

... Perché così ho scelto.

-Neo

Dedicato ai miei genitori

Indice

1	Introduzione	1
2	Elementi di Teoria della Stima	5
2.1	Triad	5
2.1.1	Implementazione del Triad Ottimizzato	5
2.1.2	Prove in Laboratorio Software-in-the-loop e sperimentali del Triad	7
2.2	Quest	12
2.2.1	Implementazione del Quest e della relativa matrice di covarianza	13
2.2.2	Prove in Laboratorio Software-in-the-loop e sperimentali del Quest	15
2.3	Elementi di teoria dei filtri di Kalman	19
2.3.1	Generalità sui filtri di Kalman	20
2.3.2	Esempi di filtri polinomiali e non polinomiali	23
2.3.3	Filtri di Kalman estesi	25
2.3.4	Filtri di Kalman linearizzati	26
2.4	Filtri F.I.R. e I.I.R.	29
2.4.1	Le equazioni da implementare	29
2.4.2	Prove sperimentali	30
3	Stima di Assetto e Posizione	33
3.1	Filtro d'assetto AHRS	36
3.2	Filtro di posizione INS	38
3.3	Implementazione del filtro AHRS	40

4	Risultati sperimentali	43
4.1	Misurazioni di deviazioni standard e varianze	44
4.2	Risultati delle prove Software-in-the-loop	45
4.3	Sensoristica e schema di acquisizione in Simulink	49
5	Conclusione e Sviluppi futuri	57
A	Fondamenti di analisi del segnale	61
A.1	La scelta dei 100 Hz	61
A.2	Analisi del segnale acquisito a 100 Hz	62
B	Calibrazione del magnetometro	65
B.1	Algoritmo autocalibrante con norma fissata	66
B.2	Implementazione dell'algoritmo	68
B.3	Risultati sperimentali della calibrazione	70
	Ringraziamenti	75

Elenco delle figure

2.1	Triad	7
2.2	Verifica del Triad	9
2.3	Verifica del Triad con l'utilizzo della piattaforma X-sens	10
2.4	Verifica del Triad con l'utilizzo della piattaforma Crossbow . .	11
2.5	Quest	16
2.6	Verifica del Quest software-in-the-loop	17
2.7	Verifica del quest con il modello UAV	18
2.8	F.I.R.	31
2.9	I.I.R.	32
3.1	Schema Generale	34
3.2	Filtro d'assetto AHRS	40
3.3	Prova del filtro d'assetto AHRS	42
4.1	Prova del filtro	45
4.2	Prova del filtro ingrandita	46
4.3	Particolare del modello in cui si nota il Bias aggiunto ai giro- scopi pari a 0.02 deg/s	47
4.4	Bias dei giroscopi stimato sull'asse delle X	47
4.5	Bias dei giroscopi stimato sull'asse delle Y	48
4.6	Bias dei giroscopi stimato sull'asse delle Z	48
4.7	Unità inerziale ADIS-16350	49
4.8	Magnetometro Applied Physics AP539	50
4.9	Modello Simulink per le prove sperimentali	51
4.10	Piattaforma con unità inerziale e magnetometro	52

4.11	Piattaforma con unità inerziale X-sens	52
4.12	Angolo di rollio- Φ	53
4.13	Angolo di beccheggio- Θ	53
4.14	Angolo di Heading magnetico- Ψ	54
4.15	Angolo di Heading magnetico- Ψ utilizzando output X-sens . .	54
4.16	Ingrandimento che compara il segnale rosso, output del filtro EKF, ed il segnale blu, output della piattaforma X-sens per ciò che riguarda l'angolo di rollio	56
5.1	Libreria creata in Simulink	58
A.1	Prova statica - segnale di output del giroscopio sull'asse delle X	63
A.2	Analisi spettrale del segnale acquisito a 100 Hz	64
B.1	Acquisizione della norma in Simulink	67
B.2	Convergenza del bias misurato sull'asse delle X	71
B.3	Convergenza del bias misurato sull'asse delle Y	71
B.4	Convergenza del bias misurato sull'asse delle Z	72
B.5	La norma del vettore campo magnetico prima e dopo la cali- brazione	72

Elenco delle tabelle

4.1	Deviazioni Standard Misurate a 100 Hz	44
-----	---	----

Capitolo 1

Introduzione

La determinazione d'assetto è una tematica fondamentale affrontata per qualsiasi tipo di velivolo, in particolare per quelli non abitati (*unmanned aerial vehicle*, UAV). La conoscenza dell'assetto non solo permette di stabilire l'orientamento del velivolo nello spazio ma è anche necessaria alle equazioni di meccanizzazione per valutarne la posizione; in linea teorica infatti sistemi di posizionamento satellitari come il GPS sarebbero inutili una volta note le condizioni iniziali (purtroppo la navigazione meramente inerziale è un'utopia a causa degli inevitabili errori di misura in cui si incorre). Inoltre negli aeromobili moderni di aviazione civile, e già da tempo nell'aviazione militare, è presente il sistema fly-by-wire le cui leggi di controllo necessitano una stima precisa dell'assetto del velivolo. Per questo scopo in commercio sono disponibili le piattaforme inerziali (IMU) e sistemi di determinazione d'assetto e di *heading* (AHRS) i quali hanno raggiunto alti standard di precisione che però possono essere gravosi in termini di ingombro e costi. La conoscenza dell'assetto è fondamentale per le leggi di controllo d'assetto e di traiettoria, non solo per i velivoli, ma anche per veicoli sottomarini, navali e terrestri, fino a quelli spaziali: la possibilità di orientare un certo *payload* distingue un satellite controllabile da un 'sasso orbitante'. La tematica della stima d'assetto è largamente trattata in letteratura, in particolare è risultato molto utile [1] per acquisire gli elementi fondamentali della teoria della stima, mentre per quanto riguarda la navigazione inerziale e le tecniche di filtraggio applicabili

si sono consultati [2] e [3]. Per la realizzazione di tale documento sono stati studiati numerosi articoli di cui è stato utilizzato il seguente materiale: le equazioni d'assetto per il filtro AHRS, [4]; l'algoritmo per l'implementazione del TRIAD, [5]; l'algoritmo per l'implementazione del QUEST e della relativa matrice di covarianza, [6]; la tecnica per la calibrazione di un magnetometro, poi tradotta in algoritmo implementabile in ambiente Matlab-Simulink, [7]. Durante l'implementazione degli algoritmi si sono sfruttate molto risorse multimediali tra cui forum della MathWorks, oltre a libri di algebra delle matrici o con appendici riguardanti tale argomento, [8] e [9]. Nel presente documento si descrive l'implementazione, in linguaggio Matlab-Simulink, di un algoritmo di determinazione d'assetto che utilizzi sensori di costo ed ingombro ridotti da installare a bordo di un UAV su cui il sistema è destinato ad essere montato. Il set di sensori è costituito da un GPS, una unità inerziale munita di accelerometri e giroscopi orientati su tre assi ortogonali e un magnetometro; l'unità inerziale ed il magnetometro saranno utili nell'algoritmo di stima dell'assetto e nel relativo filtro di Kalman, il GPS sarà utilizzato nel filtro di posizione al fine di avere in ogni momento una stima di posizione angolare e geografica soddisfacente. Fra i vari tipi di configurazioni è stata scelta una *Loosely coupled Integration* che prevede l'utilizzo dei sensori sopracitati e di due filtri di Kalman estesi (EKF): il filtro AHRS fornisce la stima dell'assetto e del bias dei giroscopi; il filtro INS fornisce la stima della posizione e del bias degli accelerometri. Grazie alla retroazione del bias è possibile compensare il rumore tipico dei sensori *low-cost*. L'idea è quella di utilizzare la piattaforma inerziale per la sua precisione nel breve periodo ed integrarla con il GPS che garantisce una precisione nel lungo periodo. Al fine di ottenere una prima stima dell'assetto, destinata ad essere successivamente filtrata, si è pensato di utilizzare il TRIAD (*Three-Axis Attitude Determination*) e successivamente il QUEST (*Quaternion Estimator*) che sono algoritmi che utilizzano i dati grezzi dei sensori. Il principio dei due algoritmi è quello di risalire all'assetto del velivolo noti due vettori sia nel sistema di riferimento corpo, utilizzando i sensori a bordo, sia nel sistema di riferimento inerziale, utilizzando l'informazione della posizione geografica fornita dal GPS e dati provenienti dai modelli matematici. La differenza

fra i due è nell'informazione in uscita: il TRIAD stima la matrice d'assetto, il QUEST stima il quaternione d'assetto (è stato scelto il QUEST compatibilmente con lo stato del filtro ad esso applicato). L'avionica disponibile permette di avere informazioni riguardanti il vettore accelerazione di gravità e il vettore campo magnetico, mentre in ambiente Simulink è disponibile, da apposite librerie, il vettore accelerazione di gravità locale in assi NED, eventualmente da integrare con accelerazioni dovute alla dinamica del velivolo, e campo magnetico locale in assi NED¹. Per valutare la bontà dell'algoritmo implementato sono state utilizzate delle piattaforme inerziali, prodotti X-sens e Crossbow, disponibili in laboratorio complete di una propria sensoristica e di un proprio algoritmo di stima. Il seguente documento è così strutturato: nel capitolo 2 vengono presentati gli elementi di teoria della stima necessari per affrontare tale trattazione, successivamente, nel capitolo 3 si affronta la struttura globale del sistema di navigazione che fornisce Posizione, Velocità e Assetto, si riportano infine i risultati sperimentali con relative conclusioni e sviluppi futuri; in appendice sono trattati l'analisi del segnale di output dei giroscopi e l'algoritmo di calibrazione del magnetometro.

¹sistema di riferimento composto dai tre assi locali *North, East, Down*

Capitolo 2

Elementi di Teoria della Stima

Fra i vari metodi utilizzati per la stima dell'assetto vengono riportati in questo capitolo il TRIAD nella sua versione ottimizzata ed il QUEST. Il TRIAD utilizza le misure di due vettori nel sistema di riferimento corpo e le compara con quelle dei medesimi vettori nel sistema di riferimento inerziale. Il QUEST procede in maniera analoga, ma invece di esprimere l'assetto tramite una matrice lo definisce utilizzando i quaternioni. Si riportano infine elementi di teoria della stima e dei filtri di Kalman.

2.1 Triad

L'algoritmo Triad o *Three-Axis Attitude Determination* fornisce la matrice di assetto A in grado di esprimere un vettore noto nel sistema di riferimento inerziale, nel sistema assi corpo del veicolo. Tale algoritmo verrà poi scaricato a favore del Quest il quale fornisce il quaternione d'assetto, più utile della matrice di assetto per le motivazioni che verranno esposte nei prossimi capitoli. Si riportano l'algoritmo tratto da [5] e le relative prove sperimentali.

2.1.1 Implementazione del Triad Ottimizzato

Siano \bar{w}_1 e \bar{w}_2 i vettori colonna i cui elementi sono quelli dei due vettori misurati nel sistema di riferimento corpo¹ e siano \bar{v}_1 e \bar{v}_2 i vettori colonna

¹vettore di campo magnetico e di accelerazione di gravità

i cui elementi sono i corrispondenti dei vettori sopracitati nel sistema di riferimento scelto (nel nostro caso il sistema NED²), ciò che si propone di fare il Triad è determinare la matrice A tale che

$$A\bar{v}_i = \bar{w}_i \quad (i = 1, \dots, n) \quad (2.1)$$

Innanzitutto è necessario creare una terna di vettori così definiti usando i vettori derivanti dalle misure a bordo del veicolo

$$\bar{r}_1 = \frac{\bar{w}_1}{|\bar{w}_1|} \quad \bar{r}_2 = \frac{\bar{r}_1 \wedge \bar{w}_2}{|\bar{r}_1 \wedge \bar{w}_2|} \quad \bar{r}_3 = \bar{r}_1 \wedge \bar{r}_2 \quad (2.2)$$

analogamente si procede creando un'altra terna di vettori usando le componenti nel sistema di riferimento NED.

$$\bar{s}_1 = \frac{\bar{v}_1}{|\bar{v}_1|} \quad \bar{s}_2 = \frac{\bar{s}_1 \wedge \bar{v}_2}{|\bar{s}_1 \wedge \bar{v}_2|} \quad \bar{s}_3 = \bar{s}_1 \wedge \bar{s}_2 \quad (2.3)$$

La matrice d'assetto A è quindi data da

$$A = \bar{r}_1 \cdot \bar{s}_1^T + \bar{r}_2 \cdot \bar{s}_2^T + \bar{r}_3 \cdot \bar{s}_3^T \quad (2.4)$$

E' possibile osservare che, con tale formulazione, viene data una maggiore importanza al vettore con pedice uno che non sempre risulta essere il vettore con minore incertezza poichè il rumore che sporca il segnale fornito dal sensore dipende da innumerevoli fattori, spesso imprevedibili. Tuttavia si può individuare una deviazione standard del segnale di output di un certo sensore e tramite l'ottimizzazione dell'algoritmo sopra riportato si può utilizzare quest'informazione per ottenere una stima migliore di entrambi i casi in cui si è cambiato il vettore con pedice uno, detto anche vettore ancora. Tralasciando i passaggi matematici viene di seguito riportata A' che al meglio stima l'assetto del veicolo

$$A' = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} A_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} A_2 \quad (2.5)$$

²North, East, Down

dove σ_1 e σ_2 sono rispettivamente le deviazioni standard dei vettori 1 e 2 mentre A_1 e A_2 sono le due matrici d'assetto calcolate con il Triad utilizzando come ancora il vettore 1 ed il vettore 2. Non è possibile dimostrare analiticamente che A' sia migliore di A_1 e A_2 ma tramite diverse prove fatte in laboratorio utilizzando una unità inerziale X-sens MTi-G è stato verificato.

2.1.2 Prove in Laboratorio Software-in-the-loop e sperimentali del Triad

L'algoritmo descritto è stato implementato in linguaggio Matlab-Simulink (2.1). Il software è stato verificato tramite prove Software-in-the-loop, utilizzando dati di assetto costruiti nell'ambiente di simulazione. Il procedimento consiste nell'utilizzare specifici blocchi Simulink (WMM-2000 e WGS-84) che facendo riferimento a dati tabulati sono in grado di fornire i vettori di campo magnetico e di accelerazione di gravità in un determinato³ luogo insieme ad una DCM (matrice di rotazione) che ruoti tali vettori, si confronta in fine la DCM esatta con quella stimata dal Triad, oppure in maniera equivalente il quaternioni d'assetto vero da quello stimato.

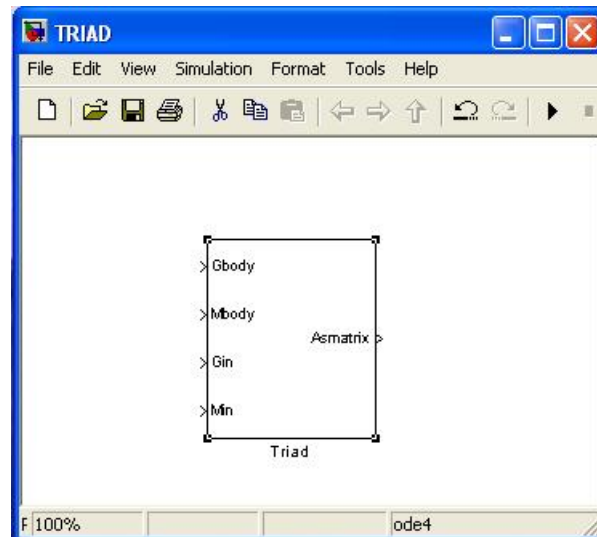


Figura 2.1: Triad

³individuato da coordinate da esprimersi in [rad rad m]

Il risultato che si può apprezzare nella figura 2.2 dimostra la validità del Triad che reagisce bene al rumore impostato che corrisponde ad una σ_1 del vettore accelerazione di gravità di 0.02 m/s^2 ed una σ_2 del vettore di campo magnetico di 0.002 Gauss : dati verosimili trovati studiando il segnale di output della piattaforma X-sens.

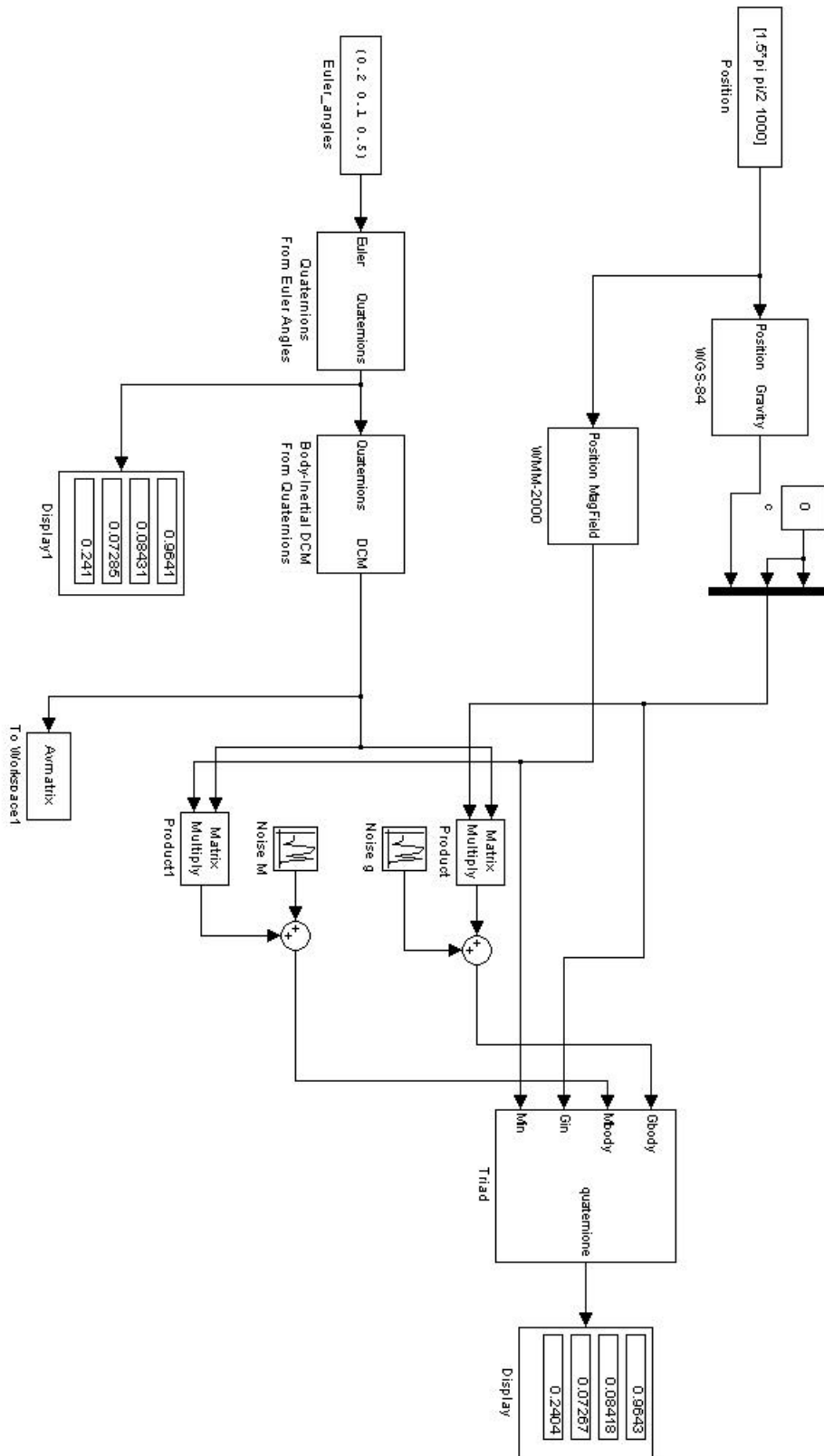


Figura 2.2: Verifica del Triad

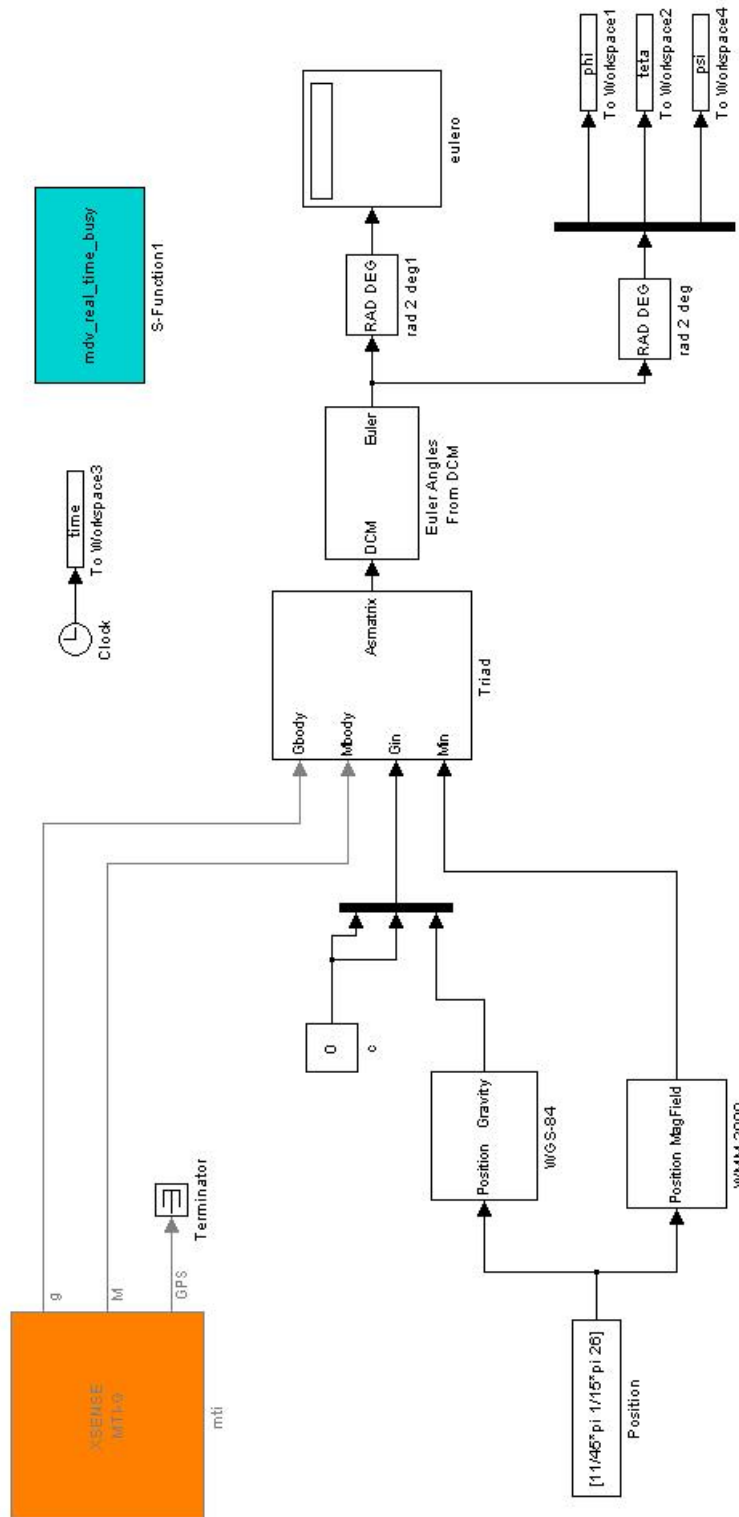


Figura 2.3: Verifica del Triad con l'utilizzo della piattaforma X-sens

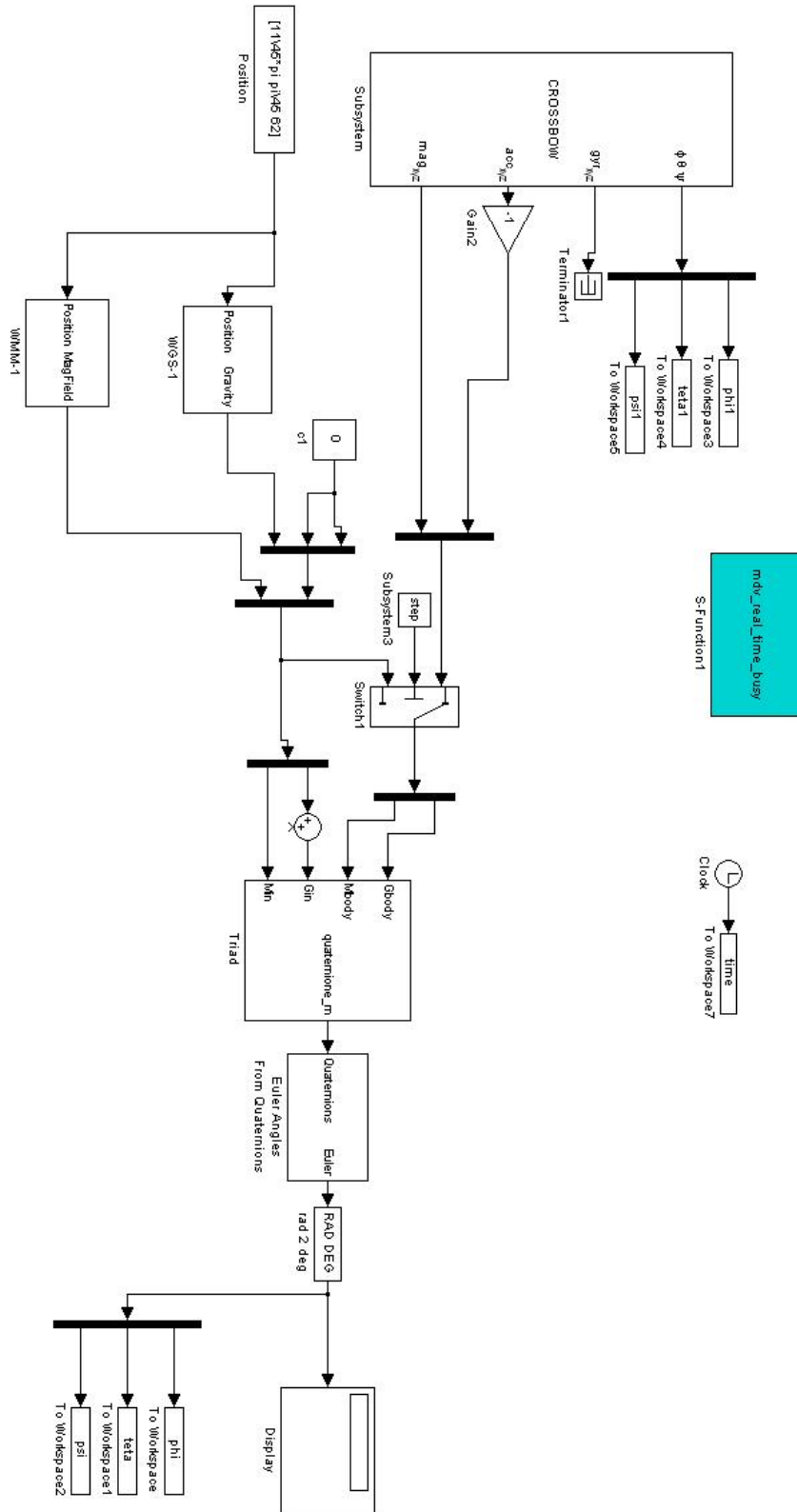


Figura 2.4: Verifica del Triad con l'utilizzo della piattaforma Crossbow

Verificata la validità dell'algoritmo implementato non restava che utilizzare dati veri, così tramite il blocchetto di acquisizione della piattaforma X-sens sono stati utilizzati i vettori di campo magnetico e di accelerazione di gravità forniti dalla piattaforma stessa, 2.3. Prove analoghe sono state fatte utilizzando la piattaforma Crossbow NAV420CA, 2.4. I risultati non sono stati soddisfacenti a causa dei forti disturbi di campo magnetico che si avvertivano in laboratorio⁴ e a causa del magnetometro della piattaforma X-sens che necessitava una calibrazione; tuttavia se la prova veniva eseguita in particolari aree del laboratorio, dove evidentemente i disturbi erano minimi, si riscontrava una buona stima del Triad che si allontanava dal valore stimato dalla piattaforma di pochi gradi, specialmente nelle prove fatte con la Crossbow il cui magnetometro era ben calibrato. In ogni caso è apparso subito necessario l'utilizzo di un filtro di Kalman per migliorare la precisione e rendere il segnale d'assetto più uniforme, privandolo delle discontinuità che presenta a causa del rumore non filtrato.

2.2 Quest

Sebbene lo scopo del Quest o *Quaternion Estimator* sia analogo a quello del Triad vi sono delle differenze nella forma con cui questo descrive l'assetto del veicolo. Viene infatti utilizzato il quaternione d'assetto il cui uso è fondamentale poichè è parte dello stato del filtro di Kalman che completerà la stima dell'assetto. Nonostante esistano algoritmi che permettano di passare dalla forma DCM a quella del quaternione tale procedura è sconsigliata più che per la pesantezza computazionale per le discontinuità che si possono verificare creando gravi errori, anche se istantanei, nella determinazione d'assetto. Per questo motivo nell'elaborato finale è stato adottato il Quest che ha pure un altro vantaggio, anche se non sfruttato dal seguente progetto, essendo in grado nella sua forma completa di far uso di più informazioni vettoriali per ottimizzare la stima; si ricorda che l'algoritmo riportato prevede la forma

⁴in parte successivamente annullati con l'algoritmo di calibrazione riportato in Appendice B

semplificata per due soli vettori. Si riportano l'algoritmo tratto da [6] e le relative prove sperimentali.

2.2.1 Implementazione del Quest e della relativa matrice di covarianza

Si tralasciano i passaggi matematici⁵ e si riporta dapprima la soluzione approssimata per *near null attitude*, semplice, ma valida se e solo se l'assetto è circa nullo⁶. Si crea innanzitutto il vettore Y definito come

$$\begin{aligned} Y = & 0.5\{a_1\bar{w}_1 \wedge \bar{v}_1 + a_2\bar{w}_2 \wedge \bar{v}_2 \\ & + a_1[(b\bar{w}_1 - a\bar{w}_2) \cdot (\bar{v}_1 \wedge \bar{v}_2)]\bar{v}_1 \\ & + a_2[(a\bar{w}_1 - b\bar{w}_2) \cdot (\bar{v}_1 \wedge \bar{v}_2)]\bar{v}_2\} \\ & + o(\delta^2) \quad (2.6) \end{aligned}$$

dove

$$a_i = \frac{\sigma_t^2}{\sigma_i^2} \quad (\sigma_t^2)^{-1} = \sum_{i=1}^n (\sigma_i^2)^{-1} \quad a = \frac{1}{|\bar{v}_1 \wedge \bar{v}_2|^2} \quad b = \frac{(\bar{v}_1 \cdot \bar{v}_2)}{|\bar{v}_1 \wedge \bar{v}_2|^2} \quad (2.7)$$

a questo punto si ricava il quaternione d'assetto q come

$$q = \frac{1}{\sqrt{1 + |Y|^2}} \begin{bmatrix} Y \\ 1 \end{bmatrix} \quad (2.8)$$

L'algoritmo è stato implementato ed è stata verificata la sua validità solo per assetti circa nulli, da cui la necessità di implementare la versione completa che possa stimare un qualunque assetto. Si definisce μ come

⁵le definizioni di \bar{w}_i e \bar{v}_1 sono le medesime del Triad ma i vettori in questo caso sono normalizzati

⁶gli angoli di eulero che descrivono l'assetto sono circa zero

$$\mu = (\bar{v}_1 \cdot \bar{v}_2)(\bar{w}_1 \cdot \bar{w}_2) + |\bar{v}_1 \wedge \bar{v}_2| |\bar{w}_1 \wedge \bar{w}_2| \quad (2.9)$$

da inserire nella formula per trovare λ_{max} che si costruisce come

$$\lambda_{max} = \sqrt{a_1^2 + 2a_1a_2\mu + a_2^2} \quad (2.10)$$

si definisce poi S e $adjS$ come

$$S = \sum_{i=1}^2 a_i (\bar{w}_i \bar{v}_i^T + \bar{v}_i \bar{w}_i^T) \quad adjS = det(S) \cdot I \cdot S^{-1} \quad (2.11)$$

per arrivare alla formulazione di q_{opt} e della relativa matrice di covarianza in una forma semplice si definiscono ancora

$$\sigma = 0.5tr(S) \quad k = tr(adjS) \quad \Delta = det(S) \quad Z = \sum_{i=1}^2 a_i (\bar{w}_i \wedge \bar{v}_i) \quad (2.12)$$

e ancora

$$\alpha = \lambda_{max}^2 - \sigma^2 + k \quad \beta = \lambda_{max} - \sigma \quad \gamma = (\lambda_{max} + \sigma)\alpha - \Delta \quad (2.13)$$

a questo punto q_{opt} è

$$q_{opt} = \frac{1}{\sqrt{1 + |X|^2}} \begin{bmatrix} X \\ 1 \end{bmatrix} \quad (2.14)$$

dove

$$X = (\alpha I + \beta S + S^2)Z \quad (2.15)$$

si riporta infine l'espressione della matrice di covarianza nel sistema di riferimento corpo⁷

⁷matrice 3x3

$$P_{QQ} = 1/4\{\sigma_t^2 I + |\bar{w}_1 \wedge \bar{w}_2|^{-2}[(\sigma_2^2 - \sigma_t^2)\bar{w}_1 \bar{w}_1^T + (\sigma_1^2 - \sigma_t^2)\bar{w}_2 \bar{w}_2^T + \sigma_t^2(\bar{w}_1 \cdot \bar{w}_2)(\bar{w}_1 \bar{w}_2^T + \bar{w}_2 \bar{w}_1^T)]\} \quad (2.16)$$

per avere la matrice di covarianza riferita al quaternione d'assetto⁸, un input importante per il filtro di Kalman, e necessario definire \bar{q} utilizzando le componenti del quaternione d'assetto $q_{opt} = [q_1 \ q_2 \ q_3 \ q_4]$ nel modo seguente

$$\bar{q} = \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \quad (2.17)$$

si può così risalire alla matrice di covarianza P_{qq} nella forma desiderata

$$P_{qq} = [\bar{q}] \begin{bmatrix} P_{QQ} & 0 \\ 0^T & 0 \end{bmatrix} [\bar{q}]^T \quad (2.18)$$

2.2.2 Prove in Laboratorio Software-in-the-loop e sperimentali del Quest

Come per il Triad anche per il Quest è stato adottato il linguaggio Matlab-Simulink ed è stato creato un pacchetto che fornisca come output non solo il quaternione d'assetto, ma anche la relativa matrice di covarianza, figura 2.5. Allo stesso modo è stato verificato l'algoritmo tramite dati simulati con prove Software-in-the-loop, sia con assetto quasi nullo per cui la versione *near null attitude* fornisce risultati soddisfacenti, sia per assetti qualsiasi in cui si è verificata la versione completa. La struttura della verifica ed il risultato sono apprezzabili in figura 2.6, dove sono stati comparati i risultati di Triad e Quest, in definitiva identici.

⁸matrice 4x4

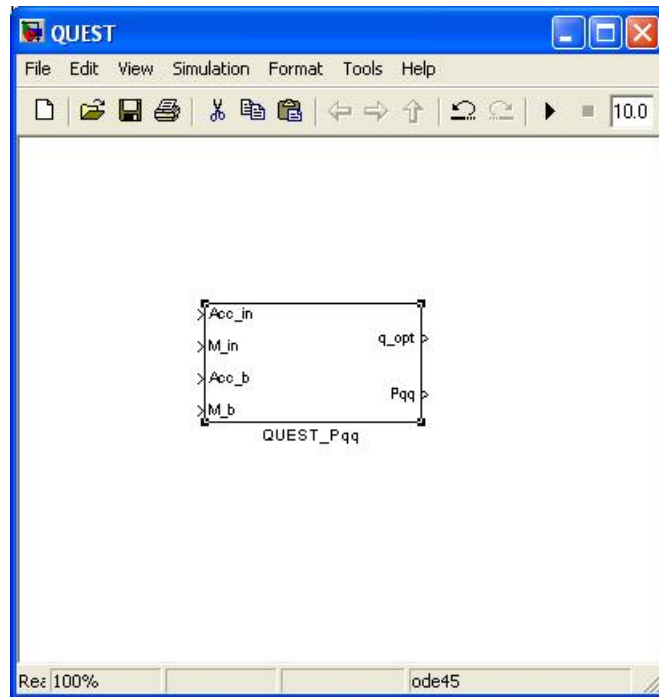


Figura 2.5: Quest

La verifica sperimentale tramite una piattaforma inerziale come crossbow o X-sens, avendo risultati analoghi al Triad, è stata fatta, ma risulta superfluo riportarla. Così è stata eseguita un'ulteriore prova software-in-the loop, figura 2.7, in cui è stato utilizzato un modello di un UAV⁹ disponibile in laboratorio che forniva i dati necessari per stabilire l'assetto tramite il Quest. La medesima prova è stata fatta applicando anche il filtro EKF per cui si analizzeranno in dettaglio i risultati nel capitolo quarto.

⁹Unmanned Aerial Vehicle

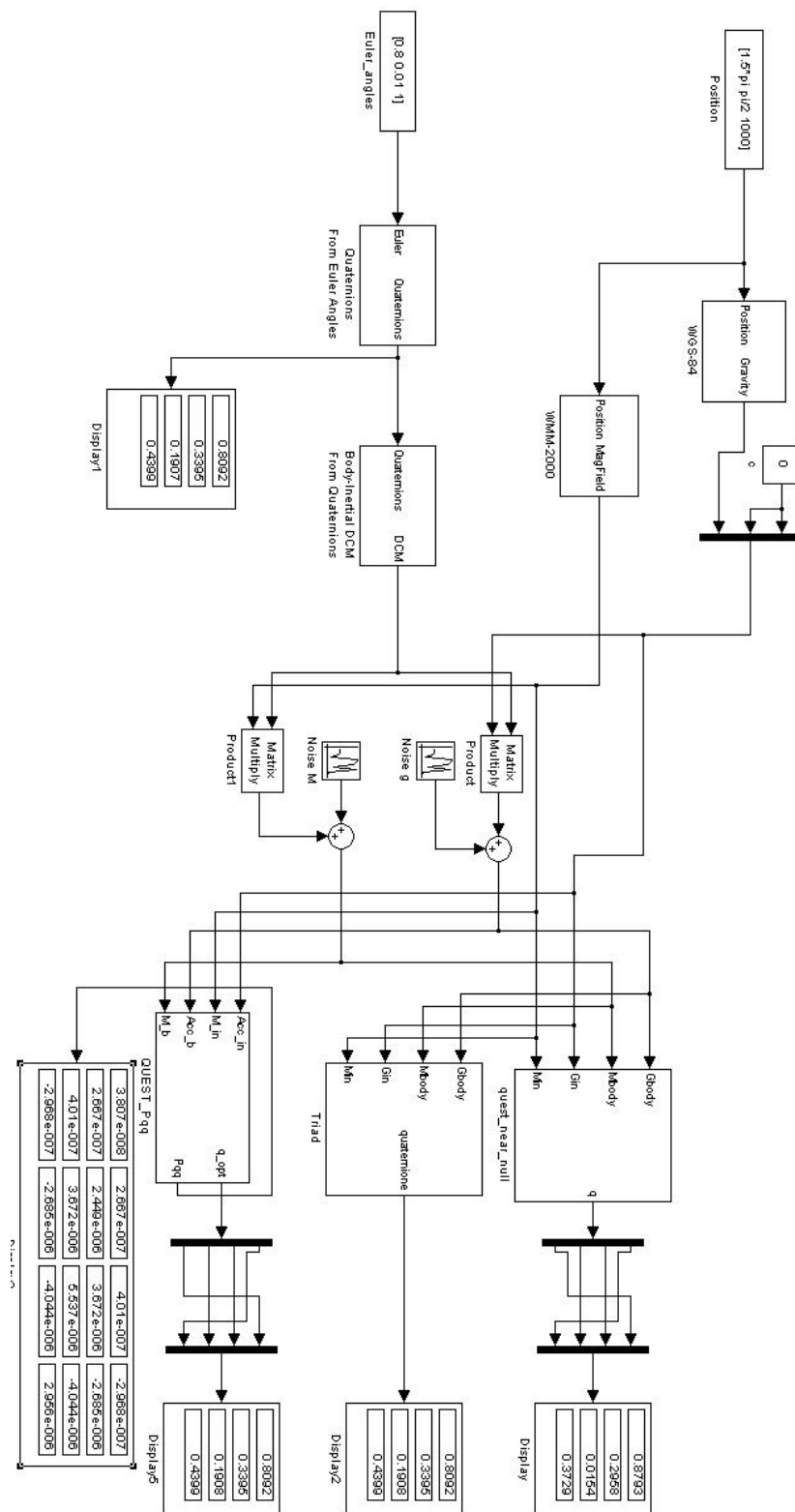


Figura 2.6: Verifica del Quest software-in-the-loop

2.3 Elementi di teoria dei filtri di Kalman

Con gli algoritmi implementati si è già giunti ad una stima d'assetto che, nelle prove Software-in-the-loop, risulta perfetta in assenza di rumore, e comunque ottima con i rumori previsti dai sensori a disposizione. Si è però visto come non appena si passa da dati simulati a dati veri forniti in output dai sensori il rumore risulta essere più instabile del previsto e il bias varia con una certa frequenza portando la stima ad errori che superano i 5 gradi. Questo non è ovviamente ammissibile, per quanto le prove fatte in laboratorio siano condizionate da eccessivi disturbi di campo magnetico, da cui la necessità di approfondire l'argomento integrandolo con i potenti mezzi di filtraggio forniti da Kalman. Il filtro di Kalman che deve essere impiegato si comporta in definitiva come un osservatore dello stato. Se il segnale è affetto da un rumore di cui non si conosce la legge di variazione e non può quindi essere modellizzato e rimosso l'unico modo per cercare di risalire al segnale vero è creare un modello del sistema fisico che tale segnale descrive. In questo modo si ha a disposizione un segnale stimato, output del modello, ed un segnale misurato, output dei sensori, ma sono entrambi affetti da errori (di incertezza del modello il primo e di misura il secondo), bisogna quindi trovare un coefficiente K detto guadagno del filtro che 'pesi' i due segnali per fare in modo di avvicinarsi il più possibile al segnale vero. Dato che il risultato è una media di due segnali sbagliati sarà impossibile arrivare al segnale vero con precisione assoluta, ma un sistema siffatto reagisce molto bene ai rumori dei sensori e permette di avere una stima decisamente più affidabile oltre che un risultato decisamente più continuo¹⁰. I filtri che sono previsti per questo progetto comprendono le equazioni del moto che descrivono la rotazione (AHRS) e le equazioni di meccanizzazione (INS), ma per capire che tipo di filtro potesse essere utilizzato e quale fosse il più conveniente è stata fatta un'analisi preliminare ed uno studio sui filtri, la fonte primaria è stata [1].

¹⁰come verrà sottolineato dai risultati sperimentali con il filtro si passa da una linea spezzata ad una linea continua

2.3.1 Generalità sui filtri di Kalman

Dato un sistema fisico la cui dinamica è descritta da un sistema di equazioni differenziali del primo ordine nella seguente forma¹¹

$$\dot{x} = Fx + Gu + w \quad (2.19)$$

dove x è il vettore colonna di stato del sistema, F è la matrice che rappresenta la dinamica del sistema, u è il vettore degli ingressi (quindi noto), G è la matrice di ingresso, w è infine il rumore bianco di processo, anch'esso rappresentabile come vettore, a cui è correlata la matrice del rumore di processo Q così definita

$$Q = E[ww^T] \quad (2.20)$$

con E che indica il valore medio. Il rumore di processo non ha un significato fisico, ma permette di definire il grado di approssimazione del modello che descrive il sistema fisico studiato. Variando questo parametro ci sono evidenti effetti sul segnale filtrato.

Inoltre è importante che le misure siano collegate linearmente con gli stati del sistema nella forma

$$z = Hx + v \quad (2.21)$$

dove z è il vettore delle misure, x sono gli stati del sistema, v è il rumore bianco dei sensori e H è la matrice delle misurazioni, in alcuni casi¹² uguale alla matrice identità. Allo stesso modo di Q viene definita R come matrice del rumore delle misurazioni

$$R = E[vv^T] \quad (2.22)$$

A questo punto deve essere creata la matrice di trasferimento Φ come

$$\Phi(t) = L^{-1}[(sI - F)^{-1}] \quad (2.23)$$

¹¹si tratta inizialmente il caso lineare

¹²se lo stato coincide con la misura

dove L^{-1} indica l'antitrasformata di Laplace. Tale matrice è fondamentale poichè evita l'integrazione da ogni step delle equazioni differenziali sostituendo tale operazione con una moltiplicazione matriciale, infatti si può dimostrare che per una data Φ si ha

$$x_k = \Phi_k x_{k-1} \quad (2.24)$$

si ricorda che la forma 2.23 vale per i sistemi continui, quindi nella realtà in cui si avrà un'acquisizione ogni T_s secondi è importante passare al caso discreto considerando quindi $\Phi_k = \Phi(T_s)$. Ovviamente anche z e R dovranno essere discretizzate e diventeranno z_k e R_k dato che si riferiranno ad un preciso step k .

Si può ora riportare l'equazione finale che filtra il segnale considerando lo stato filtrato dello step precedente opportunamente propagato con la matrice di trasferimento, eventuali ingressi come perturbazioni o controlli desiderati e lo stato misurato dai sensori

$$\hat{x}_k = \Phi_k \hat{x}_{k-1} + G_k u_{k-1} + K_k (z_k - H \Phi_k \hat{x}_{k-1} - H G_k u_{k-1}) \quad (2.25)$$

dove K_k è la matrice del guadagno di Kalman e G_k si trova come

$$G_k = \int_0^{T_s} \Phi(\tau) G d\tau \quad (2.26)$$

per trovare la matrice del guadagno di Kalman si utilizzano le equazioni di Riccati che si riportano di seguito

$$M_k = \Phi_k P_{k-1} \Phi_k^T + Q_k \quad (2.27)$$

$$K_k = M_k H^T (H M_k H^T + R_k)^{-1} \quad (2.28)$$

$$P_k = (I - K_k H) M_k \quad (2.29)$$

dove P_k è la matrice delle covarianze dopo l'aggiornamento e M_k è la matrice delle covarianze prima dell'aggiornamento. La matrice delle covarianze deve essere fornita dall'utente e dipende dallo stato usato nel filtro, se ad esempio

lo stato è il quaternione d'assetto elaborato dal Quest è necessario utilizzare un algoritmo che crei la matrice delle covarianze partendo dalle note deviazioni standard dei sensori.

Nelle equazioni compare anche la matrice del rumore di processo discretizzata Q_k che è definita come

$$Q_k = \int_0^{T_s} \Phi(\tau) Q \Phi^T(\tau) d\tau \quad (2.30)$$

si è visto quindi come il rumore di processo possa influenzare le equazioni del filtro, dato che non ci si può soffermare sul significato fisico spesso assente si descrivono brevemente gli effetti di questo parametro. Innanzitutto è bene notare come sia l'unico parametro che possa essere modificato arbitrariamente, mentre tutti gli altri sono vincolati al tipo di filtro ed ai sensori utilizzati, questo fa sì che quando si parla di ottimizzazione del filtro si parla proprio del settaggio di Q . Quello che si nota anche dai risultati sperimentali riportati nei capitoli successivi è che con un basso rumore di processo si comunica al filtro che il modello del sistema fisico è estremamente preciso, da cui la possibilità di trascurare in buona parte le misure dirette, con il rischio però che se il modello non è sufficientemente vicino al reale il segnale filtrato possa divergere da quello vero. Alzando il rumore di processo si vuole pesare in maniera più consistente il segnale misurato, si avrà quindi un rumore maggiore sul segnale che sarà in compenso più reattivo e seguirà meglio il segnale vero.

Nel caso semplificato in cui le matrici delle equazioni del filtro siano scalari e non ci siano input ($u_k=0$) è possibile risalire ad una semplificazione delle formule che possa dimostrare analiticamente le equazioni di Riccati. Sebbene non sia scopo di questo documento riportare tale dimostrazione è comunque interessante descrivere il principio base di tali equazioni: calcolare ad ogni step il guadagno K_k che minimizza la matrice delle covarianze del segnale filtrato.

2.3.2 Esempi di filtri polinomiali e non polinomiali

Fra i più semplici segnali che possano essere rintracciati ci sono quelli che possano essere descritti da un polinomio a cui si aggiunge un certo rumore. Se ad esempio il segnale è semplicemente una costante

$$x = a_0 \quad (2.31)$$

la sua derivata sarà nulla

$$\dot{x} = 0 \quad (2.32)$$

e poichè vale

$$\dot{x} = Fx \quad (2.33)$$

si può concludere che $F = 0$, il sistema non ha una dinamica e la matrice di trasferimento sarà $\Phi_k = 1$ in base alla definizione 2.23.

Se invece siamo di fronte ad un polinomio di primo grado, una rampa quindi

$$x = a_0 + a_1 t \quad (2.34)$$

si avrà una derivata prima costante

$$\dot{x} = a_1 \quad (2.35)$$

ed una derivata seconda nulla

$$\ddot{x} = 0 \quad (2.36)$$

si possono rappresentare le equazioni 2.35 e 2.36 tramite le matrici seguenti

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \quad (2.37)$$

forma in cui F è facilmente individuabile

$$F = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (2.38)$$

e la matrice di trasferimento sarà

$$\Phi_k = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \quad (2.39)$$

Supponiamo ora che il segnale non sia polinomiale, ma che sia invece sinusoidale nella forma

$$x = \sin \omega t \quad (2.40)$$

dove ω è la frequenza del segnale. La derivata sarà

$$\dot{x} = \omega \cos \omega t \quad (2.41)$$

prove sperimentali hanno verificato come un segnale siffatto possa essere rintracciato da un filtro del tipo analizzato per i polinomi¹³ ma il risultato è una stima approssimativa, specialmente per quanto riguarda la derivata, quindi utilizzabile solo per certe applicazioni. Il vantaggio di questo tipo di filtro è che in definitiva può rintracciare qualunque tipo di segnale, polinomiale e non, ma se si conosce a priori il fatto che si ha a che fare con una forma sinusoidale è possibile utilizzare un altro tipo di filtro, specifico, in questi casi molto più preciso. Innanzitutto ci si aspetta un segnale del tipo

$$x = A \sin \omega t \quad (2.42)$$

la cui derivata prima sarà

$$\dot{x} = A\omega \cos \omega t \quad (2.43)$$

mentre la derivata seconda sarà

$$\ddot{x} = -A\omega^2 \sin \omega t \quad o \quad \ddot{x} = -\omega^2 x \quad (2.44)$$

ancora una volta è possibile riscrivere le equazioni 2.43 e 2.44 nella forma

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \quad (2.45)$$

¹³meglio utilizzare quello per i polinomi di secondo grado

forma dalla quale si evince facilmente F

$$F = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \quad (2.46)$$

e di conseguenza la matrice di trasferimento

$$\Phi_k = \begin{bmatrix} \cos \omega T_s & \frac{\sin \omega T_s}{\omega} \\ -\omega \sin \omega T_s & \cos \omega T_s \end{bmatrix} \quad (2.47)$$

Un filtro fatto in questo modo ha lo svantaggio di richiedere una conoscenza aprioristica della forma del segnale, ma migliora nettamente le caratteristiche di rintracciabilità sia del segnale puro che della sua derivata.

In questo paragrafo si è quindi visto come costruire un filtro che possa isolare il segnale dal rumore da cui è affetto concentrandosi solo sulla sua forma, senza quindi indagarne la provenienza e senza modellizzare il sistema fisico di cui è output.

2.3.3 Filtri di Kalman estesi

Fin'ora sono stati esaminati filtri di Kalman applicati ad un modello del sistema fisico che può essere descritto da equazioni differenziali lineari, con misure funzioni lineari degli stati. Nella realtà si ha spesso a che fare con modelli più complessi che contengono equazioni differenziali non lineari o magari con misure che non sono direttamente funzione degli stati, in questi casi devono essere utilizzati filtri di Kalman estesi.

Le equazioni differenziali si possono esprimere come una serie di equazioni differenziali non lineari del primo ordine che hanno una forma del tipo

$$\dot{x} = f(x) + w \quad (2.48)$$

dove x è il vettore degli stati, w è il rumore di processo e $f(x)$ è una funzione non lineare. Si può creare allo stesso modo che nei filtri lineare la matrice

del rumore di processo come

$$Q = E(ww^T) \quad (2.49)$$

allo stesso modo anche le misure possono contenere un legame non lineare con gli stati, da cui

$$z = h(x) + v \quad (2.50)$$

dove z è il vettore delle misure, v è il rumore dei sensori e $h(x)$ è una funzione non lineare. Ancora, è possibile creare la matrice del rumore delle misurazioni come

$$R = E(vv^t) \quad (2.51)$$

La differenza fondamentale è che ora, per ottenere F e H è necessario linearizzare le equazioni $f(x)$ e $h(x)$ attorno ad un punto di lavoro, da cui le seguenti espressioni

$$F = \left. \frac{\delta f(x)}{\delta x} \right|_{x=\hat{x}} \quad e \quad H = \left. \frac{\delta h(x)}{\delta x} \right|_{x=\hat{x}} \quad (2.52)$$

a questo punto si utilizzano le stesse equazioni di Riccati che sono state mostrate per i filtri lineari, 2.27, 2.28, 2.29. In pratica quello che viene fatto è lo jacobiano delle funzioni $f(x)$ e $h(x)$, ma la struttura del filtro rimane la stessa.

2.3.4 Filtri di Kalman linearizzati

Esistono particolari problemi in cui ci sono diverse informazioni apriori riguardanti gli stati del sistema molto affidabili che possono essere sfruttate. Un esempio è il caso in cui si debba tracciare l'orbita di un satellite: le equazioni della meccanica orbitale permettono facilmente di prevedere l'orbita del satellite, a patto di piccole variazioni dovute a perturbazioni o a controlli compiuti dal satellite stesso. Si ha quindi a disposizione una soluzione nominale attorno alla quale si linearizza il problema.

Vengono riproposte le equazioni differenziali che descrivono il sistema, già

presentate per i filtri non lineari

$$\dot{x} = f(x) + w \quad (2.53)$$

dove x è al solito il vettore degli stati, w è il rumore di processo e $f(x)$ è una funzione non lineare. Si ha poi la matrice del rumore di processo

$$Q = E(ww^T) \quad (2.54)$$

per quanto riguarda le misure

$$z = h(x) + v \quad (2.55)$$

dove z è il vettore delle misure, v è il rumore dei sensori e $h(x)$ è una funzione non lineare. Infine si individua la matrice del rumore delle misurazioni

$$R = E(vv^t) \quad (2.56)$$

ora, se esiste una soluzione nominale del problema¹⁴ si possono scrivere le relative equazioni differenziali nella forma

$$\dot{x}_{NOM} = f(x_{NOM}) \quad (2.57)$$

dove x_{NOM} rappresenta lo stato espresso dalla soluzione nominale. Si può ora sviluppare un'equazione dell'errore che si definisce come differenza fra lo stato vero e quello nominale in questo modo

$$\Delta x = x - x_{NOM} \quad (2.58)$$

a questo punto è stato introdotto un nuovo stato, Δx , lo stato degli errori. Si può ora scrivere un'equazione differenziale approssimata e linearizzata per

¹⁴come potrebbe essere quella data dalle equazioni che risolvono la traiettoria del satellite

lo stato degli errori nella forma

$$\Delta \dot{x} = \left. \frac{\delta f(x)}{\delta x} \right|_{x=x_{NOM}} \Delta x + w \quad (2.59)$$

allo stesso modo, definito l'errore delle misure come

$$\Delta z = z - z_{NOM} \quad (2.60)$$

ci si può ricondurre all'equazione differenziale approssimata e linearizzata seguente

$$\Delta \dot{z} = \left. \frac{\delta h(x)}{\delta x} \right|_{x=x_{NOM}} \Delta x + v \quad (2.61)$$

si avranno così

$$F = \left. \frac{\delta f(x)}{\delta x} \right|_{x=x_{NOM}} \quad e \quad H = \left. \frac{\delta h(x)}{\delta x} \right|_{x=x_{NOM}} \quad (2.62)$$

in definitiva, poichè anche le equazioni di Riccati non vengono modificate, la struttura del filtro è la stessa, ma ora lo stato cambia, infatti l'output del filtro è l'errore da sommare algebricamente alla soluzione nominale che permette di risalire allo stato vero del sistema. Deve essere quindi presente un'equazione supplementare, tanto banale quanto importante, per trovare lo stato stimato nell'istante k , ovvero

$$\hat{x}_k = \Delta \hat{x}_k + x_{NOM} \quad (2.63)$$

Sono state fatte diverse verifiche sperimentali ed è stato osservato come quando la soluzione nominale è particolarmente accurata il filtro si comporta in maniera ottimale, altrimenti linearizzare attorno ad una soluzione imprecisa porta uno stato filtrato ben lontano da quello reale. Un filtro di questo tipo poteva essere utilizzato nella seconda parte del sistema complessivo di cui si occupa questa tesi: si poteva in effetti considerare la soluzione delle equazioni di meccanizzazione come nominale e trovare l'errore utilizzando il dato GPS. Il rischio è che rumorosi dati di input al sistema di equazioni di meccanizzazione, come un assetto impreciso o una velocità istantanea non corretta,

possano portare ad una soluzione nominale troppo lontana dal reale e quindi linearizzare attorno ad essa sarebbe un errore.

2.4 Filtri F.I.R. e I.I.R.

Un'alternativa ai filtri di Kalman è proposta dai filtri F.I.R. o *Finite Impulse Response* e dai filtri I.I.R. o *Infinite Impulse Response*. Si tratta di filtri molto semplici, ma efficaci, in grado di tagliare determinate frequenze dal segnale. Come viene fatto in Appendice A è possibile fare un'analisi spettrale del segnale, scomponendolo in tutte le frequenze e attribuendo ad esse la relativa ampiezza, in questo modo si possono individuare frequenze più elevate, corrispondenti a rumore, che tramite questi filtri è possibile eliminare.

2.4.1 Le equazioni da implementare

La differenza fondamentale fra FIR e IIR è evidenziata nel nome stesso, infatti un filtro FIR ha una risposta ad un impulso finita, cioè ricevuto un impulso in un intervallo di tempo finito elimina ogni traccia di esso, mentre un filtro IIR ne conserva memoria per sempre. Tali filtri sono detti filtri a media mobile, il motivo è facilmente individuabile dalla loro forma analitica, si riporta l'esempio di un semplice FIR

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j] \quad (2.64)$$

dove $y[i]$ è l'output del campione i -esimo, $x[i]$ è l'input del campione i -esimo e M è il numero di campioni utilizzati dal filtro. Quello che in effetti viene fatto è una media fra il campione appena ricevuto e gli M campioni che lo precedono, in questo modo il picco del rumore viene limato e si ha un segnale più stabile. Tuttavia la sua semplicità ha un costo, infatti, al contrario dei filtri di Kalman, con l'aumentare della sua efficacia, che cresce con il numero di campioni M utilizzati, aumenta il ritardo del segnale filtrato rispetto a quello vero. Questo effetto collaterale è intuitivo poichè se viene mediato il segnale appena ricevuto con quelli passati viene attenuato non solo il rumore

ma anche il segnale stesso, se ad esempio si ha una rampa, prima che il filtro la riconosca deve ricevere un segnale di input abbastanza forte da ‘vincere’ il peso che viene dato ai campioni precedenti. Per facilitare tale operazione è possibile pesare i vari campioni con opportuni coefficienti, passando così da una media semplice ad una media ponderata arrivando alla forma completa di un IIR

$$y[i] = b(1) \cdot x[i] + b(2) \cdot x[i - 1] + \dots + b(ib + 1) \cdot x[i - ib]$$

- $a(2) \cdot y[i - 1] - \dots - a(ia + 1) \cdot y[i - ia]$ (2.65) Si può notare come le cose siano già più complicate, innanzitutto compaiono i coefficienti sopracitati che insieme al numero di campioni vanno ad individuare una precisa frequenza di taglio, inoltre la media ponderata è fatta non solo con campioni di input passati, ma anche con campioni di output dei passaggi precedenti sempre al fine di ottimizzare l’efficacia del filtro senza dover accettare un eccessivo ritardo del segnale filtrato.

Sono state fatte diverse prove sperimentali con dati simulati e dati veri acquisiti dalle piattaforme inerziali ed è stata verificata la validità di tali strumenti, ma lo scopo di questo documento è di arrivare ad una determinazione d’assetto la più precisa possibile, il che implica l’adozione dei più complessi e più efficaci filtri di Kalman. Oltretutto l’argomento in questione è largamente trattato in letteratura e sono già disponibili le equazioni dei filtri da utilizzare.

2.4.2 Prove sperimentali

Si riportano ora le strutture fondamentali dei filtri FIR e IIR utilizzati

In figura 2.8 è riportato un semplice FIR implementato in linguaggio Matlab-Simulink che media l’ultimo campione ricevuto con i 5 precedenti, nel blocchetto *function* è possibile fare una semplice media oppure una media ponderata con opportuni coefficienti. Si può inoltre aumentare il numero di campioni trattati dal filtro ed i risultati sono decisamente visibili. Nelle prove sperimentali è stato usato un segnale con errore random ed è stata misurata la deviazione standard nei vari casi. Senza filtro si vedeva una deviazione

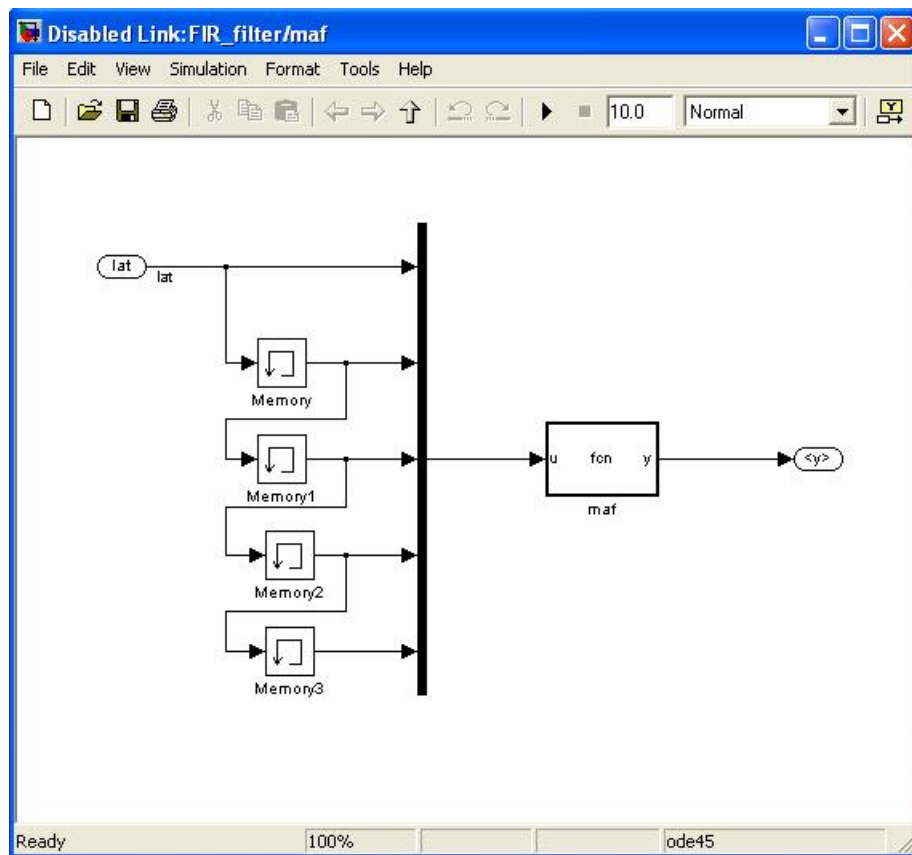


Figura 2.8: F.I.R.

standard di 0.8237 che calava a 0.3110 con un FIR di 5 campioni e arrivava a 0.1982 con un FIR di 10 campioni. Bisogna ricordare però che l'aumentare del numero di campioni provoca un ritardo del segnale che è diversamente ammissibile in base al tipo di applicazione.

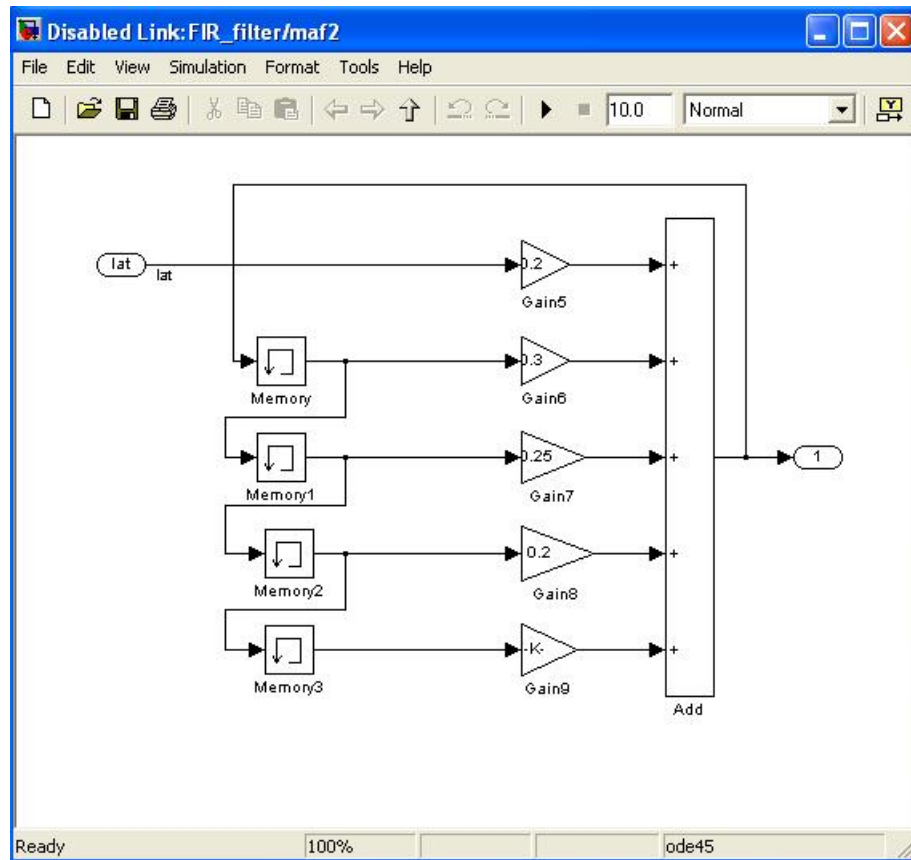


Figura 2.9: I.I.R.

Nella figura 2.9 è invece riportato un filtro I.I.R. che pesa 5 campioni. Si nota come a differenza del filtro in figura 2.8 vengano pesati i campioni filtrati precedentemente e non quelli grezzi di input. Ciò comporta un miglioramento delle prestazioni del filtro, tanto che con 5 campioni si ottiene circa lo stesso risultato di un F.I.R. che media 10 campioni. Come al solito è necessario controllare il ritardo del segnale filtrato da quello vero.

Capitolo 3

Stima di Assetto e Posizione

Dopo aver acquisito gli elementi necessari sono state studiate le possibili configurazioni dei filtri utilizzabili. Mentre per quanto riguarda il filtro AHRS si è subito pensato di usare un filtro esteso di Kalman che abbia come stato il quaternioni d'assetto, per il filtro INS che fornisce posizione e velocità ricevendo il segnale GPS e la soluzione delle equazioni di meccanizzazione sono state esplorate varie strade. In particolare è stata considerata la possibilità di utilizzare un filtro dell'errore, quindi un filtro esteso linearizzato attorno alla soluzione nominale delle equazioni di meccanizzazione, in alternativa ad un filtro esteso¹. Nel caso in cui si abbia una soluzione nominale molto vicina al reale la struttura del filtro esteso linearizzato sicuramente permette risultati migliori, ma i sensori che si hanno a disposizione sono a basso costo e quindi con standard non elevati di accuratezza, ciò comporta un input impreciso e molto variabile a causa del rumore alle equazioni di meccanizzazione che forniranno una soluzione non affidabile, da cui l'impossibilità di utilizzarla come soluzione nominale. La scelta è stata quindi per un filtro esteso che tenesse conto del fatto che mentre il dato della piattaforma inerziale ha una frequenza di campionamento di 100 Hz il dato GPS è campionato a 5 Hz o al massimo a 10 Hz. La configurazione adottata è la *Loosely Coupled Integration* riportata in figura 3.1.

¹per le differenze si richiama la trattazione al Capitolo 2

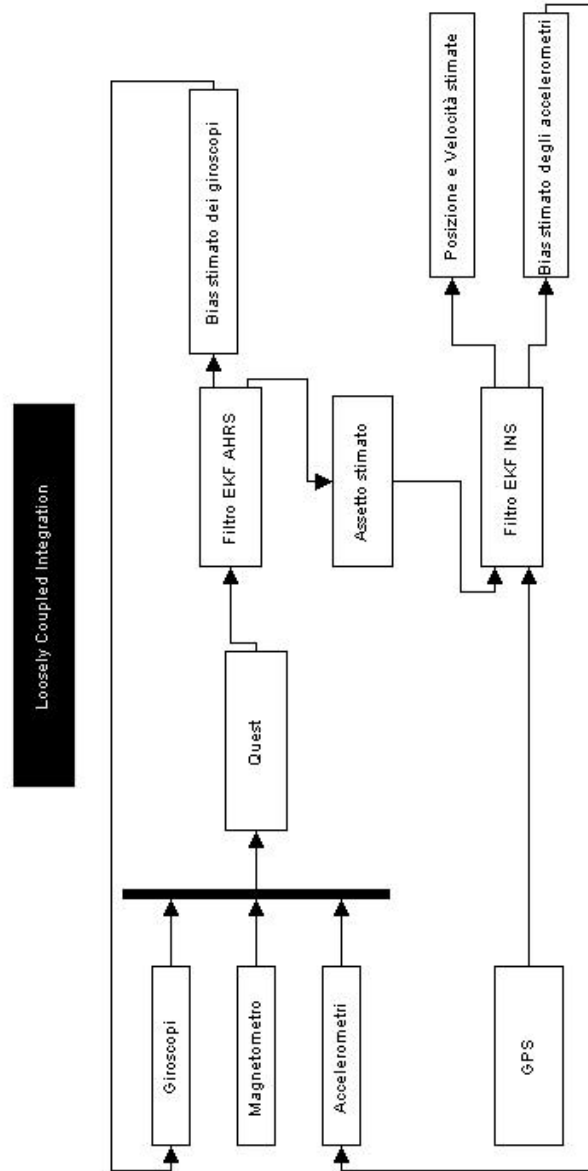


Figura 3.1: Schema Generale

Nello schema generale riportato si distinguono due input, dati della piattaforma inerziale e GPS, e tre output tra cui assetto, posizione e velocità stimati. Il percorso logico parte dalla piattaforma inerziale composta da giroscopi e accelerometri disposti su 3 assi ortogonali e da un magnetometro, tali dati sono sufficienti al Quest per elaborare una prima stima dell'assetto. Infatti il dato di campo magnetico e accelerazione di gravità misurati in assi corpo è confrontato con i medesimi vettori in assi NED, disponibili grazie ai blocchetti Simulink che contengono un aggiornato modello della terra e necessitano soltanto della posizione, fornita dal GPS², in latitudine, longitudine ed altezza con le prime due coordinate in gradi e la terza in metri. A questo punto viene utilizzato il filtro AHRS che riceve il quaternione d'assetto elaborato dal Quest e i dati dei giroscopi, vengono inoltre applicate le equazioni del moto per la rotazione che si possono scrivere nella forma seguente di cui si tratterà in seguito

$$\dot{Q} = \frac{1}{2}\Omega Q \quad (3.1)$$

così EKF-AHRS è in grado di filtrare il dato d'assetto riducendone notevolmente il rumore, come verrà riportato nei risultati sperimentali. Il filtro sarà anche in grado di stimare il bias dei giroscopi che andrà a correggere l'output di tali sensori³. Ora, una volta disponibile l'assetto stimato e il dato GPS il filtro INS applicando le equazioni di meccanizzazione è in grado di stimare Posizione e Velocità del veicolo. Allo stesso modo del filtro d'assetto anche il filtro di posizione stima il bias, degli accelerometri in questo caso, che aggiorna l'output di questi sensori. Come già ricordato i *feedback* presenti in questa configurazione complicano la struttura ma sono necessari per ovviare al fatto che si hanno a disposizione sensori con bassi standard qualitativi e con rilevanti e mutevoli bias da correggere.

²il GPS sarà importante anche per calcolare le accelerazioni dovute alla dinamica del centro di massa da aggiungere al dato in output dai blocchetti Simulink

³si vedrà in Appendice B come sia necessario questo passaggio

3.1 Filtro d'assetto AHRS

Vengono ora riportate le equazioni implementate del filtro AHRS utilizzato, tratte da [4] e da [2]. Nel caso analizzato si deve trattare un sistema dinamico non lineare che si può modellizzare nella forma generale

$$\dot{x} = f(x, \omega) + w \quad (3.2)$$

dove $x = [q \ b_\omega]^T \in R^{7 \times 1}$ è lo stato del filtro composto dal quaternione d'assetto nella particolare forma $q = [\bar{q} \ q_4]^T \in R^{4 \times 1}$, in cui quindi si ha prima la parte vettoriale e poi quella scalare, e dal bias $b_\omega \in R^{3 \times 1}$ dei giroscopi. Si ha poi $\omega = [p \ q \ r] \in R^{3 \times 1}$ che sono le note velocità angolari ed il solito rumore di processo w . Inoltre

$$y = h(x) + v \quad (3.3)$$

dove la misura y è formata dal quaternione d'assetto, $y = [q]^T \in R^{4 \times 1}$, e v è il rumore delle misurazioni. La formulazione della funzione non lineare $f(x)$ viene riportata di seguito

$$f(x, \omega) = 1/2 \begin{bmatrix} q_4 \cdot (p - bp) - q_3 \cdot (q - bq) + q_2 \cdot (r - br) \\ q_3 \cdot (p - bp) + q_4 \cdot (q - bq) - q_1 \cdot (r - br) \\ -q_2 \cdot (p - bp) + q_1 \cdot (q - bq) + q_4 \cdot (r - br) \\ -q_1 \cdot (p - bp) - q_2 \cdot (q - bq) - q_3 \cdot (r - br) \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.4)$$

si può notare come non sia altro che l'equazione generale della dinamica di rotazione con la differenza che viene introdotto il bias ed inoltre si adotta la particolare notazione del quaternione formato dalla parte vettoriale e poi da quella scalare.

Se tronchiamo lo sviluppo di Taylor della matrice di trasferimento al secondo

membro otteniamo una facile forma del tipo

$$\Phi = I + FT_s \quad (3.5)$$

dove I è la matrice identità, T_s è il tempo di campionamento ed F è la matrice che rappresenta la dinamica del sistema, nel caso dei filtri estesi è lo yacobiano di $f(x)$ quindi sarà

$$F = 1/2 \begin{bmatrix} 0 & r - b_r & -q + b_q & p - b_p & -q_4 & q_3 & -q_2 \\ -r + b_r & 0 & p - b_p & q - b_q & -q_3 & -q_4 & q_1 \\ q - b_q & -p + b_p & 0 & r - b_r & q_2 & -q_1 & -q_4 \\ -p + b_p & -q + b_q & -r + b_r & 0 & q_1 & q_2 & q_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.6)$$

per quanto riguarda la matrice $h(x)$ nel caso trattato corrisponde alla semplice matrice identità moltiplicata per lo stato, ma è stata valutata la possibilità di utilizzare un vettore delle misure alternativo, della stessa forma suggerita dall'articolo [4], $y = [a \ \psi_{mag}] \in R^{4 \times 1}$ dove a è il vettore delle accelerazioni misurate dagli accelerometri e ψ_{mag} è l'angolo di *heading* magnetico. In questo modo si avrebbe

$$h(x) = \begin{bmatrix} 2g(q_1q_3 - q_4q_2) \\ 2g(q_2q_3 + q_4q_1) \\ g(q_4^2 - q_1^2 - q_2^2 + q_3^2) \\ \tan^{-1}\left(\frac{2(q_1q_2 + q_4q_3)}{q_4^2 + q_1^2 - q_2^2 - q_3^2}\right) \end{bmatrix} \quad (3.7)$$

dove g è il modulo dell'accelerazione di gravità avvertita. Un sistema di questo tipo potrebbe essere utilizzato⁴ nel momento in cui è disponibile il segnale GPS (campionato a 5 Hz e non ai 100 Hz della piattaforma) poichè in quell'istante è anche corretto il bias degli accelerometri e la misura y definita come sopra sarebbe più precisa. Si ricorda inoltre che nelle equazioni di

⁴in Simulink questo è realizzabile tramite un blocchetto di switch

Riccati (2.27, 2.28, 2.29), è presente non $h(x)$ ma H che nel caso dei filtri estesi è lo yacobiano di $h(x)$.

3.2 Filtro di posizione INS

Anche per il filtro di posizione il sistema dinamico descritto è non lineare da cui si ha

$$\dot{x} = f(x, a) + w \quad (3.8)$$

dove lo stato $x = [p \ v \ b_a] \in R^{9 \times 1}$ è un vettore colonna composto dalla posizione, $p = [Lat \ Long \ Alt]$, dalla velocità in assi NED, $v = [v_N \ v_E \ v_D]$ e dal bias degli accelerometri. Inoltre $a = [a_N \ a_E \ a_D]$ indica il vettore contenente le accelerazioni in assi NED e w è ancora il rumore di processo. Si ha poi

$$y = h(x) + v \quad (3.9)$$

con $y = [p \ v] \in R^{6 \times 1}$ vettore delle misure contenente posizione e velocità misurate dal GPS e v rappresenta il rumore delle misurazioni. La funzione non lineare $f(x, a)$ derivante dalle equazioni di meccanizzazione è la seguente

$$f(x, a) = \begin{bmatrix} \frac{v_N}{R_\lambda + h} \\ \frac{v_E}{(R_\Phi + h) \cos \lambda} \\ \frac{-v_E^2 \sin \lambda}{(R_\Phi + h) \cos \lambda} + \frac{v_N v_D}{R_\lambda + h} + a_N - b_{a_N} \\ \frac{v_E v_N \sin \lambda}{(R_\Phi + h) \cos \lambda} + \frac{v_E v_D}{R_\lambda + h} + a_E - b_{a_E} \\ \frac{-v_E^2}{(R_\Phi + h)} - \frac{v_N^2}{R_\lambda + h} + g + a_D - b_{a_D} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.10)$$

mentre la forma di $h(x)$ è molto semplice data la corrispondenza quasi perfetta tra stato e misure, da cui $h(x) = [I^{6 \times 6} | 0^{6 \times 3}]x$.

Per comodità di visualizzazione si riportano di seguito le equazioni dei filtri estesi utilizzati suddivise nelle due parti fondamentali di Previsione e Correzione.

Previsione

$$\hat{x}_k(-) = \Phi_{k-1} \hat{x}_{k-1}(+) \quad (3.11)$$

$$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi_{k-1}^T + Q \quad (3.12)$$

dove $\Phi_k = e^{\left. \frac{\delta f}{\delta x} \right|_k \Delta t}$ ovvero troncandone lo sviluppo di Taylor al secondo membro $\Phi_k = I + F_k T_s$

Correzione

$$\hat{x}_k(+) = \hat{x}_k(-) + K_k [y_k - h(\hat{x}_k(-))] \quad (3.13)$$

$$P_k(+) = [I - K_k H_k] P_k(-) \quad (3.14)$$

$$K_k = P_k(-) H_k^T [H_k P_k(-) H_k^T + R]^{-1} \quad (3.15)$$

dove $H_k = \left. \frac{\delta h}{\delta x} \right|_k$

3.3 Implementazione del filtro AHRS

Mentre per il filtro di posizione è stato usato un filtro già presente nel laboratorio di meccanica del volo il filtro d'assetto AHRS è stato implementato in linguaggio Matlab-Simulink ed è stato creato il blocchetto di figura 3.2

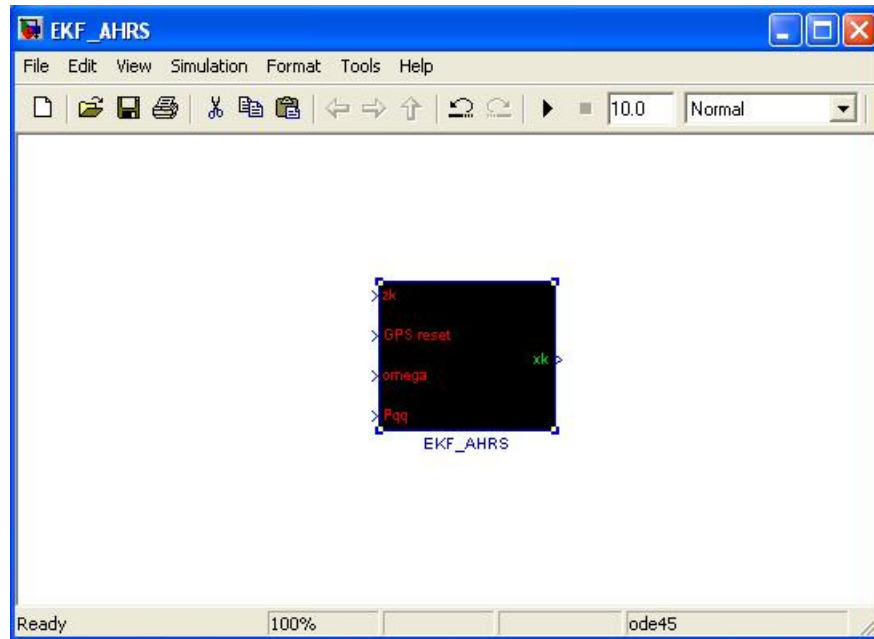


Figura 3.2: Filtro d'assetto AHRS

Per testare il filtro è stato utilizzato un modello di un UAV presente in laboratorio che sfortunatamente non forniva in output il dato di campo magnetico misurato a bordo, necessario per la determinazione d'assetto tramite Triad o Quest, che però può essere ricostruito tramite il dato di campo magnetico in assi NED opportunamente moltiplicato per la DCM⁵ d'assetto che invece il modello fornisce come output. Lo schema è riportato in figura 3.3 e viene fatto notare come il sistema utilizzato faccia uso anche del Quest che in cascata al filtro fornisce ottimi risultati, riportati nel capitolo successivo. Il sistema reagisce bene al rumore introdotto, che nello schema è prodotto dai blocchetti di *Random White Noise*, smorzandolo e riducendo la deviazione

⁵quella fornita dal modello è la DCM esatta da confrontarsi con la DCM o il quaternioni d'assetto elaborato dal sistema AHRS

standard media del segnale in output. Un'altra cosa che è possibile notare in figura 3.3 è il blocchetto che è stato introdotto per derivare le velocità misurate dal GPS risalendo alle accelerazioni in assi NED del centro di massa del veicolo. Questo è superfluo in condizioni quasi statiche, ma se il veicolo subisce accelerazioni apprezzabili dovute ad esempio ad un aumento di velocità o ad una manovra allora il vettore di accelerazioni misurate a bordo dai sensori deve essere accoppiato con il vettore $[0 \ 0 \ g]$ a cui devono essere sommate le accelerazioni in assi NED che si possono appunto ricavare derivando le velocità del GPS. Il problema che si può presentare è che derivando un segnale rumoroso è possibile ottenere dati tutt'altro che affidabili, da cui l'idea di inserire blocchi di saturazione che tengano conto dell'inviluppo di volo del velivolo o nel caso di un veicolo generico delle accelerazioni massime che ci si possa aspettare, e la derivazione non ad ogni step ma fatta in modo tale da fornire un segnale che si aggiorni ad esempio con una frequenza di 1 Hz, ciò è ragionevole dato che le accelerazioni subite a causa di manovre saranno molto gradualmente e permetterebbe di limare i picchi istantanei di accelerazione dovuti al rumore delle velocità del GPS. Un ultima considerazione è fatta sui blocchi che sono posti alla fine dello schema⁶, i quali non verranno utilizzati nella configurazione software finale ma che in fase di test si sono rilevati molto utili per acquisire il segnale di output, averlo disponibile in ambiente Matlab ed andare ad analizzarlo con metodi più accurati di una semplice visione del segnale in un blocco *scope* di Simulink.

⁶blocchi *To File* e *To Workspace*



Capitolo 4

Risultati sperimentali

In questa sezione verranno riportate tutte le prove eseguite con i relativi risultati riguardanti il Filtro EKF-AHRS integrato con il Quest. Essendo il filtro EKF-INS già disponibile in laboratorio ed essendo stato già testato con risultati positivi non è stato ritenuto necessario inserirlo nel *loop* verificato nelle prove sperimentali, soprattutto per evitare di dover fare prove all'aperto¹. Infatti un limite rilevante della fase sperimentale è stato quello di non poter uscire dal laboratorio sia a causa di problemi software riguardanti la versione Matlab-2008a nel computer portatile a disposizione, sia a causa della rete di acquisizione della piattaforma inerziale che comunica mediante porta parallela presente soltanto nei PC del laboratorio. Non poter fare prove all'aperto, oltre che non dare la possibilità di testare lo schema completo di figura 3.1, ha portato l'inevitabile manifestarsi di errori dovuti a disturbi di campo magnetico che hanno quindi impedito di andare a certificare un grado di precisione del sistema complessivo. Per questo motivo si lasciano i test fuori dal laboratorio e la conseguente ottimizzazione del filtro agli sviluppi futuri del sistema implementato.

¹per acquisire il segnale GPS

4.1 Misurazioni di deviazioni standard e varianze

Sebbene le deviazioni standard dei sensori siano certificate nei *data-sheet* dei prodotti è stato ritenuto opportuno misurarle direttamente per verificare le specifiche comunicate. I dati riguardanti l'imu X-sens Mti-G sono riportati nella tabella seguente

Tabella 4.1: Deviazioni Standard Misurate a 100 Hz

Accelerometri [m/s^2]	Giroscopi [rad/s]	Magnetometro [$Gauss$]
STD(AccX)=0.013	STD(GyrX)=0.0093	STD(MagX)=0.0008
STD(AccY)=0.009	STD(GyrY)=0.0093	STD(MagY)=0.0009
STD(AccZ)=0.017	STD(GyrZ)=0.0076	STD(MagZ)=0.0026

I risultati riportati in 4.1 sono con ottima approssimazione gli stessi della piattaforma inerziale utilizzata per le prove sperimentali, anch'essa acquisita a 100 Hz, e sono stati trovati collegando la X-sens al PC mediante porta USB², acquisendo dati e maneggiandoli in *post-processing* in ambiente Matlab. Si ricorda che la varianza della misura si costruisce nel seguente modo

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \quad (4.1)$$

dove σ è la deviazione standard, n è il numero di dati trattati che hanno valore numerico x_i e la cui media è μ .

²utilizzando un convertitore USB-SERIALE

4.2 Risultati delle prove Software-in-the-loop

Il modello di simulazione presentato in figura 3.3 ha portato risultati interessanti grazie ai quali è stata verificata la validità del filtro ed inoltre sono state provate diverse configurazioni di *process noise* cercando quella che avrebbe ottimizzato il risultato.

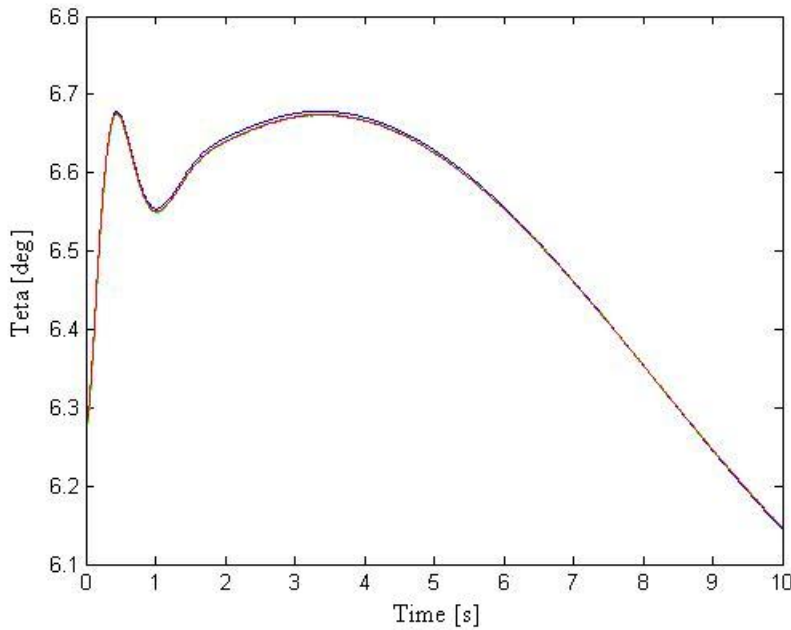


Figura 4.1: Prova del filtro

Dalla figura 4.1 si può notare come il segnale filtrato, in rosso, ricalchi perfettamente il segnale vero, in blu. Questo in realtà dice solo che in effetti il sistema Quest connesso al filtro AHRS fornisce un risultato verosimile e non diverge dalla realtà, ma non basta per giudicare l'operato del filtro.

Analizzando la figura 4.2, un ingrandimento della figura 4.1, è possibile invece apprezzare il lavoro compiuto dal filtro. In blu è riportato il segnale vero, in rosso il segnale filtrato ed in verde il segnale di output del Quest. La prima osservazione può riguardare il fatto che il Quest, e di conseguenza il filtro, abbiano un output che si scosta dal segnale vero, ma bisogna notare come questo scostamento sia dell'ordine del millesimo di grado ed è probabil-

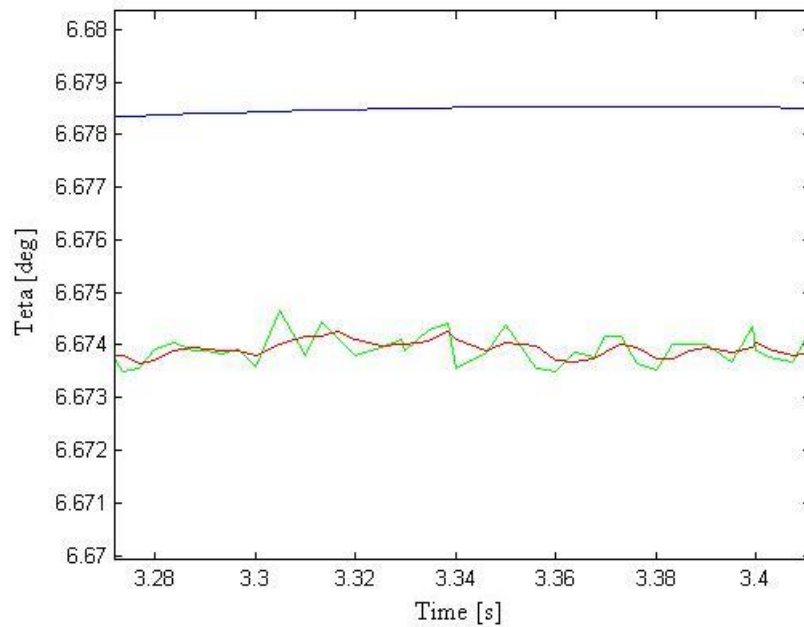


Figura 4.2: Prova del filtro ingrandita

mente causato dal fatto che nel modello non si tiene conto delle accelerazioni in assi NED del velivolo. Detto ciò si può notare come il filtro in effetti riesca a ridurre il rumore sul segnale in output del Quest riducendone la deviazione standard.

L'altro compito del filtro d'assetto è quello di stimare il bias dei giroscopi al fine di avere una navigazione inerziale più precisa negli intervalli di acquisizione in cui non è disponibile il segnale GPS. Si mostra innanzitutto in figura 4.3 un particolare del modello in cui è stato inserito un rumore a media non nulla, quindi con bias, del valore di 0,02 deg/s. Nelle figure successive, 4.4 4.5 4.6, si nota come in effetti il filtro ritrovi quel valore con uno scostamento relativamente trascurabile, ovviamente dipendente dal rumore inserito. Il tempo di risposta, la sovraelongazione e l'oscillazione intorno al valore vero sono parametri migliorabili mediante l'ottimizzazione del filtro e la relativa scelta del *process noise*.

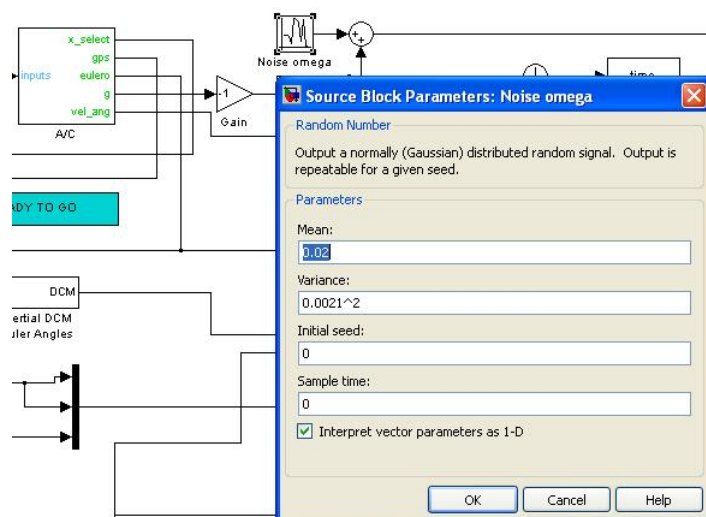


Figura 4.3: Particolare del modello in cui si nota il Bias aggiunto ai giroscopi pari a 0.02 deg/s

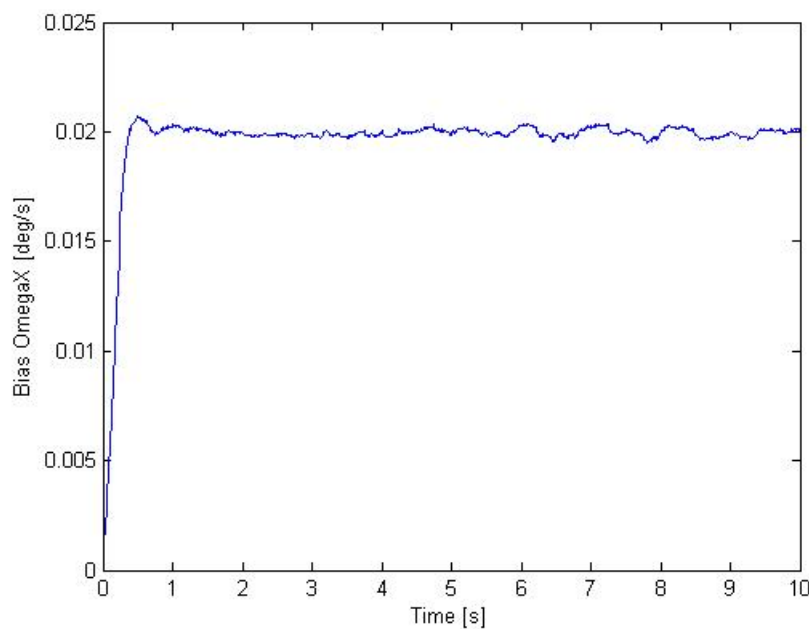


Figura 4.4: Bias dei giroscopi stimato sull'asse delle X

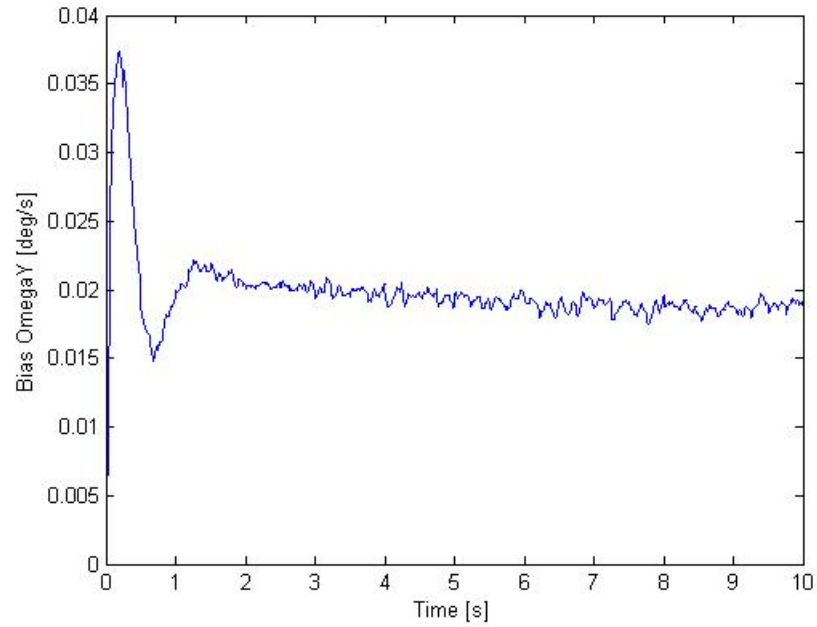


Figura 4.5: Bias dei giroscopi stimato sull'asse delle Y

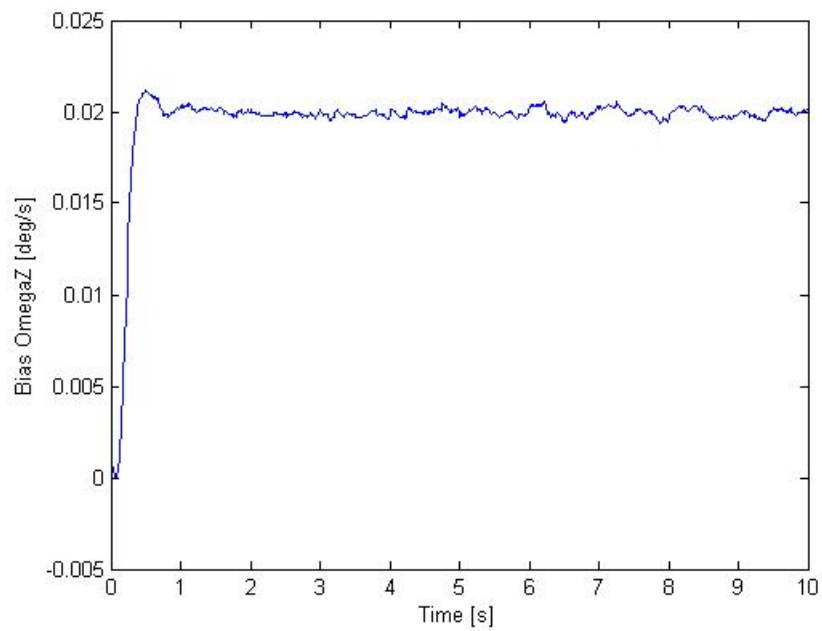


Figura 4.6: Bias dei giroscopi stimato sull'asse delle Z

4.3 Sensoristica e schema di acquisizione in Simulink

In questo paragrafo vengono descritti i sensori utilizzati per le prove sperimentali. Le stime d'assetto ottenute con il sistema sviluppato nell'ambito del presente lavoro di tesi sono confrontate con quelle fornite dalla piattaforma X-sens MTi-G disponibile in commercio. In figura 4.7 sono mostrati i sensori, accelerometri e giroscopi, utilizzati. I dati dei sensori sono acquisiti a 100 Hz e sono filtrati internamente da un filtro FIR che ne attenua in parte il rumore. Non è presente alcun software di determinazione d'assetto come quelli contenuti nelle piattaforme inerziali complete in commercio.

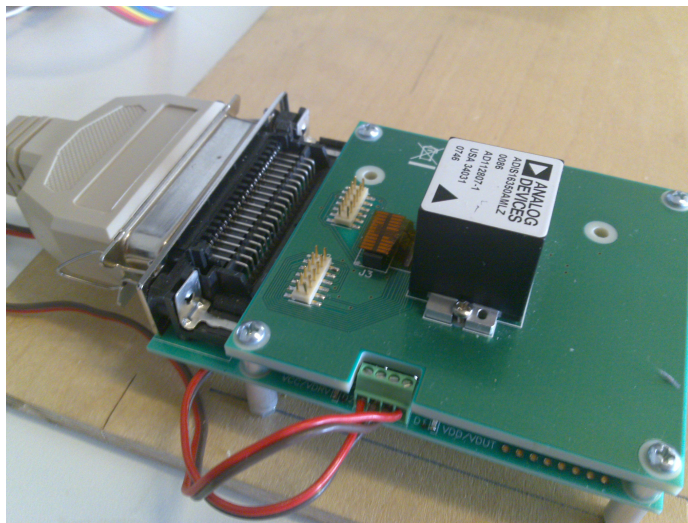


Figura 4.7: Unità inerziale ADIS-16350

L'altro sensore utilizzato è il magnetometro di figura B.11. Prima di essere usato è stato calibrato per annullare i disturbi di campo magnetico solidali alla piattaforma costruita e riportata in figura 4.10, procedura interamente descritta in Appendice B.

Si riporta ora la piattaforma costruita in laboratorio completa del sistema per la determinazione d'assetto in cui potenzialmente è alloggiabile il GPS per fare prove con la struttura di figura 3.1. Ogni volta che viene aggiunto un componente e fissato alla piattaforma è necessario ricordarsi di procedere

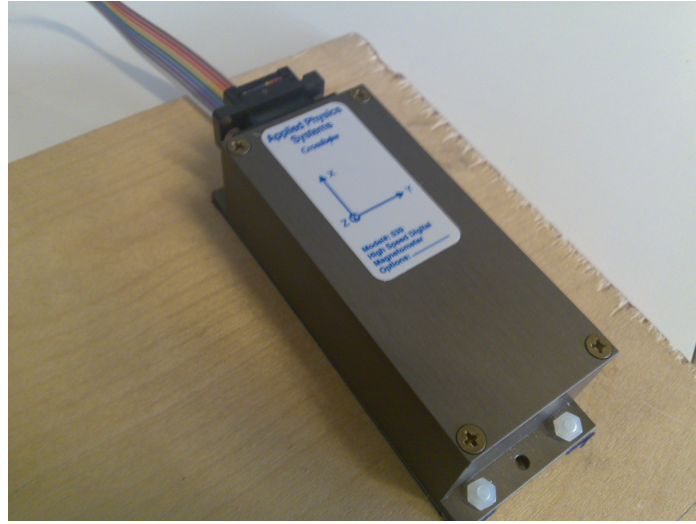


Figura 4.8: Magnetometro Applied Physics AP539

nuovamente alla calibrazione del magnetometro. Durante il montaggio della piattaforma è stato necessario fare riferimento al verso dei tre assi di unità inerziale e magnetometro al fine di garantire il perfetto funzionamento del software che deve quindi essere corretto di conseguenza³.

Per poter acquisire i dati dai sensori è stato implementato lo schema Simulink riportato in figura 4.9 in cui come si può notare è stato adottato sia il Quest che il filtro d'assetto AHRS. In base ai collegamenti è possibile fare diverse verifiche utilizzando ad esempio tutti i dati X-sens oppure gli output dei sensori grezzi al fine di comparare il risultato dei due sistemi che agiscono autonomamente.

³tramite dei segni '-' negli output dei blocchetti di acquisizione dei sensori

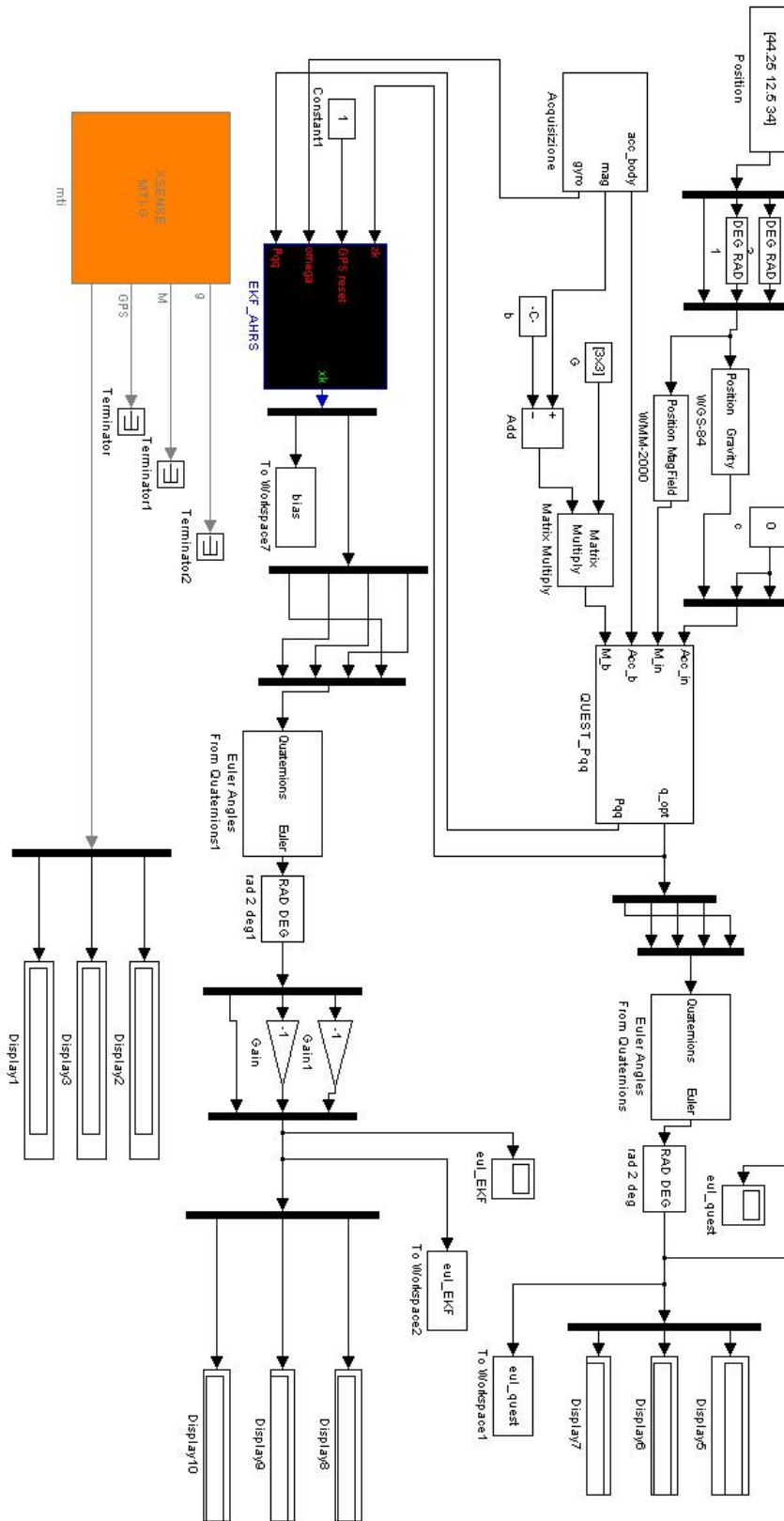


Figura 4.9: Modello Simulink per le prove sperimentali

Utilizzando la piattaforma di figura 4.11 è stato possibile verificare il filtro con l'utilizzo di sensori veri e comparare i risultati con gli output del programma di determinazione d'assetto della piattaforma X-sens. I risultati si possono apprezzare nelle figure 4.12 4.13 e 4.14.

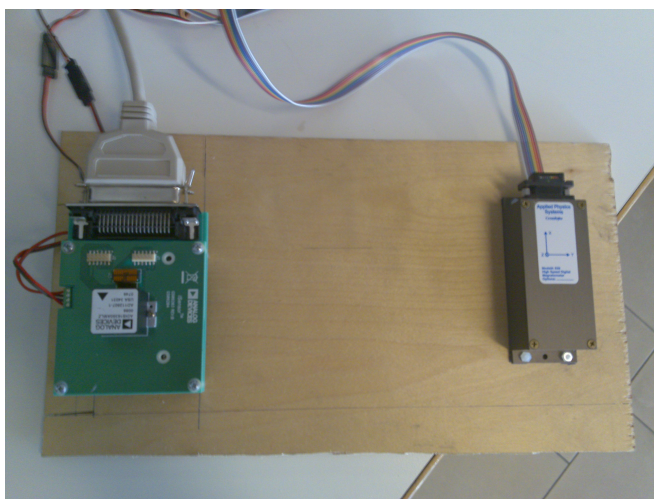


Figura 4.10: Piattaforma con unità inerziale e magnetometro

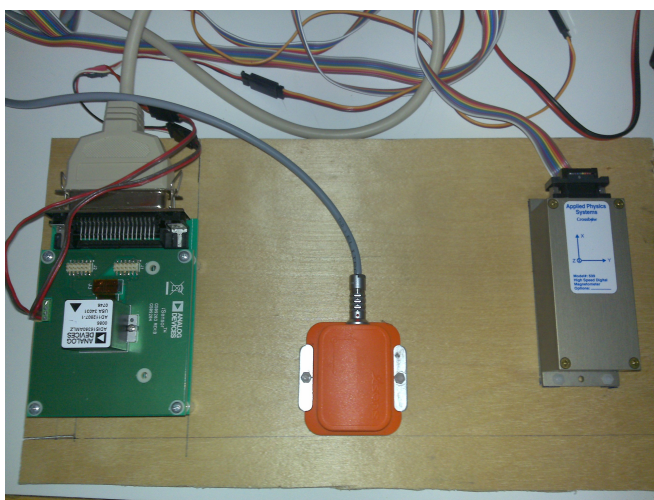


Figura 4.11: Piattaforma con unità inerziale X-sens

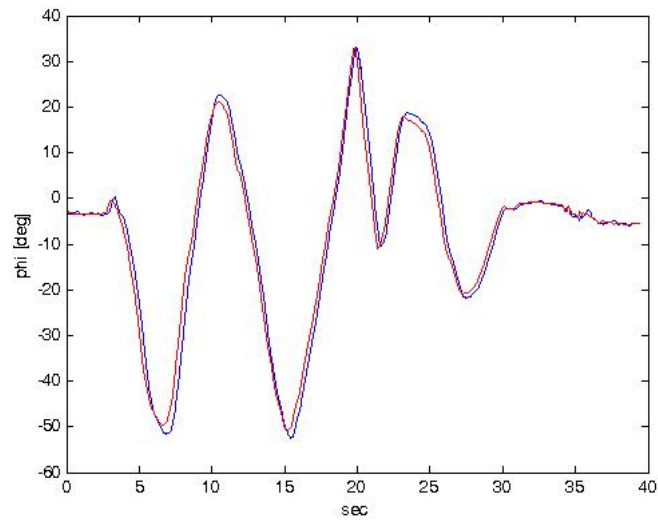


Figura 4.12: Angolo di rollio- Φ

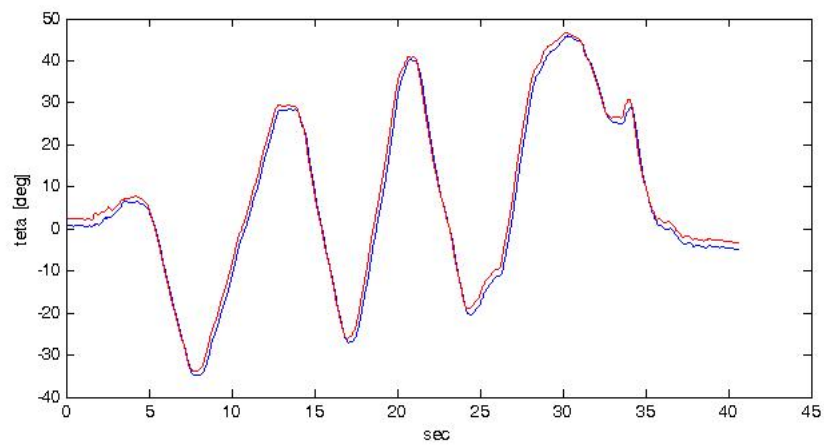
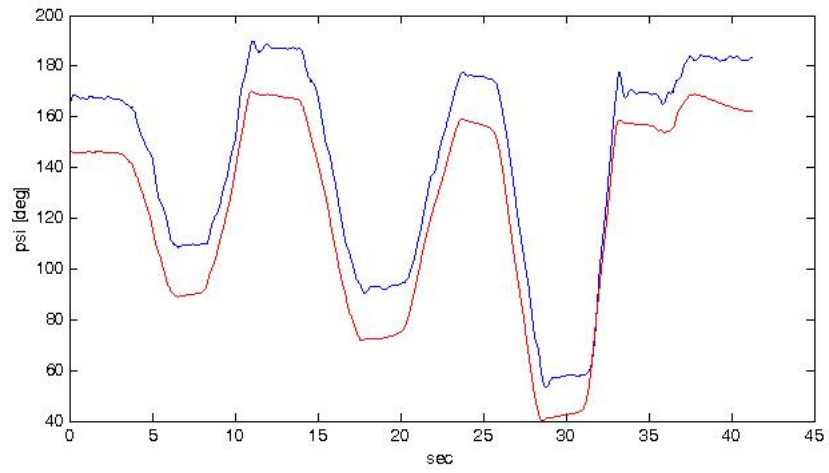
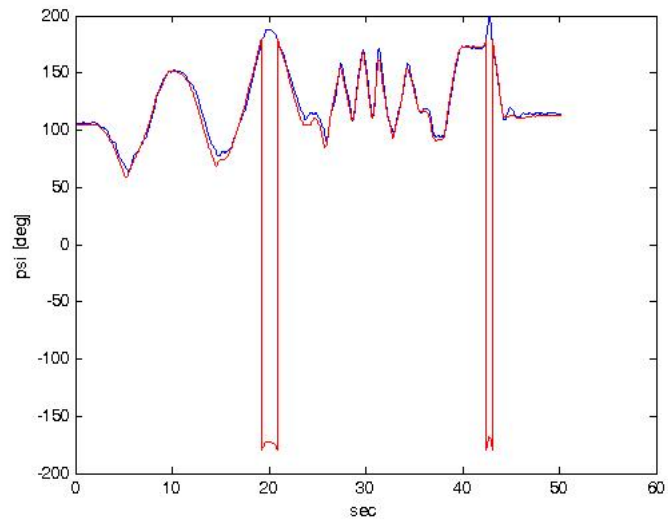


Figura 4.13: Angolo di beccheggio- Θ

Figura 4.14: Angolo di Heading magnetico- Ψ Figura 4.15: Angolo di Heading magnetico- Ψ utilizzando output X-sens

Per quanto riguarda gli angoli di rollio e beccheggio la corrispondenza è relativamente buona⁴ considerando che tali prove sono state eseguite in laboratorio in cui il rumore di campo magnetico era molto forte e non annullabile tramite la calibrazione poichè non solidale alla piattaforma di lavoro. Sorprende il risultato dell'*Heading* che ha costantemente un *bias* di circa 20 gradi. In realtà è stato verificato che il magnetometro della X-sens non è affidabile e che quindi quel bias non porta a degradare la prestazione dell'algoritmo implementato, infatti proprio per l'uso intensivo che l'algoritmo Quest fa del dato di campo magnetico sarebbe impossibile ottenere buoni risultati solo per due dei tre assi ortogonali.

E' comunque stata fatta una verifica supplementare in cui sono stati usati i dati di output della piattaforma X-sens per verificare una corrispondenza completa, in questo modo si valuta solamente l'algoritmo di determinazione d'assetto e non più i sensori a disposizione. Il risultato per l'angolo di *Heading* è riportato in figura 4.15 in cui si ritrova quella corrispondenza delle figure 4.12 e 4.13. Da notare come quando l'angolo stimato dal sistema Quest-EKF in rosso supera i 180 gradi vada immediatamente a -180 gradi, mentre l'angolo determinato dalla X-sens in blu proceda oltre i 180 gradi; questo è semplicemente determinato dalla diversità degli algoritmi adottati e dai *range* angolari che questi rispettano nel fornire l'informazione d'assetto.

Un'ultima osservazione va fatta riguardo il confronto tra il segnale prodotto dal sistema presentato in questo documento, nelle passate figure rappresentato in rosso, ed il segnale di output della piattaforma X-sens, rappresentato in blu. Si nota infatti come nonostante la vulnerabilità del Quest al rumore di campo magnetico che inevitabilmente è presente in un laboratorio il segnale rosso è più stabile e meno rumoroso del segnale blu, il che dimostra la validità del filtro e della struttura implementata nel suo complesso. Questo è visibile più chiaramente nell'ingrandimento di figura 4.16 di seguito riportata.

⁴gli scostamenti massimi fra il segnale ottenuto in rosso e quello della piattaforma X-sens in blu sono inferiori al grado

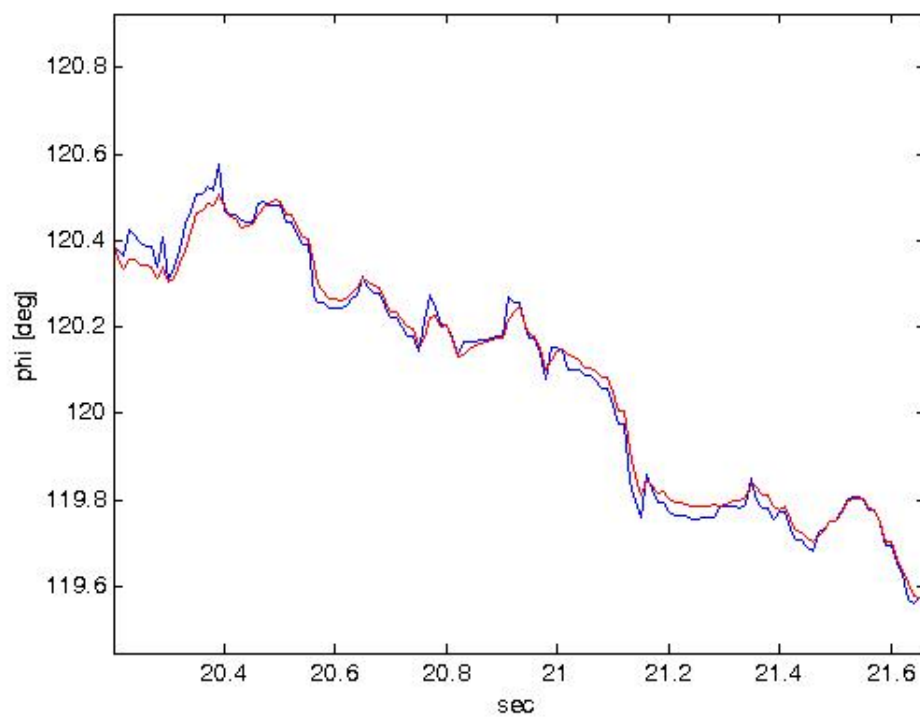


Figura 4.16: Ingrandimento che compara il segnale rosso, output del filtro EKF, ed il segnale blu, output della piattaforma X-sens per ciò che riguarda l'angolo di rollio

Capitolo 5

Conclusione e Sviluppi futuri

Nel presente documento sono riportati lo sviluppo e l'implementazione di algoritmi per la determinazione d'assetto. I blocchi Simulink implementati sono mostrati in figura 5.1 e sono composti dal *Quaternion estimator*, Quest, dal *Three-axis attitude determination*, Triad, da un filtro esteso di Kalman EKF-AHRS e da un algoritmo per la calibrazione di un magnetometro. Come riportato nei risultati sperimentali le prove fatte in Software-in-the-loop confermano la validità della teoria applicata e del lavoro di implementazione degli algoritmi, ma nel momento in cui sono stati utilizzati i sensori a disposizione del laboratorio¹ i risultati, seppur accettabili, sono peggiorati. Come già osservato il motivo risiede nel fatto che le prove fatte, per limiti di hardware, si sono svolte all'interno del laboratorio di meccanica del volo² in cui a causa delle apparecchiature elettroniche in uso era presente un disturbo magnetico che condizionava i risultati. Tuttavia tali disturbi interessavano anche le piattaforme commerciali, come la X-sens, usate come confronto ed il fatto che il segnale ricavato dall'algoritmo di determinazione studiato fosse estremamente vicino a quello di output di tali piattaforme se non migliore e meno rumoroso, figura 4.16, dimostra la validità del lavoro svolto e la discreta resistenza al rumore della struttura adottata. La precisione del sistema non è stata certificabile poichè dipendente dalla zona del laboratorio in cui

¹piattaforma ADIS-16350 e magnetmetro AP539

²della seconda facoltà di ingegneria di Bologna con sede a Forlì

veniva svolta la prova; le figure riportate rappresentano il risultato di dati acquisiti là dove il rumore di campo magnetico era meno rilevante.

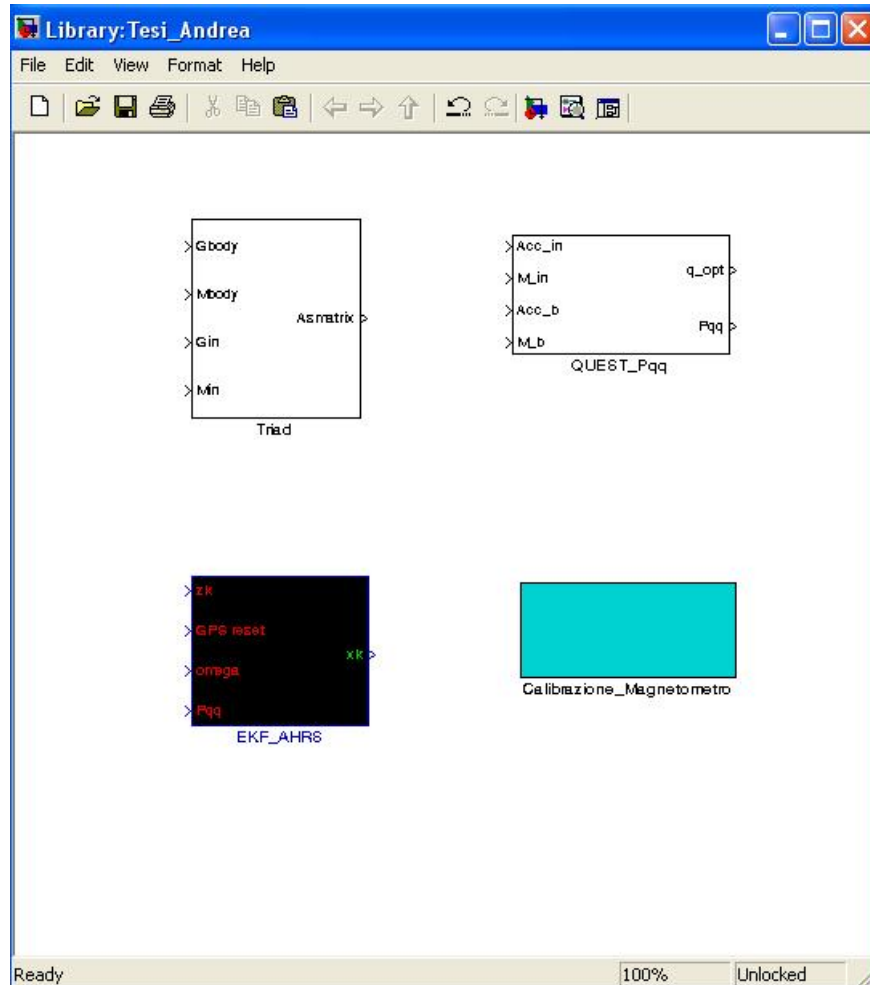


Figura 5.1: Libreria creata in Simulink

Per questo il passo successivo è quello di eseguire le prove fuori dal laboratorio dopo aver modificato la porta di comunicazione della piattaforma inerziale³ in modo da poter utilizzare un PC portatile e integrare la restante parte di stima della posizione che necessita del segnale GPS, acquisibile solamente all'esterno. A questo punto si dovrà procedere con l'ottimizzazione del filtro, che consiste nel determinare analiticamente un *process noise*

³da parallela a USB ad esempio

che minimizzi una certa funzione di costo costruita mediante il vettore di innovazioni del filtro, al fine di ottenere la migliore attenuazione del rumore evitando la divergenza dal segnale vero. Se anche dopo l'ottimizzazione non si ottengono i risultati sperati è possibile modificare la struttura del filtro EKF-AHRS considerando la non linearità delle misure ed utilizzando la matrice $h(x)$ riportata in 3.7, in questo modo si dovrebbe dipendere meno dai disturbi di campo magnetico.

Dovranno essere fatte ulteriori verifiche ma in definitiva si è riusciti ad ottenere un sistema di determinazione d'assetto che raggiunga validi risultati utilizzando sensori a basso costo grazie all'utilizzo dell'algoritmo Quest e di filtri di Kalman estesi.

Appendice A

Fondamenti di analisi del segnale

Durante la fase di acquisizione dei dati dall'unità inerziale ADIS-16350 studiando il *data-sheet* del prodotto si è visto che il campionamento poteva essere fatto a diverse frequenze, partendo da 100 Hz fino ad 800 Hz. A questo punto per ottimizzare il risultato¹ è stata affrontata la teoria dell'analisi e della gestione del segnale consultando manuali di Matlab disponibili in rete. Innanzitutto si può dire che un segnale acquisito ad una frequenza più elevata ha sicuramente una deviazione standard maggiore dello stesso acquisito ad una frequenza inferiore. Ciò porterebbe a preferire l'output a 100 Hz, anche perchè l'intera struttura della determinazione d'assetto lavora a 100 Hz, a causa dei limiti di campionamento del magnetometro, tuttavia è stata studiata l'ipotesi di acquisire a frequenze maggiori per poi compiere un sotto campionamento al fine di ridurre il rumore del segnale.

A.1 La scelta dei 100 Hz

Tramite il software dell'unità inerziale sono stati campionati dati a frequenze diverse con piattaforma ferma e sono state calcolate media e deviazioni standard attraverso un'analisi svolta in ambiente Matlab. Si è verificato ciò che prevedeva la teoria, infatti i segnali migliori erano quelli acquisiti a 100 Hz. Tuttavia il rumore, che dovrebbe aumentare con \sqrt{Hz} , aumenta molto

¹ridurre il rumore

meno del previsto con la frequenza di campionamento e questo è a causa di un filtro FIR contenuto nell'unità stessa. A questo punto è stata applicata l'idea del sotto campionamento realizzata in ambiente Matlab: innanzitutto sono stati acquisiti dati a 500 Hz, dopodichè, in *post-processing*, tramite un ciclo *for* è stato eseguito il sotto campionamento che consiste nel prendere n campioni, farne la media e fornire in output solo tale valore. In definitiva si applica un FIR che allo stesso tempo riduca il numero di campioni di output rispetto a quelli di input da 500 al secondo a 100 al secondo. Il risultato di tale operazione è stato quello di avere segnali più continui grazie alla maggiore risoluzione ottenuta, a scapito di un leggero aumento del rumore. In realtà quello che sembrava rumore era il bias² con una frequenza di cambiamento dell'ordine di 1 Hz, valore troppo basso per poterlo togliere con un filtro 'passa basso'. Di conseguenza, sebbene più reattivo, il segnale acquisito a 500 Hz non era adatto all'ottimizzazione della struttura ed è stata scelta la frequenza di campionamento di 100Hz che oltretutto è computazionalmente meno costosa.

A.2 Analisi del segnale acquisito a 100 Hz

Scartata l'alternativa di un campionamento più frequente si è analizzato il segnale dei giroscopi a 100 Hz e la figura A.1 ne mostra un'acquisizione di 10 secondi.

Si può notare come oltre all'inevitabile rumore quello che risalta è il bias variabile di questi strumenti che è dell'ordine del decimo di grado al secondo ma che se non corretto può fornire problemi alla determinazione d'assetto e al filtro AHRS che utilizza le velocità angolari per stimare il quaternion e d'assetto. Mentre la frequenza bassa di variazione del bias impedisce di poterlo eliminare tramite un filtro 'passa basso', se si isola il rumore è possibile individuare una frequenza di taglio ed ideare un filtro che elimini ogni parte del segnale che abbia una frequenza al di sopra di quella. Da cui la neces-

²dei giroscopi poichè è su quelli che sono state svolte le prove sperimentali

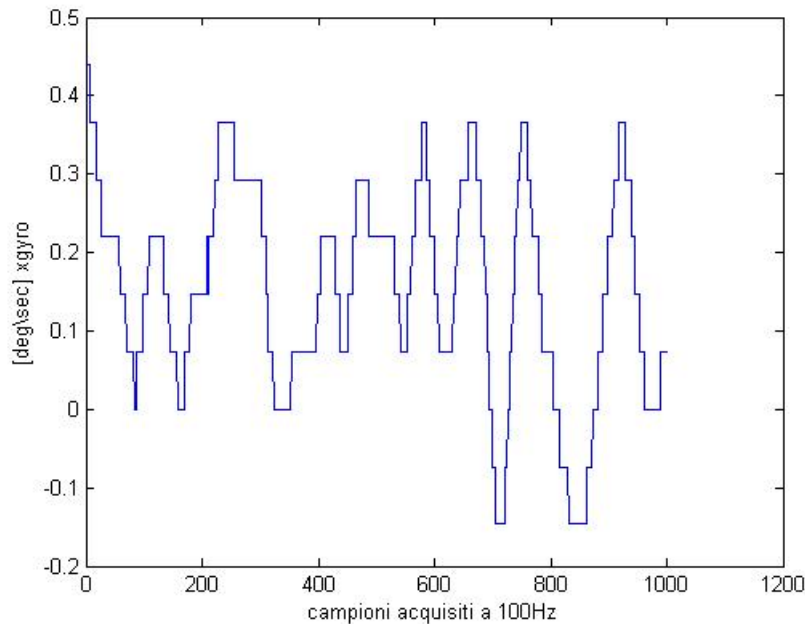


Figura A.1: Prova statica - segnale di output del giroscopio sull'asse delle X

sità di un'analisi spettrale³ del segnale di figura A.1 che viene riportata di seguito.

Dall'analisi della figura A.2 si deduce che non è presente una frequenza di taglio nettamente individuabile. Sono state fatte molte prove applicando filtri con ordini e frequenze di taglio diverse, ma quando si è passati alle prove dinamiche quello che si è visto è che si provocava un ritardo ed una attenuazione del segnale vero, evidentemente perchè si andava a tagliare una frequenza propria del segnale non facente parte del rumore. In conclusione il filtro FIR già contenuto all'interno della piattaforma è ottimizzato per togliere tutto il rumore possibile ed ha lo scopo di rendere più 'puliti' anche i segnali acquisiti a frequenze di campionamento più elevate. Per quanto riguarda lo scopo del sistema descritto dal presente documento l'acquisizione a 100Hz risulta la più conveniente e quindi quella è stata adottata. Tale analisi ha comunque verificato che i sensori a disposizione, soprattutto i giroscopi, hanno un bias non trascurabile con una frequente variazione nel tempo che

³è stata utilizzata l'analisi spettrale proposta da Welch già implementata in Matlab

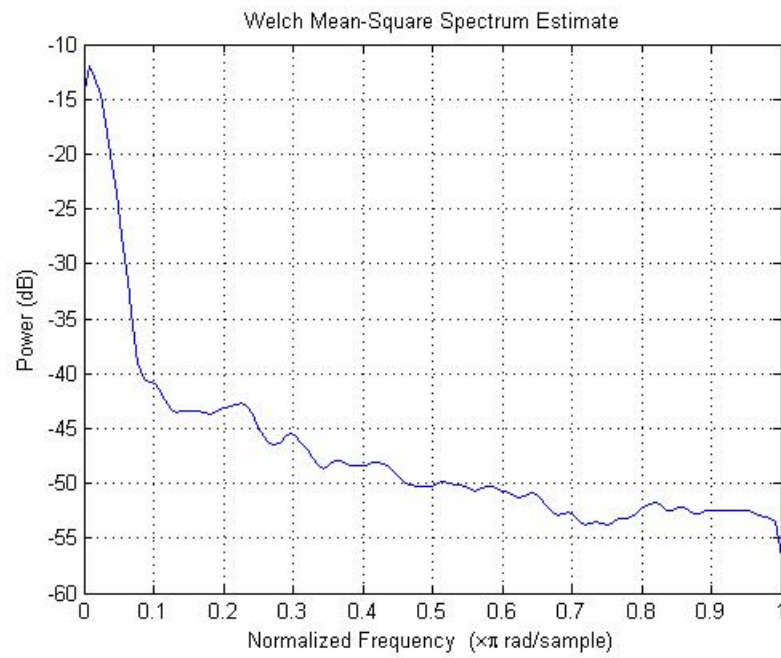


Figura A.2: Analisi spettrale del segnale acquisito a 100 Hz

deve essere individuato e corretto al fine di ottimizzare il risultato della navigazione inerziale, ciò giustifica la complessa architettura adottata in cui è presente la retroazione dei bias stimati dai filtri di Kalman.

Appendice B

Calibrazione del magnetometro

Durante le prove sperimentali ci si è resi conto che il segnale di output del magnetometro è affetto da molti disturbi di campo magnetico, alcuni esterni dovuti al fatto che si facevano prove all'interno di un laboratorio, altri invece solidali alla piattaforma su cui il magnetometro era montato dovuti alle altre apparecchiature necessarie alla determinazione d'assetto, come l'unità inerziale. Si cerca ora di implementare un algoritmo, tratto da [7], di calibrazione del magnetometro che elimini proprio questi ultimi, detti anche *Hard-Iron errors*. Il principio alla base dell'algoritmo sfrutta il fatto che in una data posizione sulla terra¹ il vettore di campo magnetico ha una precisa norma, che non varia ruotando il magnetometro, se quindi si esprime il campo magnetico misurato h^m come

$$h^m = Ch^b + b + n \quad (\text{B.1})$$

dove h^b è il campo magnetico vero, è possibile trovare valori di C di b e di n per cui vale la relazione B.1. In realtà l'algoritmo non avrà un risultato perfetto a causa di n che è il rumore del sensore dovuto ad esempio ai disturbi di campo magnetico esterni, ma stimando il bias b e la matrice di correzione C è possibile non solo avere approssimativamente il valore di norma che ci si aspetta dai dati tabulati, ma anche un valore quasi costante durante le rotazioni eseguite.

¹individuabile dalle coordinate [Lat Long Alt]

B.1 Algoritmo autocalibrante con norma fissa- ta

Fra i vari tipi di algoritmi di calibrazione è stato scelto il *Locus-Constraint Autocalibration* il quale non necessita di altri sensori ed inoltre riesce mantenere la norma del vettore di campo magnetico costante e pari al valore che viene specificato. Sebbene si potesse implementare un algoritmo che ad ogni accensione del magnetometro vada automaticamente a calibrarlo è risultato più conveniente adottarne uno più semplice che sfrutti i dati acquisiti in *post-processing* e che ogni volta fornisca i parametri per la calibrazione. Si è supposto in effetti che una volta montata la piattaforma completa il magnetometro non verrà rimosso frequentemente e quindi la calibrazione è un'operazione che dovrà essere fatta solo raramente.

Partendo dalla formula B.1 è possibile esplicitare h^b ottenendo

$$h^b = C^{-1}h^m - C^{-1}b - C^{-1}n = G(h^m - b) - Gn \quad (\text{B.2})$$

lo scopo di tale algoritmo è proprio quello di determinare G e b mentre si trascurerà n che non è valutabile con tale procedura e che la rende inutile se non risultasse essere trascurabile. Fra le ipotesi si ha il valore di norma del vettore campo magnetico che può essere ricavato dal blocco Simulink *WMM-2000*² e si può maneggiare in ambiente Matlab tramite la struttura esposta in figura B.1. In Matlab è così salvato un vettore w che corrisponde al vettore di campo magnetico in assi NED nella località indicata dal GPS. Indicando w con h per attenersi al testo dell'articolo è possibile ottenere la norma che è costante ed uguale a $\sqrt{h^T h}$. Si può avere quindi

$$h^T h = [G(h^m - b) - Gn]^T [G(h^m - b) - Gn] = (h^m)^T \Gamma h^m - 2b^T \Gamma h^m + b^T \Gamma b + \hat{n} \quad (\text{B.3})$$

dove \hat{n} è il rumore modificato dai passaggi matematici e Γ è definita come

²che come input necessita le coordinate fornite dal GPS

$$\Gamma = G^T G = \begin{bmatrix} \gamma_1 & \gamma_2 & \gamma_3 \\ \gamma_2 & \gamma_4 & \gamma_5 \\ \gamma_3 & \gamma_5 & \gamma_6 \end{bmatrix} \quad (\text{B.4})$$

L'equazione B.3 è l'equazione di un ellissoide ed indica che le misurazioni di campo magnetico sono vincolate a giacere sulla superficie di quella figura geometrica. L'algoritmo ha lo scopo di stimare tramite un processo iterativo i parametri G e b , dati valori di h^m , per far sì che l'ellissoide sia una sfera.

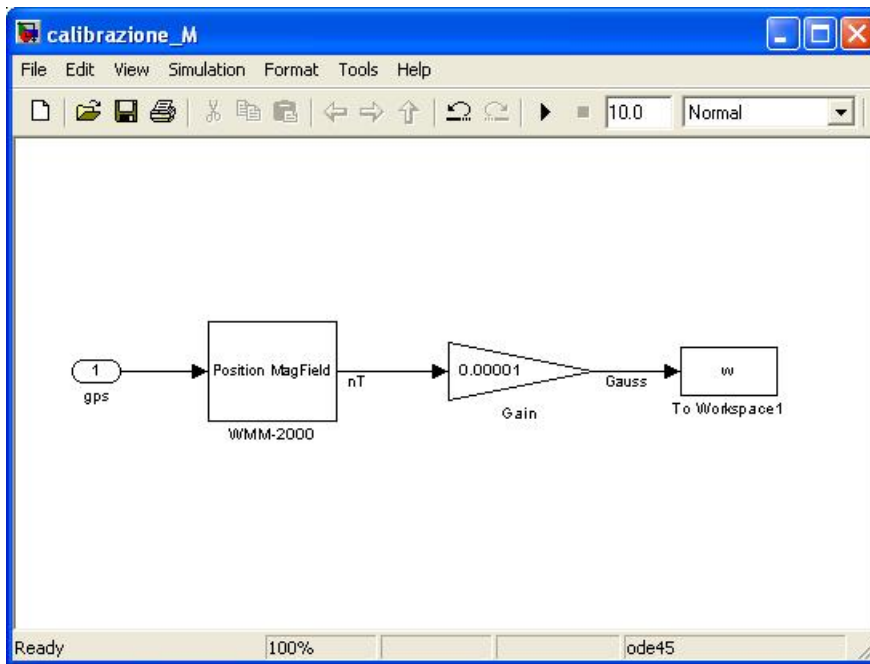


Figura B.1: Acquisizione della norma in Simulink

Date N misurazioni del magnetometro le relazioni fra G , b e le misure possono essere espresse mediante un sistema di N equazioni siffatto

$$r = H\xi \quad (\text{B.5})$$

dove $r \in R^{Nx1}$ è un vettore residuo di campo, nullo se le misure acquisite hanno errore e rumore nulli, e $\xi \in R^{10x1}$ è dato da

$$\xi = [\gamma_1 \quad \gamma_2 \quad \dots \quad \gamma_9 \quad \gamma_{10}]^T \quad (\text{B.6})$$

la riga i -esima del sistema riportato in B.5 è composta nel seguente modo

$$\begin{aligned}
 r_i &= \gamma_1(h_x^m)_i^2 + 2\gamma_2(h_x^m h_y^m)_i + 2\gamma_3(h_x^m h_z^m)_i \\
 &+ \gamma_4(h_y^m)_i^2 + 2\gamma_5(h_y^m h_z^m)_i + \gamma_6(h_z^m)_i^2 \\
 &+ \gamma_7(h_x^m)_i + \gamma_8(h_y^m)_i + \gamma_9(h_z^m)_i \\
 &+ \gamma_{10} \quad (\text{B.7})
 \end{aligned}$$

dove $\gamma_1 \dots \gamma_6$ sono gli elementi della matrice Γ e

$$\gamma_7 = -2(\gamma_1 b_x + \gamma_2 b_y + \gamma_3 b_z) \quad (\text{B.8})$$

$$\gamma_8 = -2(\gamma_2 b_x + \gamma_4 b_y + \gamma_5 b_z) \quad (\text{B.9})$$

$$\gamma_9 = -2(\gamma_3 b_x + \gamma_5 b_y + \gamma_6 b_z) \quad (\text{B.10})$$

$$\gamma_{10} = \gamma_1 b_x^2 + 2\gamma_2 b_x b_y + 2\gamma_3 b_x b_z + \gamma_4 b_y^2 + 2\gamma_5 b_y b_z + 2\gamma_6 b_z^2 - h^T h \quad (\text{B.11})$$

l'algoritmo di calibrazione risolve il sistema B.5 ponendo $r = 0$ e trovando ξ , stima così i valori γ_i e costruisce G e b .³

B.2 Implementazione dell'algoritmo

Nel primo step si risolve B.5 che diventa $0 = H\xi$. La soluzione di questo problema si può trovare facilmente se si effettua una scomposizione in valori singolari⁴ di $H^T H$ nel seguente modo

$$H^T H = V_H \Sigma_H^2 V_H^T \quad (\text{B.12})$$

³nell'articolo [7] il termine di r_i in cui compare γ_4 era moltiplicato per un fattore 2, dopo prove sperimentali si è verificato che è stato un errore di stampa ed è stata riportata la formula corretta

⁴SVD

Come riportato nell'Appendice A di [9] ogni matrice complessa A può essere fattorizzata in una scomposizione in valori singolari come in B.12 dove Σ_H^2 contiene una matrice diagonale e le colonne di V_H sono gli autovettori singolari. A questo punto è possibile risalire al valore normalizzato di ξ infatti vale

$$v_{10} = \frac{\xi}{|\xi|} \quad (\text{B.13})$$

dove v_{10} è la decima colonna di V_H . Ora è necessario determinare il modulo di ξ ed i valori di b . Dalle equazioni B.8, B.9 e B.10 è possibile ricavare b una volta noto ξ , ma a questo punto è noto solamente $\bar{\xi}$, il vettore normalizzato. Fortunatamente la norma di ξ non cambia il risultato⁵ quindi attraverso semplici passaggi matematici è possibile ricavare il valore di b come

$$b = -\frac{1}{2} \begin{bmatrix} \gamma_1 & \gamma_2 & \gamma_3 \\ \gamma_2 & \gamma_4 & \gamma_5 \\ \gamma_3 & \gamma_5 & \gamma_6 \end{bmatrix}^{-1} \begin{bmatrix} \gamma_7 \\ \gamma_8 \\ \gamma_9 \end{bmatrix} \quad (\text{B.14})$$

è ora possibile stimare il modulo di ξ da cui ricavare i veri valori di $\gamma_1 \dots \gamma_{10}$. Sempre utilizzando la notazione per cui $\bar{\gamma}_i$ è la componente normalizzata si ha

$$|\xi| = h^T h (\bar{\gamma}_1 b_x^2 + 2\bar{\gamma}_2 b_x b_y + 2\bar{\gamma}_3 b_x b_z + \bar{\gamma}_4 b_y^2 + 2\bar{\gamma}_5 b_y b_z + 2\bar{\gamma}_6 b_z^2 - \bar{\gamma}_{10})^{-1} \quad (\text{B.15})$$

si può quindi ottenere $\xi = |\xi| \cdot \bar{\xi}$. Sfruttando la scomposizione in valori singolari e i risultati ottenuti è possibile riscrivere la B.2 utilizzando quantità note, da cui si arriva alla formula che porta alla calibrazione

$$h = R_G(V_\Gamma \sqrt{\Sigma_\Gamma} V_\Gamma^T) \left(h^m + \frac{1}{2} \begin{bmatrix} \gamma_1 & \gamma_2 & \gamma_3 \\ \gamma_2 & \gamma_4 & \gamma_5 \\ \gamma_3 & \gamma_5 & \gamma_6 \end{bmatrix}^{-1} \begin{bmatrix} \gamma_7 \\ \gamma_8 \\ \gamma_9 \end{bmatrix} \right) \quad (\text{B.16})$$

⁵si possono utilizzare i $\bar{\gamma}_i$ noti al posto dei γ_i incogniti

dove R_G è uguale alla matrice identità, una volta considerato trascurabile il rumore di *Soft-Iron* che non si può correggere con questo tipo di calibrazione, e dove la matrice G è stata ricavata da una scomposizione in valori singolari della matrice Γ .

Implementare questo algoritmo in linguaggio Matlab-Simulink ha comportato l'utilizzo di un ciclo *for* che modifichi la matrice H ogni volta che si acquisisce un nuovo dato e che ripeta ad ogni step le fasi descritte. Come detto è possibile utilizzare la struttura ideata solo in *post-processing*⁶, ma dato che la procedura di calibrazione non deve essere ripetuta frequentemente non è stato ritenuto opportuno adottare una struttura più complessa che potesse usarsi in tempo reale.

B.3 Risultati sperimentali della calibrazione

Vengono ora riportate immagini che descrivono graficamente ciò che l'algoritmo di calibrazione fa una volta acquisito un sufficiente numero di dati durante rotazioni complete della piattaforma su cui è montato il magnetometro. Innanzitutto si mostra la convergenza del bias⁷ dopo un certo numero di campioni, B.2, B.3 e B.4.

Infine si mostra la norma del vettore campo magnetico prima della calibrazione, dopo la calibrazione e nel caso ideale considerando la posizione geografica indicata dal GPS.

⁶dopo aver quindi acquisito e salvato misurazioni del magnetometro

⁷per le 3 componenti

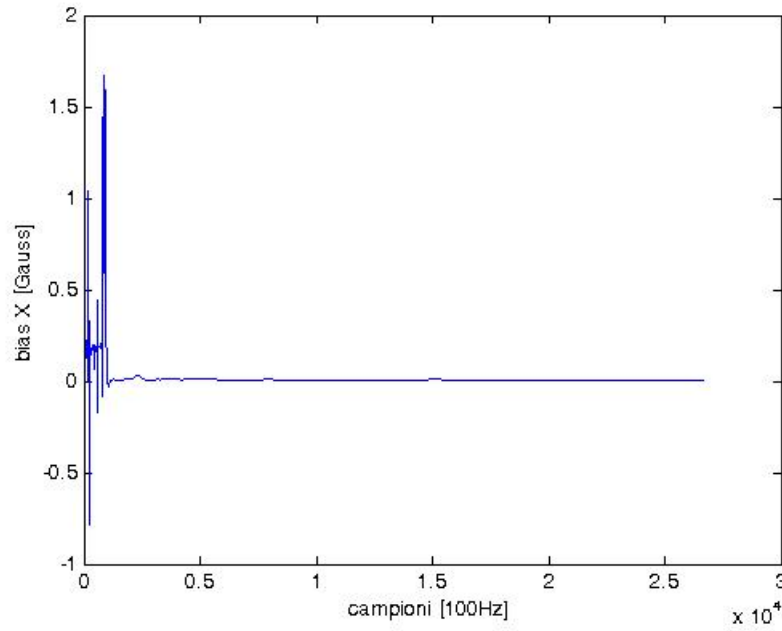


Figura B.2: Convergenza del bias misurato sull'asse delle X

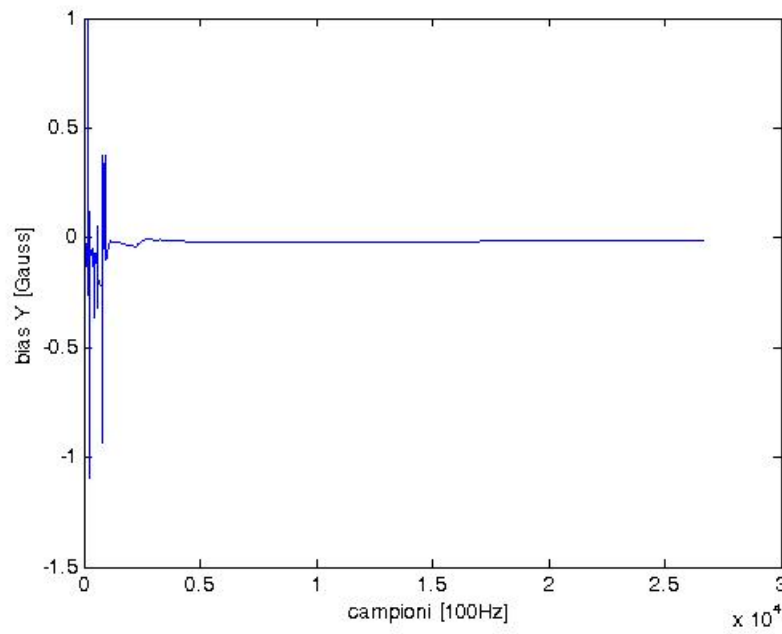


Figura B.3: Convergenza del bias misurato sull'asse delle Y

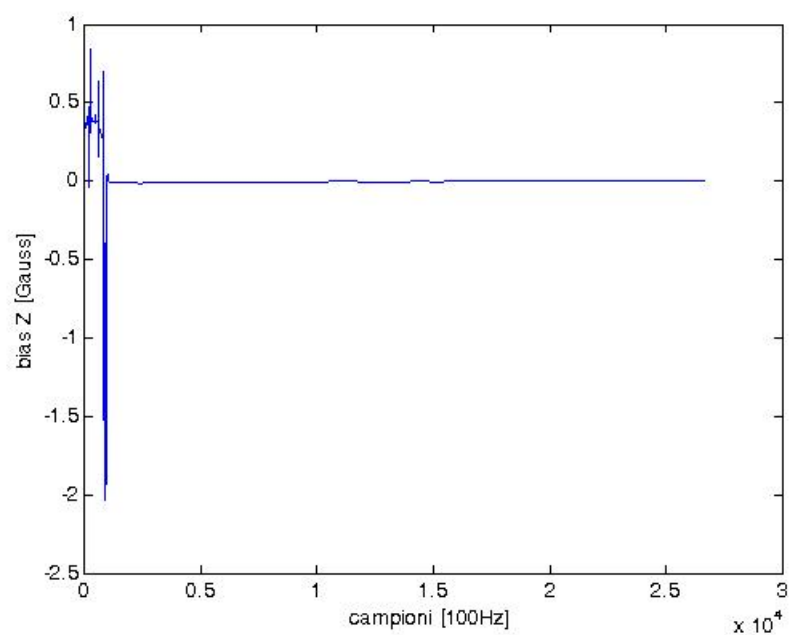


Figura B.4: Convergenza del bias misurato sull'asse delle Z

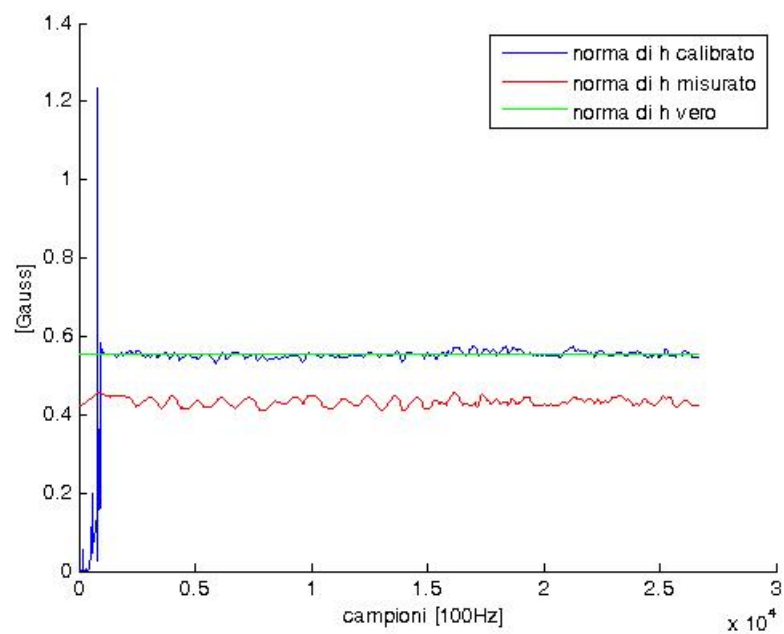


Figura B.5: La norma del vettore campo magnetico prima e dopo la calibrazione

Si può notare in figura B.5 come non solo l'output calibrato sia più stabile, ma come sia anche più vicino al valore ideale, individuato dal blocchetto Simulink, pari alla norma che dovrebbe misurarsi effettivamente in base ai dati tabulati. Ciò dimostra la validità dell'algoritmo implementato e la necessità di utilizzarlo, in particolar modo in un sistema come quello proposto da questo documento in cui il vettore campo magnetico non è importante solo per determinare l'*heading*, ma è utilizzato interamente nell'algoritmo Quest per la determinazione d'assetto.

Ringraziamenti

Ora viene la parte più difficile, in cui cercherò di non dimenticare nessuno, il fatto è che sono veramente tante le persone che mi hanno accompagnato in questo lungo viaggio dai risultati inaspettati. Innanzitutto saluto i compagni di facoltà che hanno condiviso con me interminabili ore di lezione, a partire da Flavio, che con gli anni ha imparato la puntualità, grazie di essermi amico. Grazie Carlo per la simpatia che hai dimostrato e complimenti per i gusti musicali, che sono uguali ai miei! Grazie Nicola per aver vissuto con me la facoltà in un senso più profondo, senza le tue paranoie ammetto che qualche 30 mi sarebbe scappato. Grazie Francesco per i tuoi raptus di follia che hanno ravvivato momenti noiosi. Grazie Cinzia per la presenza femminile in mezzo a decine di ragazzi e per la tua simpatia salentina. Grazie Marco per avermi ricordato le mie origini. Grazie Ami per essere stata mia confidente e compagna di tante serate. Grazie Eugenio per le lunghe chiacchierate, forse a volte troppo lunghe! Ringrazio tutti gli altri miei compagni di corso, soprattutto quelli che ho scoperto come amici recentemente tra cui Bruno, Federiccio e Daniele e quelli che avrei voluto conoscere di più come Filo e Golo. Ovviamente un ringraziamento speciale va ad Antonio e Alessandro, compagni a Corfù, ma soprattutto amici con cui ho condiviso l'emozionante vita notturna di Forlì. Arrivano ora i coinquilini, davvero tanti vivendo in uno studentato. Il primo che ringrazio è proprio lui, il Sant'Anna, che mi ha dato casa per 3 anni e spero anche per i prossimi 2! Ringrazio i Rocchi per la simpatia e le serate, poche ma memorabili, in particolare quello alto con cui ho ragionato spesso di 'fantasie ingegneristiche', grazie davvero della tua amicizia Rocco. Fra quelli che sono ormai partiti ricordo Andgy, Lucy e Tatjana, la prima per avermi regalato tanti sorrisi, la seconda per le urla

e le arrabbiate che la rendono tanto divertente quanto speciale e l'ultima per le numerose colazioni insieme. Ringrazio per le serate Ilaria, Noemi, Antonella, Francesca, Sebastiano e tutti gli altri dell'ala vecchia fino al gruppo ormai consolidato dell'ala nuova, parlo ovviamente di Margherita, Claudia, Emanuela, Maria Stella e Sebastiano. Un ringraziamento particolare va ad Enrico, per tutti i film che ci siamo visti. Grazie Ema per le emozioni che mi hai dato. Grazie anche a Marco, Nicol, Carol, Giusy, Linda, Sara, Gaspare, Davide e tutti gli altri che in 3 anni hanno vissuto con me. Saluto e ringrazio anche tutti coloro che mi hanno accompagnato nelle numerose serate di questi anni come il gruppo dello studentato in centro tra cui Ilaria e Diana, e le carpigiane di Forlì, Martina e Maria Laura e tutti quelli che ho conosciuto anche solo una sera, ma che mi hanno fatto divertire. Ringrazio i vecchi amici di Sassuolo, tutta la compagnia dei Masnadieri ed in particolare Luca, Bud e Manuel, per le vacanze insieme e le rimpatriate estive, mi mancate davvero. Ringrazio gli amici ritrovati dal lontano periodo delle medie per le nuove uscite: Tara, Gollini, la Sassa, la Scalabrini e quelli che mi hanno seguito anche alle superiori come la Silvia e la Giuly, grazie anche a quella pazza della Albicini e all'innamoratissima Sappy. Ringrazio Erik di Limidi, ci siamo conosciuti all'asilo ed ancora ci si vede, incredibile. Ringrazio la mia famiglia, davvero numerosa, a partire dagli zii, senza i quali mai sarei potuto diventare ciò che sto diventando, le mie zie, sempre disposte ad offrirmi un pasto, i miei cugini Daniele e Simone e le miei cuginette che ormai sono diventate grandi, ancora mi ricordo quando si giocava a Resident Evil insieme... Ringrazio ora chi ha permesso la mia maturazione accademica, a partire da tutti i professori, anche quello che mi ha dato 28, in particolar modo il mio relatore, l'ing. Fabrizio Giulietti e con lui anche l'ing. Matteo Turci che mi hanno aiutato ad affrontare le problematiche che si presentano ogni volta che si passa dalla teoria alla pratica. Infine ringrazio la nonna Donata ed il nonno Domenico per avermi cresciuto, grazie nonna Giancarla per il gnocco fritto e per avermi voluto bene, grazie Vittorio per il tempo passato. Grazie Paola per tutto quello che fai per me e grazie alla mia sorellina Giuly per esistere. Grazie Mamma per avermi amato, grazie papà per avermi insegnato a vivere, vi voglio bene.

Bibliografia

- [1] Paul Zarchan e Howard Musoff (2005), *Fundamentals of Kalman Filtering: A Practical Approach*, Progress in Astronautics and Aeronautics
- [2] Robert M. Rogers (2007), *Applied Mathematics in Integrated Navigation System Third edition*, AIAA EDUCATION SERIES
- [3] Francesco Torsello (2007), *Strapdown INS and GPS Integration using Kalman Filtering for Air Vehicular Navigation Applications*, Tesi di laurea specialistica, Alma Mater Studiorum-Università di Bologna
- [4] Jung Soon Jang e Darren Liccardo (2006), *Automation of small UAVS using a low cost mems sensor and embedded computing platform*, 25° Digital Avionics System Conference
- [5] Itzhack Y. Bar-Itzhack e Richard R. Harman (1996), *Optimized TRIAD Algorithm for Attitude Determination*, Journal of Guidance, VOL.20, NO.1
- [6] M.D.Shuster e S.D.Oh (1980), *Three-Axis Attitude Determination from Vector Observation*, Journal of guidance and control VOL.4, NO.1
- [7] Gebre-Egziabher (2007), *Magnetometer Autocalibration Leveraging Measurament Locus Constraints*, Journal of Aircraft VOL.44, NO.4
- [8] M. Barnabei e F.Bonetti (1992), *Sistemi lineari e matrici*, Pitagora
- [9] Sigurd Skogestad e Ian Postlethwaite (1996), *Multivariable Feedback and Control*, Wiley