



# **TECNICHE DI CONTROLLO E DIAGNOSI**

**Note applicative**

Dott. Ing. SIMANI SILVIO

Dott. Ing. BONFÈ MARCELLO

# Riepilogando...



**NOTE APPLICATIVE:** indicazioni utili per una migliore comprensione generale delle problematiche di progetto dei sistemi di controllo, al fine di:

- definire le procedure di sviluppo per le tesine richieste per l'esame
- preparare lo studente all'uso pratico delle tecniche di controllo, sia per attività di tesi di laurea che per attività professionali.

## **IL PROGETTO DI CONTROLLO + DIAGNOSI:**

1. Studio e modellazione matematica del sistema da controllare ("plant") o identificazione sperimentale delle sue caratteristiche.
2. Analisi delle caratteristiche dinamiche del *plant* al fine di definire la tecnica di controllo più idonea (es. evidenti nonlinearità, difficoltà di identificazione dei parametri, ecc.)
3. Progetto del controllore tramite una tecnica opportuna (approssimazione lineare, Feedback Linearization, Sliding Mode, Fuzzy Logic / Neural Networks)
4. Analisi numerica (simulazioni) del loop *plant / controllore*
5. Progetto di un *sistema di diagnosi*

# NOTE APPLICATIVE: modellazione e analisi

Lo studio del sistema da controllare va affrontato come un esercizio di teoria dei sistemi dinamici. Occorre anche osservare che molti sistemi “reali” sono riconducibili, introducendo eventualmente alcune approssimazioni, a modelli matematici “didattici” (es. tipico: il pendolo).

Da questo studio si otterrà come risultato una equazione differenziale:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u)$$

i cui punti di equilibrio costituiscono in genere l’obiettivo di interesse per il progetto del controllore, che dovrà stabilizzare o migliorare la stabilità di tali punti (nei problemi di regolazione).

Nei problemi di inseguimento, l’equilibrio desiderato è quello della dinamica dell’errore rispetto ad una traiettoria, ma le considerazioni sono analoghe.

L’analisi dei punti di equilibrio può anche dare indicazioni sulla struttura del controllore o sulle problematiche che si dovranno affrontare nel progettarelo...

# NOTE APPLICATIVE: analisi dell'equilibrio

L'equazione differenziale del sistema dinamico (singolo ingresso):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u)$$

ha come punti di equilibrio le soluzioni di:

$$\mathbf{f}(\mathbf{x}, u) = \mathbf{0}$$

In generale, non è detto che l'equilibrio sia determinato da condizioni con  $\mathbf{x}_e = \mathbf{0}$  né tantomeno con  $u_e = 0$ .

I valori di alcune variabili di stato nei punti di equilibrio potrebbero essere dipendenti gli uni dagli altri. In tali casi, fissato il valore di una/più delle variabili di stato ("set-point"), le altre sono determinate di conseguenza, così come il valore di ingresso necessario per quella condizione di equilibrio (**NB**: non è detto comunque che l'equilibrio sia stabile!).

$$\begin{aligned} [x_{(j,e)} \quad \dots \quad x_{(r,e)}]^T &= \mathbf{f}_e(x_{(i,e)}, \dots, x_{(m,e)}) \\ u_e &= f_{u,e}(\mathbf{x}_e) \end{aligned}$$

# NOTE APPLICATIVE: le approssimazioni lineari

Molte tecniche di controllo si basano su modelli lineari, in generale derivati dalla rappresentazione nello spazio degli stati:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

che come noto può essere ottenuta per approssimazione di un modello nonlineare:

$$\dot{\mathbf{x}} \approx \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{[x=0, u=0]} \mathbf{x} + \left. \frac{\partial \mathbf{f}}{\partial u} \right|_{[x=0, u=0]} u$$

o meglio, più in generale:

$$\dot{\mathbf{x}} \approx \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{[x=x_e, u=u_e]} (\mathbf{x} - \mathbf{x}_e) + \left. \frac{\partial \mathbf{f}}{\partial u} \right|_{[x=x_e, u=u_e]} (u - u_e)$$

che determina il modello **delle variazioni**: ( $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_e$ ;  $\tilde{u} = u - u_e$ )

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{u}$$

# NOTE APPLICATIVE: il controllo lineare

La stabilizzazione di un punto di equilibrio con un controllore lineare, ad esempio progettato come retroazione dello stato (v. controllo ottimo), si riferisce sempre al modello delle variazioni:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{u}$$

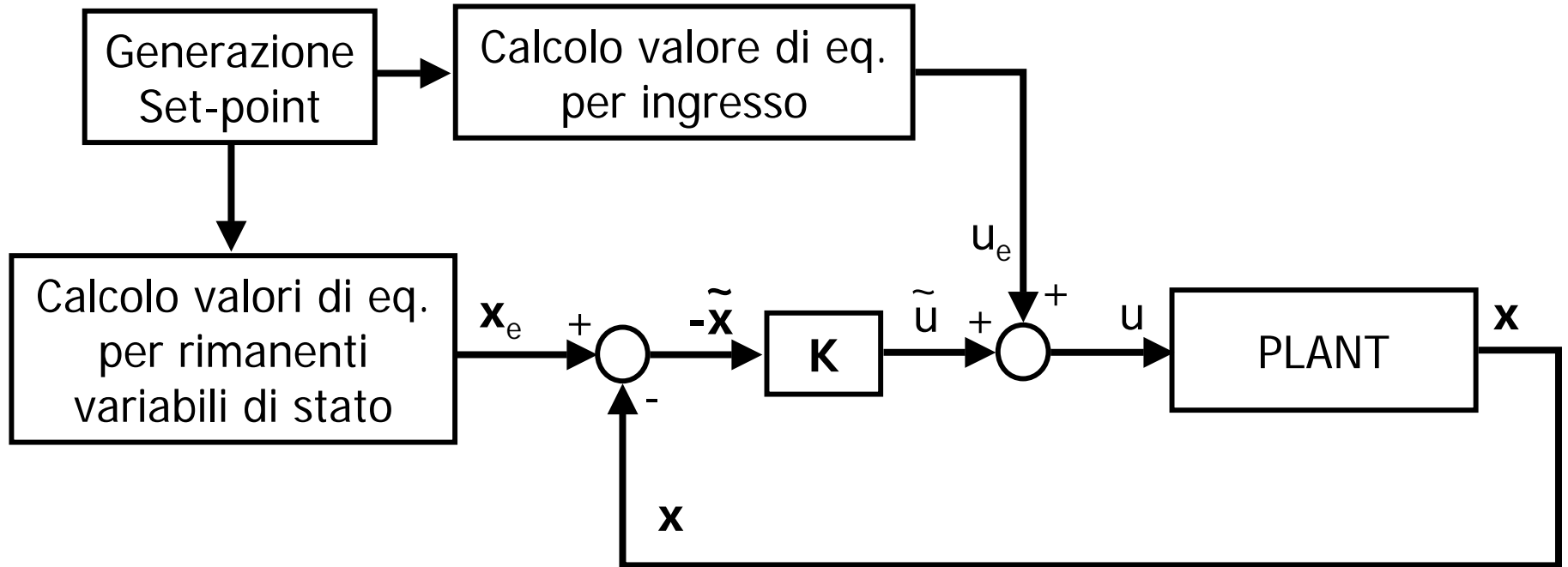
nel quale ( $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_e$ ;  $\tilde{u} = u - u_e$ ) e in generale ( $\mathbf{x}_e \neq \mathbf{0}$ ;  $u_e \neq 0$ )

Considerando come caso più generico quello nel quale l'obiettivo di controllo sia fissare alcune variabili di stato (*set-point*) in un punto di equilibrio, da queste ottenere i valori di equilibrio per le altre variabili di stato e per l'ingresso, e stabilizzare le variazioni rispetto a questo equilibrio con una retroazione negativa dello stato (v. controllo ottimo):

$$\tilde{u} = -\mathbf{K}\tilde{\mathbf{x}}$$

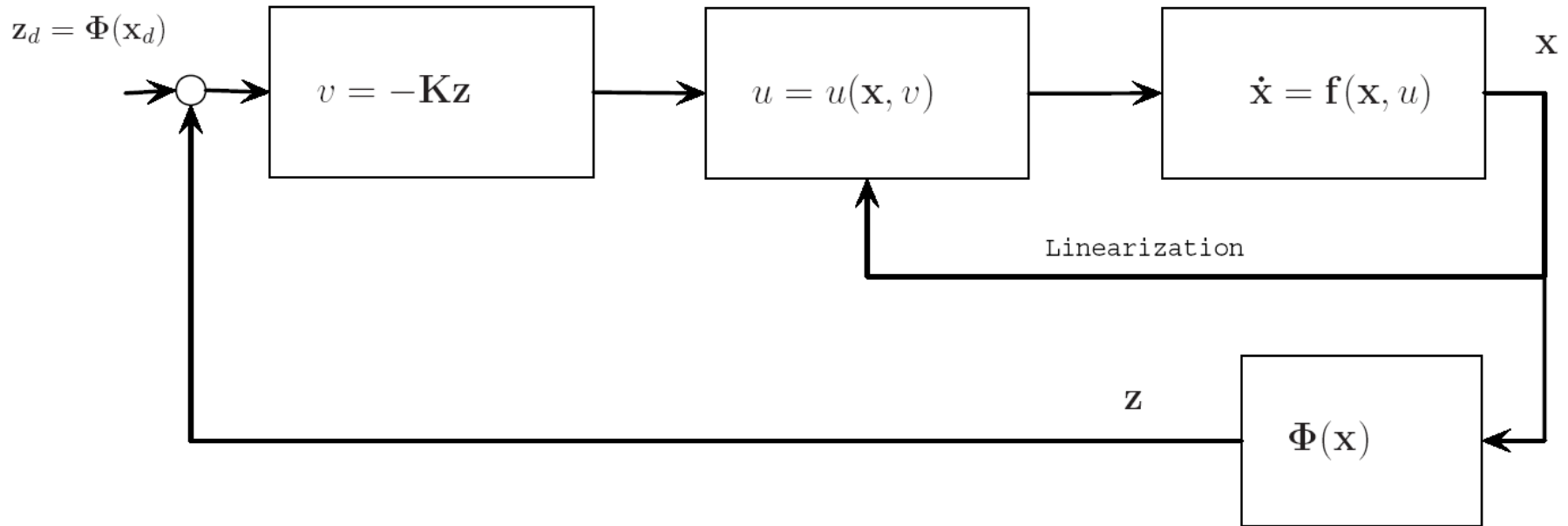
allora l'implementazione finale dello schema di controllo può essere schematizzata come segue:

# NOTE APPLICATIVE: controllo lineare - schema



**NOTARE BENE** la definizione dei segni delle variabili (retroazione negativa della variazione dello stato, controllo lineare che si somma all'ingresso di equilibrio)

# NOTE APPLICATIVE: Feedback Linearization



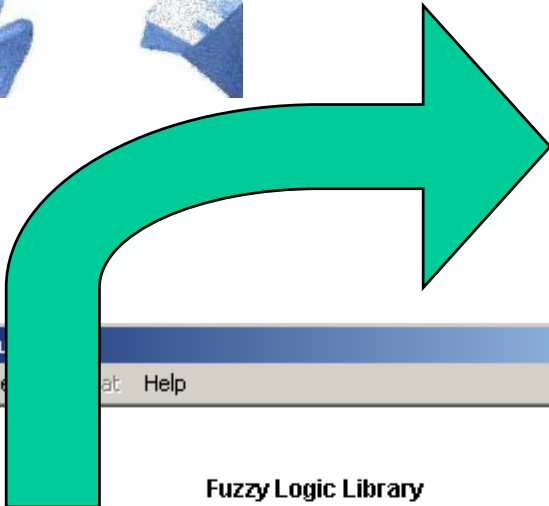
**NOTARE** la differenza rispetto allo schema precedente, nel quale il controllo lineare per essere efficace deve generare solo **variazioni** dell'ingresso di equilibrio, mentre grazie alla Feedback Linearization il sistema viene **trasformato** in un sistema lineare, il quale ha sempre come unico punto di equilibrio l'origine, seppure dello spazio degli stati in  $\mathbf{z}$  e non in  $\mathbf{x}$ .



# NOTE APPLICATIVE: Controllo Fuzzy



**IF ????**  
**THEN ???**



Library: fuzzy

File Edit View Help

### Fuzzy Logic Library

- Fuzzy Logic Controller
- Fuzzy Logic Controller with Ruleviewer
- MF Membership Functions

Ready 100% Locked

FIS Editor: slcpbnf

File Edit View

in1  
in2  
in3  
in4

slcpbnf (sugeno)

f(u)

out

FIS Name: slcpbnf FIS Type: sugeno

And method: prod  
Or method: max  
Implication: min  
Aggregation: max  
Defuzzification: wtaver

Current Variable  
Name:   
Type:   
Range:

Help Close

Ready

# NOTE APPLICATIVE: Controllo Fuzzy



I sostenitori della Logica Fuzzy dichiarano che un controllore fuzzy può essere progettato **senza conoscere il modello matematico** del sistema da controllare, ma solo le regole empiriche per “guidarlo”.

I detrattori della Logica Fuzzy sostengono invece che tali regole sono facilmente identificabili solo nei casi più semplici (**IF** livello basso **THEN** valvola molto aperta, **IF** errore piccolo **THEN** controllo piccolo), per i quali normalmente anche un controllore PID ottiene ottime prestazioni.

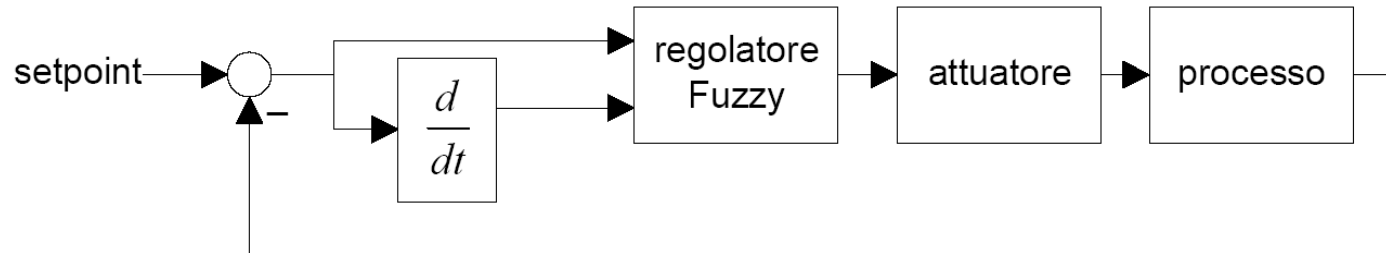
In effetti, per sistemi con più variabili di stato e fortemente nonlineari, tali regole di controllo non sono così intuitive, a meno che il sistema da controllare non venga studiato e analizzato dal punto di vista matematico...

Tuttavia, i sistemi fuzzy sono comunque degli approssimatori universali, per cui con opportune procedure di taratura automatica o di apprendimento, per quanto possano essere lunghe o applicabili solo “off-line”, si possono certamente ottenere prestazioni migliori di quelle dei controllori lineari.

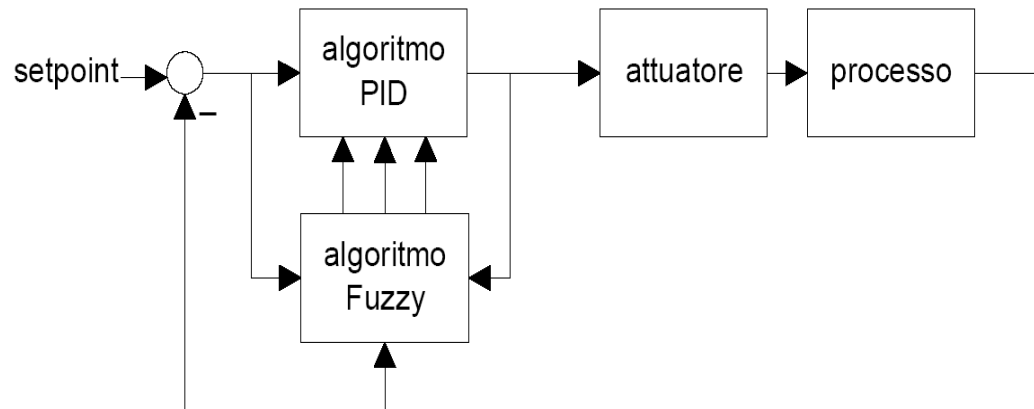
# NOTE APPLICATIVE: Controllo Fuzzy

## Possibili approcci di progetto al controllore fuzzy:

1. Fuzzificare la struttura classica PID-like



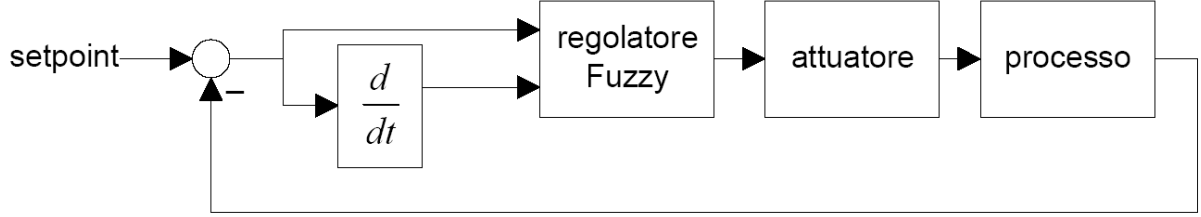
2. Usare un sistema Fuzzy per combinare diversi controllori lineari progettati in diversi punti di lavoro (Fuzzy Gain Scheduling o Supervisione Fuzzy)



3. Definire metodi di autotaratura o apprendimento (Self-Organizing Fuzzy Controller o Neuro-Fuzzy Controller)

# NOTE APPLICATIVE: Controllo Fuzzy PID-like

Struttura PD:

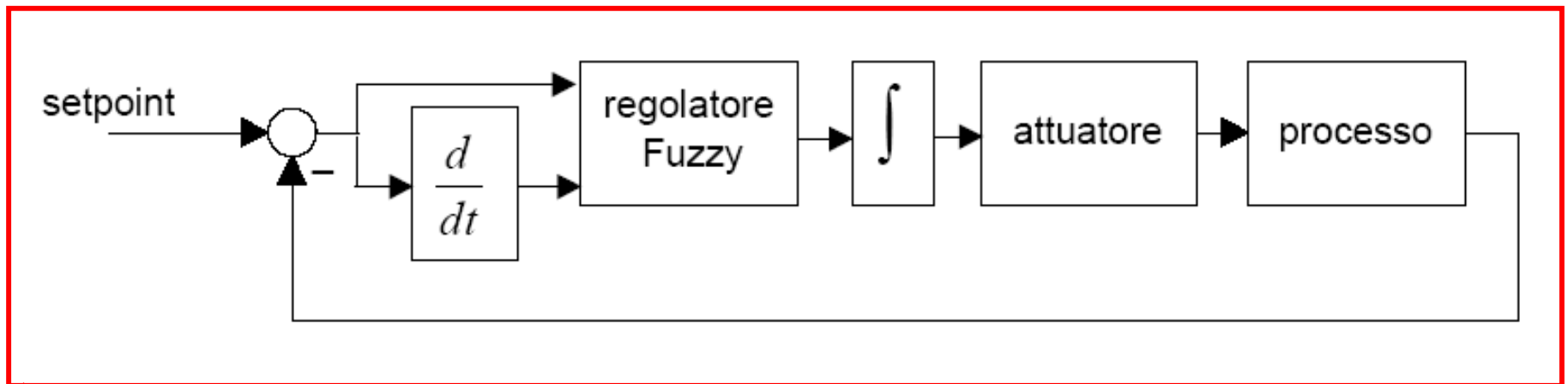
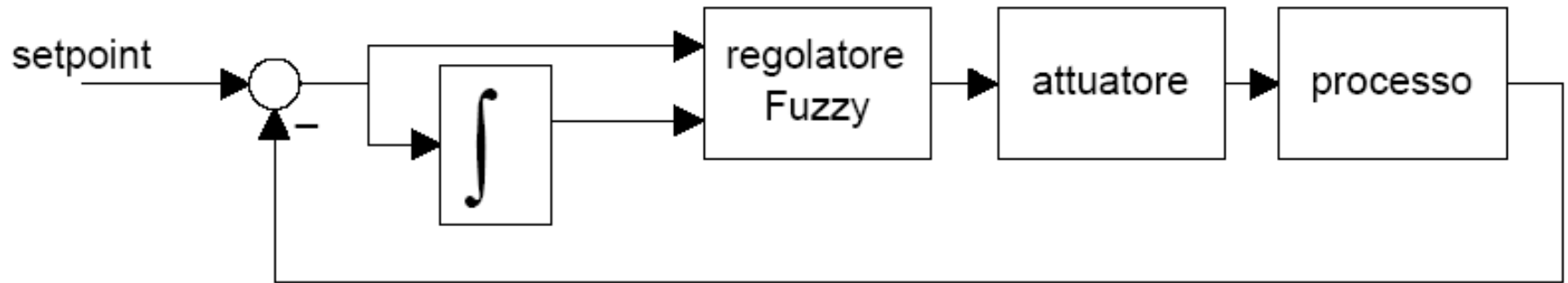


### DERIVATA DELL'ERRORE

		Negativa	Zero	Positiva
ERRORE	Negativo			
	Zero			
	Positivo			

# NOTE APPLICATIVE: Controllo Fuzzy PID-like

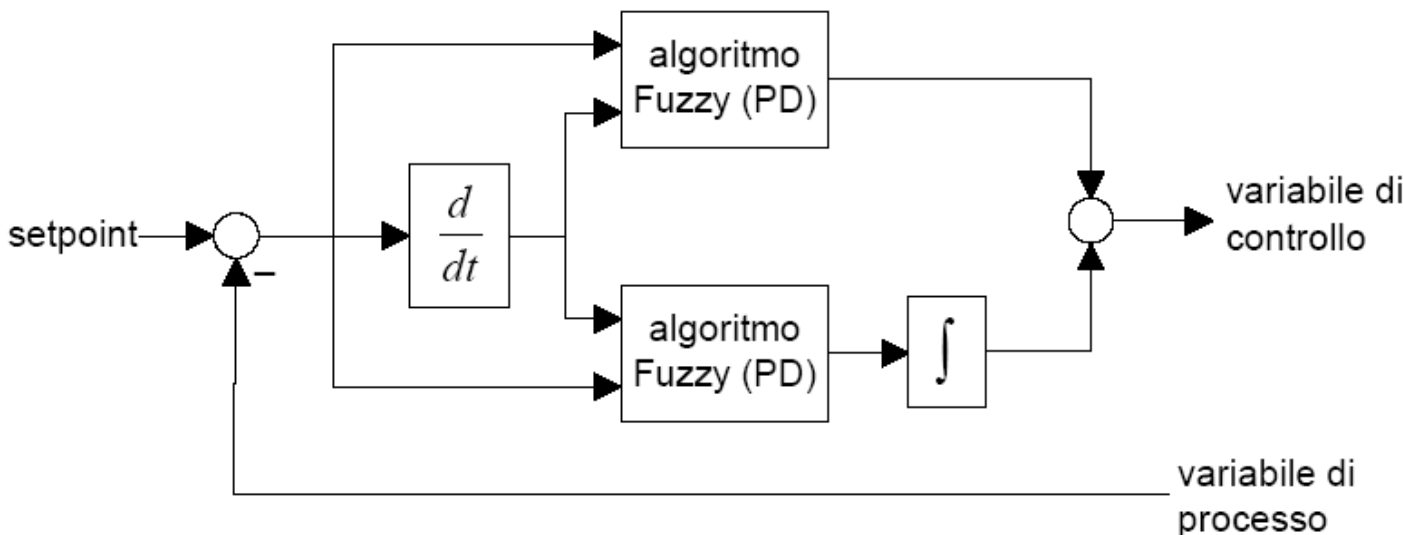
Strutture PI:



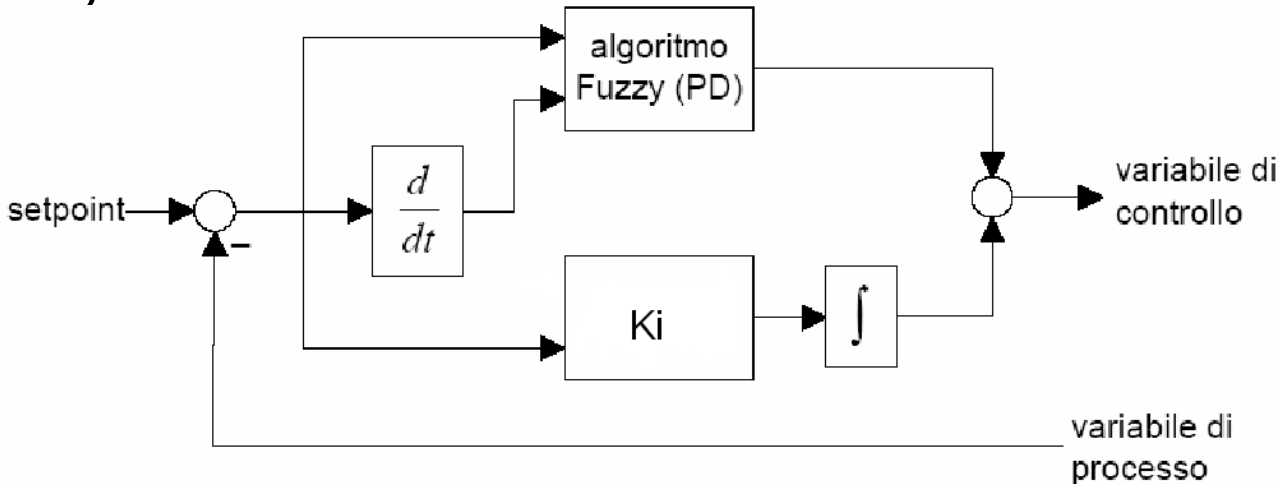
**Preferibile: stesse regole fuzzy della struttura PD!**

# NOTE APPLICATIVE: Controllo Fuzzy PID-like

Struttura PID (v. preferenza per la struttura PI con regole PD):

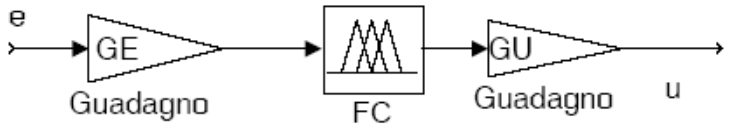


Struttura FuzzyPD+I:

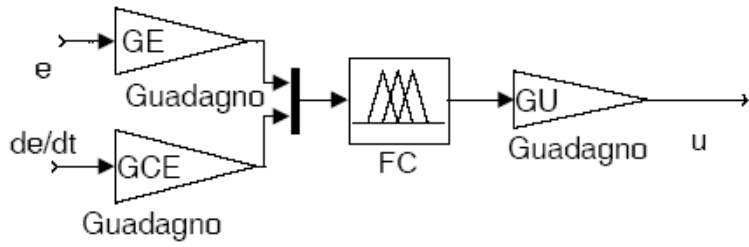


# NOTE APPLICATIVE: Fuzzy PID tuning

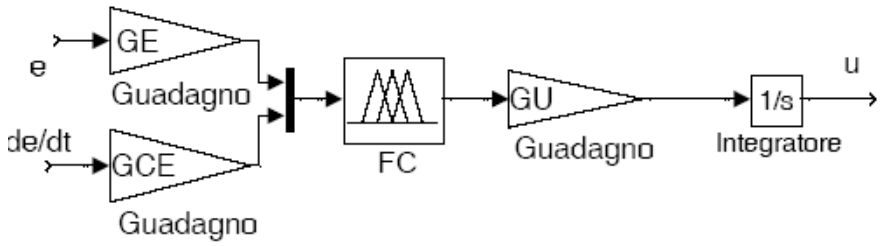
Vedi report di **Jan Jantzen "Tuning of Fuzzy PID Controllers"**:  
Il sistema Fuzzy usa variabili scalate fra  $[-100;100]$ , perciò i guadagni di messa in scala GE, GCE, GU e GIU vanno tarati e messi in relazione con quelli del PID classico.



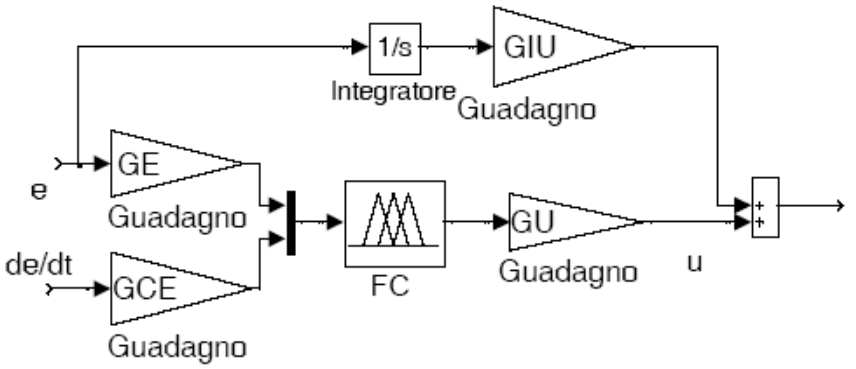
Fuzzy P



Fuzzy PD



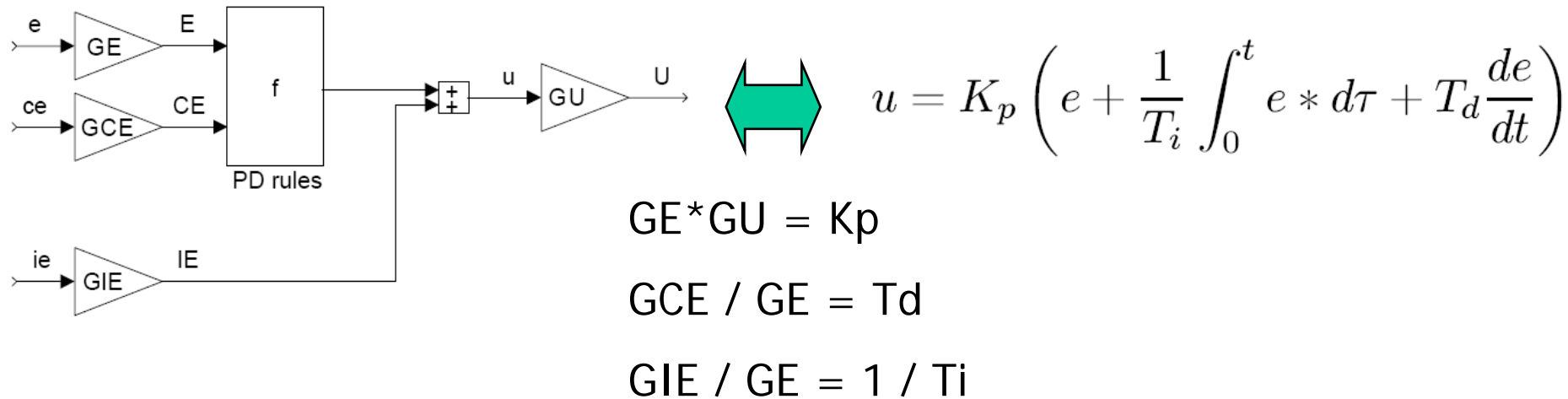
Fuzzy PI



Fuzzy PD+I

# NOTE APPLICATIVE: Fuzzy PID tuning

Esempio equivalenza Fuzzy PD+I e PID classico (formula "scolastica"):



**NOTA1:** GE e GCE vanno scelti in modo che gli ingressi del sistema Fuzzy (E e CE) siano tra [-100;100], in funzione di ipotesi sui valori massimi di errore e derivata (es. errore in [-Em;Em] → GE=100/Em)

**NOTA2:** se la struttura PID ha la formula più convenzionale con  $u = K_p * e + K_i * \text{int}(e) + K_d * de/dt$ , allora le relazioni diventano

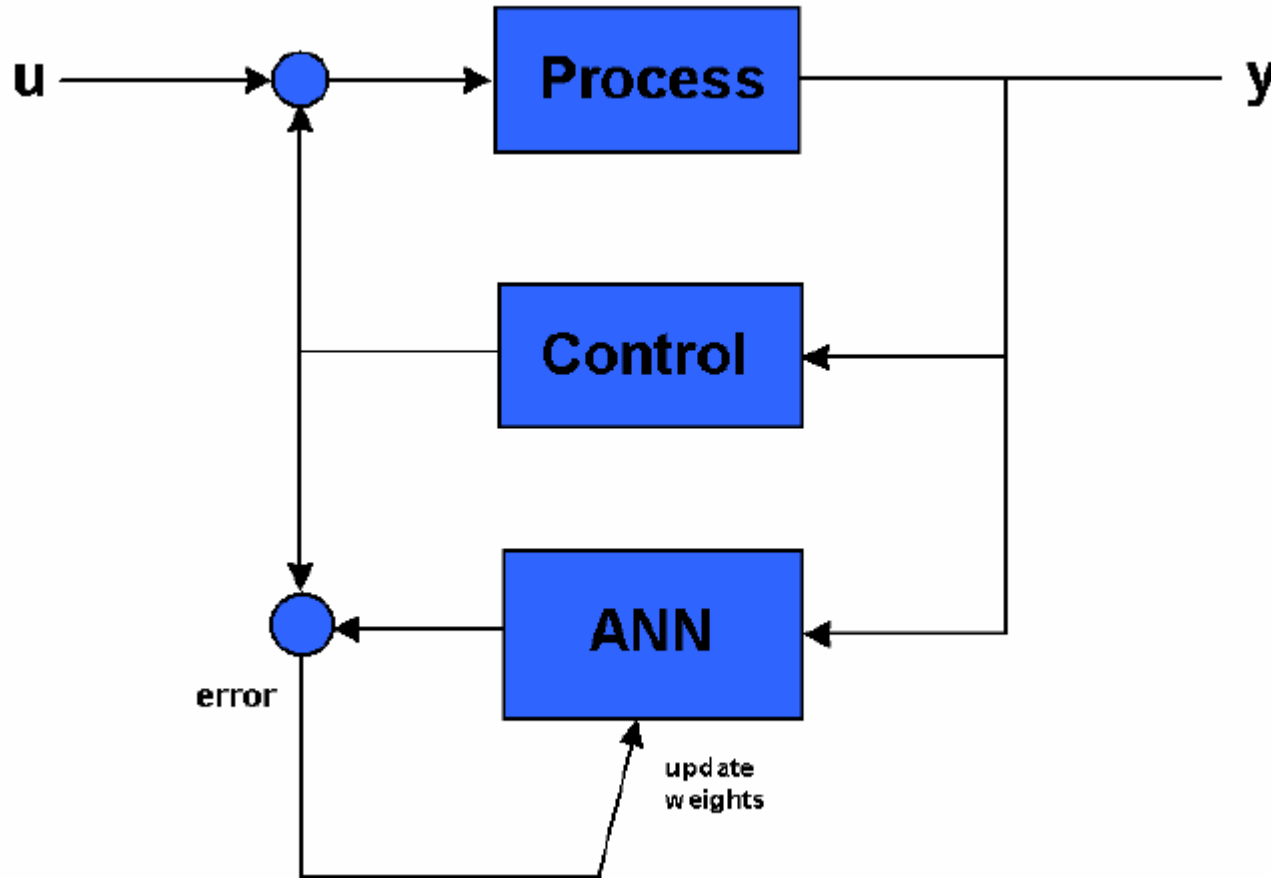
$$GE * GU = K_p; \quad GIE * GU = K_i; \quad GCE * GU = K_d$$



# NOTE APPLICATIVE: controllo con Reti Neurali

## Supervised Learning:

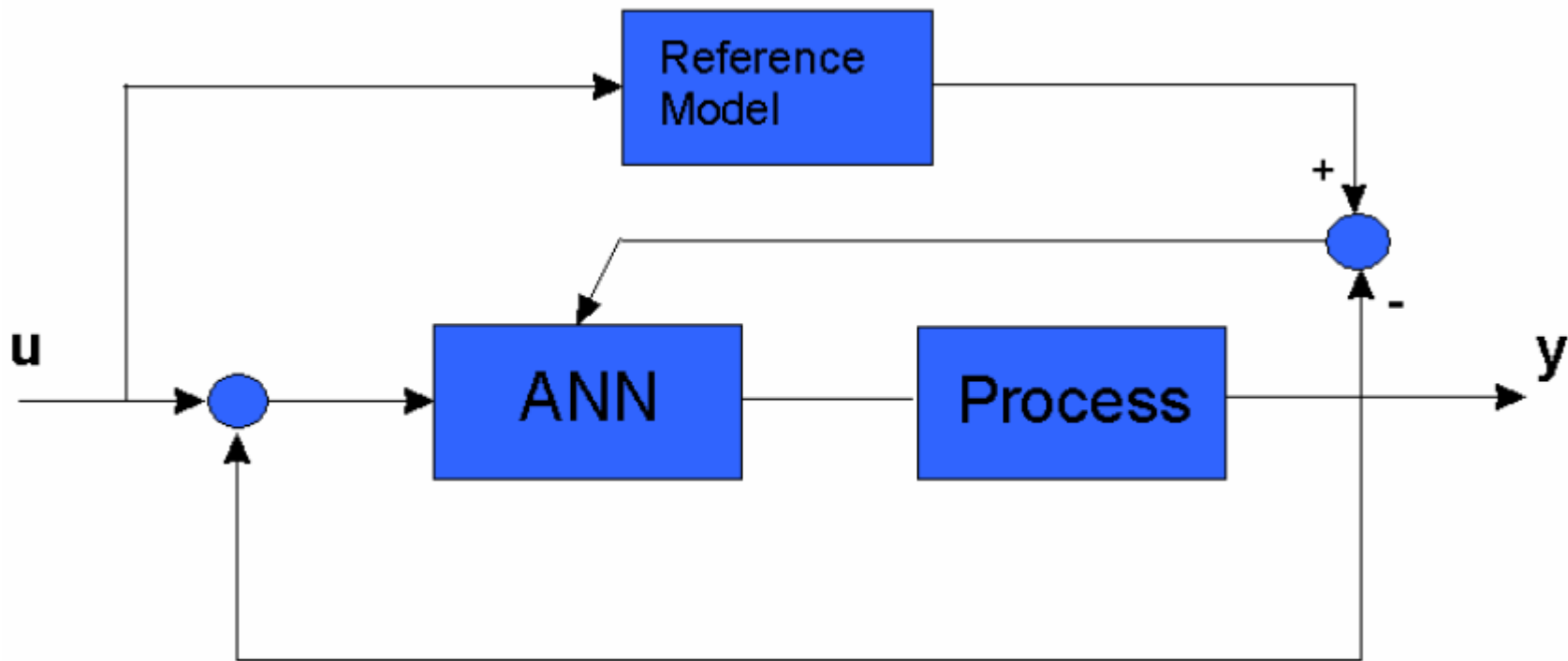
la rete neurale impara ad "emulare" un controllore esistente



# NOTE APPLICATIVE: controllo con Reti Neurali

## Model Reference Adaptive Control (MRAC):

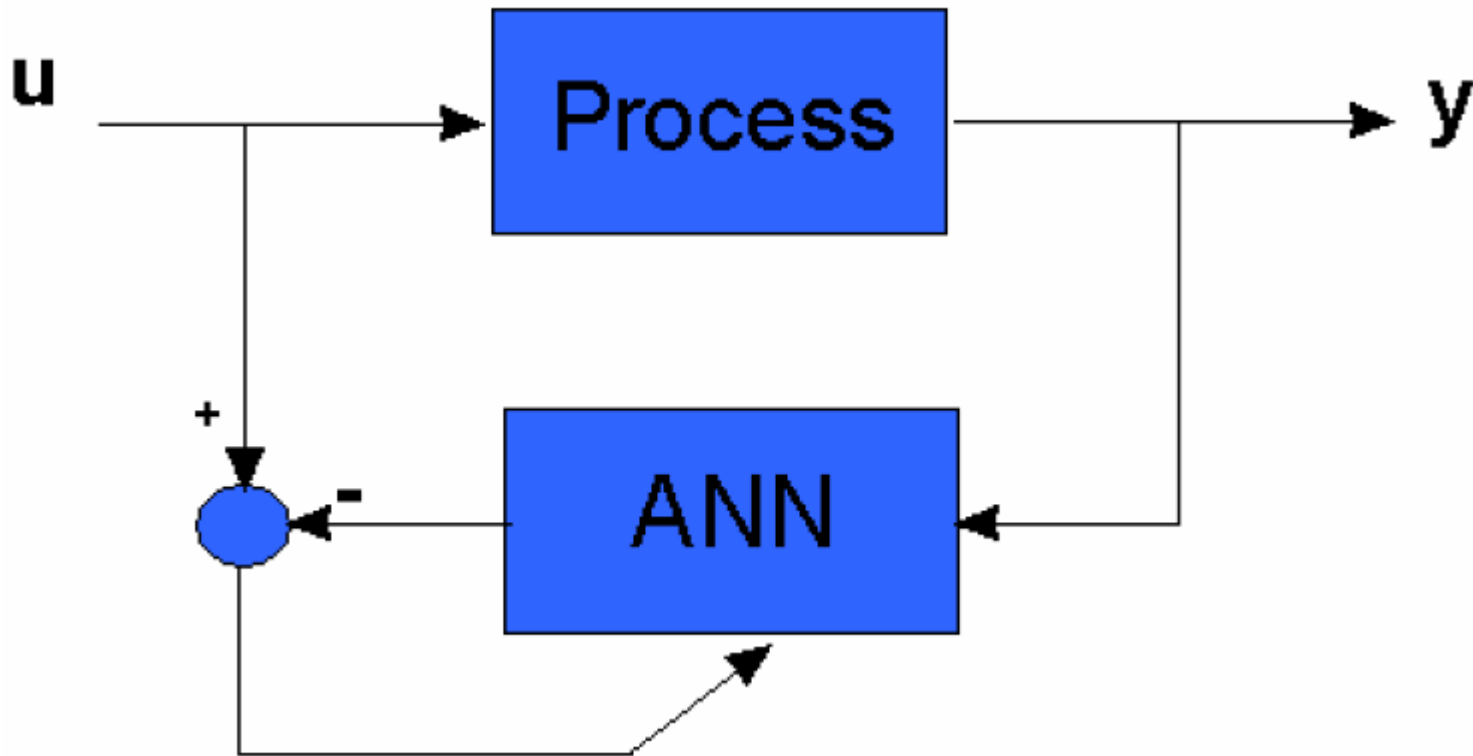
la rete neurale impara a controllare il sistema per "emulare" un modello di riferimento



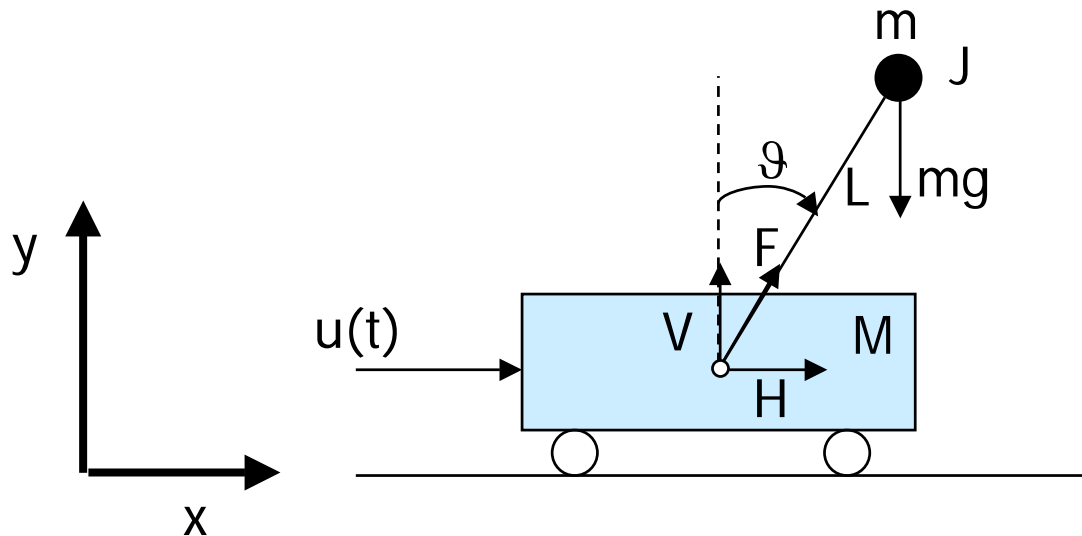
# NOTE APPLICATIVE: controllo con Reti Neurali

## Inverse Dynamics:

la rete neurale impara a comportarsi come un modello inverso del sistema (non applicabile se il sistema è instabile...)



# Esempio – Pendolo inverso su carrello



La posizione dell'asta rispetto al carrello è individuata da  $\vartheta$  mentre  $x$  individua la posizione del carrello.

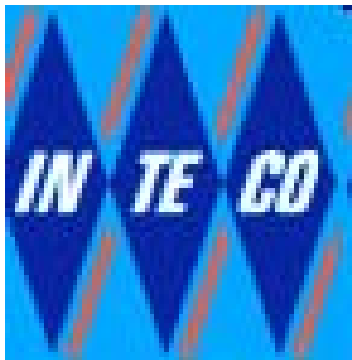
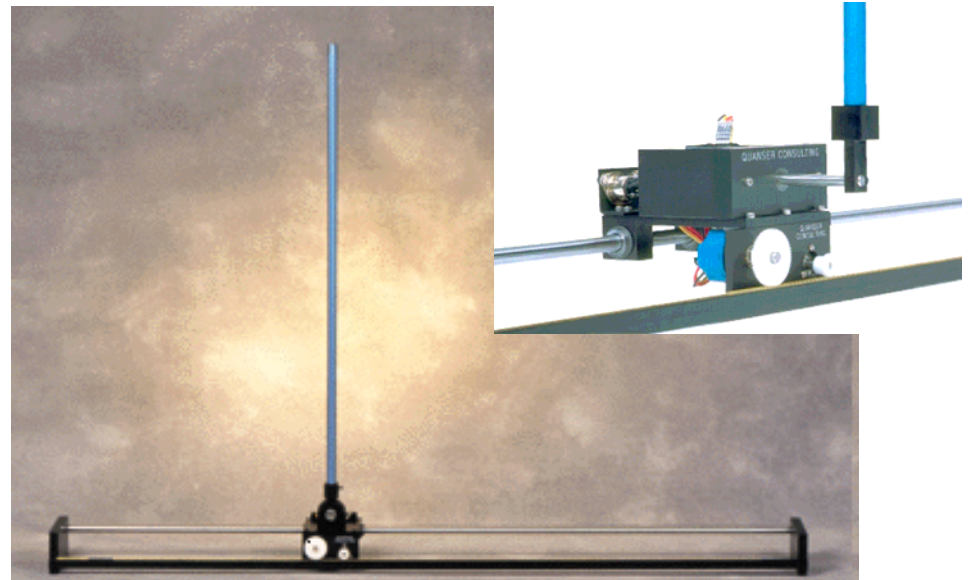
L'azione di controllo è esercitata tramite la forza lineare  $u(t)$  applicata al carrello.

Lo studio di tale sistema "didattico" è utile per le analogie con molte altre applicazioni di controllo di vario genere (es. i sistemi guida missili, il controllo di gru, il Segway® ecc.).

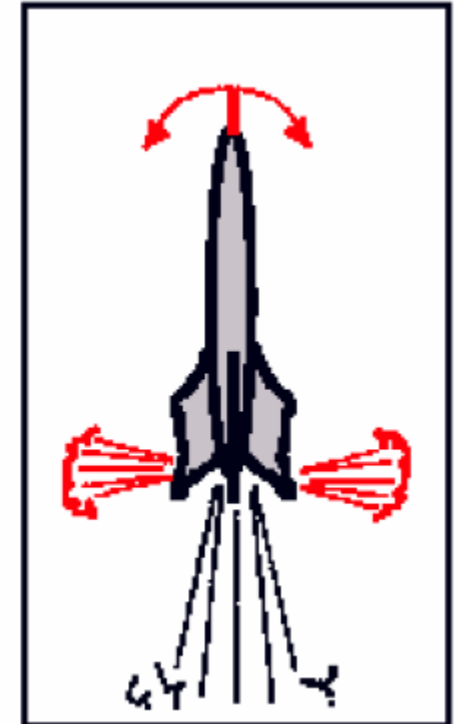
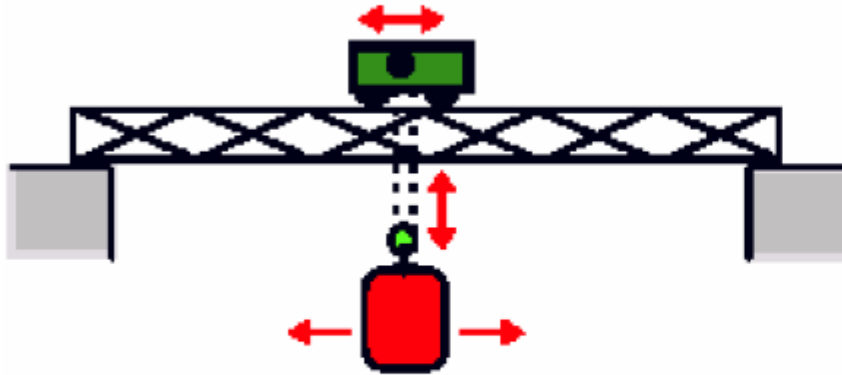
# Pendolo inverso – sistema “didattico”...



QUANSER  
INNOVATE. EDUCATE.



# Pendolo inverso – ...con applicazioni reali!



# Pendolo inverso – Calcolo del modello



Il sistema è composto da due sottosistemi: **l'asta e il carrello**.

**L'asta** è vincolata al carrello tramite una cerniera che vincola il movimento alla rotazione attorno all'asse della cerniera (perpendicolare a  $xy$ ).

**Il carrello** può muoversi solo lungo l'asse  $x$  per l'azione di una forza  $u(t)$ .

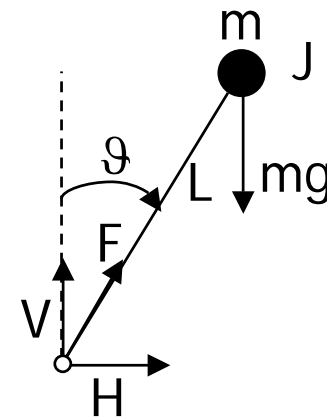
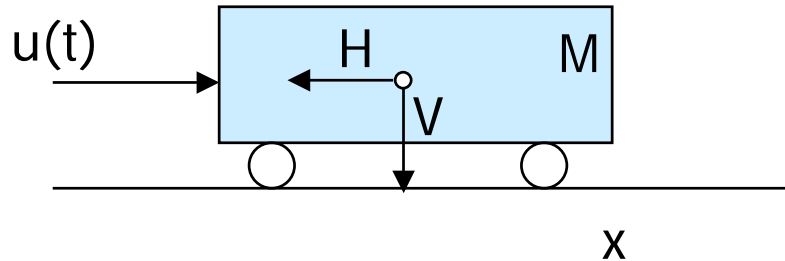
L'asta e il carrello interagiscono tramite una forza  $F$  di reazione, diretta lungo l'asta, che può essere scomposta in una componente verticale  $V$  e una componente orizzontale  $H$ .

**Ipotesi semplificative:** no attriti nella cerniera e fra carrello e piano di scorrimento, masse concentrate nei baricentri, inerzia dell'asta trascurabile.

Il modello del sistema si può ottenere facendo bilanci di forza e coppia sia per l'asta che per il carrello.

# Pendolo inverso – scomposizione

**N.B.** Forze di reazione uguali ed opposte a causa del vincolo dato dalla cerniera





# Pendolo inverso – Calcolo del modello

## Dinamica dell'asta

Dal bilancio delle forze sulla massa  $m$  (la massa dell'asta è supposta trascurabile), si ottiene che:

$$m\vec{a}_a = \vec{F} + m\vec{g}$$

Tenendo conto che l'accelerazione di gravità è diretta nel senso negativo dell'asse  $y$  e che  $H$  e  $V$  sono, rispettivamente, le componenti lungo l'asse  $x$  e lungo l'asse  $y$  della forza  $F$ , il bilancio di forze può essere scritto come una coppia di equazioni che descrivono il comportamento dell'asta lungo l'asse  $x$  e lungo l'asse  $y$ :

$$\begin{cases} m \frac{d^2}{dt^2} (x + L \sin \vartheta) = H \\ m \frac{d^2}{dt^2} (L \cos \vartheta) = V - mg \end{cases}$$

# Pendolo inverso – Calcolo del modello

Il bilancio delle coppie ci dà la seguente equazione:

$$J\ddot{\vartheta} = VL \sin \vartheta - HL \cos \vartheta$$

Che, trascurando  $J$ , diventa:  $VL \sin \vartheta = HL \cos \vartheta$

La dinamica dell'asta, pertanto, è rappresentata dalle seguenti equazioni:

$$\left\{ \begin{array}{l} m \frac{d^2}{dt^2} (x + L \sin \vartheta) = H \\ m \frac{d^2}{dt^2} (L \cos \vartheta) = V - mg \\ VL \sin \vartheta = HL \cos \vartheta \end{array} \right.$$

# Pendolo inverso – Calcolo del modello

## Dinamica del carrello

La dinamica del carrello si ottiene semplicemente da:

$$M\vec{a}_c = \vec{u} + \vec{F}_{car}$$

Siccome il carrello può muoversi solo lungo l'asse x (le forze agenti sono H e u), l'equazione che descrive la dinamica del carrello è:

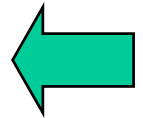
$$M\ddot{x} = u - H$$

# Pendolo inverso – Calcolo del modello

Svolgendo i calcoli, dalle equazioni dinamiche dell'asta, si trova che:

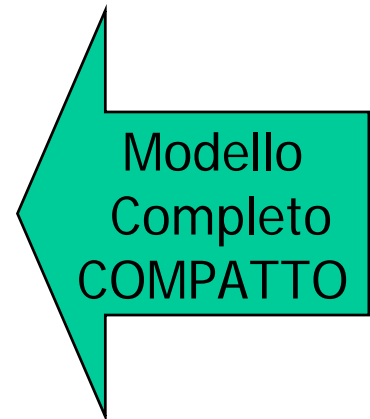
$$V(t) = mg + mL\ddot{\vartheta} \sin \vartheta - mL\dot{\vartheta}^2 \cos \vartheta$$

$$H(t) = m\ddot{x} + mL\ddot{\vartheta} \cos \vartheta - mL\dot{\vartheta}^2 \sin \vartheta$$



sostituendo le espressioni di  $V(t)$  e di  $H(t)$  nel bilancio di coppia e nella dinamica del carrello, si può **compattare il modello** come segue:

$$\begin{cases} (M + m)\ddot{x} + mL\ddot{\vartheta} \cos \vartheta - mL\dot{\vartheta}^2 \sin \vartheta = u(t) \\ m\ddot{x} \cos \vartheta + mL\ddot{\vartheta} - mg \sin \vartheta = 0 \end{cases}$$



# Pendolo inverso – Calcolo del modello

Oppure, esprimendo le equazioni nella forma più consueta per modelli nello spazio degli stati:

$$\begin{cases} \ddot{x} = \frac{mL\dot{\vartheta}^2 \sin \vartheta - mg \sin \vartheta \cos \vartheta + u}{M + m \sin^2 \vartheta} \\ \ddot{\vartheta} = \frac{-mL\dot{\vartheta}^2 \sin \vartheta \cos \vartheta + (M + m)g \sin \vartheta - u \cos \vartheta}{(M + m \sin^2 \vartheta)L} \end{cases}$$

che, ponendo

$$\mathbf{x} = [x_1 \quad x_2 \quad x_3 \quad x_4]^T = [x \quad \dot{x} \quad \vartheta \quad \dot{\vartheta}]^T$$

diventa una  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u)$  come segue...

# Pendolo inverso – Calcolo del modello

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{mLx_4^2 \sin x_3 - mg \sin x_3 \cos x_3 + u}{M + m \sin^2 x_3} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{-mLx_4^2 \sin x_3 \cos x_3 + (M + m)g \sin x_3 - u \cos x_3}{(M + m \sin^2 x_3)L} \end{array} \right.$$

# Pendolo inverso – Problema di controllo

## POSSIBILI SOLUZIONI:

- ➔ **Controllo ottimo LQ:** approssimazione lineare in  $x_e$  e progetto di  $u = -K x$
- ➔ **Controllo con Feedback Linearization / Sliding Mode**
- ➔ **Controllo Fuzzy:**
  - Controllore con “ragionamento” fuzzy (ma quali regole???)
  - Controllore tramite Fuzzy Gain Scheduling di più regolatori ottimi o Fuzzy PID-like (taratura lunga ma sistematica)
- ➔ **Controllo con Reti Neurali:** il metodo più idoneo è il Supervised Learning, vista la forte nonlinearietà e l'instabilità ad anello aperto (necessità di controllore pre-esistente il training)

# Pendolo inverso – Controllo LQ

I punti di equilibrio sono tutti i vettori  $\mathbf{x}_e = [x_d \ 0 \ 0 \ 0]^T$   
con  $x_d$  qualunque, oppure:

$$\mathbf{x}_e = [x_d \ 0 \ \pm \pi \ 0]^T$$

che corrispondono ovviamente al pendolo sulla verticale verso l'alto (instabile) o verso il basso (stabile).

Si può notare che le approssimazioni lineari in queste due tipologie di punti non dipendono da  $x_d$ , inoltre sia nell'equilibrio stabile che in quello instabile il valore di equilibrio dell'ingresso è  $u_e = 0$

Questo significa che il problema di controllo lineare è relativamente più semplice che in altri tipi di problemi, nei quali le approssimazioni lineari potrebbero essere differenti a seconda del punto di equilibrio e, soprattutto, richiedere diverse condizioni "nominali" per l'ingresso.



# Pendolo inverso – Controllo LQ

Considerando i punti di equilibrio  $\mathbf{x}_e = [x_d \ 0 \ 0 \ 0]^T$   
con  $x_d$  qualunque, possiamo ottenere:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

Con linearizzazione approssimata:  $\dot{\mathbf{x}} \approx \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{[x=0, u=0]} \mathbf{x} + \left. \frac{\partial \mathbf{f}}{\partial u} \right|_{[x=0, u=0]} u$

Oppure, considerando che:

$$\left\{ \begin{array}{l} \sin x_3 \approx x_3 \\ \cos x_3 \approx 1 \\ \sin^2 x_3 = 1 - \cos^2 x_3 \approx 0 \\ \dot{x}_3 \approx 0 \end{array} \right.$$

# Pendolo inverso – Controllo LQ

Otteniamo:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{-mg}{M}x_3 + \frac{u}{M} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{(M+m)g}{ML}x_3 - \frac{u}{ML} \end{cases}$$

Perciò:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(M+m)g}{ML} & 0 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{ML} \end{pmatrix}$$

# Pendolo inverso – Controllo LQ

Cerchiamo:  $u(t) = -\mathbf{K}\mathbf{x}(t)$

tale da minimizzare:  $\mathbf{J} = \int_0^{+\infty} (\mathbf{x}^T \mathbf{Q}\mathbf{x} + uRu)dt$

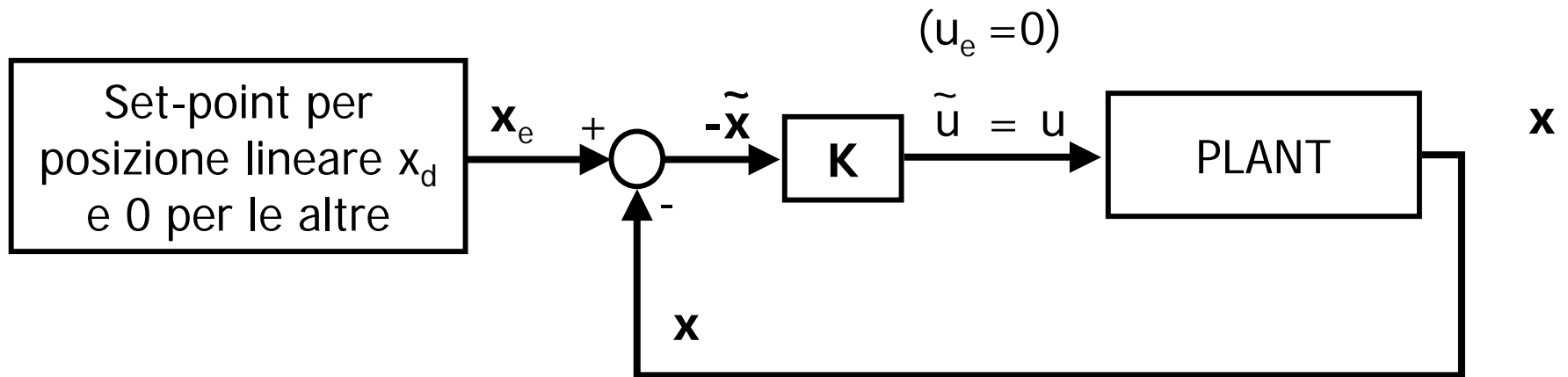
Se la coppia  $(\mathbf{A}, \mathbf{B})$  è stabilizzabile, la soluzione al progetto LQ è:

$$\mathbf{K} = R^{-1} \mathbf{B}^T \mathbf{S}$$

nella quale  $\mathbf{S}$  è la soluzione della Equazione Algebrica di Riccati (ARE):

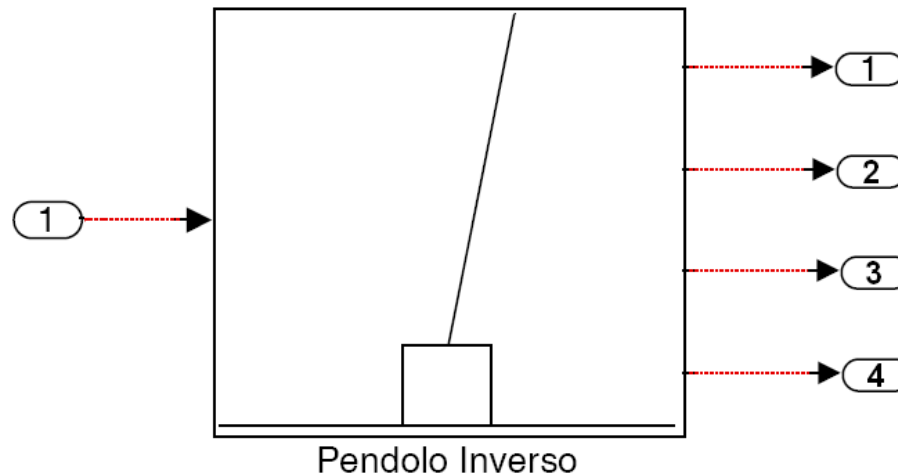
$$\mathbf{A}^T \mathbf{S} + \mathbf{S}\mathbf{A} + \mathbf{Q} - \mathbf{S}\mathbf{B}R^{-1} \mathbf{B}^T \mathbf{S} = \mathbf{0}$$

# Pendolo inverso – Controllo LQ: schema



# Pendolo inverso – Controllo LQ e MATLAB

Dato un modello Simulink con il solo sistema da controllare e nodi di I/O:



all'interno del quale le condizioni iniziali dello stato siano modificabili, è possibile usare la funzione

```
[A,B,C,D] = linmod('nome_modello_sim',...)
```

per ottenere le matrici A, B, C e D dell'approssimazione lineare.

# Pendolo inverso – Controllo LQ e MATLAB

Date le matrici A e B (C si suppone sempre l'identità, cioè lo stato completamente misurabile, e  $D = 0$ ) dell'approssimazione lineare in un determinato punto di lavoro, è possibile progettare un controllore ottimo con la funzione:

$$[K, S, E] = \text{lqr}(A, B, Q, R, N)$$

nella quale le matrici Q, R e N determinano la funzione costo:

$$J = \text{Integral} \{x'Qx + u'Ru + 2*x'Nu\} dt$$

K è la matrice dei guadagni della retroazione dello stato, S è la soluzione dell'equazione di Riccati ed E è il vettore degli autovalori del sistema in anello chiuso.

# Pendolo inverso – Controllo Sliding Mode

**Passo I - cancellazione parziale delle nonlinearità:**

Applicando la legge di controllo:

$$u = (M + m \sin^2 x_3)v - (mLx_4^2 \sin x_3 - mg \sin x_3 \cos x_3)$$

si semplifica (tramite feedback) il modello (ancora non lineare) a:

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{mLx_4^2 \sin x_3 - mg \sin x_3 \cos x_3 + u}{M + m \sin^2 x_3} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{-mLx_4^2 \sin x_3 \cos x_3 + (M + m)g \sin x_3 - u \cos x_3}{(M + m \sin^2 x_3)L} \end{array} \right. \quad \left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = v \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{g \sin x_3 - v \cos x_3}{L} \end{array} \right.$$

# Pendolo inverso – Controllo Sliding Mode

## Passo II – Feedback Linearization (approssimata):

**N.B.** La linearizzazione completa input-state non è possibile!  
Però, se consideriamo come uscita del sistema la seguente

$$y = h(\mathbf{x}) = x_1 + L \ln \left( \frac{1 + \sin x_3}{\cos x_3} \right)$$

si può ottenere il seguente cambiamento di variabili ( $y = \Phi_1(\mathbf{x}) = z_1$ )

$$\dot{y} = x_2 + \frac{Lx_4}{\cos x_3} \quad (= \Phi_2(\mathbf{x}) = z_2)$$

$$\ddot{y} = \tan x_3 \left( g + \frac{Lx_4}{\cos x_3} \right) \quad (= \Phi_3(\mathbf{x}) = z_3)$$

$$\ddot{\ddot{y}} = \left( \frac{2}{\cos^3 x_3} - \frac{1}{\cos x_3} \right) Lx_4^3 + \left( \frac{3g}{\cos^2 x_3} - 2g \right) x_4 - \underbrace{2x_4 \tan x_3 v}_{\text{trascurato!}} \quad (= \Phi_4(\mathbf{x}) = z_4)$$

$$y^{(4)} = f_1(\mathbf{x}) + g_1(\mathbf{x})v = \dot{z}_4$$



# Pendolo inverso – Controllo Sliding Mode

Per completezza, le funzioni che compaiono nell'ultima equazione sono:

$$f_1(\mathbf{x}) = \frac{\sec(x_3)}{L} \left( g^2 \tan(x_3) + 9 g L x_4^2 \sec(x_3) \tan(x_3) + 5 L^2 x_4^4 \sec(x_3)^2 \tan(x_3) + g (3 L x_4^2 + 2 g \cos(x_3)) \sin(x_3) \tan(x_3)^2 + L^2 x_4^4 \tan(x_3)^3 \right)$$

$$g_1(\mathbf{x}) = \frac{\sec(x_3) (-g - 3 L x_4^2 \sec(x_3)) + (-3 L x_4^2 - 2 g \cos(x_3)) \sin(x_3) \tan(x_3)}{L}$$

# Pendolo inverso – Controllo Sliding Mode

## Passo III – progetto Sliding Mode Control

**N.B.** L'uscita sulla quale fare il tracking è quella definita in precedenza:

$$e = y - y_d \quad (\text{comunque } y_d = 0)$$

$$\begin{aligned} s(\mathbf{x}, t) &= \left( \frac{d}{dt} + \lambda \right)^3 e \\ &= \ddot{e} + 3\lambda\ddot{e} + 3\lambda^2\dot{e} + \lambda^3 e \end{aligned}$$

# Pendolo inverso – Controllo Sliding Mode

**INFINE**, poichè:

$$\begin{aligned} \dot{s} &= e^{(4)} + 3\lambda\ddot{e} + 3\lambda^2\dot{e} + \lambda^3e \\ &= y^{(4)} + 3\lambda\ddot{y} + 3\lambda^2\dot{y} + \lambda^3y \\ &= f_1(\mathbf{x}) + g_1(\mathbf{x})v + 3\lambda\ddot{y} + 3\lambda^2\dot{y} + \lambda^3y \\ &= b(\mathbf{x}) + a(\mathbf{x})v \end{aligned}$$

Dalla precedente si può verificare che la legge di controllo:

$$v = -\frac{b(\mathbf{x})}{a(\mathbf{x})} + K\text{sign}(s)$$

porta il sistema in Sliding Mode sulla superficie  $s(\mathbf{x},t)$  con opportuno  $K$

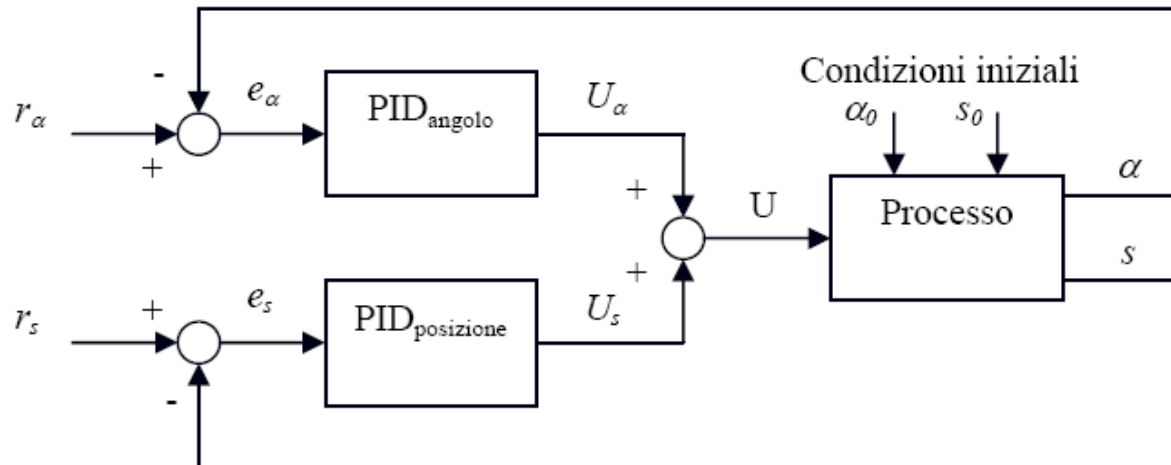
# Pendolo inverso – Controllo Fuzzy

Il controllore può essere progettato come sistema Fuzzy con 4 ingressi:

1.  $x - x_d$ , errore di posizione lineare
2.  $dx/dt$
3.  $\theta$
4.  $d\theta/dt$

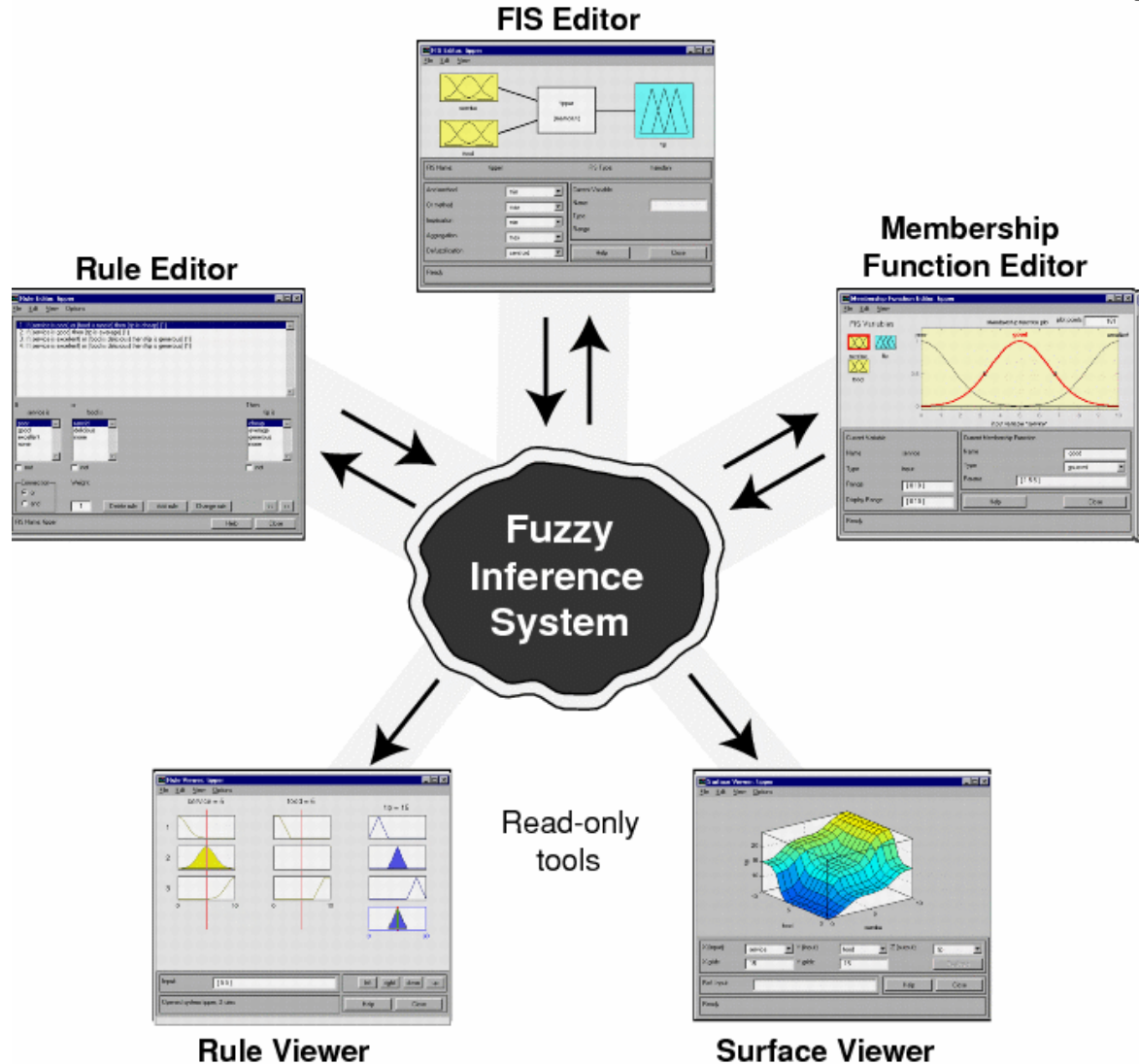
e una sola uscita, la forza agente sul carrello.

Oppure, è possibile progettare un doppio regolatore PID-like in parallelo, cioè le cui uscite si sommano per ottenere il valore della forza da applicare al carrello:



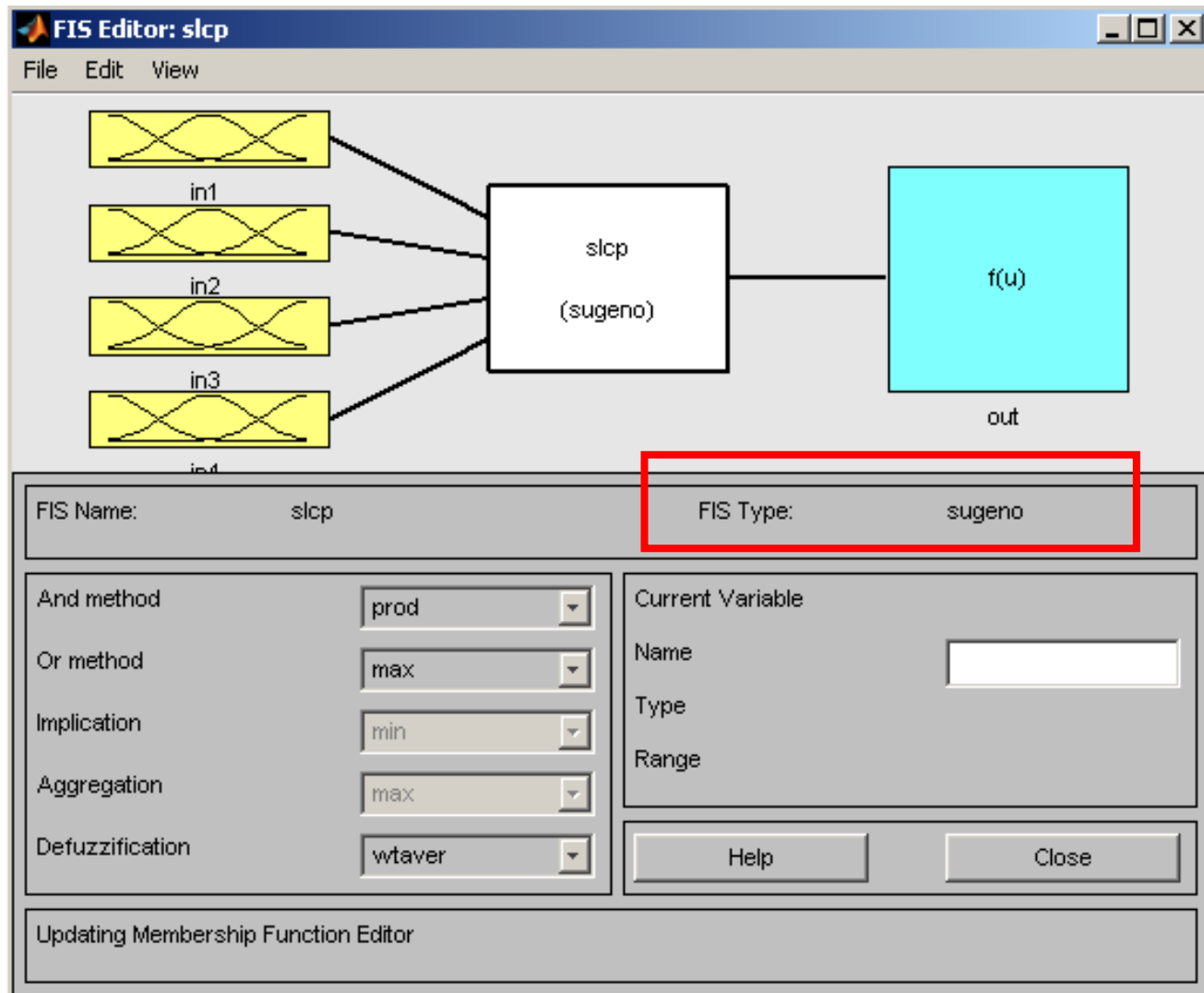
# Pendolo inverso – Controllo Fuzzy e MATLAB

## Strumenti del Fuzzy Logic Toolbox:



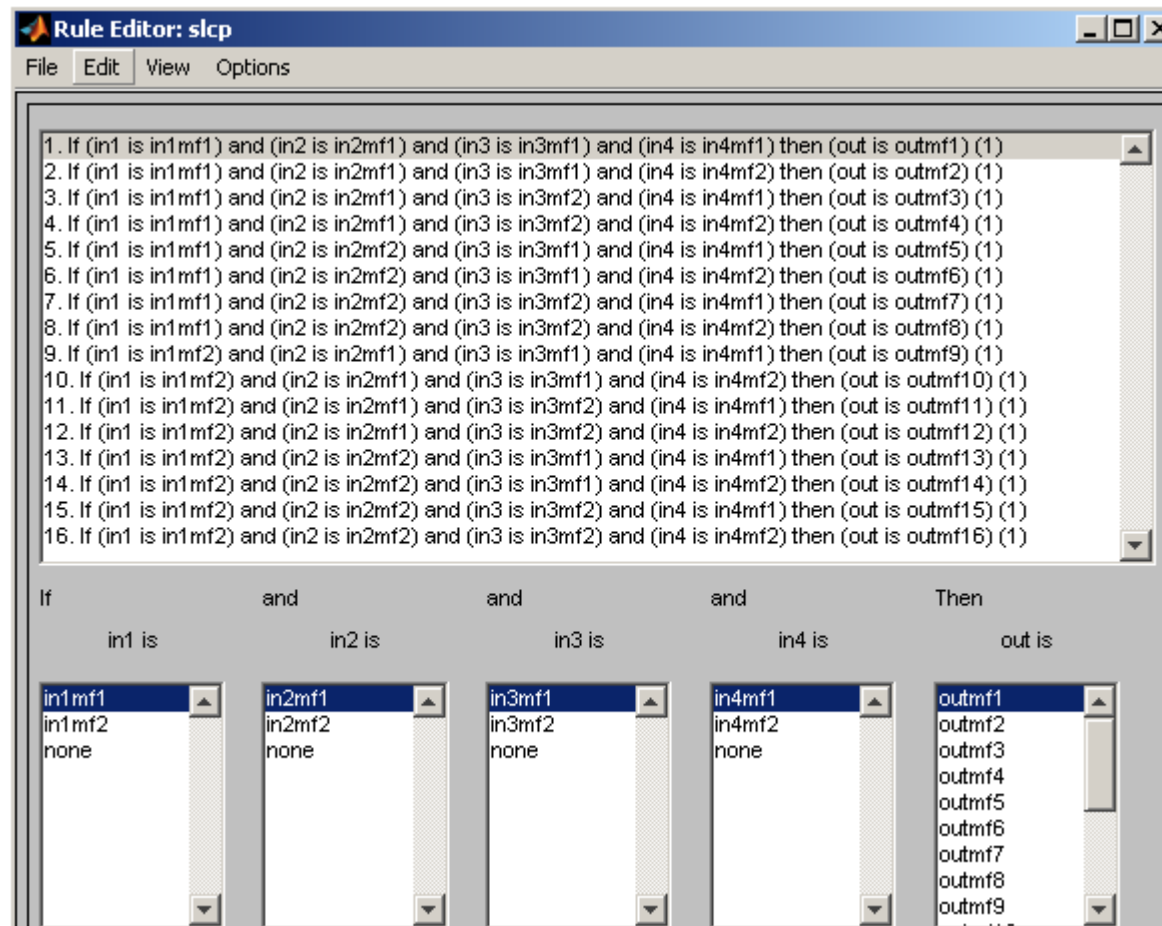
# Pendolo inverso – Controllo Fuzzy e MATLAB

Il Fuzzy Inference System per il controllo del pendolo inverso:



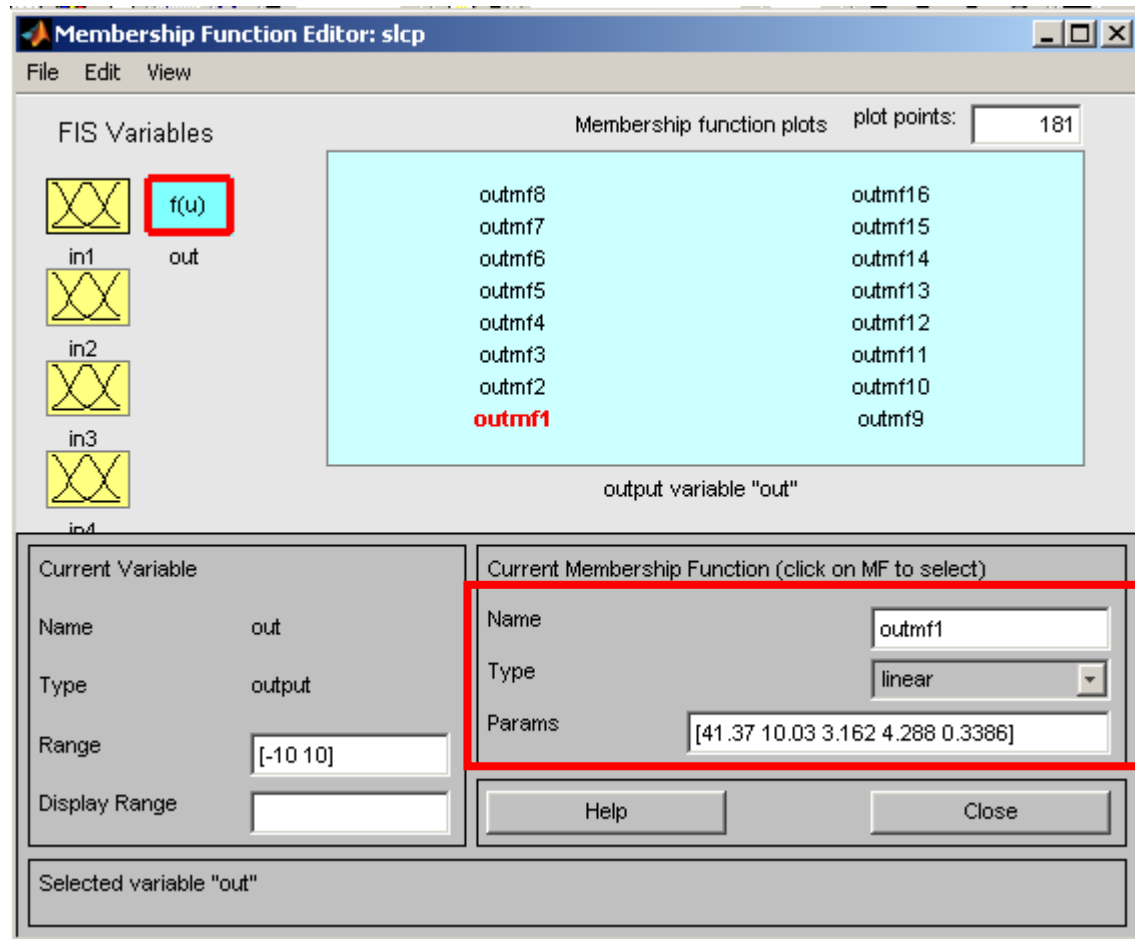
# Pendolo inverso – Controllo Fuzzy e MATLAB

Le 16 regole (visualizzabili come superfici 2D solo considerando 2 ingressi per volta):



# Pendolo inverso – Controllo Fuzzy e MATLAB

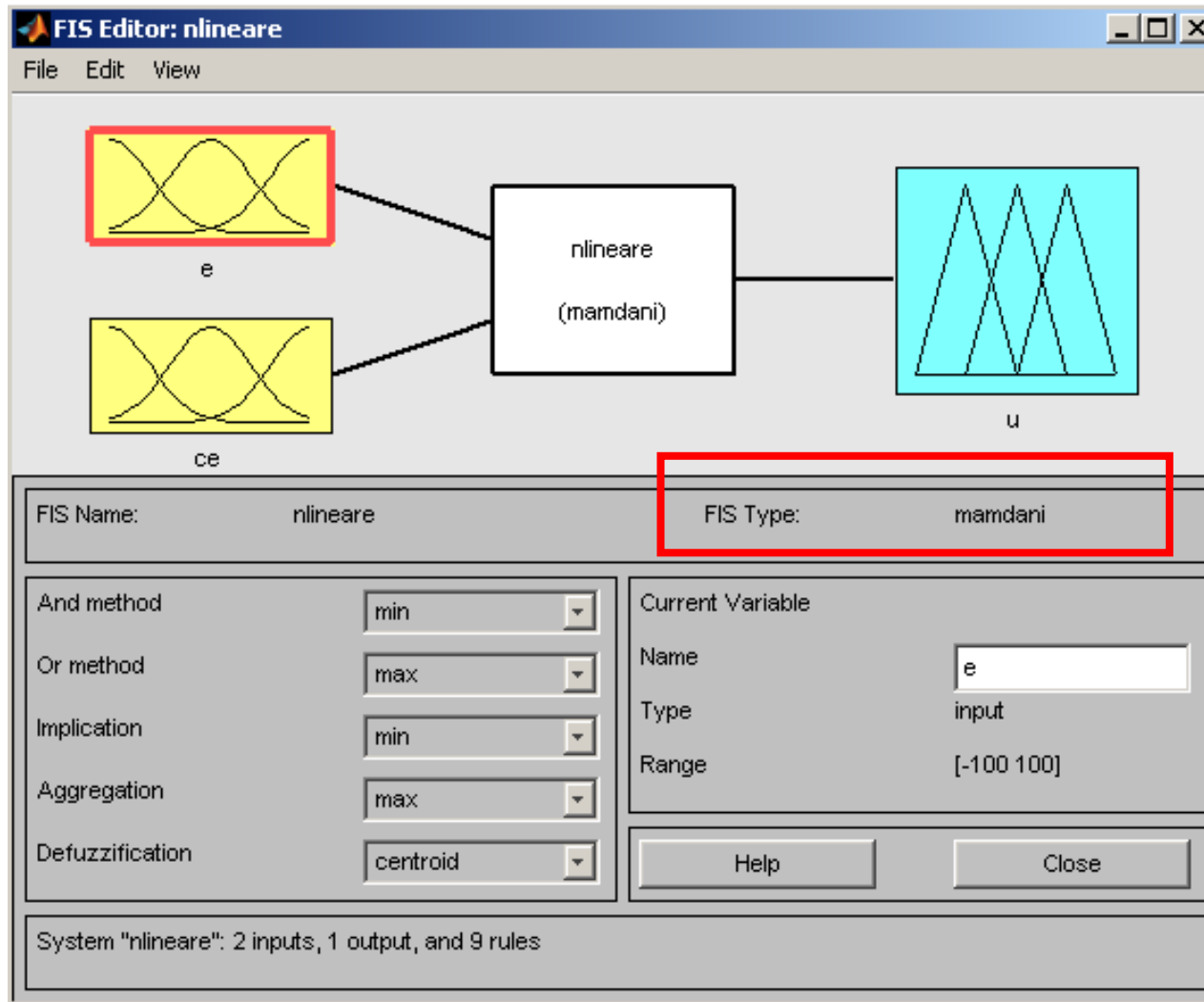
I parametri delle funzioni di uscita (modello Sugeno) del tipo:  
$$\text{out} = a \cdot \text{in1} + b \cdot \text{in2} + c \cdot \text{in3} + d \cdot \text{in4} + e$$





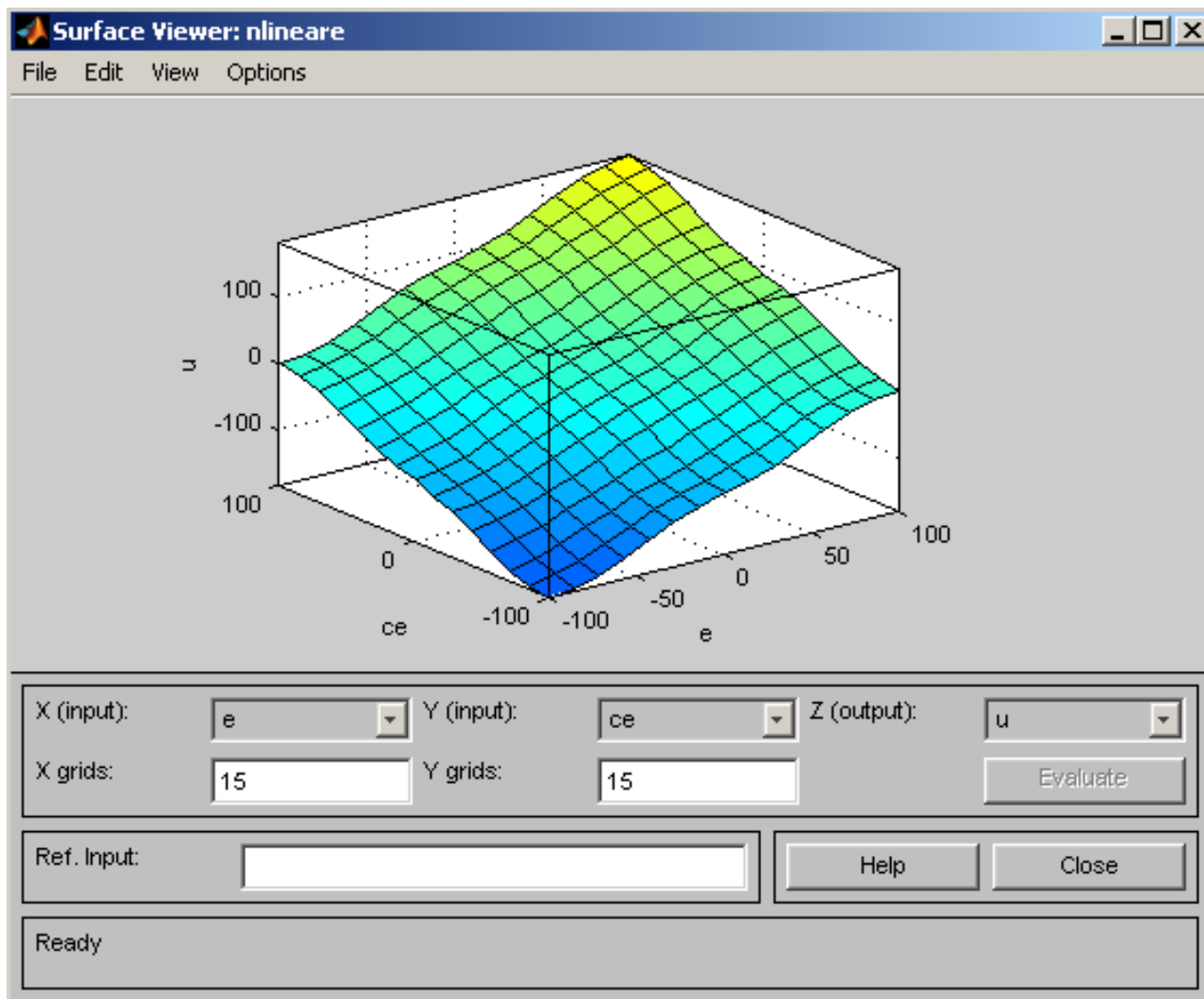
# Pendolo inverso – Controllo Fuzzy e MATLAB

Il Fuzzy Inference System per il PD (+ I non fuzzy)



# Pendolo inverso – Controllo Fuzzy e MATLAB

Le 9 regole, visualizzabili come superficie 2D (f. di E e CE)



# Pendolo inverso – Controllo con Reti Neurali



Poichè il sistema da controllare è instabile, non è possibile addestrare una rete neurale come **Inverse Dynamics**

Poichè il Neural Network Toolbox di MATLAB mette a disposizione un controllore di tipo **MRAC** come blocco per Simulink MA SOLO SISO, non può essere usato per il pendolo inverso su carrello.

L'unica soluzione "realizzabile" per simulazioni MATLAB è quella con **Supervised Learning**

E' necessario addestrare la rete sfruttando i dati generati da un modello di simulazione già stabilizzato da un controllore opportuno, sollecitato con variazioni casuali del set-point per portarlo fin quasi al limite dell'instabilità (i dati di addestramento per la rete neurale devono coprire un ampio spazio di funzionamento del sistema)

# Pendolo inverso – Reti Neurali e MATLAB

Una volta generati i dati da un modello Simulink del pendolo, stabilizzato con un controllore, si possono sfruttare le seguenti funzioni di NN toolbox:

```
net = newff(PR, [S1...SN1], {TF1...TFN1}, BTF, BLF, PF)
```

che crea una nuova rete feed-forward backpropagation,

```
net = train(net, P, T, [], [], val, test);
```

che la addestra sfruttando i dati contenuti in P e T, e

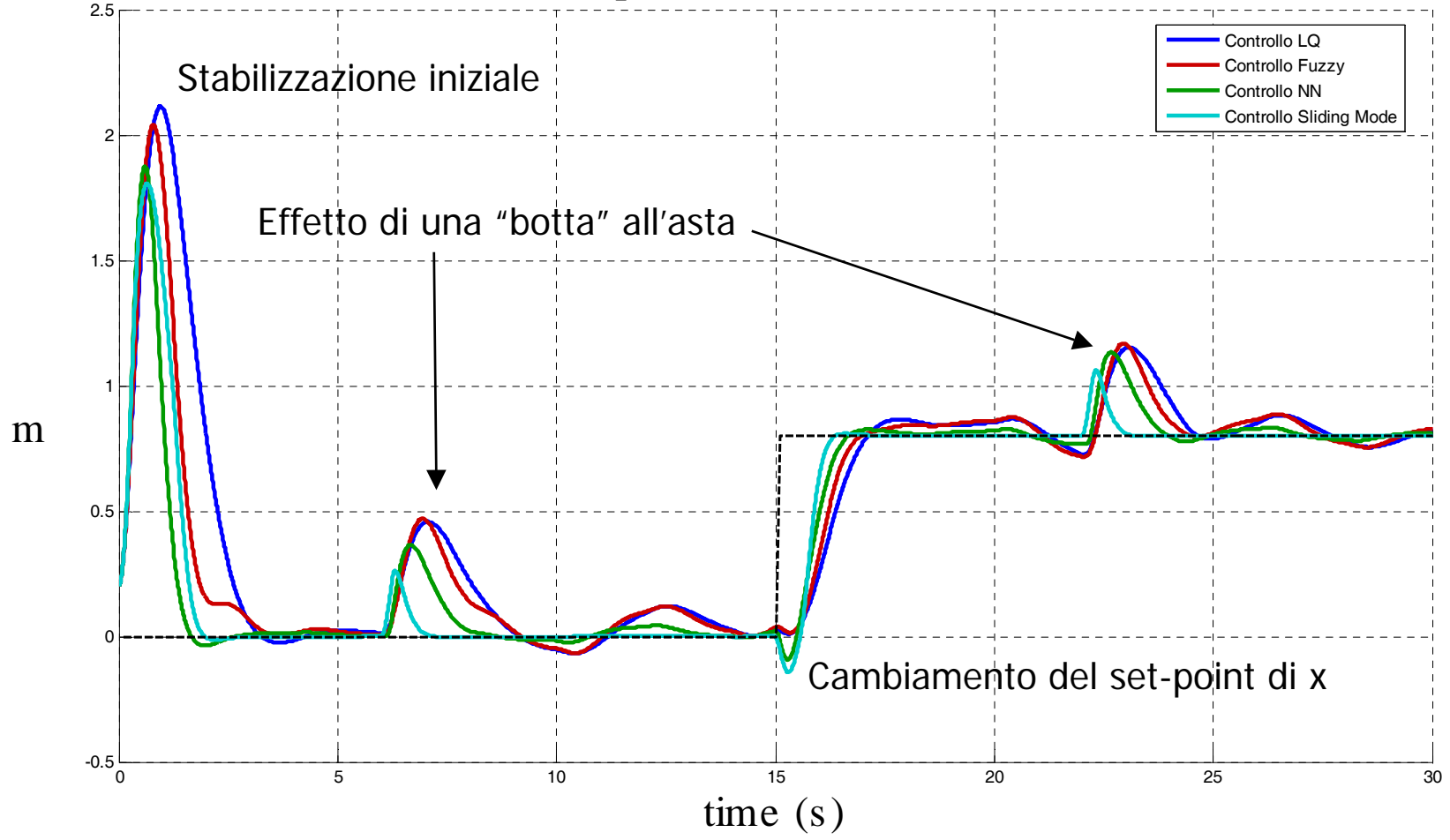
```
gensim(net, Ts);
```

che genera un blocco Simulink specifico per l'esecuzione con tempo di campionamento Ts dell'elaborazione definita dalla rete neurale

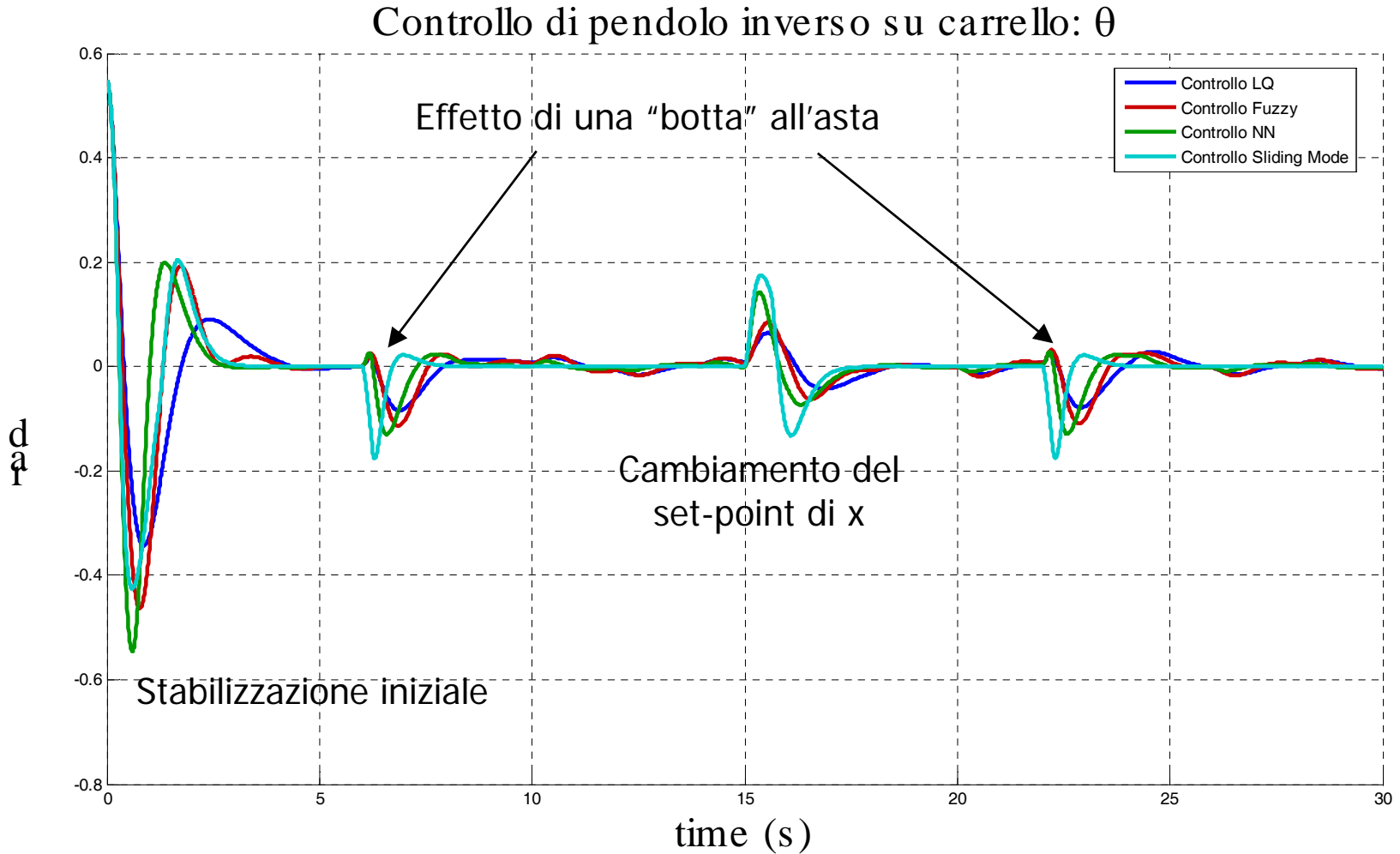
# Pendolo inverso – Simulazioni



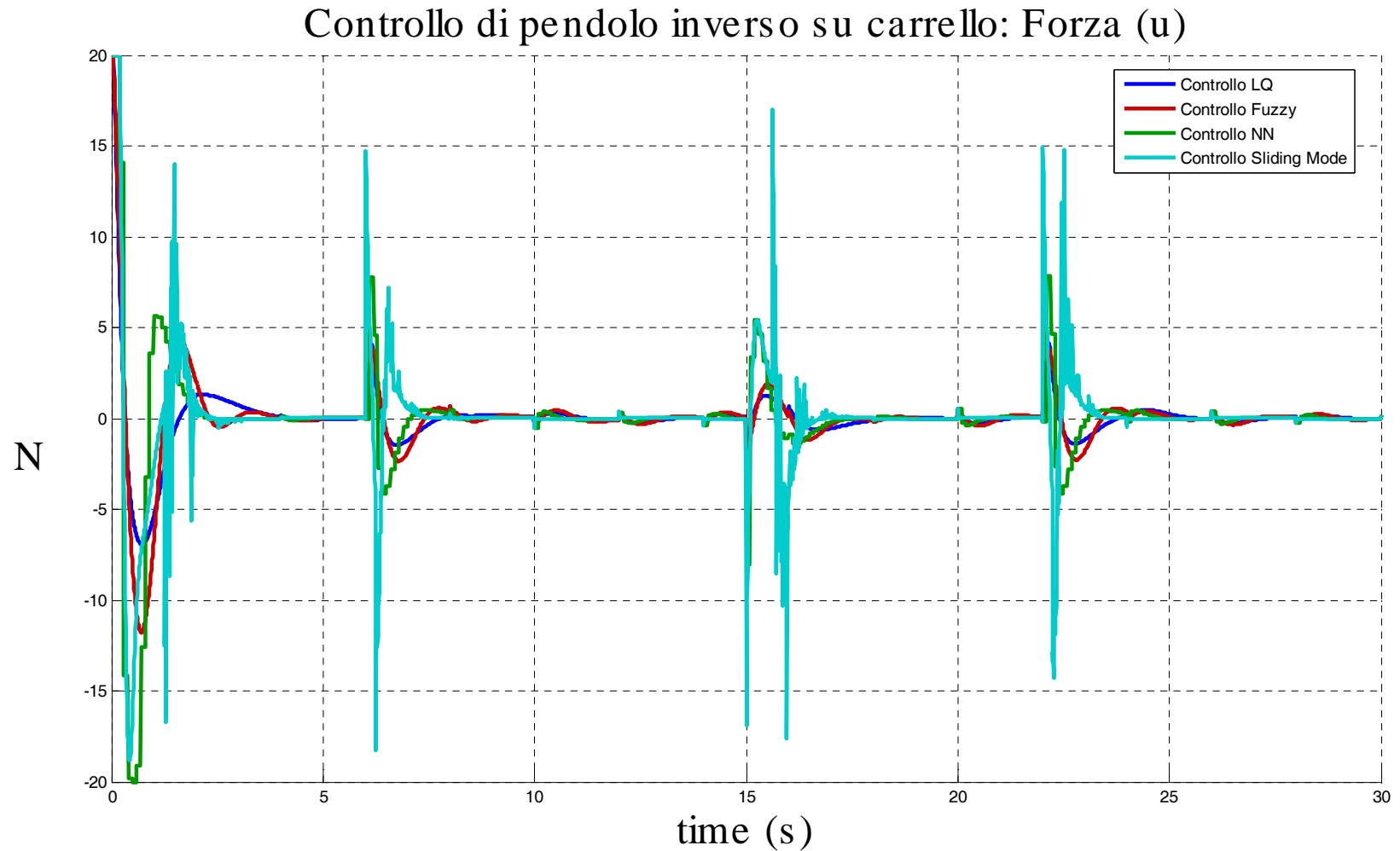
Controllo di pendolo inverso su carrello: x



# Pendolo inverso – Simulazioni



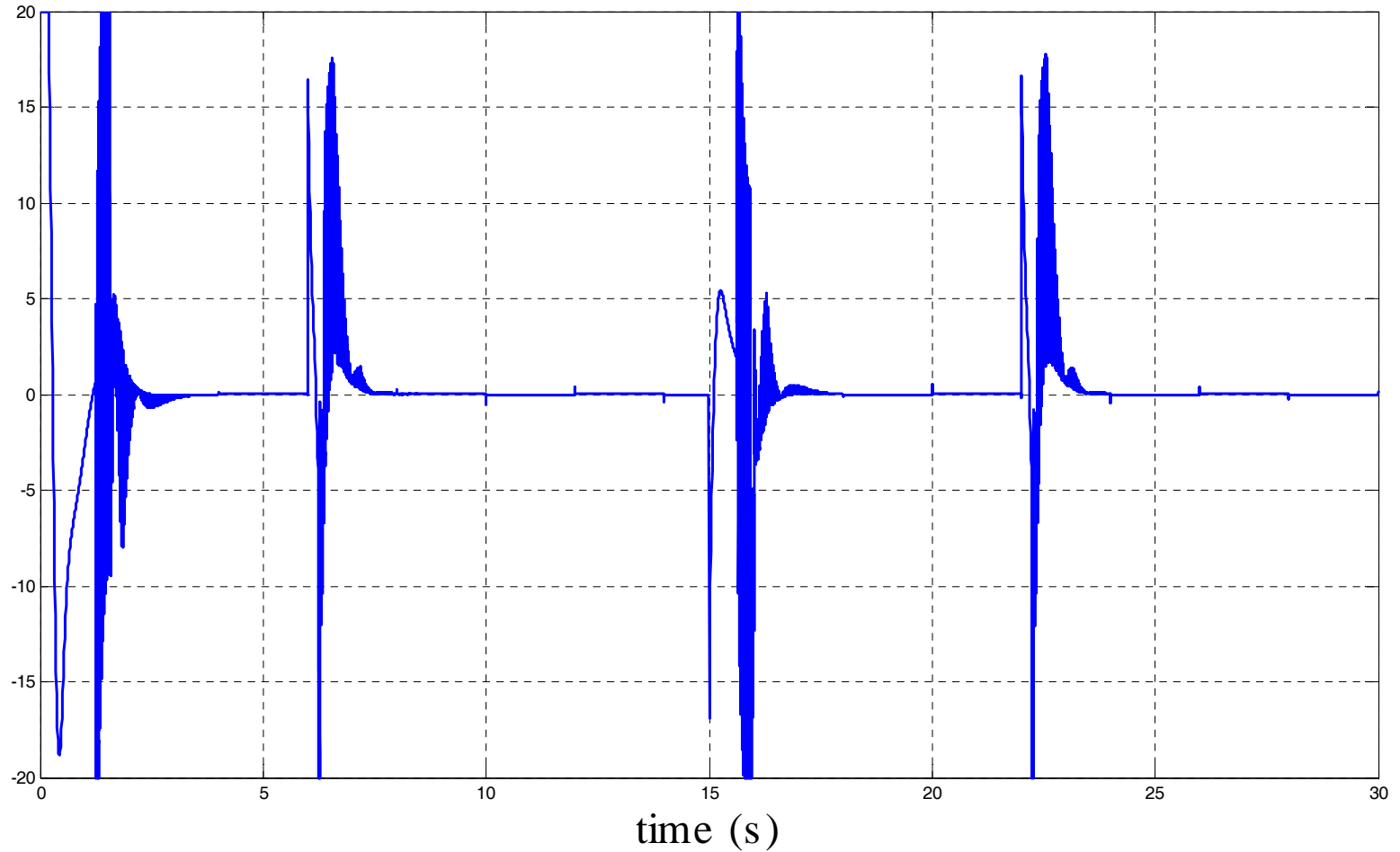
# Pendolo inverso – Simulazioni



# Pendolo inverso – Simulazioni (dettaglio..)



Forza richiesta dallo S.M.C.

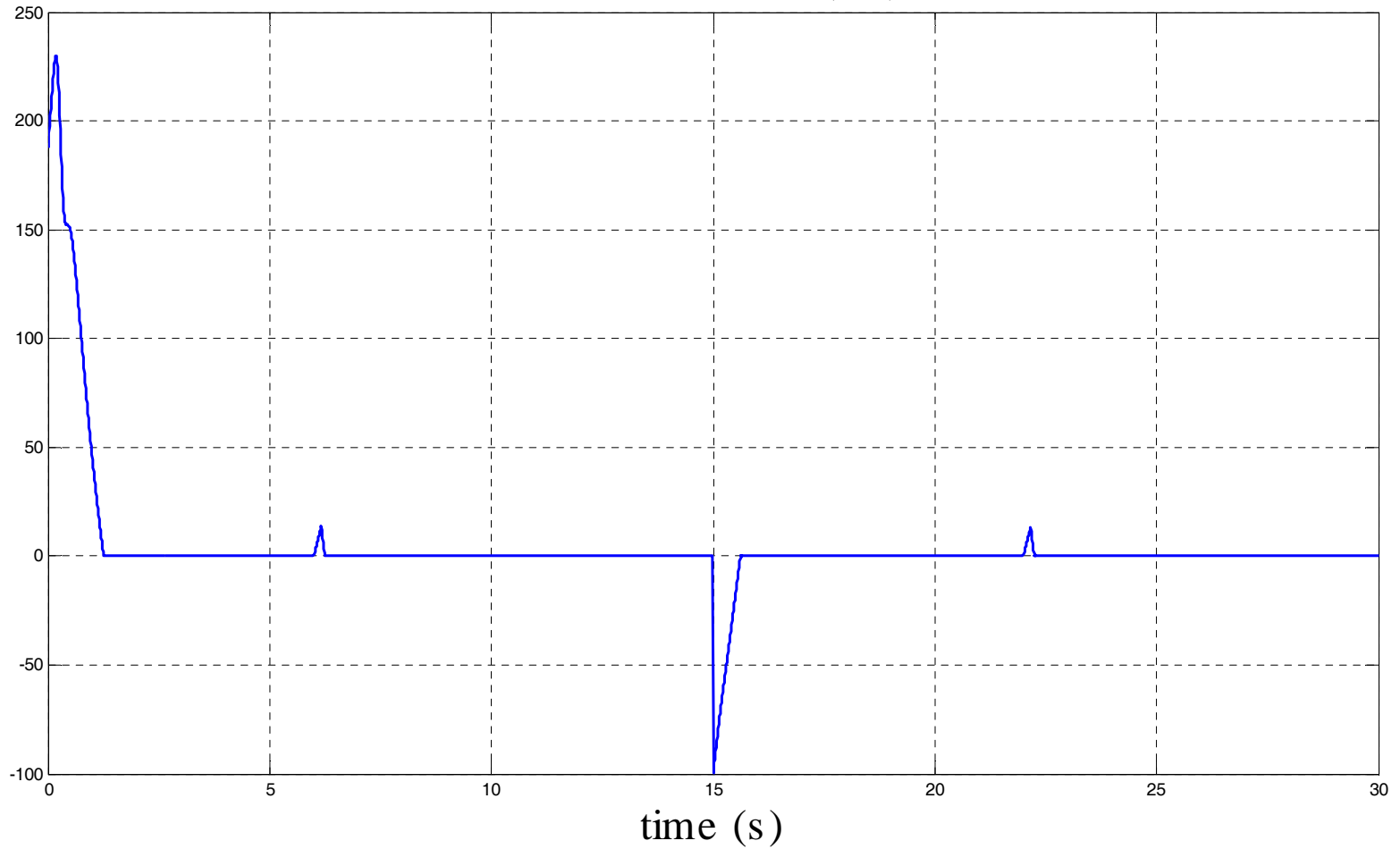




# Pendolo inverso – Simulazioni (dettaglio..)

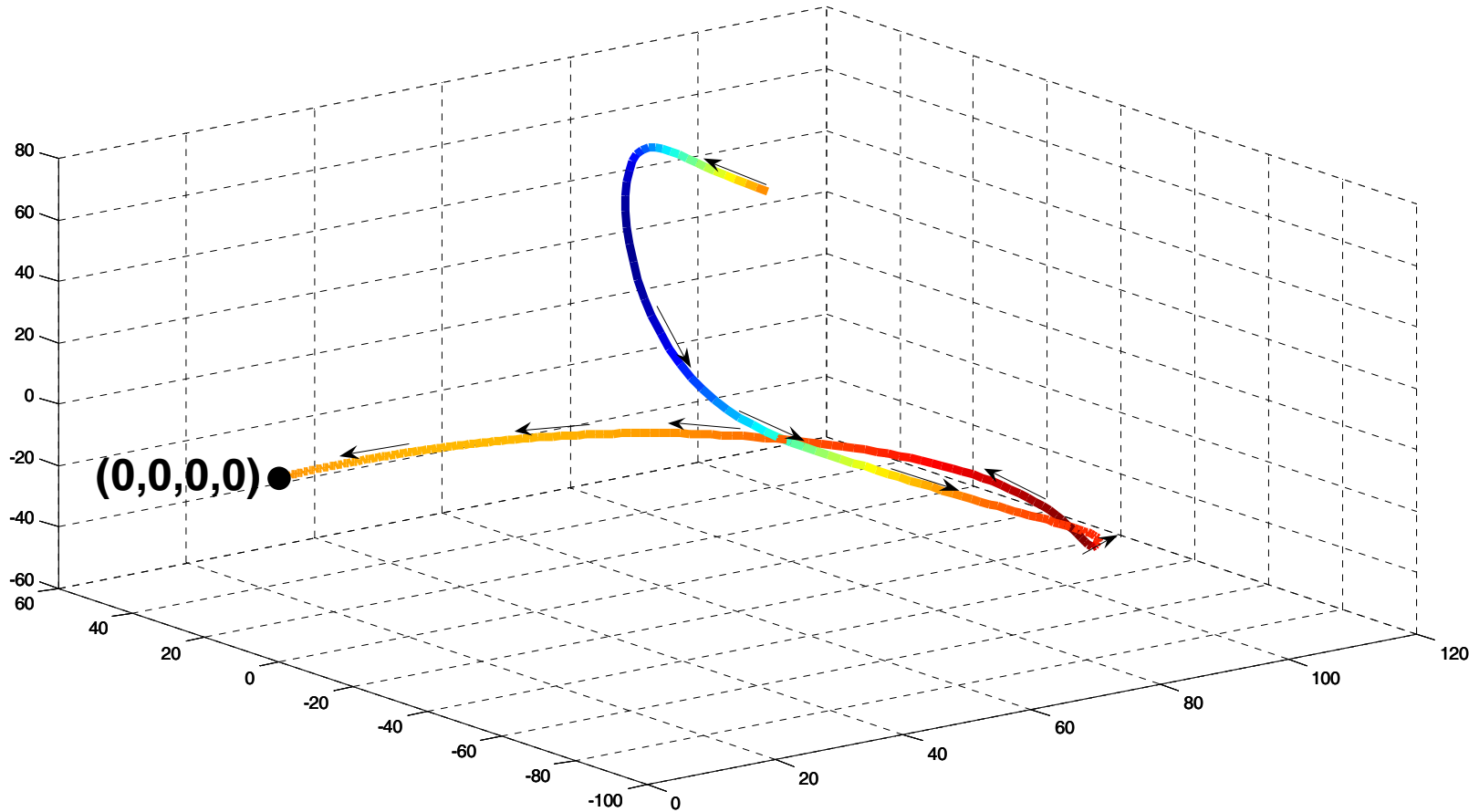


Andamento di  $S(X,t)$



# Pendolo inverso – Simulazioni (dettaglio..)

Superficie  $S(X,t)=0$  (spazio 4D)



# Pendolo inverso – Swing it up!!



Tutti i controllori visti in precedenza, sono in grado di stabilizzare la condizione con  $\theta = 0$  (pendolo in alto), purchè la condizione iniziale del pendolo non sia troppo inclinata.

Tuttavia, con opportune leggi di controllo, basate su considerazioni energetiche, è anche possibile effettuare il cosiddetto “swing-up” e riportare il pendolo da una posizione iniziale verso il basso a quella verso l’alto.

Nell’articolo di **Mark W. Spong** “**Energy Based Control of a Class of Underactuated Mechanical Systems**”, è presentato questo approccio, valido per alcuni sistemi detti appunto “sottoattuati”, cioè con più gradi di libertà rispetto agli ingressi, tra i quali anche il sistema Cart-Pole, cioè il pendolo inverso considerato in queste note..

# Pendolo inverso – Swing it up!!



## Cart-Pole swing-up!

