



# UNIVERSITÀ DI PISA

INGEGNERIA ROBOTICA E DELL'AUTOMAZIONE

---

Sistemi di Guida e Navigazione

## A MATLAB Implementation of: “A GPS-aided Inertial Navigation System in Direct Configuration”

---

*Studente:*  
Marco Borraccino  
Student ID: 501794

*Professore:*  
Pollini Lorenzo

2021-12-27

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Descrizione del Progetto</b>	<b>2</b>
2.1	Vettore di stato e specifiche del sistema . . . . .	2
2.2	Architettura del sistema . . . . .	3
2.2.1	Passo di Predizione . . . . .	3
2.2.2	Passo di Correzione . . . . .	5
2.2.3	Inizializzazione . . . . .	7
<b>3</b>	<b>Dettagli e Implementazione dell'EKF</b>	<b>8</b>
3.1	Blocco: Initialization task . . . . .	8
3.2	Blocco: System Prediction . . . . .	10
3.3	Blocco: System Update . . . . .	11
3.3.1	Blocco: flag_updates . . . . .	12
3.3.2	Funzione: virtual . . . . .	14
3.3.3	Funzione: correction . . . . .	14
3.4	Funzioni ausiliarie . . . . .	14
3.4.1	Funzione: ECEF2NED . . . . .	14
3.4.2	Funzione: extract_var . . . . .	15
<b>4</b>	<b>The Malaga 2009 Dataset</b>	<b>16</b>
4.1	Adattamento del dataset per la simulazione . . . . .	17
4.2	Grafici delle misure utilizzate . . . . .	19
<b>5</b>	<b>Parametri della Simulazione</b>	<b>22</b>
5.1	Parametri variabili . . . . .	23
5.2	Parametri assegnati empiricamente . . . . .	23
5.3	Parametri costanti . . . . .	24
<b>6</b>	<b>Risultati Sperimentali</b>	<b>25</b>
6.1	Simulazione con GPS RTK con interruzioni . . . . .	27
6.2	Simulazione con DGPS low-cost con interruzioni . . . . .	35
6.3	Simulazione con GPS RTK . . . . .	43
6.4	Simulazione con DGPS low-cost . . . . .	51
6.5	Considerazioni sul parametro <i>zero_vel_period</i> . . . . .	59
6.6	Conclusioni . . . . .	62
<b>7</b>	<b>Simulazione con dati di un drone</b>	<b>63</b>
7.1	Risultati della simulazione . . . . .	66
7.2	Risultati della simulazione con parametri aggiornati . . . . .	75

# 1 Introduzione

Questo lavoro presenta un'implementazione su Matlab dell'articolo scientifico: “*R. Munguía, A GPS-aided inertial navigation system in direct configuration, Journal of Applied Research and Technology, Volume 12, Issue 4, 2014, Pages 803-814, ISSN 1665-6423*”,[1]. L'articolo in questione propone un metodo pratico per la stima della cinematica di un veicolo, con un modello dell'errore dei sensori, attraverso l'integrazione di misure inerziali e del GPS. L'architettura è basata su un Extended Kalman Filter (EKF) in configurazione diretta. I parametri e le variabili stimate sono: l'assetto del veicolo, la velocità angolare, la posizione, la velocità lineare, l'accelerazione, il bias di accelerometro e giroscopio. Dopo un preliminare studio del problema, utilizzando Matlab e Simulink si è cercato di simulare in ogni dettaglio l'architettura del filtro e replicare fedelmente i risultati ottenuti sperimentalmente.

Nella sez. *Descrizione del Progetto* 2 si trova una panoramica del metodo proposto e una struttura di base del filtro in questione.

Procedendo, nella sez. *Dettagli e Implementazione dell'EKF* 3 è affrontata l'implementazione del metodo con riferimenti a formule e file usati per la simulazione.

La sez. *The Malaga 2009 Dataset* 4 descrive nel dettaglio il dataset utilizzato per la validazione del modello, quali dati sono disponibili e come sono stati adattati per la simulazione.

La sez. *Parametri della Simulazione* 5 espone tutti i parametri del filtro EKF, dai parametri variabili ad ogni simulazione a quelli costanti del sistema, cercando di motivare ogni loro valore.

Nella sez. *Risultati Sperimentali* 6 sono presenti i risultati di quattro simulazioni principali con grafici ed errori della stima dello stato del filtro rispetto a un riferimento dato, e ulteriori considerazioni al variare di alcuni parametri.

Infine nella sez. *Simulazione con dati di un drone* 7 si presentano i risultati utilizzando un ulteriore dataset collezionato durante il volo di un drone, nel corso di una sperimentazione dell'Università di Pisa.

## 2 Descrizione del Progetto

L'*Extended Kalman Filter (EKF)* in configurazione diretta è tutt'oggi uno dei metodi più utilizzati per la navigazione di veicoli sulla superficie terrestre. Combina le misure inerziali con quelle del GPS per fornire una stima di assetto e posizione migliore di quella che si otterrebbe con tali sensori singolarmente. Il caso in esame analizza una sua implementazione generale, adatta per essere modificata in base ai sensori disponibili e ai vincoli dinamici del veicolo. Vediamo nel dettaglio la struttura del filtro proposto in [1].

### 2.1 Vettore di stato e specifiche del sistema

L'obiettivo è stimare nel migliore dei modi il seguente vettore di stato:

$$\hat{x} = [q^{nb} \quad \omega^b \quad x_g \quad r^n \quad v^n \quad a^n \quad x_b] \quad (2.1)$$

dove:

- $q^{nb} = [q_0, q_1, q_2, q_3]$  è il quaternione unitario che rappresenta l'assetto (*yaw, pitch, roll*);
- $\omega^b = [\omega_x, \omega_y, \omega_z]$  è la velocità angolare con la compensazione del bias, espressa in body frame;
- $x_g = [x_{g_x}, x_{g_y}, x_{g_z}]$  è il bias del giroscopio;
- $r^n = [x, y, z]$  rappresenta le coordinate dell'origine del body frame espresse nel navigation frame;
- $v^n = [v_x, v_y, v_z]$  è la velocità del veicolo espressa in navigation frame;
- $a^n = [a_x, a_y, a_z]$  è l'accelerazione con la compensazione del bias, espressa in navigation frame;
- $x_a = [x_{a_x}, x_{a_y}, x_{a_z}]$  è il bias dell'accelerometro.

Il navigation frame usato è il NED (*North-East-Down*), comune per la maggior parte dei metodi usati per la navigazione di veicoli autonomi sulla Terra (Fig. 2.1).

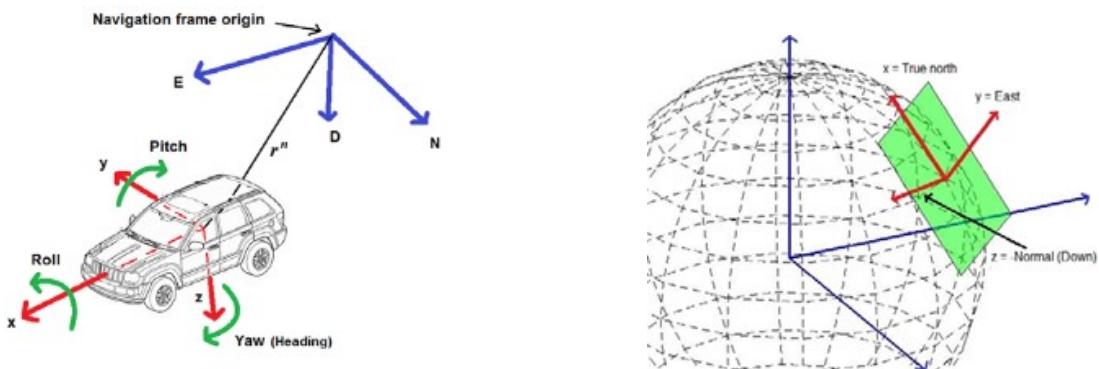


Figura 2.1: La figura mostra le relazioni che sussistono con la scelta del NED come navigation frame

La stima dello stato è ottenuta avendo a disposizione due tipologie di misure:

- *High Rate Measurement*

Le misure ad alta frequenza, tipicamente  $100 - 200\text{Hz}$ , sono fornite da un'unità inerziale IMU formata da: un giroscopio triassiale, un accelerometro triassiale e un magnetometro triassiale. Il magnetometro spesso è utilizzato per la prima fase di inizializzazione del filtro, per poi sfruttare le misure ogni istante  $k$  di giroscopio e accelerometro. In dettaglio, la misura  $y_g$  del giroscopio può essere modellata come:

$$y_g = \omega^b + x_g + v_g \quad (2.2)$$

dove  $x_g$  è un rumore additivo (bias) e  $v_g$  è un rumore bianco gaussiano con varianza pari a  $\sigma_g^2$ . Similmente il modello di misura dell'accelerometro è:

$$y_a = a^b - g^b + x_a + v_a \quad (2.3)$$

con  $x_a$  rumore additivo (bias),  $g^b$  è il vettore gravità espresso in body frame,  $v_a$  rumore bianco gaussiano con varianza  $\sigma_a^2$ .

- *Low Rate Measurement*

Le misure a bassa frequenza,  $1 - 4\text{Hz}$ , sono da associare a misure di GPS di posizione  $z_r$  e orientazione  $z_\theta$  esprimibili come:

$$\begin{bmatrix} z_r^n \\ z_\theta^n \end{bmatrix} = \begin{bmatrix} r^n + v_r \\ \theta^n + v_\theta \end{bmatrix} \quad (2.4)$$

dove  $r^n = [x, y, z]$  rappresenta la posizione dell'antenna GPS in navigation frame e  $v_r$  è un rumore bianco gaussiano con varianza  $\sigma_r^2$  legato all'accuratezza del sensore; invece  $\theta^n$  è l'angolo di traiettoria misurato dal GPS, con  $v_\theta$  rumore bianco gaussiano con varianza  $\sigma_\theta^2$ . In questo senso le misure del GPS si intendono anticipatamente trasformate dalle coordinate geodetiche in navigation frame.

## 2.2 Architettura del sistema

L'architettura del sistema si basa sulla tipica struttura di un Kalman Filter caratterizzata dai passi di predizione e correzione (Fig. 2.2). Il passo di predizione sfrutta le misure inerziali ad alta frequenza per aggiornare lo stato nel corto termine. Il passo di correzione grazie alle misure di GPS e a vincoli dinamici corregge la stima dello stato e ne migliora l'accuratezza nel lungo termine. Vediamo nel dettaglio ognuna di queste fasi.

### 2.2.1 Passo di Predizione

La frequenza di aggiornamento del sistema è posta pari a  $100\text{Hz}(\Delta t = 0,01\text{s})$ , ovvero pari alla frequenza di arrivo delle misure di giroscopio e accelerometro. Ad ogni periodo quindi il passo di predizione avviene in base al seguente modello non lineare:

$$\begin{aligned} \text{Attitude} & \left\{ \begin{array}{l} q_{(k+1)}^{nb} = q_{(k)}^{nb} * q([\omega_{(k+1)}^b \Delta t]') \\ \omega_{(k+1)}^b = y_g(k) - x_g(k) \\ x_{g(k+1)} = (1 - \lambda_{xg} \Delta t) x_{g(k)} \end{array} \right. & \text{Position} & \left\{ \begin{array}{l} r_{(k+1)}^n = r_{(k)}^n + (v_{(k+1)}^n \Delta t) + (a_{(k+1)}^n (\Delta t^2)/2) \\ v_{(k+1)}^n = v_{(k)}^n + (a_{(k+1)}^n \Delta t) \\ a_{(k+1)}^n = R^{bn}(y_a(k) - x_a(k)) + g \\ x_{a(k+1)} = (1 - \lambda_{xa} \Delta t) x_{a(k)} \end{array} \right. \end{aligned} \quad (2.5)$$

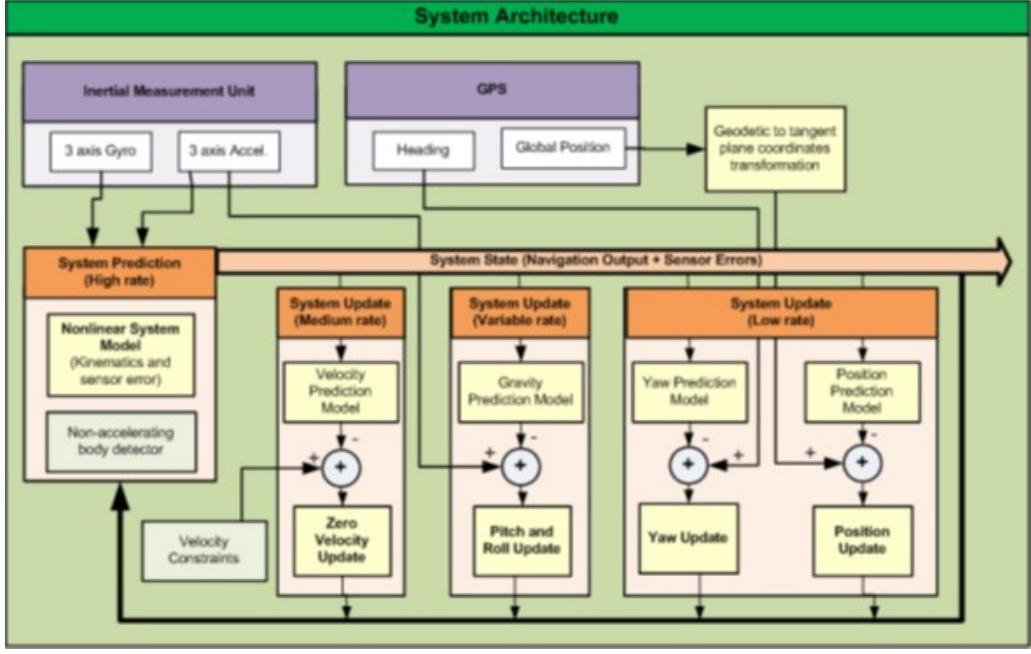


Figura 2.2: Architettura generale del filtro

in cui  $q([\omega_{(k+1)}^b \Delta t]')$  è il quaternione creato dal vettore di rotazione  $[\omega_{(k+1)}^b \Delta t]'$ . La matrice di rotazione da body a navigation frame  $R^{bn}$  è ottenuta dal quaternione di orientazione  $q_{(k+1)}^{nb}$  (le formule esplicite di questi passaggi e trasformazioni sono riportate in seguito nella sezione *Blocco: System Prediction 3.2*). I parametri  $\lambda_{xg}$  e  $\lambda_{xa}$  indicano invece quanto velocemente i bias di accelerometro e giroscopio variano nel tempo. Il modello completo per propagare l'errore del bias è:

$$bias_{(k+1)} = (1 - \lambda \Delta t) bias_{(k)} + v_b \quad (2.6)$$

con  $v_b$  rumore bianco gaussiano. Le incertezze del bias di accelerometro e giroscopio, cioè  $v_{xa}(\sigma_{xa}^2)$  e  $v_{xg}(\sigma_{xg}^2)$ , influenzano la matrice di covarianza della stima  $P$  attraverso la matrice di covarianza degli errori  $U$ . In dettaglio:

$$P_{(k+1)} = \nabla F_x P_{(k)} \nabla F'_x + \nabla F_u U \nabla F'_u \quad (2.7)$$

- $U$  è la matrice di covarianza degli errori in ingresso:

$$U = diag[\sigma_g^2 I_{3 \times 3} \quad \sigma_{xg}^2 I_{3 \times 3} \quad \sigma_a^2 I_{3 \times 3} \quad \sigma_{xa}^2 I_{3 \times 3}] \quad (2.8)$$

- $\nabla F_x = \frac{\partial F}{\partial \hat{x}}$  è il Jacobiano del modello di predizione non lineare  $F$  rispetto lo stato  $\hat{x}$ :

$$\nabla F_x = \frac{\partial F}{\partial \hat{x}} = \begin{bmatrix} \frac{\partial f q^{nb}}{\partial q^{nb}} & \frac{\partial f q^{nb}}{\partial \omega^b} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial f \omega^b}{\partial x_g} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial f x_g}{\partial x_g} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial f r^n}{\partial r^n} & \frac{\partial f r^n}{\partial v^n} & \frac{\partial f r^n}{\partial a^n} & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial f v^n}{\partial v^n} & \frac{\partial f v^n}{\partial a^n} & 0 \\ \frac{\partial f a^n}{\partial q^{nb}} & 0 & 0 & 0 & 0 & 0 & \frac{\partial f a^n}{\partial x_a} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial f x_a}{\partial x_a} \end{bmatrix} \quad (2.9)$$

- $\nabla F_u = \frac{\partial F}{\partial u}$  è il Jacobiano del modello di predizione non lineare  $F$  rispetto gli ingressi al sistema  $u$ :

$$\nabla F_u = \frac{\partial F}{\partial u} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{\partial f\omega^b}{\partial y_g} & 0 & 0 & 0 \\ 0 & \frac{\partial fx_g}{\partial v_{xg}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial fa^n}{\partial y_a} & 0 \\ 0 & 0 & 0 & \frac{\partial fx_a}{\partial v_{xa}} \end{bmatrix} \quad (2.10)$$

Dal calcolo della matrice  $\nabla F_x$  si nota che il termine  $\frac{\partial fa^n}{\partial q^{nb}} \neq 0$  implica un accoppiamento diretto tra assetto e posizione: si può assumere un asse dell'unità inerziale, solitamente l'asse x, allineato in modo fisso alla direzione di spostamento del veicolo e utilizzare quindi le misure del GPS per correggere l'assetto. Tale implicazione non è consentita per esempio con un veicolo aereo dato che l'angolo di attacco non corrisponde sempre con quello di traiettoria.

### 2.2.2 Passo di Correzione

Il sistema invece è aggiornato nel passo di correzione in base alle seguenti formule:

$$e_{(k)} = z_{(k)} - h_{(k)} \quad (2.11)$$

$$S_{(k)} = \nabla H_{(k)} P_{(k)} \nabla H'_{(k)} + R_k \quad (2.12)$$

$$K_{(k)} = P_{(k)} \nabla H'_{(k)} S_{(k)}^{-1} \quad (2.13)$$

$$\hat{x}_{(k+1)} = \hat{x}_{(k)} + K_{(k)} e_{(k)} \quad (2.14)$$

$$P_{(k+1)} = (I_{22 \times 22} - K_{(k)} H_{(k)}) P_{(k)} \quad (2.15)$$

dove:

- $e_{(k)}$  è l'innovazione ed è la differenza tra le misure  $z_{(k)}$  e lo loro predizione  $h_{(k)}$ ;
- la matrice  $R_{(k)}$  è la matrice di covarianza degli errori di misura;
- $S_{(k)}$  è la matrice di covarianza dell'innovazione;
- $K_{(k)}$  è il guadagno di Kalman;
- $\nabla H_{(k)}$  è il Jacobiano ottenuto dal modello di predizione delle misure  $h_{(k)}(\hat{x}_{(k)})$  rispetto al vettore di stato  $\hat{x}_{(k)}$ ;
- $P_{(k)}$  è la matrice di covarianza della stima.

Con frequenza diversa verranno eseguite tre diverse tipologie di correzione, come si evince dalla struttura del filtro (Fig. 2.2), che andranno a comporre i vettori e le matrici appena descritte.

### 1. Dynamical Constraints Update

Conoscendo il modello del veicolo si possono inserire i vincoli dinamici nel passo di correzione. In questo progetto sono stati utilizzati dei dati sperimentali di sensori montati su un autoveicolo, per cui, assunto che l'asse x dell'IMU sia allineato lungo la direzione di spostamento, la velocità in body frame degli assi y e z, cioè  $v_y^b$  e  $v_z^b$ , possono essere modellate come zero più un rumore additivo gaussiano  $v_v$  (di varianza  $\sigma_v^2$ ).

$$\begin{bmatrix} v_y^b \\ v_z^b \end{bmatrix} = \begin{bmatrix} 0 + v_v \\ 0 + v_v \end{bmatrix} \quad (2.16)$$

La misura predetta  $h_{v0}$  è quindi:

$$h_{v0} = [v_y^b \quad v_z^b]' \quad [v_x^b \quad v_y^b \quad v_z^b] = R^{nb}V^n \quad (2.17)$$

dove  $R^{nb}$  è la matrice di rotazione navigation to body calcolata a partire dal quaternione di orientazione  $q^{nb}$ , e  $V^n$  è la velocità in navigation frame prelevata dal vettore di stato.

Per il passo di correzione utilizzeremo quindi:

$$z_{v0} = [0 \quad 0]' \quad h_{v0} \quad R_{v0} = \sigma_v^2 I_{2 \times 2} \quad \nabla H_{v0} = \frac{\partial h_{v0}}{\partial \hat{x}} \quad (2.18)$$

### 2. Roll and Pitch Updates

Se il veicolo è assunto non accelerante, cioè  $a^b \approx 0$ , allora possiamo supporre che l'accelerometro fornisca una misura rumorosa del vettore gravità,  $y_a \approx g^b + v_a$ , al netto del bias  $x_a$ . Da tale misura possiamo correggere gli angoli *roll* e *pitch* dell'assetto del veicolo. Per valutare istante per istante se il veicolo è non accelerante, si usa il Stance Hypothesis Optimal Detector (SHOE), usato in [2]. I dettagli implementativi saranno esposti in *Blocco: System Update 3.3*. Il vettore gravità predetto dal modello delle misure sarà quindi:

$$h_g = R^{nb}[0 \quad 0 \quad g_c]' \quad (2.19)$$

dove  $g_c$  è la costante di gravità e  $R^{nb}$  è la matrice di rotazione navigation to body calcolata dal quaternione  $q^{nb}$ . Per il passo di correzione utilizzeremo quindi:

$$z_g = y_a \quad h_g \quad R_g = \sigma_a^2 I_{3 \times 3} \quad \nabla H_g = \frac{\partial h_g}{\partial \hat{x}} \quad (2.20)$$

### 3. Position and Heading Updates

Le misure del GPS sono direttamente utilizzate per correggere la posizione stimata. Quindi semplicemente:

$$h_r = r^n \quad (2.21)$$

Per migliorare la stima dello *yaw* invece, si assume che l'angolo di heading del veicolo sia fisso rispetto a quello misurato dal GPS. Se tale ipotesi è valida, il modello usato per predire la misura di heading del GPS è:

$$h_\theta = \text{atan2}(2(q_2 q_3 - q_1 q_4), \quad 1 - 2(q_3^2 + q_4^2)) \quad (2.22)$$

dove  $q^{nb} = [q_1, q_2, q_3, q_4]$  è il quaternione di orientazione corrente.

Per il passo di correzione utilizzeremo quindi:

$$z_r = z_r^n \quad h_r \quad R_r = \sigma_r^2 I_{3 \times 3} \quad \nabla H_r = \frac{\partial h_r}{\partial \hat{x}} \quad (2.23)$$

$$z_\theta = z_\theta^n \quad h_\theta \quad R_\theta = \sigma_\theta^2 \quad \nabla H_\theta = \frac{\partial h_\theta}{\partial \hat{x}} \quad (2.24)$$

Componendo quindi i vettori e le matrici con i vari tipi di correzione (eq.ni 2.18,2.20,2.23 e 2.24 ), otteniamo:

$$e_{(k)} = z_{(k)} - h_{(k)} = \begin{bmatrix} 0 \\ 0 \\ y_{a_x} \\ y_{a_y} \\ y_{a_z} \\ z_{r_x}^n \\ z_{r_y}^n \\ z_{r_z}^n \\ z_\theta^n \end{bmatrix} - \begin{bmatrix} h_{v0_y} \\ h_{v0_z} \\ h_{g_x} \\ h_{g_y} \\ h_{g_z} \\ h_{r_x} \\ h_{r_y} \\ h_{r_z} \\ h_{r_\theta} \end{bmatrix} \quad R_{(k)} = diag[\sigma_v^2 I_{2 \times 2} \quad \sigma_a^2 I_{3 \times 3} \quad \sigma_r^2 I_{3 \times 3} \quad \sigma_\theta^2] \quad (2.25)$$

$$\nabla H_{(k)} = \frac{\partial h_{(k)}}{\partial \hat{x}_{(k)}} = \begin{bmatrix} \frac{\partial h_{v0}}{\partial q^{nb}} & 0 & 0 & 0 & \frac{\partial h_{v0}}{\partial r^n} & 0 & 0 \\ \frac{\partial h_g}{\partial q^{nb}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{3 \times 3} & 0 & 0 \\ \frac{\partial h_r}{\partial q^{nb}} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.26)$$

### 2.2.3 Inizializzazione

Il sistema è inizializzato come segue:

$$\hat{x} = [q_{ini}^{nb} \quad 0_{1 \times 18}] \quad P_{ini} = diag[(\sigma(q_{ini}^{nb})^2 \quad \varepsilon_{1 \times 18})] \quad (2.27)$$

Per quanto riguarda lo stato iniziale, l'assetto del veicolo è stimato dalle prime misure dell'unità inerziale e la matrice di covarianza della stima  $P_{ini}$  è inizializzata con un rumore associato all'assetto, come specificato in [3] e in seguito nella sezione *Blocco: Initialization task 3.1*. I restanti valori  $\varepsilon$  di  $P_{ini}$  sono ragionevolmente piccoli. Tuttavia in simulazione queste definizioni saranno leggermente diverse dal metodo proposto inizialmente dall'articolo [1].

### 3 Dettagli e Implementazione dell'EKF

Sulla base quindi del metodo proposto in [1], la realizzazione di una simulazione Matlab & Simulink permette di valutarne le prestazioni. Lo schema (Fig. 3.1) estratto dal file `EKF_sim.slx`, prende spunto dall'architettura comune del modello (Fig. 2.2) con l'aggiunta di blocchi e funzioni ausiliarie.

*In questa sezione verranno esplicitati tutti i valori noti e costanti e spiegati i dettagli implementativi di ogni sezione, in riferimento ad articoli in bibliografia e script Matlab allegati alla relazione.*

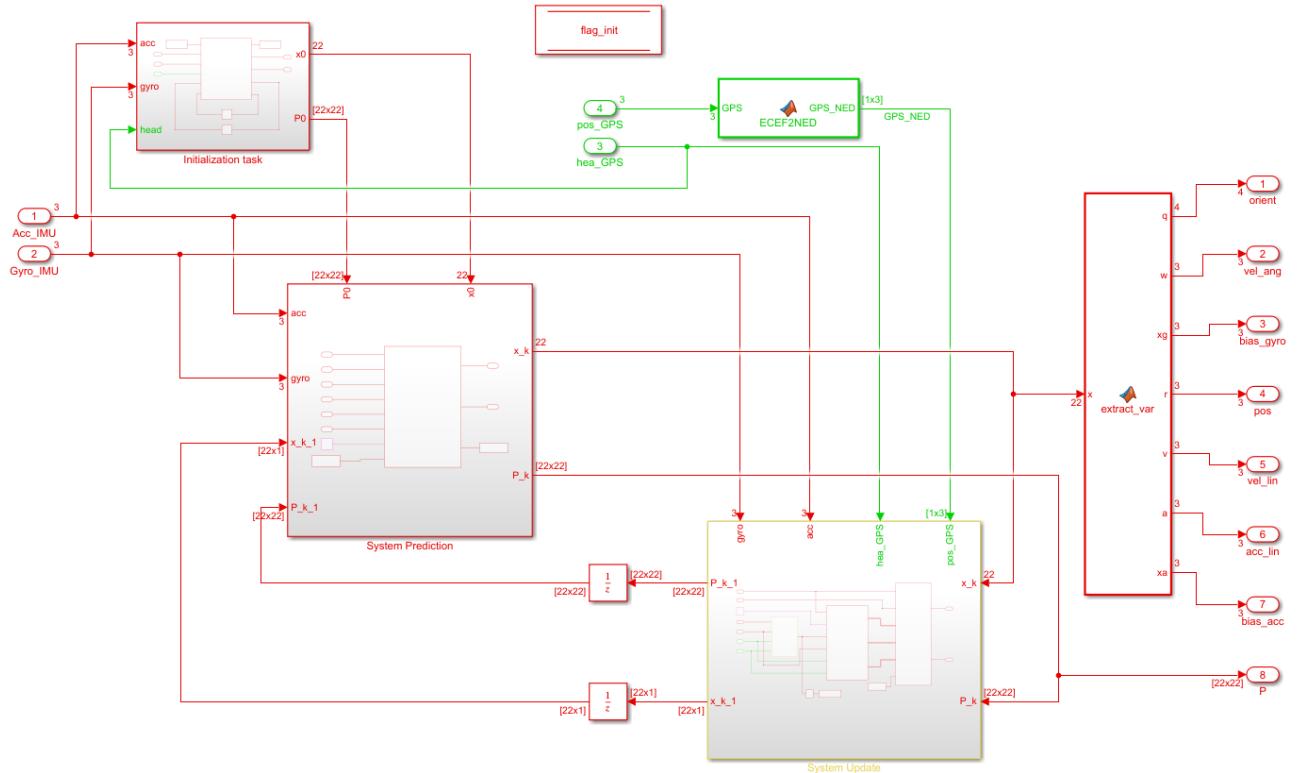


Figura 3.1: Schema Simulink Completo EKF

#### 3.1 Blocco: Initialization task

Nel blocco di inizializzazione (Fig. 3.2), composto dalla funzione `EKF_init`, ci sono le funzioni per assegnare i valori iniziali dello stato stimato  $\hat{x}_{ini}$  e matrice  $P_{ini}$  all'inizio della simulazione. Sulla base del metodo proposto in [3], per un intervallo di tempo definito (scelto pari a  $T = 4s$ ) il veicolo viene assunto non accelerante e si salvano le misure dell'accelerometro, giroscopio e dell'heading del GPS in memoria. Dalle misure dell'accelerometro si stima il vettore gravità  $g^b = [g_1, g_2, g_3]'$ :

$$g^b = \frac{1}{T} \int_0^T -y_a(t) dt \rightarrow g^b = \frac{1}{T} \sum_{t=0}^T -y_a(t) \quad (3.1)$$

dal quale si possono ricavare *roll*  $\phi$  e *pitch*  $\theta$  come:

$$\phi_{ini} = \text{atan2}(g_2, g_3) \quad \theta_{ini} = \text{atan2}(-g_1, \sqrt{(g_2)^2 + (g_3)^2}) \quad (3.2)$$

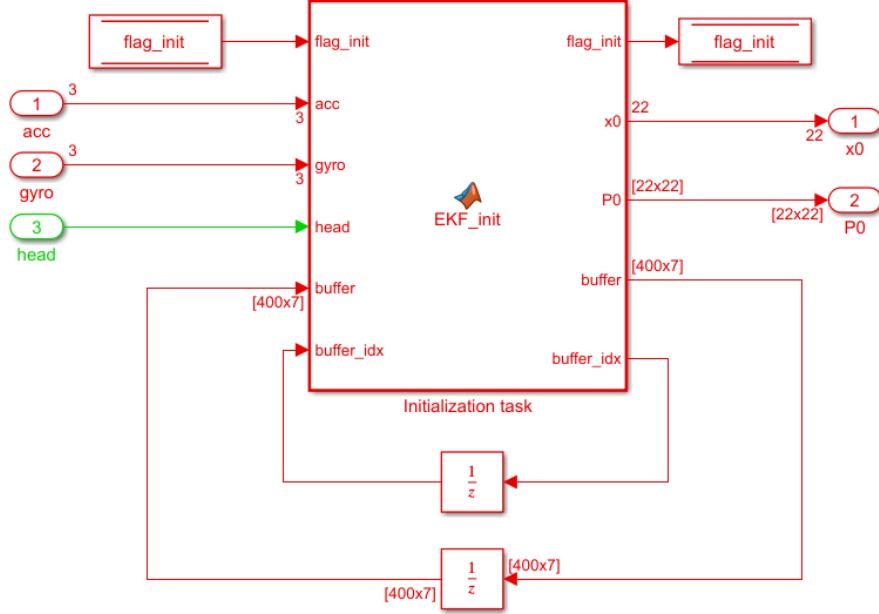


Figura 3.2: Blocco di inizializzazione

Per il calcolo dello  $yaw \psi$  iniziale è proposto un metodo basato sulla disponibilità di misure del magnetometro. In questa simulazione queste non sono state reperibili, per cui il valore iniziale dell'angolo di  $yaw$  viene o assunto noto, o stimato come una media delle prime misure dell'heading del GPS.

$$\psi_{ini} = \frac{1}{T} \sum_{t=0}^T z_\theta(t) \quad (3.3)$$

Noti gli angoli di Eulero iniziali, si calcola la matrice di rotazione navigation-to-body  $R^{nb}$  e da essa il quaternione iniziale corrispondente  $q_{ini}^{nb}$  [vedi Appendice di [3]]:

$$R^{nb} = \begin{bmatrix} c(\psi)c(\theta) & s(\psi)c(\theta) & -s(\theta) \\ -s(\psi)c(\phi) + c(\psi)s(\theta)s(\phi) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & c(\theta)s(\phi) \\ s(\psi)s(\phi) + c(\psi)s(\theta)c(\phi) & -c(\psi)s(\phi) + s(\psi)s(\theta)c(\phi) & c(\theta)c(\phi) \end{bmatrix} \quad (3.4)$$

$$q_1 = \sqrt{1 + R_{(1,1)}^{nb} + R_{(2,2)}^{nb} + R_{(3,3)}^{nb}} \quad q_{ini}^{nb} = \begin{bmatrix} q_1 \\ \frac{R_{(3,2)}^{nb} - R_{(2,3)}^{nb}}{4q_1} \\ \frac{R_{(1,3)}^{nb} - R_{(3,1)}^{nb}}{4q_1} \\ \frac{R_{(2,1)}^{nb} - R_{(1,2)}^{nb}}{4q_1} \end{bmatrix} \quad (3.5)$$

o equivalentemente a 3.4 e 3.5, con [4] [5]:

$$q([\psi_{yaw}, \theta_{pitch}, \phi_{roll}]) = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} c(\phi/2)c(\theta/2)c(\psi/2) + s(\phi/2)s(\theta/2)s(\psi/2) \\ s(\phi/2)c(\theta/2)c(\psi/2) - c(\phi/2)s(\theta/2)s(\psi/2) \\ c(\phi/2)s(\theta/2)c(\psi/2) + s(\phi/2)c(\theta/2)s(\psi/2) \\ c(\phi/2)c(\theta/2)s(\psi/2) - s(\phi/2)s(\theta/2)c(\psi/2) \end{bmatrix} \quad (3.6)$$

con  $s(\alpha) = \sin(\alpha)$  e  $c(\alpha) = \cos(\alpha)$ .

Diversamente, anche se nel lavoro proposto solo l'assetto ha un valore diverso da zero dopo l'inizializzazione, si è scelto di utilizzare comunque le prime misure del giroscopio per dare una stima iniziale del bias dello stesso. Anche in questo caso la media temporale delle prime misure ci fornisce il valore del bias  $x_{gini}$ :

$$x_{gini} = \frac{1}{T} \sum_{t=0}^T y_g(t) \quad (3.7)$$

La stima dello stato è per cui inizializzata come:

$$\hat{x} = [q_{ini}^{nb} \quad 0_{1 \times 3} \quad x_{gini} \quad 0_{1 \times 12}] \quad (3.8)$$

La matrice di covarianza  $P_{ini}$  invece è:

$$P_{ini} = \begin{bmatrix} P(q_{ini}^{nb})_{4 \times 4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & (\sigma_g^2)_{3 \times 3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & (10^{-7})_{3 \times 3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & (1)_{3 \times 3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (0.1)_{3 \times 3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & (\sigma_a^2)_{3 \times 3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & (10^{-3})_{3 \times 3} \end{bmatrix} \quad (3.9)$$

dove  $P(q_{ini}^{nb})$  è calcolata come:

$$P(q_{ini}^{nb}) = \nabla q \left[ \frac{\sigma_a^2}{T} I_{3 \times 3} \right] \nabla q' \quad (3.10)$$

con  $\nabla q$  il Jacobiano della trasformazione completa da angoli di Eulero a quaternione corrispondente (eq. 3.6), rispetto gli angoli di Eulero (si assume che la misura dello yaw, sia stata ricavata dalle misure dell'accelerometro così come gli altri angoli). I calcoli simbolici esplicativi di questa matrice sono presenti nel file `calcolo_matrice_Pq0.m`. Gli altri valori della matrice  $P_{ini}$  sono stati scelti arbitrariamente piccoli.

## 3.2 Blocco: System Prediction

Il blocco di predizione (Fig. 3.3), composto dalla sola funzione `prediction_step`, si occupa di propagare la stima dello stato  $\hat{x}$  e la matrice di covarianza della stima  $P$  sulla base delle misure inerziali di accelerometro e giroscopio. Tali operazioni sono eseguite a partire dalla fine della fase di inizializzazione. Le equazioni di predizione della stima dello stato (eq. 2.5) sono:

$$\text{Attitude} \left\{ \begin{array}{l} q_{(k+1)}^{nb} = q_{(k)}^{nb} * q([\omega_{(k+1)}^b \Delta t]') \\ \omega_{(k+1)}^b = y_g(k) - x_g(k) \\ x_{g(k+1)} = (1 - \lambda_{xg} \Delta t) x_{g(k)} \end{array} \right. \quad \text{Position} \left\{ \begin{array}{l} r_{(k+1)}^n = r_{(k)}^n + (v_{(k+1)}^n \Delta t) + (a_{(k+1)}^n (\Delta t^2)/2) \\ v_{(k+1)}^n = v_{(k)}^n + (a_{(k+1)}^n \Delta t) \\ a_{(k+1)}^n = R^{bn}(y_{a(k)} - x_{a(k)}) + g \\ x_{a(k+1)} = (1 - \lambda_{xa} \Delta t) x_{a(k)} \end{array} \right.$$

Assunta nota la frequenza di aggiornamento del filtro paria  $100Hz$ , è noto il valore di  $\Delta t = 0.01s$ . In dettaglio, l'assetto viene aggiornato tramite il prodotto di due quaternioni:  $q_{(k)}^{nb}$  è il quaternione che rappresenta la rotazione per passare dal frame navigation a quello body attuale,  $q([\omega_{(k+1)}^b \Delta t]')$  è il quaternione che rappresenta la rotazione dal frame body attuale al

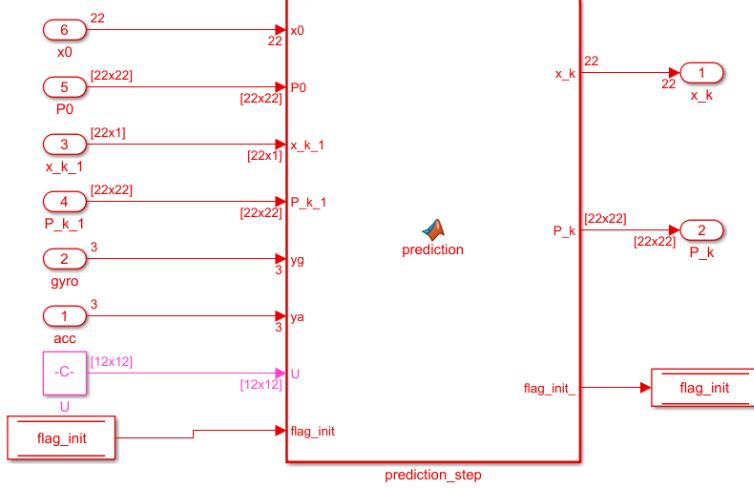


Figura 3.3: Blocco del passo di predizione

frame body futuro se il corpo ruota con velocità angolare  $\omega^b$  per un tempo  $\Delta t$ . Il quaternione a partire da angoli di Eulero [ $yaw, pitch, roll$ ] si calcola con la formula precedentemente usata 3.6.

Il prodotto tra quaternioni è uguale a:

$$q_a * q_b = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} * \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} b_1 a_1 - b_2 a_2 - b_3 a_3 - b_4 a_4 \\ b_1 a_2 + b_2 a_1 - b_3 a_4 + b_4 a_3 \\ b_1 a_3 + b_2 a_4 + b_3 a_1 - b_4 a_2 \\ b_1 a_4 - b_2 a_3 + b_3 a_2 + b_4 a_1 \end{bmatrix} \quad (3.11)$$

La matrice di rotazione  $R^{bn}$  (body-to-navigation) che ruota la misura dell'accelerometro in frame body per essere inserita nella stima in navigation frame, è calcolata a partire dal quaternione  $q^{nb} = [q_0, q_1, q_2, q_3]'$  come [4]:

$$R^{bn} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (3.12)$$

L'aggiornamento della matrice di covarianza della stima invece segue la formula 2.7:

$$P_{(k+1)} = \nabla F_x P_{(k)} \nabla F'_x + \nabla F_u U \nabla F'_u$$

Per eseguire tale calcolo, ad ogni iterazione vengono valutate le matrici Jacobiane  $\nabla F_x$  e  $\nabla F_u$  in base al valore delle varabili di stato e degli ingressi. Tali matrici sono state calcolate in modo simbolico (vedi `calcolo_matrice_F.m`) da Matlab, per poi essere inserite nel codice del blocco di predizione.

### 3.3 Blocco: System Update

Il blocco di aggiornamento (Fig. 3.4) è diviso in altri tre blocchi distinti: `flag_update` si occupa di aggiornare un vettore di flag che indicano quando sono disponibili nuove misure per la fase di correzione; la funzione `virtual` sulla base del vettore di flag, calcola tutti i parametri e le grandezze necessarie alla fase di correzione; `correction_step` infine aggiorna stima e matrice di covarianza sulla base dei calcoli effettuati dalla funzione precedente.

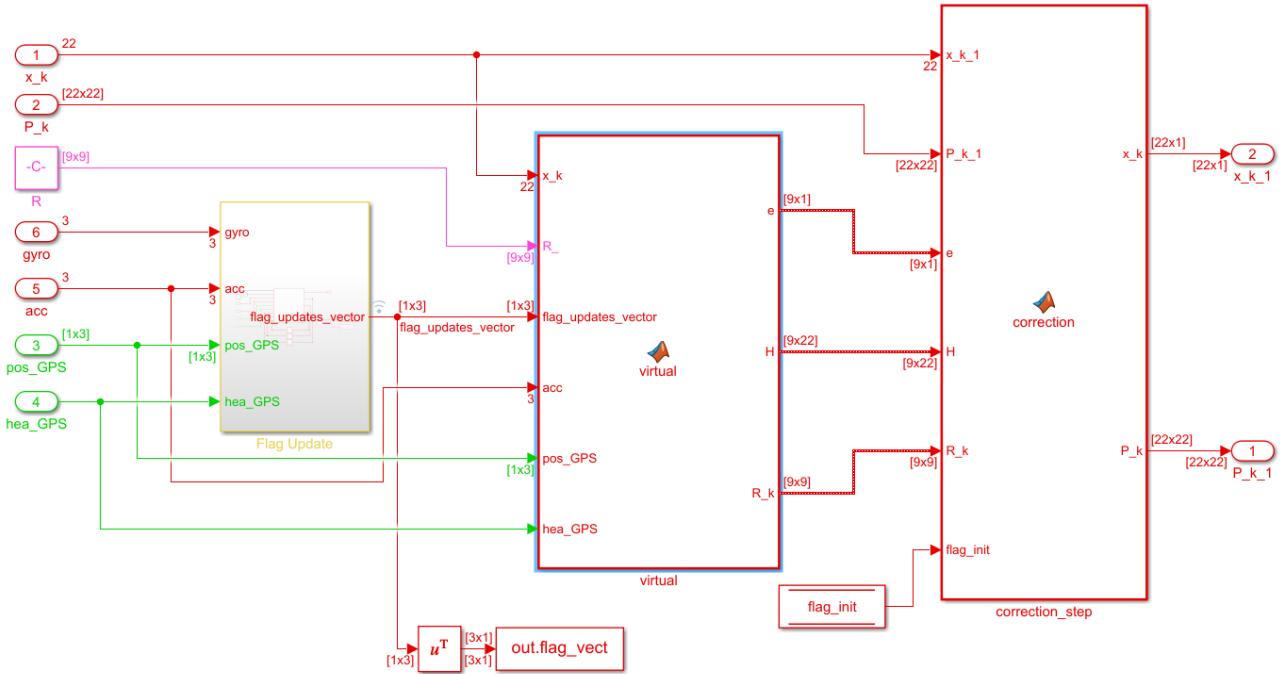


Figura 3.4: Blocco del passo di correzione

### 3.3.1 Blocco: flag\_updates

Nel metodo proposto sono tre i casi in cui avviene una correzione della stima:

1. Dynamical Constraints Update: assumendo che il veicolo è su ruote e che l'asse x del frame body punta verso la direzione di spostamento, si corregge la velocità lungo l'asse y e l'asse z ponendola pari a zero.
2. Roll and Pitch Updates: quando in veicolo è non accelerante, allora dalle misure dell'accelerometro si corregge roll e pitch stimando il vettore gravità.
3. Position and Heading Updates: l'assetto e la posizione si correggono ogni volta sono disponibili i dati del GPS.

Per il caso 1), la frequenza con la quale questa correzione avviene è variabile e dettata dal parametro del filtro *zero\_vel\_period*, calcolato empiricamente e comunque modificabile ad ogni simulazione (vedi sez. *Parametri assegnati empiricamente 5.2* per ulteriori dettagli in merito).

Per l'aggiornamento di roll e pitch 2) invece, il sistema rileva che il veicolo è non accelerante utilizzando lo *SHOE Detector* [2]. In sintesi, assunto  $\bar{y}_{a(k)}$  come la media delle ultime  $N$  misure dell'accelerometro:

$$\bar{y}_{a(k)} = \frac{1}{N} \sum_{k \in [t-N, t]} y_{a(k)} \quad (3.13)$$

tal algoritmo lavora su una finestra temporale mobile delle ultime  $N$  misure dell'accelerometro  $y_{a(k)}$  e giroscopio  $y_{g(k)}$ , calcolando tre diversi valori che insieme stabiliscono se l'accelerazione del veicolo è vicina a zero.

#### 1. Acceleration-Moving Variance Detector

$$\frac{1}{N} \sum_{k \in [t-N, t]} \|y_{a(k)} - \bar{y}_{a(k)}\|^2 < \gamma_1 \quad (3.14)$$

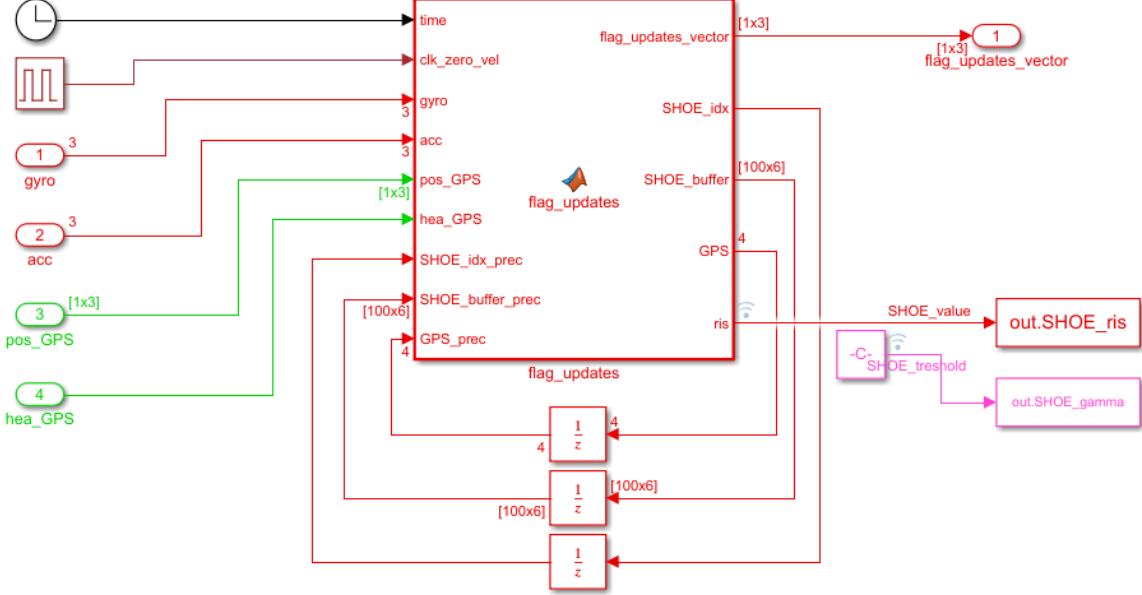


Figura 3.5: Blocco flag-updates

Se l'IMU è stazionaria nel tempo, allora l'orientazione dell'accelerometro è costante e il veicolo è assunto non accelerante. In questo caso non si valuta se l'ampiezza del vettore di forza specifica è uguale alla gravità  $g$ .

## 2. Acceleration-Magnitude Detector

$$\frac{1}{N} \sum_{k \in [t-N, t]} \left( y_{a(k)} - g \frac{y_{a(k)}}{\|y_{a(k)}\|} \right)^2 = \frac{1}{N} \sum_{k \in [t-N, t]} (\|y_{a(k)}\| - g)^2 < \gamma_2 \quad (3.15)$$

Questo secondo rilevatore proposto in letteratura è di solito usato come supplemento al precedente. Controlla se l'ampiezza del segnale misurato dall'accelerometro è vicina al valore della gravità  $g$ , ma trascura il fatto che la direzione del vettore dovrebbe essere costante.

## 3. Angular Rate Energy Detector

$$\frac{1}{N} \sum_{k \in [t-N, t]} \|y_{g(k)}\|^2 < \gamma_3 \quad (3.16)$$

Questo ultimo rilevatore considera un corpo stazionario se l'energia del segnale del giroscopio è sufficientemente piccola.

Fondendo queste tre quantità insieme, e quindi 3.14, 3.15 e 3.16, otteniamo la formula finale dello *SHOE Detector*:

$$\frac{1}{N} \sum_{k \in [t-N, t]} \left( \frac{1}{\sigma_a^2} \left\| y_{a(k)} - g \frac{\bar{y}_{a(k)}}{\|\bar{y}_{a(k)}\|} \right\|^2 + \frac{1}{\sigma_g^2} \|y_{g(k)}\|^2 \right) < \gamma \quad (3.17)$$

Si può notare come ogni peso viene scalato rispetto alla varianza del segnale misurato. Questo può comportare uno squilibrio sui diversi pesi della formula al variare del rapporto  $\sigma_a^2/\sigma_g^2$ . Tuttavia possiamo anche notare come l'*Angular Rate Energy Detector* sia un caso

speciale dello *SHOE Detector* quando  $\sigma_a^2/\sigma_g^2 \rightarrow \infty$ . Il valore della soglia limite  $\gamma$  e del valore  $N$  sono stati assegnati empiricamente dopo diverse simulazioni (vedi sez. *Parametri assegnati empiricamente 5.2*).

Infine, nell'ultimo caso 3), le misure di posizione e heading del GPS sono utilizzate in base alla frequenza del sensore utilizzato. Come specificato nella sezione *The Malaga 2009 Dataset 4*, il GPS RTK ha frequenza pari a  $4\text{Hz}$ , il DGPS low-cost  $1\text{Hz}$ . In questo blocco inoltre possono essere simulati dei malfunzionamenti del GPS in determinati intervalli di tempo, specificati nel file `init.m`, in seguito usati per valutare le performance del filtro in diverse condizioni (vedi sez. *Parametri Variabili 5.1*).

In uscita da questo blocco (Fig. 3.5) avremo quindi un vettore di flag che indica se ognuna delle tre correzioni è attiva o meno.

$$flag\_updates\_vect = \begin{bmatrix} flag_{zero.velocity} \\ flag_{SHOE} \\ flag_{GPS} \end{bmatrix} \quad (3.18)$$

### 3.3.2 Funzione: `virtual`

Nella funzione `virtual` (Fig. 3.4) si assegnano i valori dei vettori  $z$  e  $h$ , quindi del vettore di errore  $e$  (eq. 2.25), e della matrice  $H$  e  $R$  (eq. 2.26) (per il calcolo simbolico della matrice  $H$  vedere il file `calcolo_matrice_H.m`). Ad ogni istante si legge la variabile `flag_updates_vect` passata dal blocco precedente, per cui solo le correzioni ritenute valide concorrono alla creazione dei suddetti vettori e matrici. Se il flag è 1 allora si assegnano i valori come descritto nelle equazioni 2.18, 2.20, 2.23 e 2.24, oppure si lasciano i vettori e matrici con valori pari a 0. Procedere con le equazioni del passo di correzione in questo modo, sarebbe un errore: bisogna ridurre di dimensione  $e$ ,  $H$  e  $R$  eliminando le righe e le colonne non necessarie. Un esempio applicativo di questa riduzione è presente nel file Matlab `cutter_matrix.m`.

### 3.3.3 Funzione: `correction`

Nell'ultima funzione `correction` (Fig. 3.4) si aggiorna la stima dello stato  $\hat{x}$  e della matrice di covarianza della stima  $P$  secondo le equazioni 2.11-2.15, usando i valori di  $e$ ,  $H$  e  $R$  passati dalla funzione precedente.

## 3.4 Funzioni ausiliarie

### 3.4.1 Funzione: `ECEF2NED`



Figura 3.6: Funzione di trasformazione da ECEF a NED

Questa funzione (Fig. 3.6) si occupa della trasformazione delle misure in ingresso del GPS dalle coordinate geodetiche (ECEF) alle coordinate in navigation frame (NED), utilizzando il comando Matlab `geodetic2ned`. Lo sferoide di riferimento per la Terra è il *wgs84Ellipsoid*. La

trasformazione usa come origine del frame NED il valore  $O_{nav}$  (eq. 4.1) definito nella sezione successiva.

### 3.4.2 Funzione: extract\_var

Quest'ultima funzione ha il solo scopo di estrarre dal vettore di stato tutte le variabili per una maggior semplicità in fase di sviluppo e debugging.

## 4 The Málaga 2009 Dataset

Con lo scopo di testare le performance del filtro con dati reali, è stato utilizzato off-line il *Málaga 2009 Dataset* [6] disponibile in rete. Questo dataset è il risultato di una collezione di dati provenienti da diversi sensori con una rispettiva "ground truth", ovvero una posa 6D del veicolo ad alta precisione. A tale scopo è stato utilizzato un veicolo elettrico su ruote (Fig. 4.1) sul quale sono stati montati un totale di dodici sensori: cinque tipi di laser scanner, un'unità di misura inerziale, un DGPS di basso costo, tre GPS RTK e due telecamere a colori.

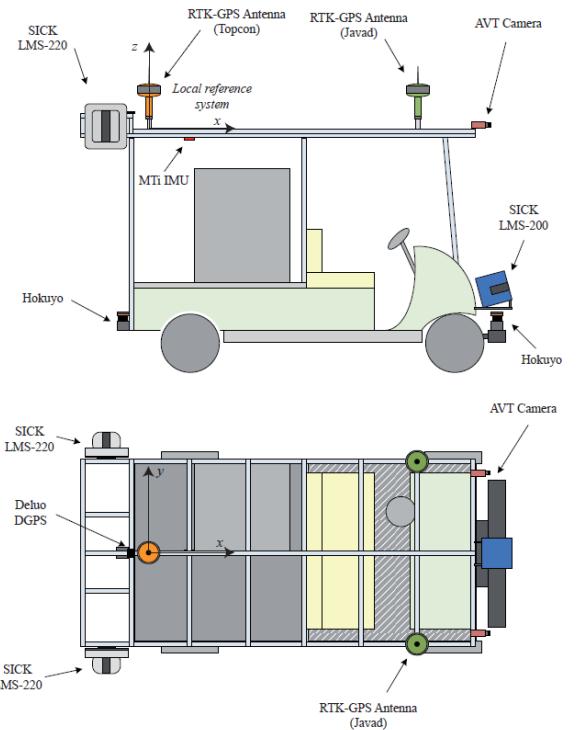


Figura 4.1: Veicolo utilizzato per la collezione di dati sperimentali, con schema delle posizioni dei sensori

Tra la totalità dei datasets collezionati solo uno di essi è stato utilizzato per la validazione del filtro: il [The Málaga 2009 Dataset: PARKING-0L](#). Tra tutti i file disponibili solo alcuni dati sono utili ai fini della simulazione, per cui dal file `parking_0L.rawlog` sono state estratte solo le misure dirette dell'IMU, del GPS RTK (left) e del DGPS low-cost (attraverso l'uso del programma open-source *Mobile Robot Programming Toolkit MRPT*). Sono stati anche utilizzati come riferimento i valori della *6D ground truth*: posizione e orientazione ricavati dall'interpolazione dei tre sensori GPS RTK, raggiungendo un livello di accuratezza dell'ordine del centimetro.

In sintesi tutti i file estrapolati e utilizzati per la simulazione sono i seguenti:

`GT_path_vehicle_100Hz.txt`, `dataset_IMU.txt`, `dataset_GPS_RTK_L.txt`,  
`dataset_GPS_NORMAL.txt`.

Per questo dataset il frame navigation scelto è stato l'ENU [*East, North, Up*] con origine in:

$$O_{nav} = \begin{bmatrix} lat_0 \\ lon_0 \\ elev \end{bmatrix} = \begin{bmatrix} -4.478958828333330 \ deg \\ 36.714459075000006 \ deg \\ 38.888700000000000 \ m \end{bmatrix} \quad (4.1)$$

In Fig. 4.2 è riportata un'immagine satellitare con riferimento al frame ENU e al percorso compiuto.

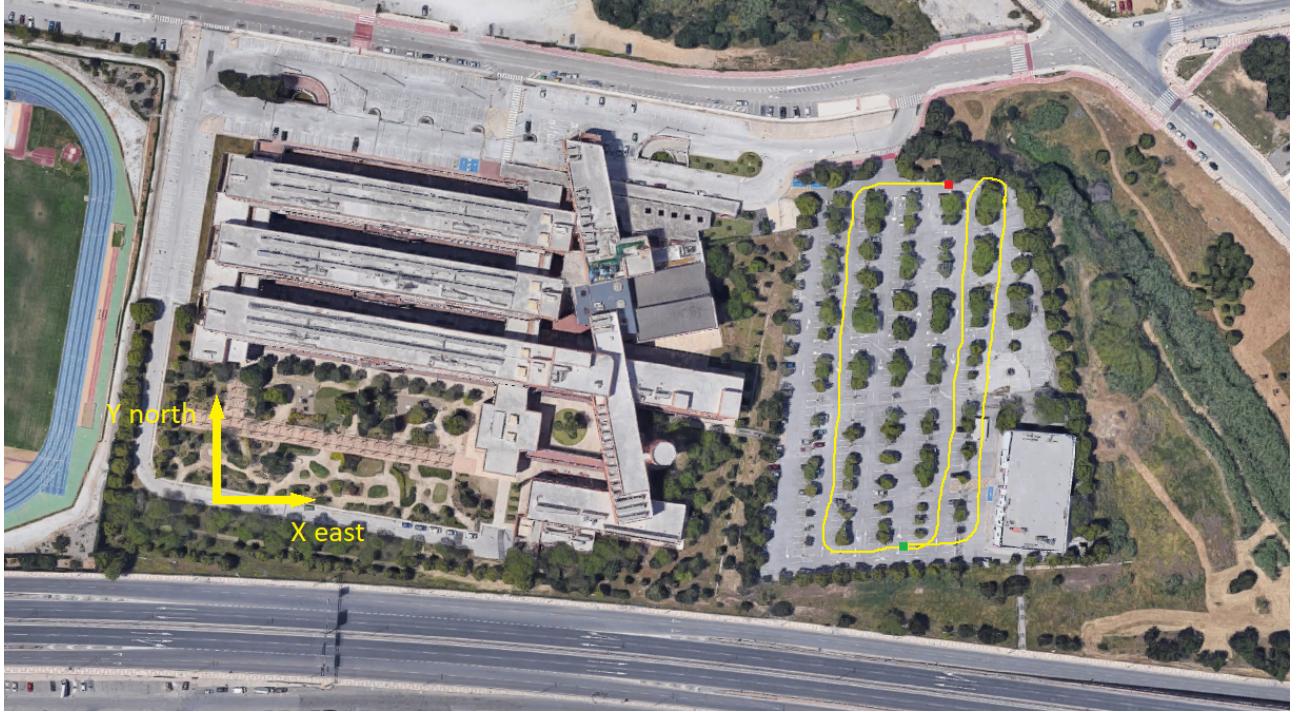


Figura 4.2: Immagine satellitare della posizione dell'origine della terna ENU (asse z verso l'alto) e del percorso compiuto dal veicolo

## 4.1 Adattamento del dataset per la simulazione

Per la simulazione completa in Matlab, le misure dirette di ogni sensore si estraggono dai file di testo eseguendo `load_dataset.m`. Successivamente prima di essere utilizzate in Simulink, alcune di esse saranno adattate in base all'uso proposto.

- `GT_path_vehicle_100Hz.txt`

Questo file contiene una posa 6D del veicolo ad una frequenza di  $100Hz$  dal quale si estraggono posizione  $(r_{ENU}^n)'$  e orientazione in angoli di Eulero  $[yaw, pitch, roll]_{body-to-ENU}$ , espresse in frame ENU.

Come specificato in [6], l'offset ottimo da sommare alle misure del GPS RTK *left* per ottenere la posizione esatta è  $\Delta^2$  [vedi Tabella 3 di [6]]:

$$\Delta^2 = [-260.168 \text{ m} \quad 23.8158 \text{ m} \quad -1.64305 \text{ m}] \quad (4.2)$$

Dal punto di vista simulativo si è deciso di sottrarre questa quantità al valore di  $(r_{ENU}^n)'$  al posto di sommarla alle misure del GPS RTK, con lo scopo di utilizzare questo riferimento di posizione esatta anche per le simulazioni con il DGPS low-cost. Inoltre, dal momento che il filtro implementato utilizza come frame navigation la terna NED, è necessario moltiplicare il vettore posizione per la matrice di rotazione  $R_{ENU-to-NED}$ :

$$R_{ENU-to-NED} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (4.3)$$

ottenendo quindi:

$$r_{NED}^n = R_{ENU-to-NED} \begin{bmatrix} (r_{ENU}^n)'_x - \Delta_x^2 \\ (r_{ENU}^n)'_y - \Delta_y^2 \\ (r_{ENU}^n)'_z - \Delta_z^2 \end{bmatrix} \quad (4.4)$$

Con l'equazione 4.4 otteniamo quindi  $r_{NED}^n$  che sarà considerata la posizione esatta di riferimento del veicolo lungo l'intero percorso.

Per quanto riguarda l'orientazione esatta di riferimento, è necessaria la trasformazione degli angoli di Eulero che descriveranno la rotazione *body-to-NED* invece di *body-to-ENU*. Il primo passo è calcolare la matrice di rotazione equivalente dagli angoli di  $[yaw, pitch, roll]_{body-to-ENU}$ :

$$R_{body-to-ENU} = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi)c(\phi) + c(\psi)s(\theta)s(\phi) & s(\psi)s(\phi) + c(\psi)s(\theta)c(\phi) \\ s(\psi)c(\theta) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & -c(\psi)s(\phi) + s(\psi)s(\theta)c(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \quad (4.5)$$

(o equivalentemente data  $R^{nb}$  di eq. 3.4,  $R_{body-to-ENU} = (R^{nb})^T$ ). Successivamente si applica la trasformazione con la matrice  $R_{ENU-to-NED}$  (eq. 4.3) componendo le rotazioni in assi mobili:

$$R_{body-to-NED} = R_{ENU-to-NED} R_{body-to-ENU} = \bar{R} \quad (4.6)$$

e infine si risale agli angoli di Eulero  $[yaw, pitch, roll]_{body-to-NED}$  corrispondenti secondo le formule [5] [7]:

$$\psi_{yaw} = \arctan\left(\frac{\bar{R}_{21}}{\bar{R}_{11}}\right) \quad \theta_{pitch} = \arctan\left(\frac{-\bar{R}_{31}}{\sqrt{1 - \bar{R}_{31}^2}}\right) \quad \phi_{roll} = \arctan\left(\frac{\bar{R}_{32}}{\bar{R}_{33}}\right) \quad (4.7)$$

Questa ultima funzione è eseguita dal comando Matlab `rotm2eul` che tiene conto di eventuali singolarità durante i cambi di rappresentazione.

- **dataset\_IMU.txt**

Questo file contiene le misure dell'unità inerziale, quindi di accelerometro  $y_a$  e giroscopio  $y_g$ , ad una frequenza di  $100Hz$ . Le misure sono in frame body quindi non subiscono alcuna trasformazione.

- **dataset\_GPS\_RTK\_L.txt dataset\_GPS\_NORMAL.txt**

Questi file contengono le misure del GPS RTK e del DGPS low-cost a frequenza rispettivamente  $4Hz$  e  $1Hz$ . Le misure estrapolate sono di posizione  $z_r = [lat, lon, h]$  e angolo di heading  $z_\theta = [rad]$ .

Le misure di posizione  $z_r$  saranno inviate direttamente al blocco *ECEF2NED* (Sez. 3.4.1) pronte per essere processate dal filtro. Le misure di heading  $z_\theta$ , invece, saranno solo rimappate tra  $[-\pi, \pi]$  tramite il comando Matlab `wrapToPi`.

## 4.2 Grafici delle misure utilizzate

Di seguito si riportano dei grafici delle misure utilizzate in ingresso al filtro.

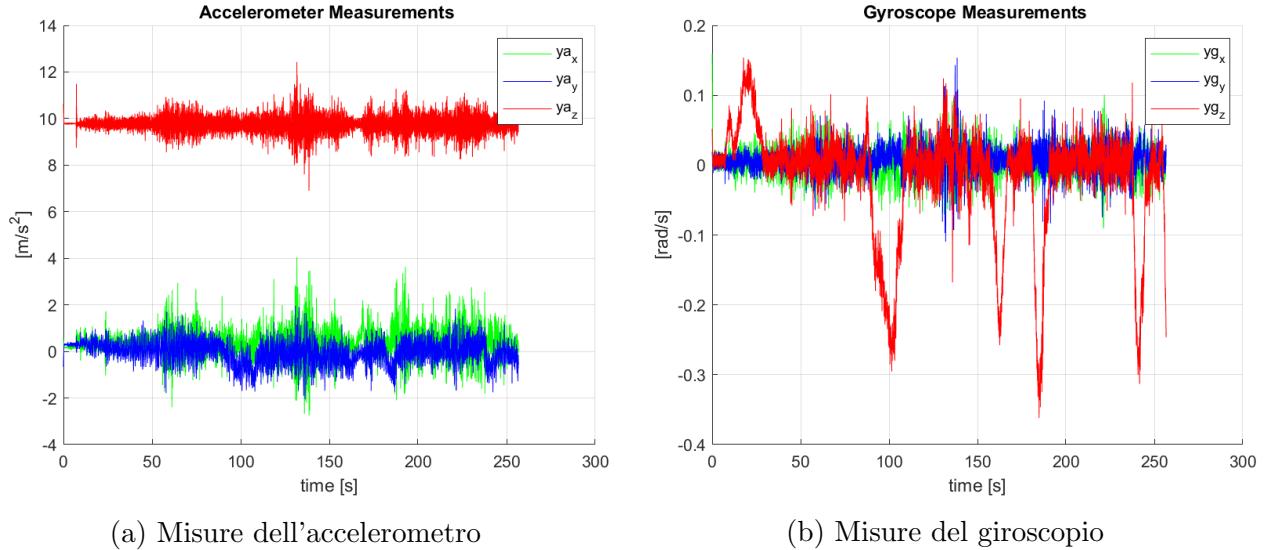


Figura 4.3: Misure provenienti dell'unità inerziale IMU

Dalle misure del giroscopio si nota infatti che solo la velocità angolare dell'angolo di yaw assume valori in media diversi da zero, questo appunto è in linea con i vincoli dinamici del veicolo.

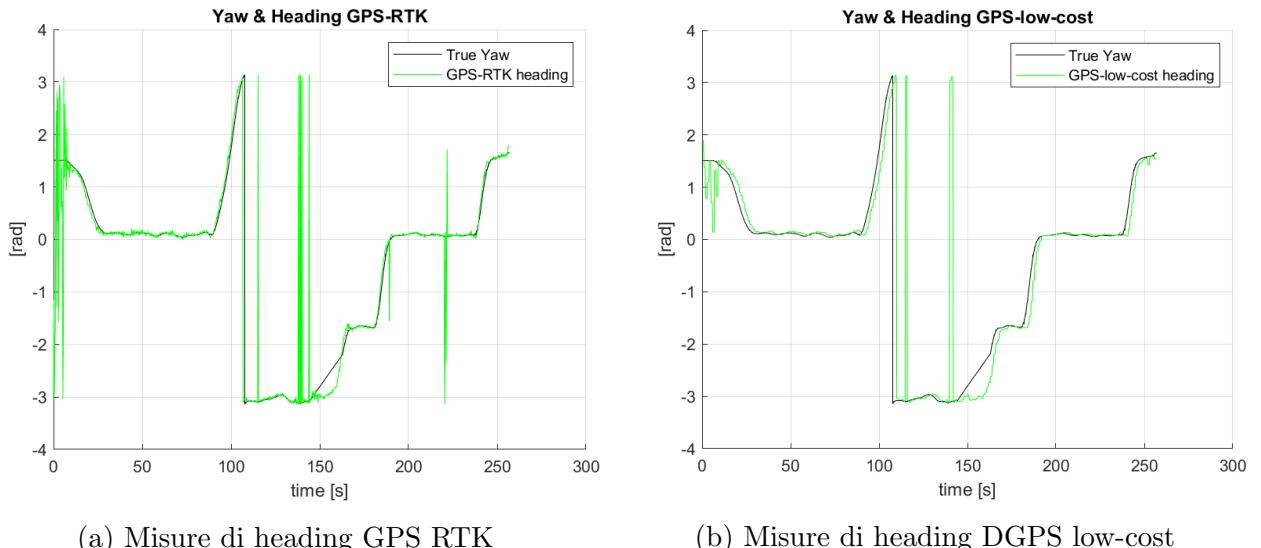
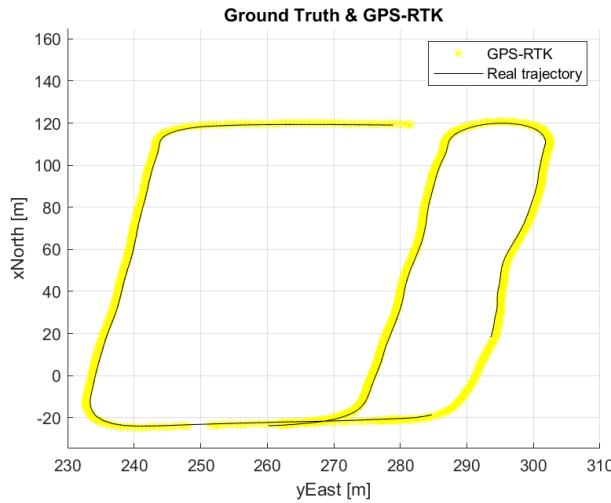
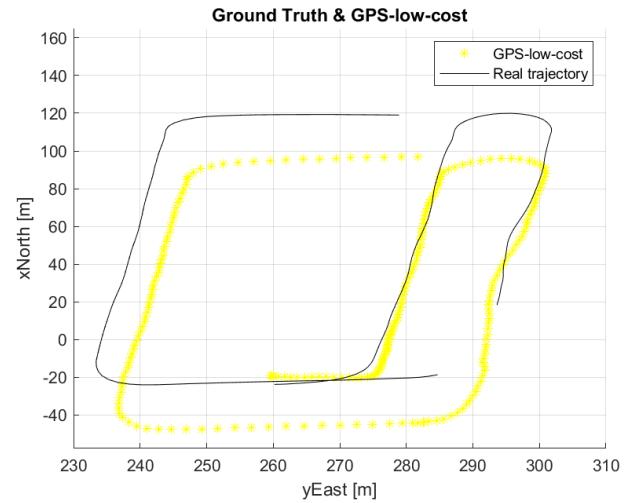


Figura 4.4: Misure dell'angolo di heading del GPS

Da un primo confronto tra le misure di heading dei due GPS, si nota che la misura proveniente dal GPS RTK (4.4a) presenta degli *outliers* di ampiezza maggiore rispetto all'altra, soprattutto nei primi istanti della simulazione quando ancora il veicolo è fermo e il sensore GPS non riesce a misurare bene tale misura.



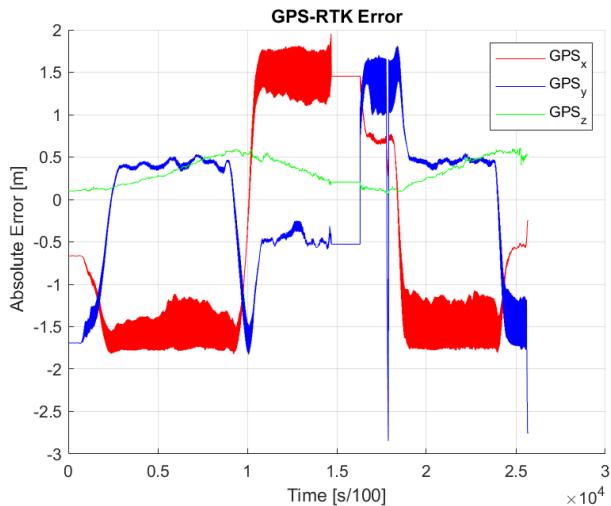
(a) Misure di posizione GPS RTK



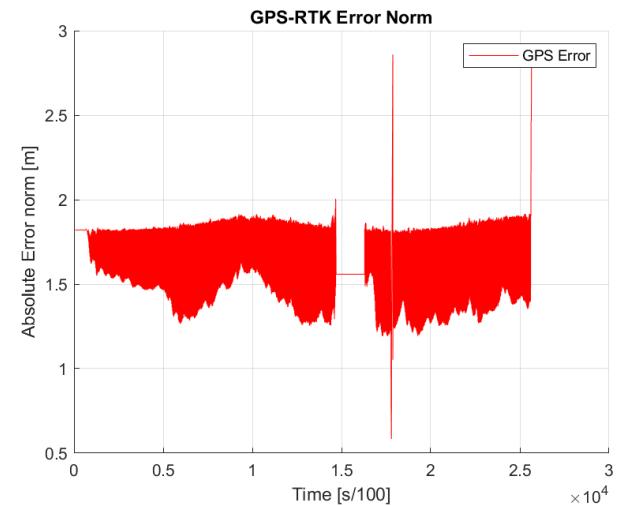
(b) Misure di posizione DGPS low-cost

Figura 4.5: Plot 2D con posizione di riferimento e misure del GPS

Data la differenza di accuratezza delle misure di posizione dei GPS, per completezza si riportano i grafici anche degli errori assoluti per componente, calcolati come differenza tra posizione di riferimento e misura del GPS, e la norma nel tempo.



(a) Errore assoluto per componente



(b) Norma dell'errore assoluto

Figura 4.6: Errore assoluto e norma tra posizione di riferimento e GPS RTK

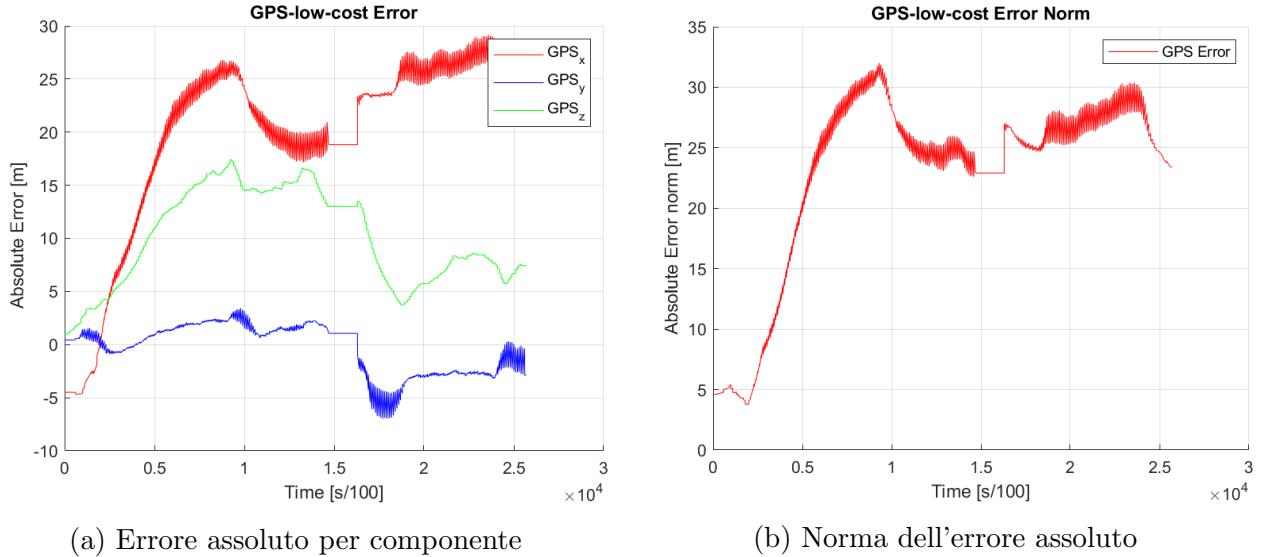


Figura 4.7: Errore assoluto e norma tra posizione di riferimento e DGPS low-cost

Per queste ultime misure, l'*RMSE* calcolato come:

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (r_{GT}^n - z_r^n)^2} \quad (4.8)$$

con  $r_{GT}^n$  posizione di riferimento e  $z_r^n$  misura del GPS, vale:

	$RMSE_x$	$RMSE_y$	$RMSE_z$	$RMSE_{tot}$
GPS RTK	1.3592m	0.83836m	0.36023m	1.637m
DGPS low-cost	21.297m	2.3438m	10.993m	24.081m

Tabella 4.1: RMSE delle misure dei GPS

Da questi errori si nota come la differenza tra i due sensori GPS sia piuttosto marcata e si può già ipotizzare che le correzioni che si basano su questi dati non renderanno la stima della posizione particolarmente accurata.

## 5 Parametri della Simulazione

Creata la struttura e preparato il dataset per la simulazione, non resta che attribuire valori concreti alle variabili e ai parametri rimanenti del sistema. Questi verranno assegnati dal file `init.m`, da eseguire prima di lanciare la simulazione su Simulink, secondo la seguente tabella.

Parametro	Simbolo	Descrizione	Valore	Unità
<code>type_of_GPS</code>	-	Tipo di GPS utilizzato	[0, 1]	-
<code>t*_start</code>	-	Inizio dell'intervallo * di interruzione GPS	[0 ÷ 257]	(s)
<code>t*_end</code>	-	Fine dell'intervallo * di interruzione GPS	[0 ÷ 257]	(s)
<code>Delta2</code>	$\Delta^2$	Offset per posizione di riferimento	-	(m)
<code>lat0</code>	$O_{nav}(1)$	Origine ENU/NED ( <i>latitude</i> )	-	(deg)
<code>lon0</code>	$O_{nav}(2)$	Origine ENU/NED ( <i>longitude</i> )	-	(deg)
<code>h0</code>	$O_{nav}(3)$	Origine ENU/NED ( <i>elevation</i> )	-	(m)
<code>gyro_variance</code>	$\sigma_g^2$	Varianza del giroscopio	$4.8 \times 10^{-6}$	(rad/s) <sup>2</sup>
<code>bias_gyro_variance</code>	$\sigma_{xg}^2$	Drift-rate del bias del giroscopio	$4.0 \times 10^{-14}$	(rad/s <sup>2</sup> ) <sup>2</sup>
<code>acc_variance</code>	$\sigma_a^2$	Varianza dell'accelerometro	$4.8 \times 10^{-2}$	(m/s <sup>2</sup> ) <sup>2</sup>
<code>bias_acc_variance</code>	$\sigma_{xa}^2$	Drift-rate del bias dell'accelerometro	$4.0 \times 10^{-14}$	(m/s <sup>3</sup> ) <sup>2</sup>
<code>lambda_xg</code>	$\lambda_{xg}$	Costante di tempo per bias del giroscopio	$1.0 \times 10^{-3}$	1/s
<code>lambda_xa</code>	$\lambda_{xa}$	Costante di tempo per bias dell'accelerometro	$1.0 \times 10^{-4}$	1/s
<code>vel_variance</code>	$\sigma_v^2$	Varianza per il vincolo dinamico	$1.0 \times 10^{-2}$	(m/s) <sup>2</sup>
<code>GPS_variance</code>	$\sigma_r^2$	Varianza delle misure di posizione del GPS	[1]	(m) <sup>2</sup>
<code>head_variance</code>	$\sigma_\theta^2$	Varianza delle misure di heading del GPS	[2]	(rad) <sup>2</sup>
<code>EKF_DT</code>	$\Delta t$	Periodo di aggiornamento dell'EKF	0.01	(s)
<code>g</code>	$g$	Costante di gravità	9.81	(m/s <sup>2</sup> )
<code>SHOE_buffer_N</code>	$N$	Numero di campioni per SHOE	100	-
<code>SHOE_threshold</code>	$\gamma$	Soglia limite per SHOE	2000	-
<code>zero_vel_period</code>	-	Frequenza correzione vincoli dinamici	0.1	(s)

Tabella 5.1: Tabella dei parametri della simulazione

[1] [2] in base al GPS utilizzato per la simulazione, si assegnano i valori:  
`GPS_variance=  $1.0 \times 10^{-2}$  ( $m$ ) $^2$`  e `head_variance=  $7.0 \times 10^{-3}$  ( $rad$ ) $^2$`  nel caso di GPS RTK;  
`GPS_variance=  $5$  ( $m$ ) $^2$`  e `head_variance=  $5.0 \times 10^{-2}$  ( $rad$ ) $^2$`  nel caso di DGPS low-cost.

## 5.1 Parametri variabili

I seguenti parametri potranno essere modificati a piacere prima di ogni simulazione.

- `type_of_GPS`

Come detto in precedenza, per valutare le prestazioni del filtro si utilizzeranno le misure di due tipi differenti di GPS. La variabile `type_of_GPS` può assumere due valori [0, 1] per selezionare i dati del DGPS low-cost o RTK rispettivamente.

- `t*_start t*_end`

Vedremo come la stima dello stato sarà influenzata durante delle interruzioni forzate del segnale GPS con l'utilizzo o meno della stima del bias di giroscopio e accelerometro. Le variabili `t*_start` e `t*_end` (con \* = [0, 1, 2, 3, 4]) determineranno gli intervalli di tempo di tali interruzioni, e dovranno assumere valori crescenti ed essere minori del tempo di simulazione  $t_{sim} = 257s$ . Per eseguire la simulazione senza interruzioni, impostare tutti questi valori pari a 1.

## 5.2 Parametri assegnati empiricamente

I seguenti parametri invece sono stati valutati empiricamente a seguito di diverse simulazioni.

- `SHOE_buffer_N SHOE_threshold`

Come descritto in [2] e riportato nella sez. *Blocco: flag\_updates 3.3.1*, l'utilizzo dello *SHOE Detector* determina se il veicolo è non-accelerante e di conseguenza si possono applicare le adeguate correzioni. L'equazione finale 3.17 qui riportata:

$$ris = \frac{1}{N} \sum_{k \in [t-N, t]} \left( \frac{1}{\sigma_a^2} \left\| y_{a(k)} - g \frac{\bar{y}_{a(k)}}{\|\bar{y}_{a(k)}\|} \right\|^2 + \frac{1}{\sigma_g^2} \|y_{g(k)}\|^2 \right) < \gamma$$

suggerisce di calcolare una quantità (chiamata *ris*) basata sulle ultime  $N$  misure di accelerometro e giroscopio e confrontarla con la *soglia*  $\gamma$ .

Di seguito sono riportati tre andamenti significativi della suddetta quantità con tre diversi valori di  $N$  ([10, 100, 1000]), in Fig. 5.1. Si nota come scegliere un valore relativamente piccolo, comporta un valore della quantità *ris* più rumoroso in quanto più esposto a repentine variazioni. Un valore esageratamente alto invece, introduce un ritardo rendendo la valutazione più *lenta* e attenuata. I picchi degli andamenti in corrispondono ai momenti in cui il veicolo ruota lungo il percorso per cambiare direzione.

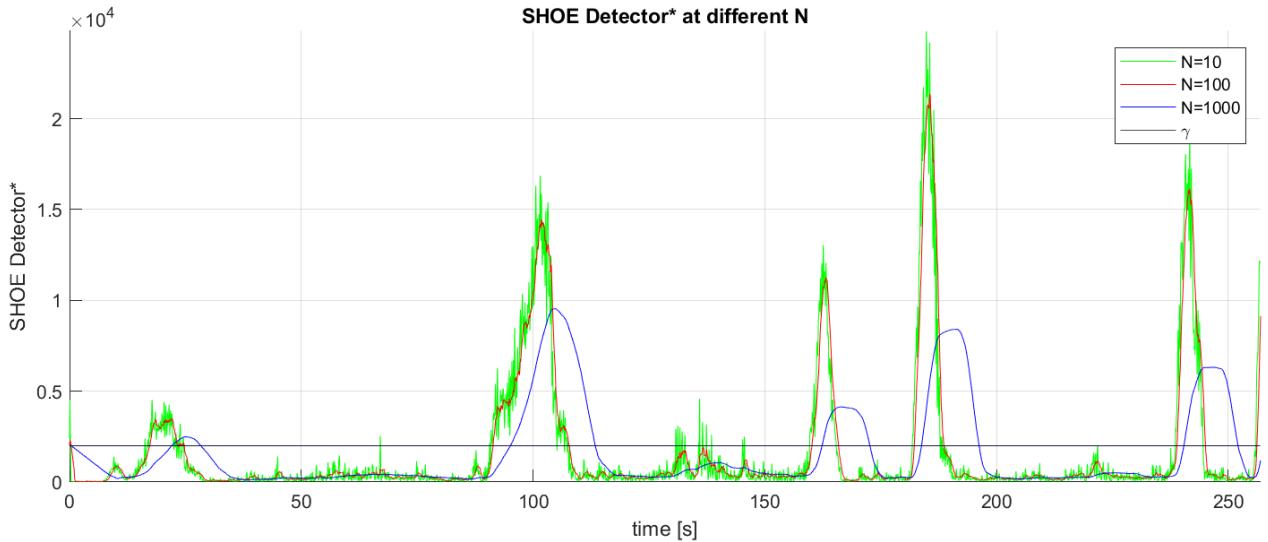


Figura 5.1: Andamento di *ris* al variare della finestra temporale di campioni  $N$

La finestra temporale scelta è quindi  $N = SHOE\_buffer\_N = 100$  che equivale a  $1\text{s}$  dato che la frequenza di aggiornamento del filtro è  $100\text{Hz}$ . Il valore della *soglia*  $\gamma$  è scelto pari a  $\gamma = SHOE\_threshold = 2000$ .

- **`init_time`**

Questo parametro esprime il tempo in secondi necessario per effettuare l'inizializzazione del filtro. Varie simulazioni con poco contenuto informativo portano a scegliere questo valore pari a qualche secondo, per cui si pone orientativamente  $init\_time = 4[\text{s}]$ .

- **`zero_vel_period`**

Il veicolo utilizzato per la collezione dei dataset è un veicolo elettrico su ruote e come tale rispetta dei vincoli dinamici. Questa correzione, descritta ampiamente nei capitoli precedenti, viene eseguita periodicamente ad una frequenza "media", o equivalentemente ad un periodo pari a *zero\_vel\_period*. Vari grafici che descrivono come la stima migliora o peggiora al variare di questo parametro sono riportati nel prossimo capitolo (Sez. 6.5), mentre il valore proposto è di  $zero\_vel\_period = 0.1[\text{s}]$ .

### 5.3 Parametri costanti

I restanti parametri della tabella 5.1 saranno invece *costanti* per ogni simulazione. Si tratta appunto delle varianze delle misure dei sensori, dei parametri fissi relativi al dataset e dei parametri di sistema.

## 6 Risultati Sperimentali

Come proposto in [1], per valutare le prestazioni del filtro e la qualità della stima degli errori dei sensori, si forzano delle interruzioni del segnale GPS, valutando lo stato stimato con il filtro completo e con il filtro senza la stima dei bias (per quest'ultimo si sono ridotte le dimensioni dello stato e di tutte le altre variabili escludendo completamente i bias  $x_g$  e  $x_a$  da ogni calcolo). Seguono quindi due simulazioni comparative in questo senso con i due tipi di sensori GPS e altre due escludendo le interruzioni dei GPS.

Una volta aperta la cartella `Matlab_files\sim_car_dataset`, per riprodurre queste simulazioni è necessario:

1. eseguire il file di inizializzazione `init.m` per assegnare tutti i parametri (vedi capitolo precedente);
2. eseguire il file Simulink `EKF_sim.slx` con un tempo di simulazione pari a  $257s$ .

La figura 6.1 rappresenta lo schema Simulink completo così come appare nel file `EKF_sim.slx`: nella parte iniziale si estraggono i dati dei sensori da variabili (*timeseries*) salvate nel workspace di Matlab; seguono i due filtri di Kalman con o senza stima dei bias; infine vengono salvati i valori dello stato stimato dei due filtri.

I grafici significativi mostrati di seguito possono essere visualizzati o attraverso il *Data Inspector* (assicurarsi che i segnali di interesse abbiano il *Data Logging* attivo), oppure eseguendo i file nella cartella "other\_functions": `plot_2D_GPS.m`, `plot_IMU.m`, `plot_SHOE.m`, `plot_state_all.m`, `plot_state_pos_2D_EKFs.m`, `plot_state_ypr.m`.

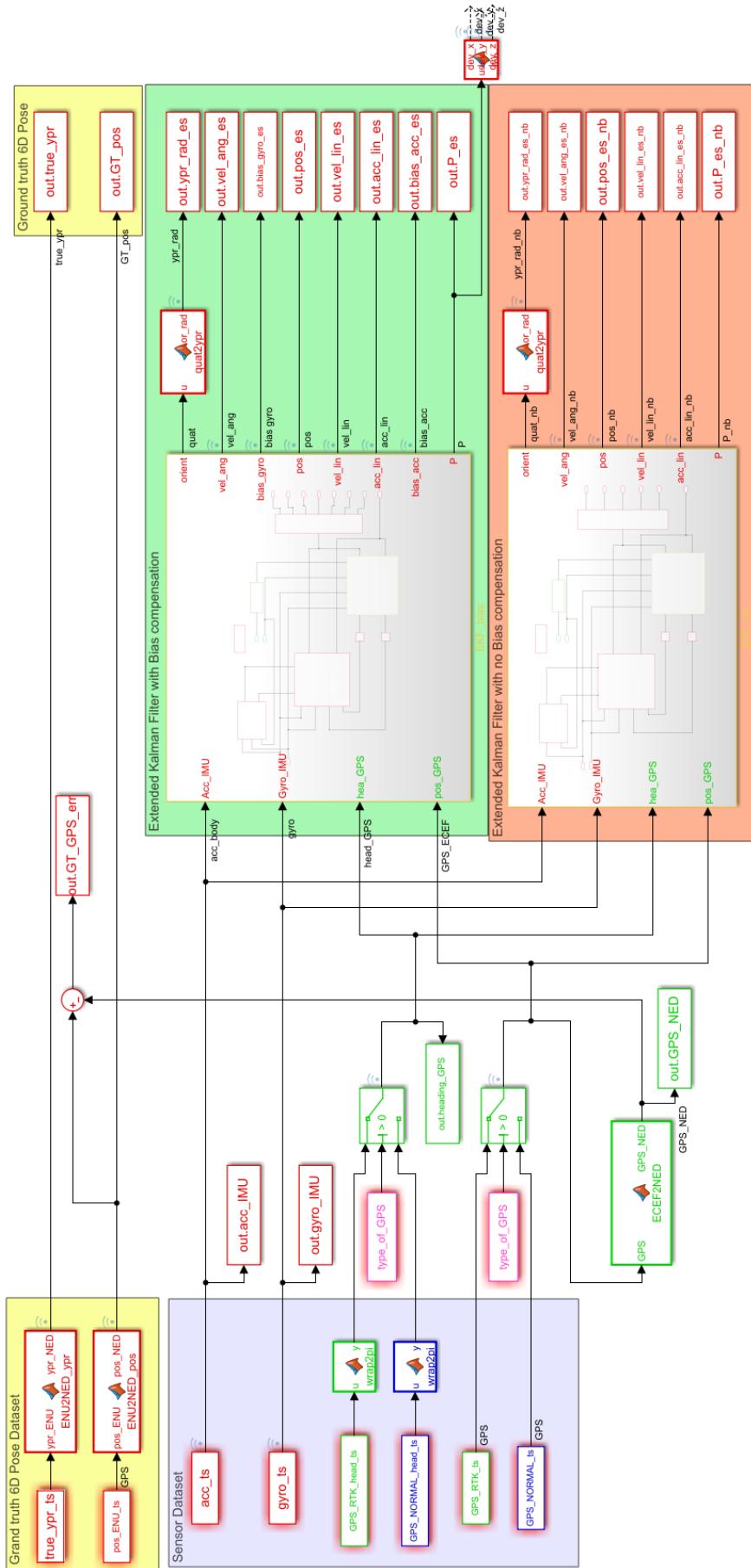


Figura 6.1: Schema Simulink completo della simulazione.

## 6.1 Simulazione con GPS RTK con interruzioni

Gli intervalli delle interruzioni del segnale GPS sono assegnati come segue (simili a [1]):

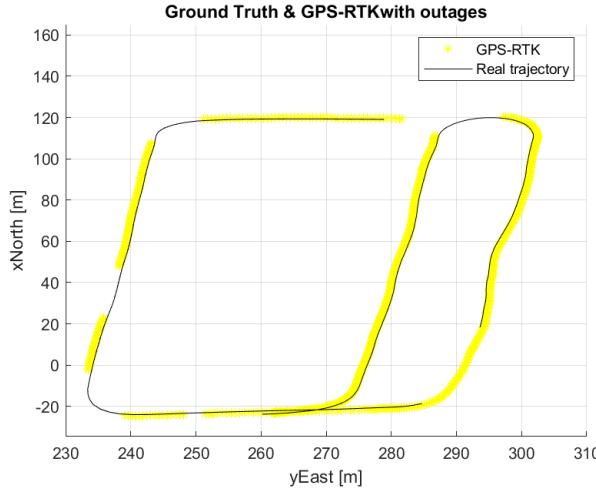
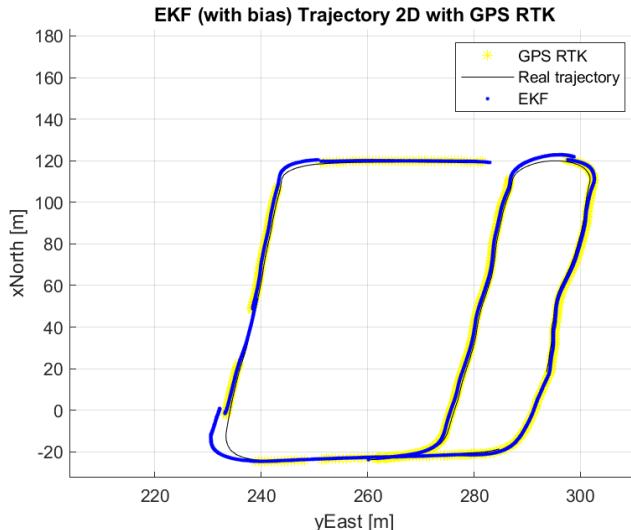


Figura 6.2: Segnale del GPS

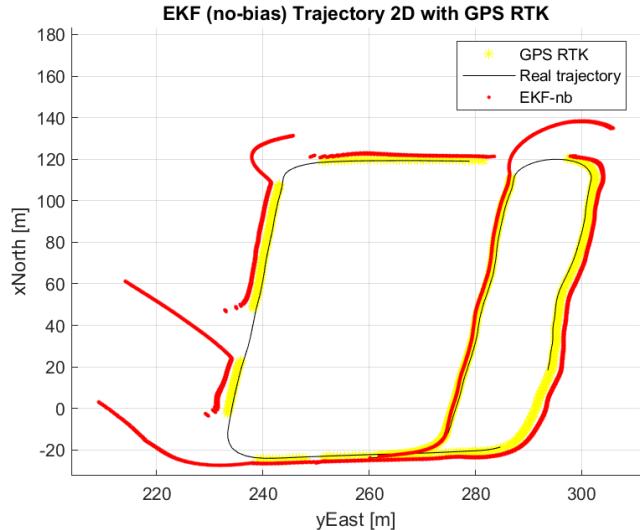
	Valore	Unità
t1_start	90	s
t1_end	100	s
t2_start	183	s
t2_end	193	s
t3_start	203	s
t3_end	213	s
t4_start	237	s
t4_end	244	s

Tabella 6.1: Intervalli di tempo degli *outages*

La posizione stimata dal filtro è:



(a) EKF con stima del bias



(b) EKF senza stima del bias

Figura 6.3: Confronto tra posizione stimata e posizione reale con EKF con stima del bias di accelerometro e giroscopio (a), e senza (b)

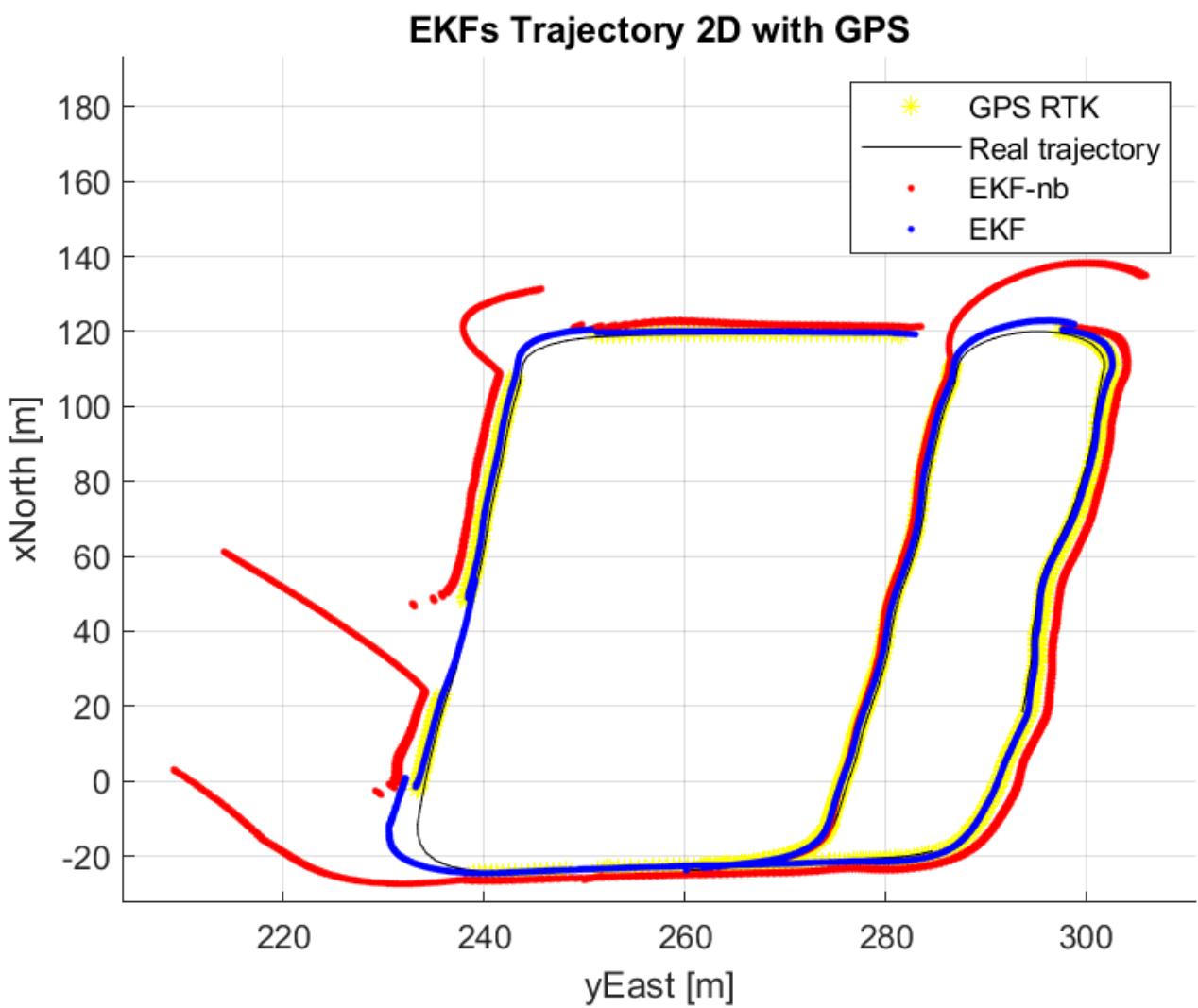


Figura 6.4: Confronto tra posizione stimata e posizione esatta

Le figure 6.3 e 6.4 mostrano quello che ci aspettavamo quando i bias del giroscopio e accelerometro sono ignorati: la stima della posizione peggiora notevolmente appena le misure del GPS non sono più disponibili, deducendo quindi che la stima di entrambi i bias è abbastanza corretta ed efficace.

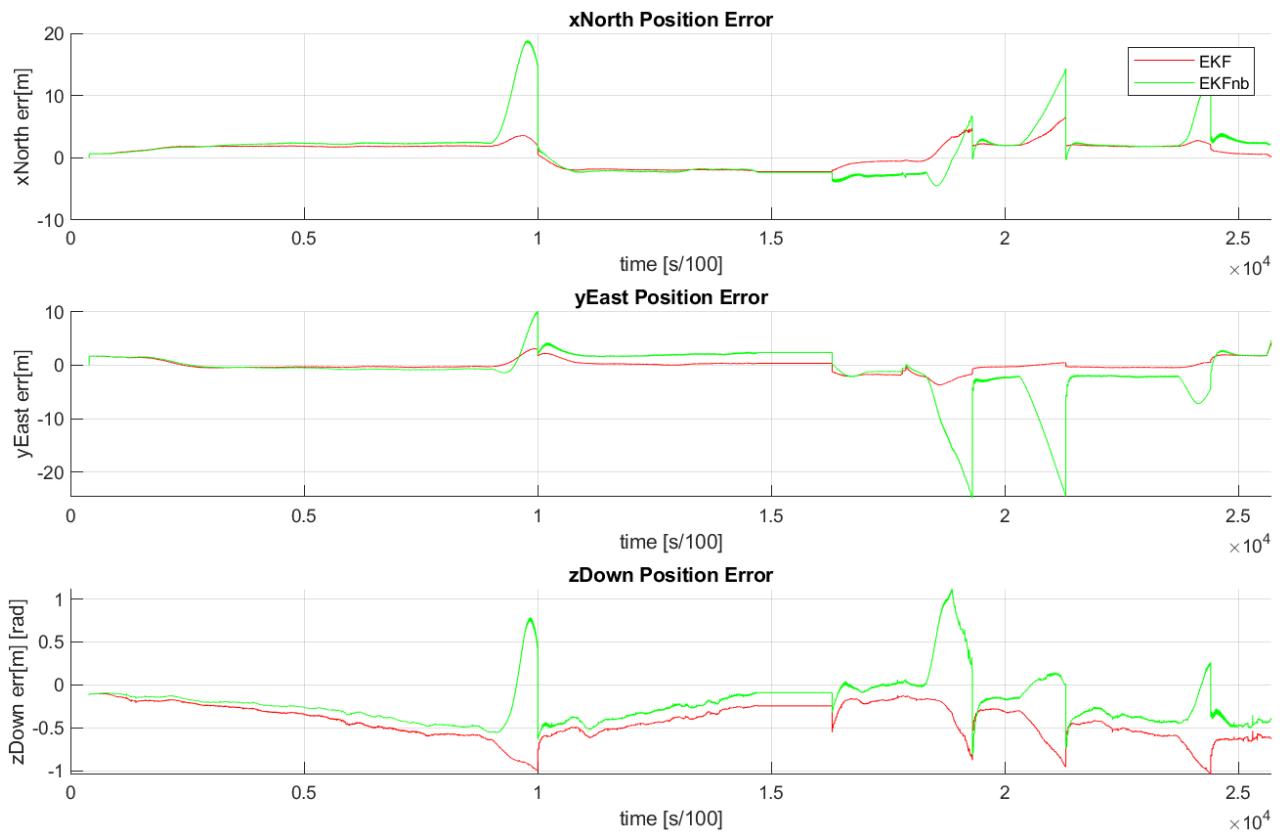


Figura 6.5: Errori assoluti delle componenti della stima di posizione

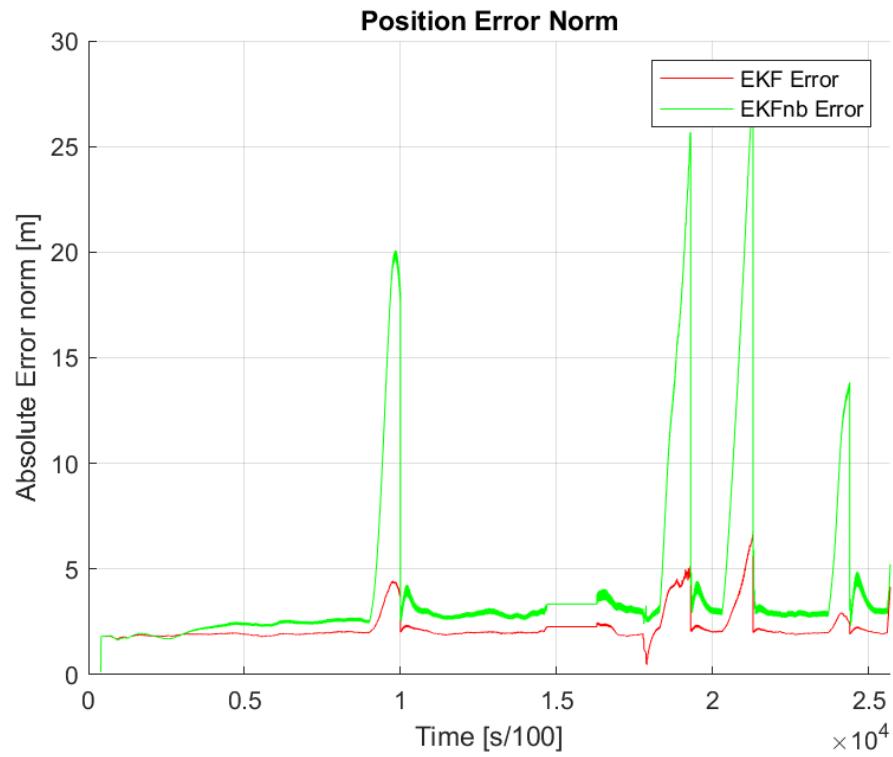


Figura 6.6: Norma dell'errore assoluto della stima di posizione

	$RMSE_x$	$RMSE_y$	$RMSE_z$	$RMSE_{tot}$	Unità
EKF	1.9883	1.0709	0.4643	2.3055	$m$
EKFnb	3.9020	4.5273	0.3336	5.9861	$m$

Tabella 6.2: RMSE della stima di posizione

Dalle figure 6.5 e 6.6, è ancora più evidente come la stima della posizione durante gli intervalli di interruzione del GPS è diversa nei due filtri, differenza più marcata nelle componenti *xNorth* e *yEast* rispetto a *zDown*, il cui errore rimane più o meno sotto il metro in valore assoluto per entrambi i filtri.

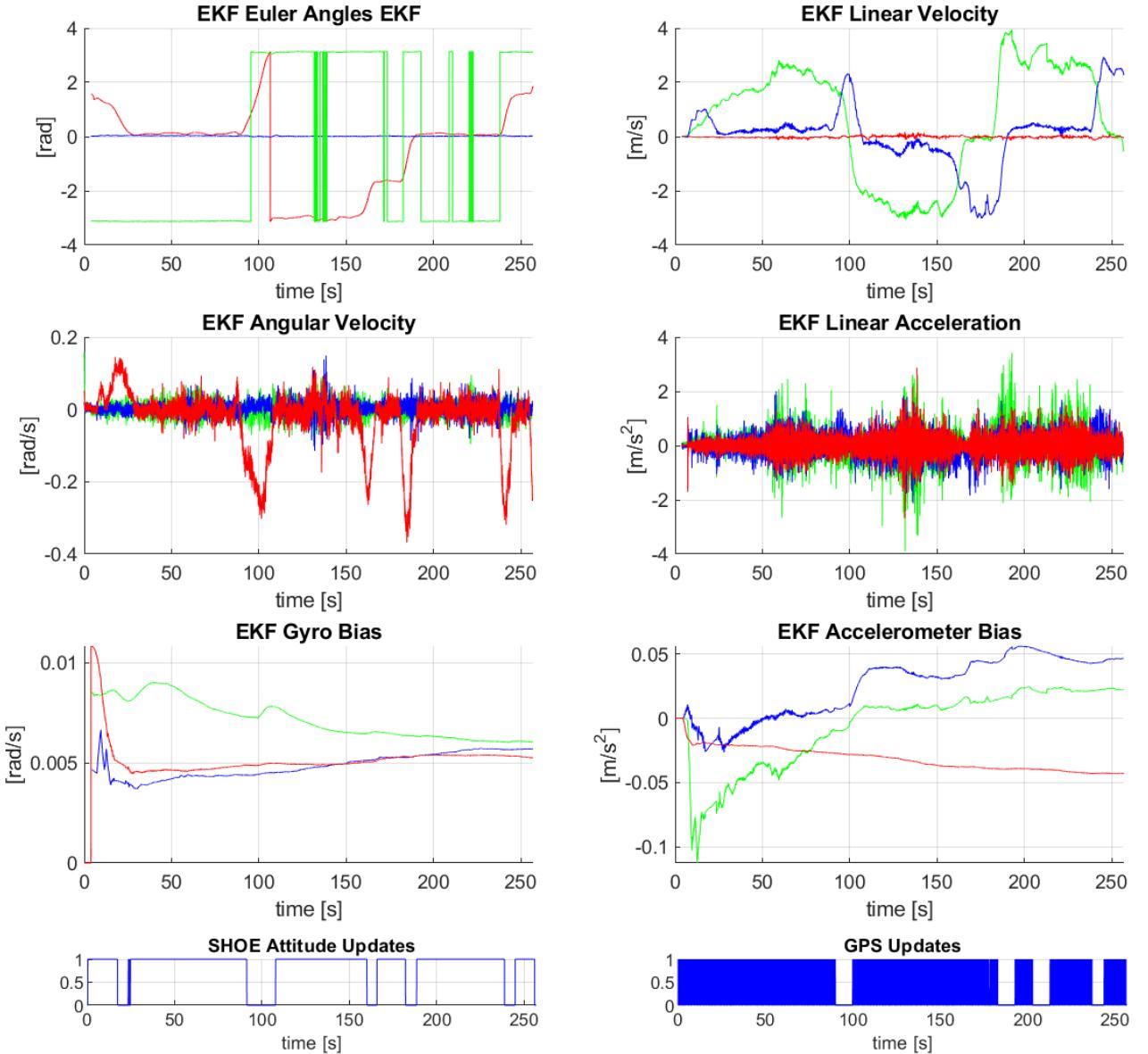


Figura 6.7: Variabili stimate dall'EKF con bias: assetto [yaw,pitch,roll], velocità lineare, velocità angolare, accelerazione e bias con componenti [Z,Y,X], i grafici in basso indicano quando le correzioni di assetto e del GPS avvengono.

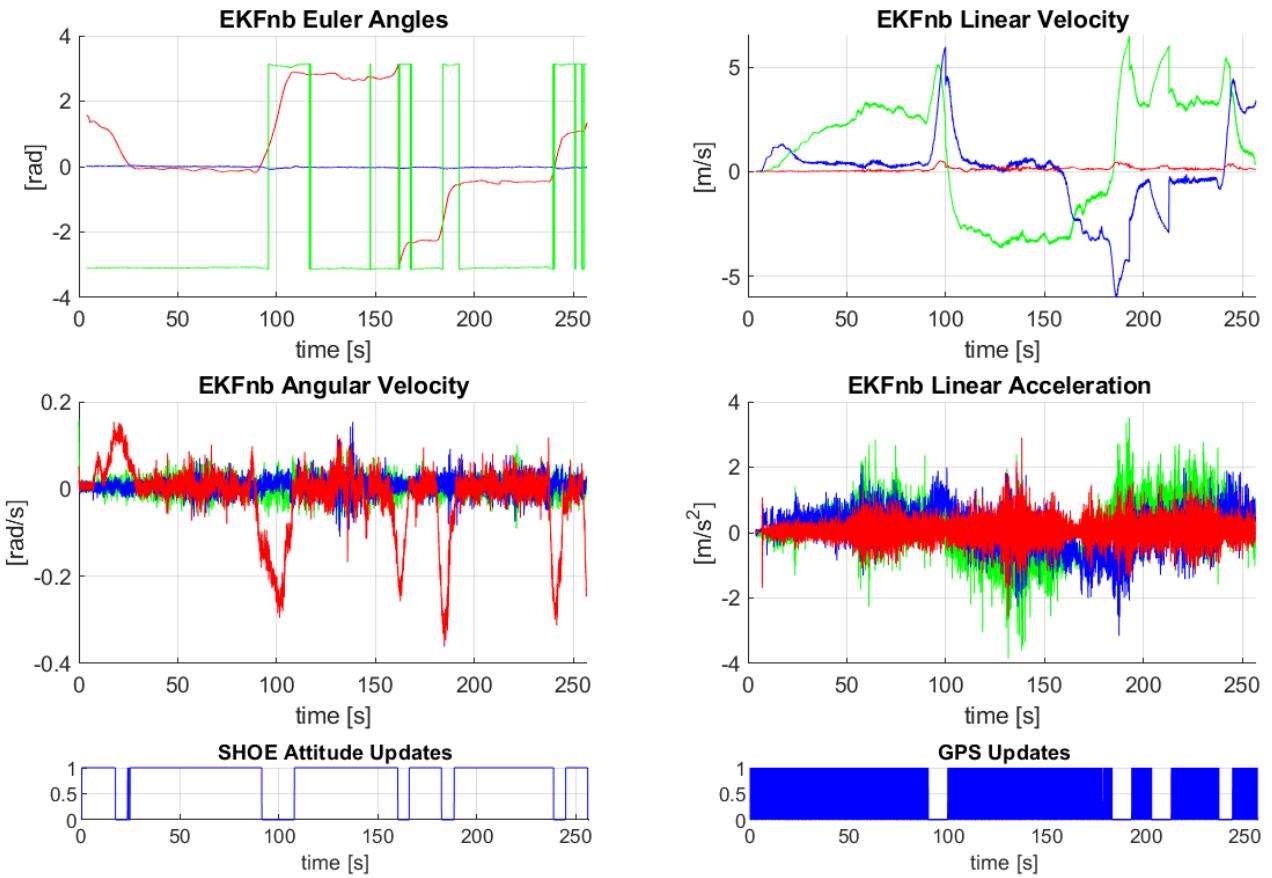


Figura 6.8: Variabili stimate dall'EKF senza bias: assetto [yaw,pitch,roll], velocità lineare, velocità angolare, accelerazione con componenti [Z,Y,X], i grafici in basso indicano quando le correzioni di assetto e del GPS avvengono.

Per quanto riguarda l'assetto in particolare:

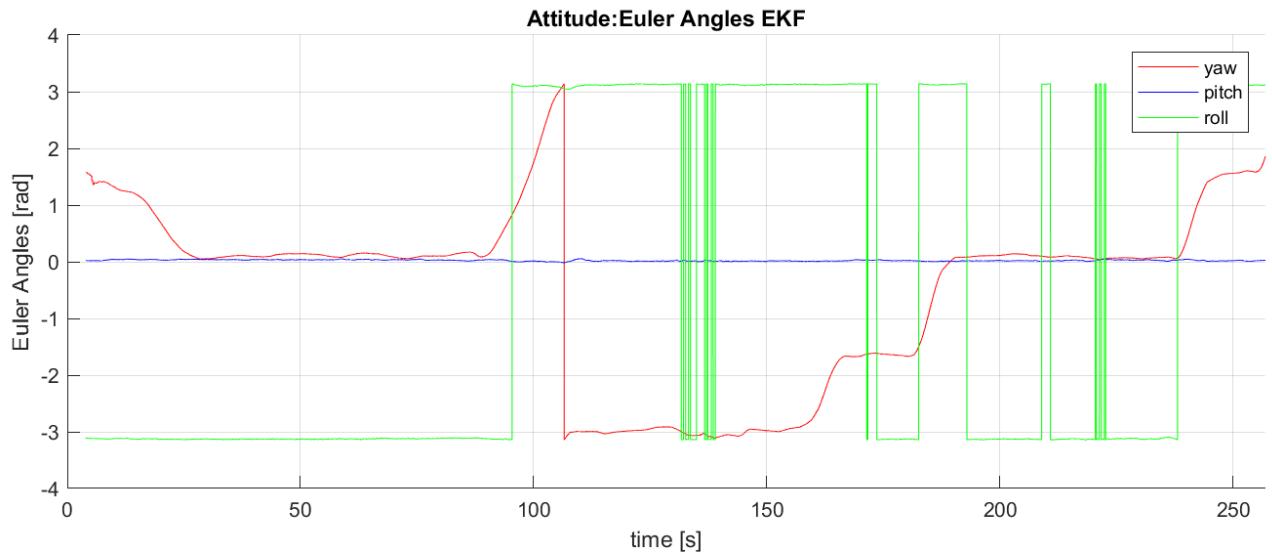


Figura 6.9: Angoli di Eulero di assetto stimati dal filtro EKF con bias

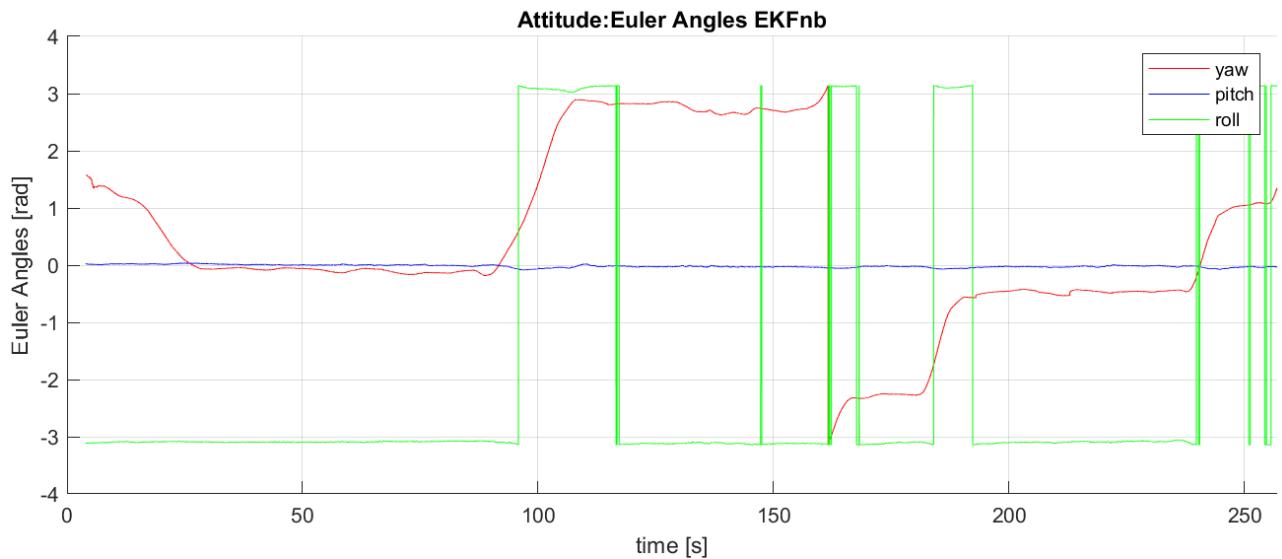


Figura 6.10: Angoli di Eulero di assetto stimati dal filtro EKF senza stima del bias

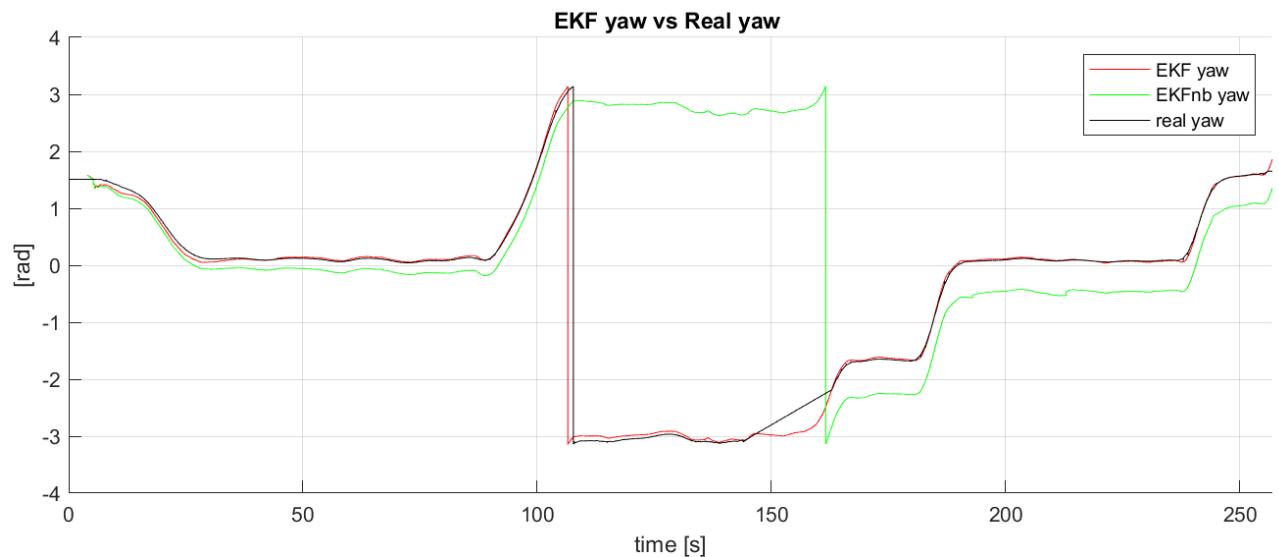


Figura 6.11: Confronto tra angoli di yaw stimati e reale

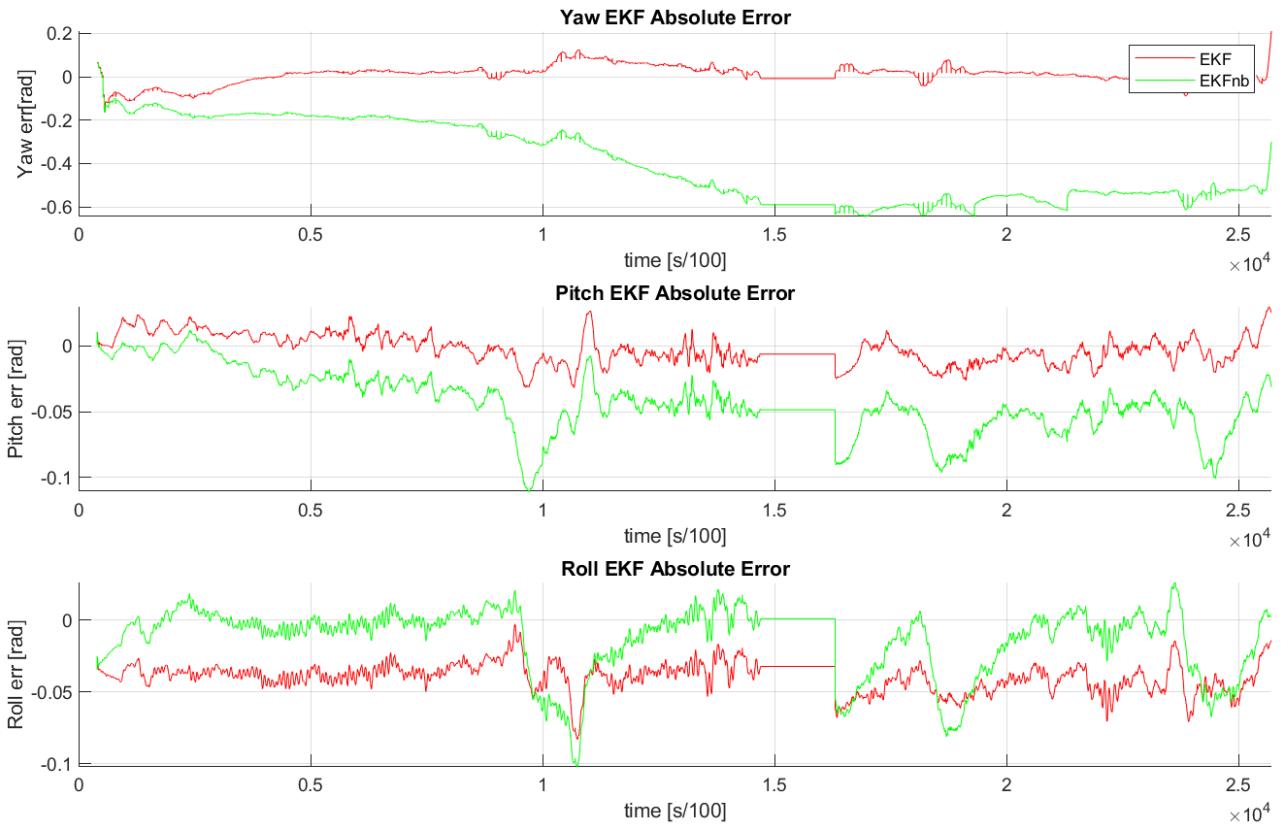


Figura 6.12: Errori assoluti degli angoli di Eulero di assetto stimati

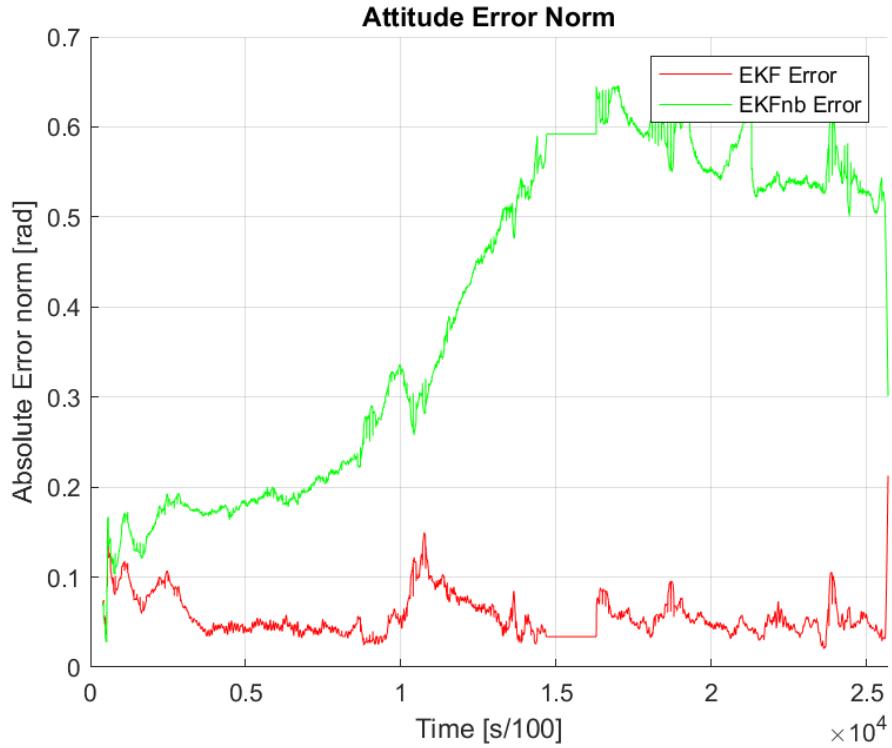


Figura 6.13: Norma dell'errore assoluto degli angoli di Eulero di assetto stimati

Le figure 6.9 e 6.10 mostrano come gli angoli di *pitch* e *roll* rimangono pressoché costanti

per tutta la simulazione, come si poteva dedurre dai vincoli dinamici del veicolo, lasciando l'asse x del frame body verso la direzione di avanzamento, l'asse z verso la superficie terrestre e l'asse y di conseguenza. L'errore assoluto rispetto l'assetto reale del veicolo, Fig. 6.12, è maggiore sulla componente di *yaw*, questo a causa principalmente della mancanza della compensazione del bias del giroscopio, e della mancanza di una correzione ad alta frequenza come quella sugli angoli di *pitch* e *roll*.

La tabella 6.3 mostra con l'indice *RMSE* questa differenza netta:  $RMSE_{tot} = 3,43 \text{ deg}$  per l'EKF con la compensazione del bias,  $RMSE_{tot} \approx RMSE_{yaw} = 24.95 \text{ deg}$  senza compensazione.

	$RMSE_{yaw}$	$RMSE_{pitch}$	$RMSE_{roll}$	$RMSE_{tot}$	Unità
EKF	0.042831	0.011174	0.040364	0.059905	rad
EKFnb	0.43200	0.048449	0.026605	0.43552	rad

Tabella 6.3: RMSE della stima dell'assetto

## 6.2 Simulazione con DGPS low-cost con interruzioni

Se invece si utilizza il DGPS low-cost, la stima peggiora notevolmente.

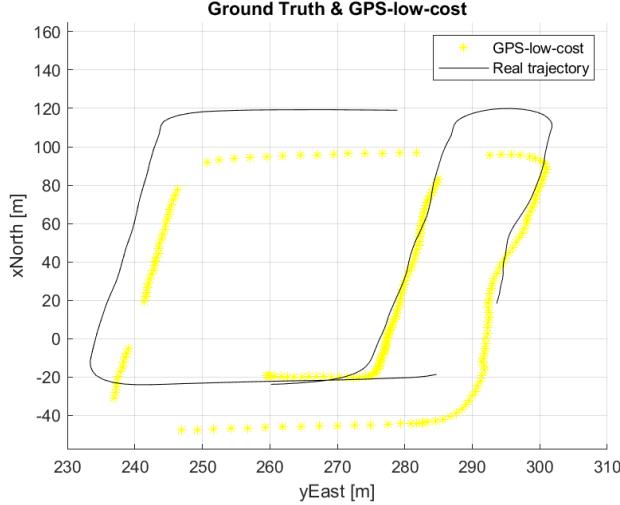
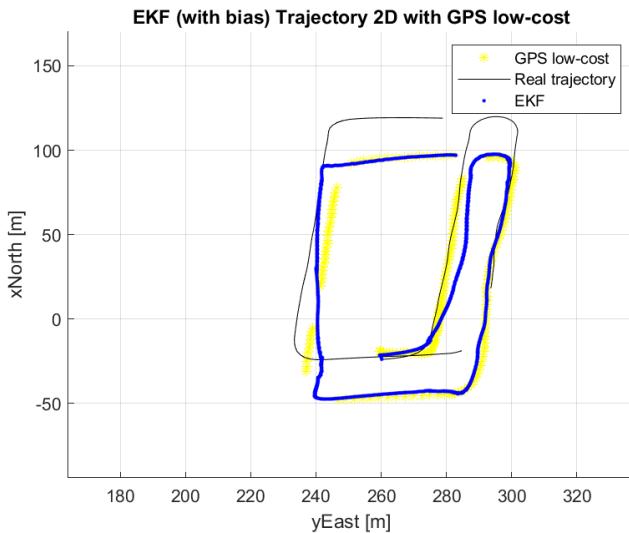


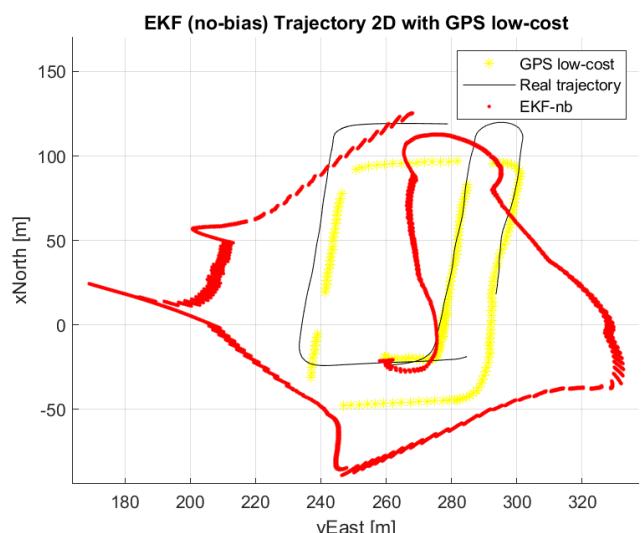
Figura 6.14: Segnale del GPS

	Valore	Unità
t1_start	90	s
t1_end	100	s
t2_start	183	s
t2_end	193	s
t3_start	203	s
t3_end	213	s
t4_start	237	s
t4_end	244	s

Tabella 6.4: Intervalli di tempo degli *outages*



(a) EKF con stima del bias



(b) EKF senza stima del bias

Figura 6.15: Confronto tra posizione stimata e posizione reale con EKF con stima del bias di accelerometro e giroscopio (a), e senza (b)

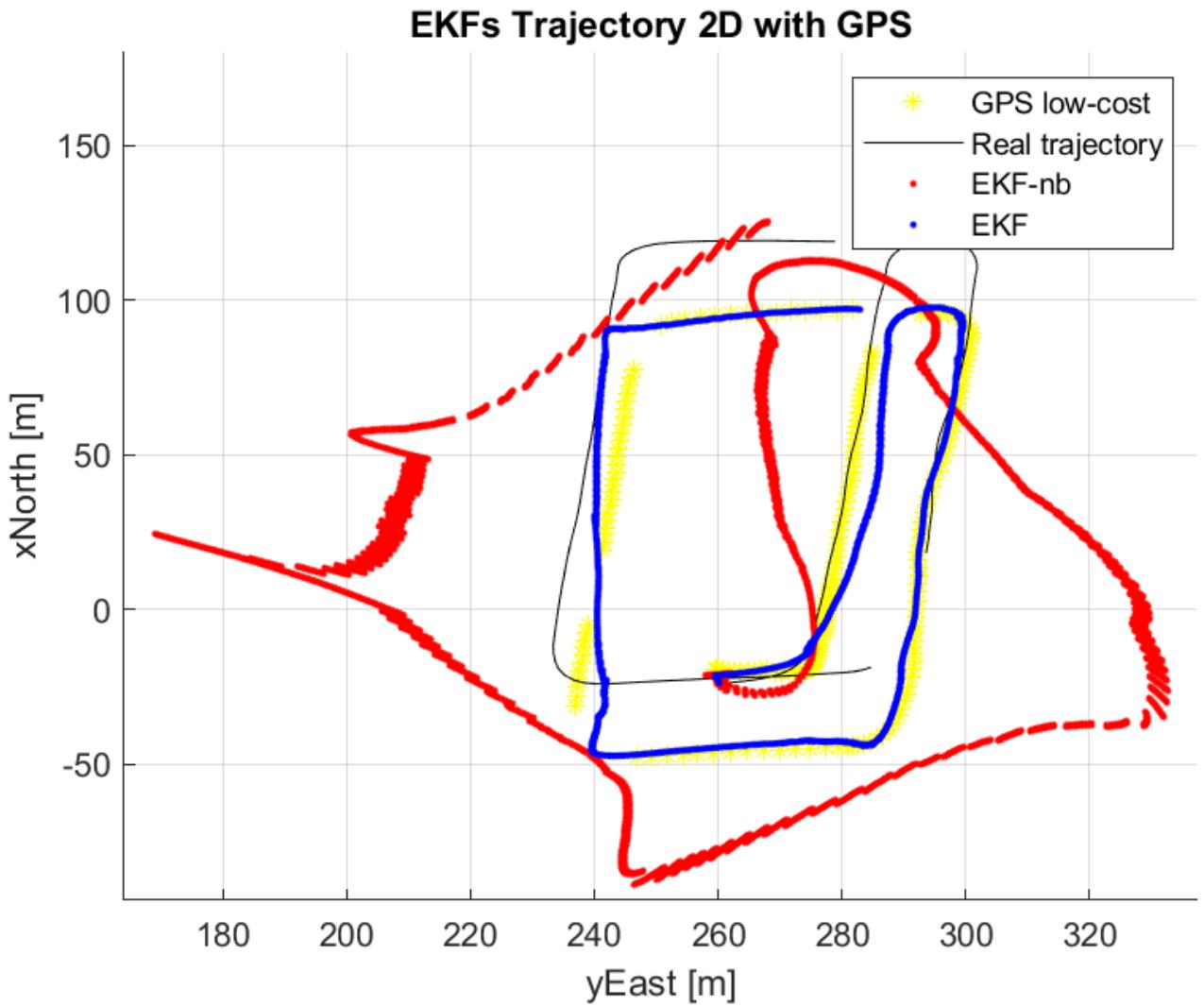


Figura 6.16: Confronto tra posizione stimata e posizione esatta

Le figure 6.15 e 6.16 mostrano risultati ben diverse rispetto al caso precedente. Nel caso di compensazione dei bias la stima di posizione è sicuramente più accurata, ma segue le misure del DGPS che non sono più precise quanto quelle del GPS RTK. Nel caso in cui si ignorino i bias, la stima della posizione peggiora ancora di più in quanto, come vedremo con i grafici relativi all'assetto, il filtro non riesce più a recuperare un'adeguata stima dell'angolo di  $yaw$ .

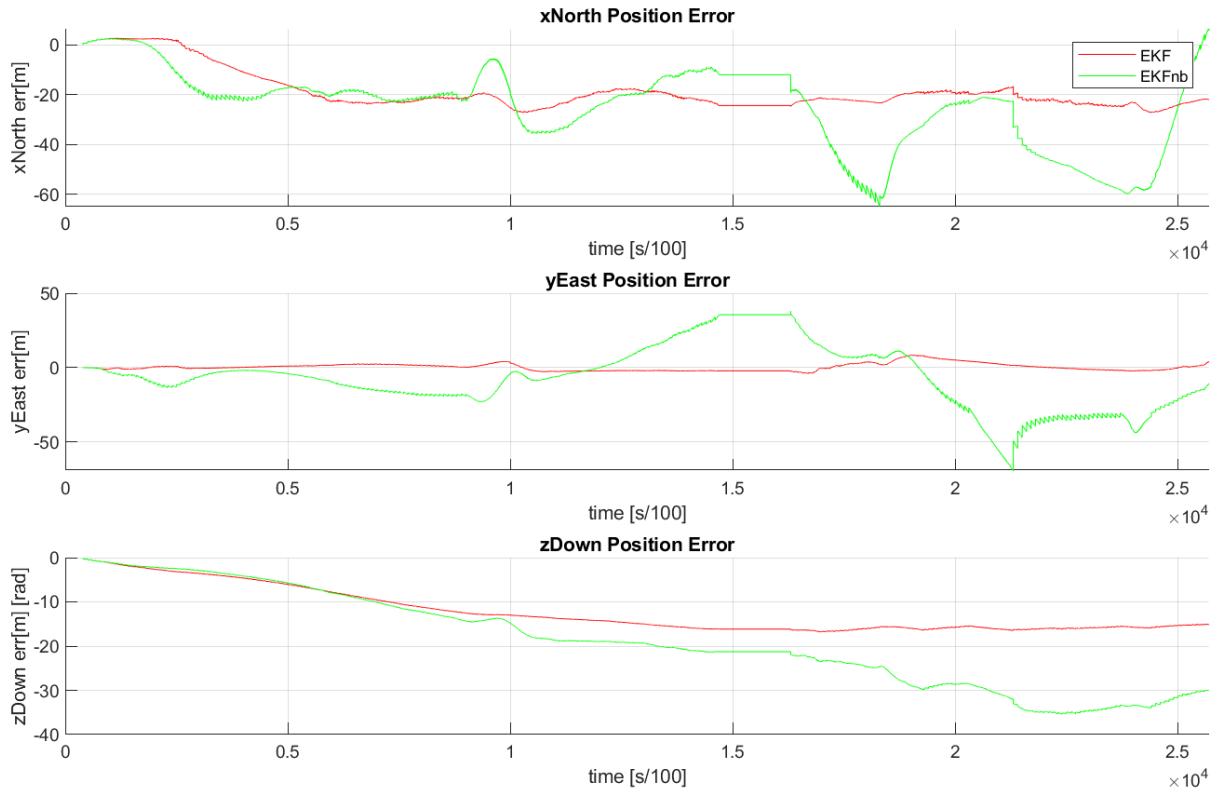


Figura 6.17: Errori assoluti delle componenti della stima di posizione

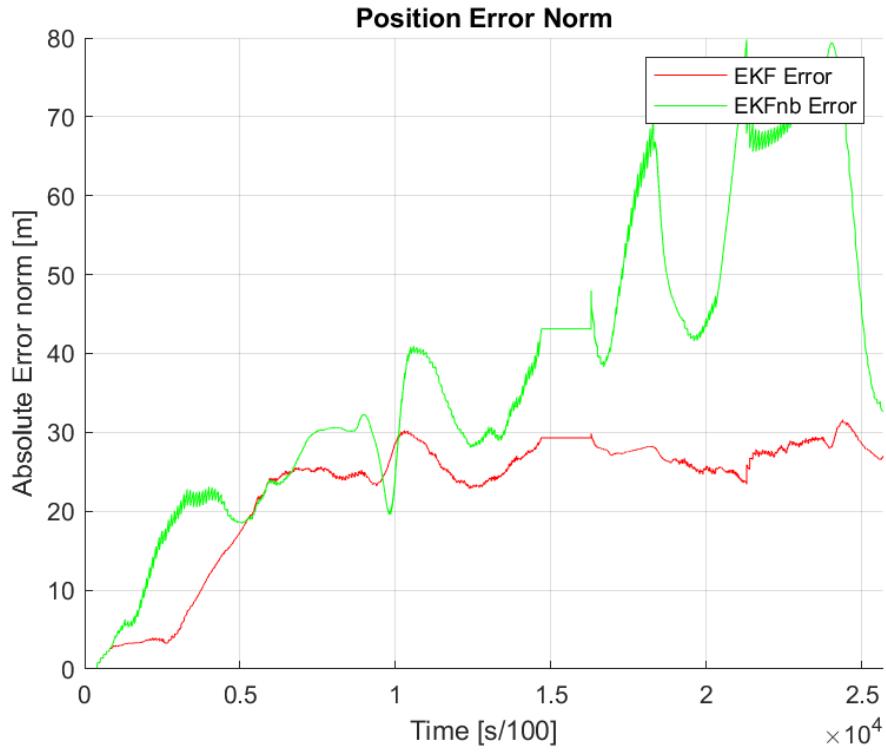


Figura 6.18: Norma dell'errore assoluto della stima di posizione

Dalla figura 6.17 si può notare che anche la componente *zDown* risente delle misure meno precise del DGPS low-cost a differenza della simulazione precedente. Infine, la tabella 6.5

	$RMSE_x$	$RMSE_y$	$RMSE_z$	$RMSE_{tot}$	Unità
EKF	20.08	2.4585	13.093	24.101	$m$
EKFnb	29.196	22.107	21.343	42.386	$m$

Tabella 6.5: RMSE della stima di posizione

indica un errore non trascurabile presente in entrambi i casi, dovuto principalmente alla scarsa accuratezza del GPS.

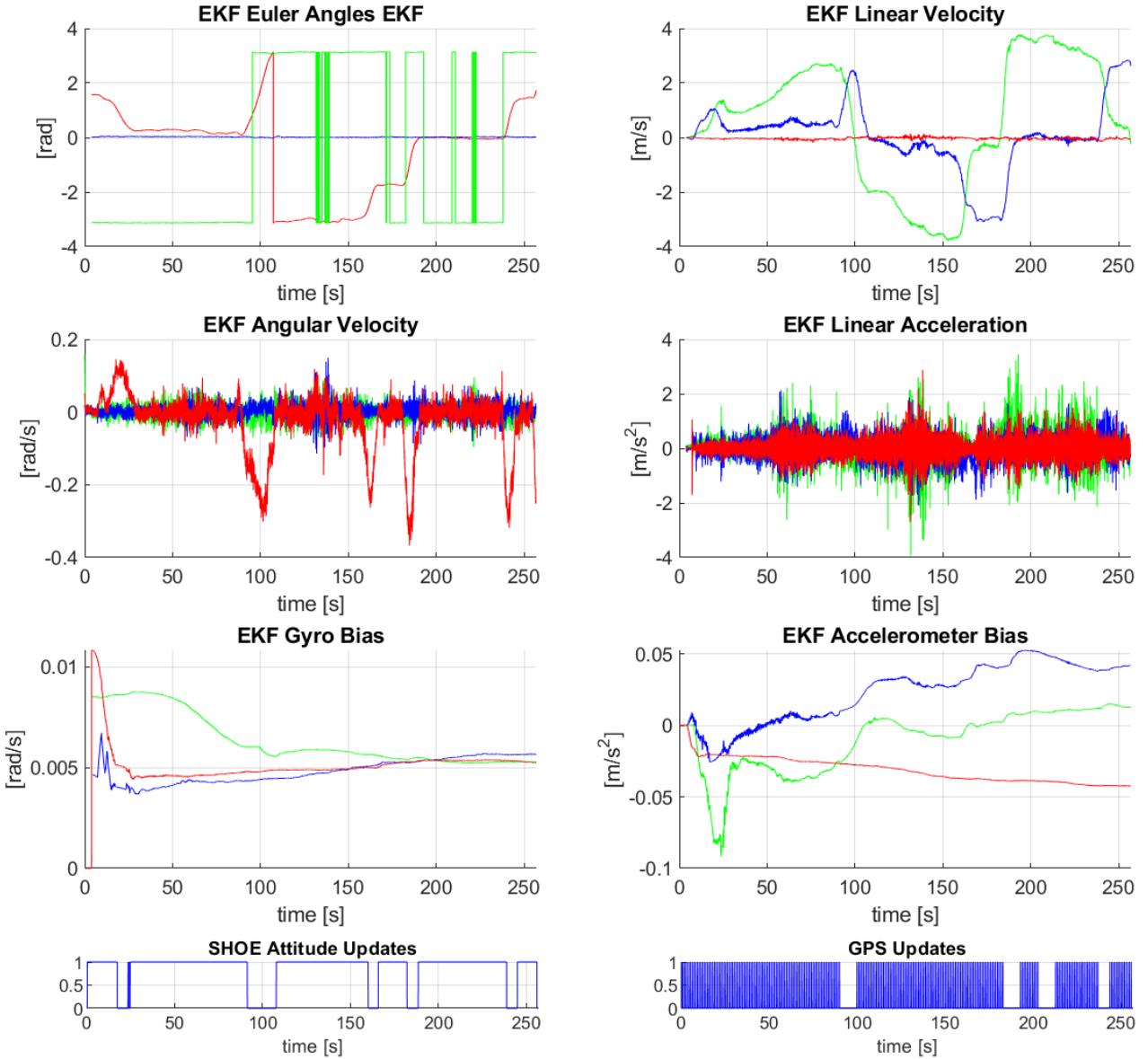


Figura 6.19: Variabili stimate dall'EKF con bias: assetto [yaw,pitch,roll], velocità lineare, velocità angolare, accelerazione e bias con componenti [Z,Y,X], i grafici in basso indicano quando le correzioni di assetto e del GPS avvengono.

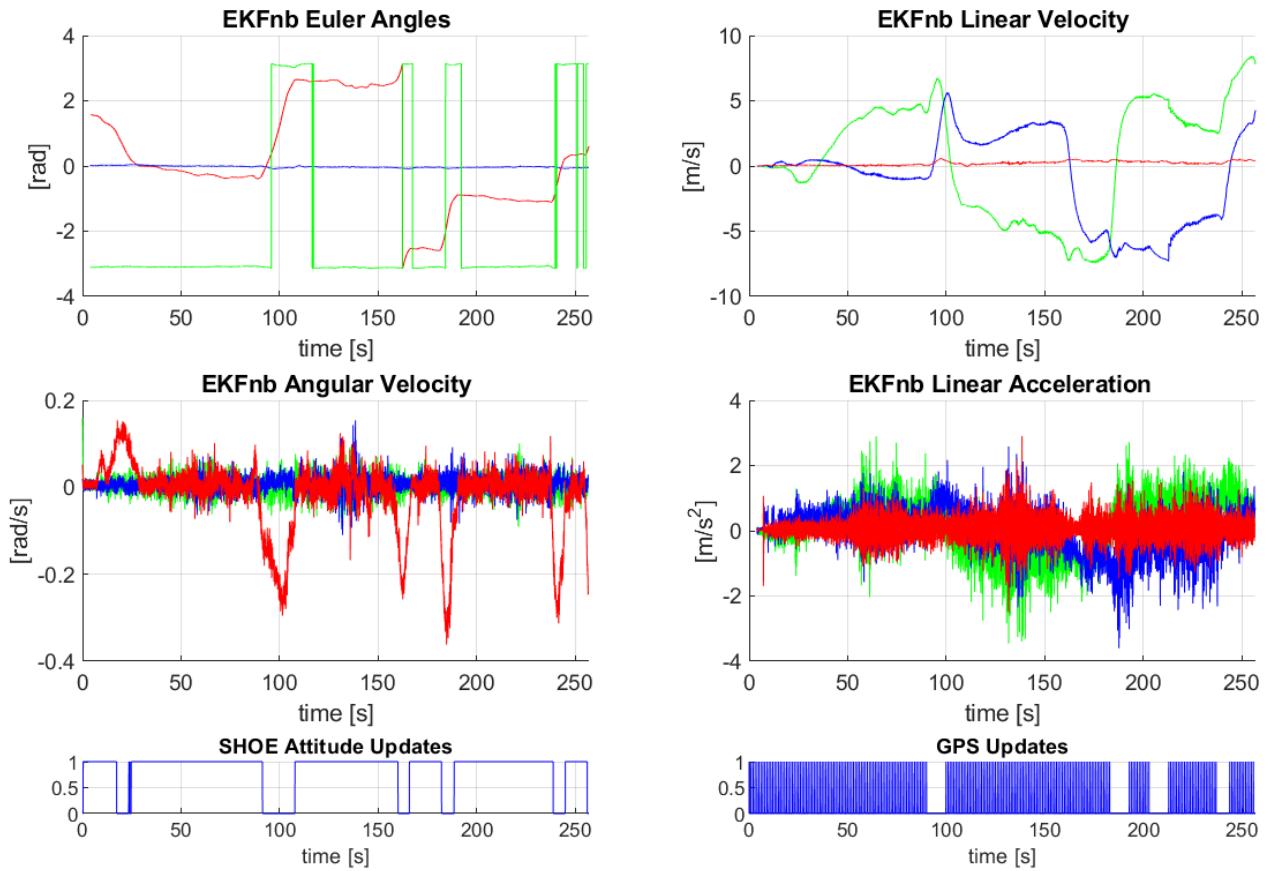


Figura 6.20: Variabili stimate dall'EKF con bias: assetto [yaw,pitch,roll], velocità lineare, velocità angolare, accelerazione con componenti [Z,Y,X], i grafici in basso indicano quando le correzioni di assetto e del GPS avvengono.

Per quanto riguarda l'assetto in particolare:

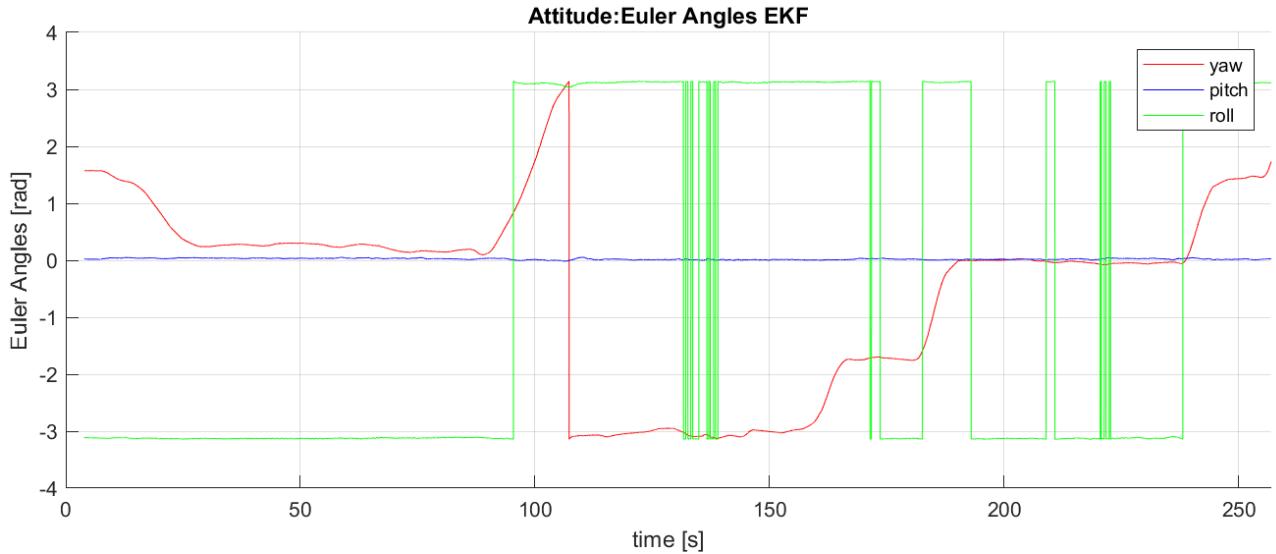


Figura 6.21: Angoli di Eulero di assetto stimati dal filtro EKF con bias

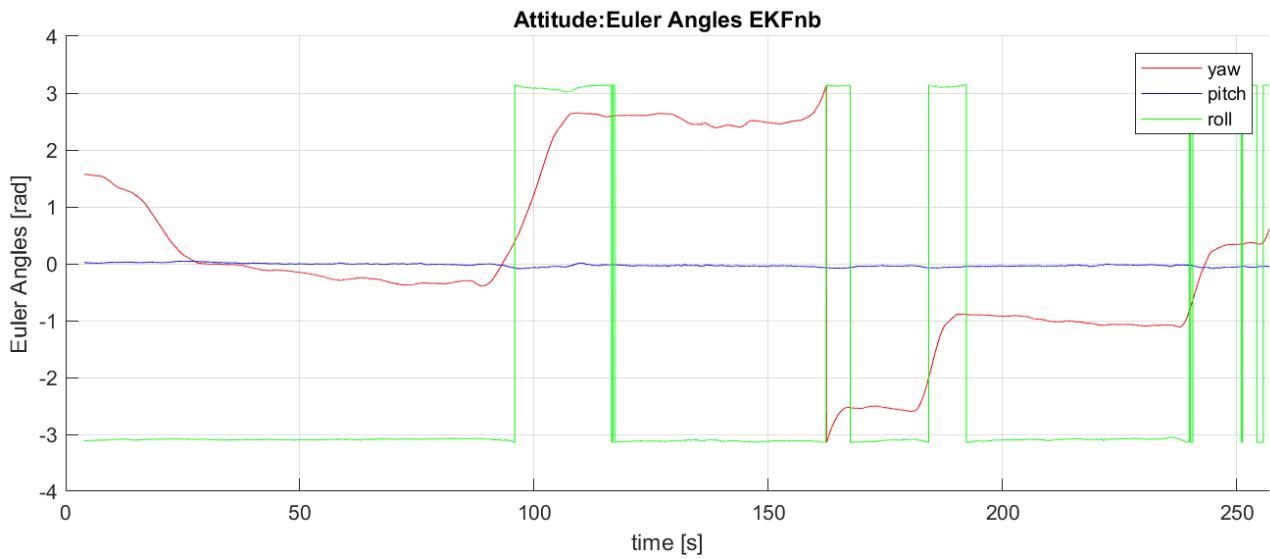


Figura 6.22: Angoli di Eulero di assetto stimati dal filtro EKF senza stima del bias

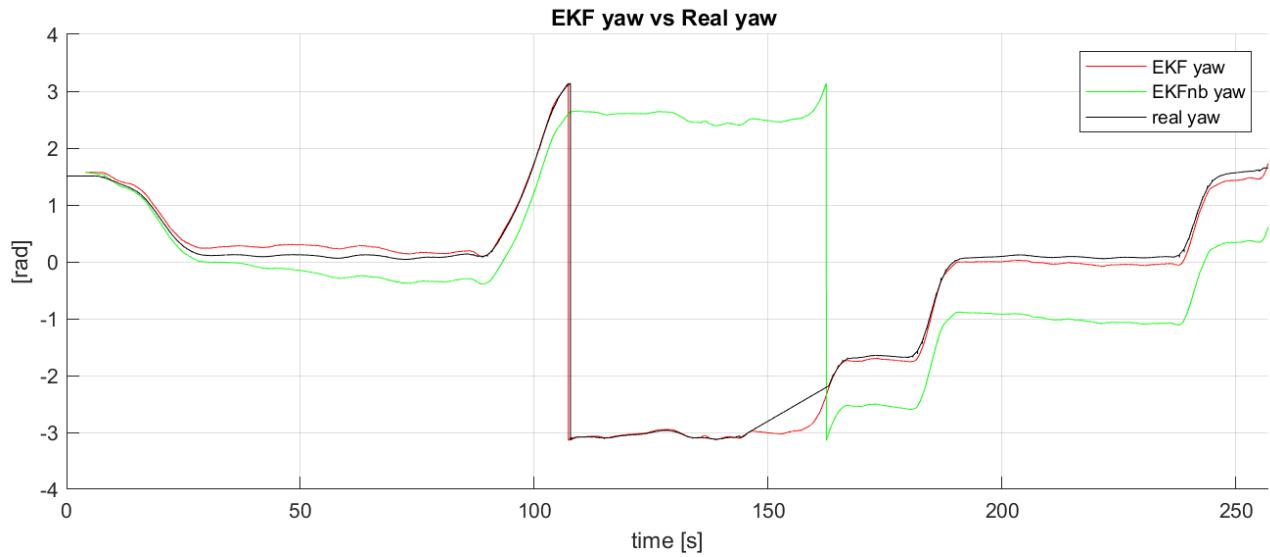


Figura 6.23: Confronto tra angoli di yaw stimati e reale

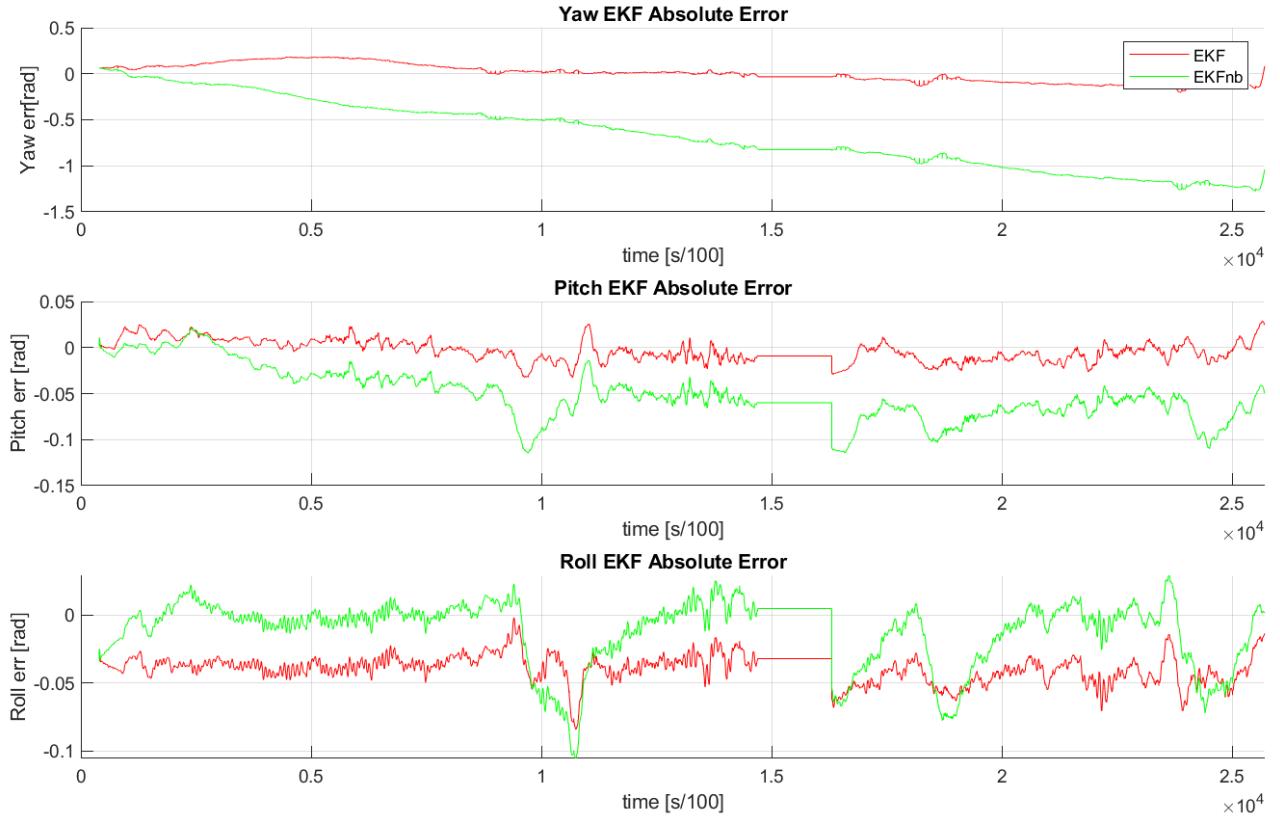


Figura 6.24: Errori assoluti degli angoli di Eulero di assetto stimati

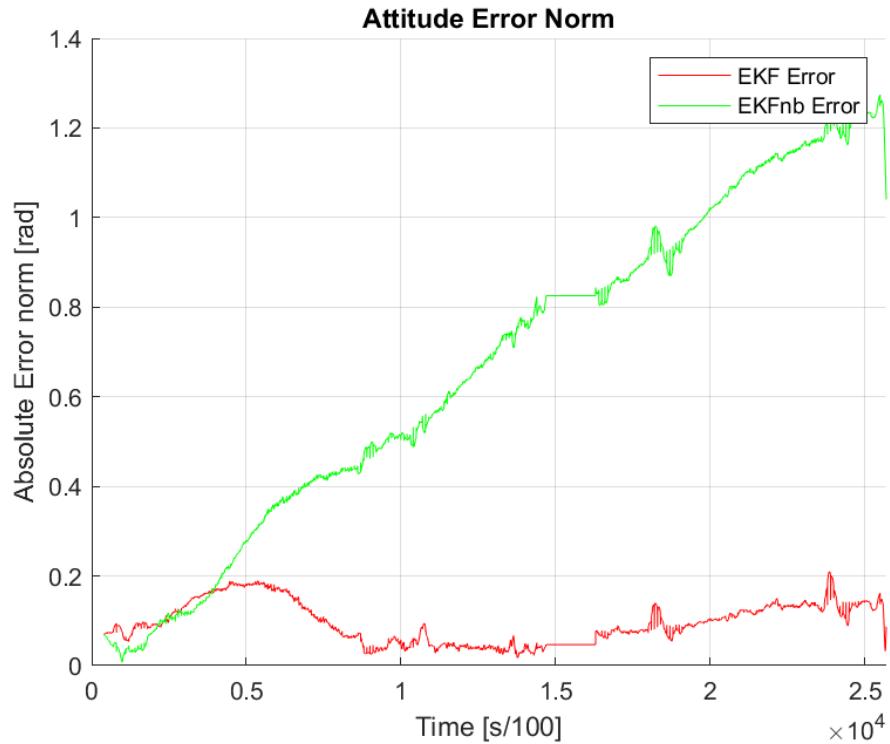


Figura 6.25: Norma dell'errore assoluto degli angoli di Eulero di assetto stimati

Come già anticipato, la stima dell'assetto del filtro senza stima dei bias risente notevolmente anche delle misure imprecise del DGPS low-cost, soprattutto l'angolo di *yaw*: la figura 6.24

mostra l'andamento nel tempo dell'errore assoluto che supera persino  $1rad = 57.3$  deg; la tabella 6.6, riporta l'indice  $RMSE_{yaw} \approx RMSE_{tot} = 43.37$  deg. Diversamente invece la stima degli angoli di Eulero del filtro completo restano piuttosto accurati.

	$RMSE_{yaw}$	$RMSE_{pitch}$	$RMSE_{roll}$	$RMSE_{tot}$	Unità
EKF	0.096583	0.011864	0.040384	0.10535	<i>rad</i>
EKFnb	0.7543	0.057588	0.026778	0.75697	<i>rad</i>

Tabella 6.6: RMSE della stima dell'assetto

### 6.3 Simulazione con GPS RTK

In questa terza simulazione si utilizzerà il GPS RTK senza interruzioni di segnale.

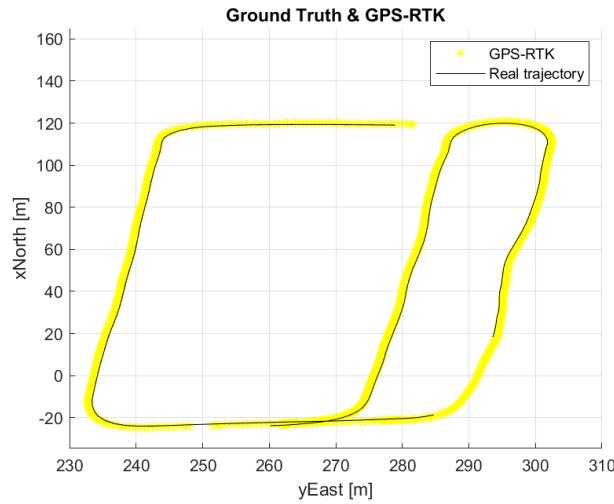


Figura 6.26: Segnale del GPS

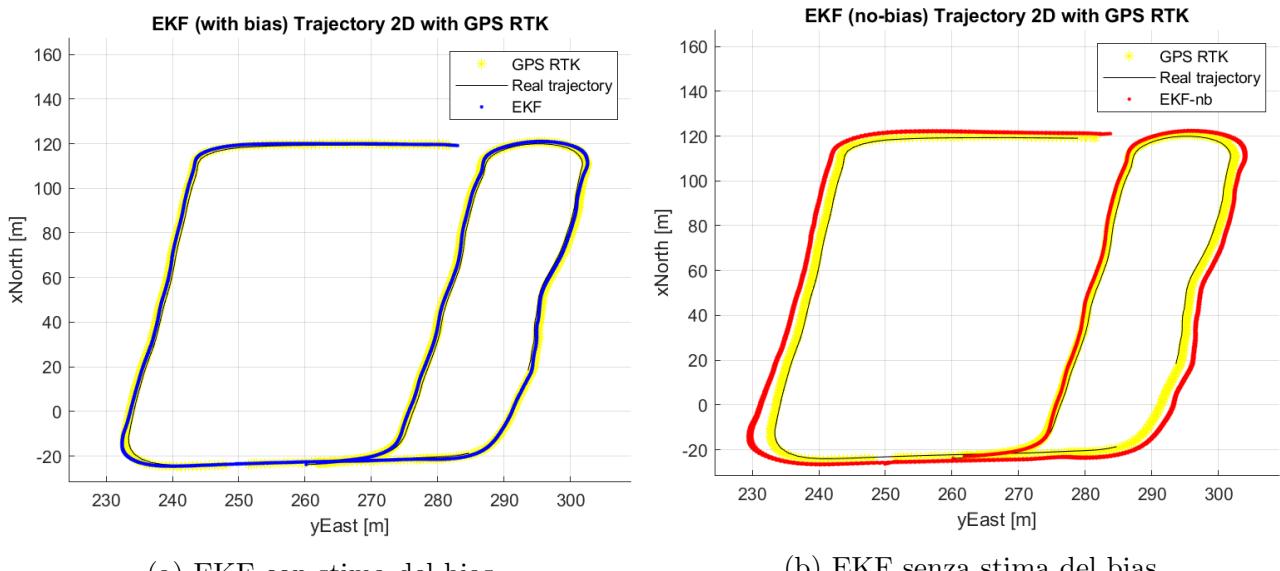


Figura 6.27: Confronto tra posizione stimata e posizione reale con EKF con stima del bias di accelerometro e giroscopio (a), e senza (b)

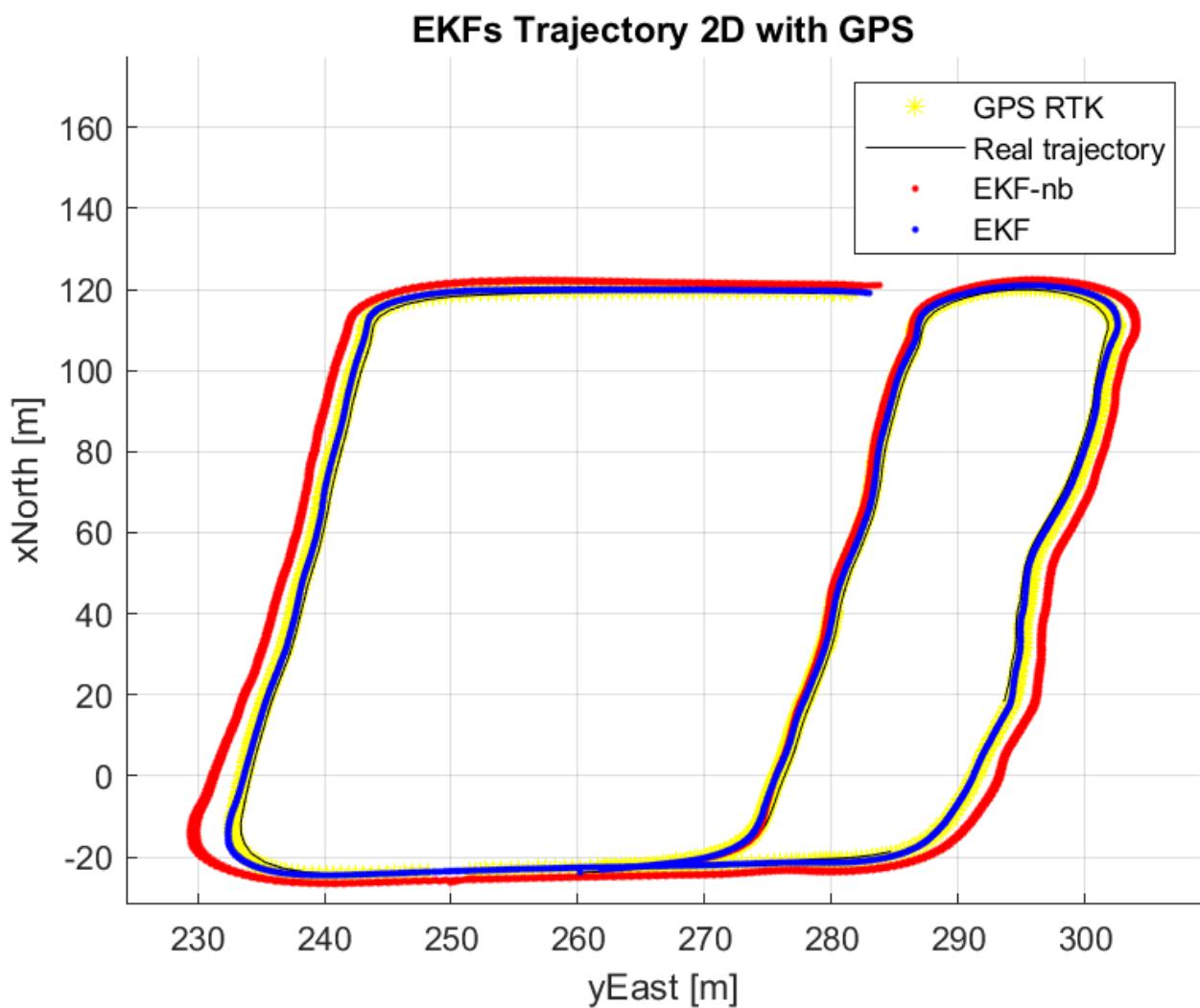


Figura 6.28: Confronto tra posizione stimata e posizione esatta

Questo è il caso più favorevole e la stima di posizione risulta essere ragionevolmente accurata in entrambi i casi.

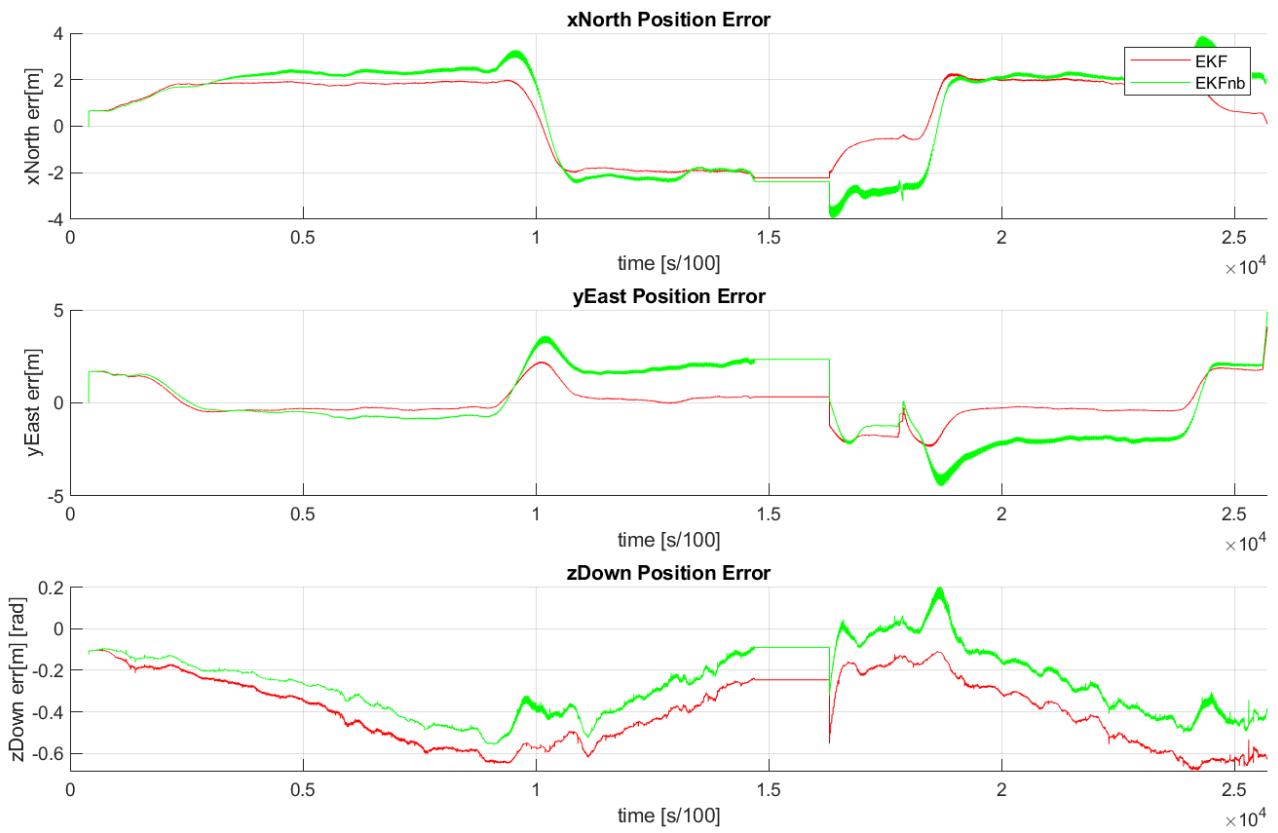


Figura 6.29: Errori assoluti delle componenti della stima di posizione

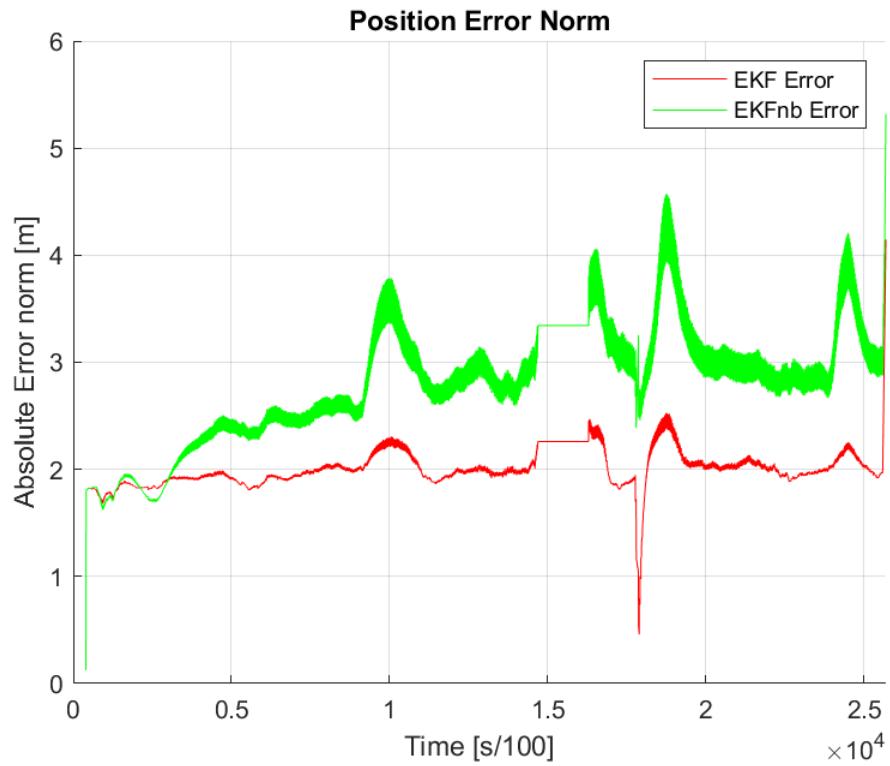


Figura 6.30: Norma dell'errore assoluto della stima di posizione

	$RMSE_x$	$RMSE_y$	$RMSE_z$	$RMSE_{tot}$	Unità
EKF	1.7211	0.93214	0.41794	2.0014	$m$
EKFnb	2.2057	1.7648	0.29924	2.8407	$m$

Tabella 6.7: RMSE della stima di posizione

Come si evince dalla figura 6.29 l'errore assoluto nelle tre componenti è notevolmente ridotto. Il filtro senza stima dei bias può appoggiarsi sulle misure sempre disponibili del GPS RTK che in questo caso monopolizza la stima della posizione data la sua accuratezza. Gli indici  $RMSE$  in tabella 6.7 risultano simili a quelli del segnale GPS confrontato con la posizione esatta in tabella 4.1.

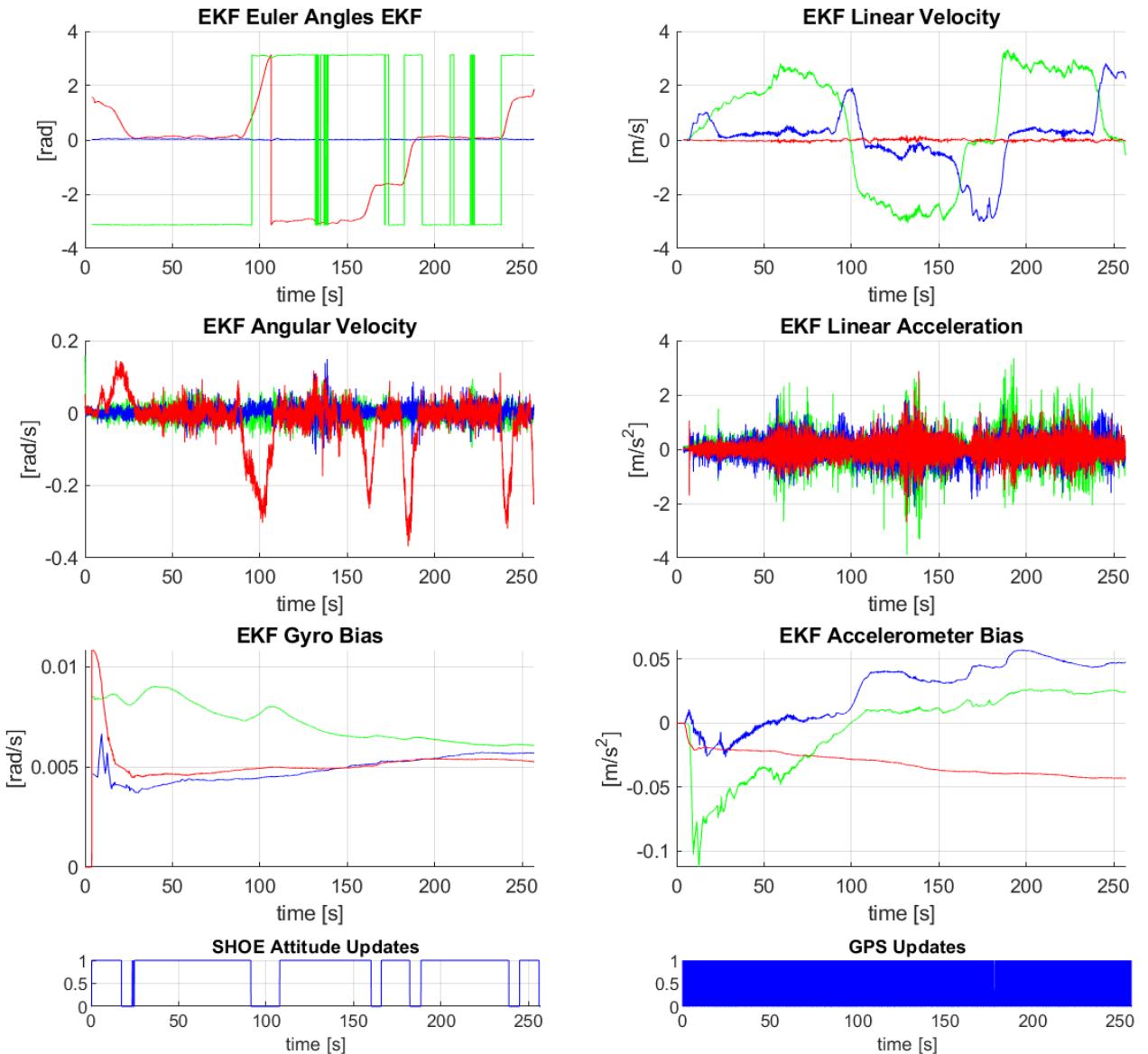


Figura 6.31: Variabili stimate dall'EKF con bias: assetto [yaw,pitch,roll], velocità lineare, velocità angolare, accelerazione e bias con componenti [Z,Y,X], i grafici in basso indicano quando le correzioni di assetto e del GPS avvengono.

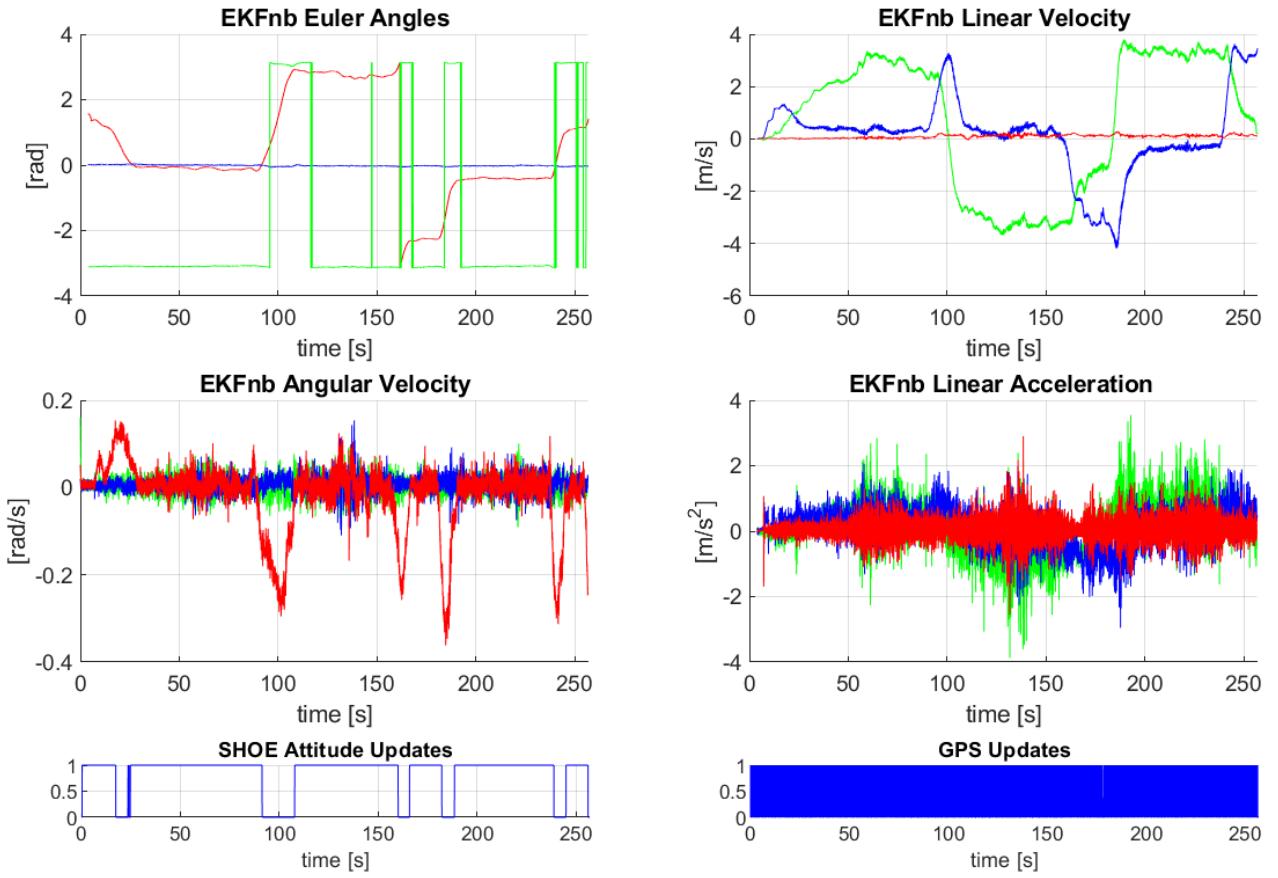


Figura 6.32: Variabili stimate dall'EKF con bias: assetto [yaw,pitch,roll], velocità lineare, velocità angolare, accelerazione con componenti [Z,Y,X], i grafici in basso indicano quando le correzioni di assetto e del GPS avvengono.

Per quanto riguarda l'assetto in particolare:

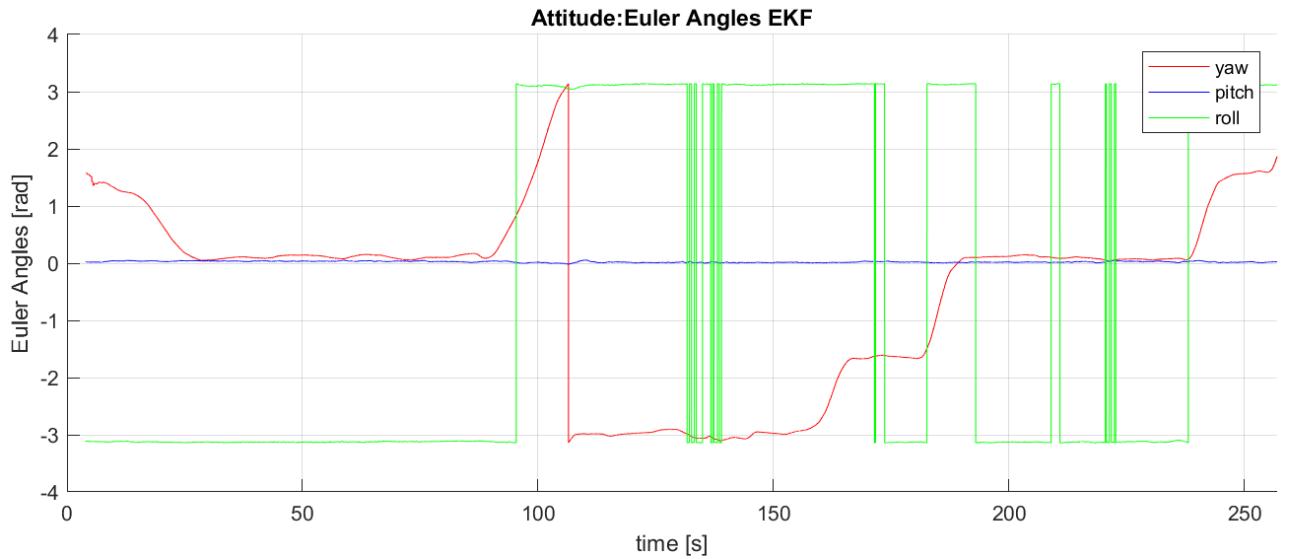


Figura 6.33: Angoli di Eulero di assetto stimati dal filtro EKF con bias

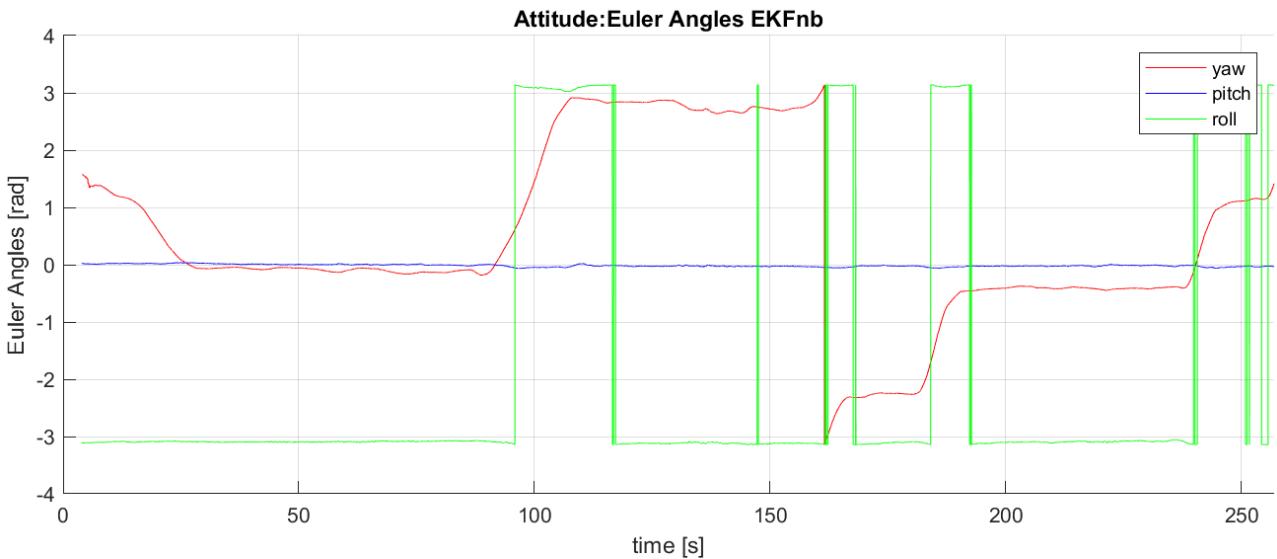


Figura 6.34: Angoli di Eulero di assetto stimati dal filtro EKF senza stima del bias

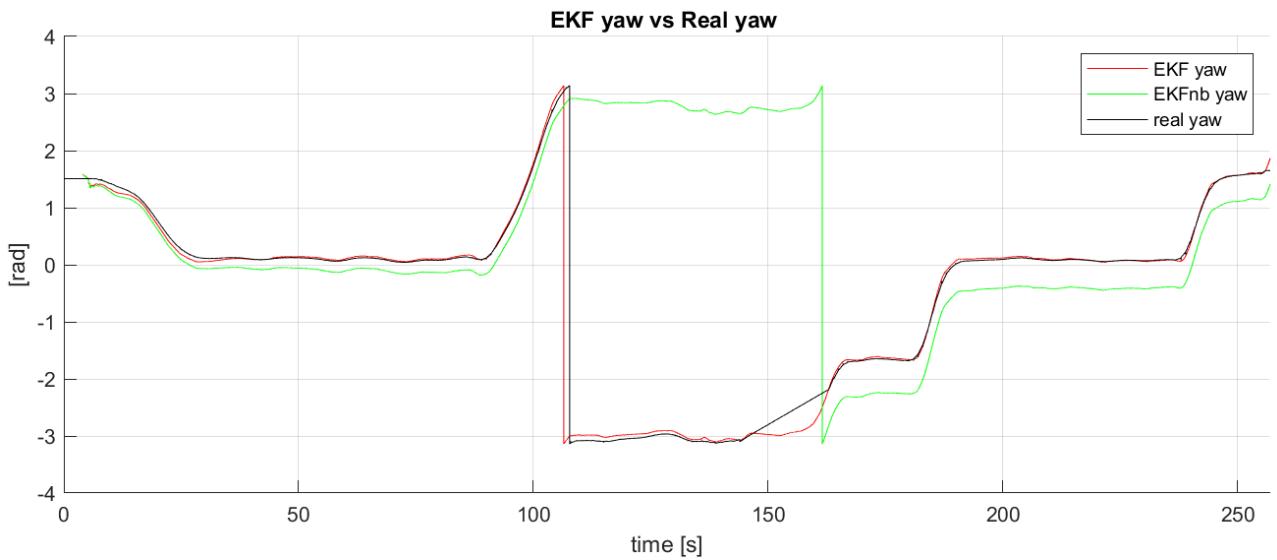


Figura 6.35: Confronto tra angoli di yaw stimati e reale

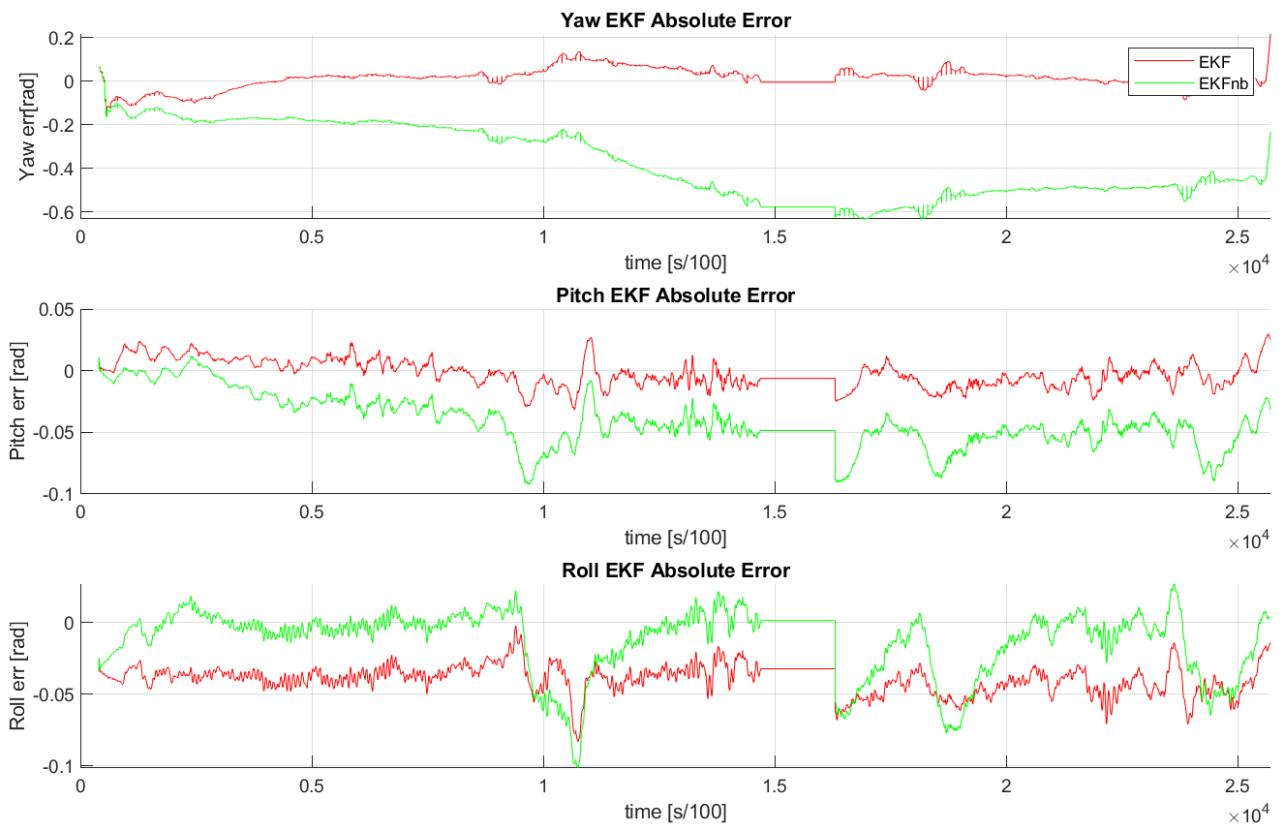


Figura 6.36: Errori assoluti degli angoli di Eulero di assetto stimati

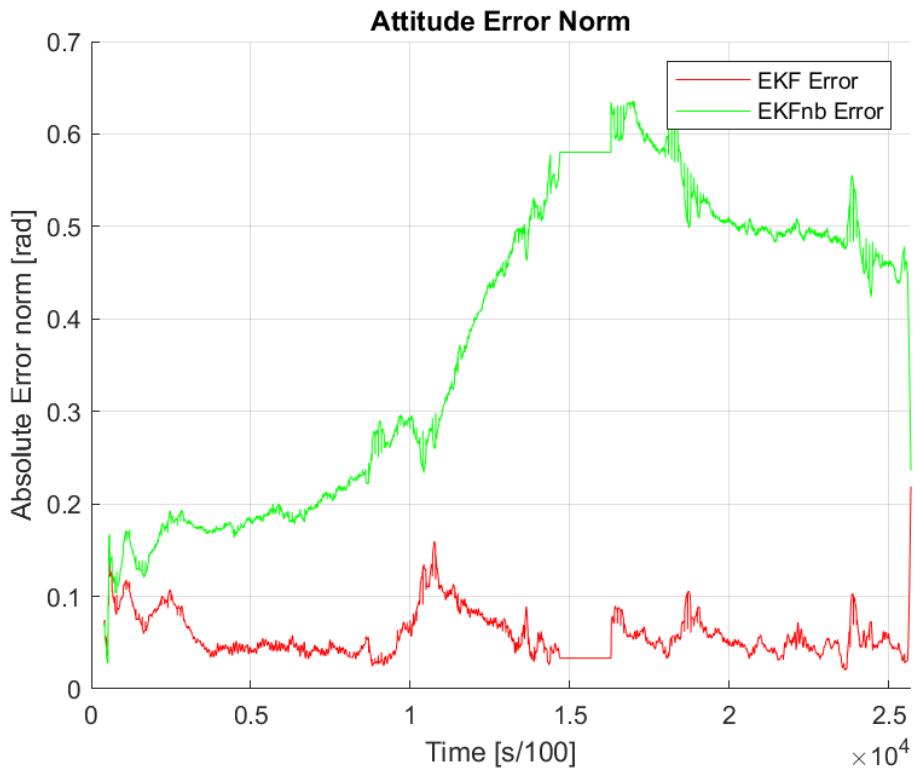


Figura 6.37: Norma dell'errore assoluto degli angoli di Eulero di assetto stimati

Anche in questo caso ignorare il bias del giroscopio porta ad un errata stima dell'angolo di  $yaw$ . Questo caso è in teoria il più favorevole tra tutti, ma è interessante notare come gli indici  $RMSE$  risultano confrontabili con la prima simulazione (tabella 6.3): le interruzioni del GPS influenzano meno rispetto la compensazione dei bias.

	$RMSE_{yaw}$	$RMSE_{pitch}$	$RMSE_{roll}$	$RMSE_{tot}$	Unità
EKF	0.046400	0.010797	0.040369	0.062443	<i>rad</i>
EKFnb	0.40491	0.045537	0.025972	0.40829	<i>rad</i>

Tabella 6.8: RMSE della stima dell'assetto

## 6.4 Simulazione con DGPS low-cost

In questa ultima simulazione si utilizzerà il DGPS low-cost senza interruzioni di segnale.

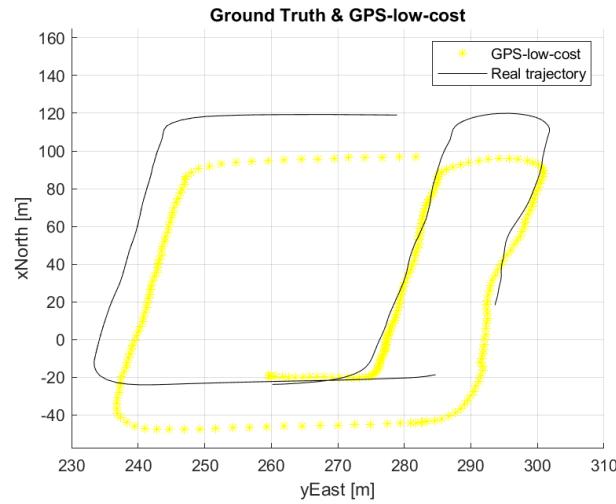


Figura 6.38: Segnale del GPS

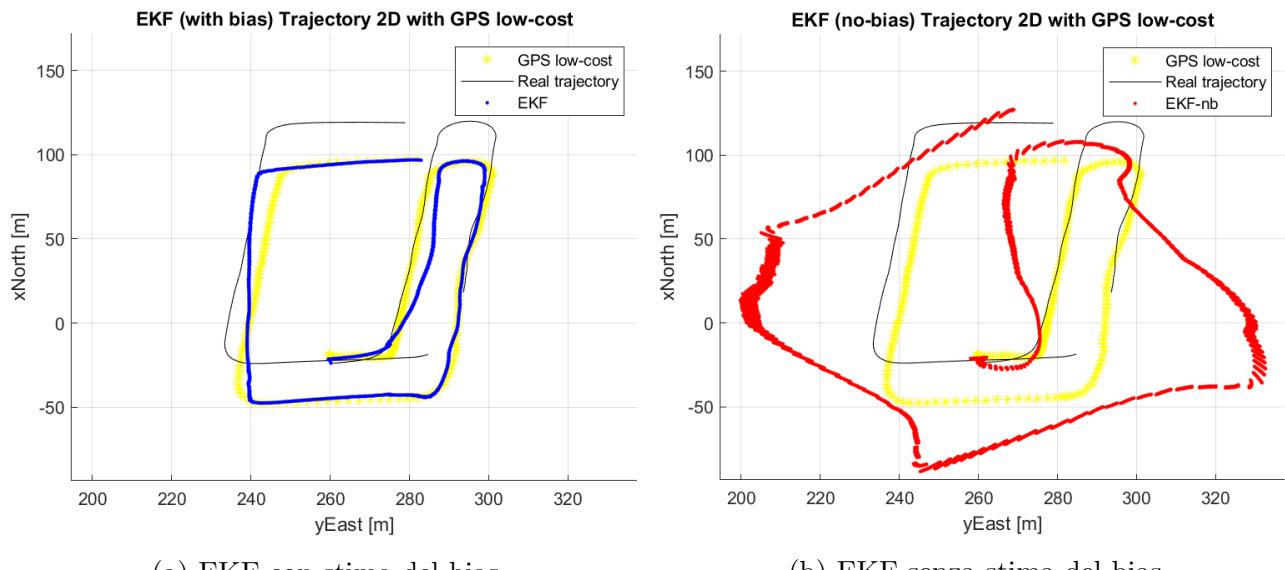


Figura 6.39: Confronto tra posizione stimata e posizione reale con EKF con stima del bias di accelerometro e giroscopio (a), e senza (b)

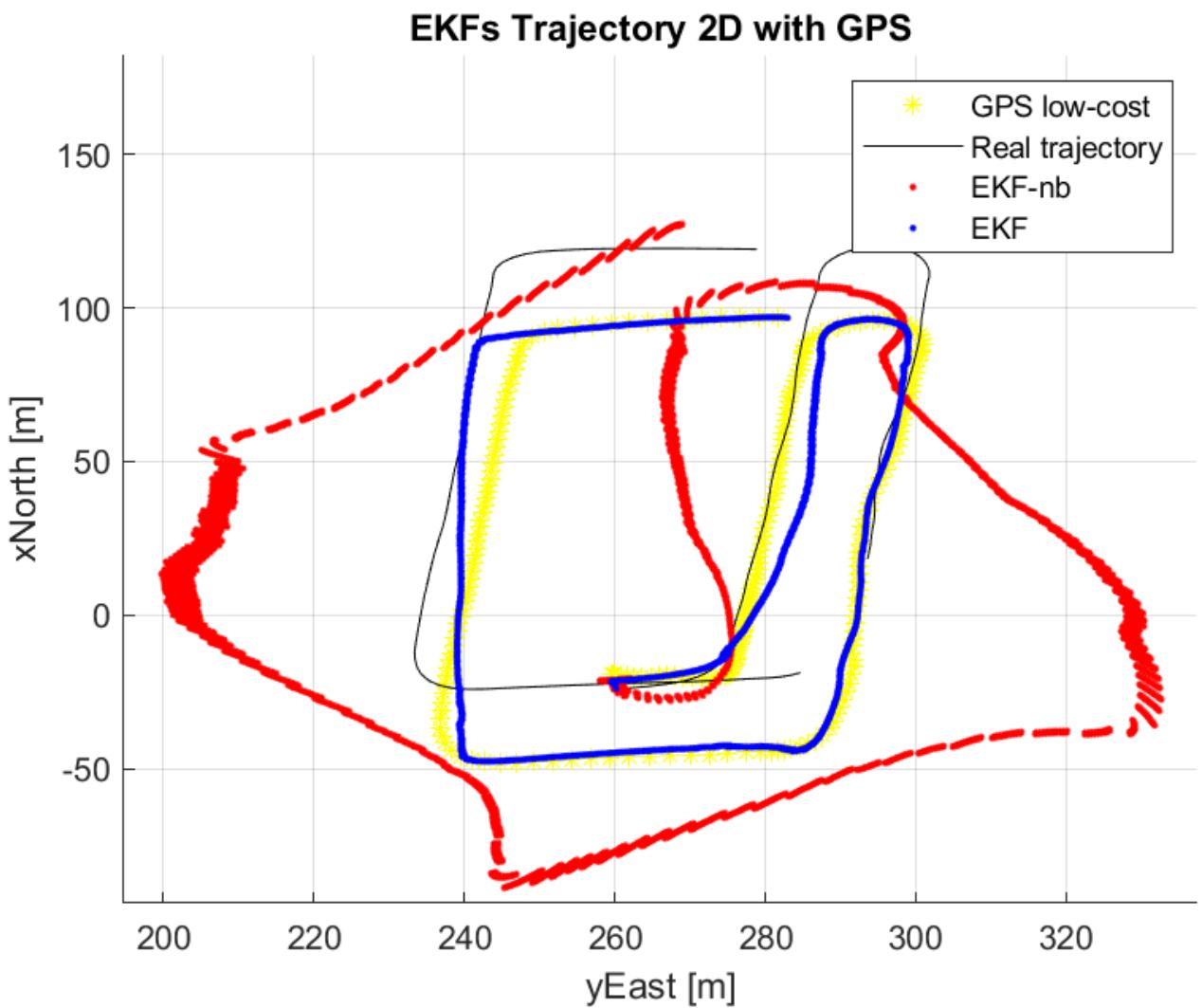


Figura 6.40: Confronto tra posizione stimata e posizione esatta

Le figure 6.39 e 6.40 mostrano come la stima di posizione risente pesantemente della mancanza della stima dei bias nel caso in cui il GPS ha un'accuratezza minore, proprio come nella seconda simulazione (Fig. 6.16).

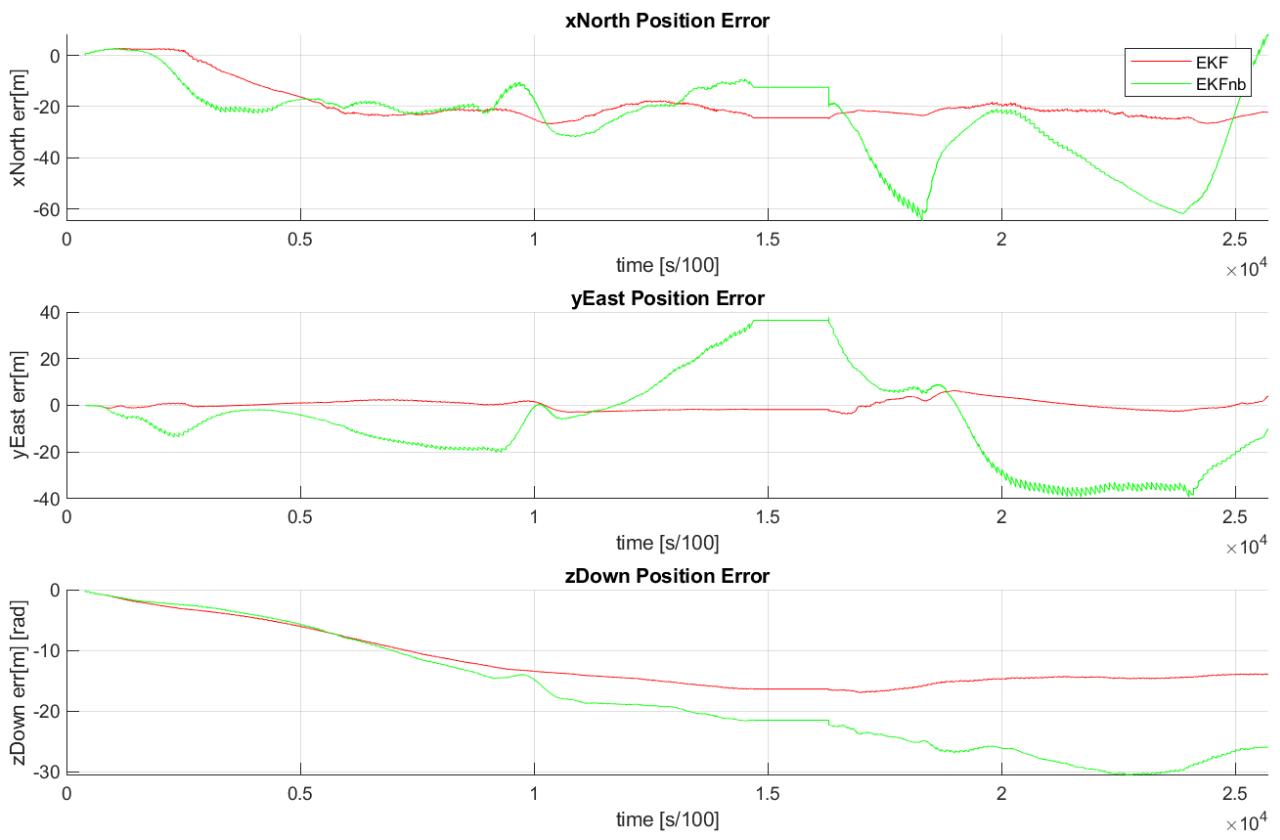


Figura 6.41: Errori assoluti delle componenti della stima di posizione

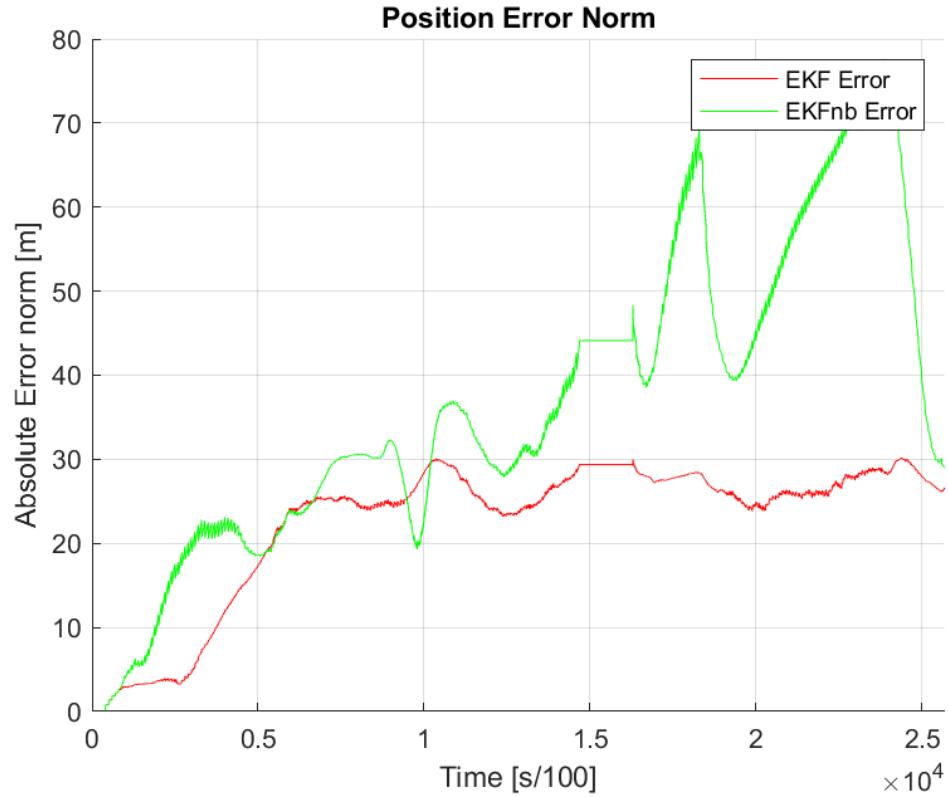


Figura 6.42: Norma dell'errore assoluto della stima di posizione

	$RMSE_x$	$RMSE_y$	$RMSE_z$	$RMSE_{tot}$	Unità
EKF	20.276	2.0767	12.770	24.052	$m$
EKFnb	28.921	21.052	19.818	40.894	$m$

Tabella 6.9: RMSE della stima di posizione

Anche gli indici  $RMSE$  in questo caso (tabella 6.9) sono simili alla stessa simulazione con l'aggiunta delle interruzioni del GPS (tabella 6.5), deducendo che appunto non sono queste la causa principale dell'errore di stima.

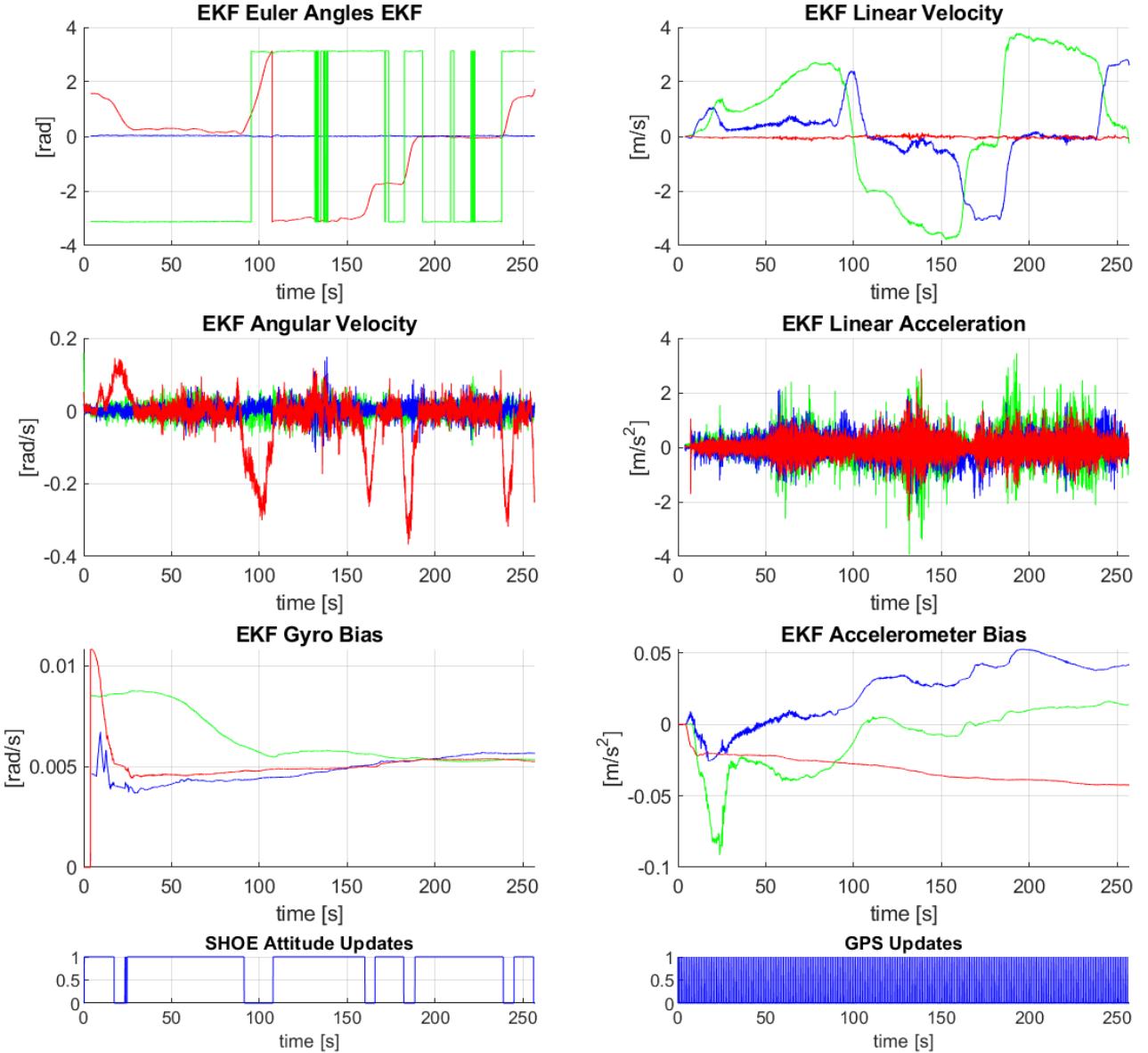


Figura 6.43: Variabili stimate dall'EKF con bias: assetto [yaw,pitch,roll], velocità lineare, velocità angolare, accelerazione e bias con componenti [Z,Y,X], i grafici in basso indicano quando le correzioni di assetto e del GPS avvengono.

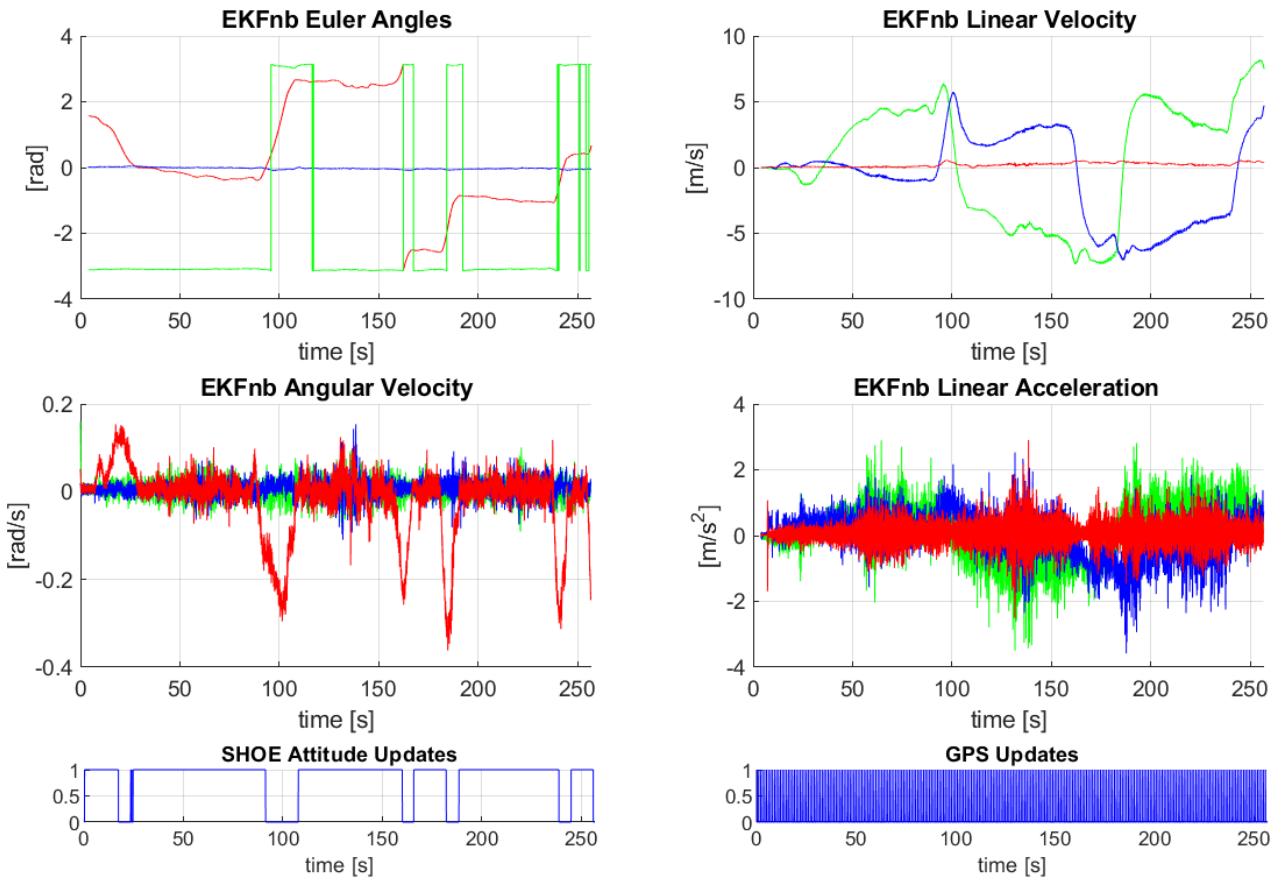


Figura 6.44: Variabili stimate dall'EKF con bias: assetto [yaw,pitch,roll], velocità lineare, velocità angolare, accelerazione con componenti [Z,Y,X], i grafici in basso indicano quando le correzioni di assetto e del GPS avvengono.

Per quanto riguarda l'assetto in particolare:

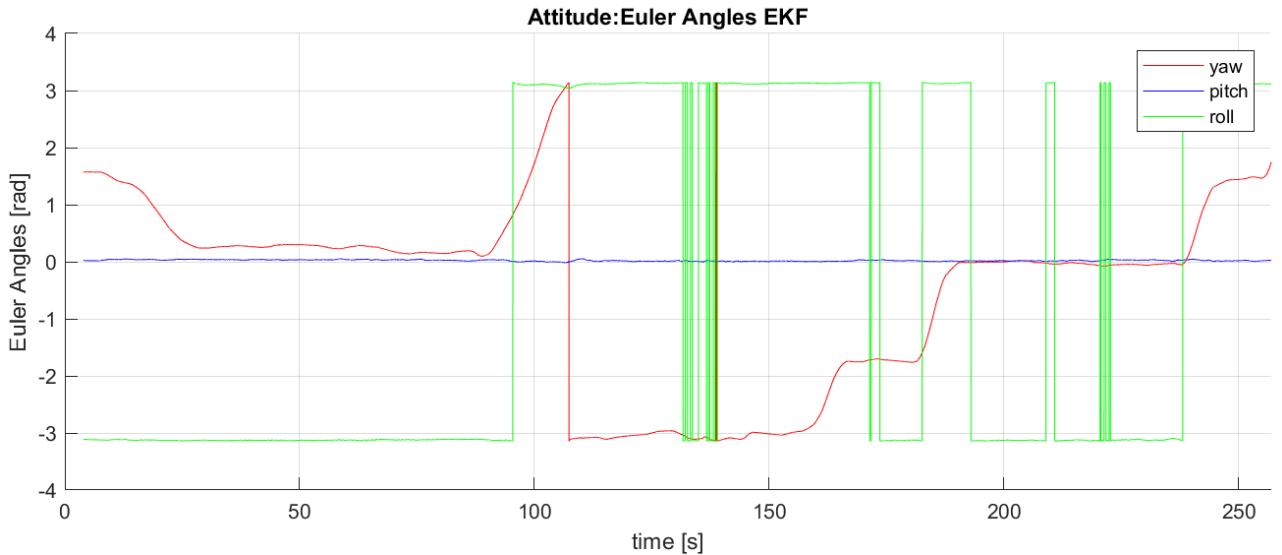


Figura 6.45: Angoli di Eulero di assetto stimati dal filtro EKF con bias

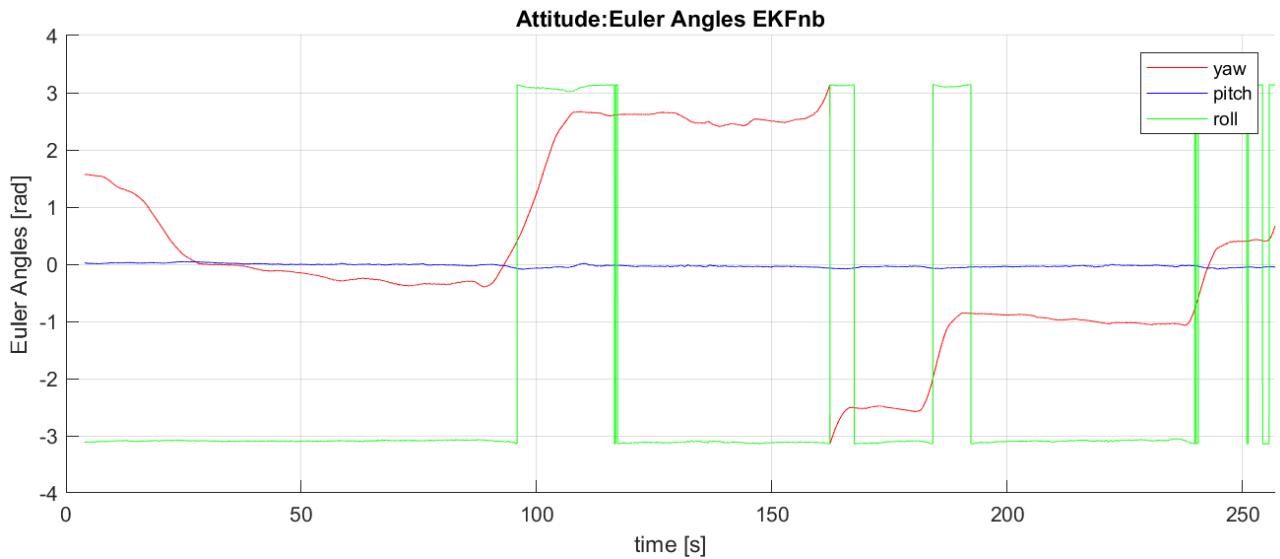


Figura 6.46: Angoli di Eulero di assetto stimati dal filtro EKF senza stima del bias

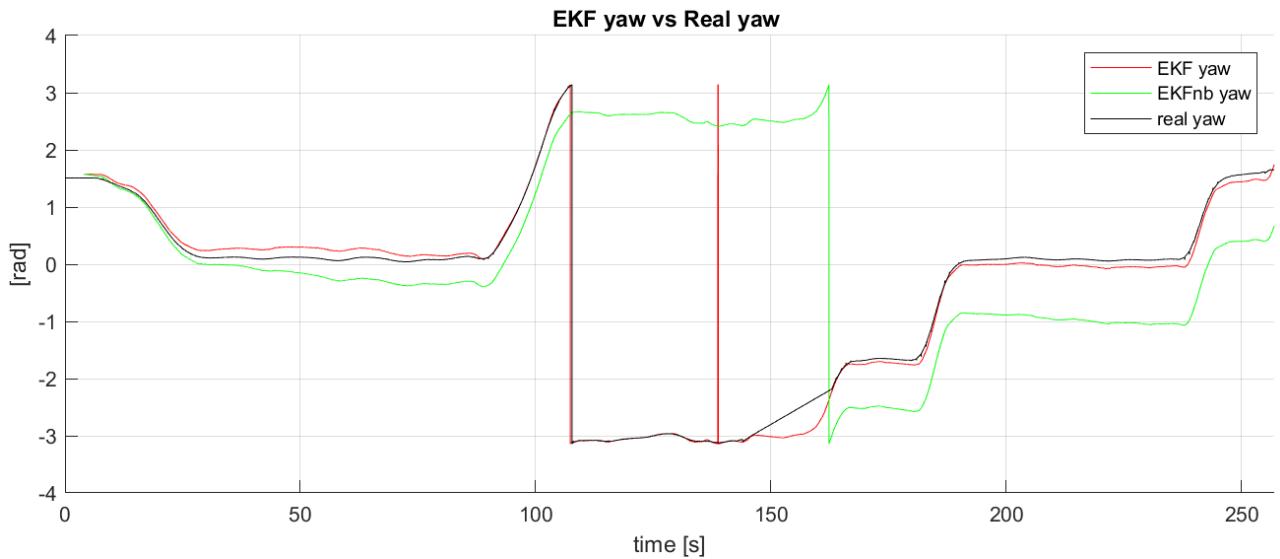


Figura 6.47: Confronto tra angoli di yaw stimati e reale

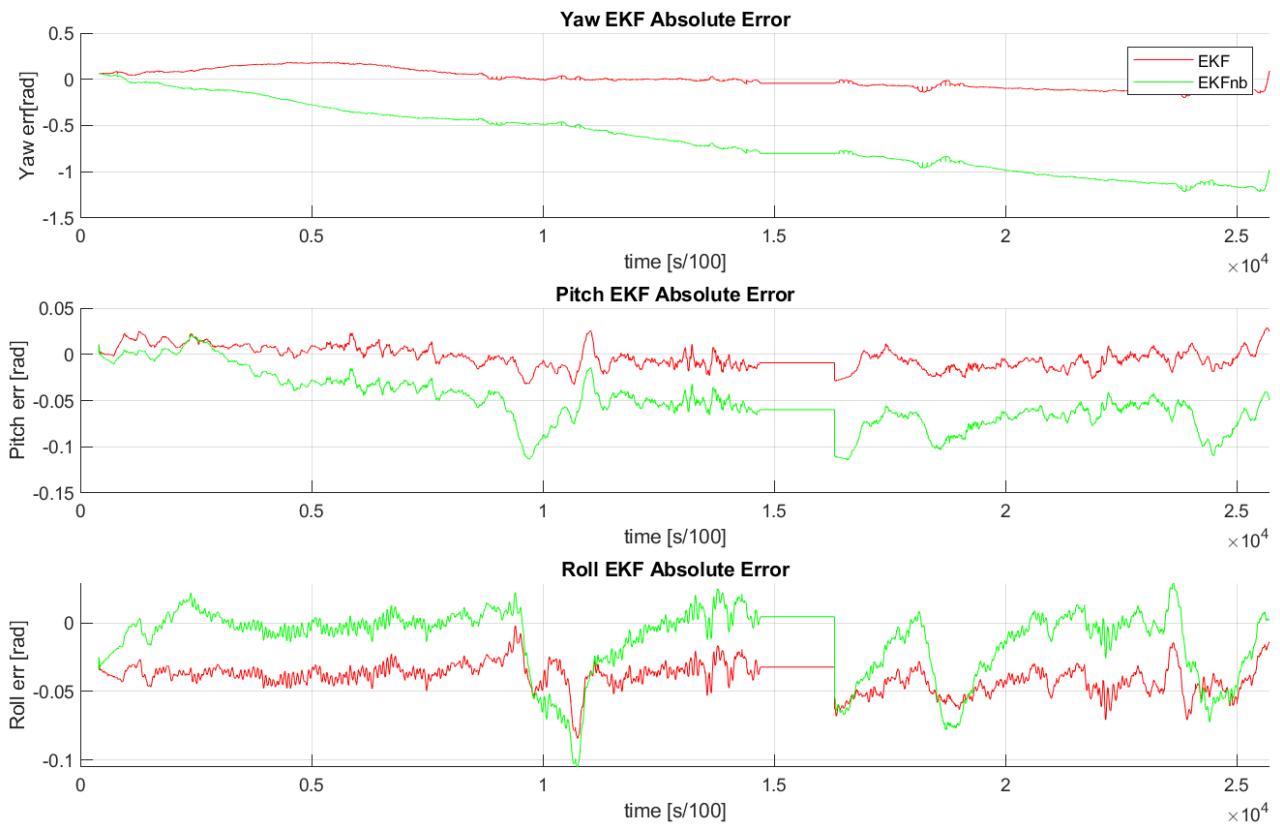


Figura 6.48: Errori assoluti degli angoli di Eulero di assetto stimati

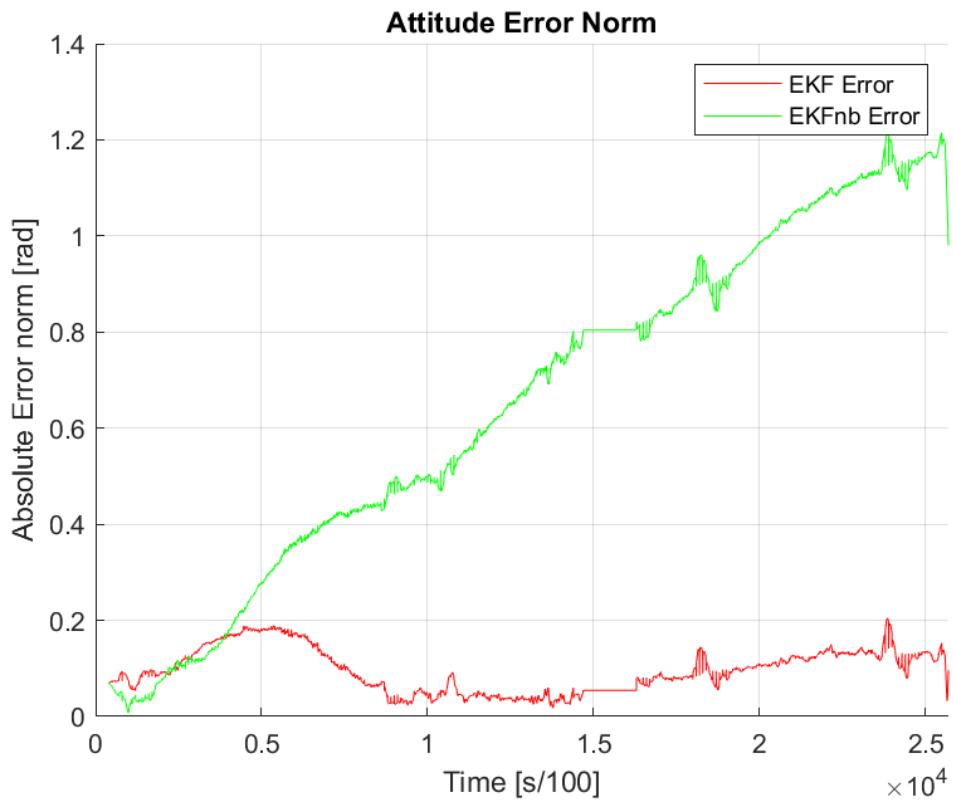


Figura 6.49: Norma dell'errore assoluto degli angoli di Eulero di assetto stimati

Le figure 6.47 e 6.48 e la tabella 6.10 mostrano risultati del tutto paragonabili in termini di andamento e ampiezza rispetto al secondo caso.

	$RMSE_{yaw}$	$RMSE_{pitch}$	$RMSE_{roll}$	$RMSE_{tot}$	Unità
EKF	0.096318	0.011855	0.040378	0.10511	rad
EKFnb	0.72918	0.05747	0.026736	0.73193	rad

Tabella 6.10: RMSE della stima dell'assetto

## 6.5 Considerazioni sul parametro $zero\_vel\_period$

Di seguito sono mostrati dei grafici qualitativi che possono orientarci nella scelta del parametro  $zero\_vel\_period$ , impostando la simulazione con l'utilizzo del GPS RTK e le interruzioni come nei casi precedenti nelle sezioni 6.1 e 6.2.

Assegnando al parametro valori pari a  $[\infty, 5, 0.1, 0.02]s$  otteniamo una stima della posizione differente:

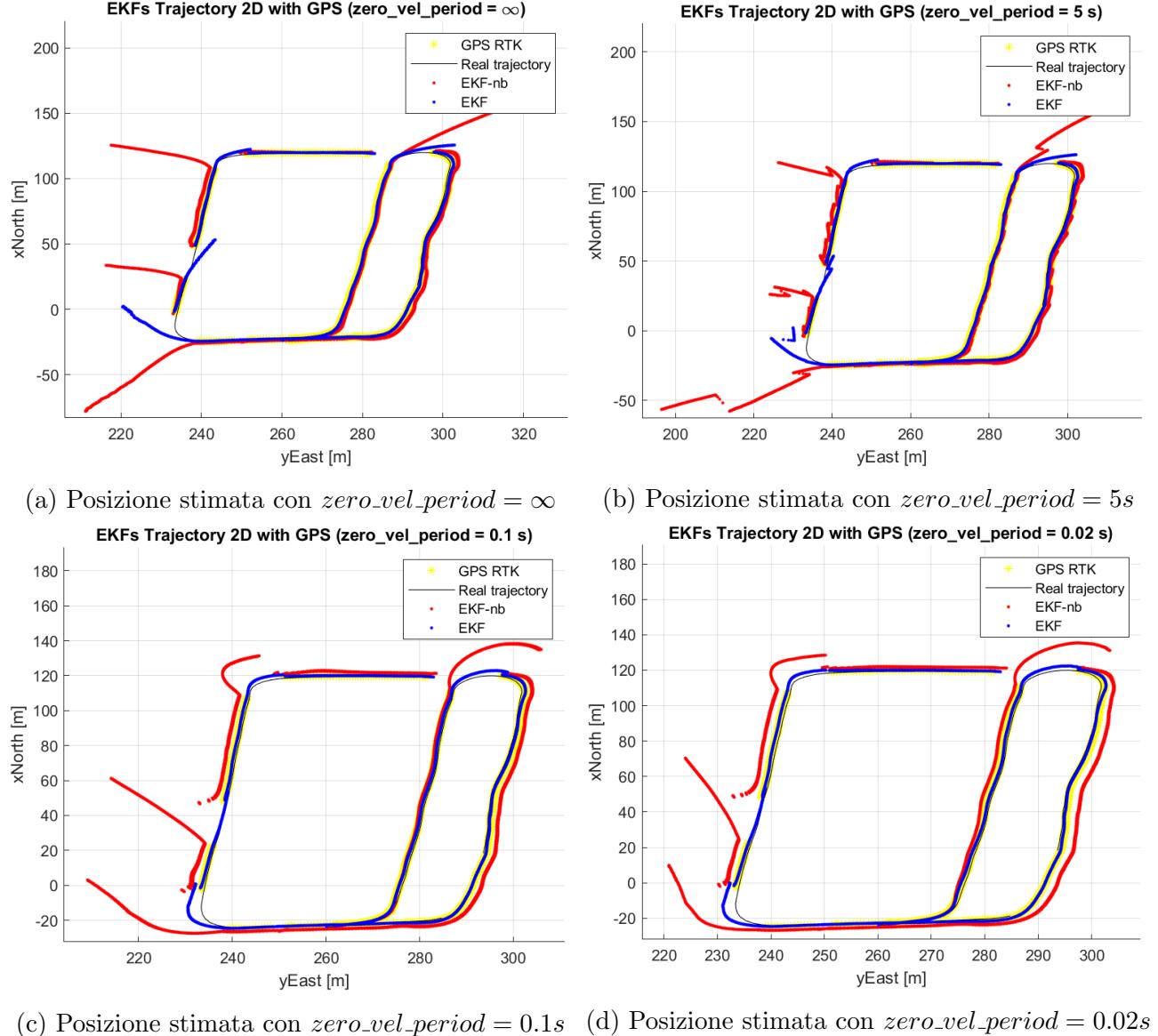


Figura 6.50: Confronto tra posizione stimata e posizione reale con EKF con e senza stima del bias di accelerometro e giroscopio, al variare del parametro  $zero\_vel\_period$

Si nota istantaneamente come più basso è il valore del parametro, più è accurata la stima della posizione. Può essere interessante confrontare anche i comportamenti delle varianze della stima di posizione. Nelle figure 6.51a e 6.51b sono mostrati gli andamenti nel tempo della varianza della stima di posizione delle componenti  $xNorth$  e  $yEast$ , prelevate direttamente dalla diagonale della matrice di covarianza della stima  $P$ . I punti in cui aumenta l'incertezza della stima, e quindi la varianza, corrispondono ai momenti in cui non è disponibile il GPS. Soffermandoci sui due picchi centrali, essi corrispondono ai momenti in cui il veicolo si muove

verso Nord, per cui, con la correzione basata sui vincoli dinamici, l'assumere nulla la velocità laterale del veicolo in terna body provoca una stima più accurata e quindi una varianza più bassa nella componente  $yEast$ .

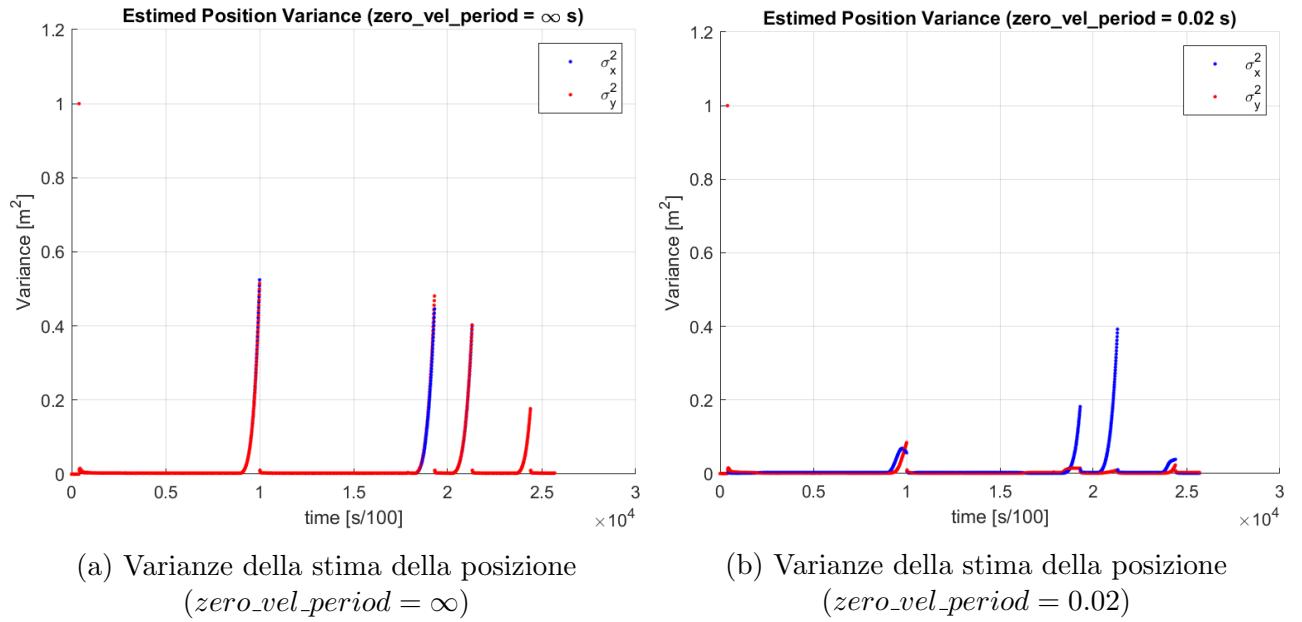


Figura 6.51: Varianze della stima di posizione in componenti  $xNorth$  e  $yEast$  al variare di  $zero\_vel\_period$

Per visualizzare meglio questo effetto, le figure 6.52 e 6.53 riportano una parte del percorso con le *ellissi di covarianza*: nel caso in cui la correzione è disattiva, le ellissi diventano cerchi, in quanto l'incertezza è uguale in tutte le direzioni rispetto allo spostamento e al veicolo; diversamente quando la correzione è attiva, le ellissi sono schiacciate verso la direzione di spostamento del veicolo proprio perché si assume che la velocità laterale è nulla e quindi il veicolo ha meno probabilità di trovarsi ai lati della traiettoria.

Per disegnare le ellissi sulla base della matrice di covarianza si vedano i file:

`plot_state_pos_2D_EKFs_withDevStd.m` e `error_ellipse.m`, con riferimento a [8].

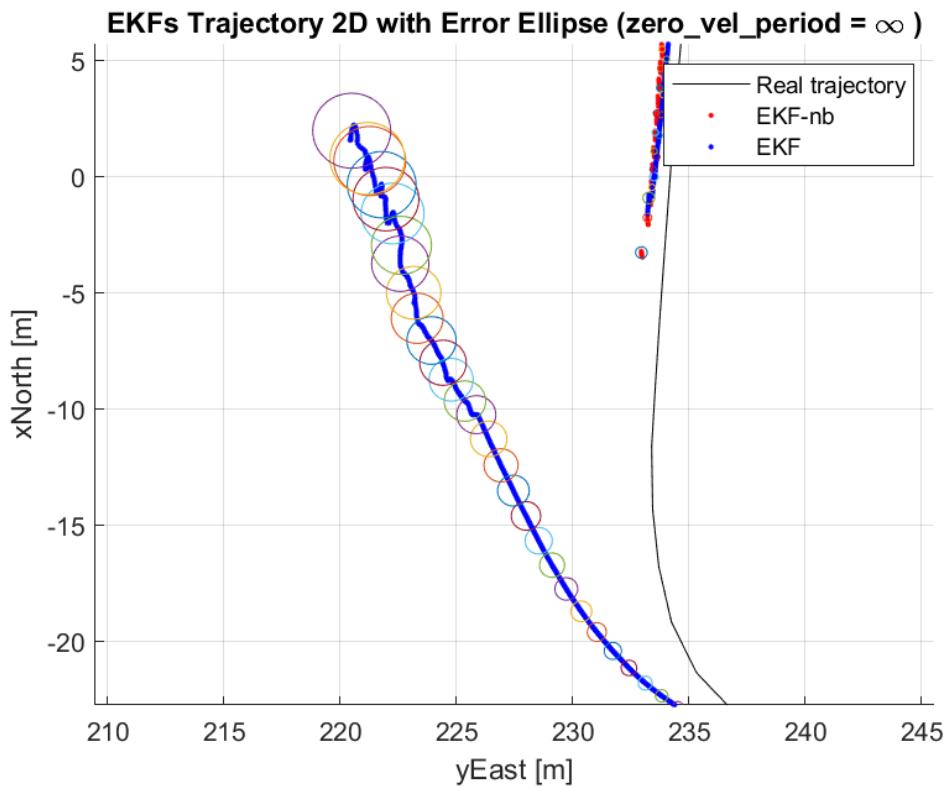


Figura 6.52: Confronto tra posizione stimata e riferimento con ellissi di covarianza  
( $zero\_vel\_period = \infty$ )

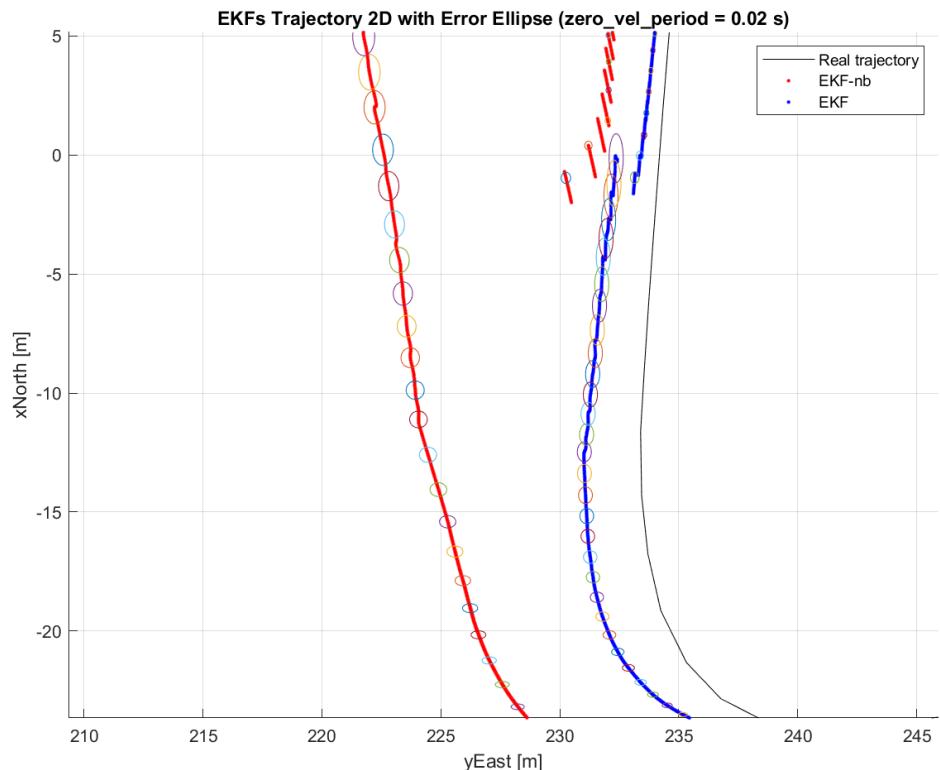


Figura 6.53: Confronto tra posizione stimata e riferimento con ellissi di covarianza  
( $zero\_vel\_period = 0.02s$ )

## 6.6 Conclusioni

In conclusione quindi si può ritenere che l'accuratezza della stima della cinematica del veicolo è in linea con gli usuali risultati ottenibili da un filtro di Kalman in configurazione diretta. La stima degli errori dei sensori inerziali migliora nettamente la stima dello stato del filtro, sia con il GPS RTK che con il DGPS low-cost: semplicemente osservando gli indici  $RMSE$  di posizione e assetto ottenuti nel capitolo precedente. Di seguito sono riportati gli indici  $RMSE$  del caso peggiore, ovvero quando si utilizza il DGPS low-cost con interruzioni del segnale.

# sim	EKF type	$RMSE_{yaw}$	$RMSE_{pitch}$	$RMSE_{roll}$	$RMSE_{tot}$	Unità
2	EKF	0.096583	0.011864	0.040384	0.10535	rad
2	EKFnb	0.7543	0.057588	0.026778	0.75697	rad

Tabella 6.11: RMSE massimi della stima degli angoli di Eulero di assetto

# sim	EKF type	$RMSE_x$	$RMSE_y$	$RMSE_z$	$RMSE_{tot}$	Unità
2	EKF	20.08	2.4585	13.093	24.101	m
2	EKFnb	29.196	22.107	21.343	42.386	m

Tabella 6.12: RMSE massimi della stima di posizione

Si nota come considerare i bias di giroscopio e accelerometro causi una variazione di circa  $0.65rad$  sull' $RMSE_{tot}$  di assetto e circa  $18m$  sull' $RMSE_{tot}$  di posizione, valori non di certo trascurabili.

Per ottenere prestazioni migliori anche con GPS ad uso commerciale, si potrebbe pensare di estendere il filtro inserendo un modello di misura del sensore stesso, cercando di stimare eventuali bias così come si è fatto per il giroscopio e accelerometro.

Naturalmente ben diversa è la qualità della stima con un sensore più preciso come un GPS RTK. Anche in tal caso però la stima dei bias risulta essere sempre efficace per ottenere maggiore precisione. La miglior stima di posizione e assetto si ottiene nel caso in cui il segnale del GPS RTK è sempre disponibile.

# sim	EKF type	$RMSE_{yaw}$	$RMSE_{pitch}$	$RMSE_{roll}$	$RMSE_{tot}$	Unità
3	EKF	0.046400	0.010797	0.040369	0.062443	rad
3	EKFnb	0.40491	0.045537	0.025972	0.40829	rad

Tabella 6.13: RMSE minimi della stima degli angoli di Eulero di assetto

# sim	EKF type	$RMSE_x$	$RMSE_y$	$RMSE_z$	$RMSE_{tot}$	Unità
3	EKF	1.7211	0.93214	0.41794	2.0014	m
3	EKFnb	2.2057	1.7648	0.29924	2.8407	m

Tabella 6.14: RMSE minimi della stima di posizione

Confrontando i valori dell' $RMSE_{yaw}$  si nota come sia essenziale, anche con la disponibilità di un GPS molto accurato, l'utilizzo della stima dei bias dei sensori inerziali.

## 7 Simulazione con dati di un drone

In quest'ultimo capitolo si presenta un'ulteriore simulazione con un dataset diverso. Si tratta di dati collezionati da dei sensori durante il volo di un drone e quindi come tale di natura diversa rispetto i dati precedenti.

Il file datalog in questione è `LOG00054_parsed_seg3.mat` dal quale vengono estratte le misure di accelerometro ( $50Hz$ ), giroscopio ( $50Hz$ ), GPS ( $10Hz$ ), yaw ( $10Hz$ ) e anche una posa 6D di riferimento, eseguendo il file `load_dataset2.m`. Seguono i grafici degli andamenti di tali misure.

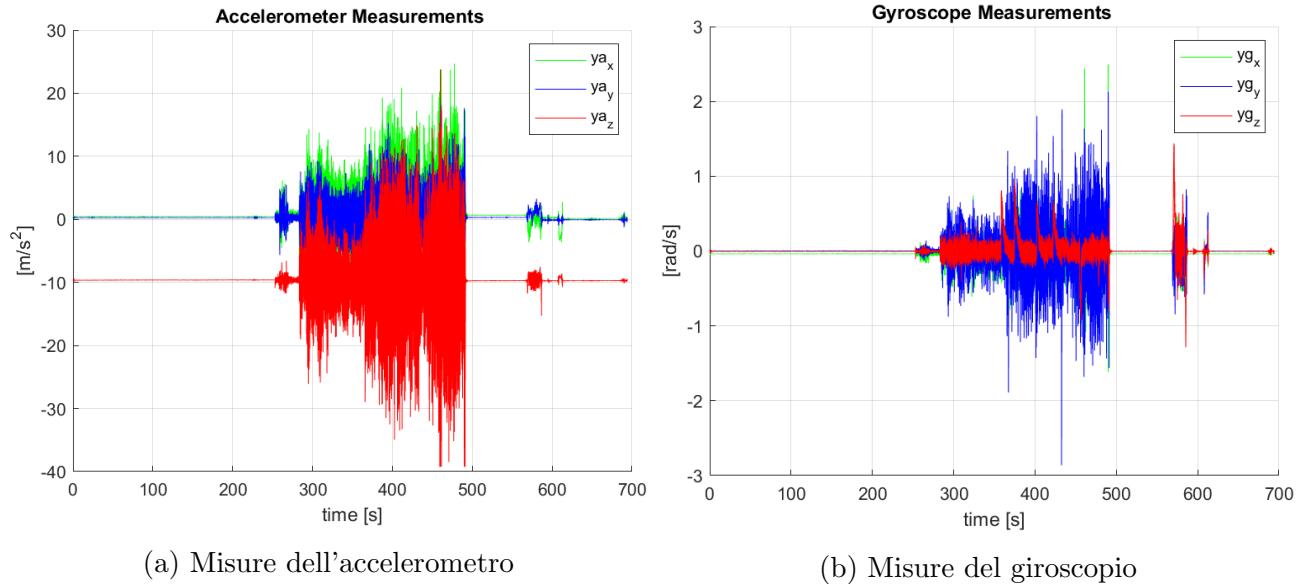


Figura 7.1: Misure provenienti dall'unità inerziale IMU

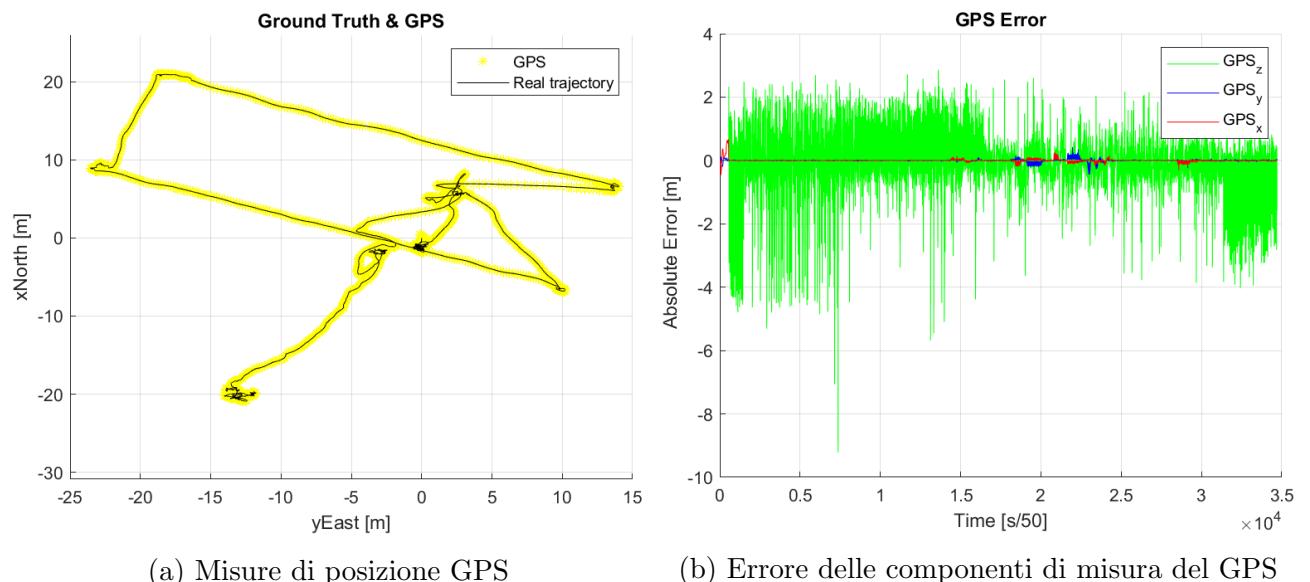


Figura 7.2: Plot 2D con posizione di riferimento e misure del GPS e errore.

Per adattare il filtro al nuovo tipo di veicolo e di misure disponibili, sono state effettuate alcune modifiche, di seguito le più importanti:

- Inizializzazione

In questo caso si è scelto di assegnare l'assetto iniziale del drone pari a  $[yaw, pitch, roll] = [0, 0, 0]$  e utilizzare le misure dell'accelerometro solo per stimare il bias dello stesso. Come per il bias del giroscopio (eq. 3.7), il valore iniziale del bias dell'accelerometro è:

$$x_{a_{ini}} = \frac{1}{T} \left( \sum_{t=0}^T y_a(t) \right) + g_z \quad (7.1)$$

La stima dello stato è per cui inizializzata come:

$$\hat{x} = [q_{ini}^{nb} \quad 0_{1 \times 3} \quad x_{g_{ini}} \quad 0_{1 \times 9} \quad x_{a_{ini}}] \quad (7.2)$$

La matrice di covarianza  $P_{ini}$  è invece invariata (eq. 3.9).

- Vincoli dinamici

Differentemente da un veicolo su ruote come era l'auto elettrica usata per l'esperimento precedente, un drone in volo può muoversi indipendentemente in tutte le direzioni senza dover rispettare alcun vincolo. Di conseguenza la correzione *Dynamical Constraint Update* basata sul modello precedente, in cui veniva corretta la velocità laterale del veicolo e posta pari a zero, non è più corretta ed è stata eliminata completamente.

- Misura e correzione di heading

Come per i vincoli dinamici, anche qui la misura dell'angolo di heading non può essere utilizzata per correggere l'assetto del veicolo, ragion per cui si è deciso di sostituire tale misura con una "fittizia" dell'angolo di yaw, prelevandola dall'orientazione di riferimento. Per adattare la struttura esistente al nuovo tipo di misura, la formula (2.22) è stata sostituita con la  $\psi_{yaw}$  di (4.7), cioè:

$$h_\theta = \psi_{yaw} = \arctan\left(\frac{\bar{R}_{21}}{\bar{R}_{11}}\right) = \text{atan2}\left(2(q_1 q_2 + q_0 q_3), \quad q_0^2 + q_1^2 - q_2^2 - q_3^2\right) \quad (7.3)$$

in cui  $\bar{R} = R^{bn}$  (body-to-navigation) di eq. (3.12):

$$R^{bn} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

Come conseguenza inoltre, è stato ricalcolato anche il nuovo valore di  $\nabla H_\theta$ , come descritto in eq. (2.24).

- Frequenza del filtro

Le nuove misure disponibili hanno frequenza diversa rispetto il dataset precedente quindi, seguendo la regola che la frequenza di aggiornamento del filtro deve essere pari a quella più alta delle misure disponibili, la nuova frequenza del filtro è posta a  $50Hz$ , come quella dei dati inerziali, e quindi  $\Delta t = 0.02s$ .

Con questo nuovo dataset disponibile e il filtro aggiornato, seguono due simulazioni: la prima mantiene gli stessi parametri del filtro precedente, la seconda invece propone un tuning diverso in ricerca di maggior precisione della stima.

Le istruzioni per eseguire le simulazioni sono le stesse del caso precedente (vedi pag. 25), accedendo alla cartella `Matlab_files\sim_drone_dataset`.

## 7.1 Risultati della simulazione

In questa simulazione verranno utilizzati gli stessi parametri definiti nel capitolo 5 come riportato nella seguente tabella. Solo due verranno modificati: `EKF_DT=0.02s` in quanto la frequenza del filtro ora è di  $50Hz$ ; `zero_vel_period=∞` in modo tale che la correzione per i vincoli dinamici non venga mai effettuata in quanto errata.

Parametro	Simbolo	Valore	Unità
<code>gyro_variance</code>	$\sigma_g^2$	$4.8 \times 10^{-6}$	$(rad/s)^2$
<code>bias_gyro_variance</code>	$\sigma_{xg}^2$	$4.0 \times 10^{-14}$	$(rad/s^2)^2$
<code>acc_variance</code>	$\sigma_a^2$	$4.8 \times 10^{-2}$	$(m/s^2)^2$
<code>bias_acc_variance</code>	$\sigma_{xa}^2$	$4.0 \times 10^{-14}$	$(m/s^3)^2$
<code>lambda_xg</code>	$\lambda_{xg}$	$1.0 \times 10^{-3}$	$1/s$
<code>lambda_xa</code>	$\lambda_{xa}$	$1.0 \times 10^{-4}$	$1/s$
<code>vel_variance</code>	$\sigma_v^2$	$1.0 \times 10^{-2}$	$(m/s)^2$
<code>GPS_variance</code>	$\sigma_r^2$	$1.0 \times 10^{-2}$	$(m)^2$
<code>head_variance</code>	$\sigma_\theta^2$	$7.0 \times 10^{-3}$	$(rad)^2$
<code>EKF_DT</code>	$\Delta t$	0.02	(s)
<code>g</code>	$g$	9.81	$(m/s^2)$
<code>init_time</code>	-	11	-
<code>SHOE_buffer_N</code>	$N$	100	-
<code>SHOE_threshold</code>	$\gamma$	2000	-
<code>zero_vel_period</code>	-	$\infty$	(s)

Tabella 7.1: Tabella dei parametri della simulazione

Anche in questo caso sono simulati dei malfunzionamenti del GPS.

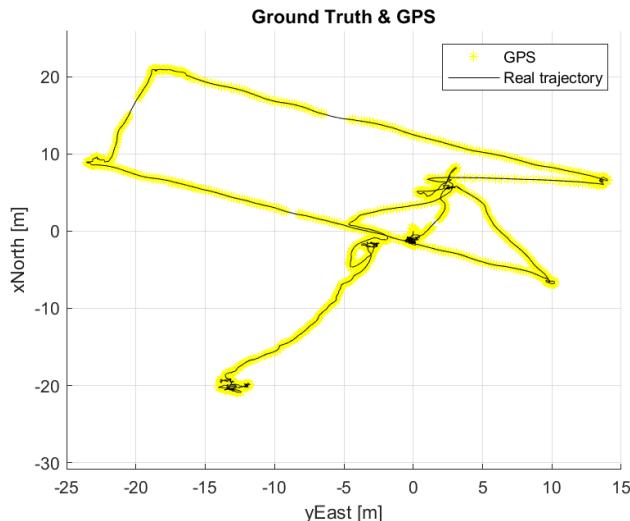


Figura 7.3: Segnale del GPS

	Valore	Unità
<code>t1_start</code>	290.0	s
<code>t1_end</code>	291.0	s
<code>t2_start</code>	392.0	s
<code>t2_end</code>	392.5	s
<code>t3_start</code>	419.0	s
<code>t3_end</code>	420.0	s
<code>t4_start</code>	438.0	s
<code>t4_end</code>	439.0	s

Tabella 7.2: Intervalli di tempo degli *outages*

Seguono vari grafici sulla stima dello stato del filtro.

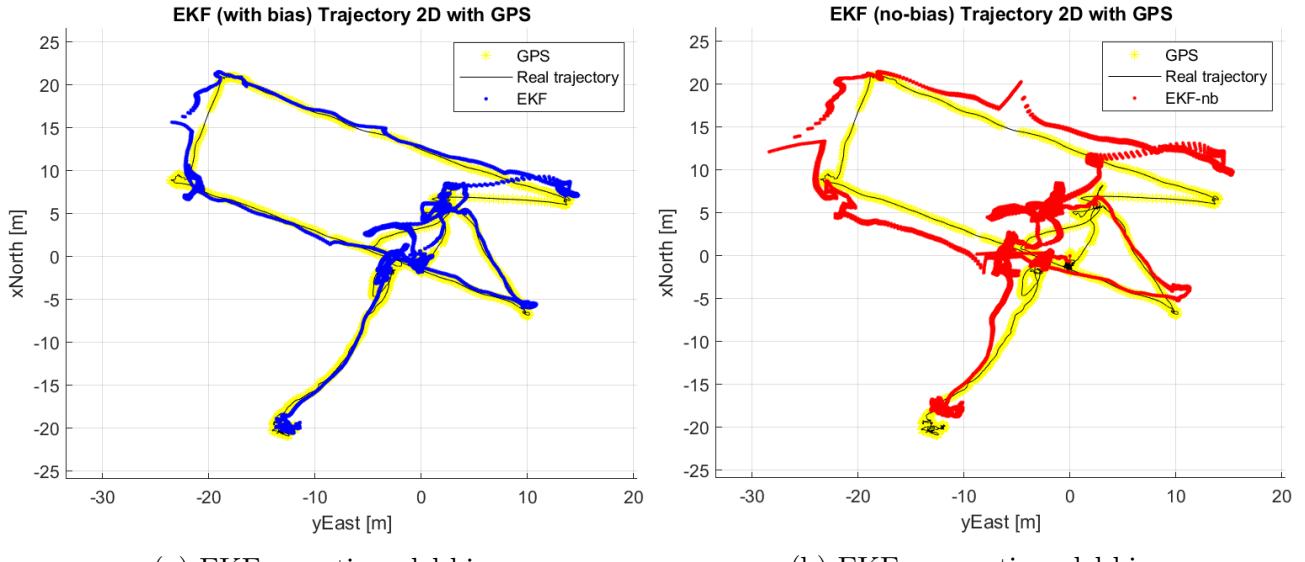


Figura 7.4: Confronto tra posizione stimata e posizione reale con EKF con stima del bias di accelerometro e giroscopio (a), e senza (b)

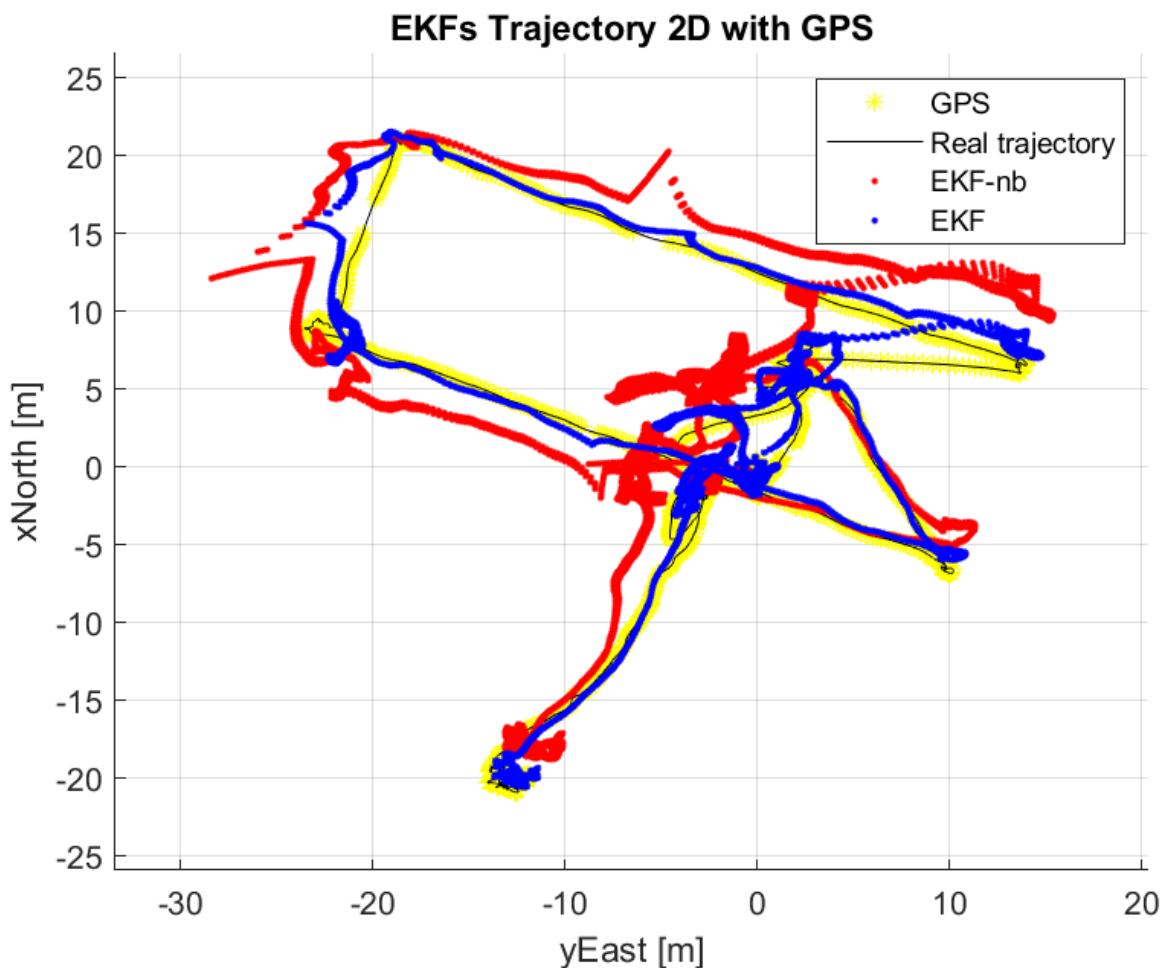


Figura 7.5: Confronto tra posizione stimata e posizione esatta

Le figure 7.4 e 7.5 mostrano come la stima di posizione sia nettamente più precisa nel caso in cui i bias vengano stimati, risultato piuttosto prevedibile anche intuitivamente. Tuttavia è anche scontato osservare una poca accuratezza rispetto la traiettoria di riferimento in entrambi i casi in quanto il veicolo e la sua dinamica, e quindi anche la natura delle misure, sono diverse rispetto il caso precedente.

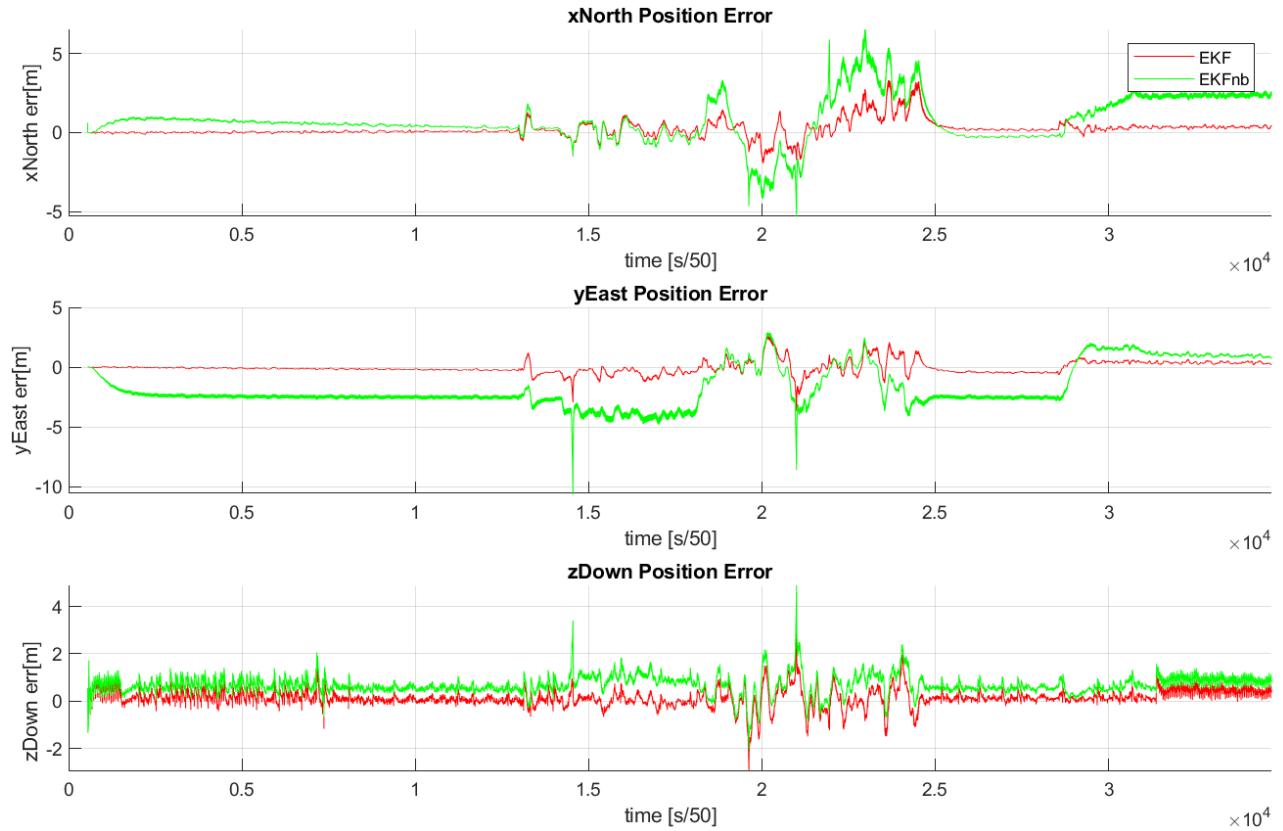


Figura 7.6: Errori assoluti delle componenti della stima di posizione

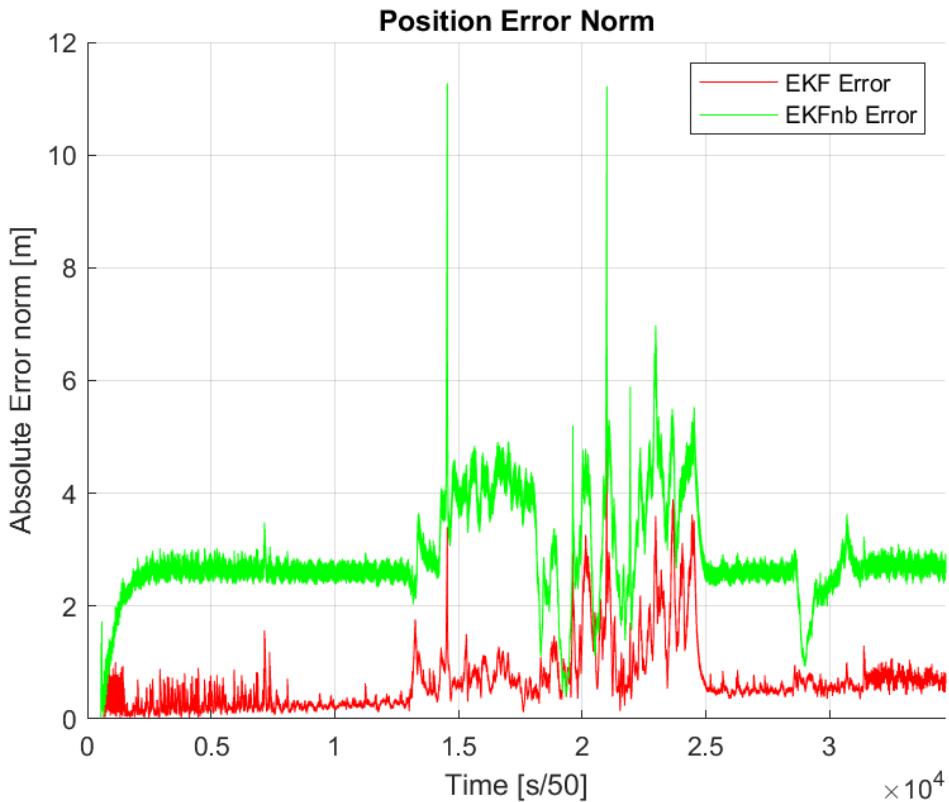


Figura 7.7: Norma dell'errore assoluto della stima di posizione

	$RMSE_x$	$RMSE_y$	$RMSE_z$	$RMSE_{tot}$	Unità
EKF	0.58237	0.53068	0.3958	0.88172	<i>m</i>
EKFnb	1.5817	2.3878	0.78638	2.9702	<i>m</i>

Tabella 7.3: RMSE della stima di posizione

Gli andamenti in figura degli errori assoluti e gli indici RMSE appena riportati, indicano una netta differenza tra i due filtri e nessuna differenza quantitativa degli errori tra le componenti della stima della posizione.

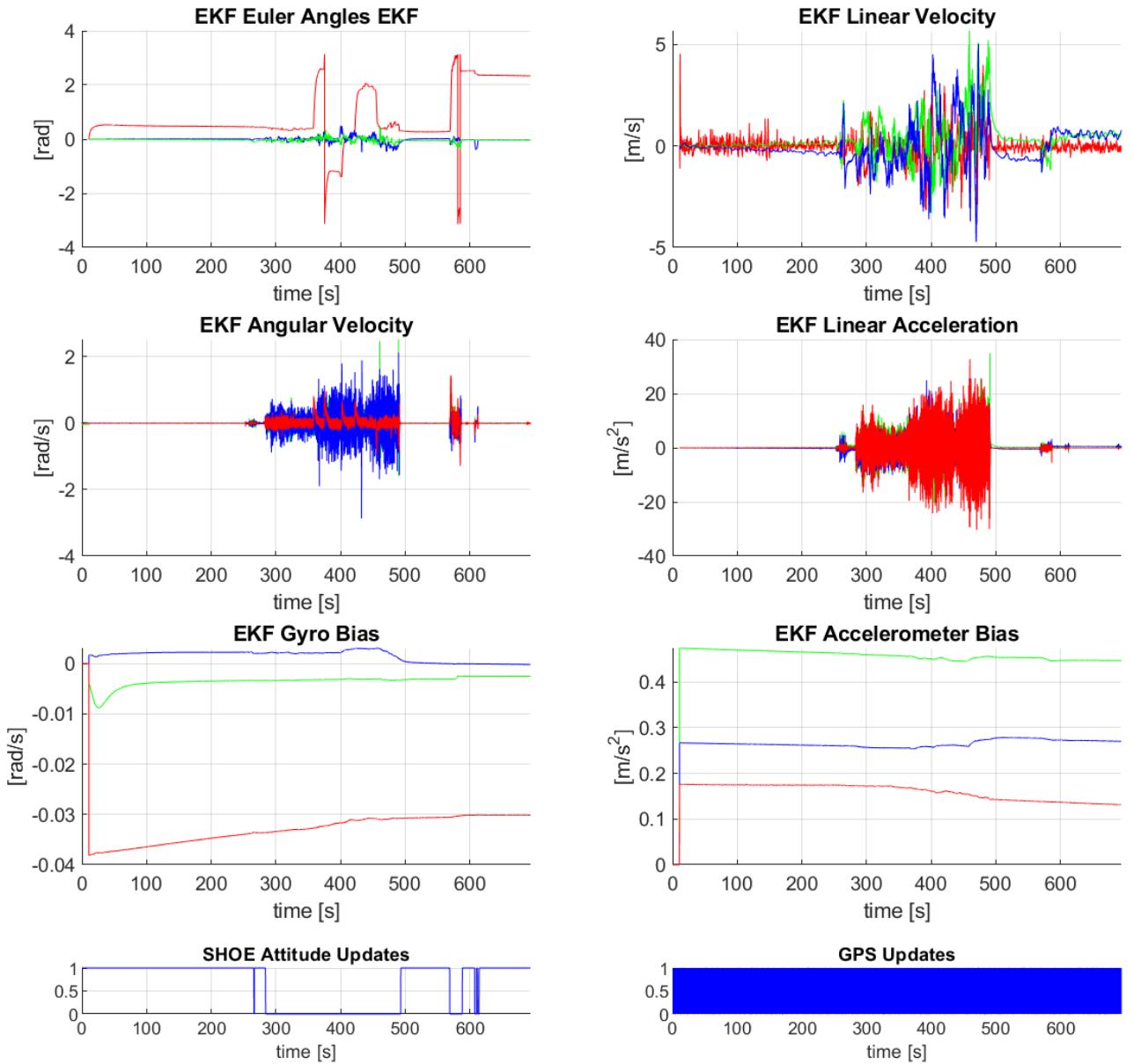


Figura 7.8: Variabili stimate dall'EKF con bias: assetto [yaw,pitch,roll], velocità lineare, velocità angolare, accelerazione e bias con componenti [Z,Y,X], i grafici in basso indicano quando le correzioni di assetto e del GPS avvengono.

In fig. 7.8 si può notare come il valore delle componenti dell'accelerazione lineare è pressoché zero nei momenti in cui il veicolo è fermo, differentemente da quello che accade al filtro senza la stima dei bias (fig. 7.9). Questo causa nel secondo filtro una errata stima della velocità lineare e quindi una stima peggiore della posa del veicolo.

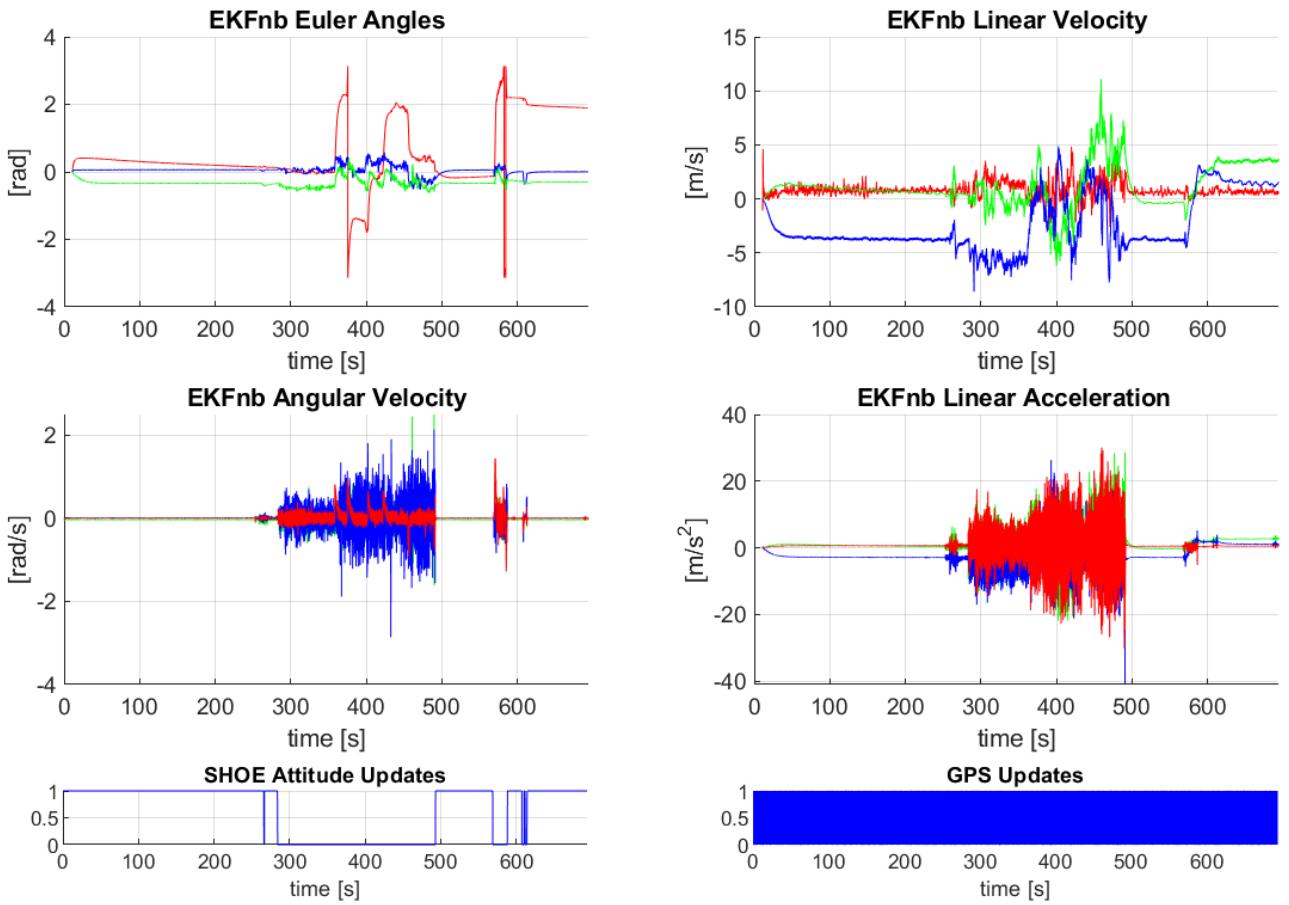


Figura 7.9: Variabili stimate dall'EKF con bias: assetto [yaw,pitch,roll], velocità lineare, velocità angolare, accelerazione con componenti [Z,Y,X], i grafici in basso indicano quando le correzioni di assetto e del GPS avvengono.

Per quanto riguarda l'assetto in particolare:

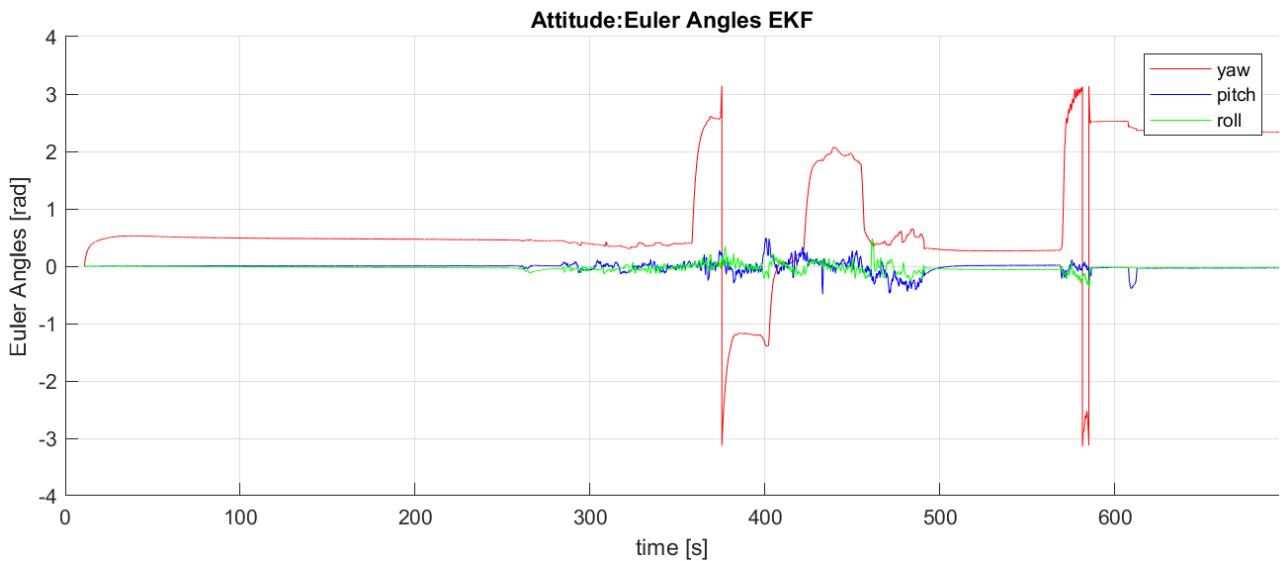


Figura 7.10: Angoli di Eulero di assetto stimati dal filtro EKF con bias

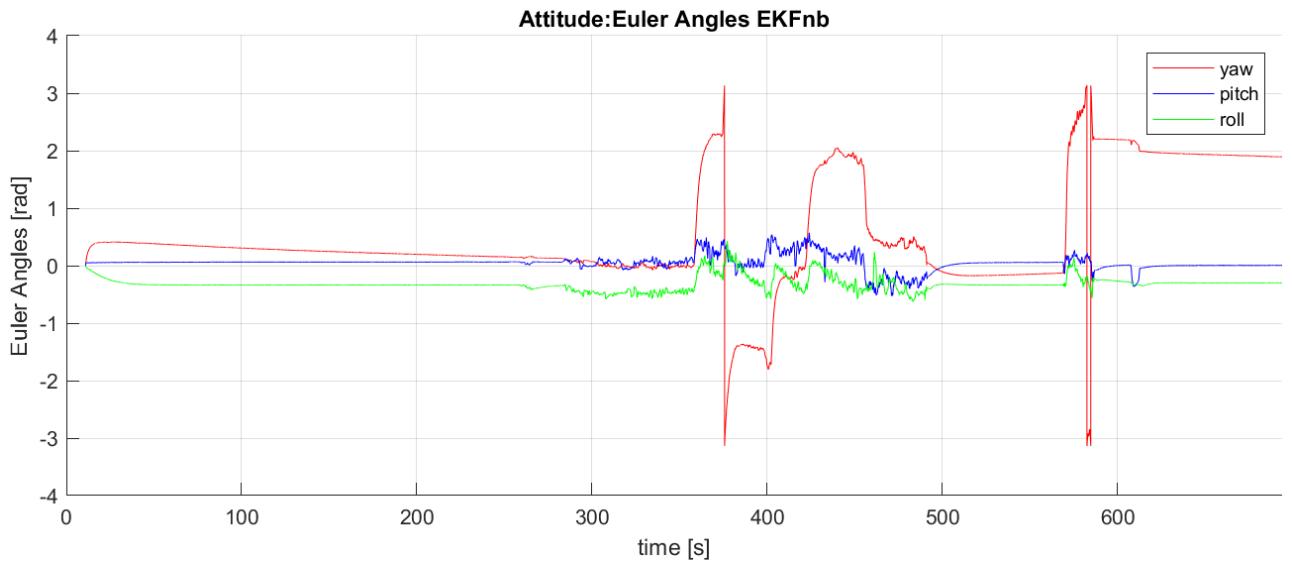


Figura 7.11: Angoli di Eulero di assetto stimati dal filtro EKF senza stima del bias

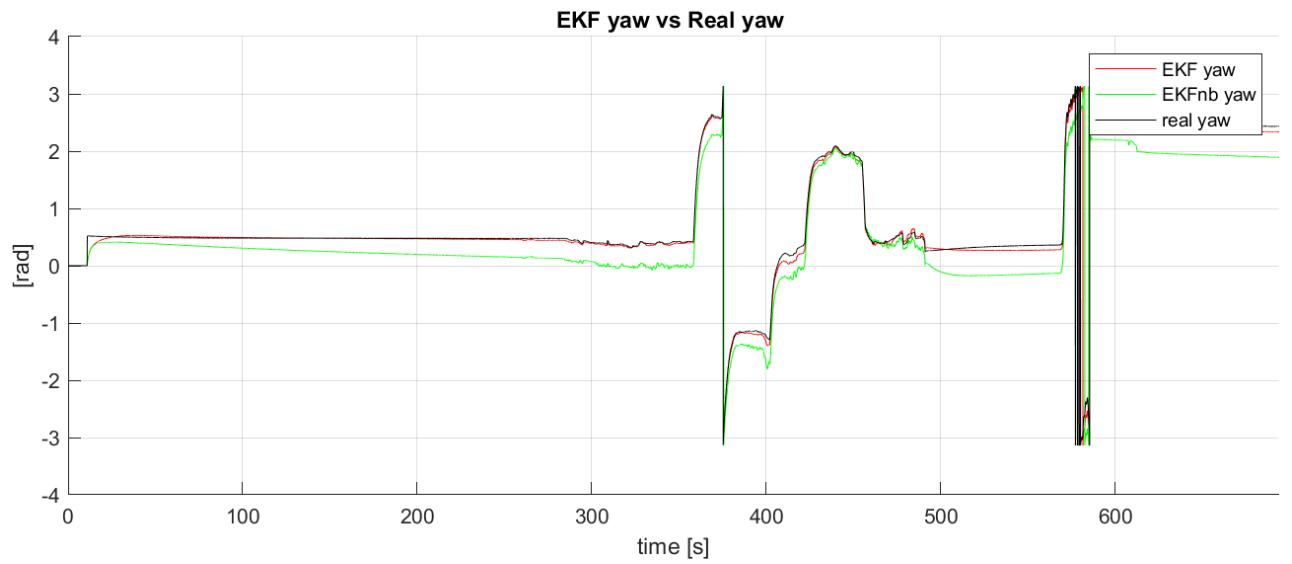


Figura 7.12: Confronto tra angoli di yaw stimati e reale

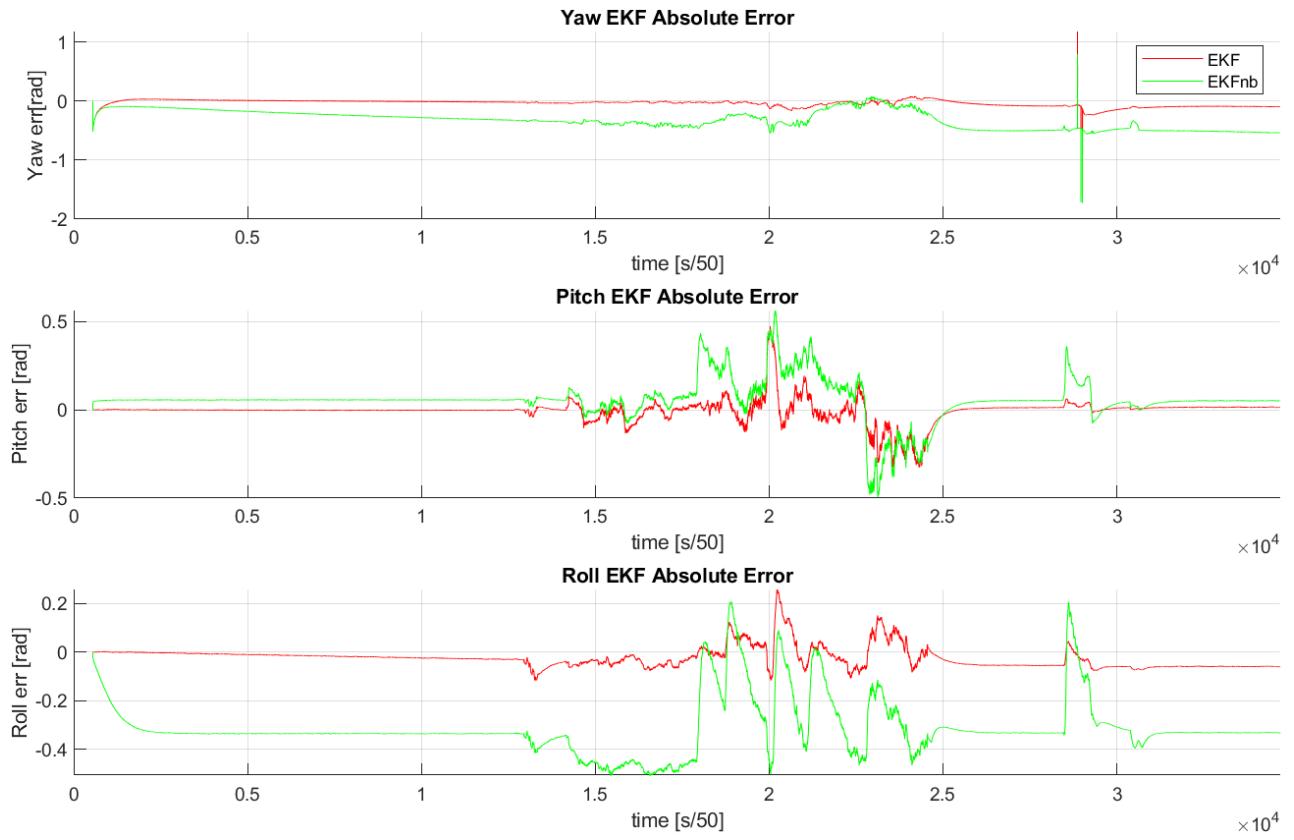


Figura 7.13: Errori assoluti degli angoli di Eulero di assetto stimati

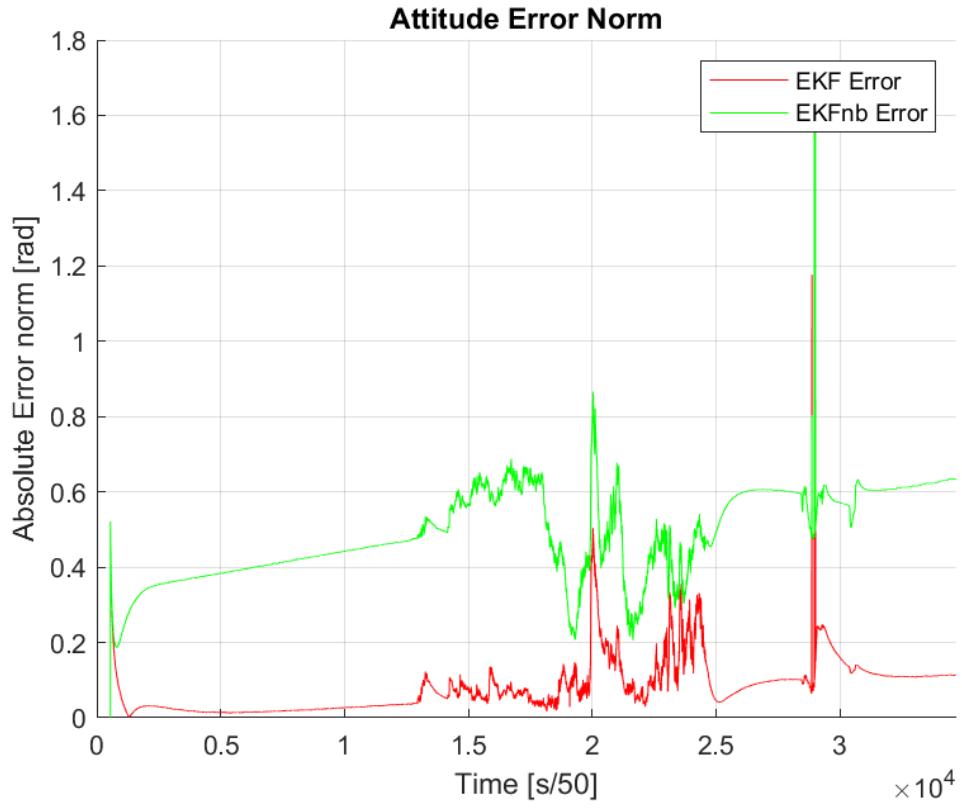


Figura 7.14: Norma dell'errore assoluto degli angoli di Eulero di assetto stimati

Le figure 7.12 e 7.13 e la tabella 7.4 mostrano come anche la stima dell'assetto risulta compromessa dall'assenza della stima dei bias. In particolare l'angolo di yaw e di roll sono nettamente mal stimati con un RMSE medio di  $RMSE_{yaw} \approx RMSE_{roll} \approx 0.33rad = 18.91deg$ . Con la stima dei bias invece gli errori sono molto più contenuti, dell'ordine di qualche grado per angolo di Eulero.

	$RMSE_{yaw}$	$RMSE_{pitch}$	$RMSE_{roll}$	$RMSE_{tot}$	Unità
EKF	0.067564	0.064411	0.049044	0.10545	rad
EKFnb	0.34879	0.12356	0.33054	0.49616	rad

Tabella 7.4: RMSE della stima dell'assetto

In conclusione quindi il filtro completo con questi parametri permette di stimare la posizione restando sotto il metro di errore quadratico medio e un errore di  $2 - 3rad$  per gli angoli di assetto. Per un veicolo come un drone in volo potrebbe comunque non essere abbastanza in base all'utilizzo, motivo per il quale la prossima sezione propone un miglior tuning dei parametri per raggiungere un'accuratezza della stima migliore.

## 7.2 Risultati della simulazione con parametri aggiornati

In questa sezione si propone un diverso tuning dei parametri del sistema come riportato nella tabella seguente, con lo scopo di avvicinare il più possibile la stima di posizione al riferimento. Tali valori sono stati assegnati tramite prove e simulazioni successive.

Parametro	Simbolo	Valore	Unità
gyro_variance	$\sigma_g^2$	$4.8 \times 10^{-2}$	(rad/s) <sup>2</sup>
bias_gyro_variance	$\sigma_{xg}^2$	$4.0 \times 10^{-14}$	(rad/s <sup>2</sup> ) <sup>2</sup>
acc_variance	$\sigma_a^2$	$4.8 \times 10^{-1}$	(m/s <sup>2</sup> ) <sup>2</sup>
bias_acc_variance	$\sigma_{xa}^2$	$4.0 \times 10^{-10}$	(m/s <sup>3</sup> ) <sup>2</sup>
lambda_xg	$\lambda_{xg}$	$1.0 \times 10^{-3}$	1/s
lambda_xa	$\lambda_{xa}$	$1.0 \times 10^{-4}$	1/s
vel_variance	$\sigma_v^2$	$1.0 \times 10^{-2}$	(m/s) <sup>2</sup>
GPS_variance	$\sigma_r^2$	$1.0 \times 10^{-2}$	(m) <sup>2</sup>
head_variance	$\sigma_\theta^2$	$7.0 \times 10^{-3}$	(rad) <sup>2</sup>
EKF_DT	$\Delta t$	0.02	(s)
g	$g$	9.81	(m/s <sup>2</sup> )
init_time	-	11	-
SHOE_buffer_N	$N$	100	-
SHOE_threshold	$\gamma$	30	-
zero_vel_period	-	$\infty$	(s)

Tabella 7.5: Tabella dei parametri della simulazione

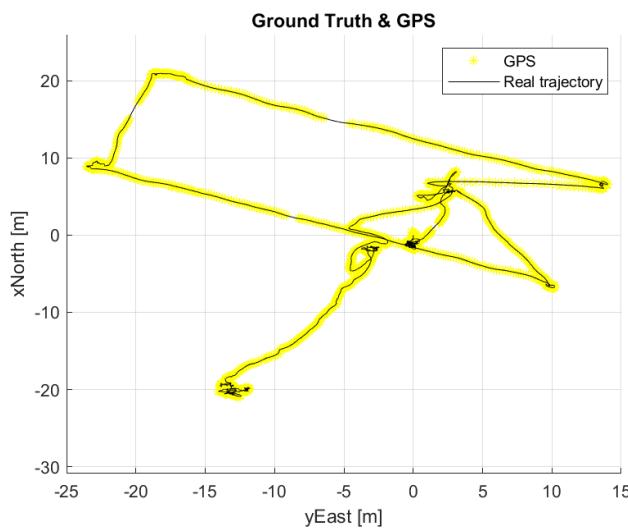
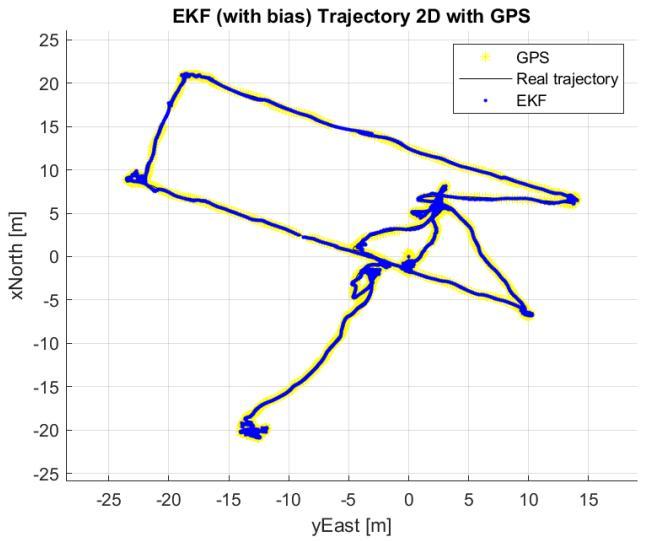


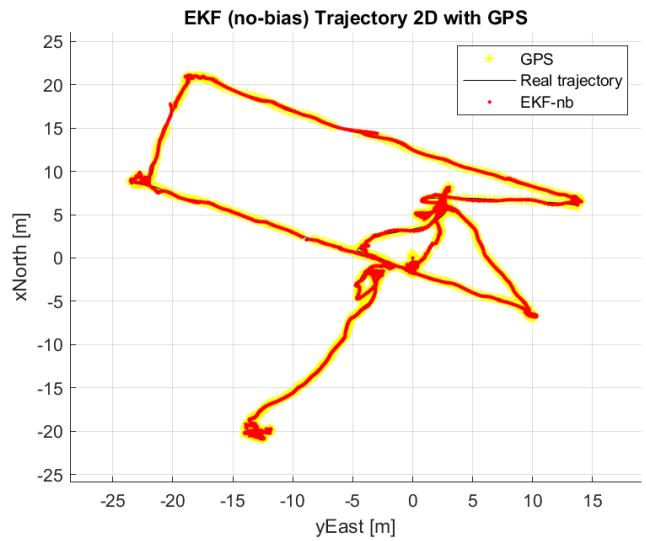
Figura 7.15: Segnale del GPS

	Valore	Unità
t1_start	290.0	s
t1_end	291.0	s
t2_start	392.0	s
t2_end	392.5	s
t3_start	419.0	s
t3_end	420.0	s
t4_start	438.0	s
t4_end	439.0	s

Tabella 7.6: Intervalli di tempo degli *outages*



(a) EKF con stima del bias



(b) EKF senza stima del bias

Figura 7.16: Confronto tra posizione stimata e posizione reale con EKF con stima del bias di accelerometro e giroscopio (a), e senza (b)

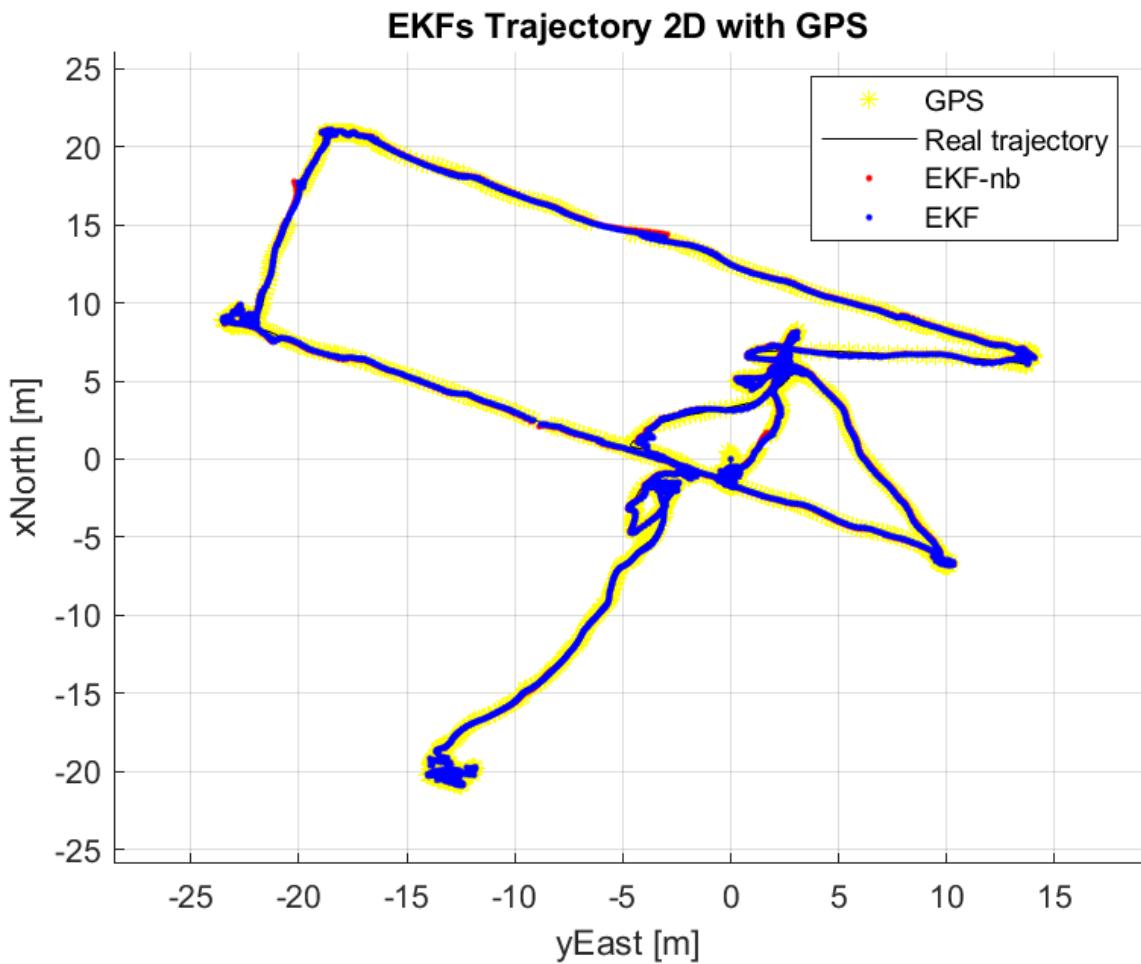


Figura 7.17: Confronto tra posizione stimata e posizione esatta

Le figure 7.16 e 7.17 mostrano come la stima della posizione sul piano è quasi identica al riferimento in entrambi i casi. Questo perché i nuovi parametri rendono molto più determinanti le misure del GPS rispetto a quelle dei sensori inerziali, cioè compensare o meno il bias è meno importante in quanto la stima di posizione è meno influenzata dalle misure di giroscopio e accelerometro. L'indice  $RMS E_{tot}$  in entrambi i casi rimane sotto il mezzo metro.

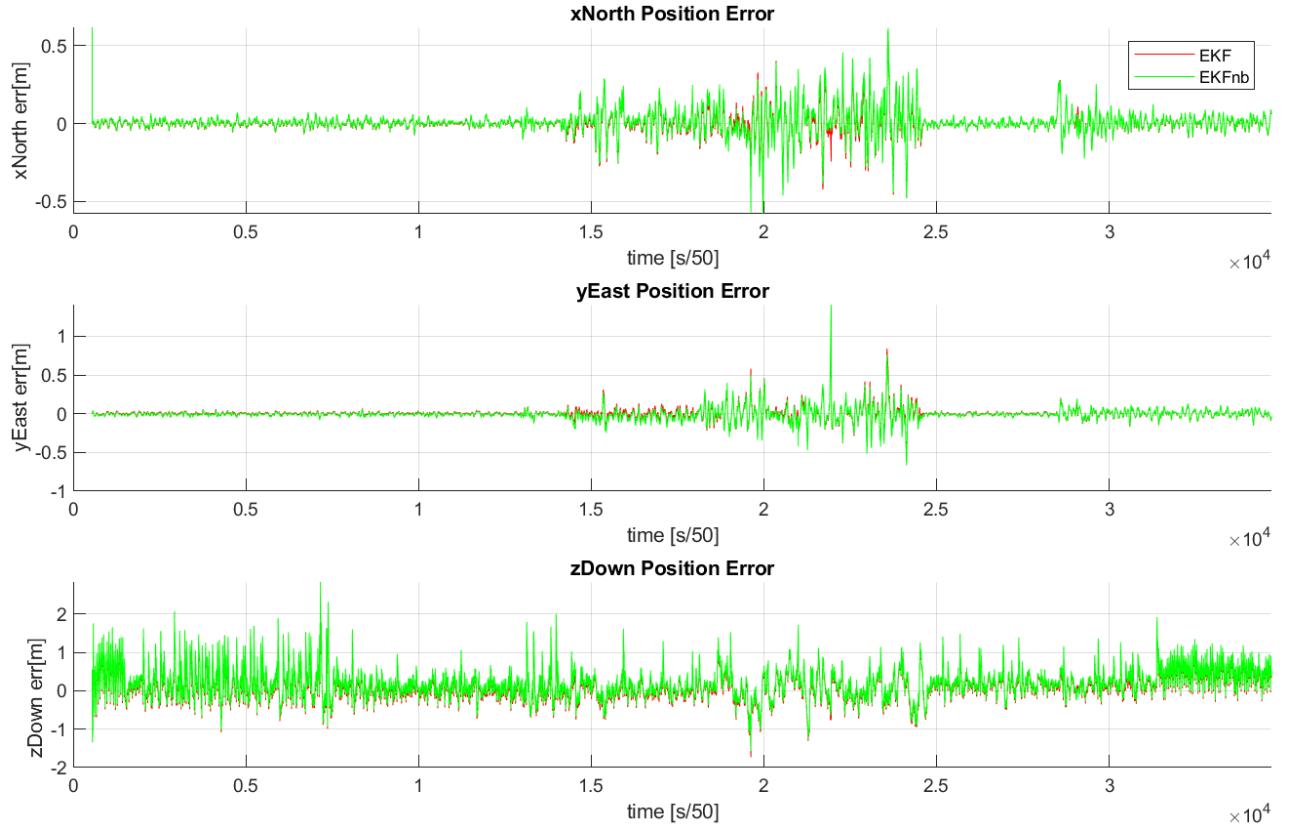


Figura 7.18: Errori assoluti delle componenti della stima di posizione

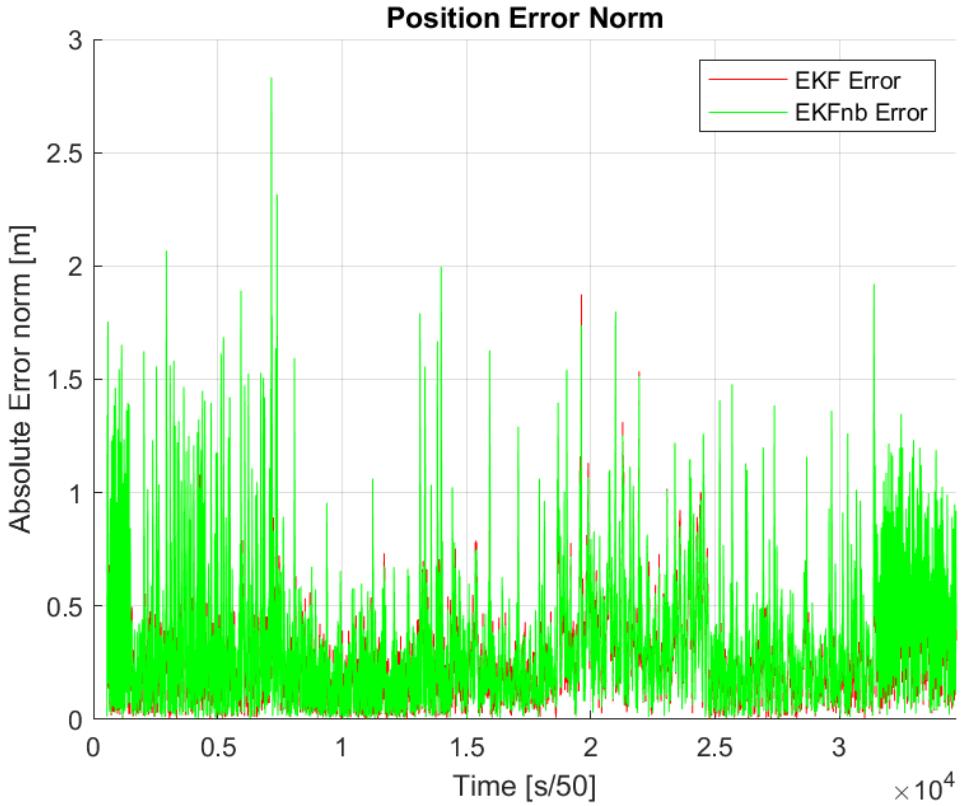


Figura 7.19: Norma dell'errore assoluto della stima di posizione

	$RMSE_x$	$RMSE_y$	$RMSE_z$	$RMSE_{tot}$	Unità
EKF	0.066803	0.074847	0.40314	0.41543	$m$
EKFnb	0.06897	0.075662	0.42299	0.43521	$m$

Tabella 7.7: RMSE della stima di posizione

Osservando i valori RMSE della stima di posizione, è ancora più evidente come il comportamento dei due filtri è quasi identico. La componente z della posizione rimane anche in questo caso la più rumorosa rispetto le altre due, dovuto anche al fatto che le misure usate sono esse stesse di natura più rumorose.

Si riportano di seguito per completezza alcune immagini del comportamento dei due filtri nei momenti in cui è simulato il malfunzionamento del GPS.

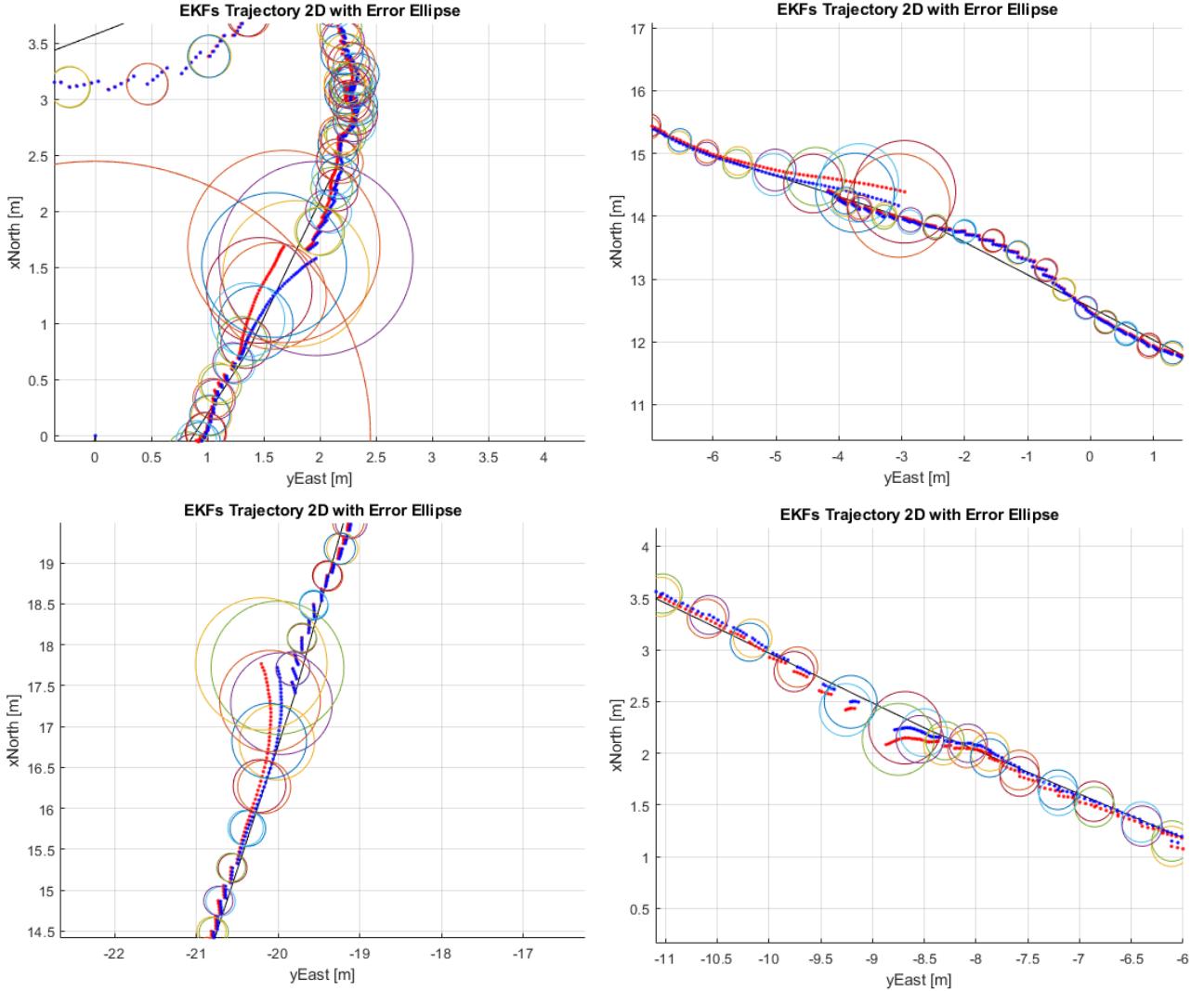


Figura 7.20: Posizione stimata del filtro con bias (blu) e senza (rosso) rispetto quella di riferimento (nero) con ellissi di covarianza dell'errore.

Si nota come, a differenza delle ellissi viste in precedenza nelle figure 6.53 per il veicolo su ruote, nel caso del drone tutte le ellissi di covarianza dell'errore sono circonference in quanto l'incertezza delle componenti x e y della posizione è calcolata in egual modo. Questo accade proprio per l'assenza della correzione basata sui vincoli dinamici del veicolo dato che un drone può muoversi in tutte le direzioni indipendentemente dal suo angolo di yaw.

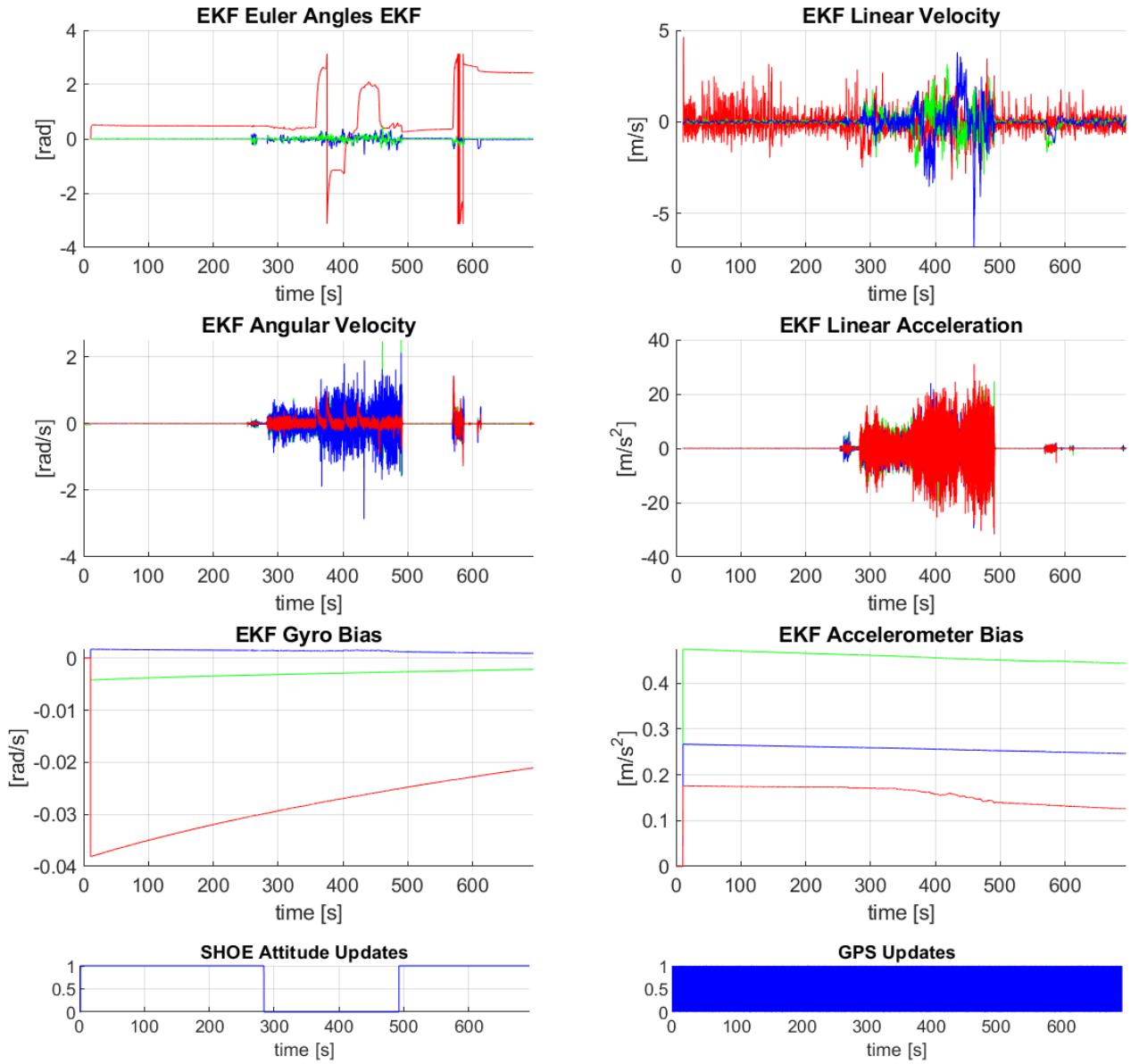


Figura 7.21: Variabili stimate dall'EKF con bias: assetto [yaw,pitch,roll], velocità lineare, velocità angolare, accelerazione e bias con componenti [Z,Y,X], i grafici in basso indicano quando le correzioni di assetto e del GPS avvengono.

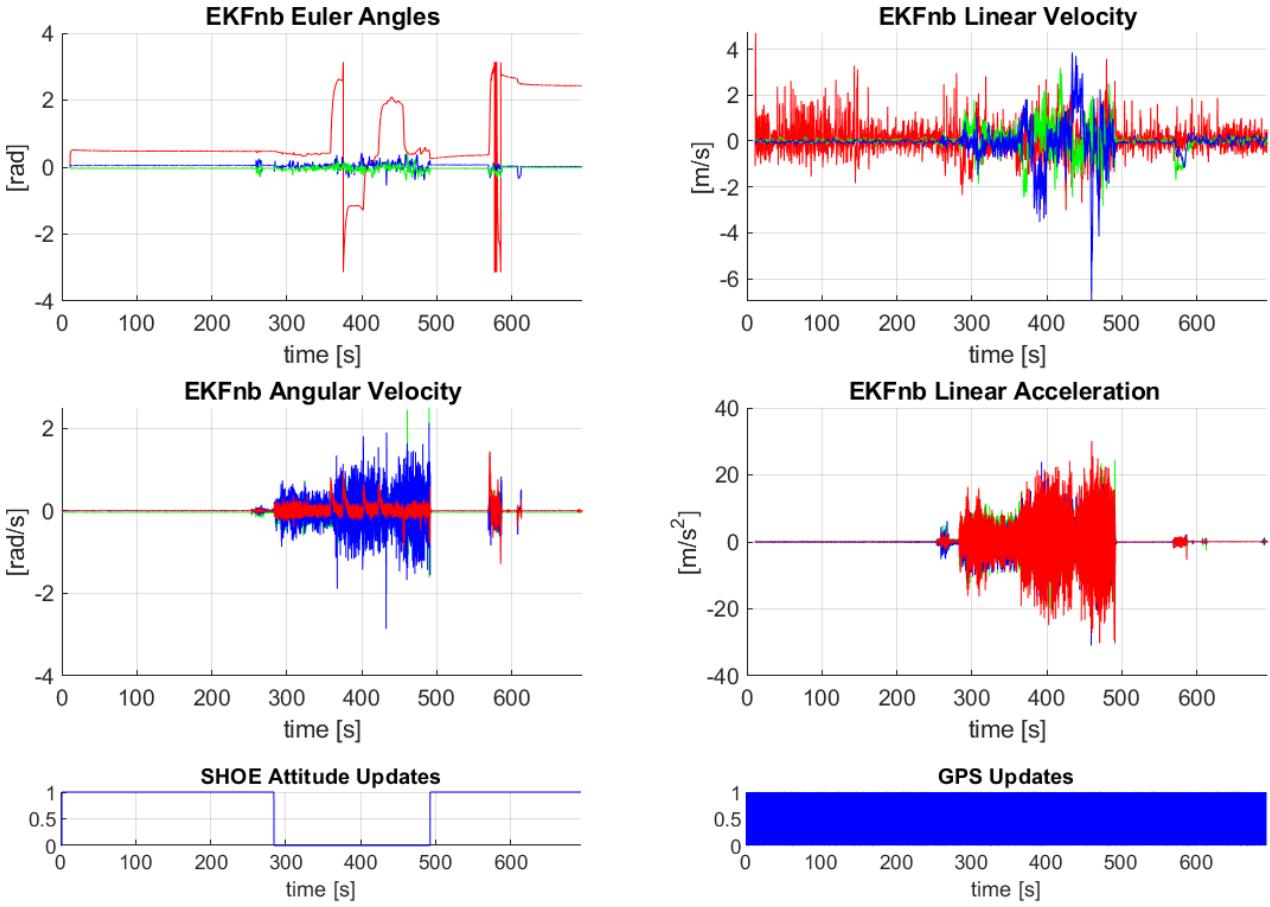


Figura 7.22: Variabili stimate dall'EKF con bias: assetto [yaw,pitch,roll], velocità lineare, velocità angolare, accelerazione con componenti [Z,Y,X], i grafici in basso indicano quando le correzioni di assetto e del GPS avvengono.

Per quanto riguarda l'assetto in particolare:

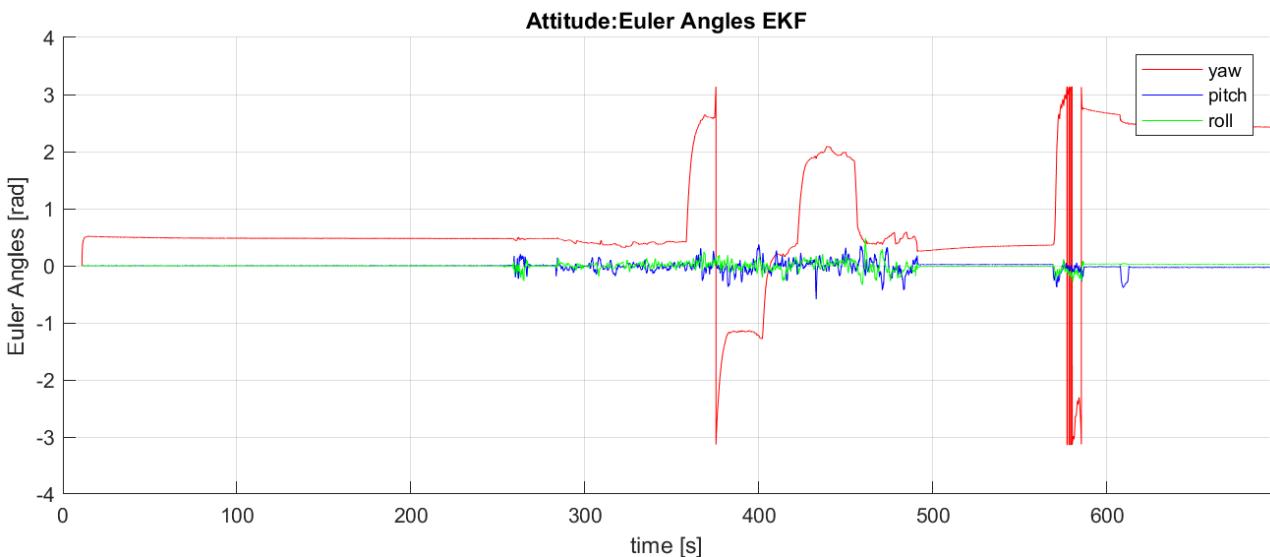


Figura 7.23: Angoli di Eulero di assetto stimati dal filtro EKF con bias

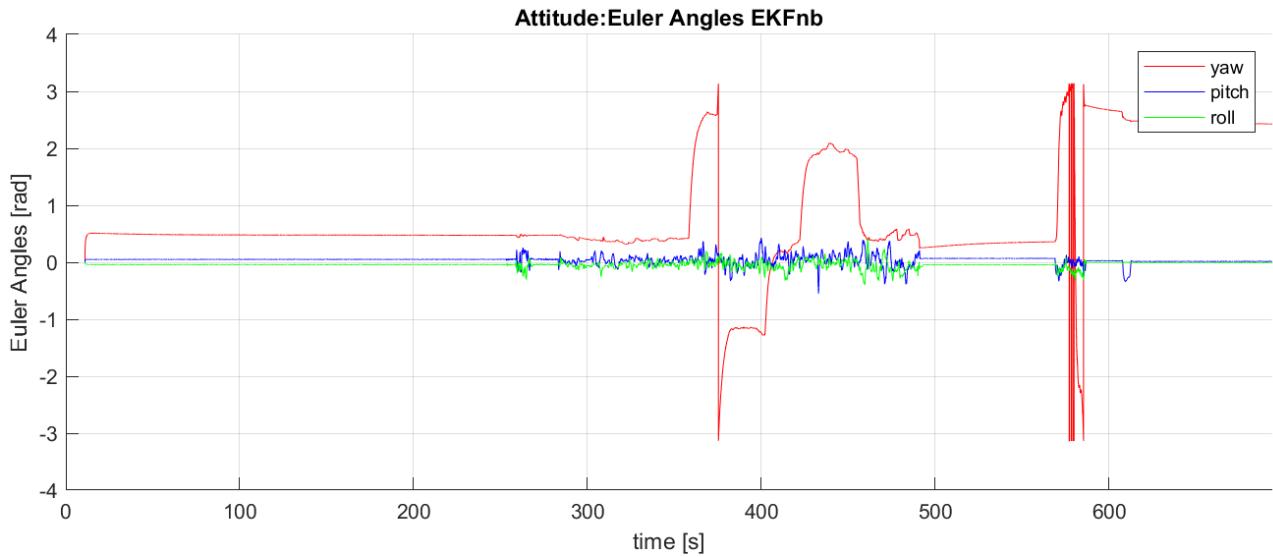


Figura 7.24: Angoli di Eulero di assetto stimati dal filtro EKF senza stima del bias

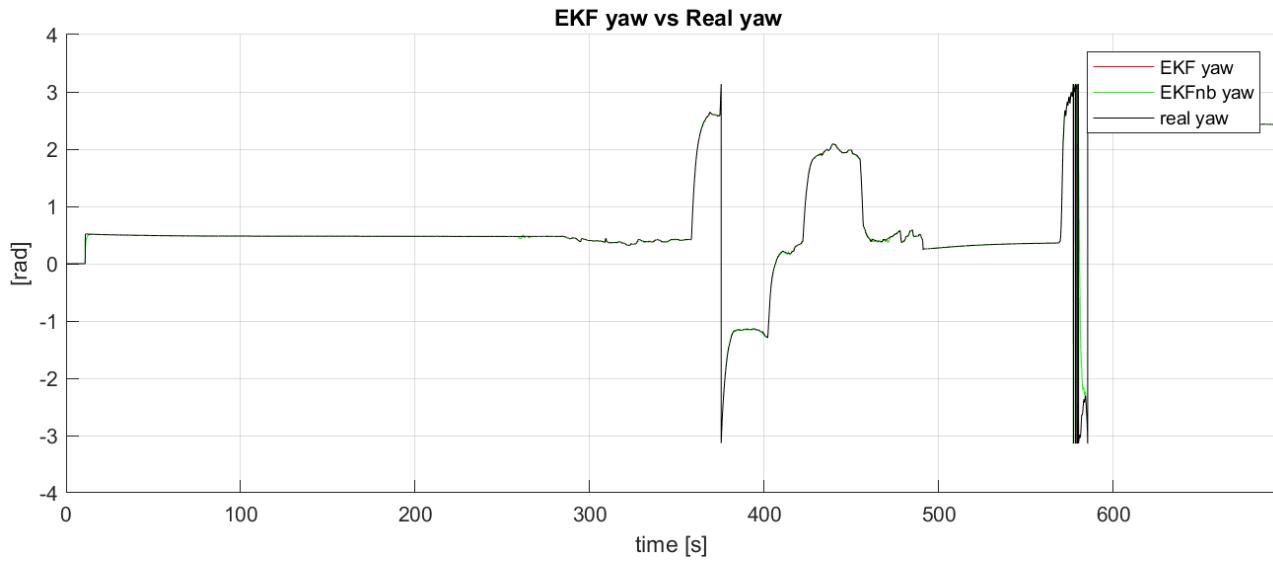


Figura 7.25: Confronto tra angoli di yaw stimati e reale

L’angolo di yaw è naturalmente ben stimato in quanto la misura utilizzata in fase correzione è proprio l’angolo di yaw di riferimento (come precedentemente discusso ad inizio capitolo), e essa stessa, così come per le misure del GPS, ha maggior peso per la stima dell’assetto rispetto ai sensori inerziali.

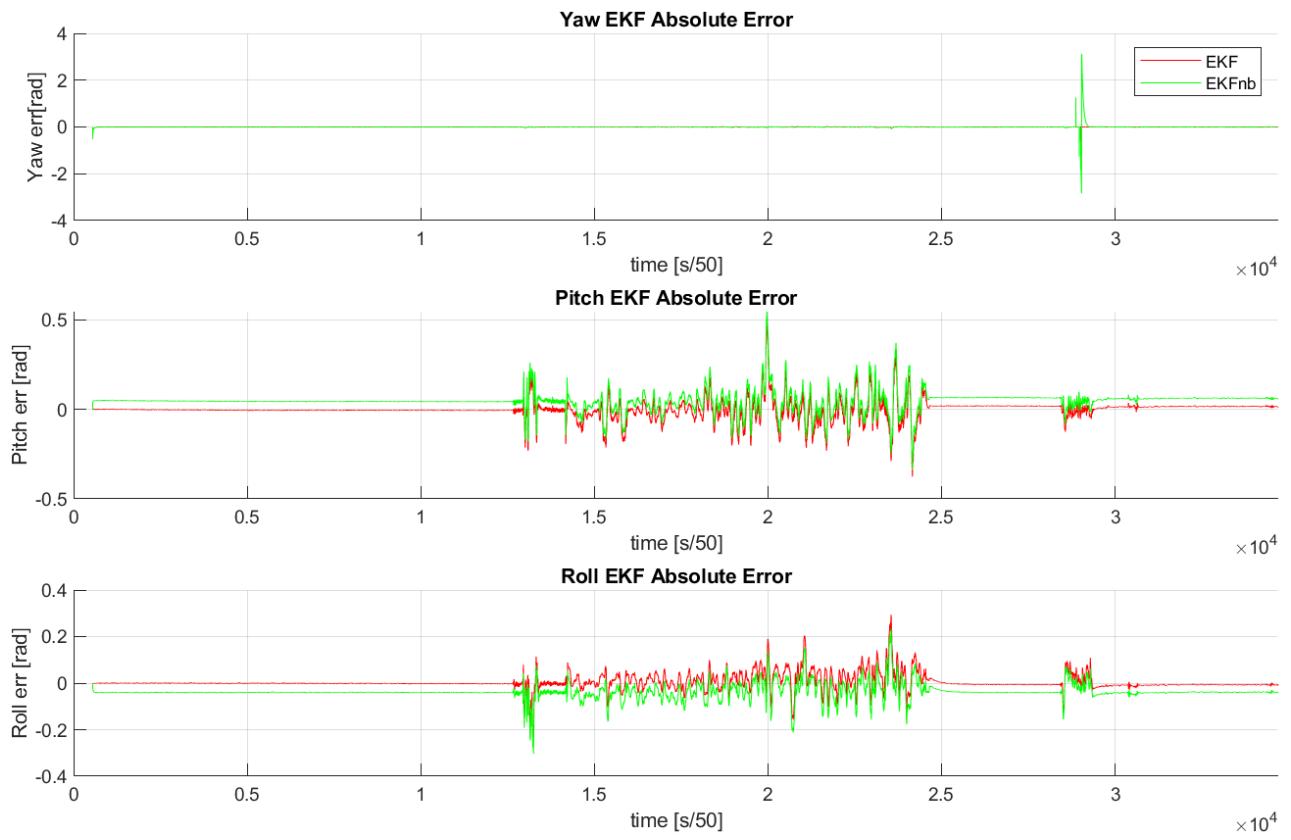


Figura 7.26: Errori assoluti degli angoli di Eulero di assetto stimati

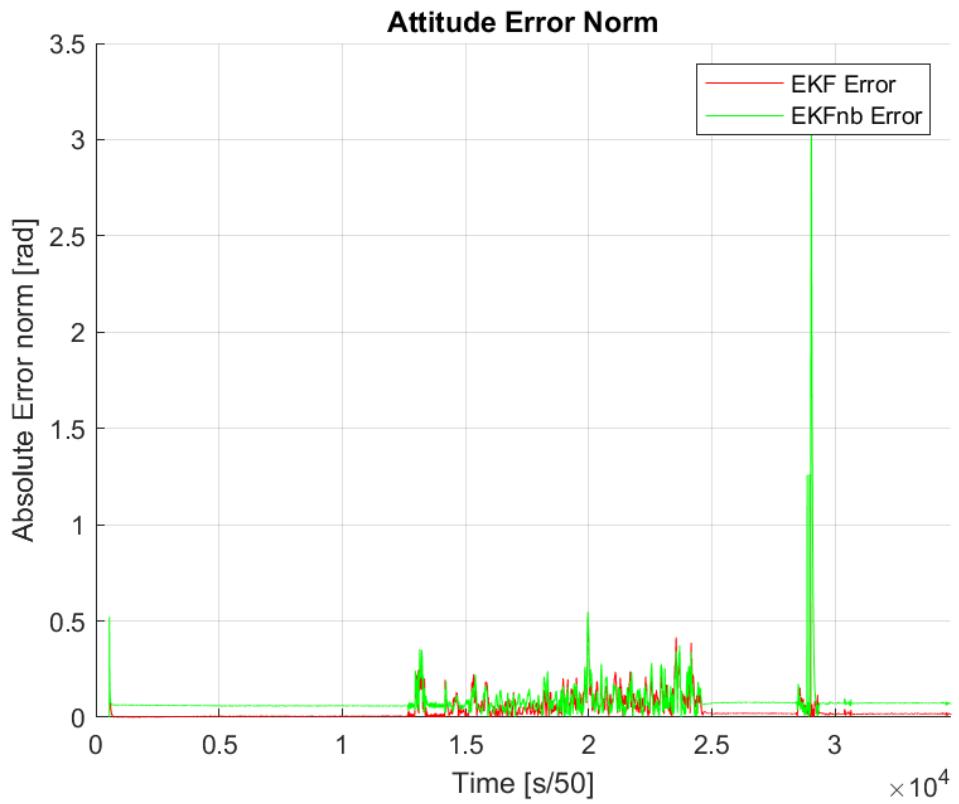


Figura 7.27: Norma dell'errore assoluto degli angoli di Eulero di assetto stimati

	$RMSE_{yaw}$	$RMSE_{pitch}$	$RMSE_{roll}$	$RMSE_{tot}$	Unità
EKF	0.014479	0.053076	0.033559	0.064443	rad
EKFnb	0.10494	0.070625	0.048745	0.13556	rad

Tabella 7.8: RMSE della stima dell’assetto

Dalle figure 7.25 e 7.26 si può notare che la stima degli angoli di roll e pitch nel caso di filtro senza la stima dei bias abbia un comportamento quasi identico a quelli del filtro completo, ma presentando un errore costante. Tale fenomeno si ripresenta quasi in ognuna delle simulazioni visto finora ed è correlato alla compensazione del bias per le misure di giroscopio e accelerometro. I valori riportati nella tabella 7.8 mostrano che anche la stima dell’assetto è migliorata rispetto al caso precedente.

In conclusione il nuovo tuning dei parametri è sicuramente più adatto a questo tipo di veicolo e misure disponibili, anche se in questa configurazione si suppone di avere a disposizione un’ottima misura del GPS, senza la quale l’accuratezza della stima ne risulterebbe sicuramente compromessa.

## Riferimenti bibliografici

- [1] R.Munguía. A gps-aided inertial navigation system in direct configuration. *Journal of Applied Research and Technology*, 12(4):803–814, 2014. ISSN 1665-6423.
- [2] I. Skog et al. Zero-velocity detection an algorithm evaluation. *IEEE Transactions on Biomedical Engineering*, 57(11):2657–2666, November 2010.
- [3] R. Munguia and A. Grau. A practical method for implementing an attitude and heading reference system. *International Journal of Advanced Robotic Systems*, 11, 2014.
- [4] Conversion between quaternions and euler angles. [https://en.wikipedia.org/wiki/Conversion\\_between\\_quaternions\\_and\\_Euler\\_angles](https://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Euler_angles), . (Last Edit: 03-Oct-2021).
- [5] D. M. Henderson. Euler angles, quaternions, and transformation matrices for space shuttle analysis. *NASA Mission Planning and Analysis Division*, June 9, 1977. Retrieved 24 May 2021.
- [6] José-Luis Blanco, Francisco-Angel Moreno, and Javier González. A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, 27(4):327–351, November 2009. ISSN 0929-5593. doi: 10.1007/s10514-009-9138-7. URL [http://www.mrpt.org/Paper:Malaga\\_Dataset\\_2009](http://www.mrpt.org/Paper:Malaga_Dataset_2009).
- [7] Euler angles. [https://en.wikipedia.org/wiki/Euler\\_angles](https://en.wikipedia.org/wiki/Euler_angles), . (Last Edit: 08-Nov-2021).
- [8] How to draw an error ellipse representing the covariance matrix? <https://www.visiondummy.com/2014/04/draw-error-ellipse-representing-covariance-matrix>. (Vincent Spruyt, 03-Apr-2014).