



Corso di Laurea Magistrale in Ingegneria Meccanica

Corso di Meccatronica

RELAZIONI DI LABORATORIO

Anno Accademico 2022/2023

Sommario

1 Servoattuatore idraulico	5
1.1 Introduzione.....	5
1.2 Modello matematico.....	5
1.2.1 Controllo	6
1.2.2 Servovalvola.....	6
1.2.3 Bypass	7
1.2.4 Attuatore.....	8
1.3 Analisi dei dati sperimentali e del modello	8
1.3.1 Analisi andamenti sperimentali.....	8
1.3.2 Confronto dati sperimentali con il modello	10
1.3.3 Variazione di K_i e K_p in set ad onda quadra	13
1.3.4 Fitting della massa mobile	14
1.3.5 Diagrammi di bode sperimentali e simulati	15
1.3.6 Risposta del modello a rampa e forza esterna fissati	17
1.4 Conclusioni.....	17
1.4.1 Risultati ottenuti.....	18
2 Banco prova ASTIB	19
2.1 Introduzione.....	19
2.2 Analisi prove	20
2.2.1 Prova 1	20
2.2.2 Prova 2	24
2.2.3 Prova 3	28
2.2.4 Prova 4	31
3 ISWEC banco Hardware in the Loop: generatore di energia da moto ondoso.....	35
3.1 Introduzione.....	35
3.2 Struttura ISWEC.....	35
3.3 Struttura banco prova HIL.....	36
3.4 Identificazione I e J	37
3.4.1 Identificazione di I e J minimizzando la potenza	37
3.4.2 Identificazione di I e J minimizzando l'errore sulla posizione.....	37
3.5 Grafici 3D	38
3.5.1 Grafico 3D sulla potenza	38
3.5.2 Grafico 3D sulla coppia	39

3.5.3 Grafico 3D sullo spostamento	39
3.6 Risultati del fitting dei dati sperimentali	39
4 Giunto di forza a 6 assi	42
4.1 Introduzione.....	42
4.1.1 Obiettivi	42
4.1.2 Introduzione teorica.....	42
4.1.3 Materiali e Strumenti	43
4.1.4 Setup sperimentale e Svolgimento metodologico.....	44
4.2 Implementazione LabVIEW.....	46
4.2.1 Lettura dati di zero (SubVI).vi e Waveform To Matrix (SubVI).vi	47
4.2.2 Creazione e inizializzazione del task per la lettura dei segnali di zero	48
4.2.3 Acquisizione dei segnali di zero	49
4.2.4 Build Conversion Matrix (subVI).vi.....	53
4.2.5 Rearrange Signals (SubVI).vi.....	55
4.2.6 Acquisizione dati.vi	56
4.2.7 Tab “Segnali di zero” (Figura 24)	56
4.2.8 Tab “Acquisizione dati” (Figura 25)	57
4.3 Modello matematico del sistema.....	65
4.3.1 Introduzione	65
4.3.2 Modello matematico	65
4.3.3 Programma MATLAB	66
4.3.4 Pre-analisi	67
4.3.5 Fitting parametrico: fminsearch.....	68
4.3.6 Risultati.....	69
4.3.7 Teorema di Nyquist	70
4.4 Modello Simulink.....	73
4.4.1 Creazione del modello	73
4.4.2 Stima dei parametri.....	74
4.4.3 Risultati ed osservazioni	76
4.5 Acquisizione dati di posizione tramite image processing	77
4.5.1 Obiettivi e setup.....	77
4.5.2 Detection & tracking.....	78
4.5.3 Plot e salvataggio	81
4.5.4 Prova sperimentale.....	82

4.6 Fitting sui dati sperimentalni dell'acquisizione immagini	83
4.6.1 Modello teorico.....	83
4.6.2 Script Matlab per il fitting dei dati	83
4.6.3 Analisi del segnale $y(t)$ e commento dei risultati	88
4.6.4 Simulazione di frequenze di acquisizione minori:.....	94
4.6.5 Sviluppi futuri e possibili migliorie.....	97

1 Servoattuatore idraulico

1.1 Introduzione

Lo scopo di questa esercitazione è lo studio di un sistema per il controllo posizione idraulico. Il sistema è costituito da un cilindro idraulico a stelo passante il cui scopo è la movimentazione di un carico esterno secondo una legge di set fissata. Il banco prova è stato progettato in modo da effettuare prove statiche e dinamiche su componenti meccanici, quali sospensioni automobilistiche.

1.2 Modello matematico

Il sistema reale è stato modellato attraverso il software Simulink in modo da individuare il comportamento teorico e poterlo confrontare con quello sperimentale in modo da validare il modello. La validazione del modello è un passaggio molto importante perché, con un modello validato, permette successivamente di eseguire dei test direttamente sul modello sapendo che i risultati possono rispecchiare quelli reali; in questo modo è molto più semplice eseguire delle prove e soprattutto si evitano eventuali problemi che si possono riscontrare sulla macchina che potrebbero anche guastarla.

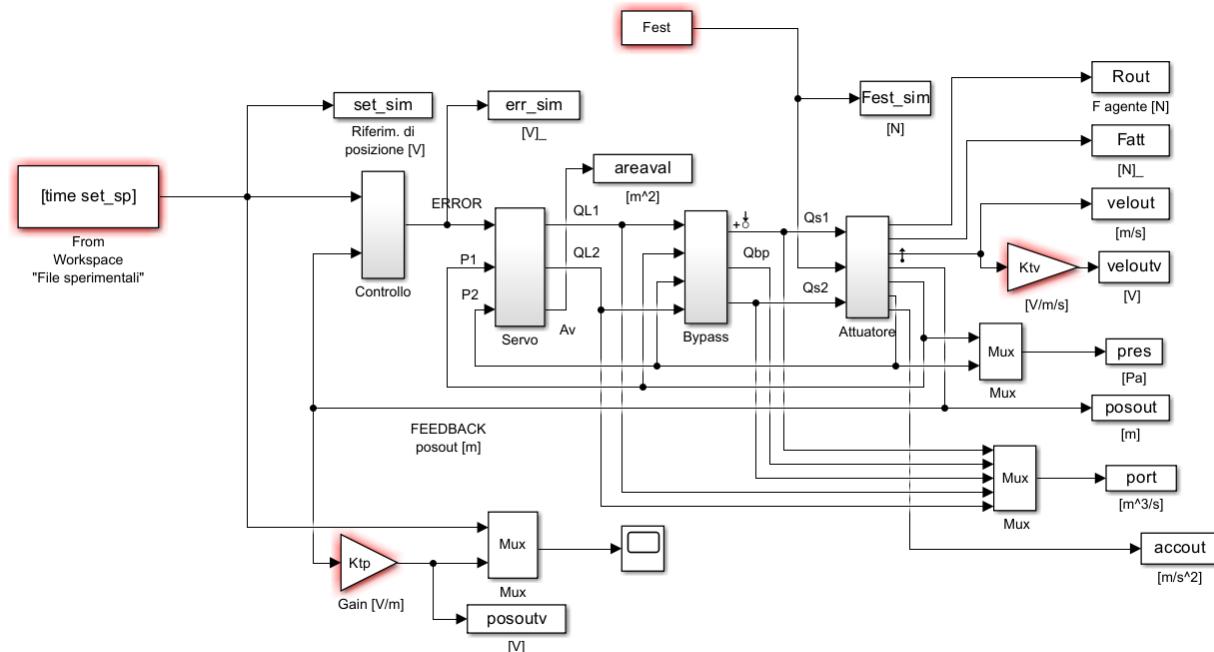


Figura 1 Modello simulink del sistema

Si analizzano ora i sottosistemi presenti nel modello.

1.2.1 Controllo

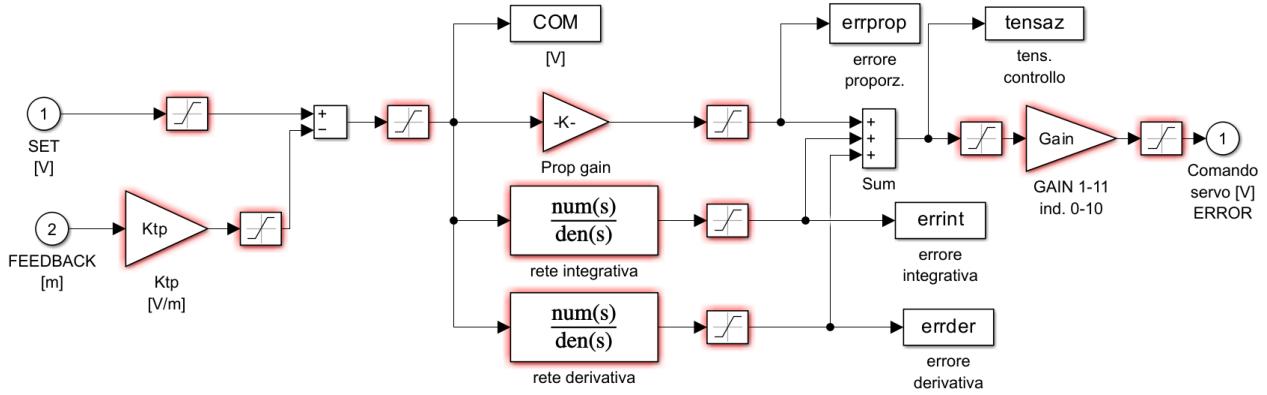


Figura 2 Parte di controllo del modello simulink

In ingresso si ha il segnale di SET in Volt e il segnale di feedback in metri che viene prontamente convertito in Volt attraverso un guadagno di feedback; in questo modo sarà possibile sottrarre le due grandezze e avere un errore espresso in Volt. Prima e dopo il calcolo dell'errore le grandezze vengono saturate tra ± 10 V perché il controllo richiede un segnale entro questi limiti (dato che viene letto nel datasheet del controllore). Una volta calcolato l'errore, esso viene mandato ai blocchi del controllore PID, successivamente i segnali della parte proporzionale, integrativa e derivativa vengono sommati tra loro. Nell'architettura del sistema è previsto inoltre un ulteriore guadagno proporzionale (che moltiplica tutti i contributi del controllo appena sommati, l'unico elemento effettivamente regolabile del banco prova, modificabile attraverso l'opportuno potenziometro). Infine, in uscita si ha il segnale di errore in Volt che viene inviato al comando della servovalvola che rappresenta il blocco successivo.

1.2.2 Servovalvola

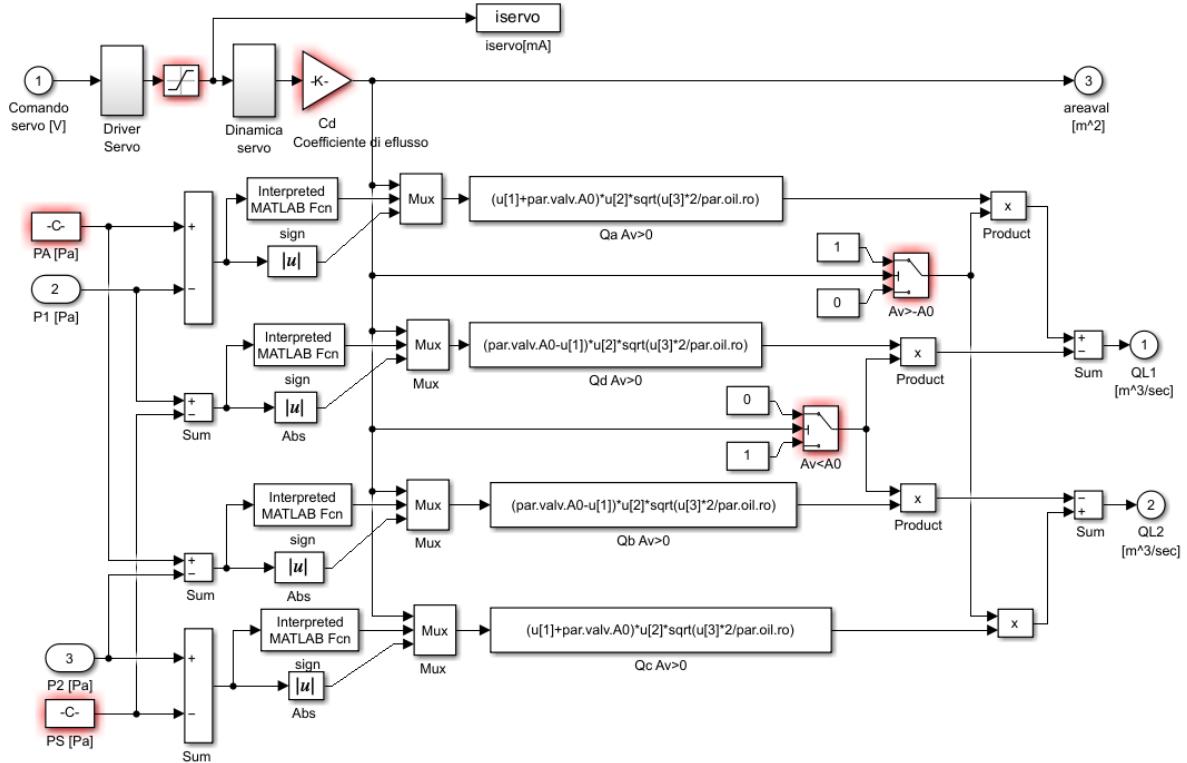


Figura 3 Modello simulink della servovalvola

Il comando, della servovalvola che è pari all'errore compensato, viene inviato al driver della servovalvola il cui scopo è quello di trasformare il segnale in tensione in un segnale in corrente necessario per il comando della valvola attraverso l'uso di solenoidi. Successivamente il segnale in corrente entra nel blocco della

dinamica della servovalvola necessario per trovare la porzione di valvola A_v che corrisponde all'apertura della servovalvola necessaria per far passare l'olio indirizzato al servoattuatore. Conoscendo le pressioni di alimentazione, di scarico e le pressioni all'interno delle camere (ricavate dal blocco attuatore) è possibile ricavare le portate che attraversano le valvole mediante la generica formula $Q = C_d A \sqrt{\frac{\Delta p}{\rho}}$ declinata caso per caso. È presente inoltre una serie di blocchi logici per identificare correttamente il segno delle portate in modo da definire il verso del loro flusso.

1.2.3 Bypass

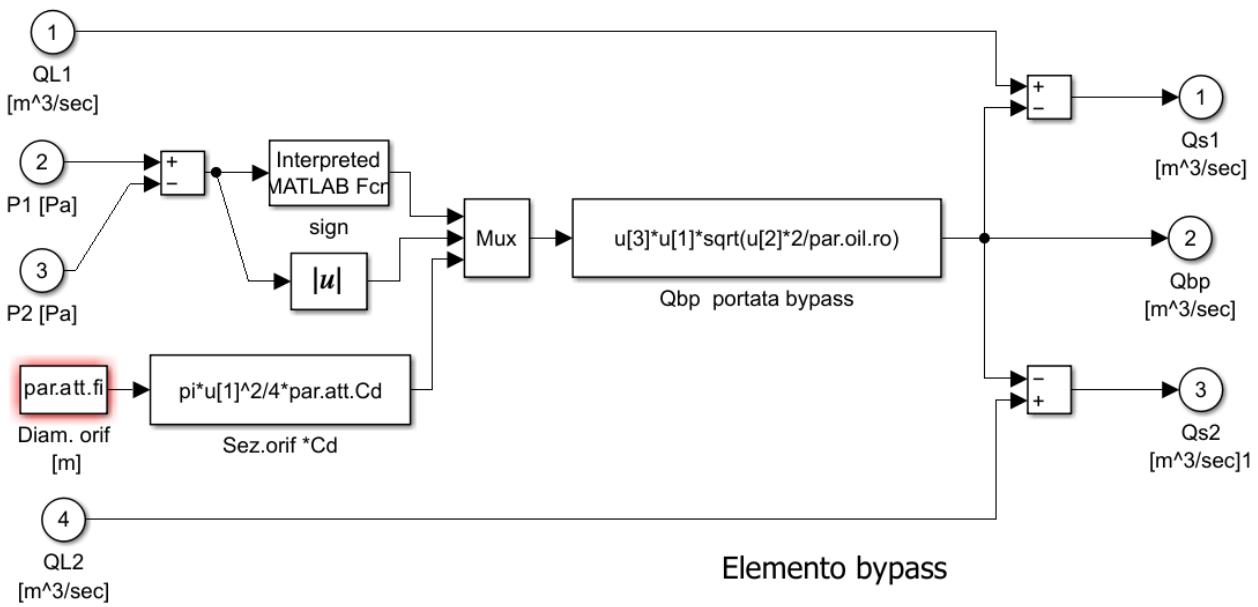


Figura 4 Modello Simulink dell'elemento di bypass

Il bypass mette in comunicazione i due condotti della servovalvola che collegano le camere dell'attuatore. Si tratta di un orifizio calibrato con sezione nota e definita in fase di progetto. Essendo sempre presenti delle perdite, previene in condizioni statiche un continuo alternarsi della valvola tra la posizione chiusa e aperta identificando invece una posizione di equilibrio con portata di bypass costante. Serve inoltre a migliorare le caratteristiche dinamiche garantendo una componente viscosa di ammortizzamento anche qualora si verificassero guasti al sistema.

1.2.4 Attuatore

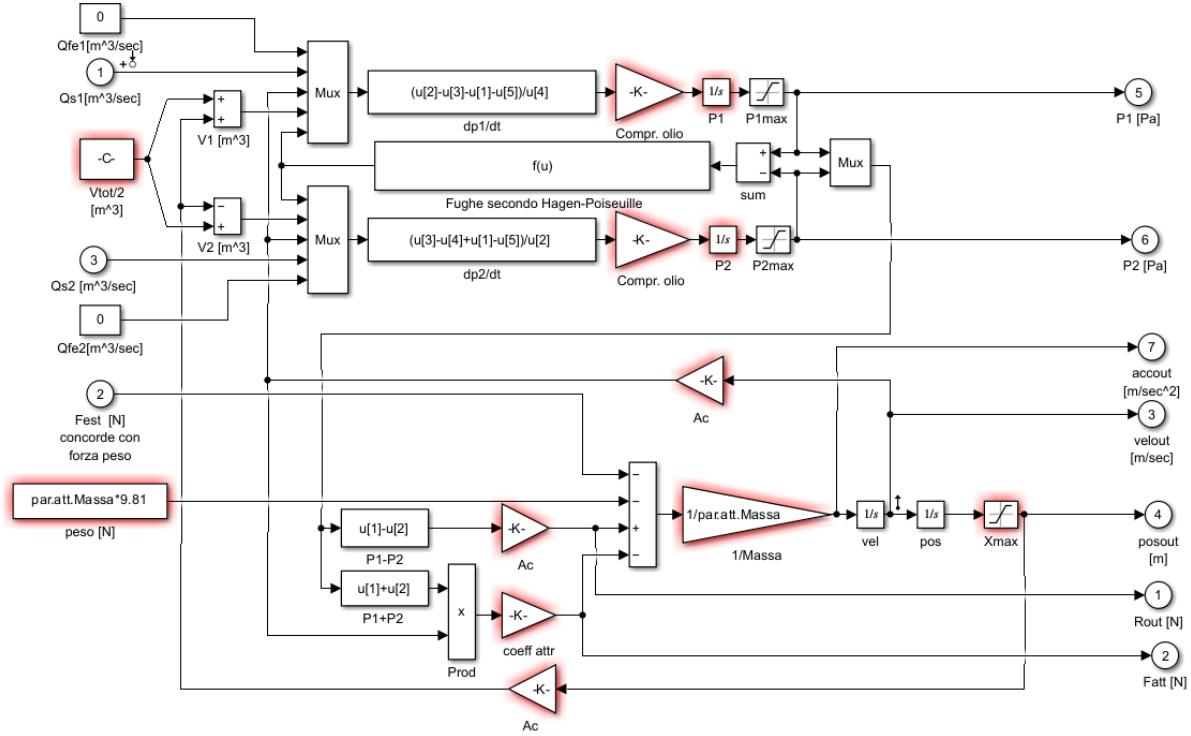


Figura 5 Modello simulink dell'attuatore

Partendo dalle portate calcolate per il bypass si riescono a ricavare le pressioni interne delle camere e i valori di posizione, velocità e accelerazione. Le pressioni vengono ricavate nel modo seguente:

$$\left(\frac{Q_{s1} - A_c \dot{x} - Q_{fe1} - Q_{HP}}{V} \right) \beta = \frac{dP_1}{dt}$$

$$\left(\frac{Q_{s2} - A_c \dot{x} - Q_{fe2} - Q_{HP}}{V} \right) \beta = \frac{dP_2}{dt}$$

Dove:

- Q_{s1} e Q_{s2} sono le portate in ingresso e in uscita della valvola
- A_c è l'area del cilindro
- Q_{fe1} e Q_{fe2} sono le fughe di olio verso l'esterno (sono state trascurate nel modello)
- Q_{HP} sono le perdite tra una camera e l'altra
- V è il volume della camera, il quale dipende dalla posizione dello stelo
- β è il modulo di comprimibilità dell'olio

La seconda equazione che viene utilizzata è quella dell'equilibrio dinamico dello stelo:

$$-F_{est} - M_{stelo} \cdot g + (p_1 - p_2) \cdot A_c - [(p_1 + p_2) \cdot A_c \cdot v_{att}] f / 2 = 0$$

Dove:

- F_{est} è il carico esterno che nel caso in esame è nullo
- f è il coefficiente di attrito

Dal bilancio di forze vengono ricavati i valori di accelerazione, velocità e posizione dello stelo, dove quest'ultima rappresenta il feedback del controllore.

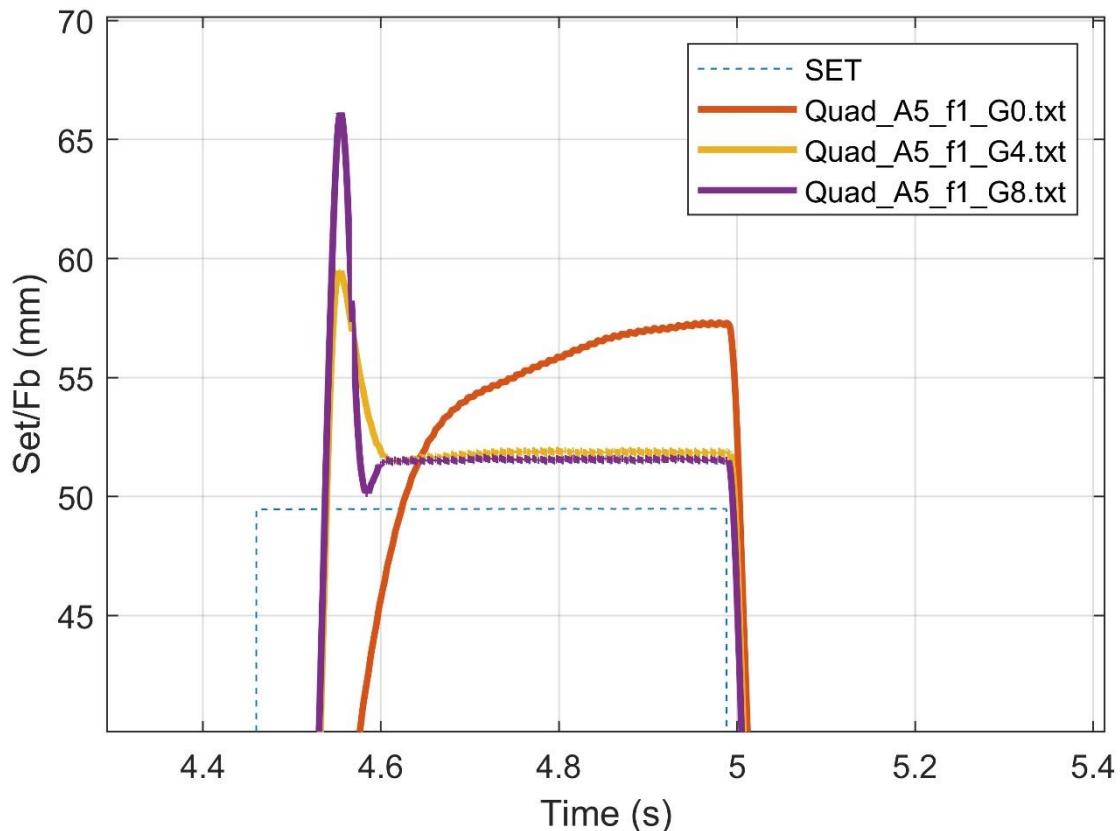
1.3 Analisi dei dati sperimentali e del modello

Dopo aver descritto il modello si eseguono delle elaborazioni sui dati e sul modello stesso.

1.3.1 Analisi andamenti sperimentali

In questa sezione si analizzano le prove sperimentali ottenute con un set a onda quadra e sinusoidale per valutare l'effetto del guadagno sulla risposta.

- Risposta onda quadra



- $G = 0$. Con guadagno nullo (curva rossa) il sistema presenta un ritardo considerevole nella risposta, non presenta overshoot e oscillazioni intorno al valore di set. Inoltre, si ha un errore a regime abbastanza consistente che lo rende poco performante.
- $G = 4$. Con guadagno pari a 4 (curva gialla) si ha una buona risposta del sistema. La risposta presenta un ritardo minore, di contro si ha un overshoot che però risulta essere un buon compromesso. Il sistema risulta quindi abbastanza pronto con un errore a regime minore rispetto a quello precedente. Probabilmente con questo guadagno si ha la miglior risposta delle 3 diagrammate.
- $G = 8$. Con guadagno pari a 8 (curva viola) si hanno le stesse caratteristiche a regime della curva gialla, si può vedere che la prontezza è pressoché la stessa, infatti, si nota dalla figura che le curve di salita sono quasi sovrapposte. Questo piccolo guadagno di prontezza e precisione a regime non migliorano complessivamente la risposta del sistema perché accompagnati da un overshoot troppo accentuato.

➤ Risposta onda sinusoidale

Figura 6 Confronto dati sperimentali ad onda quadra

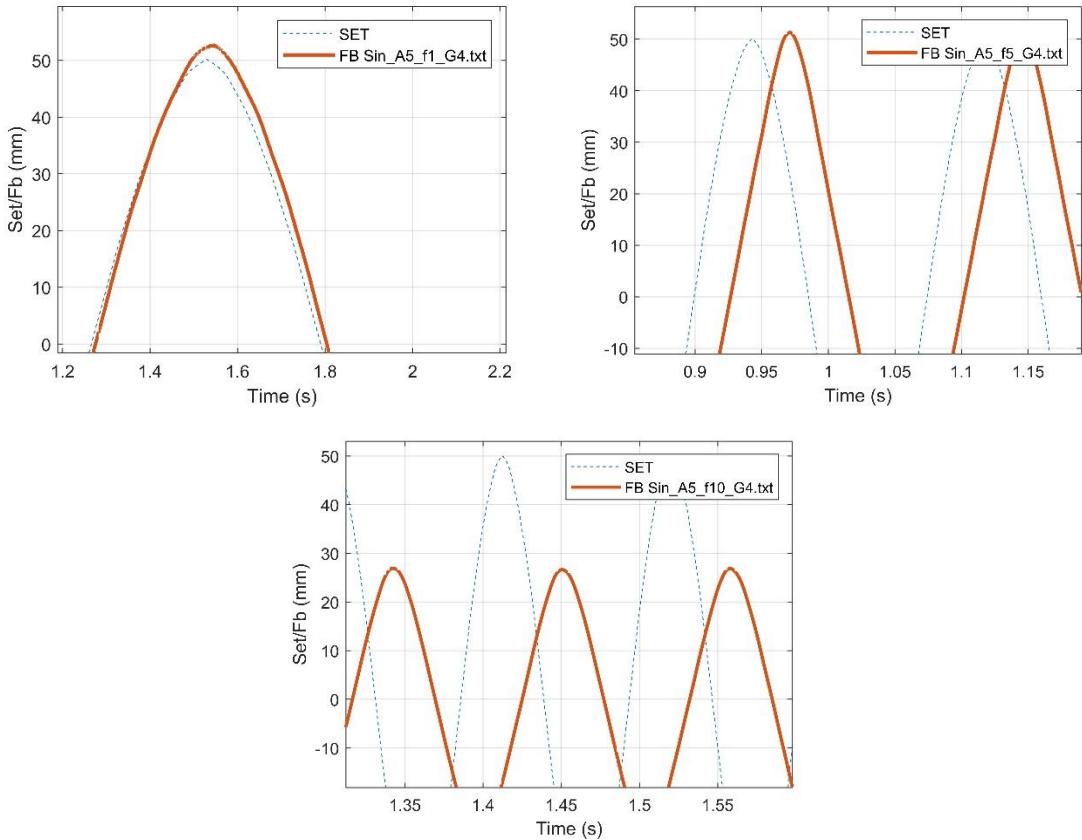


Figura 7 Confronto dati sperimentali con set sinusoidale al variare della frequenza

- $f = 1 \text{ Hz}$. In questo caso il sistema segue abbastanza fedelmente il set in ingresso sia in ampiezza che in fase.
- $f = 5 \text{ Hz}$. Con questa frequenza si vede che il sistema non attenua ancora in ampiezza perché siamo ancora in una frequenza all'interno della larghezza di banda; tuttavia, lo sfasamento risulta maggiore rispetto al caso precedente.
- $f = 10 \text{ Hz}$. Infine, con frequenze ancora più elevate, il sistema non segue più fedelmente il set. Il segnale viene attenuato in ampiezza e risulta anche considerevolmente sfasato. Con questa frequenza molto probabilmente si è oltre la frequenza di risonanza del sistema per il quale si hanno poi guadagni negativi e sfasamenti abbastanza importanti.

1.3.2 Confronto dati sperimentali con il modello

In questo paragrafo si confronteranno i dati sperimentali con quelli simulati in modo da poter validare il modello. La validazione del modello risulta utile perché permetterà successivamente di effettuare le prove sperimentali direttamente sul modello e non in laboratorio; in questo modo si possono eseguire più prove ma si hanno anche meno rischi di guasti con parametri che potrebbero essere al limite del range di funzionamento.

➤ Onda quadra

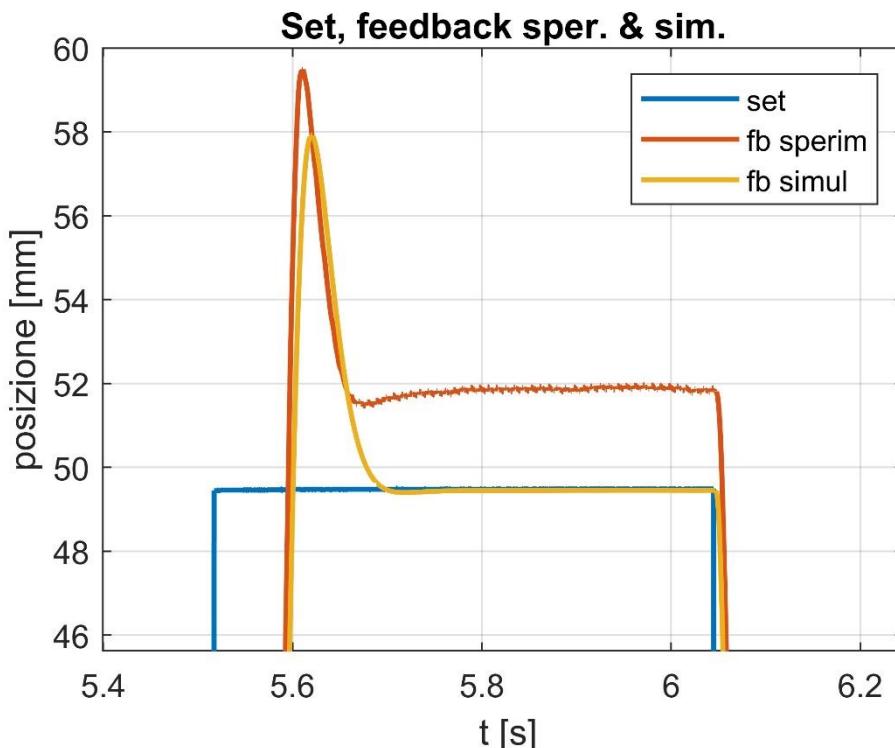


Figura 8 Confronto dati sperimentali e modello in posizione con set ad onda quadra

A regime presentano un comportamento diverso e sembrerebbe ci sia un errore di statismo, ma in realtà si tratta di un problema di lettura del segnale sperimentale.

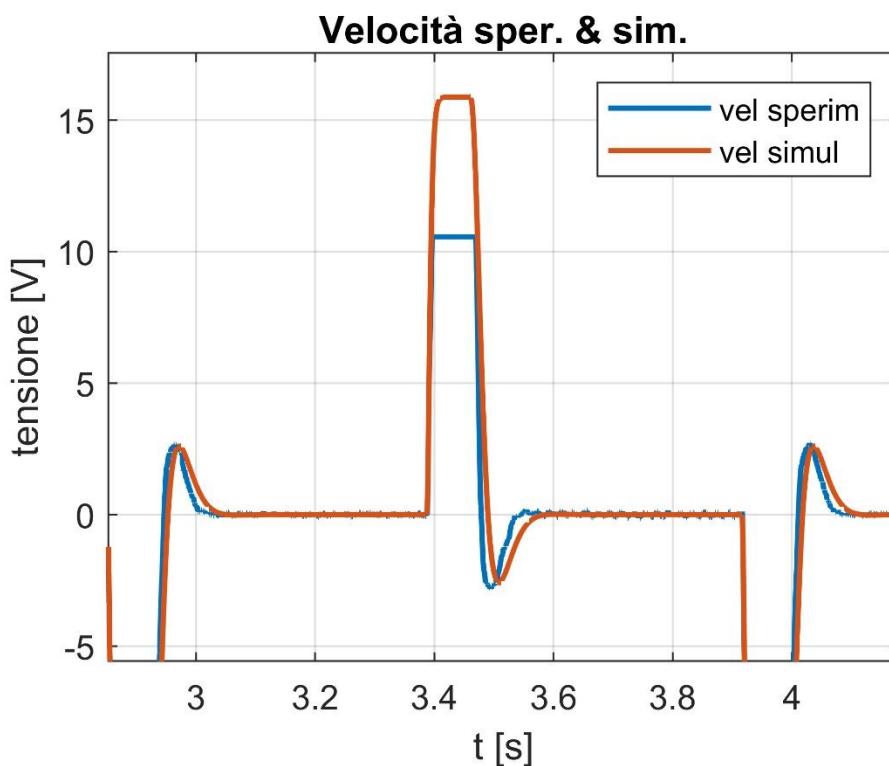


Figura 9 Confronto dati sperimentali e modello in velocità con set ad onda quadra

Nel confronto in velocità è subito evidente la presenza della saturazione a +/- 10 V inserito nel set-up sperimentale. I due segnali, tolta la saturazione non presente nel modello, sono comparabili.

Errore compensato, comando servovalvola sper. & sim.

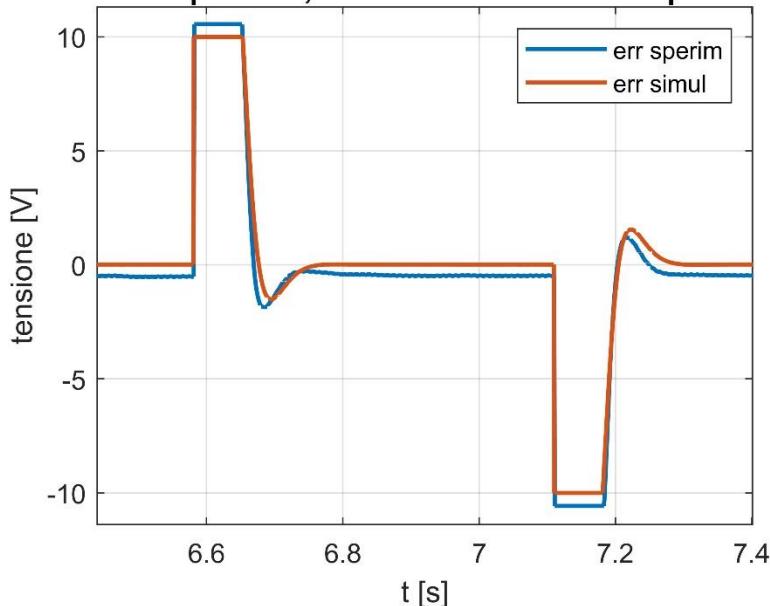


Figura 10 Confronto dati sperimentali e modello con set ad onda quadra per l'errore compensato

Per quel che riguarda l'errore compensato si vede che la saturazione è presente nell'errore simulato come si nota nella modello simulink in figura 2.

➤ Onda sinusoidale

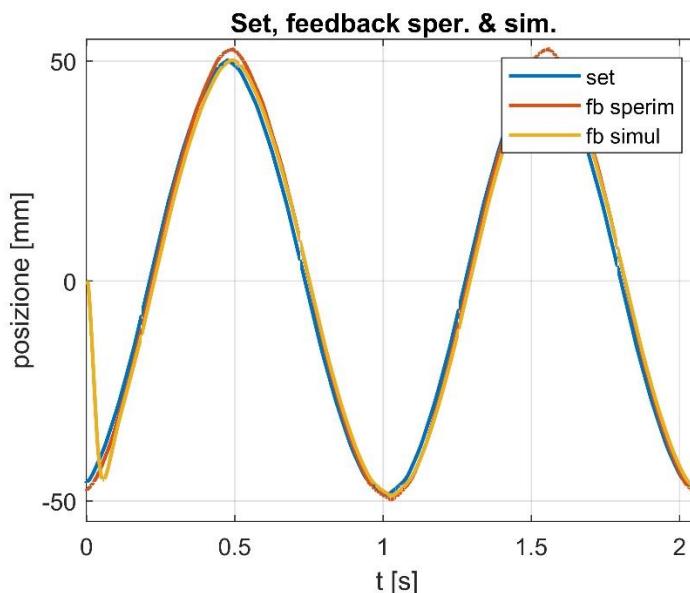


Figura 11 Confronto dati sperimentali e modello in posizione con set sinusoidale

Con un set sinusoidale il modello simulato è ben rappresentativo dei risultati sperimentali. Si ha una differenza di comportamento all'inizio della simulazione perché ovviamente il sistema simulato, senza condizioni iniziali, inizierà la simulazione con un feedback pari a 0, cosa che non accade sperimentalmente essendo che l'acquisizione parte ad un istante di tempo in cui il sistema è già in movimento.

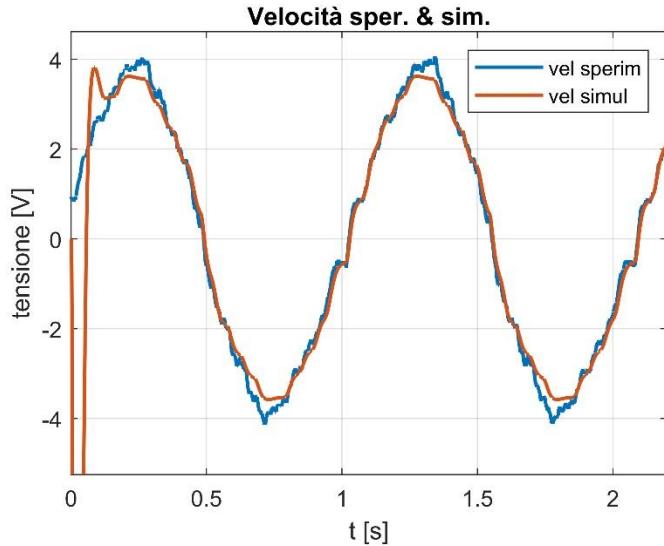


Figura 12 Confronto dati sperimentali e modello in velocità con set sinusoidale

Anche per quel che riguarda la velocità in un comando con set sinusoidale si può dire che il modello rappresenta correttamente, con differenze minime, i risultati sperimentali a meno dell'istante di tempo iniziale che crea dei problemi per i motivi già descritti in precedenza.

Si può concludere dicendo che il modello è molto rappresentativo di quello reale, quindi, è possibile usarlo per svolgere delle prove variando i guadagni o altri parametri in modo non obbligatoriamente fisico sul banco, così facendo si può avere già una buona stima del funzionamento del banco senza eseguire le prove su di esso.

1.3.3 Variazione di K_i e K_p in set ad onda quadra

Data la validazione del modello avvenuta con successo si possono valutare sul modello che influenza hanno i parametri modificabili nel controllore. Per le prove si utilizza un set ad onda quadra con ampiezza pari a 5V e frequenza di 1 Hz.

➤ Variazione di K_p

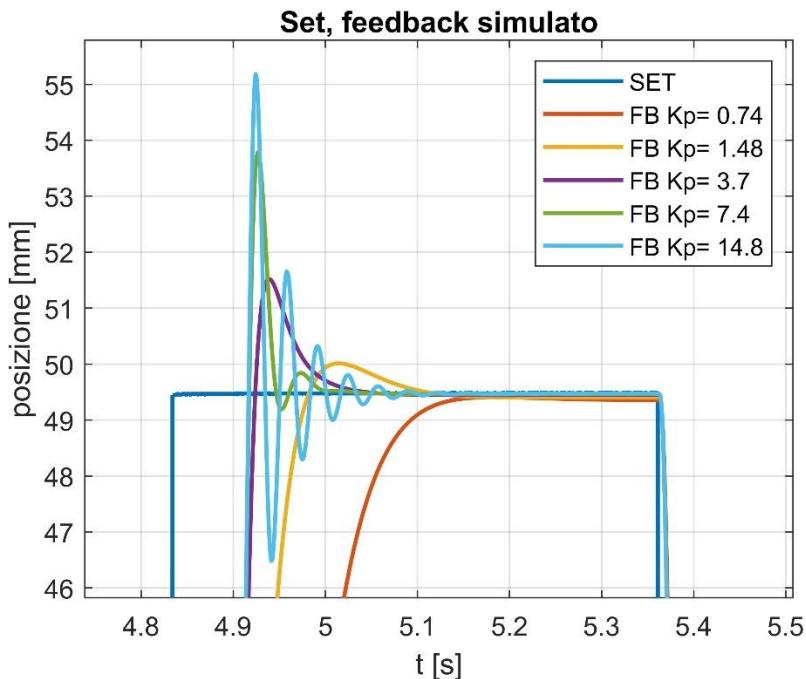


Figura 13 Analisi risposta onda quadra al variare di K_p

All'aumentare del guadagno proporzionale il sistema risulta più pronto e ha tempi di risposta quindi più brevi però le instabilità tendono ad aumentare. Infatti, all'aumentare di K_p il valore dell'overshoot

aumenta considerevolmente e di pari passo anche le oscillazioni intorno al valore di set. Al contrario con valori di K_p si ha un sistema che è molto più lento e che non raggiunge le condizioni di set fissate a causa di un errore a regime che viene ridotto all'aumentare di K_p che, come è già stato detto, ha come effetto negativo l'instabilità che dà al sistema.

➤ Variazione di K_i

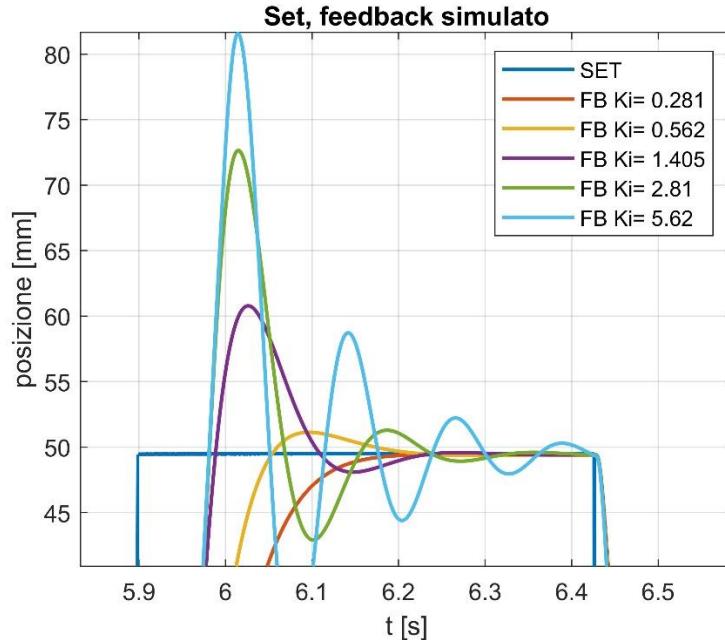


Figura 14 Analisi risposta onda quadrata al variare di K_i

Sembrerebbe si abbia un comportamento molto simile a quello ottenuto con il guadagno proporzionale, ma ci sono delle differenze. La prima differenza è che con la presenza del K_i , anche per valori bassi, l'errore a regime tende ad annullarsi ed è questo il principale effetto voluto dal guadagno integrativo. Al tempo stesso è necessario non avere un guadagno troppo elevato perché, come per quello proporzionale, un valore troppo alto genera delle instabilità. Un'ulteriore differenza è che le oscillazioni intorno al valore di set sono più dilatate nel tempo e non ravvicinate come per il guadagno proporzionale, questo dimostra che il guadagno integrativo ha effetto minore sulla prontezza del sistema.

1.3.4 Fitting della massa mobile

Lo scopo di questa sezione è il calcolo della massa mobile della cella di carico. Il funzionamento della cella di carico è il seguente: Quando la massa mobile viene soggetta ad una accelerazione si ha la presenza di una forza di inerzia che causa la deformazione dell'elemento sensibile della cella. Questa deformazione genera quindi un segnale di forza che può essere misurato. L'accelerazione necessaria per completare la formula ($F = m \cdot \ddot{x}$) si ricava derivando due volte la posizione poiché non è direttamente nota. Il fitting viene eseguito calcolando l'accelerazione derivando sia la posizione che la velocità in modo da confrontare i risultati.

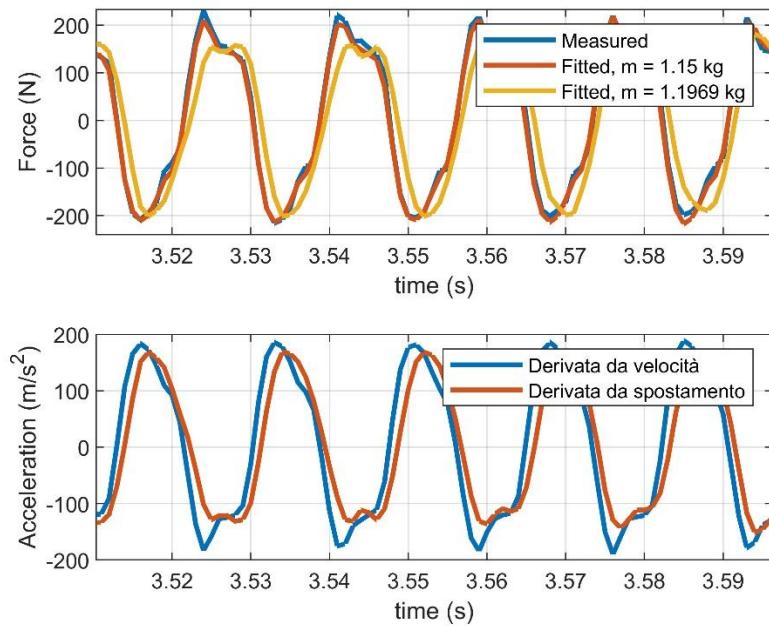


Figura 15 Fitting della massa mobile

I valori di massa mobile ottenuti sono i seguenti:

- 1.15 kg; massa ottenuta derivando la velocità
- 1.1969 kg; massa ottenuta derivando la posizione

Dal grafico si può vedere come la forza ottenuta mediante l'accelerazione espressa come derivata della velocità, segua abbastanza fedelmente quella misurata. Il motivo di ciò è dovuto al fatto che le derivando si tendono a perdere informazioni. Quindi più si eseguono derivate e più l'informazione è meno precisa.

1.3.5 Diagrammi di bode sperimentali e simulati

In questa sezione vengono confrontati i diagrammi di bode ottenuti sperimentalmente e quelli simulati.

- Confronto guadagno G=2

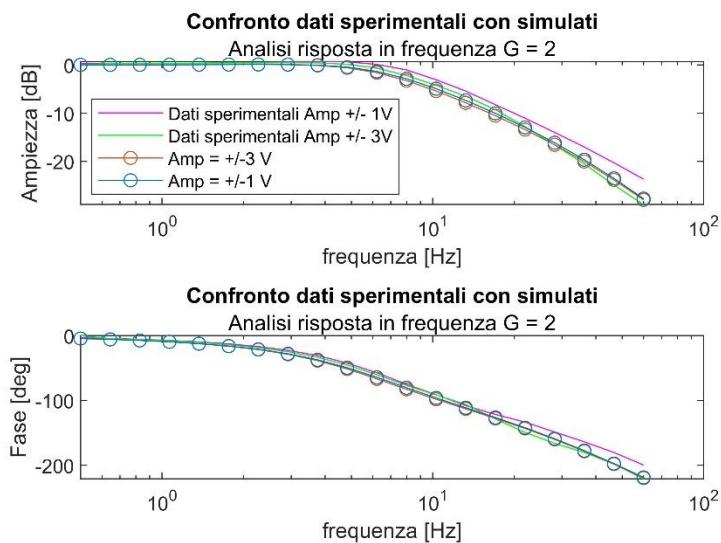


Figura 16 Diagramma di Bode con G=2

Il diagramma di Bode non dovrebbe cambiare al variare dell'ampiezza dell'onda sinusoidale in ingresso però in questo caso, essendoci delle non linearità nel sistema, si ha un Bode influenzato anche dall'ampiezza dell'onda sinusoidale. I valori sperimentali e quelli simulati si nota che hanno un andamento molto simile tra loro e che l'ampiezza della sinusoide non influisce in maniera eccessiva sull'andamento dei diagrammi.

Sinusoidi con ampiezza in ingresso maggiore aumentano la probabilità che l'errore compensato possa andare in saturazione.

➤ Confronto guadagno G=8

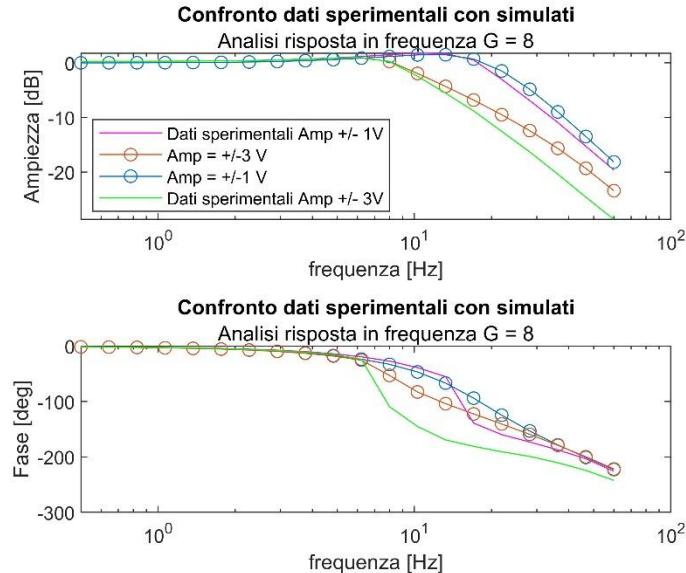


Figura 17 Diagramma di Bode con $G=8$

Aumentando il guadagno si nota come l'ampiezza della sinusode in ingresso sia molto influente sui grafici in uscita. Anche qui si può vedere come i dati simulati rispecchino abbastanza bene quelli del modello tranne per la fase dove intorno a una frequenza di 10 Hz si presentano degli andamenti diversi dovuti a delle correzioni che sono state fatte sui dati della fase ottenuti sperimentalmente; si sono ottenuti dei valori di fase positivi, cosa non possibile, e quindi si sono dovuti correggere non sapendo con certezza assoluta che il sistema abbia misurato correttamente la fase.

➤ Confronto Bode al variare del guadagno

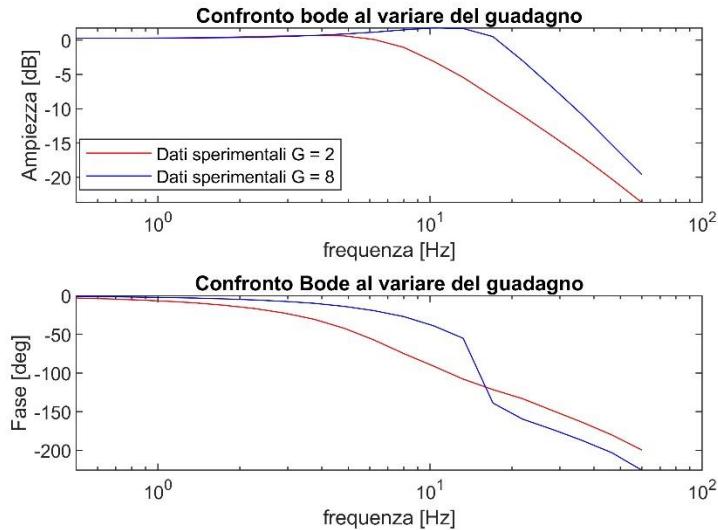


Figura 18 Confronto Bode all'aumentare del guadagno

L'aumentare o il diminuire del guadagno influenza notevolmente la banda passante che presenta il sistema. Dalla figura mostrata sopra risulta abbastanza evidente che all'aumentare del guadagno aumenta la banda passante del sistema; infatti, il sistema attenuerà e sfaserà per una frequenza più alta. L'aumento della banda passante porta però ad avere una riduzione del margine di fase (il grafico passa gli 0 dB per una frequenza maggiore e quindi la fase si è avvicinata a -180°) e del margine di guadagno (quando la fase è -180° il modulo non ha ancora attenuato). Nella configurazione con guadagno minore il fattore limitante per identificare il

campo di lavoro del sistema risulta essere il margine di guadagno, al contrario nella configurazione con margine maggiore, il netto aumento della frequenza di taglio dei 3 dB, rende più problematica la stabilità per quanto riguarda il margine di fase. Con un guadagno maggiore la curva tende a un comportamento assimilabile a un sistema del secondo ordine leggermente sottosmorzato, con la presenza di un leggero picco di risonanza per valori nell'intorno di 10 Hz.

1.3.6 Risposta del modello a rampa e forza esterna fissati

Si valuta ora la risposta del modello ad una rampa di salita da 0V a 3V nel tempo che intercorre tra 3s e 4s con un disturbo di forza fissato tra 5s e 7s; la discesa del segnale è ottenuta mediante gradino. Si analizzano quindi i risultati ottenuti per quel che riguarda posizione, errore, area di apertura della valvola, portata, pressione e velocità dell'attuatore.

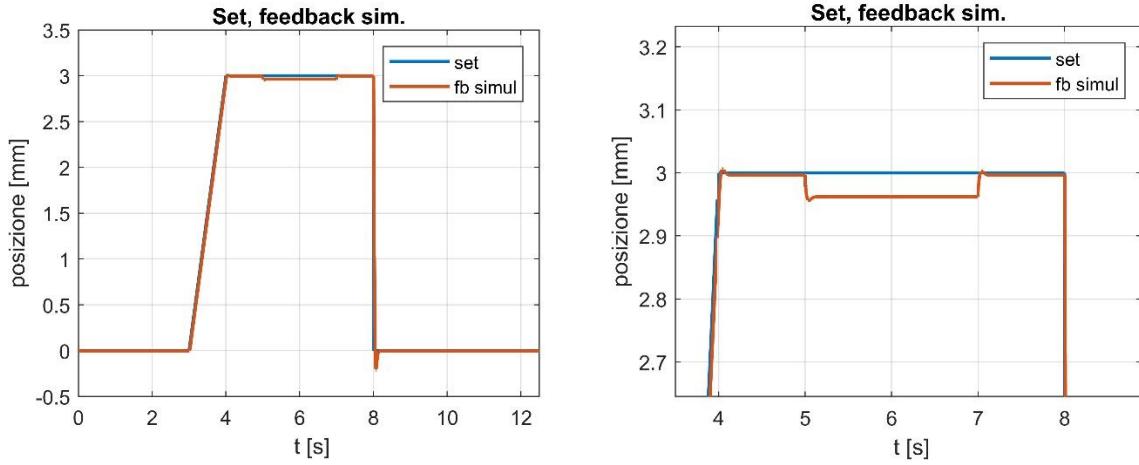


Figura 19 Dettaglio del set e del feedback fissati

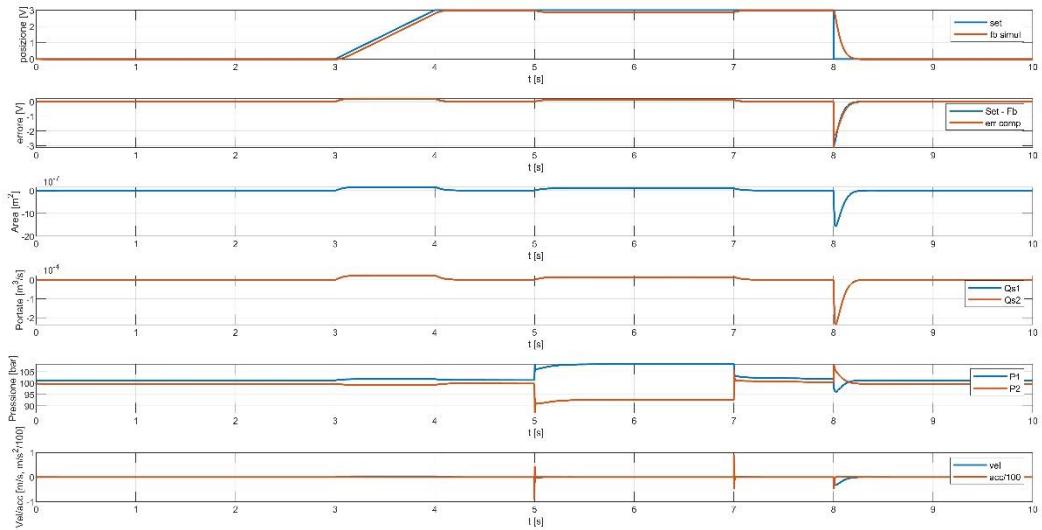


Figura 20 Figura riepilogativa di tutti i segnali del modello

La simulazione del modello è stata eseguita con un guadagno del potenziometro nullo in modo da vedere con maggior chiarezza tutti gli andamenti delle varie grandezze che altrimenti risulterebbero difficili da analizzare data la velocità di risposta del controllo.

1.4 Conclusioni

In questa esercitazione abbiamo studiato ed analizzato un servoattuatore idraulico relativamente alla sua struttura e alle sue prestazioni. In particolare la prima parte è stata eseguita in maniera sperimentale in un banco realizzato ad hoc in laboratorio, per poi, in un secondo momento, confrontare i dati ottenuti con un modello matematico realizzato tramite MatLab Simulink.

1.4.1 Risultati ottenuti

Osservando i dati sperimentali sulle onde sinusoidali e quadrate si possono subito valutare il guadagno e la frequenza impostati. Per quanto riguarda il guadagno il valore migliore risulta essere 4, poiché il sistema reagisce rapidamente ed in maniera più performante rispetto ai casi con $G=0$ e con $G=8$ dove, nel primo caso era meno reattivo e nel secondo caso presente un overshoot troppo marcato. Relativamente alla frequenza il caso migliore è quello con $f = 1$ Hz dove il feedback riesce a seguire bene il set.

Introducendo il modello matematico implementato tramite Simulink è subito evidente che è coerente con gli andamenti sperimentali. Dal confronto tra Set e Feedback a onda quadra con guadagno $G=4$ si nota che le forme sono molto simili, ma a regime l'andamento sperimentale sembra avere un errore di statismo, probabilmente dovuto ad un errore di lettura. Anche nel caso di risposta a un segnale sinusoidale il modello ottiene risultati che si avvicinano ai dati sperimentali ricavati nelle diverse prove.

Il passo successivo è stato quello di studiare la reazione del modello al variare del guadagno proporzionale e del guadagno integrativo. Riguardo K_p il miglior valore che permette dia vere il giusto compromesso tra reattività, stabilità e riduzione dell'errore a regime risulta essere $K_p=3,7$, cioè né troppo alto né troppo basso perché per valori elevati presenta un'evidente instabilità mentre per valori piccoli ottengo una risposta lenta. Invece per il guadagno integrativo la prima osservazione notata è stata quella della maggior suscettibilità del sistema al variare di K_i rispetto a K_p ed il valore migliore ottenuto è stato $K_i=1,405$ in cui a regime il Set corrisponde al Feedback quasi perfettamente senza alcun tipo di instabilità.

In seguito è stato effettuato il fitting della massa mobile opportunamente collegata alla cella di carico sulla testa del martinetto ed il risultato migliore è stato quello di $m=1,150$ kg. Successivamente è stata svolta un'analisi della risposta in frequenza confrontando i dati sperimentali con il modello matematico, osservando in particolare la dipendenza dei risultati con l'ampiezza del segnale in ingresso, evidenziando così le non linearità presenti nel sistema e di conseguenza riportate nel modello. Infine è stata studiata la risposta del modello ad un test preimpostato: comando di rampa in posizione, posizione costante, disturbo in forza a gradino. Le variabili analizzate sono posizione, errore, area di apertura della valvola, portata, pressione, e velocità di apertura della valvola.

2 Banco prova ASTIB

2.1 Introduzione

Lo scopo di questa esercitazione è lo studio di un banco prova servocomandi EMA. In particolare, il banco prova ha un attuatore lineare idraulico (HLA) controllato in forza e un attuatore elettromeccanico lineare (EMA) controllato in posizione. Lo studio verrà fatto sulla forza prodotta dal HLA il cui disturbo in posizione è prodotto dall'EMA. Uno schema del sistema analizzato è il seguente:

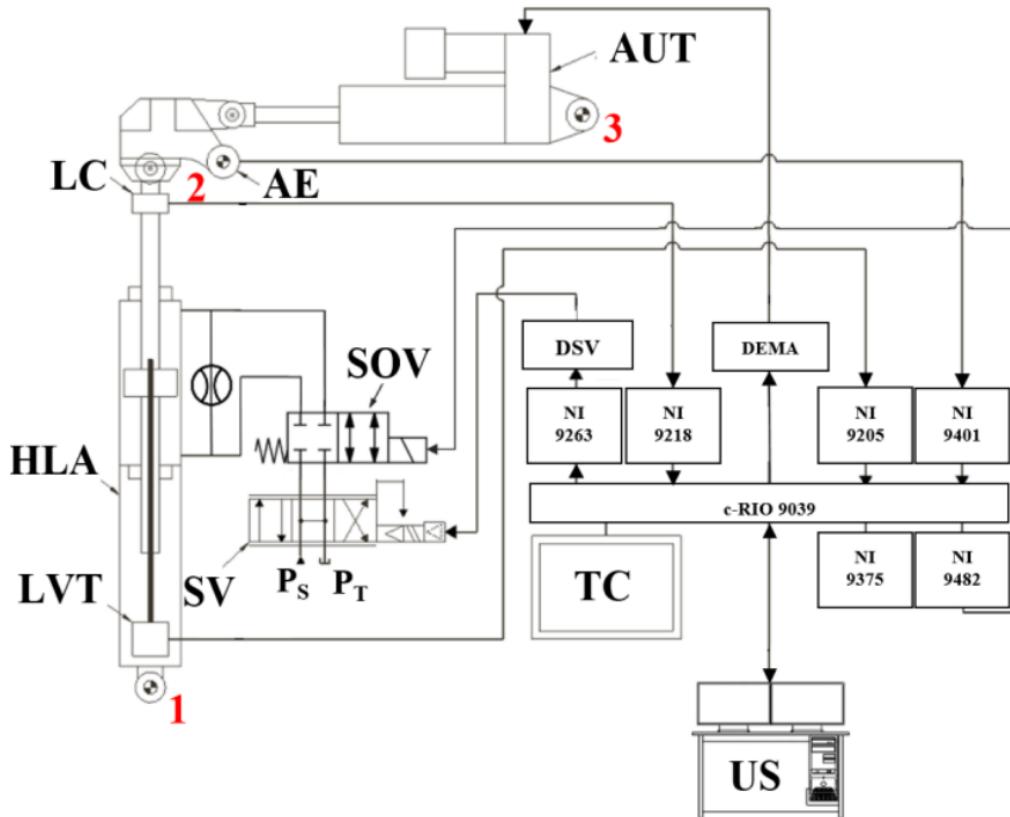


Figura 21 Schema del banco prova

I due attuatori “comunicano” tra loro essendo collegati allo stesso elemento mobile, il quale consiste in un giunto a gomito che ruota rispetto ad un punto fisso.

L'EMA, come già detto in precedenza, è un attuatore lineare controllato in posizione ad attuazione elettrica, il cui controllo posizione è eseguito dal driver DEMA. Il feedback di velocità e posizione si ottiene attraverso un resolver montato sul motore elettrico dell'EMA.

L'HLA è un servoattuatore idraulico controllato in forza, il cui feedback arriva da una cella di carico interposta tra l'attuatore e il giunto, comandato da una servoalvalvola, e alimentato da una linea idraulica, il cui schema è il seguente:

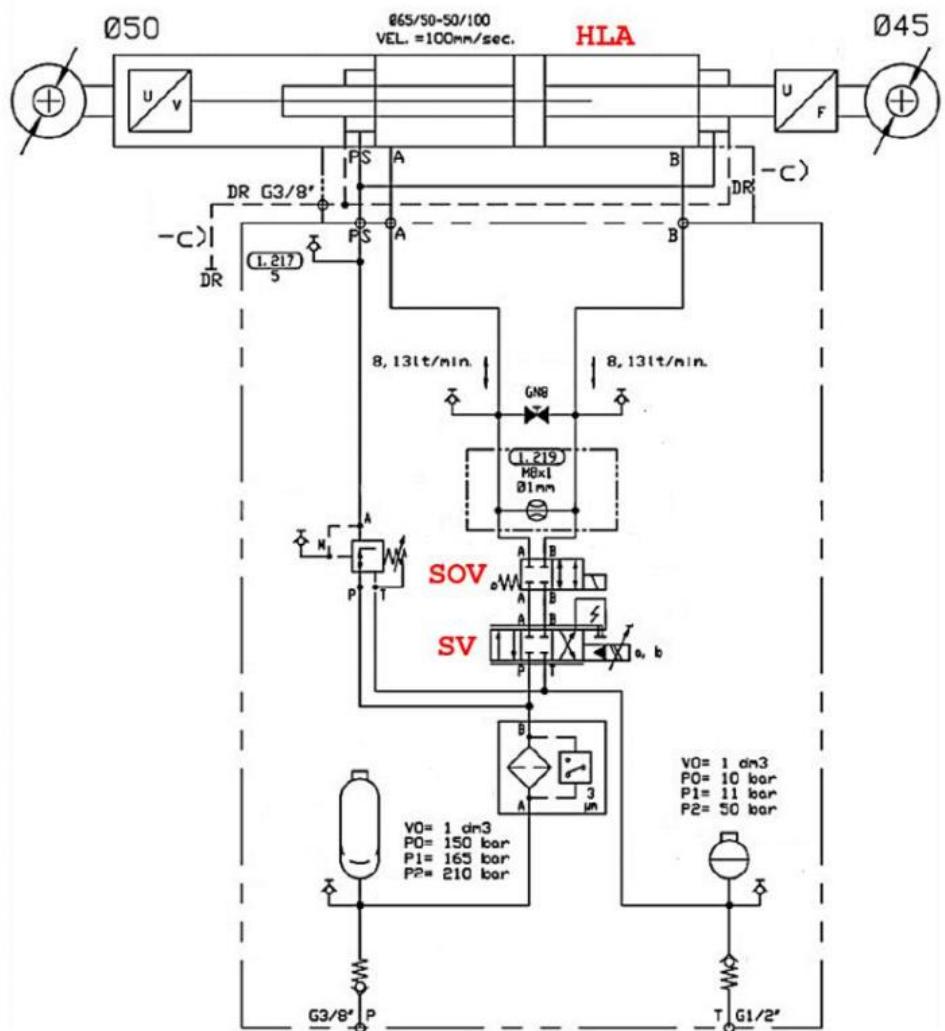


Figura 22 Schema dell'attuatore HLA

2.2 Analisi prove

In questa sezione verranno analizzati 4 prove realizzate con set di forza e disturbo differenti in modo da analizzare il comportamento dell'intero sistema. La prima prova è quella su cui ci si dilungherà maggiormente poiché alcune considerazioni sono ripetute anche per le prove successive.

2.2.1 Prova 1

In questa prova viene dato, al sistema, in input un segnale di forza ad onda quadra con un disturbo in posizione sempre ad onda quadra. Set e disturbo sono mostrati nella figura che segue:

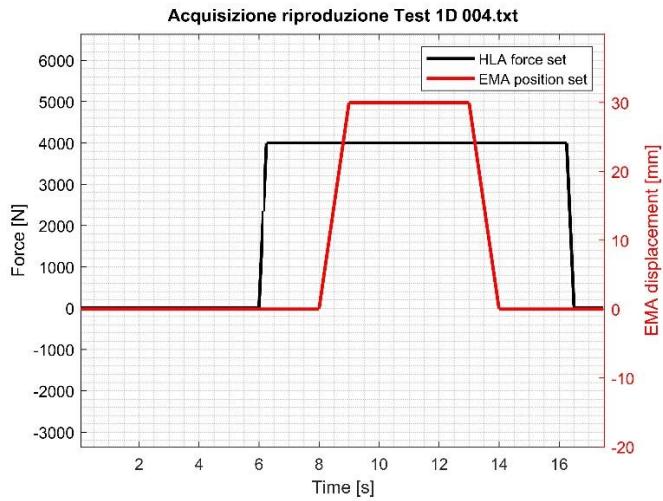


Figura 23 Set e disturbo prova 1

La primissima analisi viene fatta sul modello e sui dati sperimentali mettendoli a confronto:

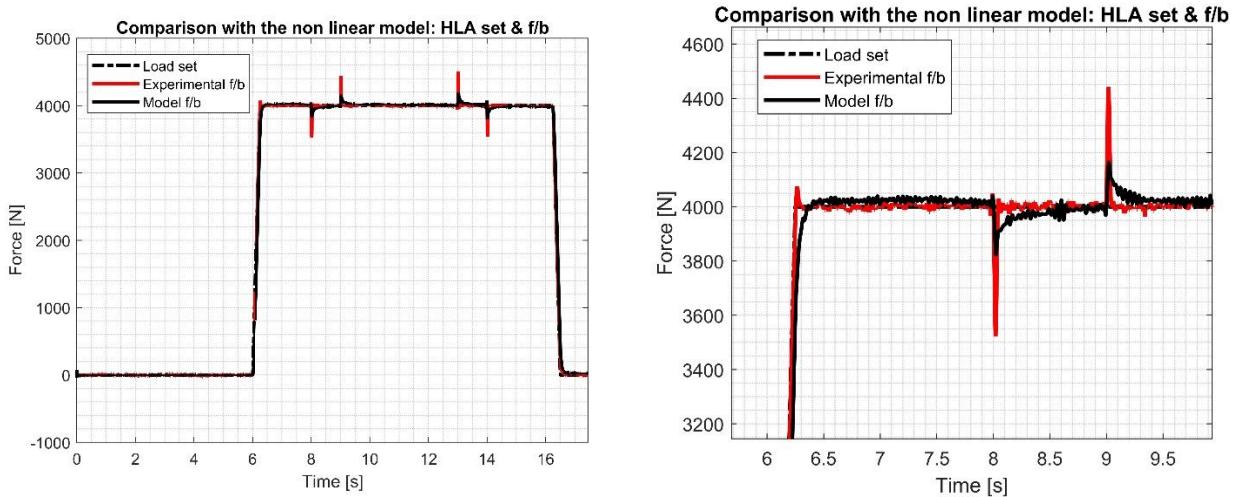


Figura 24 Confronto con il modello prova 1

Il set in forza viene eseguito abbastanza fedelmente dall'HLA tranne dove inizia e finisce la rampa del disturbo sia in salita che in discesa dove si notano dei picchi di forza. Tra i dati del modello e i dati sperimentali si nota una piccola differenza nel replicare il tratto di salita dell'onda quadra del set in forza dove il segnale sperimentale sembra essere più rapido e presenta al contempo un overshoot maggiore; questo potrebbe far pensare ad un guadagno proporzionale dell'intero sistema che sperimentale è più alto nel modello per via di alcune semplificazioni necessarie nella costruzione del modello. Altri punti da analizzare sono gli istanti di tempo in cui inizia il disturbo; si ha una diminuzione repentina della forza dovuta al set di posizione dell'EMA, il quale essendo nella stessa direzione della forza tende, per un piccolissimo istante di tempo ad alleviare la pressione all'interno della camera. Il viceversa avviene quando il set di posizione termina la rampa o quando inizia la discesa; qui il sistema si trova in una situazione in cui immediatamente dovrà fornire meno olio alla camera del cilindro idraulico e quindi per un piccolissimo istante di tempo si ha un picco di forza dovuto alla costante di tempo dell'intero sistema oleodinamico che comanda l'attuatore.

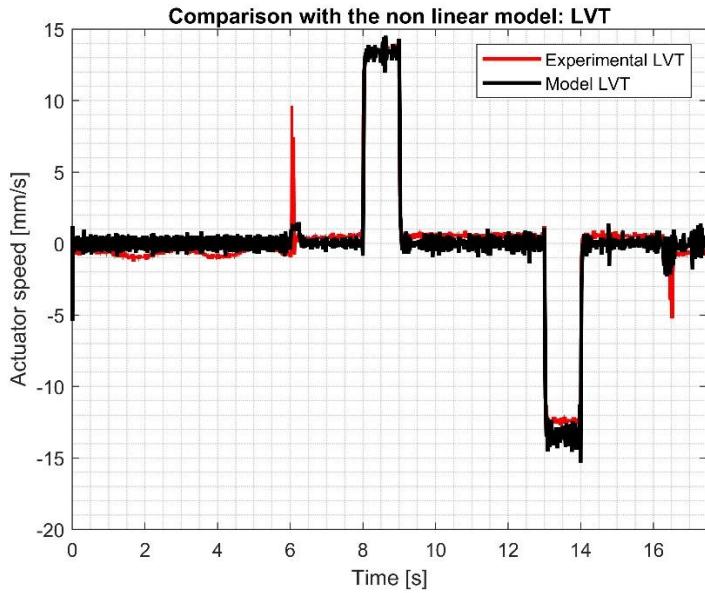


Figura 25 Velocità attuatore prova 1

Dalla figura appena mostrata si riesce a comprendere immediatamente come sia la velocità, il reale disturbo del set in forza dell’HLA. Difatti, in corrispondenza dell’istanti di tempo dove è presente la velocità si ha il sistema oleodinamico che deve “lavorare” per seguire il set in forza richiesto. Si può anche vedere come nel tratto a velocità costante, e quindi senza disturbo, il sistema riesca a replicare la forza più fedelmente. Per questa analisi bisogna confrontare la figura 24 con la figura 25.

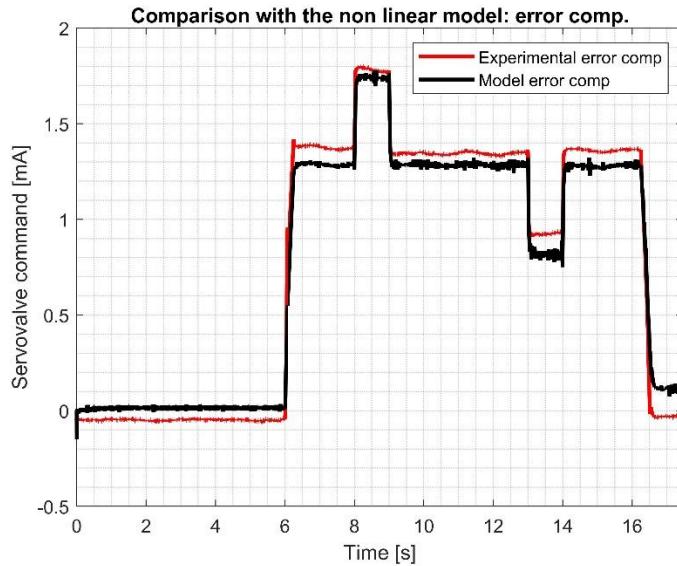


Figura 26 Comando servoalvalva prova 1

Dal comando del solenoide della servoalvalva si riesce a comprendere la portata che arriva alla camera dell’attuatore quindi risulta un andamento importante da analizzare. In corrisponde del set in forza si ha un aumento del comando della servoalvalva che aumenterà la luce di passaggio della valvola e quindi genera una portata maggiore. Durante la rampa del set in forza la valvola genera un’apertura crescente che diventa costante una volta terminata. Tutto il tratto a forza costante la valvola non passa alla configurazione a centri chiusi perché ci sono delle perdite tra le due camere dell’attuatore e perché l’olio, per quanto sia un fluido, presenta una minima comprimibilità. Analizzando il disturbo si ha che quando il disturbo è concorde alla forza applicata dall’HLA la valvola deve garantire un’apertura maggiore per cercare di seguire il moto dell’attuatore e quindi aumentando la portata in modo da garantire la pressione necessaria all’interno della camera. Per la discesa del disturbo vale il viceversa.

Si possono ora analizzare le pressioni all’interno delle due camere dell’attuatore.

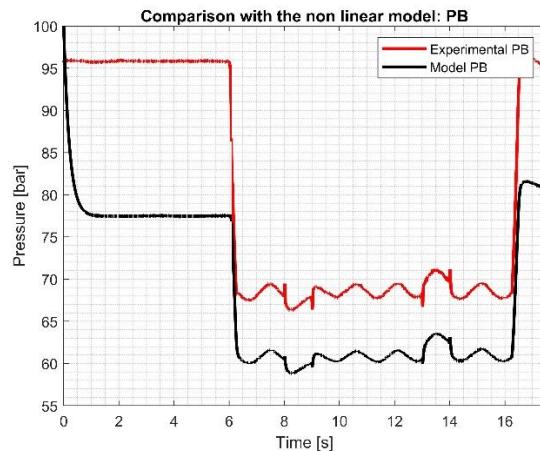
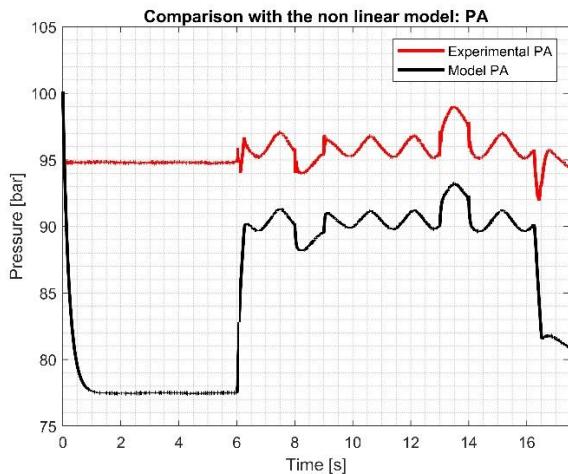


Figura 27 Pressione camera posteriore (PA) e camera anteriore (PB) prova 1

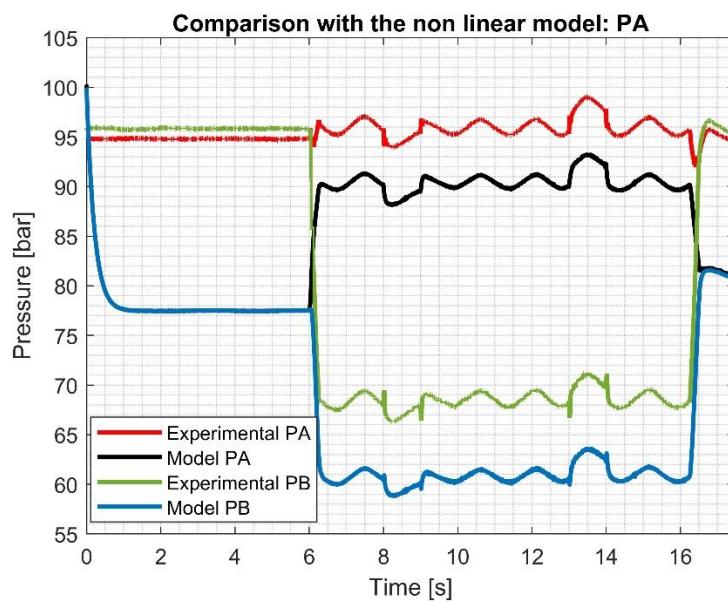


Figura 28 Confronto pressione prova 1

Confrontando le due pressioni su uno stesso grafico si nota subito come ovviamente ci sia una differenza di pressione nelle due camere che è responsabile della generazione della forza da parte dell'attuatore. Le due pressioni, sia sperimentale che del modello, non sono costanti ma presentano delle oscillazioni perché la stessa pressione di alimentazione non è costante. Ad istante di tempo iniziale si nota come tra le due pressioni sperimentali la P_B sia più alta per via della presenza dello stelo nella camera posteriore. Con l'inizio del set in forza la pressione della camera anteriore diminuisce repentinamente. Il disturbo in posizione si vede graficamente dove le due pressioni delle due camere presentano un picco che si discosta dall'andamento simile ad una sinusoida che presentano a regime. Con il disturbo crescente si ha un aumento di P_B e una diminuzione di P_A , viceversa per il disturbo decrescente.

Ora si inserisce un grafico che mostra l'andamento della pressione di alimentazione che giustifica in parte l'andamento non costante delle due pressioni:

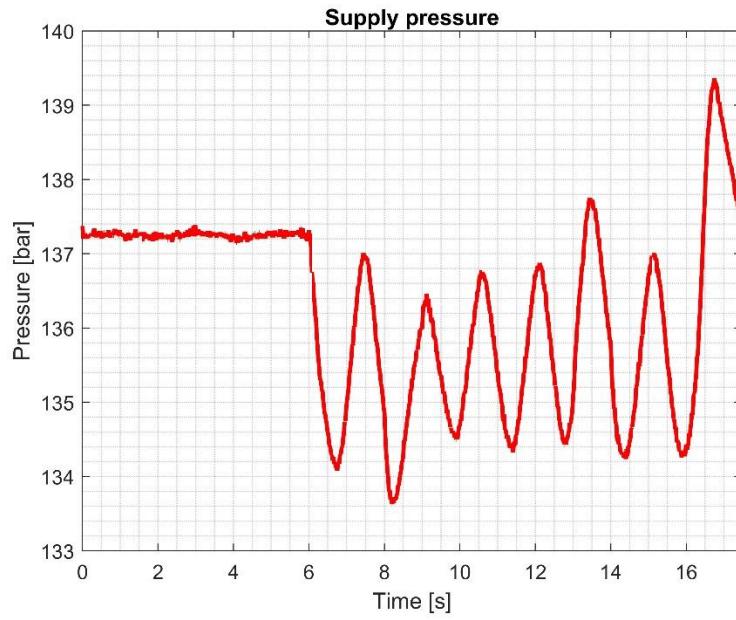


Figura 29 Pressione di alimentazione prova 1

Come ultima analisi si mostra ora l'andamento del set e del feedback del disturbo generato dall'EMA:

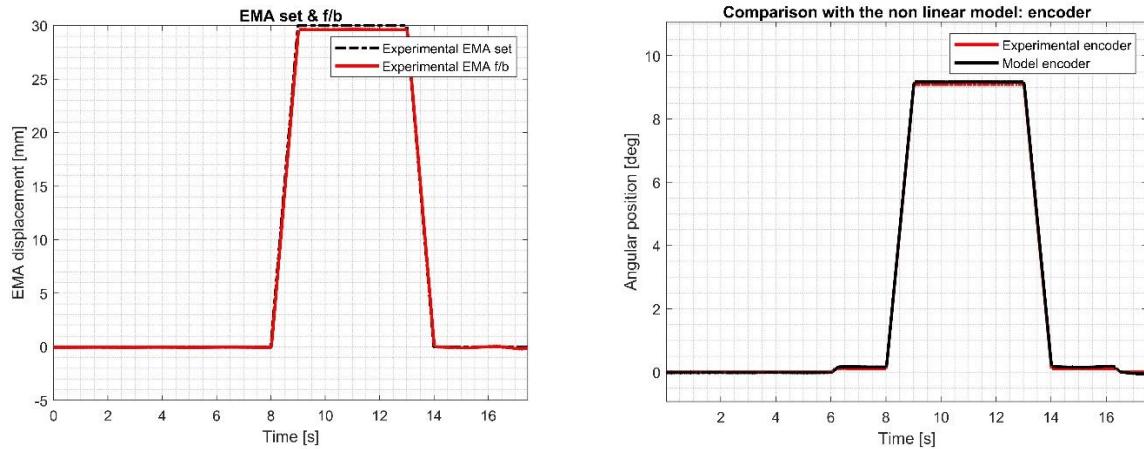


Figura 30 Set e feedback posizione dell'EMA prova 1

2.2.2 Prova 2

Per la seconda prova si è impostato un set in forza nullo con un disturbo sinusoidale con valore medio nullo.

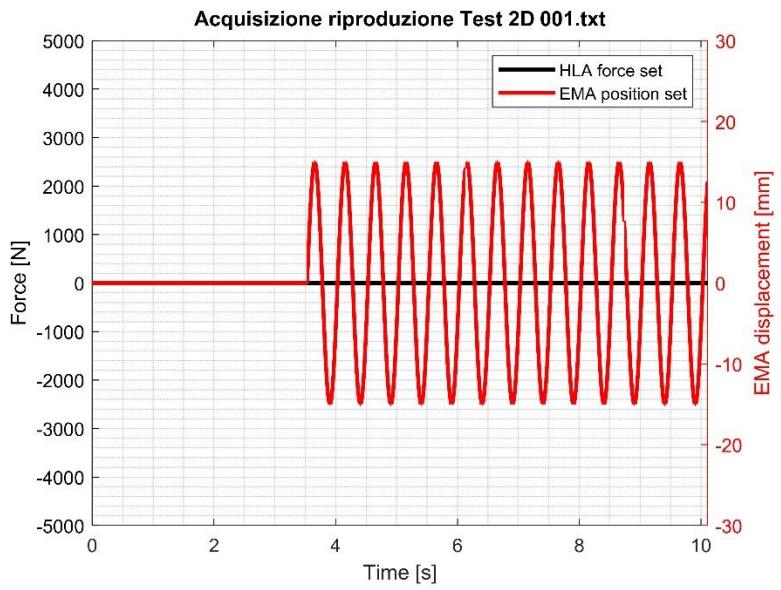


Figura 31 Set e disturbo prova 2

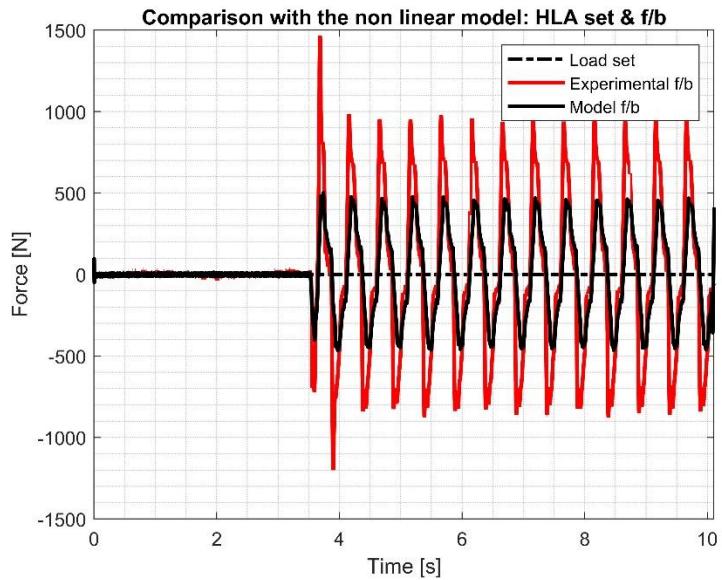


Figura 32 Confronto con il modello prova 2

Modello e dati sperimentali presentano un andamento simile tranne che per i massimi e i minimi della sinusoide; infatti, i dati sperimentali hanno valori di forza molto maggiori come se il sistema sperimentale fosse più sensibile al disturbo e che quindi riesca meno a mantenere una forza costante.

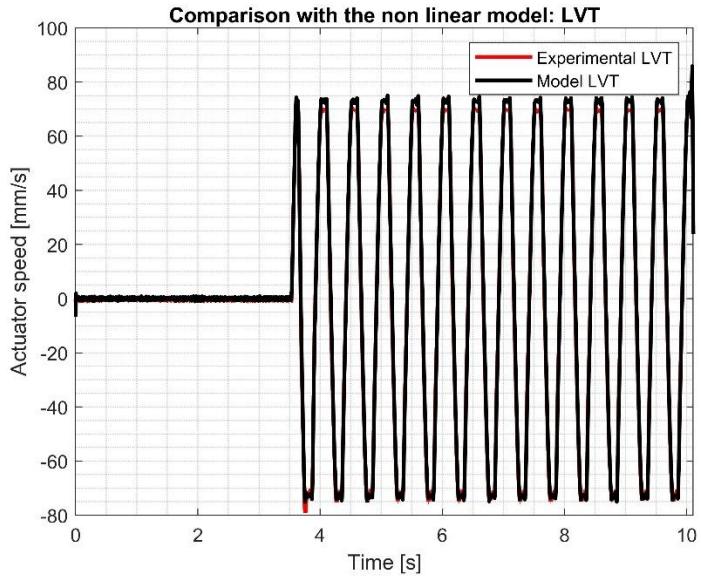


Figura 33 Velocità attuatore prova 2

La velocità dell'attuatore non ha un andamento sinusoidale preciso come quello del disturbo in posizione, questo potrebbe essere dovuto all'inerzia del sistema quando si inverte la direzione del moto dato che il sistema idraulico non è uno dei sistemi più rapidi.

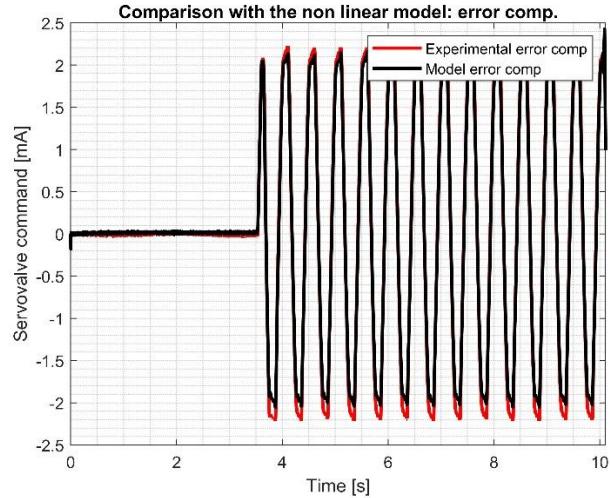


Figura 34 Comando servovalvola prova 2

Il comando della valvola sarà anch'esso sinusoidale per cercare di ovviare al disturbo e mantenere un set di forza nullo. Nella parte positiva della sinusoide del disturbo la valvola tenderà ad avere una luce di passaggio maggiore per compensare; vale il viceversa nel caso di disturbo negativo.

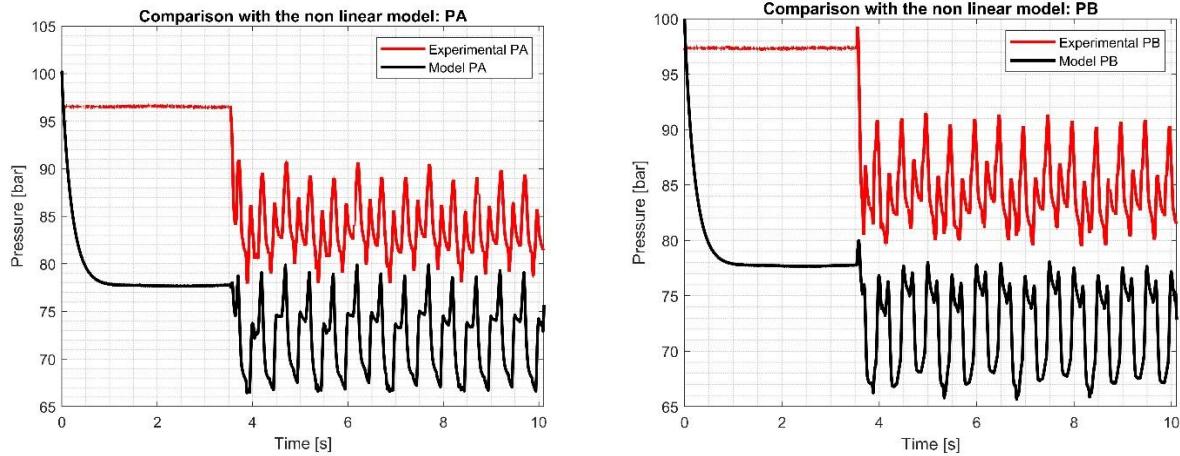


Figura 35 Pressione camera posteriore (PA) e camera anteriore (PB) prova 2

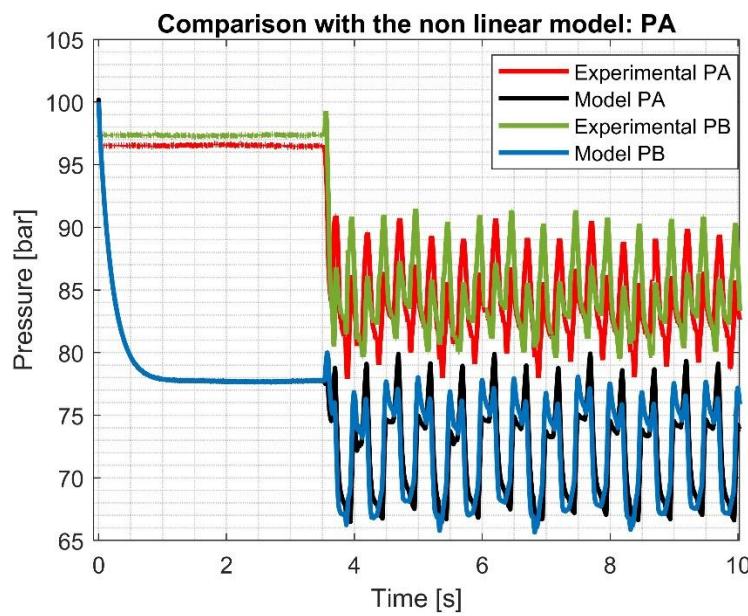


Figura 36 Confronto pressione prova 2

Dalla figura dove si mostra il confronto dell’andamento delle due pressioni si nota immediatamente come la differenza tra le due pressioni è minore rispetto a quella che si otteneva nella prova 1; questo potrebbe essere significativo in quanto mostra come il set in forza generi una differenza di pressione molto più ampia rispetto al disturbo il quale potrebbe essere indice di una buona rigidezza del sistema. L’andamento delle due pressioni è sinusoidale e quando sale la pressione della camera anteriore allora sale anche quella della camera posteriore per cercare di avere una differenza di pressione nulla in quanto il set in forza deve essere nullo.

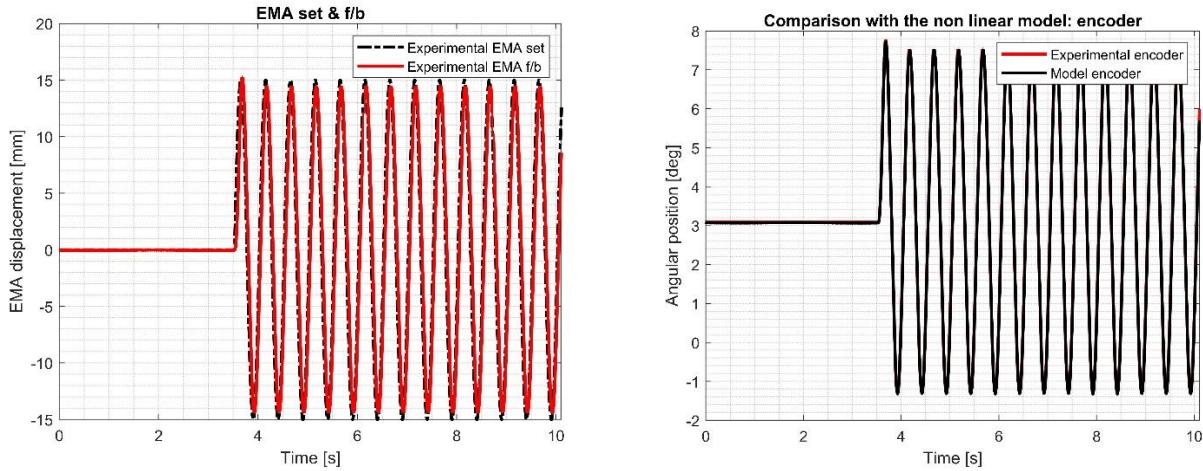


Figura 37 Set e feedback posizione dell'EMA prova 2

Si può vedere dalle figure appena sopra che l'EMA segue molto bene il set sinusoidale; cosa che ci si aspetta dato che la sinusoidale in ingresso ha una frequenza molto bassa e quindi si ha sfasamento praticamente nullo e guadagno pari a 1.

2.2.3 Prova 3

In questa prova si ha un set di forza ad onda quadra con disturbo sinusoidale in posizione.

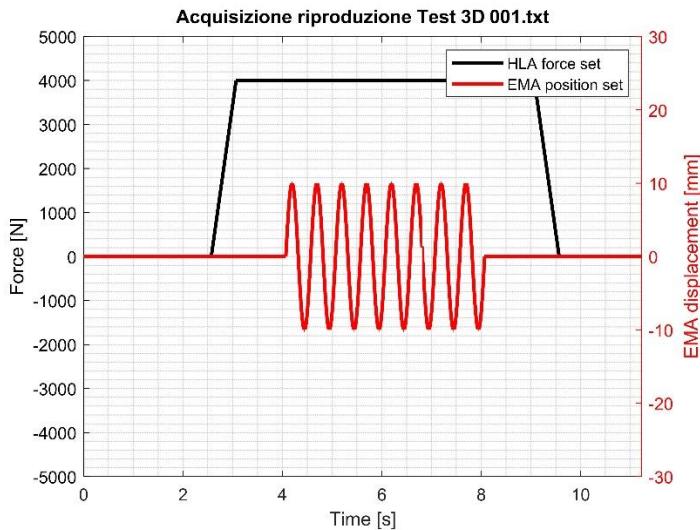


Figura 38 Set e disturbo prova 3

Dal feedback si vede come il modello e i dati sperimentali siano simili tra loro, tranne per l'inizio e la fine del disturbo in posizione dove i dati sperimentali hanno dei picchi che il modello non presenta. Anche in questo caso come è avvenuto per la prova 1, il segnale sperimentale ha un overshoot che il modello non presenta.

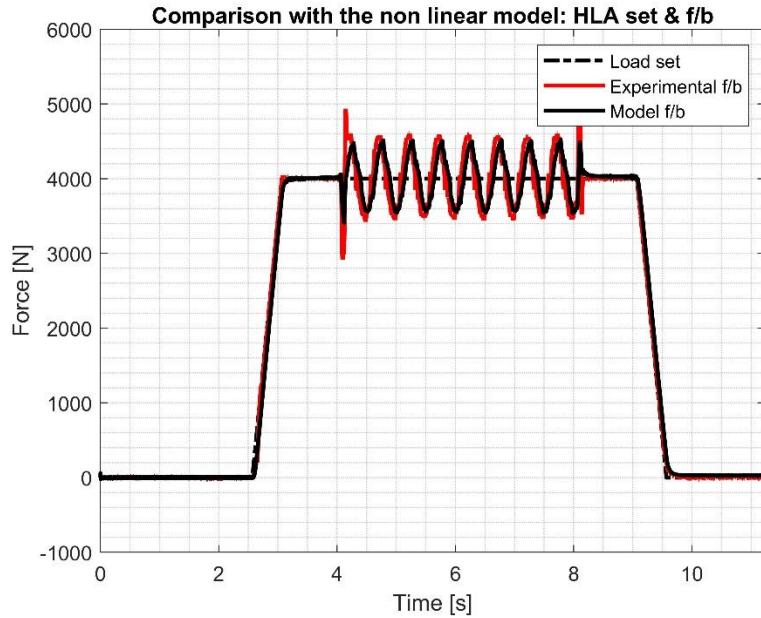


Figura 39 Confronto con il modello prova 3

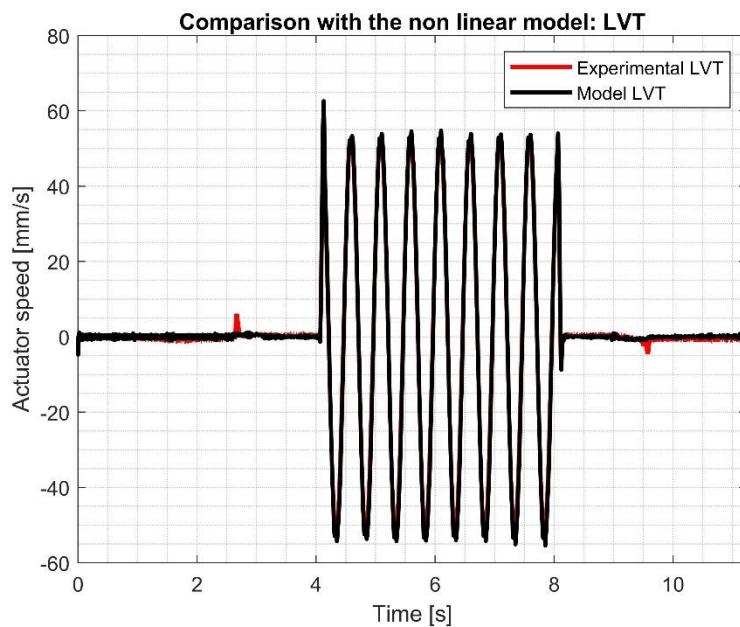


Figura 40 Velocità attuatore prova 2

Per quel che riguarda la velocità dell'attuatore si vede come segua abbastanza fedelmente l'andamento sinusoidale del disturbo. I dati sperimentali hanno due picchi intorno ai 3s e ai 9.5s; questi picchi sono dovuti alla dinamica dell'EMA. Quando inizia il set in forza l'EMA tende a non essere perfettamente fermo, essendo di rigidezza non infinita, e quindi si ha un leggero spostamento che poi l'EMA stesso si attiva per recuperare perché anche lui cerca di seguire il set in posizione che nel suo caso è un set disturbato in forza.

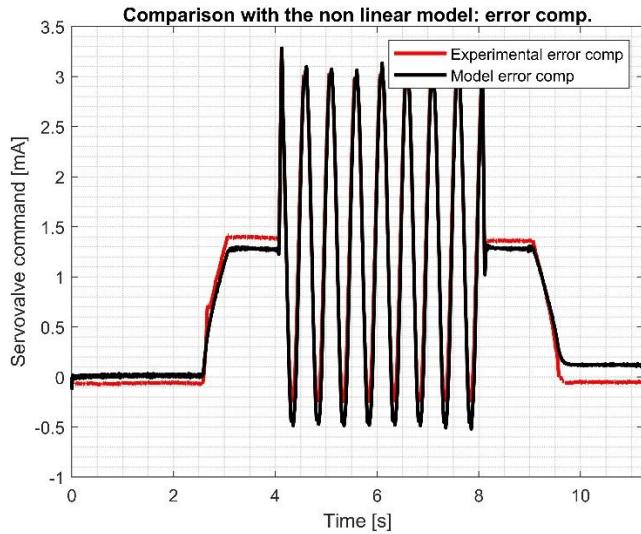


Figura 41 Comando servovalvola prova 3

Per quel che riguarda il comando della servovalvola si può notare che il modello e i dati sperimentali presentano un offset nel seguire il comportamento dell'onda quadra in forza mentre hanno lo stesso andamento quando entra il disturbo nel sistema.

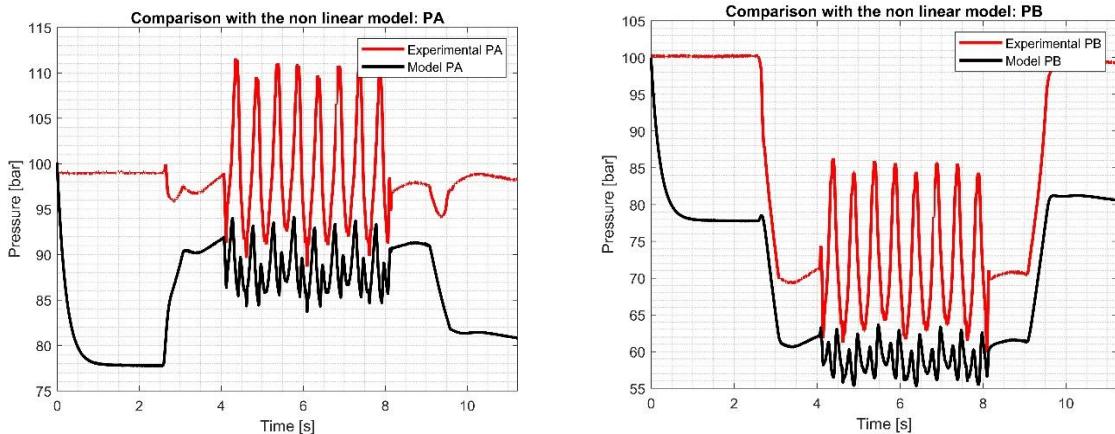


Figura 42 Pressione camera posteriore (PA) e camera anteriore (PB) prova 3

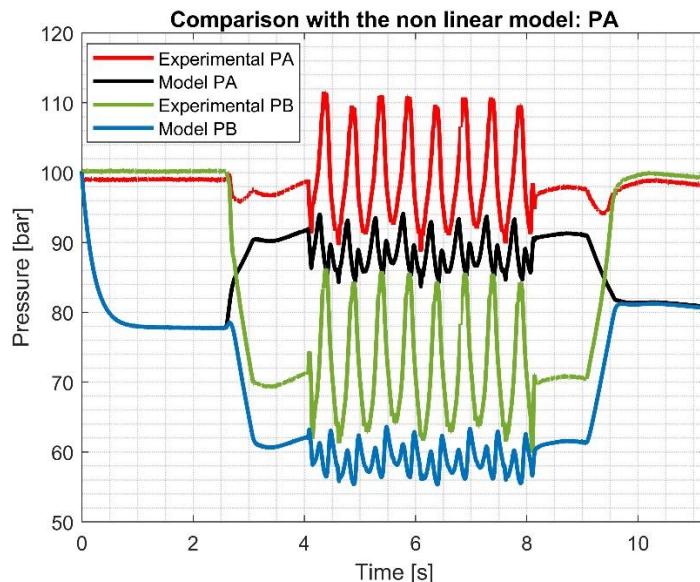


Figura 43 Confronto pressione prova 3

Nei dati sperimentali per riuscire a replicare il set in forza richiesto P_A rimane circa costante mentre P_B diminuisce bruscamente mentre per il modello l'aumento e la diminuzione di pressione delle due camere è pressoché equi ripartita tra le due. L'andamento sinusoidale che si ha quando è presente il disturbo è molto simili a quello che si è ottenuto nella prova 2 con il solo disturbo sinusoidale con set in forza nulla.

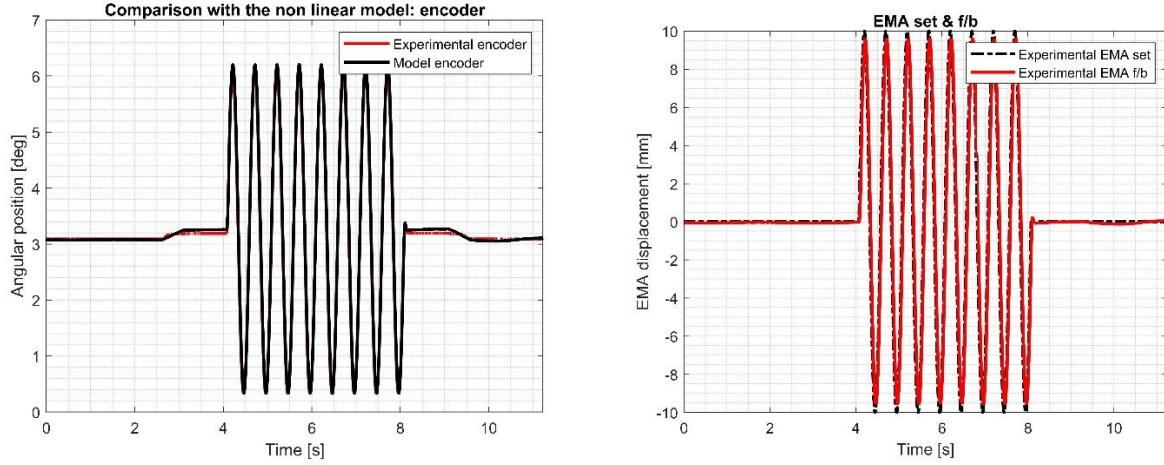


Figura 44 Set e feedback posizione dell'EMA prova 3

Da queste figure si vede che l'EMA segue bene il set sperimentalmente senza particolari problemi, si ha solo una leggerissima attenuazione che si può considerare trascurabile. Nel vedere la figura riguardante l'encoder si nota che durante il set di forza all'inizio della rampa si ha una leggera rotazione del giunto angolare che collega i due attuatori, ma risulta essere nell'ordine dei decimi di grado e quindi del tutto trascurabile.

2.2.4 Prova 4

In quest'ultima prova si ha un set di forza nullo e un disturbo a rampa a cui viene sommato un disturbo sinusoidale.

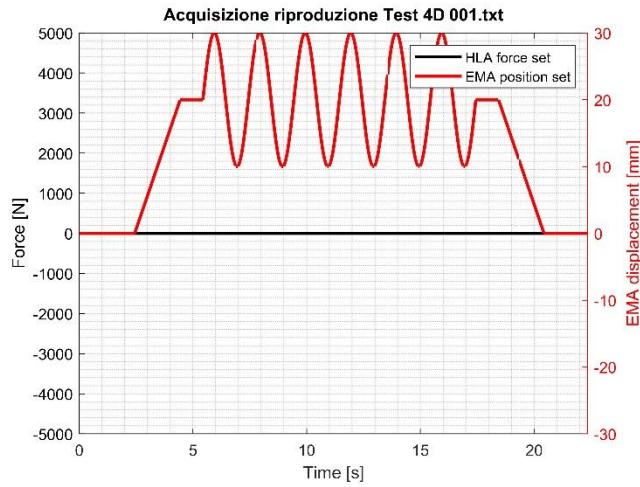


Figura 45 Set e disturbo prova 4

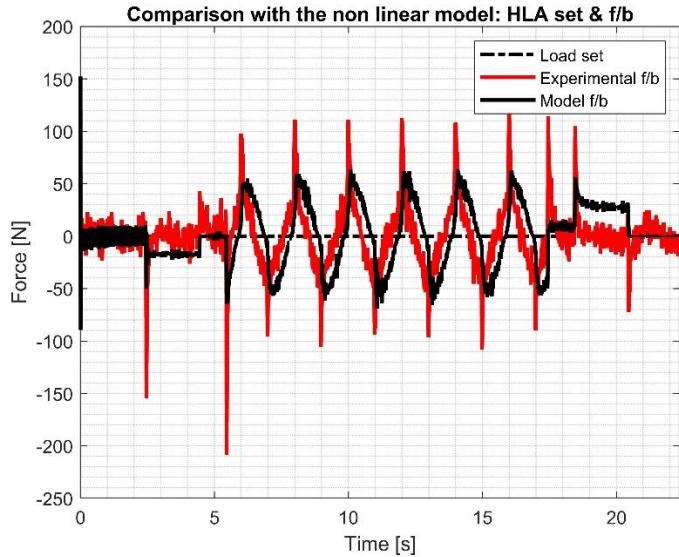


Figura 46 Confronto con il modello prova 4

L’andamento sinusoidale centrale è dovuto al disturbo sinusoidale ed è simili a quelli analizzati già in precedenza quindi ci si sofferma sulla salita e sulla discesa della rampa del disturbo. Qui si può vedere come il modello e i dati sperimentali presentino un comportamento diverso tra loro. I dati sperimentali oscillano molto nell’intorno di forza nulla mentre il modello lungo tutto il tratto di salita del disturbo si stabilizza ad un valore di forza negativo come se il sistema non riesca a rispettare il set in forza impostato.

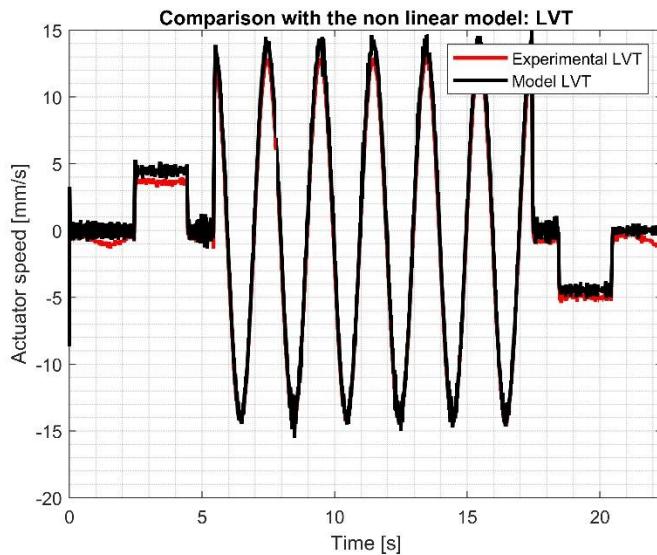


Figura 47 Velocità attuatore prova 4

Per la velocità dell’attuatore si ha un valore positivo quando il disturbo è nella fase di salita dell’onda quadra mentre si ha un valore nativo nella fase di discesa. Il modello e i dati sperimentali sono molto simili tra loro senza presentare particolari differenze.

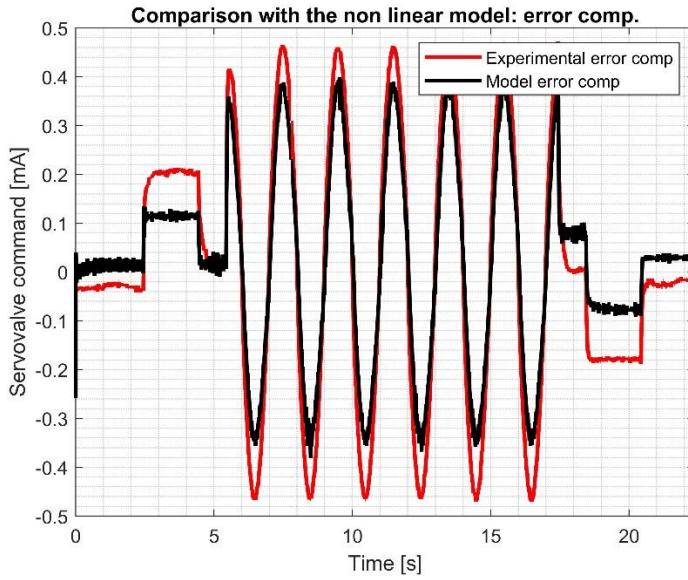


Figura 48 Comando servovalvola prova 4

Per il comando della valvola i dati sperimentali sembrano più influenzati dal disturbo; infatti, generano un'apertura o chiusura della valvola maggiore rispetto al modello. I dati sperimentali sembra che dicono che il sistema abbia una rigidezza minore al disturbo rispetto al modello.

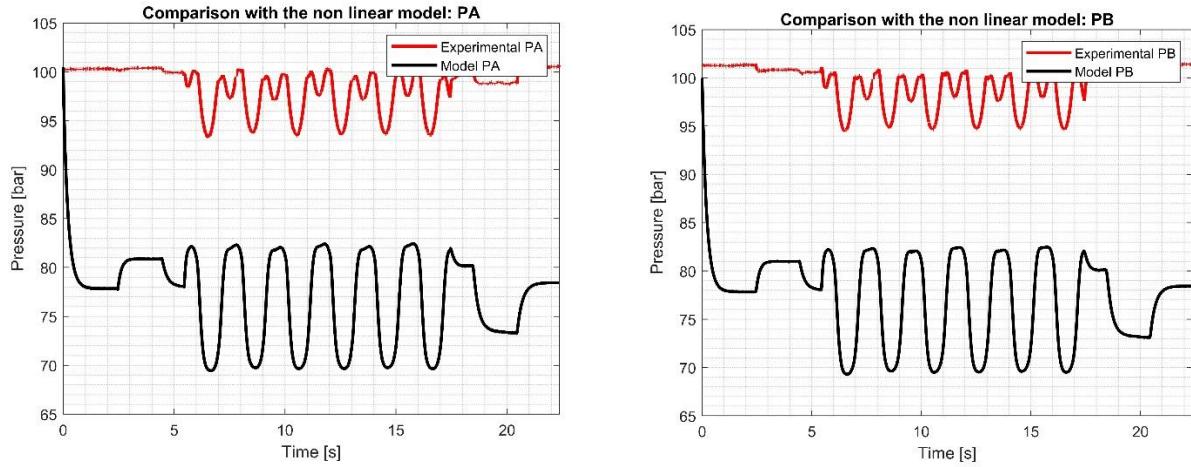


Figura 49 Pressione camera posteriore (PA) e camera anteriore (PB) prova 4

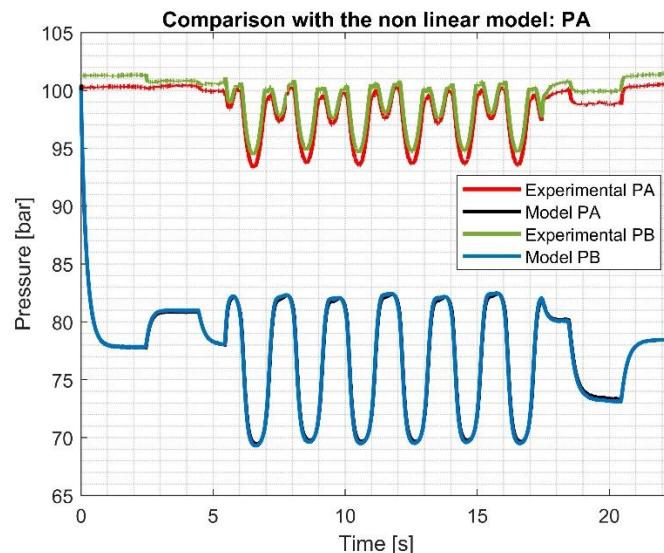


Figura 50 Confronto pressione prova 4

Nell'andamento delle pressione si riesce subito a vedere come cerchino di rimanere il più simile possibile tra loro in modo da generare il set di forza nullo. Le pressioni del modello sono praticamente sovrapposte mentre quelle sperimentali hanno una leggerissima differenza e questo può essere una spiegazione per la figura 46 dove i dati sperimentali hanno un'oscillazione maggiore intorno al valore di forza nulla rispetto al modello.

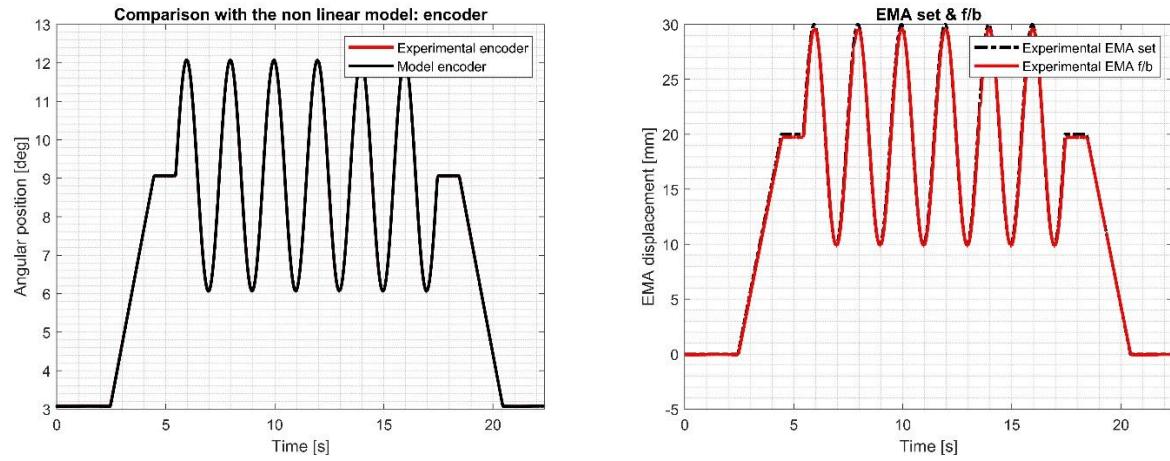


Figura 51 Set e feedback posizione dell'EMA prova 4

Per quel che riguarda l'EMA si vede come il set sia seguito molto fedelmente dal feedback anche per via dell'assenza del segnale in forza che presenta un disturbo per questo attuatore.

3 ISWEC banco Hardware in the Loop: generatore di energia da moto ondoso

3.1 Introduzione

ISWEC (Inertial Sea Wave Energy Converter) è un dispositivo per la produzione di energia dal moto ondoso del mare. L'esercitazione in esame mira ad analizzare il funzionamento di un banco prova Hardware In the Loop (HIL), utilizzato per testare un sistema per la produzione di energia tramite sfruttamento di moto ondoso. Una prima fase ha previsto l'identificazione della stima dei momenti d'inerzia del gruppo giroscopico; successivamente si è passati alla valutazione generale della performance del sistema. In quest'ottica, sono state effettuate diverse prove sperimentali, attribuendo a ciascuna di queste differenti parametri del sistema (coefficiente di smorzamento e di rigidezza) e osservando l'andamento di alcune grandezze caratteristiche del sistema in funzione di questi parametri.

La seconda parte, di tipo numerico, è invece dedicata alla valutazione dei momenti di inerzia del sistema attraverso il fitting di dati sperimentali acquisiti in laboratorio e al confronto dei risultati ottenuti in laboratorio con le prove sperimentali in vasca prova e con i risultati numerici.

La particolarità del banco è l'utilizzo della tecnica HIL: parte del sistema fisico viene simulato da un modello numerico che viene alimentato da informazioni e reazioni agli stimoli del sistema stesso. È una tecnica utilizzata in campo accademico ed industriale per ridurre tempi e costi di sistemi soprattutto in fase prototipale.

Le caratteristiche principali del sistema utilizzato sono riportate nella tabella seguente:

A photograph of the ISWEC prototype, showing a yellow cylindrical hull with a blue and white patterned section at the front. It is mounted on a black metal frame and connected to various wires and sensors.

Scafo	
Lunghezza	4 m
Larghezza	3.5 m
Massa	3800 kg
Giroscopio	
Massa	200 kg
Velocità angolare max.	1500 rpm
PTO	
Coppia max.	275 Nm
Coppia nominale	173 Nm
Velocità angolare nom.	30 rpm
Potenza nominale	540 W

Figura 52 Prototipo ISWEC e caratteristiche principali

Il processo per la produzione di energia può essere così riassunto: il moto ondoso del mare impone il moto di beccheggio allo scafo, la presenza di un volano in rotazione unito al moto di beccheggio genera una coppia giroscopica che agisce su asse sul quale è calettato un generatore elettrico che estraendo energia tende a smorzare questo movimento.

Si modelleranno dunque i seguenti sottosistemi:

- Scafo
- Interazione scafo- giroscopio
- Legge di controllo del generatore elettrico

3.2 Struttura ISWEC

Il sistema studiato, come detto in fase introduttiva, è composto da uno scafo stagno che contiene un sistema giroscopico ad un grado di libertà responsabile dell'estrazione di energia.

Ogni elemento del sistema presenta un angolo caratteristico utile per la modellazione del sistema matematico:

- Lo scafo viene identificato dall'angolo di beccheggio δ
- Il telaio viene identificato dall'angolo ε
- Il giroscopio ruota alla velocità costante imposta $\dot{\phi}$

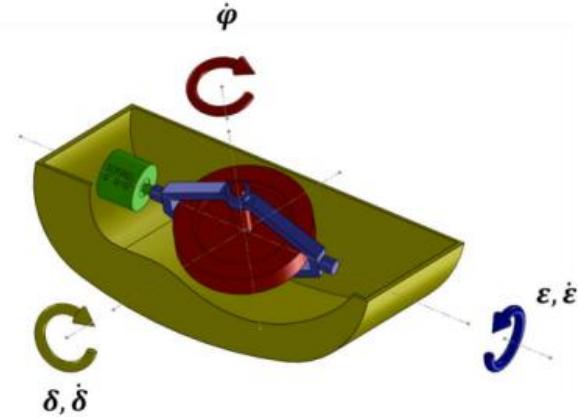


Figura 53 Schema ISWEC, angoli caratteristici

Le onde inducono sullo scafo un moto di beccheggio intorno all'asse δ . La velocità δ , combinata alla velocità angolare del volano $\dot{\phi}$, produce un momento giroscopico sul terzo asse e di conseguenza, una rotazione ε . La variazione del momento angolare (ed in particolare alla direzione del vettore momento angolare), dovuta al moto di beccheggio, induce una coppia giroscopica che può essere sfruttata dal generatore PTO (Power Take Off, ovvero “presa di potenza”), posizionato sull'asse di rotazione ε , per generare energia.

3.3 Struttura banco prova HIL

Il banco è costituito da un telaio oscillante sostenuto da supporti con cuscinetti, che viene attivato tramite l'utilizzo di un motoriduttore elettrico. Sulla piattaforma oscillante è installato il gruppo giroscopico presente sulla piattaforma ISWE, composto dal volano, contenuto all'interno della campana del vuoto per minimizzare le perdite aerodinamiche e posto in rotazione tramite un motore di azionamento. Questo gruppo è collegato ad un torsometro per misurare la coppia del generatore e al PTO stesso.

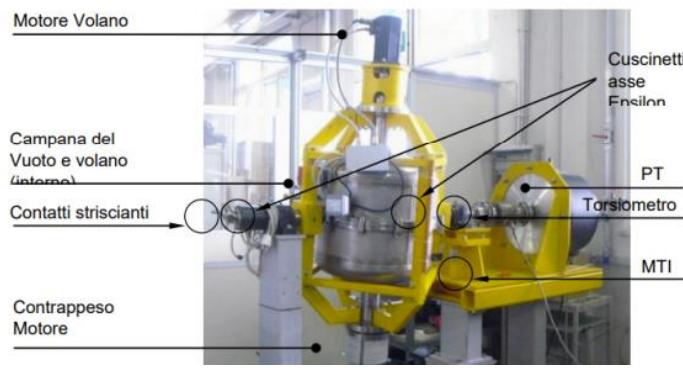


Figura 54 Banco prova HIL

Il grande vantaggio di questo sistema è la riduzione del tempo necessario per la sperimentazione in vasca di prova, operazione che può raggiungere costi molto elevati.

Con il banco è possibile riprodurre il moto di beccheggio dello scafo di ISWEC e verificare come risponde il sistema meccanico in una qualsiasi configurazione di input ondoso aggiustando due parametri statistici:

- il periodo dell'onda
- l'altezza dell'onda

La possibilità di sperimentare numerose configurazioni di input permette di identificare i parametri che massimizzano la potenza prodotta nella maggior parte delle condizioni operative ed identificare il miglior compromesso tra le variabili analizzate.

3.4 Identificazione **I** e **J**

Il primo passaggio fondamentale da eseguire è il calcolo attraverso un codice python dei due momenti di inerzia che caratterizzano il sistema dove:

- I è il momento di inerzia del gruppo giroscopico intorno a epsilon;
- J è il momento di inerzia per il solo volano

Si effettua l'identificazione di I e J minimizzando diverse relazioni del sistema.

3.4.1 Identificazione di **I** e **J** minimizzando la potenza

In questo paragrafo l'identificazione di I e J avviene minimizzando l'errore sulla potenza

- Si lascia come parametro libero solo I

Lasciando come parametro libero solo I questi sono i risultati che si ottengono su tutte le prove:

PROVA	I
Prova 1	42,875
Prova 2	44,625
Prova 3	43,094
Prova 4	42,875
Prova 5	42
Prova 6	42
Prova 7	43,75
Prova 8	46,375
Prova 9	42,875

Tabella 1 Momento inerzia I (da potenza)

- Si lascia come parametro libero sia I che J

Lasciando come parametro libero sia I che J questi sono i risultati che si ottengono su tutte le prove:

PROVA	I	J
Prova 1	42,972	5,414
Prova 2	45,217	5,376
Prova 3	42,859	5,636
Prova 4	42,947	5,585
Prova 5	43,312	5,5
Prova 6	42,328	5,621
Prova 7	43,814	5,123
Prova 8	45,578	4,981
Prova 9	42,01	5,263

Tabella 2 Momenti di inerzia I e J (da potenza)

3.4.2 Identificazione di **I** e **J** minimizzando l'errore sulla posizione

- Si lascia come parametro libero solo I

Lasciando come parametro libero solo I questi sono i risultati che si ottengono su tutte le prove:

PROVA	I
Prova 1	42
Prova 2	43,75
Prova 3	43.75
Prova 4	43.75
Prova 5	36.75
Prova 6	42

Prova 7	43.75
Prova 8	43.75
Prova 9	42

Tabella 3 Momento di inerzia I (da spostamento)

- Si lascia come parametro libero sia I che J

Lasciando come parametro libero sia I che J questi sono i risultati che si ottengono su tutte le prove:

PROVA	I	J
Prova 1	42,438	5,438
Prova 2	51,119	5,351
Prova 3	42,437	5,438
Prova 4	42,437	5,438
Prova 5	36,75	5
Prova 6	36.75	5
Prova 7	42,437	5,437
Prova 8	42,378	5,378
Prova 9	42,375	5,38

Tabella 4 Momenti di inerzia I e J (da spostamento)

3.5 Grafici 3D

In questa sezione vengono generati e si analizzano dei grafici 3D che mostrano in base alle 9 coppie possibili di smorzamento e rigidezza come cambiano alcuni parametri caratteristici del sistema.

3.5.1 Grafico 3D sulla potenza

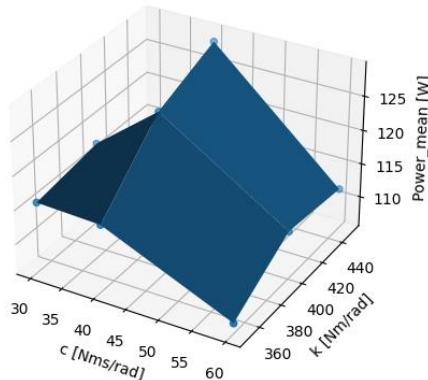


Figura 55 Grafico 3D potenza media

Il primo grafico da analizzare è quello sulla potenza media prodotta al variare dello smorzamento e della rigidezza. Dalla figura si vede immediatamente come il punto con una potenza maggiore è quello con $k=450$ n/mm e $c=40$ n*s/m; questi valori di c e k generano il miglior compresso per la creazione di potenza lungo tutto il periodo di oscillazione del sistema. In qualche modo si cerca di avere un sistema che possa andare in risonanza per far sì che l'ampiezza delle oscillazioni dell'albero sia più ampia di quelle delle onde e quindi in modo da generare una potenza maggiore. La potenza massima potrebbe anche corrispondere ad una coppia di valori diversi ma avere il picco di potenza più alto non è sempre la cosa migliore perché anche altri fattori vanno considerati.

3.5.2 Grafico 3D sulla coppia

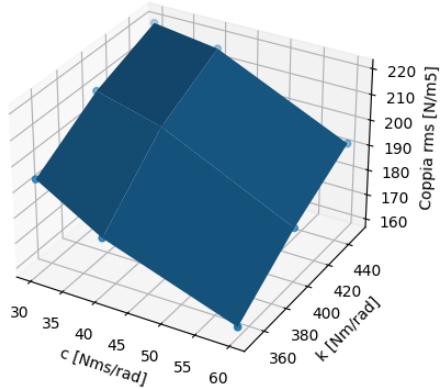


Figura 56 Grafico 3D coppia

Dalla figura appena mostrata si nota come la coppia massimo si ha in corrispondenza del valore massimo di rigidezza e del valore minimo di smorzamento. La coppia, come si può vedere dal grafico cresce all'aumentare della rigidezza e al diminuire dello smorzamento. La coppia risulta un parametro importante da tenere elevato in modo da estrarre più potenza attraverso la componente frenante del generatore elettrico.

3.5.3 Grafico 3D sullo spostamento

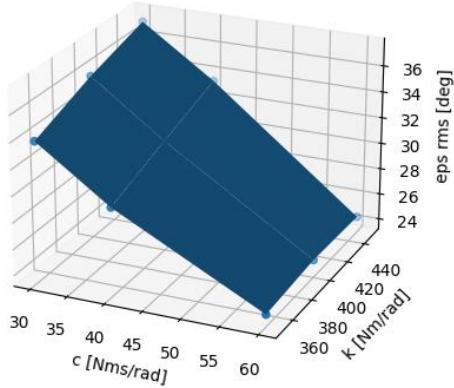


Figura 57 Grafico 3D sullo spostamento

Il punto che genera un'oscillazione maggiore è quello con rigidezza più alta e smorzamento più basso. Lo smorzamento tende a frenare la rotazione lungo l'asse del generatore e quindi è chiaro che avere smorzamenti bassi garantisce un periodo di oscillazione più lungo. La rotazione è un parametro molto importante perché il sistema è necessario che ruoti intorno all'asse del generatore non per un tempo limitato ma il più lungo possibile senza però perdere eccessivamente coppia.

Risulta quindi evidente da questo sottoparagrafo e da quelli precedenti che trovare la combinazioni di valori ottimali non è semplice; bisogna cercare un compromesso per probabilmente non avere la coppia massima o la rotazione massima ma per far sì che il sistema risulti il più bilanciato possibile.

3.6 Risultati del fitting dei dati sperimentali

Come ultima cosa è necessario confrontare i dati sperimentali con i dati del modello in modo da capire se il modello risulta essere un buon fitting dei dati sperimentali.

Confrontando i dati sperimentali e simulati della potenza si ottiene:

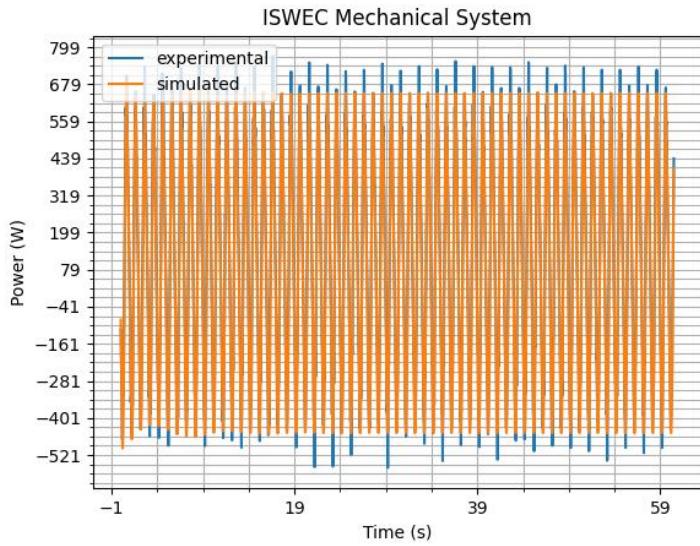


Figura 58 Potenza prodotta dal banco

Come si può vedere dalla figura appena mostrata, le due potenze non sono uguali e questo è dovuto alla mancanza dell'attrito nei dati simulati. Infatti, la simulazione è stata fatta senza considerare l'attrito viscoso ovviamente presente durante la rotazione dell'albero coassiale all'asse del generatore. Per il calcolo della potenza con il modello sono state utilizzate le seguenti formule:

$$P = T_\varepsilon \dot{\varepsilon} = -c\dot{\varepsilon}^2 - k\varepsilon\dot{\varepsilon}$$

(Che rappresenta la potenza estratta istantaneamente)

$$T_\varepsilon = -c\dot{\varepsilon} - k\varepsilon$$

(che è la coppia che agisce sul generatore)

Seguono i grafici di alcune grandezze analizzate. Nei diversi grafici vengono messi a confronto i dati sperimentali con un modello realizzato in python (curva arancione) e dei dati teorici di un modello sviluppato in Matlab (curva verde). I due modelli, avendo impostato condizioni iniziali pari a zero per ogni variabile necessitano di un transitorio prima di essere confrontabili con i dati sperimentali.

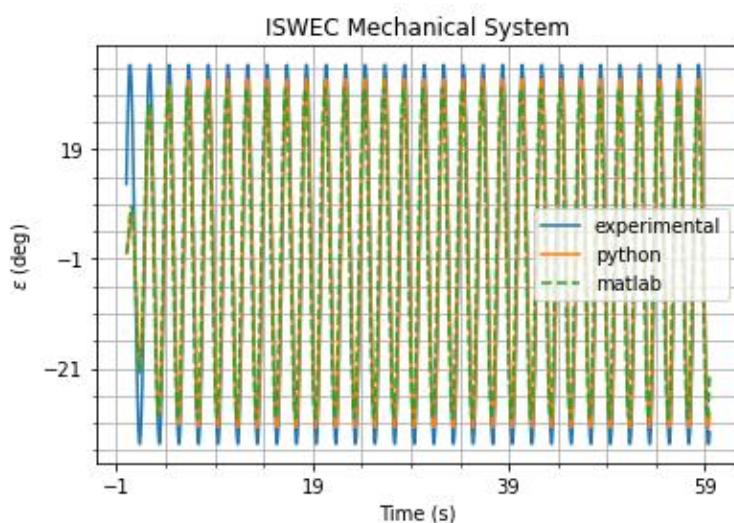


Figura 59 Eps, posizione angolare asse del PTO

La posizione angolare è correttamente riprodotta dai modelli, con l'unica differenza nel valore massimo e minimo dei picchi, leggermente inferiori per entrambe le simulazioni.

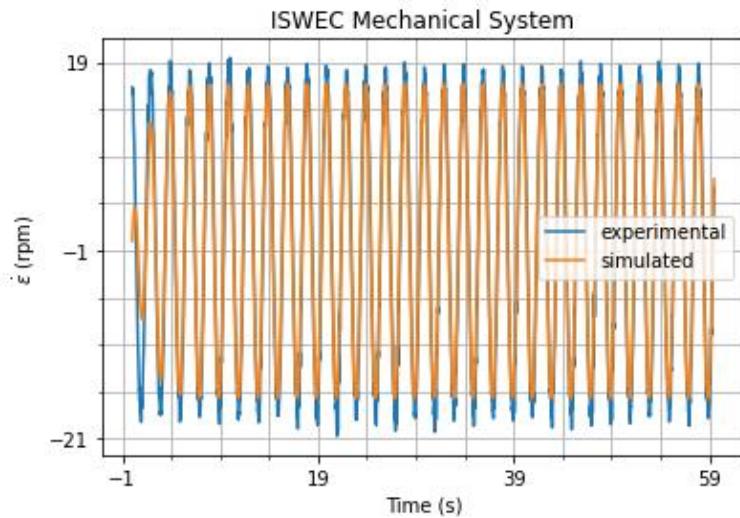


Figura 60 Eps dot, velocità angolare asse del PTO

La velocità angolare è paragonabile in entrambe le curve. Si può notare la presenza di maggiori disturbi anche nei dati sperimentabili poiché la grandezza potrebbe essere ricavata effettuando una derivata a partire dalla posizione.

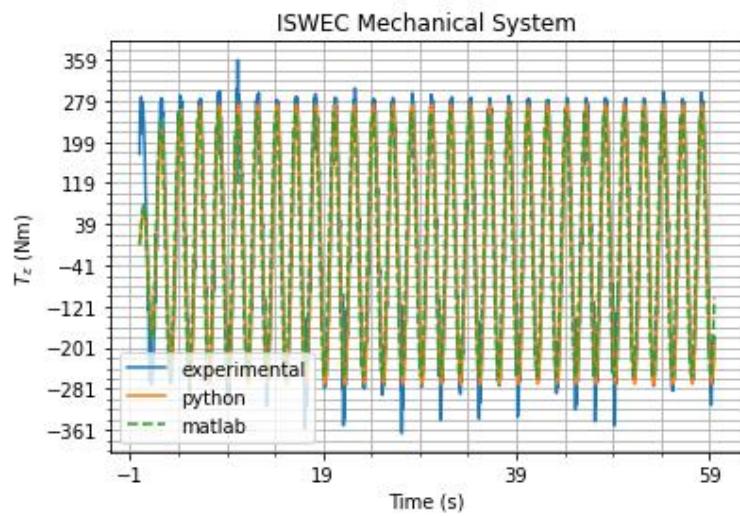


Figura 61 T eps, coppia sull'asse del PTO

La coppia T_{eps} è riprodotta correttamente da entrambi i modelli sia come andamento temporale sia come valore di picco.

Il prodotto tra la velocità angolare e la coppia determina la potenza istantanea estratta dal PTO. All'interno del periodo solo determinate zone producono energia utile, in particolare quando coppia e velocità angolari sono concordi in segno. È anche per questo motivo che per valutare la potenza prodotta si utilizza il metodo del root mean square (rms) per ricavare il valore corrente a fronte di un segnale sinusoidale (positivo per metà del periodo e negativo per il resto).

4 Giunto di forza a 6 assi

4.1 Introduzione

4.1.1 Obiettivi

L'attività di laboratorio è stata condotta per riprogettare e migliorare un sistema di acquisizione di forze e momenti che utilizza una piattaforma di Stewart a sei assi. La piattaforma è stata poi utilizzata per determinare i parametri di un sistema molla-massa-smorzatore collegato alla sua estremità grazie ad un fitting dei dati sperimentali raccolti.

Variando la frequenza è stato successivamente verificato il teorema di Nyquist, ed è stato implementato un sistema di image processing capace di ricavare gli stessi parametri ottenuti in precedenza, grazie ad immagini registrate da una videocamera.

4.1.2 Introduzione teorica

La piattaforma di Stewart è essenzialmente composta da due dischi paralleli tra loro, collegati rigidamente tramite sei bracci incernierati ad entrambi gli estremi. Ipotizzando un attrito nullo in corrispondenza delle coppie rotoidali, è possibile affermare che i bracci siano soggetti a sforzi puramente assiali, misurati tramite delle celle di carico montate lungo il loro stelo.

Alla piattaforma è stata poi fissata una massa tramite una molla, come mostrato in Figura 1 e 2.



Figura 62 – Sistema di misura a sei assi

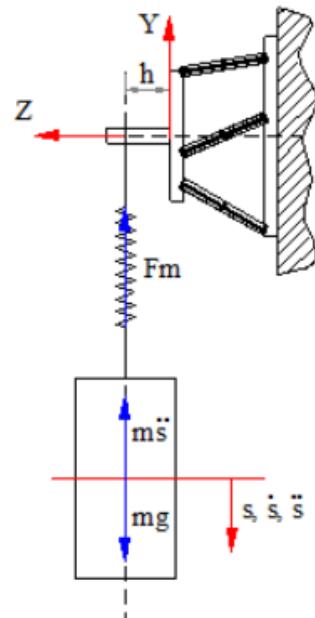


Figura 63 – Configurazione sistema fisico

Grazie alla particolare geometria della piattaforma, è possibile convertire le forze acquisite tramite la strumentazione in terne di forze e coppie di un sistema di riferimento cartesiano la cui origine è fissata ad una distanza "h" dal disco più esterno, lungo il suo asse.

La conversione avviene tramite una matrice di dimensione 6×6 opportunamente generata con le dimensioni del sistema.

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} -n & n & -\frac{\sqrt{3}}{2}m + \frac{1}{2}n & -\frac{\sqrt{3}}{2}m - \frac{1}{2}n & \frac{\sqrt{3}}{2}m + \frac{1}{2}n & \frac{\sqrt{3}}{2}m - \frac{1}{2}n \\ m & m & -\frac{1}{2}m - \frac{\sqrt{3}}{2}n & -\frac{1}{2}m + \frac{\sqrt{3}}{2}n & -\frac{1}{2}m + \frac{\sqrt{3}}{2}n & -\frac{1}{2}m - \frac{\sqrt{3}}{2}n \\ q & q & q & q & q & q \\ -qa & -qa & \frac{1}{2}qa & \frac{1}{2}qa & \frac{1}{2}qa & \frac{1}{2}qa \\ 0 & 0 & -\frac{\sqrt{3}}{2}qa & -\frac{\sqrt{3}}{2}qa & \frac{\sqrt{3}}{2}qa & \frac{\sqrt{3}}{2}qa \\ -na & na & -na & na & -na & na \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{bmatrix}$$

Figura 64 – Matrice di conversione

Grazie ad una scheda di acquisizione sono stati quindi campionati i dati di forza derivanti dalle celle di carico, e salvati in un file già nel sistema cartesiano.

Dopo aver ottenuto i dati cartesiani, è stato possibile elaborarli per stimare i seguenti parametri:

- y_0 ampiezza dell'oscillazione
- ω_d pulsazione smorzata del sistema
- φ fase del segnale
- ξ fattore di smorzamento
- ω_n pulsazione naturale

La stima è stata effettuata tramite il software Matlab, utilizzando la funzione "fminsearch" che verrà presentata in seguito. È anche stato utilizzato il software Simulink per creare un programma con le stesse finalità del primo, che elabora gli stessi dati in ingresso.

4.1.3 Materiali e Strumenti

Il sistema di misura è composto da:

- Piattaforma di Stewart
- 6 celle di carico Flintec Y1 – 50

Esso invia dati ad un sistema di acquisizione che li converte in formato digitale, composto da:

- 6 condizionatori Flintec FAA-28
- Chassis Ni cDAQ-9171 con scheda di acquisizione NI-9205



Figura 66 – Condizionatori e scheda di acquisizione

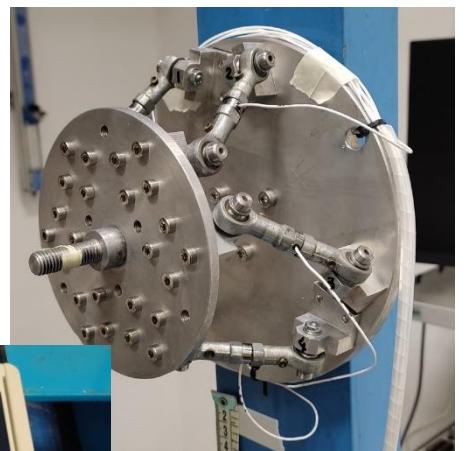


Figura 65 – Piattaforma di Stewart

Il modello in prova è poi composto di una massa ed una molla metallica.

I dati ottenuti sono poi elaborati grazie ai software:

- LabView, per l'acquisizione dei dati
- Matlab per la stima dei parametri e l'image processing
- Simulink utilizzato in alternativa per la stima dei parametri

4.1.4 Setup sperimentale e Svolgimento metodologico

L'attività di laboratorio è stata condotta a partire dall'allestimento del banco fisico. Le celle, grazie ai due codoli filettati, sono state fissate nella mezzeria dei bracci della piattaforma di Stewart.

È sorto in questo caso un problema che è stato risolto montando le celle in modo da lasciare spazio tra il loro anello esterno (esse sono di forma cilindrica) e la superficie del braccio. Ciò perché le celle sono capaci di misurare forze trasmesse solo tramite la filettatura, ma in questo caso avvitandole sui bracci si generava un contatto tra la faccia trasversale del braccio e l'anello esterno delle celle: la forza veniva quindi trasmessa tra le due superfici non permettendo una misurazione.



Figura 67 – Particolare montaggio delle celle

Una volta allestito

il banco, sono stati

cablati i componenti per connettere le celle al computer: i cavi provenienti dalle singole celle sono stati collegati ai rispettivi condizionatori. Tali strumenti sono capaci di modificare il segnale fornito dalle celle in bassa tensione (dell'ordine del mV) in una tensione variabile nel range 0 V – 10 V.

È in questo caso necessario una regolazione del guadagno e del segnale di zero dei condizionatori, operato tramite delle viti presenti sulla loro carcassa.



Figura 68 – Viti di regolazione

In Figura 7 sono mostrate le viti di selezionata la grandezza in output

regolazione: in basso è (tensione nel caso presentato), la vite "adj" è invece utilizzata per modificare i valori del guadagno e del segnale di zero, se l'altra è impostata su "G" o "Z".

Il segnale così ottenuto è elaborato dalla scheda di acquisizione che lo converte e lo invia al computer.

Parallelamente a quest'attività, sono stati sviluppati i programmi di acquisizione in LabView, ed elaborazione dei dati in Matlab e Simulink.

I programmi di elaborazione sono stati generati grazie al modello fisico del sistema, ed alle equazioni che lo governano. Esse verranno presentate in seguito.



Figura 69 – Schema del sistema di acquisizione

4.2 Implementazione LabVIEW

L'acquisizione e l'elaborazione dei segnali viene gestita mediante un apposito programma sviluppato in ambiente LabVIEW. A livello concettuale, il programma può essere descritto con riferimento allo schema di Figura 9; in particolare, le funzionalità previste sono le seguenti:

- creazione di un task di acquisizione, ovvero predisposizione della scheda di acquisizione alla misura dei segnali di interesse;
- lettura/caricamento dei segnali di zero: questa funzionalità si rende necessaria in quanto si vogliono misurare solamente le forze sulla struttura dovute alla presenza della massa oscillante; pertanto, è necessario rimuovere gli offset di tensione (“segnali di zero”) dovuti al peso proprio della struttura e misurati quando la piattaforma non è caricata con la massa oscillante;
- impostazione dei parametri di prova, ovvero durata di acquisizione e frequenza di campionamento;
- acquisizione dei segnali provenienti dal condizionatore;
- elaborazione dei segnali acquisiti (rimozione offset, conversione dei segnali in tensione in forze, calcolo di forze e momenti rispetto al sistema di riferimento della piattaforma);
- visualizzazione dell’andamento temporale dei segnali in tensione, delle forze e dei momenti;
- salvataggio dei dati per successive analisi.

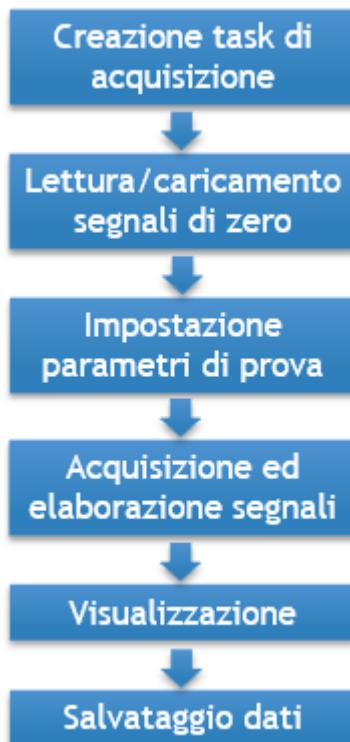


Figura 70 – Schema di funzionamento del codice LabVIEW sviluppato

Il codice di acquisizione, unitamente a tutti i sottoprogrammi (subVI) necessari al suo funzionamento, è contenuto nel file di progetto “ProgettoDAQmx.lvproj”. L’albero di progetto si presenta come mostrato in Figura 10; nello specifico, il file di progetto contiene il codice “Acquisizione Dati.vi”, che rappresenta il codice principale che deve essere eseguito per gestire l’acquisizione e l’elaborazione dei segnali. Questo codice richiama al suo interno i seguenti sottoprogrammi:

- “Lettura dati di zero (SubVI).vi”;

- “Waveform To Matrix (SubVI).vi”;
- “Build Conversion Matrix (subVI).vi”;
- “Rearrange Signals (SubVI).vi”.

Nel seguito si descriverà in dettaglio il funzionamento dei sottoprogrammi e del codice principale.

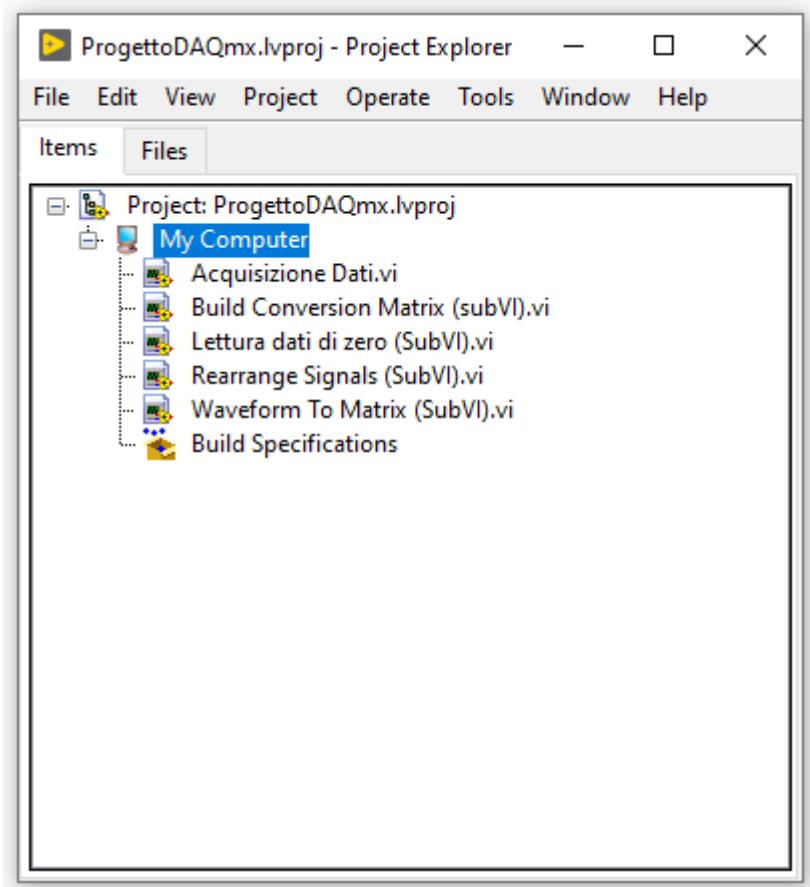


Figura 71 – Albero di progetto del programma di acquisizione

4.2.1 Lettura dati di zero (SubVI).vi e Waveform To Matrix (SubVI).vi

Si è scelto di eseguire la lettura dei segnali di zero della piattaforma di Stewart direttamente prima dell’acquisizione dati vera e propria per velocizzare le operazioni. Il codice che realizza la lettura dei dati di zero viene quindi realizzato in seno alla VI che esegue l’acquisizione dati, nella forma di una sub-VI per rendere il codice più snello e leggibile.

Il compito della Sub-VI è quindi quello di ricevere in ingresso un set di canali analogici per la lettura dei dati tramite la cDAQ e un percorso di salvataggio per restituire in uscita una matrice 6x1 contenente i segnali di zero acquisiti e mediati su ogni canale. I valori mediati devono anche essere salvati in un file “.txt”.

Alla fine di questo capitolo è riportata anche una cattura del front panel nella TAB “Lettura segnali di zero” di cui gli elementi che compaiono a schermo sono figli del modo in cui è scritto il codice della Sub-VI.

4.2.2 Creazione e inizializzazione del task per la lettura dei segnali di zero

Per discutere il codice della lettura dei dati di zero si inizierà da sinistra verso destra seguendo, ove possibile, l'ordine di esecuzione del codice. Per seguire meglio il filo del discorso si consiglia di fare riferimento alle immagini che riportano gli screenshot del codice LabView, la prima delle quali è la Figura 12.

In prima battuta si analizza la function “DAQmx Create Virtual Channel (VI)”, la più a sinistra, dalla quale inizia il percorso dell’errore. Con questa VI si crea il task che verrà usato nella sub-VI. Gli input forniti alla VI sono:

- la modalità di lettura, differenziale in questo caso, poiché la lettura avviene in maniera differenziale tra due “pin” analogici della cDAQ usata per l’acquisizione dati del banco. I pin usati per la lettura sono quelli che vanno dallo zero al cinque con i relativi differenziali come riportato in Figura 11;
- il valore massimo e minimo di voltaggio dei segnali da leggere, questi dati vengono importati dalla VI principale quando la Sub-VI viene richiamata;
- l’identificativo dei canali di acquisizione reali della cDAQ, anche questi sono importati dalla VI principali.

In uscita dalla prima VI ci sono il task appena creato che servirà per la lettura dei dati e il percorso di gestione dell’errore. Dopo aver creato il task viene quindi richiamata la DAQmx Timing (VI) che configura il numero di campioni che devono essere acquisiti per ogni canale avendo impostato l’acquisizione di un numero di campioni finiti. Si imposta inoltre una frequenza di acquisizione di 1000 Hz per velocizzare l’acquisizione.

In basso a sinistra nella Figura 12 si può inoltre osservare che il LED presente nel TAB “Segnali di zero” della VI principale viene impostato a un valore “false”, ovvero spento. Quando la lettura dei segnali di zero sarà completata il LED verrà acceso.

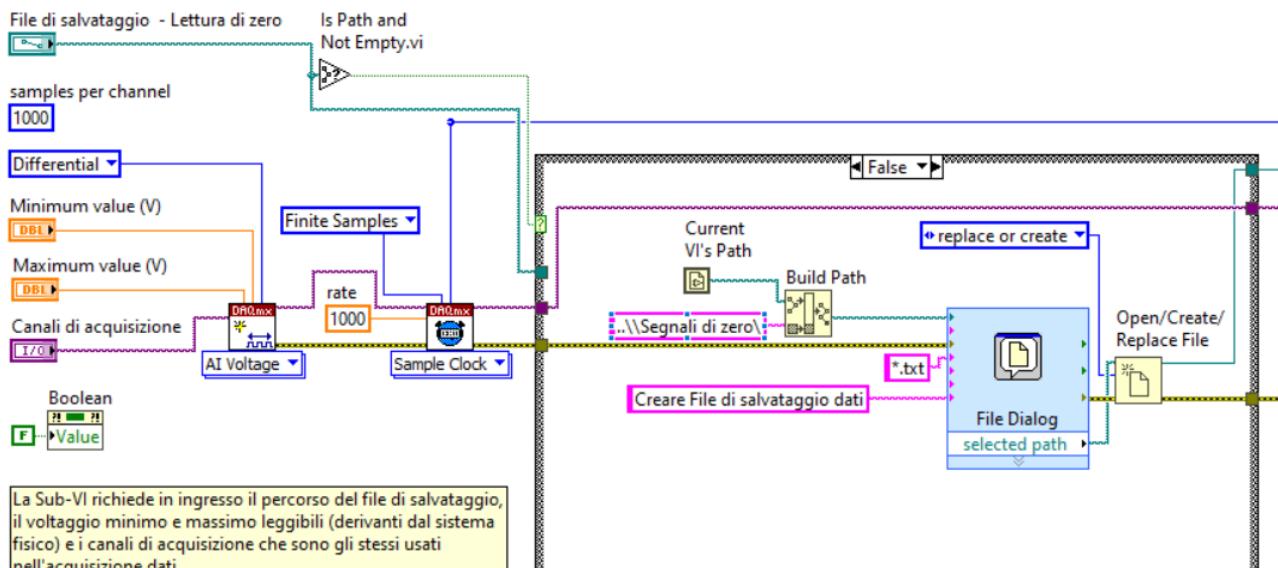


Figura 73 – Screenshot della prima parte del codice LabView della sub-VI “Lettura segnali di zero”, caso del “Case structure” FALSE

NI 9205



Figura 72 – Schema dei pin analogici della cDAQ NI 9205

Commentiamo ora il funzionamento del “Case structure” posto subito a valle del codice appena commentato. Nella Figura 12 è possibile osservare la condizione FALSE del “Case Structure”. Il codice riportato in questa condizione viene eseguito solamente quando il percorso letto dalla casella di scelta del percorso di salvataggio

(tutto in alto a sinistra in Figura 12) è vuoto o non è un percorso valido. Questa condizione viene analizzata tramite la “Is path and Not Empty.vi” che ritorna un valore TRUE solo se il percorso impostato è valido, in caso contrario riporta il valore FALSE, facendo eseguire il codice riportato in Figura 12.

4 “Case structure” FALSE: Percorso di salvataggio vuoto o non valido

Commentiamo quindi il codice eseguito se il percorso di salvataggio non è valido. Come prima cosa viene letto il percorso in cui si trova la Sub-VI della lettura dei segnali di zero tramite la function “Current VI’s path”, a questo viene accodata la cartella Segnali di zero attraverso la function “Build Path”. Il percorso di salvataggio scelto per il file viene quindi dato in ingresso alla VI “File Dialog” che forza l’uscita di una finestra di dialogo in cui l’utente deve scegliere un file di salvataggio valido nel percorso impostato. Alla VI “File Dialog” viene inoltre impostato che il file deve essere un “.txt” e che deve nominare la finestra di dialogo “Creare file di salvataggio dati”. Il percorso di salvataggio inserito dall’utente viene quindi dato in ingresso alla function “Open/create/replace file” che, impostando manualmente la condizione “replace or create” crea o sovrascrive il file “.txt” scelto.

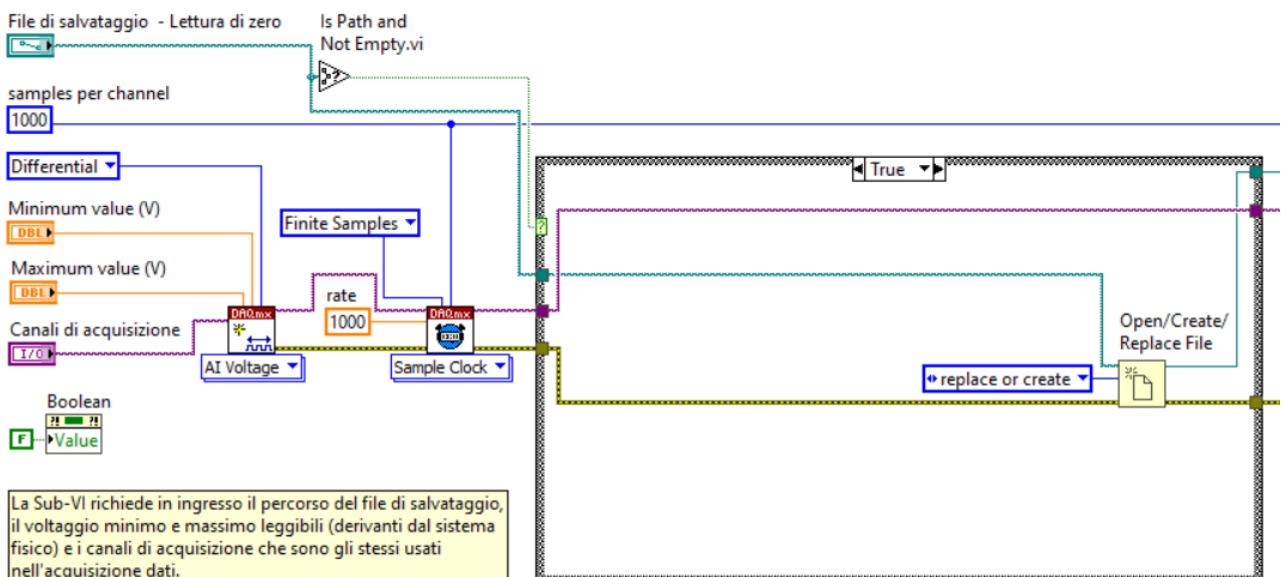


Figura 74 – Screenshot della prima parte del codice LabView della sub-VI “Lettura segnali di zero”, caso del “Case selector” TRUE

5 “Case structure” TRUE: Percorso di salvataggio valido

Se invece il percorso di salvataggio è valido viene eseguita un’alternativa al codice già commentato. Nello specifico si entra direttamente nella function “Open/create/replace” per creare o sostituire il file. Il codice nel caso si entri nella “Case structure” con valore TRUE è riportato in Figura 13.

4.2.3 Acquisizione dei segnali di zero

Passiamo ora al commento del secondo blocco del codice, quello successivo all’apertura del file su cui salvare le letture di zero. Una visione d’insieme di questo secondo blocco può essere vista in Figura 14.

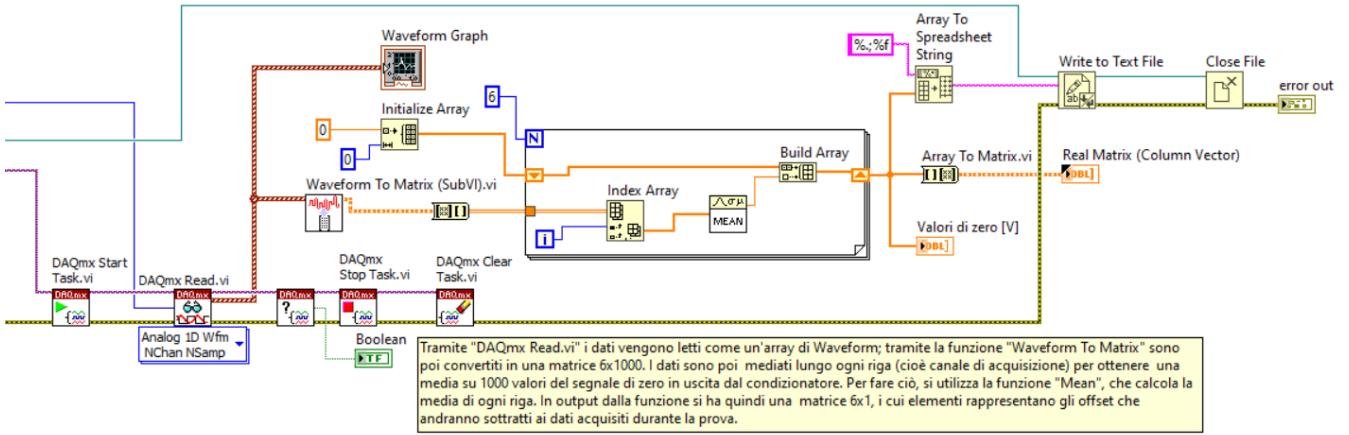


Figura 75 – Secondo blocco del codice della Sub-VI “Lettura dati di zero”

Per poter commentare più agevolmente il codice si riporta in Figura 15 un primo zoom. In questa sezione si analizza la gestione del task e quindi della lettura dei segnali di zero vera e propria.

In uscita dall’apertura del file di salvataggio viene eseguita in primis la VI “DAQmx Start Task.vi” che attiva il task creato a inizio codice iniziando l’acquisizione delle misure. In seguito, si passa alla “DAQmx Read.vi” che legge un segnale in forma di “waveform” per ogni canale di acquisizione analogico presente nel task, nel nostro caso sei. Seguendo sempre l’ordine del percorso del task (in viola) si passa alla “DAQmx Is Task Done.vi” che restituisce un valore TRUE se il task è terminato, andando ad accendere il LED presente nel TAB “Segnali di zero” a segnalare che l’acquisizione è stata completata. A questo punto si passa alla “DAQmx Stop Task.vi” che ferma il task riportandolo alla condizione antecedente alla lettura, e infine alla “DAQmx Clear Task.vi” che cancella il task.

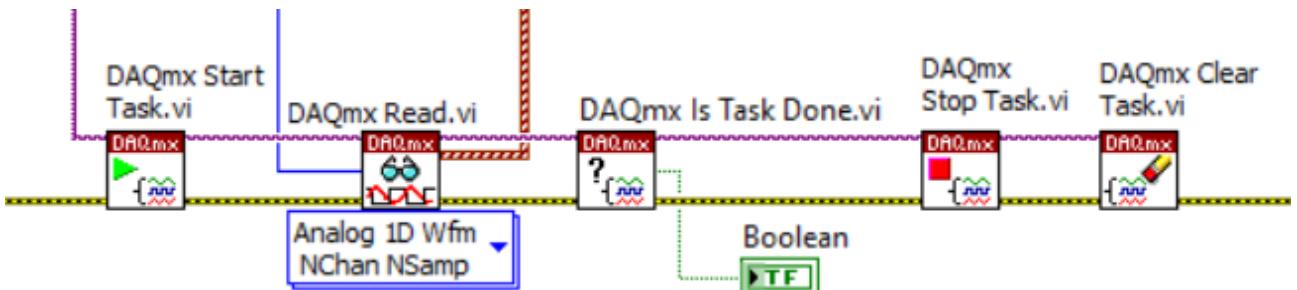


Figura 76 – Gestione task e acquisizione delle misurazioni

6 Conversione da array di waveform a matrice 6x1000

Alla fine della gestione del task ci si ritrova quindi con un array di segnali di natura “waveform” che riporta i 1000 segnali acquisiti per ogni canale di acquisizione.

Per poter gestire i dati letti in modo più agevole è stata scritta un’ulteriore Sub-VI che trasforma l’array di waveform in una matrice 6x1000, dove su ogni riga sono riportati i 1000 campioni letti per ogni canale. Il codice della Sub-VI può essere osservato in Figura 16.

Si può osservare che l’array di waveform in ingresso viene spacchettato in singoli dati waveform (uno per canale di acquisizione) tramite la “Index Waveform Array.vi”. Ogni waveform viene poi fatta passare attraverso la function “Get Waveform Components” che sostanzialmente smonta la waveform in ingresso permettendo di estrarre i singoli componenti. Prendendo in uscita solamente i dati letti e scartando quindi le informazioni temporali delle waveform è possibile ottenere un dato di tipo matrice 1x1000 su ogni canale che contiene solamente i dati letti dal sistema di acquisizione. Le sei matrici 1x1000 sono poi date in ingresso alla function “Build Matrix” che crea come risultato un’unica matrice 6x1000 che viene fornita in uscita alla Sub-VI.

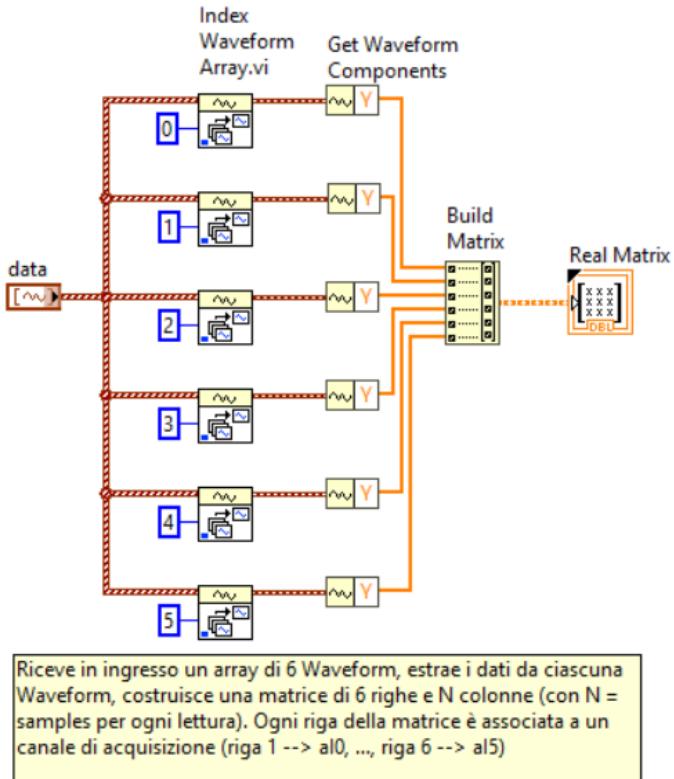


Figura 77 – “Waveform To Matrix (Sub-VI)” usata per trasformare i dati letti dalla “DAQmx Read.vi” in un formato matrice 6x1000

7 Elaborazione dei segnali di zero letti

Ritornando al codice principale della lettura dei segnali di zero in Figura 17 possiamo vedere come la Sub-VI appena commentata viene integrata nel codice. Possiamo osservare come l’array di waveform prima di venire convertito in matrice è mandato ad un “Waveform Graph” che permette di stampare i segnali letti e renderli visibili nella TAB “Segnali di zero”.

La matrice 6x1000 appena convertita viene poi trasformata in un formato array 2D tramite la “Matrix To Array.vi” per renderlo più facile da gestire nella seguente sezione del codice. L’array 2D è fornito come ingresso ad un ciclo for, il riquadro nero presente nel codice. Come ingresso al ciclo for viene anche fornito un array di dimensione 0 con 0 elementi, creato tramite la function “Initialize Array”. Sebbene possa sembrare poco sensata questa inizializzazione serve solo ad azzerare le dimensioni di un array che nel ciclo for aumenterà di dimensione a ogni ciclo. Senza questa inizializzazione zero-dimensionale il codice funzionerebbe correttamente solo alla prima esecuzione del programma. Il codice appena descritto può essere osservato in Figura 17.

Passiamo ora al codice interno al ciclo for, il quale viene eseguito esattamente sei volte, una per ogni canale di acquisizione. L’array 2D 6x1000 viene passato attraverso la function “Index Array” che estrae un singolo array ad ogni ciclo, l’indice del ciclo for (letto tramite la casella “i” in basso a sinistra) indica quale riga leggere ad ogni iterazione. L’array così letto viene quindi fatto passare attraverso la VI “Mean.vi” che esegue la media dei mille campioni letti sul canale restituendo un singolo valore in formato double.

Il valore mediato al primo ciclo for viene “accodato” all’array zero-dimensionale tramite la function “Build Array” che dà quindi come risultato una array normale. Ai cicli successivi l’array in uscita dal ciclo for viene riportato in ingresso alla function “Build array”, facendo sì che il vettore che raccoglie le medie sui vari canali accresca a ogni ciclo fino a diventare un vettore con 6 elementi a fine ciclo.

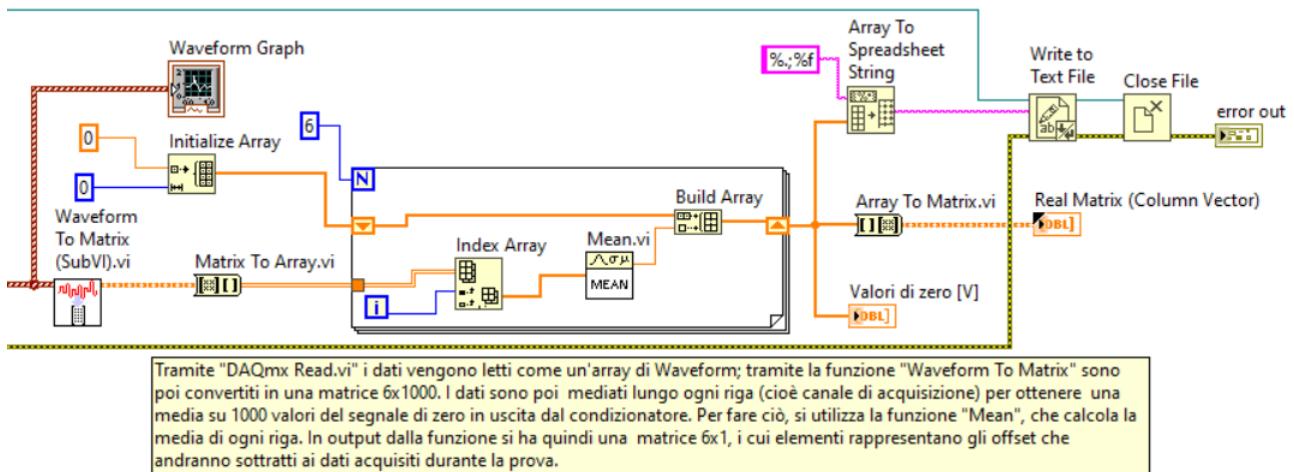


Figura 78 – Gestione dei segnali letti, media su ogni canale dei 1000 campioni e scrittura su file dei risultati

8 Scrittura su file dei segnali di zero

Dopo al ciclo for il vettore viene fornito in ingresso all'indicatore “Valori di zero [V]” presente nella TAB “Segnali di zero” che mostra i valori di zero mediati.

Il vettore viene anche convertito tramite la function “Array To Matrix.vi” ad una matrice 6x1, questo viene fatto per necessità di utilizzo dei segnali di zero nel codice principale. La matrice così ottenuta è data in output alla Sub-VI.

Il vettore viene infine fornito alla function “Array To Spreadsheet String” che converte l'array in formato Spreadsheet String, di modo che queste possano essere agevolmente scritte su file. Si passa quindi alla function “Write To Text File” che prendendo in ingresso il percorso di salvataggio scelto all'inizio dell'acquisizione dei segnali di zero (percorso azzurro) scrive sul file. Infine, la function “Close File” chiude il file dopo aver effettuato la scrittura. Il percorso di gestione degli errori è quindi fornito come uscita dalla Sub-VI, dovendo rientrare in una più ampia gestione dell'errore della VI di Acquisizione dei dati.

Nella Figura 18 è riportata una cattura della TAB “Segnali di zero” della VI “Acquisizione dati.vi”, gli elementi che compaiono a schermo sono quelli citati durante la spiegazione del codice.

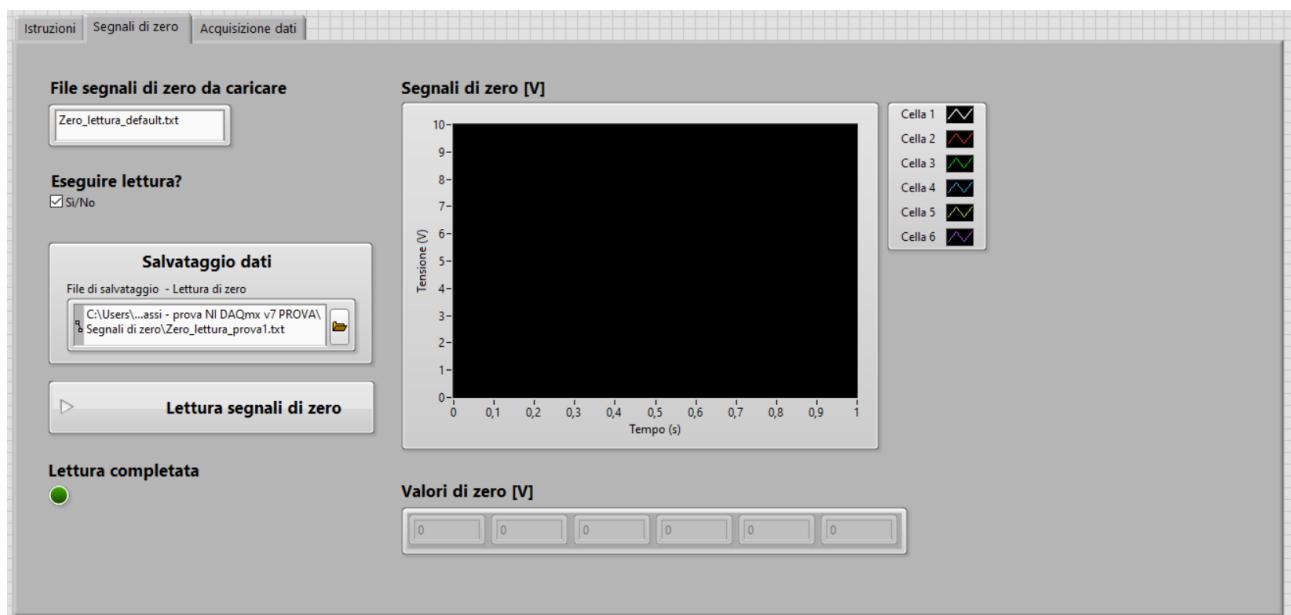


Figura 79 – TAB “Segnali di zero” della VI “Acquisizione dati.vi”

4.2.4 Build Conversion Matrix (subVI).vi

Questa subVI permette di definire la matrice di trasformazione (Figura 19) che consente di trasformare le forze misurate dai braccetti strumentati della piattaforma nelle forze e momenti riferiti al sistema di riferimento considerato.

La subVI riceve come parametri di ingresso:

- angolo α ;
- angolo β ;
- dimensione a ;

calcolati a partire dalla geometria della piattaforma, e restituisce la matrice di trasformazione.

$$\begin{bmatrix} -n & n & -\frac{\sqrt{3}}{2}m + \frac{1}{2}n & -\frac{\sqrt{3}}{2}m - \frac{1}{2}n & \frac{\sqrt{3}}{2}m + \frac{1}{2}n & \frac{\sqrt{3}}{2}m - \frac{1}{2}n \\ m & m & -\frac{1}{2}m - \frac{\sqrt{3}}{2}n & -\frac{1}{2}m + \frac{\sqrt{3}}{2}n & -\frac{1}{2}m + \frac{\sqrt{3}}{2}n & -\frac{1}{2}m - \frac{\sqrt{3}}{2}n \\ q & q & q & q & q & q \\ -qa & -qa & \frac{1}{2}qa & \frac{1}{2}qa & \frac{1}{2}qa & \frac{1}{2}qa \\ 0 & 0 & -\frac{\sqrt{3}}{2}qa & -\frac{\sqrt{3}}{2}qa & \frac{\sqrt{3}}{2}qa & \frac{\sqrt{3}}{2}qa \\ -na & na & -na & na & -na & na \end{bmatrix} \quad \begin{aligned} m &= \cos\alpha\cos\beta \\ n &= \cos\alpha\sin\beta \\ q &= \sin\alpha \end{aligned}$$

Figura 80 – Matrice di trasformazione

In particolare, all'interno della subVI vengono svolte le seguenti operazioni:

- inizializzazione di un array 6 x 6 con tutti elementi nulli;
- calcolo dei parametri m , n , q ;
- costruzione riga per riga della matrice di trasformazione all'interno di un For Loop. Durante l'iterazione i -esima, si sostituisce la riga i -esima dell'array 6 x 6 con un array 1 x 6 contenente gli elementi opportuni per la riga considerata. L'array 1 x 6 viene costruito all'interno di una Case Structure, che prevede un caso per ogni riga dell'array 6 x 6. Il Case Selector è il valore assunto dal contatore del For Loop; ricordando che il conteggio dei cicli parte da 0, alla prima riga corrisponderà il Case Selector 0 e così via per le altre righe.

In Figura 20 si riporta il codice che realizza quanto descritto in precedenza.

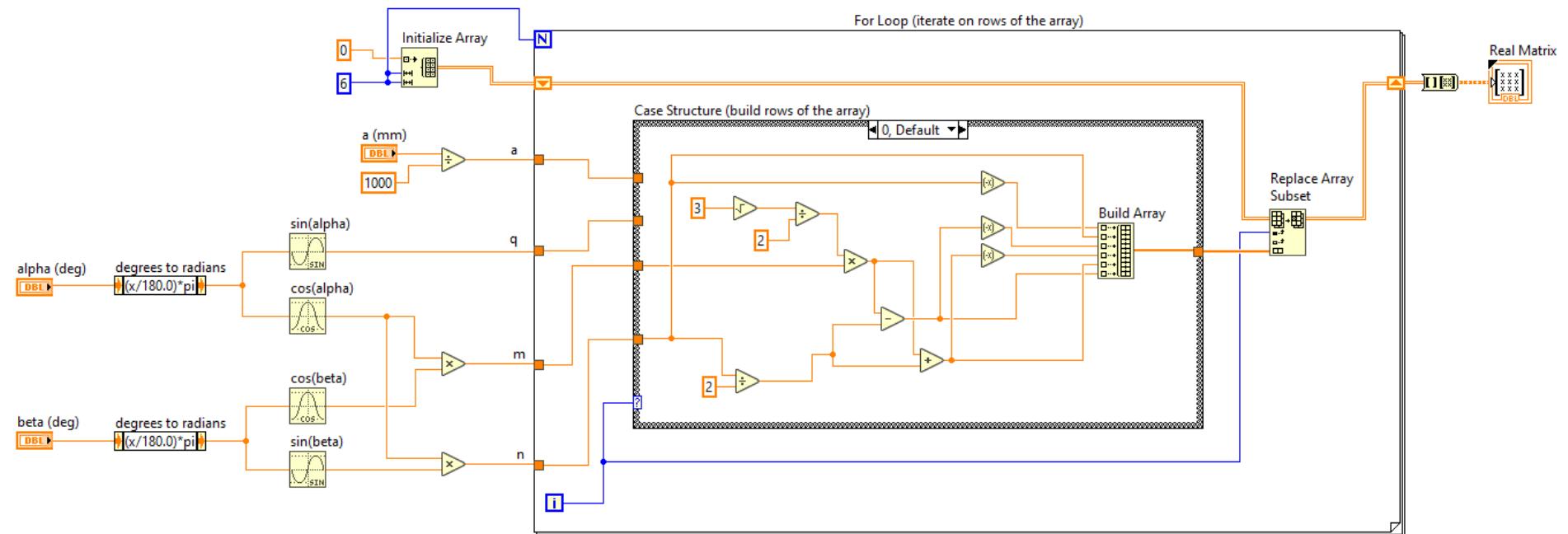


Figura 81 – Build Conversion Matrix (subVI).vi

4.2.5 Rearrange Signals (SubVI).vi

La matrice di trasformazione è riferita alla numerazione delle celle di carico nel banco prova precedente. Poiché nel nuovo banco prova le celle di carico sono numerate in modo diverso, per poter utilizzare tale matrice è necessario riordinare i segnali acquisiti. Il riordinamento dei segnali avviene come mostrato in Figura 21.

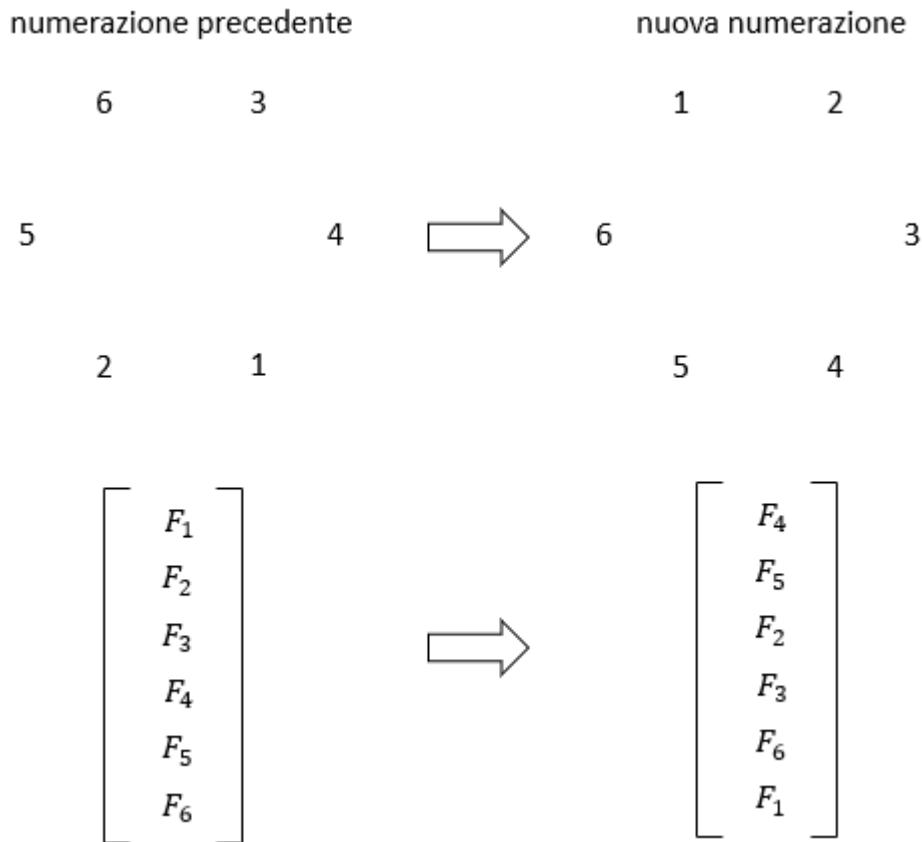


Figura 82 – Riordinamento dei segnali per tenere conto della nuova numerazione delle celle di carico

In Figura 22 si riporta il codice che esegue il riordinamento dei campioni acquisiti; come si avrà modo di vedere in seguito, i campioni acquisiti sono contenuti in una matrice $6 \times N$, con N pari al numero di campioni letti ad ogni lettura per ciascun canale. Tale matrice viene quindi fornita in ingresso alla subVI. Attraverso la funzione “Get Matrix Elements” si estrae dalla matrice la riga specificata dall’indice di riga (a tal proposito, si deve ricordare che le righe vengono numerate a partire da 0). Le singole righe estratte vengono quindi fornite alla funzione “Build Matrix” secondo l’ordine desiderato. Si ottiene quindi in uscita una nuova matrice $6 \times N$, con le righe opportunamente scambiate.

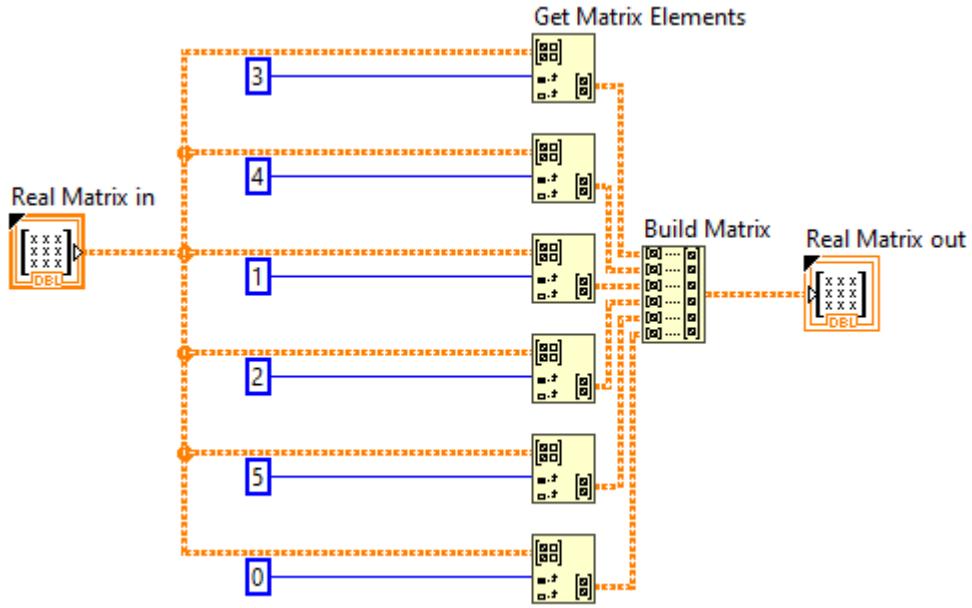


Figura 83 – Rearrange Signals (SubVI).vi

4.2.6 Acquisizione dati.vi

Il front panel del programma di acquisizione è organizzato in tre tab:

- “Istruzioni”, in cui sono riportate alcune note operative sul funzionamento del programma e sulla modalità di esecuzione delle prove;
- “Segnali zero”, in cui si può procedere alla lettura o al caricamento dei segnali di zero;
- “Acquisizione dati”, in cui si possono impostare i parametri di prova e visualizzare i segnali grezzi, le forze e i momenti.

4.2.7 Tab “Segnali di zero” (Figura 24)

È possibile procedere alla lettura dei segnali di zero (con piattaforma scarica) oppure caricare un file contenente segnali di zero acquisiti in precedenza.

La lettura sul momento dei segnali di zero è quella attiva di default (casella “Eseguire lettura?” spuntata). Prima di procedere alla lettura, è necessario creare un file di salvataggio; per far ciò è sufficiente premere sull’icona a forma di cartella: si aprirà una finestra di dialogo che chiederà di creare un file *.txt in cui salvare i valori medi dei segnali di zero. La cartella di default in cui si propone di creare il file è denominata “Segnali di zero” ed è contenuta all’interno della cartella in cui è salvato il codice di acquisizione (Figura 23).

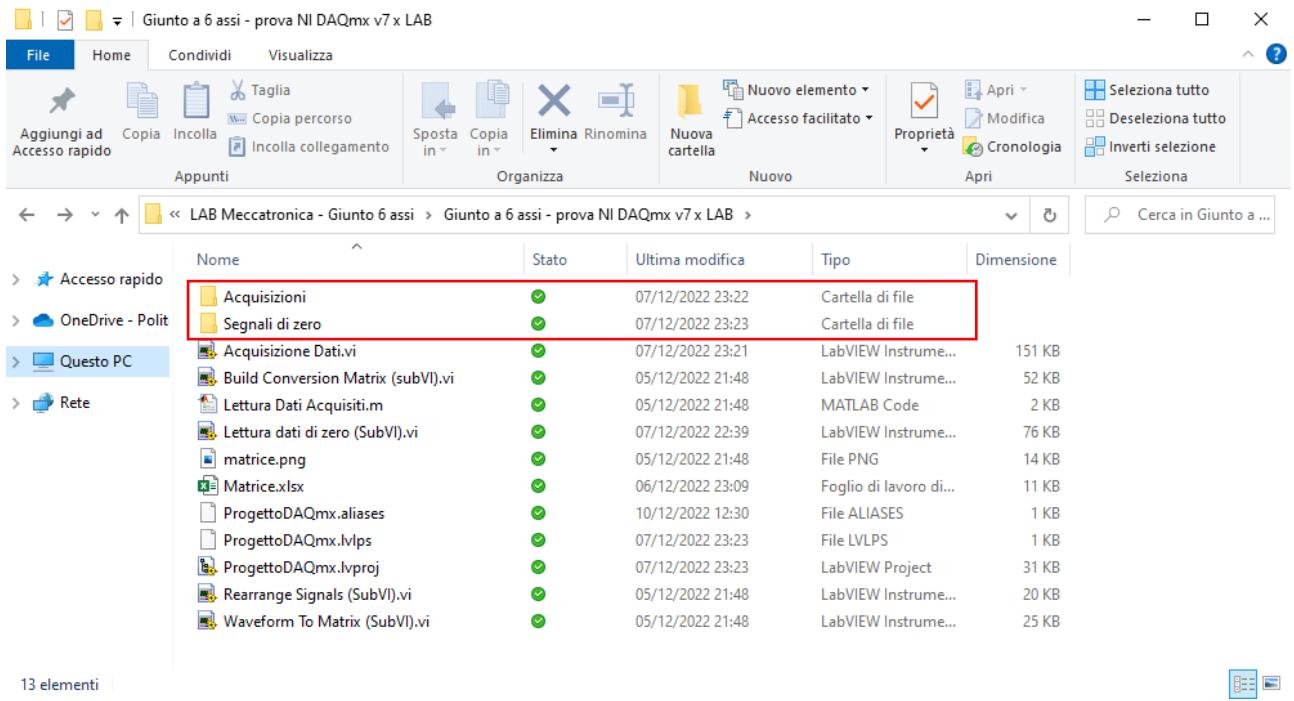


Figura 84 – Struttura della cartella contenente il progetto

Per iniziare la lettura dei segnali di zero, è necessario premere l'apposito pulsante. I segnali acquisiti vengono mostrati nel grafico posto al centro della schermata; inoltre, il valore medio calcolato di ciascun segnale viene mostrato nell'array “Valori di zero [V]”: tali valori rappresentano gli offset che verranno sottratti ai segnali acquisiti durante la prova vera e propria. Il LED “Lettura completata” si illumina nel momento in cui la lettura è terminata.

Si deve osservare che, nel caso in cui non si specificasse un percorso per il salvataggio del file (“File di salvataggio – lettura di zero” vuoto o con valore <Not A Path>), una volta premuto il pulsante di inizio acquisizione dei segnali di zero, si apre una finestra di dialogo che chiede di creare un file di salvataggio. L’acquisizione dei segnali inizia solo dopo la creazione del file.

In alternativa, è possibile caricare dei segnali di zero misurati in precedenza. Per fare ciò si deve specificare il nome del file (completo di estensione, necessariamente .txt) contenente i valori che si vogliono caricare. Il file deve essere contenuto nella cartella “Segnali di zero”, già citata in precedenza; di default viene caricato il file “Zero lettura_default.txt”. Per procedere al caricamento, si deve rimuovere la spunta dalla casella “Eseguire lettura?”; l’avvenuto caricamento è segnalato dalla comparsa dei valori nell’array “Valori di zero [V]”.

4.2.8 Tab “Acquisizione dati” (Figura 25)

In questa tab si possono impostare i parametri di prova (“Durata di acquisizione [s]” e “Frequenza di acquisizione [Hz]”). Similmente a quanto detto per la lettura dei segnali di zero, prima di procedere con la prova è necessario creare un file di salvataggio; l’unica differenza è la cartella di default in cui si chiede di creare il file (non più “Segnali di zero”, ma “Acquisizioni”; vedere Figura 23).

Per iniziare ad acquisire i segnali, si deve premere il pulsante “Inizia acquisizione”. Durante l’acquisizione, è possibile modificare l’intervallo da visualizzare sui grafici attraverso il controllo “Intervallo di visualizzazione [s]”; in pratica, sui grafici vengono mostrati gli ultimi t secondi, con t pari al valore impostato in “Intervallo di visualizzazione [s]”. Si deve tenere a mente che esiste un limite superiore all’intervallo di visualizzazione, variabile al variare della frequenza di acquisizione: in particolare, su ogni grafico è possibile visualizzare al massimo 10000 campioni in contemporanea (ciò corrisponde, ad esempio, a un intervallo di 10 s a 1000 Hz). Sono inoltre presenti due indicatori: il primo (“Tempo trascorso”) indica il tempo trascorso dall’inizio dell’acquisizione, il secondo (“Numero di samples per lettura”) indica il numero di campioni letti per ciascun canale ad ogni lettura. Si deve osservare che, per come è stato impostata la misura del tempo trascorso, il valore

indicato in “Tempo trascorso” al termine della prova sarà sempre di 1 s inferiore rispetto al valore impostato in “Durata di acquisizione [s]”.

Per terminare l’acquisizione prima che sia trascorso il tempo indicato in “Durata di acquisizione [s]” è possibile utilizzare il pulsante “Termina acquisizione”. Si consiglia di terminare l’acquisizione mediante tale pulsante e non premendo il pulsante “Abort execution” di LabVIEW, perché in questo modo si può chiudere correttamente il task di acquisizione.

Completano il front panel tre grafici, all’interno dei quali si mostra l’andamento nel tempo dei segnali in tensione (dopo rimozione degli offset) e delle forze e dei momenti calcolati.

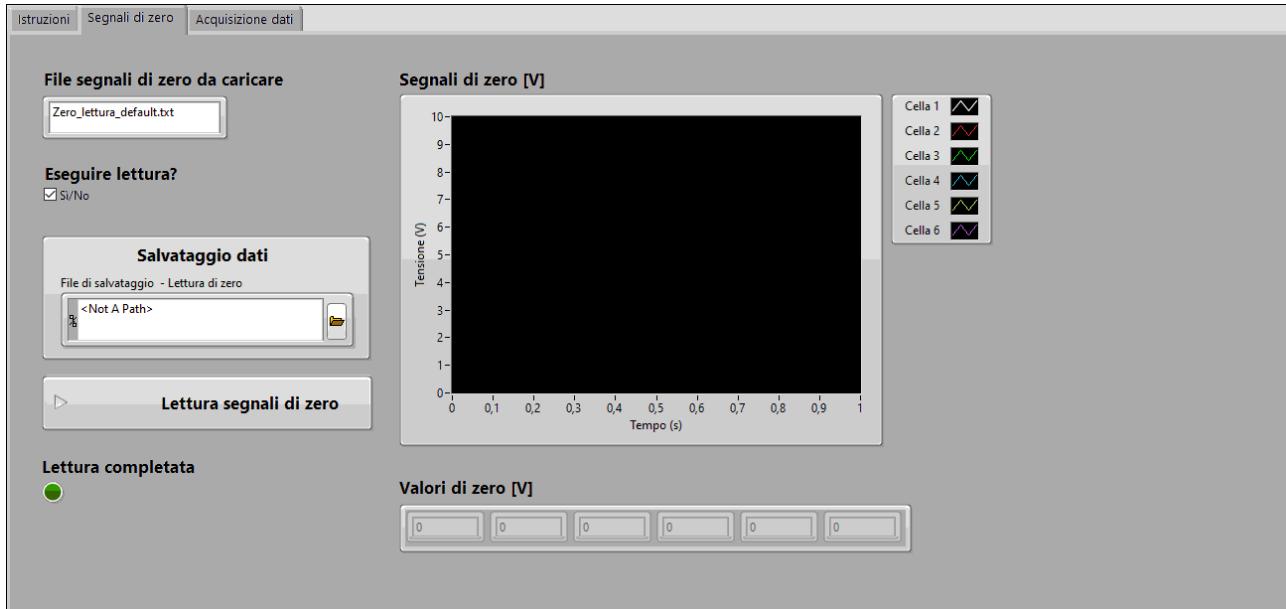


Figura 85 – Front panel “Acquisizione dati.vi” (tab “Segnali di zero”)

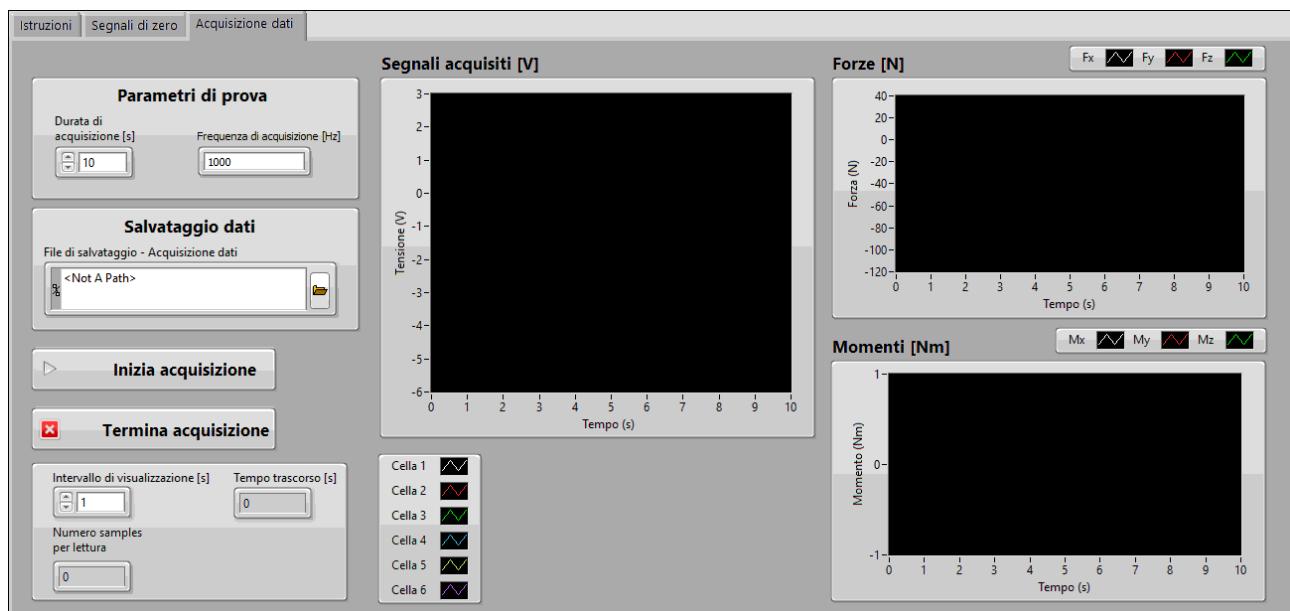


Figura 86 – Front panel “Acquisizione dati.vi” (tab “Acquisizione dati”)

Dopo aver illustrato le principali funzionalità del programma, si procede ora alla descrizione di come queste sono state implementate. Per consentire una migliore visualizzazione del codice, questo verrà illustrato una porzione per volta, ricordando che in questo caso l'ordine di esecuzione è da sinistra verso destra.

Nelle figure dalla 26 alla 28 sono riportate le porzioni di codice che realizzano quanto illustrato per la tab “Segnali di zero”.

In Figura 26 si impostano i canali di acquisizione, i valori massimo e minimo di tensione attesi e la modalità di lettura della tensione; sono inoltre presenti alcuni comandi di inizializzazione:

- definizione dei percorsi di default in cui proporre di salvare i file ;
- spegnimento del LED “Lettura completata”;
- inserimento della spunta nella casella “Eseguire lettura?”.

La presenza di una Event Structure permette di “congelare” l'esecuzione del codice fino a quando non si verifica uno dei seguenti eventi:

- pressione del pulsante “Lettura segnali di zero” (Figura 26);
- rimozione della spunta dalla casella “Eseguire lettura?” (Figura 27).

Nel primo caso verrà eseguito il caso TRUE della Case Structure, che richiama la subVI “Lettura dati di zero (SubVI)”, mentre nel secondo verrà eseguito il caso FALSE (Figura 28).

Mentre il programma è in attesa di un comando che dia inizio alla lettura o al caricamento dei segnali di zero, è possibile creare il file di salvataggio oppure specificare il nome del file contenente i segnali di zero da utilizzare.

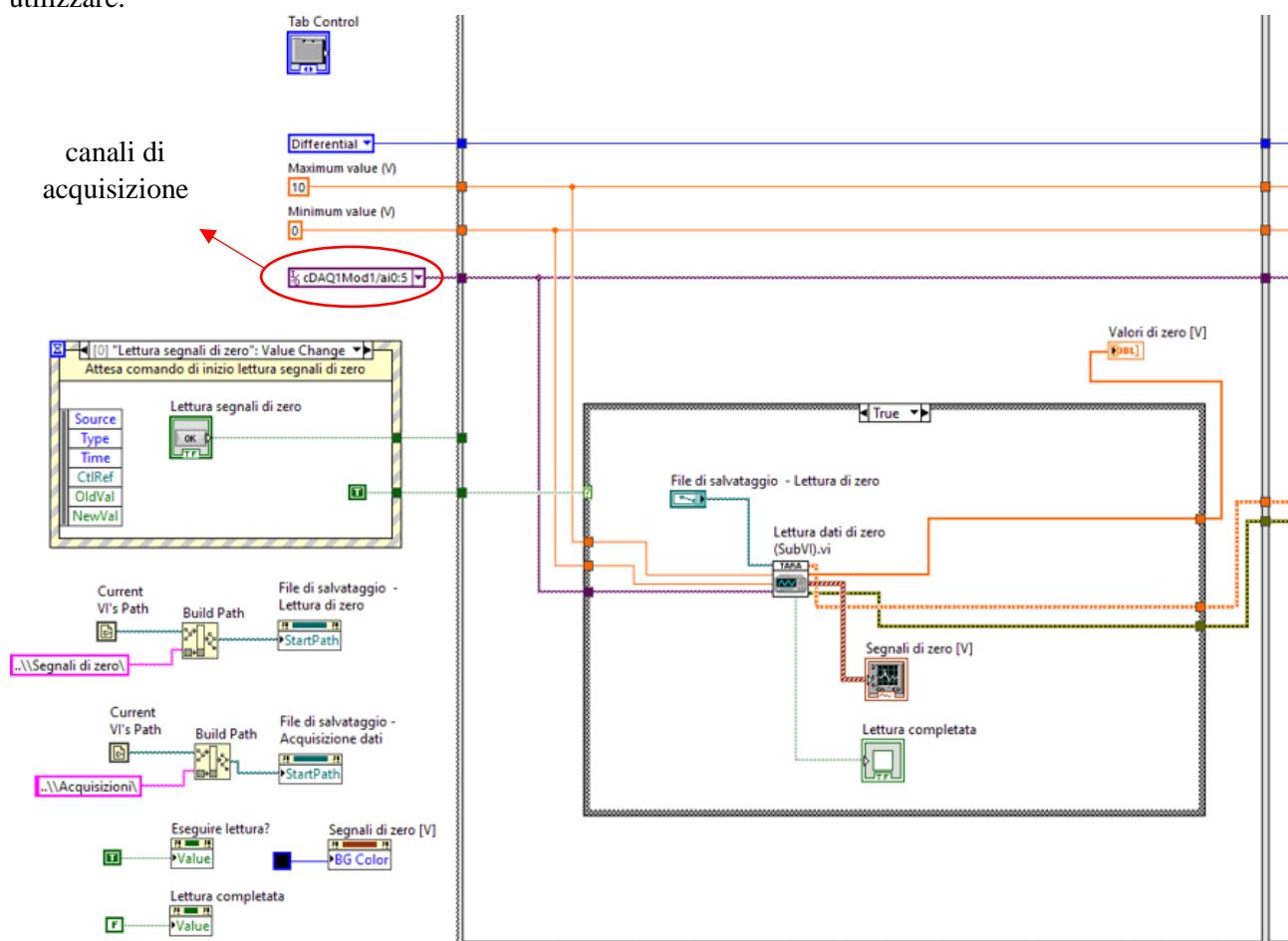


Figura 87

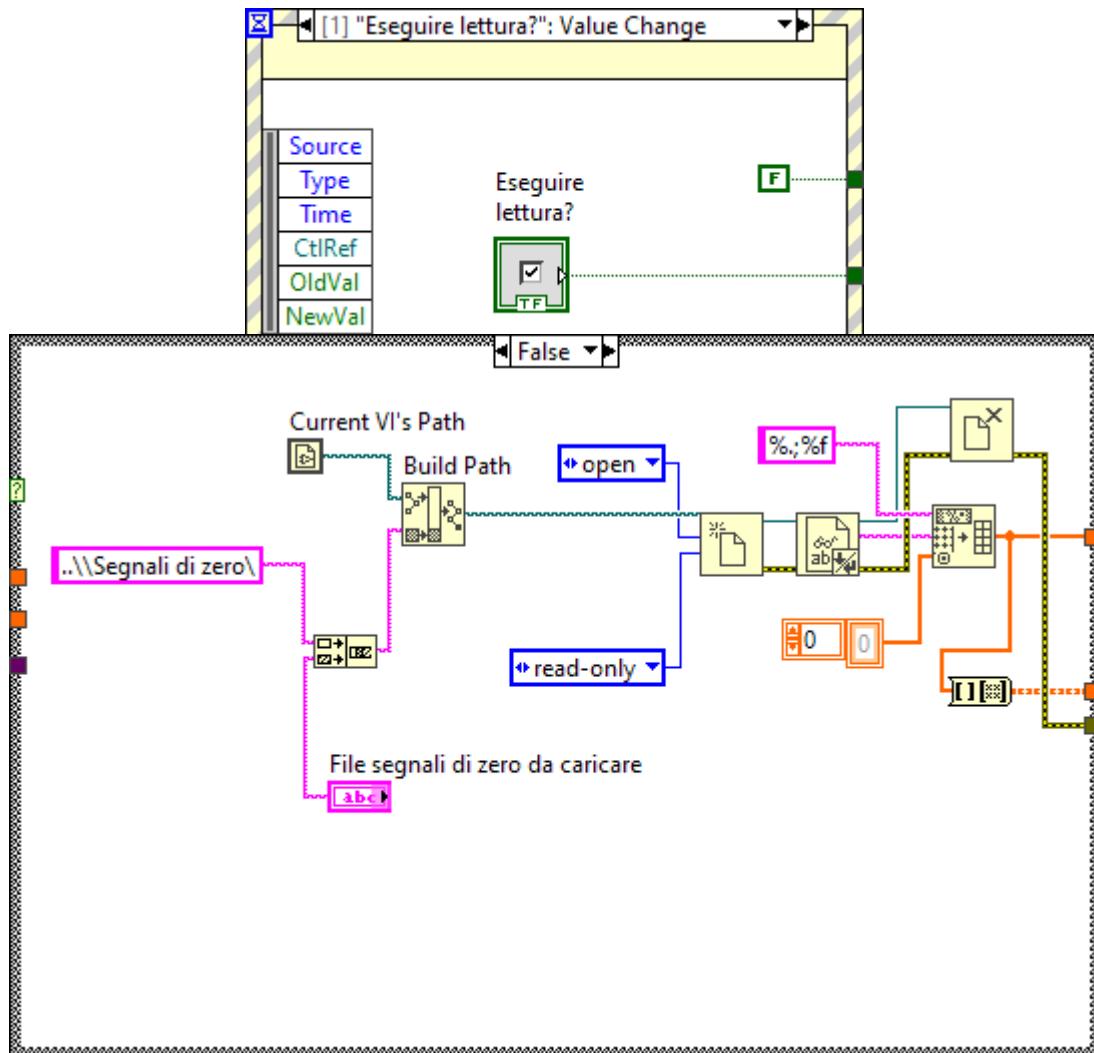


Figura 89

Nelle figure dalla 29 alla 31 sono riportate le porzioni di codice corrispondenti alla tab “Acquisizione dati”.

In Figura 29 è rappresentata una Event Structure che “congela” l’esecuzione del codice fino alla pressione del tasto “Inizia acquisizione”. Fino al momento in cui viene premuto il pulsante è possibile modificare i valori di “Durata di acquisizione [s]” e “Frequenza di acquisizione [Hz]”; inoltre, è possibile specificare il file di salvataggio dei dati acquisiti durante la prova.

Nella porzione di codice di Figura 30, si può notare la definizione del task di acquisizione: i parametri utilizzati (canali di acquisizione, valore massimo e minimo e modalità di lettura) sono gli stessi considerati per la creazione del task di lettura dei segnali di zero (se eseguita). Segue quindi la definizione della frequenza di acquisizione e del numero (finito, modalità di acquisizione “Finite samples”) di campioni da acquisire; quest’ultimo parametro viene definito indirettamente, impostando la frequenza e la durata di acquisizione. Viene infine definita la sorgente del clock con cui gestire l’acquisizione (in questo caso si utilizza il clock interno alla scheda di acquisizione).

Si procede quindi alla creazione di un file per il salvataggio dei dati nella posizione specificata in “File di salvataggio – Acquisizione dati” (Case Selector TRUE), oppure all’apertura di una finestra di dialogo che richiede la creazione di un file di salvataggio (Case Selector FALSE) all’interno della cartella “Acquisizioni”. Solo dopo aver creato il file di salvataggio può avere inizio l’acquisizione (“DAQmx Start Task.vi”).

Sempre in Figura 30 sono visibili la subVI “Build Conversion Matrix (subVI).vi”, con i relativi parametri geometrici di ingresso, e l’indicazione del guadagno da volt a newton.

Infine, si impostano alcune proprietà dei grafici; nello specifico, si impostano:

- la scala delle ascisse dei grafici, in modo che in ascissa compaia il tempo (di default, invece, LabVIEW riporta il numero di campioni acquisiti). A tal fine è sufficiente moltiplicare il numero di campioni per l’ inverso della frequenza di acquisizione;
- l’incremento lungo l’ascissa, ovvero la distanza tra i segni di graduazione sull’asse delle ascisse (valore che viene impostato a 1 s);
- il formato dei numeri riportati in ascissa (numeri decimali).

In Figura 31 è riportata la parte di codice in cui avviene la lettura e l’elaborazione dei campioni acquisiti. Per ogni canale di acquisizione (6 in totale) si leggono ad ogni iterazione N campioni mediante la funzione “DAQmx Read.vi”. Si è deciso di definire N in modo tale che ad ogni lettura corrisponda un secondo di acquisizione.

I campioni letti vengono restituiti sotto forma di un array 6 x 1 di waveform (una waveform per ogni canale di acquisizione). Prima di procedere all’elaborazione dei dati, si rende necessario manipolarli per trasformare l’array di waveform in una matrice 6 x N. A tal fine si utilizza la subVI “Waveform To Matrix (SubVI).vi”. Successivamente si rimuovono gli offset (misurati mediante un lettura di zero, oppure caricati da file). Siccome gli offset sono contenuti all’interno di un vettore 6 x 1, per poterli sottrarre alla matrice 6 x N dei segnali grezzi, è necessario moltiplicare tale vettore per un vettore 1 x N, i cui elementi sono tutti uguali a 1.

Una volta rimossi gli offset, i segnali acquisiti vengono mostrati nel corrispondente grafico sul front panel (“Segnali Acquisiti [V]”); prima di fare ciò, si trasforma la matrice in un array 2D e lo si traspone, così da ottenere un array N x 6 (per la corretta rappresentazione sul grafico, infatti, si deve avere una colonna di dati per ogni canale di acquisizione).

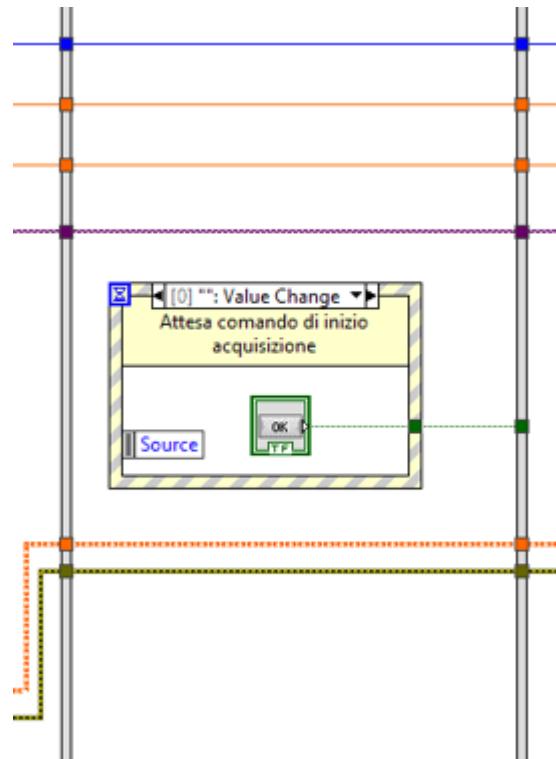


Figura 90

In parallelo a ciò, la matrice 6 x N dei segnali corretti viene manda in ingresso alla subVI “Rearrange Signals (SubVI)”, in modo da disporre nell’ordine corretto i segnali acquisiti prima di procedere con il calcolo di forze e momenti.

A questo punto la matrice 6 x N viene moltiplicata per il fattore di conversione da volt a newton: si ottengono in questo modo le forze misurate dai braccetti strumentati, le quali vengono quindi moltiplicate per la matrice di trasformazione, così da ottenere forze e momenti rispetto al sistema di riferimento della piattaforma.

Le forze e i momenti calcolati vengono mostrati sui grafici “Forze [N]” e “Momenti [Nm]”. In particolare, dopo aver trasformato in un array 2D la matrice 6 x N risultante dal prodotto tra la matrice delle forze sui braccetti e la matrice di conversione, si selezionano le prime tre righe mediante la funzione “Array Subset”, corrispondenti alle forze, e le ultime tre righe, corrispondenti ai momenti, per la rappresentazione su grafico. Anche in questo caso, per i motivi già esposti in precedenza, è necessario trasporre gli array 2D prima di procedere alla rappresentazione su grafico.

I segnali acquisiti (con l’offset già rimosso, ma non ancora riordinati) e le forze e i momenti calcolati vengono uniti in una matrice 12 x N per il salvataggio su file di testo. Prima di procedere alla scrittura sul file di salvataggio, la matrice 12 x N viene trasposta, così da ottenere una matrice N x 12, le cui prime 6 colonne corrispondono ai segnali acquisiti, mentre le restanti 6 rappresentano forze e momenti.

L’impostazione dinamica dell’intervallo di visualizzazione è stata implementata nel seguente modo:

- se il valore impostato in “Intervallo di visualizzazione [s]” è minore di quello massimo visualizzabile, si utilizza tale valore per modificare l’intervallo di visualizzazione dei grafici. In particolare, se il valore richiesto è maggiore del valore massimo attualmente mostrato in ascissa, si procede a impostare come nuovo valore massimo quello indicato in “Intervallo di visualizzazione [s]”; in caso contrario, si legge il valore massimo dell’ascissa, si sottrae ad esso il valore richiesto e si imposta il valore risultante da tale operazione come valore minimo da visualizzare;
- se il valore impostato in “Intervallo di visualizzazione [s]” è maggiore di quello massimo visualizzabile, si impone come valore massimo dell’ascissa il valore massimo visualizzabile con la frequenza di acquisizione utilizzata durante la prova.

L’acquisizione e l’elaborazione si ripetono all’interno del ciclo While fino a che non si verifica una delle seguenti condizioni:

- “DAQmx Task Is Done.vi” fornisce un valore TRUE, a significare che il task ha acquisito il numero di campioni richiesto;
- pressione del tasto “Termina acquisizione” sul front panel;
- presenza di un errore.

Una volta terminato il ciclo While si procede alla chiusura del task di acquisizione mediante la funzione “DAQmx Stop Task.vi”. Attraverso la funzione “DAQmx Clear Task.vi” si liberano le risorse di sistema che erano state riservate per l’esecuzione del task. Infine, si chiude il file di salvataggio e si valuta la presenza di eventuali errori incorsi durante l’esecuzione del programma (nel qual caso, la funzione “Simple Error Handler.vi” produce a schermo un messaggio di errore).

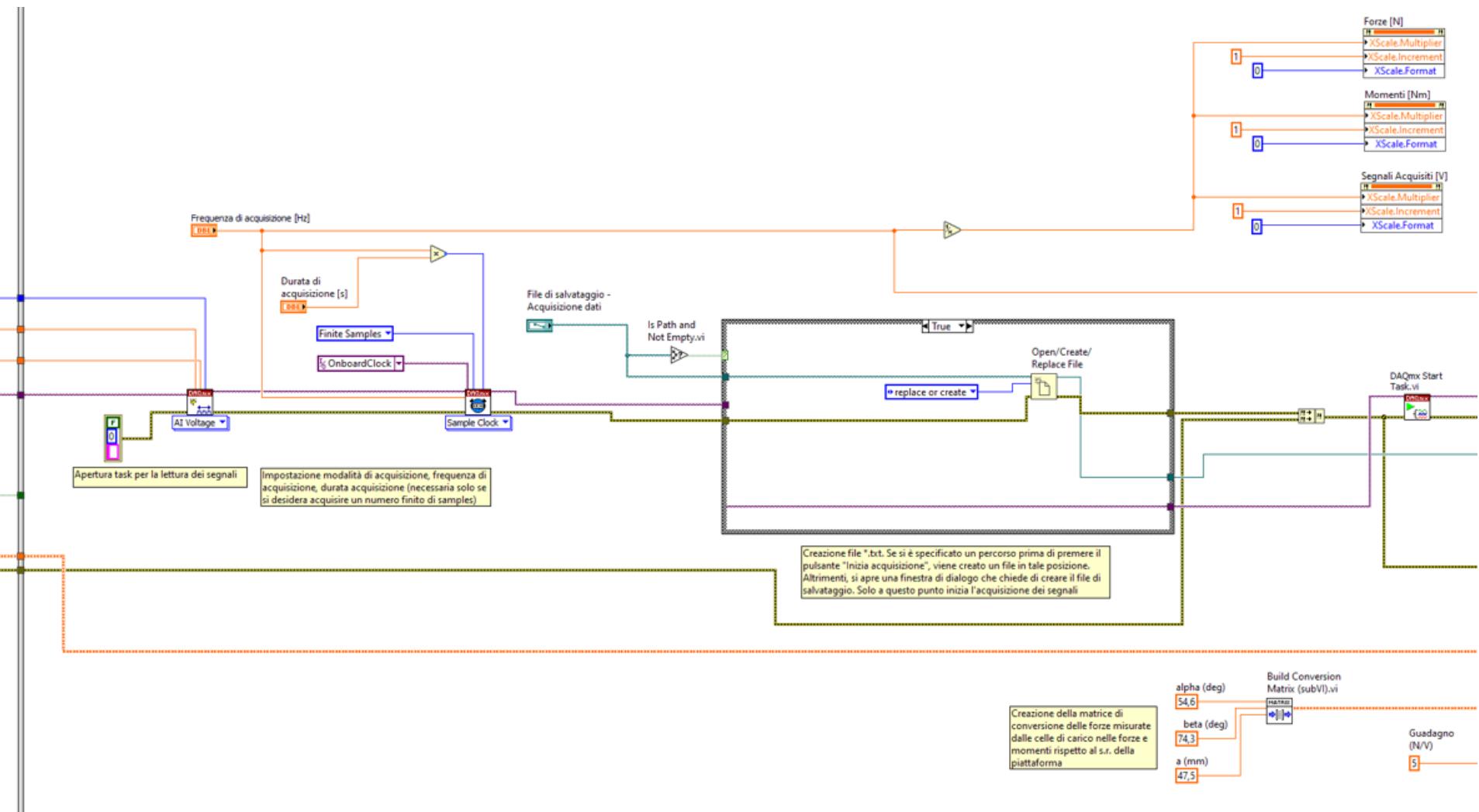


Figura 91

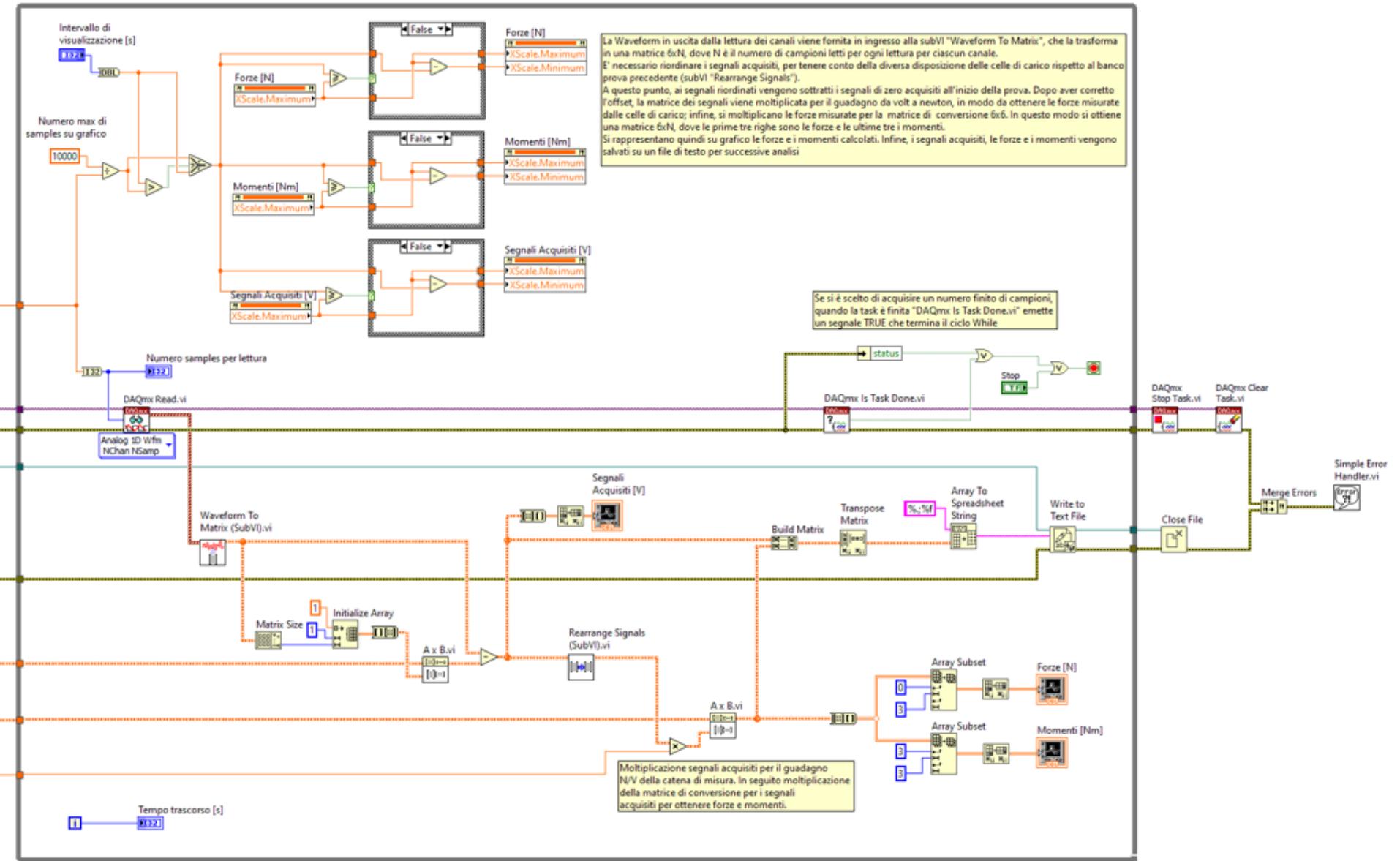


Figura 92

4.3 Modello matematico del sistema

4.3.1 Introduzione

Lo scopo di questa parte dell'esercitazione è quello di estrapolare i risultati ottenuti con le prove sperimentali, acquisiti attraverso un codice LabView, sul giunto di forza per poi confrontarli con un modello ideale necessario per ricavare i parametri fisici caratteristici del sistema che verranno usati per la sua stessa schematizzazione.

Il sistema nelle prove sperimentali è costituito da una massa, collegata al giunto di forza a sei assi tramite una molla, a cui viene fornito uno spostamento iniziale non nullo. Una volta lasciata la massa posizionata ad una certa x_0 viene lasciata oscillare liberamente fino a fermarsi completamente.

Il giunto di forza a sei assi con massa oscillante può essere schematizzato come un sistema massa-molla a cui viene aggiunto però un effetto smorzante facilmente intuibile poiché senza di esso la massa continuerebbe ad oscillare per un tempo infinito. L'effetto smorzante è quello dovuto alla presenza dell'attrito fluidodinamico dell'aria.

4.3.2 Modello matematico

Quindi il sistema può essere schematizzato come segue:

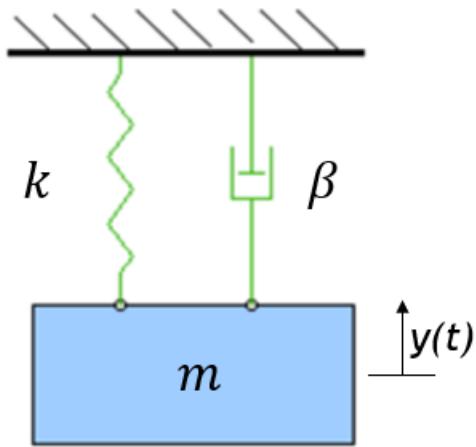


Figura 93 – Schema a parametri concentrati del modello

I passaggi matematici necessari per ricavare la forza agente sul vincolo sono i seguenti:

$$m\ddot{x} + \beta\dot{x} + kx = 0$$

$$x(t) = x_0 \sin(\omega_d t - \varphi) e^{-\zeta \omega_n t}$$

$$\omega_d = \omega_n \sqrt{1 - \zeta^2}$$

$$F_v = mg + kx(t)$$

Prima di procedere all'analisi matematica vera e propria è necessario avere dei valori che si possono definire di "primo tentativo" per la misura della massa, della rigidezza della molla e dello smorzamento dell'aria. Lo smorzamento verrà in una prima approssimazione considerato nullo poiché risulterebbe eccessivamente complicato il suo calcolo. Per quanto riguarda la massa si è usata una bilancia per la sua misura e per la rigidezza della molla si è usato un dinamometro con il quale si è misurata la forza necessaria per far allungare la molla di una lunghezza nota; tramite il loro rapporto si è poi riusciti a calcolare la rigidezza.

Per il confronto del modello matematico con i dati ottenuti dalle prove sperimentali sono state prese due strade alternative in modo da avere due metodologie di calcolo differenti operanti con due metodologie di calcolo

diverse. Si è deciso quindi di utilizzare sia un programma MATLAB che un programma SIMULINK. Da qui in avanti verranno descritti i passaggi per la stesura dei due programmi in due sezioni distinte.

4.3.3 Programma MATLAB

Come primissimo passaggio è necessario acquisire i dati che provengono dalle prove sperimentali effettuate con il codice LabView. In questo il codice LabView restituirà i valori in un file di testo secondo il seguente formato: nelle prime 6 colonne sono presenti i segnali in volt che restituiscono le 6 celle di carico della piattaforma di Stewart, nelle 3 colonne successive sono presenti i valori di forza nelle tre direzioni e infine nelle ultime 3 colonne sono presenti i valori di momento ottenuti lunghi i 3 assi del sistema di riferimento. Più precisamente nelle colonne dalla 7 alla 9 saranno presenti nel seguente ordine F_x , F_y ed F_z mentre nelle colonne dalla 10 alla 12 saranno presenti nel seguente ordine M_x , M_y , M_z . I valori dei 3 momenti e delle 3 forze sono ottenuti passando attraverso una matrice di conversione i risultati in volt ottenuti dalle 6 celle di carico che corrispondono alle prime 6 colonne del file di testo, passaggio che però viene già svolto all'interno del codice LabView. Seguendo il sistema di riferimento preso in considerazione per l'analisi del modello sarà necessario elaborare solo i segnali di F_y e M_x . Le altre due forze e gli altri due momenti non vengono utilizzati perché si presuppone che la massa venga fatta oscillare in una posizione perfettamente verticale e quindi lungo l'asse y . Le celle di carico rilevano dei valori per quel che riguarda F_x , F_z , M_y ed M_z ma sono considerabili dei disturbi dovuti allo sgancio della massa che essendo fatto manualmente non sarà perfettamente verticale e quindi lungo l'asse y . Tuttavia, le celle di carico rilevano dei valori diversi da zero per quel che riguarda F_x , F_z , M_y ed M_z poiché il rilascio manuale non garantisce un totale allineamento con l'asse verticale a cui si sommano ulteriormente i disturbi di natura elettromagnetica tipici di ogni acquisizione.

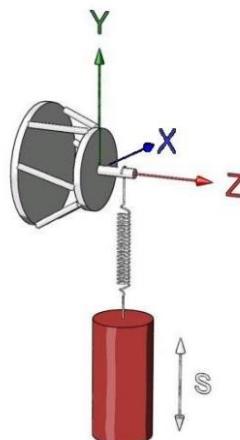


Figura 94 – Sistema di riferimento della piattaforma di Stewart

Si mostrano di seguito anche i segnali misurati delle 3 forze e dei 3 momenti in una prova eseguita in laboratorio:

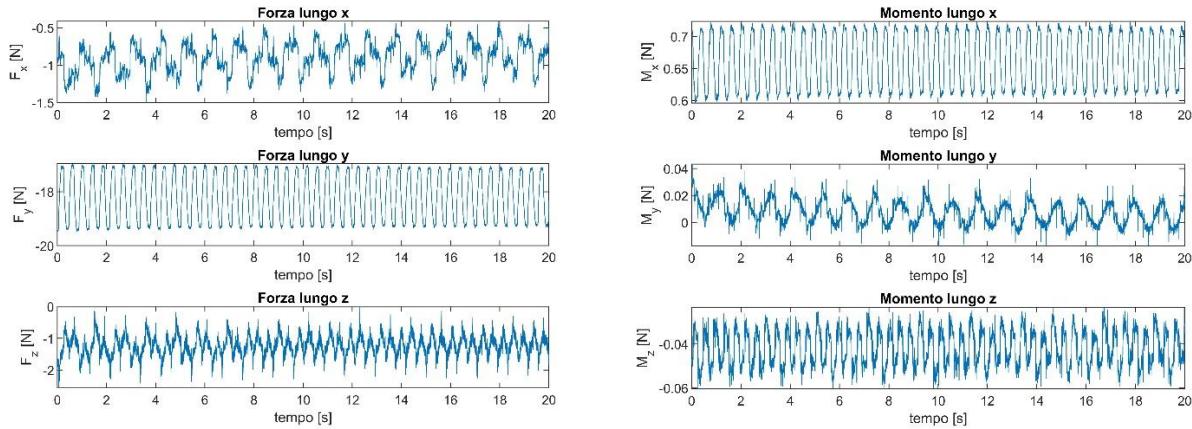


Figura 95 – Segnali di forza e momento di una prova

Dalla figura appena mostrata si nota immediatamente quali segnali sono dei disturbi perché hanno un'oscillazione molto piccola, come per F_x e M_y , oppure perché oscillano intorno ad un valore negativo, come per F_z e M_z .

Dopo aver acquisito i dati nel modo corretto si inseriscono nel modello i valori di massa e rigidezza della molla misurati in laboratorio e contestualmente si calcola sia la frequenza naturale del sistema che il periodo di oscillazione. Per una migliore visualizzazione vengono invertiti i segni dei valori di F_y negativi (risultano negativi per la scelta del sistema di riferimento).

4.3.4 Pre-analisi

Prima di effettuare il fitting vero e proprio del modello con i dati sperimentali si calcolano alcuni parametri caratteristici, quali beta (decremento logaritmico), fase e ampiezza, in modo da agevolare la funzione di fitting e non farle fare troppe iterazioni dovute ad una “partenza” troppo lontana dai dati sperimentali. I parametri caratteristici appena citati vengono calcolati sui valori di F_y ottenuti dal codice Labview.

- β : decremento logaritmico

Il decremento logaritmico è utile per avere un valore di primo tentativo di coefficiente di smorzamento. Questa procedura fa risparmiare importanti iterazioni al modello di fitting perché come si è detto nell'introduzione a questo capitolo si è deciso di non misurare sperimentalmente in laboratorio il coefficiente di smorzamento perché troppo complicato e poco preciso.

Quindi per il calcolo del decremento logaritmico è necessario prima applicare un filtro alla forza in modo da rendere meno disturbato il segnale, che renderebbe altrimenti impossibile individuare con una precisione adeguata i picchi nell'andamento della forza. La formula utilizzata per il calcolo del decremento logaritmico è la seguente:

$$\delta = \frac{1}{n} \ln \frac{x_i}{x_{i+n}}$$

Dove x_i rappresenta il valore dell'ampiezza i -esima, per misurare meglio lo smorzamento si prendono due ampiezze lontane un numero di periodi n tra loro. Con il valore del decremento logaritmico è poi possibile calcolare il valore dello smorzamento con i seguenti passaggi:

$$\zeta = \frac{\delta}{\sqrt{4\pi^2 + \delta^2}}$$

$$\beta = 2\zeta\omega_n$$

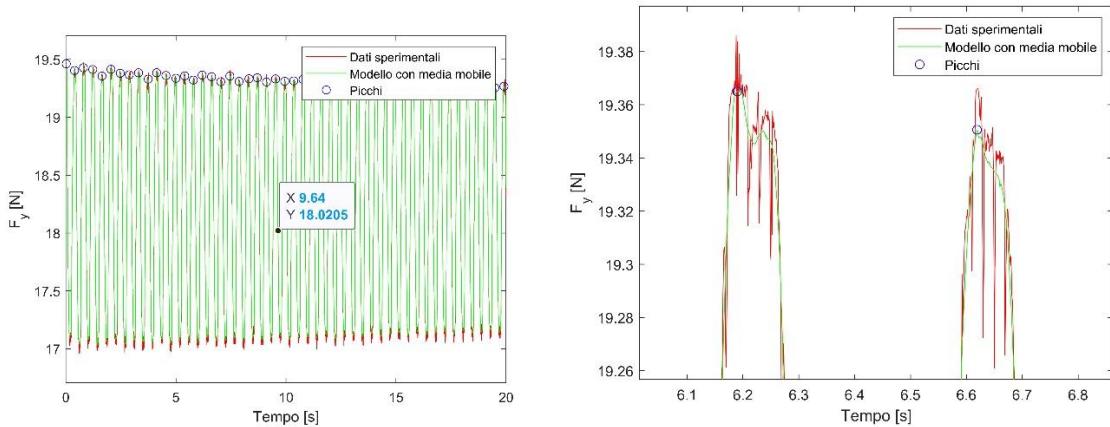


Figura 96 – Ricerca dei massimi per il decremento logaritmico

Dalla figura appena mostrata si vede immediatamente quanto sia importante l'utilizzo del filtro, senza il quale ci sarebbero troppe oscillazioni che porterebbero ad avere una ricerca dei picchi molto complessa e che quindi restituisce un risultato scadente.

- Aampiezza

Per il calcolo dell'ampiezza è necessario anche individuare i picchi negativi, ovvero i minimi relativi di ogni periodo di oscillazione. I picchi negativi vengono salvati da MATLAB come un valore negativo e quindi è necessario prendere i picchi negativi con segno opposto per il calcolo della fase che viene prendendo in considerazione il primo picco positivo e il primo picco negativo, ovvero i due picchi che risentono meno dello smorzamento dovuto all'attrito fluidodinamico con l'aria. Quindi il calcolo preliminare dell'ampiezza viene eseguito con la seguente formula:

$$A_0 = \frac{x_1^+ - x_1^-}{2k}$$

Dove x_1^+ è il valore del primo picco positivo e x_1^- è il valore del primo picco negativo.

- Fase

L'introduzione del parametro di fase si rende necessario dal momento in cui non è noto l'istante di inizio acquisizione dei dati, potendo questa iniziare in un punto qualsiasi della sinusoide.

Si tratta dunque di identificare lo sfasamento rispetto a una funzione sinusoidale con zero nell'origine. Per il calcolo della fase si individua dunque il primo picco del segnale di forza e si ricava successivamente la fase attraverso i seguenti calcoli:

$$\varphi = \frac{\pi}{2} - t_1^+ \cdot \omega_d$$

Se negativa, si può renderla positiva attraverso la formula

$$\varphi = 2 \cdot \pi + \varphi$$

NOTA: per valutare ω_d si è ricavato il periodo come media dei primi n periodi, identificati dai rispettivi picchi di massimo.

-Alternativamente si può trovare attraverso il massimo della ttf.

4.3.5 Fitting parametrico: fminsearch

Calcolati i parametri di primo tentativo si passa alla procedura di fitting del modello che viene eseguita utilizzando una funzione Matlab chiamata *fminsearch*. La funzione *fminsearch* trova il minimo di una funzione multivariabile utilizzando un metodo “derivative-free”. Per utilizzarla nel migliore dei modi viene creato un vettore contenente i valori dei parametri su cui andrà eseguita la procedura di fitting quali massa, coefficiente di smorzamento, rigidezza della molla, fase e ampiezza. I valori inseriti in questo vettore sono tutti i valori di

primo tentativo trovati sperimentalmente (massa e rigidezza della molla) o attraverso una pre-analisi sulla forza F_y (fase, ampiezza e coefficiente di smorzamento). Costruito il vettore con le variabili si scrive la formula della forza sul vincolo ricavata nell'introduzione e si crea successivamente una funzione errore che è calcolata come la norma della forza del vincolo ottenuta con il modello e la forza F_y sperimentale.

```
err= @(var) norm(F_sim(var,tt)-Fy);
option= optimset('Display','iter','TolFun',1e-6,'TolX',1e6,'MaxFunEvals',5000,'MaxIter',5000);
var_opt= fminsearch(err,var_in,option); %parametri di fitting
```

Figura 97 – Righe di codice in cui si usa fminsearch

Ora è possibile utilizzare la funzione *fminsearch* direttamente sull'errore, ovvero viene utilizzata per cercare di minimizzare la funzione errore. La funzione quindi modificando i parametri inseriti nel vettore, attraverso successive iterazioni, troverà una combinazione di valori con i quali, secondo essa, il modello si avvicina il più possibile ai valori sperimentali. I passaggi matematici e la logica matematica che è presente dietro la funzione *fminsearch* non vengono trattati perché esulerebbero dall'argomento principale di questa analisi. Per avere una certezza maggiore dei risultati si è eseguita la funzione *fminsearch* anche all'interno di un ciclo for in modo da valutare se restituisse sempre gli stessi valori. Potrebbe capitare che la soluzione capiti in un punto in cui siano presenti più minimi e che quindi cambiando le i passaggi e le iterazioni all'interno della funzione si ottengano risultati diversi. Nel caso analizzato ciò non accade infatti i valori ottenuti sono gli stessi sia passando attraverso il ciclo for che non passandoci.

4.3.6 Risultati

Per dare validità al modello realizzato si è dovuto eseguire il fitting su più prove, tutte eseguite con una frequenza di campionamento di 1000 Hz, e questi sono i risultati ottenuti:

Tabella 5 – Risultati ottenuti dal modello su matlab

	m [kg]	k [N/mm]	ζ [-]	x_0 [m]	ω_n [rad/s]	h [m]
Prova 1	1,86	425	0,0007	0,0031	15,14	0,036
Prova 2	1,88	430	0,0007	0,0053	15,14	0,035
Prova 3	1,86	426	0,0007	0,0038	15,14	0,037
Prova 4	1,86	426	0,0008	0,0042	15,15	0,037
Prova 5	1,85	424	0,0008	0,0024	15,15	0,037
Prova 6	1,89	433	0,0012	0,0048	15,13	0,032

Dalla tabella dei risultati appena mostrata si nota subito come i risultati siano ripetibili, ovvero che il modello riesce a restituire valori molto vicini tra loro anche con prove eseguite separatamente. Nonostante il modello restituisca dei valori molto simili tra loro, i valori di massa e rigidezza cambiano considerevolmente rispetto a quelli misurati in laboratorio.

```
"m" "beta" "k" "x0" "phi" "zeta" "w_n"
2.0621 0.0028 474.3200 0.0024 5.0032 0.0000 15.3967
1.8565 0.0208 425.3543 0.0031 4.4419 0.0007 15.1364
```

Figura 98 – Risultati stampanti da matlab nella Command Window

Nella Figura 37 sono mostrati i risultati che matlab mostra a schermo quando viene eseguita la prova che nella tabella dei risultati è stata chiamata “Prova 1”. Nella prima riga ci sono i valori iniziali da cui *fminsearch* inizia le iterazioni, mentre la riga sotto mostra i risultati finali. Nonostante massa e rigidezza siano dei valori misurati in laboratorio con molta accuratezza, i loro valori cambiano notevolmente prima e dopo le iterazioni e questo non è un buon risultato. Mostrando anche graficamente i risultati ottenuti si nota che il modello non riesce ad eseguire un fitting molto preciso.

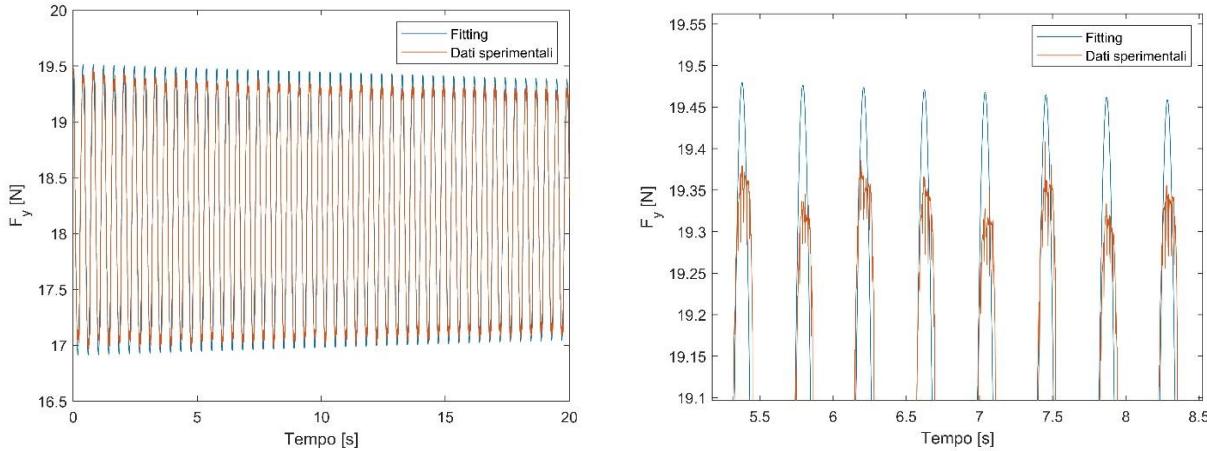


Figura 99 – Risultati del fitting su matlab

Come si vede dalla figura i risultati del fitting e i dati sperimentali non sono molto vicini tra loro, ma questo è dovuto ad un errore, che è stato riscontrato solo in un secondo momento, nel montaggio delle celle di carico da parte del fornitore. Le celle di carico presentano un gioco per il quale risulta impossibile avere dei risultati di fitting migliore.

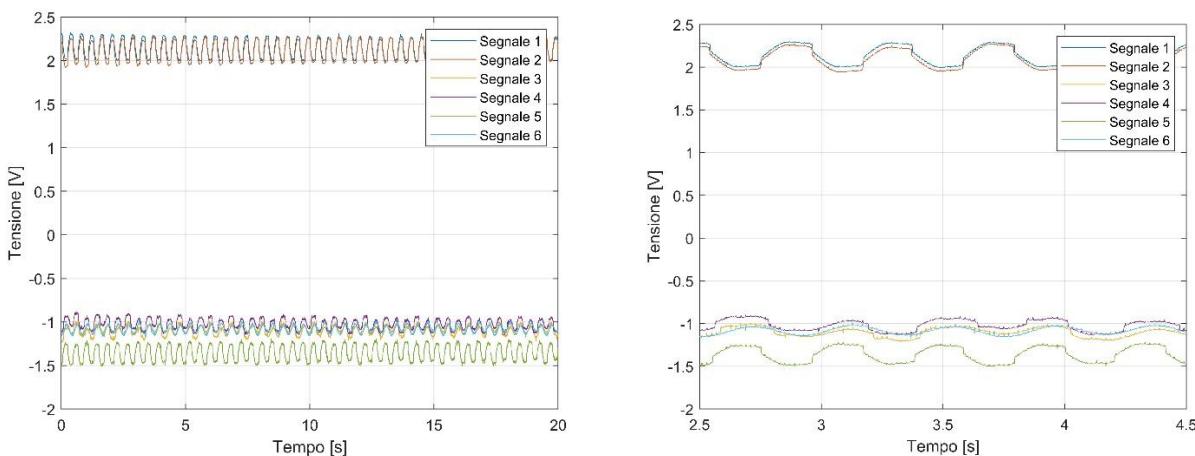


Figura 100 – Tensioni delle celle di carico

Il gioco presente nel montaggio delle celle di carico genera un segnale misurato dalle celle di carico non sinusoidale, come si può vedere dalla figura. Risulta quindi impossibile per il modello eseguire un fitting più preciso perché è pensato per avere un segnale sinusoidale in ingresso e non di questa forma tagliata. In conclusione, si può dire che il modello restituisce valori non molto precisi però si può considerare un buon modello data la sua alta ripetibilità e soprattutto dato che gli errori sono dovuti ad un errore nella parte hardware e non software.

4.3.7 Teorema di Nyquist

Un altro obiettivo dell'esercitazione è l'analisi e la verifica del teorema di Nyquist o del campionamento, che definisce la minima frequenza necessaria per campionare un segnale analogico senza perdere il contenuto in frequenza del segnale stesso. Questa frequenza, detta frequenza di Nyquist è imposta pari a

$$f_c = 2 f_{MAX}$$

Per verificare il teorema si sono eseguite prove sperimentali a diversa frequenza di campionamento. In particolare si sono scelte frequenze di campionamento minori, poco maggiori e via via ordini di grandezza più grandi della frequenza naturale (essendo un sistema molto sottosmorzato, frequenza naturale o di oscillazione smorzata sono molto prossime) teorica del sistema, pari circa a 2,4 Hz.

Le frequenze utilizzate sono:

- 2,4 Hz
- 5 Hz
- 10 Hz
- 100 Hz
- 500 Hz
- 1000 Hz

Sono di seguito riportati alcuni dei segnali ricavati dalle prove sperimentali.

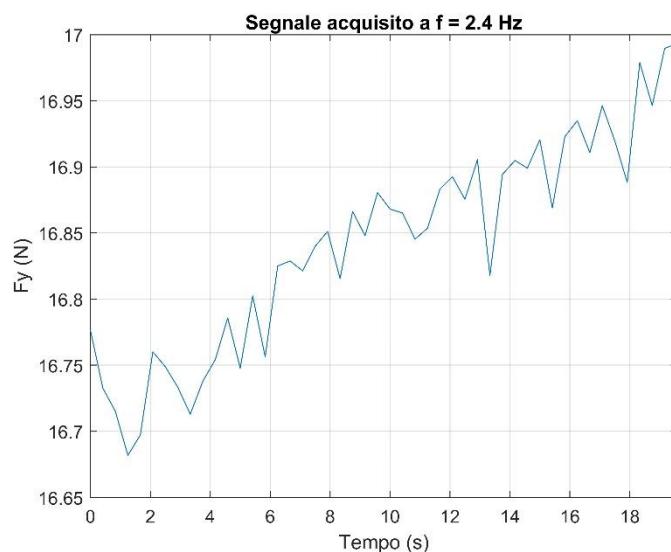


Figura 101 – Segnale con frequenza di acquisizione di 2,4 Hz

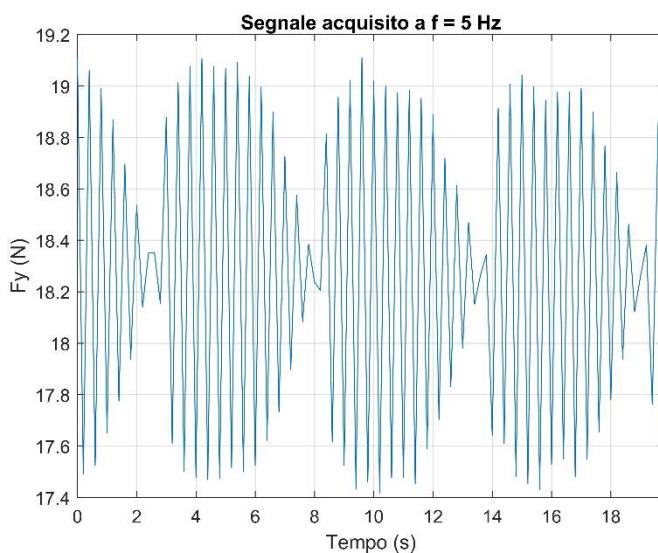


Figura 102 – Segnale con frequenza di acquisizione di 5 Hz

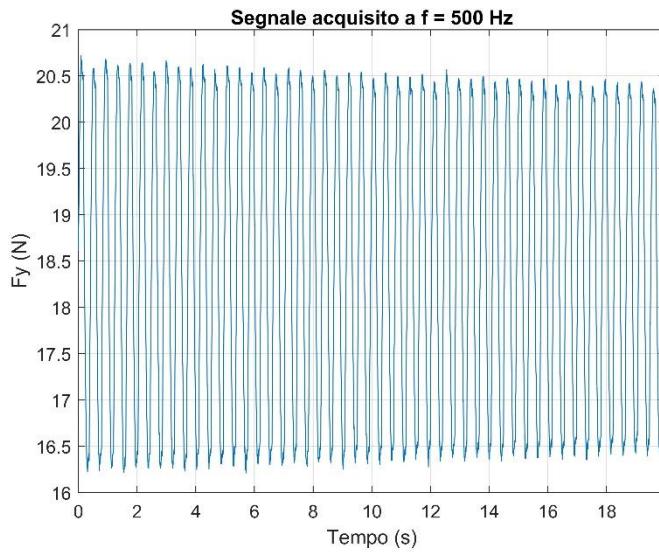


Figura 103 – Segnale con frequenza di acquisizione di 500 Hz

Dalle figure riportate si può vedere come con frequenza di acquisizione minore di 4,8 Hz non viene identificato nessun andamento riconoscibile; con frequenze appena superiori del limite imposto dal teorema è garantita l'identificazione del contenuto in frequenza ma non si è in grado di ricostruire la forma completa del segnale. Solo con frequenze di acquisizione almeno di due ordini superiori si riesce a ricavare invece correttamente sia il contenuto in frequenza sia l'informazione puntuale della forma del segnale.

Un'ulteriore analisi effettuata per la verifica del teorema di Nyquist è lo studio della trasformata di Fourier dei segnali acquisiti a diverse frequenze di campionamento.

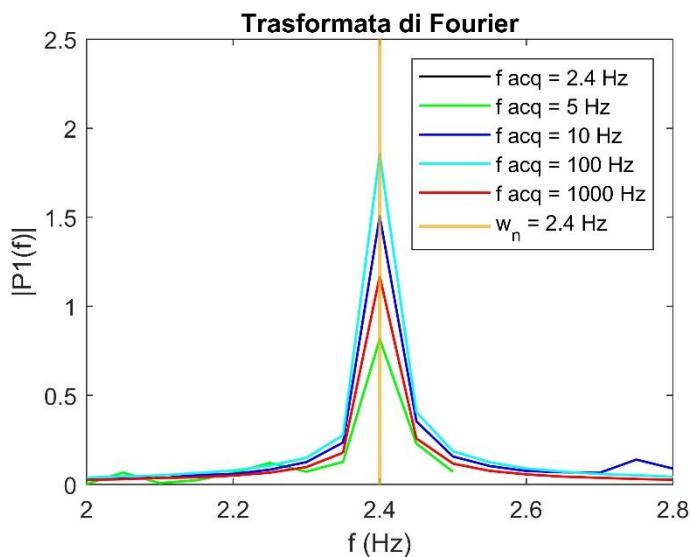


Figura 104 – Trasformata di Fourier dei segnali campionati a diverse frequenze di acquisizione

A conferma di quanto precedentemente detto il contenuto in frequenza principale (corrispondente alla frequenza di oscillazione del sistema pari circa a 2,4 Hz) è identificato correttamente da tutte le prove con frequenza di acquisizione maggiore del limite imposto dal teorema, pari al doppio del massimo contenuto in frequenza del segnale che si sta analizzando.

4.4 Modello Simulink

La stima dei parametri può essere eseguita anche su Simulink, l'ambiente di simulazione grafico di Matlab. Il modello fisico al quale si fa riferimento è sempre il medesimo; pertanto in questa sezione vengono omessi i passaggi già visti nei capitoli precedenti, soffermandosi invece sulle particolarità di questo approccio.

4.4.1 Creazione del modello

Occorre prima definire il modello numerico, in forma di schema a blocchi. In ingresso si hanno i 5 parametri che governano il sistema, in uscita si hanno i valori simulati di forza e momento trasmessi alla piattaforma.

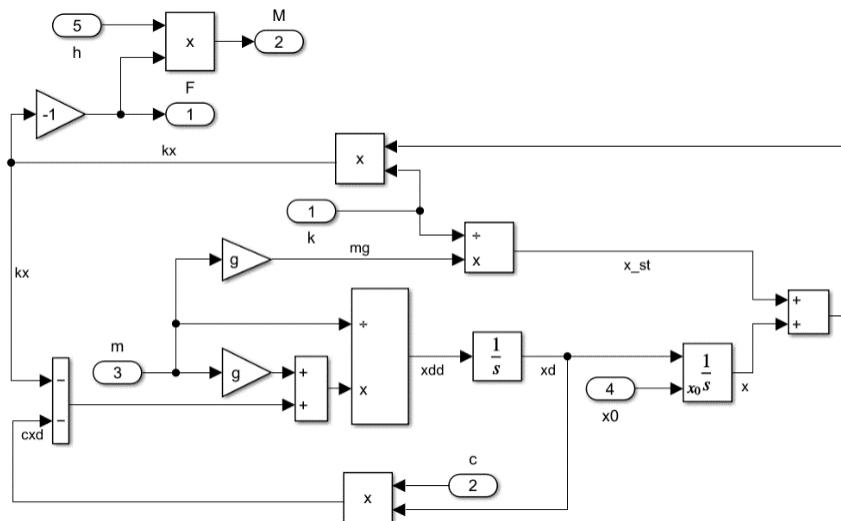


Figura 105 – Schema a blocchi del modello numerico

Questo schema a blocchi è stato poi racchiuso in un sottosistema, chiamato “Massa-molla-smorzatore”, ed inserito nella pagina principale del modello Simulink, il cui ruolo è quello di interfaccia con l'utilizzatore:

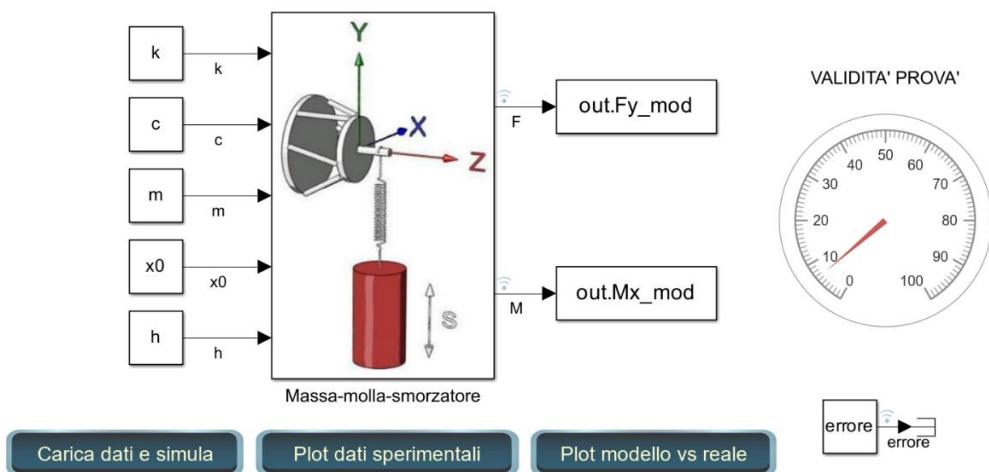


Figura 106 – Pagina principale del modello Simulink

Sono presenti tre pulsanti:

- “Carica dati e simula”: esegue il caricamento dei dati sperimentali e lancia la simulazione del modello. In particolare, esegue uno script Matlab che oltre ad effettuare le operazioni suddette contiene i valori dei parametri del sistema ed effettua i calcoli per la validità della prova (che verranno descritti più avanti);
- “Plot dati sperimentali”: visualizza tutte le sei componenti, tre di forza e tre di momento, rilevate dalla piattaforma;

- “Plot modello vs reale”: traccia i grafici che confrontano i dati simulati con quelli sperimentali.

Inoltre, sulla destra è stato inserito un indicatore a lancetta, intitolato “Validità prova”, che ora si va descrivere. Osservando il modello numerico, le uniche componenti di forza e momento che la piattaforma dovrebbe rilevare sono due: una forza verticale lungo l’asse y (F_y) ed un momento attorno all’asse x (M_x). Se tuttavia la massa non oscilla perfettamente lungo la verticale, nascono anche delle componenti “spurie” lungo le altre direzioni, sia di forza che di momento. Chiaramente questo non è desiderabile, in quanto rispecchia un comportamento che non è previsto dal modello e quindi porterebbe ad errori nella stima dei parametri. Allora per poter avere una valutazione di quanto queste grandezze spurie siano rilevanti rispetto a quelle presenti nel modello numerico, sono state calcolate quattro quantità d’errore nel modo seguente:

$$\begin{aligned} errore_{F_x} &= \frac{\|F_x\|}{\|F_y\|} \\ errore_{F_z} &= \frac{\|F_z\|}{\|F_y\|} \\ errore_{M_y} &= \frac{\|M_y\|}{\|M_x\|} \\ errore_{M_z} &= \frac{\|M_z\|}{\|M_x\|} \end{aligned}$$

Il massimo di questi valori viene poi visualizzato in percentuale nell’indicatore sopracitato. Qui sotto è stata riportata la porzione di codice dove viene eseguito questo calcolo.

```
%% CONTROLLO VALIDITA' PROVA
err_Fx=norm(Fx_exp)/norm(Fy_exp);
err_Fz=norm(Fz_exp)/norm(Fy_exp);
err_My=norm(My_exp)/norm(Mx_exp);
err_Mz=norm(Mz_exp)/norm(Mx_exp);
errore = (max([err_Fx err_Fz err_My err_Mz]))*100;
```

4.4.2 Stima dei parametri

Ora che il modello è stato definito e se ne è verificato il funzionamento, è possibile passare alla stima dei parametri. In Simulink è possibile utilizzare l’app “Parameter Estimator”, contenuta nel toolbox “Optimization”, che è stata creata appositamente per questo scopo. La procedura da seguire è la seguente:

- caricare i dati sperimentali (sotto forma di vettori);
- selezionare i parametri del sistema che devono essere stimati; è possibile anche definire un intervallo di valori entro il quale il parametro stimato deve ricadere;
- lanciare il processo di stima.

Il programma eseguirà una serie di iterazioni dove cercherà di minimizzare la funzione errore, in modo analogo a quanto visto con *fminsearch*. L’applicazione permette di vedere in tempo reale, iterazione dopo iterazione, come la risposta del modello si modifica al variare dei parametri stimati, e quindi avvicinandosi ai dati sperimentali. Inoltre, un grafico rappresenta l’andamento dei valori stimati nel corso del processo di stima.

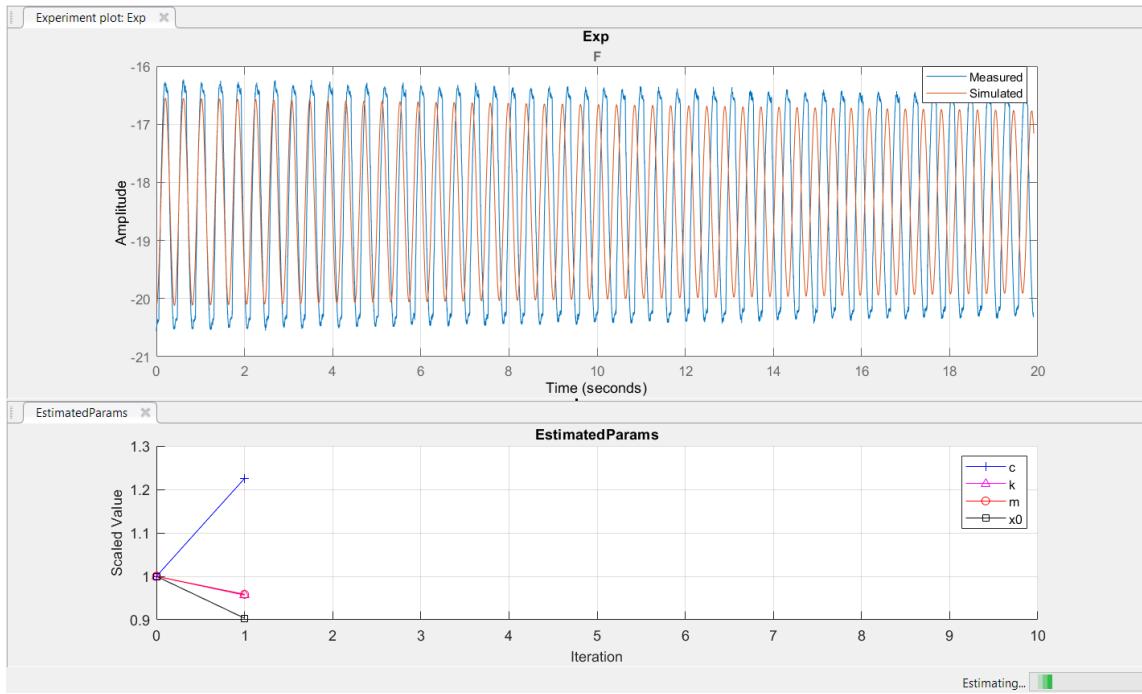


Figura 107 – Fase intermedia del processo di stima

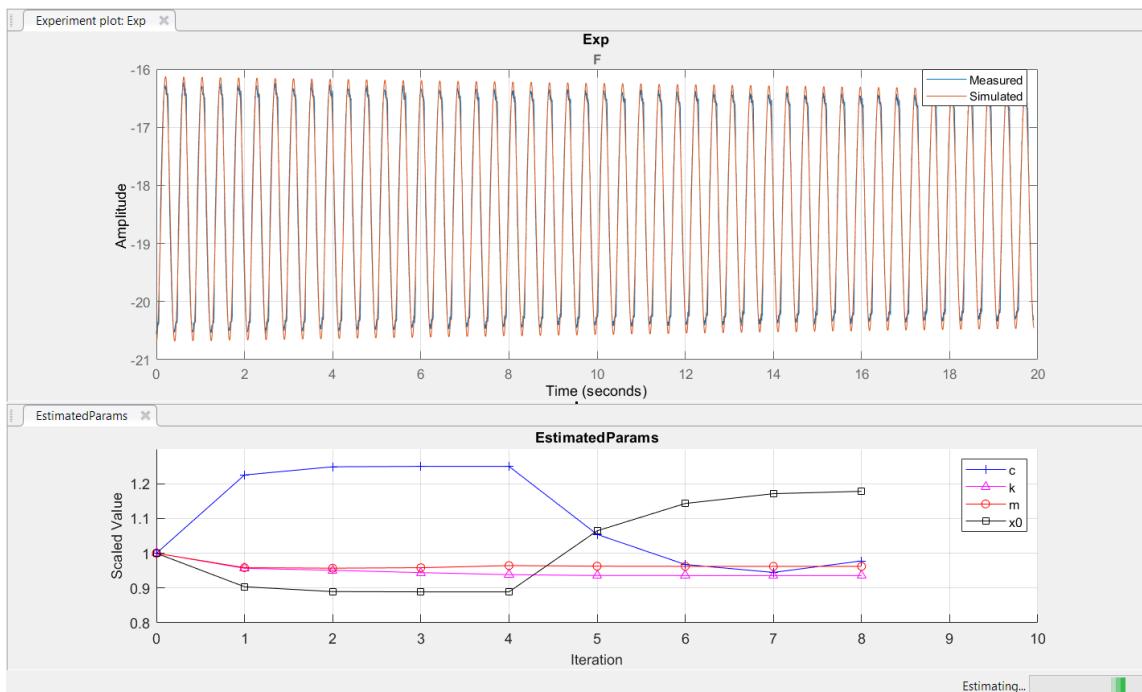


Figura 108 – Stima ultimata

L'app permette di scegliere l'algoritmo di ottimizzazione e in questo caso sono state lasciate le impostazioni di default, che utilizzano il metodo dei minimi quadrati non lineare e l'algoritmo Trust-region reflective; il funzionamento di questi esula dall'obiettivo dell'esercitazione. Inoltre, è possibile stabilire le condizioni che portano all'interruzione del processo di stima, attraverso dei valori di soglia:

- parameter tolerance: il processo di stima viene interrotto quando, tra un'iterazione e la successiva, i parametri stimati variano meno di questa soglia;
- function tolerance: il processo di stima viene interrotto quando, tra un'iterazione e la successiva, la funzione obiettivo varia meno di questa soglia.

4.4.3 Risultati ed osservazioni

I risultati ottenuti nelle varie prove sono i seguenti:

Tabella 6 – Risultati ottenuti su Simulink

	m [kg]	k [N/mm]	ζ [-]	x_0 [m]	ω_n [rad/s]	h [m]
Prova 1	1,86	428	0,0004	0,0027	15,19	0,036
Prova 2	1,88	430	0,0004	0,0053	15,14	0,035
Prova 3	1,86	425	0,0004	0,0037	15,13	0,037
Prova 4	1,86	426	0,0004	0,0042	15,15	0,037
Prova 5	1,85	427	0,0003	0,0021	15,20	0,037
Prova 6	1,89	432	0,0006	0,0047	15,11	0,032

Questi valori sono in linea con quelli stimati attraverso il codice Matlab. Le maggiori discrepanze si riscontrano nel calcolo dello smorzamento, che di per sé è difficile a causa del valore molto basso dello stesso.

In conclusione, si può dire che la stima dei parametri attraverso Simulink risulta più semplice ed immediata perché non è necessario scrivere le soluzioni dell'equazione differenziale del modello numerico: è sufficiente descrivere l'equazione stessa sotto forma di schema a blocchi. La soluzione dell'equazione viene ricavata numericamente, pertanto non è neanche necessario conoscerla in forma chiusa.

Tuttavia, il processo di stima su Simulink risulta più lento se confrontato con Matlab. Questo ha a che fare con l'elaborazione degli elementi grafici presenti nel modello, che rallentano inevitabilmente la stima. Inoltre, l'app consente di eseguire simultaneamente il fitting sulle due grandezze di interesse (forza F_y e momento M_x); tuttavia questo aumenta il costo computazionale, e quindi in questo caso si è preferito procedere con due step successivi.

4.5 Acquisizione dati di posizione tramite image processing

4.5.1 Obiettivi e setup

Parallelamente all'acquisizione dei dati di forze e coppie a partire dalle celle di carico del banco prova, ci si pone l'obiettivo di acquisire i dati di posizione della massa oscillante tramite un sistema di image processing. Utile osservare come sia di particolare interesse, nel panorama odierno dello studio della sensoristica, la capacità di ottenere delle rilevazioni affidabili tramite apparecchiature ottiche, le quali non generano disturbi nel sistema osservato. In questo particolare caso è stata utilizzata una comune webcam come apparecchiatura ottica.

Nella fase di setup iniziale è necessario impostare inizialmente l'algoritmo che si occupa della ricerca dei punti di interesse iniziali e di seguirli durante la prova. Per questa sperimentazione è stato scelto l'algoritmo KLT, anche detto di Viola-Jones, definito tramite ‘CascadeObjectDetector()’, il quale riesce a definire i suoi punti di interesse dall'analisi dei frame in bianco e nero. Successivamente si imposta il PointTracker, corrispondente a quello del “Computer Vision Toolbox”. Infine si definisce la webcam da utilizzare e si richiamano i frame raccolti e il VideoPlayer che visualizzerà il video.

```
%% Setup

% Create the detector object.
Detector = vision.CascadeObjectDetector();

% Create the point tracker object.
pointTracker = vision.PointTracker('MaxBidirectionalError', inf);

% Create the webcam object.
cam = webcam(1);    % 1=Internal camera; 2=External camera

% Capture one frame to get its size.
videoFrame = snapshot(cam);
frameSize = size(videoFrame);

% Create the video player object.
videoPlayer = vision.VideoPlayer('Position', [100 100 [1100, 600]]);
```

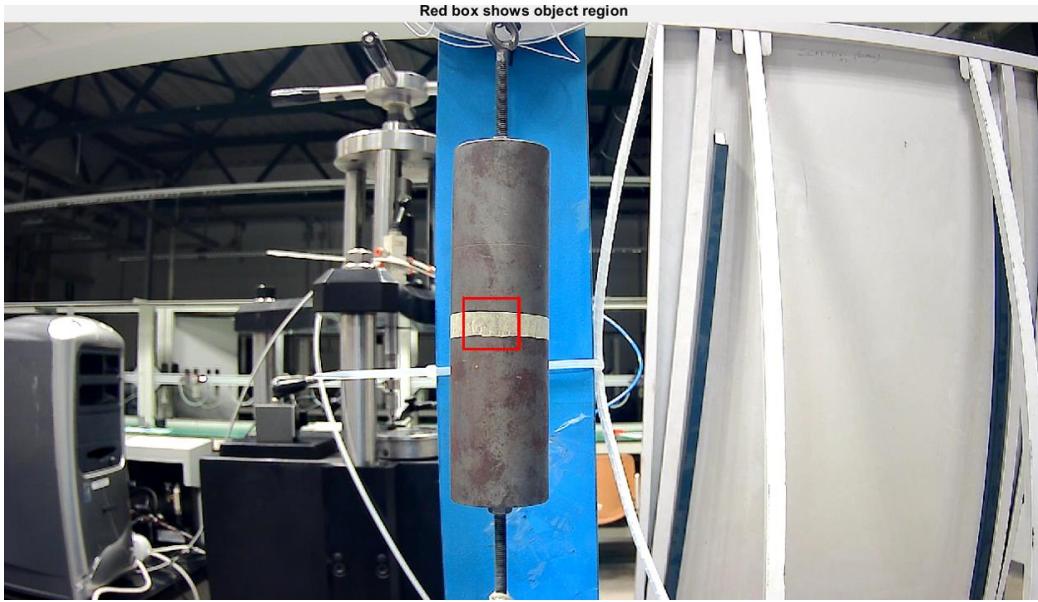
“Image_Processing.m” Pt.1 – Setup

Il codice sfrutta una preimpostata ‘ObjectRegion’ che permette di definire la zona di ricerca iniziale dei punti. Nella fase di setup si sfrutta un ulteriore script al fine di riuscire a definire correttamente la ObjectRegion ed il posizionamento della webcam.

```
objectRegion = [850,500,100,100];

objectFrame = videoFrame;
objectImage = insertShape(objectFrame, 'rectangle', objectRegion, 'Color', 'red');
figure;
imshow(objectImage);
title('Red box shows object region');
```

“Help_define_ObjectRegion.m”



4.5.2 Detection & tracking

Il core dello script è nel while loop, che analizza ogni frame raccolto dalla webcam. Nella prima fase inizializza il Detector, impostando poi la ricerca solo sulla ObjectRegion. Successivamente salva la posizione dei punti trovati, creando quindi una matrice che permette di visualizzare un rettangolo che indica l'orientazione dell'oggetto inquadrato.

```
%% Detection and Tracking
% Capture and process video frames from the webcam in a loop to detect and
% track a face. The loop will run until the video player window is closed.

runLoop = true;
numPts = 0;
frameCount = 0;

while runLoop

    % Get the next frame.
    videoFrame = snapshot(cam);
    videoFrameGray = rgb2gray(videoFrame);
    frameCount = frameCount + 1;

    % Define the region where the detector will search for points
    objectRegion = [850,500,100,100];      % Depends on the camera used

    if numPts < 10
        % Detection mode.
        bbox = Detector.step(videoFrameGray);

        if ~isempty(bbox)

            % Find corner points inside the detected region.
            points = detectMinEigenFeatures(videoFrameGray, 'ROI', objectRegion);

            % Re-initialize the point tracker.
            xyPoints = points.Location;
            numPts = size(xyPoints,1);
            release(pointTracker);
            initialize(pointTracker, xyPoints, videoFrameGray);

        end
    end
end
```

```

% Save a copy of the points.
oldPoints = xyPoints;

% Convert the rectangle represented as [x, y, w, h] into an
% M-by-2 matrix of [x,y] coordinates of the four corners. This
% is needed to be able to transform the bounding box to display
% the orientation of the object.
bboxPoints = bbox2points(objectRegion);

% Convert the box corners into the [x1 y1 x2 y2 x3 y3 x4 y4]
% format required by insertShape.
bboxPolygon = reshape(bboxPoints', 1, []);

% Display a bounding box around the detected face.
videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon, 'LineWidth', 3);

% Display detected corners.
videoFrame = insertMarker(videoFrame, xyPoints, '+', 'Color', 'white');

end

```

“Image_Processing.m” Pt.2 – Initialize

Dopo la prima ricerca di punti, confermando che essi siano almeno 10 per avere una certa affidabilità, si calcola un valore di soglia tra i valori di posizione dei punti di un frame e quello successivo tramite la funzione “estimateGeometricTransform2D”. È possibile regolare tale valore di soglia (‘MaxDistance’) per valutare sperimentalmente quale sia il più adatto: un valore basso implica maggiore accuratezza e precisione nel seguire i punti da frame a frame, ma rischia di essere di difficile applicazione pratica poiché l’attrezzatura utilizzata, l’oggetto da seguire e le condizioni visive dell’ambiente non sono molto favorevoli, portando dunque ad una facile perdita dei punti. Si continua sempre a visualizzare i punti considerati e a definirli dentro un rettangolo giallo per definirne l’orientazione. Infine si salva su una variabile separata la media dei valori di posizione verticale dei punti.

```

else
    % Tracking mode.
    [xyPoints, isFound] = step(pointTracker, videoFrameGray);
    visiblePoints = xyPoints(isFound, :);
    oldInliers = oldPoints(isFound, :);

    numPts = size(visiblePoints, 1);

    if numPts >= 10
        % Estimate the geometric transformation between the old points
        % and the new points.
        [xform, inlierIdx] = estimateGeometricTransform2D(
            oldInliers, visiblePoints, 'similarity', 'MaxDistance', 4);
        oldInliers = oldInliers(inlierIdx, :);
        visiblePoints = visiblePoints(inlierIdx, :);

        % Apply the transformation to the bounding box.
        bboxPoints = transformPointsForward(xform, bboxPoints);

        % Convert the box corners into the [x1 y1 x2 y2 x3 y3 x4 y4]
        % format required by insertShape.
        bboxPolygon = reshape(bboxPoints', 1, []);

        % Display a bounding box around the object being tracked.
        videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon, 'LineWidth', 3);

        % Display tracked points.
    end

```

```

videoFrame = insertMarker(videoFrame, visiblePoints, '+', 'Color', 'white');

% Reset the points.
oldPoints = visiblePoints;
setPoints(pointTracker, oldPoints);
end

% Save the position of the object for every frame
position(frameCount,1)=mean(visiblePoints(:,2));

end

```

“Image_Processing.m” Pt.3 – GeometricTransform

Alla fine del while loop si imposta l’apertura del VideoPlayer e contemporaneamente la variabile che termina il programma quando esso viene chiuso.

Per comodità di visualizzazione si imposta l’adattamento automatico dello schermo del VideoPlayer. Dovrebbe essere possibile anche la rimozione automatica dell’impostazione “*Drop Frames to improve performance*” che riduce i frame raccolti dal VideoPlayer, purtroppo però si è rivelato essere un bug di Matlab, o perlomeno della versione utilizzata (R2022a), dovendo dunque procedere manualmente.

```

% Display the annotated video frame using the video player object.
step(videoPlayer, videoFrame);

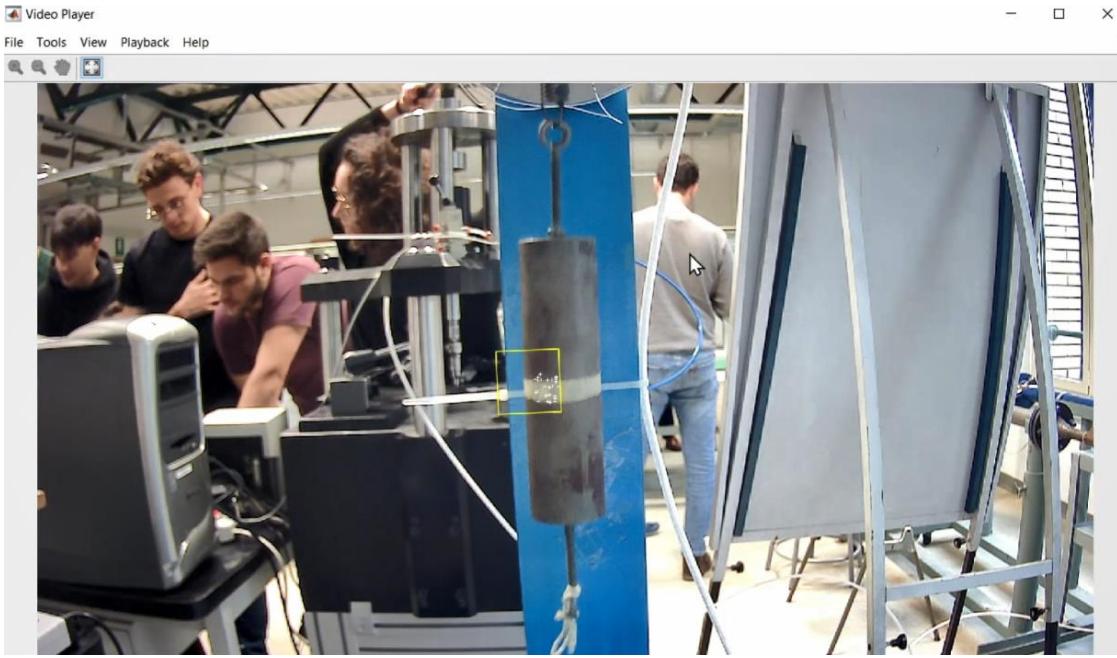
% Check whether the video player window has been closed.
runLoop = isOpen(videoPlayer);

% Check the videoPlayer option "Maintain fit to window".
% Should also work for the "Drop frames to improve performance", but
% that is a videoPlayer bug.
if frameCount==1
    set(0,'showHiddenHandles','on');
    fig_handle = gcf;
    fig_handle.findobj; % to view all the linked objects with the vision.VideoPlayer
    ftw = fig_handle.findobj ('TooltipString', 'Maintain fit to window');
    ftw.ClickedCallback();
    % df = fig_handle.findobj ('Tag','uimgr.uimenu_SynchronousLock')
    % df.Checked='off';
end

% Clean up.
clear cam;
release(videoPlayer);
release(pointTracker);
release(Detector);

```

“Image_Processing.m” Pt.4 – Visualizzazione VideoPlayer



4.5.3 Plot e salvataggio

Nel momento in cui il VideoPlayer viene chiuso manualmente, il while loop termina. A questo punto si analizza il vettore posizione ricavato. In prima analisi si inverte rispetto all'asse, in quanto il programma salva la posizione a partire dal lato in alto del VideoPlayer. Sotto forma di commento vi è il procedimento che permette la conversione dell'unità di misura della posizione sotto forma di correlazione lineare tra pixel e centimetri. Purtroppo non è stato possibile effettuarla per difficoltà pratiche riscontrate. Si procede quindi con la creazione del vettore temporale, in modo da poter effettuare un plot dei dati ricavati. Infine si salvano i dati di posizione in un file testo poiché servirà al programma di post-processing.

```
%% Plot

position(:,1)=frameSize(1)-position(:,1);

%position(pixel)-->position(cm)
% Pos1pix=150;
% Pos2pix=610;
% Pos1cm=33;
% Pos2cm=59;
% K=(Pos2cm-Pos1cm)/(Pos2pix-Pos1pix);
% q=Pos1cm-K*Pos1pix;
% position=position.*K + q;

%frameCount-->seconds (see fps)
fps=30;
sec=frameCount/fps;
seconds=linspace(0,sec,(size(position,1)))';

% Plot position.
figure
hold on
xlabel('Time (s)')
ylabel('Position (cm)')
plot(seconds,position)

%Save data to a .txt file
A = [seconds';position'];
fileID = fopen('Prova1.txt','w+');
fprintf(fileID, '%6.2f %12.8f\n',A);
```

```
fclose(fileID);
```

“Image_Processing.m” Pt.5 – Plot e salvataggio

4.5.4 Prova sperimentale

Si esplicitano nel seguito i vari passaggi eseguiti per compiere interamente la procedura sperimentale di acquisizione dei dati:

- Preparazione del banco di prova, comprendendo il sistema massa-molla (si è rivelato utile apporre del nastro carta sulla massa in modo da agevolare la rilevazione da parte dell'algoritmo), un tavolo aggiuntivo su cui posizionare il computer ed un treppiede per il posizionamento della webcam alla stessa altezza della massa appesa;
- Posizionamento corretto dell'inquadratura della webcam, ottenuta tramite l'utilizzo dello script ausiliario “Help_define_ObjectRegion” precedentemente descritto;
- Avviamento del codice “Image_Processing”, assicurandosi di rimuovere l'impostazione “*Drop Frames to improve performance*”, e che i punti visibili siano sulla massa;
- Messa in moto manuale della massa, osservando che i punti continuino a seguire fedelmente la massa oscillante;
- Chiusura del VideoPlayer e salvataggio dei dati di posizione acquisiti.

4.6 Fitting sui dati sperimentali dell'acquisizione immagini

4.6.1 Modello teorico

Similmente a come è stato fatto per i dati acquisiti dalle celle di carico della piattaforma di Stewart si vuole eseguire il fitting dei dati sperimentali per validare il modello teorico scelto per rappresentare il sistema e per calcolare i parametri di tale modello.

Il sistema fisico del giunto a sei assi con massa-molla oscillante è lo stesso già descritto nelle parti precedenti, l'unica cosa che cambia è il sistema di acquisizione dati. Nella parte di fitting precedentemente descritta venivano acquisiti dei segnali in tensione dalle celle di carico, poi opportunamente convertiti in forze e momenti tramite LabView. In questa sezione invece vengono utilizzati i dati relativi alla posizione della massa derivanti dal codice Matlab dell'acquisizione immagini. Vista la diversa natura dei dati sperimentali acquisiti ne consegue che anche il modello teorico con cui questi vengono confrontati debba essere diverso.

Nella pratica il modello risulta molto semplice e soprattutto facilmente riconducibile al modello usato nel fitting in forza dei dati. Se infatti si considera un semplice modello massa-molla-smorzatore come in Figura 48, possiamo descrive la sua posizione nel tempo come segue:

$$y(t) = y_0 \sin(\omega_d t - \varphi) e^{-\xi \omega_n t}$$

dove:

- y_0 ampiezza dell'oscillazione compiuta dalla massa
- ω_d pulsazione smorzata del sistema
- φ fase del segnale
- ξ fattore di smorzamento
- ω_n pulsazione naturale

Questo diventa quindi il modello teorico del quale andremo a cercare i parametri attraverso il fitting dei dati in posizione raccolti dal codice dell'acquisizione immagini. Si può inoltre notare come il modello di questa procedura è molto simile al fitting in forza, in quel caso il modello è un equilibrio di forze sul vincolo della molla composto dalla forza elastica (derivante dalla y sopra descritta) dalla reazione della piattaforma F_y e dal peso della massa.

$$F_y(t) = mg + ky(t)$$

4.6.2 Script Matlab per il fitting dei dati

Di seguito verrà riportata la descrizione del codice Matlab "fitting_ImageProcessing.m" usato per validare il modello teorico e calcolarne i parametri caratteristici.

La prima cosa che il codice Matlab fa è caricare i dati da un file .txt su cui sono salvati posizione della massa e tempo, questi valori derivano a loro volta dallo script "Image_Processing.m" con cui viene effettuata l'acquisizione da video. Dopo aver caricato i dati questi sono salvati in un vettore y per le posizioni e un vettore tempo per i relativi istanti temporali. Dopo questa sezione è presente un plot del vettore y che permette di visualizzare l'andamento della posizione. Successivamente sono presenti alcune righe che permettono di tagliare il segnale se dal precedente plot questo risultasse essere non omogeneo o presentasse dei momenti non utilizzabili in quanto comprensivi della parte in cui si inizia a far oscillare la massa. Infine, è possibile applicare un filtro al segnale di posizione appena salvato. La porzione di codice appena commentata è riportata di seguito:

```
%> ACQUISIZIONE DATI  
% carico dati
```

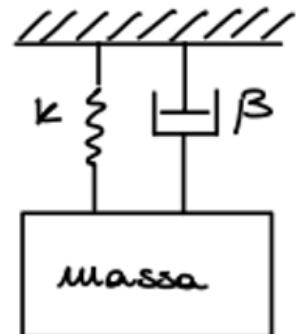


Figura 109 – Schema modello massa-molla-smorzatore

```

Folder = 'Acquisizioni';
FileName = 'Prova1.txt';

dati = importdata([Folder,filesep,FileName]);

% frequenza di acquisizione, la frequenza è di 30 Hz poichè deriva dalla
% webcam utilizzata
f_acq = 30; % Hz

%Lettura dati per prova con il tempo:
tempo = dati((1:end),1);
y = dati((1:end),2);

% %DATI SOLO PER LA PROVA SENZA TEMPO:
% y = dati((1:end),1);
% tempo = (0 : 1/f_acq*ones(size(dati(:,1))) : 1/f_acq*(length(dati(:,1))-1))'; % vettore
tempi

%PRIMO PLOT UTILE PER VISUALIZZARE E ELIMINARE I DATI NON RILEVANTI PER LA MISURA
figure(1)
plot(y),grid

%SE NECESSARIO TAGLIARE y e prendere il segnale dove esso risulta più
%pulito (in caso di interferenze con le rilevazioni). Dopo dichè si porta
%il segnale di posizione a oscillare attorno allo zero per praticità e si
%riazzerà il tempo a seconda del taglio effettuato
y = y(1500:2600);
tempo = tempo(1500:2600);
y = y -mean(y);
tempo = tempo - tempo(1);

%Viene applicato un filtro ai dati acquisiti in caso questi presentino
%disturbi molto rilevanti
y_f=movmean(y,1);

```

"fitting ImageProcessing.m" Pt.1 – Caricamento dati, taglio e filtro del segnale

Dopo aver tagliato ed eventualmente filtrato il segnale viene effettuata un'analisi di Fourier del segnale $y(t)$ attraverso la funzione `fft` di Matlab. In un subplot di Matlab sono poi stampati la y dopo il taglio in funzione del tempo e un solo lato dello spettro di potenza in funzione delle frequenze.

```

%% ANALISI IN FREQUENZA DEL SEGNALE
%Trasformata di Fourier
Y = fft(y);
L=length(y);
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);      %Un lato dello spettro di potenza della trasformata
di fourier
f = f_acq*(0:(L/2))/L;

%Stampa di y analizzato nelle fft e periodi e frequenze presenti nel
%segnale analizzato
figure(2)
subplot(2,1,1)
plot(tempo,y),grid
title('Posizione y')

```

```

xlabel('tempo [s]')
subplot(2,1,2)
loglog(f,P1),grid
title('Spettro delle frequenze del segnale y(t)')
xlabel('Frequenza [Hz]')

```

[“fitting ImageProcessing.m” Pt.2 – Analisi di Fourier del segnale y\(t\)](#)

Operativamente lo script Matlab che esegue il fitting del modello teorico ha bisogno di un set di parametri iniziali da cui iniziare la ricerca per poter ottenere più facilmente un risultato veritiero e minimizzare i tempi di calcolo. Nel codice è quindi inserita una sezione in cui sono inseriti dei dati rilevati in laboratorio e viene poi fatta una basilare analisi del segnale y per ricavare alcuni di questi parametri iniziali da fornire in input alla funzione che esegue il fitting.

Il coefficiente di smorzamento del segnale di posizione y(t):

Nello specifico viene valutato il decremento logaritmico sul segnale per poter calcolare il coefficiente di smorzamento del segnale. Il decremento logaritmico è utile per avere un valore di primo tentativo di coefficiente di smorzamento. Questa procedura fa risparmiare importanti iterazioni al modello di fitting perché come si è detto nell'introduzione a questo capitolo si è deciso di non misurare sperimentalmente in laboratorio il coefficiente di smorzamento perché troppo complicato e poco preciso.

La formula utilizzata per il calcolo del decremento logaritmico è la seguente:

$$\delta = \frac{1}{n} \ln \frac{y_i}{y_{i+n}}$$

dove x_i rappresenta il valore del picco i-esimo del segnale e x_{i+n} il picco $i+n$ -esimo, per misurare meglio lo smorzamento, infatti, si prendono due ampiezze lontane un numero di n periodi tra loro. Con il valore del decremento logaritmico è poi possibile calcolare il valore del fattore di smorzamento δ e quindi del coefficiente di smorzamento β con i seguenti passaggi:

$$\zeta = \frac{\delta}{\sqrt{4\pi^2 + \delta^2}}$$

$$\beta = 2\zeta\omega_n m$$

L'ampiezza del segnale di posizione y(t):

Per il calcolo dell'ampiezza è necessario anche individuare i picchi negativi, ovvero i minimi relativi di ogni periodo di oscillazione. I picchi negativi vengono salvati da MATLAB come un valore negativo e quindi è necessario prendere i picchi negativi con segno opposto per il calcolo della fase che viene prendendo in considerazione il primo picco positivo e il primo picco negativo, ovvero i due picchi che risentono meno dello smorzamento dovuto all'attrito fluidodinamico con l'aria. Quindi il calcolo preliminare della fase viene eseguito con la seguente formula:

$$A_0 = \frac{y_1^+ - y_1^-}{2}$$

Dove y_1^+ è il valore del primo picco positivo e y_1^- è il valore del primo picco negativo.

Fase del segnale di posizione y(t):

La fase del segnale serve per poter correttamente sincronizzare il segnale sperimentale con quello creato per il fitting dei dati. Prima di calcolare la fase viene calcolata la pulsazione smorzata del segnale, questo viene fatto a partire dai dati sperimentali invertendo il periodo del segnale calcolato su n picchi.

$$T_s = \frac{t_{n+i} - t_i}{n} \quad \omega_s = \frac{2\pi}{T_s}$$

Per calcolare la fase viene localizzato il primo picco del segnale e avendo calcolato il tempo t_1 in cui questo si verifica lo si può moltiplicare per la pulsazione smorzata ω_s a cui sta oscillando il segnale per capire la fase del primo picco in radianti. Togliendo $\pi/2$ si ottiene la fase φ_0 del segnale.

$$\varphi_0 = t_1 \omega_s - \frac{\pi}{2}$$

```
%% Parametri da controllare
%Dati misurati sperimentalmente
m=2.0417; % massa [kg]
k=484; %rigidezza molla [N/m]
g=9.81; %Accelerazione di gravità [m/s^2]
f_acq=30; %[Hz] % fps della webcam
w_n=sqrt(k/m); %frequenza naturale 15.39 (rad/s)
T=2*pi/w_n; %Periodo di oscillazione

%% PRE-ANALISI
%Vengono stimati alcuni parametri di prima ipotesi var_hp da utilizzare come var_in per fminsearch

%A) Decremento logaritmico
%Massimi
[pks,loc]=findpeaks(y_f,f_acq,"MinPeakDistance",0.9*T);

figure(3)
plot(tempo, y, 'r',tempo, y_f, 'g', loc, pks, 'bo')
legend("Dati sperimentali","Modello con media mobile","Picchi")

n=min([50, length(pks)]); %numero di picchi da considerare (prendo il minimo tra n e size pks)
delta=1/n*log(pks(1)/pks(n+1));
zeta_dec_log=delta/(sqrt(4*pi^2+delta^2));
beta_dec_log=2*zeta_dec_log*w_n*m;

%B) Ampiezza
%Minimi
[pks_min,loc_min]=findpeaks(-y_f,f_acq,"MinPeakDistance",0.9*T);
pks_min=-pks_min;

figure (3)
hold on
plot(loc_min + tempo(1), pks_min, 'ko')
legend("Dati sperimentali","Modello con media mobile","Picchi","Minimi")

%Mi calcolo un'ampiezza di primo tentativo mediata su n dati
y0_hp=(pks(1)-pks_min(1))/2; %Doppio meno perchè pks_min è un valore negativo

%C) Fase
n=50; %numero di picchi da considerare
Ts_hp=(loc(n+1)-loc(1))/n; %Periodo (media primi n picchi)
```

```

ws_hp=2*pi/Ts_hp;
ws_dec_log=w_n*sqrt(1-zeta_dec_log^2);
[max_idx, idx]=max(P1(10:end));
ws_ttf=f(9+idx)*2*pi;

phi_hp=loc(1)*ws_hp-pi/2;
if phi_hp<0
    phi_hp=2*pi+phi_hp; %+ perchè phi è negativo
end

```

“fitting ImageProcessing.m” Pt.3 – Calcolo dei parametri iniziali da fornire alla funzione che esegue il fitting

Calcolati i parametri di primo tentativo necessari si passa alla procedura di fitting del modello che viene eseguita utilizzando una funzione Matlab *fminsearch*. La funzione *fminsearch* trova il minimo di una funzione multivariabile utilizzando un metodo derivativo (derivative-free method). Per utilizzarla nel migliore dei modi viene creato un vettore contenente i valori dei parametri su cui andrà eseguita la procedura di fitting quali massa, coefficiente di smorzamento, rigidezza della molla, fase e ampiezza. I valori inseriti in questo vettore sono tutti i valori di primo tentativo trovati sperimentalmente (massa e rigidezza della molla) o attraverso una pre-analisi sul segnale di posizione y (fase, ampiezza e coefficiente di smorzamento).

Costruito il vettore con le variabili si scrive la formula della posizione della molla nel sistema massa-molla-smorzatore che rappresenta il modello teorico e si crea successivamente una funzione errore della posizione della massa ottenuta con il modello e con il segnale sperimentale. Ora è possibile utilizzare la funzione *fminsearch* direttamente sull'errore, ovvero per cercare di minimizzare la funzione errore. La funzione quindi modificando i parametri inseriti nel vettore attraverso successive iterazioni troverà una combinazione di valori con i quali, secondo essa, il modello si avvicina il più possibile ai valori sperimentali.

```

%% FITTING PARAMETRI
%POSIZIONE DELLA MASSA
%NOTA: var=[m, beta, k, y0, phi]
%Equazione modello con variabile vettoriale. Sono riportati due modelli con
%anneesse variabili, uno con i parametri e l'altro con frequenza naturale e
%coefficiente di smorzamento. Decommentare per usare l'altro modello
y_sim=@(var, tempo) var(4)*sin(sqrt(var(3)/var(1)))*sqrt(1-
(var(2)/(2*sqrt(var(3)/var(1))*var(1)))^2).*tempo-var(5)).*exp(-
(var(2)/(2*sqrt(var(3)/var(1))*var(1)))*sqrt(var(3)/var(1)).*tempo));
% y_sim=@(var, tempo) var(3)*sin((var(1)*sqrt(1-(var(2))^2)).*tempo-var(4)) .* exp(-
var(2)*var(1).*tempo);

%Valori primo tentativo di test
m_in=m;
beta_in=beta_dec_log;
k_in=k;
y0_in=y0_hp;
phi_in=phi_hp;
w_n_in = w_n;
zeta_in=zeta_dec_log;

var_in=[m_in, beta_in, k_in, y0_in, phi_in]; %vettore variabili primo tentativo
% var_in=[w_n_in,zeta_in,x0_in,phi_in]; %vettore variabili primo tentativo

%Funzione errore - fminsearch
err= @(var) norm(y_sim(var,tempo)-y);
option= optimset('Display','iter','TolFun',1e-6,'TolX',1e-6);
var_opt= fminsearch(err,var_in,option); %parametri di fitting
disp("Confronto var in - opt")
disp(["m ", "beta", "k ", "y0 ", "phi "])
% disp(["w_n ", "zeta ", "y0 ", "phi "])
disp([var_in;var_opt])

```

```
%Comparazione_in_exp_opt=[var_in;var_exp;var_opt]

figure (4)
plot(tempo, y_sim(var_opt,tempo)+mean(y))
hold on
plot(tempo, y, '-')
legend('Fitting','Dati sperimentali')
```

“fitting ImageProcessing.m” Pt.4 – Fitting dei dati sperimentali

Nell’ultima porzione di codice è poi riportato il codice Matlab che prova a variare i parametri in ingresso alla *fminsearch* per assicurarsi che la minimizzazione trovata della funzione non sia una situazione di minimo locale. In nessuna delle prove eseguite si è registrata una differenza tra il primo fitting e questa prova di variazione dei parametri iniziali.

```
%% FOR VAR_OPT
for j=1:10
    %Valori primo tentativo di test
    m_in=m*(100+randn)/100;
    beta_in=beta_dec_log;
    k_in=k*(100+randn)/100;
    y0_in=y0_hp;
    phi_in=phi_hp;

    var_in_for(j,:)=[m_in,beta_in,k_in,y0_in,phi_in]; %vettore variabili primo tentativo
%     var_in_for(j,:)=[w_n_in,zeta_in,y0_in,phi_in]; %vettore variabili primo tentativo

    %Funzione errore - fminsearch
    err= @(var) norm(y_sim(var,tempo)-y);
    option= optimset('TolFun',1e-3,'TolX',1e-3);
    var_opt_for(j,:)= fminsearch(err,var_in,opt); %parametri di fitting
end

disp("      Confronto var in - opt CICLO FOR")
disp(["m ", "beta", "k ", "y0 ", "phi "])
% disp(["w_n ", "zeta ", "y0 ", "phi "])
disp([mean(var_in_for);mean(var_opt_for)])
```

“fitting ImageProcessing.m” Pt.5 – Ciclo For di ottimizzazione del fitting

4.6.3 Analisi del segnale $y(t)$ e commento dei risultati

Vengono ora analizzati i dati $y(t)$ derivanti dall’acquisizione immagini e verranno commentati i risultati ottenuti tramite la procedura del fitting dei dati. In Figura 49 è possibile osservare il segnale di $y(t)$ per come è stato acquisito. Si evidenzia come i valori assoluti della misura siano in pixel, non è quindi stata implementata una conversione efficacie dei segnali acquisiti dal codice “Image_processing.m”. Questo risulta sicuramente un punto di miglioramento per future esercitazioni svolte sul banco del giunto a sei assi.

Nonostante non sia stato possibile effettuare una conversione efficacie dei segnali letti è comunque possibile effettuare l’analisi del segnale per valutare il modello teorico. Questo risulta infatti essere un modello a parametri concentrati e non risente quindi del valore assoluto in ampiezza delle oscillazioni che la massa subisce.

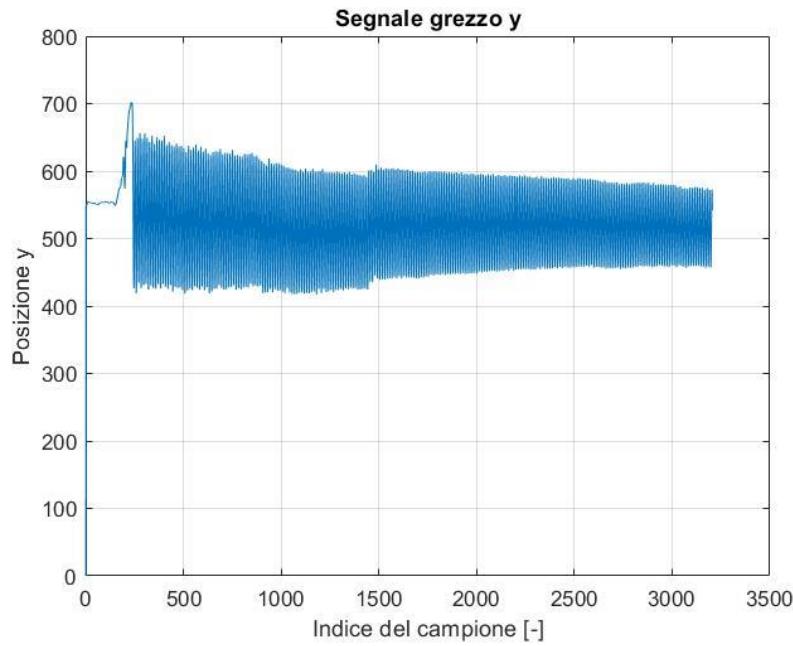


Figura 110 – Dati grezzi acquisiti dal codice “Image_processing.m”

Come si può osservare in Figura 49 i dati acquisiti dalla webcam non sono utilizzabili subito, questo perché per garantire una corretta acquisizione dati la massa deve partire da ferma. In ogni prova sarà quindi necessario tagliare una parte di segnale. Nella prova presentata nel grafico inoltre c’è una prima parte in cui i dati sembrano essere poco affidabili, probabilmente a causa dell’eccessiva ampiezza di oscillazione fornita. Si esegue quindi un taglio del segnale tra l’indice 1500 e 2600, dove l’acquisizione è riuscita meglio.

In Figura 50 viene invece riportato il segnale di $y(t)$ dopo il taglio nella parte in alto, mentre nella parte in basso viene riportata la densità spettrale di potenza del segnale $y(t)$ alle varie frequenze ottenuto dall’analisi di Fourier. Si può osservare che il segnale non presenta particolari disturbi e il picco corrisponde alla frequenza smorzata a cui sta oscillando il sistema, nella pratica molto simile alla frequenza naturale visto il basso smorzamento. Di seguito è riportato un confronto tra la frequenza naturale teorica e la frequenza smorzata calcolata con l’analisi di Fourier:

$$\omega_n = 15.4 \left[\frac{\text{rad}}{\text{s}} \right] \quad \omega_s = 15.24 \left[\frac{\text{rad}}{\text{s}} \right]$$

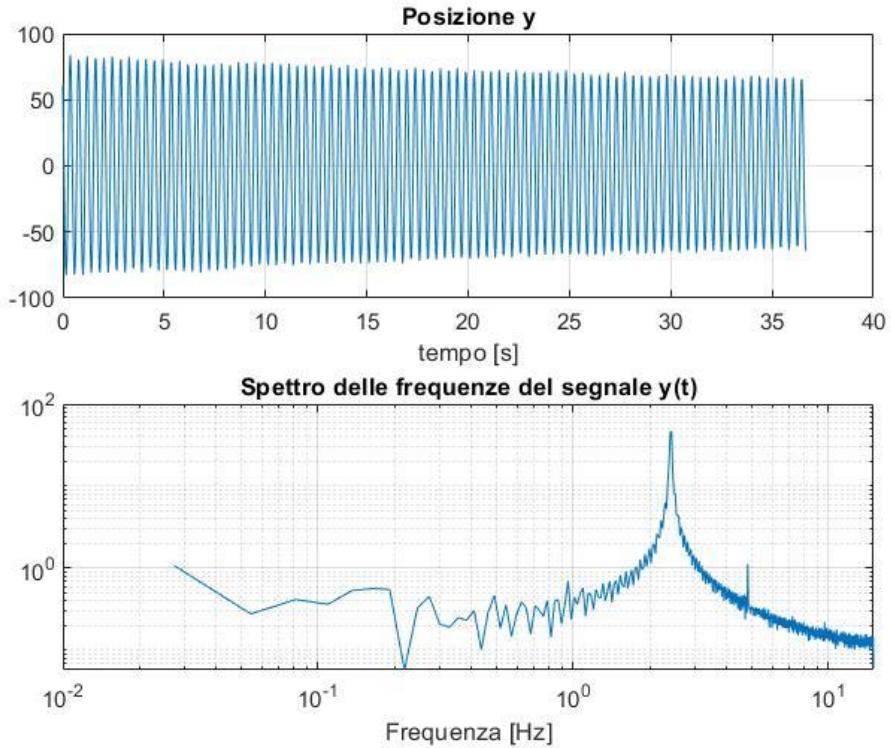


Figura 111 – Segnale $y(t)$ tagliato e Analisi spettrale del segnale

In Figura 51 sono invece riportate la y sperimentale tagliata, la y filtrata con il filtro a media mobile e i picchi positivi e negativi del segnale ricavati sul segnale filtrato. Nello specifico di questa acquisizione nessun filtro è stato applicato vista la bontà del segnale, e quindi i due andamenti sono sovrapposti.

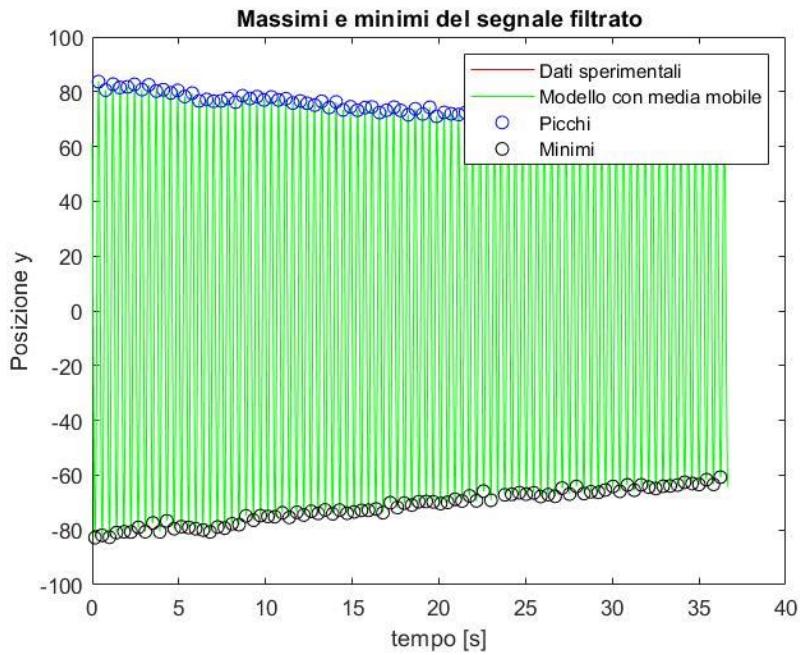


Figura 112 – y sperimentale e filtrata con i picchi del segnale filtrato

Infine, è riportato in Figura 52 l'andamento dei dati sperimentali con il modello teorico costruito con i parametri trovati tramite la funzione fminsearch di Matlab. Come si può osservare il modello rispecchia molto fedelmente i dati sperimentali. Non verrà discussa la modalità e le finalità del fitting in quanto sono le stesse già spiegate per il modello in forza del sistema.

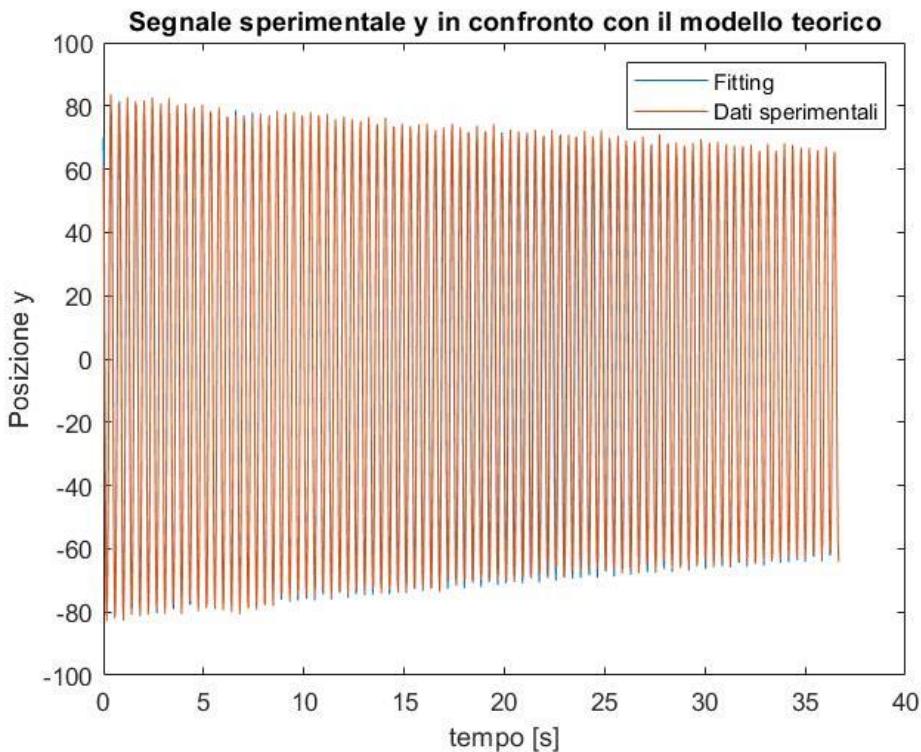


Figura 113 – Modello teorico e modello sperimentale

Di seguito è riportata una tabella con i risultati del fitting del modello teorico in base ai dati sperimentali raccolti durante tre prove distinte per dimostrarne la stabilità ed eventuali limiti. Il primo segnale è quello di cui sono stati mostrati e commentati alcuni risultati sopra.

Per gli altri due segnali si riportano brevemente i grafici mostrati in precedenza per il primo segnale per mostrare come questi sono diversi e a tratti meno di qualità rispetto al primo, questo per far vedere che possono essere usati anche segnali più scarsi applicando i dovuti tagli e filtri.

In Figura 53 è riportata la seconda prova di acquisizione eseguita, si specifica che si tratta sempre di una prova fatta a 30 Hz. Il segnale è stato tagliato tra i campioni 260 e 1060 per eliminare il primo picco fuori scala e l'ultima parte dei dati dove probabilmente il software ha perso e riacquisito rapidamente l'obiettivo facendo scalare leggermente la media del segnale. I picchi fuori scala visibili nel segnale sono causati da una perdita del target da parte del codice di acquisizione dati che non viene subito compensata.

Per questo segnale è stato impostato un filtro a media mobile su 5 campioni per volta, per cercare di abbassare il picco del segnale. Non è possibile applicare filtri a media mobile molto maggiori in quanto avendo solo 30 campioni al secondo il segnale perde di ampiezza molto rapidamente, cosa che influenza negativamente il procedimento di fitting.

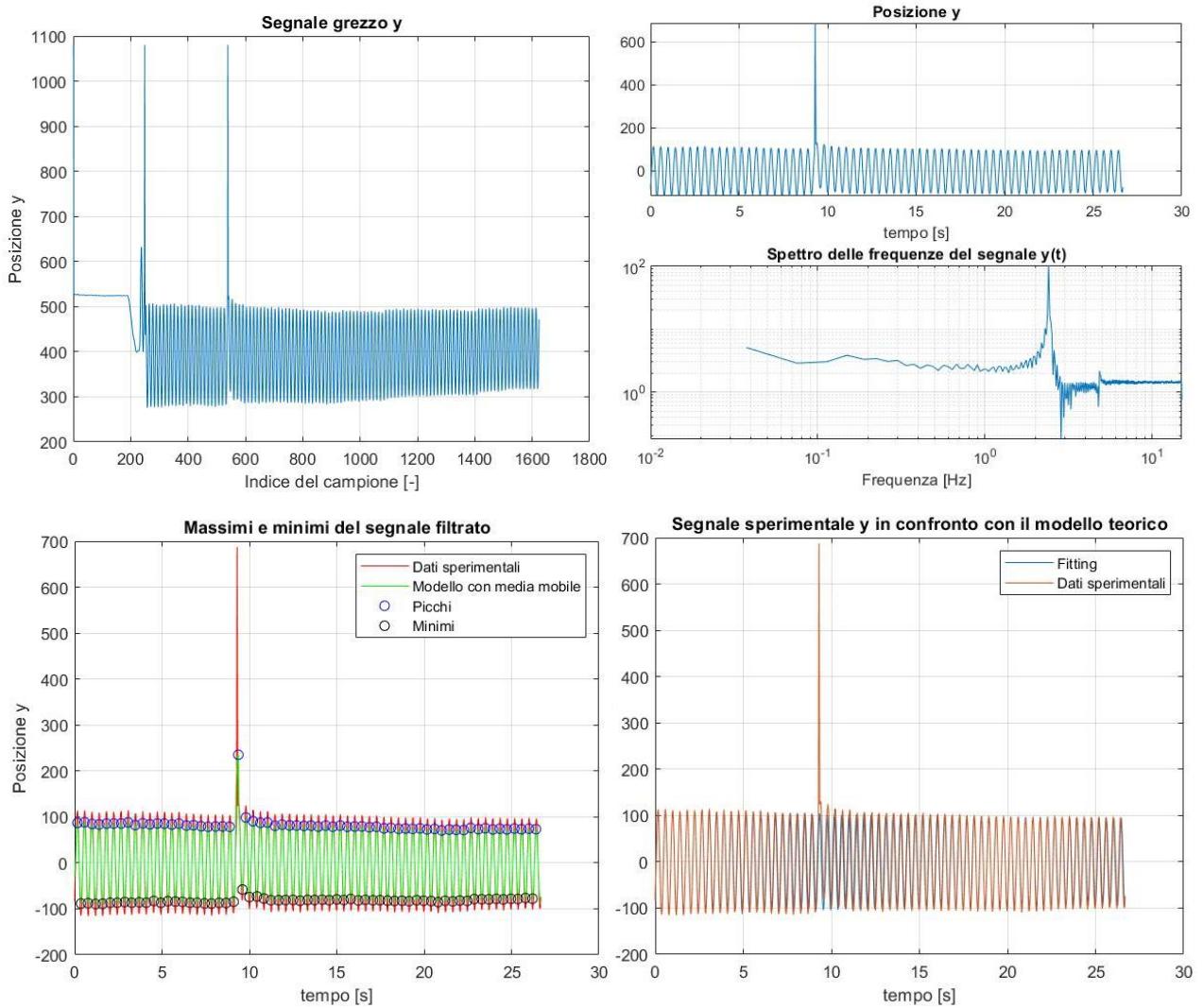


Figura 114 – Prova di acquisizione numero due

In Figura 54 è invece riportata una serie di immagini relative al terzo segnale che verrà trattato. Anche questo è a 30 Hz e con un filtro a media mobile su cinque elementi, il segnale è inoltre stato tagliato dal campione 900 fino alla fine. Su questo segnale il problema è invece una sorta di media del segnale decrescente nel tempo dovuto probabilmente a un leggero spostamento del tracker del sistema di acquisizione per cause sconosciute, forse un cambio nell'illuminazione della massa oscillante. Anche su questo segnale è stato applicato un filtro a media mobile su cinque valori.

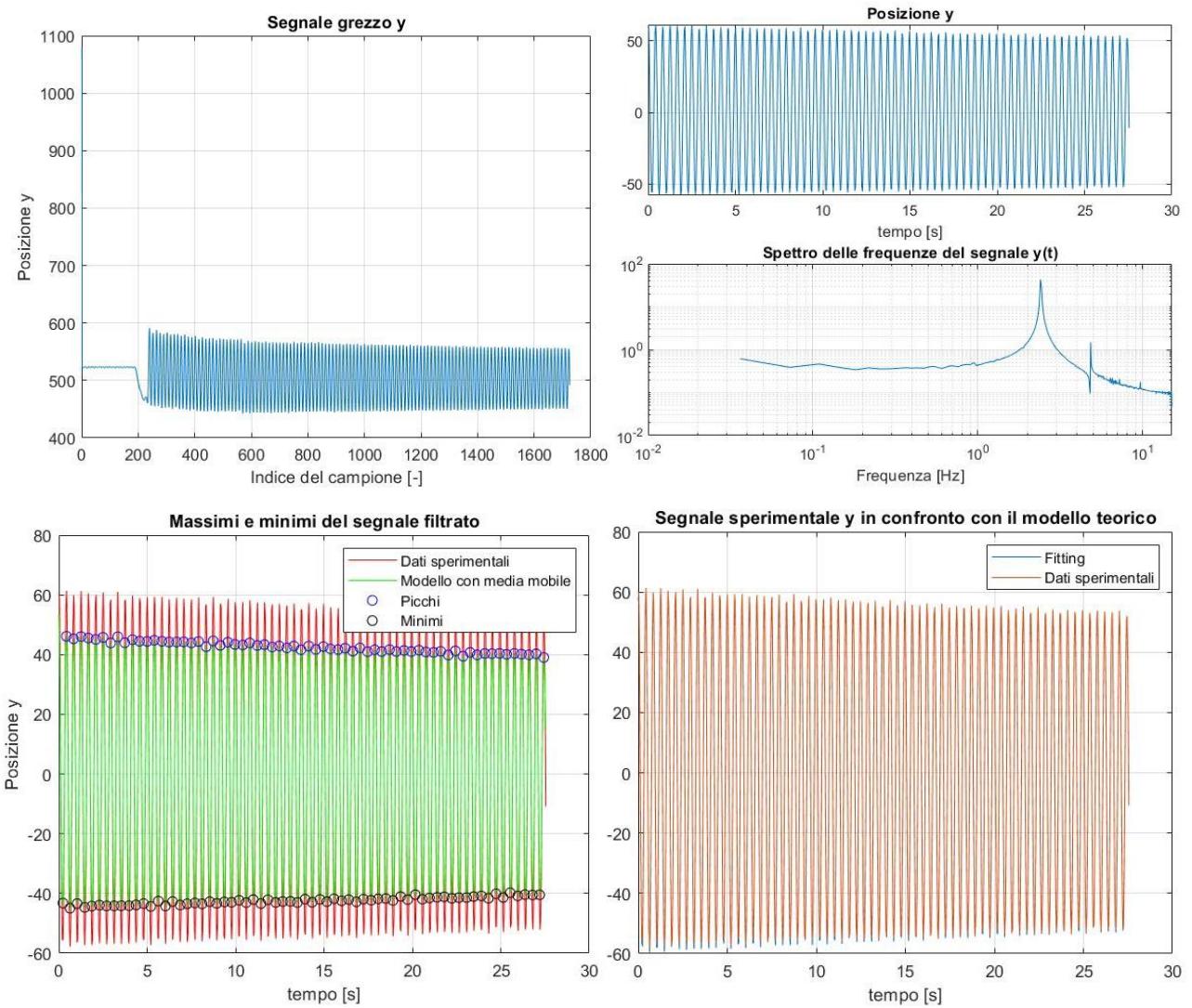


Figura 115 – Prova di acquisizione numero tre

Viene ora riportata una tabella comparativa dei risultati ottenuta tramite la funzione fminsearch usata per trovare i parametri per il modello teorico che lo fanno comportare al meglio come i dati sperimentali.

Tabella 7 – Comparazione dei risultati

Prova		m [kg]	β [-]	k [N/m]	y_0 [pixel]	ϕ [rad]	ω_n [rad/s]
1	Appross. iniziale	2,04	0,0271	484,00	83	3,98	15,4
	Risultati	2,09	0,0297	479,84	83	4,15	15,15
2	Appross. iniziale	2,04	0,0297	484,00	88	0,94	15,4
	Risultati	2,07	0,0243	468,30	111	0,72	15,05
3	Appross. iniziale	2,04	0,0218	484,00	45	4,98	15,4
	Risultati	2,11	0,022	481,83	60	4,94	15,12

Si specifica come il modello predica dei valori di ampiezza decimali che sono poi arrotondati all'unità essendo che stiamo parlando di pixel. Per migliorare i risultati del fitting sarebbe quindi auspicabile utilizzare una videocamera con una risoluzione molto maggiore della webcam usata nelle prove sopra presentate. Il modello teorico nella realtà dovrebbe lavorare in metri, è semplicemente necessaria una conversione dei dati che non è purtroppo stata operata.

Si può comunque osservare come ci sia una predizione dei parametri abbastanza fedele rispetto alle misure di primo tentativo eseguite in laboratorio. Per provare a migliorare ulteriormente il fitting del modello sarebbe necessario effettuare alcune misurazioni a una frequenza di acquisizione maggiore, e quindi avere più dati in ogni periodo. Aumentare esageratamente la durata di acquisizione introduce infatti dei possibili errori di perdita dell'obbiettivo da parte del software di acquisizione.

4.6.4 Simulazione di frequenze di acquisizione minori:

Uno degli obiettivi dell'esercitazione innovativa era quello di verificare il teorema di Nyquist. L'idea iniziale era quella di modificare la frequenza di acquisizione della webcam direttamente durante le acquisizioni, esiste infatti una function implementata negli add-ons di Matlab richiesti per il codice di acquisizioni immagini che permette facilmente di modificare il "Frame rate" della videocamera. Questo, tuttavia, è possibile solamente se l'hardware a disposizione permette di farlo, e nel nostro caso purtroppo non è così.

Per poter comunque argomentare un discorso sul teorema di Nyquist si è quindi deciso di simulare una frequenza di acquisizione minore a partire dai campioni registrati a 30 fps. Questa procedura può essere svolta aggiungendo al codice Matlab già presentato in precedenza le seguenti righe di codice:

```
%FREQUENZA SIMULATA, Si taglia artificialmente la frequenza per verificare
%il teorema di Nyquist. Se non si è interessati commentare questa sezione.
%Inserire solo sottomultipli di 30
f_sim = 2; %Hz
t=f_acq/f_sim;
y_r=y;
y=zeros(ceil(length(y_r)/t),1);
for i=t:t:length(y_r)
    y(i/t)=y_r(i);
end
f_acq = f_sim; %Solo se si taglia il segnale
.
.
.
%La divisione per t serve per correggere in caso di frequenza simulata
y = y(1500/t:2600/t);
tempo = tempo(1500/t:2600/t);
y = y -mean(y);
tempo = tempo - tempo(1);
tempo = tempo.*t;
```

["fitting ImageProcessing.m" Pt.6 – Modifiche per simulare un segnale con una frequenza di acquisizione minore](#)

Una nota importante per questo "scarto" di dati è che si può solamente simulare una frequenza che è un sottomultiplo della frequenza di acquisizione. Questo perché è impossibile prendere dei dati che non sono presenti nel segnale originale, e crearli con medie tra i segnali più vicini introdurrebbe un'arbitrarietà troppo elevata nella simulazione.

Prima di iniziare con quest'analisi si ricordano le frequenze naturale e smorzata del sistema e quindi la frequenza di Nyquist.

$$\omega_n = 15.4 \left[\frac{\text{rad}}{\text{s}} \right] = 2.45 \text{ [Hz]} \quad \omega_s = 15.24 \left[\frac{\text{rad}}{\text{s}} \right] = 2.43 \text{ [Hz]}$$

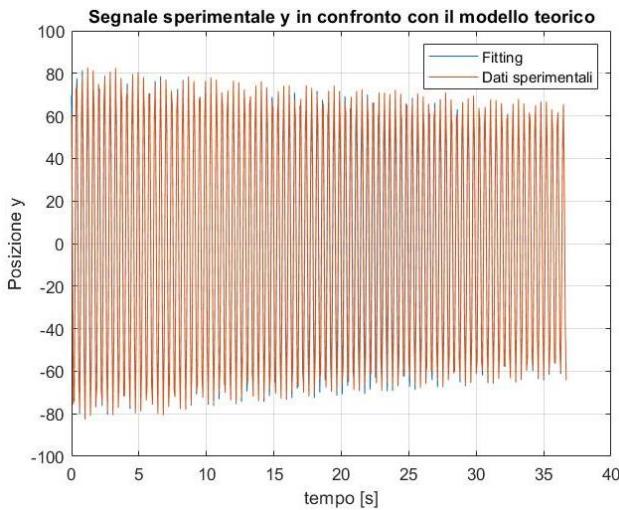
$$\omega_{Nyquist} = 2\omega_s = 30.47 \left[\frac{\text{rad}}{\text{s}} \right] = 4,85 \text{ [Hz]}$$

In Figura 55 vengono riportati i segnali simulati con frequenze di 15 Hz, 10 Hz, 5Hz, 3Hz. Osservando come il segnale vari a diverse frequenze di acquisizione si possono trarre delle interessanti conclusioni. In prima battuta osserviamo come a 15 Hz il segnale risulti ancora abbastanza simile all'originale preso a 30 Hz che è visibile in Figura 52. Si iniziano tuttavia a osservare delle creste sui picchi del segnale, questo avviene perché la frequenza di campionamento non è successivamente alta da garantire che ad ogni picco del segnale venga effettuata un'acquisizione. Questo concetto può essere osservato ancora meglio nell'acquisizione a 10 Hz, dove queste oscillazioni nei picchi sono ancora più evidenti.

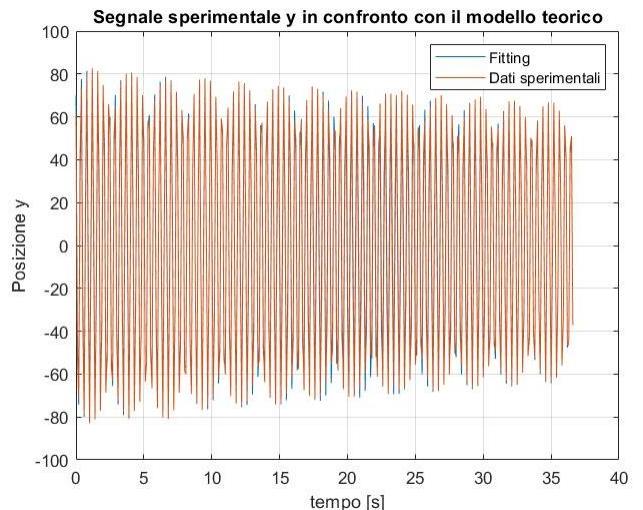
Sebbene non sia stato possibile variare agilmente la frequenza di acquisizione possiamo confermare la regola indicativa per cui 10 volte la frequenza a cui oscilla un sistema come frequenza di acquisizione dà risultati accettabili. In questo caso infatti la frequenza sarebbe 24,25 Hz. L'acquisizione a 30 Hz risulta ben riuscita mentre quella a 15 Hz presenta delle imperfezioni.

Continuando a scendere con la frequenza si arriva a 5 Hz dove il problema nel campionamento risulta evidente. Quello che sta accadendo è sostanzialmente il fenomeno fisico del battimento, la frequenza a cui oscilla il sistema è circa 2,4 Hz mentre il doppio di tale frequenza è 4,8 Hz. Con un'acquisizione a 5 Hz si prendono poco più di 2 campioni per ogni periodo di oscillazione del sistema massa-molla-smorzatore, questo causa uno spostamento relativo tra un periodo e l'altro del punto a cui sta venendo effettuata l'acquisizione, motivo per cui ci si sposta da momenti in cui si riesce a rilevare il segnale massimo in ampiezza a momenti in cui viene solamente letto lo zero.

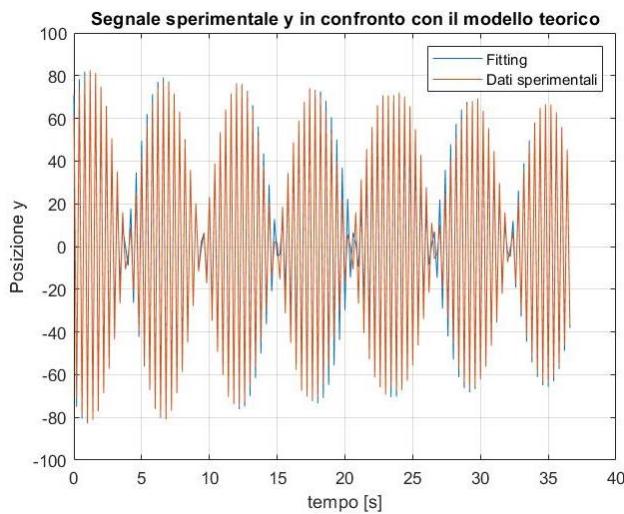
Scendendo ancora con la frequenza di acquisizione si arriva a 3 Hz. Se apparentemente sembra che il segnale sia più pulito del caso precedente questo è solamente perché oltre ad essersi persa l'informazione sulla forma del segnale si è persa anche quella in frequenza per cui il segnale sembra regolare ma non ha più la frequenza dei segnali precedenti. 3 Hz sono infatti inferiori alla frequenza di Nyquist di questo segnale che è 4,8 Hz. Possiamo quindi validare sperimentalmente le affermazioni del teorema di Nyquist.



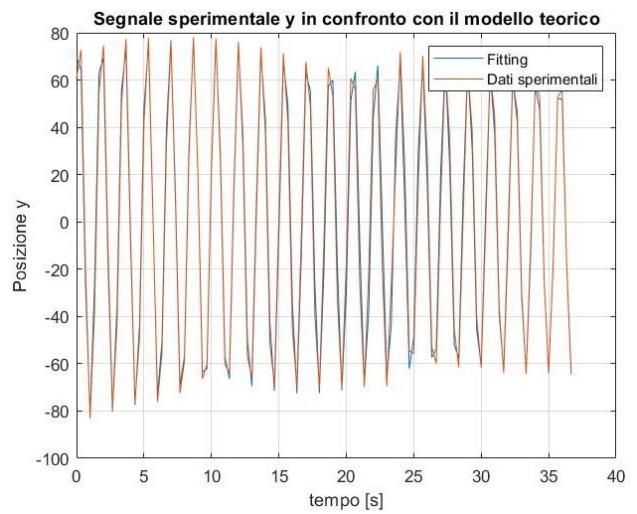
a) frequenza 15 Hz



b) frequenza 10 Hz



c) frequenza 5 Hz



d) frequenza 3 Hz

Figura 116 – Segnali di $y(t)$ con frequenze simulate di 15 Hz, 10 Hz, 5 Hz e 3 Hz

Di seguito viene proposta una tabella che riporta l'andamento della frequenza del segnale ricavata tramite l'analisi di Fourier al variare della frequenza di acquisizione:

Tabella 8 – Frequenza del segnale tramite analisi di Fourier a diverse frequenze di acquisizioni

Frequenza di acquisizione [Hz]	Frequenza del segnale [Hz]
30	2,43
15	2,42
10	2,43
5	2,42
3	0,59
2	0,405

Lo stesso risultato è riportato in Figura 56. Sebbene non correli molti dati da comunque un'idea intuitiva di quello che il teorema di Nyquist implica per questa applicazione. I segnali acquisiti a più di 4,8 Hz riescono a leggere correttamente la frequenza di oscillazione del sistema massa molla smorzatore mentre scendendo sotto questa soglia si perde l'informazione di frequenza.

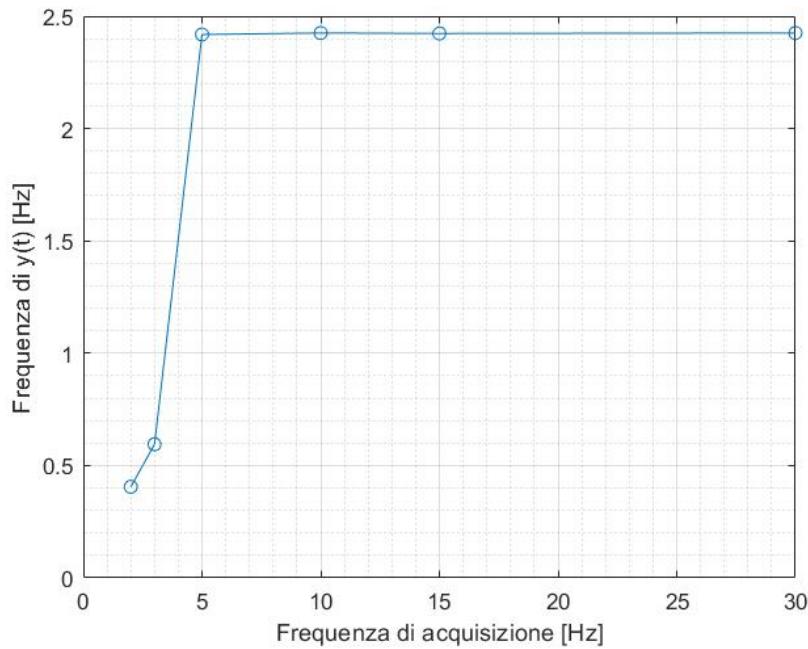


Figura 117 – Frequenza del segnale tramite analisi di Fourier a diverse frequenze di acquisizioni

4.6.5 Sviluppi futuri e possibili migliorie

In quest'ultimo capitolo vengono discussi i possibili miglioramenti che possono essere applicati al metodo di acquisizione immagini. Come primo punto c'è sicuramente quello di utilizzare una telecamera più prestante, possibilmente con la possibilità di modificare gli fps. La telecamera utilizzata nelle prove sopra commentate è infatti una normale webcam USB che lavora a 30 fps fissi, avere la possibilità di aumentare la quantità di campionamenti al secondo garantirebbe una maggiore accuratezza dei segnali acquisiti. Allo stesso tempo poter ridurre a piacere il “Frame rate” consentirebbe di eliminare tutta la porzione di codice decisamente “artificiale” di abbassamento a posteriori della frequenza di acquisizione, dando anche la possibilità di indagare meglio il teorema di Nyquist.

Un secondo punto è poi sicuramente quello di avere un Hardware per posizionare la telecamera più professionale, o comunque stabile. Questo aumenterebbe la ripetibilità delle misure eliminando alcuni degli errori di acquisizione più banali. In Figura 57 è riportata una foto che mostra il “Set-up” usato per alcune delle prove. Nonostante il risultato ottenuto non sia stato scarso si capisce come potrebbe essere facilmente migliorato.

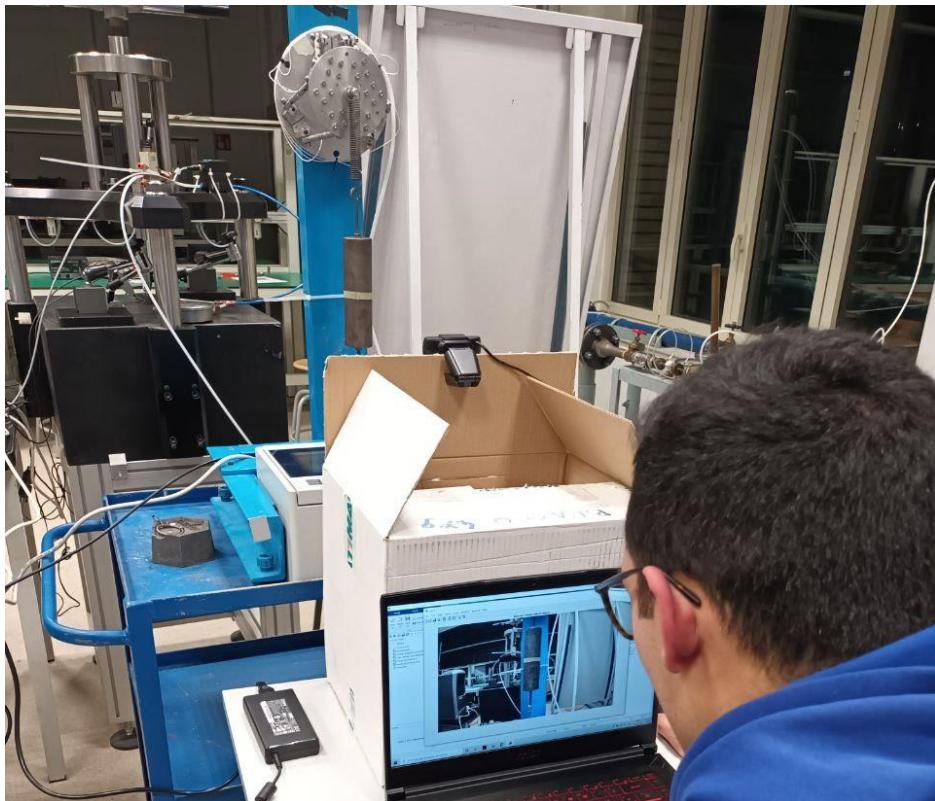


Figura 118 – Set-up per l’acquisizione immagini

Un’ulteriore punto su cui sarebbe ottimo lavorare in futuro è la conversione dei dati acquisiti dal codice di acquisizione immagini in metri. L’analisi dei segnali eseguita nelle pagine precedenti si basa infatti su una $y(t)$ che è misurata in pixel, cosa che fa perdere quindi qualsiasi legame con la realtà. Come già detto l’analisi del segnale è comunque corretta, essendo che il modello teorico da validare è a parametri concentrati è possibile non operare alcuna conversione e lavorare con i segnali acquisiti. Convertire in metri causerebbe solo uno scalamento del segnale, nella misura in cui l’oscillazione avviene in uno spazio molto ridotto, e quindi non entrano in gioco errori di angolazione, cosa che effettivamente è verificata per le prove eseguite.

Un tentativo di conversione del segnale è comunque stato provato, nello specifico l’idea era quella di leggere due posizioni su un metro messo in secondo piano rispetto alla massa e poi operare una conversione rispetto alla lettura fatte su quelle due conversioni. Il risultato non è stato ottimale per cui si è deciso di non implementarlo nella relazione, il Set-up può comunque essere osservato in Figura 58.

Sarebbe utile implementare una sorta di lettura del segnale di zero per leggere la posizione della molla prima e dopo l’aggiunta della massa. Questo consentirebbe, noto il coefficiente elastico della molla k (sia da misura iniziale che da modello) di ricavarsi l’allungamento x in metri che la molla subisce. Questo potrebbe essere usato come una primitiva forma di conversione da pixel a metri ma anche per potersi portare dal modello in posizione del sistema al modello in forza usato nella parte di analisi del segnale delle celle di carico con un’affidabilità maggiore che non solo moltiplicando i parametri trovati dal fitting.



Figura 119 – Set-up di primo tentativo per la conversione da pixel a metri