



**UNIVERSITÀ
DI PARMA**

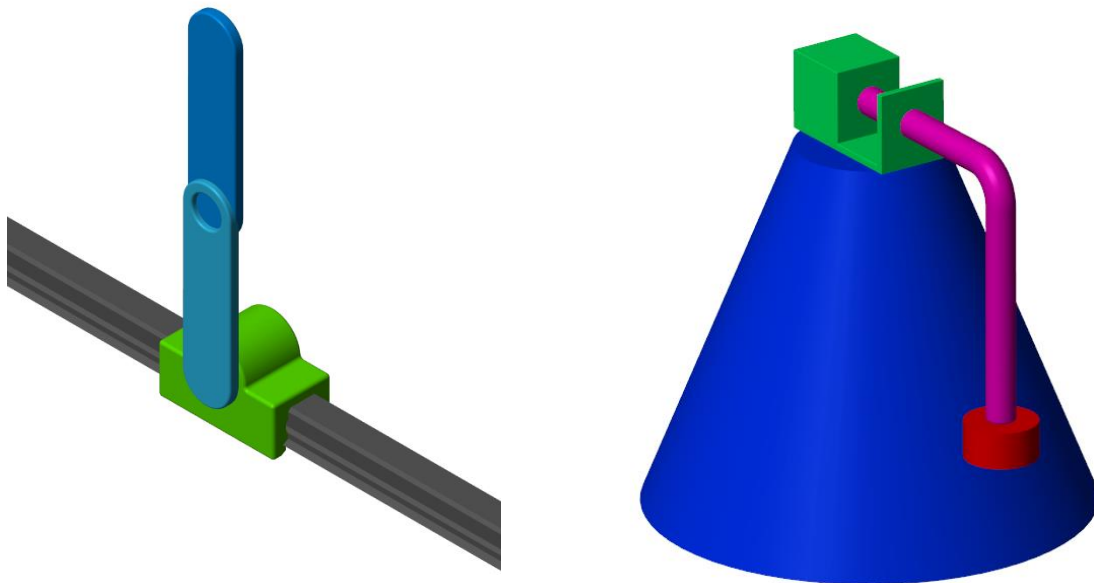
DIPARTIMENTO DI INGEGNERIA E ARCHITETTURA
Corso di Laurea Magistrale in Ingegneria Meccanica

Esame di:

“DINAMICA E CONTROLLO DEI SISTEMI MECCANICI”

REPORT TECNICO

“Analisi e controllo di sistemi meccanici a n -gradi di libertà”



Studenti:

Elisa Bertozzi	352195
Andrea Meni	353456
Gianluca Montomoli	353452
Roberto Ravaglia	356094

Anno Accademico 2023/24

Sommario

INTRODUZIONE	3
1. SINGOLO PENDOLO.....	4
1.1 Legge di moto: accelerazione – velocità - spostamento	5
1.2 Ampiezza oscillazioni	6
1.3 Simulink pendolo semplice	8
1.4 Simscape pendolo semplice.....	10
1.5 Variazione di parametri significativi	15
2. DOPPIO PENDOLO.....	18
2.1 Equazione del moto	19
2.2 Legge di moto: accelerazione – velocità – spostamento.....	20
2.3 Andamento oscillazioni.....	21
2.4 Simulink doppio pendolo	23
2.5 Simscape doppio pendolo	26
3. APPROFONDIMENTI	31
3.1 Doppio pendolo inverso	31
3.2 Pendolo di Furuta	35
4. APPENDICE	41
4.1 Pendolo semplice accelerazione ottimizzata.....	41
4.2 Function Legge_moto	42
5.3 Function Ampiezza_oscillazioni	42
4.4 Pendolo doppio accelerazione ottimizzata	43
4.5 Function Confr_amplitude.....	44
4.6 Torque.....	45

INTRODUZIONE

Un problema di controllo nasce nel momento in cui si vuole imporre ad un “oggetto” (la cui natura va di volta in volta precisata) un comportamento desiderato, per mezzo di opportune azioni esercitate sull’oggetto stesso. Il controllo rappresenta la scelta a disposizione di un controllore esterno per guidare opportunamente l’evoluzione di un sistema affinché produca la risposta, che in virtù di un determinato criterio ottimale, sia la migliore possibile.

Il sistema in oggetto è caratterizzato da due variabili distinte: input e output. Le prime, ossia le grandezze in ingresso che possono essere manipolate dal progettista, sono dette variabili di controllo, le seconde, che non possono essere manipolate dal progettista, sono dette disturbi.

L’evoluzione del sistema è descritta da un sistema di equazioni differenziali ordinarie (variabili di stato) in cui la dinamica dipende dalla scelta di parametri (controlli). L’insieme di queste informazioni costituisce il modello matematico del sistema, conosciuto il quale si può giungere alla soluzione di ciascun problema di controllo.

Chiaramente l’andamento effettivo delle variabili di controllo non sarà mai quello desiderato, poiché intervengono molteplici fattori quali i disturbi ed errori di approssimazione.

In particolare, nel seguente report, il primo sistema da controllare è il pendolo semplice. La grandezza in ingresso è rappresentata da una forzante esterna che applicata al carrello causa delle oscillazioni sullo stesso. La grandezza in uscita è costituita dall’ampiezza delle oscillazioni, che saranno valutate attraverso la risoluzione del modello matematico rappresentato dal pendolo semplice linearizzato.

Il secondo sistema riportato è costituito dal doppio pendolo fisico, caratterizzato da due gradi di libertà e con fulcro libero di traslare lungo l’asse X. Anche nel secondo caso in esame la grandezza in ingresso è rappresentata da un forzante applicata al carrello, mentre la grandezza in uscita è l’ampiezza delle oscillazioni che tenderà ad annullarsi poiché segue il modello matematico ottimizzato.

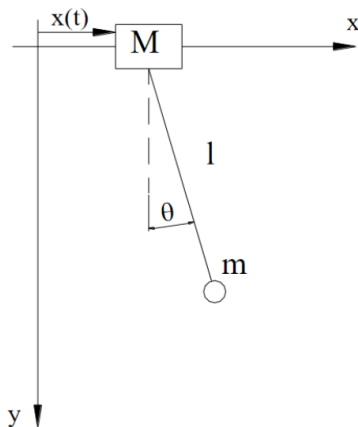
Per approfondire ulteriormente il secondo sistema analizzato è stato anche riportato il controllo del doppio pendolo inverso.

Il terzo sistema analizzato è il modello di un pendolo di Furuta, con implementazione del controllo in velocità angolare per il raggiungimento della posizione di equilibrio stabile.

Per ciascun tema vengono riportate la legge di moto ad accelerazione costante a tratti che minimizza le vibrazioni residue al termine dell’azionamento, lo spostamento del carrello, il numero di step ottimale per soddisfare i vincoli cinematici dell’azionamento, il modello Simulink per simulare il comportamento del sistema utilizzando un modello a blocchi e confrontare i risultati ottenuti precedentemente e la valutazione per l’introduzione di un controllo in feedback per ottenere vibrazioni residue nulle anche in presenza di errori di modellazione o non linearità.

Saranno infine verificate l’efficacia del controllo rispetto all’accuratezza richiesta e rispetto ad una legge di azionamento qualsiasi e la robustezza rispetto ad errori di modellazione (frequenze naturali del sistema fisico diverse da quelle ipotizzate nel modello).

1. SINGOLO PENDOLO



Massa	1 Kg
Lunghezza	0.7 m
Accelerazione di gravità	$9.81 \frac{m}{s^2}$
Spostamento richiesto	1.5 m
Tempo di fine azionamento	2.5 s

Figura 1 Singolo Pendolo

Il primo sistema analizzato è rappresentato dal seguente singolo pendolo. Sull'estremità superiore è presente un carrello fisico libero di traslare lungo l'asse X. Al carrello è collegata attraverso una fune inestensibile, di lunghezza "l", una massa "m". Le oscillazioni del pendolo saranno la diretta conseguenza della forza esterna applicata al carrello che permetterà a quest'ultimo di muoversi lungo l'asse orizzontale.

Il modello matematico utile per risolvere il sistema si può ricavare utilizzando l'equazione di Lagrange, legge che descrive il moto di un sistema soggetto a forze conservative. La legge del moto specifica per il sistema in esame è la seguente:

$$ml^2\ddot{\theta} + mgl\theta = m\ddot{x}$$

L'equazione in funzione del pendolo risulta essere:

$$\ddot{\theta} + \omega^2\theta = q\ddot{x}$$

È necessario assicurarsi che la legge utilizzata soddisfi tutti i vincoli del problema. Dovrà infatti garantire che alla fine dell'azionamento (la cui durata t_f è fissa) il carico si trovi nella posizione stabilita con velocità ed oscillazioni nulle. Limitare le oscillazioni residue del sistema significa rendere minima l'energia del sistema al termine dell'azionamento, rappresentata dalla seguente espressione:

$$E_{res} = \frac{1}{2}m\dot{\theta}^2 + \frac{1}{2}k\theta^2$$

Il ciclo di calcolo principale incrementerà il numero di step "n" fino a soddisfare tutti i vincoli imposti.

Per analizzare i grafici è stato necessario implementare le seguenti "function", il cui codice completo sarà riportato in appendice:

```
[aa,v,s,t]= Legge_moto(t_f,h,n);
[ta,x_1]= Ampiezza_oscillazioni(t_f,n,wn,h,FF,K1);
```

Figura 2 Function Matlab

1.1 Legge di moto: accelerazione – velocità - spostamento

Vengono riportati differenti grafici di accelerazione – velocità – spostamento poiché sono stati considerati molteplici valori dell'energia residua.

In primo luogo, è stata valutata la legge di moto assicurandosi che l'energia residua risultasse sempre minore di 10^{-4} .

La soluzione al problema è rappresentata da una legge di accelerazione in 4 step. Il risultato ottenuto è concorde con le aspettative poiché è immediato verificare che una legge di questo tipo soddisfi tutte e tre le condizioni imposte.

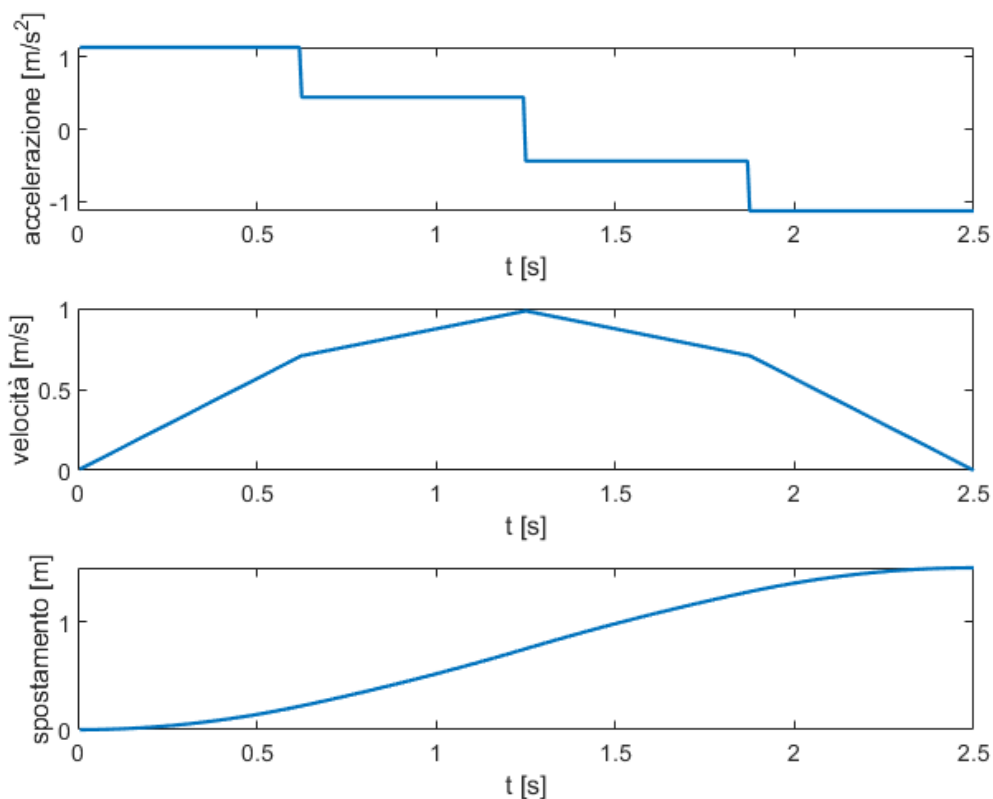


Figura 3 Legge di moto

Nel caso in esame si è mantenuto come vincolo riguardante la velocità finale un valore assoluto minore di 10^{-6} . Infatti, dal grafico della velocità è chiaramente visibile che al termine dell'azionamento, raggiunta la posizione imposta, il pendolo torna ad avere velocità nulla.

1.2 Ampiezza oscillazioni

È stato riportato anche l'andamento delle oscillazioni, plottato per un tempo maggiore di quello richiesto per verificare l'effettivo annullamento dopo 2.5 secondi. Il pendolo, come noto, dal grafico, durante lo spostamento del carrello, compie due oscillazioni con la medesima ampiezza ma di segno opposto. Ciò è coerente con la legge dell'accelerazione, infatti, durante i primi due gradini di accelerazione con valore positivo, l'oscillazione assume un valore negativo dovuto all'inerzia, durante gli ultimi due gradini con accelerazione negativa, il pendolo compie un'oscillazione positiva.

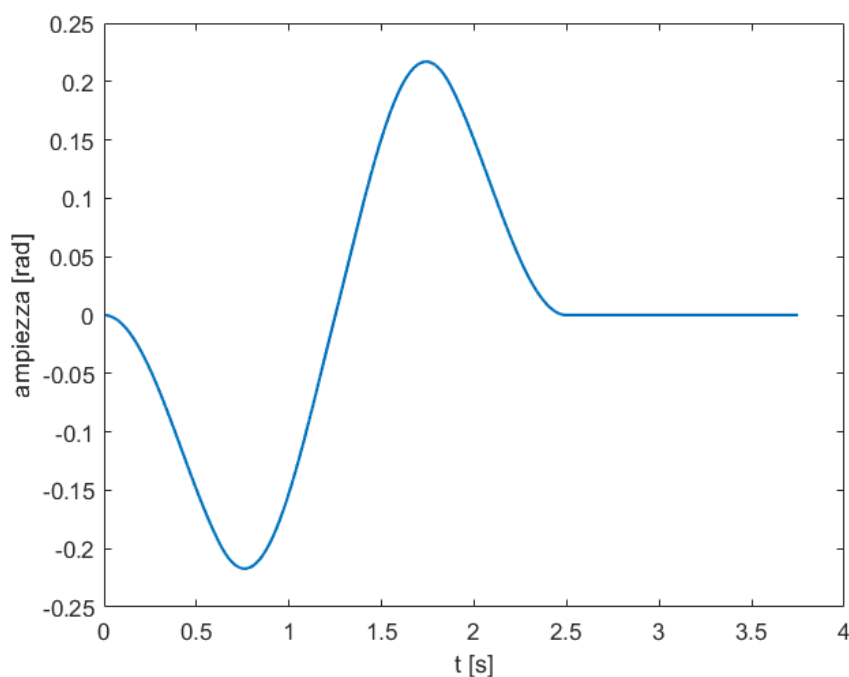


Figura 4 Ampiezza oscillazioni

Il valore minimo e massimo raggiunto dalle oscillazioni è pari a 0.205 radianti, cioè 11.75 gradi, e ciò risulta in linea con le ipotesi di piccole oscillazioni nell'intorno della posizione di equilibrio fatte inizialmente.

Nei grafici successivi vengono riportati le leggi di moto considerando come vincolo l'energia residua minore di 10^{-32} . Per quanto riguarda l'ampiezza delle oscillazioni, rimane costante nei diversi casi analizzati.

Nel grafico dell'accelerazione il numero di step risulta essere pari a 212. L'andamento dell'accelerazione è chiaramente ottimizzato e la velocità risulta annullarsi dopo 2.5 secondi come desiderato.

L'ampiezza delle oscillazioni rimane invariata e presenta inizialmente un andamento negativo dovuto all'inerzia e conseguentemente un valore opposto positivo.

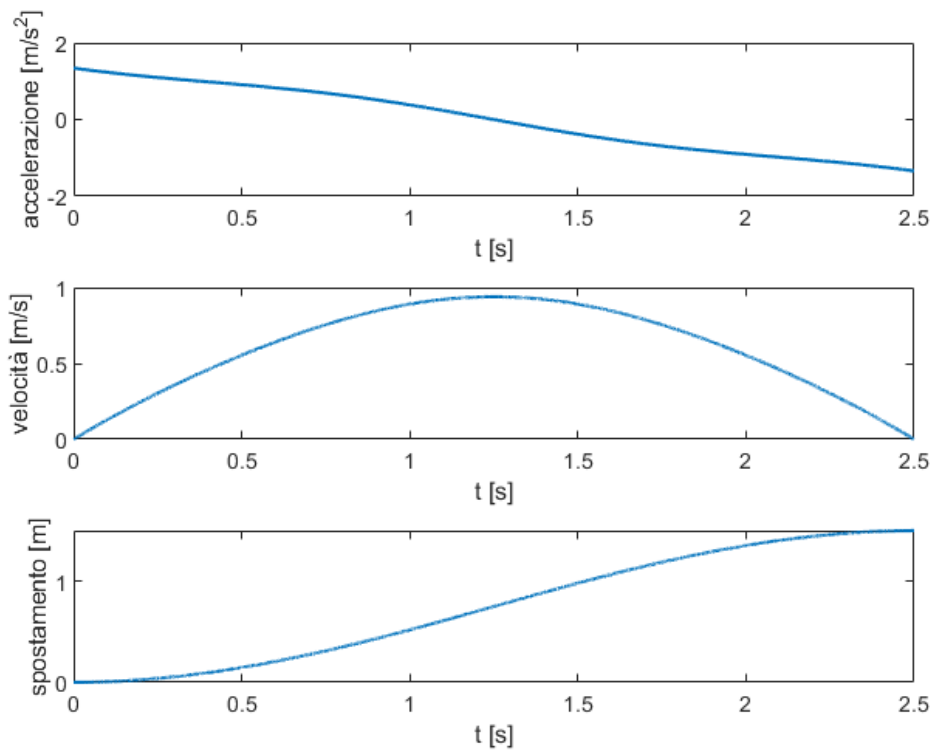


Figura 5 Legge di moto ottimizzata

Infine, viene riportato il grafico considerando come vincolo riguardante la velocità finale un valore assoluto minore di 10^{-16} .

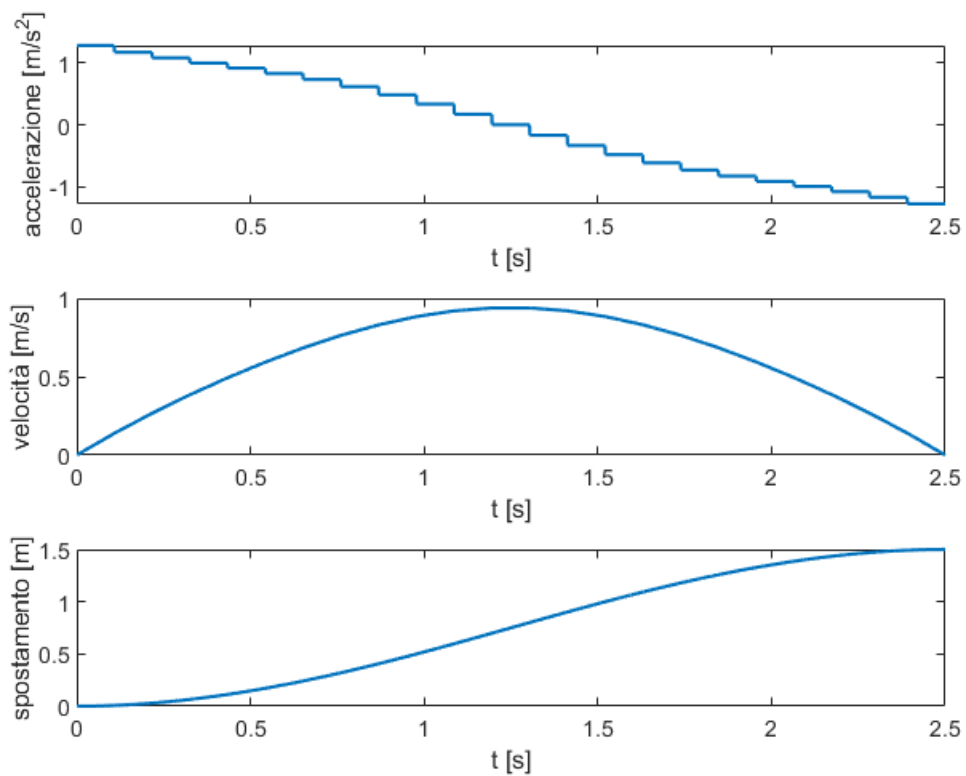


Figura 6 Legge di moto con $\text{abs}(v_f) < 10^{-16}$

1.3 Simulink pendolo semplice

Si è poi passati alla risoluzione del problema tramite l'utilizzo di Simulink, uno strumento intuitivo e veloce per la costruzione di modelli matematici a blocchi.

Si è andati a schematizzare l'equazione differenziale per il moto del pendolo semplice nel seguente modo, ponendo in ingresso la legge di accelerazione generata dal codice Matlab mostrato in precedenza, riuscendo così ad ottenere in uscita i valori di $\ddot{\theta}, \dot{\theta}, \theta$.

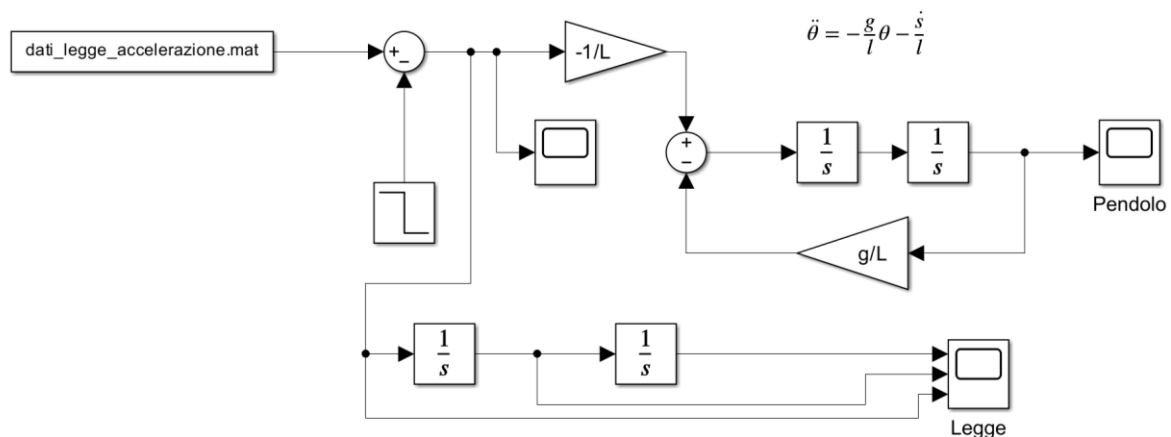


Figura 7 Schema Simulink

Di seguito vediamo la legge di accelerazione che è contenuta in “*dati_legge_accelerazione.mat*”, che porta il carrello ad una posizione di 1.5 metri dopo un tempo di 2.5 secondi, come da richiesta.

Si è andati ad inserire uno step a fine azionamento in modo da annullare l'accelerazione e poter così valutare le oscillazioni al termine della movimentazione. Il valore inserito all'interno dello step è pari al negativo dell'ultimo valore di accelerazione ottenuto, per ottenere un valore perfettamente nullo.

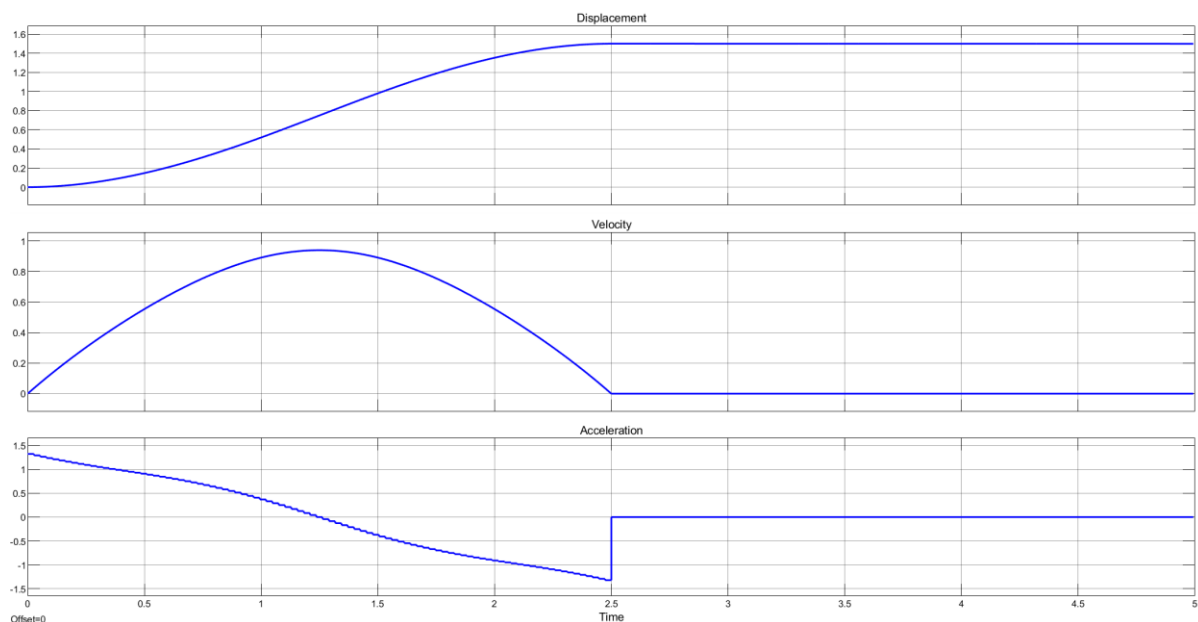


Figura 8 Andamento Accelerazione-Velocità-Spostamento Simulink

Il risultato ottenuto è quello di una oscillazione perfettamente sinusoidale senza nessuna oscillazione residua al termine dell'azionamento. L'oscillazione massima ottenuta è pari a 0.205 radianti, cioè 11.75 gradi, in linea con ciò che si è ottenuto su Matlab.

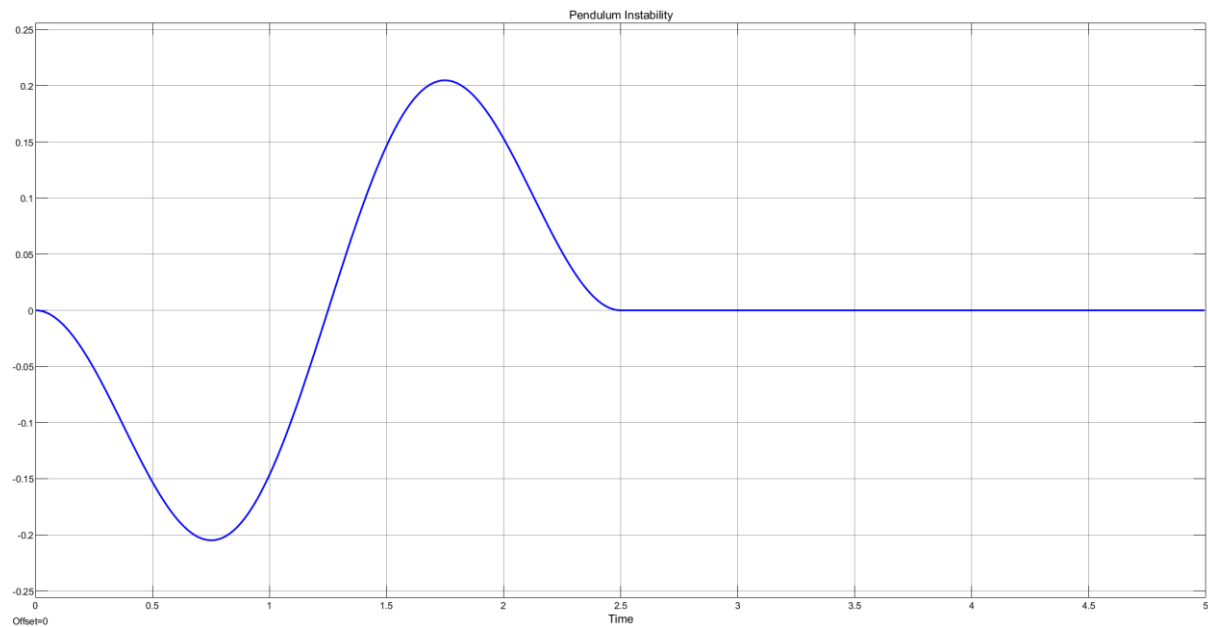


Figura 9 Andamento oscillazioni Simulink

1.4 Simscape pendolo semplice

Si introduce un modello di simulazione avanzato, implementato in Simscape, del sistema dinamico complesso, costituito da un pendolo semplice collegato a un carrello traslante.

Questa rappresentazione visiva consentirà una comprensione più approfondita della struttura e dell'interazione dei componenti nel contesto del simulatore.

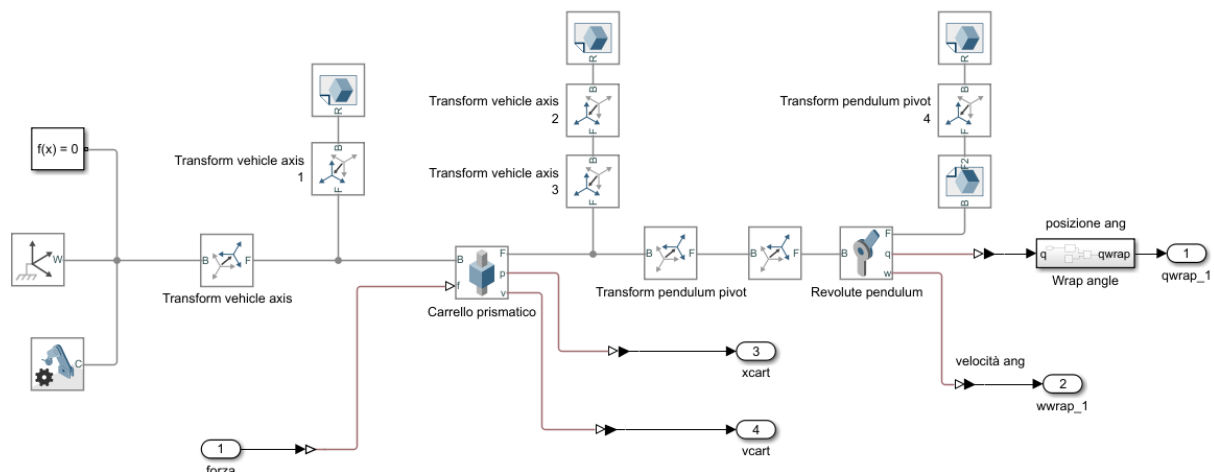


Figura 10 Simscape pendolo semplice

Si è implementata la legge di accelerazione con vincolo su energia residua $<10^{-32}$, progettata specificamente per minimizzare le oscillazioni residue al termine dell'azionamento. Questo approccio è finalizzato a ottimizzare la dinamica del pendolo e del carrello, consentendo una risposta più veloce e precisa alle variazioni nella posizione desiderata.

Si può vedere che è possibile implementare anche ulteriori leggi di moto, come la legge sinusoidale oppure la legge a tre step di accelerazione.

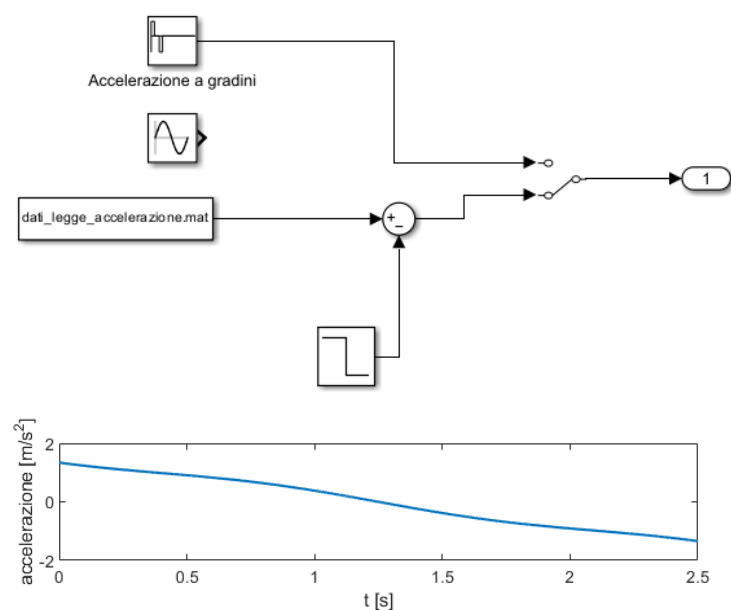


Figura 11 Input Simulink forzante

Si riporta in seguito il grafico dello spostamento e della velocità del carrello:

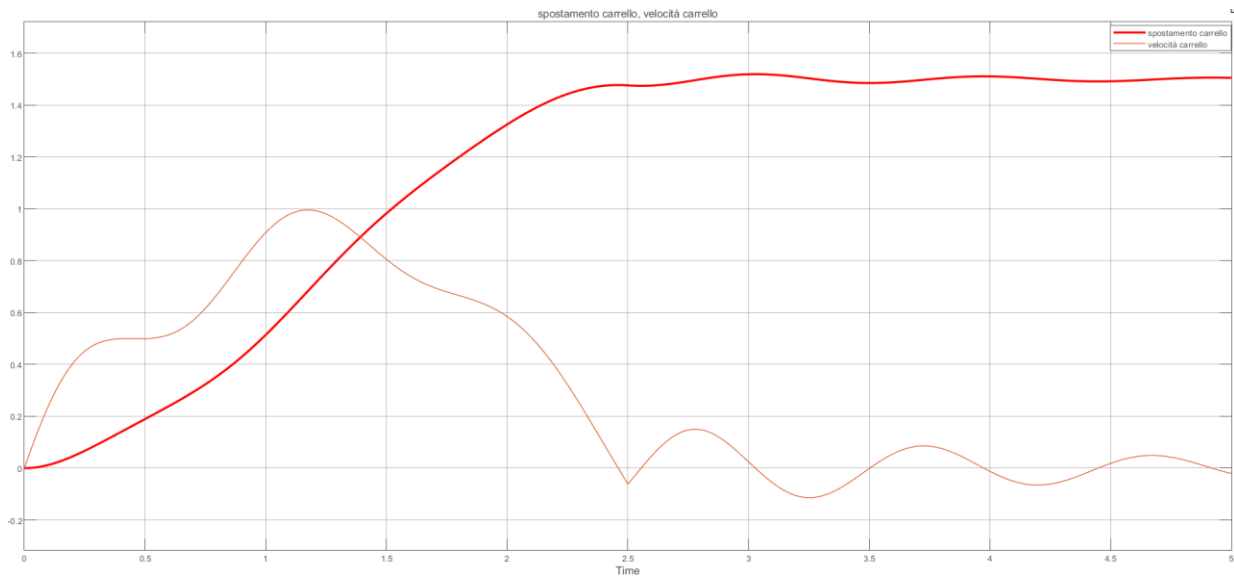


Figura 12 Spostamento e velocità del carrello

Al fine di potenziare la robustezza del sistema a errori di modellazione oppure di fronte a disturbi esterni, è stato integrato un sistema di controllo proporzionale, integrale e derivativo (PID).

Le costanti K_p , K_i e K_d sono state calcolate utilizzando il metodo manuale di Ziegler-Nichols che permette di trovarle con semplici e rapidi calcoli. Si riportano i calcoli effettuati in Excel:

Control Type	K_p	T_i	T_d	K_i	K_d
P	$0.5K_u$	–	–	–	–
PI	$0.45K_u$	$0.83T_u$	–	$0.54K_u/T_u$	–
PD	$0.8K_u$	–	$0.125T_u$	–	$0.10K_uT_u$
classic PID ^[2]	$0.6K_u$	$0.5T_u$	$0.125T_u$	$1.2K_u/T_u$	$0.075K_uT_u$

CALCOLI ZIEGLER-NICHOLS PER PID PENDOLO SEMPLICE CON CARRELLO					
Periodo T_u	9,5				
K_u	5				
	K_p	T_i	T_d	K_i	K_d
P	2,5				
PI	2,25	4		0,284211	
PD	4	2,5	1,1875		4,75
classic PID	3	2,5	1,1875	0,631579	3,5625

Figura 13 Calcoli Excel

Questo meccanismo agisce successivamente all'azionamento, apportando correzioni dinamiche che garantiscono una stabilità superiore e una maggiore resistenza ad eventuali perturbazioni.

È stato inserito, inoltre, un segnale casuale di rumore che simula eventuali disturbi esterni che influenzano il comportamento del sistema.

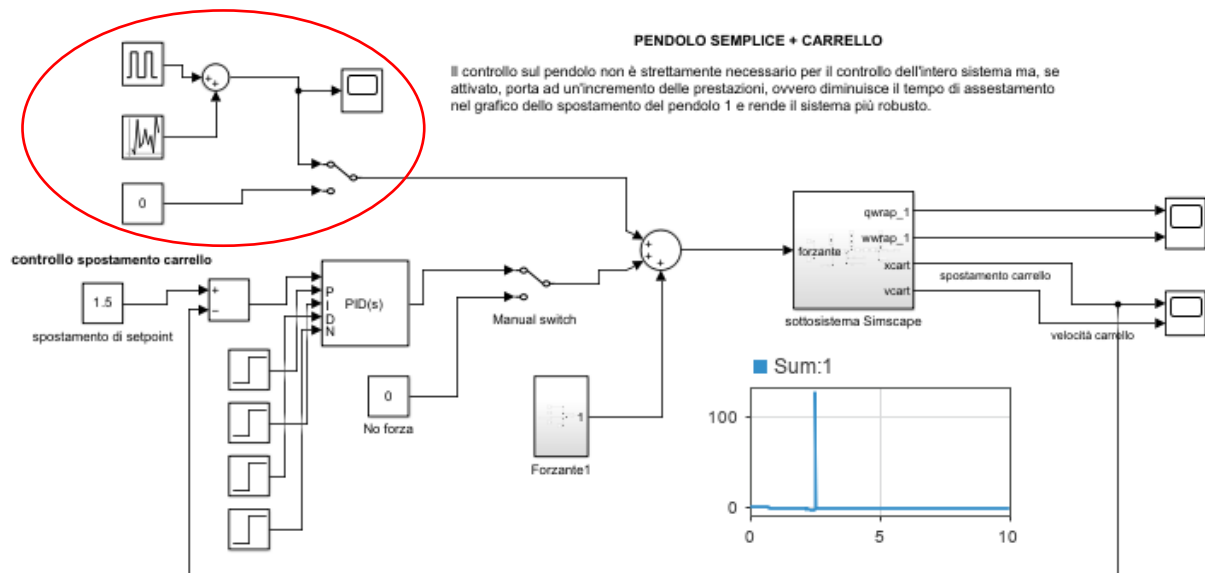


Figura 15 Simulink Disturbi

Sono riportati i nuovi grafici di spostamento e velocità del carrello con il controllo PID attivato dopo 2.5s e con l'influenza dei disturbi casuali esterni.

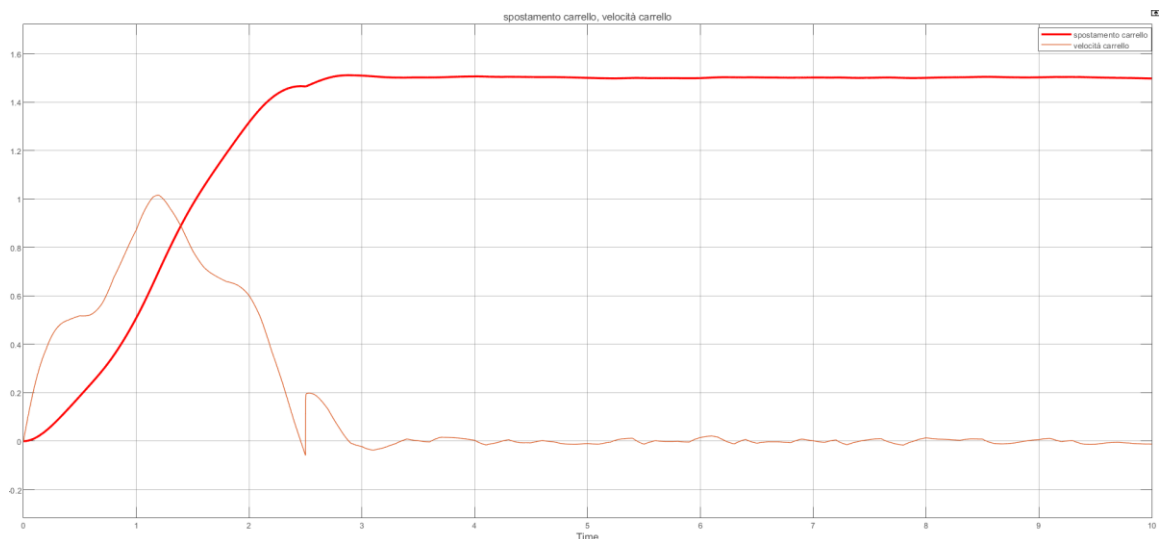


Figura 16 Spostamento e velocità carrello

Si può notare che il controllo PID agisce in corrispondenza della fine dell'azionamento a 2.5s, lo spostamento finale del sistema rimane nell'intorno di 1.5m, con oscillazioni residue circa nulle.

La velocità finale risulta circa nulla e con oscillazioni residue trascurabili.

È stato considerato anche il caso in cui siano presenti errori di modellazione. Di conseguenza la frequenza propria del sistema non risulta essere quella calcolata in precedenza ma presenta un'incertezza del $\pm 10\%$.



ω_n 3.7436

I nuovi grafici che si riportano in seguito sono i grafici dello spostamento del carrello in funzione del tempo, rispettivamente nel caso in cui $\omega_n = 3.3692(-10\%)$ e $\omega_n = 4.1179(+10\%)$.

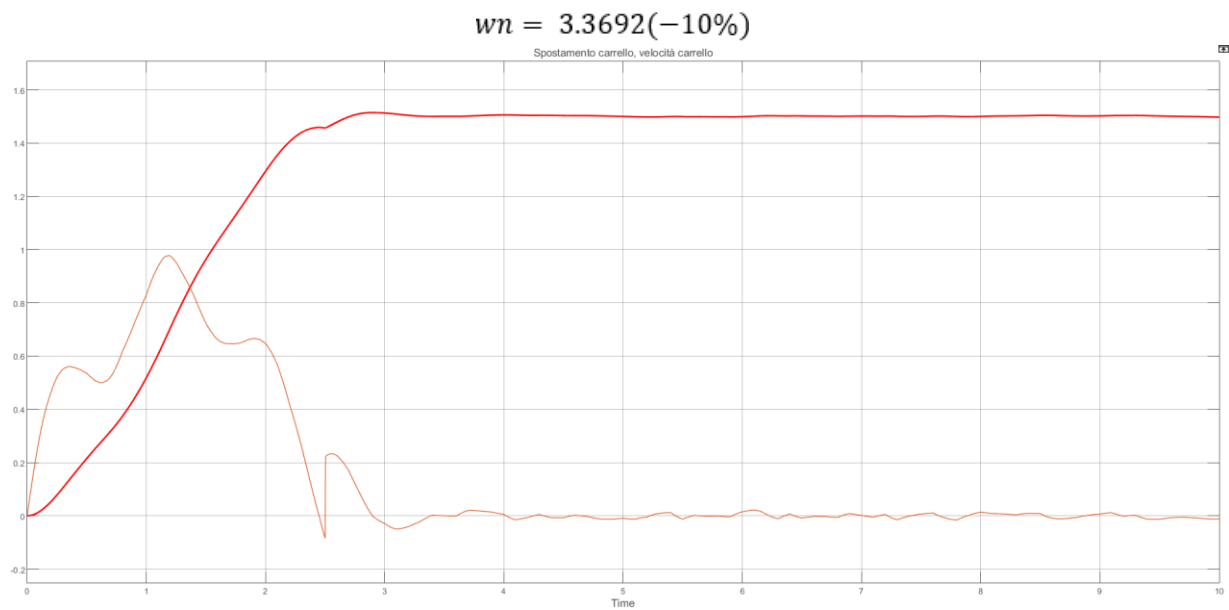


Figura 17 Spostamento e velocità carrello $\omega_n -10\%$

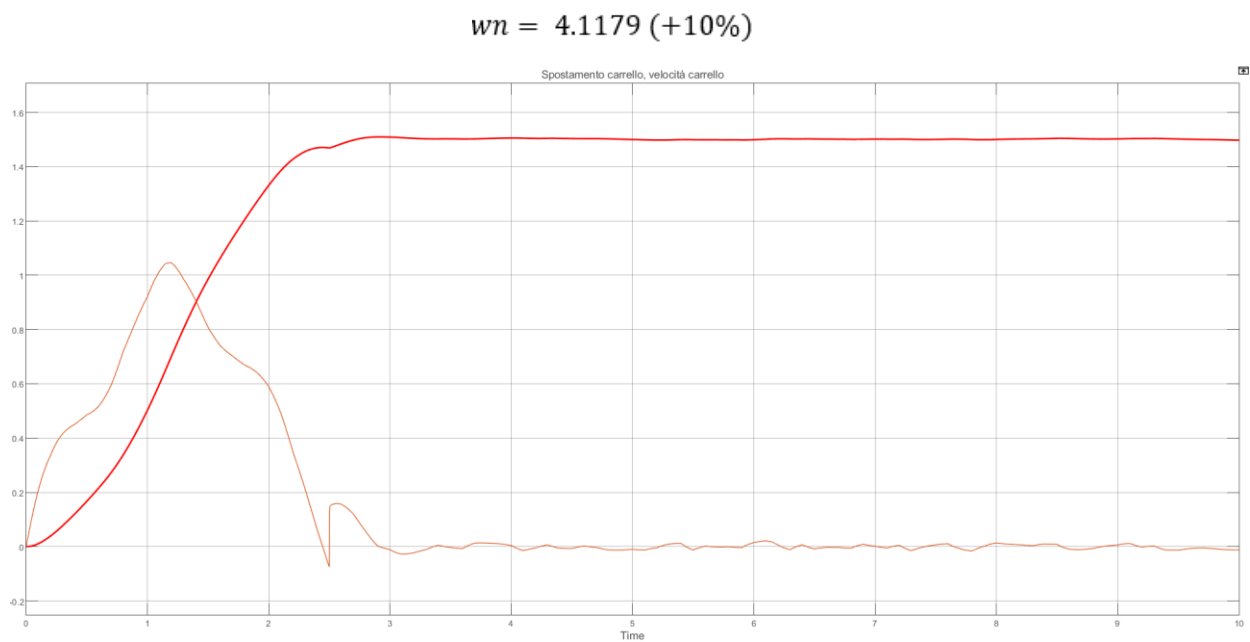
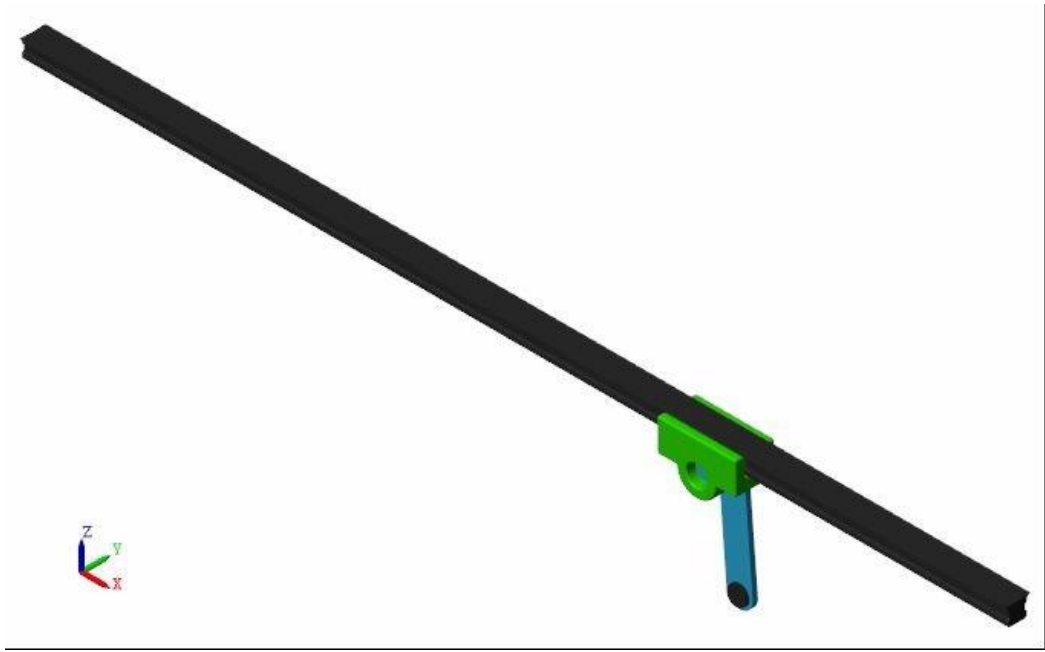


Figura 18 Spostamento e velocità carrello $\omega_n +10\%$

Il controllo risulta stabile anche in presenza di piccole incertezze sulla frequenza propria del sistema.

Dopo aver completato la simulazione, è possibile procedere alla visualizzazione dell'ambiente Multibody di Matlab "Mechanics Explorers", il quale illustra il movimento del pendolo nel corso del tempo simulato:



1.5 Variazione di parametri significativi

Per valutare la robustezza del modello realizzato abbiamo provato a variare alcuni parametri significativi, con i risultati che esponiamo di seguito. Per prima cosa abbiamo valutato la variazione dell'ampiezza delle oscillazioni al variare del tempo di azionamento, mantenendo l'energia residua costante. Un primo tentativo è stato quello di raddoppiare e dimezzare il tempo di azionamento mantenendo un'energia residua $E_{res} > 10^{-16}$.

Come si osserva nei grafici riportati di seguito, all'aumentare del tempo di azionamento l'ampiezza delle oscillazioni diminuisce ma si ripetono una volta in più. Diminuendolo, invece, l'ampiezza delle oscillazioni aumenta. Un tempo di azionamento minore significa dover raggiungere la distanza con una velocità maggiore, quindi con un'accelerazione maggiore. Ciò significa maggior forza di inerzia e maggior ampiezza delle oscillazioni.

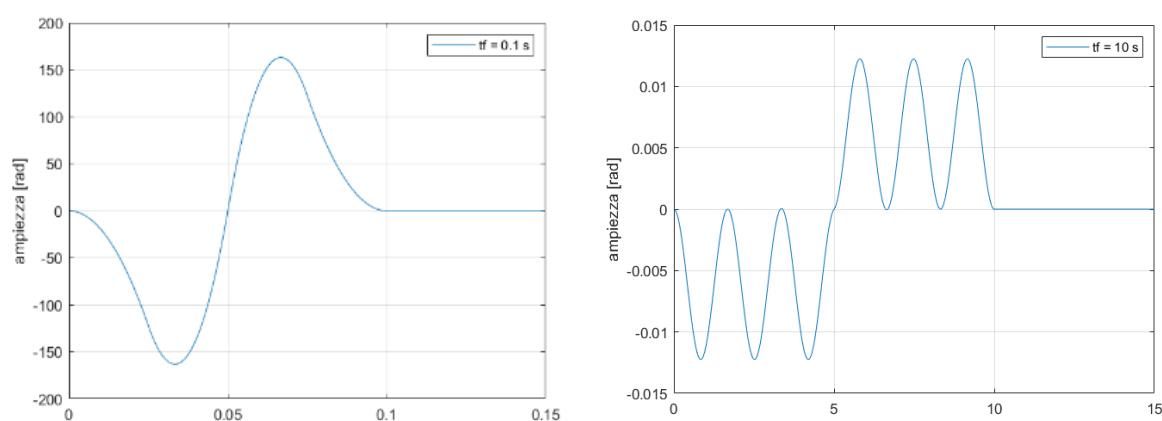


Figura 19 Confronto al variare del tempo di azionamento

Il caso mostrato nella figura di sinistra è sicuramente un caso limite, in cui il carrello deve raggiungere la posizione desiderata in soli 0.1 s. Resta, però, particolarmente esplicativo in merito all'aumento dell'ampiezza delle oscillazioni.

In secondo luogo, abbiamo analizzato la risposta del sistema al variare della lunghezza del pendolo. Osserviamo che l'equazione che determina l'ampiezza delle oscillazioni implementata in Matlab non dipende direttamente dalla lunghezza del pendolo ma all'interno della funzione coseno c'è la frequenza propria del sistema, che dipende da \sqrt{L} . Infatti, i termini FF e K1 dipendono da L ma, nel rapporto, L si semplifica. In Appendice (Paragrafo 4.1) è riportato lo script Matlab cui si fa riferimento.

La legge che determina l'ampiezza delle oscillazioni è la seguente:

$$x(t) = \frac{FF}{K1} (1 - \cos(\omega_n t))$$

Con $FF = -\frac{1}{L}$ e $K1 = \frac{g}{L}$.

Essendo L la lunghezza del pendolo, questa entra in gioco, in merito alle oscillazioni, solo tramite la frequenza propria, cioè

$$\omega_n = \sqrt{\frac{g}{L}}$$

Per comprendere come variano dapprima le oscillazioni al variare di L, senza tener conto di energia residua e annullamento dell'ampiezza, bisogna analizzare la funzione x(t), riportata sopra.

All'aumentare di L, si avrà una diminuzione della frequenza della funzione coseno e, al contrario, al diminuire di L la frequenza della funzione coseno aumenterà.

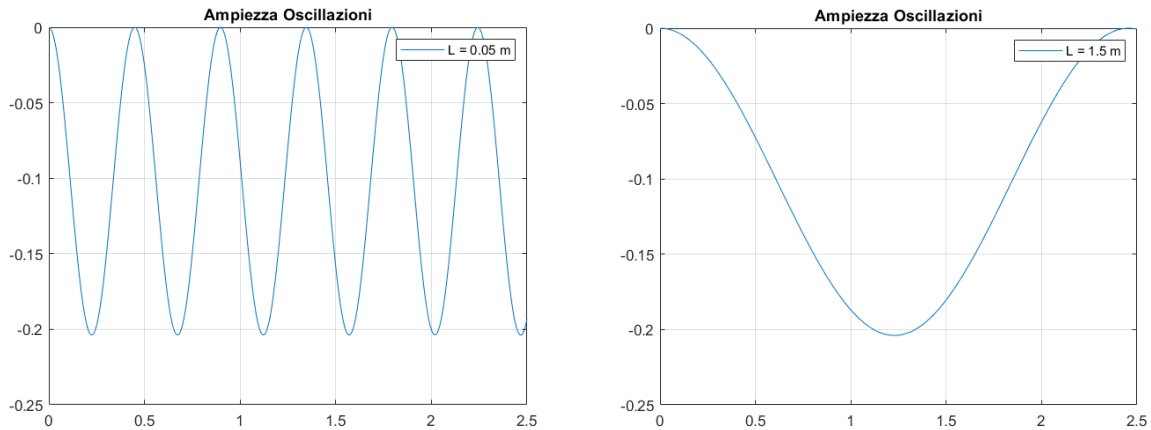


Figura 20 Confronto al varare della lunghezza del pendolo

Con l'implementazione del controllo sull'energia residua, presente nello script riportato nel Paragrafo 4.1, il comportamento delle oscillazioni non varia come un coseno. Mantenendo invariati i due valori di lunghezza analizzati precedentemente, si osserva che le oscillazioni variano più significativamente al diminuire di L piuttosto che all'aumentare dello stesso.

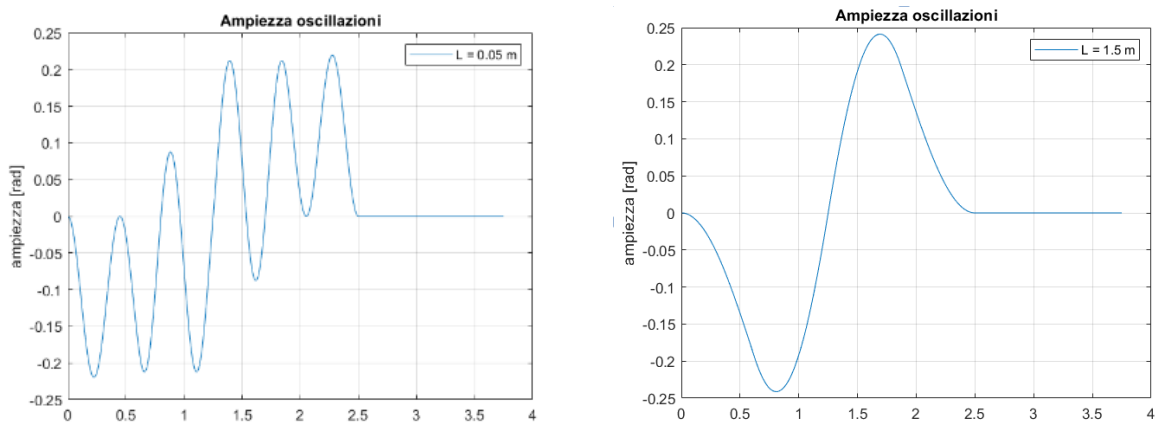


Figura 21 Confronto sulla lunghezza con implementazione del controllo sull'energia residua

L'ampiezza massima rimane pressoché costante ma diminuendo L varia significativamente la frequenza di oscillazione. Si conclude, quindi, che, per quanto riguarda il controllo delle oscillazioni, un pendolo più corto risulta più critico rispetto ad uno lungo, in un intorno di 0.7 m fisicamente significativo.

Infine, abbiamo aumentato la massa del pendolo del 20% ($M=1.2$ kg) e valutato la robustezza del controllo implementato nel modello Simulink. Di seguito si riporta il diagramma con spostamento e velocità del carrello.

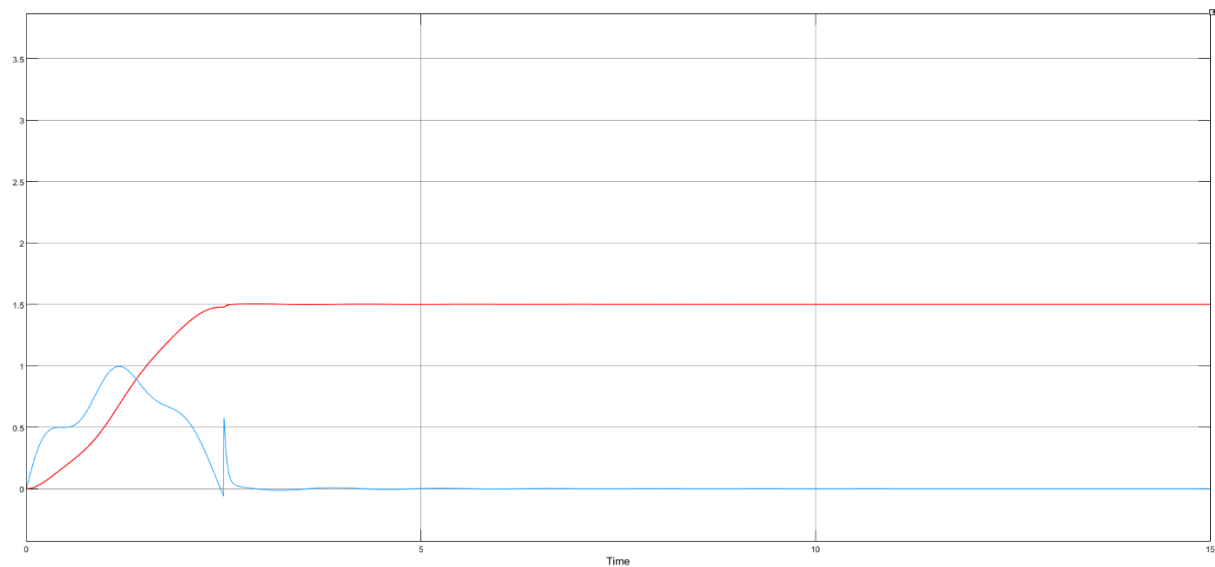


Figura 22 Spostamento e velocità carrello

Si osserva che, nonostante la variazione significativa apportata alla massa del pendolo, il carrello riesce a raggiungere la posizione di setpoint senza particolari difficoltà. Possiamo, quindi, concludere che il controllo è efficace nonostante un errore di modellazione della massa del pendolo del 20%.

Di seguito si riportano anche le oscillazioni del pendolo (posizione e velocità angolare).

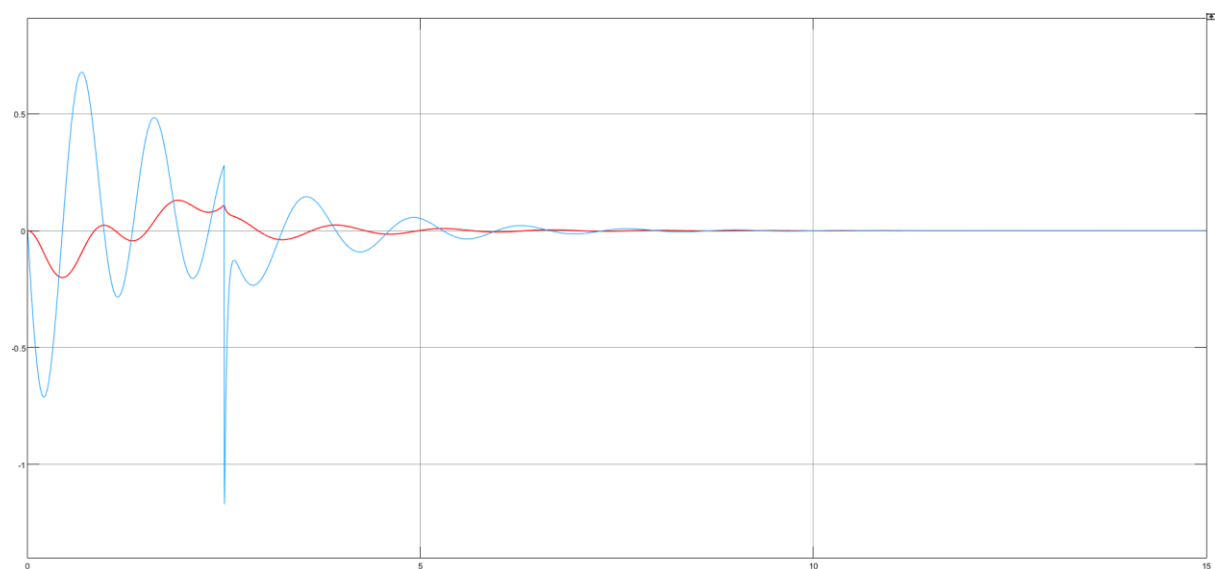
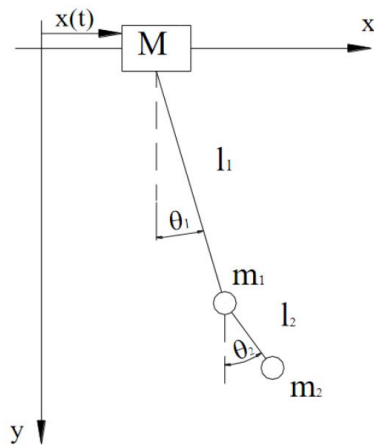


Figura 23 Oscillazione pendolo

2. DOPPIO PENDOLO



Massa 1	0.2 Kg
Massa 2	0.5 Kg
Lunghezza 1	1.05 m
Lunghezza 2	0.35 m
Accelerazione di gravità	$9.81 \frac{m}{s^2}$
Spostamento richiesto	1.5 m
Tempo di fine azionamento	2.7 s

Figura 24 Pendolo doppio

L'approccio al problema utilizzato per il singolo pendolo può essere facilmente esteso al doppio pendolo. L'obiettivo finale riguarda sempre la minimizzazione dell'energia residua e l'annullamento delle oscillazioni al termine dell'azionamento.

In figura 15 è schematizzato il sistema a due gradi di libertà. È presente un carrello, che può traslare orizzontalmente lungo l'asse x, e due masse m_1 e m_2 , unite tra loro da due funi inestensibili di lunghezza rispettivamente l_1 e l_2 . Muovendo il carrello attraverso una generica legge, le masse m_1 e m_2 iniziano ad oscillare fino al termine dello spostamento.

Nel piano, le rispettive posizioni delle masse possono essere individuate dagli angoli θ_1 e θ_2 , che costituiscono i due gradi di libertà del sistema. Come nel caso in esame, la massa M si considera trascurabile e rappresenta il fulcro del primo pendolo; in questo modo il carrello non risulta essere un grado di libertà del sistema.

Per il sistema costituito dal doppio pendolo, avente due gradi di libertà, le equazioni da risolvere saranno due e saranno disaccoppiate. Questa caratteristica semplifica la trattazione poiché sarà possibile utilizzare la trattazione del singolo pendolo in coordinate modali per ciascuno dei due modi.

Partendo dal codice utilizzato per il singolo pendolo, apportando opportune modifiche, si è giunti a rappresentare la legge di moto dell'intero sistema. L'obiettivo rimane quello di minimizzare le vibrazioni residue minimizzando l'energia residua e garantire che al termine dell'azionamento il carico si trovi nella posizione desiderata con velocità nulla.

Il procedimento dovrà essere ripetuto per ciascuno dei due modi.

2.1 Equazione del moto

Si è andati quindi a determinare l'equazione del moto per il sistema. Si è deciso di procedere alla risoluzione delle equazioni del moto tramite le funzioni simboliche presenti in Matlab, che permettono di ottenere un sistema di equazioni differenziali letterali che esprimono le variabili indipendenti di interesse $\ddot{\theta}_1, \ddot{\theta}_2$.

Si è andati a definire tutte le variabili simboliche e, come da commenti, la matrice di rigidità K, la matrice delle forze F e le matrici relative a θ . L'equazione del moto è definita nella penultima riga, derivata dall'equazione 1.11 trattata nel capitolo 1 di "Riduzione delle Vibrazioni Residue".

Tramite la funzione "solve" Matlab calcola le equazioni del moto, che vengono successivamente semplificate attraverso il comando "simplify".

```
syms m1 m2 l1 l2 theta2ddot theta1ddot theta2 theta1 g xddot F1 F2

M=([(m1+m2)*l1^2,m2*l1*l2;m2*l1*l2,m2*l2^2]); %Matrice di massa

F=(-(m1+m2)*l1;-m2*l2)*xddot; %Matrice delle forze

K=[(m1+m2)*g*l1,0;0,m2*g*l2]; %Matrice di rigidità

thetaddot=[theta2ddot;theta1ddot]; %Matrice accelerazioni ang.

theta=[theta1;theta2]; %Matrice spostamenti ang.

eqn= thetaddot==(M^-1)*(F-K*theta); %Equazione del moto

C=solve(eqn,thetaddot);
```

Figura 25 Definizione variabili

Le equazioni ottenute grazie allo script sono le seguenti:

$$\ddot{\theta}_1 = -\frac{m_1 \ddot{x} + g(m_1 \theta_1 + m_2 \theta_1 - m_2 \theta_2)}{l_1 m_1}$$

$$\ddot{\theta}_2 = \frac{g(m_1 + m_2)(\theta_1 - \theta_2)}{l_2 m_1}$$

- m_1, m_2 sono le masse dei pendoli
- θ_1, θ_2 sono gli angoli di deviazione dei pendoli dalla verticale
- l_1, l_2 sono le lunghezze dei pendoli
- g è l'accelerazione dovuta alla gravità
- $\ddot{\theta}_1, \ddot{\theta}_2$ sono le seconde derivate rispetto al tempo degli angoli
- \ddot{x} rappresenta l'accelerazione fornita al vincolo tra il carrello e il pendolo 1.

2.2 Legge di moto: accelerazione – velocità – spostamento

Di seguito vengono riportati i grafici relativi all'accelerazione, alla velocità e allo spostamento ottenuti, avendo impostato lo spostamento del carrello pari 1.5 m e il tempo di fine azionamento pari a 2.7 secondi.

In primo luogo, è stata valutata la legge di moto assicurandosi che l'energia residua risultasse sempre minore di 10^{-4} e il valore assoluto di velocità minore di 10^{-6} .

La soluzione al problema è rappresentata da una legge di accelerazione in cinque step. Il risultato ottenuto è concorde con le aspettative poiché è immediato verificare che una legge di questo tipo soddisfi tutte e tre le condizioni imposte.

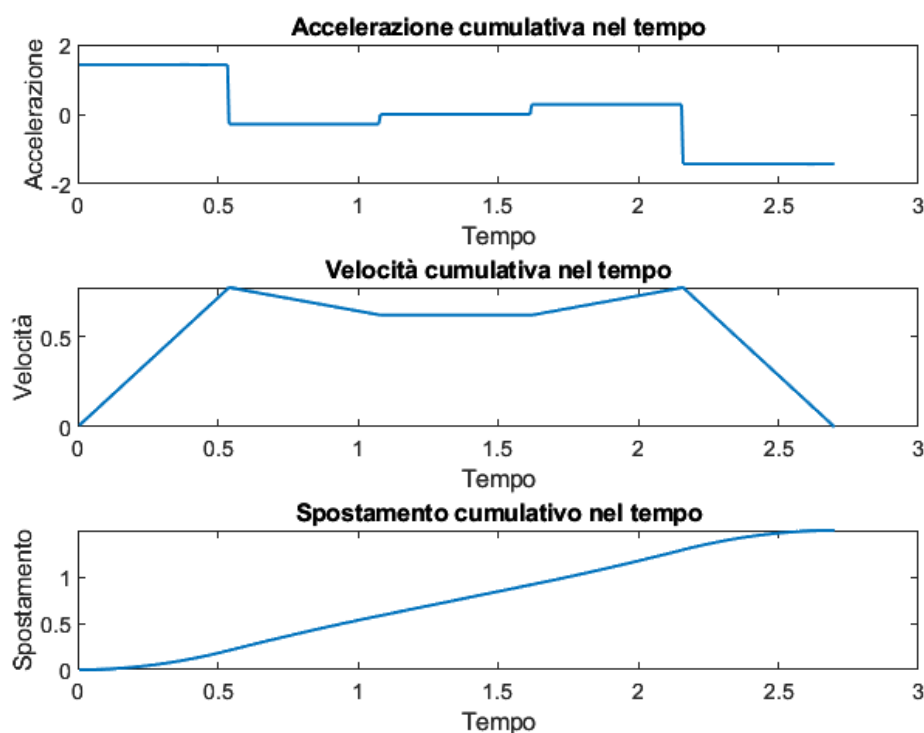


Figura 24 Legge di Moto

Nella figura 16 si può notare che la legge non è perfettamente lineare ed il motivo principale è la minimizzazione del "jerk" ossia, la minimizzazione della derivata dell'accelerazione. L'obiettivo rimane quello di frenare l'intero sistema in modo graduale.

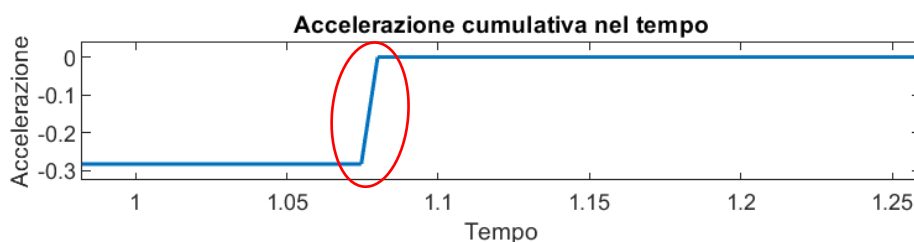


Figura 25 Gradino di Accelerazione

2.3 Andamento oscillazioni

Per valutare le oscillazioni del doppio pendolo si è utilizzata la funzione “*confr_amplitude.m*” presente in GARZIERA R., MANCONI E., *Preshaping Command Inputs For Reducing Residual Vibration In Point-To-Point Motion*, Proc. IMAC-XXIII, Orlando, Florida, 2005. Anche nel caso del doppio pendolo le oscillazioni si annullano al termine dell’azionamento. Il pendolo, come noto, dal grafico, durante lo spostamento del carrello, compie due oscillazioni con la medesima ampiezza ma di segno opposto.

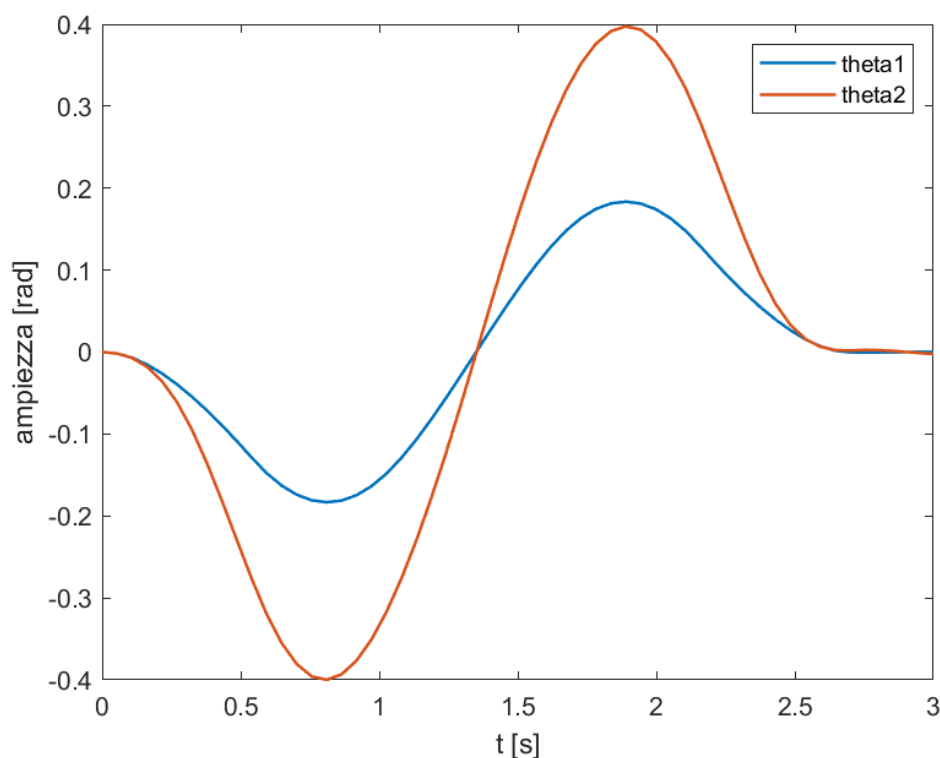


Figura 26 Andamento oscillazioni

Il valore minimo e massimo raggiunto dalle oscillazioni è pari a 0.4 radianti e ciò risulta in linea con le ipotesi di piccole oscillazioni nell’intorno della posizione di equilibrio fatte inizialmente.

Nei grafici successivi vengono riportati le leggi di moto e l’ampiezza delle oscillazioni considerando come vincolo l’energia residua minore di 10^{-29} .

Nel grafico dell’accelerazione il numero di step risulta essere pari a sette. L’andamento dell’accelerazione è chiaramente ottimizzato e la velocità risulta annullarsi dopo 2.7 secondi come desiderato.

L’ampiezza delle oscillazioni rimane invariata e presenta inizialmente un andamento negativo dovuto all’inerzia e conseguentemente un valore opposto positivo.

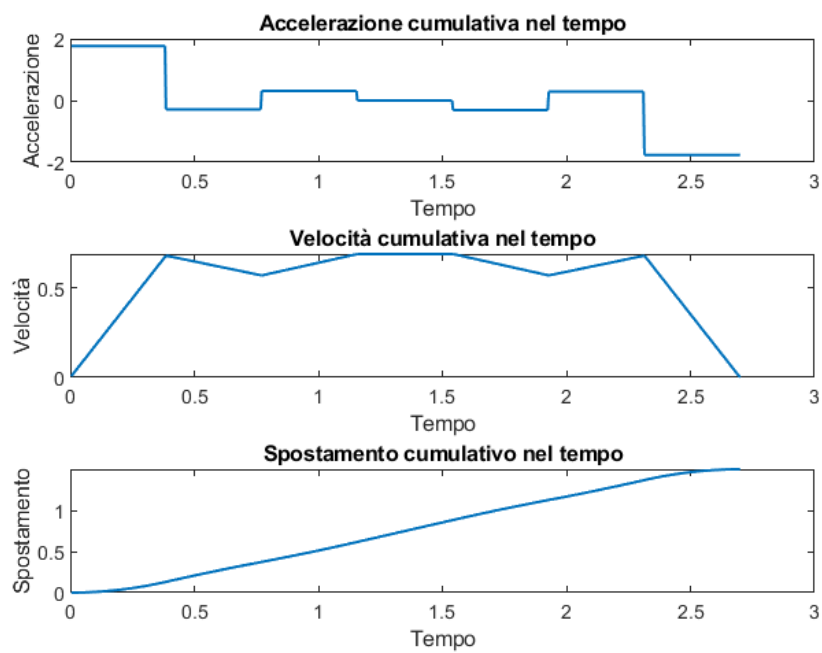


Figura 27 Legge di moto ottimizzata

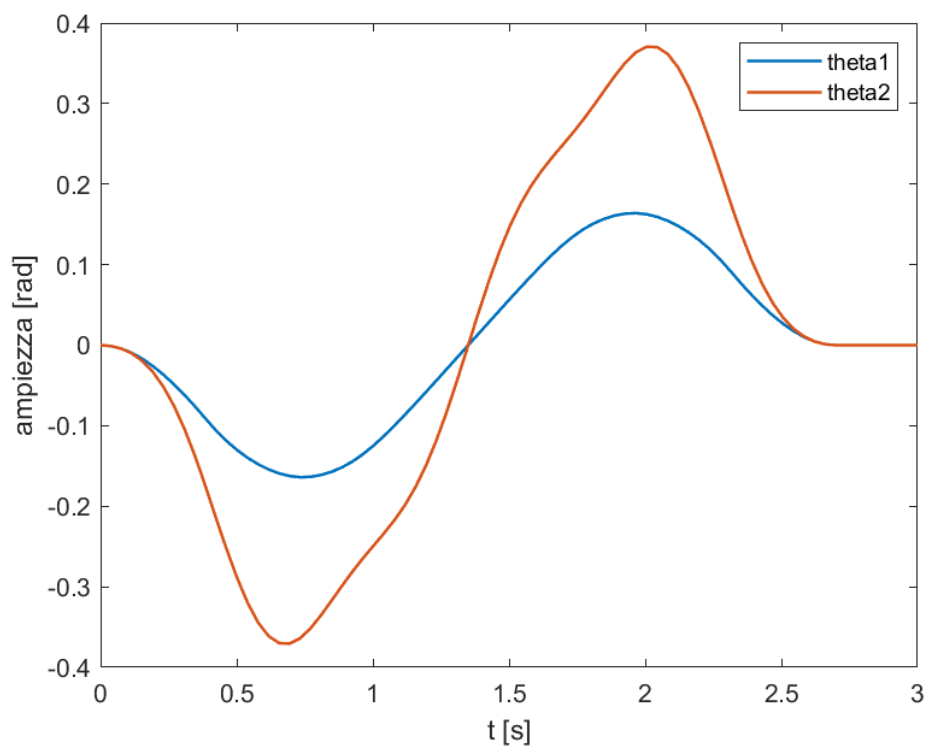


Figura 28 Andamento oscillazioni ottimizzato

2.4 Simulink doppio pendolo

Si è poi passati alla risoluzione del problema tramite l'utilizzo di Simulink.

Si è andati a schematizzare il sistema di equazioni differenziali per il moto del pendolo doppio nel seguente modo, ponendo in ingresso la legge di accelerazione generata dal codice Matlab mostrato in precedenza, riuscendo così ad ottenere in uscita i valori di $\ddot{\theta}, \dot{\theta}, \theta$ per entrambi i pendoli.

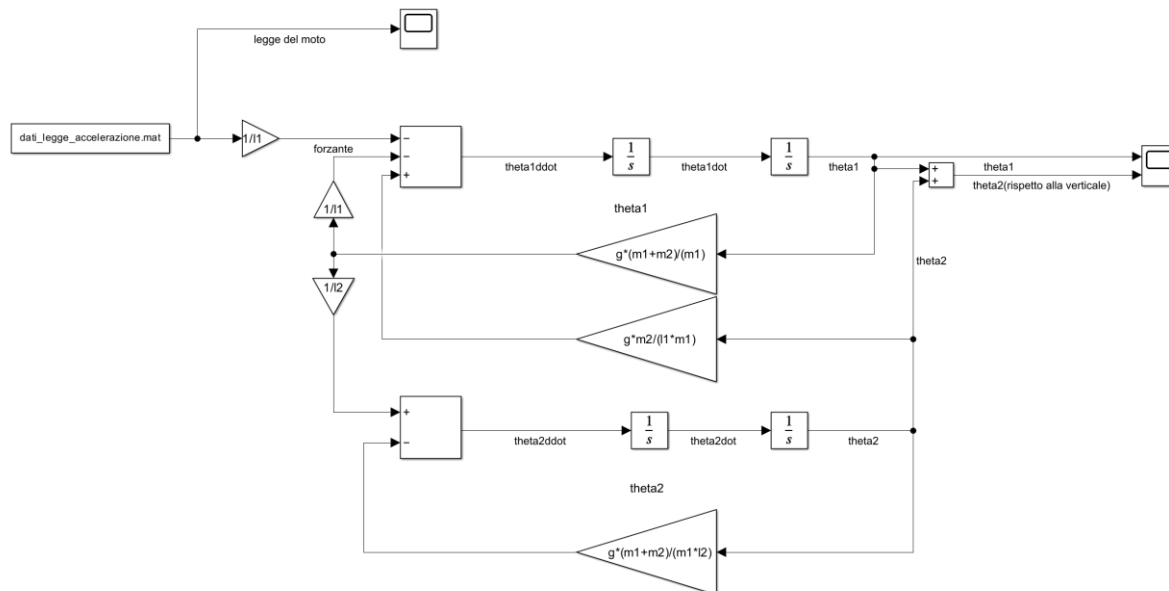


Figura 29 Simulink doppio pendolo

Il sistema di equazioni si presenta nel seguente modo. La forzante è stata fornita al sistema utilizzando un file ". mat" come fatto in precedenza. Il valore di θ_2 viene rappresentato prendendo come riferimento la verticale, andando a sommare il valore di θ_1 al valore proprio di θ_2 .

La legge del moto fornita è la legge ottimizzata, che si può visualizzare nel seguente grafico, con valori di energia residua $<10^{-29}$:

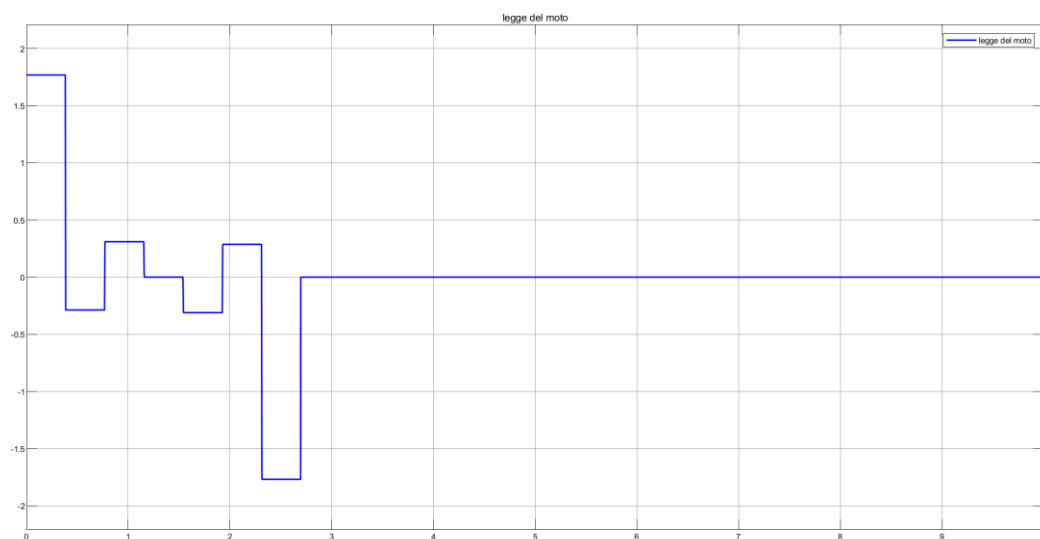


Figura 30 Legge accelerazione

Vediamo di seguito i risultati ottenuti per quanto riguarda gli angoli θ_1 , θ_2 .

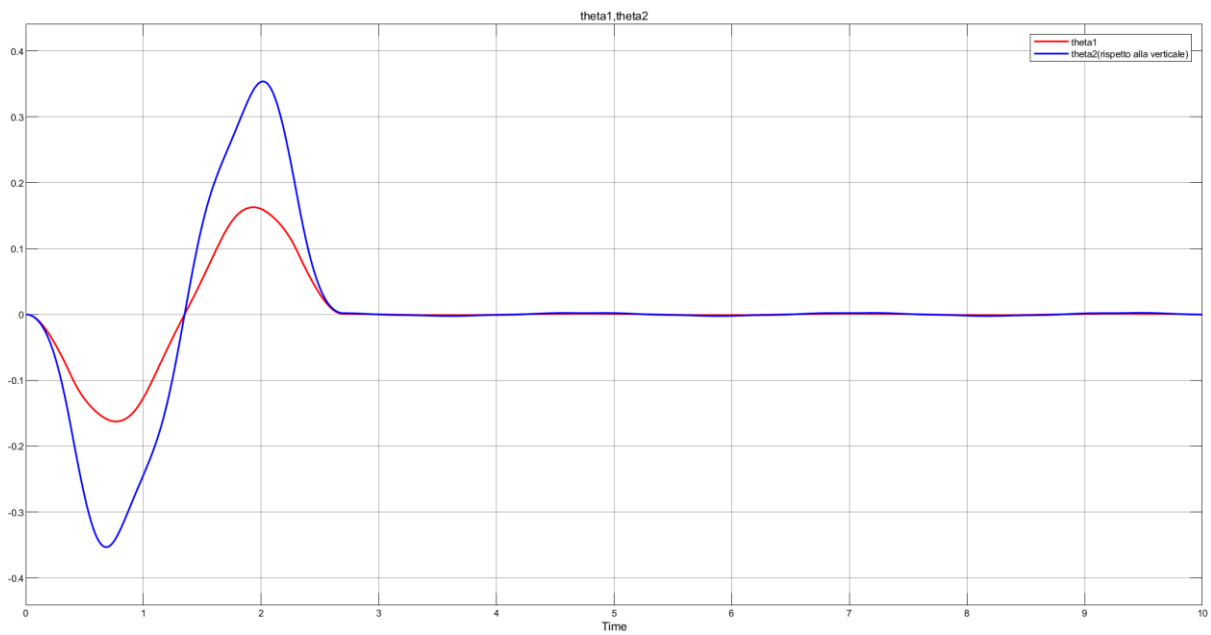


Figura 31 Andamento oscillazioni

Possiamo ora visualizzare lo spostamento delle masse m_1 e m_2 . La legge del moto consente di mantenere perfettamente la posizione di 1.5 metri alla fine dell'azionamento. I valori di spostamento x_1, x_2 sono calcolati attraverso le seguenti equazioni:

$$x_1 = x_{\text{carrello}} + l_1 * \sin(\theta_1)$$

$$x_2 = x_1 + l_2 * \sin(\theta_2)$$

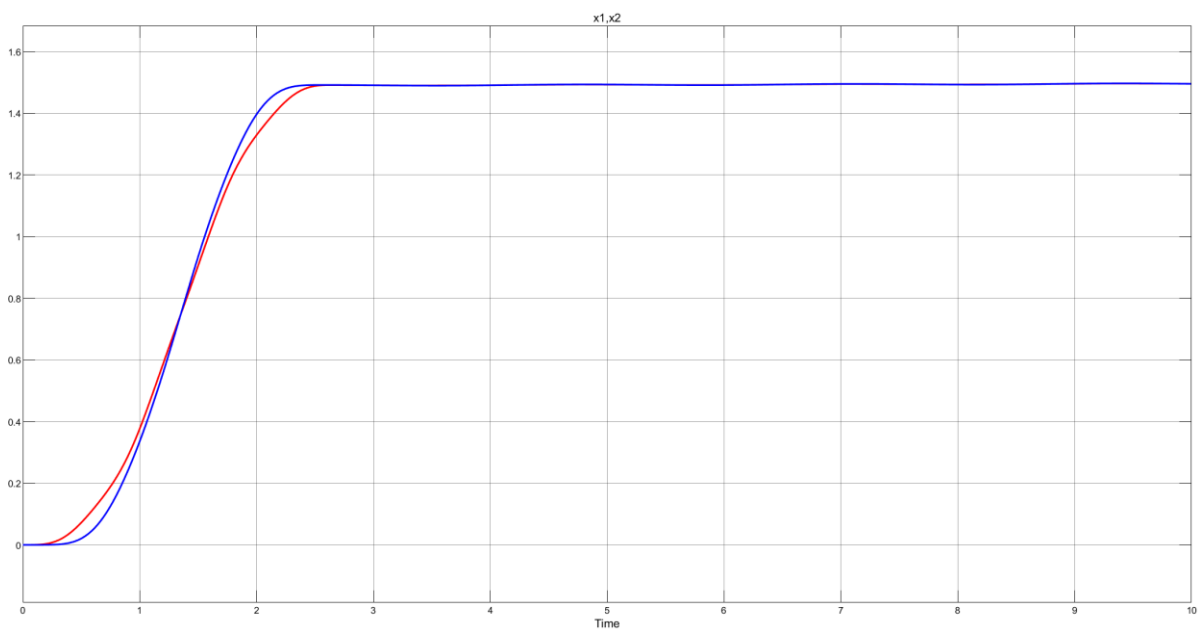


Figura 32 Spostamento

Viene poi valutata la risposta del sistema nel caso vengano forniti dei dati scorretti o contenenti errori di misurazione. Nel seguente caso è stato scelto di simulare il sistema utilizzando una lunghezza l_1 con un errore di misurazione del 10%, cioè

$$l_1 = 1.505 \text{ m}$$

Si può vedere dal seguente grafico che la posizione angolare dei due pendoli è ben differente dal caso precedente.

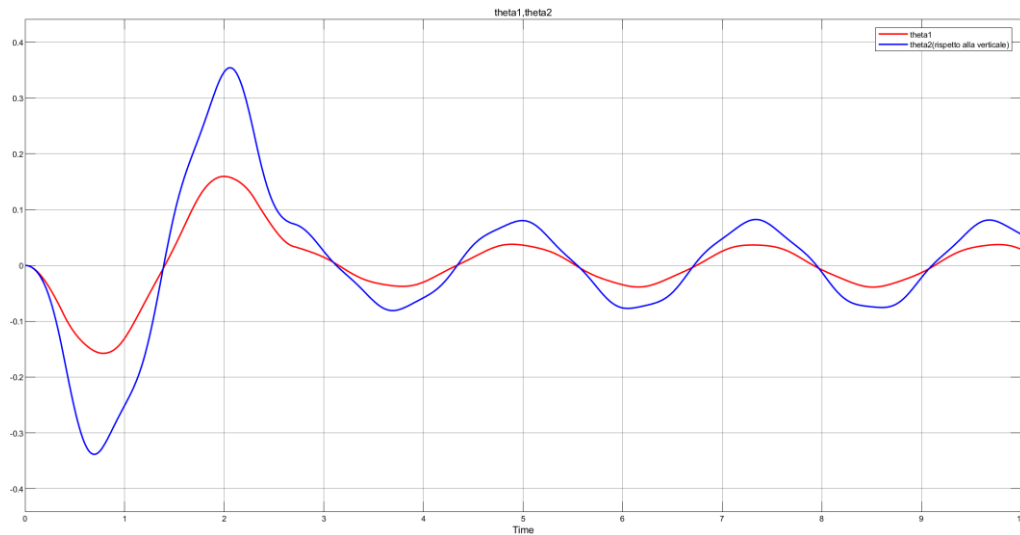


Figura 33 Andamento oscillazioni (errori di modellazione)

Le oscillazioni residue, nel caso di un doppio pendolo non controllato sono presenti e portano il sistema ad oscillare in maniera ben visibile attorno alla posizione di equilibrio pari a 1.5 m.

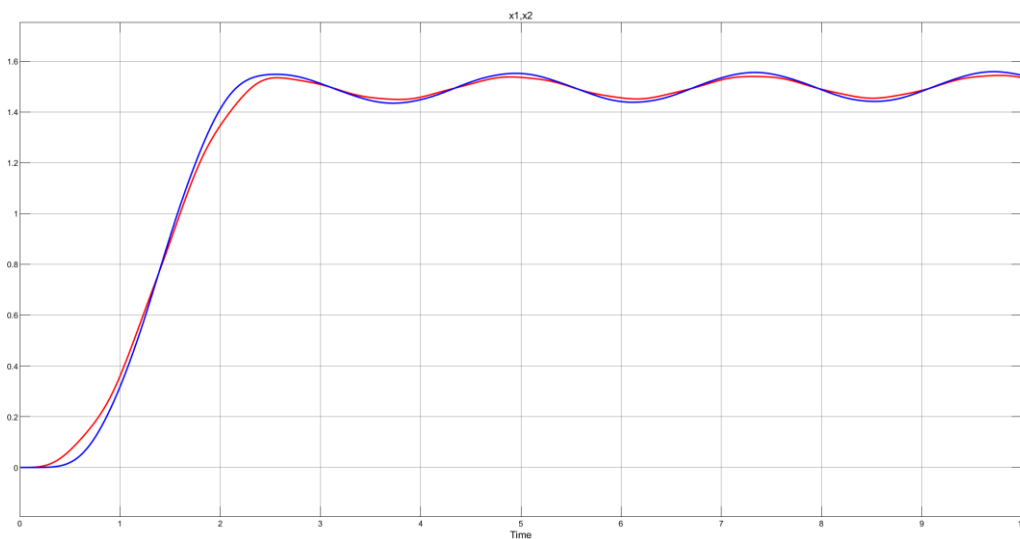


Figura 34 Spostamento errori di modellazione

Dai seguenti risultati si può capire l'importanza di inserire un controllo che consenta di rimediare a errori dovuti a valutazioni incorrette dei parametri presenti all'interno del sistema.

2.5 Simscape doppio pendolo

Si introduce, come per il pendolo semplice, il modello simulativo implementato in Simscape. Questa rappresentazione consentirà una comprensione più approfondita del comportamento del sistema complessivo soggetto alla forzante.

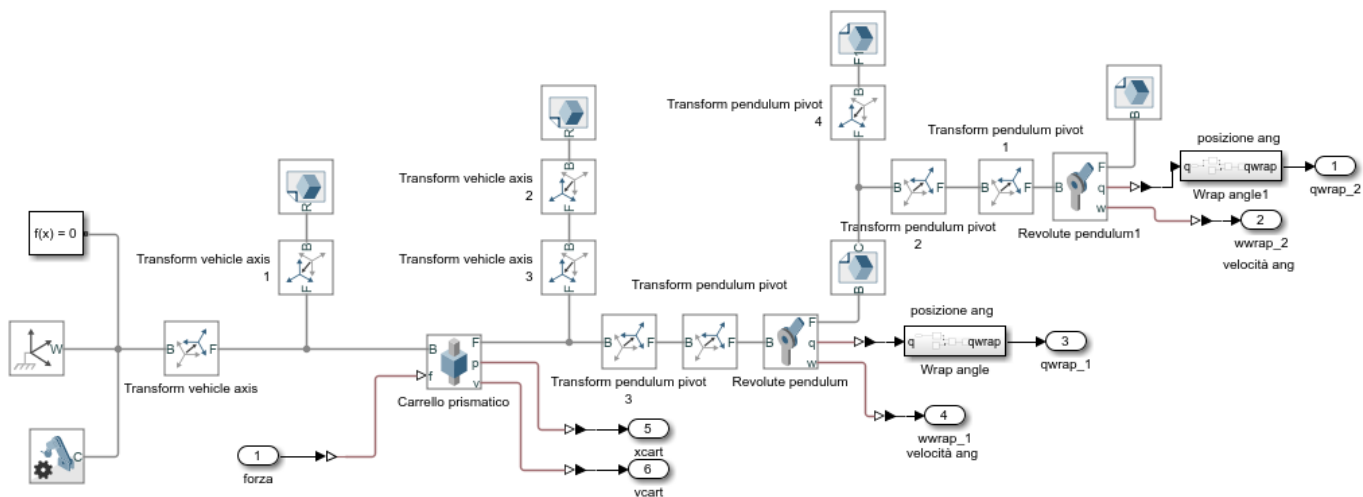


Figura 35 Simscape doppio pendolo

Si è implementata la legge di accelerazione con vincolo su energia residua $<10^{-29}$, progettata specificamente per minimizzare le oscillazioni residue al termine dell'azionamento.

Si può vedere che la legge è a 7 step ed è stata inserita come file .mat nel blocchetto di Simulink e collegata come input al sistema. Anche per il doppio pendolo sono presenti altre tipologie di forzanti (es. l'accelerazione a tre step).

Inoltre è possibile implementare una legge di accelerazione con il seguente andamento.



Figura 36 Legge di moto trapezoidale smooth

La legge che minimizza le accelerazioni rimane comunque quella calcolata dallo script Matlab, con i vincoli progettuali sull'energia residua e sul valore della velocità finale.

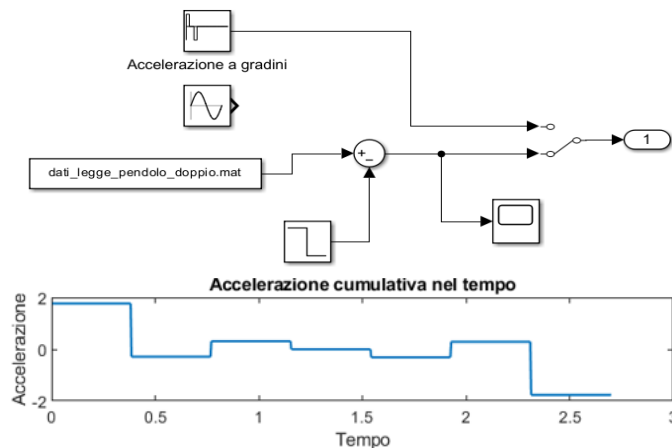


Figura 37 Forzante input

Si riporta in seguito, una volta effettuata la simulazione, il grafico dello spostamento e della velocità del carrello una volta implementata la legge di accelerazione che minimizza le oscillazioni.

Si può vedere che lo spostamento del carrello (tratto rosso) non rimane nell'intorno dello spostamento richiesto ma ha un andamento decrescente e inoltre sono presenti oscillazioni non trascurabili. Le oscillazioni del sistema derivano dalla presenza delle inerzie rotazionali dei due pendoli che giocano un ruolo significativo nell'opporre resistenza al movimento lineare del carrello.

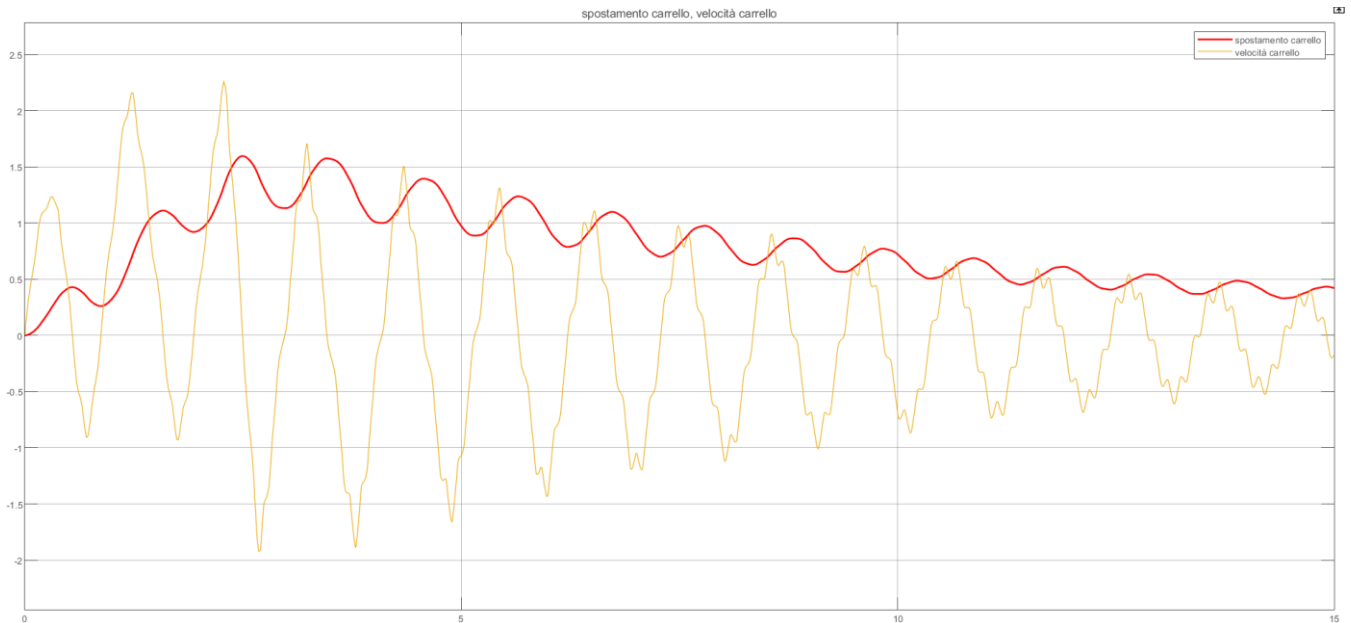


Figura 38 Spostamento e velocità carrello

È stato necessario, quindi, non solo per aumentare la robustezza del sistema ma per far sì che lo spostamento finale risulti quello richiesto, inserire un controllore proporzionale, integrale e derivativo (PID). Il controllore ha permesso, inoltre, di avere il valore di velocità finale pari a zero alla fine dell'azionamento.

Le costanti K_p , K_i e K_d sono state calcolate utilizzando il metodo manuale di Ziegler-Nichols.

Si riportano i calcoli effettuati in Excel:

Control Type	K_p	T_i	T_d	K_i	K_d
P	$0.5K_u$	–	–	–	–
PI	$0.45K_u$	$0.83T_u$	–	$0.54K_u/T_u$	–
PD	$0.8K_u$	–	$0.125T_u$	–	$0.10K_uT_u$
classic PID^[2]	$0.6K_u$	$0.5T_u$	$0.125T_u$	$1.2K_u/T_u$	$0.075K_uT_u$

Figura 411 Risultati in Excel

CALCOLI ZIEGLER-NICHOLS PER PID PENDOLO DOPPIO CON CARRELLO						
Periodo Tu	5					
Ku	2					
	Kp	Ti	Td	Ki	Kd	
P	1					
PI	0,9	4,15		0,216		
PD	1,6		0,625		4,75	
classic PID	1,2	2,5	0,625	0,48	0,75	

Figura 42 Risultati in Excel

Si è implementato nel modello a blocchi di Simulink il controllore PID con i valori calcolati dal metodo di Ziegler-Nichols. È importante sottolineare che il controllo agirà soltanto una volta terminato l'azionamento, quindi dopo il tempo finale di 2.7s. Per fare ciò si sono inseriti i valori di Kp, Ki e Kd come segnali esterni attraverso i blocchi "step".

Si poteva implementare, in alternativa, un segnale di clock con un blocco "delay $\geq 2.7s$ " e successivamente collegare i valori delle costanti come blocchi "constant".

Si è provato ad inserire un controllore direttamente sull'angolo del primo pendolo ma con scarsi risultati in termini di oscillazioni finali. Si riporta lo schema a blocchi del sistema con il controllo:

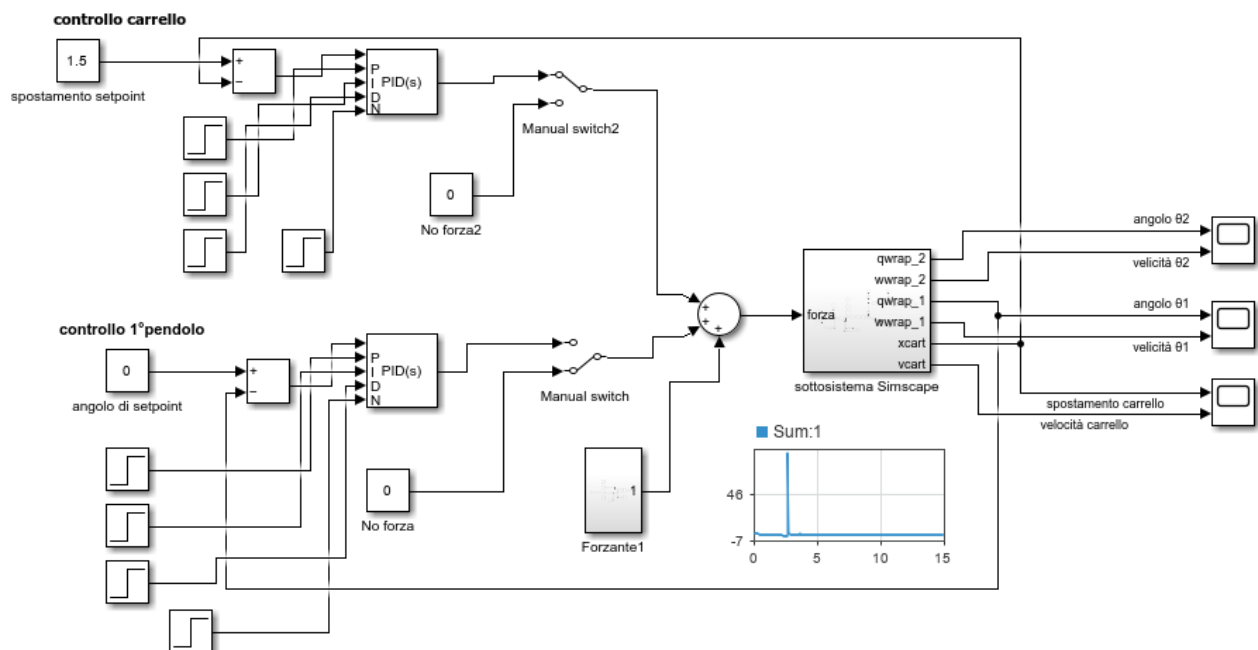


Figura 43 Modello Simulink generale con controllo

Si riporta quindi il grafico dello spostamento e della velocità del carrello dopo che sul sistema agisce il controllore che si è implementato:

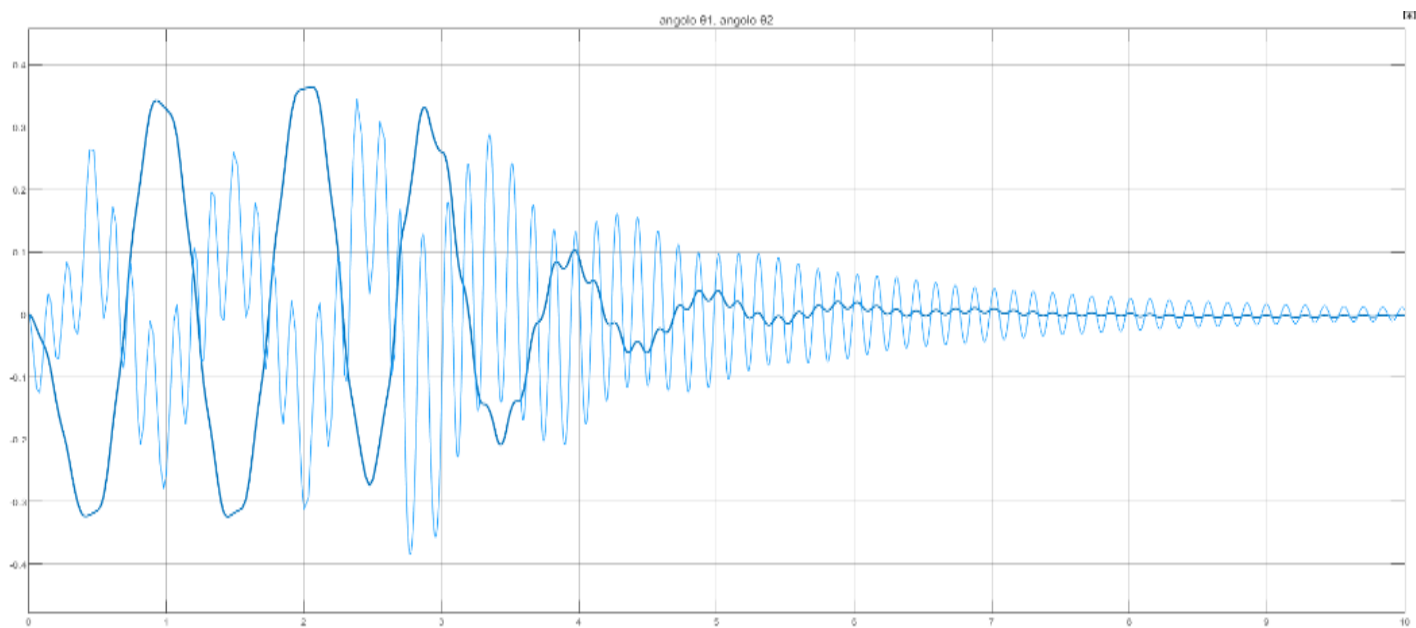


Figura 44 Oscillazioni dei pendoli

Per completezza, è riportato il grafico degli angoli dei pendoli 1 e 2, dove si vede che anche le oscillazioni dei pendoli vengono smorzate e tendono a zero grazie al controllo sul carrello.

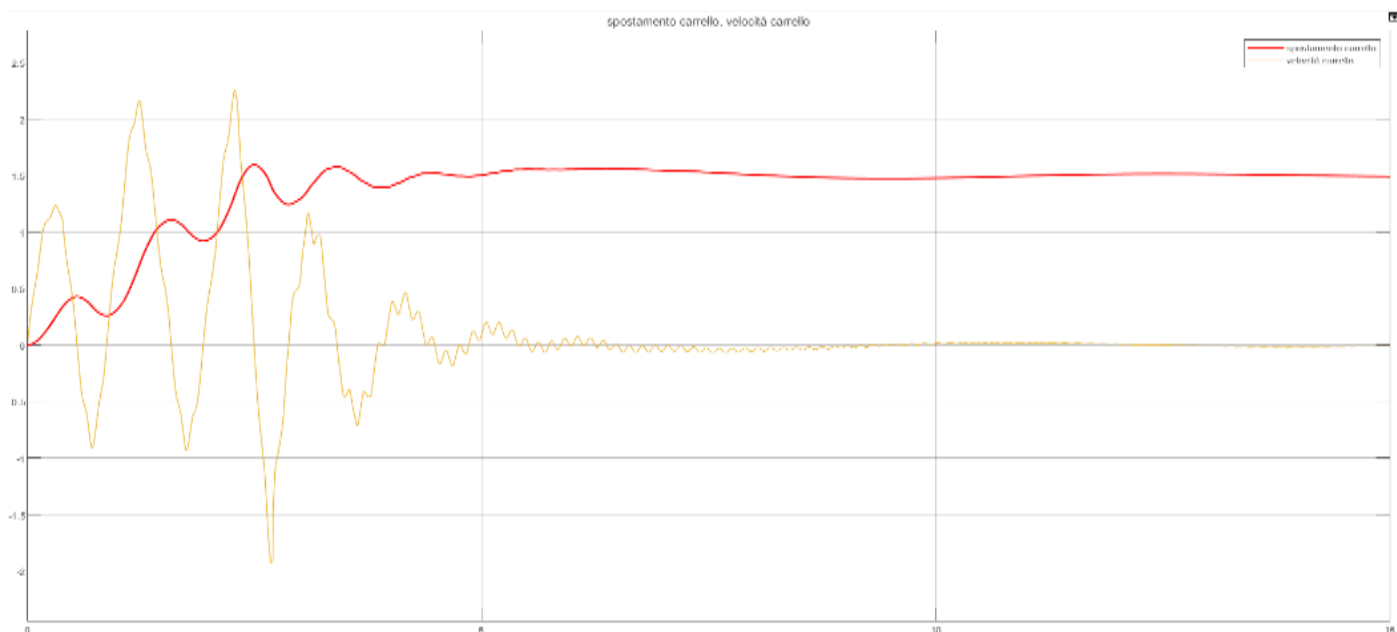
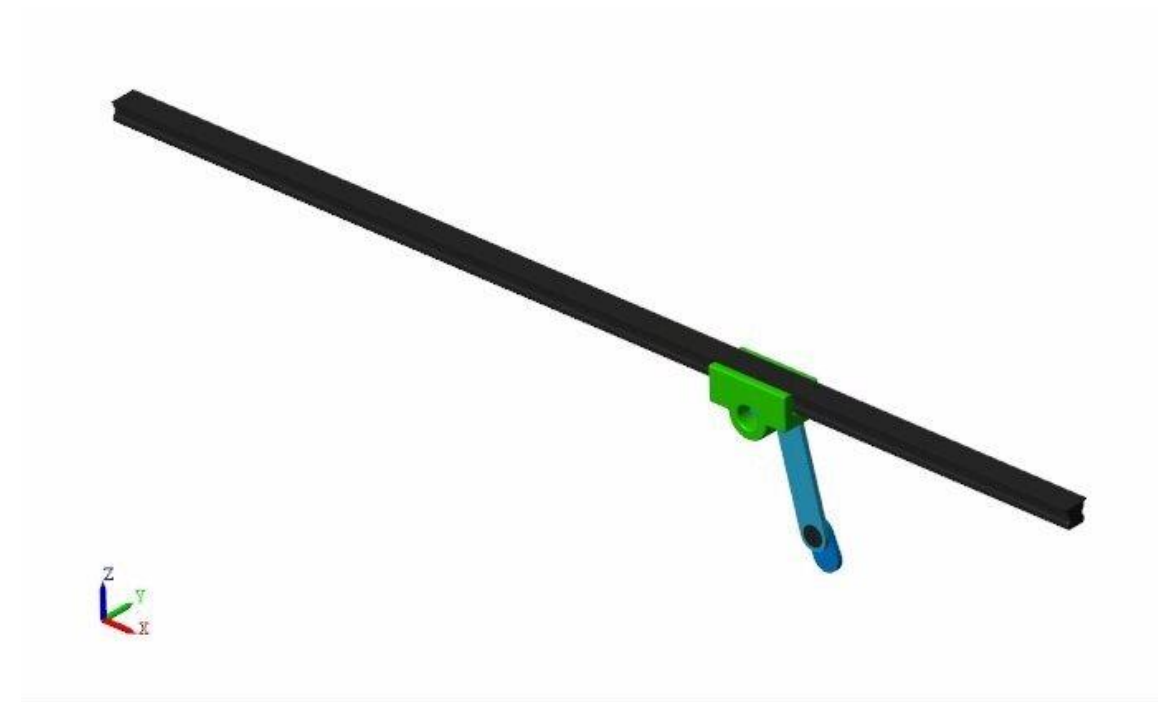


Figura 45 Spostamento e velocità carrello

Si precisa che il grafico degli angoli è stato ottenuto imponendo una massa di 0.3Kg al carrello.

Infine, si riporta la visualizzazione dell'ambiente Multibody di Matlab "Mechanics Explorers", che illustra il movimento del pendolo nel corso del tempo simulato:



Vengono riportati le immagini di due istanti di tempo significativi:

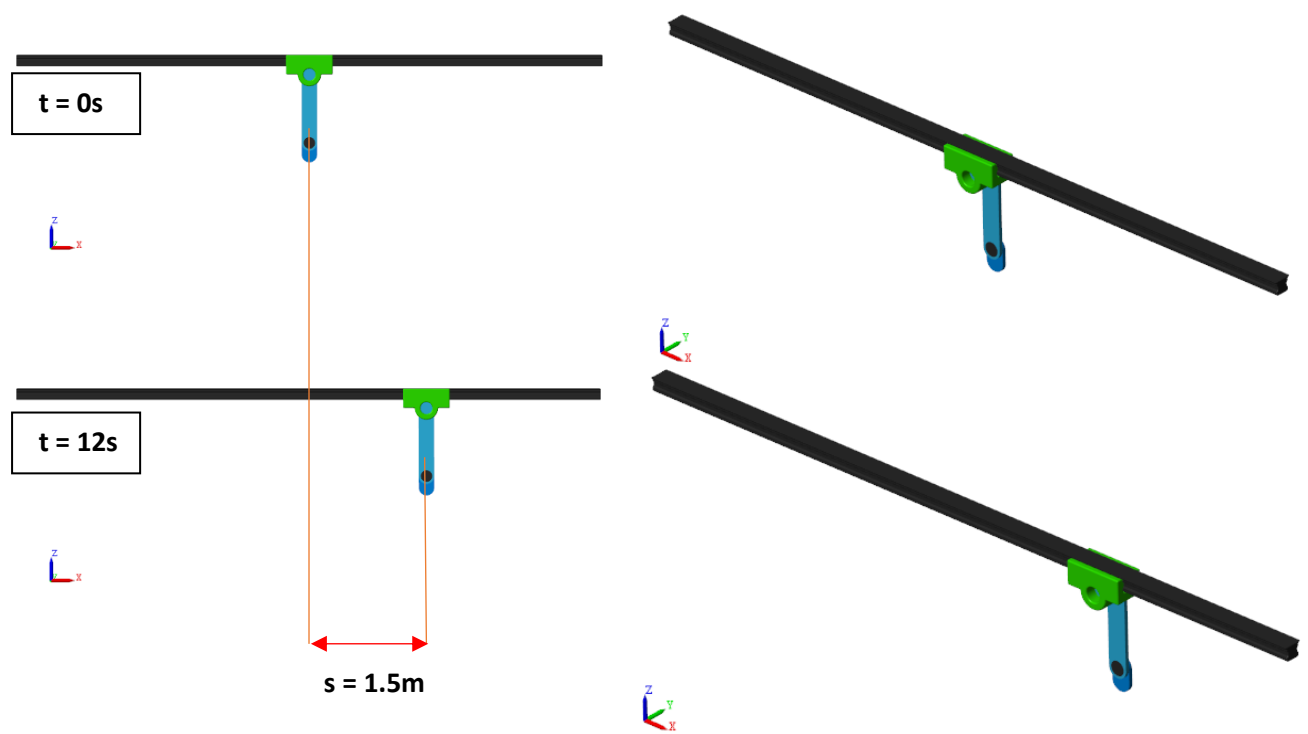


Figura 46 Sistema con doppio pendolo in istanti significativi

Di seguito viene riportato lo schema dei collegamenti generato tramite Simulink:

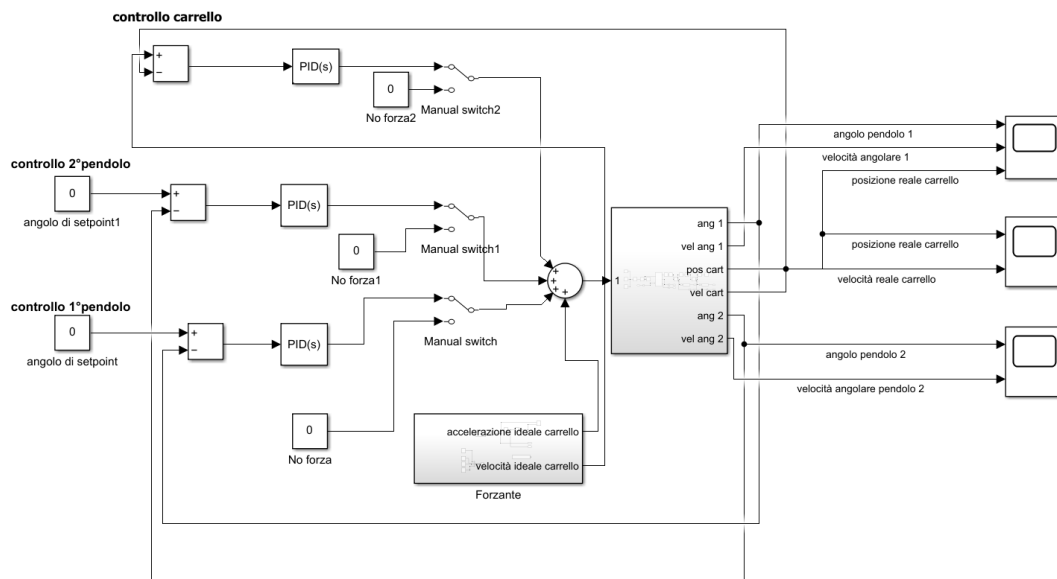


Figura 48 Schema modello Simulink generale doppio pendolo inverso con controllo

Il sistema è mosso dalla seguente legge del moto, che consente al carrello di raggiungere la posizione target di 1.5 metri dopo un tempo pari a 2.7 secondi:

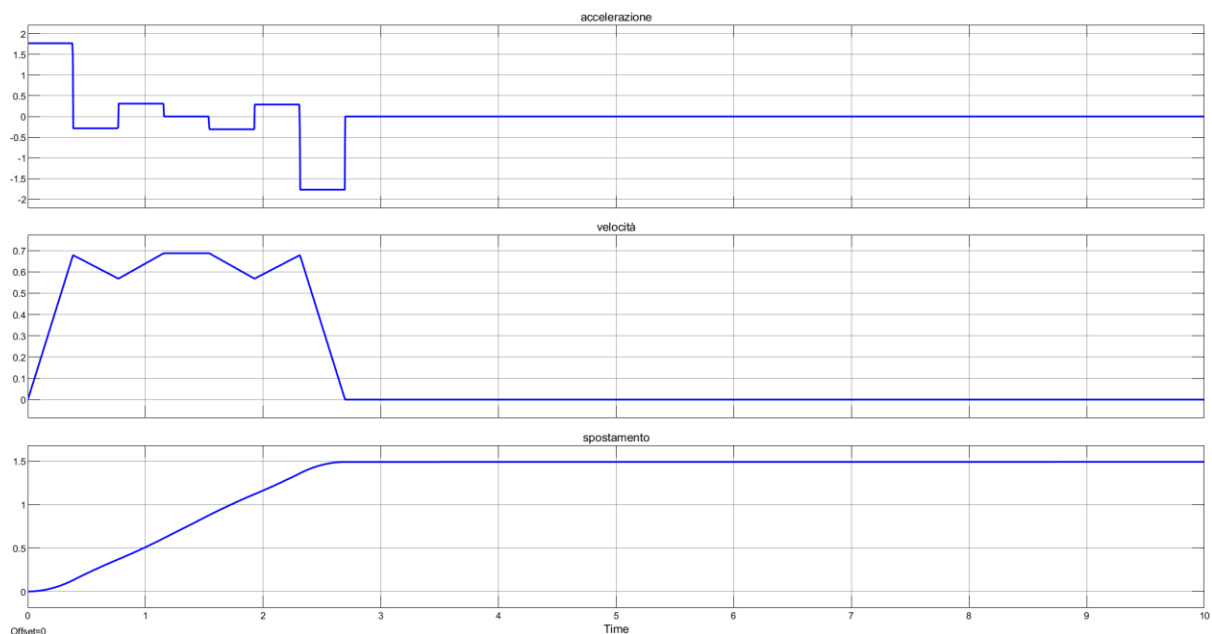


Figura 49 Legge di moto applicata al carrello

I risultati ottenuti verranno mostrati per un tempo di simulazione pari a 20 secondi. Il controllo del pendolo, infatti, richiede un tempo decisamente lungo per funzionare nella maniera corretta, per via della natura instabile del doppio pendolo inverso. Il tempo di stabilizzazione del sistema è di circa 10 secondi.

Nel grafico sottostante possiamo vedere i valori di $\theta_1, \theta_2, \ddot{\theta}_1, \ddot{\theta}_2$.

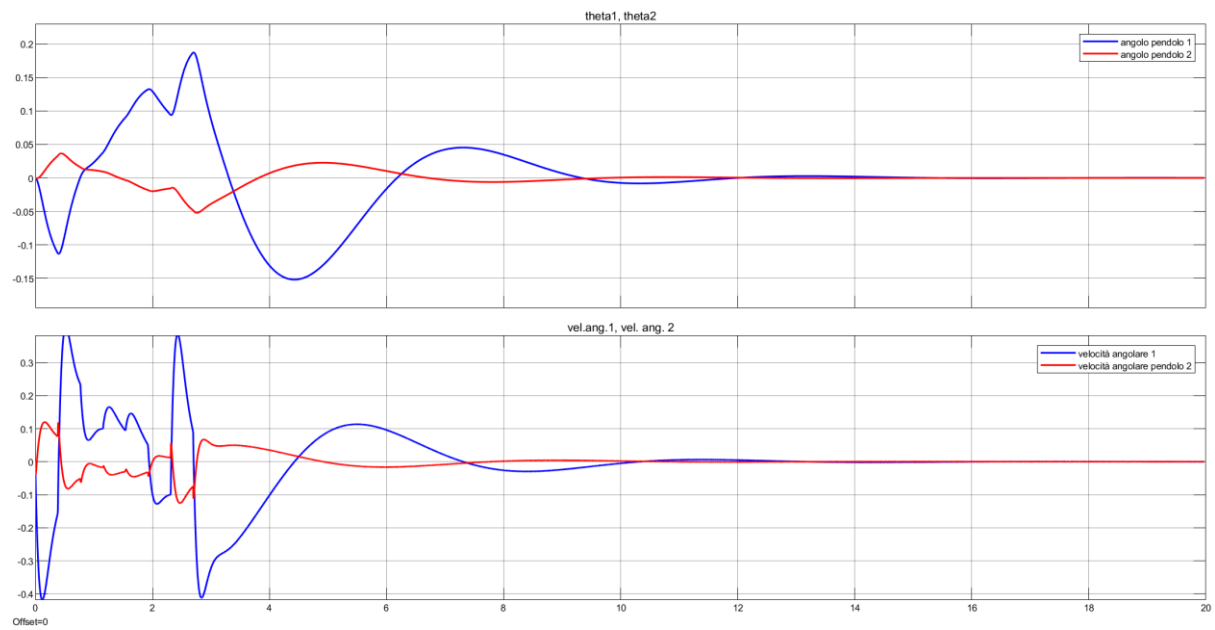


Figura 50 spostamenti e velocità angolari

Di seguito i risultati ottenuti relativi alla posizione e alla velocità del carrello. Possiamo vedere come il carrello si fermi effettivamente in un intorno di 1.5 metri, passato un intervallo di tempo pari a circa 10 secondi.

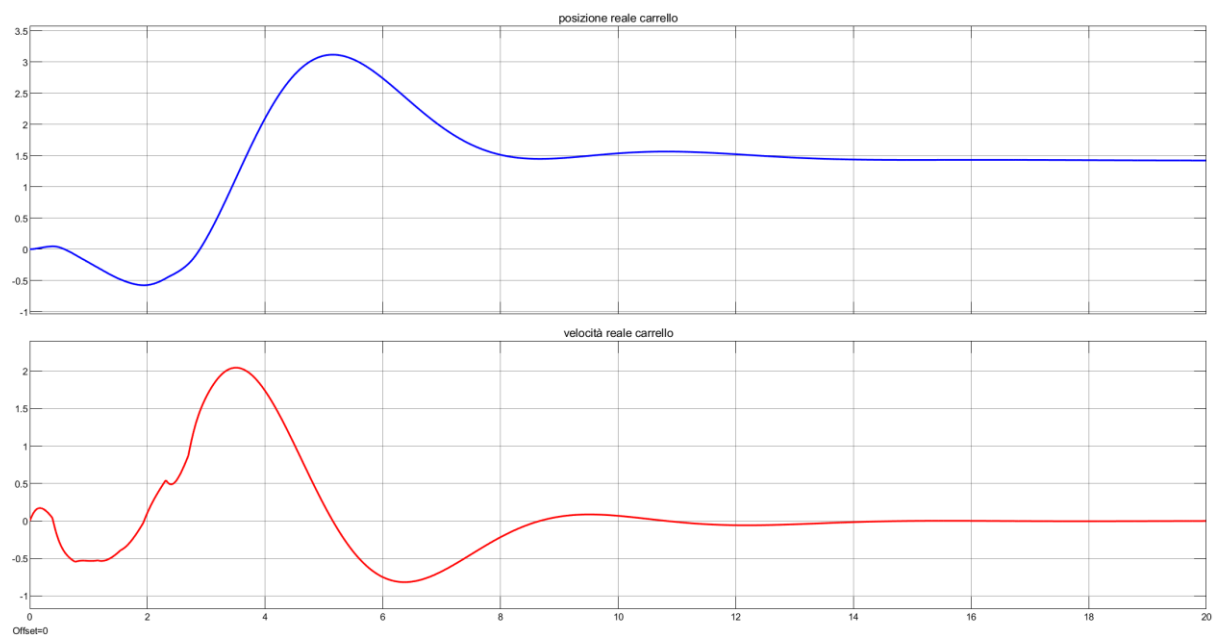
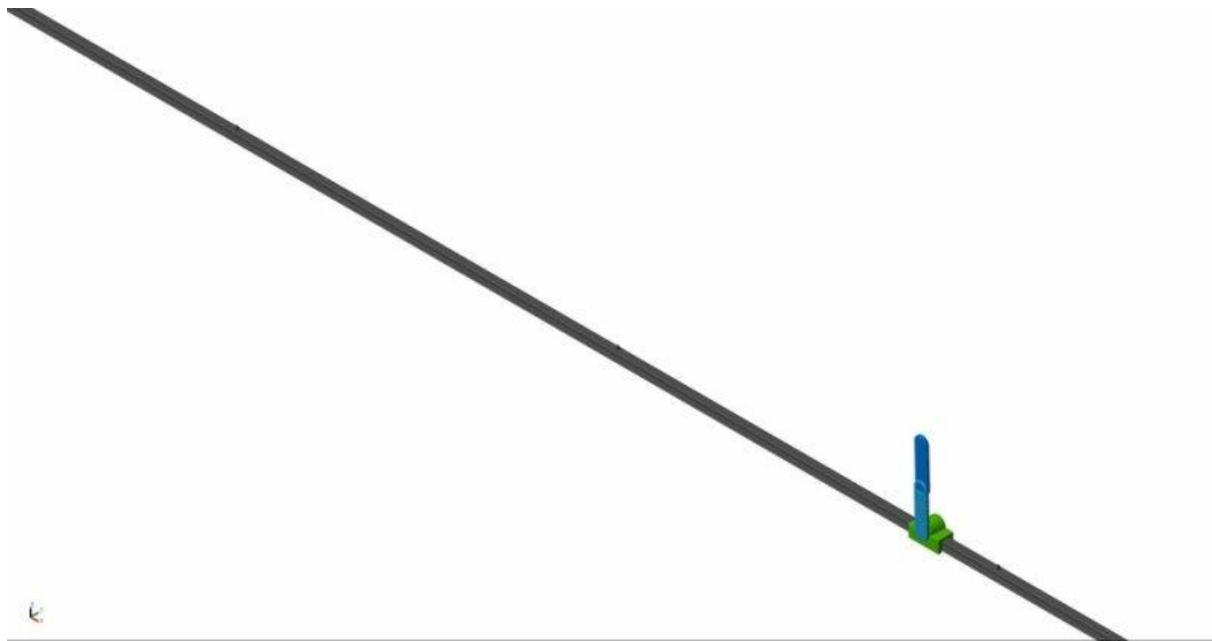


Figura 51 spostamento e velocità carrello

Di seguito possiamo osservare il moto del doppio pendolo simulato attraverso “Mechanics Explorer”:



I risultati ottenuti non controllano perfettamente il sistema ma sono un buon risultato per proseguire con un'analisi più approfondita.

3.2 Pendolo di Furuta

L'analisi e la modellazione del pendolo di Furuta è stata eseguita in conformità con la lezione del Professor Luca Consolini, in cui un pendolo di Furuta reale veniva controllato tramite Arduino e Simulink. Il pendolo reale è stato riprodotto su Simulink in ambiente Simscape con blocchi 'File Solid'. Il sistema in questione presenta due gradi di libertà rotazionali, rappresentati in Simscape da due Revolute Joint, in corrispondenza del collegamento base-testa e testa-pendolo. Infatti, il pendolo può ruotare rispetto alla testa e questa rispetto all'asse della base. Di seguito il modello Simscape:

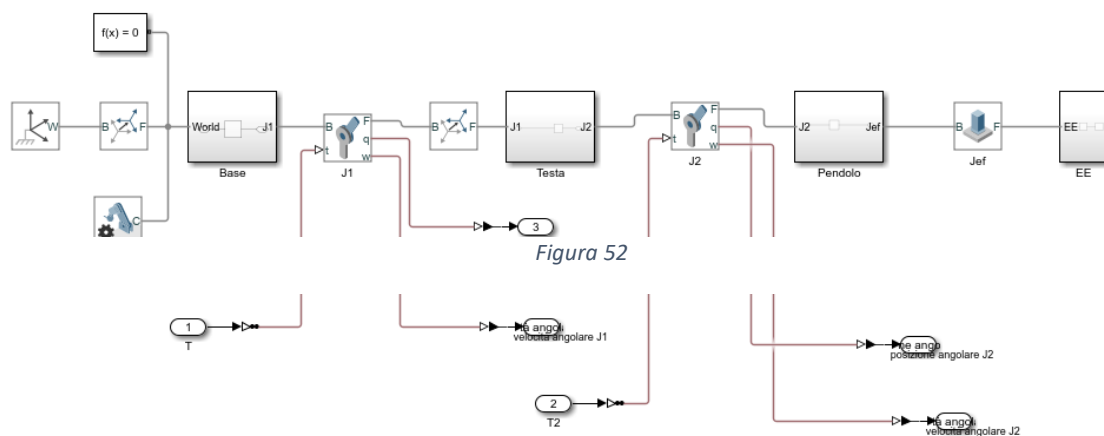


Figura 52

I sottosistemi Base, Testa, Pendolo ed End Effector contengono i blocchi Solid, importati da CAD 3D (Solidworks).

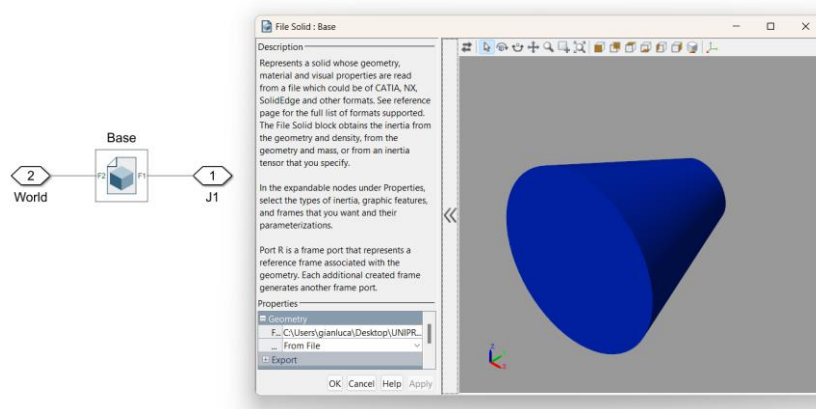


Figura 53 Simscape pendolo di Furuta

L'obiettivo è quello di controllare il pendolo durante il ritorno nella sua posizione di equilibrio stabile. Per farlo, abbiamo implementato su Simulink un modello a blocchi con un controllo in retroazione di tipo P.I.D., riportato di seguito.

Controllo sulla velocità angolare del J1

In primo approccio, il controllo è stato effettuato sulla velocità angolare in uscita dal giunto base-testa, come nel modello riportato di seguito, sfruttando l'auto tuning di Simulink, già integrato nel blocco PID.

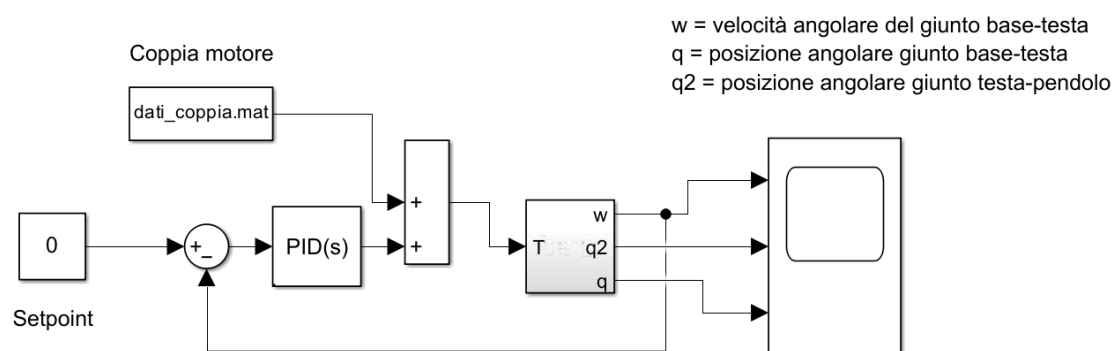


Figura 54 schema Simulink controllo velocità angolare del giuntonto J1

Tramite un file Matlab è stata creata una curva di coppia tipica di un motore passo, così che rappresentasse la forzante da applicare al sistema. Per fare in modo che il pendolo rimanga nella posizione finale la curva di coppia implementata come forzante al J1 è la seguente:

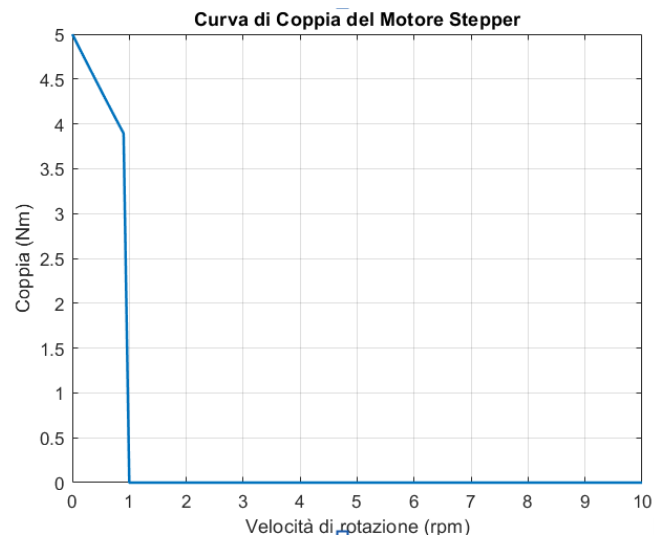


Figura 55 Curva di coppia motore elettrico

Il set point della velocità angolare è stato posto a 0. Di seguito si riporta un video che mostra il comportamento del sistema con coppia massima di 10 Nm e i relativi grafici ottenuti in merito a posizione e velocità dei due revolute joint.

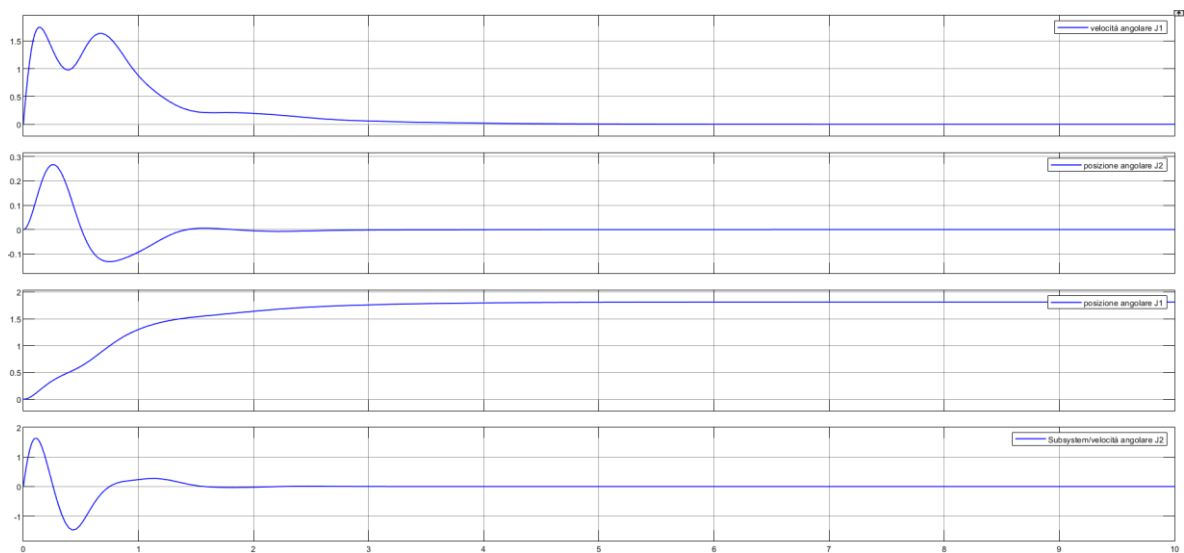
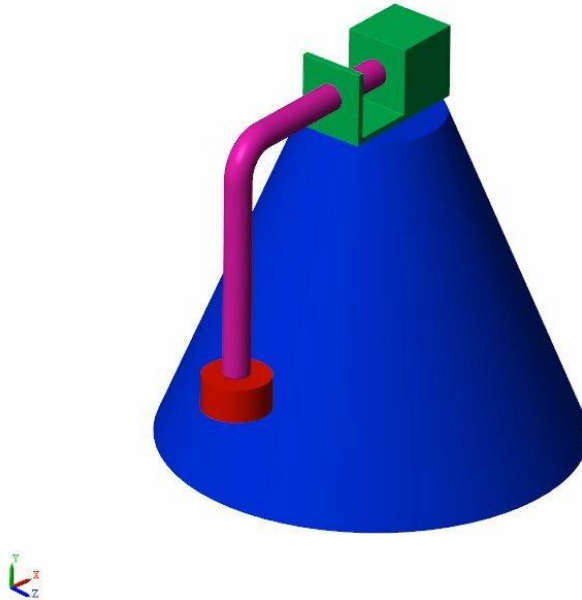


Figura 56 Posizione e velocità di J1 e J2

Si osserva che la velocità angolare raggiunge il valore di set point in circa 2 secondi dal momento in cui cessa la forzante. Di seguito si riporta la simulazione multibody attraverso “Mechanics Explorer” del comportamento del sistema in riferimento allo schema riportato sopra.



Doppio controllo sulle velocità

Con lo schema Simulink riportato di seguito, abbiamo implementato un doppio controllo P.I.D. su una velocità. Per testare la robustezza del sistema abbiamo dato in input al pendolo una forzante casuale, che vuole rappresentare ad esempio del vento random non stazionario.

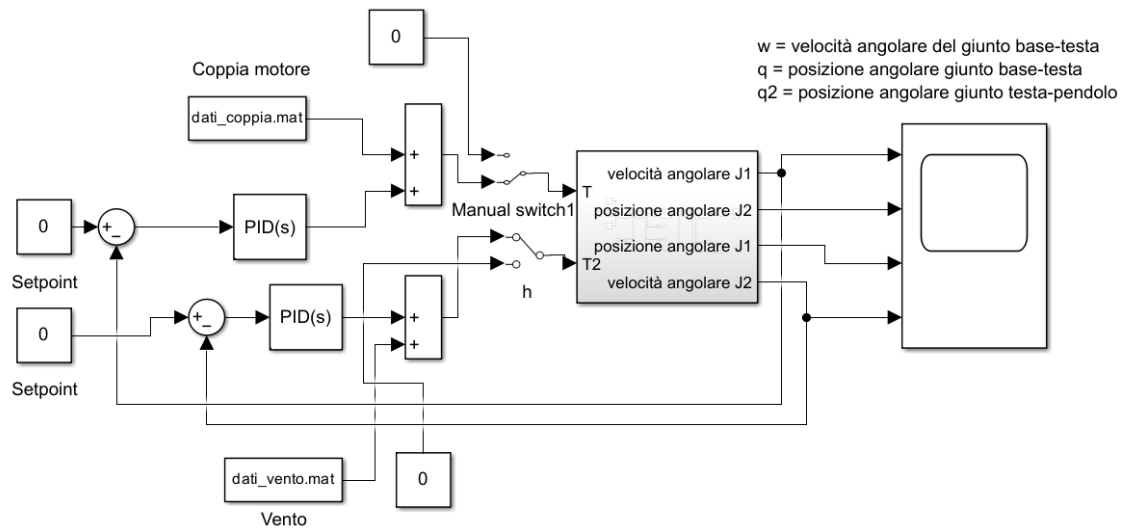


Figura 58 Modello Simulink Pendolo di Furuta con doppio controllo

Una volta conclusa l'analisi del caso con controllo in velocità del motore (J1) abbiamo voluto testare il modello in caso di applicazione di disturbi esterni. Per farlo, abbiamo ipotizzato il caso in cui il pendolo fosse eccitato da un segnale random, con ampiezza compresa tra 0 ed 1, che vuole rappresentare del vento casuale, per tutta la durata della simulazione. Con lo schema riportato sopra abbiamo implementato il caso in questione e abbiamo ottenuto come risultato il pendolo che vibra intorno alla posizione di equilibrio, con ampiezza delle vibrazioni di tre ordini di grandezza in meno. Di seguito i grafici ottenuti allo Scope:



Figura 59 Grafici ottenuti con coppia Figura 60 Grafici ottenuti con coppia costante dopo un certo intervallo

In questa analisi abbiamo applicato una forzante che raggiunge un valore e poi rimane costante, con vincolo di rotazione a -180 gradi per il J1. L'altro approccio è quello di applicare la coppia già usata per il singolo controllo e di seguito si riportano i rispettivi risultati.

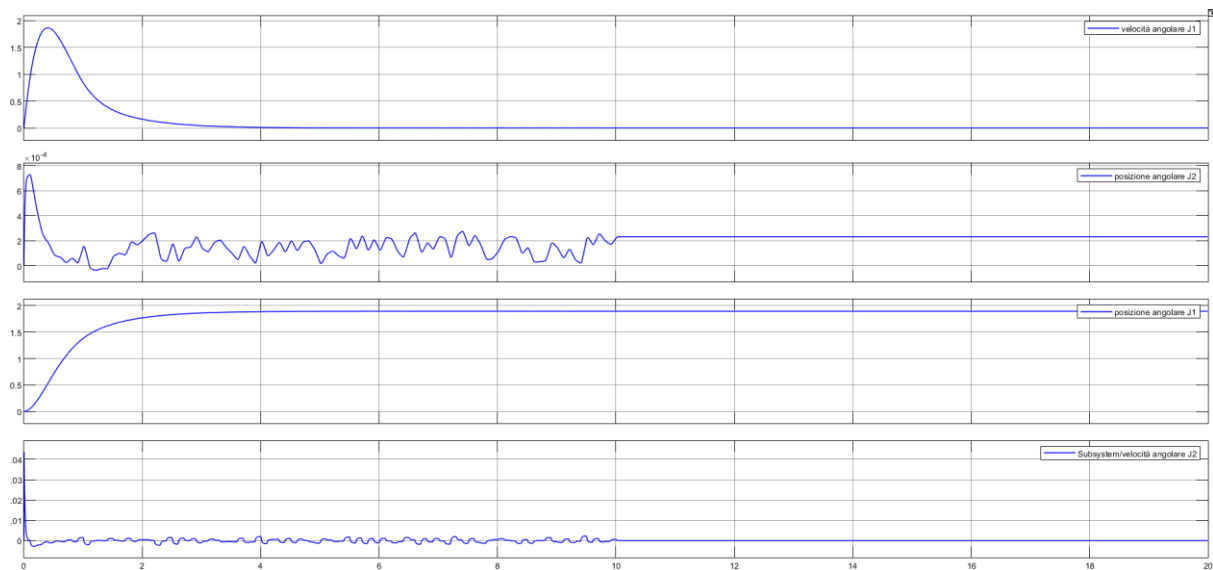
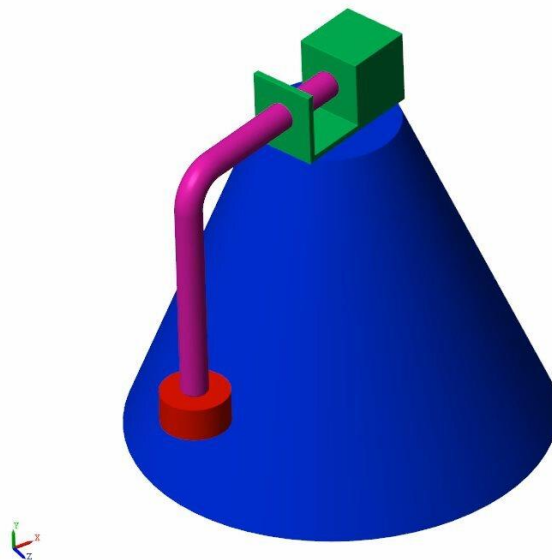


Figura 61 Grafici ottenuti con coppia nulla dopo intervallo di 10 secondi

L'andamento di velocità e posizione del J1 è particolarmente pulito ed esente da fenomeni vibratorii osservabili, mentre per quanto riguarda il pendolo sono presenti vibrazioni che cessano all'istante $t_f=10s$, perché a t_f cessa l'applicazione del disturbo casuale.

Il modello è efficace perché in presenza di vento casuale mantiene il pendolo con vibrazioni di quattro ordini di grandezza inferiori rispetto allo spostamento compiuto; dunque, presenta una significativa robustezza ai disturbi casuali esterni.

Si riporta la simulazione multibody dell'ultimo caso affrontato, con doppio controllo.



Il doppio controllo può essere ulteriormente sviluppato ed approfondito variando i disturbi in funzione di un'applicazione di interesse e con una certa ampiezza delle vibrazioni accettabile.

4. APPENDICE

Si riportano di seguito gli script matlab e le function utilizzate per ottenere i risultati proposti nei diversi capitoli del report tecnico:

4.1 Pendolo semplice accelerazione ottimizzata

```
clc
close all;
clear;

format long
s=1.5; %spostamento
t_f=2.5; %tempo di azionamento

%parametri fisici pendolo
L=0.7; %lunghezza
g=9.81; %gravità
cost = 1;
W=sqrt(g/L)*cost; %pulsazione propria
T=2*pi/W; %periodo

%altri parametri
n=1; %numero di steps iniziale
q=1; %modi di vibrare
m=2; %numero di vincoli cinematici

% matrice massa e rigidezza
FF=-1/L; M1=1; K1=g/L; V=1;

%% ciclo che si ripete se i vincoli della riga while non sono soddisfatti
E_res=1; rs_f=0; rs_max=ones(1,q);
while E_res>10^(-32) || (round(rs_f)~=round(rs_max)) || (abs(v_f)>10^(-15))
n=n+1;
Dt=t_f/n;
for i=1:n
x(i,1)=(FF/K1)*(-cos(W*(t_f-(i-1)*Dt))+cos(W*(t_f-i*Dt)));
x(i,2)=(FF/K1)*(W*sin(W*(t_f-(i-1)*Dt))-W*sin(W*(t_f-i*Dt)));
g(i,1)=Dt;
g(i,2)=(Dt^2/2+(n-i)*Dt^2);
end
E_r=0;
k=1;
for i=1:n
for z=(i+1):n
H(z,i)=W^2*x(i,k)*x(z,k)+(x(i,k+1)*x(z,k+1));
H(i,z)=H(z,i);
end
H(i,i)=W^2*x(i,k)^2+(x(i,k+1)^2);
end
H=2*H;
E_r=H+E_r;
H1=[E_r,g]; % riga 1 matrice [H U]
H2=[g',zeros(m)]; % riga 2 matrice[U^T 0]
C=[H1;H2];
TN=zeros(n+m,1);
TN(n+m)=s;
h=pinv(C)*TN;
v_tf=0;
x_tf=0;
for i=1:n
v_tf=v_tf+h(i)*Dt;
x_tf=x_tf+h(i)*(Dt^2/2+(n-i)*Dt^2);
end
rs_f=x_tf;
v_f=v_tf;
rs_max=s;
while abs(round(rs_f))<10000
rs_f=rs_f*10;
rs_max=rs_max*10;
end
wn=W;
x_2=(h(1)*(FF/K1))*(1-cos(wn*Dt));
xv_2=h(1)*(FF/K1)*(wn*sin(wn*Dt));
for i=1:n
x=(h(i+1)-h(i))*(FF/K1)*(1-cos(wn*(Dt-i*Dt)));
xv=(h(i+1)-h(i))*(FF/K1)*(wn*sin(wn*(Dt-i*Dt)));
x_2=x_2+x;
xv_2=xv_2+xv;
end
E_res=0.5*((xv_2^2)*M1+K1*(x_2^2));
```

```

end

%grafici
%grafici
[aa,v,s,t]= Legge_moto(t_f,h,n);
[ta,x_1]= Ampiezza_oscillazioni(t_f,n,wn,h,FF,K1);
%%
figure ('Name', 'Legge di moto carrello')
subplot(3,1,1);
plot(t,aa,LineWidth=1.3);
ylabel('accelerazione [m/s^2]')
xlabel('t [s]')
subplot(3,1,2);
plot(t,v,LineWidth=1.3);
ylabel('velocità [m/s]')
xlabel('t [s]')
subplot(3,1,3);
plot(t,s,LineWidth=1.3);
ylabel('spostamento [m]')
xlabel('t [s]')
figure ('Name', 'Ampiezza oscillazioni')
plot(ta,x_1,LineWidth=1.3);
ylabel('ampiezza [rad]')
xlabel('t [s]')

```

4.2 Function Legge_moto

```

function[a,v,s,t]=motion_law(t_f,h,n)

cost = 1000;

t_step=t_f/(cost*n);
for j=1:cost
t(j)=t_step*j;
a(j)=h(1);
v(j)=a(j)*t(j);
s(j)=(1/2)*a(j)*t(j)^2;
end

for i=2:n
for k=cost*(i-1):cost*i
t(k)=t_step*k;
a(k)=h(i);
v(k)=v(cost*(i-1))+a(k)*(t(k)-t(cost*(i-1)));
s(k)=s(cost*(i-1))+v(cost*(i-1))*(t(k)-t(cost*(i-1)))+(1/2)*a(k)*(t(k)-t(cost*(i-1)))^2;
end
end

```

5.3 Function Ampiezza_oscillazioni

```

function[tempo,teta]=angular_displacement(t_f,n,wn,h,FF,K1)
cost = 1000;
t_step=t_f/(cost*n);

for j=1:(cost*n)
ta(j)=t_step*j;
x_1(j)=(h(1)*(FF/K1))*(1-cos(wn*ta(j)));
end

ts(1)=0;

for i=2:n
ts(i)=ta(cost*(i-1));
for kk=cost*(i-1):(cost*n)
ta(kk)=t_step*kk;
x_1(kk)=x_1(kk)+(h(i)-h(i-1))*(FF/K1)*(1-cos(wn*(ta(kk)-ts(i))));
end
end

for p=1:200
tf(p)=t_f+t_step*p;
x_f(p)=h(1)*(FF/K1)*(1-cos(wn*(t_step*p+t_f)));
for i=1:n
x(p)=(h(i+1)-h(i))*(FF/K1)*(1-cos(wn*(t_step*p+t_f)));
x_f(p)=x_f(p)+x(p);
end
end

teta=horzcat(x_1,x_f);
tempo=horzcat(ta,tf);

```

4.4 Pendolo doppio accelerazione ottimizzata

```

clc
clear all
close all

m1=0.2; %kg
m2=0.5; %kg

l1=1.05;
l2=0.35;

g=9.81;
t_f=2.7;
s=1.5;

n=1;
q=2;
mo=q+1;
m=2;
M=[(m1+m2)*l1^2, m2*l1*l2; m2*l1*l2, m2*l2^2];
K=[(m1+m2)*g*l1, 0; 0, m2*g*l2];
F=[-(m1+m2)*l1; -(m2)*l2];
[V,v]=eig(K,M);
lambda=diag(v);
W=sqrt(abs(lambda));
T=2*pi./W;
M1=V'*M*V;
K1=V'*K*V;
FF=V'*F;
E_res_a=1; E_res_b=1; rs_f=0; rs_max=ones(1,q);
while E_res_a>10^(-29)*E_res_b || (round(rs_f)~=round(rs_max)) || (abs(v_f)>10^(-6))
n=n+1;
Dt=t_f/n;
c=0;
for i=1:n
a=1:2:(2*q)
x(i,a)=(FF(q-c)/K1(q-c,q-c))*(-cos(W(q-c)*(t_f-(i-1)*Dt))+cos(W(q-c)*(t_f-i*Dt))); % eta
x(i,a+1)=(FF(q-c)/K1(q-c,q-c))*(W(q-c)*sin(W(q-c)*(t_f-(i-1)*Dt))-W(q-c)*sin(W(q-c)*(t_f-i*Dt))); % eta punto
c=c+1;
end
g(i,1)=Dt;
g(i,2)=(Dt^2/2+(n-i)*Dt^2);
c=0;
end
j=q;
E_r=0;
for k=1:2:(2*q)
for i=1:n
for z=(i+1):n
H(z,i)=W(j)^2*x(i,k)*x(z,k)+(x(i,k+1)*x(z,k+1));
H(i,z)=H(z,i);
end
H(i,i)=W(j)^2*x(i,k)^2+(x(i,k+1)^2);
end
H=2*H;
E_r=H+E_r;
j=j-1;
end
H1=[E_r,g];
H2=[g',zeros(m)];
C=[H1;H2];
TN=zeros(n+m,1);
TN(n+m)=s;
h=pinv(C)*TN;
v__tf=0;
x__tf=0;
for i=1:n
v__tf=v__tf+h(i)*Dt;
x__tf=x__tf+h(i)*(Dt^2/2+(n-i)*Dt^2);
end
rs_f=x__tf;
v_f=v__tf;
rs_max=s;
while abs(round(rs_f))<10000
rs_f=rs_f*10;
rs_max=rs_max*10;
end
W1=[0;W];
FF1=[0;FF];
KK1=[zeros(1,q+1);0,K1(1,:);0,K1(2,:)];
[tt, x_mod, v_mod, x_mod_tff, v_mod_tff, A1, A2, tf_position] = confr_amplitude (n, mo, V, W1, Dt, h, FF1, KK1,
Dt/10, 1.5*t_f);
[r, ~]=size(x_mod);
E_res=zeros(1,r);

```

```

for j=1:r
E_res(j)=0.5*(v_mod(j,:)*M1*v_mod(j,:)+x_mod(j,:)*K1*x_mod(j,:));
end
E_res_b=max(E_res(1:tf_position));
E_res_a=max(E_res(tf_position:r));
end
% grafici
[aa,v,s,t]=legge_moto(t_f,h,n);

figure
subplot(3,1,1)
plot(t,aa)
title('Accelerazione cumulativa nel tempo')
xlabel('Tempo')
ylabel('Accelerazione')
subplot(3,1,2)
plot(t,v)
title('Velocità cumulativa nel tempo')
xlabel('Tempo')
ylabel('Velocità')
subplot(3,1,3)
plot(t,s)
title('Spostamento cumulativo nel tempo')
xlabel('Tempo')
ylabel('Spostamento')

figure ('Name', 'Ampiezza oscillazioni')
plot(tt,x_mod*V);
ylabel('ampiezza [rad]')
xlabel('t [s]')
xlim([0 3]);

```

4.5 Function Confr_amplitude

```

%*****
% PROGRAMMA "confronto_amplitud.m"
%*****
function [tt, x_mod, v_mod, x_mod_tff, v_mod_tff, A1, A2, tf_position] = confr_amplitude(n, mo, V, W, Dt, h, FF, K1,
time_step, t_plus)
    g = 1;

    for e = 2:mo
        wn = W(e);
        x = 0; x_2 = 0; x_1 = 0; xv_2 = 0;
        h(n + 1) = 0;
        k = zeros(1, n);
        q = 1; z = 1; a = 1;
        t = 0;

        for z = 1:n
            while t <= z * Dt
                x_1 = (h(1) * FF(e) / K1(e, e)) * (1 - cos(wn * t));
                xv_1 = h(1) * (FF(e) / K1(e, e)) * (wn * sin(wn * t));

                for i = 1:n
                    x = k(i) * (h(i + 1) - h(i)) * (FF(e) / K1(e, e)) * (1 - cos(wn * (t - i * Dt)));
                    xv = k(i) * (h(i + 1) - h(i)) * (FF(e) / K1(e, e)) * (wn * sin(wn * (t - i * Dt)));
                end
                x_2 = x_2 + x;
                xv_2 = xv_2 + xv;
            end

            x_mod_t(q, g) = x_2 + x_1;
            v_mod_t(q, g) = xv_2 + xv_1;
            tt(q) = t;

            t = t + time_step;
            q = q + 1;

            clear x_1 xv_1 x xv
            x_2 = 0; xv_2 = 0;
        end

        x = 0;
        k(z) = 1;
    end

    x_2 = 0; xv_2 = 0;

    while t <= t_plus
        x_1 = h(1) * (FF(e) / K1(e, e)) * (1 - cos(wn * t));
        xv_1 = h(1) * (FF(e) / K1(e, e)) * (wn * sin(wn * t));
    end

```

```

for i = 1:n
    x = (h(i + 1) - h(i)) * (FF(e) / K1(e, e)) * (1 - cos(wn * (t - i * Dt)));
    xv = (h(i + 1) - h(i)) * (FF(e) / K1(e, e)) * (wn * sin(wn * (t - i * Dt)));

    x_2 = x_2 + x;
    xv_2 = xv_2 + xv;
end

x_mod_end(a, g) = x_2 + x_1;
v_mod_end(a, g) = xv_2 + xv_1;
ttt(a) = t;

t = t + time_step;
a = a + 1;

clear x_1 xv_1 x xv
x_2 = 0; xv_2 = 0;
end

g = g + 1;
end

tf_position = q - 1;
x_mod_tff = x_mod_t(q - 1, :);
v_mod_tff = v_mod_t(q - 1, :);
A1 = max(x_mod_t);
A2 = max(x_mod_end);
x_mod = [x_mod_t; x_mod_end];
v_mod = [v_mod_t; v_mod_end];
tt = [tt, ttt];
end

```

4.6 Torque

```

clear
close all
clc

Cmax = 10;
S = 4;
a = 0.5;
w0 = 10;
w = 0:100;

% T = -K*sin((pi/2)/S)*theta);
C = Cmax./(1+exp(a*(w/w0)));
C_end = C(1:10);
Plus = C_end(end)*ones((length(C)-length(C_end)),1);
C_tot = [C_end Plus];
w_end = linspace(0,10,length(C_tot));

plot(w_end, C_tot,'LineWidth',1.5);
xlabel('Velocità di rotazione (rpm)');
ylabel('Coppia (Nm)');
title('Curva di Coppia del Motore Stepper');
grid on;

% salvo i parametri di coppia in un file .mat

% Crea un oggetto timeseries per i dati sull'accelerazione
coppia_timeseries = timeseries(C_tot, w_end);

% Salva l'oggetto timeseries in un file MAT con versione 7.3
save('dati_coppia.mat', 'coppia_timeseries', '-v7.3');

vento = rand(101,1);
time = 0:0.1:10;

% salvo i parametri di coppia in un file .mat

% Crea un oggetto timeseries per i dati sull'accelerazione
vento_timeseries = timeseries(vento, time);

% Salva l'oggetto timeseries in un file MAT con versione 7.3
save('dati_vento.mat', 'vento_timeseries', '-v7.3');

```