



Università di Pisa

Ingegneria Robotica e dell'Automazione

# Progetto di Sistemi di Guida e Navigazione

Stima del bias di un accelerometro

Chiara Sammarco

Rachele Nebbia Colomba

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Descrizione Filtro di Navigazione</b>	<b>3</b>
<b>3</b>	<b>Filtro di Kalman per la stima del Bias</b>	<b>6</b>
3.1	Prove preliminari . . . . .	8
3.1.1	Filtro ad 1 grado di libertà . . . . .	8
3.1.2	Filtro a 3 gradi di libertà . . . . .	15
<b>4</b>	<b>Ruolo dello stimatore di assetto AHRS</b>	<b>22</b>
4.1	Filtro con feedback del bias stimato . . . . .	22
4.1.1	Creazione del primo dataset fittizio . . . . .	22
4.1.2	Risultati della simulazione . . . . .	27
4.1.3	Creazione del secondo dataset fittizio . . . . .	34
4.1.4	Risultati della simulazione . . . . .	36
<b>5</b>	<b>Simulazione su dati reali</b>	<b>41</b>
<b>6</b>	<b>Conclusioni</b>	<b>47</b>
	Appendice . . . . .	47
A	Appendice . . . . .	48
A.1	Background: Filtro di Kalman . . . . .	48

# 1. Introduzione

I sistemi di navigazione inerziale hanno il compito di stimare la posizione e velocità dei sistemi a partire da misure di sensori quali accelerometri, barometri e giroscopi. In generale integrando due volte l'accelerazione si è in grado di risalire allo spostamento, mentre dall'integrazione della velocità angolare si può calcolare l'orientazione. I problemi vengono fuori proprio quando sono implementate le integrazioni per trovare la posizione e l'orientamento. La presenza, anche minima, di offset sulla misura comporta un errore che cresce illimitatamente nel tempo. Per questo motivo si utilizza in aggiunta a queste misure il GPS, strumento utile per correggere in fase di **update** la stima di posizione. In questo caso lo svantaggio è che il filtro di navigazione risulta sbilanciato verso le correzioni di GPS; se dal filtro si interrompono o si eliminano le misure di quest'ultimo sensore la stima diverge con una velocità lineare nel tempo.

L'obiettivo di questo progetto è di inserire nel vettore di stima del filtro di Kalman anche il bias degli accelerometri; in questo modo la stima del bias può essere utilizzata per ottenere una navigazione basata più sulla predizione rispetto alla correzione e ovviare al problema di divergenza della stima in caso di guasto del sistema GPS.

Il sistema che andremo ad analizzare è costituito da:

- flag computation
- *Filtro di Kalman*
- blocco AHRS

Inoltre sono forniti:

- dati inerziali (body frame)
- dati misurati (navigation frame)

Il blocco **flag computation** tiene conto del funzionamento o meno dei sensori tramite i *flags* raccolti in un unico scalare secondo la mappa:

$$flags = init * 100 + gps * 10 + baro. \quad (1.1)$$

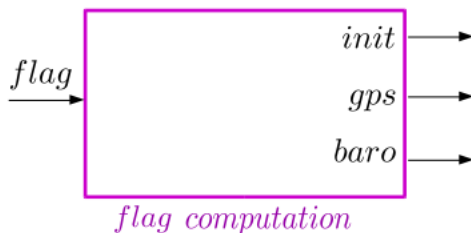


Figura 1.1: flag computation block

Il blocco **Kalman Filter** contiene l'implementazione del filtro, che verrà descritta nella sez.2.



Figura 1.2: Kalman Filter block

Il blocco **AHRS** contiene il filtro di assetto del sistema, che permette di ricavare l'orientazione a partire da misure di accelerometro,giroscopio e magnetometro.

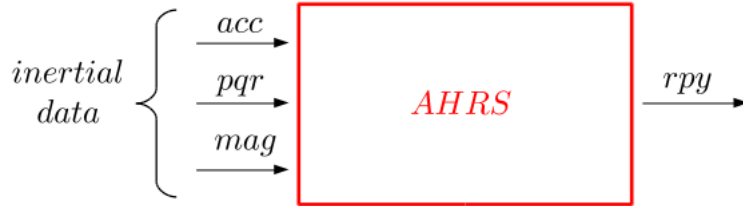


Figura 1.3: AHRS block

Nelle sezioni seguenti vengono riportati le descrizioni dei filtri utilizzati, e dei risultati ottenuti sia su dati simulati che su dati veri registrati durante una prova di volo di un drone.

## 2. Descrizione Filtro di Navigazione

In questo capitolo viene descritto un algoritmo di navigazione inerziale che calcola posizione, assetto e velocità del veicolo (stati della navigazione) ad ogni passo mediante l'integrazione di dati di piattaforme inerziali e GPS. L'algoritmo implementa un filtro di Kalman tempo discreto, successivamente adattato a un sistema aumentato per includere la stima del bias degli accelerometri.

Consideriamo la seguente equazione in tempo discreto:

$$p(k+1) = p(k) + v(k)T + \frac{1}{2}a(k)T^2 \quad (2.1)$$

dove  $p(k)$ ,  $v(k)$ ,  $a(k)$  sono i vettori  $\in \mathbb{R}^{3 \times 1}$  che descrivono rispettivamente posizione, velocità e accelerazione del sistema, mentre  $T$  è il tempo di campionamento (in questo caso  $T = 1/50$  s).

Nel filtro di Kalman scelto i dati inerziali sono utilizzati come misure e non ingressi esogeni: conseguentemente anche l'accelerazione rientra tra le variabili di stato.

Il vettore di stato in esame è quindi:

$$\mathbf{x}(k) = \begin{bmatrix} p_x(k) \\ v_x(k) \\ a_x(k) \\ p_y(k) \\ v_y(k) \\ a_y(k) \\ p_z(k) \\ v_z(k) \\ a_z(k) \end{bmatrix} \quad (2.2)$$

Per semplicità prendendo in esame la sola componente  $x$  dei vettori, il sistema in tempo discreto può essere riscritto come segue:

$$\begin{cases} p_x(k+1) = p_x(k) + v_x(k)T + \frac{1}{2}a_x(k)T^2 \\ v_x(k+1) = v_x(k) + a_x(k)T \\ a_x(k+1) = a_x(k) \end{cases} \quad (2.3)$$

Per l'implementazione del filtro è opportuno riportare il sistema in forma di stato come in 2.4:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + \omega(k) \\ y(k) = Cx(k) + \nu(k) \end{cases} \quad (2.4)$$

dove  $\omega(k)$  e  $\nu(k)$  rappresentano i rumori di processo e misura rispettivamente, considerati bianchi e a media nulla e con matrici di covarianze rispettivamente :

$$\begin{aligned} E[w_k w_k^T] &= Q_k \\ E[\nu_k \nu_k^T] &= R_k \end{aligned} \quad (2.5)$$

La matrice dinamica  $A \in \mathbb{R}^{9 \times 9}$  assume dunque la seguente forma:

$$\mathbf{A} = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{1}{2}T^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T & \frac{1}{2}T^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

la matrice  $B \in \mathbb{R}^{9 \times 1}$  viene scelta considerando un vettore unitario:

$$\mathbf{B} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (2.7)$$

Si sottolinea che la scelta di questo vettore è arbitraria in quanto il vettore degli ingressi esogeni al sistema considerato è identicamente nullo ( $u = 0$ ).

Il vettore di uscita tiene conto delle misurazioni di posizione, velocità e accelerazione.

E' doveroso ricordare che la misura della velocità lungo  $z$  viene trascurata: il GPS è uno strumento utile ma poco affidabile nella misura di quest'ultima componente; per questo motivo  $y \in \mathbb{R}^{8 \times 1}$ .

La matrice  $C \in \mathbb{R}^{8 \times 9}$  (nell'ipotesi in cui tutti i flag delle misure dei sensori disponibili siano attivi) è definita quindi nel seguente modo:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

In definitiva il sistema finale assume la seguente forma:

$$\mathbf{x}(k+1) = \begin{bmatrix} p_x(k+1) \\ v_x(k+1) \\ a_x(k+1) \\ p_y(k+1) \\ v_y(k+1) \\ a_y(k+1) \\ p_z(k+1) \\ v_z(k+1) \\ a_z(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{1}{2}T^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T & \frac{1}{2}T^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x(k) \\ v_x(k) \\ a_x(k) \\ p_y(k) \\ v_y(k) \\ a_y(k) \\ p_z(k) \\ v_z(k) \\ a_z(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} u \quad (2.9)$$

$$\mathbf{y}(\mathbf{k}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x(k) \\ v_x(k) \\ a_x(k) \\ p_y(k) \\ v_y(k) \\ a_y(k) \\ p_z(k) \\ v_z(k) \\ a_z(k) \end{bmatrix} \quad (2.10)$$

Partendo da questo sistema si può quindi applicare l'algoritmo di stima degli stati di navigazione espresso nei due passi di predizione e correzione:

- **Predizione**

$$\begin{cases} x = Ax + Bu; \\ P = APA^T + Q \end{cases} \quad (2.11)$$

- **Correzione**

$$\begin{cases} K = PC^T(R + CPC^T)^{-1} \\ x = x + K(y - Cx); \\ P = (I_{9 \times 9} - KC)P \end{cases} \quad (2.12)$$

Per tenere conto della diversa frequenza dei sensori utilizzati, e quindi della disponibilità o meno delle misure nei diversi istanti di campionamento, si utilizza un *flag* in ingresso al filtro (come spiegato nell'introduzione). In base alla disponibilità o meno delle misure, la matrice  $C$  descritta in 2.8 viene opportunamente modificata all'interno del passo di **update**, in modo da considerare solo le righe relative alle misure in quell'istante disponibili.

### 3. Filtro di Kalman per la stima del Bias

Le misure dell'accelerometro sono soggette a un errore sistematico quale è il bias; l'integrazione di quest'ultime può portare quindi a una deriva dell'errore e a peggiorare notevolmente le performance dell'algoritmo di navigazione.

Il bias rappresenta una caratteristica del sensore non misurabile e viene espressa in *body frame*.

Per effettuare una stima del **bias** lo consideriamo come una nuova componente aggiuntiva del vettore di stato da stimare attraverso il filtro di Kalman: viene preso quindi un nuovo vettore di stato aumentato  $\tilde{x}$ , in cui includere le componenti del *bias* lungo le tre direzioni.

Il sistema aumentato ottenuto è rappresentato dalla seguente equazione:

$$\begin{cases} \tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{B}\tilde{u}(k) \\ \tilde{y}(k) = \tilde{C}\tilde{x}(k) + \nu(k) \end{cases} \quad (3.1)$$

in cui  $\tilde{A}$ ,  $\tilde{B}$ ,  $\tilde{C}$  rappresentano le matrici caratteristiche del nuovo sistema su cui è possibile applicare i passi di predizione e correzione del filtro di Kalman.

Detto quindi:

$$\epsilon^b = \begin{bmatrix} \epsilon_x^b \\ \epsilon_y^b \\ \epsilon_z^b \end{bmatrix} \quad (3.2)$$

il vettore delle componenti del bias lungo le tre dimensioni espresse in terna **body**, il vettore di stato aumentato diventa:

$$\tilde{x}(k) = \begin{bmatrix} p_x(k) \\ v_x(k) \\ a_x(k) \\ p_y(k) \\ v_y(k) \\ a_y(k) \\ p_z(k) \\ v_z(k) \\ a_z(k) \\ \epsilon_x^b(k) \\ \epsilon_y^b(k) \\ \epsilon_z^b(k) \end{bmatrix} \in \mathbb{R}^{12 \times 1} \quad (3.3)$$

Considerando il nuovo vettore di stato la matrice  $\tilde{A} \in \mathbb{R}^{12 \times 12}$  diventa:

$$\tilde{A} = \begin{bmatrix} subA & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & subA & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & subA & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \quad (3.4)$$



dove  $0_{3 \times 3}$  rappresenta il blocco di zeri,  $I_{3 \times 3}$  la sottomatrice identità relativa alle componenti di bias e  $subA$  la sottomatrice  $3 \times 3$  legata a posizione, velocità e accelerazione riportata di seguito:

$$subA = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Il vettore  $B \in R^{12 \times 1}$  è scelto arbitrariamente come un vettore unitario per gli stessi motivi espressi precedentemente.

Il vettore delle misure è quindi rappresentato da  $y \in \mathbb{R}^{8 \times 1}$  che tiene conto delle misure di posizione, velocità e accelerazione lungo le tre componenti (anche in questo caso viene esclusa la velocità lungo z dalle misure).

E' opportuno notare che il bias è espresso in terna **body**, mentre le restanti uscite del sistema sono espresse in terna **navigation**; risulta necessario introdurre una matrice di rotazione che permetta il cambio di coordinate dal frame body a navigation, descritta dalla composizione di rotazioni elementari rispetto agli angoli di roll ( $\phi$ ), pitch ( $\theta$ ) e yaw ( $\psi$ ):

$$C_b^n = C_z^T(\psi)C_y^T(\theta)C_x^T(\phi) \quad (3.6)$$

La matrice ottenuta viene riportata di seguito:

$$C_b^n = \begin{bmatrix} c(\theta)c(\psi) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) \\ s(\psi)c(\theta) & s(\psi)c(\theta)s(\phi) + c(\psi)c(\phi) & -c(\psi)s(\phi) + s(\psi)s(\theta)c(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \quad (3.7)$$

Per cui il bias espresso in terna navigation risulta:

$$\epsilon^n = C_b^n \epsilon^b \quad (3.8)$$

in componenti:

$$\begin{cases} \epsilon_x^n = c_{11}\epsilon_x^b + c_{12}\epsilon_y^b + c_{13}\epsilon_z^b \\ \epsilon_y^n = c_{21}\epsilon_x^b + c_{22}\epsilon_y^b + c_{23}\epsilon_z^b \\ \epsilon_z^n = c_{31}\epsilon_x^b + c_{32}\epsilon_y^b + c_{33}\epsilon_z^b \end{cases} \quad (3.9)$$

dove  $c_{ij}$  rappresentano gli elementi corrispondenti della matrice di rotazione  $C_b^n$ .

Le misure di accelerazione prese in esame (già compensate della forza di gravità) sono quindi della forma seguente :

$$a^n = C_b^n * (a^b + \epsilon^b) + \nu \quad (3.10)$$

dove  $a^b$  rappresente l'accelerazione dinamica del sistema espressa in body frame e  $\nu$  il rumore di misura relativo.

La matrice  $\tilde{C} \in R^{8 \times 12}$  delle uscite (in cui si considera la componente di bias riportata in navigation frame alla misura di accelerazione) assume quindi la seguente forma:

$$\tilde{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & c_{11} & c_{12} & c_{13} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & c_{21} & c_{22} & c_{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & c_{31} & c_{32} & c_{33} \end{bmatrix} \quad (3.11)$$

con  $c_{ij}$  elementi della matrice di rotazione riportata in 3.8.

Si noti come in questo caso il filtro di navigazione prenda in ingresso anche gli angoli di assetto del drone. Partendo quindi dal sistema aumentato riportato in 3.1 con le matrici appena descritte è possibile dunque applicare l'algoritmo di Kalman per la stima dello stato descritto nel capitolo 2, e valutare se una stima del bias degli accelerometri può portare vantaggi in termini di filtro di navigazione.

## 3.1 Prove preliminari

### 3.1.1 Filtro ad 1 grado di libertà

Data la struttura descritta nel capitolo precedente, per iniziare si è deciso di provare il fitro su un sistema semplice ad un solo grado di libertà, in modo da valutare l'osservabilità del nuovo stato da stimare.

Sotto queste ipotesi infatti risulta più facile trovare il tuning dei parametri del filtro senza incorrere nelle problematiche legate agli errori presenti nella stima dell'assetto del drone.

Per fare ciò è stato creato un dataset 'fittizio' (espresso in *navigation frame*) di misure di posizione, velocità e accelerazione da utilizzare in simulazione e campionate alla frequenza di interesse (nel nostro caso  $f = 50Hz$ ). L'accelerazione 'vera' viene ottenuta come somma di tre sinusoidi ad ampiezza, frequenza e fase diversa come riportato di seguito:

$$acc = 0.02 * \sin(0.01 * t) + (0.03) * \sin(0.05 * t + \pi) + (0.04) * \sin(0.03 * t + \pi/2); \quad (3.12)$$

Le espressioni analitiche di velocità e posizione vengono invece ricavate con meccanizzazione di questo segnale, ovvero integrando direttamente nell'ambiente di simulazione Matlab l'accelerazione appena descritta.

Per modellizzare al meglio le misure questi segnali sono stati 'sporcati' opportunamente sommando del rumore. Per quanto riguarda l'accelerazione, al segnale viene sommato direttamente un valore costante noto a rappresentazione del **bias** che vogliamo stimare.

Per i primi 100s di simulazione è stato scelto di mantenere fermo il sistema, quindi i segnali di accelerazione, velocità e posizione sono a 0.

Una rappresentazione dei segnali utilizzata è riportata nei grafici di seguito.

Inoltre si riporta un dettaglio della posizione vera confrontata con la misura nei primi 200s di simulazione per evidenziare l'andamento del rumore di misura intorno alla posizione iniziale.

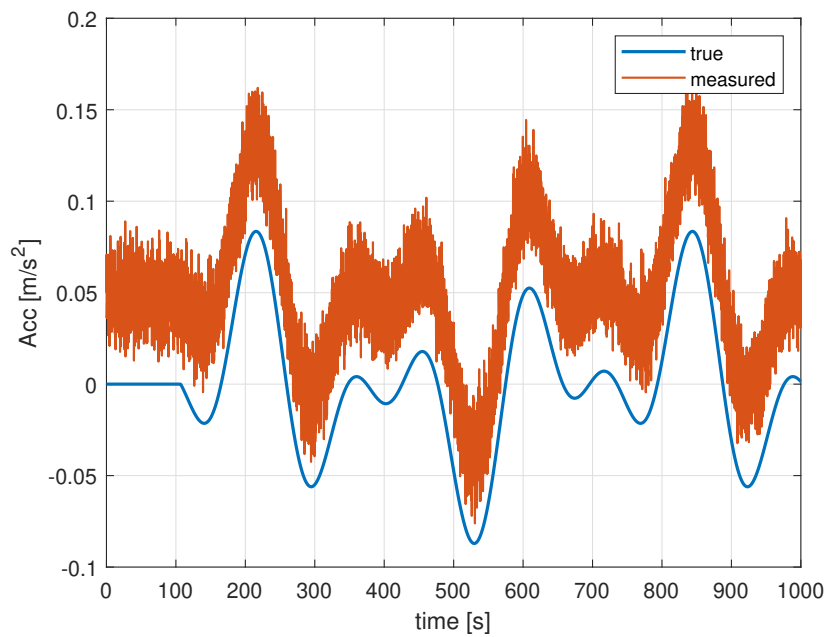


Figura 3.1: Accelerazione

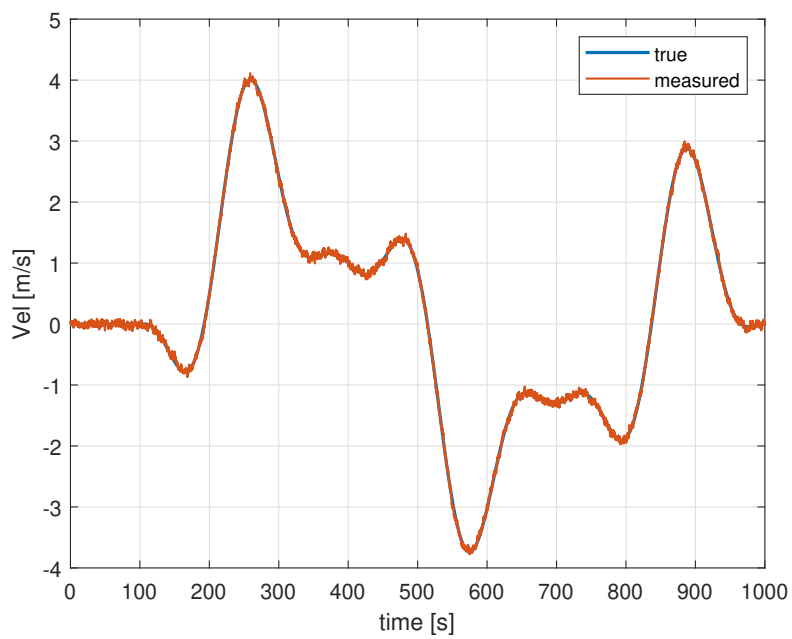


Figura 3.2: Velocità

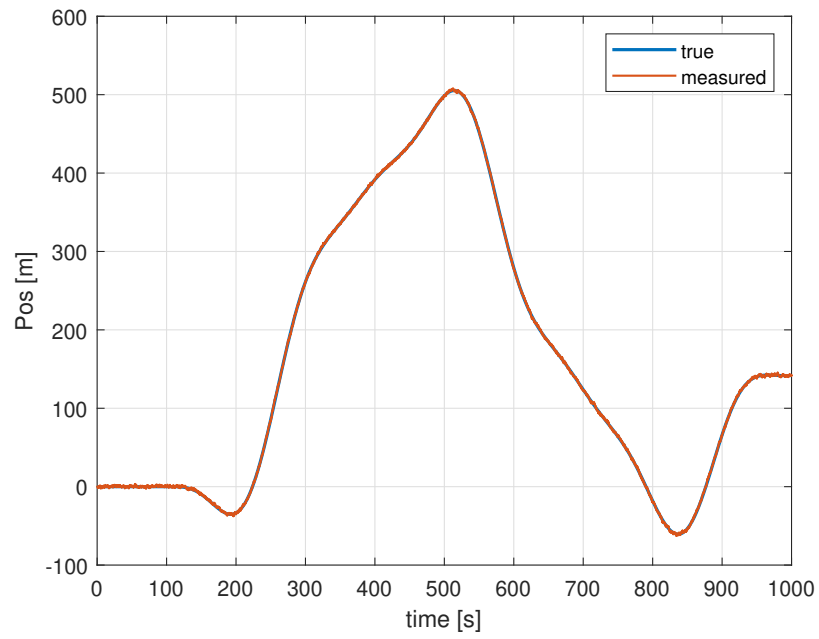


Figura 3.3: Posizione

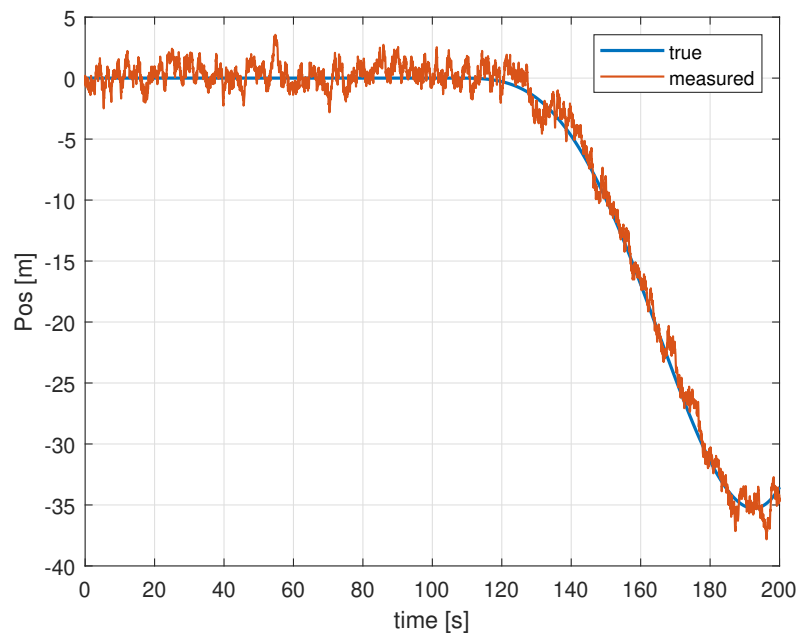


Figura 3.4: Posizione nei primi 200s

In questa analisi si è deciso di utilizzare la versione ridotta del filtro descritto in precedenza, andando quindi a considerare come vettore di stato  $x \in R^{4 \times 1}$ :

$$\mathbf{x}(\mathbf{k}) = \begin{bmatrix} p(k) \\ v(k) \\ a(k) \\ \epsilon(k) \end{bmatrix} \quad (3.13)$$

La matrice dinamica  $A$  viene modificata di conseguenza in:

$$A = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 & 0 \\ 0 & 1 & T & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

La matrice degli ingressi  $B \in R^{4 \times 1}$  assume valori arbitrari a piacere in assenza di ingressi esogeni.

Per avere un'analisi il più simile possibile ad una situazione reale si è deciso di mantenere in ingresso al filtro i vettore dei *flags* che indica per ogni istante di simulazione la disponibilità o meno delle misure di GPS e barometro.

Per questi motivi la matrice  $C$  del sistema varia in base a questi flags come segue:

- GPS, BARO, IMU attivi,  $C \in R^{3 \times 4}$ :

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.15)$$

- BARO, IMU attivi,  $C \in R^{2 \times 4}$ :

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad (3.16)$$

- solo IMU attivo,  $C$  è scalare e pari a 1.

In questa prima simulazione lo stato  $x_0$  e la matrice di covarianza dell'errore di stima  $P_0$  vengono inizializzate come segue:

$$x(0) = [p, v, a, 0] \quad (3.17)$$

$$P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix} \quad (3.18)$$

dove  $p, v, a$  rappresentano le misure di posizione, velocità e accelerazioni iniziali.

Visto che nessuna conoscenza a priori sul valore di bias da stimare è disponibile si è deciso di imporre come condizione iniziale sullo stesso un valore nullo, e conseguentemente a questa stima è stata associata un valore iniziale di  $P$  più elevato.

Come riportato in appendice, la matrice  $Q_k$  è relativa al rumore di processo. La scelta dei parametri di questa matrice è delicata e definita esclusivamente da un "tuning".

Dopo qualche simulazione, i parametri scelti per la matrice di  $Q$  sono i seguenti:

$$Q = \begin{bmatrix} 10^{-2} & 0 & 0 & 0 \\ 0 & 10^{-6} & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 \\ 0 & 0 & 0 & 10^{-12} \end{bmatrix} * 2 \quad (3.19)$$

Per quanto riguarda invece la matrice  $R_k$  del filtro, nel caso di tutte le misure disponibili, è scelta come:

$$R = \begin{bmatrix} GPS_{cov}^{(1)} & 0 & 0 \\ 0 & GPS_{cov}^{(2)} & 0 \\ 0 & 0 & IMU_{cov} \end{bmatrix} \quad (3.20)$$

con:

$$GPS_{cov} = [1 \ 2] \quad (3.21)$$

$$IMU_{cov} = 0.525 \quad (3.22)$$

relative rispettivamente alle misure di posizione, velocità e accelerazione.

La stima del bias ottenuta con i parametri del filtro appena descritti è riportata di seguito:

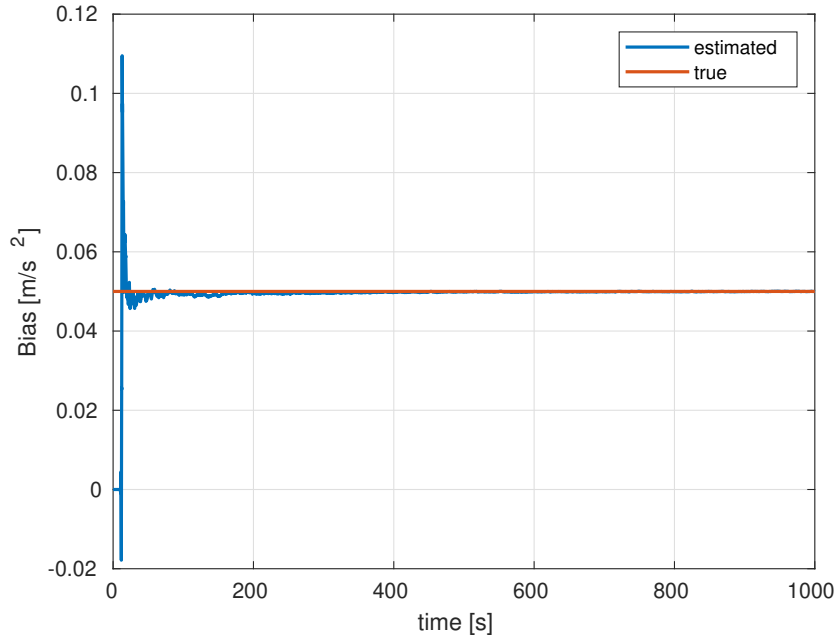


Figura 3.5: Bias stimato dal filtro di Kalman ad un grado di libertà.

Come si nota in figura 3.5, la stima dopo un primo picco converge al valore 'vero', indicando in questa prima analisi semplificata come il bias rappresenti una caratteristica del tutto osservabile.

Nei seguenti grafici vengono riportati i risultati ottenuti, confrontando la stima di posizione e velocità con i dati simulati ideali e quelli misurati. In particolare nella figura 3.8 si evidenzia come la stima di accelerazione sia in effetti 'pulita dal bias' inserito.

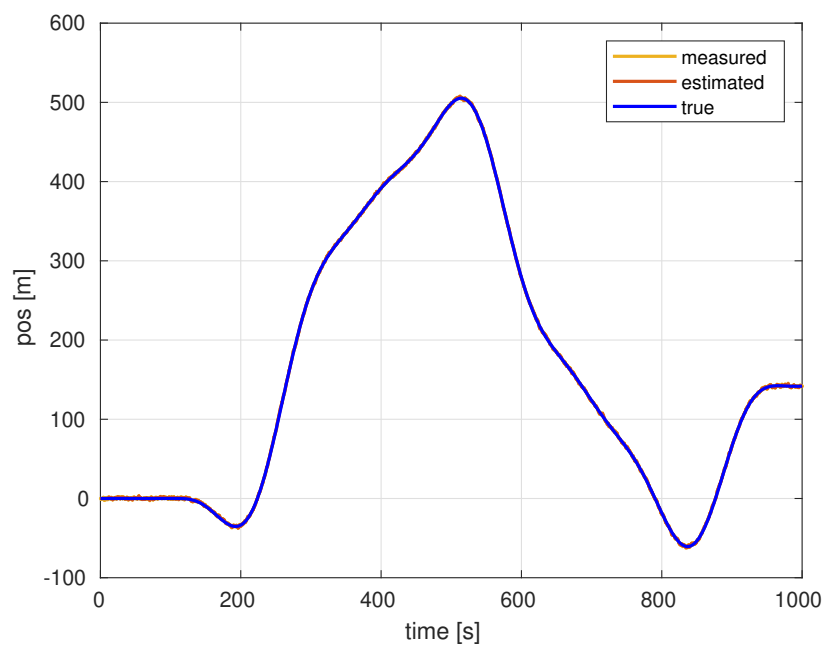


Figura 3.6: Confronto posizione stimata e reale

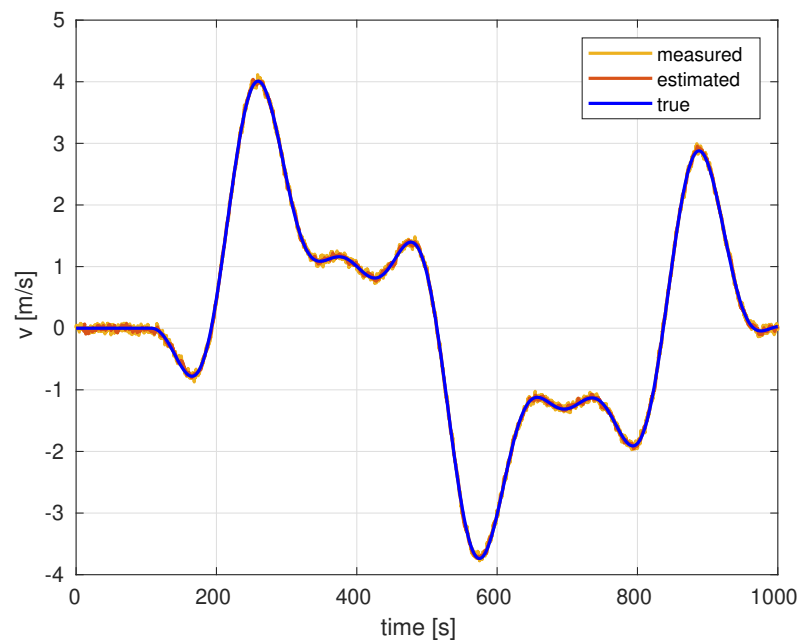


Figura 3.7: Confronto velocità stimata e reale

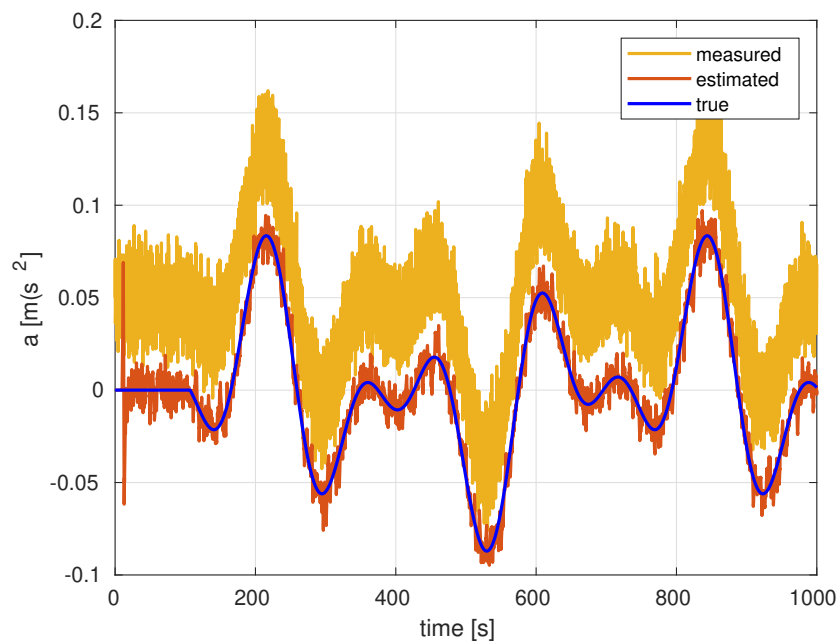


Figura 3.8: Confronto accelerazione stimata, misurata e reale



### 3.1.2 Filtro a 3 gradi di libertà

Per valutare ulteriormente le prestazioni di un filtro come quello descritto per la stima del bias di un accelerometro triassiale, le prove di simulazione sono state effettuate sui 3 gradi di libertà.

Anche in questo caso viene creato un dataset fittizio corrispondente alle misure in ingresso al filtro.

In queste prove, il segnale di accelerazione dinamica espresso in *navigation frame* assume una forma nelle tre componenti del tipo:

$$a_x = 0.02 \sin(0.01 t) + 0.03 \sin(0.05 t + \pi) + 0.04 \sin(0.03 t + \pi/2); \quad (3.23)$$

$$a_y = 0.02 \sin(0.01 t) + 0.05 \sin(0.07 t + \pi) + 0.04 \sin(0.05 t + \pi/2); \quad (3.24)$$

$$a_z = 0.02 \sin(0.01 t) + 0.03 \sin(0.05 t + \pi) + 0.04 \sin(0.03 t + \pi/2); \quad (3.25)$$

Come in precedenza, i segnali 'veri' di velocità e posizione sono ottenuti direttamente tramite integrazione su simulink delle rispettive misure di accelerazioni e poi opportunamente sporcati per assomigliare il più possibile ai *dati inerziali* e alle misure del *Dataset* reali; anche qui, per i primi 100sec posizione, velocità e accelerazione sono nulle.

Una valutazione a parte merita la simulazione del bias aggiuntivo alle misure. In questa analisi per mantenere le ipotesi semplificative viste in precedenza, ovvero considerando per ora la stima dell'assetto scollegata dalle misure di accelerazione, si è deciso di utilizzare gli angoli di roll, pitch, yaw registrati durante una prova di volo reale, considerandoli quindi come parametrizzazione dell'orientazione del sistema durante tutta la nostra simulazione.

Questi angoli vengono utilizzati per ruotare opportunamente il *bias* da noi considerato in body frame e aggiunto alle misure di accelerazione in *navigation frame* riportate nelle tre equazioni precedenti.

Gli stessi angoli sono necessari in ingresso al filtro di navigazione per computare la matrice  $C$  del modello, come spiegato in 3.11. Una rappresentazione degli angoli assetto utilizzati è mostrata nella figura seguente.

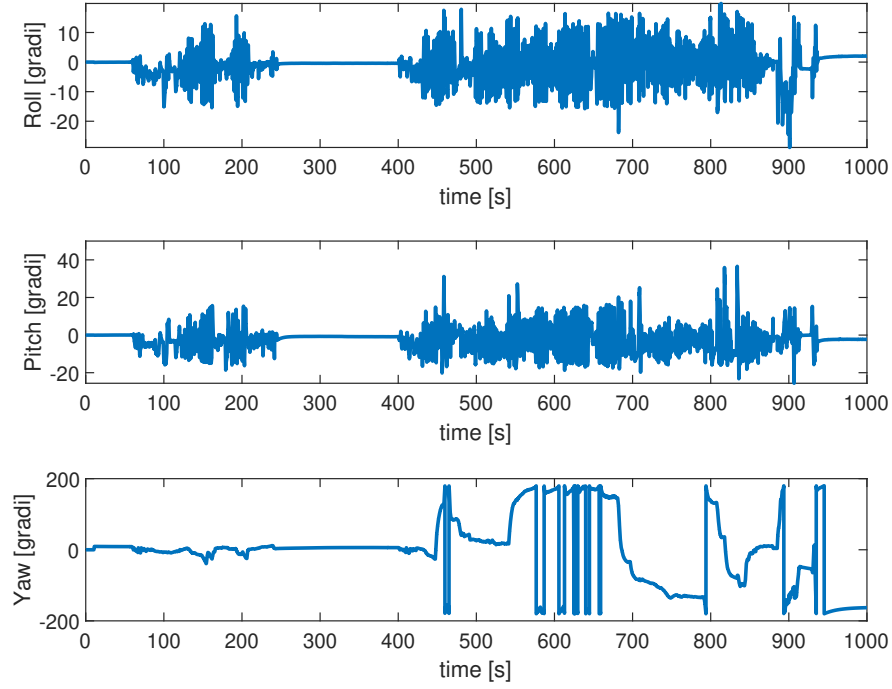


Figura 3.9: Angoli di Eulero

Il filtro di Kalman utilizzato quindi sui tre gradi di libertà presenta la stessa struttura spiegata nel capitolo 3. Mantenendo le considerazioni viste sul filtro implementato su un singolo grado di libertà, abbiamo ora quindi inizializzato come segue l'algoritmo:

$$x(0) = [p, v, a, 0] \quad (3.26)$$

con  $p, v, a \in R^{3 \times 1}$  le misure sui 3 gradi di libertà di posizione, velocità e accelerazione. La matrice  $P_0 \in R^{12 \times 12}$ :

$$P_0 = \begin{bmatrix} I_{3 \times 3} & 0 & 0 & 0 \\ 0 & I_{3 \times 3} & 0 & 0 \\ 0 & 0 & I_{3 \times 3} & 0 \\ 0 & 0 & 0 & 50 * I_{3 \times 3} \end{bmatrix} \quad (3.27)$$

I valori delle matrici di tuning del filtro  $Q$  e  $R$  scelte sono:

$$Q = \begin{bmatrix} 10^{-2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-12} \end{bmatrix} \quad (3.28)$$

$$R = \begin{bmatrix} GPS_{cov}^{(1)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & GPS_{cov}^{(3)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & IMU_{cov}^{(1)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & GPS_{cov}^{(2)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & GPS_{cov}^{(4)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & IMU_{cov}^{(2)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & BARO_{cov} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & IMU_{cov}^{(3)} \end{bmatrix} \quad (3.29)$$

con:

- $GPS_{cov} = [2 \ 2 \ 1 \ 1 \ 0.36]/2$
- $IMU_{cov} = 7.5*[0.07 \ 0.07 \ 10*0.09]$
- $BARO_{cov} = (1.7)/2$ .

La stima del bias ottenuta sui tre gradi di libertà è riportata in figura 3.10.

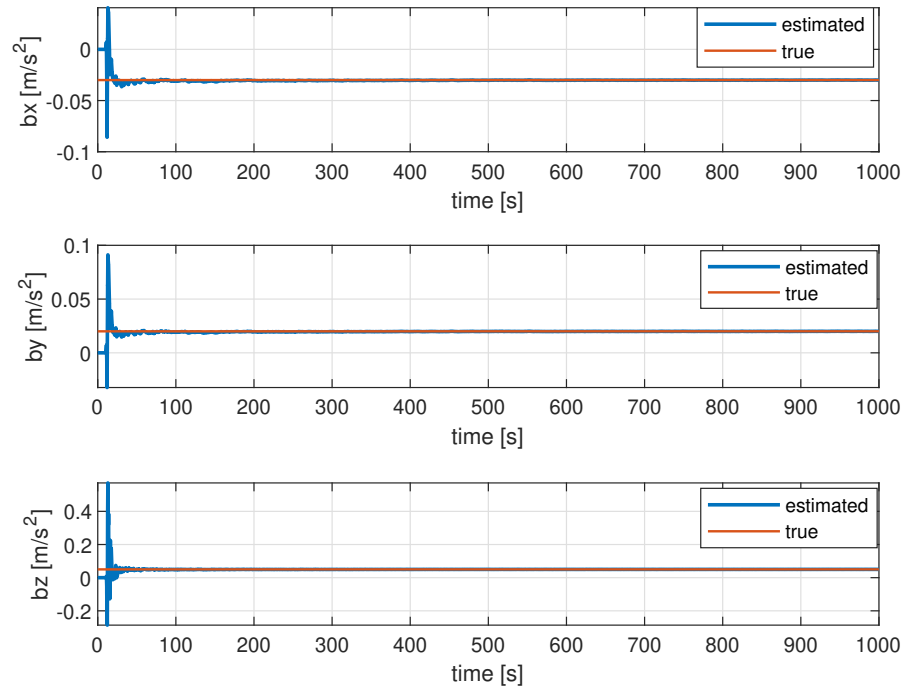


Figura 3.10: Stima dei bias

Anche in questo caso si osserva che dopo un picco iniziale, le stime dei tre bias convergono ai valori desiderati. Nei grafici seguenti vengono riportati i risultati del filtro di Kalman per quanto riguarda gli stati di navigazione:

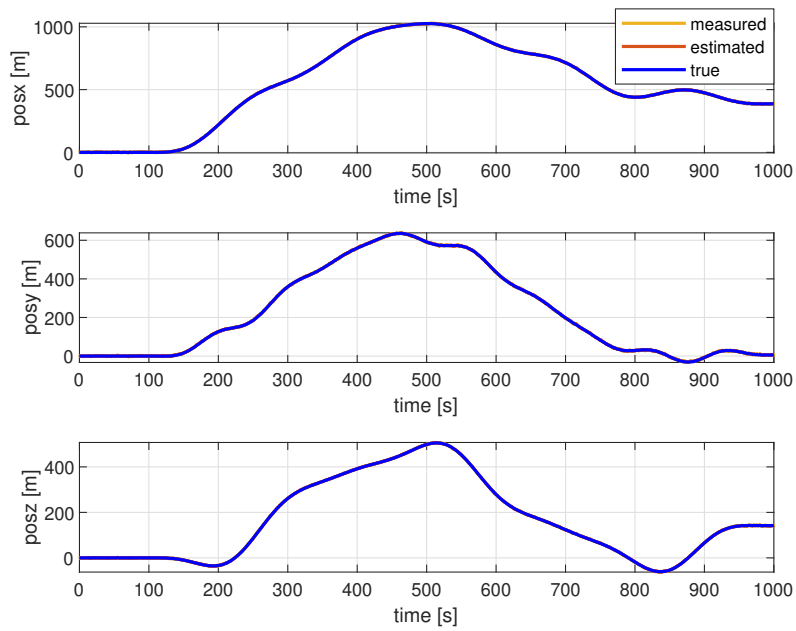


Figura 3.11: Posizione confronto

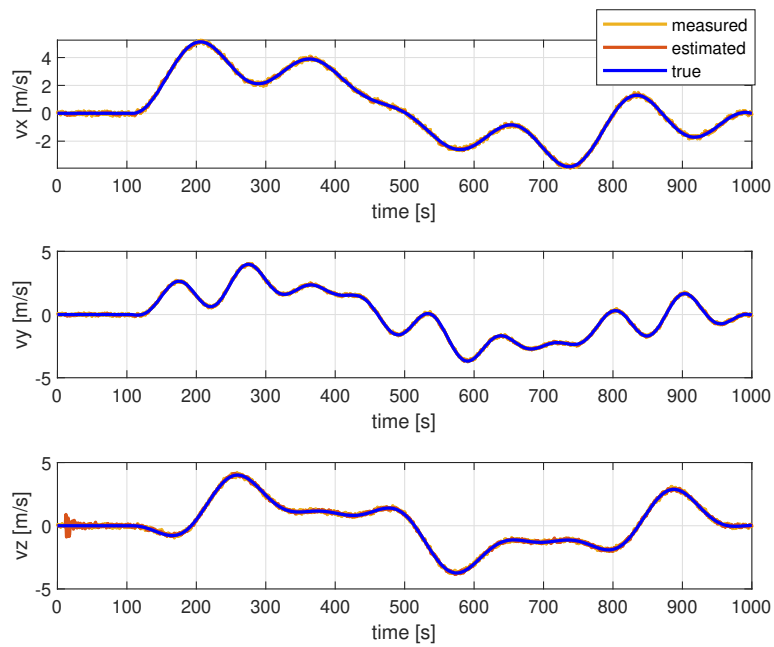


Figura 3.12: Velocità confronto

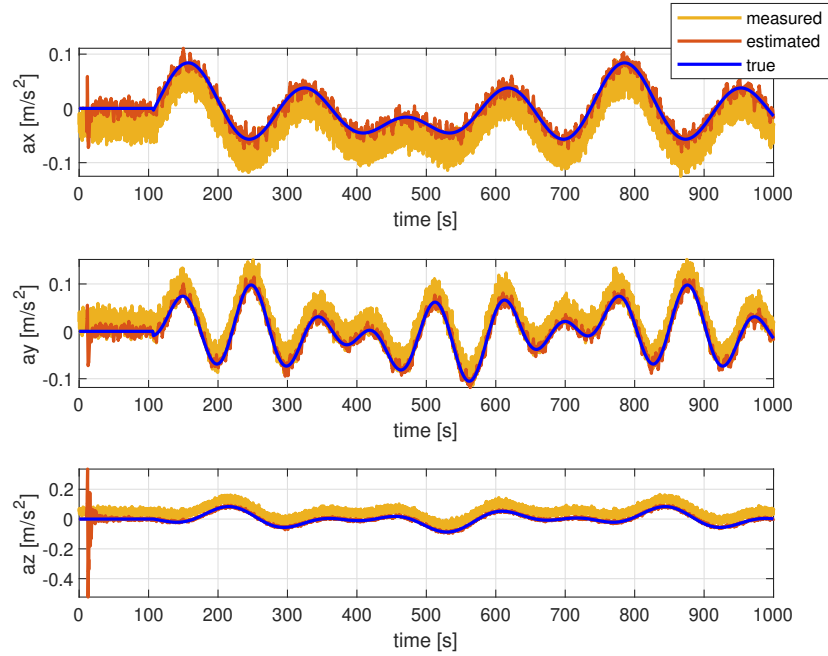


Figura 3.13: Accelerazione confronto

Come visto nel caso di un singolo grado di libertà, il filtro di Kalman modificato per la stima del bias permette di avere in uscita l'accelerazione stimata 'pulita' dal bias.

Questo si ripercuote nella stima dei dati 'veri' di posizione e velocità, infatti con la scelta di valori della matrice  $Q$  relativi all'equazione  $\dot{v} = a$  piccoli, 'fidandosi' molto del modello, si impone quindi che la velocità stimata sia sempre di più l'integrazione di *accelerazione - bias*.

L'effetto di quanto appena detto sulla stima del filtro si può osservare nei grafici di confronto tra la stima di posizione e velocità precedenti, e quella ottenuta con il filtro di navigazione senza stima del bias descritto nel capitolo 2.

Come si nota soprattutto in figura 3.14, la stima fatta con il nuovo filtro segue bene i dati veri simulati, mentre il filtro vecchio si attacca alle misure rumorose.

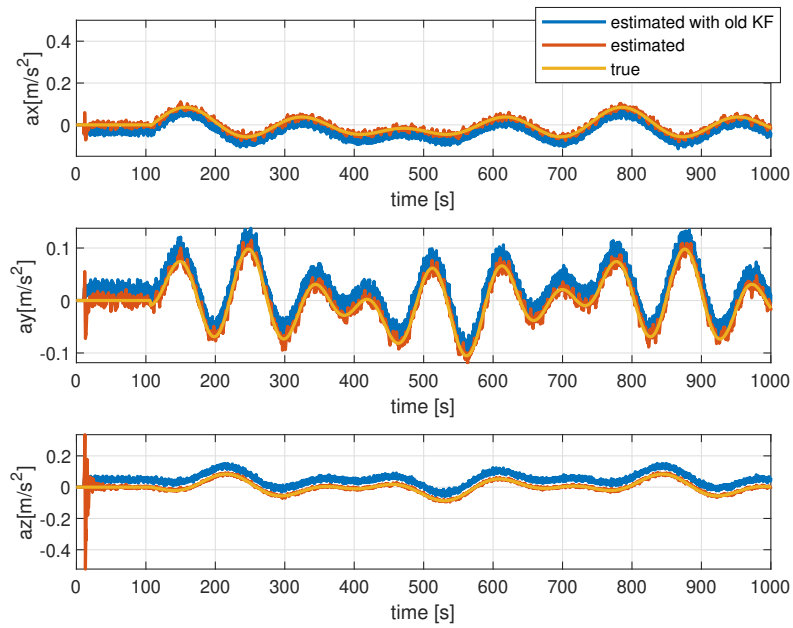


Figura 3.14: Confronto stima accelerazione

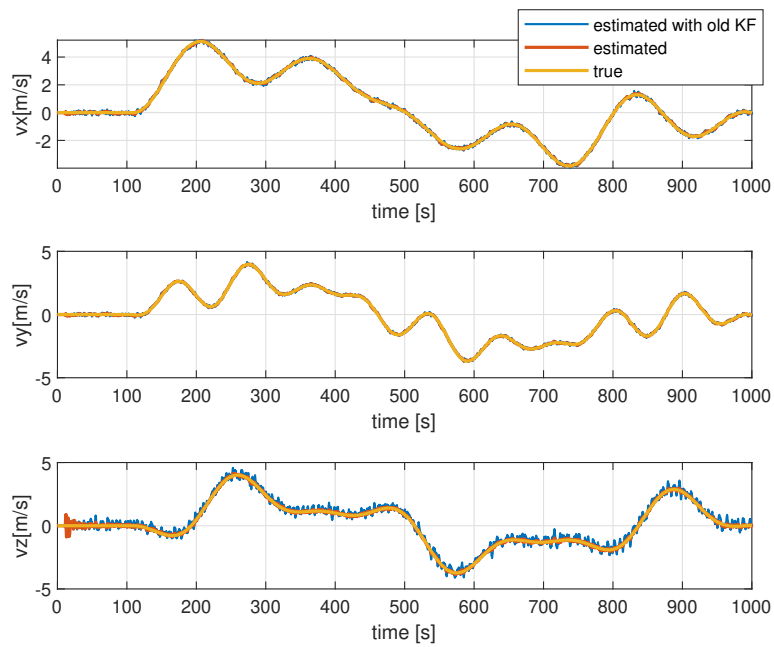


Figura 3.15: Confronto stima velocità

## 4. Ruolo dello stimatore di assetto AHRS

Nei capitoli precedenti per semplicità sono state analizzate le prestazioni del filtro considerando i dati simulati direttamente come misure in navigation frame, senza considerare il ruolo del bias dell'accelerometro sulla stima di assetto del sistema.

Infatti sul drone è presente un filtro AHRS: tramite la *sensor fusion* di misure provenienti da accelerometro, giroscopio e magnetometro stima gli angoli di eulero roll, pitch e yaw che parametrizzano l'orientazione. Nell'AHRS l'integrazione dell'angular rate viene fatta tramite quaternioni, e opportunamente corretta tramite le misure provenienti dagli altri due sensori.

L'algoritmo implementato all'interno del filtro AHRS fa in modo di trovare un assetto in cui roll e pitch siano tali che l'accelerazioni misurate su x e y siano pari a 0 (sensore fermo e livellato). Se è presente quindi un bias sulle accelerazioni l'AHRS modifica l'assetto per annullarlo, e quindi si verifica un errore nella stima.

Questo errore di stima degli angoli di orientazione si ripercuote direttamente sulla performance del filtro di Kalman, in quanto gli stessi angoli vengono utilizzati per ruotare le misure dei sensori in navigation frame.

Per ovviare a questa problematica, l'idea potrebbe essere di retroazionare la stima del bias e sottrarla alle misure di accelerometro in ingresso al filtro AHRS, in modo da avere una migliore convergenza della stima.

### 4.1 Filtro con feedback del bias stimato

#### 4.1.1 Creazione del primo dataset fittizio

L'obiettivo di questa sezione è di ottenere una 'situazione' più vicina alla realtà, ovvero simulare le misure dei sensori da ruotare opportunamente tramite l'assetto stimato dal filtro AHRS.

Come primo passo in ambiente di simulazione è stato creato un segnale ideale per le misure di un giroscopio. Il segnale scelto, mantenuto a 0 per i primi 107 secondi, è semplicemente riportato di seguito:

$$p = 0.04 \sin(2 \pi F_c(t - 107)); \quad (4.1)$$

$$q = 0.05 \sin(2 \pi F_c(t - 107)); \quad (4.2)$$

$$r = 0.02 \sin(2 \pi F_c(t - 107)); \quad (4.3)$$

con  $F_c$  che indica la frequenza scelta pari a  $0.025Hz$ .



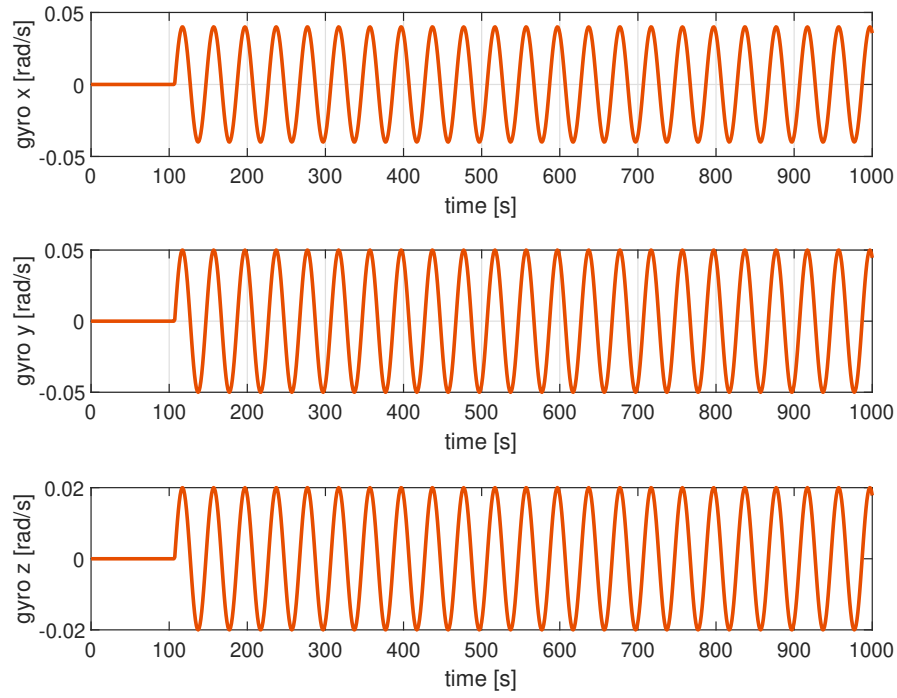


Figura 4.1: Angular rate ideale

Gli angoli di roll,pitch yaw ideali sono stati ottenuti tramite meccanizzazione dell'angular rate, utilizzando il filtro AHRS in cui le misure di accelerometro e magnetometro sono nulle, e si ha quindi semplicemente un'integrazione dell' angular rate ideale.

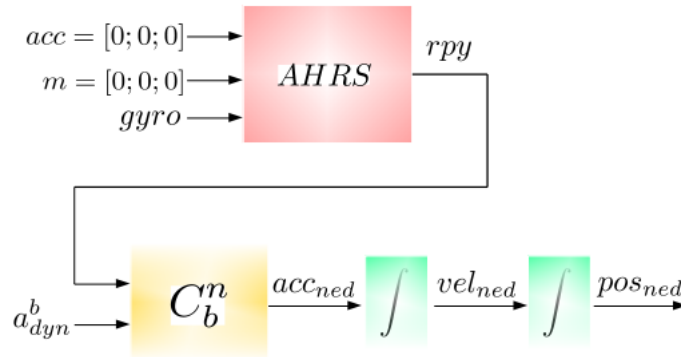


Figura 4.2: Schematizzazione Dataset fittizio

L'andamento degli angoli di eulero così ottenuto viene rappresentato nel seguente grafico:

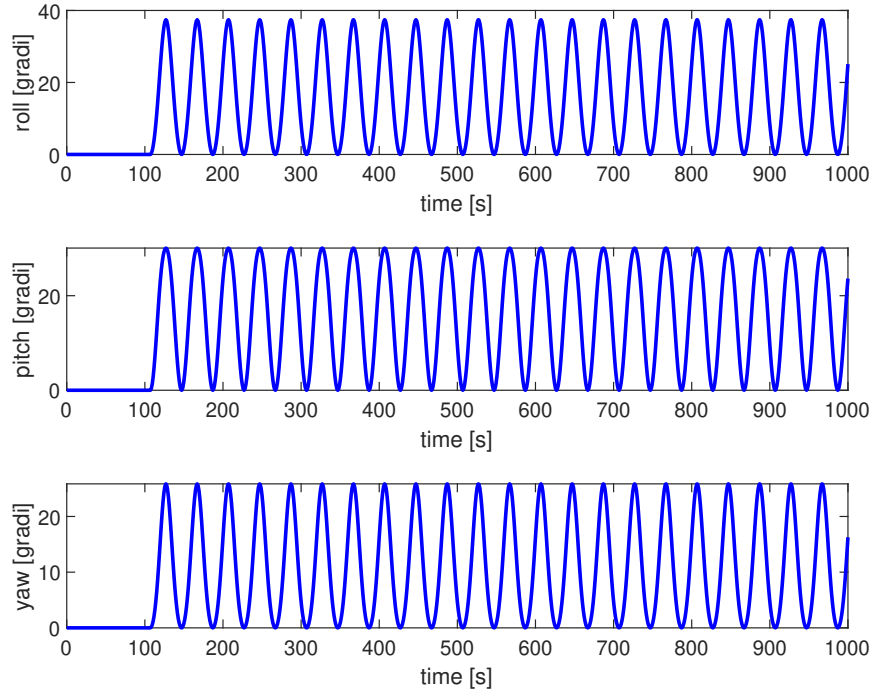


Figura 4.3: Angoli di roll,pitch e yaw ideali

La misura ideale dell'accelerometro (in assenza di rumore e bias) è rappresentata dall'equazione seguente:

$$acc^b = a_{dyn}^b + g^b \quad (4.4)$$

dove  $g^b = (C_b^n)^T g^n$  rappresenta il contributo di gravità espresso in body frame.

L'accelerazione dinamica misurata sui 3 gradi di libertà (posta uguale a 0 per i primi 107 secondi di simulazione) è la seguente:

$$a_{dyn} = 0.02 \sin(0.01 t) + 0.03 \sin(0.05 t + \pi) + 0.04 \sin(0.03 t + \pi/2); \quad (4.5)$$

Il segnale  $a_{dyn}$  appena descritto è stato ruotato tramite la  $C_b^n$  calcolata con gli angoli di assetto ideali rappresentati in figura 4.3 per ottenere l'accelerazione ideale in NED, successivamente integrata per simulare velocità e posizione ideali rispettivamente.

Nei seguenti grafici vengono riportati i segnali simulati ottenuti per posizione, velocità ed accelerazione.

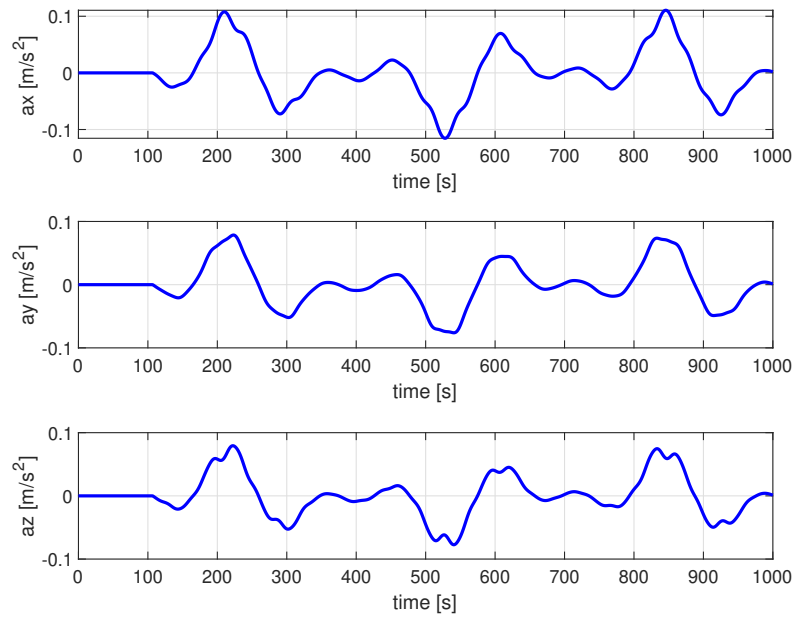


Figura 4.4: Accelerazione vera

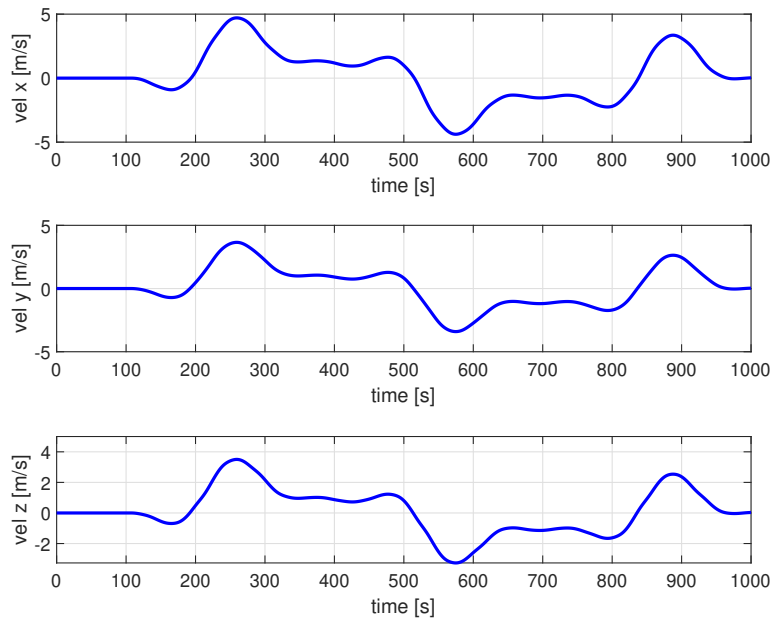


Figura 4.5: Velocità vera

b

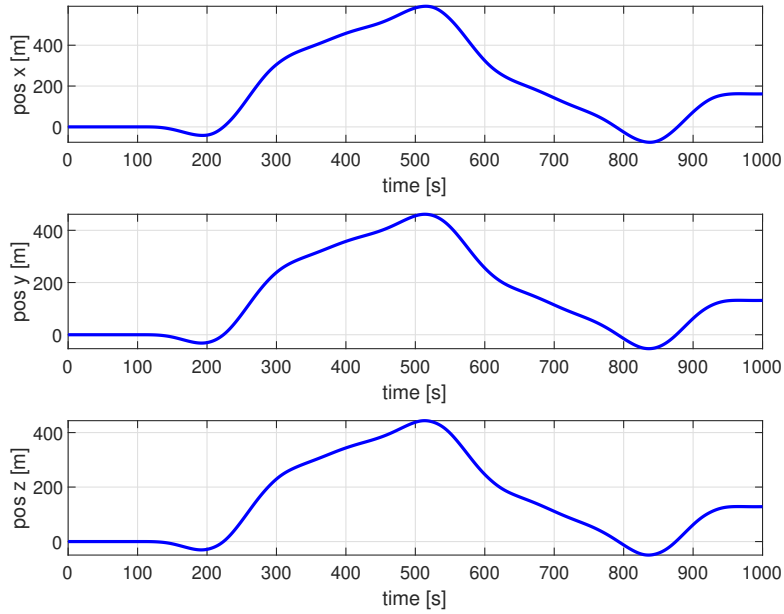


Figura 4.6: Posizione vera

Questi ultimi due segnali vengono poi opportunamente sporcati con del rumore prima di essere utilizzati come misure in ingresso al filtro di Kalman di navigazione.

Per quanto riguarda l'accelerazione, al fine di modellizzare un accelerometro 'reale', al segnale descritto in 4.4 viene sommato un bias e un rumore come segue:

$$acc_{noise}^b = a_{dyn}^b + g^b + \epsilon^b + \nu \quad (4.6)$$

Questo segnale diventa l'ingresso di un blocco AHRS che simula il filtro di bordo, in cui si utilizzano le misure di giroscopio ideali sporcate opportunamente con del rumore, mentre il magnetometro non viene considerato. L'accelerazione ottenuta ruotando con l'assetto calcolato da questo blocco, diventa la misura di accelerazione da fornire al filtro di Kalman.

### 4.1.2 Risultati della simulazione

Il dataset appena descritto è stato utilizzato per simulare il comportamento del Filtro di Kalman. Per valutare se fosse possibile migliorare la stima d'assetto, e di conseguenza le prestazioni del filtro, è stato deciso di retroazionare il bias stimato sottraendolo alla misura dell'accelerometro in ingresso al blocco AHRS. La matrice di rotazione calcolata con gli angoli di roll, pitch e yaw stimati viene quindi utilizzata per ruotare le misure in navigation frame in ingresso al filtro di navigazione; in particolare, per la misura di accelerazione, il dato inerziale viene ruotato con le stime di assetto e sommato con la gravità in navigation frame. Una schematizzazione della simulazione eseguita è riportata in fig 4.7:

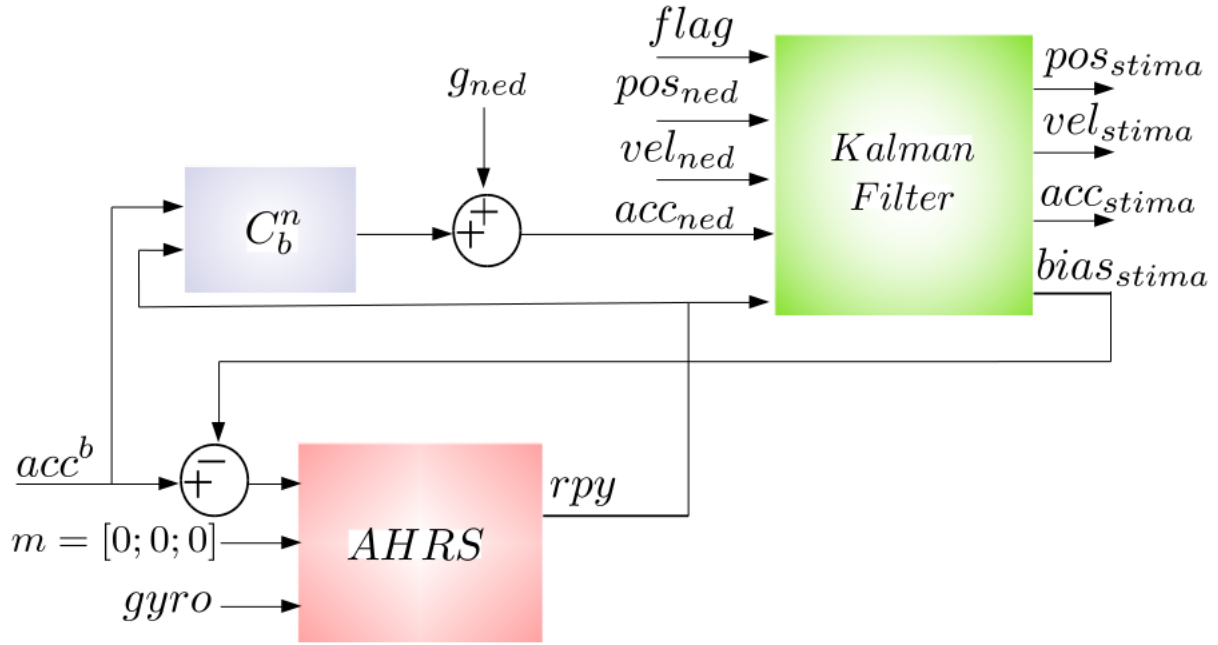


Figura 4.7: Schematizzazione Feedback del bias

Il vettore di stima è stato inizializzato nel modo seguente:

$$x(0) = [px, vx, ax, py, vy, ay, pz, vz, az, 0, 0, 0]^T \quad (4.7)$$

con  $p, v, a \in R^{3 \times 1}$  le misure sui 3 gradi di libertà di posizione, velocità e accelerazione. La matrice  $P_0 \in R^{12 \times 12}$ :

$$P_0 = \begin{bmatrix} I_{3 \times 3} & 0 & 0 & 0 \\ 0 & I_{3 \times 3} & 0 & 0 \\ 0 & 0 & I_{3 \times 3} & 0 \\ 0 & 0 & 0 & 100 * I_{3 \times 3} \end{bmatrix} \quad (4.8)$$

Dopo vari tentativi di tuning di seguito sono riportati i valori di  $Q$  e  $R$  scelti.

$$Q = \begin{bmatrix} p_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & v_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & v_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_y & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_z & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_z & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \epsilon_x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \epsilon_y & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \epsilon_z \end{bmatrix} \quad (4.9)$$

dove:

- $p_x = 1e-8$ ;
- $v_x = 1e-9$ ;
- $a_x = 1e-5$ ;
- $p_y = 1e-8$ ;
- $v_y = 1e-9$ ;
- $a_y = 1e-5$ ;
- $p_z = 1e-8$ ;
- $v_z = 1e-9$ ;
- $a_z = 1e-6$ ;
- $\epsilon_x = 1e-12$ ;
- $\epsilon_y = 1e-12$ ;
- $\epsilon_z = 1e-12$ ;

Ricordando che dalle misure in ingresso al filtro di navigazione viene esclusa la velocità lungo z, la matrice di covarianza  $R \in R^{8 \times 8}$ , nell'ipotesi di tutte le misure disponibili, assumerà la seguente forma:

$$R = \begin{bmatrix} GPS_{cov}^{(1)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & GPS_{cov}^{(3)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & IMU_{cov}^{(1)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & GPS_{cov}^{(2)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & GPS_{cov}^{(4)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & IMU_{cov}^{(2)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & BARO_{cov} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & IMU_{cov}^{(3)} \end{bmatrix} \quad (4.10)$$

Si noti che i valori di covarianza dei sensori sono ordinati nel seguente modo:

$$GPS_{cov} = [p_x \ p_y \ v_x \ v_y \ v_z] \quad (4.11)$$

con:

- $p_x = 0.1 \cdot 50$ ;
- $p_y = 0.1 \cdot 50$ ;
- $v_x = 200 \cdot 5$ ;
- $v_y = 200 \cdot 5$ ;
- $v_z = 0.36$ ;

$$IMU_{cov} = [a_x \ a_y \ a_z] \quad (4.12)$$

con:

- $a_x = 0.525 \cdot 50$ ;
- $a_y = 0.525 \cdot 50$ ;
- $a_z = 0.67 \cdot 50$ ;

$$BARO_{cov} = [p_z] \quad (4.13)$$

con  $p_z = 1.7/5$ .

Come primo risultato viene riportato il confronto tra l'assetto stimato dal filtro AHRS appena descritto con correzione del bias in ingresso all'accelerometro, e un filtro in cui questa correzione non viene effettuata.

Nella figura 4.8 si nota come l'unica differenza significativa in termini di stima si abbia nel caso dell'angolo di pitch, in cui si nota un piccolo miglioramento della stima nel caso di retroazione del bias stimato.

Negli altri angoli questa differenza non è significativa, e la stima di assetto in generale dei due schemi risulta paragonabile.

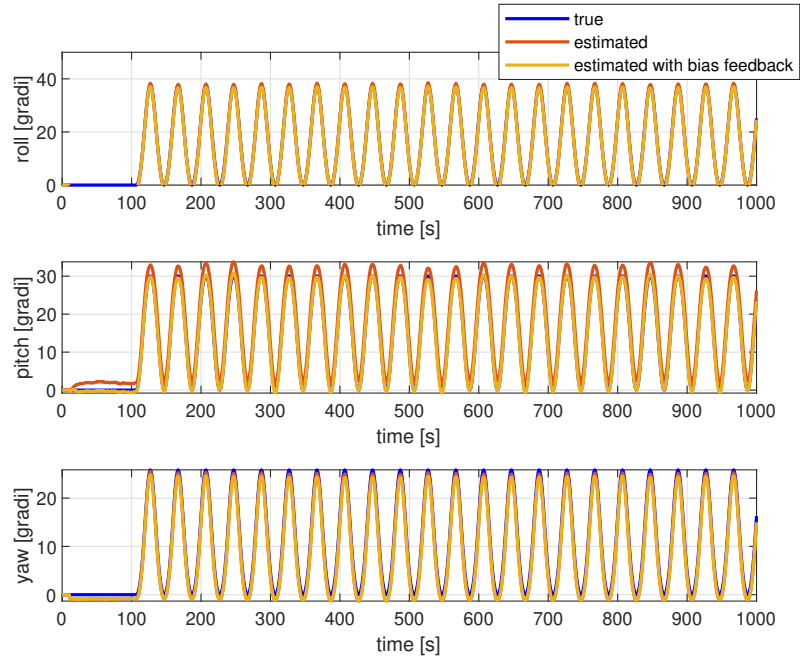


Figura 4.8: Confronto angoli di roll,pitch,yaw

I risultati in termini di stima degli stati di navigazione ottenuti con questo filtro di Kalman vengono mostrati nelle figure successive.



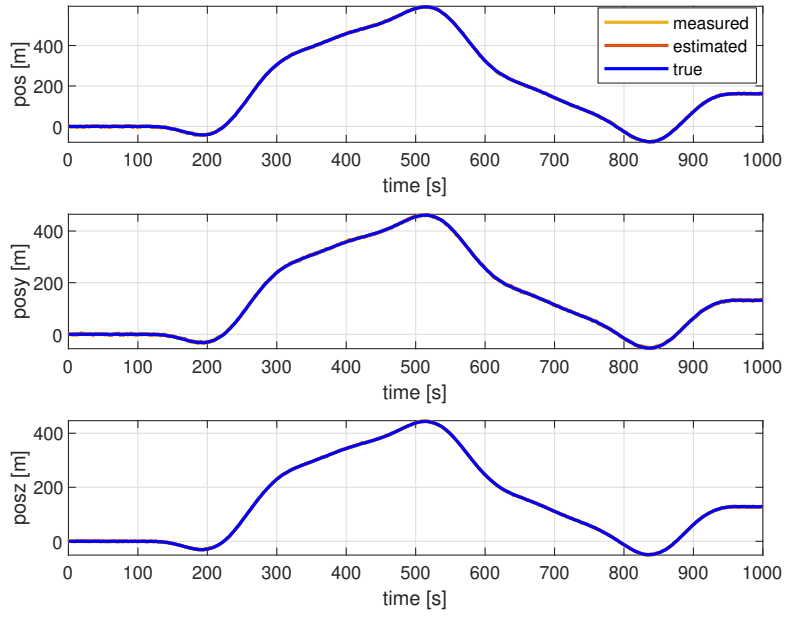


Figura 4.9: Confronto posizione stimata, misurata e reale

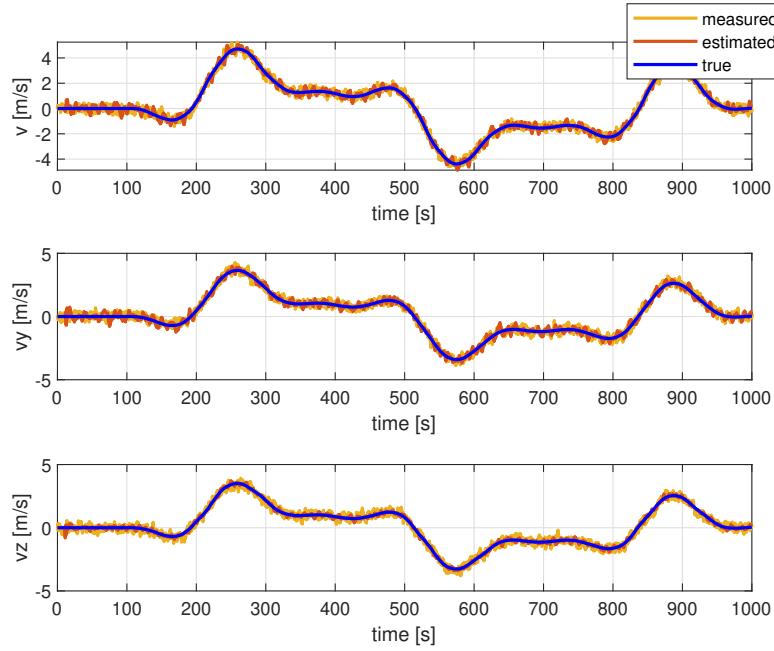


Figura 4.10: Confronto velocità stimata, misurata e reale

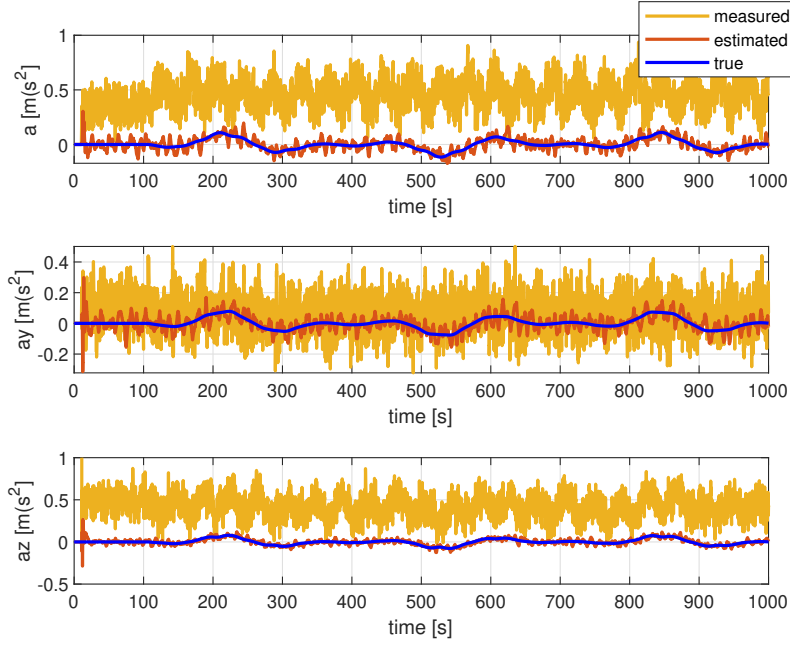


Figura 4.11: Confronto accelerazione misurata, stimata e reale

Infine in figura 4.12 viene riportata la stima dei bias ottenuta. Come si può notare la stima converge ad un valore costante diverso da quello desiderato sui tre gradi di libertà. Ciò può essere dovuto in parte al dataset scelto per la simulazione: si nota infatti come ad un'ampia traiettoria di posizione non corrisponda in effetti un altrettanto ampio moto di rotazione. In particolare il segnale di velocità angolare simulato e riportato in 4.1 se integrato dà luogo ad angoli di roll, pitch solo positivi, e valori di angoli di yaw piccoli e poco significativi in un intervallo di simulazione pari a 1000s.

Per queste ragioni nella sezione seguente viene mostrato un'altra simulazione con dataset diverso, per verificare se queste accortezze nella scelta del dataset possano influire in un miglioramento della stima dei bias.

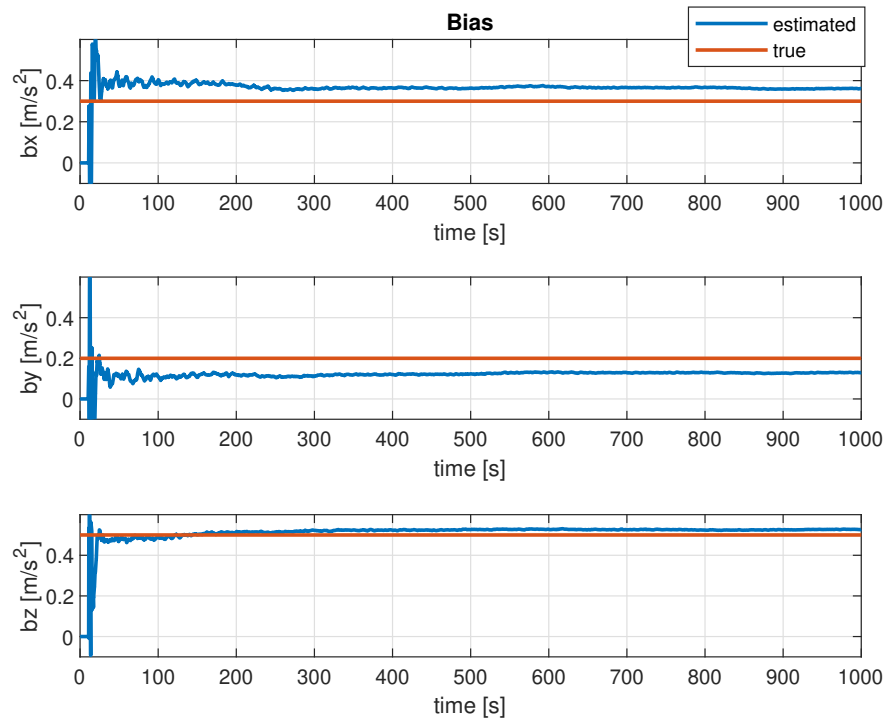


Figura 4.12: Stima del bias

### 4.1.3 Creazione del secondo dataset fittizio

In questa sezione, per indagare il possibile ruolo di accoppiamento dei segnali simulati sulla stima del bias, viene proposto un nuovo dataset.

Questo è costituito da un segnale di tipo *jerk* per l'accelerazione dinamica in *navigation frame*, differente per ogni grado di libertà.

Successivamente questa accelerazione viene integrata per ottenere velocità e posizione rispettivamente.

Una rappresentazione dei segnali creati è fornita nelle figure 4.13, 4.14 e 4.15.

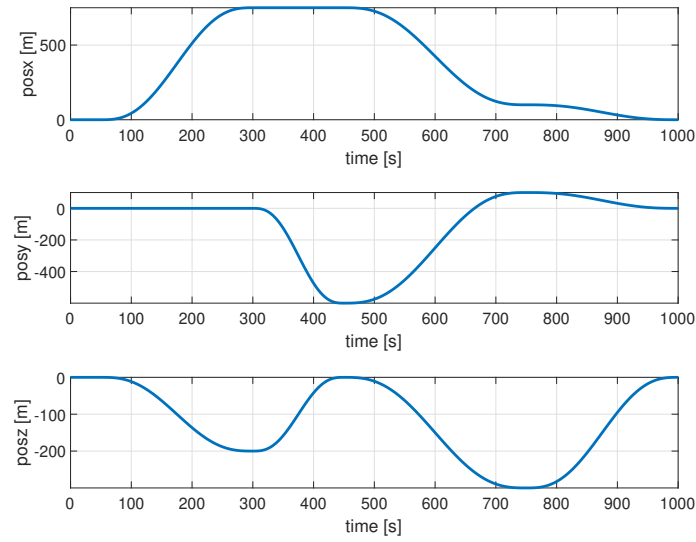


Figura 4.13: Posizione simulata

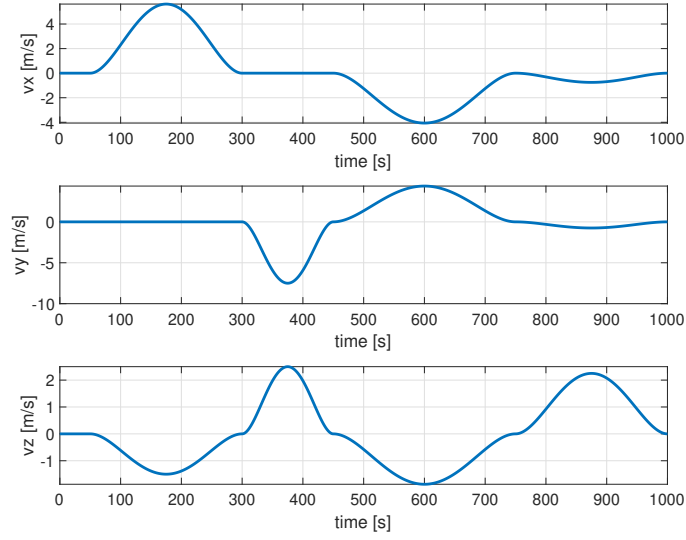


Figura 4.14: Velocità simulata

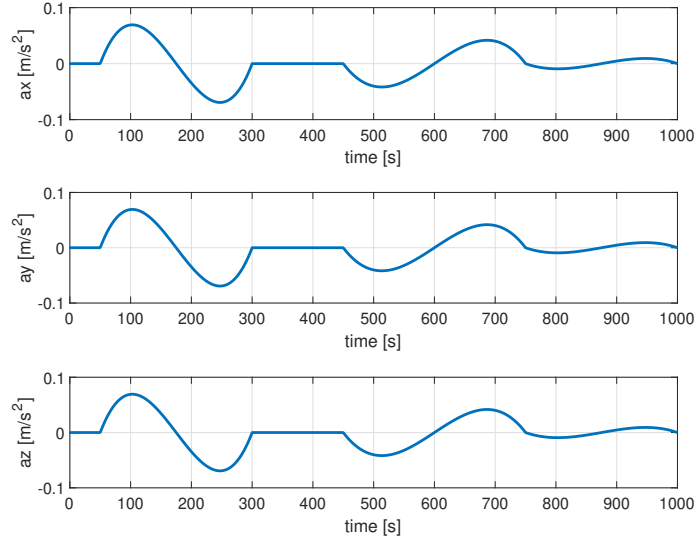


Figura 4.15: Accelerazione simulata

Per quanto riguarda l'assetto, è stato ottenuto con gli stessi passaggi descritti nella sezione 4.1.1. In questo caso, per avere una rotazione più ampia in accordo con la traiettoria scelta, la velocità angolare implementata è la seguente:

$$p = 0.04 \sin(2 \pi F_c(t - 107) + \pi/2); \quad (4.14)$$

$$q = 0.05 \sin(2 \pi F_c(t - 107) + \pi/2); \quad (4.15)$$

$$r = 0.02 \sin(2 \pi F_c(t - 107) + \pi/2) + (2\pi/1000); \quad (4.16)$$

Gli angoli di eulero così ottenuti vengono mostrati in figura 4.16 ed utilizzati per ruotare il segnale di accelerazione vero creato da navigation a body frame; quest'ultimo viene poi utilizzato come misura di accelerometro in ingresso al blocco AHRS come visto nella sezione precedente.

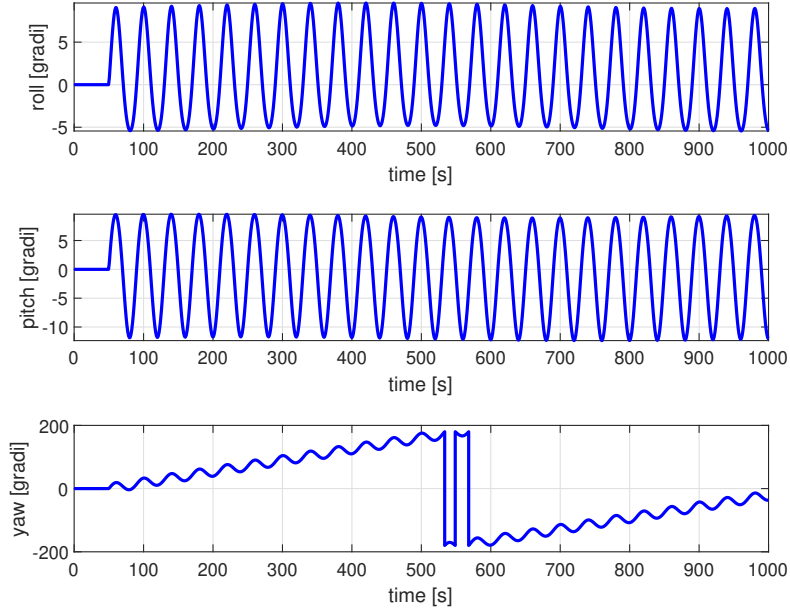


Figura 4.16: Angoli di eulero

#### 4.1.4 Risultati della simulazione

Anche in questo caso viene per prima riportato il confronto tra la stima di assetto ottenuta retroazionando il bias stimato per correggere le misure di accelerometro e quella in cui la retroazione non viene effettuata. Nella figura 4.17 risulta evidente come la correzione del bias dell'accelerometro porti ad un miglioramento significativo della stima degli angoli di roll e pitch in uscita dal blocco AHRS.

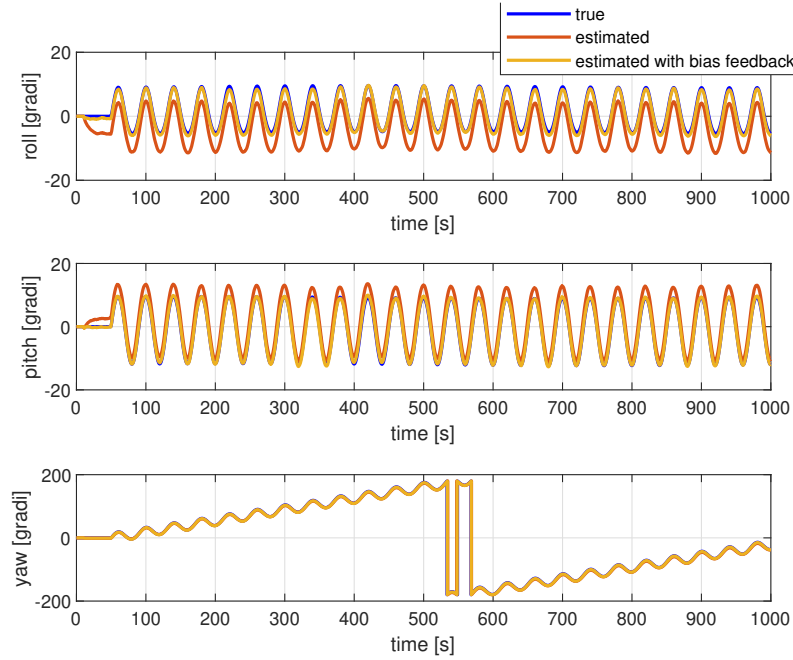


Figura 4.17: Confronto stima assetto

La stima dei bias ottenuta è riportata nel grafico seguente.

Si nota come in questo caso la stima dei bias converge, con piccoli errori, al valore desiderato, e permette quindi un algoritmo di navigazione più predittivo che correttivo, in accordo con gli obiettivi del lavoro presentato.

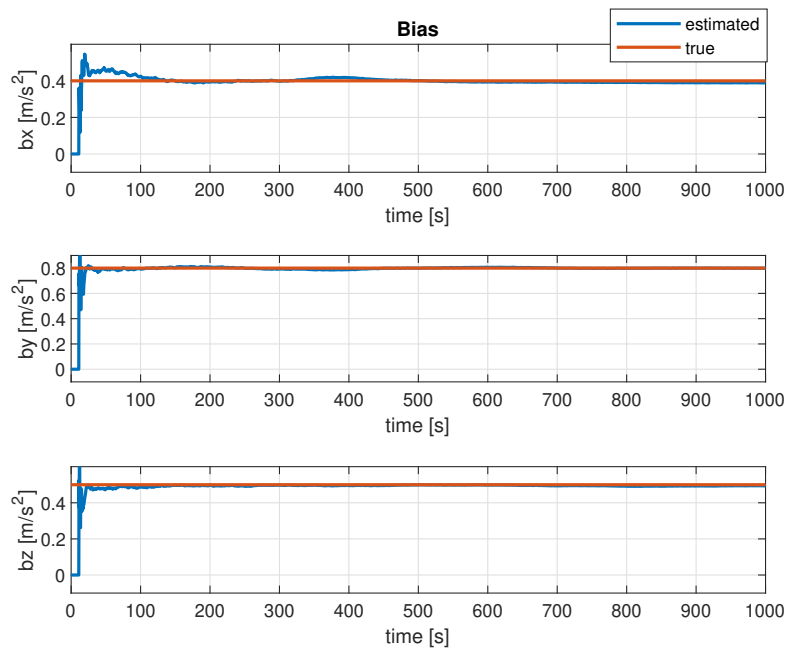


Figura 4.18: Stima del bias

Infine vengono riportate anche le stime di posizione, velocità e accelerazione ottenute a partire dal dataset descritto, utilizzando il tuning del filtro di Kalman della sezione 4.1.2.



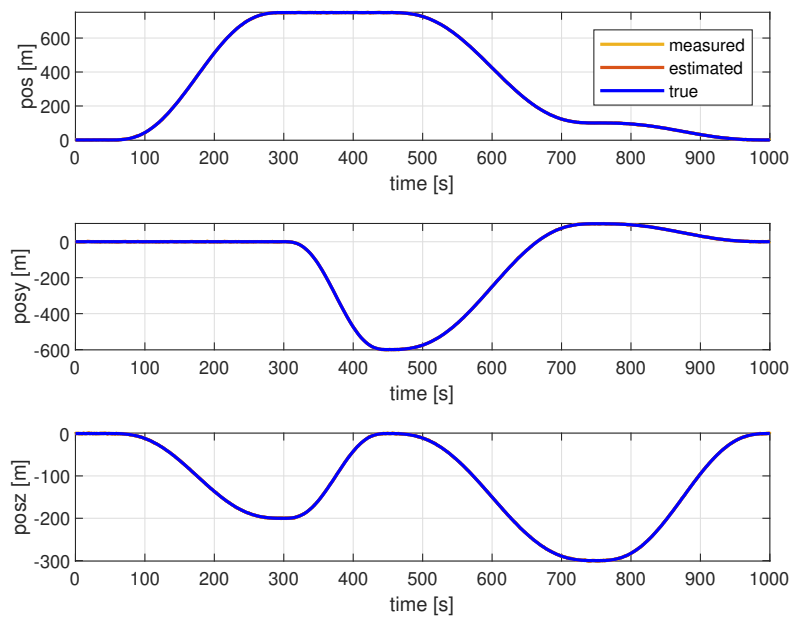


Figura 4.19: Confronto Posizione vera, misurata e stimata

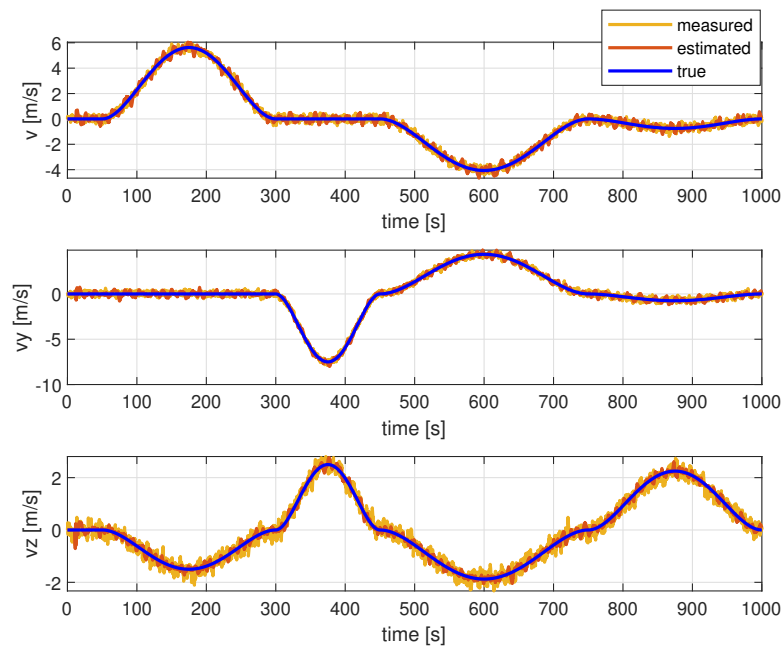


Figura 4.20: Confronto Velocità vera, misurata e stimata

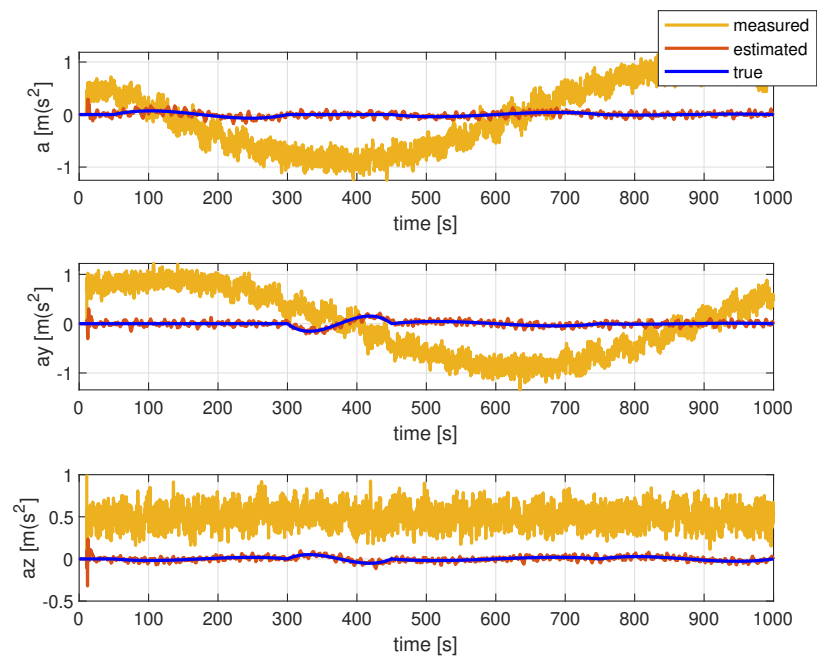


Figura 4.21: Confronto Accelerazione vera, misurata e stimata

## 5. Simulazione su dati reali

In questo capitolo le performance del filtro descritto vengono valutate su un dataset reale.

Si è deciso di utilizzare i dati registrati dai sensori durante una prova di volo reale del drone.

Il segmento di volo scelto è il *test\_volo\_20200218*, e la traiettoria eseguita misurata dai sensori di bordo viene riportata nella figura 5.1.

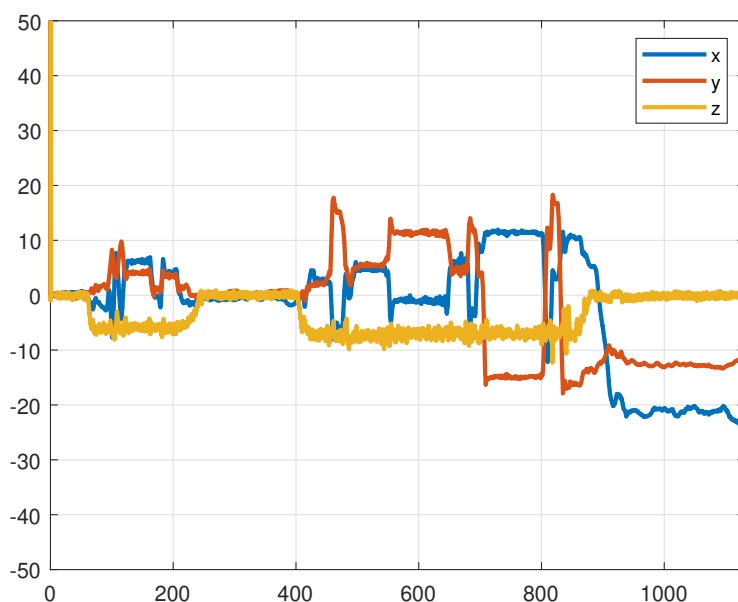


Figura 5.1: Traiettoria misurata

Visti i risultati ottenuti in simulazione, anche sui dati reali si implementa un feedback del bias stimato come schematizzato in figura 4.7. Il filtro di Kalman utilizzato è lo stesso implementato nelle ultime prove in simulazione.

La stima degli stati di navigazione ottenuta con il nostro filtro viene riportata nelle figure seguenti; viene confrontata con le misure disponibili e con le stime ottenute con il filtro descritto nel capitolo 2, senza stima del bias.

Si nota come sui dati reali le due stime siano confrontabili, e solo sull'asse  $z$  si possono notare delle piccole differenze, in particolare nel grafico di velocità.

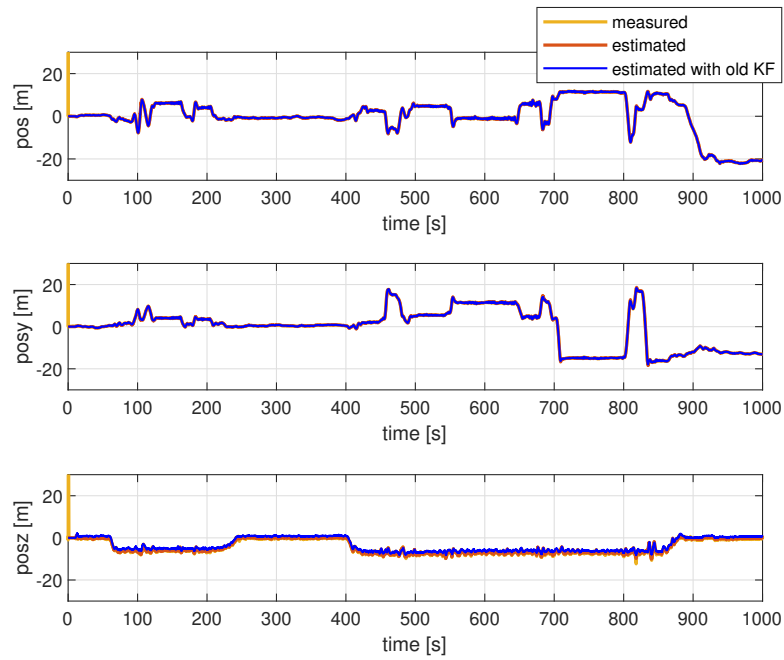


Figura 5.2: Confronto posizione

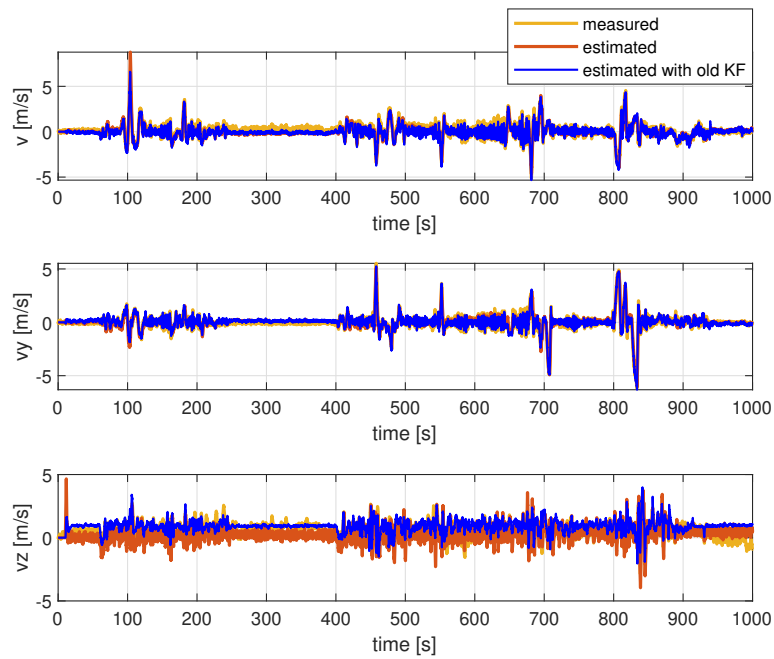


Figura 5.3: Confronto velocità

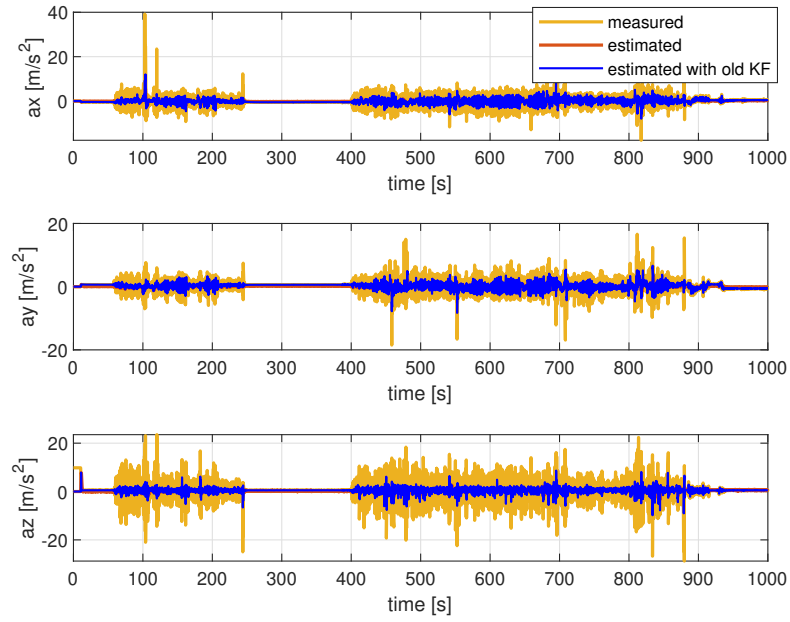


Figura 5.4: Confronto accelerazione

Di seguito viene riportato il confronto di stima degli angoli di Eulero, nel caso di bias retroazionato e non, per valutare gli effetti sulla stima di orientazione.

Come si nota in figura 5.5, la stima degli angoli con la correzione del bias non pare migliorata, anzi, nel caso dell'angolo di roll, presenta degli errori visibili.

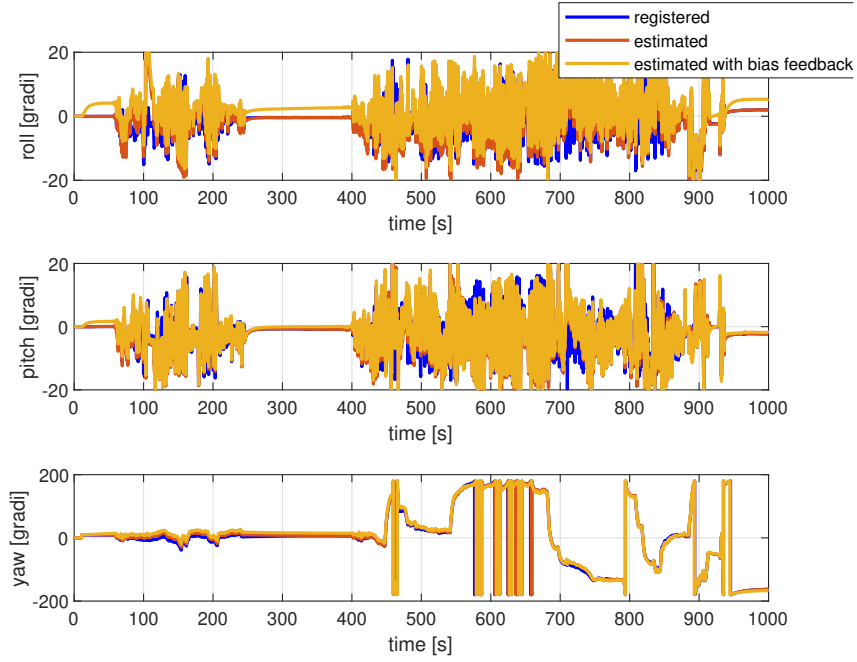


Figura 5.5: Confronto angoli di Eulero

Ciò, può essere spiegato dalla stima del bias ottenuta poi retroazionata.

Infatti, risulta evidente in 5.6 che la stima del bias, non appena inizia il moto intorno ai 50s, si stacca dal valore desiderato, e si allontana in particolare sugli assi  $x$  e  $y$ .

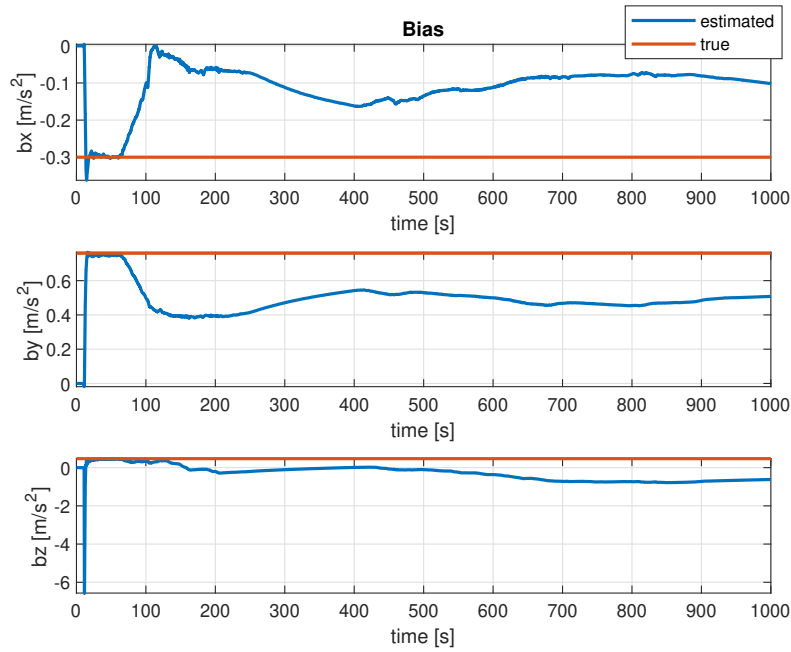


Figura 5.6: Stima bias

Questo problema potrebbe essere legato in parte alla presenza di un bias anche sulle misure di giroscopio. Infatti anche per questo sensore si nota uno scostamento dal valore nullo negli intervalli in cui il drone è fermo (ad es. tra circa *250s* e *400s* come si vede in 5.7. Questo errore si ripercuote direttamente sulla stima di assetto, che a sua volta influisce direttamente la performance del filtro di navigazione.

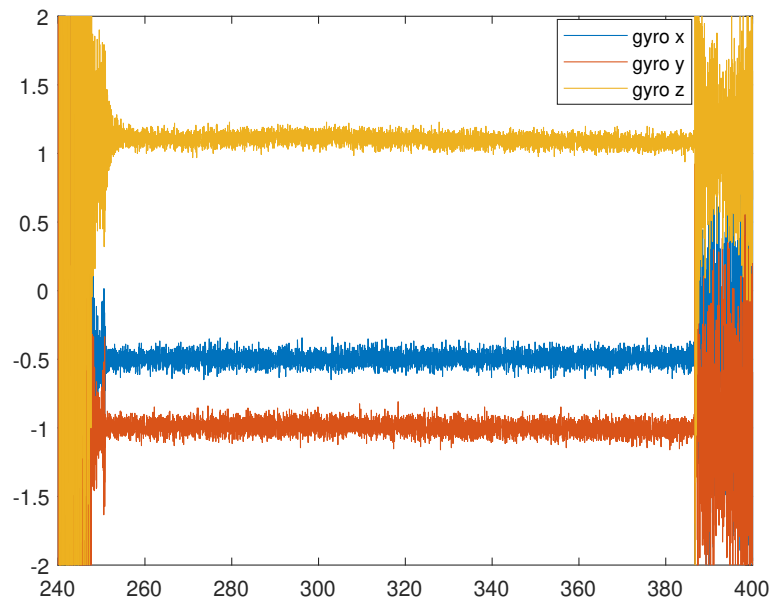


Figura 5.7: Bias giroscopio dettaglio



## 6. Conclusioni

A seguito di queste prove e di queste varie simulazioni, abbiamo notato che la stima del bias converge al valore desiderato solo per il dataset fittizio.

Per quanto riguarda i dati reali, la stima si discosta notevolmente dal valore vero e non migliora la stima di assetto del drone. Ciò potrebbe essere dovuto al fatto che il rumore dei sensori utilizzati è molto grande e influenza negativamente le performance del filtro.

Inoltre, l'assetto calcolato per ruotare le misure in navigation frame è basato su dati sottocampionati a  $50Hz$ , mentre il filtro AHRS funziona a  $200Hz$ ; questo sottocampionamento potrebbe in parte essere di motivo di ulteriori errori sulla stima.

Un possibile sviluppo futuro potrebbe essere quello di inserire anche il bias del giroscopio nel vettore di stima del filtro di Kalman per correggere ulteriormente la stima di assetto.

## A Appendice

### A.1 Background: Filtro di Kalman

Il filtro di Kalman è un'implementazione ricorsiva di algoritmi di stima che si applica al problema della ricostruzione dello stato di un sistema lineare affetto da rumore.

Si consideri quindi il seguente sistema lineare tempo continuo:

$$\begin{cases} \dot{x}(t) = A(t)x(t) + B(t)u(t) + w(t) \\ y(t) = C(t)x(t) + \nu(t) \end{cases} \quad (1)$$

con stato iniziale  $x(t_i) = x_0$  e dove  $w(t)$  e  $\nu(t)$  rappresentano rispettivamente i rumori di processo e misura. Si ipotizzano questi ultimi come processi stocastici Gaussiani bianchi, mutuamente incorrelati e indipendenti tra loro con matrici di covarianza note:

$$\begin{aligned} E[w(t)w(t)^T] &= Q(t) \\ E[\nu(t)\nu(t)^T] &= R(t) \end{aligned} \quad (2)$$

Lo stato iniziale  $x_0$  è anche esso un vettore aleatorio con valor medio  $\bar{x}_0$  e matrice di covarianza  $P_0$ . Si ipotizzi che i processi stocastici  $w(t)$  e  $\nu(t)$  sono a loro volta incorrelati con il vettore  $x_0$ .

Il vettore di stato stimato  $\hat{x}$  risulta anche esso quindi una variabile aleatoria con matrice di covarianza  $P$ . Partendo dall'equazione dinamica dell'osservatore asintotico (Luenberger):

$$\dot{\hat{x}}(t) = A(t)\hat{x}(t) + B(t)u(t) + K(t)[y(t) - C(t)\hat{x}(t)] \quad (3)$$

e definito  $e(t) = x(t) - \hat{x}(t)$  l'errore di stima la sua dinamica sarà quindi rappresentata dal seguente sistema:

$$\begin{cases} \dot{e}(t) = (A(t) - K(t)C(t))e(t) + K(t)\nu(t) + w(t) \\ e(t_0) = e_0 \end{cases} \quad (4)$$

L'obiettivo dell'osservatore **ottimo** consiste nello scegliere  $\hat{x}_0$  e  $K(t)$  in modo da minimizzare l'errore quadratico medio di stima:

$$\min_{e_0, K} E[e(t)e(t)^T] \quad (5)$$

e quindi applicare il funzionale di costo seguente:

$$J = 1/2 \int_{t_i}^{t_f} [e(t)^T Q(t) e(t) + u(t)^T R(t) u(t)] dt \quad (6)$$

Se la componente della matrice  $R(t)$  è definita positiva (ovvero tutte le componenti del vettore di uscita sono affette da rumore), allora la soluzione del problema ottimo è unica:

$$K(t) = P(t)C^T(t)R^{-1}(t) \quad (7)$$

dove  $P(t)$  è la soluzione dell'equazione di Riccati (DRE).

$$\dot{P}(t) = A(t)P(t) + P(t)A^T(t) - P(t)C^T(t)R^{-1}(t)C(t)P(t) + Q(t) \quad (8)$$

La matrice  $P(t)$  rappresenta la varianza dell'errore di stima, e la sua traccia l'errore quadratico medio. Il filtro di Kalman non è altro quindi che un osservatore asintotico il cui guadagno  $K$  (che 'pesa' l'innovazione) è direttamente proporzionale alla varianza della stima dello stato corrente ( $P(t)$ ) e inversamente proporzionale

alla varianza del rumore sulle uscite (come evidente in 7).

Se si considera un sistema TD del tipo:

$$x_{k+1} = A_k x_k + B_k w_k \quad (9)$$

$$y_k = C_k x_k + v_k \quad (10)$$

dove  $w_k$  e  $v_k$  rappresentano il rumore di processo e di misura con matrici di covarianze  $Q_k$  e  $R_k$  rispettivamente. Si può quindi facilmente *proiettare* in avanti la matrice di covarianza della stima:

$$\begin{aligned} P_{k+1} &= E[\tilde{x}_{k+1} \tilde{x}_{k+1}^T] = E[(A_k \tilde{x}_k - B_k w_k)(A_k \tilde{x}_k - B_k w_k)^T] = \\ &= E[A_k \tilde{x}_k \tilde{x}_k^T A_k^T - A_k \tilde{x}_k w_k^T B_k^T - B_k w_k \tilde{x}_k^T A_k^T + B_k^T w_k w_k^T B_k] = A_k P_k A_k^T + B_k Q_k B_k^T \end{aligned} \quad (11)$$

Da questa relazione si può osservare che la covarianza del termine forzante si somma a quella della stima ad ogni passo.

Quando sia disponibile una misura, è possibile aggiornare la stima dello stato sino ad allora disponibile fondendo le informazioni relative in una stima a posteriori.

Ipotizzando di avere al  $k$ -esimo istante una stima dello stato  $\hat{x}_{k|k-1}$  con covarianza  $P_{k|k-1}$  e di misurare quindi  $y_k = C_k x_k + v_k$ .

Distinguiamo quindi due passi nella stima di  $x_k$ , una prima stima  $x_{k|k+1}$  costruita conoscendo le misure sino a  $y_{k-1}$ , fase di **predizione**, ed una seconda  $\hat{x}_{k|k}$  che utilizza anche la misura  $y_k$ , fase di **correzione**.

Viste le considerazioni precedenti otteniamo quindi le seguenti equazioni dell'algoritmo in tempo discreto:

$$x_{k+1|k} = A_k \hat{x}_k \quad (12)$$

$$P_{k+1|k} = A_k P_k A_k^T + B_k Q_k B_k^T \quad (13)$$

e l'innovazione diventa:

$$P_{k+1|k+1} = P_{k+1|k} + C_{k+1}^T R_{k+1}^{-1} C_{k+1} \quad (14)$$

$$x_{k+1|k+1} = x_{k+1|k} + P_{k+1|k+1} C_{k+1}^T R_{k+1}^{-1} (y_{k+1} - C_{k+1} x_{k+1|k}) \quad (15)$$

Il guadagno di Kalman altro non è che  $K_k = P_{k|k} C_k^T R_k^{-1}$ , ed è fondamentalmente un rapporto tra l'incertezza nella stima dello stato  $P$  e l'incertezza della misura  $R$ .

Assumendo costanti le matrici di covarianza del rumore e del modello si può riscrivere in forma compatta come:

$$x_{k+1} = A \hat{x}_k + A K_k (y_k - C \hat{x}_k) \quad (16)$$

$$\hat{y}_k = C \hat{x}_k + C K_k (y_k - C \hat{x}_k) \quad (17)$$

dove  $K_k$  è il guadagno di Kalman e la stima della matrice di covarianza dell'errore di stima è:

$$P_{k+1}^{-1} = (A P_k A^T + B Q B^T)^{-1} + C^T R^{-1} C \quad (18)$$

La variabile  $\hat{y}_k$  è la versione filtrata dell'uscita, utile per sopprimere gli effetti dei disturbi sulle uscite misurate.

### Parametri di Tuning

Nel filtro di Kalman si devono fornire due matrici di covarianza, rispettivamente sul rumore d'uscita e quello dello stato. La matrice  $R_k$  ha spesso una chiara interpretazione fisica, rappresenta il rumore dei sensori

associati alla uscita. Nel caso di più uscite incorrelate tra loro,  $R_k$  è una matrice diagonale contenete le covarianze di rumore dei differenti sensori.

La stessa cosa non può dirsi per  $Q_k$ : il filtro di Kalman viene spesso utilizzato per stimare lo stato che quindi non è misurabile. Una scelta di  $Q_k$  con valori molto bassi, supposto il sistema stabile, significa dire al filtro che il modello matematico è perfetto, il risultato è che il filtro considera come stima ottima l'evoluzione del suo stato senza considerare la misura delle uscite che sono affette da rumore.

A valle di queste considerazioni appare evidente come sia necessario fornire una stima, spesso euristica, della matrice  $Q_k$ . Una buona scelta, in assenza di informazioni ulteriori, è quella di prendere una matrice diagonale definita positiva; il valore degli elementi, poi, va opportunamente scelto. Il valore relativo tra  $R_k$  e  $Q_k$  ha quindi un significato ben preciso, fornendo un alto rapporto  $Q_k$  vs  $R_k$  ( in particolare  $\|Q_k\| \gg \|R_k\|$  significa tarare il filtro in maniera tale da dare maggior peso alle misure rispetto alla evoluzione dello stato.

La matrice  $P_k$  è effettivamente una misura della covarianza della stima dello stato solo se le matrici di covarianza dei rumori sono state ottenute da considerazioni fisiche.