

Progetto di laboratorio: Primo lavoro di gruppo

Davide Salerno, Edoardo Storti, Nicola Taddei & Luca Varotto
Dipartimento di Ingegneria dell'Automazione
Politecnico di Milano

I. DEFINIZIONE DEL PROBLEMA

Lo scopo del laboratorio è ricostruire un modello lineare di un sistema dinamico (motore elettrico con inerzia controllato in DC) attraverso un'analisi black-box a partire da un set di input-output. Il risultato finale, fornito sotto forma di funzione di trasferimento (che lega la tensione in ingresso alla velocità di rotazione in uscita), verrà testato confrontandolo con il sistema reale attraverso il calcolo dello scarto quadratico medio fra la risposta ad un dato input (a noi sconosciuto) determinata dal nostro modello e quella del sistema stesso.

II. PROCEDIMENTO ADOTTATO

Il modello che vogliamo ottenere, fissato ad n l'ordine, può essere scritto nella seguente formulazione in forma di stato

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \\ \mathbf{y}(t) = \mathbf{c}\mathbf{x}(t) + du(t) \end{cases} \quad (1)$$

dove \mathbf{A} (matrice $n \times n$), \mathbf{b} (vettore $n \times 1$), \mathbf{c} (vettore $1 \times n$) e d (scalare) sono parametri incogniti.

Tali valori sono legati alla funzione di trasferimento dalla relazione

$$G(s) = \mathbf{c}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + d \quad (2)$$

che a sua volta è parametrizzata da due vettori che definiscono i coefficienti del polinomio di s al numeratore e quelli al denominatore. Possiamo raggruppare tali parametri nel vettore $\boldsymbol{\theta}$.

A partire da tale modello è possibile calcolare numericamente la risposta del sistema ad un ingresso generico u_n (successione corrispondente alla funzione di input valutata negli istanti di campionamento t_1, \dots, t_n). Si ottiene così una successione $\tilde{y}_{\boldsymbol{\theta},n}$ che può essere confrontata con quella dei valori dell'uscita del sistema misurati sperimentalmente y_n .

Chiamando *sim* una generica procedura che permette di calcolare numericamente la traiettoria del sistema con parametri $\boldsymbol{\theta}$ sottoposto ad un input $u(t)$, di cui sono noti i valori agli istanti t_n , possiamo riscrivere quanto sopra detto nel seguente modo

$$\tilde{y}_{\boldsymbol{\theta},n} = \text{sim}(\boldsymbol{\theta}, u_n, t_n) \quad (3)$$

Una possibile metrica di distanza fra l'uscita calcolata a partire dal nostro modello e quella reale è lo scarto quadratico medio

$$MSE(y_n, \tilde{y}_{\boldsymbol{\theta},n}) = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_{\boldsymbol{\theta},i})^2 \quad (4)$$

Si può quindi definire una funzione del vettore $\boldsymbol{\theta}$ che esprime l'errore fra il nostro modello e il sistema reale



Figura 1. Il sistema reale

$$errore(\theta) = MSE(y_n, sim(\theta, u_n, t_n)) \quad (5)$$

A questo punto possiamo dare una formulazione matematica più rigorosa al compito che ci proponiamo di svolgere

$$\min errore(\theta) \quad (6)$$

Esistono vari algoritmi di ottimizzazione che permettono di dare una soluzione a tale problema. Un esempio, forse il più semplice, è l'algoritmo di discesa del gradiente, che prevede di calcolare, ad ogni iterazione, il gradiente della funzione di errore rispetto ai parametri, per poi modificare i valori di questi ultimi di uno scarto

$$\delta\theta = -k * \nabla_{\theta} errore \quad (7)$$

dove k è un valore positivo (misura della lunghezza del "passo" che si compie nello spazio dei parametri) che può essere mantenuto costante o variare al passare delle iterazioni a seconda delle implementazioni.

Fortunatamente in *Matlab* esiste già la funzione *tfest* che implementa la procedura sopra esposta (mettendo a disposizione algoritmi più efficienti di quelli da noi proposti).

A. Funzionamento di *tfest*

$$SYS = tfest(DATA, NP, OPT)$$

tfest è una funzione di Matlab che si occupa di stimare una funzione di trasferimento a tempo continuo usando dati campionati o nel dominio del tempo o nel dominio delle frequenze.

tfest richiede in input un oggetto di tipo *DATA* e un numero intero positivo *NP* (che indica i numeri di poli) e ne ricava una funzione di trasferimento. *SYS* è un oggetto *IDTF* (transfer function model with identifiable parameters) che incapsula la funzione di trasferimento identificata. *OPT* specifica le opzioni aggiuntive che vogliamo andare ad utilizzare per migliorare la funzione ottenibile.

Algoritmo

Inizializzazione dei parametri:

L'algoritmo inizializza i parametri stimabili utilizzando il metodo specificato dall'opzione di stima *InitializeMethod*. Il metodo predefinito è l'Instrument Variable.

Questo metodo ha un prefiltro (che è il denominatore del "modello corrente" inizializzato con SVF, State-Variable Filters). Questo prefiltro viene iterato fino a quando la modifica del modello non è inferiore all'errore minimo desiderato rimanendo sotto un tetto massimo di tentativi possibili. *MaxIterations* e *Tolerance* sono opzioni che è possibile specificare utilizzando la struttura *InitializeOptions*.

Aggiornamento dei parametri:

I parametri inizializzati vengono aggiornati utilizzando un metodo di ricerca non lineare dei minimi quadrati, specificato dall'opzione di stima *SearchMethod*. L'obiettivo del metodo di ricerca è ridurre al minimo la norma ponderata dell'errore di previsione.

III. CRITICITÀ INCONTRATE E PROBLEM SOLVING

A. Errori numerici

Osservando i grafici relativi alle rampe ci siamo resi conto della presenza di una forte discontinuità nell'ultimo campione di velocità. Abbiamo dedotto che essa sia dovuta al modo in cui quest'ultima è stata ricavata: a partire da misurazioni della posizione e conoscendo gli intervalli di campionamento è stata eseguita una divisione fra gli incrementi (decrementi) delle due variabili. Per poter valutare la velocità nell'ultimo punto (necessitando di due valori di posizione) si è scelto di inventare l'ultimo punto oppure di scegliere una posizione relativa ad un ingresso non registrato (presumibilmente 0). Abbiamo optato quindi per eliminare l'ultimo campione da ciascun set di dati.

B. Non linearità

Un problema significativo che ha fin da subito attirato la nostra attenzione è lo stallo subito dal motore in corrispondenza di valori molto bassi della tensione in ingresso, particolarmente evidente nei grafici di *ramp1.mat* e *ramp3.mat*

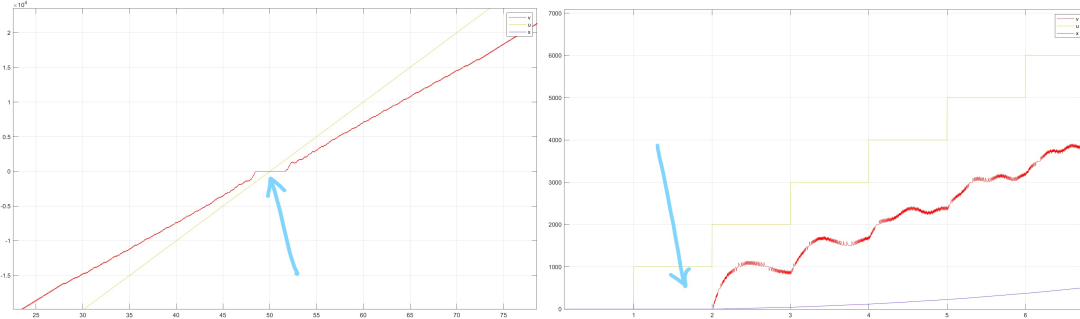


Figura 2. *ramp1*, *ramp2* (le frecce azzurre evidenziano il fenomeno in questione)

Questo ci ha spinti a pensare a cosa potesse causare tale fenomeno, cercando indizi meno evidenti anche nei grafici delle risposte alle sinusoidi. Il risultato di tale indagine ci ha portati a sospettare l'attrito (statico e dinamico) come causa, ponendo degli interrogativi sulla massima accuratezza raggiungibile dal nostro modello. L'attrito infatti introduce un certo grado di non linearità nel sistema, che per forza di cose non potrà essere catturato dal nostro modello del sistema, basato sull'assunzione di linearità. Inoltre abbiamo notato che l'effetto dell'attrito è più evidente nei casi in cui l'input sinusoidale abbia ampiezza ridotta e derivata massima (e quindi pendenza nell'origine) più bassa: ciò è coerente con l'ipotesi di attrito dinamico costante opposto al moto, e nel caso in cui "l'albero motore" abbia una velocità di rotazione maggiore (pendenza nell'origine elevata) e una "coppia" elevata (ampiezza più alta), l'effetto dell'attrito risulta ridotto.

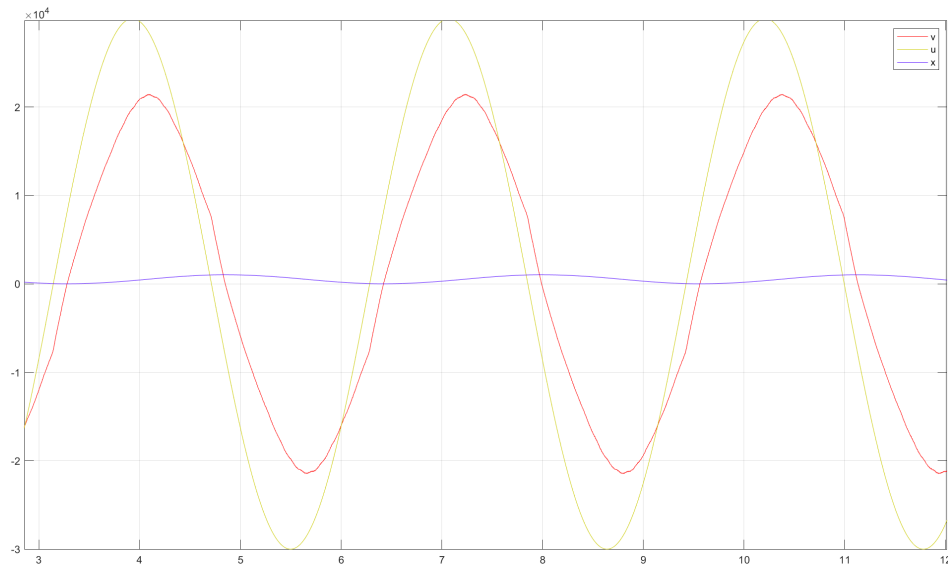


Figura 3. *sine3*, si può notare come la fase di accelerazione in valore assoluto sia più lenta della decelerazione, a causa del contributo contrario al moto dell'attrito

C. Ordine

Un altro problema che ci siamo trovati a dover affrontare è stato quello della determinazione dell'ordine del modello. Un'analisi qualitativa dei grafici suggerisce la presenza di dinamiche lente, compatibili con un sistema di ordine 1 o 2 e dinamiche oscillatorie, più veloci, imputabili alla presenza di poli complessi coniugati o all'azione del rumore (che però non può essere prevista dal nostro modello). Inoltre osservando il grafico della risposta a `square.mat` e `scalini.mat`, applicando il teorema del valore iniziale abbiamo capito che il grado relativo del sistema doveva essere uno, dato che $y(0^+) = 0$ e $\dot{y}(0^+) \neq 0$.

Procedendo per tentativi e giocando con i parametri della funzione (tolleranza e numero di iterazioni massime) ci siamo resi conto che i risultati migliori venivano ottenuti usando un modello con due poli. Questo può essere spiegato dal fatto che la dinamica del sistema si presta ad essere approssimata da una del secondo ordine e anche dal fatto che l'introduzione di nuovi poli rende più complessa la forma della funzione da ottimizzare, aumentando le probabilità di rimanere "intrappolati" in minimi locali con scarse prestazioni.

D. Posizione vs velocità

Avendo a disposizione i dati relativi sia alla posizione del motore che alla sua velocità abbiamo tentato di sfruttare entrambi: nel caso della velocità il procedimento non necessita ulteriori spiegazioni, mentre nel caso della posizione ci siamo limitati ad usare `tfest` sul set di dati relativo alla posizione e poi moltiplicare la f.d.t. per s per ottenere la funzione relativa alla derivata. In quest'ultima situazione ci siamo accorti che la risposta agli scalini prevista conteneva delle forti discontinuità in corrispondenza di quelle dell'input, dando luogo ad un'uscita visibilmente innaturale. Abbiamo quindi deciso di abbandonare questa strada e limitarci all'analisi della velocità.

E. Bontà dell'input

Nel determinare la funzione di trasferimento, abbiamo osservato che in base agli input la funzione `tfest` dava risultati diversi.

Ad esempio, utilizzando solo i dati relativi agli ingressi `ramp1.mat`, `ramp2.mat` e `ramp3.mat` ha generato una determinata funzione di trasferimento, che però non dava risultati soddisfacenti con gli altri input.

Utilizzando anche le funzioni sinusoidali c'è stato un miglioramento significativo nella funzione di trasferimento, e anche considerando soltanto gli input sinusoidali la funzione di trasferimento ottenuta riusciva ad approssimare bene le uscite degli input "a rampa".

IV. FUNZIONI DI TRASFERIMENTO OTTENUTE AL VARIARE DEGLI INPUT

Per comodità, si è scelto di raggruppare gli input nel seguente modo

- `rampe = {ramp1.mat, ramp2.mat, ramp3.mat}`
- `seni f=cost = {sine1.mat, sine2.mat, sine3.mat, sine4.mat, sine5.mat, sine6.mat}`
- `seni f≠cost = {seni_largeamplitude_nooffset.mat, seni_nooffset.mat, seni_offset2.mat}`
- `onde quadre = {scalini.mat, square.mat}`

A. rampe

Utilizzando gli input raggruppati in `rampe`, la funzione di trasferimento che si ottiene è:

$$G_1(s) = \frac{1.397 \times 10^4 s + 3685}{s^2 + 2.207s + 0.4934} \quad (\text{A.8})$$

Inoltre anche utilizzando gli input presenti in `seni f≠cost` e in `rampe` si ottiene sempre $G_1(s)$, che è anche quella ottenuta utilizzando tutti gli input a nostra disposizione.

B. $\text{seni } f = \text{cost}$

Utilizzando gli input raggruppati in $\text{seni } f = \text{cost}$, la funzione di trasferimento che si ottiene è:

$$G_2(s) = \frac{3.199 \times 10^3 s + 3.865 \times 10^6}{s^2 + 70.58s + 567.8} \quad (\text{A.9})$$

C. $\text{seni } f \neq \text{cost}$

Utilizzando gli input raggruppati in $\text{seni } f \neq \text{cost}$, la funzione di trasferimento che si ottiene è:

$$G_3(s) = \frac{4.13 \times 10^4 s + 5.386 \times 10^3}{s^2 + 5.456s + 0.7409} \quad (\text{A.10})$$

Anche utilizzando tutti gli input sinusoidali si ottiene la stessa funzione di trasferimento, come anche se a questi input si aggiungono quelli di rampe. Anche usando $\text{seni } f \neq \text{cost}$ e rampe si ottiene $G_3(s)$.

D. onde quadre

Utilizzando gli input raggruppati in onde quadre, la funzione di trasferimento che si ottiene è:

$$G_4(s) = \frac{-1.684 \times 10^4 s + 8.145 \times 10^6}{s^2 + 129.9s + 1074} \quad (\text{A.11})$$

Anche utilizzando gli input presenti in onde quadre unite agli input di un qualunque altro raggruppamento la funzione di trasferimento risultante è sempre $G_4(s)$.

V. OSSERVAZIONI

A. Comportamento t_{fest}

Si può osservare che al variare degli input utilizzati ci sia, in generale, una conseguente variazione della funzione di trasferimento ottenuta tramite t_{fest} .

Analizzando i risultati possiamo dire che:

- Alcuni input influenzano maggiormente la FDT rispetto ad altri
- Ci sono input che migliorano notevolmente la FDT
- Ci sono input che peggiorano notevolmente la FDT

Ad esempio gli input in onde quadre influenzano notevolmente e positivamente la FDT. Infatti calcolare la FDT solo con questi input o aggiungendone anche altri non porta a nessuna variazione, tranne nel caso in cui si usino tutti gli input.

Il motivo per cui utilizzando tutti gli input la FDT vari è che le non linearità legate ai singoli input vanno a sommarsi creando delle problematiche non indifferenti all'algoritmo di t_{fest} . Per verificare ciò abbiamo fatto un tentativo usando tutti gli input tranne `ramp3`. Abbiamo scelto di togliere questo input poiché ha un comportamento fortemente non lineare quando il voltaggio in input è vicino al valore nullo. Il risultato ottenuto ci riporta alla stessa FDT ottenuta con solo gli input in onde quadre.

Un'altra osservazione riguarda le `OPT` di t_{fest} . Si osserva che aumentando il numero di iterazioni di t_{fest} non si ha un aumento significativo della bontà della funzione di trasferimento.

B. Verifica consistenza funzione di trasferimento

Prendiamo come riferimento la funzione di trasferimento $G(s) = \frac{-1.684 \times 10^4 s + 8.145 \times 10^6}{s^2 + 129.9s + 1074}$ e verifichiamo che il risultato ottenuto sia coerente con i dati sperimentali.

Possiamo ricavare il guadagno μ direttamente dalla funzione di trasferimento. Infatti, poiché il tipo di $G(s)$ è zero, il guadagno generalizzato è $\mu = G(0) = \frac{8.145 \times 10^6}{1074} \approx 7584$ ed è uguale al guadagno statico ($\mu = \frac{\bar{y}}{\bar{u}}$). Possiamo provare a ricavare \bar{y} e \bar{u} per via grafica. Ci serve un ingresso che rimane costante per un certo tempo tale per cui l'uscita termina il suo transitorio e riesce a stabilirsi su un valore fisso. Tra i dati sperimentali quelli che posseggono queste caratteristiche sono quelli contenuti in onde quadre.

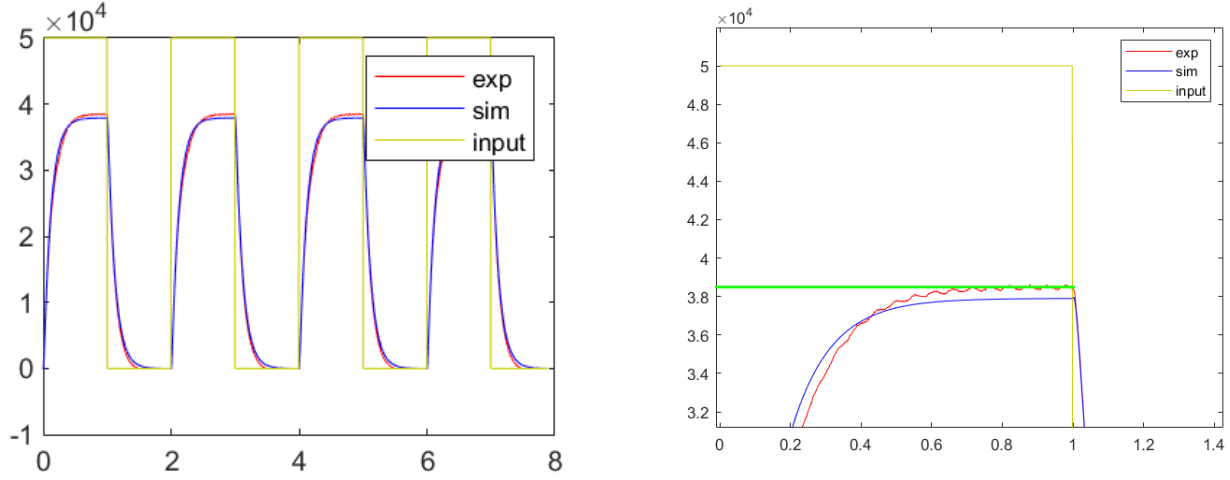


Figura 4. Comportamento square.mat

Analizziamo l'onda quadra da $t = 0$ a $t = 1$. L'ingresso vale $\bar{u} = 5$, l'uscita \bar{y} si stabilizza a circa $\bar{y} = 38500$ e il guadagno statico vale $\mu = \frac{\bar{y}}{\bar{u}} \approx \frac{38500}{5} = 7700$.

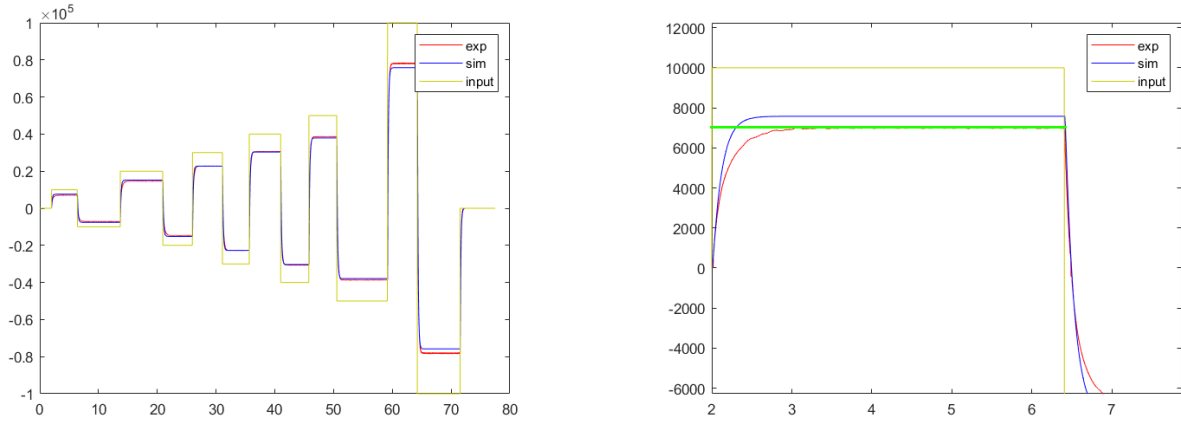


Figura 5. comportamento gradini.mat

Valutando il primo gradino di gradini.mat l'ingresso vale $\bar{u} = 1$, l'uscita \bar{y} si stabilizza a circa $\bar{y} = 7000$ e di conseguenza il guadagno è $\mu = \frac{\bar{y}}{\bar{u}} \approx \frac{7000}{1} = 7000$.

Riassumendo abbiamo trovato un valore $\mu \approx 7700$ per `square.mat` e $\mu \approx 7000$. Se calcolassimo il guadagno anche per tutti gli altri gradini otterremmo un valore di μ che si aggira intorno a questi valori quindi possiamo dire che il valore del guadagno di $G(s)$ è coerente con i dati sperimentali analizzati ($\mu_G \approx 7584$).

Sempre dai grafici in onde quadre possiamo ricavare il tempo di transitorio e di conseguenza il polo reale più vicino all'asse immaginario. Infatti il tempo di assestamento è $T_A = 5\tau_{max}$ e il polo vale $p_1 = \frac{-1}{\tau_{max}}$ (metto il segno meno poiché il polo deve essere minore di zero affinché il sistema sia asintoticamente stabile). I poli di $G(s)$ sono $p_{1G(s)} \approx -8.88$ e $p_{2G(s)} \approx -121$.

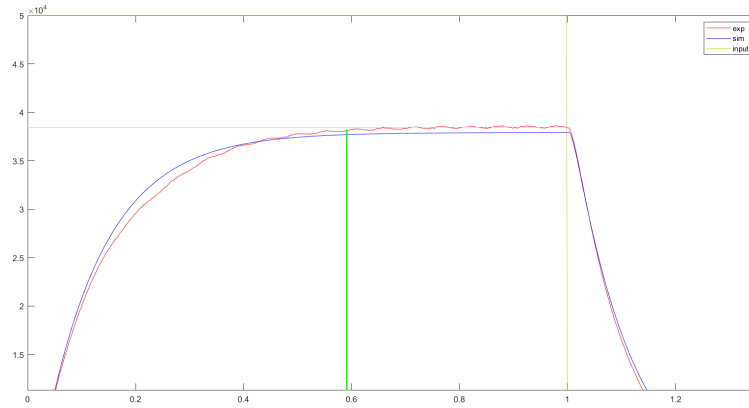


Figura 6. La linea verde indica il T_A

Graficamente ricaviamo che $T_A = \tau_1 \times 5 \approx 0.59$. Allora $p_1 \approx \frac{-1}{\tau_1} = \frac{-1}{0.59} \times 5 \approx -8.47$. Il polo $p_{1G(s)}$, che tra i due poli di $G(s)$ è quello più vicino all'asse immaginario, non discosta particolarmente dal polo p_1 ricavato per via analitica. Quindi possiamo concludere che il valore del polo $p_{1G(s)}$ è coerente con i dati sperimentali.

Per valutare la bontà dello zero e del secondo polo sfruttiamo il teorema della risposta in frequenza. Esso dice che se il sistema è asintoticamente stabile allora $y(t) \sim_{t \rightarrow \infty} \hat{y}(t)$ dove, dato un ingresso $u(t) = U \times \sin(\omega t)$, $\hat{y}(t) = U \times |G(j\omega)| \times \sin(+\angle G(j\omega))$.

Analizziamo `sine1.mat` e `sine2.mat`. Partiamo calcolando il valore di ω , infatti esso sarà $\omega = \frac{2\pi}{T}$.

Per via grafica vediamo che $T_1 \approx 12.55$ e $T_2 \approx 6.28$ e quindi i valori di $\omega = \frac{2\pi}{T}$ saranno $\omega_1 \approx 0.50$ e $\omega_2 \approx 1.00$. Il modulo $|G(j\omega)|$ nei due casi vale $|G(j\omega_1)| \approx 7568.6$ e $|G(j\omega_2)| \approx 7532.8$. Per via grafica possiamo anche valutare l'ampiezza U delle due sinusoidi che sono $U_1 = 10$ e $U_2 = 5$. Ci aspettiamo che $U_i \times |G(j\omega_i)| = Amp_i$ (ampiezza della i -esima sinusoide d'uscita). Le ampiezze delle uscite $y(t)_i$ le possiamo valutare tramite i grafici e valgono $Amp_1 \approx 78000$ e $Amp_2 \approx 38000$.

$U_1 \times |G(j\omega_1)| \approx 10 \times 7568.6 = 75686$ e $U_2 \times |G(j\omega_2)| \approx 5 \times 7532.8 = 37664$ che sono circa le ampiezze valutate per via grafica.

Passiamo a valutare le fasi $\angle G(j\omega)$. Chiameremo ϕ le fasi valutate graficamente.

Valgono $\phi_i = (t_{picco,u,i} - t_{picco,y,i}) \times \omega_i$ e dunque avremo:

$$\phi_1 = (3.15 - 3.25) \times \omega_1 = -0.05 \text{ rad}$$

$$\phi_2 = (1.57 - 1.7) \times \omega_2 = -0.13 \text{ rad}$$

Calcoliamo ora le fasi $\angle G(j\omega) = \sum \arctan \frac{\text{Im}\{z_i\}}{\text{Re}\{z_i\}} - \sum \arctan \frac{\text{Im}\{p_i\}}{\text{Re}\{p_i\}}$. Avremo quindi:

$$\angle G(j\omega)_1 = -0.0614 \text{ rad}$$

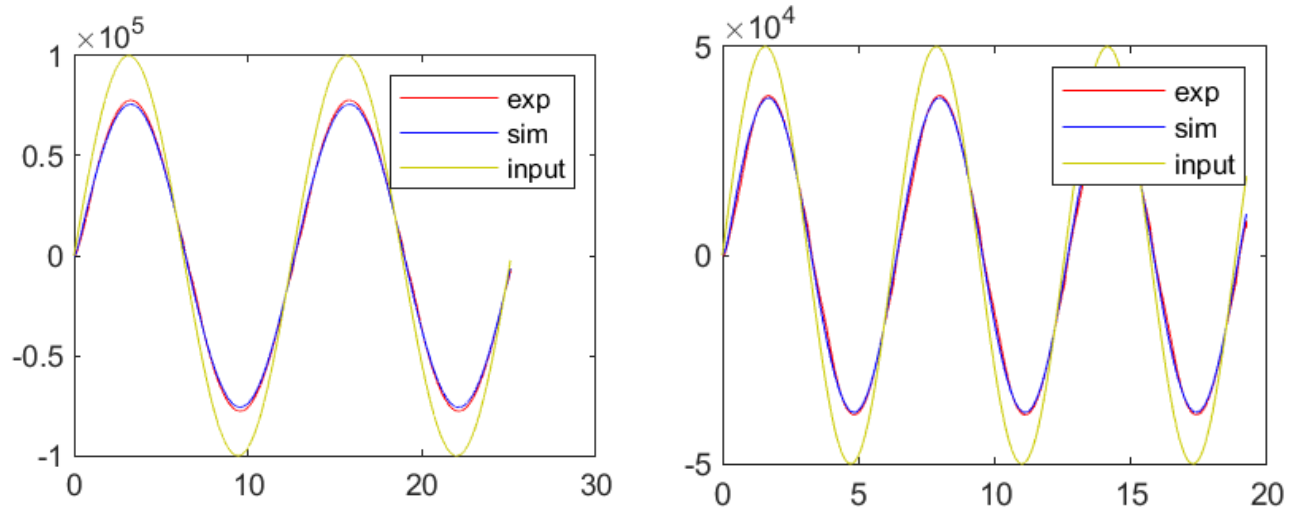


Figura 7. sine1.mat e sine2.mat

$$\angle G(j\omega)_2 = -0.1225 \text{ rad}$$

Anche in questo caso i risultati ricavati per via analitica non si discostano particolarmente da quelli ottenuti tramite la funzione di trasferimento $G(s)$.

VI. CONSIDERAZIONI FINALI

La miglior funzione di trasferimento individuata, valutata sia graficamente che attraverso il valore di MSE è:

$$G(s) = \frac{-1.684 \times 10^4 s + 8.145 \times 10^6}{s^2 + 129.9s + 1074}$$

APPENDICE A GRAFICI

Per ogni funzione di trasferimento, ricavata utilizzando diversi input, mostriamo alcuni grafici che rappresentano quanto si discosti la simulazione ottenuta (*sim*) rispetto ai dati sperimentali utilizzati per stimare la funzione di trasferimento (*exp*).

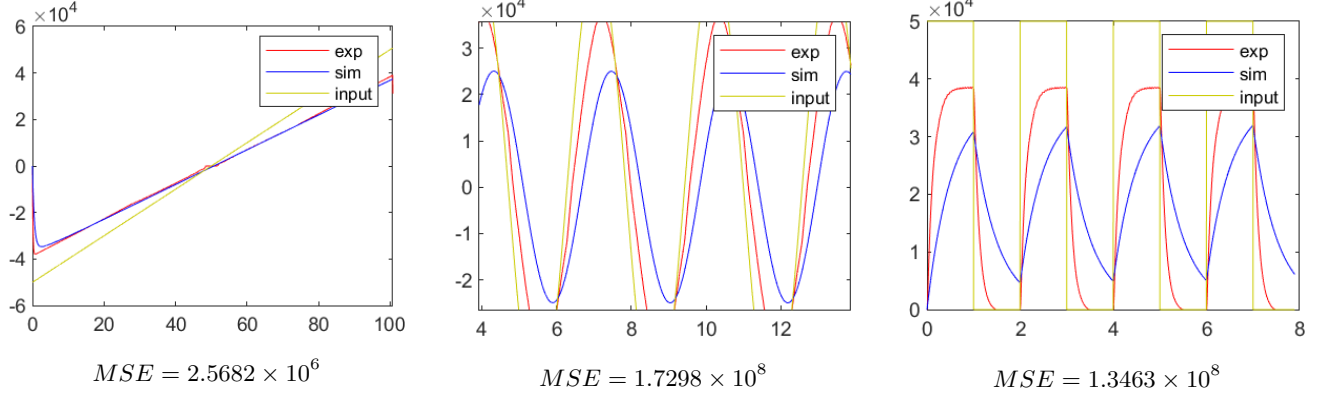


Figura 8. Caso A: Rampa 1, Seno 4, Square

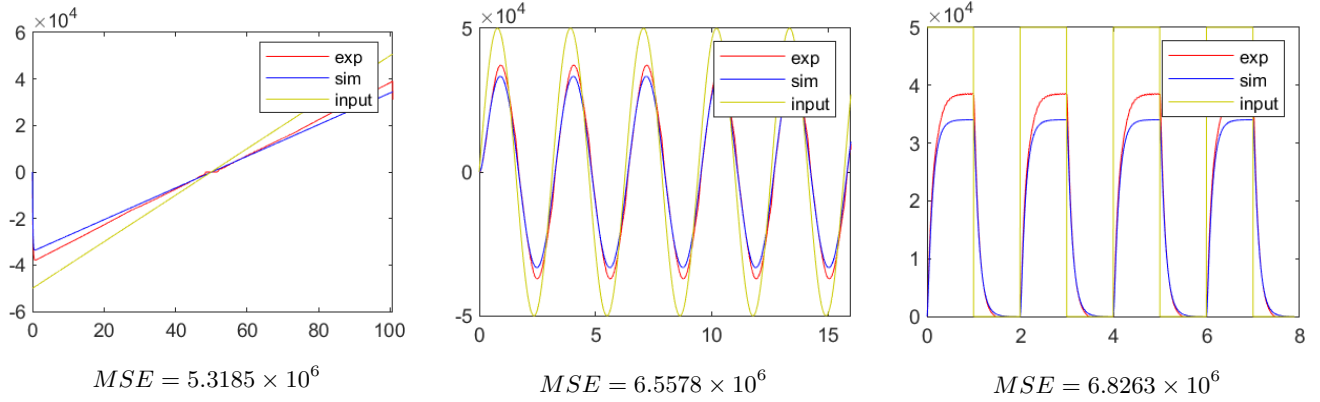


Figura 9. Caso B: Rampa 1, Seno 4, Square

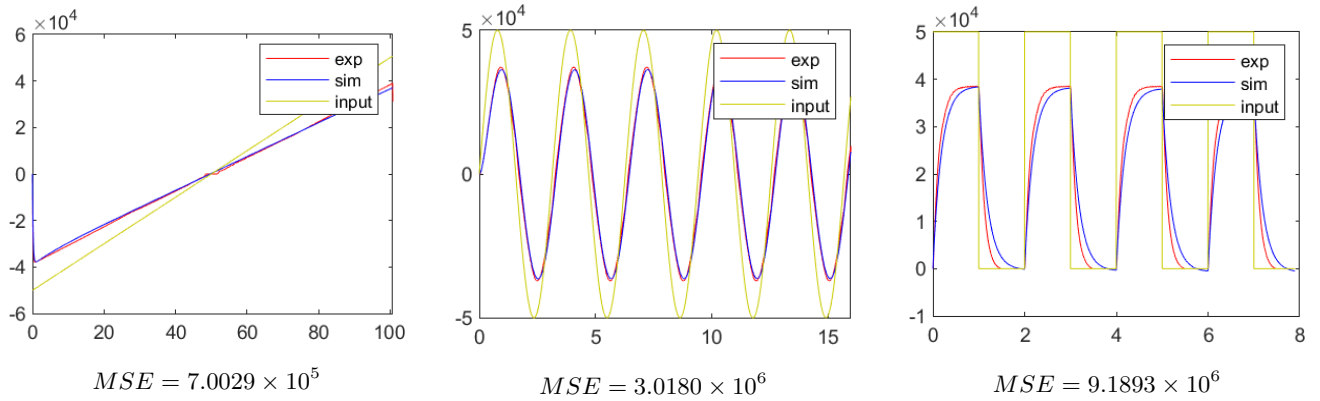


Figura 10. Caso C: Rampa 1, Seno 4, Square

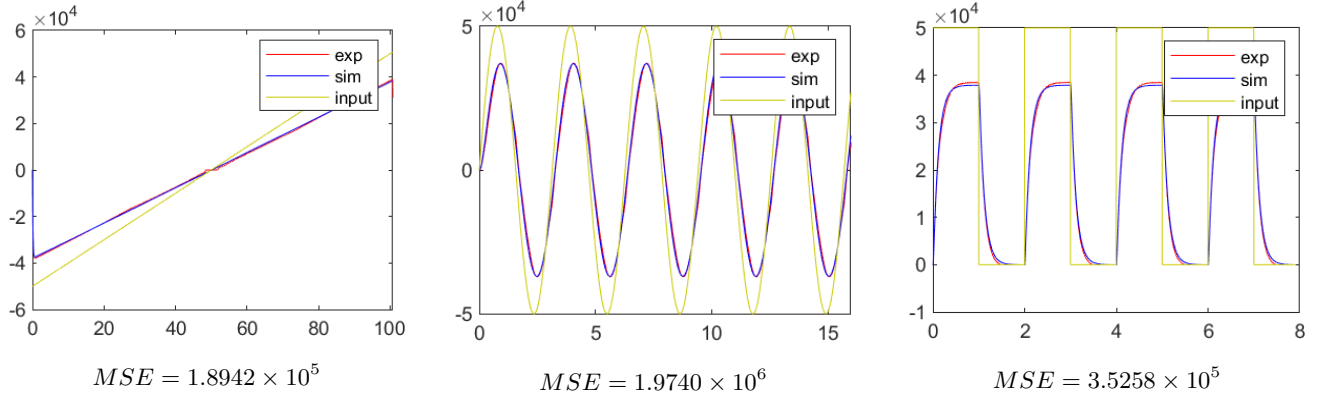


Figura 11. Caso D: Rampa 1, Seno 4, Square

APPENDICE B TABELLE DEL MSE

	$G_1(s)$	$G_2(s)$	$G_3(s)$	$G_4(s)$
ramp1.mat	2.564×10^6	5.326×10^6	7.001×10^5	1.880×10^5
ramp2.mat	2.791×10^6	3.820×10^6	4.666×10^5	2.164×10^5
ramp3.mat	3.480×10^6	3.776×10^5	2.123×10^5	3.805×10^5

Tabella I. RAMPE

	$G_1(s)$	$G_2(s)$	$G_3(s)$	$G_4(s)$
square.mat	1.347×10^8	6.838×10^6	9.187×10^6	3.529×10^5
scalini.mat	1.216×10^8	2.111×10^7	5.966×10^6	1.304×10^6

Tabella II. ONDE QUADRE

	$G_1(s)$	$G_2(s)$	$G_3(s)$	$G_4(s)$
sine1.mat	1.240×10^8	4.477×10^7	1.032×10^6	2.245×10^6
sine2.mat	7.496×10^7	7.920×10^6	9.466×10^5	1.321×10^6
sine3.mat	5.303×10^7	1.723×10^6	8.357×10^5	2.182×10^6
sine4.mat	1.730×10^8	6.567×10^6	3.019×10^6	1.971×10^6
sine5.mat	7.498×10^8	9.383×10^5	2.962×10^6	2.319×10^6
sine6.mat	4.973×10^7	8.164×10^4	4.437×10^6	1.152×10^6

Tabella III. SENI CON F=COST

	$G_1(s)$	$G_2(s)$	$G_3(s)$	$G_4(s)$
seni_largeamplitude	3.558×10^8	1.959×10^7	3.212×10^7	2.821×10^6
_nooffset.mat	1.818×10^6	1.181×10^5	2.765×10^5	3.335×10^5
seni_offset2.mat	2.134×10^6	1.169×10^5	5.841×10^4	7.378×10^5

Tabella IV. SENI CON F≠COST

APPENDICE C CODICE

Il codice completo è disponibile sulla repository pubblica al link <https://github.com/Death-Bode/FdA1>