

Identificazione dei modelli

- Complesso dei procedimenti che consente di risalire **dai dati al modello**.
- **Identificazione a scatola trasparente:** il modello viene ricavato dalla descrizione delle parti costituenti il sistema dato mediante le leggi (*fisiche*) che regolano il comportamento delle parti stesse. Il problema dell'identificazione nasce quando un parametro è incognito e va stimato in base ad osservazioni sperimentali.
- **Identificazione a scatola nera:** si considera una famiglia di modelli che possono essere applicati in maniera del tutto generale. I parametri di tali modelli non hanno un'interpretazione fisica diretta, ma sono solo un mezzo per descrivere delle relazioni di ingresso-uscita del sistema.

- **Bibliografia essenziale:**

- Sergio Bittanti, *Identificazione dei Modelli e Sistemi Adattativi*, Pitagora Editrice, Bologna, 2004.
- T. Söderström e P. Stoica, *System Identification*, Prentice-Hall, 1989
- L. Ljung, *System Identification-theory for the user*, Prentice-Hall, 1999

Stima ai minimi quadrati

- Si consideri il modello di *regressione lineare* nei parametri

$$y_k = \phi_1(x_k)\alpha_1 + \phi_2(x_k)\alpha_2 + \dots + \phi_n(x_k)\alpha_n + \varepsilon_k$$

dove

- y è la variabile osservata;
- $\alpha_i, i = 1, \dots, n$ sono parametri incogniti;
- $\phi_i(\cdot), i = 1, \dots, n$ sono funzioni note che dipendono dalla variabile nota x
- ε è il *residuo*, ossia l'errore di modello.

Il modello ha come indice la variabile discreta k , la quale spesso ha il significato di variabile temporale.

- **Problema:** Avendo a disposizione le coppie $\{y_k, x_k\}$, $k = 1, \dots, N$ ($N > n$), si vogliono determinare i parametri $\alpha_i, i = 1, \dots, n$ in modo da minimizzare la funzione costo

$$V = \sum_{k=1}^N \varepsilon_k^2, \quad \text{cioè} \quad \min_{\alpha_i} \sum_{k=1}^N \varepsilon_k^2$$

- Con le notazioni

$$\mathbf{y} = [y_1, y_2, \dots, y_N]^T, \quad \boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N]^T, \quad \boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_n]^T$$

$$\boldsymbol{\phi}(k) = [\phi_1(x_k), \phi_2(x_k), \dots, \phi_n(x_k)]^T \quad \boldsymbol{\Phi} = \begin{bmatrix} \phi^T(1) \\ \phi^T(2) \\ \dots \\ \phi^T(N) \end{bmatrix}$$

la funzione di costo può essere riscritta come

$$V(\boldsymbol{\alpha}) = \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon}$$

e il problema riformulato in forma vettoriale come

$$\begin{cases} \min_{\boldsymbol{\alpha}} V(\boldsymbol{\alpha}) = \min_{\boldsymbol{\alpha}} \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} = \mathbf{y} - \boldsymbol{\Phi} \boldsymbol{\alpha} \end{cases}$$

- Supponendo che la matrice $\Phi^T \Phi$ sia definita positiva, la funzione $V(\alpha)$ ha un unico minimo dato da

$$\hat{\alpha} = (\Phi^T \Phi)^{-1} \Phi^T y$$

La matrice $(\Phi^T \Phi)^{-1} \Phi^T$ è detta *pseudoinversa* di Φ .

Il corrispondente valore minimo di $V(\alpha)$ risulta

$$\min_{\alpha} V(\alpha) = V(\hat{\alpha}) = y^T y - y^T \Phi (\Phi^T \Phi)^{-1} \Phi^T y.$$

Dimostrazione:

$$\begin{aligned} V(\alpha) &= \varepsilon^T \varepsilon \\ &= (y - \Phi \alpha)^T (y - \Phi \alpha) \\ &= y^T y - y^T \Phi \alpha - \alpha^T \Phi^T y + \alpha^T \Phi^T \Phi \alpha \\ &= [y^T y - y^T \Phi (\Phi^T \Phi)^{-1} y] + [\alpha - (\Phi^T \Phi)^{-1} \Phi^T y]^T (\Phi^T \Phi) [\alpha - (\Phi^T \Phi)^{-1} \Phi^T y] \end{aligned}$$

Il primo termine non dipende da α , mentre poichè si è ipotizzato che $(\Phi^T \Phi)$ fosse definito positivo il secondo termine è sempre maggiore o uguale a zero. Perciò $V(\alpha)$ può essere minimizzato assumendo che questo termine sia esattamente uguale a zero (cioè $\alpha - (\Phi^T \Phi)^{-1} \Phi^T y = 0$). Da cui il risultato.

- il calcolo della pseudo-inversa $(\Phi^T \Phi)^{-1} \Phi^T$ può dare luogo a errori numerici nel caso la matrice $\Phi^T \Phi$ risulti malcondizionata numericamente. In questi casi conviene utilizzare algoritmi ottimizzati, ad.es. basati sulla fattorizzazione QR, che consentono di risolvere il sistema lineare $y = \Phi \alpha$ senza il calcolo dell'inversa di $\Phi^T \Phi$. Proprio sulla fattorizzazione QR si basa la funzione per il calcolo della pseudo-inversa "pinv" disponibile in matlab.

Uso: >> alpha = pinv(Phi)*y

Fattorizzazione QR

Una generica matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$, può essere fattorizzata come

$$\mathbf{A} = \mathbf{QR}$$

dove $\mathbf{Q} \in \mathbb{R}^{m \times m}$ è una matrice ortogonale, e $\mathbf{R} \in \mathbb{R}^{m \times n}$ è una matrice triangolare superiore.

Dato un sistema di equazioni lineari $\mathbf{Ax} = \mathbf{b}$, considerando la suddetta fattorizzazione e il fatto che per una matrice ortogonale $\mathbf{Q}^T \mathbf{Q} = \mathbf{Q} \mathbf{Q}^T = \mathbf{I}$, è possibile riscriverlo come

$$\begin{aligned} \mathbf{QRx} &= \mathbf{b} \\ \mathbf{Q}^T \mathbf{QRx} &= \mathbf{Q}^T \mathbf{b} \\ \mathbf{Rx} &= \mathbf{Q}^T \mathbf{b} = \mathbf{c}. \end{aligned}$$

Essendo \mathbf{R} triangolare superiore il sistema può essere risolto con una procedura di risoluzione all'indietro.

Esempio Matlab

```
A = rand(6,3)
```

```
A =
```

```
0.8181    0.9730    0.0835
0.8175    0.6490    0.1332
0.7224    0.8003    0.1734
0.1499    0.4538    0.3909
0.6596    0.4324    0.8314
0.5186    0.8253    0.8034
```

```
b = [1 2 3 4 5 6]'
```

```
b =
```

```
1
2
3
4
5
6
```

```
[Q,R]=qr(A)
```

```
Q =
```

```
-0.5088    0.2278   -0.4345   -0.3714   -0.1610   -0.5802
-0.5084   -0.3699   -0.2201    0.4297   -0.5112    0.3320
-0.4492    0.0924   -0.2438    0.0827    0.7778    0.3440
-0.0932    0.5511    0.2032    0.7342    0.0111   -0.3272
-0.4102   -0.4671    0.6729   -0.0467    0.1894   -0.3503
-0.3225    0.5299    0.4574   -0.3595   -0.2679    0.4527
```

R =

-1.6082	-1.6703	-0.8246
0	0.5410	0.2386
0	0	0.8985
0	0	0
0	0	0
0	0	0

c = Q'*b

c =

-7.2317
2.8138
5.3159
1.2827
0.5345
0.7721

- La soluzione al problema di stima ai minimi quadrati

$$\hat{\alpha} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$$

può essere dedotta anche seguendo una interpretazione geometrica del problema. Si riscriva infatti

$$y_k = \phi_1(x_k)\alpha_1 + \phi_2(x_k)\alpha_2 + \dots + \phi_n(x_k)\alpha_n + \varepsilon_k, \text{ per } k = 1, \dots, N$$

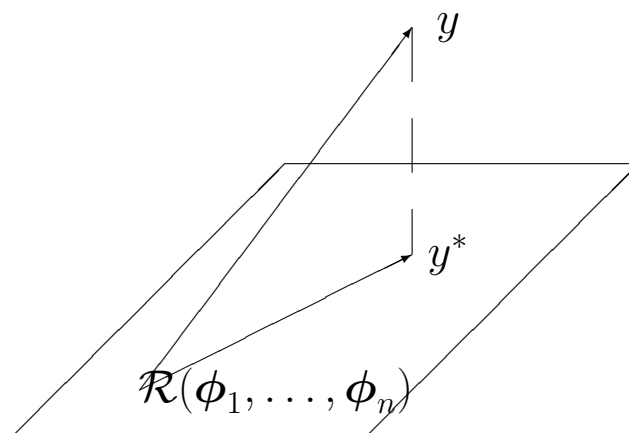
come

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix} = \begin{bmatrix} \phi_1(x_1) \\ \phi_1(x_2) \\ \dots \\ \phi_1(x_N) \end{bmatrix} \alpha_1 + \dots + \begin{bmatrix} \phi_n(x_1) \\ \phi_n(x_2) \\ \dots \\ \phi_n(x_N) \end{bmatrix} \alpha_n = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \dots \\ \varepsilon_N \end{bmatrix}$$

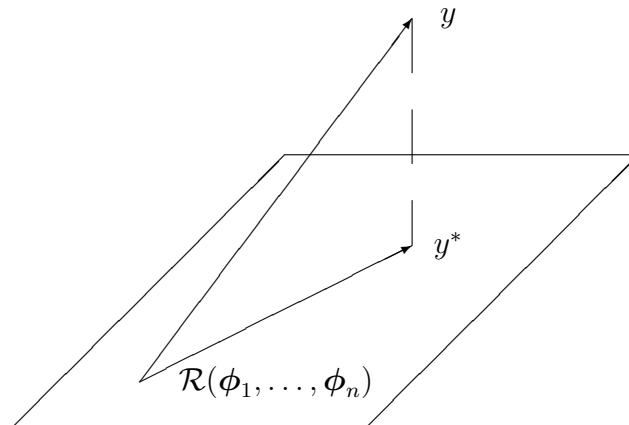
ovvero, con ovvia corrispondenza dei simboli,

$$\mathbf{y} - \phi_1\alpha_1 - \phi_2\alpha_2 - \dots - \phi_n\alpha_n = \boldsymbol{\varepsilon}$$

Si considerino ora $\mathbf{y}, \phi_1, \dots, \phi_n$ come vettori in uno spazio Euclideo a N dimensioni con norma $\|\mathbf{x}\| = \mathbf{x}^T \mathbf{x}$. Il problema dei minimi quadrati può allora essere visto come quello della approssimazione del vettore \mathbf{y} con una combinazione lineare dei vettori $\phi_i, i = 1, \dots, n$ in modo che la norma dell'errore $\boldsymbol{\varepsilon}$ sia minima. La soluzione, indicata con \mathbf{y}^* , è data chiaramente dalla proiezione ortogonale di \mathbf{y} sul sottospazio $\mathcal{R}(\phi_i)$ generato dai vettori ϕ_1, \dots, ϕ_n .



Dimostrazione:



Poiché $(y - y^*) \perp \phi_i, \dots, \phi_n$ si può scrivere

$$\begin{cases} (y - y^*)^T \phi_1 = 0 \\ \dots \\ (y - y^*)^T \phi_n = 0 \end{cases}$$

Questo sistema, tenendo conto della condizione $y^* = \alpha_1 \phi_1 + \dots + \alpha_n \phi_n$, equivale a

$$\begin{bmatrix} \phi_1^T \phi_1 & \phi_1^T \phi_2 & \dots & \phi_1^T \phi_n \\ \phi_2^T \phi_1 & \phi_2^T \phi_2 & \dots & \phi_2^T \phi_n \\ \vdots & \vdots & \ddots & \vdots \\ \phi_n^T \phi_1 & \phi_n^T \phi_2 & \dots & \phi_n^T \phi_n \end{bmatrix} \alpha = \begin{bmatrix} y^T \phi_1 \\ y^T \phi_2 \\ \vdots \\ y^T \phi_n \end{bmatrix}$$

che è esattamente

$$(\Phi^T \Phi) \hat{\alpha} = \Phi^T y$$

ovvero

$$\hat{\alpha} = (\Phi^T \Phi)^{-1} \Phi^T y.$$

La condizione che $(\Phi^T \Phi)$ sia non singolare è equivalente alla lineare indipendenza dei vettori ϕ_1, \dots, ϕ_n , che costituiscono una base del sottospazio \mathcal{R} .

Applicazione della stima ai minimi quadrati ai sistemi dinamici

- Si assume generalmente che il sistema da identificare sia un sistema a dati campionati descritto da una funzione di trasferimento discreta

$$G(z) = \frac{b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}} = \frac{Y(z)}{U(z)}$$

per cui l'uscita y al generico istante k può essere espressa come

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) - \dots - a_n y(k-n) + b_1 u(k-1) + b_2 u(k-2) + \dots + b_n u(k-n) + \varepsilon(k) \quad (1)$$

in cui i parametri $a_i, b_i, i = 1, \dots, n$ sono non noti e $\varepsilon(k)$ è il residuo del modello, ovvero la differenza tra l'uscita vera e quella stimata:

$$\varepsilon(k) = y(k) - \hat{y}(k, \alpha)$$

- Il modello (1) è un caso particolare di un modello noto in letteratura con il nome di modello **ARX** (*AutoRegresive model with eXternal input*), la cui struttura generale è

$$y(k) + \sum_{i=1}^{n_a} a_i y(k-i) = \sum_{i=0}^{n_b-1} b_i u(k-i-n_k) + \varepsilon(k)$$

da cui

$$\frac{Y(z)}{U(z)} = z^{-n_k} \frac{b_0 + b_1 z^{-1} + \dots + b_{n_b-1} z^{-(n_b-1)}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{n_a} z^{-n_a}}$$

con l'ordine del modello (n_a, n_b) e il numero di ritardi puri n_k tra ingresso e uscita che possono assumere valori arbitrari.

- Supponendo di avere a disposizione N campioni della variabile manipolabile $u(k)$ ($k = 1, \dots, N$) e i corrispondenti campioni dell'uscita $y(k)$, è possibile scrivere il seguente sistema di equazioni lineari nei parametri incogniti

$$y(n+1) = -a_1 y(n) - \dots - a_n y(1) + b_1 u(n) + \dots + b_n u(1) + \varepsilon(n+1)$$

$$y(n+2) = -a_1 y(n+1) - \dots - a_n y(2) + b_1 u(n+1) + \dots + b_n u(2) + \varepsilon(n+2)$$

⋮

$$y(N) = -a_1 y(N-1) - \dots - a_n y(N-n) + b_1 u(N-1) + \dots + b_n u(N-n) + \varepsilon(N)$$

che riscritto in forma vettoriale con dà luogo a un problema di stima ai minimi quadrati

$$\mathbf{y} = \Phi \boldsymbol{\alpha} + \boldsymbol{\varepsilon}$$

dove

$$\mathbf{y} = \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(N) \end{bmatrix}, \quad \boldsymbol{\alpha} = \begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ -a_n \\ b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon(n+1) \\ \varepsilon(n+2) \\ \vdots \\ \varepsilon(N) \end{bmatrix}$$

e

$$\Phi = [\Phi_1 \quad \Phi_2]$$

con

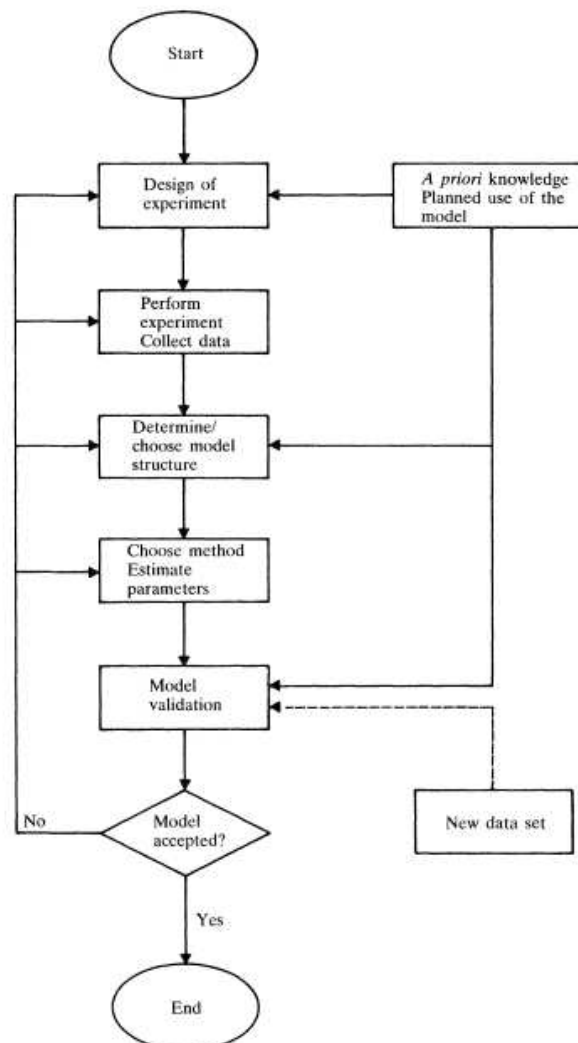
$$\Phi_1 = \begin{bmatrix} y(n) & y(n-1) & \cdots & y(1) \\ y(n+1) & y(n) & \cdots & y(2) \\ y(n+2) & y(n+1) & \cdots & y(3) \\ \vdots & & & \\ y(N-1) & y(N-2) & \cdots & y(N-n) \end{bmatrix}$$

$$\Phi_2 = \begin{bmatrix} u(n) & u(n-1) & \cdots & u(1) \\ u(n+1) & u(n) & \cdots & u(2) \\ u(n+2) & u(n+1) & \cdots & u(3) \\ \vdots & & & \\ u(N-1) & u(N-2) & \cdots & u(N-n) \end{bmatrix}$$

- La soluzione al problema di stima che minimizza la somma dei quadrati dei residui è data da

$$\hat{\boldsymbol{\alpha}} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$$

- La scelta e validazione del modello sono il punto iniziale e il punto terminale dell'intero processo di identificazione.
- Con riferimento ai sistemi dinamici lineari, la scelta del modello consiste nella selezione dell'ordine n del sistema, come rappresentato in (1) (o alternativamente degli ordine n_a, n_b, n_k del modello arx).
- La bontà del modello (in relazione al successivo impiego) può essere quantificata testando la capacità del modello stesso di riprodurre i dati sperimentali (possibilmente non utilizzati per l'identificazione del modello stesso), ad es. verificando che $|\varepsilon(k)| < C$.
- Gli errori di modello sono tipicamente dovuti a:
 - ingressi non misurati;
 - una struttura di modello troppo semplice (dinamiche non modellate, nonlinearità non previste).



- La procedura di identificazione si basa sul fatto che il periodo T con cui i dati sono campionati sia stato scelto opportunamente. T sarà chiaramente il tempo di campionamento del sistema tempo-discreto che si andrà ad ottenere.

Una regola pratica per la scelta di T è

$$\lambda_m T \leq \alpha, \text{ con } \alpha = 0.1 \div 0.5$$

dove λ_m è il modulo del più grande degli autovalori del modello tempo-continuo che descrive il sistema in oggetto. Chiaramente λ_m risulta incognito (prima del processo di indentificazione), ma in qualche maniera può essere stimato da un'analisi preliminare della risposta del sistema (e delle costanti di tempo che approssimativamente sembrano caratterizzarla).

- Se lo scopo del processo di indentificazione del sistema è la realizzazione di uno schema di controllo, T può essere scelto sulla base delle specifiche richieste, e in particolare della velocità di risposta del sistema, che come noto dipende dalla banda passante dello stesso. In questo caso si può assumere

$$T \leq \alpha \frac{2\pi}{\omega_c}, \text{ con } \alpha = 0.1 \div 0.5$$

dove ω_c è la pulsazione di taglio del sistema chiuso in retroazione.

- Scegliere un periodo di campionamento T il più piccolo possibile non è una buona soluzione, per due ragioni:
 1. aumento del carico computazionale
 2. problema di identificazione mal condizionato da un punto di vista numerico

Effetti del periodo di campionamento su un sistema a dati campionati:
come noto, dato un sistema tempo-continuo

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t)\end{aligned}$$

il corrispondente sistema tempo-discreto, ottenuto campionando i segnali con un periodo T risulta

$$\begin{aligned}\mathbf{x}((k+1)T) &= \mathbf{F}\mathbf{x}(kT) + \mathbf{G}\mathbf{u}(kT) \\ \mathbf{y}(kT) &= \mathbf{H}\mathbf{x}(kT)\end{aligned}$$

con $\mathbf{F} = e^{\mathbf{A}T}$ e $\mathbf{G} = \int_0^T e^{\mathbf{A}\sigma}\mathbf{B}d\sigma$.

Se quindi consideriamo gli autovalori della matrice \mathbf{F} , detti $\lambda_i, i = 1, \dots, n$ gli autovalori della matrice \mathbf{A} , dai risultati generali sulla matrice esponenziale si deduce immediatamente che gli autovalori di \mathbf{F} sono dati da $e^{\lambda_i T}, i = 1, \dots, n$. In particolare, se $\lambda_i = \alpha_i + j\beta_i$, l'autovalore corrispondente di \mathbf{F} risulta

$$e^{\alpha_i T} e^{j\beta_i T}$$

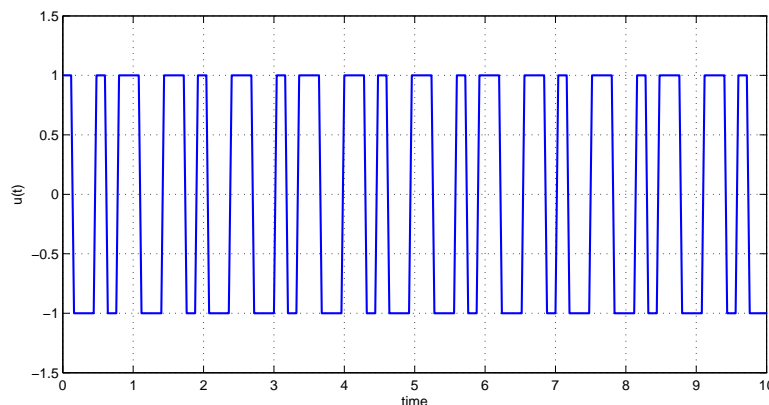
il cui modulo è semplicemente $e^{\alpha_i T}$. Conseguentemente risulta chiaro come al diminuire di T , gli autovalori del sistema a dati-campionati siano confinati in una regione sempre più stretta nell'intorno del cerchio unitario anche se i corrispondenti poli del sistema tempo-continuo sono a parte reale (ben) negativa. Tutto ciò implica che a seguito di errori numerici, in particolare in presenza di rumore di misura, i poli del sistema che si sta cercando di identificare possano risultare a modulo maggiore di 1 (quindi instabili) anche se il sistema fisico che ha generato i dati su cui l'identificazione è stata svolta è un sistema stabile. In questi casi si può tentare di migliorare la stima dei parametri aumentando il tempo di campionamento T .

- La scelta del segnale di ingresso $u(k)$ risulta critica per il successo dell'operazione di identificazione.
- E' necessario garantire che $u(k)$ sia “*persistentemente eccitante*”, cioè che sia in grado di eccitare tutti i modi del sistema durante il processo di identificazione.
- I segnali di riferimento comunemente utilizzati sono

– *funzione a gradino*:

$$u(t) = \begin{cases} 0 & t < 0 \\ u_0 & t \geq 0 \end{cases}$$

– *sequenza binaria pseudorandom (PRBS)*: sequenza che commuta tra due livelli con un certo periodo



– *somma di sinusoidi*:

$$u(t) = \sum_{j=1}^m a_j \sin(\omega_j t + \varphi_j)$$

in cui le pulsazioni ω_j sono distinte. Nota: il numero di sinusoidi dovrà essere maggiore dell'ordine del modello assunto.

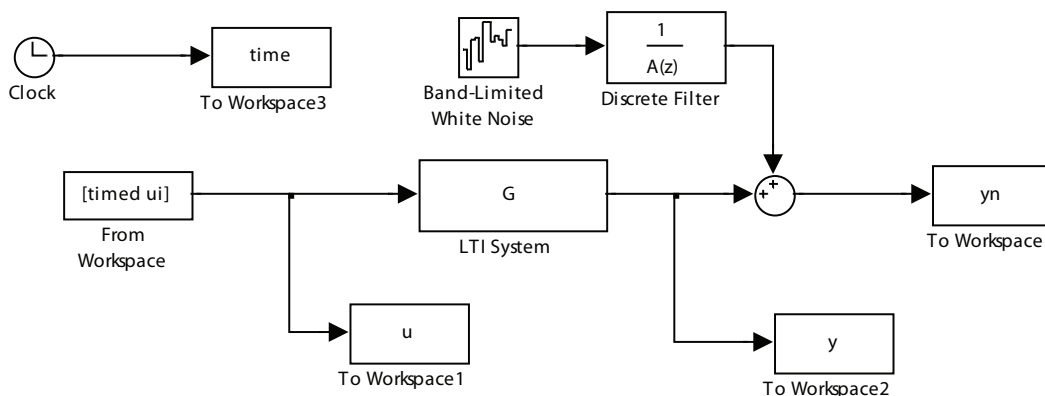
Esempio Matlab (IdentificazioneG.m)

- Si vuole procedere all'identificazione del sistema

$$G(s) = \frac{5(s+3)}{(s+0.2)(s+0.8)(s+5)} \xrightarrow{T=0.1s} G_d(z) = \frac{B(z)}{A(z)} = \frac{0.02274z^2 + 0.003743z - 0.01525}{z^3 - 2.51z^2 + 2.059z - 0.5488}$$

$$= \frac{B(z^{-1})}{A(z^{-1})} = \frac{0.02274z^{-1} + 0.003743z^{-2} - 0.01525z^{-3}}{1 - 2.51z^{-1} + 2.059z^{-2} - 0.5488z^{-3}}$$

```
>> G = zpk(-3, [-0.2 -0.8, -5], 5);
>> Gd = c2d(G,T, 'zoh');
```



- Si fa l'ipotesi che i dati campionati siano affetti da rumore bianco. Nello schema simulativo esso viene filtrato da $1/A(z^{-1})$, in questo modo i dati corrispondono effettivamente ad un modello arx:

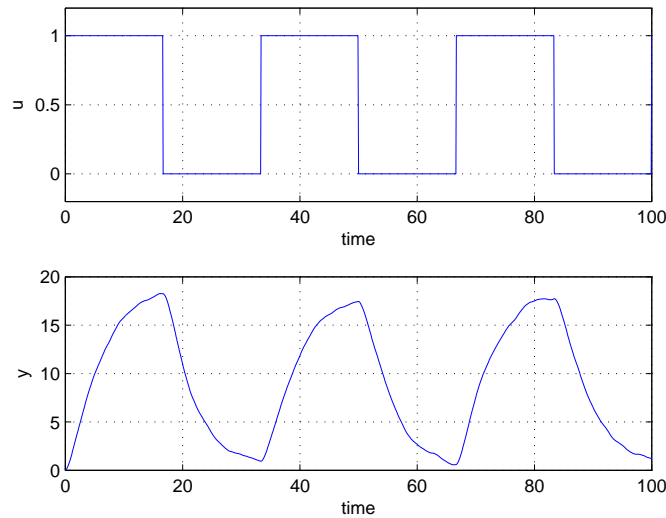
$$y_n(k) = \frac{B(z^{-1})}{A(z^{-1})}u(k) + \frac{1}{A(z^{-1})}\varepsilon(k) \Rightarrow A(z^{-1})y_n(k) = B(z^{-1})u(k) + \varepsilon(k)$$

Senza il filtro sul rumore, cioè

$$y_n(k) = \frac{B(z^{-1})}{A(z^{-1})}u(k) + \varepsilon(k)$$

il modello prende il nome di *output error* (oe). Si veda la funzione matlab *oe*.

- Applicando un segnale di ingresso $u(k)$ del tipo *onda quadra* si ottiene la risposta di figura



- Dai dati acquisiti (con un periodo di campionamento $T = 0.1s$), applicando la procedura di stima ai minimi quadrati si ottiene

$$n = 1 \rightarrow G_i(z) = \frac{0.2706}{z-0.9864} \quad \left(\frac{1}{N} \sum_N \varepsilon^2 = 0.0018 \right)$$

$$n = 2 \rightarrow G_i(z) = \frac{0.03466z-0.003916}{z^2-1.898z+0.8995} \quad \left(\frac{1}{N} \sum_N \varepsilon^2 = 1.6294e-005 \right)$$

$$n = 3 \rightarrow G_i(z) = \frac{0.03211z^2-0.01314z-0.006504}{z^3-2.472z^2+1.987z-0.5146} \quad \left(\frac{1}{N} \sum_N \varepsilon^2 = 1.0511e-005 \right)$$

$$n = 4 \rightarrow G_i(z) = \frac{0.03193z^3-0.01373z^2-0.003541z-0.001757}{z^4-2.5z^3+2.098z^2-0.6496z+0.05238} \quad \left(\frac{1}{N} \sum_N \varepsilon^2 = 1.0472e-005 \right)$$

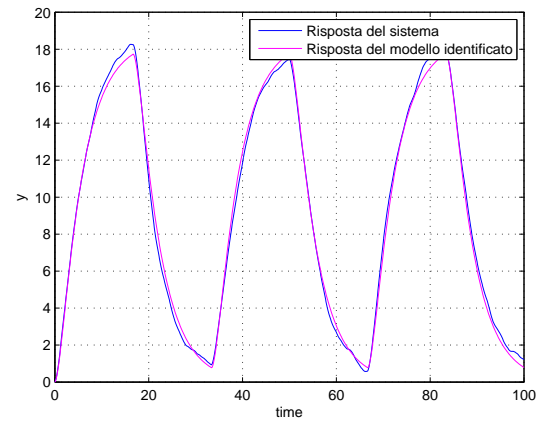
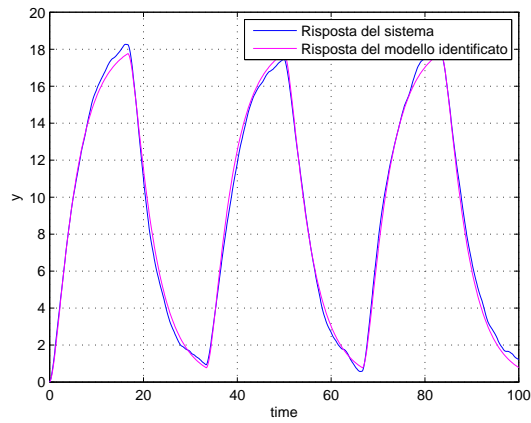
- Notare che il miglioramento, in termini di errore quadratico, indotto da un aumento dell'ordine per $n > 3$ risulta del tutto trascurabile.

- Nelle seguenti figure sono confrontati i risultati ottenuti per due diversi valori di n :

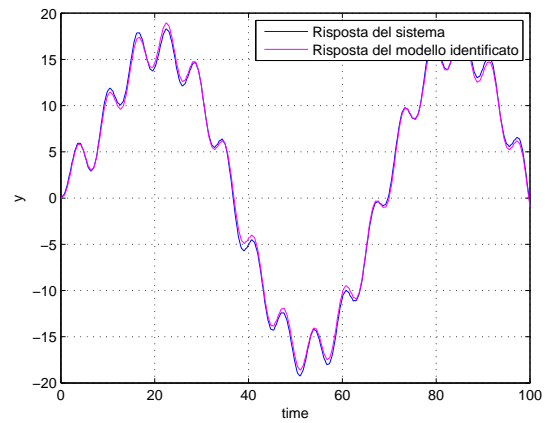
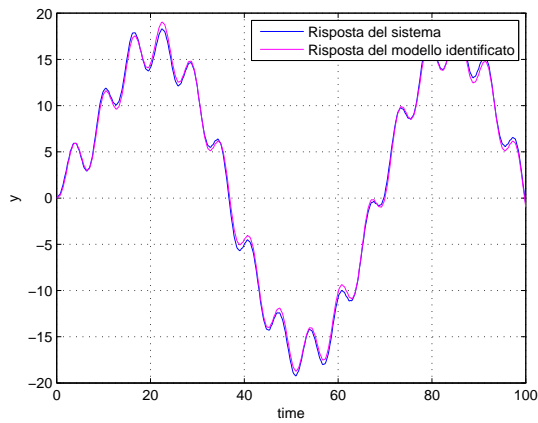
$$n = 2$$

$$n = 4$$

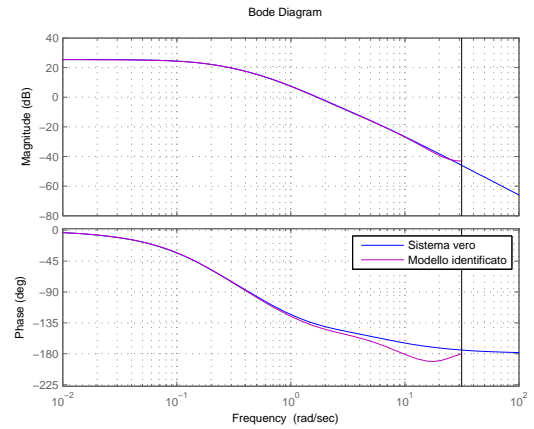
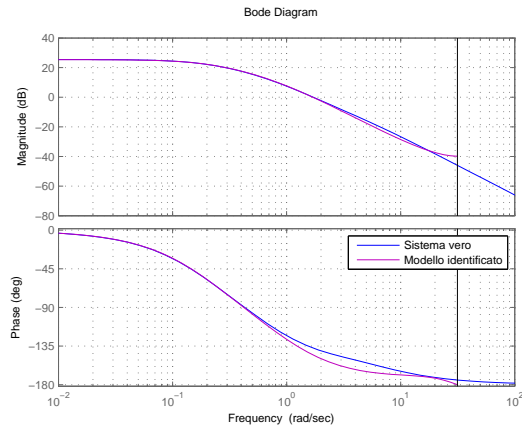
Risposta del sistema identificato a $u(k)$



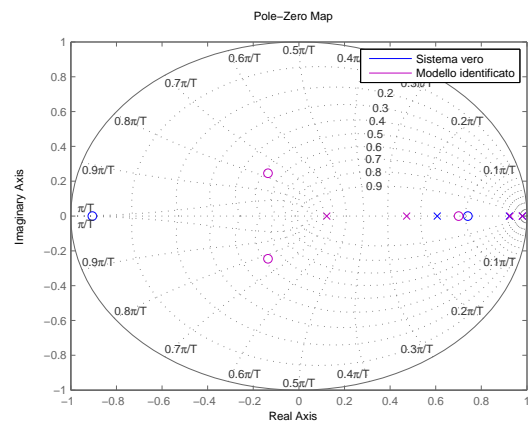
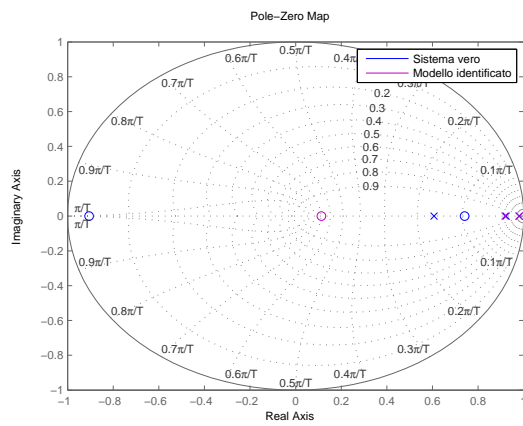
Risposta del sistema identificato a un segnale sinusoidale



Risposta in frequenza del sistema identificato



Mappa poli-zeri del sistema identificato



- Il *System Identification Toolbox* di Matlab mette a disposizione un libreria completa di funzioni per la stima dei parametri (e la successiva elaborazione) di sistemi dinamici. Tra queste la funzione `arx` che calcola la stima ai minimi quadrati dei coefficienti di un modello `arx`:

Uso: `m = arx(data,orders)`
 dove
`data` è l'insieme dei campioni dell'ingresso e dell'uscita definito tramite la funzione `iddata`
`orders = [na nb nk]`

Esempio Matlab

```
>> data = iddata(y,u,Ts)
```

```
Time domain data set with 1001 samples.  
Sampling interval: 0.1
```

```
Outputs      Unit (if specified)  
  y1
```

```
Inputs      Unit (if specified)  
  u1
```

```
>> m = arx(data,[np nz 1])
```

```
Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + e(t)
```

```
A(q) = 1 - 1.22 q-1 - 0.3137 q-2 + 0.5378 q-3
```

```
B(q) = 0.04807 q-1 + 0.008134 q-2 + 0.01457 q-3
```

```
Estimated using ARX from data set data
```

```
Loss function 0.000307986 and FPE 0.000311679
```

```
Sampling interval: 0.1
```

```
>> [A,B] = arxdata(m)
```

```
A =
```

```
    1.0000   -1.2204   -0.3137    0.5378
```

```
B =
```

```
    0    0.0481    0.0081    0.0146
```

```
>> G=tf(B,A,T)
```

```
Transfer function:
```

```
0.04804 z2 + 0.008068 z + 0.01406
```

```
-----  
z3 - 1.222 z2 - 0.3119 z + 0.5379
```

```
Sampling time: 0.1
```

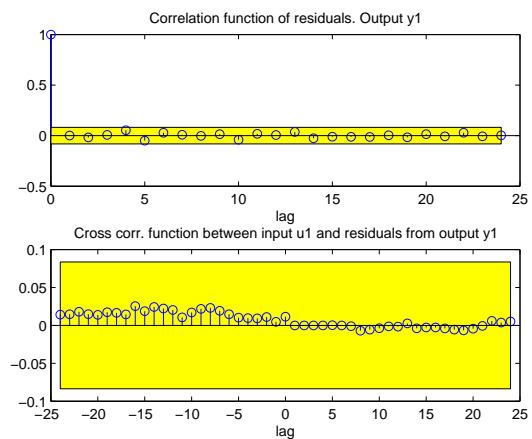
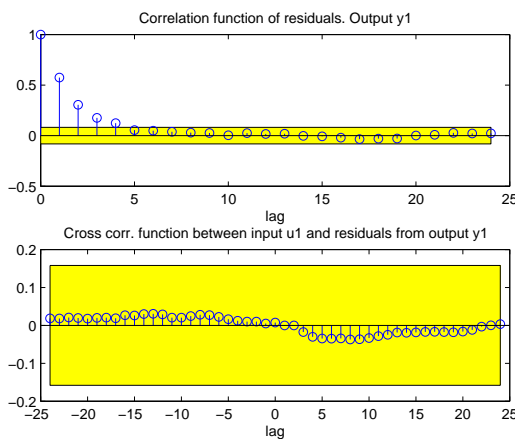
- In una interpretazione *probabilistica* del processo di indentificazione, se il rumore sovrapposto ai dati è effettivamente bianco (non correlato) e il modello è stato indentificato correttamente, allora i residui di modello dovranno risultare incorrelati.
- Il *System Identification Toolbox* di Matlab fornisce la funzione `resid` che calcola delle stime della funzione di correlazione del residuo, e plotta un intervallo di confidenza all'interno del quale *tutte le funzioni di correlazione eccetto la prima* (la varianza) si dovrebbero trovare. Se così risulta il modello indentificato può essere considerato soddisfacente.

Uso: `>> resid(m,data)`

$$n = 2$$

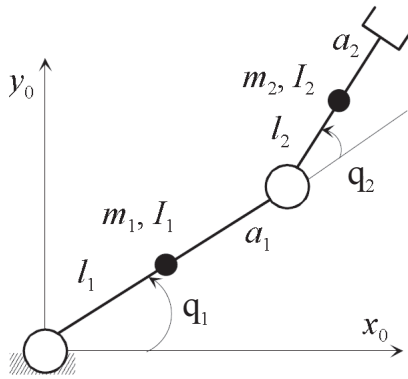
$$n = 4$$

Analisi dei residui



Identificazione dei parametri di un modello fisico: applicazione alla robotica

- Consideriamo il manipolatore robotico planare a 2 gradi di libertà, rappresentato in figura



Esso è caratterizzato da:

- masse: m_1 e m_2
- lunghezze: a_1 e a_2
- distanze dei baricentri dagli assi dei giunti: l_1 e l_2
- momenti di inerzia baricentrali intorno a z_0 : I_1 e I_2

Uscita: coordinate generalizzate $\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$

Ingresso: coppie applicate ai giunti $\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$

- In assenza di attriti e forze esterne sull'end-effector, le equazioni del moto risultano

$$(m_1 l_1^2 + I_1 + m_2(a_1^2 + l_2^2 + 2a_1 l_2 c_2) + I_2) \ddot{q}_1 + (m_2(l_2^2 + a_1 l_2 c_2) + I_2) \ddot{q}_2 - 2m_2 a_1 l_2 s_2 \dot{q}_1 \dot{q}_2 - m_2 a_1 l_2 s_2 \dot{q}_2^2 + (m_1 l_1 + m_2 a_1) g c_1 + m_2 g l_2 c_{12} = \tau_1$$

$$(m_2(l_2^2 + a_1 l_2 c_2) + I_2) \ddot{q}_1 + (m_2 l_2^2 + I_2) \ddot{q}_2 + m_2 a_1 l_2 s_2 \dot{q}_1^2 + m_2 l_2 g c_{12} = \tau_2$$

dove

$$c_i = \cos(q_i)$$

$$s_i = \sin(q_i)$$

$$c_{ij} = \cos(q_i + q_j)$$

- In termini vettoriali le equazioni possono essere riscritte come

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (2)$$

con le matrici

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} m_1 l_1^2 + I_1 + m_2(a_1^2 + l_2^2 + 2a_1 l_2 c_2) + I_2 & m_2(l_2^2 + a_1 l_2 c_2) + I_2 \\ m_2(l_2^2 + a_1 l_2 c_2) + I_2 & m_2 l_2^2 + I_2 \end{bmatrix}$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -m_2 a_1 l_2 s_2 \dot{q}_2 & -m_2 a_1 l_2 s_2 (\dot{q}_1 + \dot{q}_2) \\ m_2 a_1 l_2 s_2 \dot{q}_1 & 0 \end{bmatrix}$$

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} (m_1 l_1 + m_2 a_1) g c_1 + m_2 g l_2 c_{12} \\ m_2 l_2 g c_{12} \end{bmatrix}$$

e dove i vettori \mathbf{q} e $\boldsymbol{\tau}$ valgono

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

- Come si nota la (2) risulta un'equazione nonlineare che tuttavia può essere riscritta come un'equazione lineare nei parametri, ovvero

$$\tau = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\pi}$$

- Per ispezione diretta è possibile ricavare i parametri dinamici rispetto ai quali il modello è lineare. Risulta:

$$\boldsymbol{\pi} = \begin{bmatrix} m_1 l_1 \\ I_1 + m_1 l_1^2 \\ m_2 \\ m_2 l_2 \\ I_2 + m_2 l_2^2 \end{bmatrix}$$

e

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} & y_{15} \\ 0 & 0 & 0 & y_{24} & y_{25} \end{bmatrix}$$

con

$$y_{11} = g c_1$$

$$y_{12} = \ddot{q}_1$$

$$y_{13} = a_1^2 \ddot{q}_1 + a_1 g c_1$$

$$y_{14} = 2a_1 c_2 \ddot{q}_1 + a_1 c_2 \ddot{q}_2 - 2a_1 s_2 \dot{q}_1 \dot{q}_2 - a_1 s_2 \dot{q}_2^2 + g c_{12}$$

$$y_{15} = \ddot{q}_1 + \ddot{q}_2$$

$$y_{24} = a_1 c_2 \ddot{q}_1 + a_1 s_2 \dot{q}_1^2 + g c_{12}$$

$$y_{25} = \ddot{q}_1 + \ddot{q}_2$$

- Si noti che i coefficienti di \mathbf{Y} dipendono da q_1 , q_2 , dalle loro derivate prime e seconde, da g e da a_1 (quest'ultimo parametro può essere facilmente misurato trattandosi della lunghezza di un link).
- Il sistema

$$\tau = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\pi}$$

risulta un modello di regressione lineare, i cui parametri incogniti $\boldsymbol{\pi}$ si possono stimare con la tecnica vista precedentemente.

- Si dovranno eseguire opportune traiettorie del manipolatore robotico, misurando le posizioni dei giunti \mathbf{q} , misurando o ricavando per derivazione le velocità $\dot{\mathbf{q}}$ e misurando o ricavando per derivazione (filtrata) le accelerazioni $\ddot{\mathbf{q}}$. E' necessario poi misurare le coppie $\boldsymbol{\tau}$, direttamente (con opportuni sensori) o indirettamente dalle misure di corrente nei motori.
- Una volta note le misure di tutte le variabili negli istanti di tempo t_1, \dots, t_N si può scrivere

$$\bar{\boldsymbol{\tau}} = \begin{bmatrix} \boldsymbol{\tau}(t_1) \\ \vdots \\ \boldsymbol{\tau}(t_N) \end{bmatrix} = \begin{bmatrix} \mathbf{Y}(t_1) \\ \vdots \\ \mathbf{Y}(t_N) \end{bmatrix} \boldsymbol{\pi} = \bar{\mathbf{Y}} \boldsymbol{\pi}$$

- risolvendo con la tecnica ai minimi quadrati si ottiene

$$\boldsymbol{\pi} = (\bar{\mathbf{Y}}^T \bar{\mathbf{Y}})^{-1} \bar{\mathbf{Y}}^T \bar{\boldsymbol{\tau}}$$

- Chiaramente sono identificabili solo gli elementi di $\boldsymbol{\pi}$ la cui corrispondente colonna di $\bar{\mathbf{Y}}$ non è nulla (che quindi danno un contributo alla dinamica del robot). Inoltre, alcuni parametri sono identificabili solo in combinazione lineare con altri
- Le traiettorie da utilizzare devono essere sufficientemente ricche (buon condizionamento di $\bar{\mathbf{Y}}^T \bar{\mathbf{Y}}$) ma non tali da eccitare dinamiche non modellate (solitamente vengono impiegate traiettorie di tipo polinomiale).

Esempio Matlab (StimaParametriRobot.m)

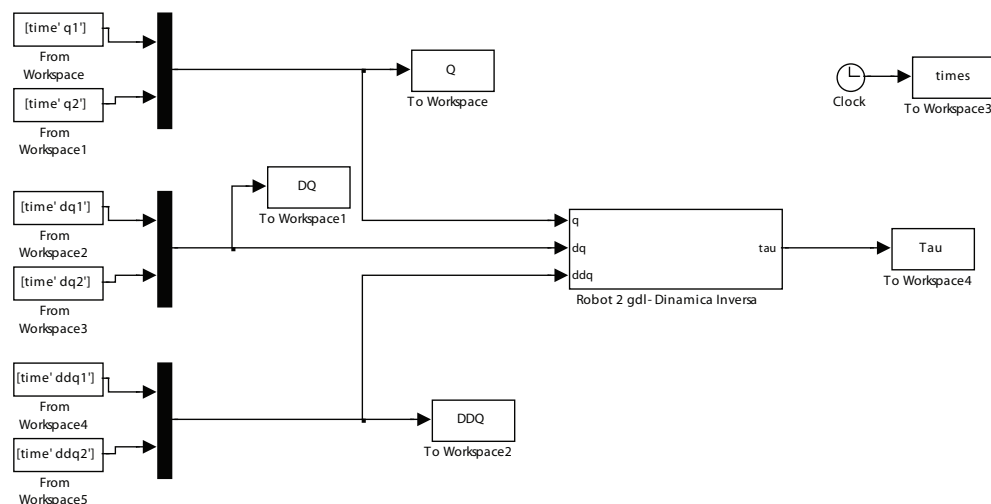
- Si vuole applicare il procedimento di stima dei parametri a un robot caratterizzato dai seguenti parametri fisici:
 - Lunghezze dei link: $a_1 = 1\text{m}$, $a_2 = 1\text{m}$
 - Distanze dei baricentri dagli assi dei giunti: $l_1 = 0.5\text{m}$, $l_2 = 0.5\text{m}$
 - Masse: $m_1 = 50\text{kg}$, $m_2 = 50\text{kg}$
 - Momenti di inerzia baricentrali intorno ad un asse parallelo a z_0 : $I_1 = 10\text{ Kg m}^2$, $I_2 = 10\text{Kg m}^2$
- Si applicano ai giunti traiettorie di tipo polinomiale caratterizzate dalle equazioni

$$q(\tau) = h(10\tau^3 - 15\tau^4 + 6\tau^5) + q_0, \quad \tau = \frac{t - t_0}{T}$$

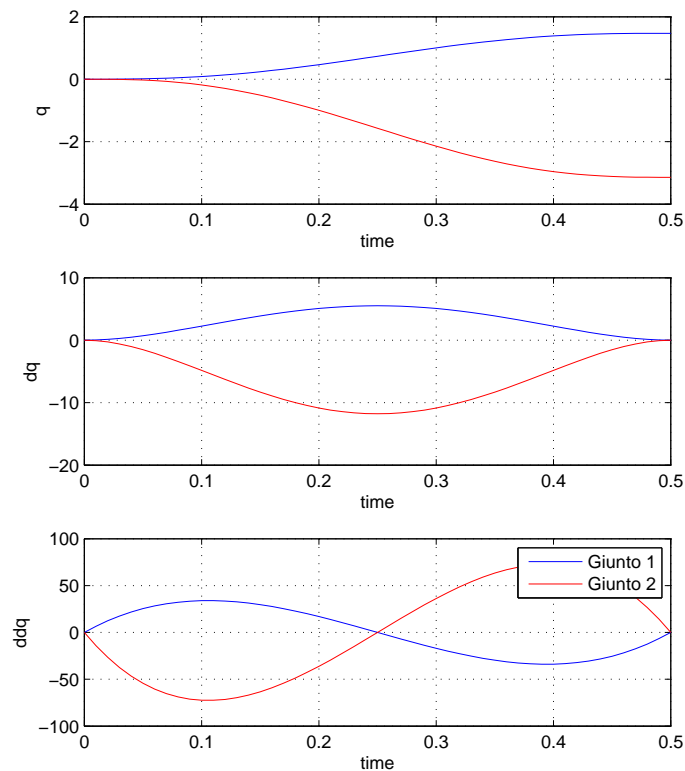
$$\dot{q}(\tau) = \frac{h}{T}(30\tau^2 - 60\tau^3 + 30\tau^4)$$

$$\ddot{q}(\tau) = \frac{h}{T^2}(60\tau - 180\tau^2 + 120\tau^3)$$

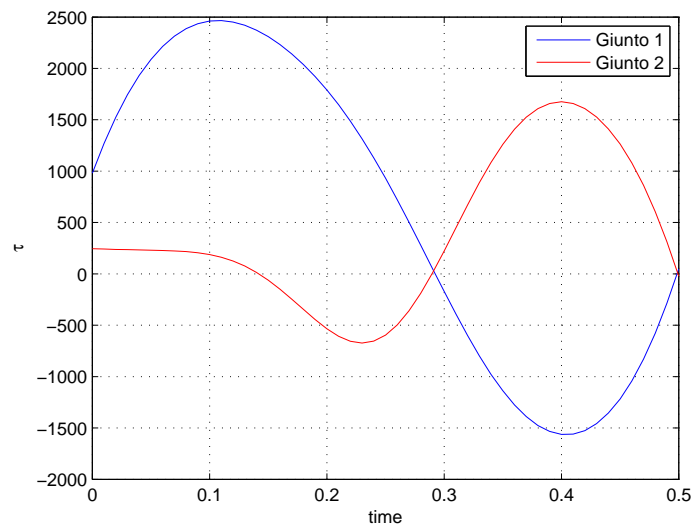
dove $T = t_1 - t_0$ è la durata della traiettoria, e $h = q_1 - q_0$ è lo spostamento richiesto, e si “registrano” le coppie ai giunti.



Traiettorie di posizione



Coppie ai giunti



- In seguito al processo di identificazione si ricava il seguente vettore dei parametri

$$\hat{\boldsymbol{\pi}} = [25.8333, 23.3333, 49.1667, 25.0000, 22.5000]^T$$

a fronte del vettore "vero"

$$\boldsymbol{\pi} = [25.0000, 22.5000, 50.0000, 25.0000, 22.5000]^T$$