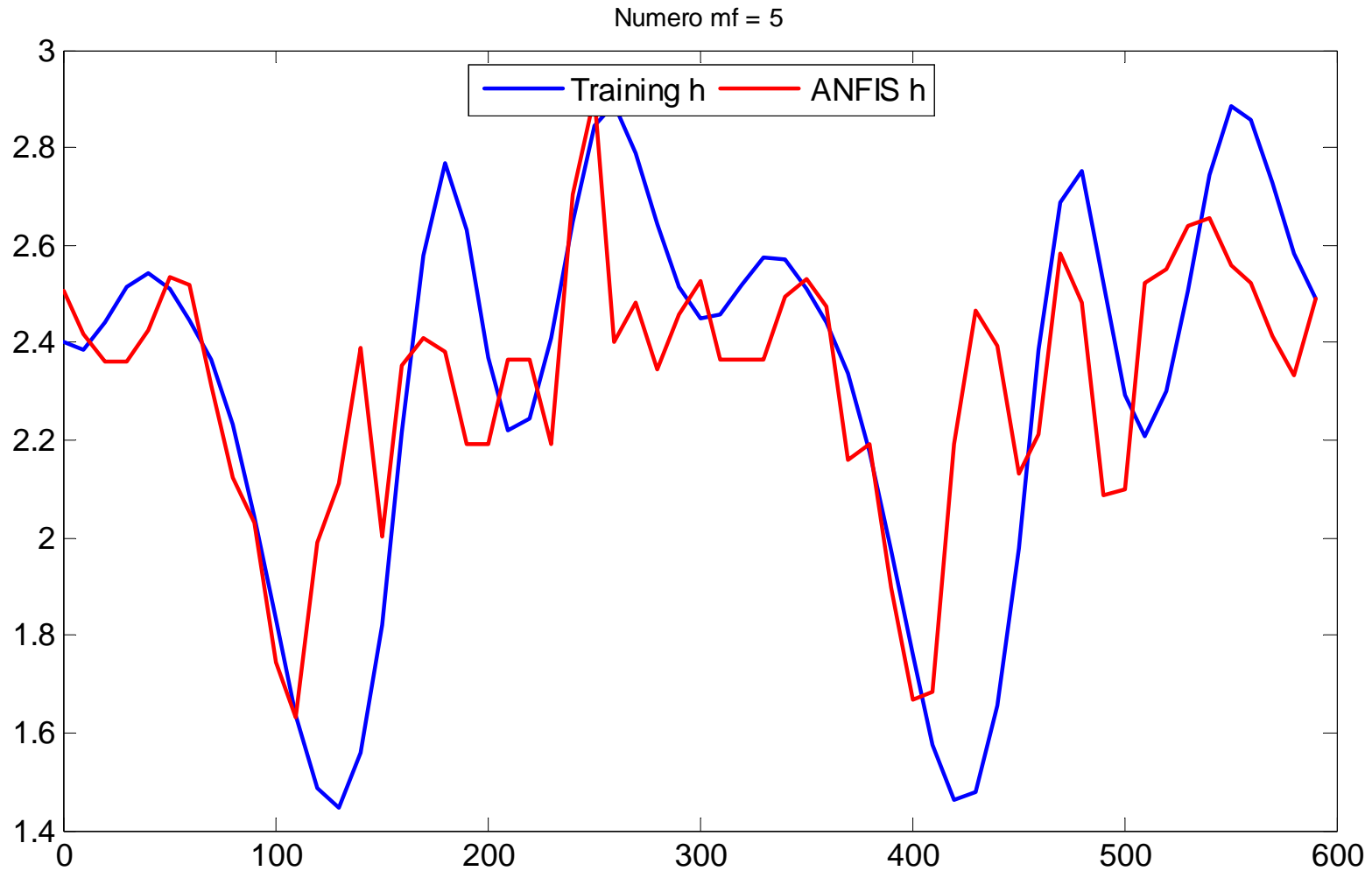


# Variazione delle membership functions



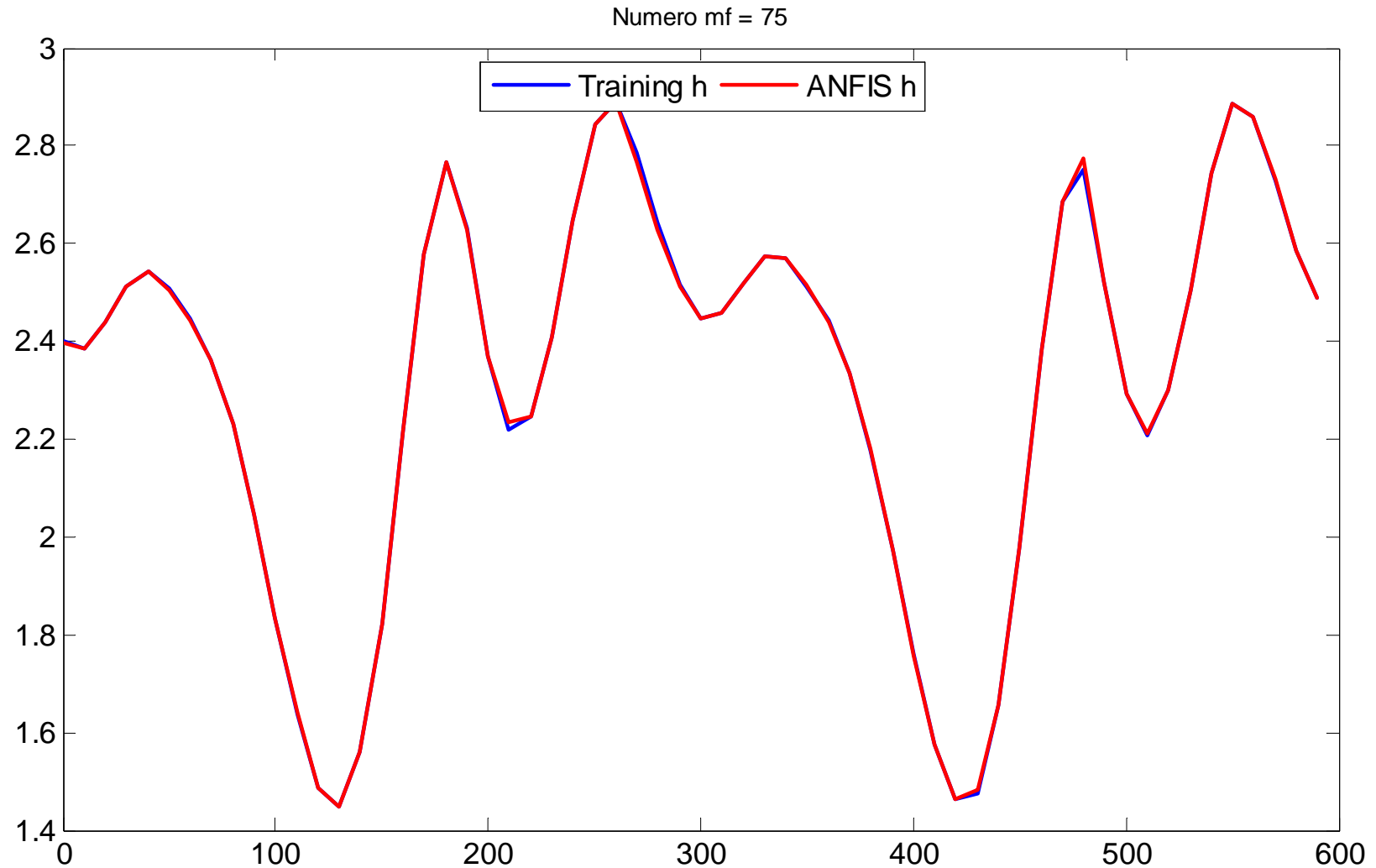
# Modello del CSTR

Con poche membership il modello è scadente come quello fatto a mano

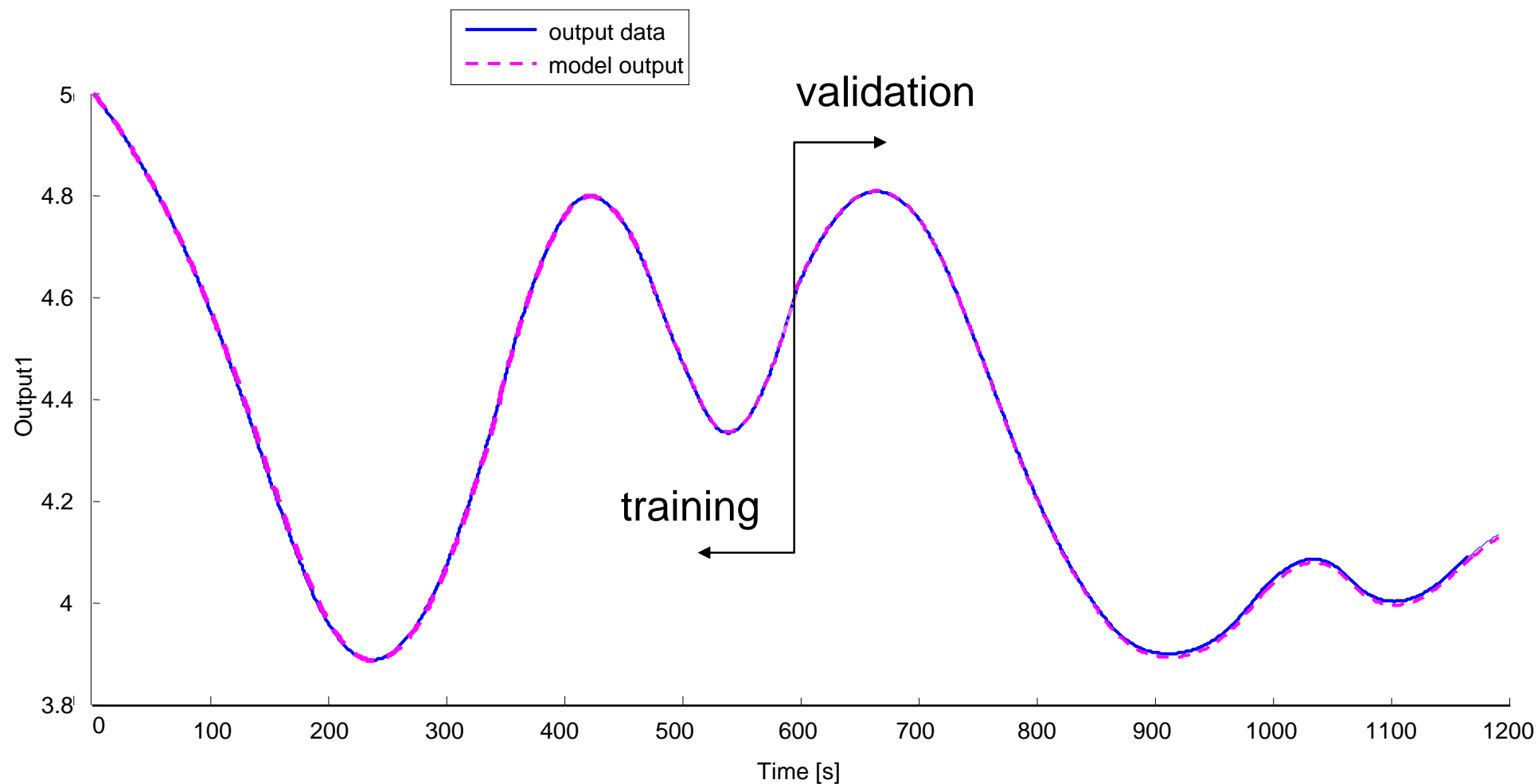


# Modello del CSTR

Una buona modellazione richiede un numero di membership elevato



# Modello del CSTR con clusterizzazione/Sugeno



# Conclusioni

---

- ☞ L'approccio alla Mamdani è “rassicurante” perché
  - ⇒ Permette di usare il normale buonsenso per costruire modelli
  - ⇒ E' intuitivo
  - ⇒ Permette di inserire nel modello nozioni non quantitative, ma di tipo linguistico
  - ⇒ Adatto per piccole semplici applicazioni
  - ⇒ Non è facilmente calibrabile in modo rigoroso
  - ⇒ La sua limitazione sta nel suo approccio empirico
- ☞ L'approccio alla Sugeno è più impegnativo, ma performante perché
  - ⇒ Offre una grande flessibilità nella definizione dei conseguenti
  - ⇒ Permette l'uso degli strumenti classici della teoria dei sistemi per imporre proprietà desiderabili (stabilità, raggiungibilità, etc.) ai conseguenti
  - ⇒ Apre un collegamento con la teoria classica della modellistica e del controllo
  - ⇒ Permette l'uso di strumenti sofisticati per la calibrazione sistematica del modello contro i dati (ANFIS, Babuska, etc....)
  - ⇒ Adatto ad applicazioni complesse

# Bibliografia

---

- 👉 Yager R.R. e Filev D.P. (1994) *Essentials of Fuzzy Modelling and Control*, Wiley.
- 👉 Klir. G.J. e T.A. Folger (1988) *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall.
- 👉 Ross T.J. (1995) *Fuzzy Logic with Engineering Applications*, McGraw-Hill.
- 👉 Driankov D, Hellendoorn H., Reinfrank M. (1993) *An introduction to Fuzzy Control*, Springer-Verlag.
- 👉 Patyra M.J. e Mlynek D.M. (editors) (1996) *Fuzzy Logic, Implementation and Applications*, Wiley Teubner.
- 👉 Nguyen H.T., Sugeno M., Tong R. Yager R.R. (editors) (1995) *Theoretical Aspects of Fuzzy Control*, Wiley.
- 👉 Yager R.R. e Zadeh L.A. (editors) (1994) *Fuzzy Sets, Neural Networks, and Soft Computing*, Van Nostrand Reinhold.
- 👉 Wang, L. X. (1994) *Adaptive Fuzzy Systems and Control*. PTR Prentice Hall.