



POLITECNICO DI MILANO
Facoltà di Ingegneria Industriale
Dipartimento di Ingegneria Aeronautica

**MODELLAZIONE, SIMULAZIONE E CONTROLLO DI UN SENSORE IRST
IMBARCATO SU UN CACCIA**

Relatore:

Chiar.mo Prof. GIANANTONIO MAGNANI

Tesi di Laurea di:

EDOARDO VILLA
751268

A.A. 2010–2011

a Valentina

Ringraziamenti

La prima persona che desidero ringraziare è il professor GianAntonio Magnani. I suoi consigli ed i suoi insegnamenti sono stati fondamentali; senza il suo interessamento e la sua straordinaria disponibilità non avrei mai raggiunto questo traguardo.

Desidero inoltre ringraziare la SELEX Galileo, per l'opportunità di lavorare su un loro prodotto e di conoscere un ambiente stimolante e innovativo. In particolare sono profondamente grato all'ingegnere Giorgio Balzarotti, all'ingegnere Giuseppe Colombi e all'ingegnere Stefano Marangoni, che mi hanno aiutato in questo lavoro.

Un grazie speciale va a Marco, per essere stato un esempio e un maestro da ben prima che iniziassi l'università.

Un pensiero particolare a tutti i miei amici, il vostro affetto ha reso più pieno ogni momento di felicità e mi ha sostenuto ed aiutato nei giorni più difficili; ai compagni dell'università, grazie a voi questo percorso è stato più bello e meno faticoso. Un ultimo ringraziamento va alla mia famiglia, per tutto.

Indice

Ringraziamenti	i
Indice	iii
Sommario	vii
Introduzione	I
I. 1. Motivazioni ed obiettivi	I
I. 2. Risultati principali	2
I. 3. Organizzazione del lavoro.	3
Capitolo 1	
Inquadramento del problema	5
I. 1. Il programma Gripen NG	5
I. 2. Descrizione di un sensoreIRST	6
I. 2. 1. Storia del sensore	7
I. 2. 2. Principio di funzionamento	7
I. 2. 3. Movimentazione del sensore.	9
Capitolo 2	
Schema dell'anello di controllo e modello <i>Simulink</i>	II
2. 1. Schema generale.	II
2. 2. Descrizione delle schede	12
2. 3. Trasmissione del moto	15
2. 4. Modello <i>Simulink</i>	15
2. 4. 1. Definizione del riferimento	18
2. 4. 2. Controllore di posizione.	20
2. 4. 3. Motore ed elettronica di controllo	21
2. 4. 4. Attrito	21
2. 4. 5. Dinamica meccanica del sensore	23
2. 4. 6. Sensori di posizione e velocità.	23
2. 4. 7. Finestre grafiche.	24
Capitolo 3	
Modellazione multicorpo	29
3. 1. Sistemi di riferimento.	29
3. 1. 1. Sistema di riferimento terrestre	30

INDICE

3. 1. 2.	Sistema di riferimento del velivolo	31
3. 1. 3.	Sistema di riferimento del sensore.	32
3. 1. 4.	Sistema di riferimento del supporto	33
3. 1. 5.	Sistema di riferimento dello specchio	34
3. 2.	Assunzioni sul moto del velivolo.	34
3. 3.	Coordinate libere	35
3. 4.	Equazioni di Lagrange	36
3. 4. 1.	Energia cinetica	37
3. 4. 2.	Energia potenziale	38
3. 4. 3.	Lavori virtuali	39
3. 4. 4.	Scrittura delle equazioni.	40
3. 4. 5.	Considerazioni sulle equazioni	42
3. 5.	Disturbi sul sistema.	42
3. 6.	Implementazione al calcolatore.	43
Capitolo 4		
Modellazione del motore elettrico		47
4. 1.	Modello della dinamica elettrica.	49
4. 2.	Elettronica di controllo.	51
4. 3.	PWM.	53
4. 3. 1.	PWM indipendente bipolare	55
4. 3. 2.	PWM indipendente unipolare	56
4. 3. 3.	PWM complementare bipolare.	57
4. 3. 4.	PWM complementare unipolare	58
4. 3. 5.	Modellazione <i>Simulink</i>	59
Capitolo 5		
Prove sperimentali sul sensore IRST		63
5. 1.	Prove previste	63
5. 1. 1.	Prove a scalino	63
5. 1. 2.	Prove a velocità costante.	64
5. 2.	Esperimenti effettuati.	66
5. 2. 1.	Risposta a scalino.	66
5. 2. 2.	Prove a velocità costante, $\Delta t = 0.5$ ms	70
5. 2. 3.	Prove a velocità costante, $\Delta t = 0.25$ ms	72
5. 2. 4.	Rumore del segnale	74

Capitolo 6	
Individuazione e modellazione dei disturbi	75
6. 1. Analisi dei dati delle rampe di velocità	75
6. 2. Analisi in frequenza	83
6. 2. 1. Basse frequenze	83
6. 2. 2. Alte frequenze	92
6. 3. Modellazione dei disturbi	97
6. 3. 1. Disturbi di frequenza lineare con la velocità	97
6. 3. 2. Disturbo di frequenza decrescente con la velocità	101
Capitolo 7	
Controllore di posizione	103
7. 1. Regolatore P/PI	103
7. 1. 1. PID	103
7. 1. 2. Controllo in cascata	104
7. 1. 3. Realizzazione in <i>Simulink</i>	104
7. 2. Taratura dei regolatori	107
7. 2. 1. Taratura dell'anello interno	108
7. 2. 2. Taratura dell'anello esterno	109
7. 3. Confronto con i dati sperimentali e verifica delle prestazioni	110
7. 3. 1. Confronto con i dati sperimentali	110
7. 3. 2. Verifica delle prestazioni	112
7. 4. Osservatore del disturbo di coppia	114
7. 4. 1. Descrizione generale	114
7. 4. 2. Modello <i>Simulink</i>	117
7. 4. 3. Verifica finale delle prestazioni	119
Capitolo 8	
Interfaccia grafica di controllo	121
8. 1. Descrizione generale	121
8. 2. Menu principale	122
8. 3. Menu simulazione	123
8. 4. Menu grafici	125
8. 5. Menu FFT	128
Considerazioni finali e sviluppi futuri	133
Indice delle tabelle	135

Indice delle figure	137
Appendice A	
Lista degli acronimi	149
Appendice B	
Listato delle funzioni e degli <i>script</i> MATLAB	151
B. 1. animIRST.m	151
B. 2. bracci.m	155
B. 3. dati_antenna.m	157
B. 4. dist_AZ.m	160
B. 5. fftIRST.m	161
B. 6. figureIRST.m	165
B. 7. gen_SP.m	168
B. 8. graficiIRST.m	168
B. 9. inertiadist.m	171
B. 10. IRSTdin.m	172
B. 11. MAIN.m	174
B. 12. PWM.m	178
Bibliografia	181
Sitografia	183

Sommario

Lo Skyward-G è un sensoreIRST sviluppato dalla SELEX Galileo. Poiché è stato scelto per il programma Gripen NG, la sua parte di controllo è nuovamente in fase di progettazione.

Questa tesi riguarda la costruzione di un modello di simulazione del sensore che possa essere un aiuto per la definizione della strategia di controllo e la sua messa a punto e permetta inoltre di provare nuove configurazioni

I *software* usati sono *Simulink* e *MATLAB*.

Il modello è particolarmente dettagliato nella modellizzazione del motore, della sua elettronica di potenze e della *pulse width modulation*.

È stata inoltre sviluppata una interfaccia grafica che consenta di semplificare il controllo dei parametri di simulazione e faciliti la visualizzazione dei risultati.

Parole chiave: sensoreIRST, modello di simulazione, *pulse width modulation*.

Abstract

The Skyward-G is anIRST sensor developed by SELEX Galileo. As it was chosen for the Gripen NG program, its control part is under design phase again.

This graduation thesis focuses on the design of a sensor simulation model that could be an aid in control strategy design and calibration. It can also allow testing new configurations.

The software used are *Simulink* and *MATLAB*.

The model is particularly well detailed in the motor's modelling, as well as its power control and pulse width modulation.

A graphical interface has also been developed in order to simplify the simulation parameters control and visualize its results more easily.

Keywords:IRST sensor, simulation model, pulse width modulation.

Introduzione

I.1. Motivazioni ed obiettivi

Questo lavoro di tesi riguarda la simulazione ed il controllo di un sensore IRST, *Infra Red Search and Track*, sistema di ricerca e inseguimento ad infrarossi.

Il lavoro è stato svolto in stretta collaborazione con la società SELEX Galileo. L'analisi è stata svolta infatti sul sensore SKYWARD-G, su cui è anche stata inoltre effettuata una campagna di prove sperimentali volte a identificarne alcune caratteristiche.

Lo scopo principale del lavoro è stato quello di ottenere un modello sufficientemente raffinato del sensore, che comprendesse la parte di controllo ed una corretta modellazione del motore.

Tale lavoro di simulazione è stato eseguito principalmente tramite il *software MATLAB* e *Simulink*. Il *software Simulink* ha permesso di creare un modello del sistema dinamico da simulare, descritto attraverso uno schema a blocchi del sistema stesso.

Simulink è strettamente interconnesso con *MATLAB*, può infatti leggere e scrivere nello *workspace* di *MATLAB* ed inoltre possono essere inseriti dei blocchi all'interno dello schema che eseguono delle funzioni in codice *MATLAB*, eventualmente prendendo da *Simulink* dei dati in ingresso e potendone restituire in uscita.

Grazie a questi *software* sono state scritte le equazioni della meccanica multi corpo del sensore e delle componenti elettriche, oltre ad essere simulate le leggi di controllo.

Alla scrittura del modello ha fatto seguito una serie di prove sperimentali su un sensore opportunamente strumentato che ha permesso di identificare dei parametri importanti, in particolar modo riguardo ai disturbi agenti sul sistema.

L'identificazione dei disturbi ha permesso quindi di raffinare ulteriormente il modello, con la simulazione dei disturbi stessi.

Si è quindi studiata e calibrata una strategia di controllo del sensore, in particolare si è implementato un controllo in cascata P/PI sulla posizione e sulla velocità, con uno stimatore del disturbo di coppia.

1.2. Risultati principali

I risultati principali di questa tesi sono stati:

- la costruzione di un modello *Simulink* del sensore;
- l'individuazione e la caratterizzazione dei disturbi di coppia agenti sui motori del sensore;
- lo sviluppo di un'interfaccia grafica per il controllo delle simulazioni.

Il primo risultato è sicuramente quello principale e che ha richiesto l'impegno più grande, anche in termini di tempo.

In particolare si è cercato di creare un modello che potesse riprodurre nel maggior dettaglio possibile il funzionamento del sensore e dei suoi componenti, compatibilmente anche con dei tempi di simulazione accettabili. Si è inoltre dato importanza al fatto che il modello potesse essere facilmente modificabile cambiando opportuni parametri o funzioni, al fine di poter adattarsi ad eventuali aggiornamenti del sensore. Questo aspetto risulta di importanza notevole se si considera che la piattaforma è in fase di aggiornamento da parte dell'azienda.

Un elemento particolare del modello è costituito dalla dettagliata simulazione del funzionamento dei due motori elettrici bifase, che è stata svolta considerando la dinamica elettrica dei due avvolgimenti.

Questo ha inoltre permesso di dettagliare anche la parte di elettronica di controllo del motore stesso, in cui è implementata la logica di controllo della corrente in quadratura e diretta, tramite opportune trasformazioni delle correnti negli avvolgimenti.

Un altro risultato che è stato consentito dalla simulazione dei singoli avvolgimenti del motore è stato quello di poter simulare nel dettaglio la modulazione PWM.

Al fine di permettere un controllo maggiore sulla logica di funzionamento del PWM, la simulazione ha incluso la modellazione del circuito elettrico del ponte ad H di ogni avvolgimento.

Un altro elemento particolare del sensore, e quindi anche del modello, è la mancanza di una trasmissione tra il motore ed il carico. Si tratta infatti di motori *direct drive*, collegati direttamente al carico, senza alcun tipo di trasmissione o riduttore, questo ha avuto delle ovvie implicazioni nella costruzione del modello.

L'individuazione dei disturbi di coppia e la loro caratterizzazione è servita a dettagliare ulteriormente il modello. Tale lavoro ha richiesto l'esecuzione di alcune prove sperimentali e l'analisi dettagliata di un numero consistente di dati.

La costruzione di un'interfaccia grafica è stata resa necessaria dal notevole numero di parametri di cui è dotato il modello e dalla necessità di poterli modificare in modo agevole. Essa ha inoltre permesso di rendere più immediate alcune operazioni di visualizzazione dei risultati e di elaborazione degli stessi.

I. 3. Organizzazione del lavoro

La tesi è strutturata in otto capitoli, oltre all'introduzione e le conclusioni, e due appendici, il cui argomento è il seguente.

- Capitolo 1:* Nel primo capitolo viene descritto il velivolo su cui sarà montato il sensore, il Gripen NG. Viene inoltre descritto un generico sensore IRST, spiegandone brevemente la storia ed il principio di funzionamento.
- Capitolo 2:* Viene inizialmente descritto nel dettaglio il sensore Skyward-G, si descrive quindi il modello *Simulink* dello stesso, con una breve descrizione dei blocchi che lo compongono.
- Capitolo 3:* Si ricavano le equazioni di moto del sensore e vengono descritti i blocchi del modello che le implementano.
- Capitolo 4:* Viene effettuata la descrizione del motore elettrico e dell'elettronica di controllo, descrivendo nel dettaglio i blocchi del modello che ne simulano la dinamica.
- Capitolo 5:* Vengono descritti gli esperimenti che sono stati effettuati sul sensore, presentandone anche l'andamento nel tempo.
- Capitolo 6:* Sono analizzati i dati ricavati dagli esperimenti descritti nel capitolo precedente, vengono caratterizzati i disturbi di coppia agenti sul sistema e sono quindi ricavati i blocchi che simulano nel modello tali disturbi.
- Capitolo 7:* Sono descritti i blocchi dei controllori di posizione, viene spiegata la procedura di taratura dei regolatori ed è introdotto nel modello uno stimatore del disturbo di coppia, di cui viene descritto il funzionamento.
- Capitolo 8:* Viene spiegato il funzionamento e le capacità dell'interfaccia grafica di controllo delle simulazioni.
- Appendice A:* Vengono elencati gli acronimi utilizzati nella trattazione e ne viene fornita una breve descrizione.
- Appendice B:* Sono riportati i listati dei programmi che sono stati utilizzati per la simulazione del sensore.

Capitolo 1

Inquadramento del problema

Verrà ora brevemente illustrato il velivolo su cui è previsto che venga montato il sensore, il Gripen NG. In seguito verrà anche fatta una concisa descrizione del funzionamento di un sensoreIRST ed una schematizzazione dello SKYWARD-G.

1.1. Il programma Gripen NG

Il Saab JAS 39 Gripen (*Italiano: Grifone*) è un caccia monomotore multiruolo svedese.

Gli studi di progetto del velivolo iniziarono all'inizio degli anni 80, con il primo volo che avvenne il 9 dicembre del 1988, e l'aeroplano entrò in servizio alla fine del 1997.



Figura 1.1. Fotografia dall'alto di un Saab JAS 39 Gripen.

Nel 2007 fu deciso di avviare un programma di sviluppo di una versione migliorata del caccia, dotata anche di un nuovo propulsore di maggiore potenza, che sfruttasse tutte le nuove tecnologie disponibili.

Questa nuova versione, denominata NG, *Next Generation*, ebbe il suo volo inaugurale il 27 Maggio del 2008.

Tra i cambiamenti di cui beneficia il Gripen NG c'è anche l'introduzione di un sensore IRST integrato nel velivolo, la cui fornitura è stata assegnata alla SELEX Galileo. Nelle versioni precedenti del velivolo la capacità di visione infrarossa era infatti eventualmente garantita da opportuni *pod* FLIR posizionati sotto il ventre del velivolo.



Figura 1.2. *Rendering* del Gripen NG, si noti come sia presente la cupola del sensore IRST sul lato sinistro del muso, assente nelle versioni meno recenti del caccia.

1.2. Descrizione di un sensore IRST

Il termine IRST è una sigla che indica *Infra-Red Search and Track*, traducibile come Ricerca e Inseguimento tramite InfraRossi. Un sensore IRST permette infatti l'individuazione e l'inseguimento di obiettivi che emettono radiazioni infrarosse e quindi, come verrà descritto in seguito, che emettono calore.

Differentemente dal Radar, con cui condivide la finalità di ricerca di un bersaglio, il sensore IRST è totalmente passivo, dovendo solo rilevare la radiazione emessa dal bersaglio e non avendo quindi da generare a sua volta delle radiazioni elettromagnetiche. Questo rende l'illuminazione del bersaglio da parte del sensore non identificabile da parte del bersaglio stesso e permette quindi di usare il sensore anche in situazioni in cui si vuole mantenere la maggiore invisibilità al nemico possibile.

1.2.1. Storia del sensore

Il primo uso dei sensoriIRST avvenne nella metà degli anni cinquanta, sui caccia statunitensi F-101 ed F-102. Tali velivoli imbarcavano sensori notevolmente semplici, che potevano ruotare solo nel piano orizzontale. I sensoriIRST infatti, piuttosto che essere dei sistemi autonomi, venivano usati allo scopo di rendere più agevole il puntamento del Radar verso il bersaglio.

Quando la procedura di aggancio di un bersaglio da parte del Radar fu automatizzata, con l'introduzione di Radar di maggiore autonomia, i sensoriIRST persero quindi, la loro funzione e furono temporaneamente abbandonati dalla dotazione avionica dei caccia.

I sensoriIRST vennero riconsiderati attorno agli anni ottanta, grazie a due importanti cambiamenti nel sensore:

- una maggiore sensibilità, che permetteva una migliore risoluzione ed un migliore *range*;
- la possibilità di essere orientati ruotando sia orizzontalmente che verticalmente.

Furono inizialmente introdotti sul caccia statunitense imbarcato F-14 ed in seguito anche su macchine russe, come il Su-27, ed europee, come l' Eurofighter Typhoon.



Figura 1.3. SensoreIRST del Su-27 e dell' Eurofighter Typhoon.

Il loro utilizzo divenne indipendente e sostitutivo rispetto al Radar, in particolare per identificare e tracciare i bersagli in situazioni in cui l'utilizzo del Radar non era possibile, considerando inoltre che la nuova tecnologia di missili, *fire and forget*, permetteva di colpire un bersaglio senza la necessità di accendere il Radar da parte dell'aereo aggressore.

1.2.2. Principio di funzionamento

Il principale elemento di trasduzione di un sensoreIRST è una telecamera sensibile allo spettro della radiazione elettromagnetica noto come infrarosso ad onde medie.

Essa funziona come un sensore termico, rilevando il calore emesso dai corpi, questo perché ogni corpo, a temperature superiori rispetto allo zero assoluto, emette una radiazione elettromagnetica chiamata radiazione termica, nello spettro dell'infrarosso.

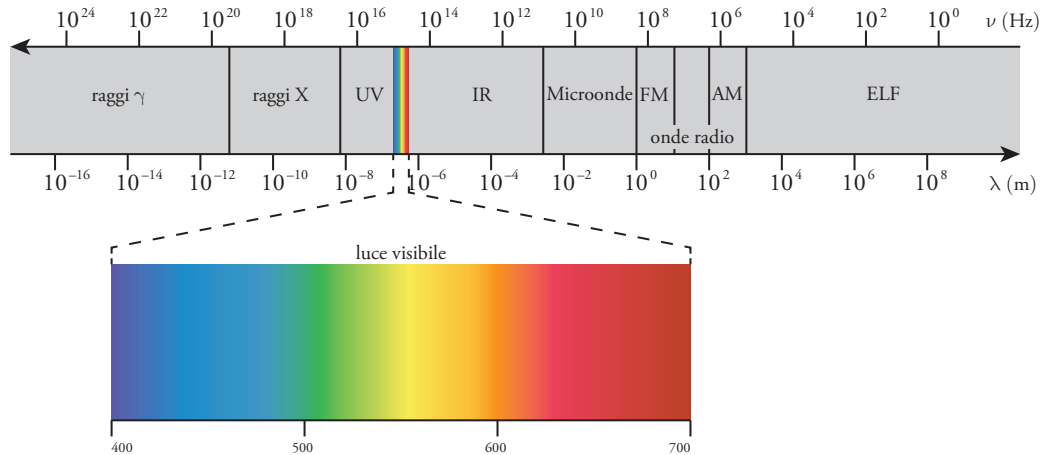


Figura 1.4. Suddivisione delle diverse categorie di radiazioni elettromagnetiche in base alla loro lunghezza d'onda.

Tale radiazione è legata al movimento degli atomi del corpo e la frequenza dominante dello suo spettro di emissione aumenta all'aumentare della temperatura come descritto dalla legge di Wien.

Questa relazione è dedotta sperimentalmente ed esprime il legame fra la radiazione emessa da un corpo nero e da una massa generica, la temperatura e la lunghezza d'onda massima.

$$[1.1] \quad T \cdot \lambda_{\max} = b;$$

dove:

- $b = 2.8977685 \times 10^{-3} \text{ K} \cdot \text{m}$ è detta costante dello spostamento di Wien;
- T è la temperatura assoluta, in kelvin, della corpo nero;
- λ_{\max} è lunghezza d'onda espressa in metri per la quale è massima la radiazione emessa dal corpo.

Nell'ambito aeronautico, la principale fonte di calore, che permette un funzionamento efficiente del sensore è costituita dai gas di scarico dei motori.

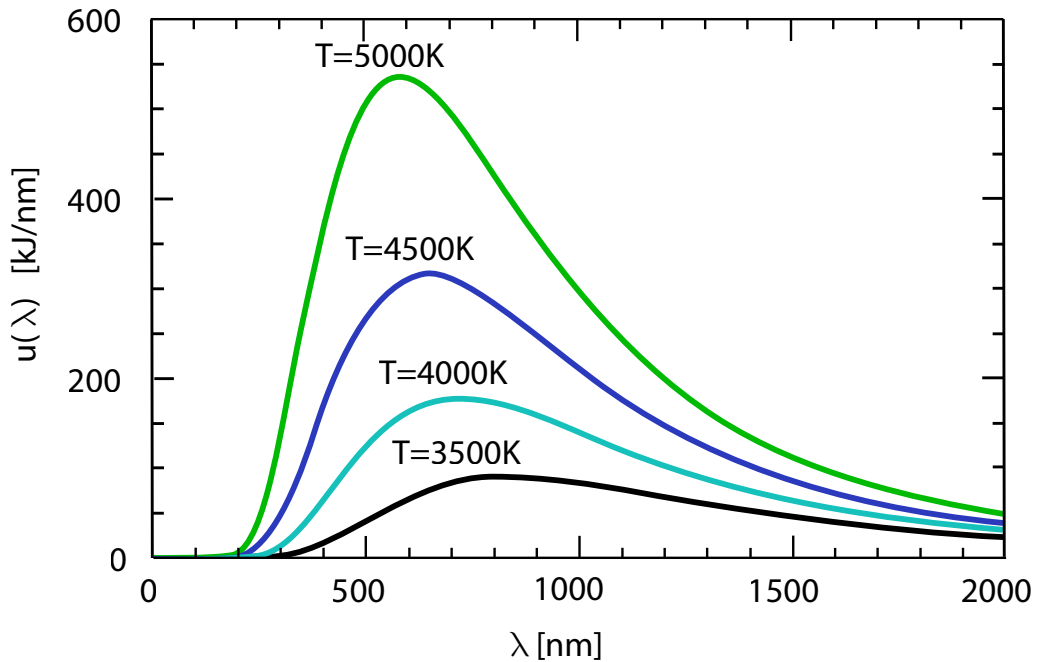


Figura 1.5. Spettri di emissione di corpi a differenti temperature, si noti come, coerentemente a quanto formulato dalla legge di Wien, all'aumentare della temperatura il picco dello spettro si sposta verso frequenze più basse.

1.2.3. Movimentazione del sensore

A causa di un campo visuale limitato il sensore deve avere la possibilità di essere orientato.

La strategia utilizzata per ottenere ciò è di due tipi:

- movimentazione dell'intero sensore;
- orientamento di uno specchio che rifletta l'immagine su una telecamera fissa.

La soluzione maggiormente diffusa è la prima, essendo più semplice ed economica, tuttavia nell'ambito aeronautico la seconda soluzione è più idonea poiché si riescono ad ottenere dei volumi più contenuti, ottenendo dei positivi impatti sulla aerodinamica del velivolo. Si ha inoltre un maggiore contenimento del peso del sistema.

Tale soluzione è anche quella che è stata utilizzata dalla SELEX Galileo per il sensore SKYWARD-G.

In particolare lo specchio ha la possibilità di ruotare attorno ad un asse perpendicolare alla base del sensore, detto di *azimuth*, e attorno ad uno perpendicolare al primo, giacente nel piano dello specchio, detto di *elevation*, come visibile in Figura 1.6 a pagina 10.

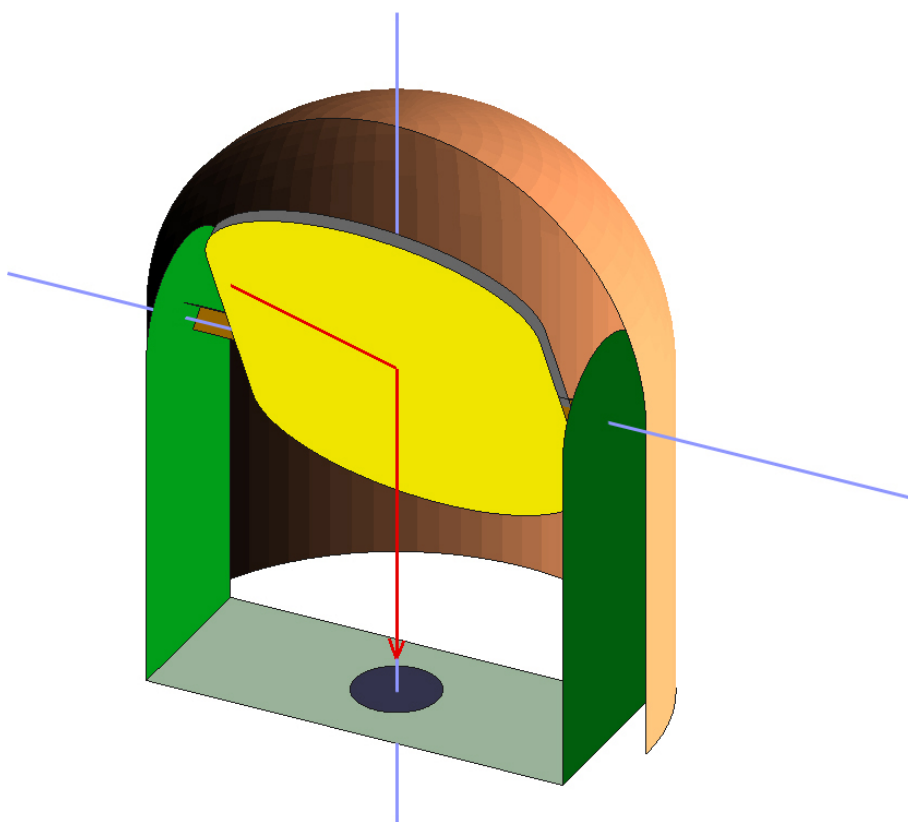


Figura 1.6. Schematizzazione del sensore, in azzurro sono visibili i due assi di rotazione del sensore, mentre in rosso è visibile la traiettoria del raggio luminoso.

Capitolo 2

Schema dell'anello di controllo e modello *Simulink*

In questo capitolo viene descritto lo schema concettuale generale del sensore, per quanto riguarda lo schema di controllo, e verrà anche mostrata la sua implementazione in *Simulink*.

In seguito saranno anche descritte alcune porzioni del modello che, per la loro semplicità, non richiedono di essere approfondite in un capitolo dedicato.

2.1. Schema generale

Lo schema generale di funzionamento del servomeccanismo può essere spiegato nel modo seguente.

Il parametro controllato dal sistema è la LOS del sensore IR. L'elettronica di controllo (*LOS control and Stabilization*) muove, mediante dei motori (*Actuators*), uno specchio che riflette lo scenario sulla pupilla del sensore IR.

La posizione dello specchio viene misurata mediante sensori angolari (*Resolvers*) e confrontata con la posizione comandata (*LOS Command*).

La misura del moto del velivolo (ottenuta via bus dal sistema inerziale del velivolo stesso) e le vibrazioni locali (misurate mediante giroscopi) sono opportunamente miscelate (dal *Navigation Processor*) per creare un segnale di compensazione della LOS.

Questo schema, mostrato nelle figura seguente, ingloba, come visto, anche una parte piuttosto complessa di generazione del riferimento.

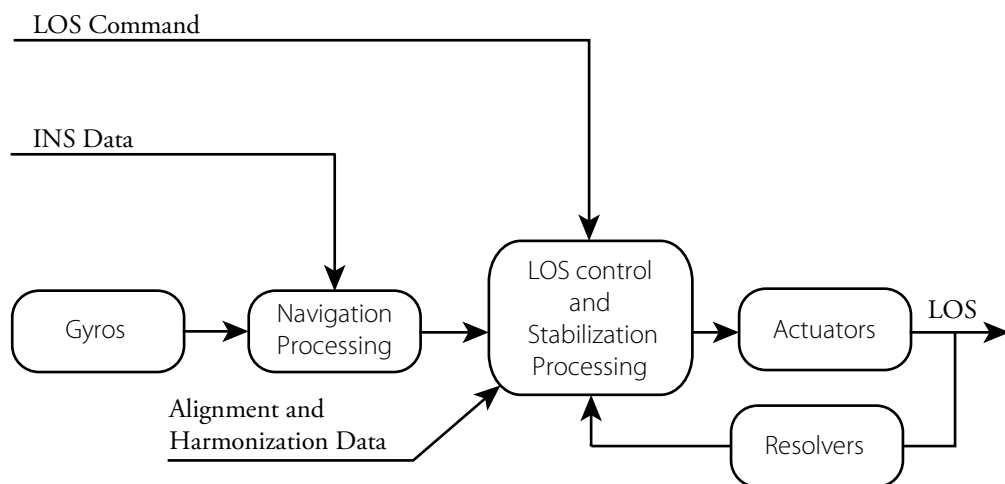


Figura 2.1. Schema generale del sistema di controllo del sensore, comprendente anche la parte relativa ai dati di navigazione.

Tale parte, oltre ad essere ancora in sviluppo da parte di SELEX Galileo, non fornirebbe particolari vantaggi nell'essere inserita nel modello e non sarà quindi descritta.

Quello su cui si sono focalizzati il lavoro e l'attenzione è l'anello di controllo della posizione del motore, avente in ingresso degli opportuni *set-point* di posizione.

Lo schema di tale parte del sensore è rappresentato nella figura seguente, in cui sono evidenziate anche le schede che svolgono le funzioni descritte, che verranno elencate nella sezione successiva.

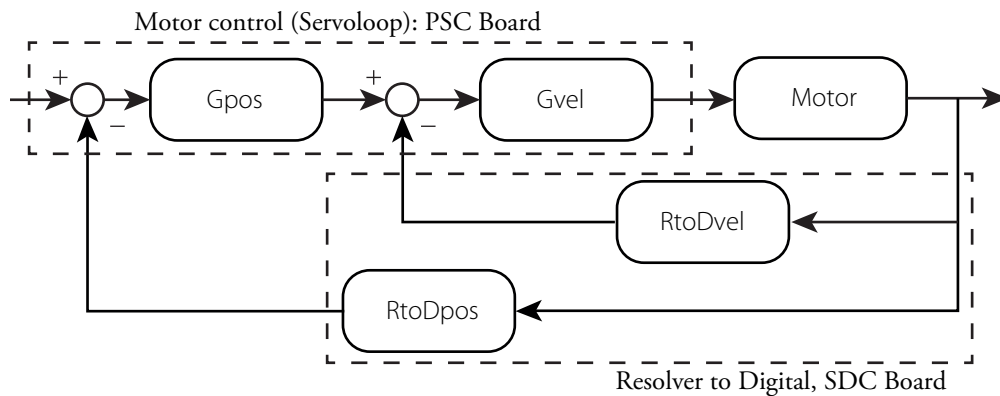


Figura 2.2. Schematizzazione funzionale dell'anello di controllo di posizione del motore, con in evidenza le schede su cui sono implementate le funzioni

Come già visto nel paragrafo 1. 2. 3, il sensore è dotato di due gradi di libertà, uno di *azimuth* ed uno di *elevation*. Lo schema presentato in Figura 2.2 è quindi identico e ripetuto per i due gradi di libertà, ovviamente considerando dei segnali di riferimento differenti.

2.2. Descrizione delle schede

Analizzeremo adesso nel dettaglio lo schema dello SKYWARD-G, per quanto riguarda la disposizione interna delle schede. Viene ora presentato, in Figura 2.3 a pagina 13, lo schema generale dell'architettura, con in evidenza le differenti schede.

Nello schema sono presenti i seguenti circuiti:

- SDC: Circuito per l'interfaccia tra le componenti analogiche e quelle digitali del meccanismo, converte i dati da analogici a digitali e viceversa, da digitali ad analogici;
- PSC: Circuito per il *motion control* dello specchio, implementa gli anelli di controllo della LOS del sensore;
- PSM: Circuito per la gestione dell'alimentazione e l'implementazione ed il controllo del PWM.

Vengono ora riportati nel particolare i diagrammi a blocchi dei singoli circuiti.

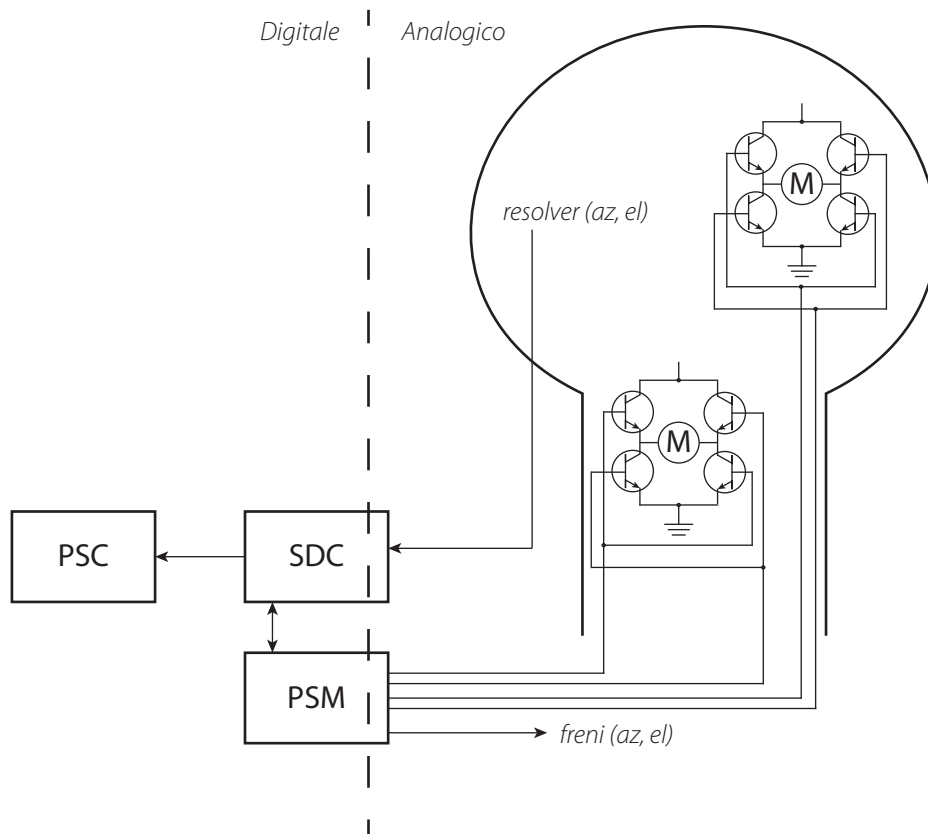


Figura 2.3. Schema generale delle schede del servomeccanismo, con in evidenza la distinzione tra segnali analogici e segnali digitali.

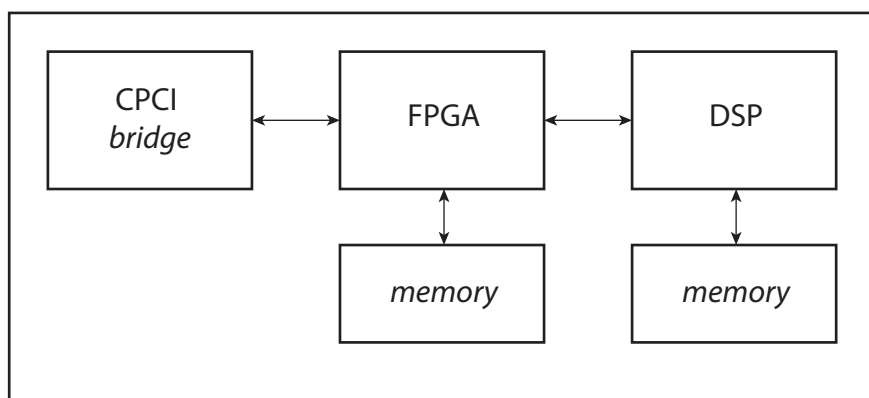


Figura 2.4. Schema a blocchi della scheda PSC, con in evidenza i sottoblocchi.

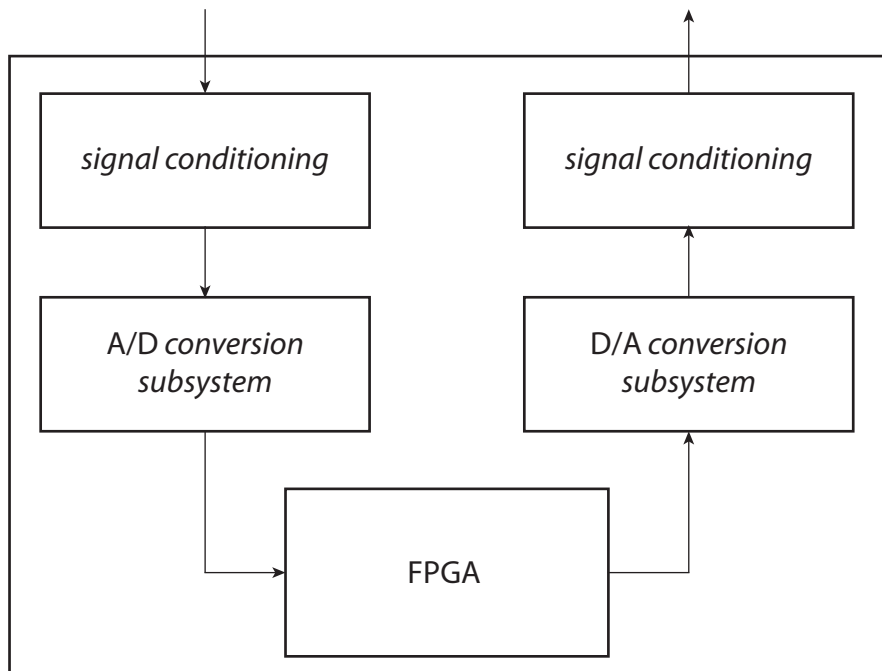


Figura 2.5. Schema a blocchi della scheda SDC, con in evidenza i sottoblocchi.

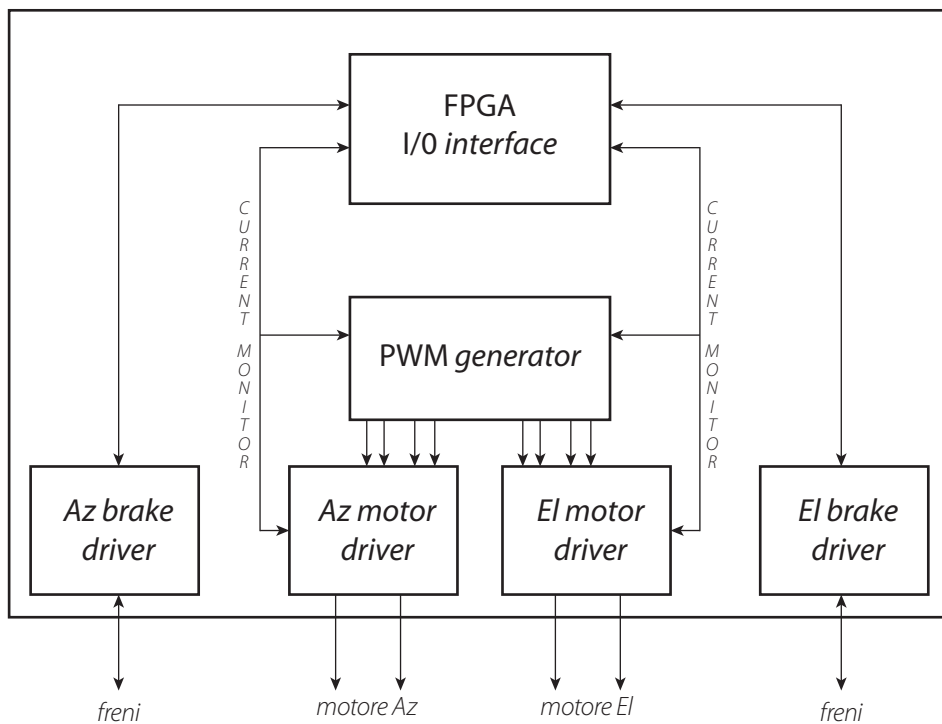


Figura 2.6. Schema a blocchi della scheda PSM, con in evidenza i sottoblocchi.

2.3. Trasmissione del moto

È di fondamentale importanza considerare da subito il particolare modo con cui il carico è connesso al motore.

Differentemente da soluzioni più tradizionali in cui è presente un riduttore, i motori di entrambi i gradi di libertà sono di tipo *direct drive*.

Questo significa quindi che il motore sia direttamente collegato allo specchio, nel caso del grado di libertà di *elevation* e al supporto, per il grado di libertà di *azimuth*.

Tale fatto dovrà essere tenuto in considerazione nella costruzione del modello, in particolare la coppia prodotta dal motore entrerà, a meno degli attriti, come *input* direttamente nelle equazioni di moto.

2.4. Modello *Simulink*

Lo schema *Simulink* del modello ricalca ovviamente le configurazioni già mostrate del sensore. Lo schema generale è presentato in Figura 2.3 a pagina 13. Esso può essere diviso in due parti, praticamente identiche, corrispondenti all'anello di controllo di *azimuth* ed *elevation*. I due anelli sono ovviamente accoppiati a livello di dinamica meccanica del sensore.

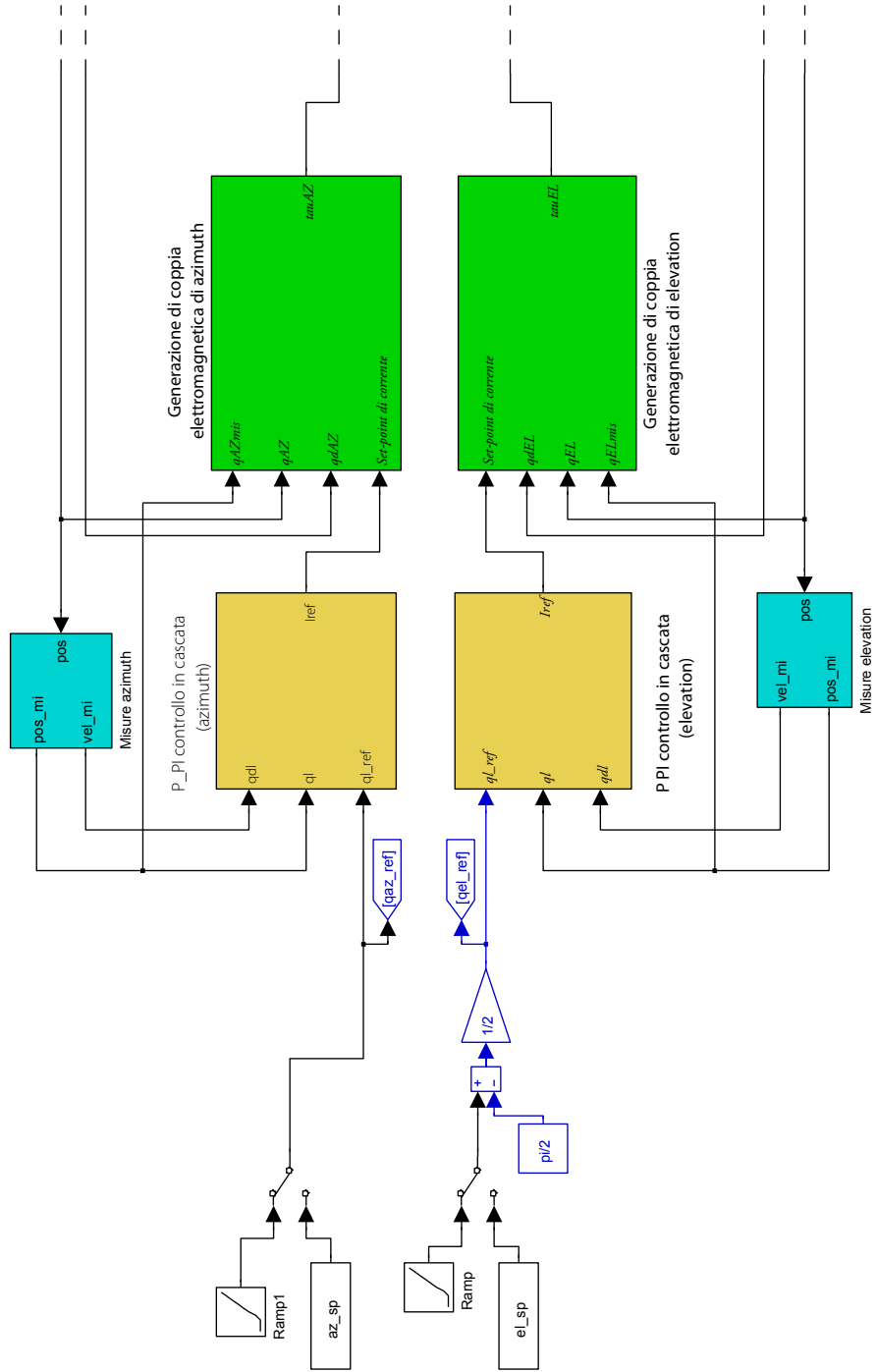
Lo schema è quindi costituito da:

- Regolatore di corrente;
- Meccanismo di *azimuth/elevation*;
- Dinamica dello specchio;
- Grafica;
- Blocchi di misura;
- Segnali di riferimento.

Nell'implementazione in *Simulink* si è preferito tenere in un blocco separato l'attrito, che concettualmente potrebbe essere legato al motore o alla dinamica dello specchio, poiché esso è notevolmente complesso e sfrutta anche un blocco già prodotto, validato e messo a punto dal professor GianAntonio Magnani e dal Dipartimento di Elettronica del Politecnico di Milano.

Si nota inoltre come nel blocco di attrito ed in quello relativo al motore ci siano in ingresso la posizione e la velocità non misurate. Questo perché i blocchi necessitano di tali grandezze per modellizzare fenomeni fisici che dipendono intrinsecamente da esse e non dalla loro misura che è invece l'ingresso dei regolatori.

Simulazione Sensore IRST



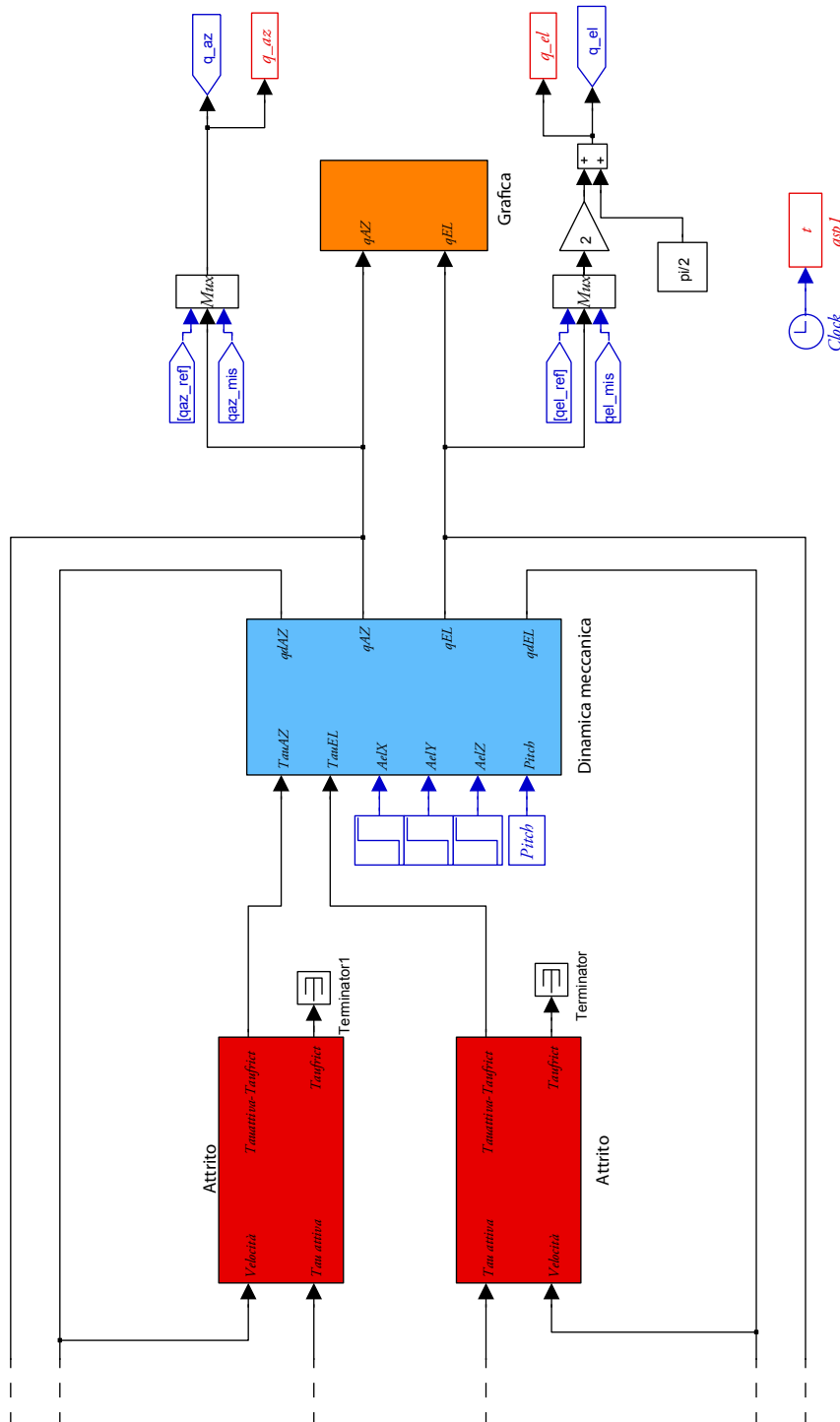


Figura 2.7. Schema del modello *Simulink* del sensore.

Andremo ora ad analizzare i singoli blocchi componenti il modello.

2.4.1. Definizione del riferimento

Per la definizione del *set-point* di posizione con cui comandare il sensore si è utilizzato una duplice strategia. Si è infatti previsto di poter sia leggere un file che di seguire un segnale generato da *Simulink*.

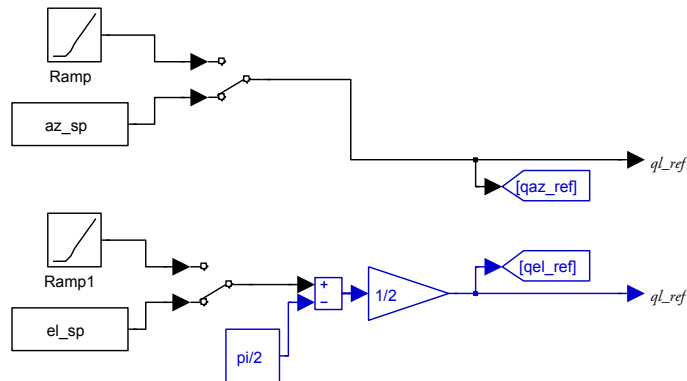


Figura 2.8. Blocchi di generazione del riferimento di posizione, lo *switch* è impostato per leggere i *file*, su entrambi i gradi di libertà.

La lettura dei *file* rende infatti possibile utilizzare gli stessi *set-point* che sono utilizzati per comandare il sensore, mentre il segnale generato tramite *Simulink* permette di simulare velocemente determinate condizioni come rampe o scalini senza dover generare il *file* apposito.

La commutazione su quale ingresso scegliere è fatta semplicemente con un blocco *Manual Switch* di *Simulink*.

Il file di *set-point* viene processato da uno script *MATLAB*, *genSP.m*, vedere Appendice B a pagina 151, prima dell'avvio della simulazione. Vengono creati due vettori, memorizzati nello *Workspace*, una per ogni grado di libertà, che contengono su ogni riga l'istante di tempo e la posizione comandata, in radianti. Tali vettori sono quindi letti in fase di simulazione da due opportuni blocchi di *Simulink*.

Per quanto riguarda il *set-point* di *elevation* devono inoltre essere tenute presenti alcune considerazioni aggiuntive, in particolare il fenomeno della leva ottica e la differente convenzione usata per gli angoli.

Con il termine leva ottica si indica il fatto che lo specchio fa compiere al raggio di luce una rotazione pari al doppio della rotazione dello specchio stesso. Si deve inoltre tenere in considerazione che nei *file* del riferimento l'*elevation* viene considerato nullo quando la linea di vista è orizzontale e lo specchio si presenta quindi con una rotazione che, secondo le convenzioni evidenziate nel capitolo 3, in Figura 3.7 a pagina 36, è pari a -45° .

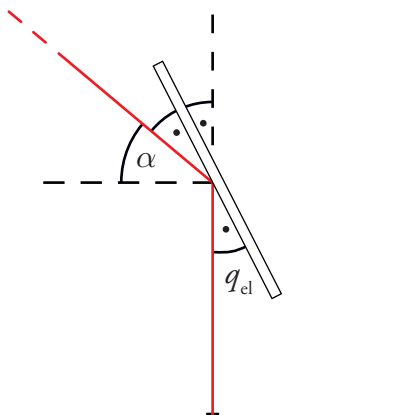


Figura 2.9. Rappresentazione del fenomeno della leva ottica e visualizzazione degli angoli.

Si deve inoltre considerare che per le leggi dell'ottica nella riflessione speculare gli angoli che il raggio incidente e quella riflesso formano con il piano di riflessione sono tra loro coincidenti, si veda la Figura 2.9.

Questo, insieme ad alcune semplici considerazioni geometriche, porta a scrivere la seguente equazione che lega l'angolo di riferimento e l'angolo di *elevation* secondo le convenzioni adottate:

$$[2.2] \quad q_{el} = \frac{\alpha - \frac{\pi}{2}}{2}.$$

Questo ha portato all'implementazione dei blocchi di condizionamento del segnale.

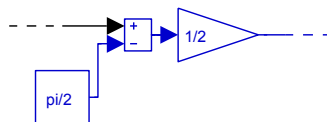


Figura 2.10. Blocchi di condizionamento del segnale di *elevation* che cambiano il sistema di riferimento con cui sono descritti gli angoli.

Si è preferito aggiungere tali blocchi nel modello e non nello *script MATLAB* poiché anche il riferimento fornito da *Simulink* fosse relativo alla linea di vista e non alla posizione dello specchio e fosse inoltre più evidente, anche nel modello, che il *set-point* contenuto nel *file* è relativo alla linea di vista ed ha una convenzione diversa da quella adottata nei blocchi relativi alla dinamica.

Per ragioni analoghe, poiché la posizione che interessa visualizzare nei grafici è quella della LOS del sensore piuttosto che quella dello specchio, anche alle grandezze

in uscita dal sistema, relative alla posizione di *elevation* è applicata la stessa trasformazione, seppur invertita.

$$[2.3] \quad \alpha = 2q_{el} + \frac{\pi}{2}.$$

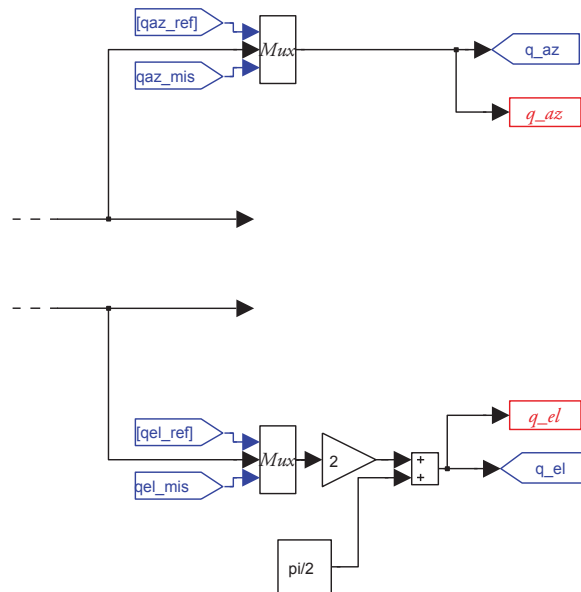


Figura 2.11. Blocchi di scrittura dei dati di *azimuth* ed *elevation* in opportune variabili, si notino i blocchi di condizionamento del segnale di *elevation*.

2. 4. 2. Controllore di posizione

Come evidenziato anche in Figura 2.2 a pagina 12 gli anelli di controllo della posizione dello specchio del sensore sono chiusi da un controllore in cascata che ha in ingresso la posizione e la velocità. I due blocchi sono praticamente identici come schema di controllo, per entrambi i gradi di libertà ed hanno in uscita i *set-point* di corrente che sono l'ingresso dei *driver* dei motori elettrici.

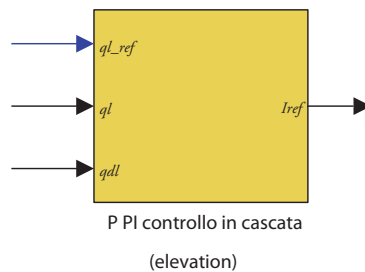


Figura 2.12. Controllore di posizione, per il grado di libertà di *elevation*, quello per l'*azimuth* è identico.

Una analisi più approfondita del controllore verrà fatta nel capitolo, dove si mostrerà anche la regolazione dei parametri di controllo.

2.4.3. Motore ed elettronica di controllo

La trattazione del controllo del motore elettrico e della generazione di coppia richiede una trattazione più approfondita e sarà affrontata nel capitolo 4.

Per quanto riguarda la comprensione dello schema generale del modello è sufficiente sapere che i blocchi dei motori, anche essi praticamente coincidenti tra loro, ricevono in ingresso la posizione, la velocità ed il *set-point* di corrente, generato dal controllore e hanno in uscita la coppia generata.

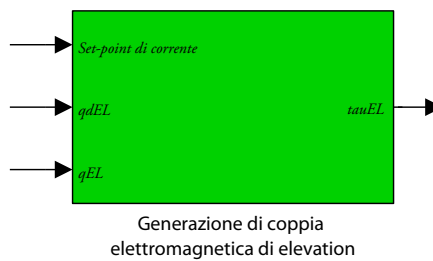


Figura 2.13. Blocco che modella il motore e l'elettronica di controllo.

2.4.4. Attrito

Come già spiegato il blocco di attrito è di notevole complessità, come si nota anche visualizzando lo schema interno, Figura 2.15 a pagina 22.



Figura 2.14. Blocco dell'attrito, l'uscita *Taufric* non viene più usata ed è quindi mandata in un blocco terminatore.

Approfondire il funzionamento di tale blocco non rientra nel campo di interesse di questa tesi, è sufficiente sapere che esso riesce a modellare in modo corretto i fenomeni di attrito, in particolare quello di primo distacco, che interviene in modo preponderante all'avvio del moto e durante le fasi di inversione, in cui quindi la velocità si annulla per qualche istante.

Il blocco riceve come *input* la velocità e la coppia generata dal motore e restituisce in *output* la coppia di attrito e la coppia generata al motore meno la coppia di attrito.

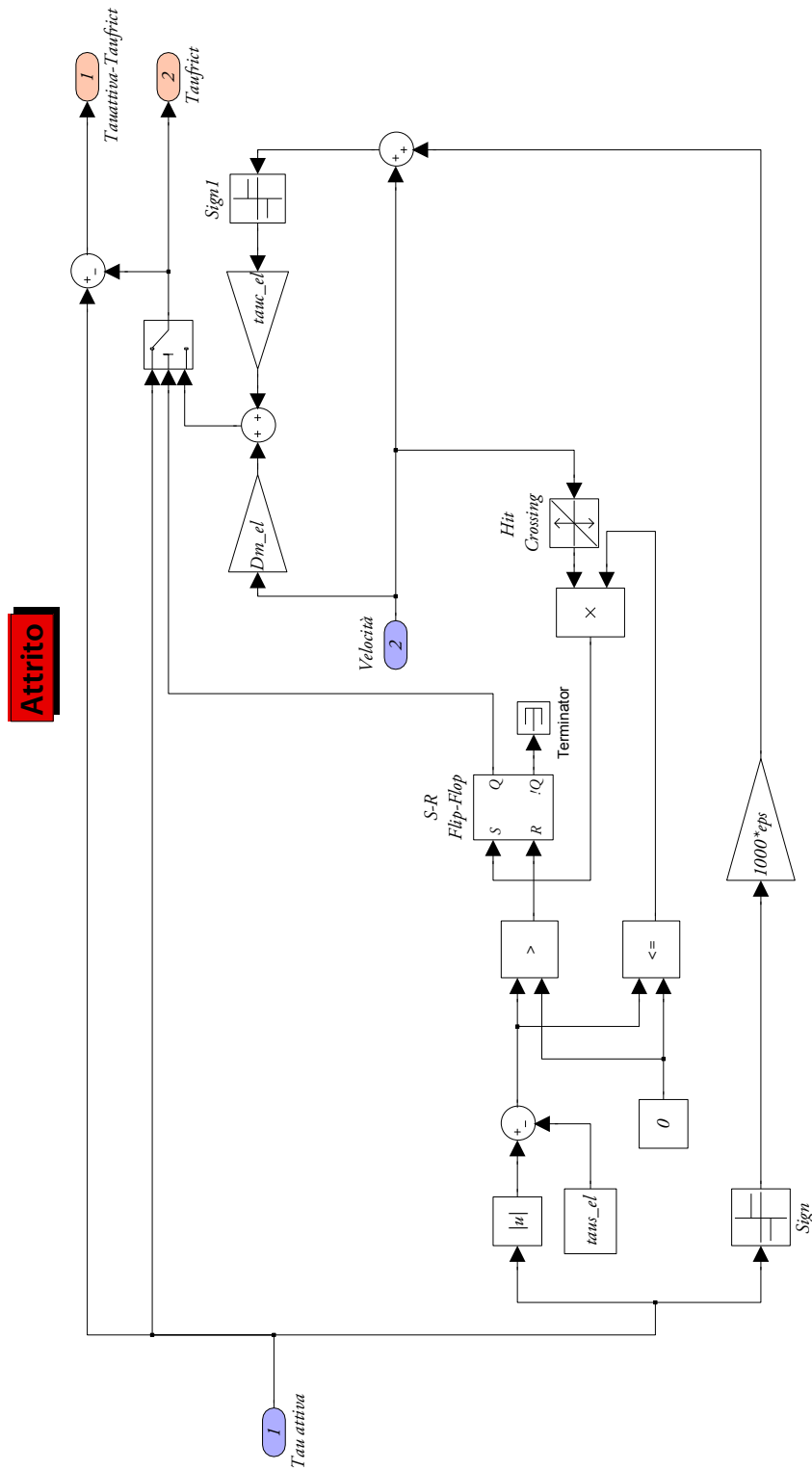


Figura 2.15. Schema interno del blocco di attrito.

2.4.5. Dinamica meccanica del sensore

Anche la trattazione della dinamica meccanica del sensore è troppo complessa per essere qui affrontata e sarà invece discussa con più attenzione nel capitolo 3.

Per quanto riguarda la comprensione dello schema generale del modello è sufficiente sapere che il blocco della dinamica riceve in ingresso la coppia generata dai motori di *azimuth* ed *elevation*, a cui è stata sottratta la coppia di attrito, oltre all'angolo di beccheggio e le accelerazioni sui tre assi del velivolo. Le uscite sono invece costituite dalla posizione e dalla velocità di *azimuth* ed *elevation*.

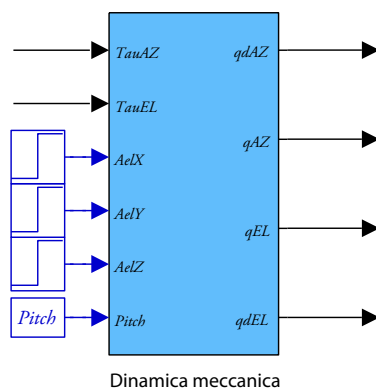


Figura 2.16. Blocco che modella la dinamica meccanica del sensore.

2.4.6. Sensori di posizione e velocità

La modellazione dei sensori è stata fatta secondo quanto riportato da SELEX Galileo nella documentazione degli stessi.

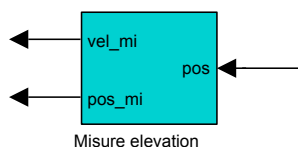


Figura 2.17. Blocco che simula i sensori di posizione e velocità

Essi sono dei *Resolver to Digital*, modellabili tramite la seguente funzione di trasferimento.

$$[2.4] \quad G_{RtoD} = \frac{G_{RtoD,open}}{G_{RtoD,open} + 1}; \quad G_{RtoD,open} = g \left(\frac{A_1}{s} \frac{s + B_1}{s + B} \frac{A_2}{s} \right);$$

con i parametri seguenti:

$$\begin{aligned}
 A_1 &= 2.85 \text{ Hz rad;} \\
 B_1 &= 526.5 \text{ Hz rad;} \\
 [2.5] \quad g &= 10; \\
 A_2 &= 2.85 \cdot 10^5 \text{ Hz rad;} \\
 B_2 &= 5265 \text{ Hz rad.}
 \end{aligned}$$

Il segnale di velocità è ricavato dall'uscita di posizione del convertitore mediante opportuna derivazione effettuata in modo digitale.

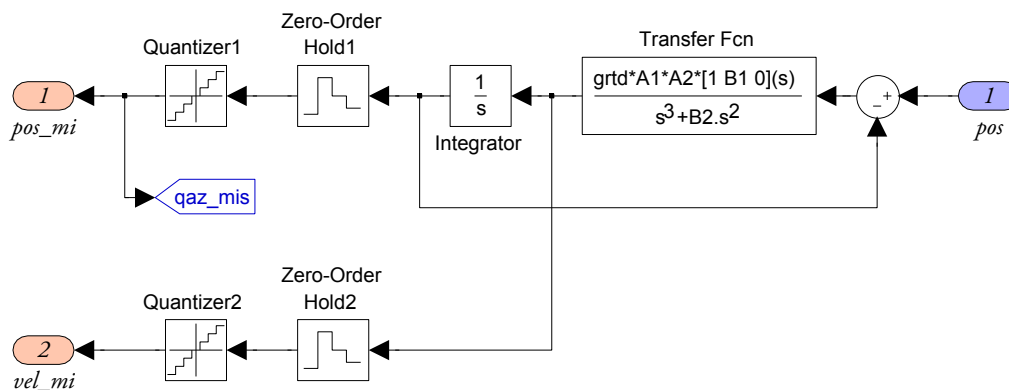


Figura 2.18. Schema interno del blocco che simula i sensori.

2.4.7. Finestre grafiche

Il blocco *Grafica* che è presente nello schema *Simulink* non ha un significato fisico ma permette la realizzazione di opportune finestre grafiche che mostrano l'andamento della simulazione, potendo avere quindi anche un immediato riscontro dei risultati che si stanno ottenendo.

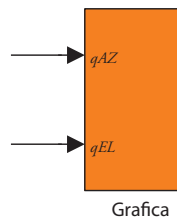


Figura 2.19. Blocco di grafica del modello *Simulink*.

Esso è costituito internamente da due *S-function*, una *animIRST.m* ed una *graficiIRST.m*, che verranno adesso brevemente descritte.

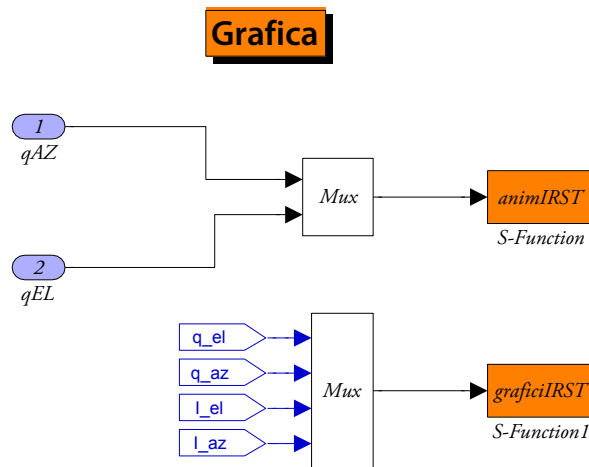


Figura 2.20. Schema interno del blocco *Grafica*.

La funzione `animIRST.m` riceve in ingresso i valori di posizione angolare di *azimuth* ed *elevation*, vengono inoltre automaticamente passate ad essa anche le informazioni relative all'istante di simulazione.

Con questi dati viene creata una figura che contiene nella parte superiore una rappresentazione del sensore e nella parte inferiore un barra indicatrice dello scorrere del tempo, che permette di visualizzare immediatamente a quale punto della simulazione ci si trova e quanto manca alla fine.

Un esempio di tale finestra è visibile in Figura 2.21 a pagina 26.

La rappresentazione tridimensionale schematizza in modo molto semplice il sensore, mostrando tuttavia gli elementi principali, sono infatti presenti:

- La copertura;
- Il supporto;
- Lo specchio;
- La telecamera.

Questo permette di visualizzare con chiarezza la configurazione attuale. È inoltre anche rappresentato un raggio luminoso, che permette di sapere quale sia la LOF attuale del sensore.

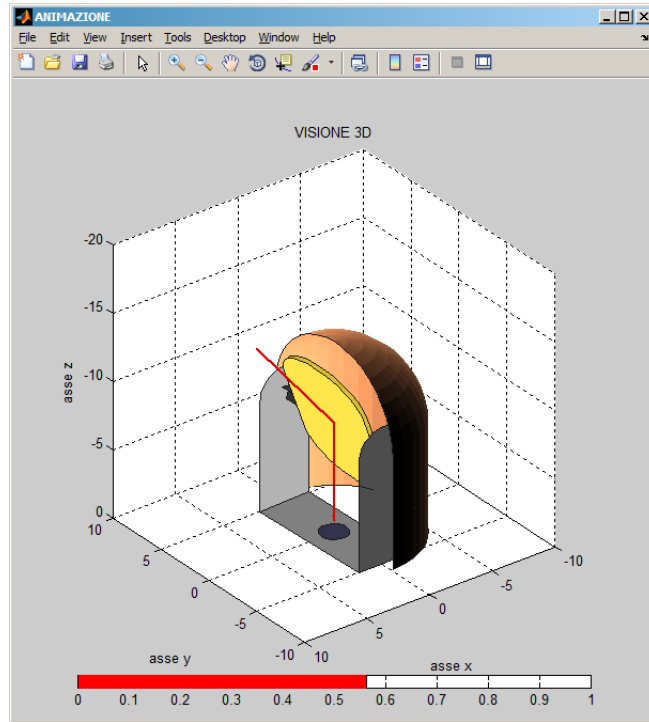


Figura 2.21. Esempio di finestra generata da animIRST.m.

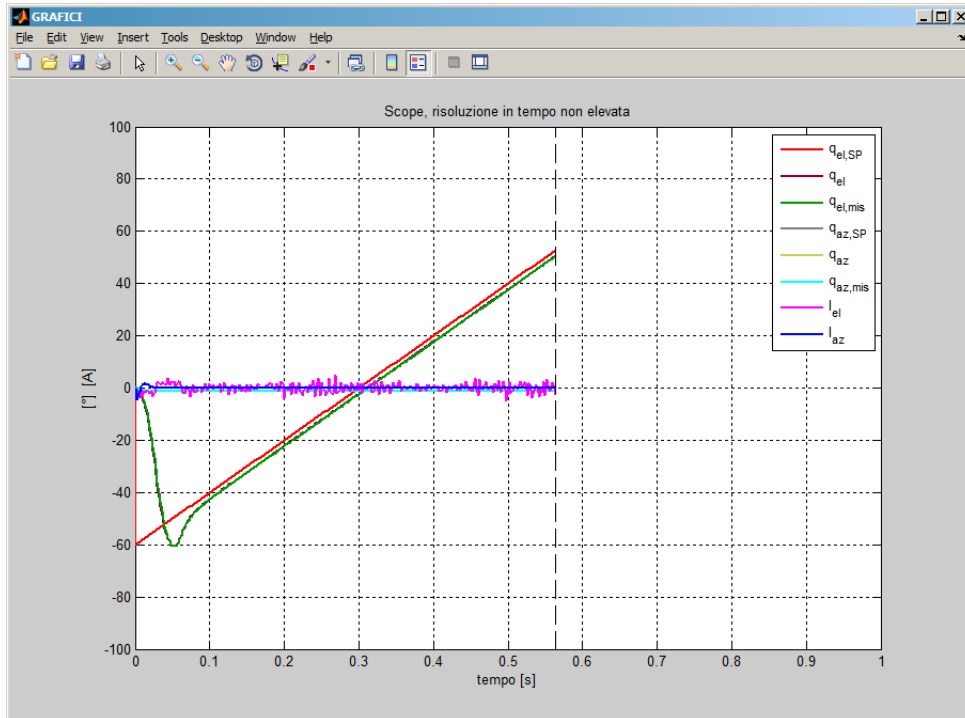


Figura 2.22. Esempio di finestra generata da graficiIRST.m.

La funzione `graficiIRST.m` riceve in ingresso i valori di alcune grandezze caratteristiche della simulazione riceve inoltre l'istante attuale della simulazione e la durata totale della simulazione.

Tale dato permette di creare un grafico che mostra l'andamento delle grandezze fino all'istante attuale della simulazione, mostrando contemporaneamente anche quanto manca alla fine della simulazione.

Entrambe le finestre grafiche vengono generate con una frequenza inferiore rispetto a quella con cui sono memorizzati i dati nel *Workspace*, per evitare di rendere eccessivamente lenta la simulazione. Per tale motivo tuttavia la finestra generata da `graficiIRST.m` non ha una risoluzione elevata.

Capitolo 3

Modellazione multicorpo

Una corretta simulazione del sensore IRST rende necessaria in primo luogo una accurata modellazione della sua dinamica meccanica.

Si è quindi dovuto scrivere un modello multicorpo del servomeccanismo. I passaggi per ricavare le equazioni che descrivono la dinamica meccanica del sensore e la loro implementazione verranno trattati in questo capitolo.

Come approccio alla scrittura delle leggi di moto si è scelto quello delle equazioni di Lagrange.

Si analizzeranno prima i passaggi che portano alla scrittura delle equazioni. Verranno in primo luogo descritti i sistemi di riferimento adottati. Successivamente si procederà all'individuazione delle opportune coordinate libere. Dopo avere individuato quindi il corretto insieme di variabili da utilizzare si analizzeranno tutti i termini che verranno coinvolti nella scrittura delle equazioni di Lagrange e verrà infine ricavato il sistema di equazioni stesso.

In seguito, e per ultimo punto, si descriverà come la dinamica è stata implementata al calcolatore tramite un modello *Simulink*, interfacciato a *MATLAB* tramite opportune *S-Function*.

Si andranno ora a descrivere i vari sistemi di riferimento che hanno permesso una corretta scrittura delle equazioni dinamiche.

3.1. Sistemi di riferimento

Vengono ora descritti i cinque sistemi di riferimento che si sono adottati. Tali sistemi sono stati introdotti sia al fine di rendere possibile la scrittura delle equazioni dinamiche della meccanica del servomeccanismo e per poter individuare le corrette coordinate libere, sia per altre operazioni che verranno affrontate in seguito. Si è scelto di descrivere comunque tutti i sistemi di riferimento ora per maggiore chiarezza.

Si sono definiti un sistema di riferimento solidale alla terra, uno solidale al velivolo, uno solidale alla base del sensore, uno solidale al supporto dello specchio ed un ultimo solidale allo specchio stesso.

Verranno ora descritti tutti e cinque tali sistemi di riferimento.

Tabella 3.1. Differenti sistemi di riferimento.

$\{x_g, y_g, z_g\}$	$\{x_b, y_b, z_b\}$	$\{x_s, y_s, z_s\}$	$\{x_a, y_a, z_a\}$	$\{x_m, y_m, z_m\}$
solidale alla terra	solidale al velivolo	solidale al sensore	solidale al supporto	solidale allo specchio

3.1.1. Sistema di riferimento terrestre

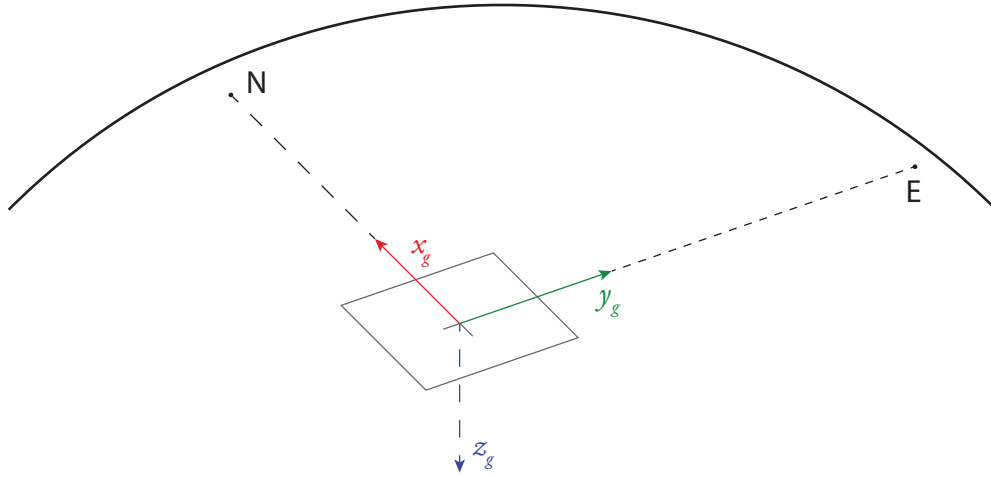


Figura 3.1. Disposizione degli assi $\{x_g, y_g, z_g\}$ nel sistema di riferimento terrestre.

La prima terna di riferimento considerata è solidale con la terra, considerabile fissa. Il piano x - y è tangente alla superficie terrestre, con l'origine coincidente con il punto di tangenza.

Gli assi invece sono disposti nel modo seguente:

- asse x : diretto verso il Nord geografico locale;
- asse y : diretto verso Est;
- asse z : diretto verso il basso, a formare quindi una terna destra.

Nel seguito tale terna sarà indicata dal pedice g , $\{x_g, y_g, z_g\}$.

3.1.2. Sistema di riferimento del velivolo

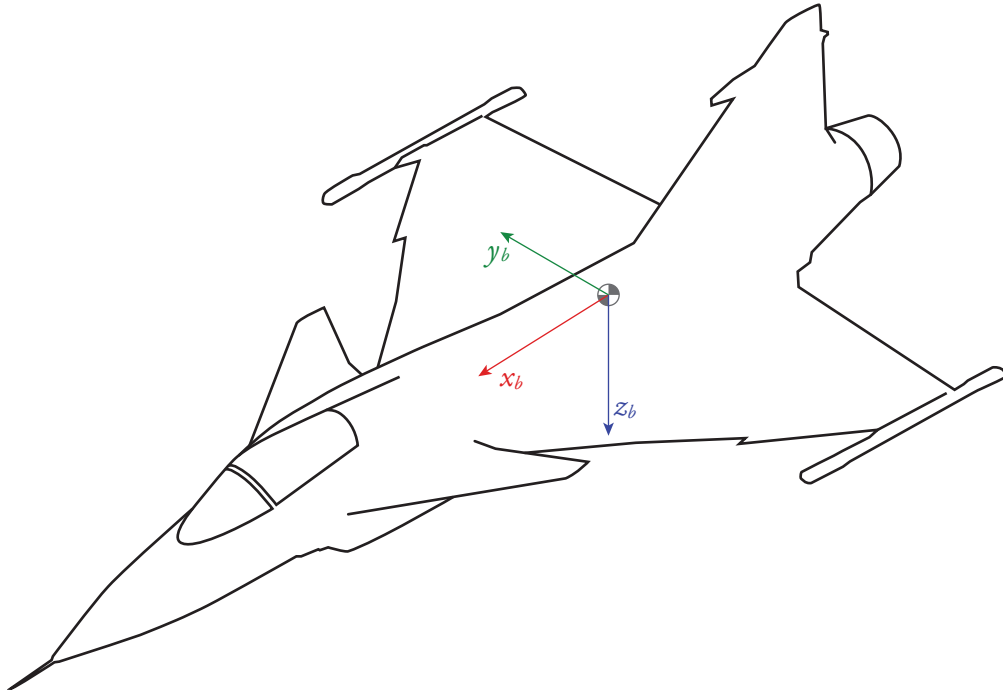


Figura 3.2. Disposizione degli assi $\{x_b, y_b, z_b\}$ nel sistema di riferimento solidale all'aereo.

La seconda terna di riferimento è mobile rispetto alla prima ed è invece solidale al velivolo.

L'origine è posta nel baricentro del velivolo stesso. Gli assi invece sono disposti nel modo seguente:

- asse x : allineato con l'asse del velivolo e diretto verso la prua;
- asse y : diretto verso la semiala destra;
- asse z : diretto verso il basso, a formare quindi una terna destra.

Nel seguito tale terna sarà indicata dal pedice $b, \{x_b, y_b, z_b\}$.

3.1.3. Sistema di riferimento del sensore

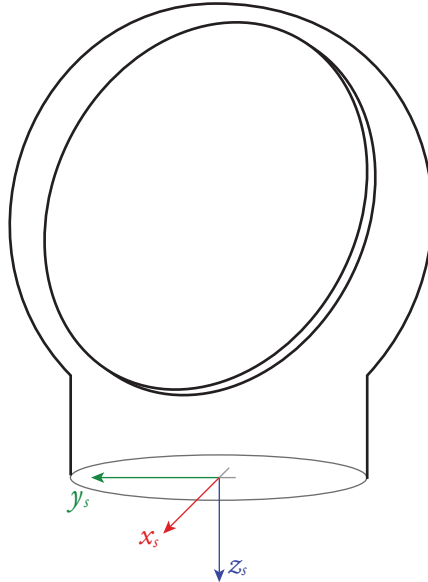


Figura 3.3. Disposizione degli assi $\{x_s, y_s, z_s\}$ nel sistema di riferimento solidale alla base del sensore.

Un sistema di riferimento è considerato disposto sulla base del sensore, il piano $x-y$ è coincidente con la base stessa.

L'origine del sistema di riferimento è posta nel centro della base. Gli assi invece sono disposti nel modo seguente:

- asse x : diretto verso la prua dell'aereo
- asse y : perpendicolare agli altri due assi, a formare una terna destra
- asse z : diretto verso il basso.

Nel seguito tale terna sarà indicata dal pedice s , $\{x_s, y_s, z_s\}$.

Tale sistema di riferimento può essere considerato orientato come il sistema precedente, b , ed è anche esso solidale con il velivolo.

3.1.4. Sistema di riferimento del supporto

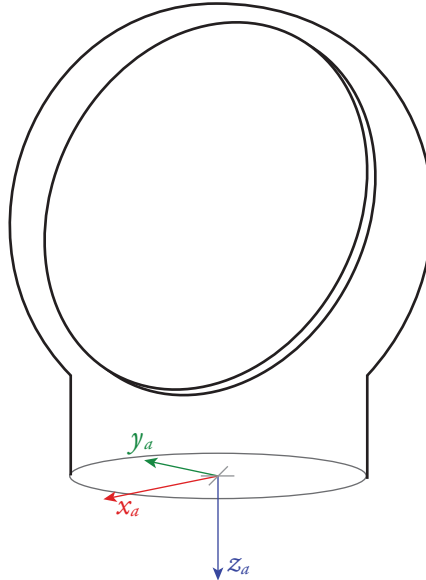


Figura 3.4. Disposizione degli assi $\{x_a, y_a, z_a\}$ nel sistema di riferimento solidale al supporto dello specchio del sensore.

Un ultimo sistema di riferimento è considerato solidale al supporto dello specchio del sensore.

L'origine del sistema di riferimento è posta al centro della base. Gli assi invece sono disposti nel modo seguente:

- asse x : parallelo alla proiezione della perpendicolare al piano dello specchio;
- asse y : perpendicolare agli altri due assi, a formare una terna destra;
- asse z : diretto verso il basso.

Nel seguito tale terna sarà indicata dal pedice a , $\{x_a, y_a, z_a\}$.

Si noti come tale terna sia ottenibile dalla terna, $\{x_i, y_i, z_i\}$, attraverso una rotazione variabile attorno all'asse z_i , che, come verrà descritto in seguito, corrisponde all'angolo di *azimuth*.

3.1.5. Sistema di riferimento dello specchio

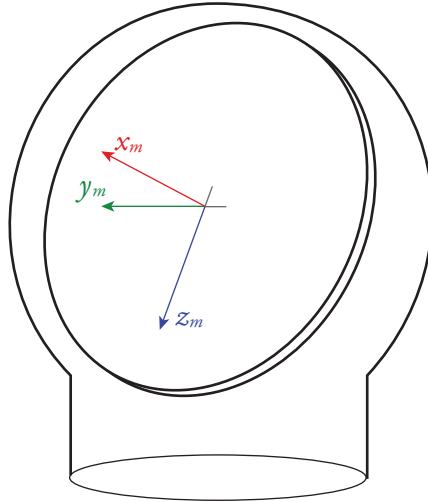


Figura 3.5. Disposizione degli assi $\{x_m, y_m, z_m\}$ nel sistema di riferimento solidale allo specchio del sensore.

Un ultimo sistema di riferimento è considerato disposto sullo specchio del sensore, il piano $z-y$ è parallelo allo specchio. L'origine del sistema di riferimento è posta nel centro dello specchio. Gli assi invece sono disposti nel modo seguente:

- asse x : perpendicolare al piano dello specchio, diretto verso l'esterno;
- asse y : perpendicolare agli altri due assi, a formare una terna destra;
- asse z : diretto verso la base del sensore.

Nel seguito tale terna sarà indicata dal pedice m , $\{x_m, y_m, z_m\}$.

Si noti come tale terna sia ottenibile dalla terna precedentemente indicata, $\{x_a, y_a, z_a\}$, attraverso una rotazione attorno all'asse y_a .

3.2. Assunzioni sul moto del velivolo

Nel seguito della trattazione verranno fatte delle opportune ipotesi, in particolare sulla dinamica del velivolo, al fine di semplificare la scrittura delle equazioni. Si è infatti assunto costante l'assetto del velivolo, anche se opportunamente modificabile.

In aggiunta a questo le accelerazioni del velivolo, che vengono ovviamente trasmesse anche al sensore, non sono state considerate direttamente nella scrittura delle equazioni ma sono invece inserite come disturbi sul sensore stesso.

Tali ipotesi hanno permesso di considerare la base del servomeccanismo ferma, agli effetti della scrittura delle equazioni.

Anche con queste ipotesi semplificative rimane tuttavia vincolante poter definire l'assetto del velivolo, e quindi della base del sensore, nello spazio. Per fare questo ci si serve dalla parametrizzazione di rotazione descritta dagli angoli di *Roll*, *Pitch* e *Yaw*, indicati rispettivamente da $\{\phi, \vartheta, \psi\}$.

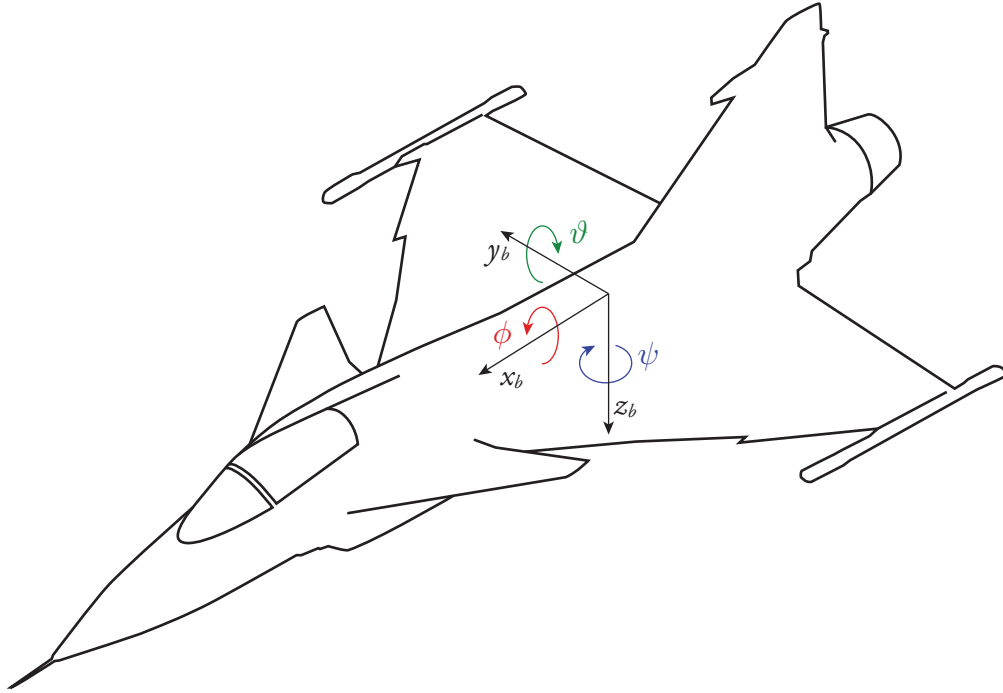


Figura 3.6. Parametrizzazione di rotazione per il velivolo, descritta dagli angoli di *Roll*, *Pitch* e *Yaw*.

3.3. Coordinate libere

Per la scrittura delle equazioni dinamiche del meccanismo è necessario individuare le opportune coordinate libere. Il numero di coordinate libere deve essere corrispondente ai gradi di libertà del sistema dinamico.

Grazie alle ipotesi espresse precedentemente, il velivolo, e quindi la base del servomeccanismo stesso, è considerabile fissa. Questo limita i gradi di libertà alle due rotazioni permesse dai giunti del servomeccanismo.

Tali rotazioni sono corrispondenti alle coordinate libere di *azimuth* ed *elevation*, che nel seguito verranno indicate con la notazione di $\{q_{az}, q_{el}\}$.

La rotazione di *azimuth* avviene attorno all'asse z , e corrisponde alla rotazione necessaria per trasformare il sistema di riferimento s in a . La rotazione di *elevation*

avviene attorno all'asse y_m corrisponde invece alla rotazione che orienta il sistema di riferimento s come il sistema m .

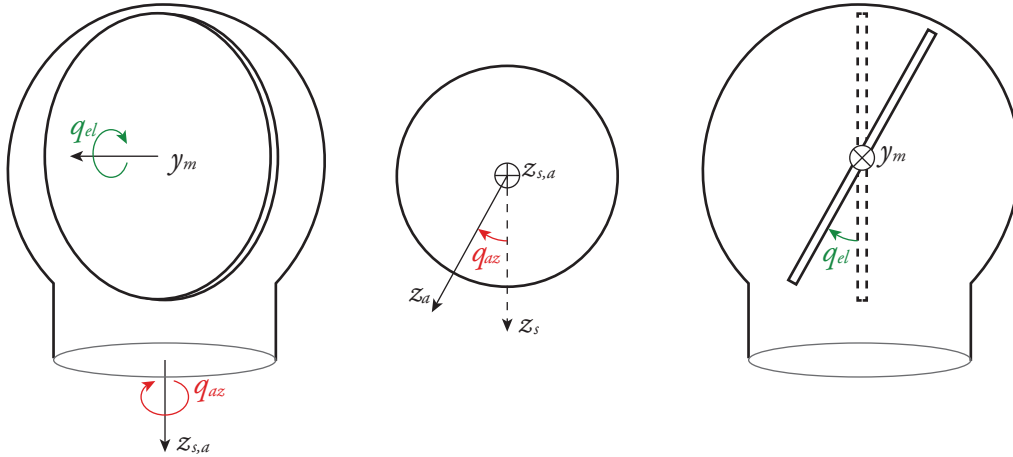


Figura 3.7. Coordinate libere scelte per la scrittura delle equazioni di moto del sensore, rotazione attorno agli assi di *azimuth* ed *elevation*.

Si definisce inoltre la distanza tra i sistemi di riferimento s ed m , ed il centro di massa del meccanismo.

$$[3.1] \quad \begin{aligned} n_{,CM} &= \{r_{2,CM}^x \ r_{2,CM}^y \ r_{2,CM}^z\}; \\ n_{,CM} &= \{r_{2,CM}^x \ r_{2,CM}^y \ r_{2,CM}^z\}. \end{aligned}$$

3.4. Equazioni di Lagrange

La scrittura del sistema di equazioni di Lagrange richiede di identificare:

- *L'energia cinetica;*
- *L'energia potenziale;*
- *I lavori virtuali.*

Il servomeccanismo è stato considerato composto da due corpi, uno rotante attorno al grado di libertà di *azimuth* ed un secondo, coincidente con lo specchio, vincolato al primo attorno al gradi di libertà di *elevation*.

Il primo corpo risulta essere solidale alla terna s e verrà indicato dal pedice \square_1 , mentre il secondo risulta essere solidale alla terna m e verrà indicato dal pedice \square_2 .

Si andranno ora a scrivere i termini necessari per le equazioni per i due corpi.

3.4.1. Energia cinetica

Per un corpo rigido l'energia cinetica è esprimibile come la somma di un contributo dovuto alla velocità lineare del centro di massa del corpo ed alla sua massa ed un secondo contributo dovuto alla velocità di rotazione del corpo ed alla sua matrice di inerzia:

$$\begin{aligned}
 T &= \frac{1}{2} m v_{CM} \cdot v_{CM} + \frac{1}{2} \omega^T I^{CM} \omega = \\
 [3.2] \quad &= \frac{1}{2} \left(m(\dot{x}_{CM} + j_{CM} + \dot{z}_{CM}) + \left\{ \omega_x \ \omega_y \ \omega_z \right\} \begin{bmatrix} I_x^{CM} & I_{xy}^{CM} & I_{xz}^{CM} \\ I_{yx}^{CM} & I_y^{CM} & I_{yz}^{CM} \\ I_{zx}^{CM} & I_{zy}^{CM} & I_z^{CM} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \right).
 \end{aligned}$$

Per quanto riguarda la velocità del centro di massa essa può essere scritta come la somma della velocità dell'origine del sistema di riferimento ed il prodotto della velocità di rotazione per la distanza del baricentro dall'origine del sistema di riferimento stesso.

$$[3.3] \quad v_{CM} = v_0 + \omega \times r_{CM}.$$

È necessario quindi ricavare le velocità angolari espresse nel sistema di riferimento di ciascuno dei due corpi.

Per il primo essa, esprimibile nel sistema di riferimento s , è:

$$[3.4] \quad \omega_1^s = \begin{Bmatrix} 0 \\ 0 \\ \dot{q}_{AZ} \end{Bmatrix}.$$

Per il secondo corpo va invece considerato che la velocità di *azimuth* nel sistema di riferimento m ha componenti sia lungo il primo che lungo il terzo asse. Si ottiene quindi:

$$[3.5] \quad \omega_2^m = \begin{Bmatrix} -\sin(q_{EL}) \dot{q}_{AZ} \\ \dot{q}_{EL} \\ \cos(q_{EL}) \dot{q}_{AZ} \end{Bmatrix}.$$

Considerando nulle per le ipotesi già espresse le velocità del centro del sistema di riferimento s , si ottiene la seguente velocità per il centro di massa del primo corpo.

$$[3.6] \quad v_{1,CM}^s = (-\dot{q}_{AZ} r_{1,CM}^y) \hat{x}_s + (\dot{q}_{AZ} r_{1,CM}^x) \hat{y}_s.$$

Similmente per il secondo corpo.

$$\begin{aligned}
 v_{2,CM}^m &= (-\dot{q}_{EL} r_{2,CM}^z - \cos(q_{EL}) \dot{q}_{AZ} r_{2,CM}^y) \hat{x}_m + \dots \\
 [3.7] \quad &\dots + (\cos(q_{EL}) \dot{q}_{AZ} r_{2,CM}^x - \sin(q_{EL}) \dot{q}_{AZ} r_{2,CM}^z) \hat{y}_m + \dots \\
 &\dots + (\sin(q_{EL}) \dot{q}_{AZ} r_{2,CM}^y - \dot{q}_{AZ} r_{2,CM}^x) \hat{z}_m.
 \end{aligned}$$

Tale termine viene riportato per motivi di completezza. Verrà infatti evidenziato dai dati forniti dal costruttore come per il sensore in esame il baricentro coincide esattamente con il centro dello specchio e la distanza $r_{2,CM}$ risulta quindi nulla.

Noti i termini appena scritti può quindi essere definita l'energia cinetica dei due corpi, il supporto rotante e lo specchio, che risulta infatti pari a:

$$[3.8] \quad T_1 = \frac{1}{2} m_1 (r_{1,CM}^{x^2} + r_{1,CM}^{y^2}) \dot{q}_{AZ}^2 + \frac{1}{2} I_{1,z}^{CM,a} \dot{q}_{AZ}^2;$$

$$\begin{aligned}
 T_2 &= \frac{1}{2} m_2 \left((r_{2,CM}^{x^2} + r_{2,CM}^{z^2}) \dot{q}_{EL}^2 + (r_{2,CM}^{x^2} + r_{2,CM}^{y^2}) \cos^2(q_{EL}) \dot{q}_{AZ}^2 + \dots \right. \\
 &\quad \dots + (r_{2,CM}^{y^2} + r_{2,CM}^{z^2}) \sin^2(q_{EL}) \dot{q}_{AZ}^2 + 2r_{2,CM}^x r_{2,CM}^z \sin(q_{EL}) \cos(q_{EL}) \dot{q}_{AZ}^2 + \dots \\
 [3.9] \quad &\quad \left. \dots - 2r_{2,CM}^y r_{2,CM}^z \cos(q_{EL}) \dot{q}_{EL} \dot{q}_{AZ} + 2r_{2,CM}^x r_{2,CM}^y \sin(q_{EL}) \dot{q}_{EL} \dot{q}_{AZ} \right) + \dots \\
 &\quad \dots + \frac{1}{2} (I_{2,x}^{CM,m} \sin^2(q_{EL}) \dot{q}_{AZ}^2 - 2I_{2,xy}^{CM,m} \sin(q_{EL}) \dot{q}_{EL} \dot{q}_{AZ} + \dots \\
 &\quad \dots - 2I_{2,xz}^{CM,m} \sin(q_{EL}) \cos(q_{EL}) \dot{q}_{AZ}^2 + I_{2,y}^{CM,m} \dot{q}_{EL}^2 + \dots \\
 &\quad \dots + 2I_{2,yz}^{CM,m} \cos(q_{EL}) \dot{q}_{EL} \dot{q}_{AZ} + I_{2,z}^{CM,m} \cos^2(q_{EL}) \dot{q}_{EL}^2).
 \end{aligned}$$

L'energia cinetica del sistema è quindi esprimibile come:

$$[3.10] \quad T = T_1 + T_2.$$

3.4.2. Energia potenziale

La formula per il calcolo dell'energia potenziale è la seguente:

$$[3.11] \quad U = - m g z_g.$$

La distanza z_g dipende dall'assetto del velivolo, in particolare dal suo angolo di beccheggio ed è pari a:

$$[3.12] \quad z_g = z_s \cos(\vartheta) + x_s \sin(\vartheta).$$

Si deve quindi ottenere la distanza del baricentro dall'origine, nel sistema di riferimento s . Tale distanza può essere ottenuta premoltiplicando la distanza nelle coordinate locali per le opportune matrici di rotazione.

$$[3.13] \quad r_{1,cm}^s = R_1 r_{1,CM}^a = \begin{bmatrix} \cos(q_{AZ}) & -\sin(q_{AZ}) & 0 \\ \sin(q_{AZ}) & \cos(q_{AZ}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} r_{1,CM}^x \\ r_{1,CM}^y \\ r_{1,CM}^z \end{Bmatrix};$$

$$[3.14] \quad \begin{aligned} r_{2,cm}^s &= R_1 r_{2,CM}^a = R_1 R_2 r_{2,CM}^m = \dots \\ \dots &= \begin{bmatrix} \cos(q_{AZ}) & -\sin(q_{AZ}) & 0 \\ \sin(q_{AZ}) & \cos(q_{AZ}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(q_{EL}) & 0 & \sin(q_{EL}) \\ 0 & 1 & 0 \\ -\sin(q_{EL}) & 0 & \cos(q_{EL}) \end{bmatrix} \begin{Bmatrix} r_{2,CM}^x \\ r_{2,CM}^y \\ r_{2,CM}^z \end{Bmatrix}. \end{aligned}$$

Le relazioni appena scritte permettono di ottenere le seguenti espressioni per l'energia potenziale dei due corpi che costituiscono il sistema.

$$[3.15] \quad U_1 = -m_1 g r_{1,CM}^z \cos(\vartheta) + m_1 g r_{1,CM}^x \cos(q_{AZ}) \sin(\vartheta) - m_1 g r_{1,CM}^y \sin(q_{AZ}) \sin(\vartheta);$$

$$[3.16] \quad \begin{aligned} U_2 &= m_2 g r_{2,CM}^x \sin(q_{AZ}) \cos(\vartheta) - m_2 g r_{2,CM}^z \cos(q_{AZ}) \cos(\vartheta) + \dots \\ \dots &+ m_2 g r_{2,CM}^x \cos(q_{EL}) \cos(q_{AZ}) \sin(\vartheta) + \dots \\ \dots &+ m_2 g r_{2,CM}^z \sin(q_{EL}) \cos(q_{AZ}) \sin(\vartheta) - m_2 g r_{2,CM}^y \sin(q_{AZ}) \sin(\vartheta); \end{aligned}$$

Come per l'energia cinetica, l'energia potenziale del sistema risulta essere:

$$[3.17] \quad U = U_1 + U_2.$$

3.4.3. Lavori virtuali

I lavori virtuali agenti sul sistema sono dovuti alle due coppie introdotte dal motore e sono quindi pari a:

$$[3.18] \quad \begin{cases} W_{AZ} = Q_{AZ} \delta q_{AZ} = \tau_{AZ} \delta q_{AZ} \\ W_{EL} = Q_{EL} \delta q_{EL} = \tau_{EL} \delta q_{EL} \end{cases}.$$

3.4.4. Scrittura delle equazioni

Le relazioni ricavate nelle sezioni precedenti rendono possibile scrivere le equazioni di Lagrange. Esse nella forma più generale sono:

$$[3.19] \quad \begin{aligned} \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_{AZ}} \right) - \frac{\partial \mathcal{L}}{\partial q_{AZ}} &= Q_{AZ}; \\ \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_{EL}} \right) - \frac{\partial \mathcal{L}}{\partial q_{EL}} &= Q_{EL}; \end{aligned}$$

dove:

$$[3.20] \quad \mathcal{L} = T - U.$$

Attraverso alcuni passaggi si ottiene una equazione vettoriale che descrive la dinamica del sensore, esprimibile nella forma:

$$[3.21] \quad M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \Gamma;$$

che permette di ottenere il termine di accelerazione:

$$[3.22] \quad \ddot{q} = M(q)^{-1} [-C(q, \dot{q})\dot{q} - G(q) + \Gamma];$$

dove:

$$[3.23] \quad q = \begin{Bmatrix} q_{AZ} \\ q_{EL} \end{Bmatrix}; \quad \Gamma = \begin{Bmatrix} \tau_{AZ} \\ \tau_{EL} \end{Bmatrix}.$$

Andremo ora a scrivere i termini che compongono le matrici.

$$[3.24] \quad M = \begin{bmatrix} A + B \cos^2(q_{EL}) + C \sin^2(q_{EL}) + D \sin(q_{EL}) \cos(q_{EL}), & \dots \\ I \cos(q_{EL}) + L \sin(q_{EL}), & \dots \\ \dots & \dots \\ \dots & E \cos(q_{EL}) + F \sin(q_{EL}) \\ \dots & H \end{bmatrix};$$

$$[3.25] \quad C = \begin{bmatrix} 2(C - B) \sin(q_{EL}) \cos(q_{EL}) + D(\cos^2(q_{EL}) - \sin^2(q_{EL})), & \dots \\ -M \sin(q_{EL}) \cos(q_{EL}) + N \cos^2(q_{EL}) + O \sin^2(q_{EL}), & \dots \\ \dots & \dots \\ \dots & F \cos(q_{EL}) - E \sin(q_{EL}) \\ \dots & (L - P) \cos(q_{EL}) - (Q + I) \sin(q_{EL}) \end{bmatrix};$$

$$[3.26] \quad G = \begin{cases} G_1 \sin(q_{AZ}) + G_2 \cos(q_{AZ}) + G_3 \cos(q_{EL}) \sin(q_{AZ}) + G_4 \cos(q_{AZ}) + G_5 \cos(q_{AZ}) \\ G_6 \cos(q_{EL}) + G_7 \sin(q_{EL}) + G_8 \sin(q_{EL}) \cos(q_{AZ}) + G_9 \cos(q_{EL}) \cos(q_{AZ}) \end{cases}$$

Dove le lettere che compaiono nelle equazioni precedenti sostituiscono le seguenti quantità:

$$[3.27] \quad \begin{aligned} A &= m_1 (r_{1,CM}^{x^2} + r_{1,CM}^{y^2}) + I_{1,z}^{CM,a}; \\ B &= m_2 (r_{2,CM}^{x^2} + r_{2,CM}^{y^2}) + I_{2,z}^{CM,m}; \\ C &= m_2 (r_{2,CM}^{y^2} + r_{2,CM}^{z^2}) + I_{2,x}^{CM,m}; \\ D &= 2(m_2 r_{2,CM}^x r_{2,CM}^z - I_{2,xz}^{CM,m}) \\ E &= -m_2 r_{2,CM}^y r_{2,CM}^z + I_{2,yz}^{CM,m} \\ F &= m_2 r_{2,CM}^x r_{2,CM}^y \end{aligned}$$

$$[3.28] \quad \begin{aligned} H &= m_1 (r_{2,CM}^{x^2} + r_{2,CM}^{z^2}) + I_{2,y}^{CM,m}; \\ I &= -m_2 r_{2,CM}^y r_{2,CM}^z + I_{2,yz}^{CM,m}; \\ L &= m_2 r_{2,CM}^y r_{2,CM}^x - I_{2,xy}^{CM,m}; \\ M &= m_2 (r_{2,CM}^{z^2} + r_{2,CM}^{x^2}) + I_{2,x}^{CM,m} - I_{2,z}^{CM,m}; \\ N &= m_2 r_{2,CM}^x r_{2,CM}^z - I_{2,xz}^{CM,m}; \\ O &= -m_2 r_{2,CM}^x r_{2,CM}^z + I_{2,xz}^{CM,m}; \\ P &= m_2 r_{2,CM}^x r_{2,CM}^y - I_{2,xy}^{CM,m}; \\ Q &= m_2 r_{2,CM}^y r_{2,CM}^z - I_{2,yz}^{CM,m}; \end{aligned}$$

$$[3.29] \quad \begin{aligned} G_1 &= -m_1 g r_{1,CM}^x \sin(\vartheta); \\ G_2 &= -m_1 g r_{1,CM}^y \sin(\vartheta); \\ G_3 &= -m_2 g r_{2,CM}^x \sin(\vartheta); \\ G_4 &= -m_2 g r_{2,CM}^z \sin(\vartheta); \\ G_5 &= -m_2 g r_{2,CM}^y \sin(\vartheta); \end{aligned}$$

$$[3.30] \quad \begin{aligned} G_6 &= m_2 g r_{2,CM}^x \cos(\vartheta); \\ G_7 &= m_2 g r_{2,CM}^z \cos(\vartheta); \\ G_8 &= -m_2 g r_{2,CM}^x \sin(\vartheta); \\ G_9 &= m_2 g r_{2,CM}^z \sin(\vartheta). \end{aligned}$$

3.4.5. Considerazioni sulle equazioni

Sull'equazione dinamica vettoriale di Lagrange, Equazione [3.21] a pagina 40, che descrive il moto del servomeccanismo possono essere fatte alcune utili considerazioni.

- La matrice di massa, M , risulta essere simmetrica e definita positiva, infatti $I = E$ ed $L = F$.
- La matrice di smorzamento, C , è composta da termini di forze centrifughe e di Coriolis.
- Il vettore G , che rappresenta un termine forzante aggiuntivo oltre alle coppie di *circular* ed *elevation* è dovuto all'azione della forza peso.

3.5. Disturbi sul sistema.

Nella trattazione precedente non è stato considerato alcun moto da parte del velivolo, poiché si è scelto di introdurlo come un disturbo sul sistema.

Per fare questo è comunque necessario eseguire alcune trasformazioni. Si consideri i vettori che descrivono i due gradi di libertà, essi possono essere scritti come:

$$\begin{aligned} [3.31] \quad \hat{n}_{AZ} &= \{0 \ 0 \ 1\}; \\ \hat{n}_{EL} &= \{0 \ 1 \ 0\}. \end{aligned}$$

L'accelerazione del velivolo va scritta nel sistema di riferimento a , tramite la formula:

$$[3.32] \quad a_{vel}^a = R_1^T a_{vel}.$$

Al fine di valutare la coppia agente sui due gradi di libertà per tale accelerazione che i due corpi subiscono è necessario valutare la distanza dal baricentro sempre nel sistema di riferimento a . Per il secondo corpo essa si ottiene con la formula.

$$[3.33] \quad r_{2,CM}^a = R_2 r_{2,CM}.$$

Le matrici R_1 ed R_2 sono le stesse che comparivano nelle equazioni [3.13] e [3.14] a pagina 39.

Noti questi valori, si deve considerare inoltre che un'accelerazione del primo corpo non può provocare una coppia sul grado di libertà di *elevation* poiché non è vincolato al velivolo attraverso questo asse, diversamente dallo specchio. La coppia attorno all'asse di *elevation* dovuta all'accelerazione del primo corpo andrà quindi ad aumentare la reazione vincolare.

Le coppie di disturbo risultano essere quindi:

$$\begin{aligned}
 \tau_{1,AZ} &= -(m_1 a_{vel}^a \times r_{1,CM}) \cdot \hat{n}_{AZ}; \\
 \tau_{2,AZ} &= -(m_2 a_{vel}^a \times r_{2,CM}^a) \cdot \hat{n}_{AZ}; \\
 \tau_{1,EL} &= 0; \\
 \tau_{2,EL} &= -(m_2 a_{vel}^a \times r_{2,CM}^a) \cdot \hat{n}_{EL}.
 \end{aligned}
 \tag{3.34}$$

Da cui si ottiene il vettore:

$$\Gamma_{dist} = \begin{Bmatrix} \tau_{1,AZ} + \tau_{2,AZ} \\ \tau_{1,EL} + \tau_{2,EL} \end{Bmatrix}.
 \tag{3.35}$$

3.6. Implementazione al calcolatore

Il sistema dinamico descritto dalle equazioni che sono state appena ricavate è stato implementato al calcolatore.

Il blocco *Simulink* che implementa le equazioni appena ricavate ha in ingresso le coppie di *azimuth* e di *elevation*, oltre ai valori delle accelerazioni del velivolo lungo i tre assi *b* ed all'angolo di beccheggio.

Le uscite sono invece le posizioni e le velocità angolari relative ai due gradi di libertà.

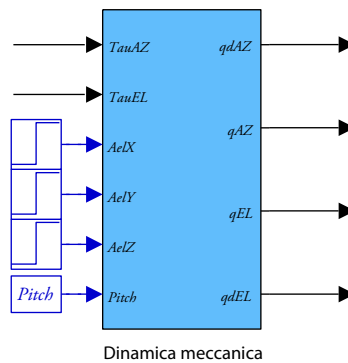


Figura 3.8. Blocco *Simulink* che implementa la dinamica meccanica del sensore.

Internamente il blocco della dinamica del sensore è costituito fondamentalmente da due *S-Function*. Una implementa le equazioni di Lagrange mentre la seconda calcola i disturbi dovuti alle accelerazioni del velivolo

Le uscite delle equazioni di Lagrange, che sono le accelerazioni del sistema nei due gradi di libertà, sono integrate due volte per ottenere la velocità e la posizione di *azimuth* ed *elevation*.

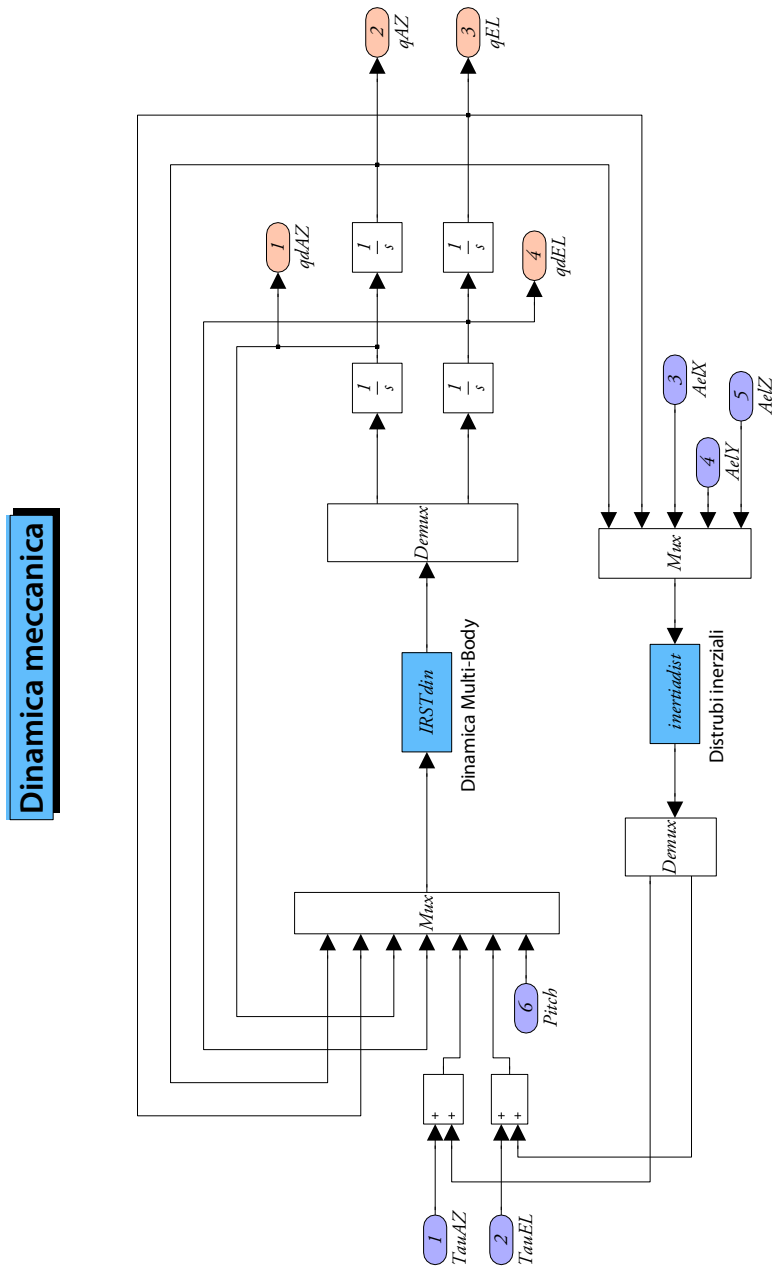


Figura 3.9. Schema interno del blocco Simulink che descrive la dinamica del sensore.

Le coppie di disturbo sono invece sommate a quelle di controllo e la loro somma costituisce uno degli ingressi del blocco che implementa le equazioni di moto.

Il listato delle *S-Function* è riportato insieme alle altre funzioni e *script MATLAB* nell'Appendice 26.

Capitolo 4

Modellazione del motore elettrico

La modellazione della dinamica elettrica del motore è stata abbastanza approfondita, anche per poter eventualmente tenere conto di disturbi dovuti a *offset* di correnti e per considerare la modulazione PWM. Tali elementi non si potrebbero infatti introdurre in modo appropriato se si utilizzasse un unico circuito equivalente resistenza–induttanza–forza controelettromotrice.

I motori installati nel sensore sono due motori *brushless* bifase DC, con le caratteristiche riportate nella tabella seguente.

Tabella 4.1. Caratteristiche del motore di *azimuth*.

	Valore	Tolleranza	Unità
Numero di fasi:	2		
Coppia di picco:	1.8		Nm
Costante di coppia:	0.34	±10%	Nm/A
Costante elettromagnetica:	0.34	±10%	Vs/rad
Induttanza degli avvolgimenti statorici:	$6.5 \cdot 10^{-4}$	±20%	H
Resistenza degli avvolgimenti statorici:	2.95	±5%	Ω
Corrente alla coppia di picco:	5.5		A
Velocità nominale:	15		rad/s
Corrente massima (impostate per protezione):	5.5		A

Tabella 4.2. Caratteristiche del motore di *elevation*.

	Valore	Tolleranza	Unità
Numero di fasi:	2		
Coppia di picco:	0.12		Nm
Costante di coppia:	0.064	±10%	Nm/A
Costante elettromagnetica:	0.064	±10%	Vs/rad
Induttanza degli avvolgimenti statorici:	$1.3 \cdot 10^{-3}$	±20%	H
Resistenza degli avvolgimenti statorici:	5.4	±5%	Ω
Corrente alla coppia di picco:	1.9		A
Velocità nominale:	15		rad/s
Corrente massima (impostate per protezione):	2		A

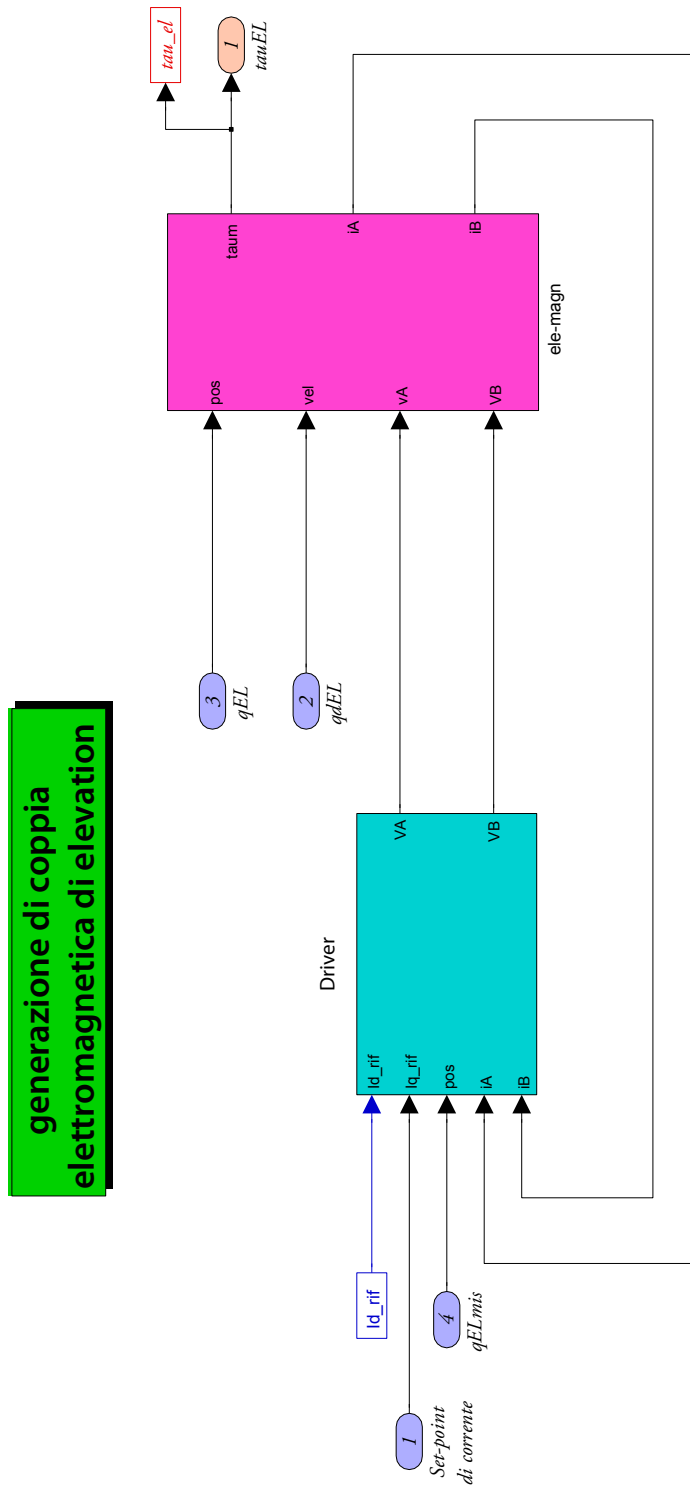


Figura 4.1. Schematizzazione del motore elettrico.

Entrambi i motori sono stati simulati con un modello identico, differenziato solamente per le grandezze caratteristiche del modello.

Verrà ora descritta quindi la schematizzazione *Simulink* che si è costruita, visibile in Figura 4.1 a pagina 48.

Il motore può essere diviso in due sottoblocchi abbastanza distinti, una parte di controllo di corrente, che simula l'elettronica di controllo del motore stesso ed un parte che invece simula il motore vero e proprio, in particolare la sua dinamica elettrica.

4.1. Modello della dinamica elettrica

Il blocco che schematizza il motore è rappresentato in Figura 4.3 a pagina 50.

I motori scelti sono dei motori bifase, con gli avvolgimenti collegati in modo indipendente al generatore di tensione. Il comportamento dei due avvolgimenti è modellizzabile come quello di una resistenza ed un'induttanza collegate in serie.

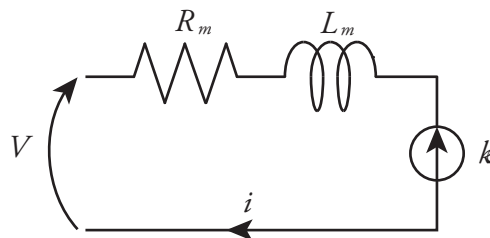


Figura 4.2. Circuito equivalente di ogni avvolgimento.

Il comportamento di tale circuito è descritto dalla seguente funzione di trasferimento, utilizzata anche nel modello *Simulink*.

$$[4.1] \quad G = \frac{1}{L s + R}.$$

La differenza di potenziale applicata al circuito è pari alla tensione di alimentazione meno la forza contro elettromotrice.

Essa ha come valore il prodotto della velocità moltiplicato per la costante di coppia e per il coseno della posizione relativa tra la spira ed il motore. Per la posizione del motore deve essere considerato il suo angolo elettrico, che è il prodotto tra l'angolo fisico ed il numero di coppie polari.

I due circuiti, alimentati da due differenti tensioni, hanno in uscita la corrente circolante la quale, moltiplicata per la costante di coppia ed il valore della posizione relativa già considerato precedentemente, consente di ottenere il valore di coppia generato dal motore stesso.

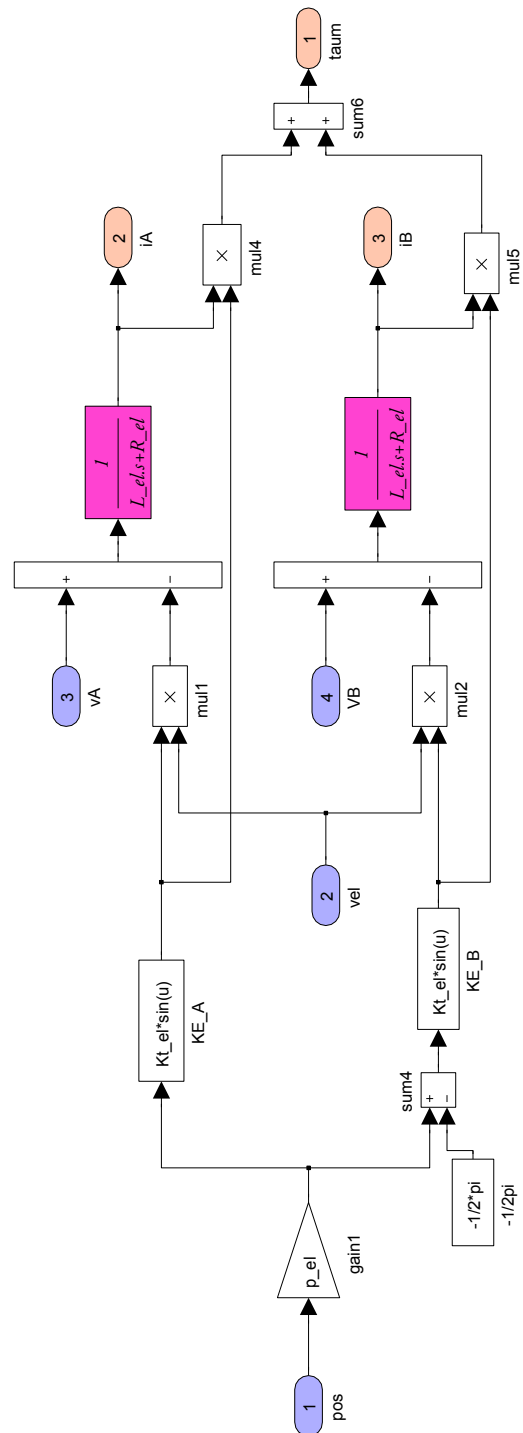


Figura 4.3. Schema del blocco di simulazione della dinamica elettrica.

Generalmente nei modelli elettrici tale valore di coppia, valutando opportunamente l'inerzia del motore, entra in una trasmissione che consente quindi di calcolare la coppia agente sul carico.

Nel sensore considerato i motori sono tuttavia *direct drive*, cioè collegati direttamente al carico, senza alcuna forma di riduttore e tale giunto può essere pensato come di rigidità infinita. Per tale motivo il valore di coppia generata dal motore, dopo essere stato opportunamente decurtato di quello relativo all'attrito, entra come ingresso nel blocco che simula la dinamica multicorpo.

4.2. Elettronica di controllo

Le tensioni in ingresso al motore vengono opportunamente generate dal *driver*. Esso ha in ingresso le correnti sui due avvolgimenti, generalmente denominati A e B.

Tali correnti, relative ad avvolgimenti fissi, possono essere facilmente trasformate in correnti relative a due direzioni mobili, fisse rispetto al rotore, una denominata in quadratura ed una denominata diretta.

Tale trasformazione, che implica principalmente un prodotto per opportuni valori di seno e coseno dell'angolo elettrico, prende il nome di trasformazione di Park ed è descritta dalle seguenti equazioni.

$$[4.2] \quad \begin{cases} i_d = i_A \cos(\vartheta) + i_B \sin(\vartheta) \\ i_q = -i_A \sin(\vartheta) + i_B \cos(\vartheta) \end{cases}$$

Si deve ora quindi considerare che la coppia è generata solo dalla componente di corrente in quadratura.

Per tale motivo il riferimento di corrente può essere considerato il riferimento della corrente in quadratura, mentre la corrente diretta ha un *set-point* sempre nullo.

Gli errori relativi alle due correnti sono quindi processati da due regolatori PI, il cui modello è mostrato nella figura seguente.

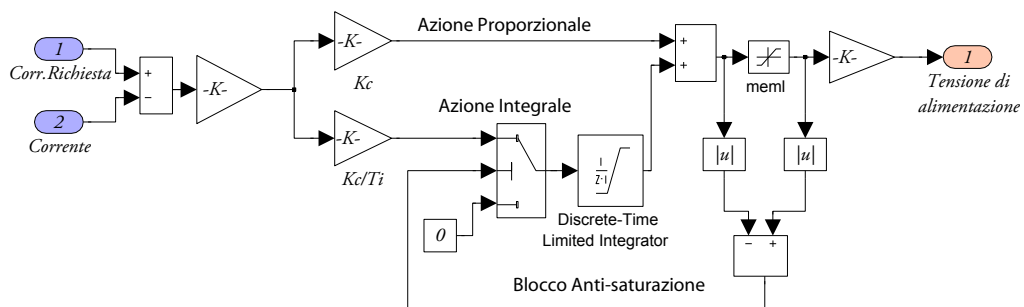


Figura 4.4. Regolatore di corrente PI.

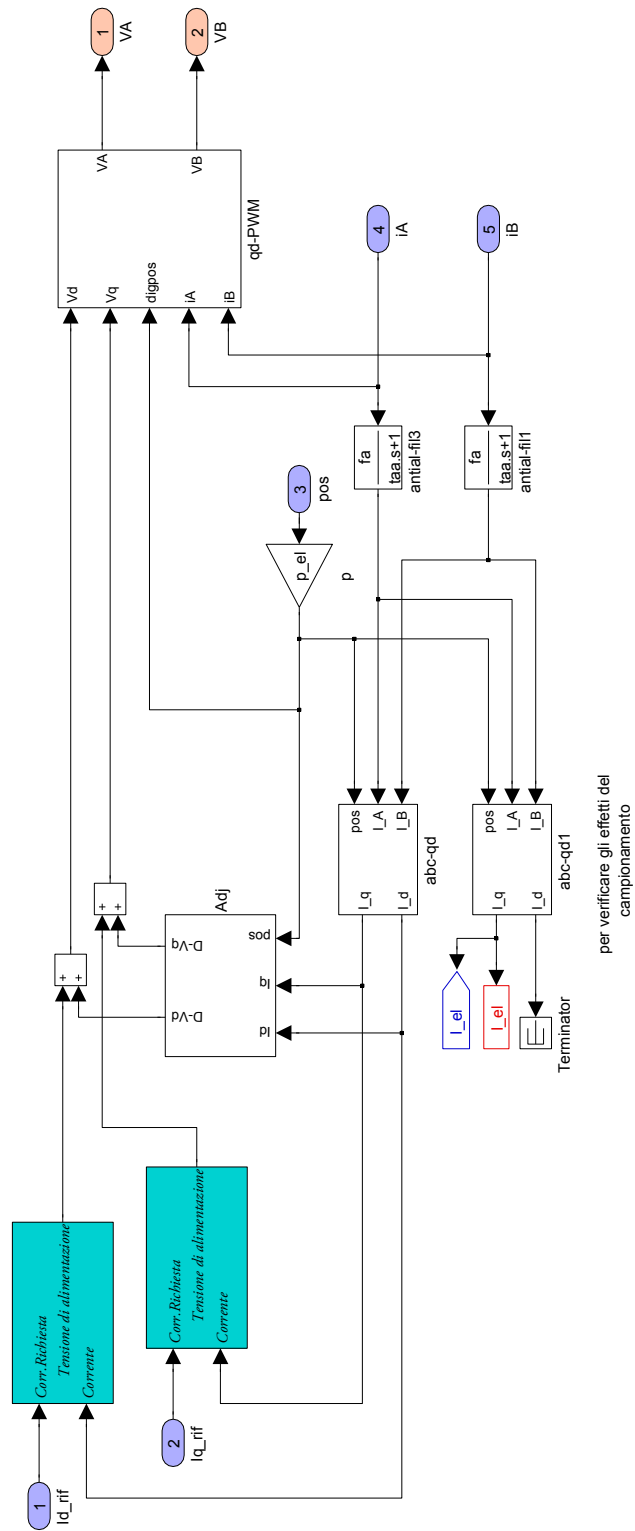


Figura 4.5. Schema del blocco di simulazione dell'elettronica di controllo.

Le uscite dei due regolatori sono le tensioni da applicare al motore, relative tuttavia anch'esse alla riferimento in quadratura e diretto. Per tale motivo esse vanno antitrasformate, con le seguenti equazioni.

$$[4.3] \quad \begin{cases} V_A = V_q \sin(\vartheta) + V_d \cos(\vartheta) \\ V_B = V_q \cos(\vartheta) - V_d \sin(\vartheta) \end{cases}$$

Le tensioni relative ai due avvolgimenti verranno quindi opportunamente modulate tramite la *Pulse Width Modulation*, come descritto nel prossimo paragrafo.

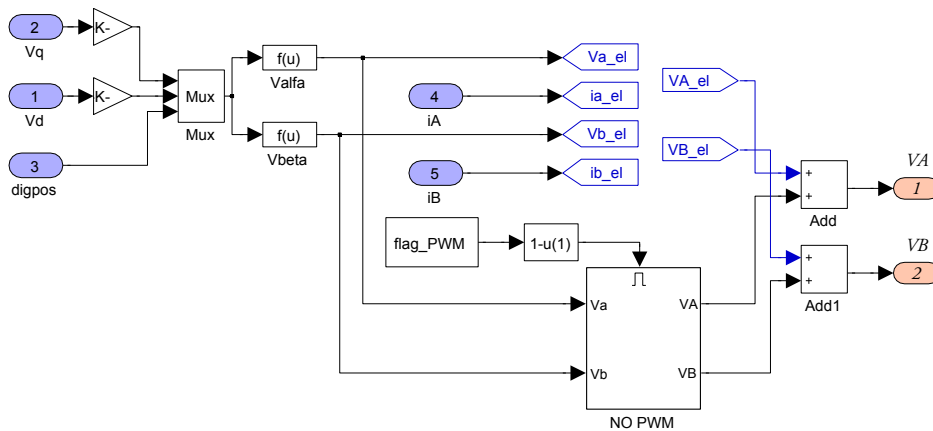


Figura 4.6. Blocco *qd-PWM*, che antitrasforma i segnali ed applica, eventualmente, la PWM, i segnali in ingresso ed in uscita ai blocchi *go-to* sono collegati al blocco *PWM*.

4.3. PWM

Il motore è collegato al generatore di tensione tramite due ponti ad H, uno per avvolgimento. Il ponte ad H è un circuito che permette di applicare una differenza di potenziale ad un carico, in questo caso, l'avvolgimento del motore in entrambe le direzioni. Questo viene reso possibile tramite la opportuna commutazione degli *switch* che lo compongono, che nel sensore in esame sono costituiti da degli IGBT, associati a dei diodi.

Poiché la differenza di potenziale che può essere applicata al carico è, in modulo, costante, è necessario utilizzare lo schema di modulazione della tensione noto come *Pulse Width Modulation*, PWM. Esso prevede di applicare al motore ad intervalli di tempo regolari una serie di impulsi di tensione, di intensità costante, tali per cui il valore medio della tensione applicata su ogni intervallo sia pari a quello della tensione che si vuole modulare.

Le tecniche di modulazione possibili non sono uniche ma sono invece possibili diverse tipologie, che possono essere distinte attraverso alcuni parametri.

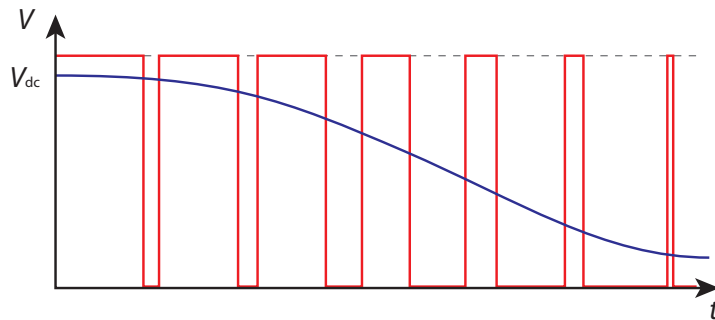


Figura 4.7. Andamento della tensione da modulare, in blu, e quella modulata, in rosso.

Una prima differenziazione è possibile farla a seconda di come la differenza di potenziale è applicata al carico. Si può infatti applicare una possibile polarità oppure zero in un ciclo di PWM, tale tecnica di modulazione è nota come unipolare o *soft switching*. In alternativa la tecnica di modulazione nota come bipolare, o *hard switching*, prevede di applicare al carico entrambe le polarità.

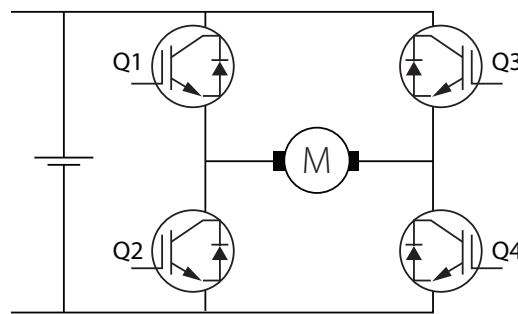


Figura 4.8. Ponte ad H, composto da quattro *switch* commutabili.

Un'altra differenziazione possibile riguarda la commutazione degli *switch* opposti, Q1–Q2 e Q3–Q4, essi possono infatti essere commutati in modo indipendente, oppure in modo complementare, dove se uno è attivato, l'altro non lo è. Queste tecniche si possono combinare insieme, ottenendo quindi quattro possibili combinazioni, come visualizzato in Tabella 4.3.

Tabella 4.3. Possibili tecniche di modulazione PWM.

	Bipolare	Unipolare
Indipendente	PWM Indipendente Bipolare	PWM Indipendente Unipolare
Complementare	PWM Complementare Bipolare	PWM Complementare Unipolare

Si andranno ora a descrivere ciascuna delle quattro tipologie.

4.3.1. PWM indipendente bipolare

L'andamento della modulazione indipendente bipolare è rappresentato in Figura 4.9.

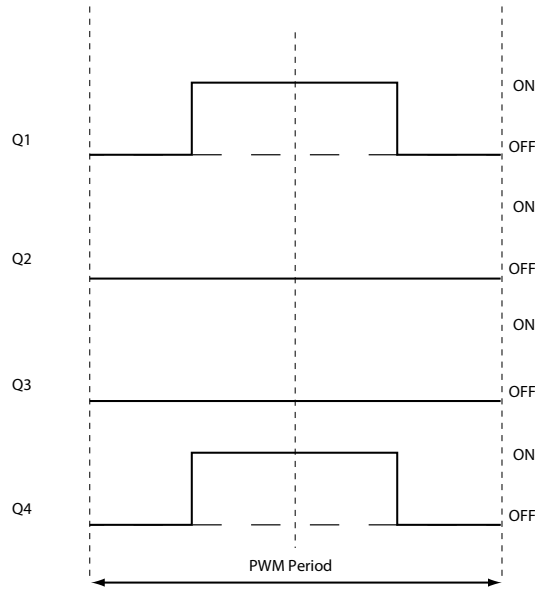


Figura 4.9. Andamento dei segnali di commutazione degli *switch* nel periodo del PWM per la modulazione indipendente e bipolare.

E' possibile distinguere una prima fase in cui i *transistor* Q1 e Q4 sono attivati. Il carico si trova quindi connesso alla polarità positiva del generatore di tensione e cresce la corrente sul motore, che come già detto ha un comportamento resistivo-induttivo.

Nella seconda fase i *transistor* vengono spenti, la corrente decresce continuando ad avere lo stesso verso ed il carico è connesso con la polarità negativa del generatore, Figura 4.10, secondo riquadro. Se la corrente circolante diventa nulla il motore si trova ad essere disconnesso dal generatore di tensione.

Per muovere il motore in direzione opposta si deve invertire il segnale di pilotaggio tra i *transistor* Q1 e Q3 e tra Q4 e Q2. Tale considerazione è valida anche per le modalità di modulazione che verranno descritte successivamente.

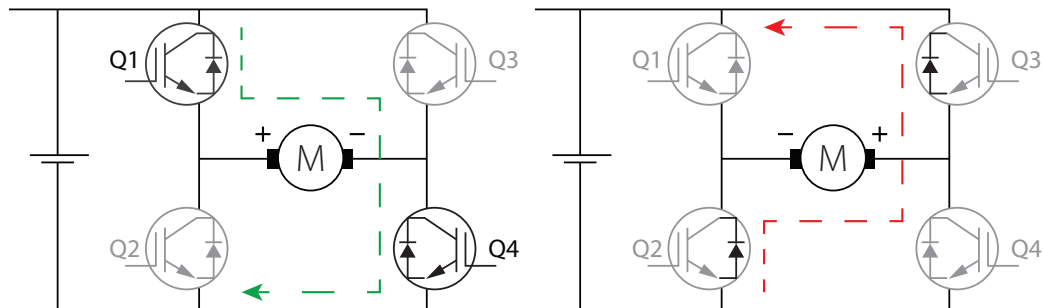


Figura 4.10. Fase uno, a sinistra, e due, a destra, della modulazione indipendente bipolare.

4.3.2. PWM indipendente unipolare

L'andamento della modulazione indipendente unipolare è rappresentato in Figura 4.11.

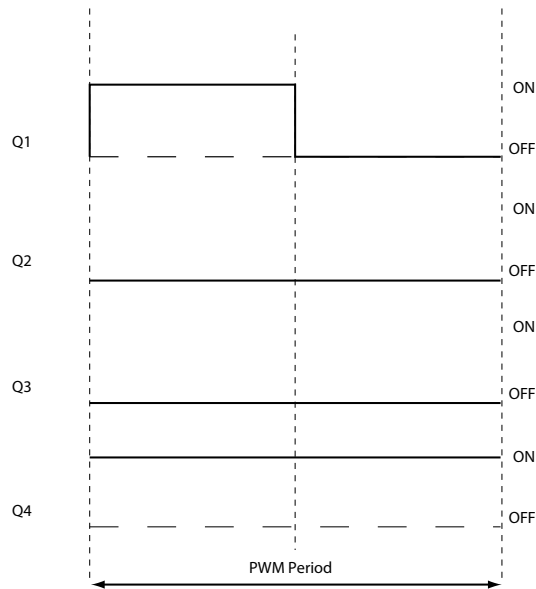


Figura 4.11. Andamento dei segnali di commutazione degli *switch* nel periodo del PWM per la modulazione indipendente unipolare.

La prima fase, Figura 4.12, primo riquadro, in cui i *transistor* Q1 e Q4 sono attivati, è sostanzialmente identica a quella già vista per il PWM indipendente bipolare, il carico si trova quindi connesso con la polarità positiva del generatore di tensione e cresce la corrente. Nella seconda fase invece il *transistor* Q1 viene spento ma il Q4 rimane acceso; la corrente, che continua ad avere lo stesso verso, fluisce tuttavia attraverso il diodo Q2.

Il motore si trova, conseguentemente, ad essere sottoposto ad una differenza di potenziale nulla.

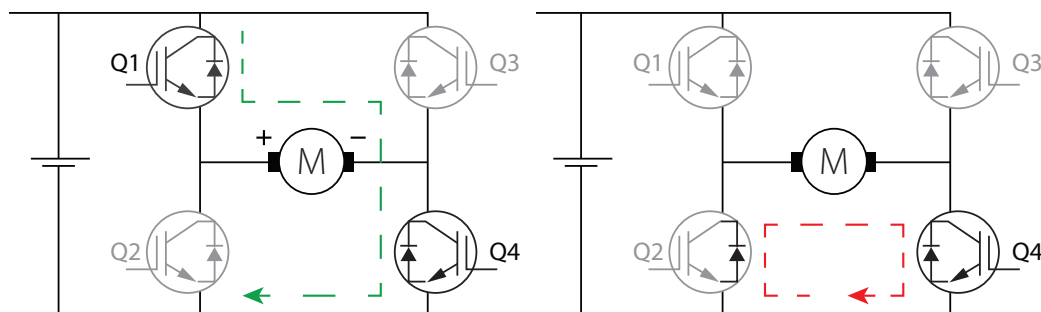


Figura 4.12. Fase uno, a sinistra, e due, a destra, della modulazione indipendente unipolare.

4.3.3. PWM complementare bipolare

L'andamento della modulazione complementare bipolare è rappresentato in Figura 4.13.

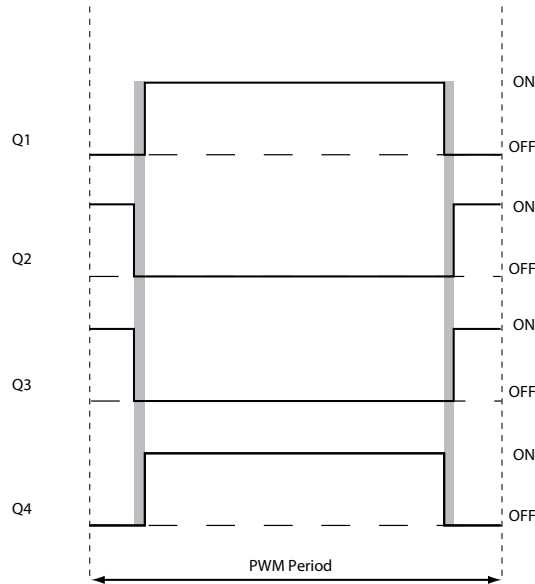


Figura 4.13. Andamento dei segnali di commutazione degli *switch* nel periodo del PWM per la modulazione complementare bipolare, in grigio i tempi morti.

Poiché la modulazione è complementare, è possibile incorrere nel fenomeno del *cross conduction*, in cui i *transistor* di uno stesso ramo sono attivati. Si devono introdurre quindi dei tempi morti tra la commutazione di due *transistor* complementari.

Nella prima fase in cui i *transistor* Q1 e Q4 sono attivati, il carico si trova connesso con la polarità positiva e la corrente cresce. Nella seconda fase invece tutti i *transistor* commutano e si trovano quindi ad essere attivati i Q2 e Q4.

Il motore si trova quindi ad essere sottoposto alla polarità negativa. La corrente decresce, mantenendo inizialmente lo stesso verso. Può tuttavia diventare anche negativa, differentemente dalla modulazione indipendente bipolare. In tal caso il motore si trova ad essere frenato attivamente dal *driver*.

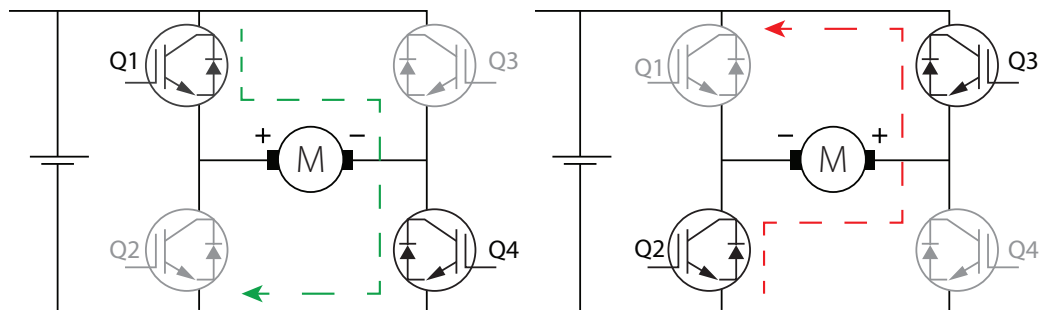


Figura 4.14. Fase uno, a sinistra, e due, a destra, della modulazione complementare bipolare.

4.3.4. PWM complementare unipolare

L'andamento della modulazione complementare unipolare è rappresentato in Figura 4.15.

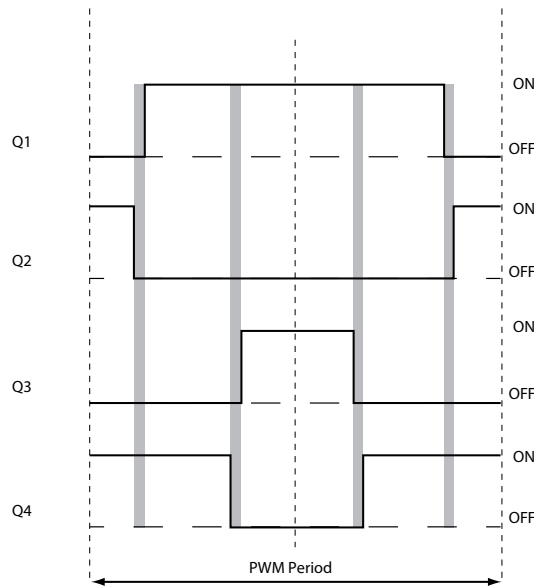


Figura 4.15. Andamento dei segnali di commutazione degli *switch* nel periodo del PWM per la modulazione complementare unipolare, in grigio i tempi morti.

Anche in questo caso, per i motivi già descritti, tra le commutazioni intercorre un tempo morto. Differentemente dai casi precedenti il periodo può essere diviso in tre fasi.

La prima fase prevede che i *transistor* Q2 e Q4 siano attivati; la corrente fluisce quindi attraverso il *transistor* Q4 ed il diodo Q2 ed il motore è sottoposto ad una differenza di potenziale nulla.

Commutando i *transistor* Q1 e Q2, il motore si trova ad essere sottoposto alla polarità positiva del generatore e la corrente conseguentemente cresce.

Nella terza fase sono invece i *transistor* Q3 e Q4 a commutare e, similamente alla prima fase, il motore si trova sottoposto ad una differenza di potenziale nulla.

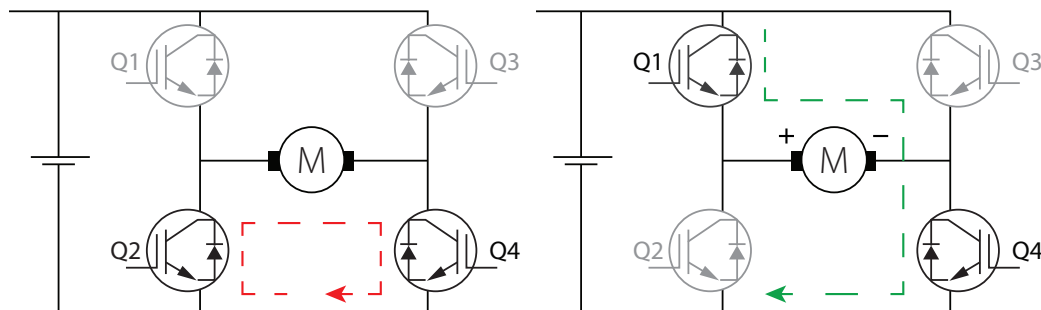


Figura 4.16. Fase uno, a sinistra, e due, a destra, della modulazione complementare unipolare.

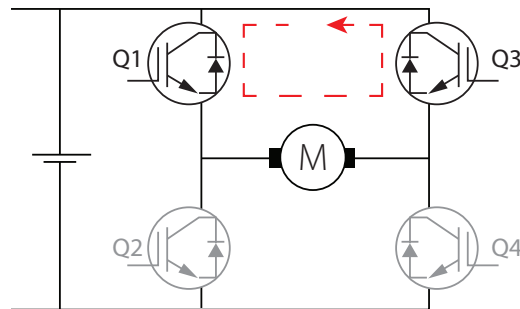


Figura 4.17. Fase tre della modulazione complementare unipolare.

4.3.5. Modellazione *Simulink*

La modulazione PWM ha in genere frequenze molto elevate, potrebbe quindi essere preferibile avere una simulazione che non la consideri ma che sia maggiormente veloce, si deve infatti considerare che ad ogni periodo di PWM devono venire eseguite delle istruzioni per il pilotaggio degli *switch*

Per tale motivo sono stati introdotti due blocchi, uno in cui la modulazione PWM viene simulata, che verrà descritto successivamente, ed uno in cui invece le tensioni vengono date in ingresso al motore senza alcuna variazione.

L'esecuzione dei due blocchi è comandata da un opportuno valore di ingresso che può essere 1 o 0, a seconda che il blocco debba essere eseguito o meno.

Si è fatto in modo che il valore in ingresso fosse opposto per i due blocchi, andando quindi ad attivarne uno e spegnere l'altro a seconda che sia necessario fare una simulazione più precisa o più veloce.

Mentre il blocco che lascia invariate le tensioni è replicato per *azimuth* ed *elevation*, per ragioni legate al funzionamento di *Simulink** si è dovuto riunire la schematizzazione della modulazione PWM in un unico blocco per entrambi due gradi di libertà.

* Un blocco comandato costituisce per *Simulink* un *Nonvirtual subsystem*, come viene evidenziato dal bordo spesso del blocco. I componenti con cui sono stati schematizzati i quattro ponti ad H, appartenenti alla libreria *SimPowerSystems*, necessitano della presenza di un blocco, unico per tutto il modello, chiamato *powergui*. Tale blocco non riesce a comunicare con gli altri componenti, se essi si trovano in un *Nonvirtual subsystem*, a meno che anche esso non appartenga al sottosistema stesso. Per tale motivo si è dovuto creare un unico blocco comandato per tutti gli schemi PWM, avente all'interno il blocco *powergui*.

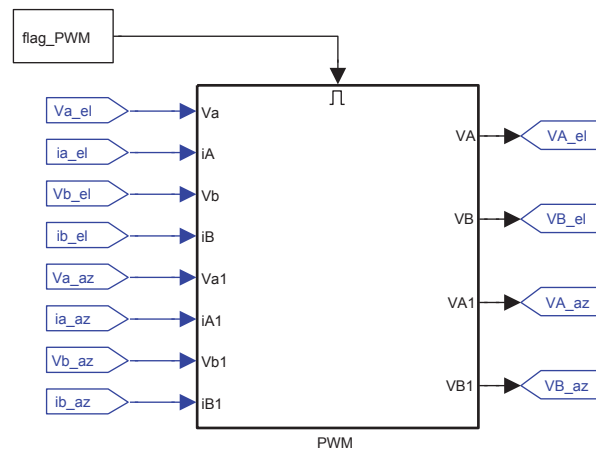


Figura 4.18. Blocco *PWM*, i blocchi *go-to* sono connessi ai segnali dei due blocchi *qd-PWM*, vedi Figura 4.6 a pagina 53.

Al fine di garantire la maggiore flessibilità possibile al modello, ed anche per avere una simulazione corretta del funzionamento si è scelto di simulare in modo circuitale il ponte ad H. Questo è reso possibile dalla libreria *SimPowerSystems*, i cui blocchi simulano alcuni componenti elettronici e possono comporre uno schema elettrico.

Si sono simulati quattro circuiti, corrispondenti ai quattro ponti ad H presenti nel sensore, due per ogni grado di libertà, pari al numero di avvolgimenti.

I circuiti sono tra loro identici, differendo solo per la tensione da modulare e, tra *azimuth* ed *elevation*, per i valori di alimentazione. Un esempio di uno di questi quattro circuiti è riportato in Figura 4.19 a pagina 61.

Il segnale di tensione da modulare, campionato ad ogni periodo di PWM entra in una *S-Function*. Tale funzione implementa le tecniche di PWM viste precedentemente, restituendo in uscita l'opportuno segnale di pilotaggio per ciascuno dei quattro *transistor* che compongono il circuito.

La differenza di potenziale nell'opportuno ramo del ponte, dove deve essere presente il carico, viene letta e convertita da un opportuno blocco in un segnale *Simulink* tradizionale. Tale segnale arriva, attraverso vari blocchi, al blocco che simula il motore, vedere Figura 4.3 a pagina 50. In tale blocco viene quindi calcolata la corrente circolante nell'avvolgimento, che viene imposta al circuito, grazie ad un blocco di generazione di corrente, pilotato in ampiezza dal segnale *Simulink* della corrente stessa.

Tale modalità di simulazione ha consentito di evitare la costruzione di un modello circuitale del motore, adottando invece la schematizzazione a blocchi precedentemente descritta, che risulta più facilmente modificabile.

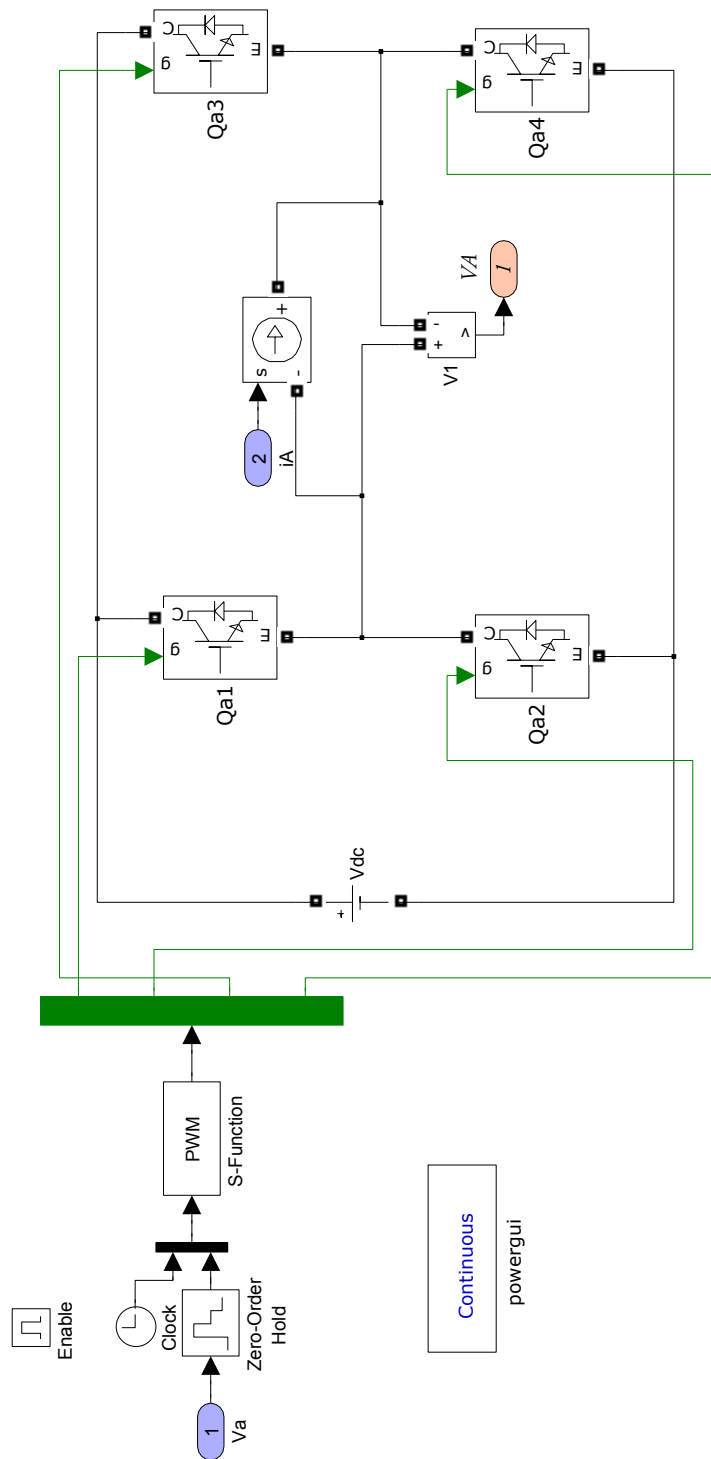


Figura 4.19. Schema di uno dei quattro ponti ad H simulati nel blocco *PWM*. Il segnale di pilotaggio degli IGBT è verde per distinguerlo dal ponte ad H che non rappresenta un ramo di uno schema a blocchi ma un circuito elettrico vero e proprio.

Capitolo 5

Prove sperimentali sul sensore IRST

Il modello elettrico del motore che è stato implementato nello schema *Simulink*, è chiaramente un modello ideale, che non tiene conto di nessuna forma di disturbo.

Dovranno invece essere modellati alcune forme di disturbo connaturate alla tipologia di motore elettrico utilizzato.

Al fine di determinare tali importanti caratteristiche del sistema si sono quindi organizzate delle prove sperimentali su un sensore IRST opportunamente strumentato, identico a quello che si andrà a simulare, con l'eccezione che quello usato da banco prove sperimentali è controllato tramite un controllore PID analogico, diversamente da quello simulato che implementerà invece una soluzione digitale.

Oltre a prove per identificare eventuali disturbi di coppia, sono state eseguite anche delle prove di risposta del sistema per caratterizzarne il comportamento

Il meccanismo è comandato da un segnale di posizione per entrambi i gradi di libertà ed è strumentato al fine di rendere possibile l'acquisizione della sua posizione effettiva. Sono stati quindi registrati dal calcolatore sia i dati relativi al *set-point* che quelli relativi alla posizione. È importante notare come, volendo andare a caratterizzare il motore, ci si sia riferito alla posizione dello specchio e non della linea di vista, quindi gli angoli di *elevation* sono dimezzati rispetto ai valori relativi alla LOF.

A causa della tipologia di impianto di prova non è stato possibile effettuare delle prove a coppia costante né misurare la corrente circolante nel motore.

Nelle sezioni successive verranno descritte le prove che sono state effettuate e verrà mostrato il loro andamento, in termini macroscopici. Un'analisi dettagliata dei risultati verrà invece affrontata nel capitolo successivo.

5.1. Prove previste

Le prove che si è previsto di effettuare sul servomeccanismo sono le seguenti:

- Risposte ad un segnale a scalino di diversa ampiezza.
- Prove a velocità costante, per differenti velocità.

5.1.1. Prove a scalino

Non è possibile per come è comandato il servomeccanismo eseguire degli scalini ideali, saranno invece eseguite delle rampe in cui l'intervallo di salita sarà pari a 0.5 ms. Tale intervallo corrisponde al tempo minimo di campionamento con cui è fornito il segnale di *set-point* al sensore.

Questo non rappresenta tuttavia un grave problema per gli scalini di ampiezza maggiore, mentre potrebbe essere più critico per gli scalini di ampiezza limitata.

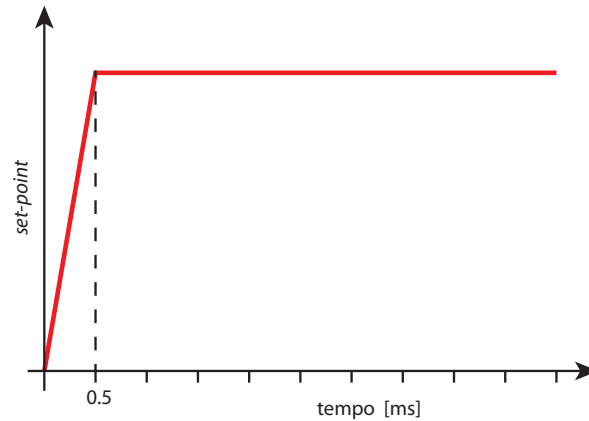


Figura 5.1. Andamento del tempo del *set-point* utilizzato per le prove di risposta a scalino, equivalente ad una rampa con tempo di salita pari ad un intervallo di campionamento.

Le prove di risposta a scalino possono essere divise in due tipologie con diverse finalità

- **Scalini di piccola ampiezza:** per determinare l'attrito di primo distacco.
- **Scalini di grande ampiezza:** determinare i parametri della risposta del sistema.

5.1.2. Prove a velocità costante

Per quanto riguarda le prove a velocità costante si deve invece considerare che entrambi i gradi di libertà hanno delle posizioni limite dovute alla configurazione meccanica del servomeccanismo.

Per tale motivo si sono eseguiti diversi tratti a velocità costante, uguale in modulo ma opposta in verso, intervallati da tratti di inversione del moto.

Per le prove si è preso in considerazione un *range* di velocità che ha come estremi $50^\circ/s$ e $400^\circ/s$.

Tutti i *set-point* qui visti vanno applicati singolarmente ai motori di entrambi gli assi per cui durante la prova di un motore l'altro grado di libertà, a cui non si sta applicando il *set-point* a rampa, subisce invece delle variazioni a gradino, come evidenziato in Figura 5.2 a pagina 65.

Si riporta ora una tabella riassuntiva delle prove che vanno eseguite:

Tabella 5.1. Prove da eseguire sul servomeccanismo.

Scalini di piccola ampiezza	Determinare l'attrito di primo distacco
Scalini di grande ampiezza	Caratterizzare la risposta del sistema
Tratti a velocità costante	Determinare il ripple di coppia

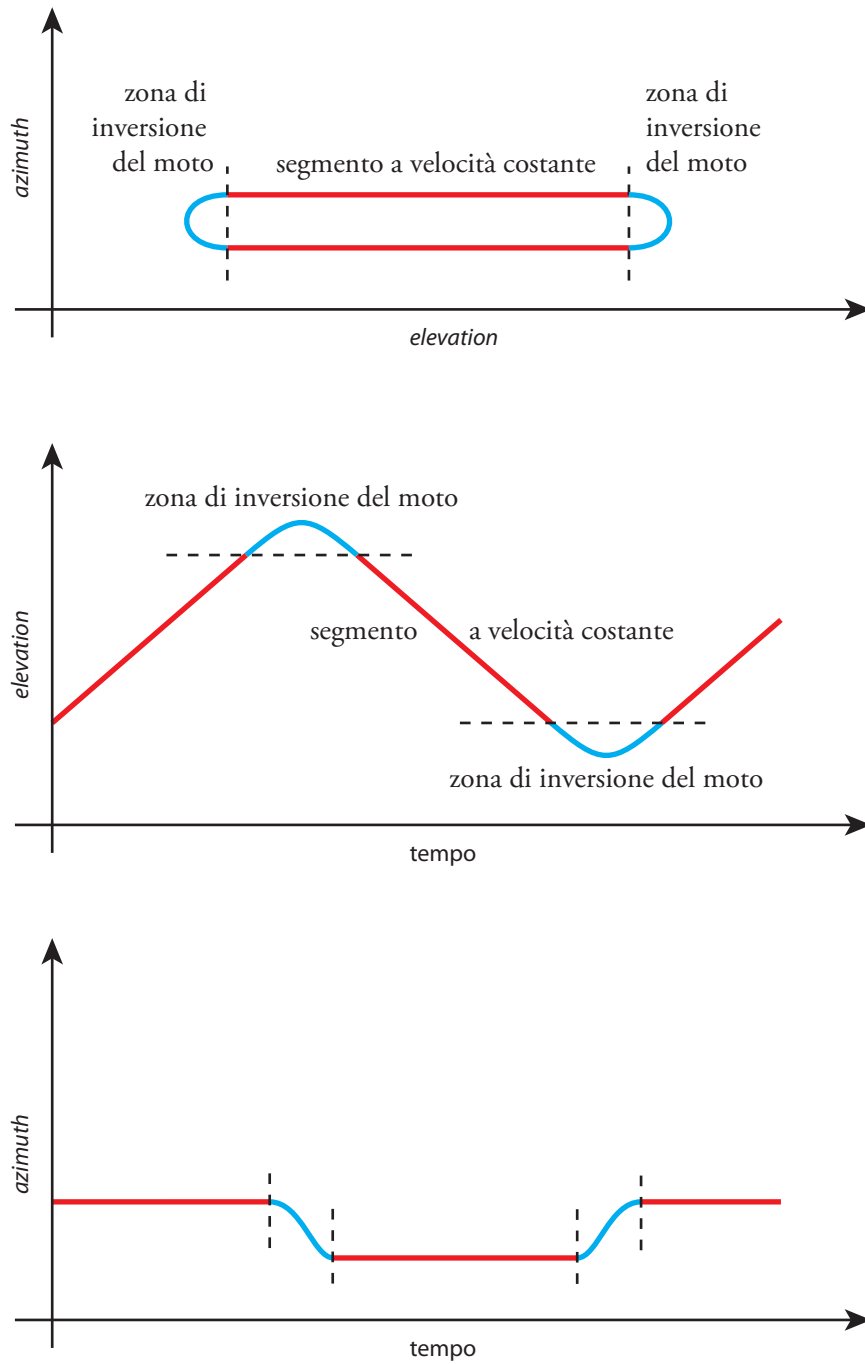


Figura 5.2. Rappresentazione dell'andamento dei gradi di libertà in una prova relativa all'asse di *elevation*. Nel caso di una prova per l'*azimuth*, gli assi *azimuth* ed *elevation* risulterebbero invertiti.

5.2. Esperimenti effettuati

5.2.1. Risposta a scalino

Sono state svolte delle prove di risposta a scalino. Per queste prove si è usato un tempo di campionamento di $\Delta t = 0.5$ ms ad esclusione dello scalino di intensità minore per cui si è usato un tempo di campionamento dimezzato, $\Delta t = 0.25$ ms.

Si riporta solamente l'andamento della grandezza sottoposta al *set-point* a scalino poiché all'altra viene imposto un riferimento nullo.

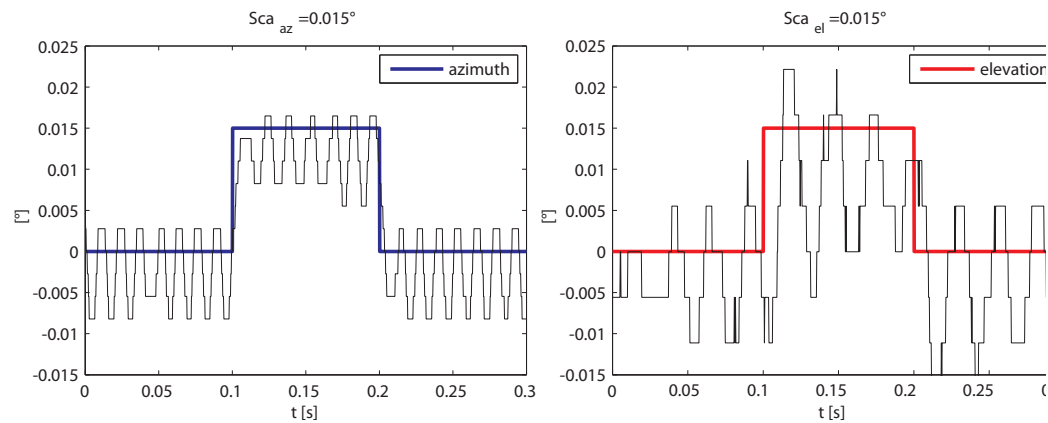


Figura 5.3. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 0.015° , per entrambi i gradi di libertà. Il segnale misurato è rappresentato in nero mentre il *set-point* è rappresentato dalle linee colorate. Si noti che la scale dei tempi per questa prova è differente rispetto a tutte le altre.

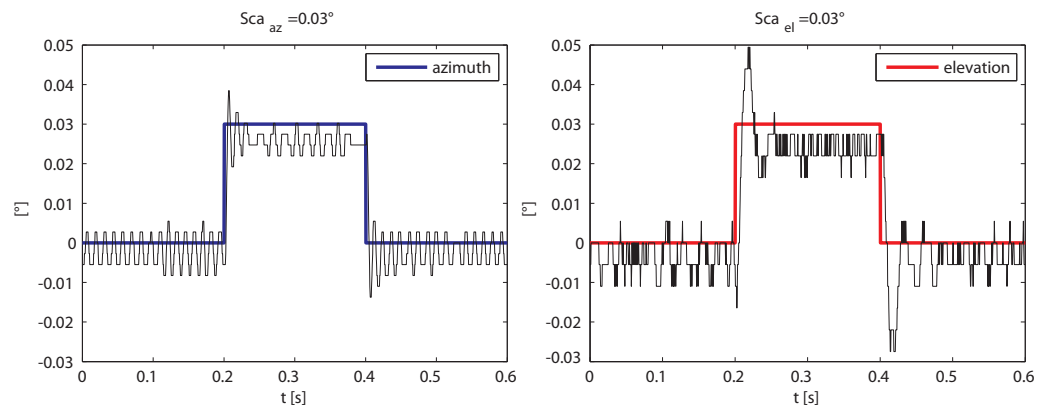


Figura 5.4. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 0.03° , per entrambi i gradi di libertà.

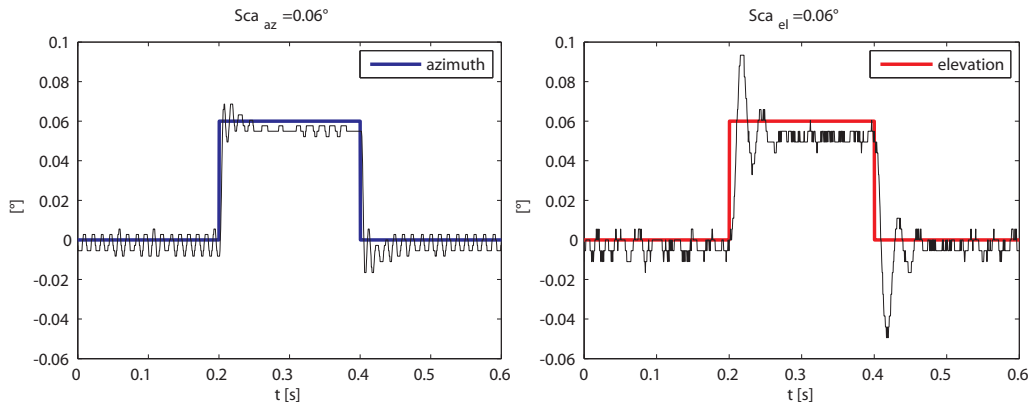


Figura 5.5. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 0.06° , per entrambi i gradi di libert .

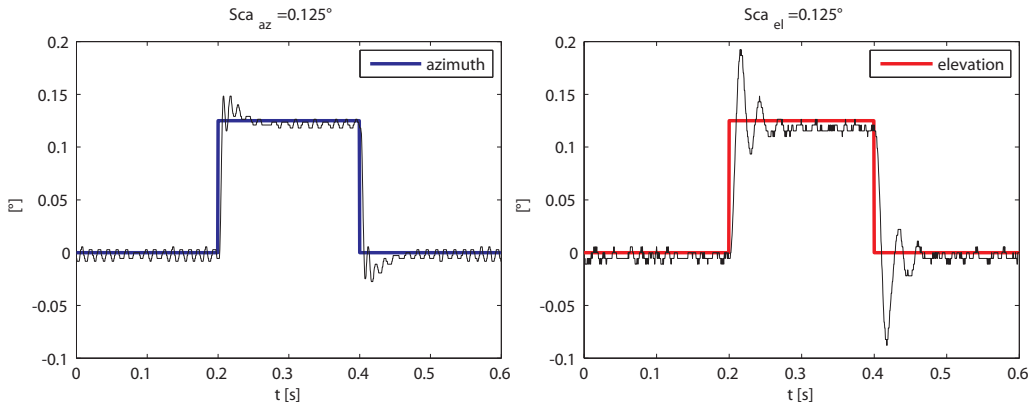


Figura 5.6. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 0.125° , per entrambi i gradi di libert .

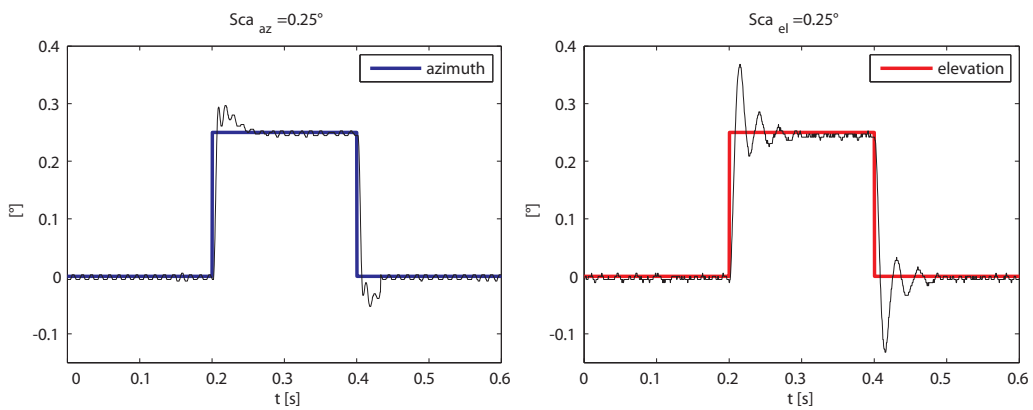


Figura 5.7. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 0.25° , per entrambi i gradi di libert .

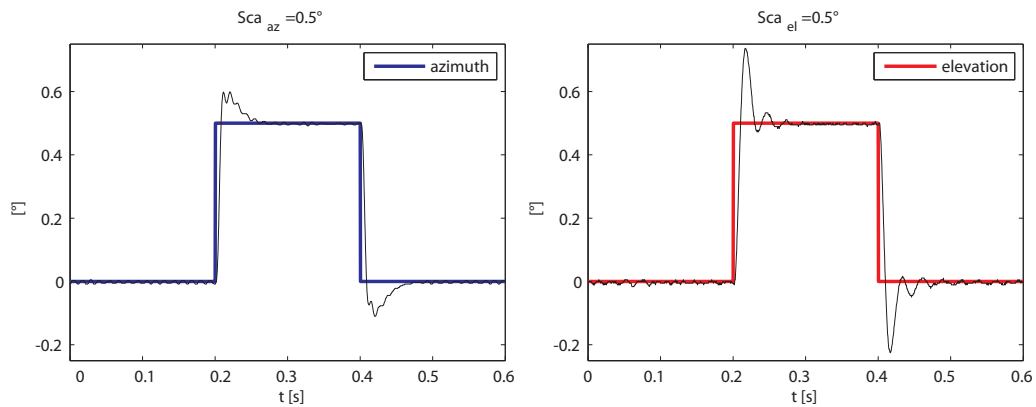


Figura 5.8. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 0.5° , per entrambi i gradi di libertà.

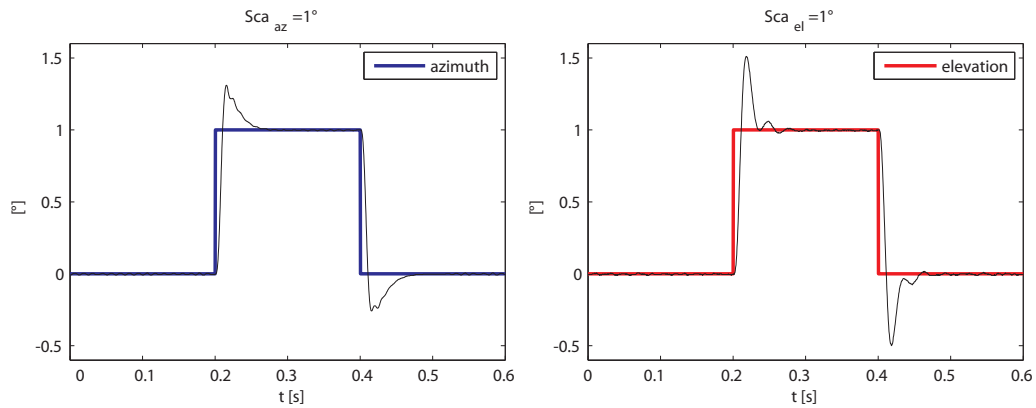


Figura 5.9. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 1° , per entrambi i gradi di libertà.

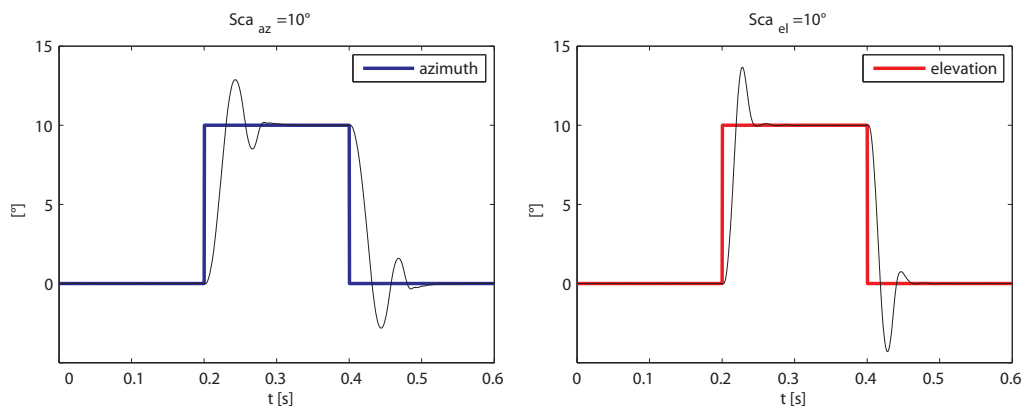


Figura 5.10. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 10° , per entrambi i gradi di libertà.

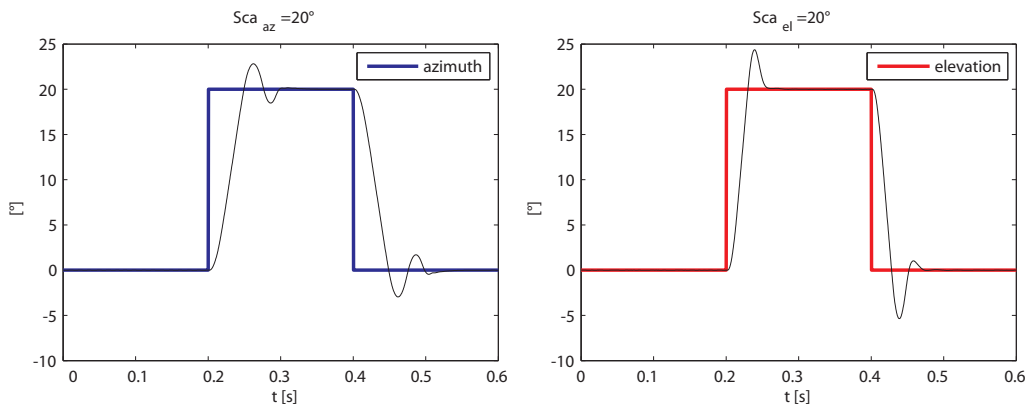


Figura 5.11. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 20° , per entrambi i gradi di libert .

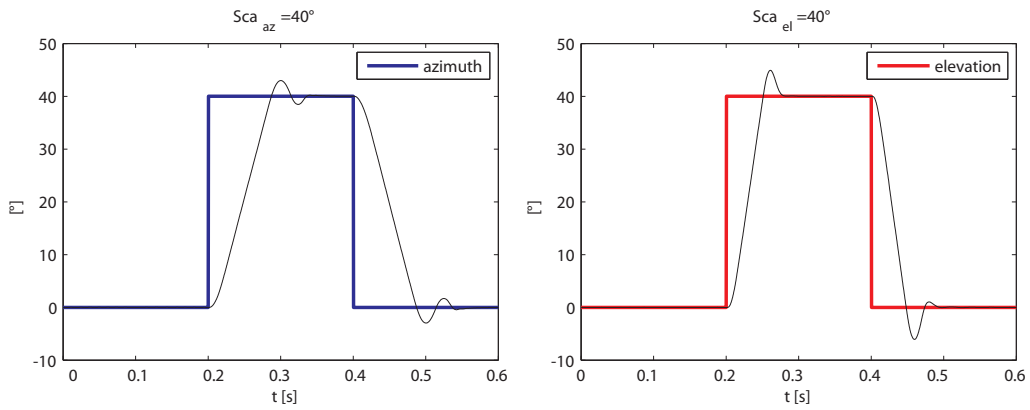


Figura 5.12. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 40° , per entrambi i gradi di libert .

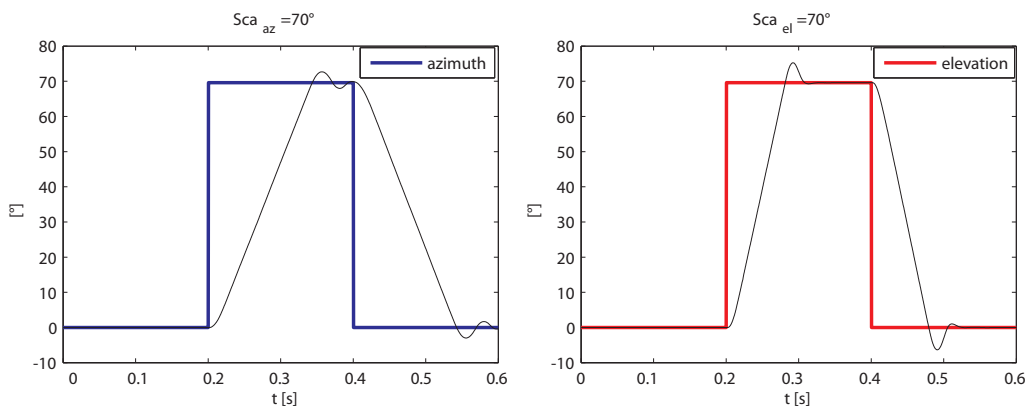


Figura 5.13. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 70° , per entrambi i gradi di libert .

5.2.2. Prove a velocità costante, $\Delta t = 0.5$ ms

Si sono inizialmente eseguite le prove relative ai segmenti a velocità costante. Le prove sono state eseguite alle seguenti velocità:

Tabella 5.2. Velocità eseguite nei diversi esperimenti.

50°/s	100°/s	150°/s	200°/s	400°/s
-------	--------	--------	--------	--------

Tutte le misure sono state eseguite con un intervallo di campionamento pari a 0.5 ms.

Si riporta nello stesso grafico l'andamento dei due segnali di posizione, *azimuth* ed *elevation*, insieme al *set-point* per il primo ciclo. Si presti attenzione al fatto che la scala temporale non è uniforme tra i grafici, per una migliore lettura degli stessi

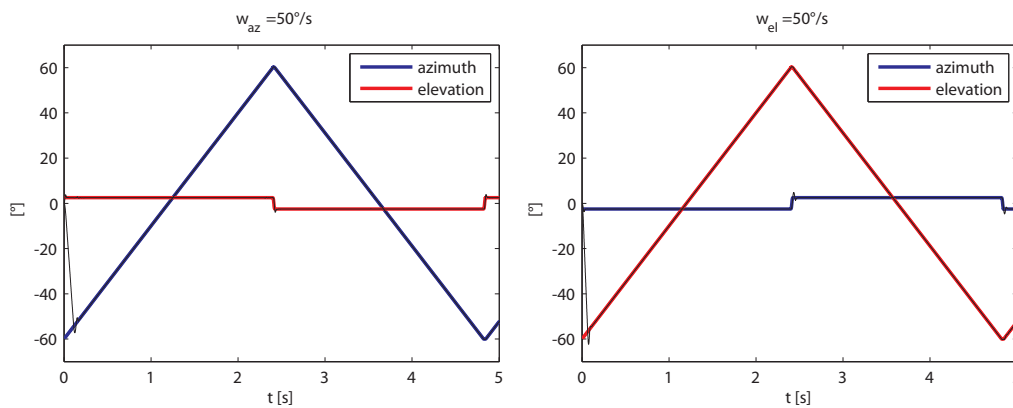


Figura 5.14. Andamento nel tempo del *set-point* e del segnale misurato per le prove a velocità costante, $\omega = 50^\circ/s$, per entrambi i gradi di libertà, $\Delta t = 0.5$ ms. Il segnale misurato è rappresentato in nero mentre il *set-point* è rappresentato dalle linee colorate.

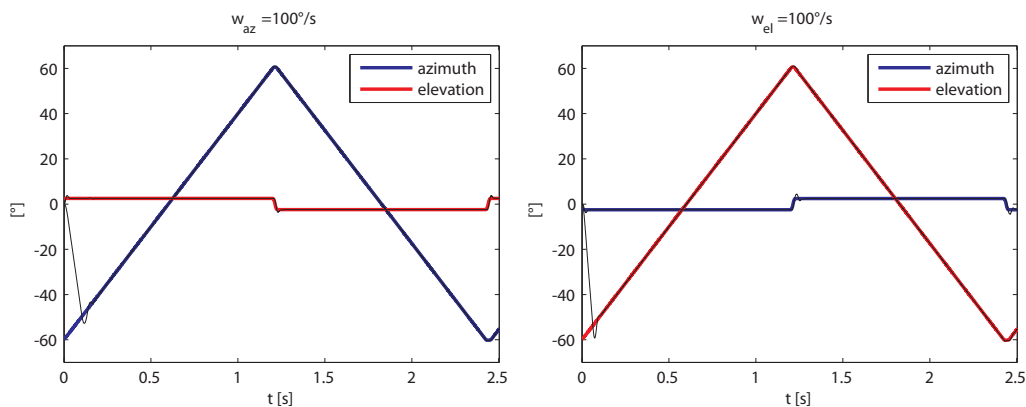


Figura 5.15. Andamento nel tempo del *set-point* e del segnale misurato per le prove a velocità costante, $\omega = 100^\circ/s$, per entrambi i gradi di libertà, $\Delta t = 0.5$ ms.

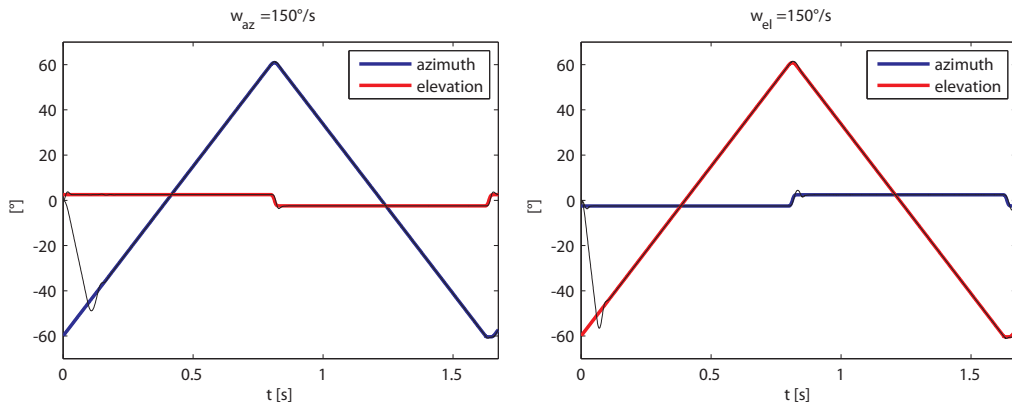


Figura 5.16. Andamento nel tempo del *set-point* e del segnale misurato per le prove a velocità costante, $\omega = 150^\circ/s$, per entrambi i gradi di libertà, $\Delta t = 0.5$ ms.

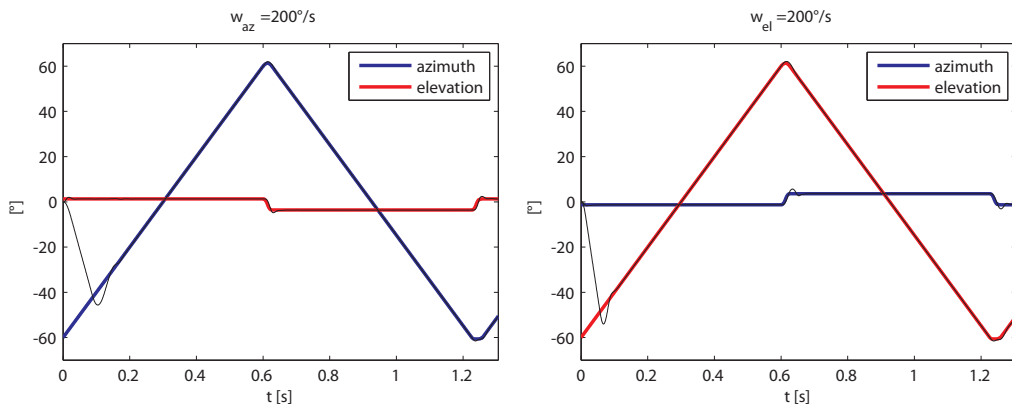


Figura 5.17. Andamento nel tempo del *set-point* e del segnale misurato per le prove a velocità costante, $\omega = 200^\circ/s$, per entrambi i gradi di libertà, $\Delta t = 0.5$ ms.

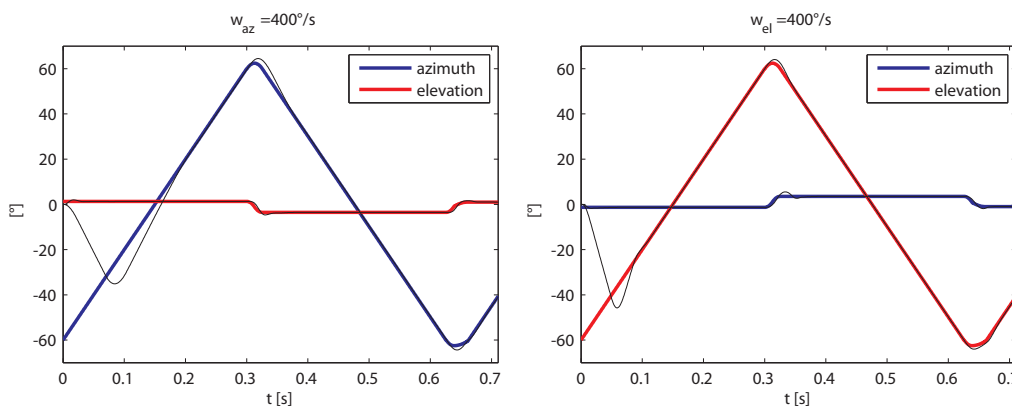


Figura 5.18. Andamento nel tempo del *set-point* e del segnale misurato per le prove a velocità costante, $\omega = 400^\circ/s$, per entrambi i gradi di libertà, $\Delta t = 0.5$ ms.

5.2.3. Prove a velocità costante, $\Delta t = 0.25$ ms

Oltre alle prove precedenti sono anche eseguiti degli esperimenti con un intervallo di campionamento minore, pari a 0.25 ms.

Si sono eseguite le prove per le stesse velocità analizzate in precedenza, ad esclusione di quella a $150^\circ/s$ che non è stata eseguita e di una prova addizionale svolta a $800^\circ/s$. Si presti attenzione al fatto che la scala temporale non è uniforme tra i grafici, per una migliore lettura degli stessi.

Dall'analisi dei dati si nota come la prova ad $800^\circ/s$ non possa essere considerata valida poichè il sensore non riesce a seguire in modo soddisfacente il *set-point*.

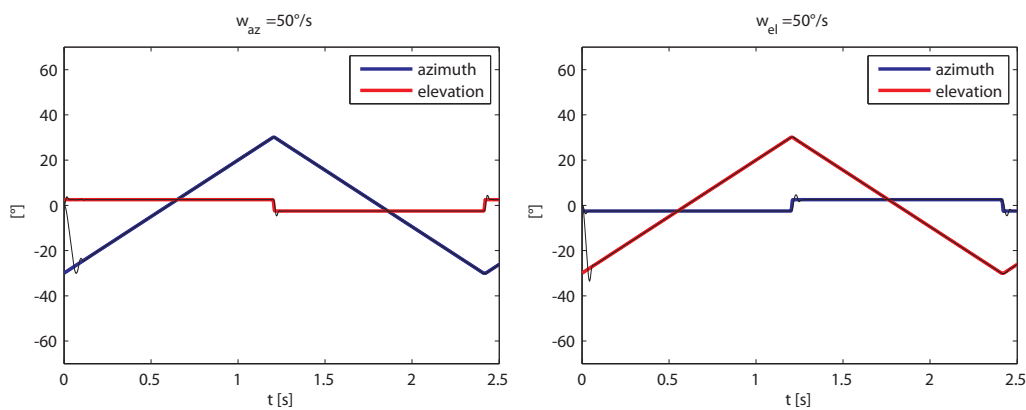


Figura 5.19. Andamento nel tempo del *set-point* e del segnale misurato per le prove a velocità costante, $\omega = 50^\circ/s$, per entrambi i gradi di libertà, $\Delta t = 0.25$ ms. Il segnale misurato è rappresentato in nero mentre il *set-point* è rappresentato dalle linee colorate.

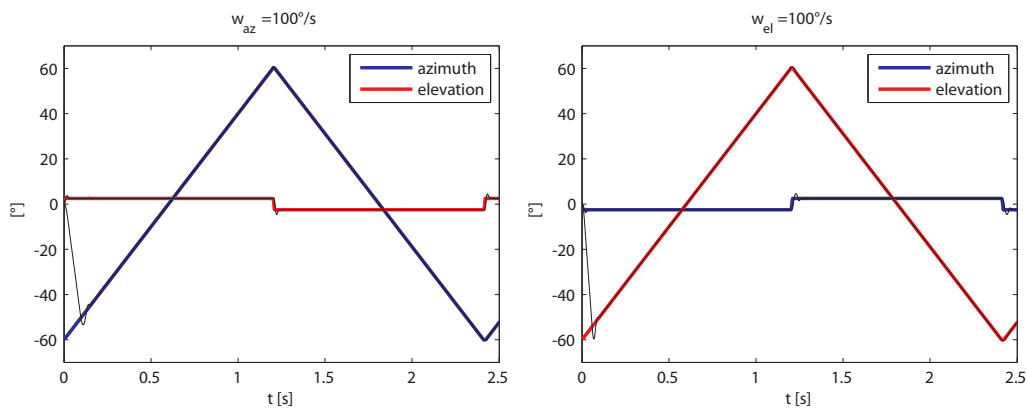


Figura 5.20. Andamento nel tempo del *set-point* e del segnale misurato per le prove a velocità costante, $\omega = 100^\circ/s$, per entrambi i gradi di libertà, $\Delta t = 0.25$ ms.

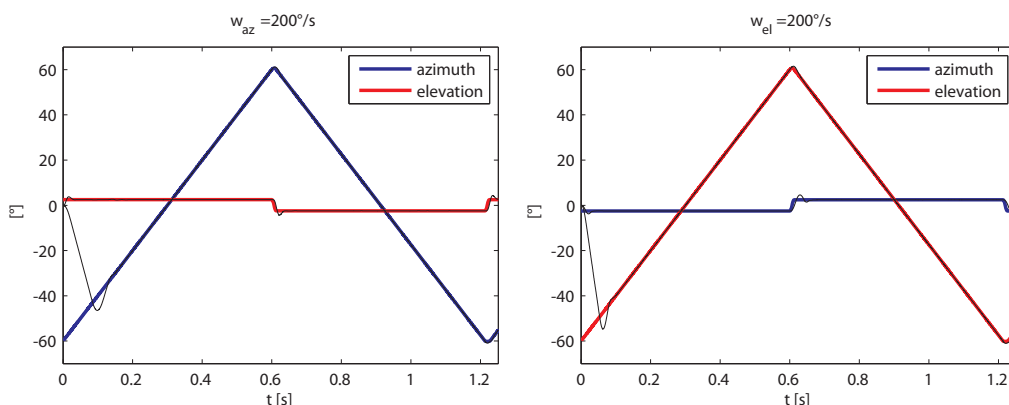


Figura 5.21. Andamento nel tempo del *set-point* e del segnale misurato per le prove a velocità costante, $\omega = 200^\circ/s$, per entrambi i gradi di libertà, $\Delta t = 0.25$ ms.

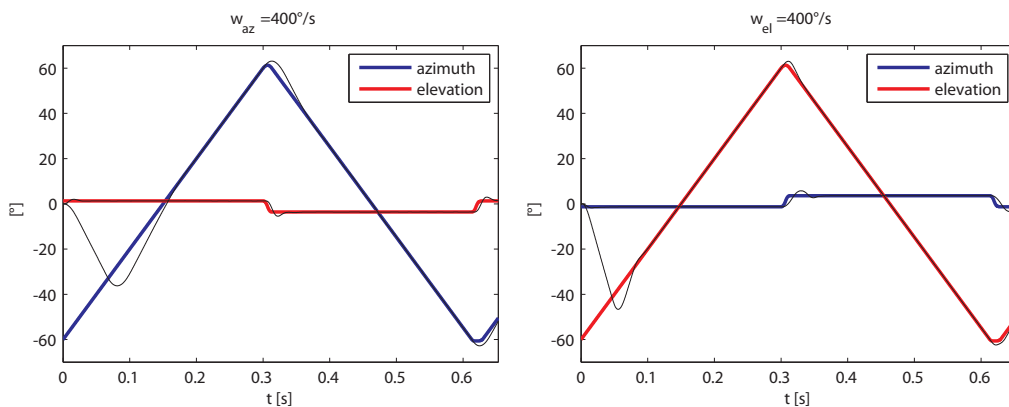


Figura 5.22. Andamento nel tempo del *set-point* e del segnale misurato per le prove a velocità costante, $\omega = 400^\circ/s$, per entrambi i gradi di libertà, $\Delta t = 0.25$ ms.

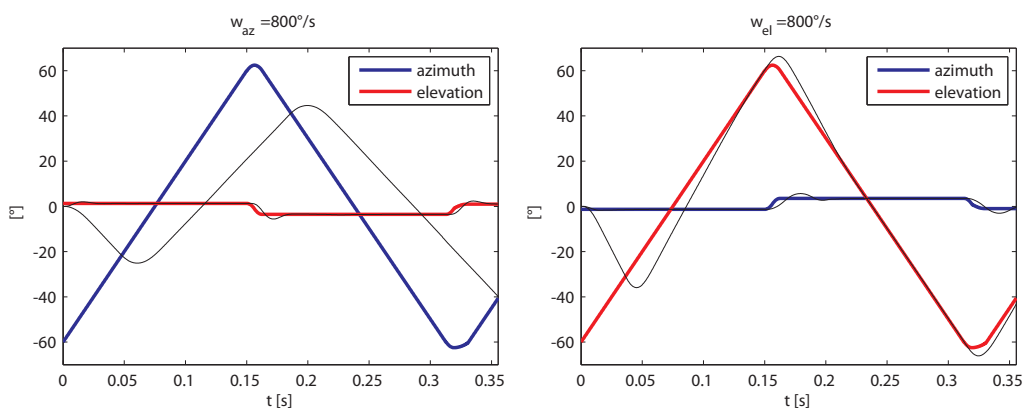


Figura 5.23. Andamento nel tempo del *set-point* e del segnale misurato per le prove a velocità costante, $\omega = 800^\circ/s$, per entrambi i gradi di libertà, $\Delta t = 0.25$ ms.

5.2.4. Rumore del segnale

Per caratterizzare il rumore di misura dei due sensori di posizione si è anche acquisito il segnale lasciando nullo il *set-point*, essendo impossibile disattivare il controllo sul sistema mantenendo attivi i sensori di misura.

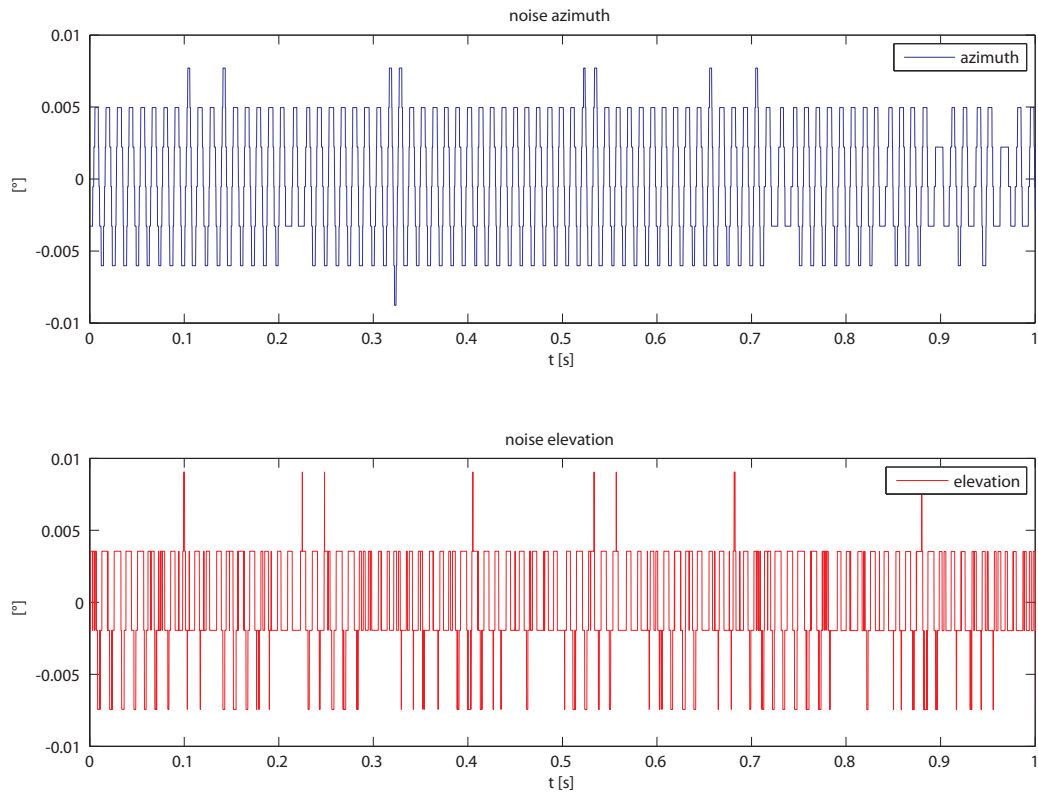


Figura 5.24. Andamento del segnale di posizione del sistema quando sottoposto ad un ingresso nullo. Il segnale può essere considerato come rumore della misura.

Capitolo 6

Individuazione e modellazione dei disturbi

In questo capitolo verranno analizzati i dati ottenuti dagli esperimenti descritti nel capitolo precedente.

In particolare ci si è focalizzati sull'analisi e l'elaborazione dei dati ottenuti dalle prove a rampe. Essi sono stati processati al fine di ottenere, attraverso una analisi in frequenza, le principali componenti di disturbo di coppia.

Infine si descriverà la modellazione in *Simulink* dei disturbi.

6.1. Analisi dei dati delle rampe di velocità

Come descritto anche nel capitolo precedente il segnale delle prove a velocità costante ha un andamento a rampa, alternato tuttavia a zone di inversione del moto.

Questo ha reso necessario l'individuazione corretta di opportuni intervalli nei quali la velocità fosse costante, che potessero essere analizzati.

L'individuazione di tali intervalli ha quindi reso possibile il *detrend* del segnale di posizione, la rimozione cioè della parte costante e di quella lineare, come rappresentato anche in Figura 6.1 a pagina 75. Tale procedura permette di ricavare quindi le variazioni rispetto ad una rampa ideale a velocità costante.

Si è ricavato tale segnale dal momento che si è preferito applicare un'analisi spettrale ad esso e non al segnale di velocità depurato del valore medio, poiché da una analisi del segnale di velocità sarebbero state evidenziate maggiormente le componenti ad alta frequenza, potendo invece rendere più difficoltoso identificare le componenti più lente dei disturbi.

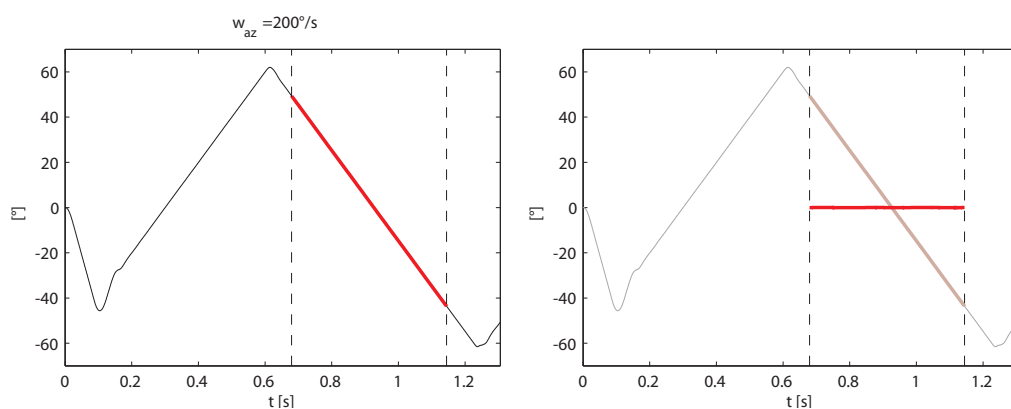


Figura 6.1. Identificazione di un segmento a velocità costante, riquadro sinistro, e successivo *detrend* del segnale stesso, riquadro a destra.

Si riporta ora nelle figure seguenti quanto si è ottenuto dal *detrend* dei segnali di posizione, prima per le prove attorno all'asse di *azimuth* poi per quelle attorno all'asse di *elevation*.

I grafici riportati sono relativi alle prove eseguite con $\Delta t = 0.5\text{ms}$, quelli con il tempo di campionamento dimezzato risultano qualitativamente uguali.

Alcune differenze, come si vedrà in seguito, sono emerse quando si è andati a visualizzare lo spettro in frequenza del segnale, in particolare però per le componenti a frequenza molto elevata.

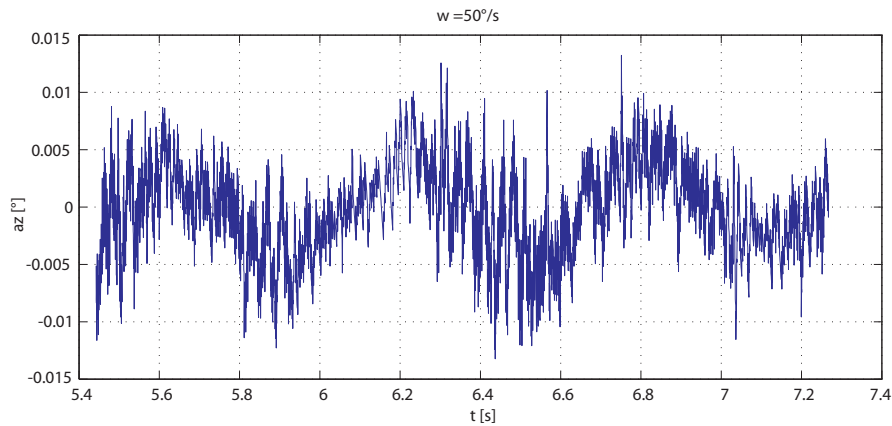


Figura 6.2. Andamento nel tempo del *detrend* di un intervallo del segnale di *azimuth*, per la prova a velocità costante, $\omega = 50^\circ/\text{s}$.

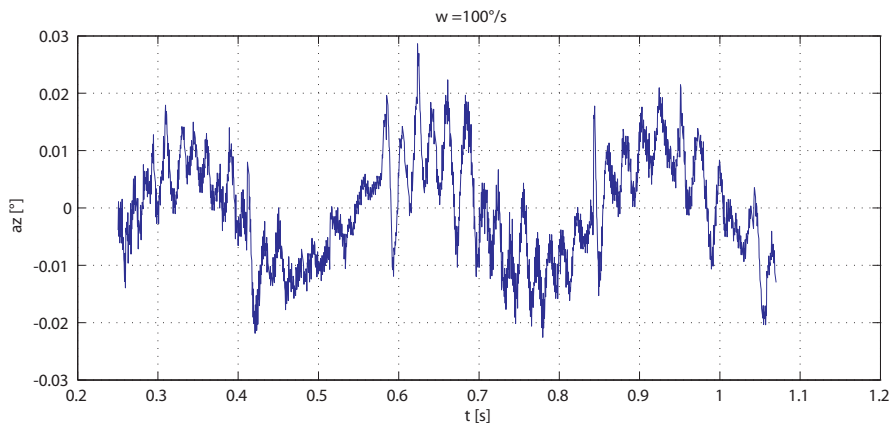


Figura 6.3. Andamento nel tempo del *detrend* di un intervallo del segnale di *azimuth*, per la prova a velocità costante, $\omega = 100^\circ/\text{s}$.

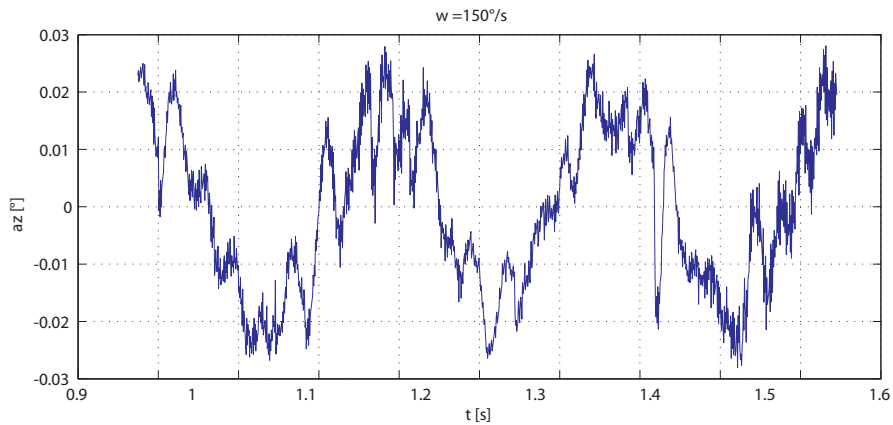


Figura 6.4. Andamento nel tempo del *detrend* di un intervallo del segnale di *azimuth*, per la prova a velocità costante, $\omega = 150^\circ/\text{s}$.

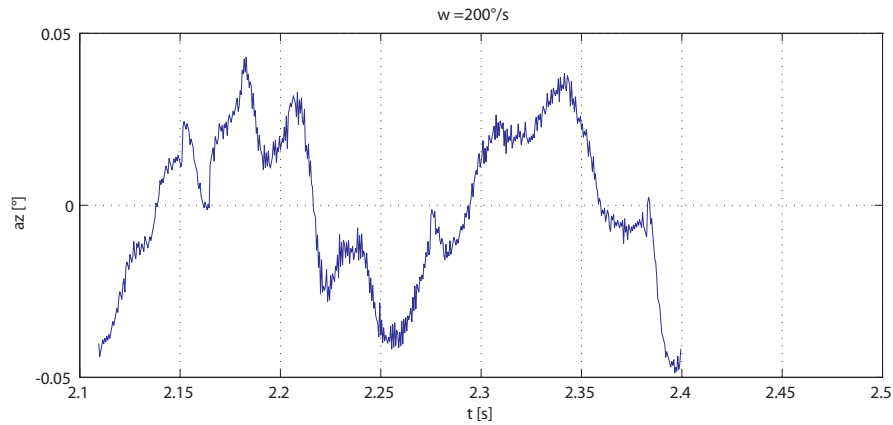


Figura 6.5. Andamento nel tempo del *detrend* di un intervallo del segnale di *azimuth*, per la prova a velocità costante, $\omega = 200^\circ/\text{s}$.

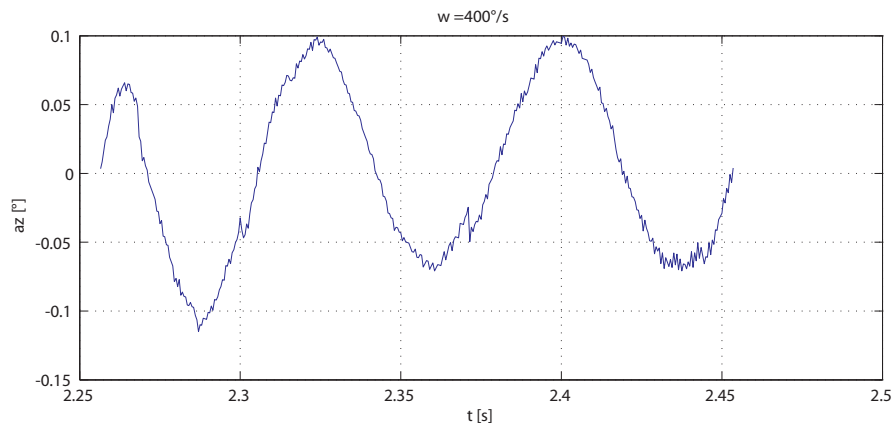


Figura 6.6. Andamento nel tempo del *detrend* di un intervallo del segnale di *azimuth*, per la prova a velocità costante, $\omega = 400^\circ/\text{s}$.

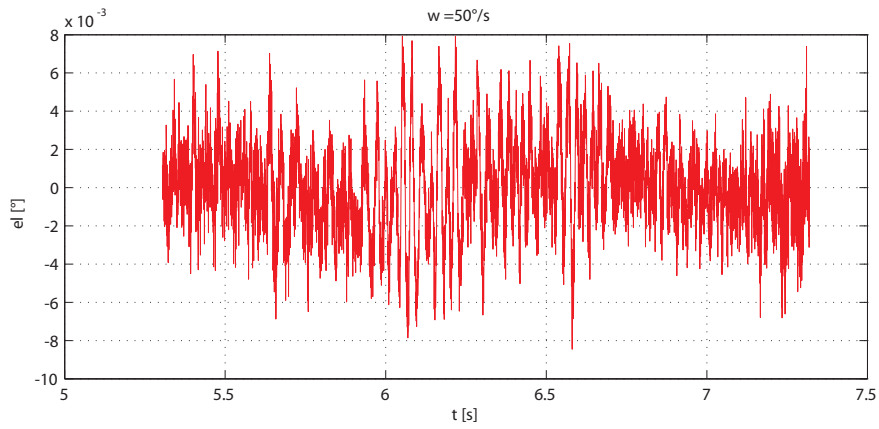


Figura 6.7. Andamento nel tempo del *detrend* di un intervallo del segnale di *elevation*, per la prova a velocità costante, $\omega = 50^\circ/\text{s}$.

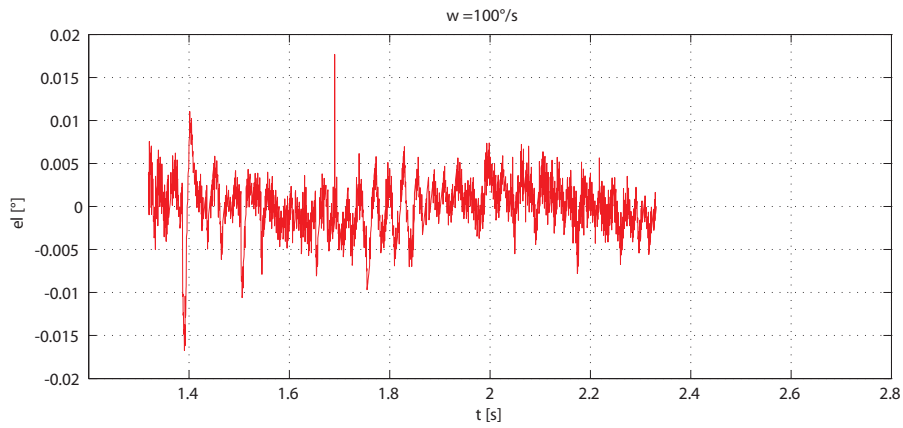


Figura 6.8. Andamento nel tempo del *detrend* di un intervallo del segnale di *elevation*, per la prova a velocità costante, $\omega = 100^\circ/\text{s}$.

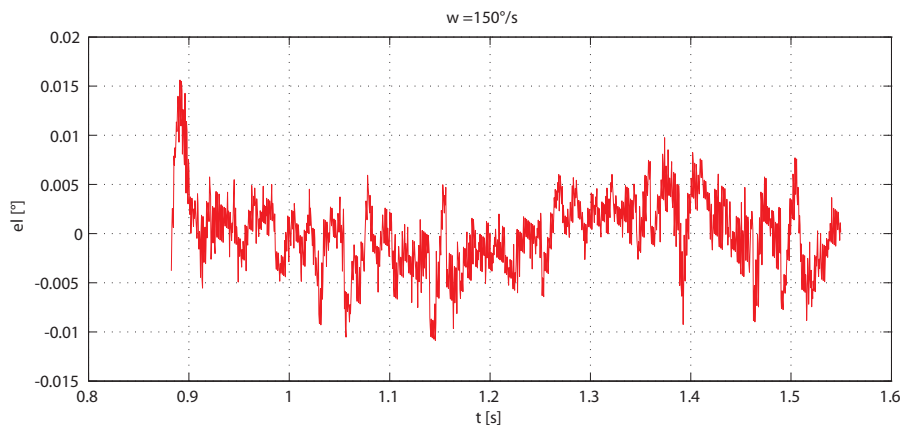


Figura 6.9. Andamento nel tempo del *detrend* di un intervallo del segnale di *elevation*, per la prova a velocità costante, $\omega = 150^\circ/\text{s}$.

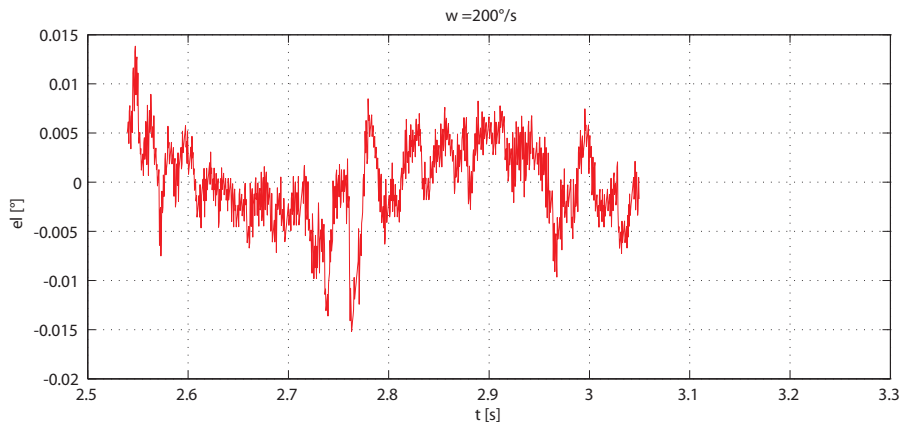


Figura 6.10. Andamento nel tempo del *detrend* di un intervallo del segnale di *elevation*, per la prova a velocità costante, $\omega = 200^\circ/\text{s}$.

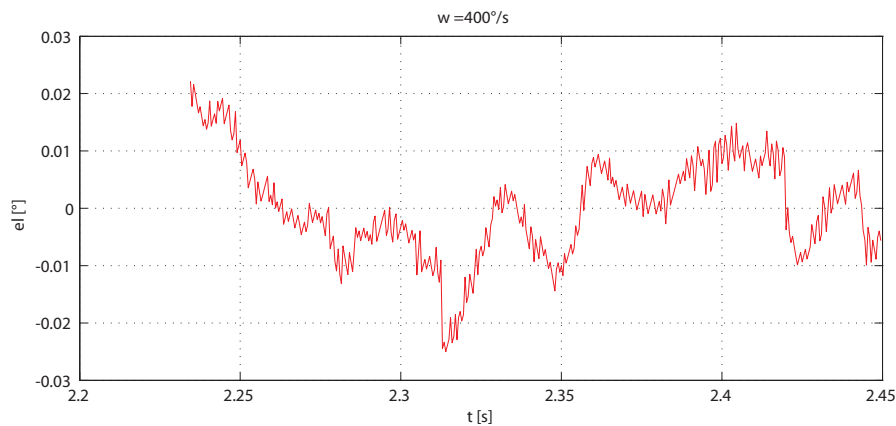


Figura 6.11. Andamento nel tempo del *detrend* di un intervallo del segnale di *elevation*, per la prova a velocità costante, $\omega = 400^\circ/\text{s}$.

Si evidenzia immediatamente, in particolare per i dati relativi all'*azimuth*, come sia presente una componente sinusoidale. Per i grafici relativi all'*elevation* si nota invece una forte componente di altre frequenze.

Al fine di determinare quale sia l'errore dovuto da tali disturbi, non solo in termini di posizione ma anche in termini di velocità, a regime, è interessante riportare anche i grafici relativi alle velocità, depurate della loro componente costante, in corrispondenza degli intervalli di valori selezionati.

Poiché come spiegato in precedenza, l'unico segnale acquisito era quello di posizione, si è quindi dovuto ottenere la velocità come uscita di un sistema dinamico equivalente ad un derivatore reale ed avente in ingresso la posizione.

Come si nota dalle figure seguenti è ovviamente maggiormente presente una componente di disturbo ad alta frequenza.

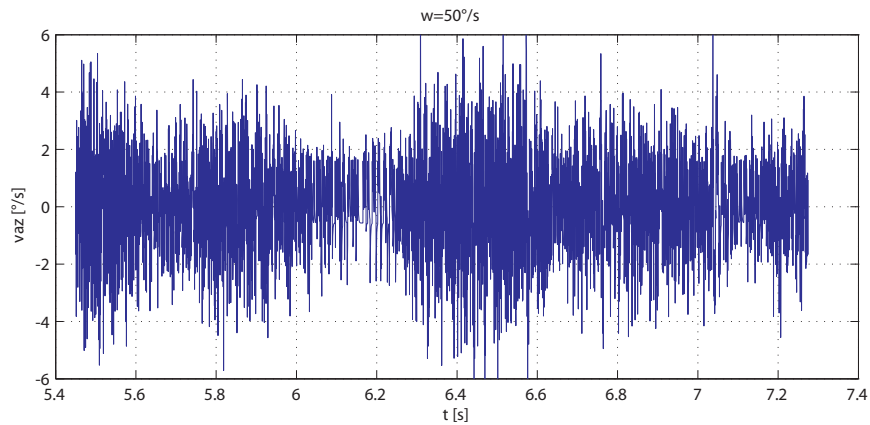


Figura 6.12. Andamento nel tempo di un intervallo della velocità di *azimuth*, per la prova a velocità costante, $\omega = 50^\circ/\text{s}$.

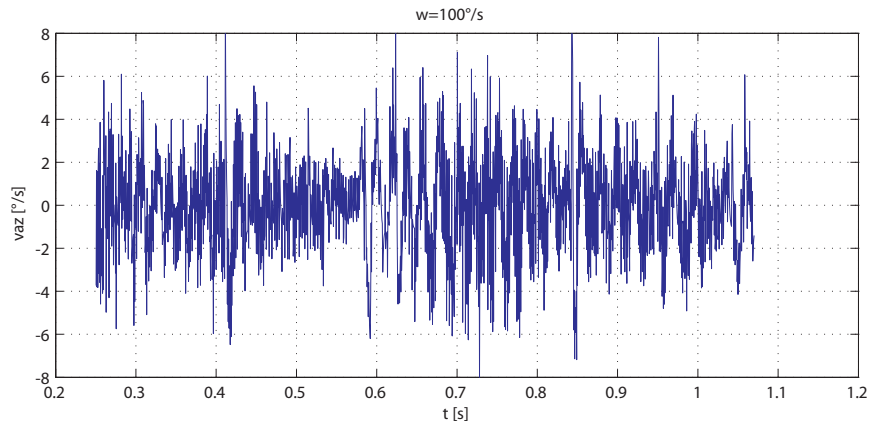


Figura 6.13. Andamento nel tempo di un intervallo della velocità di *azimuth*, per la prova a velocità costante, $\omega = 100^\circ/\text{s}$.

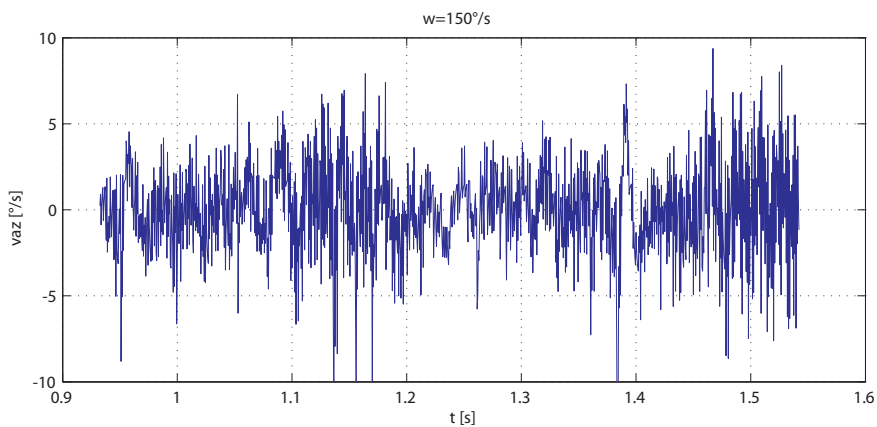


Figura 6.14. Andamento nel tempo di un intervallo della velocità di *azimuth*, per la prova a velocità costante, $\omega = 150^\circ/\text{s}$.

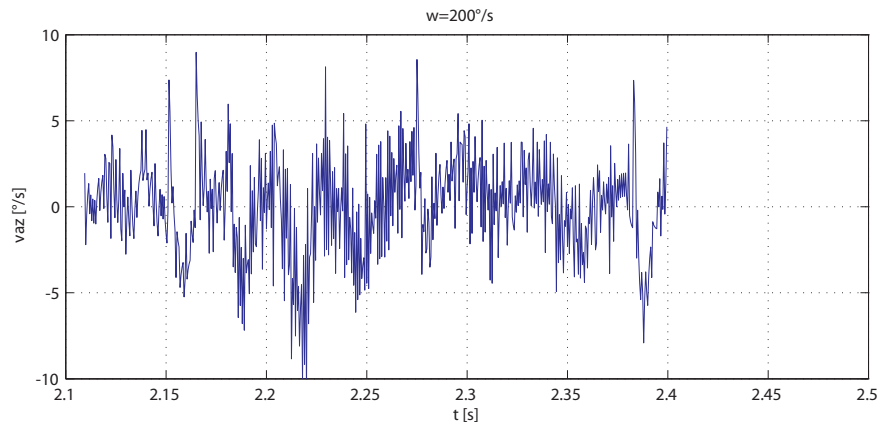


Figura 6.15. Andamento nel tempo di un intervallo della velocità di *azimuth*, per la prova a velocità costante, $\omega = 200^\circ/\text{s}$.

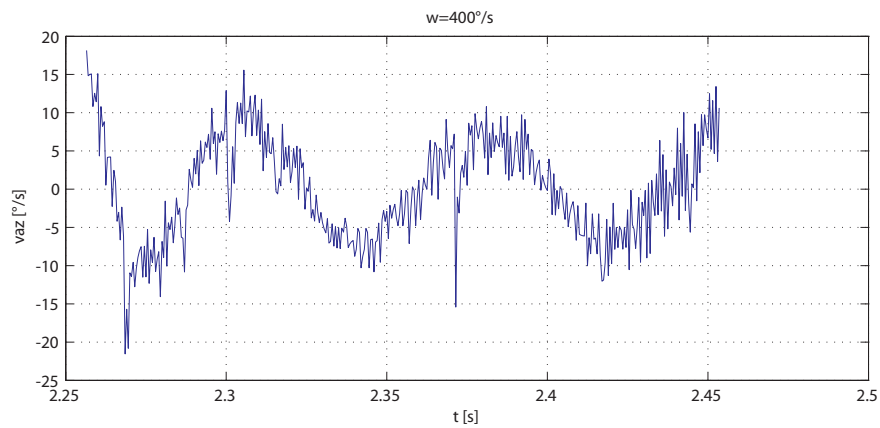


Figura 6.16. Andamento nel tempo di un intervallo della velocità di *azimuth*, per la prova a velocità costante, $\omega = 400^\circ/\text{s}$.

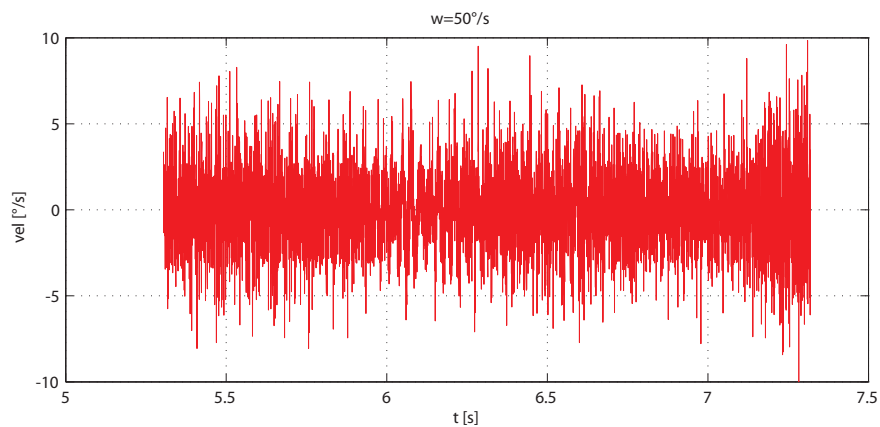


Figura 6.17. Andamento nel tempo di un intervallo della velocità di *elevation*, per la prova a velocità costante, $\omega = 50^\circ/\text{s}$.

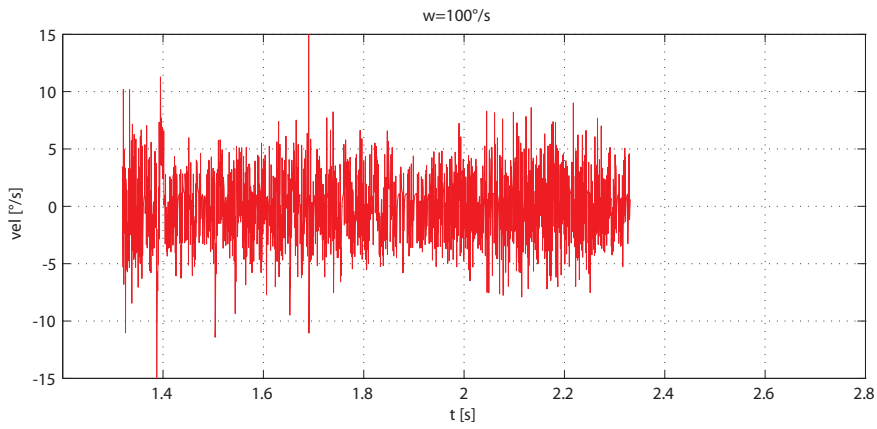


Figura 6.18. Andamento nel tempo di un intervallo della velocità di *elevation*, per la prova a velocità costante, $\omega = 100^\circ/\text{s}$.

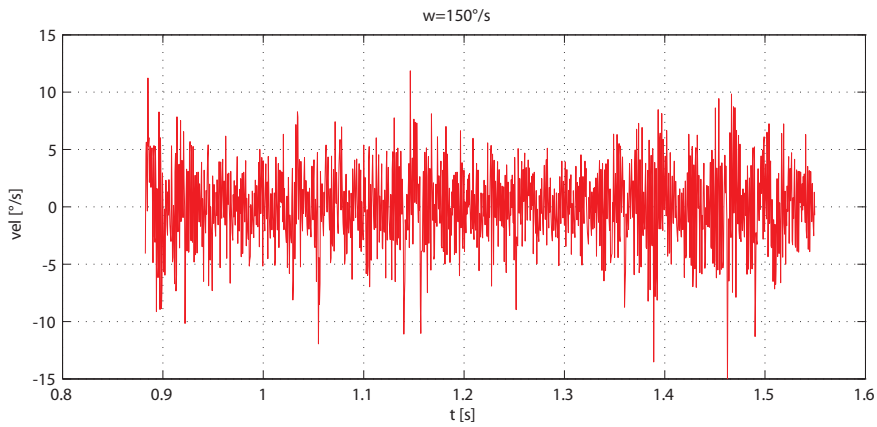


Figura 6.19. Andamento nel tempo di un intervallo della velocità di *elevation*, per la prova a velocità costante, $\omega = 150^\circ/\text{s}$.

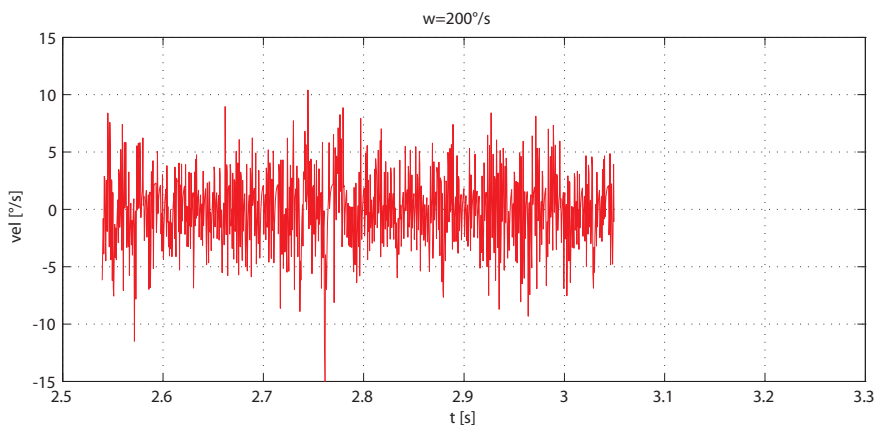


Figura 6.20. Andamento nel tempo di un intervallo della velocità di *elevation*, per la prova a velocità costante, $\omega = 200^\circ/\text{s}$.

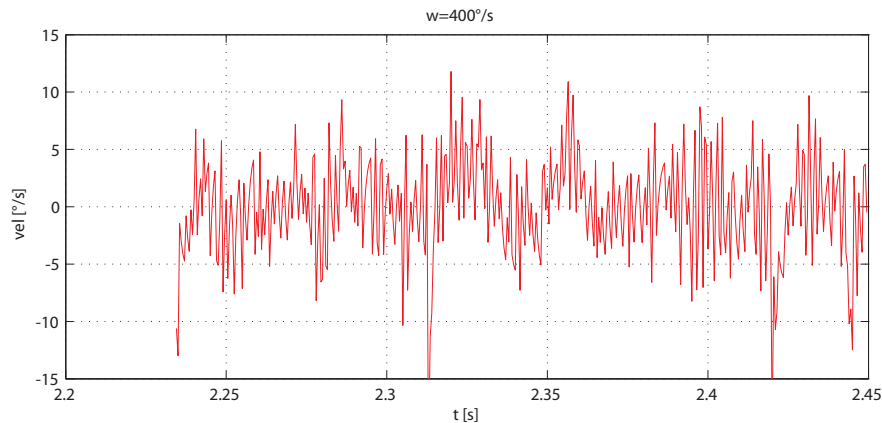


Figura 6.21. Andamento nel tempo di un intervallo della velocità di *elevation*, per la prova a velocità costante, $\omega = 400^\circ/\text{s}$.

Si nota come l'errore, che in termini di posizione era abbastanza limitato, in velocità diventa rilevante, dell'ordine di qualche grado.

6.2. Analisi in frequenza

Al fine di determinare in modo preciso le componenti armoniche dei segnali costruiti è stata fatta una analisi spettrale.

È stata quindi applicata la trasformata discreta di Fourier ai dati ottenuti dal *detrend* del segnale.

$$[6.1] \quad X(k) = \sum_{n=1}^N x_n e^{-\frac{2\pi}{N}i(n-1)(k-1)}.$$

Si è preferito per ragioni di velocità di analisi non implementare l'equazione sopra scritta ma utilizzare invece la *Fast Fourier Trasfor* che ottiene lo stesso risultato tramite opportune fattorizzazioni con un numero di operazioni notevolmente minore.

6.2.1. Basse frequenze

Si riportano ora gli spettri del segnale, limitatamente alle basse frequenze, per *azimuth* ed *elevation*.

La trasformata discreta ha un limite nella risoluzione in frequenza che può avere, determinato dal tempo di osservazione. Esso è pari a, considerando la frequenza angolare:

$$[6.2] \quad \Delta\omega = \frac{1}{T}2\pi.$$

Per tale motivo gli spettri corrispondenti a velocità maggiori, per i quali si è potuto isolare un intervallo più breve, hanno una risoluzione peggiore.

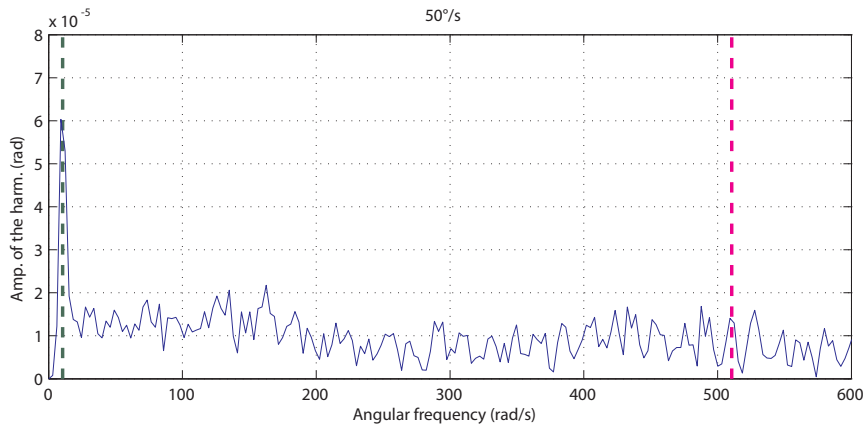


Figura 6.22. Spettro in frequenza per il segnale di *azimuth*, $\omega = 50^\circ/\text{s}$, basse frequenze, sono evidenziati i due picchi principali.

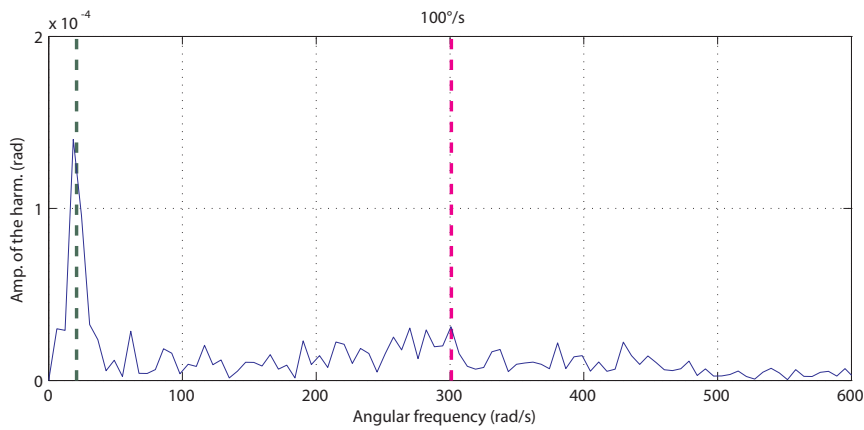


Figura 6.23. Spettro in frequenza per il segnale di *azimuth*, $\omega = 100^\circ/\text{s}$, basse frequenze, sono evidenziati i due picchi principali.

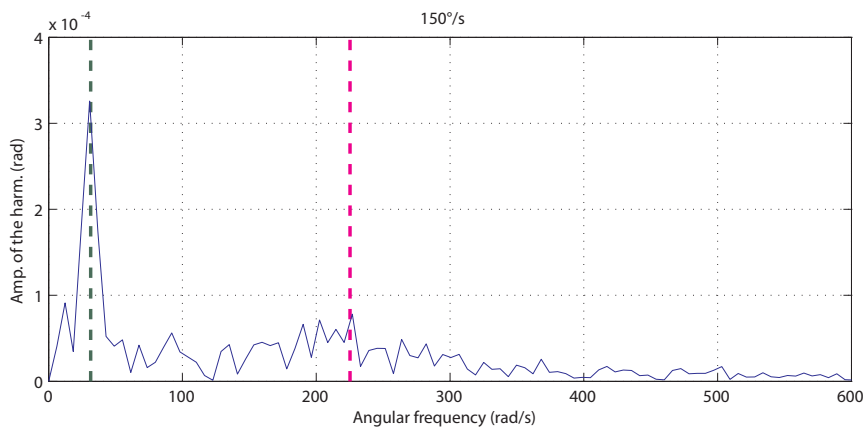


Figura 6.24. Spettro in frequenza per il segnale di *azimuth*, $\omega = 150^\circ/\text{s}$, basse frequenze, sono evidenziati i due picchi principali.

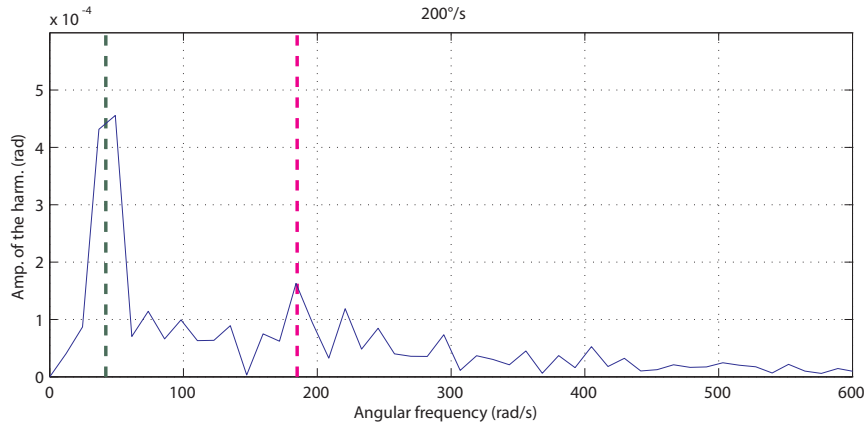


Figura 6.25. Spettro in frequenza per il segnale di *azimuth*, $\omega = 200^\circ/s$, basse frequenze, sono evidenziati i due picchi principali.

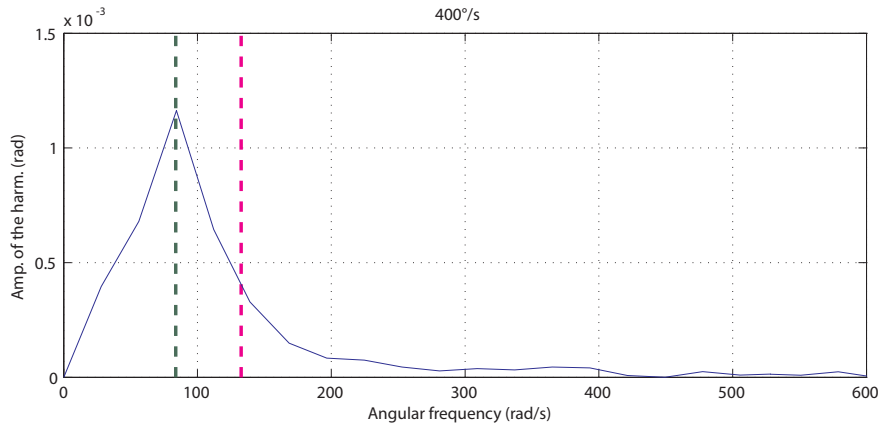


Figura 6.26. Spettro in frequenza per il segnale di *azimuth*, $\omega = 400^\circ/s$, basse frequenze, sono evidenziati i due picchi principali.

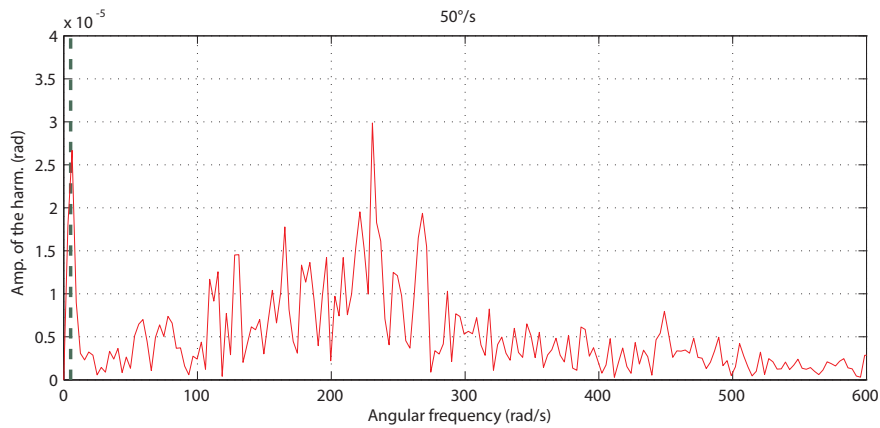


Figura 6.27. Spettro in frequenza per il segnale di *elevation*, $\omega = 50^\circ/s$, basse frequenze, è evidenziato il picco principale.

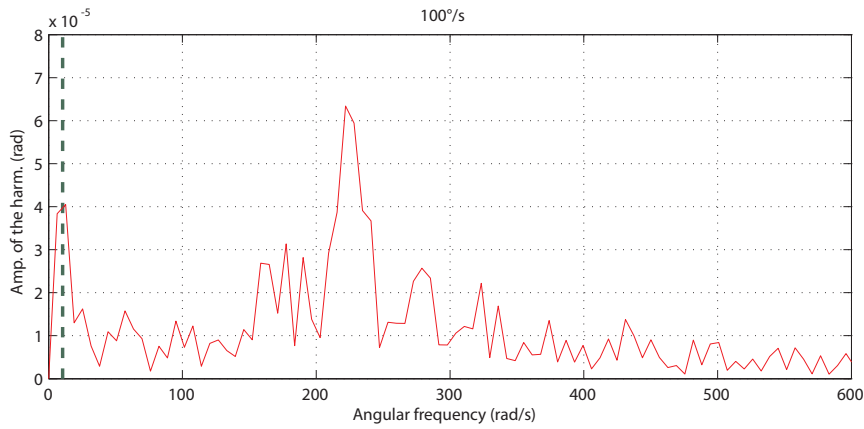


Figura 6.28. Spettro in frequenza per il segnale di *elevation*, $\omega = 100^\circ/s$, basse frequenze, è evidenziato il picco principale.

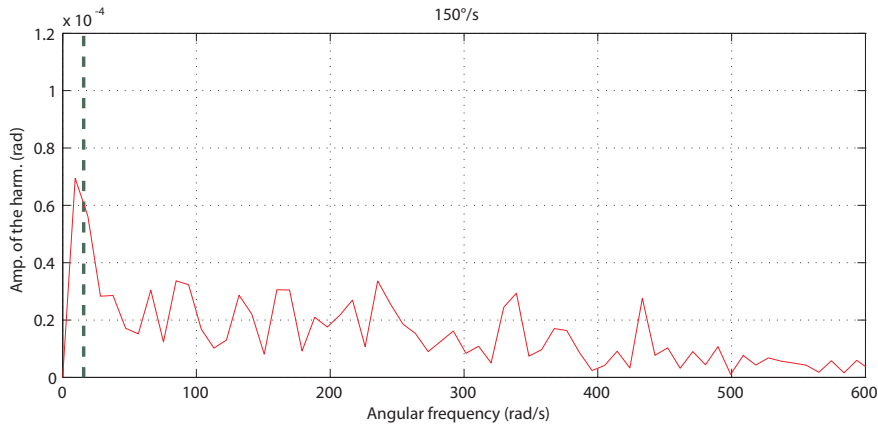


Figura 6.29. Spettro in frequenza per il segnale di *elevation*, $\omega = 150^\circ/s$, basse frequenze, è evidenziato il picco principale.

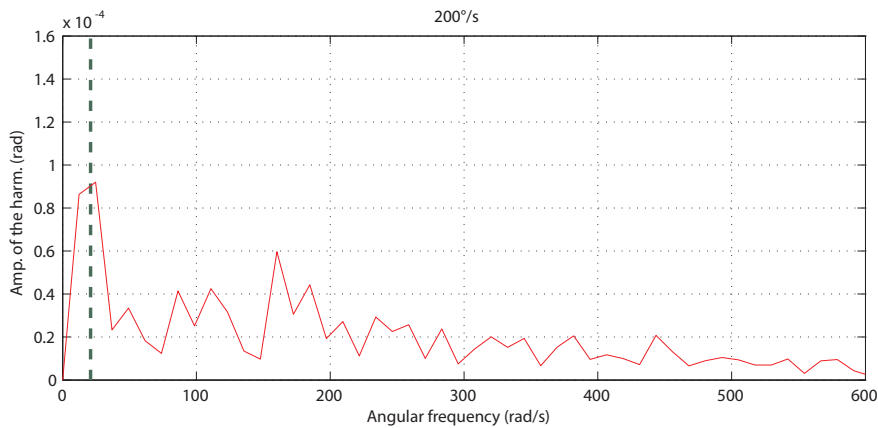


Figura 6.30. Spettro in frequenza per il segnale di *elevation*, $\omega = 200^\circ/s$, basse frequenze, è evidenziato il picco principale.

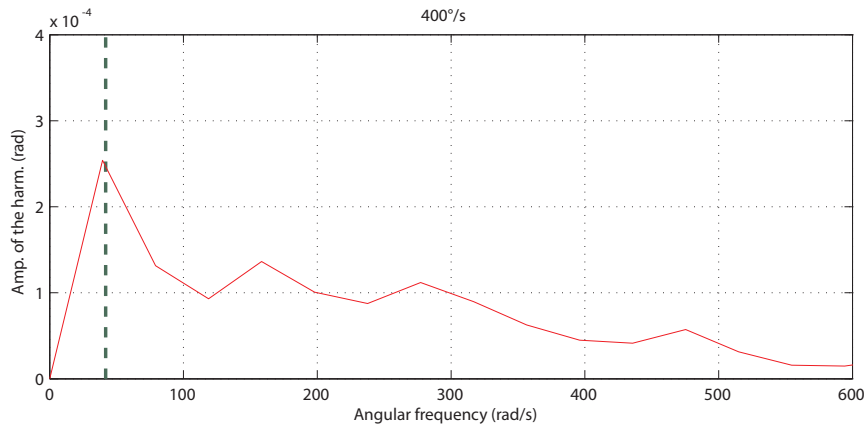


Figura 6.31. Spettro in frequenza per il segnale di *elevation*, $\omega = 400^\circ/\text{s}$, basse frequenze, è evidenziato il picco principale.

Dagli spettri del segnale appena rappresentati posso essere fatte alcune considerazioni.

Emerge, per entrambi i gradi di libertà, un primo picco a frequenze molto basse, la cui frequenza ha un andamento lineare con la velocità, come evidenziato in Figura 6.32 ed in Figura 6.33 a pagina 88. Tale picco è associato al numero di coppie del motore, 12 per quello di *azimuth* e 6 per quello di *elevation*. Andando a rappresentare i segnali analizzati non in funzione del tempo ma in funzione dell'angolo si nota infatti come i disturbi abbiano la stessa frequenza spaziale, corrispondente ad un angolo di 30° per l'*azimuth* e di 60° per l'*elevation*, come può essere notato in Figura 6.34 a pagina 89 ed in Figura 6.35, anche se in misura minore a causa del maggior rumore.

Per quanto riguarda l'*elevation* non possono essere fatte considerazioni aggiuntive, per quanto riguarda le componenti a bassa frequenza, perché i picchi che si notano a bassa velocità intorno ai 200–300 rad/s non si ripetono per le prove a velocità maggiore, scomparendo totalmente.

Per l'*azimuth* invece può essere notato un secondo picco, di ampiezza crescente con la velocità e di frequenza inversamente proporzionale. Esso a basse velocità si confonde con il rumore, essendo praticamente non identificabile a $50^\circ/\text{s}$, e ad alte velocità coalesce con il primo picco, tuttavia per le velocità intermedie è abbastanza evidente, come si nota anche dall'andamento temporale se al segnale analizzato viene sottratta la prima frequenza, Figura 6.37 a pagina 90.

Andando ad analizzare in particolare le frequenze si nota come il loro legame con la velocità viene ottimamente approssimato dall'inverso del quadrato della velocità di rotazione.

$$[6.3] \quad f \propto \frac{1}{\omega^2}.$$

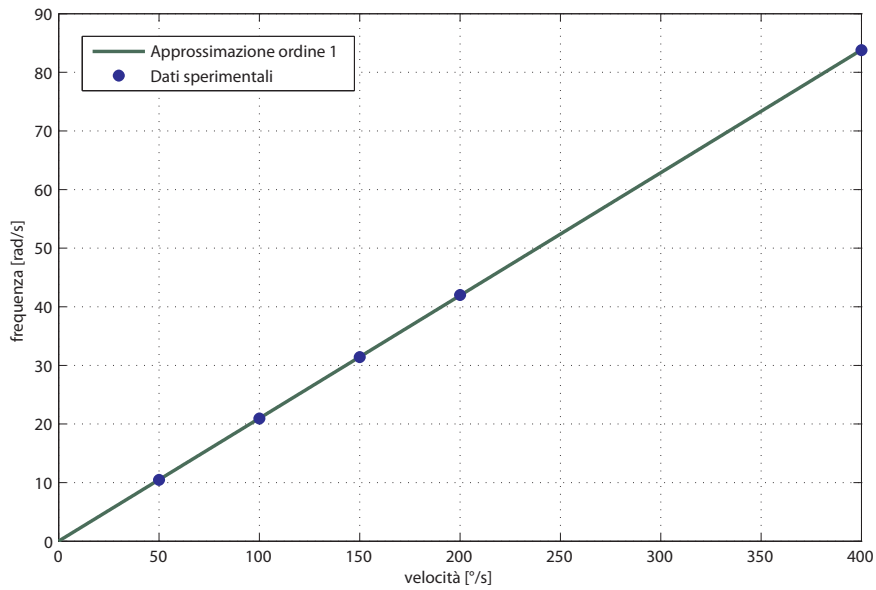


Figura 6.32. Approssimazione lineare del legame tra frequenza del primo picco per il segnale di *azimuth* e velocità, confrontata con i dati sperimentali.

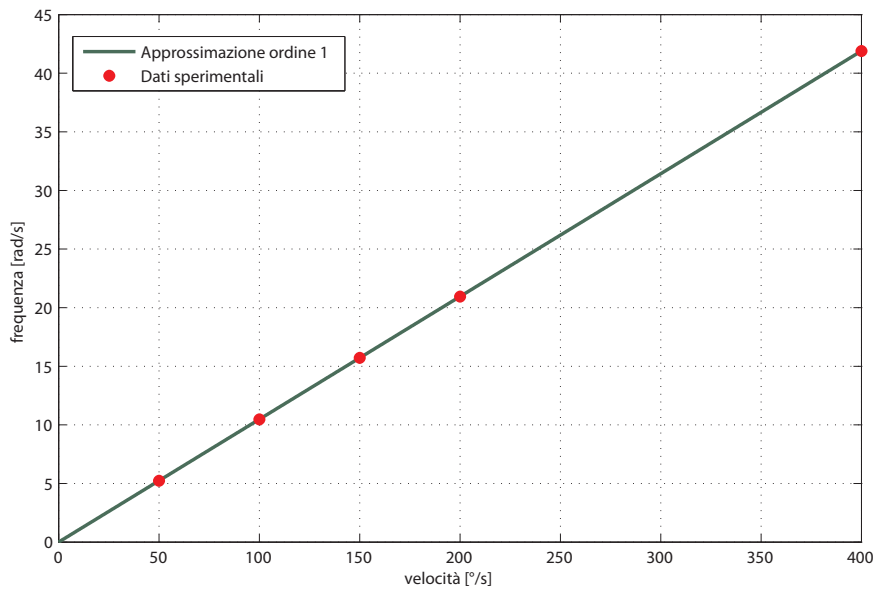


Figura 6.33. Approssimazione lineare del legame tra frequenza del primo picco per il segnale di *elevation* e velocità, confrontata con i dati sperimentali.

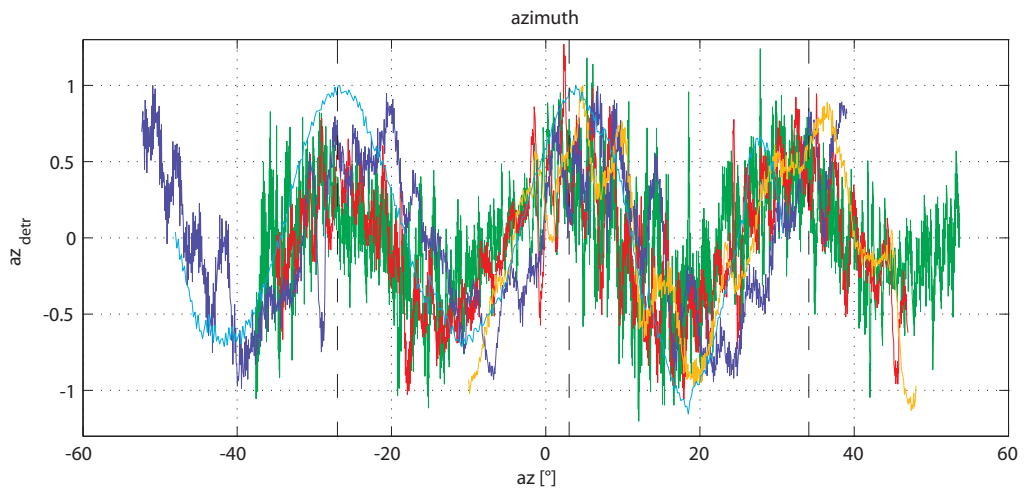


Figura 6.34. Andamento del *detrend* del segnale di *azimuth* rispetto all'*azimuth* stesso, per le differenti velocità. Si è normalizzata l'ampiezza dei segnali per poterli confrontare tra loro.

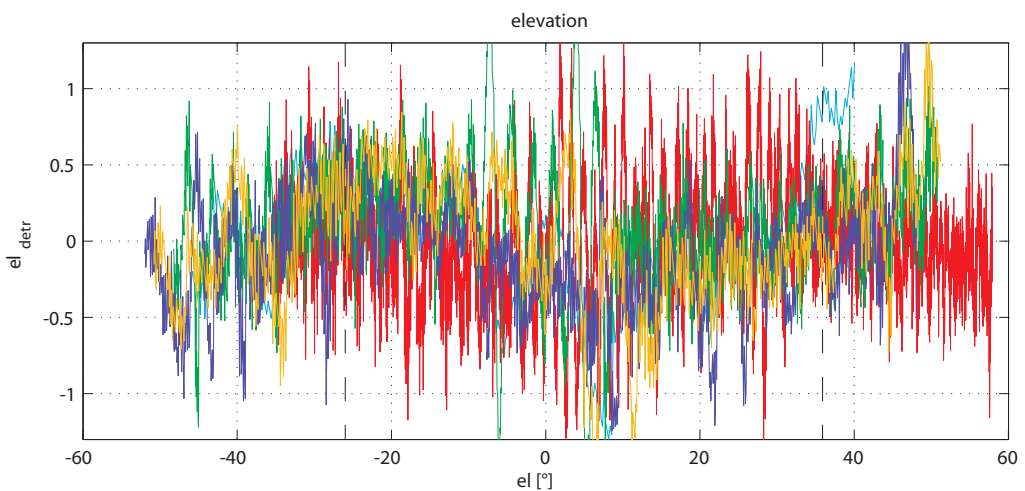


Figura 6.35. Andamento del *detrend* del segnale di *elevation* rispetto all'*elevation* stesso, per le differenti velocità. Si è normalizzata l'ampiezza dei segnali per poterli confrontare tra loro.

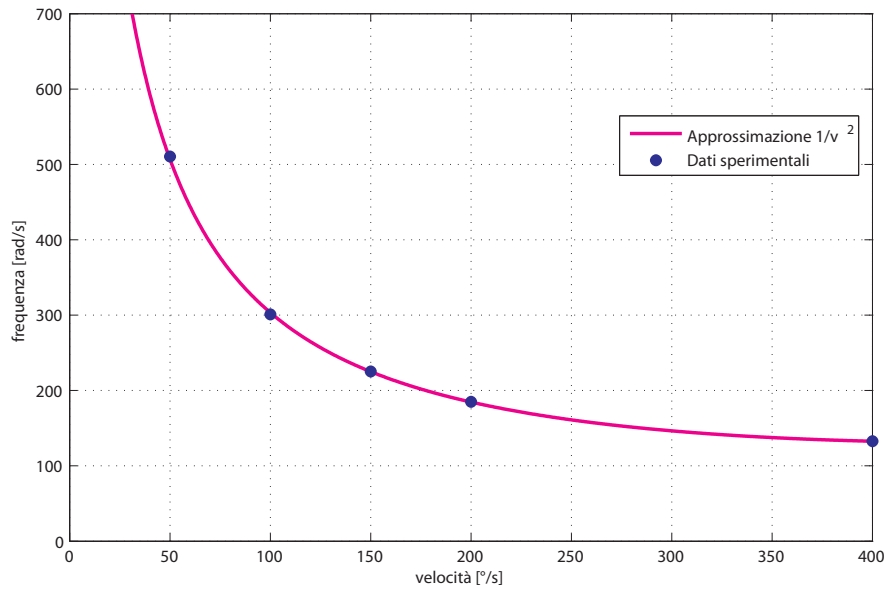


Figura 6.36. Approssimazione del legame tra frequenza del secondo picco per il segnale di *azimuth* e velocità, confortata con i dati sperimentali.

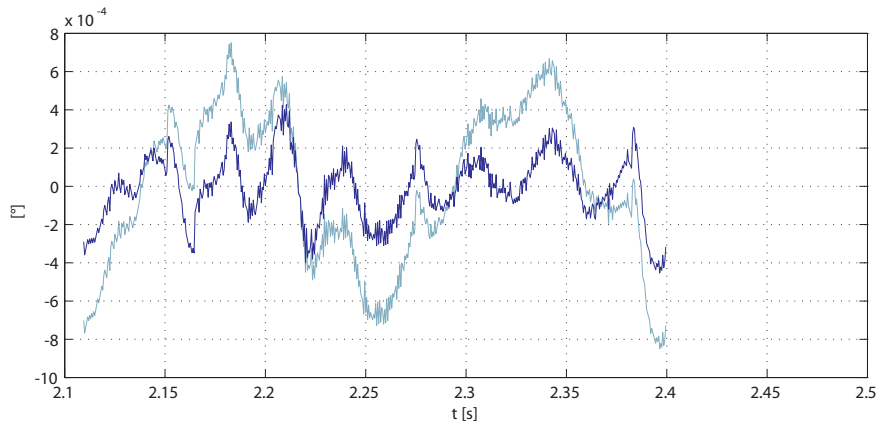


Figura 6.37. Andamento nel tempo del *detrend* del segnale di *azimuth*, a cui è stata sottratta la prima componente in frequenza, in scuro, in chiaro il segnale originale.

Si analizza ora l'ampiezza dei tre picchi che sono stati evidenziati, due per il segnale di *azimuth* ed uno per il segnale di *elevation*.

Quello che si nota è un andamento quadratico rispetto alla velocità per i due picchi relativi all' *azimuth* ed un andamento lineare per quanto riguarda l' *elevation*.

Tali andamenti sono rappresentati nelle figure seguenti.

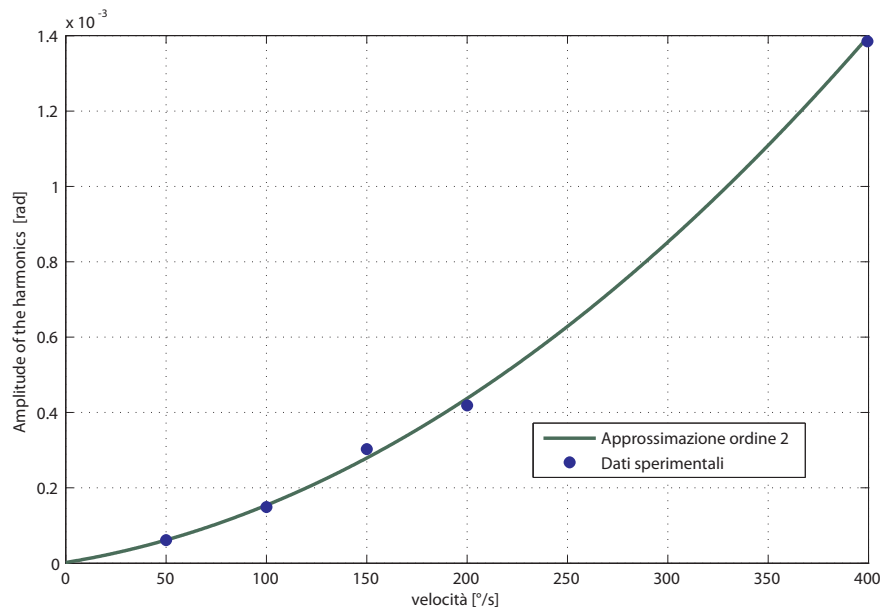


Figura 6.38. Approssimazione quadratica del legame tra ampiezza del primo picco per il segnale di *azimuth* e velocità, confortata con i dati sperimentali.

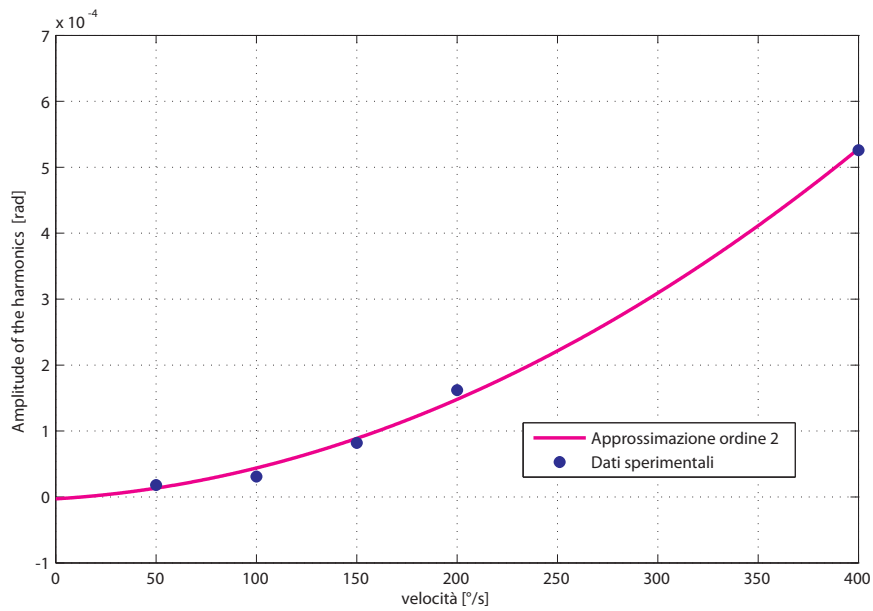


Figura 6.39. Approssimazione quadratica del legame tra ampiezza del secondo picco per il segnale di *azimuth* e velocità, confortata con i dati sperimentali.

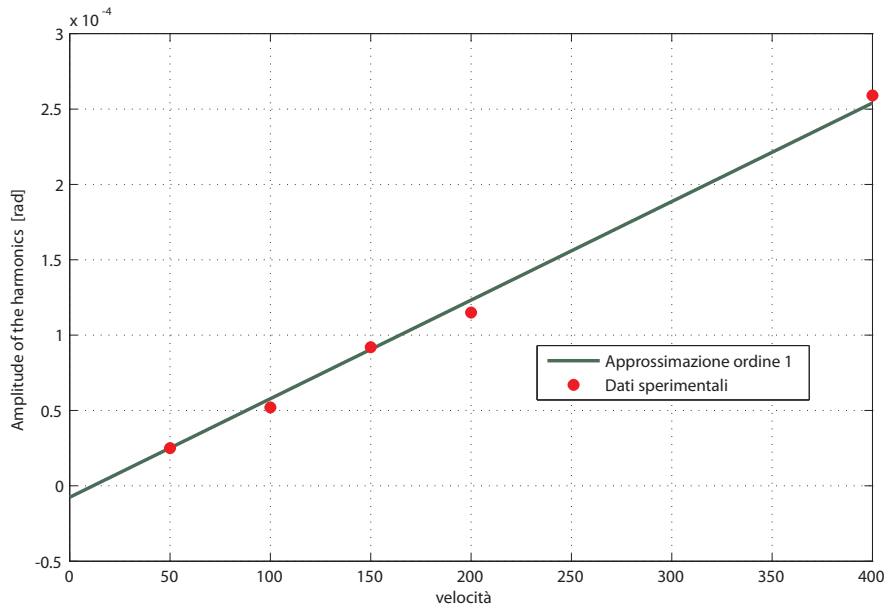


Figura 6.40. Approssimazione lineare del legame tra ampiezza del secondo picco per il segnale di *elevation* e velocità, confortata con i dati sperimentali.

6.2.2. Alte frequenze

Si analizzano ora le alte frequenze dello spettro.

Nel fare questo dovrà essere tenuto in considerazione il fenomeno dell'*aliasing*, per cui solo i picchi fino alla metà della frequenza di campionamento sono analizzati correttamente. I picchi di frequenza superiore sono invece visti come picchi di frequenza differente.

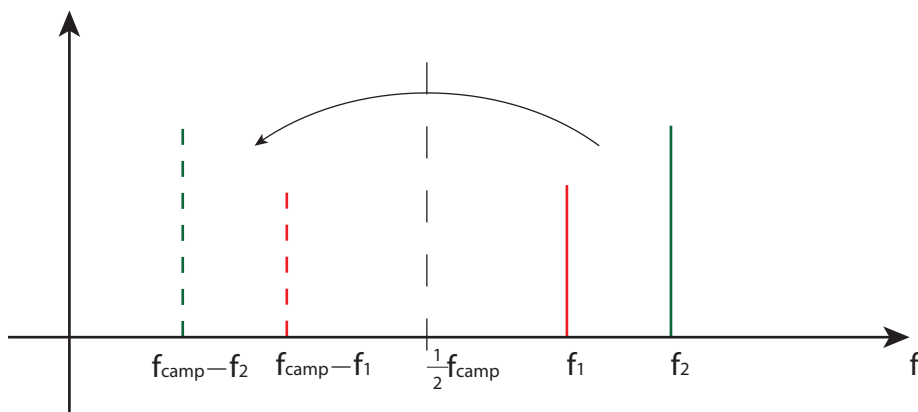


Figura 6.41. Rappresentazione del fenomeno dell'*aliasing*, in linea continua i picchi reali, in linea tratteggiata i picchi ottenuti dall'analisi del segnale campionato a f_{camp} .

Le due sessioni di acquisizioni, essendo state fatte con tempi di acquisizione differenti hanno anche due frequenze di campionamento differenti.

Tabella 6.1. Frequenza di campionamento

Δt [ms]	f_{camp} [rad/s]	$\frac{1}{2} f_{\text{camp}}$ [rad/s]
0.5	12566	6283
0.25	25133	12566

Si riportano ora i differenti spettri che si sono ottenuti, per *azimuth* e per l'*elevation*.

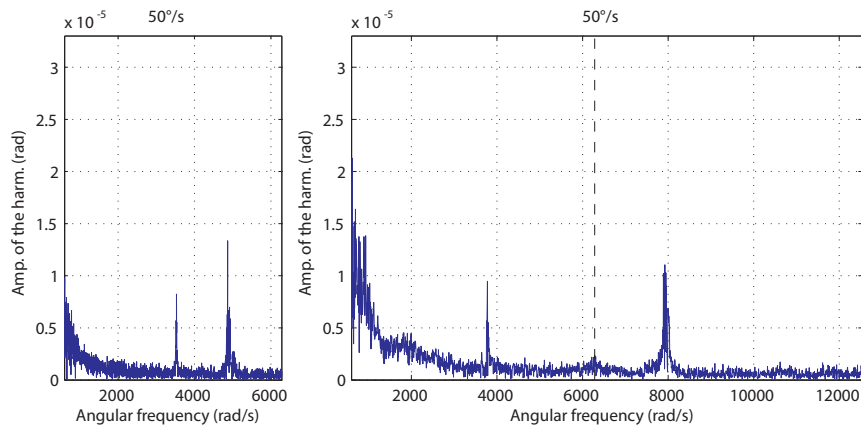


Figura 6.42. Spettro in frequenza per il segnale di *azimuth*, $\omega = 50^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5\text{ms}$, a destra $\Delta t = 0.25\text{ms}$.

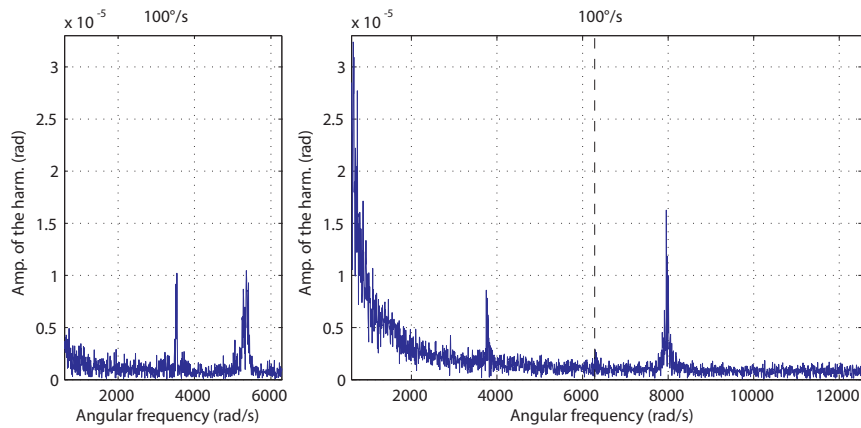


Figura 6.43. Spettro in frequenza per il segnale di *azimuth*, $\omega = 100^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5\text{ms}$, a destra $\Delta t = 0.25\text{ms}$.

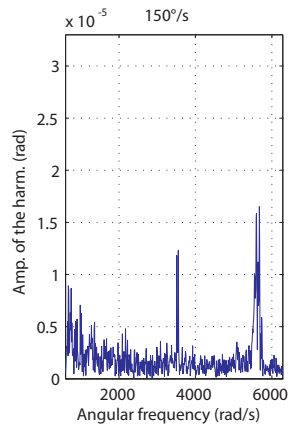


Figura 6.44. Spettro in frequenza per il segnale di *azimuth*, $\omega = 150^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5\text{ms}$.

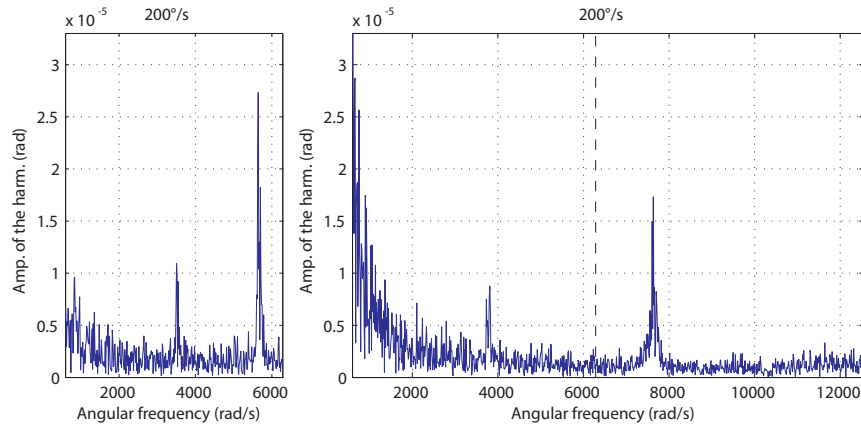


Figura 6.45. Spettro in frequenza per il segnale di *azimuth*, $\omega = 200^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5\text{ms}$, a destra $\Delta t = 0.25\text{ms}$.

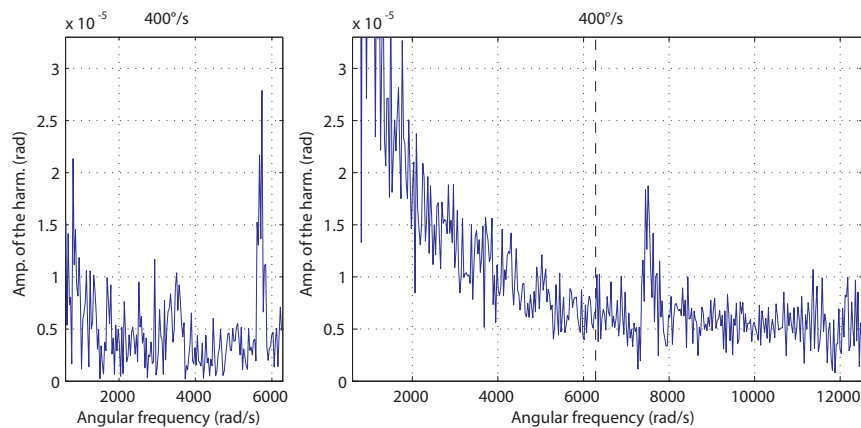


Figura 6.46. Spettro in frequenza per il segnale di *azimuth*, $\omega = 400^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5\text{ms}$, a destra $\Delta t = 0.25\text{ms}$.

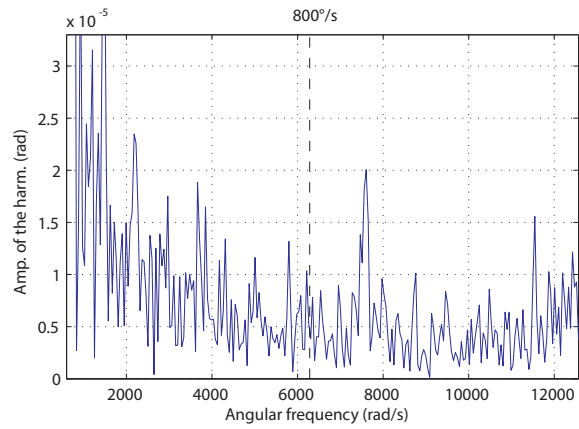


Figura 6.47. Spettro in frequenza per il segnale di *azimuth*, $\omega = 800^\circ/s$, alte frequenze, a destra $\Delta t = 0.25ms$.

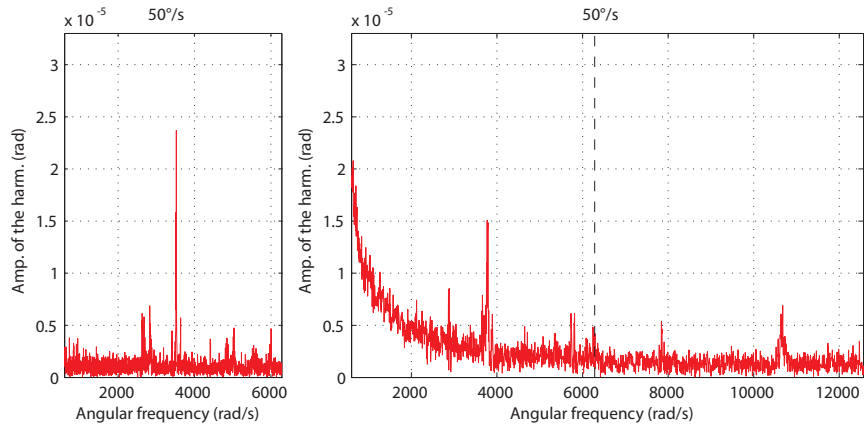


Figura 6.48. Spettro in frequenza per il segnale di *elevation*, $\omega = 50^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5ms$, a destra $\Delta t = 0.25ms$.

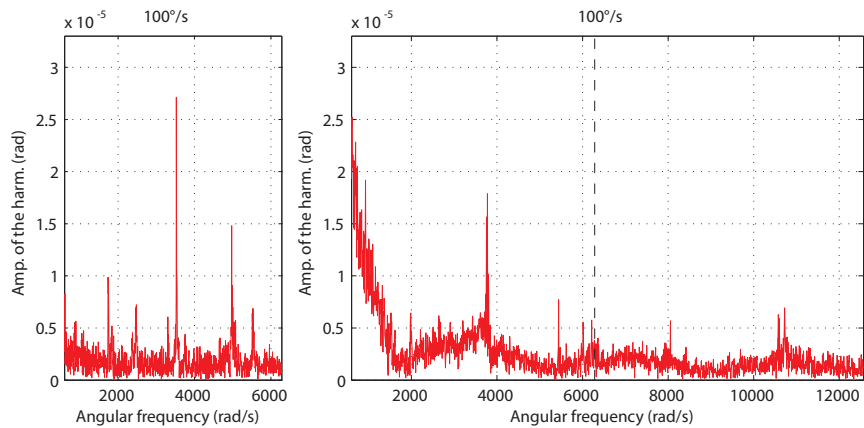


Figura 6.49. Spettro in frequenza per il segnale di *elevation*, $\omega = 100^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5ms$, a destra $\Delta t = 0.25ms$.

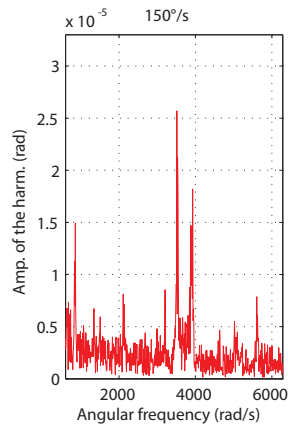


Figura 6.50. Spettro in frequenza per il segnale di *elevation*, $\omega = 150^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5\text{ms}$, a destra $\Delta t = 0.25\text{ms}$.

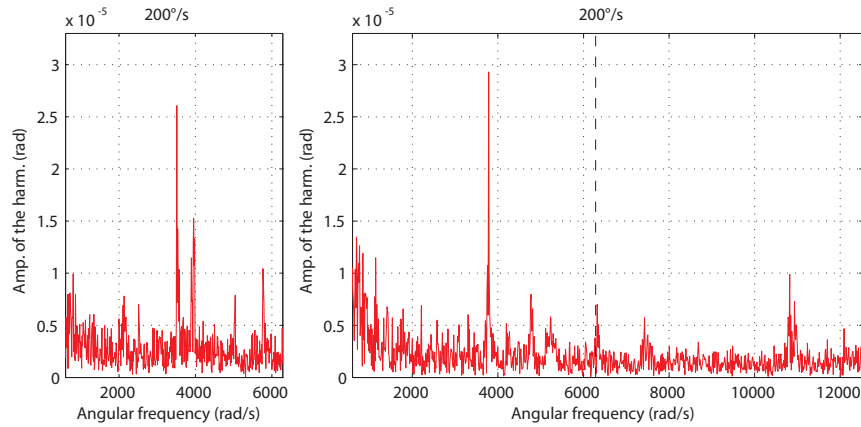


Figura 6.51. Spettro in frequenza per il segnale di *elevation*, $\omega = 200^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5\text{ms}$, a destra $\Delta t = 0.25\text{ms}$.

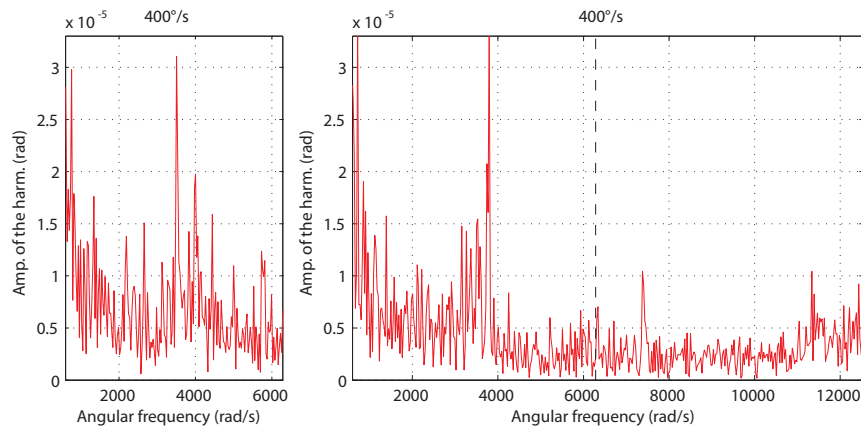


Figura 6.52. Spettro in frequenza per il segnale di *elevation*, $\omega = 400^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5\text{ms}$, a destra $\Delta t = 0.25\text{ms}$.

Per entrambi i gradi di libertà si notano due picchi sia nel campionamento con passo di mezzo millisecondo, a 3500 rad/s e 5000 rad/s, che in quello più veloce, a poco meno di 4000 rad/s e poco più di 7000 rad/s. Tali picchi si spostano poco in frequenza con la velocità ed anche l'ampiezza rimane pressochè costante.

Lo spettro ottenuto con il tempo di campionamento dimezzato sembrerebbe mostrare un risultato discordante, rispetto a quanto si ottiene con $\Delta t = 0.5\text{ms}$.

Tuttavia, ipotizzando un tempo di campionamento diverso da quello previsto, leggermente superiore e pari a 0.26 ms, il primo picco, a poco meno di 4000 rad/s coincide, con quello ottenuto dal campionamento più lento.

Tale ipotesi potrebbe inoltre giustificare il picco a 5000 rad/s come un effetto di *aliasing* dovuto al picco a 7000 rad/s.

In ogni caso la frequenza di tali disturbi è molto elevata, molto fuori dalla banda di controllo e questo, insieme alla loro ampiezza relativamente bassa, permette di non considerarli problematici per il controllo del sensore stesso.

6.3. Modellazione dei disturbi

Gli esperimenti effettuati sul sensore e la successiva analisi dei dati ottenuti ha permesso di identificare dei disturbi di coppia che agiscono sul motore, sia per il grado di libertà di *azimuth*, che per quello di *elevation*.

Si sono identificati in modo abbastanza chiaro due tipologie di disturbo, una di disturbi ad alta frequenza ed una seconda a frequenze più basse.

I disturbi ad alta frequenza, come già visto, sono molto oltre le bande di controllo che si desidera ottenere. È inoltre difficile capire quale sia la loro origine, considerando anche che le prove effettuate, per limitazioni della apparecchiatura di prova, sono state effettuate ad anello chiuso e questo porta ad avere delle interazioni dell'elettronica di controllo, che potrebbe avere causato i disturbi misurati.

A causa di queste considerazioni si eviterà quindi di modellare i disturbi ad alta frequenza, considerando invece solo quelli a bassa frequenza.

Come emerge dai dati i disturbi che sono stati individuati sono tre, due per l'*azimuth* e uno per l'*elevation*. Considereremo ora i due disturbi la cui frequenza cresce linearmente con la velocità.

6.3.1. Disturbi di frequenza lineare con la velocità

Come già osservato i due disturbi che hanno una frequenza temporale proporzionale con la velocità mostrano invece una frequenza angolare costante. Tale fatto porta ad individuare il legame che sussiste tra essi e le coppie polari dei motori.

Si può quindi ipotizzare che il disturbo sia imputabile ad uno squilibrio nella corrente di uno dei due avvolgimenti del motore. Se infatti l'andamento della corrente nel tempo non è perfettamente sinusoidale ma presenta un *offset*, la coppia generata dal motore, che è il prodotto della corrente per la costante di coppia, non è più co-

stante ed indipendente dall'angolo elettrico, ma mostra invece un andamento sinusoidale che, rispetto alla posizione angolare, ha un periodo pari a

$$[6.4] \quad \vartheta = \frac{2\pi}{p};$$

dove p è il numero di coppie polari. Si è quindi introdotto un disturbo che si sommasse alla corrente in uno dei due avvolgimenti, e che fosse proporzionale alla corrente ed al suo quadrato, opportunamente filtrata per evitare variazioni troppo brusche.

Sono state effettuate alcune simulazioni variando i coefficienti di proporzionalità, per ottenere che l'ampiezza avesse l'andamento desiderato rispetto alla velocità.

Noti i valori, si sono quindi aggiunti gli opportuni blocchi allo schema del *Driver* dei motori, in modo equivalente per entrambi i gradi di libertà, come rappresentato in Figura 6.53.

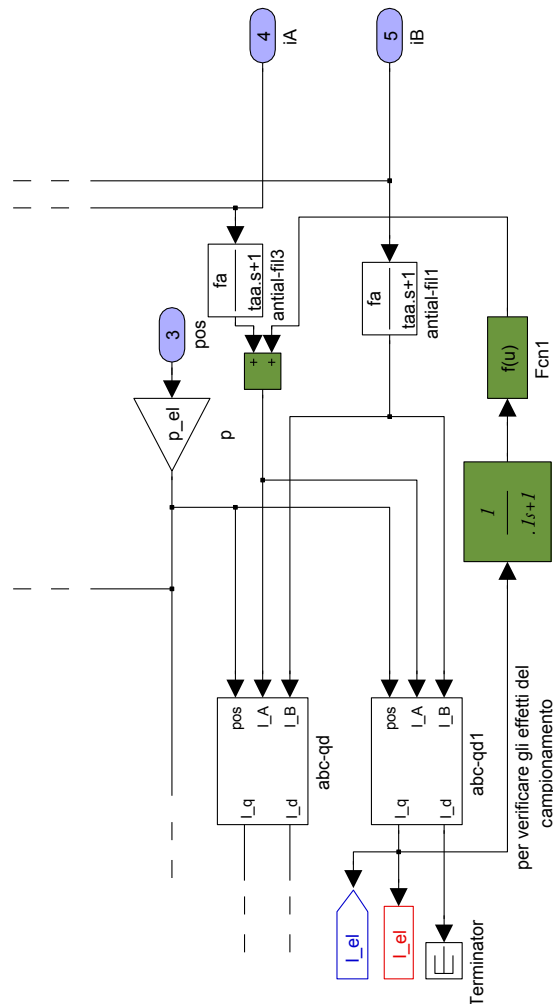


Figura 6.53. In verde i blocchi aggiunti allo schema rappresentato in Figura 4.5 a pagina 52, al fine di poter simulare i disturbi.

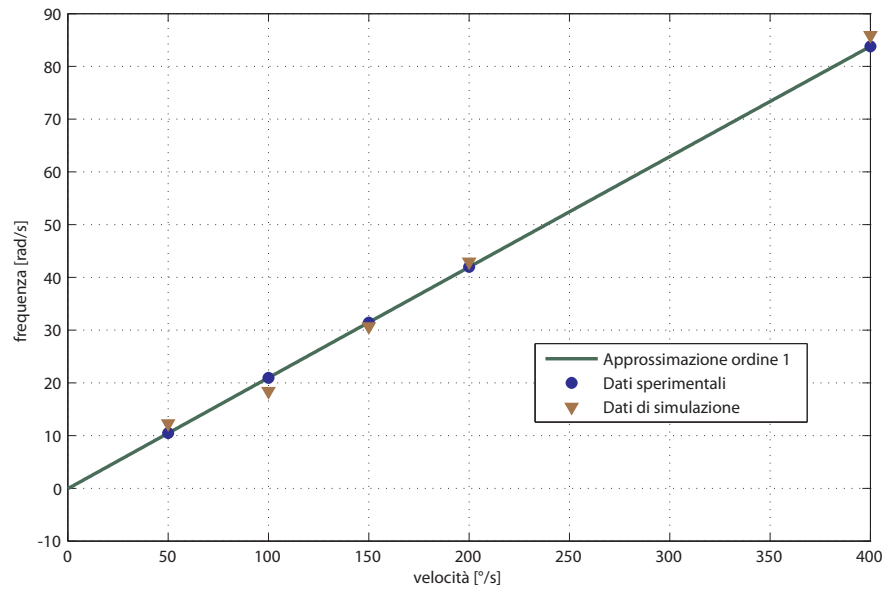


Figura 6.54. Andamento della frequenza dei disturbi di *azimuth* in funzione della velocità per i dati sperimentali e per i dati ottenuti in simulazione.

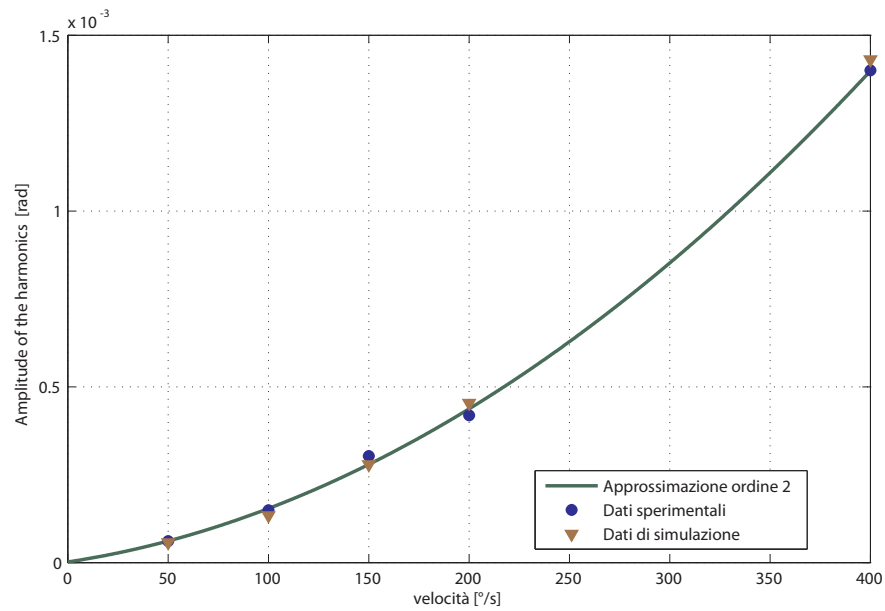


Figura 6.55. Andamento della ampiezza dei disturbi di *azimuth* in funzione della velocità per i dati sperimentali e per i dati ottenuti in simulazione.

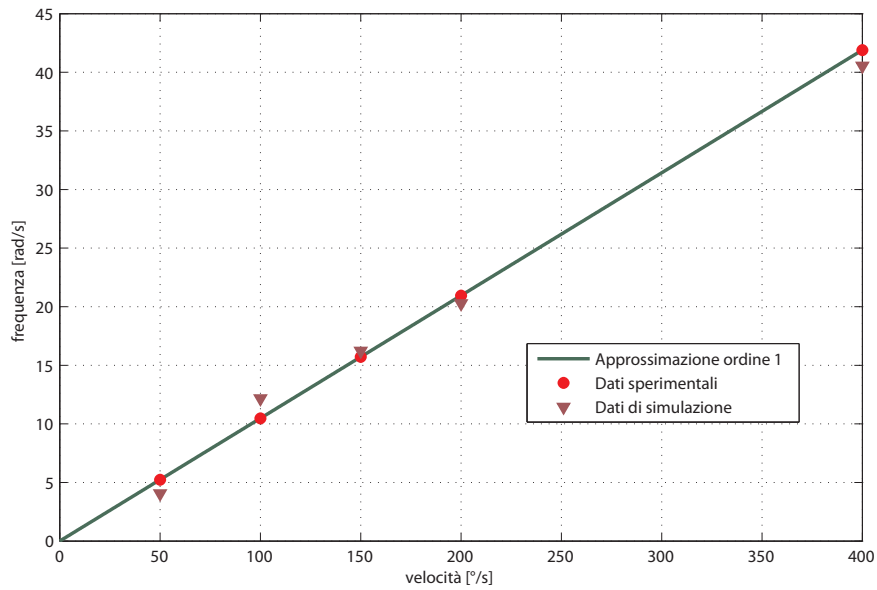


Figura 6.56. Andamento della frequenza dei disturbi di *elevation* in funzione della velocità per i dati sperimentali e per i dati ottenuti in simulazione.

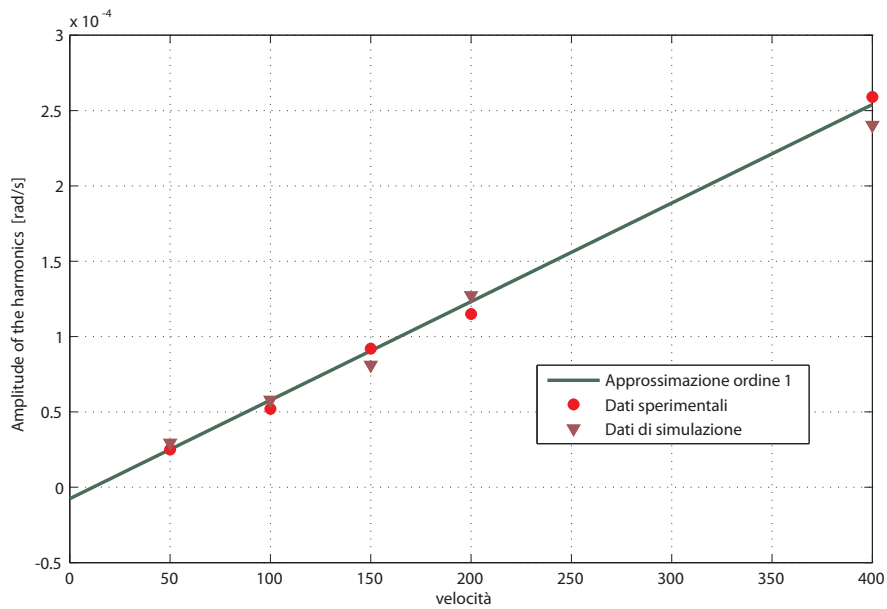


Figura 6.57. Andamento della ampiezza dei disturbi di *elevation* in funzione della velocità per i dati sperimentali e per i dati ottenuti in simulazione.

6.3.2. Disturbo di frequenza decrescente con la velocità

Diverse sono le considerazioni che possono essere fatte per quanto riguarda il disturbo, riscontrato sul grado di libertà di *azimuth*, che ha una frequenza che decresce con la velocità, in particolare proporzionalmente all'inverso del quadrato della velocità.

Tale disturbo infatti non ha una identificazione chiara, ma non può essere trascurato nella simulazione, avendo una frequenza compresa nella banda del controllore ed essendo di ampiezza non trascurabile.

Si è quindi deciso di applicarlo direttamente nel blocco del motore, come una coppia che va a sommarsi a quella prodotta dal motore stesso.

Il calcolo della coppia di disturbo viene fatto da una opportuna *S-Function* di *MATLAB* che riceve in ingresso la posizione e la velocità dando in uscita la coppia di disturbo.

Anche in questo caso si sono svolte delle simulazioni per ottenere un andamento della frequenze e della ampiezza pari a quello dei dati sperimentali.

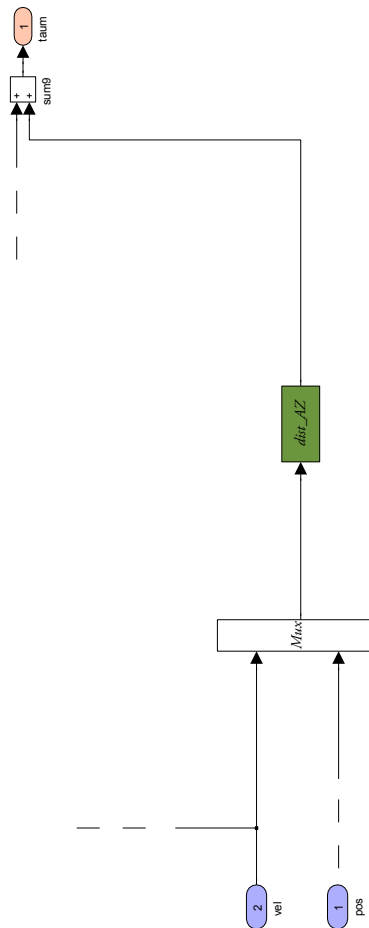


Figura 6.58. In verde blocco aggiunto al modello di motore di *azimuth*, identico a quello visualizzato in Figura 4.3 a pagina 50, per simulare un disturbo.

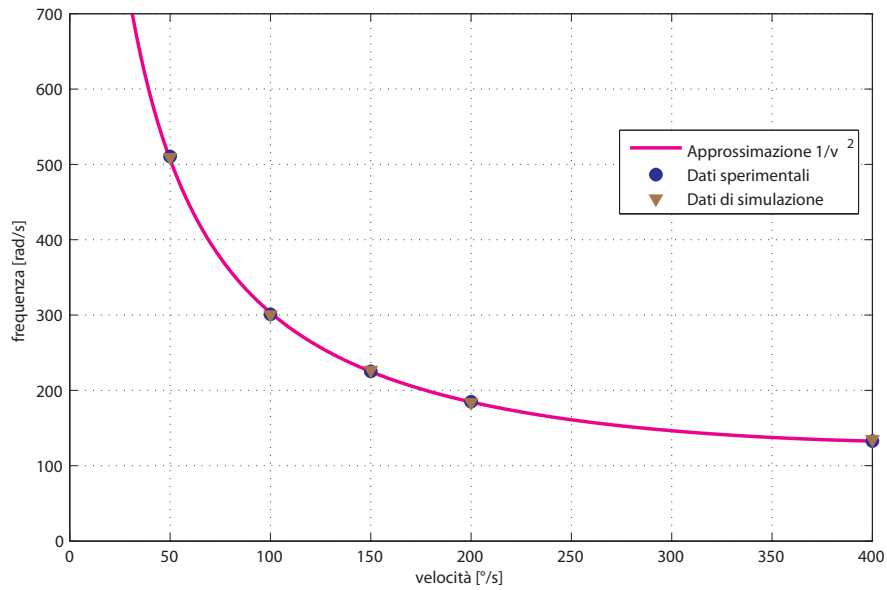


Figura 6.59. Andamento della frequenza dei secondi disturbi di *azimuth* in funzione della velocità per i dati sperimentali e per i dati ottenuti in simulazione.

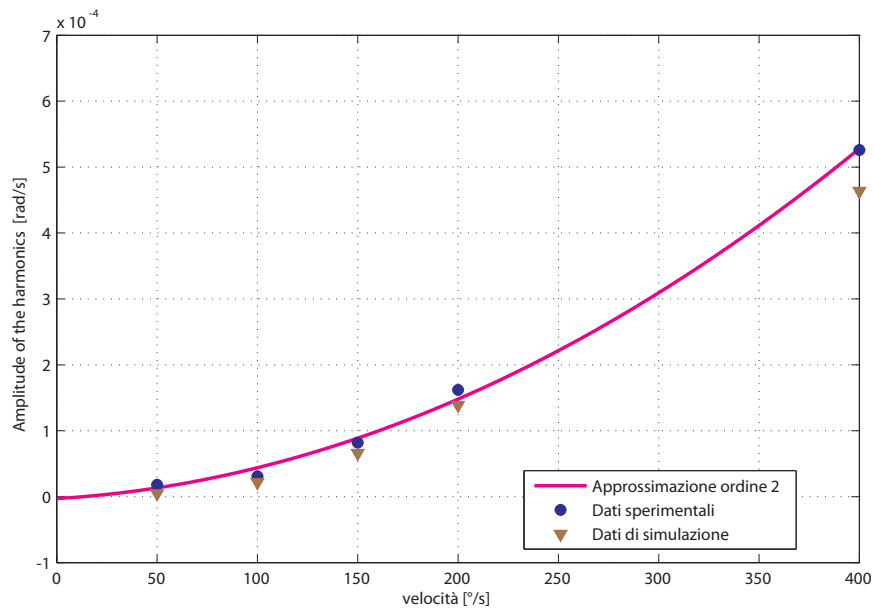


Figura 6.60. Andamento della ampiezza dei secondi disturbi di *azimuth* in funzione della velocità per i dati sperimentali e per i dati ottenuti in simulazione.

Capitolo 7

Controllore di posizione

In questo capitolo verrà affrontata la descrizione del regolatore che chiude l'anello di controllo di posizione del sensore.

Verrà descritta la tipologia di controllore utilizzata e spiegato il modello *Simulink* con il quale esso è stato implementato.

La taratura dei parametri del regolatore è stata quindi effettuata tenendo in considerazione sia le specifiche che i risultati delle prove sperimentali a scalino, descritte precedentemente. In ultimo verrà mostrato l'osservatore del disturbo di coppia, che si è reso necessario per soddisfare le specifiche.

7.1. Regolatore P/PI

Il regolatore utilizzato è un regolatore di tipo PID, che implementa un controllo a cascata sulla posizione e sulla velocità.

7.1.1. PID

Un regolatore di tipo PID è un regolatore che prevede un uscita che è la somma di tre contributi:

$$[7.1] \quad u = u_p + u_i + u_D ;$$

- $u_p = K_C e$, uno proporzionale all'errore;
- $u_i = \frac{K_C}{T_I} \int_0^i e \, d\tau$, uno proporzionale all'integrale dell'errore;
- $u_D = K_C T_D \frac{de}{dt}$, uno proporzionale alla derivata dell'errore.

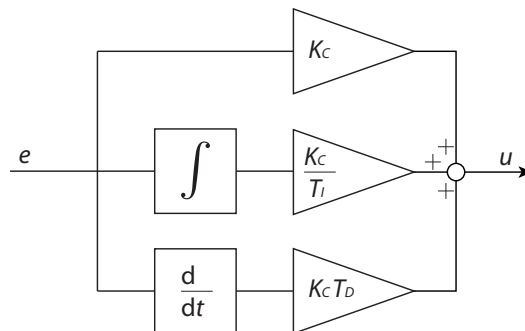


Figura 7.1. Schema a blocchi di un controllore PID.

Tale tipo di regolatore rappresenta una delle soluzioni maggiormente diffuse nel campo dell'automazione, principalmente per la sua efficacia a fronte di una grande affidabilità ed economicità, dovute alla grande diffusione e standardizzazione.

È ovviamente possibile realizzare controllori che presentino solo uno o due dei contributi sopra esposti, sono in particolare usati diffusamente soluzioni P, PD, PI.

Un'altro uso comune fatto dei controllori PID, è quello di utilizzarli come sotto-blocchi di uno schema più avanzato di controllo. Una delle possibili realizzazioni è quella del controllo in cascata, che è stata utilizzata per il sensore descritto e che verrà ora descritta.

7.1.2. Controllo in cascata.

Il controllo in cascata è uno schema avanzato di controllo che permette di migliorare la risposta al *set-point*, particolarmente se si è in presenza di un disturbo di carico.

Esso prevede di realizzare due anelli di controllo, uno interno all'altro.

L'anello più interno, o secondario, retroaziona una misura addizionale, indicata in Figura 7.2. come w , ed è chiuso da un regolatore R_2 . Il *set-point* di questo regolatore è invece l'uscita del regolatore R_1 che chiude l'anello di controllo esterno, o principale, e retroaziona la variabile controllata.

Per essere efficiente lo schema deve essere fatto in modo tale che la banda dell'anello di controllo interno sia sufficientemente più ampia di quella dell'anello esterno.

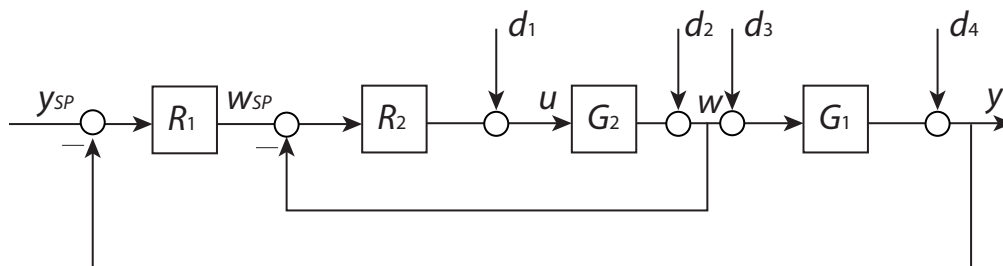


Figura 7.2. Schema del controllo in cascata.

7.1.3. Realizzazione in Simulink

Come visto anche nel paragrafo 2. 4. 2 a pagina 20 lo schema di controllo da implementare prevede un controllo in cascata della velocità, come misura addizionale, e della posizione, come variabile controllata.

Si prevede di utilizzare come regolatore più esterno un regolatore di tipo proporzionale, P, e nell'anello più interno un regolatore proporzionale integrativo, PI.

La scelta del regolatore PI per l'anello di velocità garantisce, grazie alla presenza dell'integratore, errore statico nullo a regime, anche per disturbi costanti come sono quelli generati dalle coppie gravitazionali.

P PI Controllo in cascata

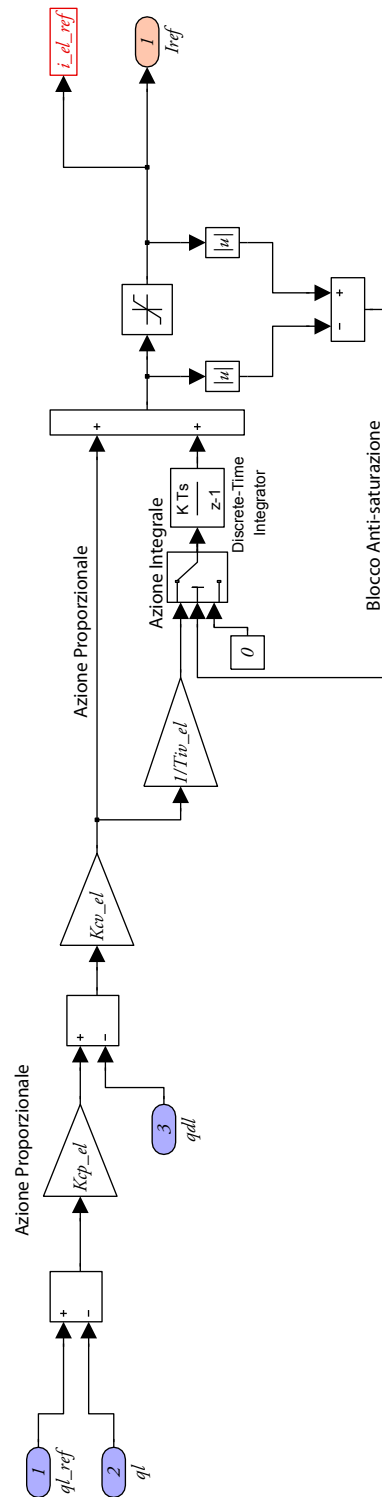


Figura 7.3. Schema interno del regolatore di posizione.

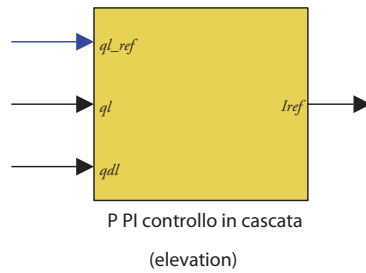


Figura 7.4. Controllore di posizione, per il grado di libertà di *elevation*, quello per l'*azimuth* è identico.

Il blocco del controllore ha quindi in ingresso tre variabili, che sono rispettivamente:

- Segnale di riferimento;
- Segnale di posizione misurata;
- Segnale di velocità misurata.

Internamente il blocco segue lo schema presentato in Figura 7.3 a pagina 105. L'ingresso di posizione viene comparato con il segnale di riferimento e controllato tramite un controllore proporzionale, andando a generare il *set-point* dell'anello più interno, di velocità, che è controllato da un regolatore PI, proporzionale integrativo

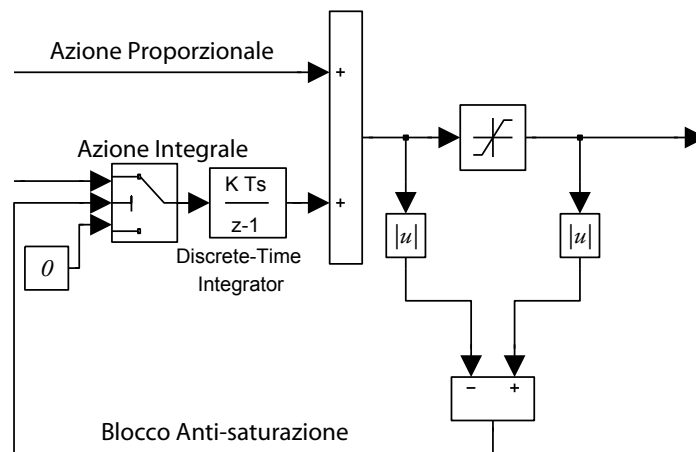


Figura 7.5. Blocchi anti saturazione, lo *switch* si attiva quando il segnale non è nullo.

Il controllore interno, PI, presenta un blocco anti-saturazione, per evitare i problemi collegati all'utilizzo di un'azione integrale. Il segnale prima del blocco di saturazione viene infatti confrontato, in valore assoluto, con il segnale dopo il blocco. Se la loro differenza non è nulla e quindi il segnale ha superato il valore di soglia viene attivato un blocco *Switch* che rende nullo il contributo dell'azione integrale.

7.2. Taratura dei regolatori

Per regolare il controllore PID vanno tarati i suoi tre parametri:

- $K_{C,p}$ guadagno proporzionale del regolatore esterno;
- $K_{C,v}$ guadagno proporzionale del regolatore interno;
- $T_{I,v}$ tempo integrale del regolatore interno.

La scelta di tali valori è stata fatta grazie ad alcune considerazioni che emergono analizzando i diagrammi di Bode del servomeccanismo.

Per eseguire l'analisi che è adesso riportata devono essere tuttavia fatte due ipotesi.

- *dinamica elettrica quasi-statica*, il motore elettrico ha una sua dinamica che è accoppiata a quella meccanica tramite la forza controelettrica.

Si deve tuttavia considerare che il controllore di corrente, di fatto un terzo anello ancora più interno, ha una banda molto più elevata rispetto a quelle ottenibili dagli anelli di posizione e velocità.

Questo permette di poter trascurare la dinamica e considerare compensato ogni effetto accoppiante della forza controelettrica.

Il motore può essere quindi considerato come un guadagno, di valore pari alla costante di coppia.

- *dinamica meccanica semplificata*, la dinamica del motore prevede degli accoppiamenti tra i due gradi di libertà, tuttavia essi sono di piccola entità.

Può quindi essere considerato un modello meccanico più semplice, che trascura anche l'attrito, in cui il motore è pari ad un integratore di guadagno pari all'inverso della sua inerzia attorno al suo asse di rotazione.

Tali ipotesi permettono di scrivere un modello molto semplificato del servomeccanismo, relativo ad un grado di libertà, la cui funzione di trasferimento è:

$$[7.2] \quad G_m = \frac{k_c/J_m}{s}.$$

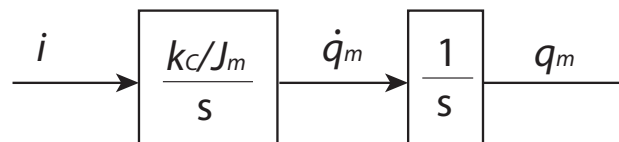


Figura 7.6. Schema semplificato del motore.

Si è quindi andati a regolare prima l'anello più interno, quindi l'anello principale di posizione.

7.2.1. Taratura dell'anello interno

La funzione di trasferimento del regolatore PI è la seguente:

$$\begin{aligned}
 [7.3] \quad G_{PI} &= K_{C,v} \left(1 + \frac{1}{sT_{I,v}} \right) \\
 &= K_{C,v} \frac{1 + sT_{I,v}}{sT_{I,v}}.
 \end{aligned}$$

Considerando il modello semplificato di motore elettrico espresso nell'equazione [7.3] si ottiene la seguente funzione di trasferimento d'anello.

$$[7.4] \quad L_v = G_{PI} G_m = K_{C,v} \frac{k_C}{J_m} \frac{1 + sT_{I,v}}{s}.$$

Se si considera un valore della costante di tempo integrale sufficientemente grande da poter considerare lo zero del regolatore a bassa frequenza si può approssimare la pulsazione critica prendendo l'approssimazione in alta frequenza di L_v

$$[7.5] \quad L_v \approx \frac{\omega_{c,v}}{s};$$

dove la pulsazione critica è quindi pari a:

$$[7.6] \quad \omega_{c,v} = K_{C,v} \frac{k_C}{J_m}$$

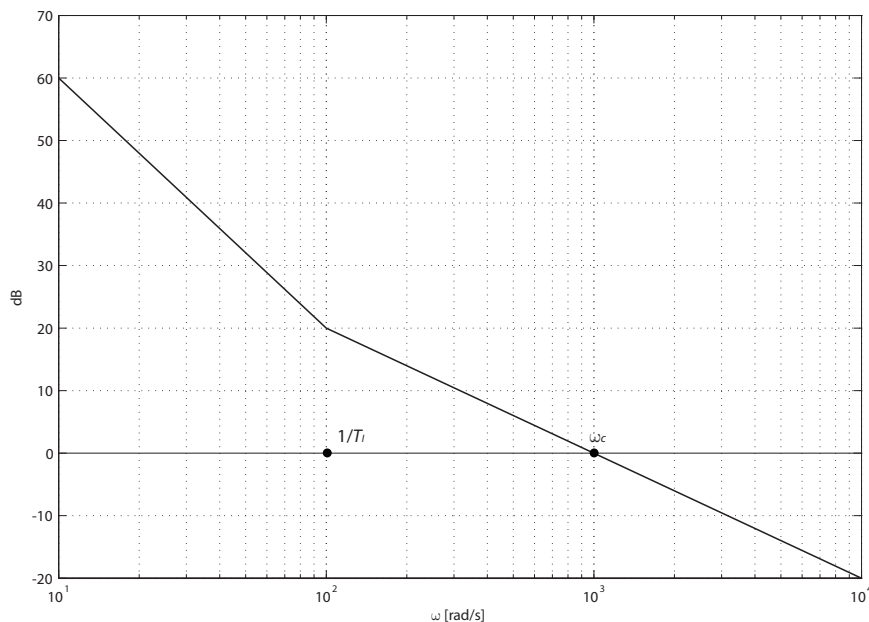


Figura 7.7. Diagramma di Bode del modulo di L_v .

Scelto quindi l'estremo superiore della banda passante, che coincide con la pulsazione critica si ha il valore del guadagno.

Per quanto invece riguarda il valore della costante di tempo integrale, si deve fare in modo che lo zero ad essa associato sia a frequenze più basse della pulsazione critica, si sceglie quindi generalmente un valore di:

$$[7.7] \quad T_{i,v} = (3 \div 10) / \omega_{c,v}.$$

7.2.2. Taratura dell'anello esterno

Per la taratura dell'anello esterno è possibile considerare il sistema costruito dall'anello interno, come un sistema del primo ordine, con funzione di trasferimento:

$$[7.8] \quad F_v = \frac{1}{1 + s/\omega_{c,v}}.$$

Sotto tale ipotesi l'anello di posizione è espresso dalla seguente funzione di trasferimento:

$$[7.9] \quad L_p = K_{C,p} F_v \frac{1}{s} = \frac{K_{C,p}}{s(1 + s/\omega_{c,v})}.$$

Il diagramma di Bode di tale funzione di trasferimento è rappresentato nella figura seguente.

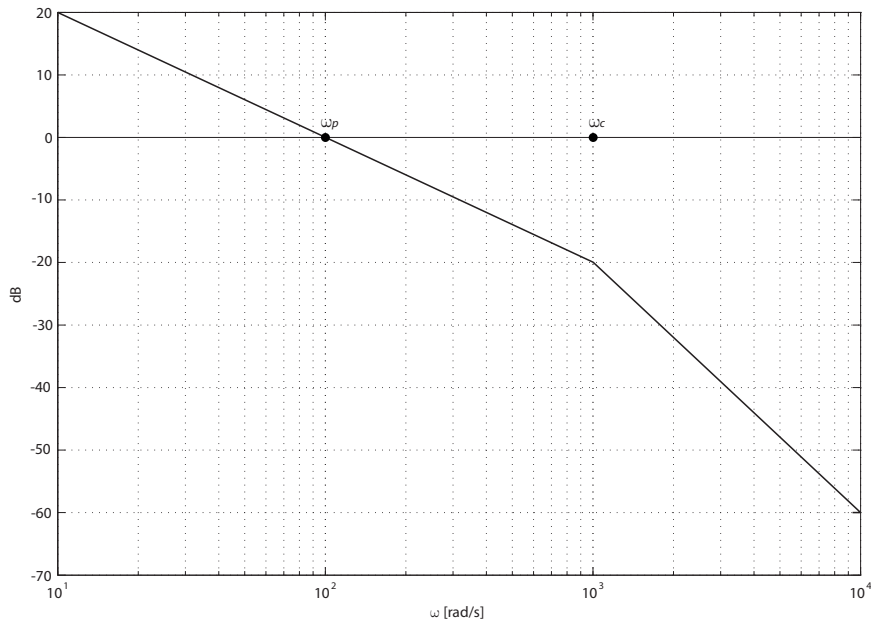


Figura 7.8. Diagramma di Bode del modulo di L_p .

Considerando quindi che la banda passante dell'anello di velocità è maggiore di quella dell'anello di corrente, si ottiene che

$$[7.10] \quad \omega_{c,p} = K_{C,p}.$$

Si sono scelti i seguenti valori di pulsazione, uguali per entrambi i gradi di libertà:

$$\omega_{c,p} = 50; \quad \omega_{c,v} = 500.$$

Si è inoltre scelto per entrambi:

$$T_{l,v} = 0.2.$$

Considerando le caratteristiche del motore si ottengono quindi i seguenti parametri del regolatore.

- per l'*azimuth*

$$K_{C,p} = 50; \quad K_{C,v} = 6.9; \quad T_{l,v} = 0.02.$$

- per l'*elevation*

$$K_{C,p} = 50; \quad K_{C,v} = 2.4; \quad T_{l,v} = 0.02.$$

7.3. Confronto con i dati sperimentali e verifica delle prestazioni

Per verificare questi valori sono stati presi in considerazione due criteri, da una parte rispettare i vincoli forniti da SELEX Galileo, dall'altra ottenere una banda confrontabile e se possibile superiore a quella ottenuta dalle prove sperimentali.

Tabella 7.1. Caratteristiche obiettivo del sistema di controllo.

	Valore
Banda ad anello chiuso	> 6Hz
Profilo di comando con <i>step</i> di 5°.	
Verifica della massima coppia raggiungibile	
Errore di velocità	< 4°/s
Errore di posizione	< 35 μrad
Tempo di mantenimento del <i>set-point</i>	> 500 μs

7.3.1. Confronto con i dati sperimentali

Si è confrontato quanto ottenuto sperimentalmente con i ottenuti dalla simulazione, per degli scalini di ampiezza pari a 10°, 20° e 40°.

Come si nota i risultati sono abbastanza simili, avendo dei tempi di risposta comparabili.

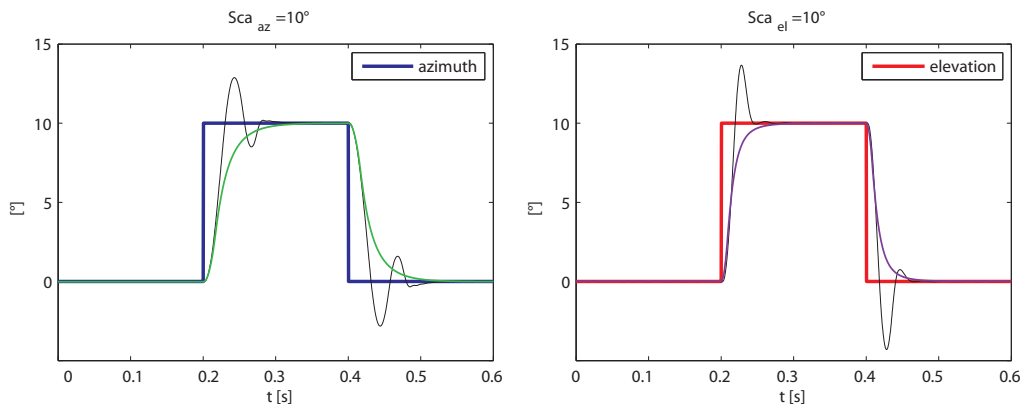


Figura 7.9. Andamenti dei segnali di posizione, in simulazione, verde e viola, confrontato con i dati sperimentali, in linea nera.

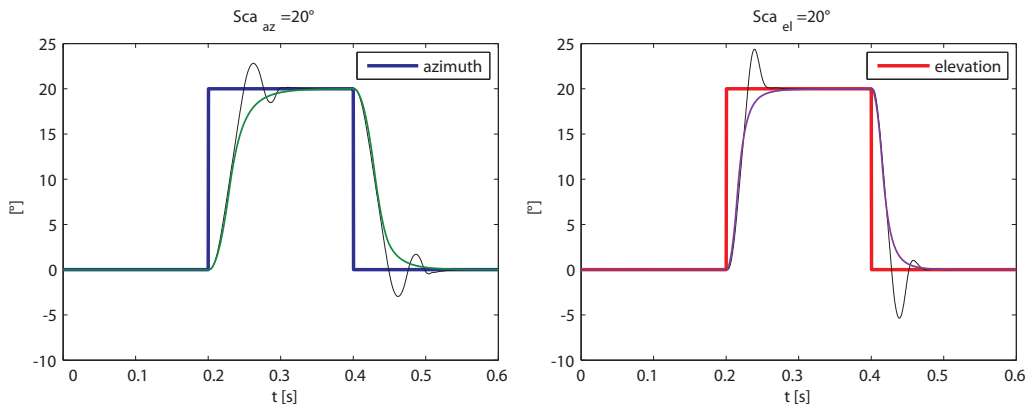


Figura 7.10. Andamenti dei segnali di posizione, in simulazione, verde e viola, confrontato con i dati sperimentali, in linea nera.

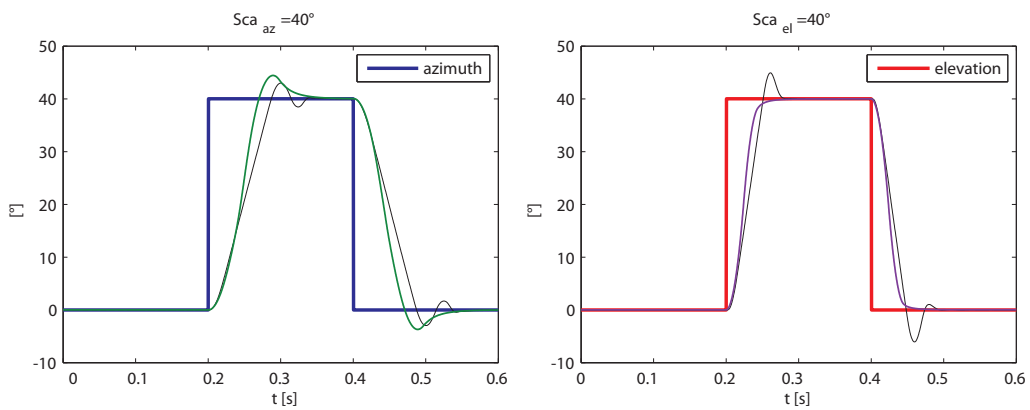


Figura 7.11. Andamenti dei segnali di posizione, in simulazione, verde e viola, confrontato con i dati sperimentali, in linea nera.

7.3.2. Verifica delle prestazioni

Le prestazioni richieste sono soddisfatte in termini di banda passante, considerando infatti che la pulsazione critica è pari a 50 rad/s, poco meno di 8 Hz.

Andando invece a simulare delle rampe di pendenza $200^\circ/\text{s}$ e $400^\circ/\text{s}$ si nota come, particolarmente a 400° siano presenti delle oscillazioni, che risultano essere dovute ai disturbi. Tali oscillazioni, particolarmente per l'*azimuth* hanno un'ampiezza elevata.

Per tale motivo si è ipotizzato di implementare uno schema di controllo aggiuntivo, l'osservatore del disturbo di coppia, descritto nella sezione successiva.

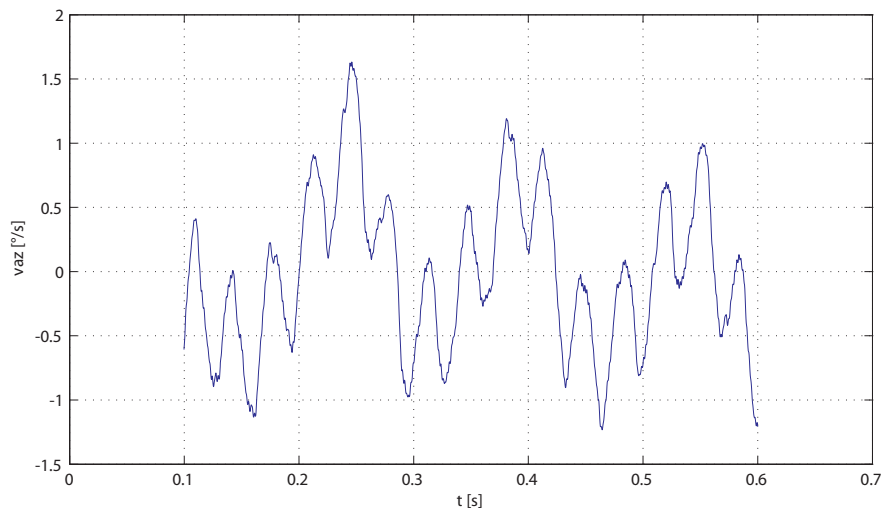


Figura 7.12. Andamento nel tempo della velocità di *azimuth*, decurtata del valore medio, $\omega = 200^\circ/\text{s}$.

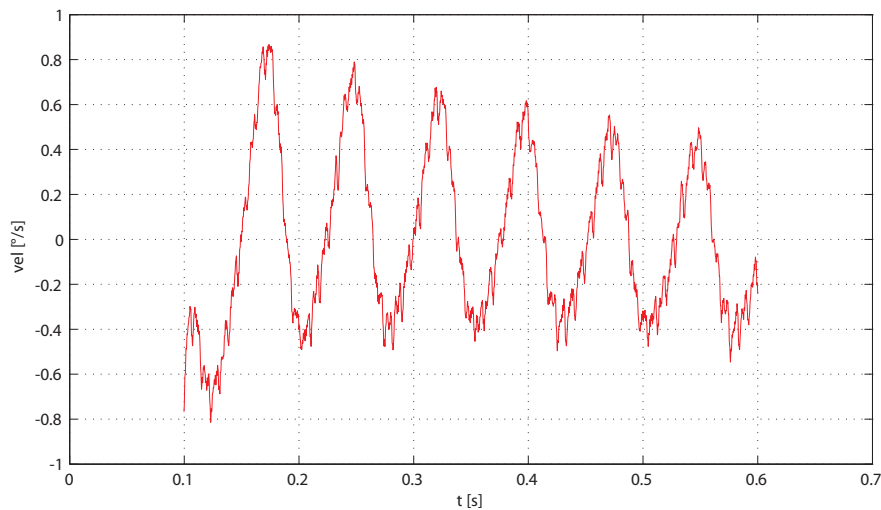


Figura 7.13. Andamento nel tempo della velocità di *elevation*, decurtata del valore medio, $\omega = 200^\circ/\text{s}$.

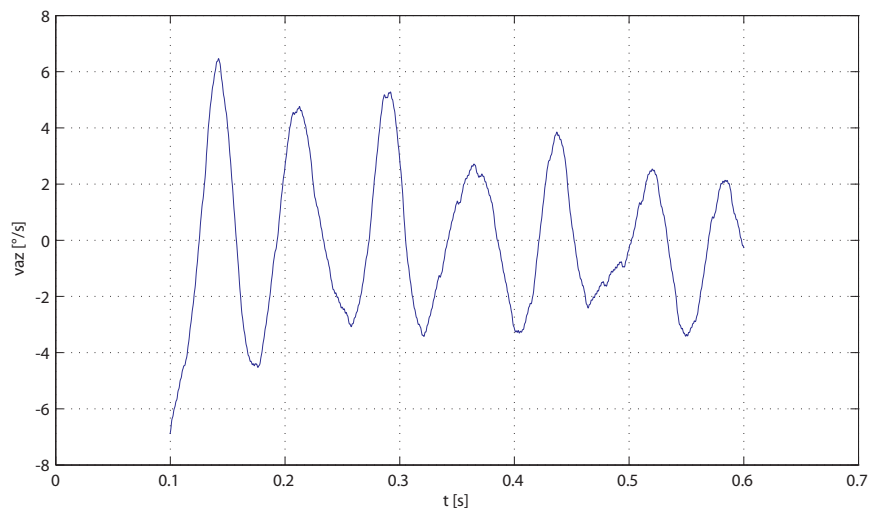


Figura 7.14. Andamento nel tempo della velocità di *azimuth*, decurtata del valore medio, $\omega = 400^\circ/\text{s}$.

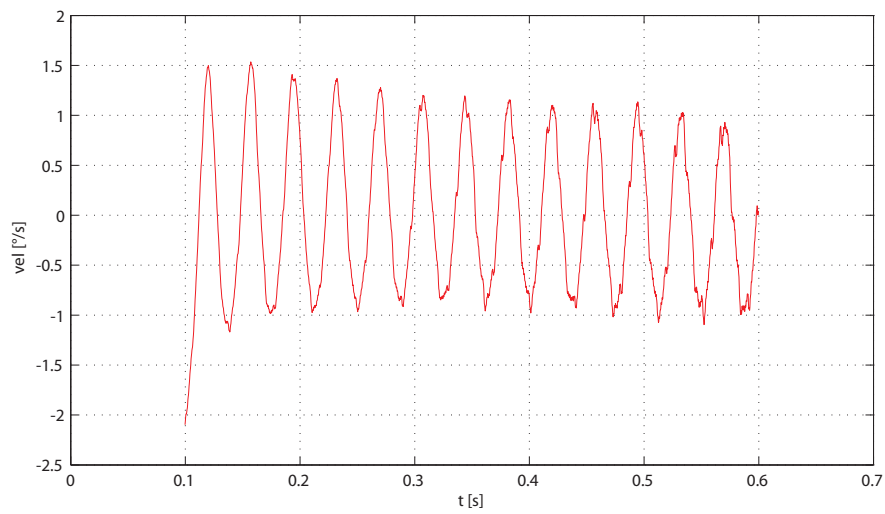


Figura 7.15. Andamento nel tempo della velocità di *elevation*, decurtata del valore medio, $\omega = 400^\circ/\text{s}$.

7.4. Osservatore del disturbo di coppia

Per aumentare la capacità di reiezione dei disturbi del sistema, si è implementato nel modello *Simulink* uno schema che stimasse opportunamente la coppia di disturbo e la compensasse. Tale schema è noto in letteratura con il nome di osservatore del disturbo di coppia, spesso indicato con il suo acronimo inglese TDO.

7.4.1. Descrizione generale*

L'idea fondamentale del TDO è quella di progettare un sistema che, dato il generico sistema dinamico rappresentato in Figura 7.16, riesca a ricostruire una stima del disturbo, sulla base di u e delle misure y .

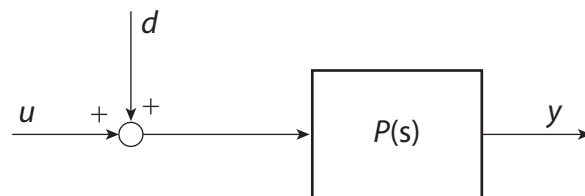


Figura 7.16. Schema di un generico sistema dinamico a cui è applicato un disturbo di carico.

Ipotizzando che la funzione di trasferimento $P^{-1}(s)$ fosse realizzabile, dando in ingresso ad essa il segnale y si otterrebbe ovviamente il segnale in ingresso del sistema $P(s)$, che opportunamente confrontato con u , darebbe il valore del disturbo.

Si deve tuttavia tenere in considerazione che in un caso reale non è nota la funzione di trasferimento del sistema, quanto piuttosto un modello nominale, $P_n(s)$ di esso ed inoltre $P_n^{-1}(s)$ non è realizzabile.

Questo porta alla costruzione dello schema rappresentato in Figura 7.17 a pagina 115, in cui il segnale y è l'ingresso della funzione di trasferimento $Q(s)P_n^{-1}(s)$ con $Q(s)$ funzione di trasferimento di un filtro passabasso di guadagno unitario e grado relativo sufficientemente elevato da rendere $Q(s)P_n^{-1}(s)$ realizzabile.

L'uscita di tale funzione di trasferimento è quindi confrontata con il valore di u , filtrato tuttavia da $Q(s)$.

* La trattazione presentata riprende in parte quanto esposto in [8] G. Magnani; "Tecnologie dei sistemi di controllo," McGraw-Hill, Marzo 2000.

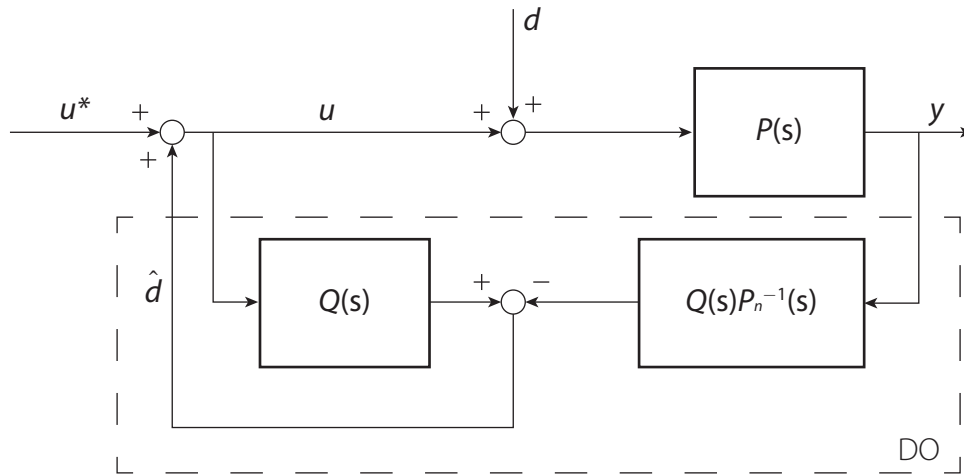


Figura 7.17. Schema di un osservatore del disturbo.

La stima del disturbo \hat{d} è espressa dalla seguenti equazioni:

$$\begin{aligned}
 \hat{d} &= Q(s)u - P_n^{-1}(s)Q(s)y = \\
 [7.1] \quad &= Q(s)u - P_n^{-1}(s)Q(s)P(s)(d + u) = \\
 &= Q(s)\left(1 - P_n^{-1}(s)P(s)\right)u - Q(s)P_n^{-1}(s)P(s)d.
 \end{aligned}$$

Se il sistema nominale è una buona approssimazione del sistema reale si può considerare $P_n(s) \approx P(s)$, tale ipotesi porta a semplificare l'equazione [7.1], ottenendo quindi:

$$[7.2] \quad \hat{d} \approx Q(s)d.$$

L'uscita del sistema, y , è invece descritta dalla seguente equazione:

$$[7.3] \quad y = \frac{P(s)(1 - Q(s))}{1 - Q(s) + P_n^{-1}(s)P(s)Q(s)}d + \frac{P(s)}{1 - Q(s) + P_n^{-1}(s)P(s)Q(s)}u^*.$$

Considerando $P_n(s) \approx P(s)$ l'equazione diventa:

$$[7.4] \quad y = P(s)\left(1 - Q(s)\right)d + P(s)u^*.$$

Andando ad analizzare quindi le equazioni [7.2] e [7.4] si può concludere che nella banda passante del filtro $Q(s)$ il disturbo è correttamente stimato e, conseguentemente, il sistema ha una reiezione totale del disturbo.

Nel caso di un motore il sistema sotto controllo è rappresentabile dalla funzione di trasferimento tra la corrente e la velocità:

$$[7.5] \quad P(s) = \frac{k_t}{J_m s + D_m}.$$

Supponendo che il modello nominale coincida con il modello vero è sufficiente adottare un filtro del primo ordine:

$$[7.6] \quad Q(s) = \frac{1}{T_f s + 1}.$$

Considerando le funzioni di trasferimento [7.5] e [7.6] il sistema rappresentato in Figura 7.17 a pagina 115 si riduce allo schema mostrato in Figura 7.18.

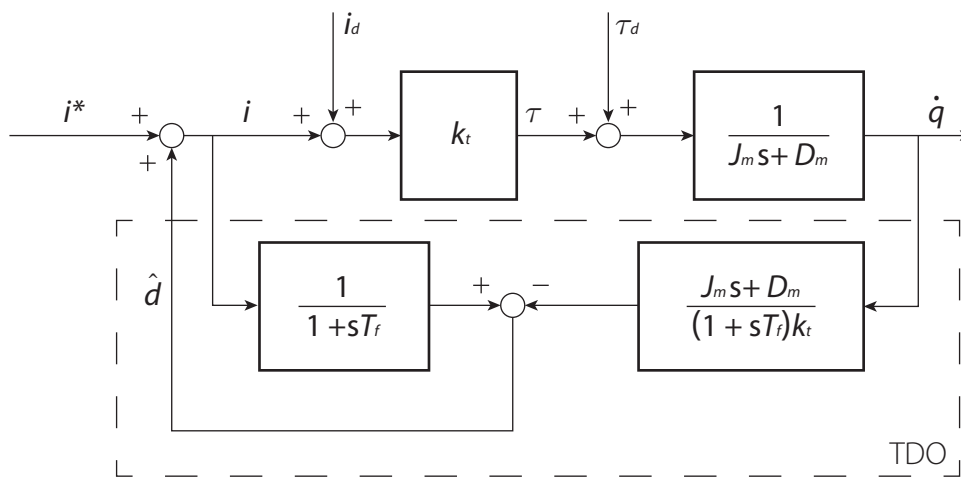


Figura 7.18. Schema dell'osservatore del disturbo di coppia.

Tale schema può essere ulteriormente rielaborato, al fine di ottenere uno schema che faccia uso solo di un sistema dinamico del primo ordine. Lo schema che si ottiene è rappresentato in Figura 7.19 a pagina 117.

Poiché, come già osservato, l'osservatore riesce a filtrare efficacemente i disturbi di coppia entro la banda passante di $Q(s)$, se ne deduce che la reiezione è tanto più efficace quanto più è piccola la costante di tempo T_f .

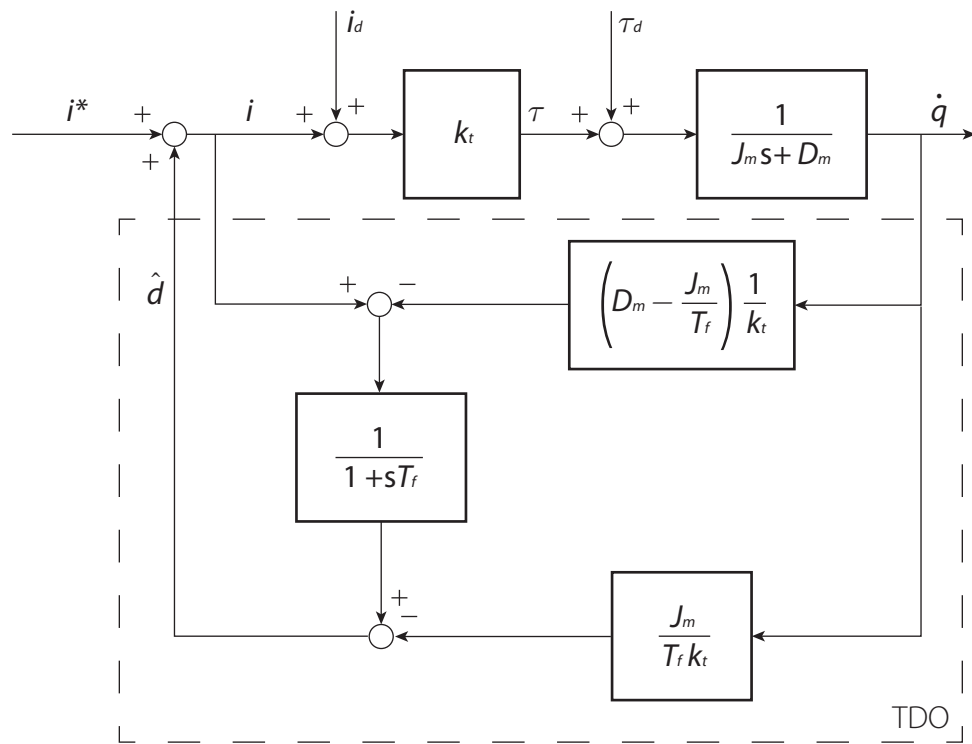


Figura 7.19. Schema dell'osservatore del disturbo di coppia.

7.4.2. Modello Simulink

La modellazione in *Simulink* riprende del tutto lo schema presentato in Figura 7.19. Trattandosi tuttavia di un controllore digitale viene aggiunto un ritardo ed il filtro non è continuo ma digitale.

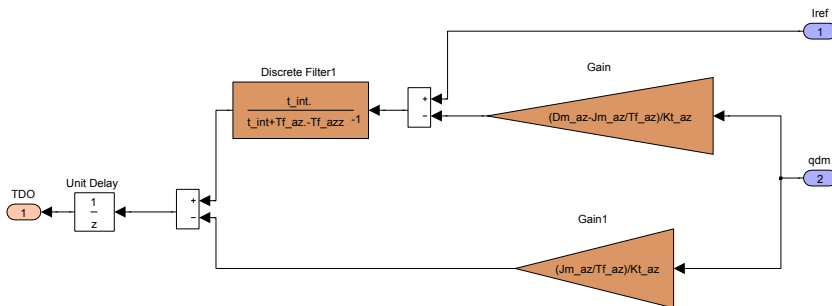


Figura 7.20. Implementazione *Simulink* del TDO.

Simulazione Sensore IRST

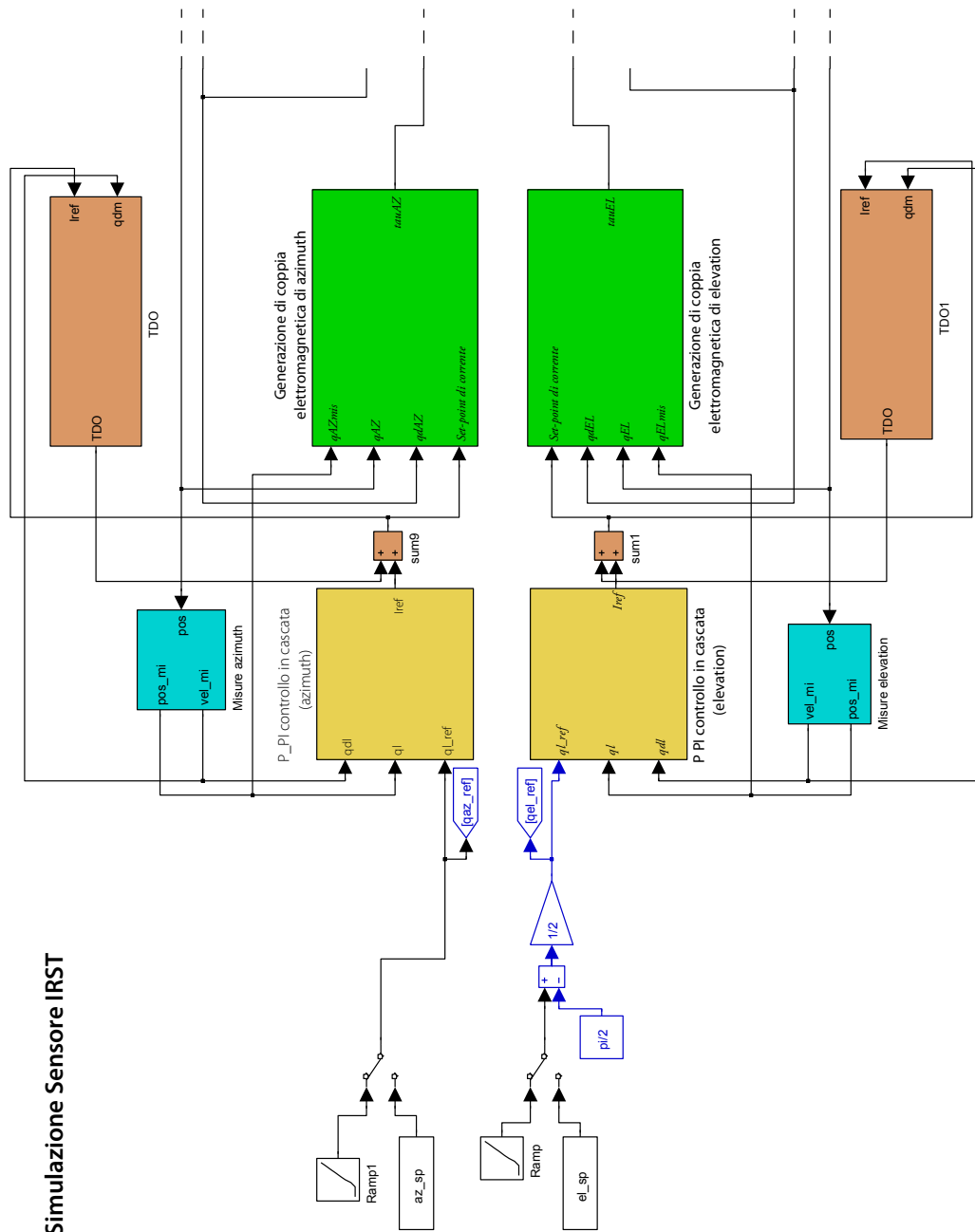


Figura 7.21. Modello del sistema con i due *Torque Disturbance Observer*, in marrone, la seconda parte del modello rimane invariata ed è rappresentata in Figura 2.7 a pagina 17.

7.4.3. Verifica finale delle prestazioni

Per verificare l'effetto del TDO sono state eseguite le stesse prove descritte nel paragrafo 2.1.

Come è possibile notare dalle figure seguenti si ha una reiezione molto migliore dei disturbi. Inoltre la specifica di errore di velocità minore di $4^\circ/s$ è sempre verificata, anche per le velocità maggiori

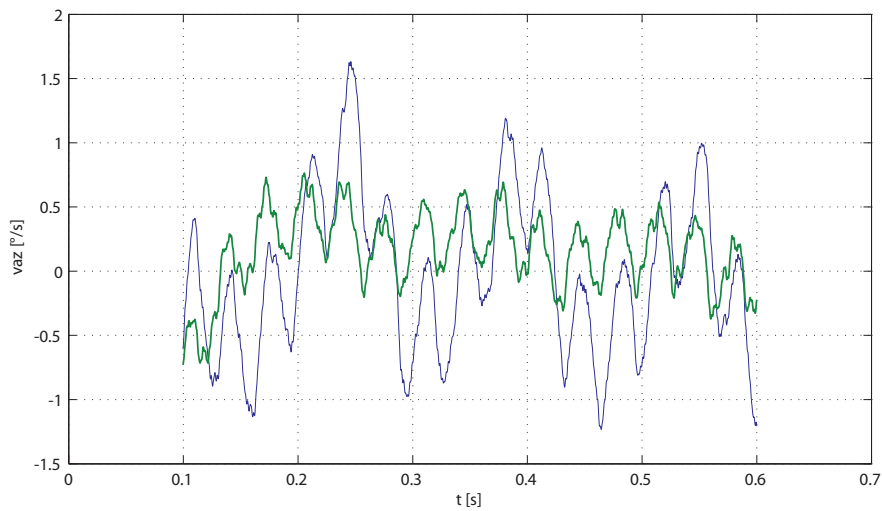


Figura 7.22. Andamento nel tempo della velocità di *azimuth*, decurtata del valore medio, $\omega = 200^\circ/s$. In verde il segnale con TDO, in blu senza.

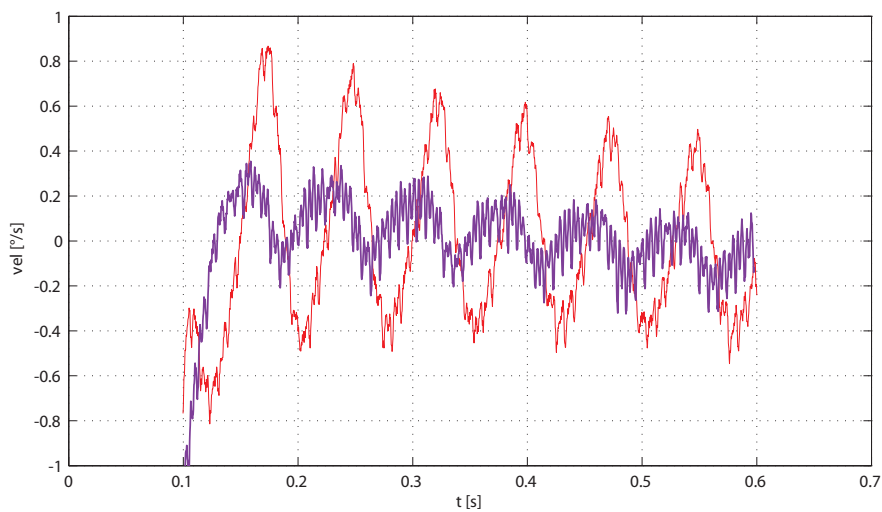


Figura 7.23. Andamento nel tempo della velocità di *elevation*, decurtata del valore medio, $\omega = 200^\circ/s$.

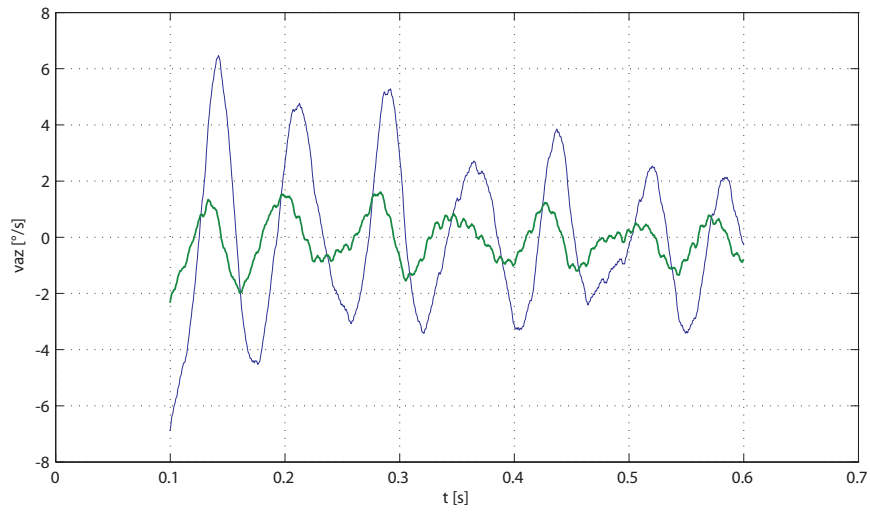


Figura 7.24. Andamento nel tempo della velocità di *azimuth*, decurtata del valore medio, $\omega = 400^\circ/\text{s}$. In verde il segnale con TDO, in blu senza.

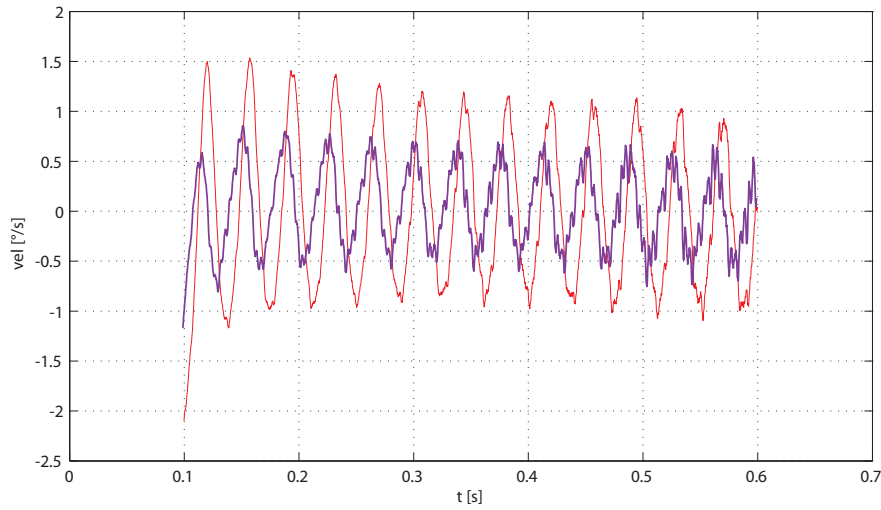


Figura 7.25. Andamento nel tempo della velocità di *elevation*, decurtata del valore medio, $\omega = 400^\circ/\text{s}$.

Capitolo 8

Interfaccia grafica di controllo

Il modello *Simulink* che è stato descritto nei capitoli precedenti, richiede un certo numero di parametri per poter funzionare correttamente. Inoltre devono essere eseguite alcune necessarie operazioni come la generazione di un opportuno *set-point*.

Questo ha portato all'esigenza di costruire un'interfaccia grafica che, seppur semplice, rendesse possibile interagire con il modello in modo rapido.

Ad essa è anche stato possibile aggiungere alcuni moduli per la visualizzazione dei risultati durante e dopo la elaborazione.

In questo capitolo verrà descritto il funzionamento dell'interfaccia e le sue funzioni.

8.1. Descrizione generale

L'interfaccia si compone principalmente di:

- Menu composti da diversi bottoni;
- Finestre di inserimento di valori;
- Finestre con grafici.

Il menu principale consente di andare al menu di simulazione o a quelli relativi alle funzioni di *post processing* dei dati, di esecuzione della FFT e dei grafici.

Uno schema generico del funzionamento dell'interfaccia è presente nella figura seguente.

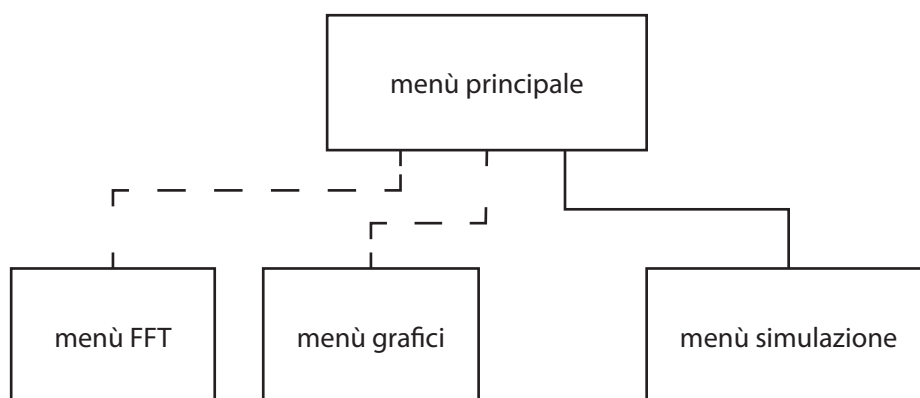


Figura 8.1. Schema generale dell'interfaccia grafica, la linea che collega i menu FFT e grafici è tratteggiata perché la possibilità di accedervi è subordinata all'aver eseguito una simulazione.

Tutti i menu inferiori a quello principale permettono di tornare indietro e da quello principale può essere chiusa l'interfaccia.

Andremo ora a descrivere ed analizzare nel dettaglio i singoli menu.

8.2. Menu principale

Il menu principale si presenta diversamente a seconda se è già stata eseguita una simulazione o se invece è la prima volta che si apre l'interfaccia.

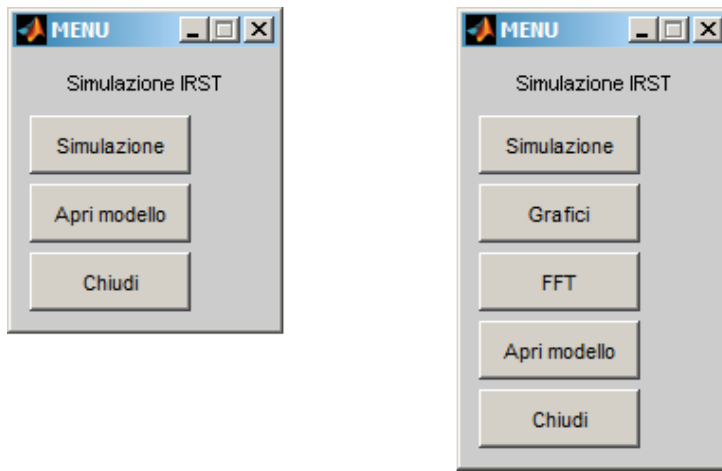


Figura 8.2. Due differenti aspetti del menu principale, prima di una simulazione, a sinistra, e dopo una simulazione, a destra.

In tal caso i pulsanti possibili sono solo:

- *Simulazione* consente di aprire il menu simulazione, che verrà descritto in seguito;
- *Apri modello* apre la finestra di *Simulink* con il modello;
- *Chiudi* chiude l'interfaccia ed eventuali finestre aperte.

Nel caso invece in cui sia già stata eseguita una simulazione e siano quindi disponibili i risultati da essa prodotti, sono invece anche presente due ulteriori pulsanti:

- *Grafici*;
- *FFT*.

Tali pulsanti aprono i menu e le finestre grafiche relative, il primo, alla trasformata di Fourier dei dati ed il secondo alla rappresentazione del loro andamento nel tempo.

Le funzioni di questi menu saranno comunque maggiormente approfondite nella sezione ad esse relative.

8.3. Menu simulazione

Questo menu è quello che permette di far partire la simulazione del modello *Simulink* e attraverso il quale possono essere modificati alcuni parametri reattivi alla simulazione stessa.

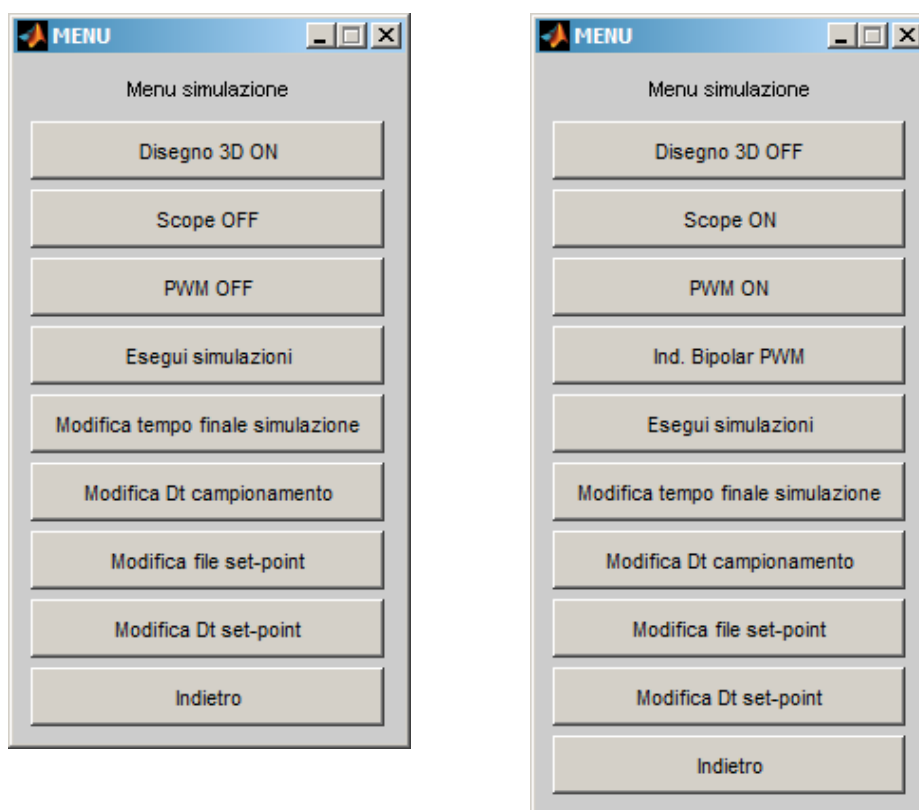


Figura 8.3. Rappresentazione del menu principale, che mostra la possibilità di attivare e disattivare alcune funzioni e vederne lo stato nel nome del pulsante; si noti che se il pulsante *PWM* è su ON è presente un ulteriore pulsante.

I primi tre pulsanti sono:

- *Disegno 3D ON* [*Disegno 3D OFF*];
- *Scope OFF* [*Scope ON*];
- *PWM OFF* [*PWM ON*].

Essi consentono l'attivazione o la disattivazione di particolari funzioni ed il loro stato corrente viene mostrato nel pulsante stesso.

Di *default* sono impostati come riportato nell'elenco puntato, non tra parentesi quadre.

I pulsanti *Scope* e *Disegno 3D* permettono la visualizzazione, durante la simulazione, di opportune finestre grafiche che mostrano l'andamento della simulazione stessa, generate dalla *S-Function* relative, come descritto nel paragrafo 2. 4. 7. a pagina 24.

Il pulsante *PWM* invece permette di simulare o meno la modulazione della corrente tramite PWM. Tale simulazione rende infatti il modello più corretto ma contemporaneamente più lento nella simulazione.

Se attivato compare un secondo pulsante che cambia a rotazione il suo stato se premuto e permette di scegliere il tipo di modulazione PWM da simulare.

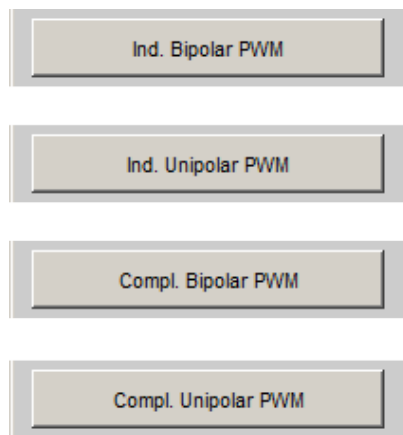


Figura 8.4. Pulsanti corrispondenti alle diverse tipologie di modulazione PWM che è possibile simulare.

Il pulsante *Esegui simulazioni* fa partire la simulazione del modello in *Simulink*, al termine della quale si riapre il menu principale.

Sono presenti quindi i seguenti pulsanti:

- *Modifica tempo finale simulazione*, permette di impostare un tempo finale di simulazione, aprendo la finestra di inserimento dati relativa. Viene eseguito un controllo affinché il tempo immesso non sia nullo o negativo.

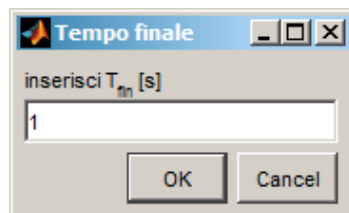


Figura 8.5. Finestra di inserimento del valore di tempo finale della simulazione.

- *Modifica Dt campionamento*, modifica il tempo di campionamento dei dati in

uscita. Anche in questo caso viene eseguito un controllo, affinché il tempo non sia nullo o negativo, viene tuttavia permesso l'inserimento del valore -1 che corrisponde al campionamento di tutti i valori, anche se non equidistanziati temporalmente.

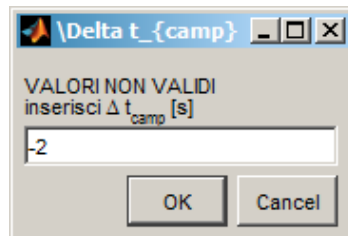


Figura 8.6. Finestra di inserimento del valore di tempo di campionamento, si noti il messaggio di errore poiché si è inserito un valore negativo, non valido.

- *Modifica file set-point*, di *default* vengono presi come *file* per la generazione del riferimento i file *az.txt* e *e1.txt* presenti nella cartella del modello, tramite questo pulsante vengono tuttavia aperte delle finestre da cui si può selezionare dei file diversi. I file devono contenere un elenco sequenziale con i valori di *set-point* e devono avere lo stesso numero di elementi. Su queste specifiche non viene fatto un controllo e questo, se i *file* inseriti non sono corretti, potrebbe portare a degli errori in fase di simulazione.
- *Modifica Dt set-point*, i *file* relativi al riferimento non hanno nessuna indicazione temporale, deve essere quindi fornito separatamente l'intervallo di tempo tra due campioni. Il valore può essere opportunamente modificato tramite questo bottone, che apre una finestra di selezione.
- *Indietro*, permette di tornare indietro al menu principale.

8.4. Menu grafici

Una volta eseguita una simulazione, l'interfaccia torna sul menu principale, dove tuttavia c'è la possibilità di selezionare i pulsanti relativi ai grafici ed alla FFT.

Selezionando il pulsante *Grafici* si apre una finestra che presenta un piano cartesiano vuoto, in alternativa se durante la simulazione si era attivata la opzione *Scope ON* la finestra di *Scope* viene selezionata e ripulita dei grafici visualizzati.

Oltre alla finestra grafica è anche presente il Menu grafici che mostra una serie di pulsanti relativi ad alcune grandezze di interesse per la simulazione.

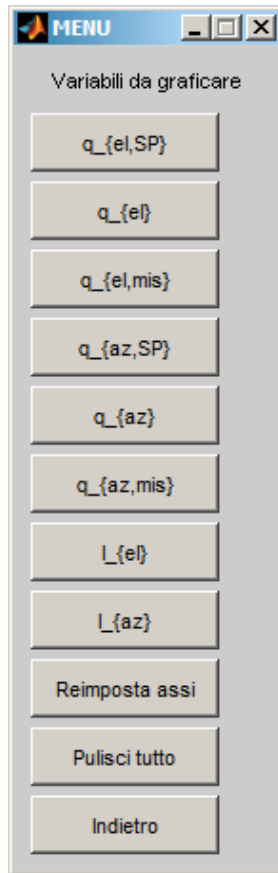


Figura 8.7. Menu relativo ai grafici.

La selezione di uno di questi pulsanti permette di far apparire sulla finestra il grafico corrispondente e la legenda.

Oltre ai pulsanti relativi alle grandezze che possono essere rappresentate ci sono anche i seguenti pulsanti:

- *Reimposta assi*, apre una maschera di selezione dove possono essere inseriti dei diversi estremi sugli assi delle ascisse e delle ordinate. Vengono eseguiti inoltre dei controlli che non permettono di inserire valori non validi.
- *Pulisci tutto*, permette di riportarsi alla situazione pulita, senza nessun grafico e con gli estremi degli assi di *default*.
- *Indietro*, permette di andare indietro al menu principale.

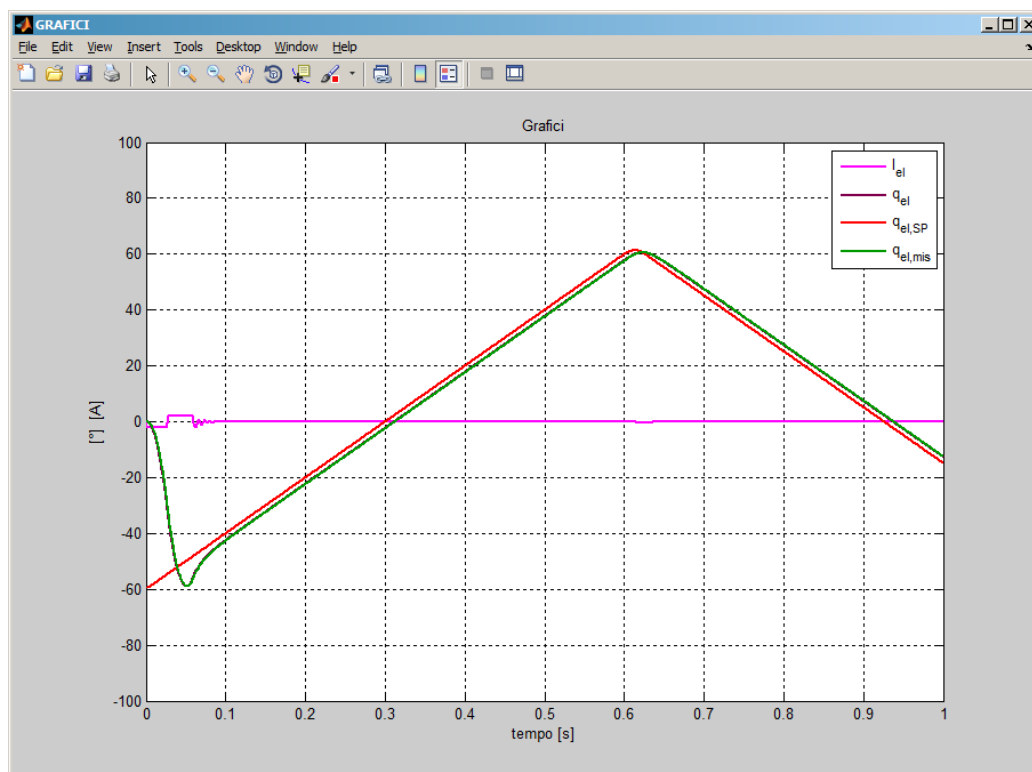


Figura 8.8. Esempio di una finestra dei grafici con in evidenza le grandezze relative all'elevation.

The figure shows a dialog box titled "limiti assi" with four input fields for axis limits: x_{min} (0), x_{max} (1), y_{min} (-100), and y_{max} (100). There are "OK" and "Cancel" buttons at the bottom.

Figura 8.9. Finestra di inserimento dei limiti per gli assi x ed y del grafico.

8.5. Menu FFT

Prima di introdurre il menu FFT si deve considerare che la operazione di FFT può essere eseguita solo su un intervallo equispaziato nel tempo di valori, a meno di non utilizzare algoritmi modificati.

Per tale motivo se i dati non sono stati campionati ad intervalli di tempo regolari, il pulsante FFT restituisce una finestra di errore e torna al menu principale.

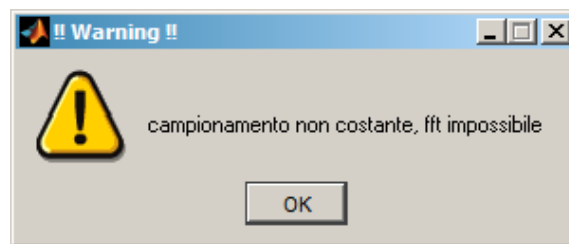


Figura 8.10. Finestra di errore che compare se si cerca di accedere al menu FFT anche se si è scelto un passo di campionamento non costante.

Nel caso in cui invece i dati siano campionati in modo regolare, il pulsante di FFT apre un menu intermedio che permette di selezionare il grado di libertà che si desidera analizzare.

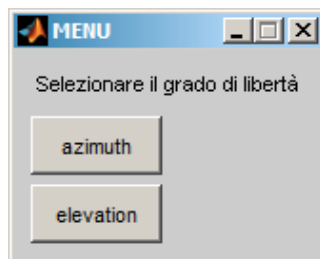


Figura 8.11. Finestra di selezione del grado di libertà che verrà analizzato.

Una volta selezionata la grandezza in analisi, viene aperta una nuova finestra grafica ed il menu dell'FFT vero e proprio.

La finestra grafica presenta tre piani cartesiani, disposti verticalmente.

Nel primo viene rappresentato l'andamento nel tempo della grandezza scelta e, tramite un colore differente ed una linea più marcata, viene evidenziato un tratto che è quello su cui si va ad effettuare la trasformata veloce di Fourier.

La velocità media relativa al tratto selezionato viene riportata nel titolo.

Nel secondo piano viene rappresentato l'andamento temporale dell'intervallo evidenziato nel grafico superiore, decurtato della parte costante e di quella lineare.

Nell'ultima finestra, infine, viene mostrato lo spettro del tratto selezionato, viene inoltre mostrato il picco massimo, tramite un colore differente. La frequenza e l'ampiezza del picco massimo vengono riportati nel titolo.

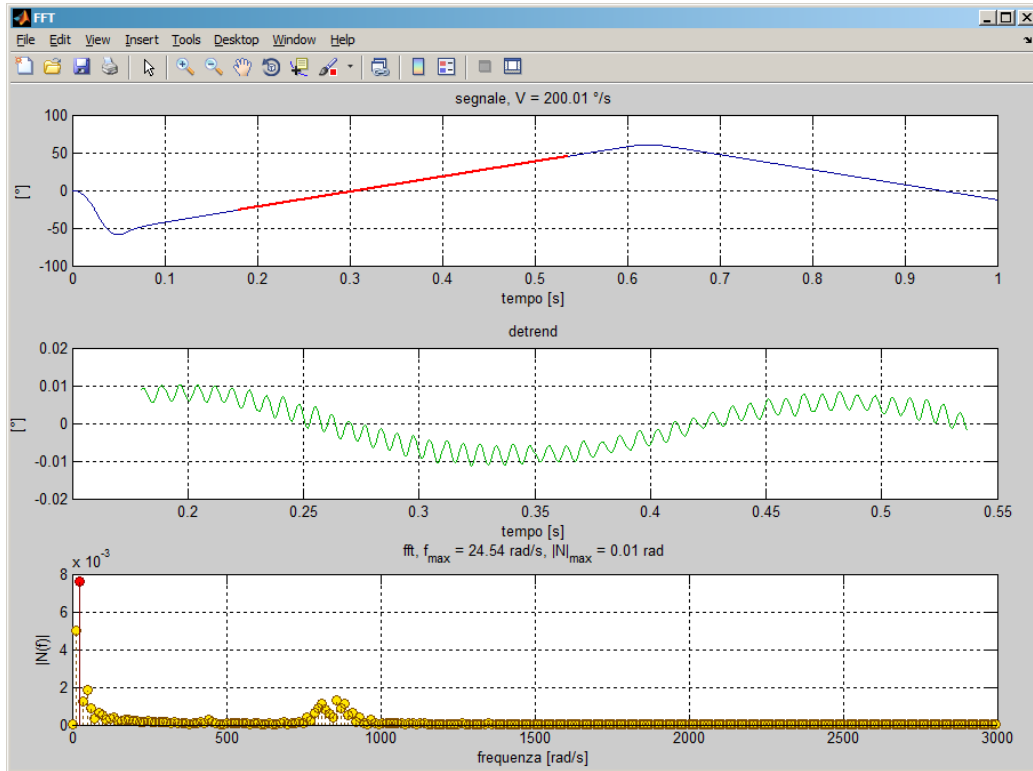


Figura 8.12. Esempio di una finestra relativa al menu FFT.

Il menu FFT permette di modificare opportunamente sia l'intervallo che alcune opzioni relative al grafico delle frequenze .

Sono in particolare presenti i pulsanti:

- <- sposta a destra l'intervallo;
- -> sposta a sinistra l'intervallo;
- <= sposta a sinistra l'intervallo di una quantità pari all'intervallo stesso, l'istante finale del nuovo intervallo è l'istante iniziale di quello vecchio;
- => sposta a destra l'intervallo di una quantità pari all'intervallo stesso, l'istante iniziale del nuovo intervallo è l'istante finale del vecchio;
- + aumenta le dimensioni dell'intervallo;
- - diminuisce le dimensioni dell'intervallo;



Figura 8.13. Menu relativo all'FFT.

- ++ raddoppia le dimensioni dell'intervallo;
- -- dimezza le dimensioni dell'intervallo;
- *Inserisci estremi intervallo*, permette di inserire tramite una maschera di inserimento dati gli estremi dell'intervallo da selezionare. Per tutti i tasti precedenti vengono eseguiti dei controlli affinché l'intervallo di dati sia sempre valido, non venga scelto un intervallo nullo o si superi la dimensione del vettore di dati stesso.

Oltre al pulsante *Indietro* è anche presente un ulteriore pulsante:

- *Inserisci estremi frequenze*, viene aperta una maschera di inserimento in cui possono essere inserite gli estremi dell'intervallo di frequenze che si desidera analizzare e quindi in pratica i limiti dell'asse x dell'ultimo piano cartesiano. Tale modifica non ha alcun effetto sulla risoluzione in frequenza che è legata

alla lunghezza dell'intervallo di tempo selezionato. Vengono fatti opportuni controlli che non sia superata la frequenza di *Shannon* e che non si immettano estremi non validi.

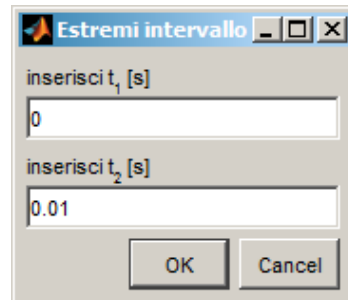


Figura 8.14. Finestra di inserimento dei limiti temporali della selezione.

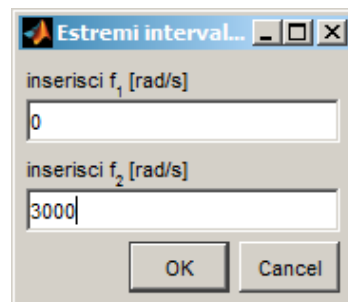


Figura 8.15. Finestra di inserimento dei limiti delle frequenze.

Considerazioni finali e sviluppi futuri

Il lavoro svolto ha permesso la creazione di un modello abbastanza dettagliato del sensore. Si è inoltre riuscito a rendere facilmente modificabili i parametri di funzionamento della simulazione e del modello stesso, grazie alla creazione di un'interfaccia grafica.

Dal punto di vista della schematizzazione il modello può essere considerato quindi sufficientemente completo.

Si è proceduto ad una prima taratura dei regolatori, con l'implementazione di un osservatore del disturbo di coppia. Lo schema di controllo è risultato efficace, ma i parametri scelti, anche se danno buoni risultati e possono essere un punto di partenza per future tarature, non sono definitivi. Questo dal momento che manca una adeguata solidità per quanto riguarda alcuni valori caratteristici dei sotto blocchi del modello.

Si deve infatti considerare che alcuni componenti, come ad esempio i *resolver*, potrebbero in futuro essere variati e, inoltre, alcune grandezze, come la frequenza del PWM, non sono state ancora definite.

Si può ipotizzare quindi di usare il modello come strumento di analisi per la ricerca di soluzioni ottimali per la scelta dei componenti e dei valori caratteristici degli stessi.

Oltre che come aiuto alla progettazione del sistema di controllo, si può prevedere di utilizzare il modello anche per caratterizzare il comportamento del sensore al variare di alcuni parametri, tramite simulazioni statistiche Monte Carlo.

Indice delle tabelle

<i>Tabella 3.1.</i>	Differenti sistemi di riferimento.	29
<i>Tabella 4.1.</i>	Caratteristiche del motore di <i>azimuth</i>	47
<i>Tabella 4.2.</i>	Caratteristiche del motore di <i>elevation</i>	47
<i>Tabella 4.3.</i>	Possibili tecniche di modulazione PWM.	54
<i>Tabella 5.1.</i>	Prove da eseguire sul servomeccanismo.	64
<i>Tabella 5.2.</i>	Velocità eseguite nei diversi esperimenti.	70
<i>Tabella 6.1.</i>	Frequenza di campionamento	93
<i>Tabella 7.1.</i>	Caratteristiche obiettivo del sistema di controllo.	110

Indice delle figure

Capitolo 1

Inquadramento del problema

<i>Figura 1.1.</i>	Fotografia dall'alto di un Saab JAS 39 Gripen.	5
<i>Figura 1.2.</i>	<i>Rendering</i> del Gripen NG, si noti come sia presente la cupola del sensoreIRST sul lato sinistro del muso, assente nelle versioni meno recenti del caccia.	6
<i>Figura 1.3.</i>	SensoreIRST del Su-27 e dell' Eurofighter Typhoon.	7
<i>Figura 1.4.</i>	Suddivisione delle diverse categorie di radiazioni elettromagnetiche in base alla loro lunghezza d'onda.	8
<i>Figura 1.5.</i>	Spettri di emissione di corpi a differenti temperature, si noti come, coerentemente a quanto formulato dalla legge di Wien, all'aumentare della temperatura il picco dello spettro si sposta verso frequenze più basse.	9
<i>Figura 1.6.</i>	Schematizzazione del sensore, in azzurro sono visibili i due assi di rotazione del sensore, mentre in rosso è visibile la traiettoria del raggio luminoso.	10

Capitolo 2

Schema dell'anello di controllo e modello *Simulink*

<i>Figura 2.1.</i>	Schema generale del sistema di controllo del sensore, comprendente anche la parte relativa ai dati di navigazione.	11
<i>Figura 2.2.</i>	Schematizzazione funzionale dell'anello di controllo di posizione del motore, con in evidenza le schede su cui sono implementate le funzioni	12
<i>Figura 2.3.</i>	Schema generale delle schede del servomeccanismo, con in evidenza la distinzione tra segnali analogici e segnali digitali.	13
<i>Figura 2.4.</i>	Schema a blocchi della scheda PSC, con in evidenza i sotto-blocchi.	13
<i>Figura 2.5.</i>	Schema a blocchi della scheda SDC, con in evidenza i sotto-blocchi.	14
<i>Figura 2.6.</i>	Schema a blocchi della scheda PSM, con in evidenza i sotto-blocchi.	14
<i>Figura 2.7.</i>	Schema del modello <i>Simulink</i> del sensore.	17
<i>Figura 2.8.</i>	Blocchi di generazione del riferimento di posizione, lo <i>switch</i> è impostato per leggere i <i>file</i> , su entrambi i gradi di libertà.	18
<i>Figura 2.9.</i>	Rappresentazione del fenomeno della leva ottica e visualizzazione degli angoli.	19

Figura 2.10. Blocchi di condizionamento del segnale di *elevation* che cambiano il sistema di riferimento con cui sono descritti gli angoli. 19

Figura 2.11. Blocchi di scrittura dei dati di *azimuth* ed *elevation* in opportune variabili, si notino i blocchi di condizionamento del segnale di *elevation*. 20

Figura 2.12. Controllore di posizione, per il grado di libertà di *elevation*, quello per l'*azimuth* è identico. 20

Figura 2.13. Blocco che modella il motore e l'elettronica di controllo. 21

Figura 2.14. Blocco dell'attrito, l'uscita *Taufric* non viene più usata ed è quindi mandata in un blocco terminatore. 21

Figura 2.15. Schema interno del blocco di attrito. 22

Figura 2.16. Blocco che modella la dinamica meccanica del sensore. 23

Figura 2.17. Blocco che simula i sensori di posizione e velocità 23

Figura 2.18. Schema interno del blocco che simula i sensori. 24

Figura 2.19. Blocco di grafica del modello *Simulink*. 24

Figura 2.20. Schema interno del blocco *Grafica*. 25

Figura 2.21. Esempio di finestra generata da *animIRST.m*. 26

Figura 2.22. Esempio di finestra generata da *graficiIRST.m*. 26

Capitolo 3

Modellazione multicorpo

Figura 3.1. Disposizione degli assi $\{x_g, y_g, z_g\}$ nel sistema di riferimento terrestre. 30

Figura 3.2. Disposizione degli assi $\{x_b, y_b, z_b\}$ nel sistema di riferimento solidale all'aereo. 31

Figura 3.3. Disposizione degli assi $\{x_s, y_s, z_s\}$ nel sistema di riferimento solidale alla base del sensore. 32

Figura 3.4. Disposizione degli assi $\{x_a, y_a, z_a\}$ nel sistema di riferimento solidale al supporto dello specchio del sensore. 33

Figura 3.5. Disposizione degli assi $\{x_m, y_m, z_m\}$ nel sistema di riferimento solidale allo specchio del sensore. 34

Figura 3.6. Parametrizzazione di rotazione per il velivolo, descritta dagli angoli di *Roll*, *Pitch* e *Yaw*. 35

Figura 3.7. Coordinate libere scelte per la scrittura delle equazioni di moto del sensore, rotazione attorno agli assi di *azimuth* ed *elevation*. 36

Figura 3.8. Blocco *Simulink* che implementa la dinamica meccanica del sensore. 43

<i>Figura 3.9.</i> Schema interno del blocco <i>Simulink</i> che descrive la dinamica del sensore.	44
Capitolo 4	
Modellazione del motore elettrico	
<i>Figura 4.1.</i> Schematizzazione del motore elettrico.	48
<i>Figura 4.2.</i> Circuito equivalente di ogni avvolgimento.	49
<i>Figura 4.3.</i> Schema del blocco di simulazione della dinamica elettrica.	50
<i>Figura 4.4.</i> Regolatore di corrente PI.	51
<i>Figura 4.5.</i> Schema del blocco di simulazione dell'elettronica di controllo. . .	52
<i>Figura 4.6.</i> Blocco <i>qd-PWM</i> , che antitrasforma i segnali ed applica, eventualmente, la PWM, i segnali in ingresso ed in uscita ai blocchi <i>go-to</i> sono collegati al blocco <i>PWM</i>	53
<i>Figura 4.7.</i> Andamento della tensione da modulare, in blu, e quella modulata, in rosso.	54
<i>Figura 4.8.</i> Ponte ad H, composto da quattro <i>switch</i> commutabili.	54
<i>Figura 4.9.</i> Andamento dei segnali di commutazione degli <i>switch</i> nel periodo del PWM per la modulazione indipendente e bipolare.	55
<i>Figura 4.10.</i> Fase uno, a sinistra, e due, a destra, della modulazione indipendente bipolare.	55
<i>Figura 4.11.</i> Andamento dei segnali di commutazione degli <i>switch</i> nel periodo del PWM per la modulazione indipendente unipolare.	56
<i>Figura 4.12.</i> Fase uno, a sinistra, e due, a destra, della modulazione indipendente unipolare.	56
<i>Figura 4.13.</i> Andamento dei segnali di commutazione degli <i>switch</i> nel periodo del PWM per la modulazione complementare bipolare, in grigio i tempi morti.	57
<i>Figura 4.14.</i> Fase uno, a sinistra, e due, a destra, della modulazione complementare bipolare.	57
<i>Figura 4.15.</i> Andamento dei segnali di commutazione degli <i>switch</i> nel periodo del PWM per la modulazione complementare unipolare, in grigio i tempi morti.	58
<i>Figura 4.16.</i> Fase uno, a sinistra, e due, a destra, della modulazione complementare unipolare.	58
<i>Figura 4.17.</i> Fase tre della modulazione complementare unipolare.	59
<i>Figura 4.18.</i> Blocco <i>PWM</i> , i blocchi <i>go-to</i> sono connessi ai segnali dei due blocchi <i>qd-PWM</i> , vedi Figura 4.6 a pagina 53.	60

Figura 4.19. Schema di uno dei quattro ponti ad H simulati nel blocco PWM. Il segnale di pilotaggio degli IGBT è verde per distinguerlo dal ponte ad H che non rappresenta un ramo di uno schema a blocchi ma un circuito elettrico vero e proprio. 61

Capitolo 5

Prove sperimentali sul sensore IRST

Figura 5.1. Andamento del tempo del *set-point* utilizzato per le prove di risposta a scalino, equivalente ad una rampa con tempo di salita pari ad un intervallo di campionamento. 64

Figura 5.2. Rappresentazione dell'andamento dei gradi di libertà in una prova relativa all'asse di *elevation*. Nel caso di una prova per l'*azimuth*, gli assi *azimuth* ed *elevation* risulterebbero invertiti. 65

Figura 5.3. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 0.015°, per entrambi i gradi di libertà. Il segnale misurato è rappresentato in nero mentre il *set-point* è rappresentato dalle linee colorate. Si noti che la scale dei tempi per questa prova è differente rispetto a tutte le altre. 66

Figura 5.4. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 0.03°, per entrambi i gradi di libertà. 66

Figura 5.5. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 0.06°, per entrambi i gradi di libertà. 67

Figura 5.6. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 0.125°, per entrambi i gradi di libertà. 67

Figura 5.7. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 0.25°, per entrambi i gradi di libertà. 67

Figura 5.8. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 0.5°, per entrambi i gradi di libertà. 68

Figura 5.9. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 1°, per entrambi i gradi di libertà. 68

Figura 5.10. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 10°, per entrambi i gradi di libertà. 68

Figura 5.11. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 20°, per entrambi i gradi di libertà. 69

Figura 5.12. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 40°, per entrambi i gradi di libertà. 69

Figura 5.13. Andamento nel tempo del *set-point* e del segnale misurato per le prove a scalino, ampiezza = 70°, per entrambi i gradi di libertà. 69

<i>Figura 5.14.</i> Andamento nel tempo del <i>set-point</i> e del segnale misurato per le prove a velocità costante, $\omega = 50^\circ/\text{s}$, per entrambi i gradi di libertà, $\Delta t = 0.5$ ms. Il segnale misurato è rappresentato in nero mentre il <i>set-point</i> è rappresentato dalle linee colorate.	70
<i>Figura 5.15.</i> Andamento nel tempo del <i>set-point</i> e del segnale misurato per le prove a velocità costante, $\omega = 100^\circ/\text{s}$, per entrambi i gradi di libertà, $\Delta t = 0.5$ ms.	70
<i>Figura 5.16.</i> Andamento nel tempo del <i>set-point</i> e del segnale misurato per le prove a velocità costante, $\omega = 150^\circ/\text{s}$, per entrambi i gradi di libertà, $\Delta t = 0.5$ ms.	71
<i>Figura 5.17.</i> Andamento nel tempo del <i>set-point</i> e del segnale misurato per le prove a velocità costante, $\omega = 200^\circ/\text{s}$, per entrambi i gradi di libertà, $\Delta t = 0.5$ ms.	71
<i>Figura 5.18.</i> Andamento nel tempo del <i>set-point</i> e del segnale misurato per le prove a velocità costante, $\omega = 400^\circ/\text{s}$, per entrambi i gradi di libertà, $\Delta t = 0.5$ ms.	71
<i>Figura 5.19.</i> Andamento nel tempo del <i>set-point</i> e del segnale misurato per le prove a velocità costante, $\omega = 50^\circ/\text{s}$, per entrambi i gradi di libertà, $\Delta t = 0.25$ ms. Il segnale misurato è rappresentato in nero mentre il <i>set-point</i> è rappresentato dalle linee colorate.	72
<i>Figura 5.20.</i> Andamento nel tempo del <i>set-point</i> e del segnale misurato per le prove a velocità costante, $\omega = 100^\circ/\text{s}$, per entrambi i gradi di libertà, $\Delta t = 0.25$ ms.	72
<i>Figura 5.21.</i> Andamento nel tempo del <i>set-point</i> e del segnale misurato per le prove a velocità costante, $\omega = 200^\circ/\text{s}$, per entrambi i gradi di libertà, $\Delta t = 0.25$ ms.	73
<i>Figura 5.22.</i> Andamento nel tempo del <i>set-point</i> e del segnale misurato per le prove a velocità costante, $\omega = 400^\circ/\text{s}$, per entrambi i gradi di libertà, $\Delta t = 0.25$ ms.	73
<i>Figura 5.23.</i> Andamento nel tempo del <i>set-point</i> e del segnale misurato per le prove a velocità costante, $\omega = 800^\circ/\text{s}$, per entrambi i gradi di libertà, $\Delta t = 0.25$ ms.	73
<i>Figura 5.24.</i> Andamento del segnale di posizione del sistema quando sottoposto ad un ingresso nullo. Il segnale può essere considerato come rumore della misura.	74

Capitolo 6

Individuazione e modellazione dei disturbi

Figura 6.1. Identificazione di un segmento a velocità costante, riquadro sinistro, e successivo *detrend* del segnale stesso, riquadro a destra. 75

Figura 6.2. Andamento nel tempo del *detrend* di un intervallo del segnale di *azimuth*, per la prova a velocità costante, $\omega = 50^\circ/\text{s}$ 76

Figura 6.3. Andamento nel tempo del *detrend* di un intervallo del segnale di *azimuth*, per la prova a velocità costante, $\omega = 100^\circ/\text{s}$ 76

Figura 6.4. Andamento nel tempo del *detrend* di un intervallo del segnale di *azimuth*, per la prova a velocità costante, $\omega = 150^\circ/\text{s}$ 77

Figura 6.5. Andamento nel tempo del *detrend* di un intervallo del segnale di *azimuth*, per la prova a velocità costante, $\omega = 200^\circ/\text{s}$ 77

Figura 6.6. Andamento nel tempo del *detrend* di un intervallo del segnale di *azimuth*, per la prova a velocità costante, $\omega = 400^\circ/\text{s}$ 77

Figura 6.7. Andamento nel tempo del *detrend* di un intervallo del segnale di *elevation*, per la prova a velocità costante, $\omega = 50^\circ/\text{s}$ 78

Figura 6.8. Andamento nel tempo del *detrend* di un intervallo del segnale di *elevation*, per la prova a velocità costante, $\omega = 100^\circ/\text{s}$ 78

Figura 6.9. Andamento nel tempo del *detrend* di un intervallo del segnale di *elevation*, per la prova a velocità costante, $\omega = 150^\circ/\text{s}$ 78

Figura 6.10. Andamento nel tempo del *detrend* di un intervallo del segnale di *elevation*, per la prova a velocità costante, $\omega = 200^\circ/\text{s}$ 79

Figura 6.11. Andamento nel tempo del *detrend* di un intervallo del segnale di *elevation*, per la prova a velocità costante, $\omega = 400^\circ/\text{s}$ 79

Figura 6.12. Andamento nel tempo di un intervallo della velocità di *azimuth*, per la prova a velocità costante, $\omega = 50^\circ/\text{s}$ 80

Figura 6.13. Andamento nel tempo di un intervallo della velocità di *azimuth*, per la prova a velocità costante, $\omega = 100^\circ/\text{s}$ 80

Figura 6.14. Andamento nel tempo di un intervallo della velocità di *azimuth*, per la prova a velocità costante, $\omega = 150^\circ/\text{s}$ 80

Figura 6.15. Andamento nel tempo di un intervallo della velocità di *azimuth*, per la prova a velocità costante, $\omega = 200^\circ/\text{s}$ 81

Figura 6.16. Andamento nel tempo di un intervallo della velocità di *azimuth*, per la prova a velocità costante, $\omega = 400^\circ/\text{s}$ 81

Figura 6.17. Andamento nel tempo di un intervallo della velocità di *elevation*, per la prova a velocità costante, $\omega = 50^\circ/\text{s}$ 81

Figura 6.18. Andamento nel tempo di un intervallo della velocità di *elevation*, per la prova a velocità costante, $\omega = 100^\circ/\text{s}$ 82

Figura 6.19. Andamento nel tempo di un intervallo della velocità di *elevation*, per la prova a velocità costante, $\omega = 150^\circ/\text{s}$ 82

Figura 6.20. Andamento nel tempo di un intervallo della velocità di *elevation*, per la prova a velocità costante, $\omega = 200^\circ/\text{s}$ 82

Figura 6.21. Andamento nel tempo di un intervallo della velocità di *elevation*, per la prova a velocità costante, $\omega = 400^\circ/\text{s}$ 83

Figura 6.22. Spettro in frequenza per il segnale di *azimuth*, $\omega = 50^\circ/\text{s}$, basse frequenze, sono evidenziati i due picchi principali. 84

Figura 6.23. Spettro in frequenza per il segnale di *azimuth*, $\omega = 100^\circ/\text{s}$, basse frequenze, sono evidenziati i due picchi principali. 84

Figura 6.24. Spettro in frequenza per il segnale di *azimuth*, $\omega = 150^\circ/\text{s}$, basse frequenze, sono evidenziati i due picchi principali. 84

Figura 6.25. Spettro in frequenza per il segnale di *azimuth*, $\omega = 200^\circ/\text{s}$, basse frequenze, sono evidenziati i due picchi principali. 85

Figura 6.26. Spettro in frequenza per il segnale di *azimuth*, $\omega = 400^\circ/\text{s}$, basse frequenze, sono evidenziati i due picchi principali. 85

Figura 6.27. Spettro in frequenza per il segnale di *elevation*, $\omega = 50^\circ/\text{s}$, basse frequenze, è evidenziato il picco principale. 85

Figura 6.28. Spettro in frequenza per il segnale di *elevation*, $\omega = 100^\circ/\text{s}$, basse frequenze, è evidenziato il picco principale. 86

Figura 6.29. Spettro in frequenza per il segnale di *elevation*, $\omega = 150^\circ/\text{s}$, basse frequenze, è evidenziato il picco principale. 86

Figura 6.30. Spettro in frequenza per il segnale di *elevation*, $\omega = 200^\circ/\text{s}$, basse frequenze, è evidenziato il picco principale. 86

Figura 6.31. Spettro in frequenza per il segnale di *elevation*, $\omega = 400^\circ/\text{s}$, basse frequenze, è evidenziato il picco principale. 87

Figura 6.32. Approssimazione lineare del legame tra frequenza del primo picco per il segnale di *azimuth* e velocità, confrontata con i dati sperimentali. . . 88

Figura 6.33. Approssimazione lineare del legame tra frequenza del primo picco per il segnale di *elevation* e velocità, confrontata con i dati sperimentali. . . 88

Figura 6.34. Andamento del *detrend* del segnale di *azimuth* rispetto all'*azimuth* stesso, per le differenti velocità. Si è normalizzata l'ampiezza dei segnali per poterli confrontare tra loro. 89

Figura 6.35. Andamento del *detrend* del segnale di *elevation* rispetto all'*elevation* stesso, per le differenti velocità. Si è normalizzata l'ampiezza dei segnali per poterli confrontare tra loro. 89

<i>Figura 6.36.</i> Approssimazione del legame tra frequenza del secondo picco per il segnale di <i>azimuth</i> e velocità, confortata con i dati sperimentali.	90
<i>Figura 6.37.</i> Andamento nel tempo del <i>detrend</i> del segnale di <i>azimuth</i> , a cui è stata sottratta la prima componente in frequenza, in scuro, in chiaro il segnale originale.	90
<i>Figura 6.38.</i> Approssimazione quadratica del legame tra ampiezza del primo picco per il segnale di <i>azimuth</i> e velocità, confortata con i dati sperimentali.	91
<i>Figura 6.39.</i> Approssimazione quadratica del legame tra ampiezza del secondo picco per il segnale di <i>azimuth</i> e velocità, confortata con i dati sperimentali.	91
<i>Figura 6.40.</i> Approssimazione lineare del legame tra ampiezza del secondo picco per il segnale di <i>elevation</i> e velocità, confortata con i dati sperimentali. . .	92
<i>Figura 6.41.</i> Rappresentazione del fenomeno dell' <i>aliasing</i> , in linea continua i picchi reali, in linea tratteggiata i picchi ottenuti dall'analisi del segnale campionato a f_{camp}	92
<i>Figura 6.42.</i> Spettro in frequenza per il segnale di <i>azimuth</i> , $\omega = 50^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5ms$, a destra $\Delta t = 0.25ms$	93
<i>Figura 6.43.</i> Spettro in frequenza per il segnale di <i>azimuth</i> , $\omega = 100^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5ms$, a destra $\Delta t = 0.25ms$	93
<i>Figura 6.44.</i> Spettro in frequenza per il segnale di <i>azimuth</i> , $\omega = 150^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5ms$	94
<i>Figura 6.45.</i> Spettro in frequenza per il segnale di <i>azimuth</i> , $\omega = 200^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5ms$, a destra $\Delta t = 0.25ms$	94
<i>Figura 6.46.</i> Spettro in frequenza per il segnale di <i>azimuth</i> , $\omega = 400^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5ms$, a destra $\Delta t = 0.25ms$	94
<i>Figura 6.47.</i> Spettro in frequenza per il segnale di <i>azimuth</i> , $\omega = 800^\circ/s$, alte frequenze, a destra $\Delta t = 0.25ms$	95
<i>Figura 6.48.</i> Spettro in frequenza per il segnale di <i>elevation</i> , $\omega = 50^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5ms$, a destra $\Delta t = 0.25ms$	95
<i>Figura 6.49.</i> Spettro in frequenza per il segnale di <i>elevation</i> , $\omega = 100^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5ms$, a destra $\Delta t = 0.25ms$	95
<i>Figura 6.50.</i> Spettro in frequenza per il segnale di <i>elevation</i> , $\omega = 150^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5ms$, a destra $\Delta t = 0.25ms$	96
<i>Figura 6.51.</i> Spettro in frequenza per il segnale di <i>elevation</i> , $\omega = 200^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5ms$, a destra $\Delta t = 0.25ms$	96
<i>Figura 6.52.</i> Spettro in frequenza per il segnale di <i>elevation</i> , $\omega = 400^\circ/s$, alte frequenze, a sinistra $\Delta t = 0.5ms$, a destra $\Delta t = 0.25ms$	96

Figura 6.53. In verde i blocchi aggiunti allo schema rappresentato in Figura 4.5 a pagina 52, al fine di poter simulare i disturbi. 98

Figura 6.54. Andamento della frequenza dei disturbi di *azimuth* in funzione della velocità per i dati sperimentali e per i dati ottenuti in simulazione. . 99

Figura 6.55. Andamento della ampiezza dei disturbi di *azimuth* in funzione della velocità per i dati sperimentali e per i dati ottenuti in simulazione. . 99

Figura 6.56. Andamento della frequenza dei disturbi di *elevation* in funzione della velocità per i dati sperimentali e per i dati ottenuti in simulazione. . 100

Figura 6.57. Andamento della ampiezza dei disturbi di *elevation* in funzione della velocità per i dati sperimentali e per i dati ottenuti in simulazione. . 100

Figura 6.58. In verde blocco aggiunto al modello di motore di *azimuth*, identico a quello visualizzato in Figura 4.3 a pagina 50, per simulare un disturbo. 101

Figura 6.59. Andamento della frequenza dei secondi disturbi di *azimuth* in funzione della velocità per i dati sperimentali e per i dati ottenuti in simulazione. 102

Figura 6.60. Andamento della ampiezza dei secondi disturbi di *azimuth* in funzione della velocità per i dati sperimentali e per i dati ottenuti in simulazione. 102

Capitolo 7

Controllore di posizione

Figura 7.1. Schema a blocchi di un controllore PID. 103

Figura 7.2. Schema del controllo in cascata. 104

Figura 7.3. Schema interno del regolatore di posizione. 105

Figura 7.4. Controllore di posizione, per il grado di libertà di *elevation*, quello per l'*azimuth* è identico. 106

Figura 7.5. Blocchi anti saturazione, lo *switch* si attiva quando il segnale non è nullo. 106

Figura 7.6. Schema semplificato del sensore. 107

Figura 7.7. Diagramma di Bode del modulo di L_v 108

Figura 7.8. Diagramma di Bode del modulo di L_p 109

Figura 7.9. Andamenti dei segnali di posizione, in simulazione, verde e viola, confrontato con i dati sperimentali, in linea nera. III

Figura 7.10. Andamenti dei segnali di posizione, in simulazione, verde e viola, confrontato con i dati sperimentali, in linea nera. III

Figura 7.11. Andamenti dei segnali di posizione, in simulazione, verde e viola, confrontato con i dati sperimentali, in linea nera. III

<i>Figura 7.12.</i> Andamento nel tempo della velocità di <i>azimuth</i> , decurtata del valore medio, $\omega = 200^\circ/s$	112
<i>Figura 7.13.</i> Andamento nel tempo della velocità di <i>elevation</i> , decurtata del valore medio, $\omega = 200^\circ/s$	112
<i>Figura 7.14.</i> Andamento nel tempo della velocità di <i>azimuth</i> , decurtata del valore medio, $\omega = 400^\circ/s$	113
<i>Figura 7.15.</i> Andamento nel tempo della velocità di <i>elevation</i> , decurtata del valore medio, $\omega = 400^\circ/s$	113
<i>Figura 7.16.</i> Schema di un generico sistema dinamico a cui è applicato un disturbo di carico.	114
<i>Figura 7.17.</i> Schema di un osservatore del disturbo.	115
<i>Figura 7.18.</i> Schema dell'osservatore del disturbo di coppia.	116
<i>Figura 7.19.</i> Schema dell'osservatore del disturbo di coppia.	117
<i>Figura 7.20.</i> Implementazione <i>Simulink</i> del TDO.	117
<i>Figura 7.21.</i> Modello del sistema con i due <i>Torque Disturbance Observer</i> , in marrone, la seconda parte del modello rimane invariata ed è rappresentata in Figura 2.7 a pagina 17.	118
<i>Figura 7.22.</i> Andamento nel tempo della velocità di <i>azimuth</i> , decurtata del valore medio, $\omega = 200^\circ/s$. In verde il segnale con TDO, in blu senza.	119
<i>Figura 7.23.</i> Andamento nel tempo della velocità di <i>elevation</i> , decurtata del valore medio, $\omega = 200^\circ/s$	119
<i>Figura 7.24.</i> Andamento nel tempo della velocità di <i>azimuth</i> , decurtata del valore medio, $\omega = 400^\circ/s$. In verde il segnale con TDO, in blu senza.	120
<i>Figura 7.25.</i> Andamento nel tempo della velocità di <i>elevation</i> , decurtata del valore medio, $\omega = 400^\circ/s$	120

Capitolo 8

Interfaccia grafica di controllo

<i>Figura 8.1.</i> Schema generale dell'interfaccia grafica, la linea che collega i menu FFT e grafici è tratteggiata perché la possibilità di accedervi è subordinata all'aver eseguito una simulazione.	121
<i>Figura 8.2.</i> Due differenti aspetti del menu principale, prima di una simulazione, a sinistra, e dopo una simulazione, a destra.	122
<i>Figura 8.3.</i> Rappresentazione del menu principale, che mostra la possibilità di attivare e disattivare alcune funzioni e vederne lo stato nel nome del pulsante; si noti che se il pulsante <i>PWM</i> è su ON è presente un ulteriore pulsante.	123

<i>Figura 8.4.</i> Pulsanti corrispondenti alle diverse tipologie di modulazione PWM che è possibile simulare.	124
<i>Figura 8.5.</i> Finestra di inserimento del valore di tempo finale della simulazione.	124
<i>Figura 8.6.</i> Finestra di inserimento del valore di tempo di campionamento, si noti il messaggio di errore poiché si è inserito un valore negativo, non valido.	125
<i>Figura 8.7.</i> Menu relativo ai grafici.	126
<i>Figura 8.8.</i> Esempio di una finestra dei grafici con in evidenza le grandezze relative all' <i>elevation</i>	127
<i>Figura 8.9.</i> Finestra di inserimento dei limiti per gli assi <i>x</i> ed <i>y</i> del grafico.	127
<i>Figura 8.10.</i> Finestra di errore che compare se si cerca di accedere al menu FFT anche se si è scelto un passo di campionamento non costante.	128
<i>Figura 8.11.</i> Finestra di selezione del grado di libertà che verrà analizzato.	128
<i>Figura 8.12.</i> Esempio di una finestra relativa al menu FFT.	129
<i>Figura 8.13.</i> Menu relativo all'FFT.	130
<i>Figura 8.14.</i> Finestra di inserimento dei limiti temporali della selezione.	131
<i>Figura 8.15.</i> Finestra di inserimento dei limite delle frequenze.	131

Appendice A

Lista degli acronimi

- DSP: *Digital Signal Processor*, microprocessore con una architettura ottimizzata per operare in modo veloce ed efficiente delle sequenze di istruzioni che servono al condizionamento dei segnali digitali.
- FFT: *Fast Fourier Transform*, algoritmo che permette l'esecuzione della trasformata discreta di *Fourier*, con un numero di operazioni ridotto, $O(N \log(N))$ invece di $O(N^2)$.
- FLIR: *Forward Looking InfraRed*, sensore di visione, sensibile alle radiazioni infrarosse.
- FPGA: *Field Programmable Gate Array*, circuito integrato digitale, programmabile tramite *hardware description language* (HDL).
- IGBT: *Insulated Gate Bipolar Transistor*, dispositivo elettronico di potenza che permette di commutare correnti e tensioni.
- IRST: *InfraRed Search and Track*, sensore ottico passivo, basato su una telecamera a raggi infrarossi, con capacità di ricerca ed inseguimento di un obiettivo.
- INS: *Inertial Navigation System*, sistema di navigazione inerziale del velivolo, che misura con degli accelerometri le accelerazioni subite dall'aereo stesso nei sue tre assi.
- LOS: *Line of Sight*, linea di vista del sensore, direzione di un raggio luminoso che entra nella telecamera, comandata dalla posizione dello specchio.
- PID: *Proporzionale Integrale Derivativo*, tipologia di controllore che fornisce un'uscita proporzionale ad un'opportuna combinazione dell'errore, del suo integrale e della sua derivata nel tempo.
- PSC: *Programmable Servo Control*, circuito per il *motion control* dello specchio, implementa l'anello di controllo.
- PSM: *Power Supply Management*, circuito per la gestione dell'alimentazione e l'implementazione ed il controllo del PWM.

- PWM: *Pulse–Width Modulation*, modulazione analogica che genera una serie di impulsi di ampiezza costante, il cui valore medio corrisponde a quello di una opportuna forma d'onda richiesta.
- SDC: *Signal Data Converter*, circuito per l'interfaccia e la conversione dei dati tra le componenti analogiche e quelle digitali del meccanismo.
- SPWM: *Synchronized (oppure Sinusoidal) Pulse–Width Modulation*, particolare modalità di PWM nel quale il segnale viene generato comparando la sinusoide richiesta con un'onda di forma triangolare.
- TDO: *Torque Disturbance Observer*, metodo che consente di stimare la coppia di disturbo e di compensarla opportunamente.

Appendice B

Listato delle funzioni e degli *script* *MATLAB*

B.1. animIRST.m

```
1 function [sys,x0] = animIRST (t,x,u,flag,ts,T_fin,flag_fig)
2 global Antenna
3
4 if flag==2,
5     if flag_fig
6         if any(get(0,'Children')==Antenna),
7             set(0,'currentfigure',Antenna);
8             armhand=get(gca,'UserData');
9
10
11             bracci;
12
13             Aaz=[cos(u(1)), -sin(u(1)), 0, 0;
14                 sin(u(1)), cos(u(1)), 0, 0;
15                 0, 0, 1, 0;
16                 0, 0, 0, 1];
17
18             Ael=[ cos(u(2)), 0, sin(u(2)), 0;
19                 0, 1, 0, 0;
20                 -sin(u(2)), 0, cos(u(2)), h_sp;
21                 0, 0, 0, 1];
22
23             A1=Aaz';
24             A2=Ael'*A1;
25
26             link0      = basello;
27             link1_dx   = spalla_dx*A1;
28             link1_sx   = spalla_sx*A1;
29             link1_hat  = cappello*A1;
30             link2_sup  = supp_sup*A2;
31             link2_inf  = supp_inf*A2;
32             link2_radar = radar*A2;
33             link2_ombra = ombra*A2;
34
35             set(armhand(1),'XData',xra,'YData',yra,'ZData',zra);
36             set(armhand(3),'XData',link1_dx(:,1),'YData',link1_
dx(:,2),'ZData',link1_dx(:,3));
37             set(armhand(4),'XData',link1_sx(:,1),'YData',link1_
sx(:,2),'ZData',link1_sx(:,3));
```

```

38         set(armhand(5),'XData',link1_hat(:,1),'YData',link1_
hat(:,2),'ZData',link1_hat(:,3));
39         set(armhand(6),'XData',link2_sup(:,1),'YData',link2_
sup(:,2),'ZData',link2_sup(:,3));
40         set(armhand(7),'XData',link2_inf(:,1),'YData',link2_
inf(:,2),'ZData',link2_inf(:,3));
41         set(armhand(8),'XData',link2_radar(:,1),'YData',link2_
radar(:,2),'ZData',link2_radar(:,3));
42         set(armhand(9),'XData',link2_ombra(:,1),'YData',link2_
ombra(:,2),'ZData',link2_ombra(:,3));
43         set(armhand(10),'XData',xc,'YData',yc,'ZData',zc);
44         set(armhand(11),'XData',xs,'YData',ys,'ZData',zs);
45         set(armhand(12),'XData',xco,'YData',yco,'ZData',zco);
46         set(armhand(13),'YData',t);
47
48         hold on,grid on
49
50         drawnow;
51         title('VISIONE 3D');
52         %         t_el=toc;
53         %         t_rim =round((T_fin-t)*t_el/t);
54         %         m_rim = floor(t_rim/60);
55         %         s_rim = t_rim - m_rim*60;
56         %         title(['VISIONE 3D,  t_{sim} = ',num2str(t,'% 10.3f' ),
...
57         %         ' s, fine simulazione tra ',num2str(m_rim,'%
4.0f'), ...
58         %         ' m e ',num2str(s_rim,'% 4.0f'),' s']);
59
60
61         end
62     end
63     sys=[];
64
65     elseif flag==4 %return next sample hit
66         % ns stores the number of samples
67         ns=t/ts;
68         % This is the time of the next sample hit.
69         sys=(1+floor(ns+1e-13*(1+ns)))*ts;
70
71
72     elseif flag==0,
73         tic
74         if flag_fig
75             % Initialize the figure for use with this simulation
76             namestr='ANIMAZIONE';
77             [existFlag,Antenna]=figflag(namestr);
78
79             if ~existFlag,

```

```

80     posizione=[50 50 600 600];
81     Antenna=figure( ...
82         'Name',namestr, ...
83         'NumberTitle','off', ...
84         'BackingStore','off', ...
85         'Position',posizione);
86     axes( ...
87         'Units','normalized', ...
88         'DrawMode','normal');
89     end;
90
91     LINKCOLOR=[...
92         % colore per i link
93         .7     .7     .7; % spalla dx
94         .3     .3     .3; % spalla sx
95         .5     .5     .5; % cappello
96         0.2   0.2   0.3; % basello
97         0.2   0.2   .2;  % sup. inf, supp. sup
98         0.85  0.75  0.2; % ombra antenna
99         1   0.9   0.3;  % antenna
100        0.4   0.4   0.4; % cupola
101        % colore dei bordi
102        0.1   0.1   0.1;
103        % colore del raggio
104        0.9   0     0;
105        ];
106
107     bracci;
108
109     link0      = basello;
110     link1_dx   = spalla_dx;
111     link1_sx   = spalla_sx;
112     link1_hat  = cappello;
113     link2_sup  = supp_sup;
114     link2_inf  = supp_inf;
115     link2_radar = radar;
116     link2_ombra = ombra;
117
118         subplot('Position',[.1 .05 .8 .02])
119     armhand(13)=barh(0,'r');
120     axis([0 T_fin .7 .9])
121
122     subplot('Position',[.1 .12 .8 .77])
123     % raggio
124     armhand(1)=plot3(xra,yra,zra,'EraseMode','normal','Color',LINKC
OLOR(10,:),'Linewidth',2);
125     hold on,grid on
126     quiver3(xra(2),yra(2),zra(2),-xra(2),-yra(2),-zra(2),'EraseMode','
normal','Color',LINKCOLOR(10,:),'Linewidth',2);

```

APPENDICE B

```

127
128
129     % struttura interna
130     armhand(2:9)=fill3(...
131         link0(:,1),    link0(:,2),    link0(:,3),
LINKCOLOR(4,:),...
132         link1_dx(:,1), link1_dx(:,2), link1_dx(:,3),
LINKCOLOR(1,:),...
133         link1_sx(:,1), link1_sx(:,2), link1_sx(:,3),
LINKCOLOR(2,:),...
134         link1_hat(:,1), link1_hat(:,2), link1_hat(:,3),
LINKCOLOR(3,:),...
135         link2_sup(:,1), link2_sup(:,2), link2_sup(:,3),
LINKCOLOR(5,:),...
136         link2_inf(:,1), link2_inf(:,2), link2_inf(:,3),
LINKCOLOR(5,:),...
137         link2_radar(:,1), link2_radar(:,2), link2_radar(:,3),
LINKCOLOR(7,:),...
138         link2_ombra(:,1), link2_ombra(:,2), link2_ombra(:,3),
LINKCOLOR(6,:),...
139
    'EraseMode','normal','EdgeColor',LINKCOLOR(9,:),'LineWidth',1);
140
141
142     % copertura
143     armhand(10)=surf(xc,yc,zc,yc,'EraseMode','normal','EdgeColor','
none','LineWidth',1);
144     armhand(11)=surf(xs,ys,zs,ys,'EraseMode','normal','EdgeColor','
none','LineWidth',1);
145     colormap(copper)
146     armhand(12)=plot3(xco,yco,zco,'EraseMode','normal','Color',LINK
COLOR(9,:),'LineWidth',1);
147
148     %     title(strvcat('     VISIONE 3D','     t_{sim} = 0 s',...
149     %     '     fine simulazione tra -- s'));
150     xlabel('asse x');ylabel('asse y');zlabel('asse z');
151
152     set(gca,'DataAspectRatio',[1 1 1], ...
153         'PlotBoxAspectRatio',[1 1 1], ...
154         'XLim',[-10 10], ...
155         'YLim',[-10 10], ...
156         'ZLim',[-20 0], ...
157         'ZDir','reverse', ...
158         'XDir','reverse');
159
160
161
162     set(gca,'UserData',armhand);
163
154

```

```

164
165         drawnow
166
167     %         h = waitbar(0,'Simulazione in corso');
168         end
169         sys=[0 0 0 2 0 0];
170         x0=[];
171
172     end

```

B.2. bracci.m

```

1     ris = 10;
2     ris = ris*2;
3
4     d_sp = 5;
5     l_sp = -2;
6     h_sp = -8;
7
8     a_ra = 4;
9     h_ra = 2.5;
10    b_ra = 2.5;
11
12    r_ci = d_sp *1.2;
13    b_sf = -(b_ra+h_ra) *1.2;
14
15    spalla_dx= [ l_sp    d_sp    0    1;
16                -l_sp   d_sp    0    1;
17                -l_sp   d_sp    h_sp   1;
18                [-cos(pi/ris*[0:ris])*l_sp]',d_sp*ones(ris+1,1),[h_
19                sp+sin(pi/ris*[0:ris])*l_sp]',ones(ris+1,1);
20                l_sp    d_sp    h_sp    1;];
21
22    spalla_sx = [l_sp    -d_sp    0    1;
23                -l_sp   -d_sp    0    1;
24                -l_sp   -d_sp    h_sp   1;
25                [-cos(pi/ris*[0:ris])*l_sp]',-d_sp*ones(ris+1,1),[h_
26                sp+sin(pi/ris*[0:ris])*l_sp]',ones(ris+1,1);
27                l_sp    -d_sp    h_sp    1;];
28
29    cappello = [ l_sp    -d_sp    0    1;
30                -l_sp   -d_sp    0    1;
31                -l_sp    d_sp    0    1;
32                l_sp    d_sp    0    1;];
33
34    basello = [[-cos(pi/ris*[0:2*ris])*l_sp/2]',[sin(pi/ris*[0:2*ris])*l_
35                sp/2]',-.05+zeros(2*ris+1,1),ones(2*ris+1,1)];

```

```
35
36  supp_sup = [-.5  5  -.3  1;
37             -.5 -5  -.3  1;
38             0  -5  -.5  1;
39             0   5  -.5  1;];
40
41
42  supp_inf = [-.5  5  -.3  1;
43             -.5 -5  -.3  1;
44             0  -5  .5  1;
45             0   5  .5  1;];
46
47
48  radar = [zeros(ris+1,1),[-cos(pi/ris*[0:ris])*a_ra]',h_ra+[sin(pi/
ris*[0:ris])*b_ra]',ones(ris+1,1);
49          zeros(ris+1,1),[-cos(pi/ris*[ris:-1:0])*a_ra]',-h_ra-[sin(pi/
ris*[ris:-1:0])*b_ra]',ones(ris+1,1)];
50
51
52  ombra = [-.3*ones(ris+1,1),[-cos(pi/ris*[0:ris])*a_ra]',h_ra+[sin(pi/
ris*[0:ris])*b_ra]',ones(ris+1,1);
53          -.3*ones(ris+1,1),[-cos(pi/ris*[ris:-1:0])*a_ra]',-h_ra-[sin(pi/
ris*[ris:-1:0])*b_ra]',ones(ris+1,1)];
54
55  % cilindro copertura
56  R = r_ci;
57  r = [R R]';
58  if flag == 0 theta = (0:ris)/ris*pi+pi/2; else theta = (0:ris)/
ris*pi+pi/2+u(1); end
59  sintheta = sin(theta);
60  xc = r * cos(theta);
61  yc = r * sintheta;
62  zc = [0;h_sp] * ones(1,ris+1);
63
64  % sfera copertura
65  phi = -(0:(ris/2))'/(ris/2)*pi/2;
66  cosphi = cos(phi);
67
68  xs = R * cosphi*cos(theta);
69  ys = R * cosphi*sintheta;
70  zs = h_sp-b_sf * sin(phi)*ones(1,ris+1);
71
72  % contorno
73
74  if flag ~= 0
75  xco = [R * cos(theta) , -R * cos((0:ris)/ris*pi)*cos(pi/2+u(1))];
76  yco = [R* sin(theta)   , -R * cos((0:ris)/ris*pi)*sin(pi/2+u(1))];
77  zco = [zeros(1,ris+1) ,h_sp+b_sf * sin((0:ris)/ris*pi)];
78  else
```



```

79 xco = [R * cos(theta) , -R * cos((0:ris)/ris*pi)*1];
80 yco = [R* sin(theta) , -R * cos((0:ris)/ris*pi)*0];
81 zco = [zeros(1,ris+1) ,h_sp+b_sf * sin((0:ris)/ris*pi)];
82 end
83
84 xco(end+1)=xco(1);
85 yco(end+1)=yco(1);
86 zco(end+1)=zco(1);
87
88 % raggio
89 if flag ~= 0
90 xra = [-10*cos(-2*u(2)+pi/2)*cos(u(1)) 0];
91 yra = [-10*cos(-2*u(2)+pi/2)*sin(u(1)) 0];
92 zra = [h_sp-10*sin(-2*u(2)+pi/2) h_sp];
93 else
94 xra = [1 0];
95 yra = [1 0];
96 zra = [h_sp h_sp];
97 end

```

B.3. dati_antenna.m

```

1 % Inertia characteristics
2
3 m1 = 0.88123; % [kg]
4 m2 = 0.00123; % [kg]
5
6 r1cm=[0.00136; 0.00016; -0.124]; % [m]
7 r2cm=[0; 0; 0]; % [m]
8
9 J0=[754032.789 904.745 -67920.156
10 904.745 198910.908 740.926
11 -67920.156 740.926 764428.6];
12
13 J2=[198904.718 0 0
14 0 686180.907 0
15 0 0 822440.586];
16
17 J3=[1392749.365 693.497 54275.872
18 693.497 615877.422 -9190.44
19 54275.872 -9190.44 1184242.253];
20
21 J4 = [17158787.79 -22864.13 406147
22 -22864.13 17251721.1 -68806.63
23 406147 -68806.63 1952572.52];
24 Jm_az = 0.0047;
25 Dm_az = 0;
26 Jm_e1 = 1.9891e-004;
27 Dm_e1 = 0;

```

APPENDICE B

```
28 Tf_az = .02;
29 Tf_el = .02;
30
31 I1=(J2+J3+J4)/1000^3;
32 I2=(J0)/1000^3;
33
34 Pitch=0;%pi/3; % [rad]
35
36
37 t_int = .3e-3;
38 pol   = 3*10^3;
39
40
41 % Azimuth Resolver
42 d_az = 0.17*1/1000;
43 A1=2.85;
44 A2 = 2.85*10^5;
45 B1=526.5;
46 B2=5265;
47 grtd=10;
48
49 % Elevation Resolver
50 d_el = 0.17*1/1000;
51
52 % Azimuth Plant Parameters
53
54 taus_az = 26.5*10^-3; %[N m] sticktion friction torque
55 tauc_az = 23*10^-3;  %[N m] Coulomb friction torque
56 Dm_az   = 0.3*10^-3; %[N m/rad/s] viscous friction coefficient
57 Kt_az   = 0.34;      %[N m/A] torque sensitivity
58 Ke_az   = Kt_az;    %[V/rad/s] electromechanical constant
59 R_az    = 2.95;     %[Ohm] stator winding electrical resistance
60 L_az    = 6.5e-4;   %[H] stator winding electrical inductance
61 Imax_az = 5.5;     %[A] Corrente massima
62 Vci_az  = 10;
63 Vlim_az = 28;
64 p_az    = 12;
65
66 % Elevation Plant Parameters
67
68 taus_el = 3.5*10^-3; %[N m] sticktion friction torque
69 tauc_el = 3.5*10^-3; %[N m] Coulomb friction torque
70 Dm_el   = 0.33*10^-3; %[N m/rad/s] viscous friction coefficient
71 Kt_el   = 0.064;     %[N m/A] torque sensitivity
72 Ke_el   = Kt_el;    %[V/rad/s] electromechanical constant
73 R_el    = 5.4;     %[Ohm] stator winding electrical resistance
74 L_el    = 1.3e-3;   %[H] stator winding electrical inductance
75 Imax_el = 2;       %[A] Corrente massima MANCA
76 Vci_el  = 10;
```

```

77  Vlim_el = 28;
78  p_el    = 6;
79
80  % Controllo di corrente azimuth
81
82  f_az    = 200;
83  Kci_az = 2*pi*L_az*f_az*(Imax_az/Vci_az)*(Vci_az/Vlim_az);
84  Tii_az = (L_az/R_az);
85
86  % Controllo di corrente elevation
87
88  f_el    = 200;
89  Kci_el = 2*pi*L_el*f_el*(Imax_el/Vci_el)*(Vci_el/Vlim_el);
90  Tii_el = (L_el/R_el);
91
92  % Controllo di posizione/velocità azimuth
93  wv_az   = 500;
94  Kcv_az = wv_az*Jm_az/Kt_az;
95  Kcp_az = 50;
96  Tiv_az = 10/wv_az;
97
98  % Controllo di posizione/velocità elevation
99  wv_el   = 500;
100 Kcv_el = wv_az*Jm_el/Kt_el;
101 Kcp_el = 50;
102 Tiv_el = 10/wv_el;
103
104 % Condizioni iniziali
105
106 qEL0 = -pi/4; % specchio inclinato per avere fascio luminoso
    dritto
107 qdEL0 = 0;      %          \
108 qAZ0  = 0;      %          ---\
109 qdAZ0 = 0;      %          /\
110
111
112 kcd_br = -100;
113 kcp_br = -100;
114
115
116 hc = 0.000106; % periodo di campionamento in secondi
117 tpwm = 1/(5*10^3); % periodo aggiornamento uscita PWM
118 sdt = 1.0000e-006;
119
120 %sottoblocco adj correz. su Vd e Vq
121 % derivata numerica vel con filtro s/(1+Tflv*s)
122 Tflv= 0.1*hc;
123 Vfilc1=1;
124 Vfilc0= -0.0000454;

```

APPENDICE B

```
125
126 % par1=KE0/p;
127 % par2=L;
128 par1=0;
129 par2=0;
130
131 % parametri dei sensori di Hall
132 % ixoff : corrente di offset fase x [A]
133 % fx : guad sensore fase x [adim]
134
135 fa=1;
136 fb=1;
137 % i1off=0.075;
138 % i2off=0.395;
139
140 i2off=22;%0.007*93.0842;
141
142 i0offel=0.0500;
143 i1offel=0.00;%1;%*93.0842;
144 i2offel=0;%0.007*93.0842;
145
146 taa = 1/(2*pi*1200);
147
148 nbit11=11;
149 nbit15=15;
150 Ishift=1;
151 Icontamp=1;
152 nbitres=14;
```

B.4. dist_AZ.m

```
1 function [sys,x0] = dist_AZ(t,x,u,flag)
2
3 if abs(flag)==0,
4 % N° stati cont., N° stati discr., N° uscite, N° ingr., ...
5
6 sys = [0;0;1;2;0;1];
7 x0 = [];
8
9 elseif abs(flag)==3,
10
11 az = 180/pi*u(2);
12 vaz = 180/pi*u(1);
13
14 P = [-3.5256e-8 3.1765e-5 4.7244e-4]; fre2 = 1./polyval(P,vaz);
15 P = [15*2.8627e-9 420*1.8165e-7 1000*2.8689e-6]; pot2 =
16 polyval(P,vaz);
17 if isinf(fre2)
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
```

```

18     fre2 = 0;
19 end
20
21 sys = pot2*sin(fre2*t);
22 x0 = [];
23
24
25 else
26     sys=[];
27 end

```

B.5. fftIRST.m

```

1     menu3 = 1;
2     az    = q_az(:,2)*180/pi;
3     el    = q_el(:,2)*180/pi;
4     if dt_samp == -1
5         warndlg('campionamento non costante, fft impossibile','!!
Warning !!')
6         menu3 = 0;
7     else
8         jgd1 = menu ('Selezionare il grado di
libertà','azimuth','elevation');
9         if jgd1 ==1
10            sign =az;
11        else
12            sign = el;
13        end
14
15        f_Ny = 1/dt_samp*pi;    %[rad/s]
16        f1 = 0;                %[rad/s]
17        f2 = f_Ny;             %[rad/s]
18        k=1;
19        intervallo = floor(length(az)/10);
20        namestr = 'FFT';
21
22        while menu3
23            % toglie il trend al segnale nell'intervallo
24            detrend=detrend(sign(k:k+intervallo));
25            pp = polyfit(t(k:k+intervallo),sign(k:k+intervallo), 1);
26            % ne fa la FFT
27            NFFT = 2^nextpow2(length(detrend));
28            Y=fft(detrend,NFFT)/length(detrend);
29            f = 1/dt_samp/2*linspace(0,1,NFFT/2+1);
30            % trova i limiti corrispondenti alle freq scelte
31            [void,i2] = min(abs(f-f2/2/pi));
32            [void,i1] = min(abs(f-f1/2/pi));
33            % trova i limiti corrispondenti alle freq scelte
34            [void,imax] = max(2*abs(Y(i1:i2)));

```

```

35
36     [existFlag,Grafici]=figflag(namestr);
37     if ~existFlag, %creagrafici
38         posizione=[770 50 900 600];
39         Grafici=figure('Name',namestr, 'NumberTitle','off', ...
40             'BackingStore','off', 'Position',posizione);
41         axes('Units','normalized','DrawMode','normal');
42         subplot('Position',[.06 .065+.67 .9 .22])
43         fftgr(1) = plot(t,sign,'Color',[0 0 .6]); hold on
44         fftgr(2) = plot(t(k:k+intervallo),sign(k:k+intervallo),'r'
,'Linewidth',2);
45         xlabel('tempo [s]'); ylabel(['°']); title(['segnale, V
= ',num2str(pp(1),'% 10.2f'),' °/s']);
46         grid on
47         box on
48
49         subplot('Position',[.06 .065+.33 .9 .22])
50         fftgr(3) = plot(t(k:k+intervallo),detr,'Color',[0 .7 0]) ;
51         xlabel('tempo [s]'); ylabel(['°']); title('detr');
52         grid on
53         box on
54
55         subplot('Position',[.06 .065 .9 .22])
56         fftgr(4) = stem(f(i1:i2)*2*pi,2*abs(Y(i1:i2)),'fill',':','Col
or',[.5 .3 0]);
57         set(fftgr(4),'MarkerFaceColor',[1 .9 0])
58         xlabel('frequenza [rad/s]'); ylabel(['|N(f)|']);
title(['fft, f_{max} = ',num2str(f(i1+imax-1)*2*pi,'% 10.2f'),' rad/s,
|N|_{max} = ',num2str(2*abs(Y(i1+imax-1)),'% 10.2f'),' rad']);
59         grid on
60         box on
61         hold on
62         fftgr(5) = stem(f(i1+imax-1)*2*pi,2*abs(Y(i1+imax-
1)),'fill','Color',[.5 0 0]);
63         set(fftgr(5),'MarkerFaceColor',[1 .0 0])
64
65     else % aggiornamento valori
66         set(fftgr(1),'XData',t,'YData',sign);
67
set(fftgr(2),'XData',t(k:k+intervallo),'YData',sign(k:k+intervallo));
68         set(fftgr(3),'XData',t(k:k+intervallo),'YData',detr);
69         set(fftgr(4),'XData',f(i1:i2)*2*pi,'YData',2*abs(Y(i1:i2)));
70         set(fftgr(5),'XData',f(i1+imax-1)*2*pi,'YData',2*abs(Y(i1+i
max-1)));
71         subplot('Position',[.06 .065+.67 .9 .22])
72         title(['segnale, V = ',num2str(pp(1),'% 10.2f'),' °/s']);
73         subplot('Position',[.06 .065 .9 .22])

```

```

74         title(['fft, f_{max} = ',num2str(f(i1+imax-1)*2*pi,'%
10.2f'),' rad/s, |N|_{max} = ',num2str(2*abs(Y(i1+imax-1)),'% 10.2f'),'
rad']);
75         end
76
77         kk = menu ('Modificare intervallo','<-','->','<=','=>','+','-','+
+','- -','Inserisci estremi intervallo','Inserisci estremi
frequenze','Indietro');
78         switch kk
79             case 1
80                 k = k-floor(intervallo/5);
81                 if k<=0; k=1; end
82             case 2
83                 k = k+floor(intervallo/5);
84                 if k> (length(t)-intervallo); k=length(t)-
intervallo; end
85             case 3
86                 k = k-intervallo;
87                 if k<=0; k=1; end
88             case 4
89                 k = k+intervallo;
90                 if k> (length(t)-intervallo); k=length(t)-
intervallo; end
91             case 5
92                 intervallo = intervallo + floor(intervallo/5);
93                 if intervallo > (length(t)); intervallo =
length(t)-1; end
94                 if intervallo > (length(t)-k); k=length(t)-
intervallo; end
95             case 6
96                 intervallo = intervallo - floor(intervallo/5);
97                 if intervallo<=0; intervallo=1; end
98             case 7
99                 intervallo = intervallo *2;
100                if intervallo > (length(t)); intervallo =
length(t)-1; end
101                if intervallo> (length(t)-k-0); k=length(t)-
intervallo-0; end
102             case 8
103                 intervallo = floor(intervallo /2);
104                 if intervallo<=0; intervallo=1; end
105             case 9
106                 inizio = 0;
107                 while (k<=0 || k> (length(t)-intervallo)
||intervallo<=0 ||intervallo> (length(t)-k) || intervallo >
(length(t)) || inizio ==0)
108                     if inizio == 0
109                         prompt={'inserisci t_1 [s]','inserisci t_2
[s]'};

```

APPENDICE B

```

110             else
111                 prompt={strvcat('VALORI NON
VALIDI','inserisci t_1'),'inserisci t_2'};
112             end
113             name='Estremi intervallo';
114             numlines=1;
115             defaultanswer= {num2str(t(k)),
num2str(t(k+intervallo))};
116             options.Resize='on';
117             options.WindowStyle='normal';
118             options.Interpreter='tex';
119             awr=inputdlg(prompt,name,numlines,defaultanswer
,options);
120             if isempty(awr)
121                 awr=defaultanswer;
122             end
123             t1 = str2num(awr{1});          t2 =
str2num(awr{2});
124             [tt,i1] = min(abs(t-t1)); [tt,i2] =
min(abs(t-t2));
125             k = i1;
126             intervallo = i2-i1;
127             inizio = 1;
128         end
129     case 10
130         inizio = 0;
131         while (f1<0 || f2 >f_Ny || f2<=f1 || inizio ==0)
132             if inizio == 0
133                 prompt={'inserisci f_1 [rad/s'],'inserisci
f_2 [rad/s]'};
134             else
135                 if f2 >f_Ny
136                     prompt={strvcat('VALORI NON
VALIDI',strcat('Non superare f_{Ny}=',num2str(f_Ny),'rad/s'),'inserisci
f_1'),'inserisci f_2'};
137                 else
138                     prompt={strvcat('VALORI NON
VALIDI','inserisci f_1'),'inserisci f_2'};
139                 end
140             end
141             name='Estremi intervallo frequenze';
142             numlines=1;
143             defaultanswer={num2str(f1),num2str(f2)};
144             options.Resize='on';
145             options.WindowStyle='normal';
146             options.Interpreter='tex';
147             awr=inputdlg(prompt,name,numlines,defaultanswer
,options);
148             if isempty(awr)

```



```
149             awr=defaultanswer;
150             end
151             f1 = str2num(awr{1});    f2 = str2num(awr{2});
152             inizio = 1;
153             end
154             case 11
155                 menu3 = 0;
156             end
157         end
158     end
159 end
```

B. 6. figureIRST.m

```
1     menu3 = 1;
2
3     namestr='GRAFICI';
4     [existFlag,Grafici]=figflag(namestr);
5
6     if ~existFlag,
7         posizione=[700 50 900 600];
8         Grafici=figure( ...
9             'Name',namestr, ...
10            'NumberTitle','off', ...
11            'BackingStore','off', ...
12            'Position',posizione);
13         axes( ...
14            'Units','normalized', ...
15            'DrawMode','normal');
16     end;
17
18     clf
19     title('Grafici');
20     xmin = 0;
21     xmax = T_fn;
22     lw = 1.5;
23
24     ymin=-100;
25     ymax=100;
26     axis([xmin xmax, ymin ymax]);
27     xlabel('tempo [s]');
28     ylabel('[°] [A]');
29     grid on
30     box on
31     hold on
32     leg = '';
33     fatto(1:11) = 0;
34
35     while menu3
```

```
36     sel = menu ('Variabili da graficare','q_{el,SP}','q_{el}','q_{el,mis}','q_{az,SP}','q_{az}','q_{az,mis}','I_{el}','I_{az}','Reimposta assi','Pulisci tutto','Indietro');
37     inizio = 0;
38     if fatto(sel) == 0;
39         switch sel
40             case 1
41                 figflag(namestr);
42                 fatto(sel) = 1;
43                 plot(t,q_el(:,1)*180/pi,'r','linewidth',lw)
44                 leg = strvcat(leg,'q_{el,SP}');
45                 legend(leg);
46             case 2
47                 figflag(namestr);
48                 fatto(sel) = 1;
49                 plot(t,q_el(:,2)*180/pi,'Color',[0.5 0
0.3],'linewidth',lw)
50                 leg = strvcat(leg,'q_{el}');
51                 legend(leg);
52             case 3
53                 figflag(namestr);
54                 fatto(sel) = 1;
55                 plot(t,q_el(:,3)*180/pi,'Color',[0 0.6
0.], 'linewidth',lw)
56                 leg = strvcat(leg,'q_{el,mis}');
57                 legend(leg);
58             case 4
59                 figflag(namestr);
60                 fatto(sel) = 1;
61                 plot(t,q_az(:,1)*180/pi,'Color',[0.5 0.5
0.5],'linewidth',lw)
62                 leg = strvcat(leg,'q_{az,SP}');
63                 legend(leg);
64             case 5
65                 figflag(namestr);
66                 fatto(sel) = 1;
67                 plot(t,q_az(:,2)*180/pi,'Color',[.8 .8
0.4],'linewidth',lw)
68                 leg = strvcat(leg,'q_{az}');
69                 legend(leg);
70             case 6
71                 figflag(namestr);
72                 fatto(sel) = 1;
73                 plot(t,q_az(:,3)*180/pi,'c','linewidth',lw);
74                 leg = strvcat(leg,'q_{az,mis}');
75                 legend(leg);
76             case 7
77                 figflag(namestr);
78                 fatto(sel) = 1;
```

```

79         plot(t,I_e1,'m','linewidth',lw);
80         leg = strvcat(leg,'I_{e1}');
81         legend(leg);
82     case 8
83         figflag(namestr);
84         fatto(sel) = 1;
85         plot(t,I_az,'b','linewidth',lw);
86         leg = strvcat(leg,'I_{az}');
87         legend(leg);
88     case 9
89         while (xmin<0 || xmax>T_fin || xmax<=xmin ||
ymax<=ymin || inizio ==0)
90             if inizio == 0
91                 prompt={'x_{min}','x_{max}','y_{min}','y_
{max}'};
92             else
93                 prompt={'strvcat('VALORI NON VALIDI','x_
{min}'),'x_{max}','y_{min}','y_{max}'};
94             end
95             name='limiti assi';
96             numlines=1;
97             defaultanswer={num2str(xmin),num2str(xmax),num2s
tr(ymin),num2str(ymax)};
98             options.Resize='on';
99             options.WindowStyle='normal';
100            options.Interpreter='tex';
101            awr=inputdlg(prompt,name,numlines,defaultanswer
,options);
102            xmin = str2num(awr{1});
103            xmax = str2num(awr{2});
104            ymin = str2num(awr{3});
105            ymax = str2num(awr{4});
106            [existFlag,Grafici]=figflag(namestr);
107            if not(xmin<0 || xmax>T_fin || xmax<=xmin ||
ymax<=ymin)
108                axis([xmin xmax, ymin ymax]);
109                end
110                inizio = 1;
111            end
112    case 10
113        [existFlag,Grafici]=figflag(namestr);
114        clf
115        title('Grafici');
116        xmin = 0;
117        xmax = T_fin;
118        lw = 1.5;
119        ymin=-100;
120        ymax=100;
121        axis([xmin xmax, ymin ymax]);

```

```
122             xlabel('tempo [s]');
123             ylabel(['°  [A]');
124             grid on
125             box on
126             hold on
127             leg = '';
128             fatto(1:11) = 0;
129             case 11
130                 menu3 = 0;
131             end
132         end
133     end
```

B.7. gen_SP.m

```
1     az     = load (strcat(PNaz, FNaz))*pi/180;
2     el     = load (strcat(PNel, FNel))*pi/180;
3     tt     = 0:dt:(length(az)-1)*dt;
4
5     az_sp = [];
6     el_sp = [];
7     for i = 1:floor(T_fin/tt(end))
8         az_sp = [az_sp ; az];
9         el_sp = [el_sp ; el];
10    end
11    tt     = [0:dt:(length(az_sp)-1)*dt]';
12    if isempty(tt)
13        tt=0;
14    end
15
16    az_sp(1:end+ceil((T_fin-tt(end))/dt),1) = [az_sp ; az(1:ceil((T_fin-
17    tt(end))/dt))];
18    el_sp(1:end+ceil((T_fin-tt(end))/dt),1) = [el_sp ; el(1:ceil((T_fin-
19    tt(end))/dt))];
20    tt     = [0:dt:(length(az_sp)-1)*dt]';
21    az_sp = [tt,az_sp];
22    el_sp = [tt,el_sp];
```

B.8. graficiIRST.m

```
1     function [sys,x0] = graficiIRST (t,x,u,flag,ts,T_fin,flag_gra)
2     global Grafici
3
4     if flag==2,
5         if flag_gra
6             if any(get(0,'Children')==Grafici),
7                 set(0,'currentfigure',Grafici);
8                 graf=get(gca,'UserData');
```

```
9
10     u(1:6) = u(1:6)*180/pi;
11     tt    = [get(graf(1),'XData'),t];
12
13     for j = 1:8
14         yy    = [get(graf(j),'YData'),u(j)];
15         set(graf(j),'XData',tt,'YData',yy);
16     end
17     set(graf(9),'XData',[t tt]);
18
19     drawnow;
20     title('Scope, risoluzione in tempo non elevata');
21     box on
22     grid on
23
24
25     end
26 end
27 sys=[];
28
29 elseif flag==4 %return next sample hit
30     % ns stores the number of samples
31     ns=t/ts;
32     % This is the time of the next sample hit.
33     sys=(1+floor(ns+1e-13*(1+ns)))*ts;
34
35
36 elseif flag==0,
37     if flag_gra
38
39     % Initialize the figure for use with this simulation
40     namestr='GRAFICI';
41     [existFlag,Grafici]=figflag(namestr);
42
43     if ~existFlag,
44         posizione=[700 50 900 600];
45         Grafici=figure( ...
46             'Name',namestr, ...
47             'NumberTitle','off', ...
48             'BackingStore','off', ...
49             'Position',posizione);
50         axes( ...
51             'Units','normalized', ...
52             'DrawMode','normal');
53     end;
54
55     t = 0;
56     u = zeros(8,1);
57     q_e1 = u(1:3);
```

APPENDICE B

```

58     q_az = u(4:6);
59     I_el = u(7);
60     I_az = u(8);
61     leg  = [];
62
63     lw = 1.5;
64     hold on
65
66     leg = strvcatscat(leg,'q_{el,SP}');
67     leg = strvcatscat(leg,'q_{el}');
68     leg = strvcatscat(leg,'q_{el,mis}');
69     leg = strvcatscat(leg,'q_{az,SP}');
70     leg = strvcatscat(leg,'q_{az}');
71     leg = strvcatscat(leg,'q_{az,mis}');
72     leg = strvcatscat(leg,'I_{el}');
73     leg = strvcatscat(leg,'I_{az}');
74
75     graf(1) = plot(t,q_el(1)*180/pi,'r','linewidth',lw);
76     graf(2) = plot(t,q_el(2)*180/pi,'Color',[0.5 0
0.3],'linewidth',lw) ;
77     graf(3) = plot(t,q_el(3)*180/pi,'Color',[0. 0.6
0.], 'linewidth',lw);
78     graf(4) = plot(t,q_az(1)*180/pi,'Color',[0.5 0.5
0.5],'linewidth',lw);
79     graf(5) = plot(t,q_az(2)*180/pi,'Color',[.8 .8
0.4],'linewidth',lw);
80     graf(6) = plot(t,q_az(3)*180/pi,'c','linewidth',lw);
81     graf(7) = plot(t,I_el,'m','linewidth',lw);
82     graf(8) = plot(t,I_az,'b','linewidth',lw);
83     graf(9) = plot([t t],[-100 100],'k --','linewidth',1);
84
85     legend(leg);
86     % copertura
87
88     xlabel('asse x');ylabel('asse y');zlabel('asse z');
89
90     set(gca, 'XLim',[0 T_fin], ...
91           'YLim',[-100 100]);
92           xlabel('tempo [s]');
93           ylabel('[°] [A]');
94
95     set(gca,'UserData',graf);
96     drawnow
97     end
98     sys = [0 0 0 8 0 0];
99     x0 = [];
100
101 end

```

B. 9. inertiadist.m

```

1  function [sys,x0] = inertiadist (t,x,u,flag,m1,m2,r1cm,r2cm)
2  % calcola disturbi sul senore dovuti alle accelerazione
   dell'aeromobile
3
4  if abs(flag)==0,
5  % N° stati cont., N° stati discr., N° uscite, N° ingr., ...
6  sys = [0;0;2;5;0;1];
7  x0 = [];
8
9  elseif abs(flag)==3,
10
11  qci = u(1);           % posizione di circular
12  qel = u(2);           % posizione di elevation
13  ax = u(3);           % accelerazione lungo x, aereo
14  ay = u(4);           % accelerazione lungo y, aereo
15  az = u(5);           % accelerazione lungo z, aereo
16
17  % definizione del vettore accelerazione lineare dell'elicottero
18  a = [ax;ay;az];
19
20  % definizione di seni e coseni degli angoli di rotazione
21  cci = cos(qci);
22  sci = sin(qci);
23  cel = cos(qel);
24  sel = sin(qel);
25
26  % definizione delle matrici di rotazione
27  R1 = [cci , -sci , 0;
28        sci ,  cci , 0;
29        0 ,   0 , 1];
30
31  R2 = [cel , 0 , sel;
32        0 , 1 , 0;
33        -sel , 0 , cel];
34
35  % definizione degli assi di rotazione
36  nci = [0 , 0 , 1]';
37  nel = [0 , 1 , 0]';
38
39  % calcolo dei contributi di coppia di inerzia
40  a1 = R1'*a;
41  tau1ci = ( cross(r1cm , -m1*a1) )' * nci;
42  r2cm1 = R2*r2cm;
43  tau2ci = ( cross(r2cm1 , -m2*a1) )' * nci;
44  tau1el = 0; % il link 1 non è vincolato all'elicottero mediante
   l'asse in questione
45  tau2el = ( cross(r2cm1 , -m2*a1) )' * nel;

```

```
46
47 % calcolo del vettore taudist
48 sys = [tau1ci + tau2ci ;
49         tau1el + tau2el];
50
51 else
52     sys=[];
53 end
```

B.10. IRSTdin.m

```
1 function [sys,x0] = IRSTdin (t,x,u,flag,m1,m2,r1cm,r2cm,I1,I2);
2 % calcola accelerazioni per modello multicorpo a 2 link
3 % escursioni angolari consentite:  $0 \leq qaz \leq 360^\circ$ ,  $-45^\circ \leq qel \leq 45^\circ$ 
4
5 if abs(flag)==0,
6 % N° stati cont., N° stati discr., N° uscite, N° ingr., ...
7
8 sys = [0;0;2;7;0;1];
9 x0 = [];
10
11 elseif abs(flag)==3,
12
13 qaz = u(1); % posizione di azimuth
14 qel = u(2); % posizione di elevation
15 qpaz = u(3); % velocità di azimuth
16 qpel = u(4); % velocità di elevation
17 tau1 = u(5); % coppia di azimuth
18 tau2 = u(6); % coppia di elevation
19 pitch = u(7); % angolo di pitch
20
21 %% definizione costanti
22
23 % acc. di gravità
24 g = 9.81;
25
26 % distanza del baricentro I
27 r1x = r1cm(1);
28 r1y = r1cm(2);
29 r1z = r1cm(3);
30
31 % distanza del baricentro II
32 r2x = r2cm(1);
33 r2y = r2cm(2);
34 r2z = r2cm(3);
35
36 % termini non nulli della matrice di inerzia I
37 I1x = I1(1,1);
38 I1y = I1(2,2);
```



```

39  I1z = I1(3,3);
40  I1xy = I1(1,2);
41  I1xz = I1(1,3);
42  I1yz = I1(2,3);
43
44  % termini non nulli della matrice di inerzia II
45  I2x = I2(1,1);
46  I2y = I2(2,2);
47  I2z = I2(3,3);
48  I2xy = I2(1,2);
49  I2xz = I2(1,3);
50  I2yz = I2(2,3);
51
52  %% I equazione di Lagrange
53
54  % termini inerziali
55  A = m1 * ( (r1x^2) + (r1y^2) ) + I1z;
56  B = m2 * ( (r2x^2) + (r2y^2) ) + I2z;
57  C = m2 * ( (r2y^2) + (r2z^2) ) + I2x;
58  D = 2*( m2 * r2x * r2z - I2xz);
59  E = - m2 * r2y * r2z + I2yz;
60  F = m2 * r2x * r2y - I2xy;
61
62  % termini dovuti al campo gravitazione
63  G1 = -m1 * g * sin(pitch) * r1x;
64  G2 = -m1 * g * sin(pitch) * r1y;
65  G3 = -m2 * g * sin(pitch) * r2x;
66  G4 = -m2 * g * sin(pitch) * r2z;
67  G5 = -m2 * g * sin(pitch) * r2y;
68
69  %% II equazione di Lagrange
70
71  % termini inerziali
72  H = m2 * ( (r2x^2) + (r2z^2) ) + I2y;
73  I = -m2 * r2y * r2z + I2yz;
74  L = m2 * r2x * r2y - I2xy;
75  M = m2 * ( (r2z^2) - (r2x^2) ) + I2x - I2z;
76  N = m2 * r2x * r2z - I2xz;
77  O = -m2 * r2x * r2z + I2xz;
78  P = m2 * r2x * r2y - I2xy;
79  Q = m2 * r2y * r2z - I2yz;
80
81  % termini dovuti al campo gravitazione
82  G6 = m2 * g * r2x * cos(pitch);
83  G7 = m2 * g * r2z * cos(pitch);
84  G8 = -m2 * g * r2x * sin(pitch);
85  G9 = m2 * g * r2z * sin(pitch);
86
87  %% definizione del sistema di equazioni

```

```
88
89 % definizione di vettori con i dati in ingresso
90 tau = [tau1;tau2]; % vettore delle coppie
91 q = [qaz;qel]; % vettore degli angoli
92 qp = [qpaz;qpel]; % vettore delle velocità angolari
93
94 % definizione di seni e coseni degli angoli di rotazione
95 saz = sin(q(1));
96 caz = cos(q(1));
97 cel = cos(q(2));
98 sel = sin(q(2));
99
100 % definizione delle matrici M(q),C(q,qp),g(q)
101 Mq = [ A+B*(cel^2)+C*(sel^2)+D*sel*cel , E*cel+F*sel;
102 I*cel+L*sel , H ];
103
104 Cqqd = [(2*(C-B)*sel*cel+D*((cel^2)-(sel^2)))*qp(2) , (F*cel-
E*sel)*qp(2);
105 -(M*cel*sel+N*(cel^2)+O*(sel^2))*qp(1) , ((L-P)*cel-
(Q+I)*sel)*qp(1)];
106
107 gq = [G1*saz+G2*caz+G3*cel*saz+G4*sel*saz+G5*caz ; G6*cel+G7*sel+G8
*sel*caz+G9*cel*caz];
108
109 %% risoluzione del sistema di equazioni di Lagrange
110
111 % calcolo del vettore qpp,
112 sys = Mq\(-Cqqd*qp-gq+tau);
113
114 else
115 sys=[];
116 end
```

B. 11. MAIN.m

```
1 clear all;
2 close all;
3 clc;
4 inizio1 = 0;
5 inizio2 = 0;
6
7 menu1 = 1;
8 simul = 0;
9 vel = 100;
10 dt = 0.0005;
11 dt_samp = 0.0005;
12 dt_gra = 1/500;
13 T_fin = .1;
14 pwm_t = 1;
```

```

15
16 FNaz = [];
17 PNaz = 'az.txt';
18 FNel = [];
19 PNel = 'el.txt';
20 flag_fig = 1;
21 flag_gra = 0;
22 flag_PWM = 0;
23 sino = {' OFF', ' ON'};
24 while menu1
25     if simul == 0
26         sel1 = menu ('Simulazione IRST','Simulazione','Apri
modello','Chiudi');
27         if sel1 >=2
28             sel1=sel1+2;
29         end
30     else
31         sel1 = menu ('Simulazione IRST','Simulazione','Grafici','FFT',
'Apri modello','Chiudi');
32     end
33     menu2 = 1;
34     switch sel1
35         case 1
36             while menu2
37                 if flag_PWM
38                     switch pwm_t
39                         case 1
40                             strpwm = 'Ind. Bipolar PWM';
41                         case 2
42                             strpwm = 'Ind. Unipolar PWM';
43                         case 3
44                             strpwm = 'Compl. Bipolar PWM';
45                         case 4
46                             strpwm = 'Compl. Unipolar PWM';
47                     end
48                     sel2 = menu ('Menu
simulazione',char(strcat('Disegno 3D',sino(flag_fig+1))),
char(strcat( 'Scope',sino(flag_gra+1))), char(strcat('PWM',
sino(flag_PWM+1))),strpwm,'Esegui simulazioni','Modifica tempo
finale simulazione','Modifica Dt campionamento','Modifica file set-
point','Modifica Dt set-point','Indietro');
49                     else
50                         sel2 = menu ('Menu
simulazione',char(strcat('Disegno 3D',sino(flag_fig+1))),char(strcat('
Scope',sino(flag_gra+1))),char(strcat('PWM',sino(flag_PWM+1))),'Esegui
simulazioni','Modifica tempo finale simulazione','Modifica
Dt campionamento','Modifica file set-point','Modifica Dt set-
point','Indietro');
51                     if sel2>3

```

```
52         sel2=sel2+1;
53     end
54 end
55 switch sel2
56     case 1
57         flag_fig = 1-flag_fig;
58     case 2
59         flag_gra = 1-flag_gra;
60     case 3
61         flag_PWM = 1-flag_PWM;
62     case 4
63         pwm_t = pwm_t+1;
64         if pwm_t>4
65             pwm_t = 1;
66         end
67     case 5
68         close all
69         dati_antenna
70         gen_SP
71         tic
72         sim('modelloIRST')
73         toc
74         simul = 1;
75         menu2 = 0;
76     case 6
77         inizio1 = 0;
78         while (T_fin<=0) || (inizio1 ==0)
79             if inizio1 == 0
80                 prompt={'inserisci T_{fin} [s]'};
81             else
82                 prompt={strvcat('VALORI NON
VALIDI','inserisci T_{fin} [s]')};
83             end
84             name = 'Tempo finale';
85             numlines = 1;
86             defaultanswer = {num2str(T_fin)};
87             options.Resize='on'; options.
WindowStyle='normal'; options.Interpreter='tex';
88             awr = inputdlg(prompt,name,numlines,de
faultanswer,options);
89             if isempty(awr)
90                 awr = defaultanswer;
91             end
92             T_fin = str2num(awr{1});
93             inizio1 = 1;
94         end
95     case 7
96         inizio2 = 0;
```

```

97         while ((dt_samp~-=-1) && (dt_samp<=0)) ||
(inizio2 ==0)
98             if inizio2 == 0
99                 prompt={'inserisci \Delta t_{camp}
[s]'};
100             else
101                 prompt={strvcat('VALORI NON
VALIDI','inserisci \Delta t_{camp} [s]')};
102             end
103             name = '\Delta t_{camp}';
104             numlines = 1;
105             defaultanswer = {num2str(dt_samp)};
106             options.Resize='on'; options.
WindowStyle='normal'; options.Interpreter='tex';
107             awr = inputdlg(prompt,name,numlines,de
faultanswer,options);
108             if isempty(awr)
109                 awr = defaultanswer;
110             end
111             dt_samp = str2num(awr{1});
112             inizio2 = 1;
113         end
114     case 8
115         [FNaz,PNaz] = uigetfile('*.','Selezionare il
file di set point di azimuth');
116         [FNe1,PNe1] = uigetfile('*.','Selezionare il
file di set point di elevation');
117         if FNaz == 0
118             PNaz = [];
119             FNaz = 'az.txt';
120         end
121         if FNe1 == 0
122             PNe1 = [];
123             FNe1 = 'el.txt';
124         end
125     case 9
126         inizio2 = 0;
127         while (dt<=0) || (inizio2 ==0)
128             if inizio2 == 0
129                 prompt={'inserisci \Delta t [s]'};
130             else
131                 prompt={strvcat('VALORI NON
VALIDI','inserisci \Delta t [s]')};
132             end
133             name = '\Delta t';
134             numlines = 1;
135             defaultanswer = {num2str(dt)};
136             options.Resize='on'; options.
WindowStyle='normal'; options.Interpreter='tex';

```

```
137                                     awr = inputdlg(prompt,name,numlines,de
    faultanswer,options);
138                                     if isempty(awr)
139                                         awr = defaultanswer;
140                                     end
141                                     dt = str2num(awr{1});
142                                     inizio2 = 1;
143                                     end
144                                     case 10
145                                         menu2 = 0;
146                                     end
147                                     end
148                                     case 2
149                                         figureIRST;
150                                     case 3
151                                         fftIRST;
152                                     case 4
153                                         open('modelloIRST')
154                                     case 5
155                                         menu1 = 0;
156                                         close all;
157                                     end
158     end
```

B.12. PWM.m

```
1     function [sys,x0] = PWM (tt,x,u,flag,tpwm,Vmax,pwm_t,sdt)
2     if abs(flag)==0,
3         sys = [0;0;4;2;0;1];
4         x0 = [];
5     elseif abs(flag)==3,
6         V = u(2);
7         t = u(1);
8         tc = t-floor(t/tpwm)*tpwm;
9         dc = abs(V)/Vmax*tpwm;
10
11         switch pwm_t
12             case 1 %Independent Bipolar PWM
13                 dc = abs(V+V_max)/2/Vmax*tpwm;
14                 t1 = (tpwm-dc)/2;
15                 Q1 = tc>=t1 && tc<=(tpwm-t1);
16                 Q2 = 0;
17                 Q3 = 0;
18                 Q4 = Q1;
19             case 2 %Independet Unipolar PWM
20                 Q1 = tc<=dc;
21                 Q2 = 0;
22                 Q3 = 0;
23                 Q4 = 1;
```

```
24         case 3 %Complementary Bipolar PWM
25             dc = abs(V+V_max)/2/Vmax*(tpwm-2*sdt);
26             t1 = (tpwm-dc)/2-sdt;
27             Q1 = tc>(t1+sdt) && tc<(tpwm-t1-sdt);
28             Q2 = tc<t1 || tc>(tpwm-t1);
29             Q3 = Q2;
30             Q4 = Q1;
31         case 4 %Complementary Unipolar PWM
32             dc = abs(V)/Vmax*(tpwm-4*sdt);
33             t1 = (tpwm-dc)/4-sdt;
34             Q1 = tc>(t1+sdt) && tc<(tpwm-t1-sdt);
35             Q2 = tc<t1 || tc>(tpwm-t1);
36             Q3 = tc>(t1+dc/2+2*sdt) && tc<(tpwm-t1-dc/2-2*sdt);
37             Q4 = tc<(t1+dc/2+sdt) || tc>(tpwm-t1-dc/2-sdt);
38         end
39         if V>0
40             sys = real([Q1,Q2,Q3,Q4]);
41         else % se V è minore di zero Q1 si scambia con Q3 e Q2 con Q4
42             sys = real([Q3,Q4,Q1,Q2]);
43         end
44     else
45         sys=[];
46     end
```


Bibliografia

- [1] P. ATKINS, J. DE PAULA; “Chimica Fisica,” *Zanichelli*, Settembre 2004.
- [2] P. BOLZERN, R. SCATTOLINI, N. SCHIAVONI; “Fondamenti di controlli automatici,” *Mc Graw-Hill*, 2008.
- [3] R. G. DRIGGERS, P. COX; T. Edwards; “Introduction to infrared and electro-optical systems,” *Artech House*, 1999.
- [4] G. FERRETTI, G. MAGNANI, P. ROCCO; “Modelling Identification and Compensation of Pulsating Torque in Permanent Magnet AC Motors”, *IEEE Trans. on Industrial Electronics*, Vol. 45, No. 6, pp. 912–920, USA, December 1998.
- [5] P. GRASBLUM; “Using the HCS08 TPM Module in Motor Control Applications,” *Beyond Bits Issue II, Freescale*, pp. 74–80.
- [6] D. HANSELMAN; “Brushless Permanent Magnet Motor Design”, *McGraw-Hill*, 1994.
- [7] G. C. HOLST; “Testing and evaluation of infrared imaging systems,” *JCD Pub.*, 1998.
- [8] G. MAGNANI; “Tecnologie dei sistemi di controllo,” *McGraw-Hill*, Marzo 2000.
- [9] G. MAGNANI, P. ROCCO, A. RUSCONI; “Modeling and position control of a joint prototype of DEXARM,” *IEEE Int. Workshop on Advanced Motion Control*, Trento, Italy, 2008.
- [10] N. MOHAN, T. UNDELAND, W. ROBBINS; “Power Electronics, Second Edition,” *John Wiley & Sons*, 1995.
- [11] K. OHNISHI, M. SHIBATA, T. MURAKAMI; “Motion control for advanced mechatronics”, *IEEE/ASME Trans. on Mechatronics*, Vol. 1, pp. 56–57, 1999
- [12] C. A. SCHULER, M. L. CHUGANI; “Digital signal processing: a hands-on approach,” *McGraw-Hill*, 2005.
- [13] J. VITHAYATHIL; “Power electronics-Principles and applications,” *McGraw-Hill*, Ch. 9, 1995.

Sitografia

- [1] Eurofighter Typhoon, Infra Red Search & Track,
<<http://www.eurofighter.com/capabilities/technology/sensor-fusion/infra-red-search-and-track.html>>.
- [2] Lockheed Martin,IRST,
<<http://www.lockheedmartin.com/products/InfraredSearchTrack/index.html>>.
- [3] MathWorks, SimPowerSystems,
<<http://www.mathworks.com/products/simpower/index.html>>.
- [4] Saab group, Gripen Fighter System,
<<http://www.saabgroup.com/en/Air/Gripen-Fighter-System/>>.
- [5] Selex Galileo,
<<http://www.selexgalileo.com/SelexGalileo/EN/index.sdo>>.

