



UNIVERSITÀ DI PISA

Corso di Laurea Magistrale in Ingegneria Robotica
e dell'Automazione

Progetto del corso di Identificazione dei Sistemi Incerti:

Filtro di Kalman Esteso, Trasformata Unscented e Filtro di Rauch-Tung-Striebel di un drone vincolato da una barra rigida

Candidato:

Piero Campitelli

Indice

1	Introduzione	4
2	Modello dinamico del sistema	5
2.0.1	Parametri del sistema	6
3	I sensori	7
3.1	Accelerometro	7
3.2	DOA	7
3.3	Modello dei sensori	7
4	Validazione del modello	8
4.1	Prima simulazione	8
4.2	Seconda simulazione	9
4.3	Terza simulazione	9
4.3.1	Caso senza smorzamento	10
4.3.2	Caso con smorzamento	10
4.4	Quarta simulazione	11
4.5	Quinta simulazione	12
5	Validazione dei sensori	14
5.1	Sensore DOA	14
5.2	Accelerometri	15
6	Stima dello stato	18
6.1	EKF	18
6.1.1	Predizione	18
6.1.2	Correzione	19
6.2	Applicazione EKF	19
6.2.1	Discretizzazione del sistema	20
6.2.2	Ingressi del sistema	20
6.2.3	Condizioni iniziali	20
6.2.4	Sensori	21
6.2.5	Jacobiani	21
6.2.6	Simulazioni	22
6.2.7	Test di bianchezza	23

6.3	UKF	26
6.3.1	Applicazione UKF	27
6.3.2	Predizione	28
6.3.3	Correzione	28
6.3.4	Simulazioni	29
6.3.5	Test di bianchezza	30
6.4	Distanza di Mahalanobis	32
6.4.1	Senza usare la distanza di Mahalanobis	32
6.4.2	Usando la distanza di Mahalanobis	33
6.5	Conclusioni	36
7	Regolarizzazione	37
7.0.1	Analisi dei risultati	37

1 Introduzione

La presente relazione ha lo scopo di mostrare il lavoro svolto ed i risultati ottenuti durante lo studio del progetto del corso "Identificazione dei sistemi incerti". I principali obiettivi del progetto sono:

- **Modellazione dinamica:** Scrittura del modello dinamico di un drone con due eliche vincolato al terreno tramite una barra rigida.
- **Scelta e modellazione dei sensori:** Scelta delle grandezze fisiche da misurare e degli opportuni sensori.
- **Filtro di Kalman Esteso(EKF):** Progettazione del Filtro di Kalman Esteso per la stima dello stato del sistema.
- **Filtro di Kalman con Trasformata Unscented(UKF):** Implementazione del Filtro di Kalman con Trasformata Unscented per la stima dello stato del sistema.
- **Confronto tra i due filtri utilizzati:** Analisi delle performance e robustezza dei due filtri implementati.
- **Algoritmo di smoothing di Rauch–Tung–Striebel:** Implementazione dell'algoritmo di smoothing di Rauch–Tung–Striebel sullo stato stimato del sistema.

2 Modello dinamico del sistema

Il sistema in analisi è rappresentato nella seguente Fig. 2.1, il drone di massa \mathbf{m}_R è vincolato da una barra rigida: ad un estremo è attaccata al centro di massa del drone ed all'altra estremità al terreno consentendo un moto circolare intorno alla cerniera del terreno (angolo ϕ) ed un moto rotatorio intorno a quella del centro di massa (angolo θ). Il velivolo è dotato di due eliche identiche, per semplicità la spinta che esercita ogni singola elica viene rappresentata da un sistema equivalente dove la forza \mathbf{f}_R complessiva viene applicata nel centro di massa.

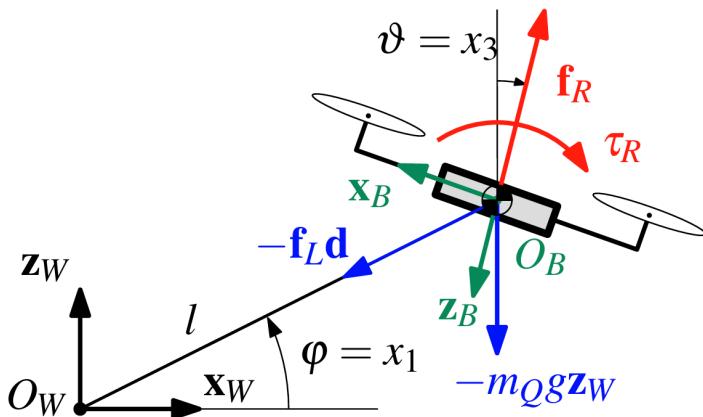


Figura 2.1: Sistema drone vincolato dalla barra rigida.

Il sistema può essere descritto dalle equazioni della dinamica di Eulero-Lagrange:

$$M(\mathbf{q})\ddot{\mathbf{q}} + D\dot{\mathbf{q}} + g(\mathbf{q}) = Q(\mathbf{q})u \quad (2.1)$$

dove:

- \mathbf{q} è il vettore: $\begin{bmatrix} \phi \\ \theta \end{bmatrix}$ e $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ le sue derivate.
- $M(\mathbf{q})$ è la matrice delle inerzie:

$$\begin{bmatrix} J_\phi & 0 \\ 0 & J_\theta \end{bmatrix}$$

dove J_ϕ e J_θ sono i momenti d'inerzia lungo l'asse perpendicolare al piano rispettivamente dell'asta di lunghezza \mathbf{l} che ruota intorno all'origine del *world frame* \mathbf{O}_W e del drone che ruota attorno all'origine del *body frame* \mathbf{O}_B .

- \mathbf{D} è il termine di smorzamento:

$$\begin{bmatrix} \mathbf{D}_\phi & 0 \\ 0 & \mathbf{D}_\theta \end{bmatrix}$$

\mathbf{D}_ϕ è il termine di smorzamento dovuto all'attrito dell'aria associato alla velocità $\dot{\phi}$, mentre \mathbf{D}_θ alla velocità $\dot{\theta}$.

- \mathbf{g} è il vettore delle forze gravitazionali:

$$\begin{bmatrix} m_R l g d^\perp \cdot e_3 \\ 0 \end{bmatrix}$$

- $\mathbf{Q}(\mathbf{q})$ esprime come sono legati tra di loro gli ingressi del sistema:

$$\begin{bmatrix} -l \mathbf{R}_B^W e_3 \cdot d^\perp & 0 \\ 0 & 1 \end{bmatrix}$$

la matrice \mathbf{R}_B^W esprime la rotazione intorno l'asse y dal *body frame* al *world frame*.

- \mathbf{u} è il vettore degli ingressi del sistema:

$$\begin{bmatrix} \mathbf{f}_R \\ \boldsymbol{\tau}_R \end{bmatrix}$$

Definiamo infine lo **stato** \mathbf{x} del sistema: $\mathbf{x} = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \end{bmatrix}$

2.0.1 Parametri del sistema

Di seguito vengono riportati i parametri fisici del sistema con le rispettive incertezze:

- $m_R = 2kg$, $\sigma_{m_R} = 0.03kg$ massa ed incertezza del drone
- $l = 2m$, $\sigma_l = 0.26m$ lunghezza ed incertezza della barra rigida
- $J_R = 0.25kg \cdot m^2$, $\sigma_{J_R} = 0.04kg \cdot m^2$ momento ed incertezza del momento d'inerzia del robot
- $D_\phi = 0.5kg/s$, $D_\theta = 0.025kg/s$ coefficienti di smorzamento dell'aria
- $g = 9.81m/s^2$ accelerazione di gravità
- $\sigma_{f_R} = 0.03N$ incertezza sull'ingresso di thrust
- $\sigma_{\tau_R} = 0.01Nm$ incertezza sulla coppia rotazionale di ingresso

3 I sensori

Per leggere i componenti del vettore di stato sono state scelte due tipologie di sensori: l'**accelerometro** ed il sensore che misura la **Direction of Arrival(DOA)**

3.1 Accelerometro

L'accelerometro scelto è di tipo biassiale, misura l'accelerazione sia sull'asse x che sull'asse z. Supponiamo di lavorare con degli accelerometri che funzionano ad una frequenza:

$$f_{acc} = 1000Hz.$$

con un'incertezza:

$$\sigma_{acc} = 0.1m/s^2$$

3.2 DOA

La direction of arrival viene misurata tramite una camera posta in (S_{sens}, L_{sens}) che riesce a seguire il baricentro del drone.

$$S_{sens} = L_{sens} = 5m$$

$$f_{DOA} = 120Hz$$

$$\sigma_{DOA} = 0.1rad$$

3.3 Modello dei sensori

Mettendo insieme tutti i sensori si può scrivere il seguente vettore di misure:

$$y_k = \begin{bmatrix} y_{DOA} \\ y_{ax} \\ y_{az} \end{bmatrix} = \begin{bmatrix} \text{atan}2\left(\frac{S_{sens}-l\cos(\phi)}{L_{sens}-l\sin(\phi)}\right) \\ -g\sin(\theta) \\ g\cos(\theta) \end{bmatrix} \quad (3.1)$$

4 Validazione del modello

Per verificare che il modello matematico rappresenti in modo realistico l'andamento del sistema sono state effettuate delle simulazioni in assenza di disturbi ed incertezze.

4.1 Prima simulazione

Si è considerato il sistema nella condizione iniziale $x_0 = [0 \ 0 \ 0 \ 0]^T$, ovvero il drone all'istante iniziale coincide con l'asse \mathbf{X}_W e forma un angolo $\boldsymbol{\theta}$ di zero gradi con l'asse \mathbf{Z}_B . Sul drone agisce solamente la forza peso in quanto gli ingressi forniti in questa simulazione impongono un thrust nullo (eliche non in rotazione). Inoltre i coefficienti di smorzamento \mathbf{D}_ϕ e \mathbf{D}_θ in questo caso sono stati scelti nulli, quindi ci si aspetta che il drone cada verso il basso oscillando con un moto periodico tra $\phi = -\pi$ e $\phi = 0$. Effettuando la simulazione si ottengono gli andamenti riportati in Fig. 4.1.

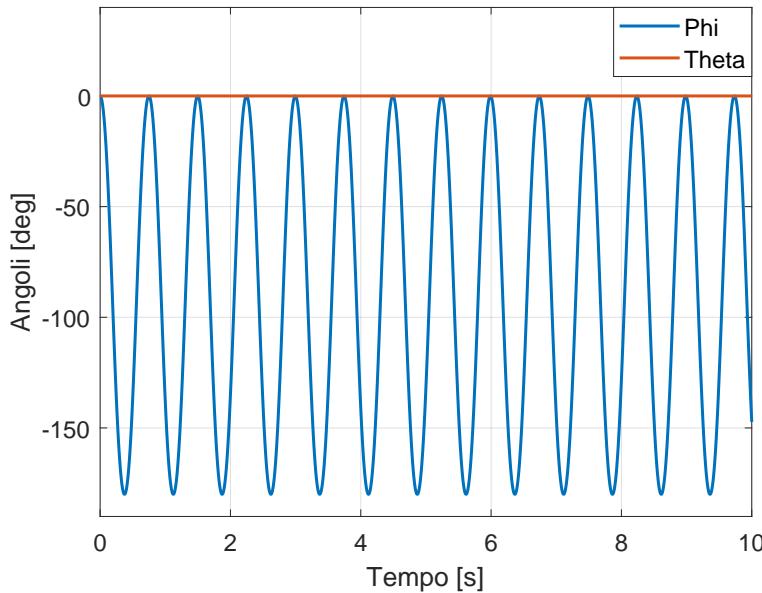


Figura 4.1: Simulazione del sistema con $D_\phi = 0 \text{ kg/s}$ e $D_\theta = 0 \text{ kg/s}$, $f_R = 0 \text{ N}$, $x_0 = [0 \ 0 \ 0 \ 0]^T$.

Come ci si aspettava il sistema parte con un angolo ϕ uguale a zero ed a causa della forza peso senza smorzamenti inizia ad oscillare tra $\phi = -\pi$ e $\phi = 0$, mentre l'angolo θ rimane nullo.

4.2 Seconda simulazione

Nella seconda simulazione è stato ripetuto l'esperimento imponendo questa volta che gli smorzamenti non siano nulli, i valori scelti sono: $D_\phi = 0,5 \text{ kg/s}$ e $D_\theta = 0,025 \text{ kg/s}$, questi valori sono stati scelti dopo diversi test in cui si cercavano dei valori realistici che smorzassero il sistema entro un limite di tempo ragionevole come nella realtà. Si prevede che il drone si stabilizzi intorno alla posizione di equilibrio stabile $\phi = -\pi/2$. Il risultato ottenuto è riportato in Fig. 4.2.

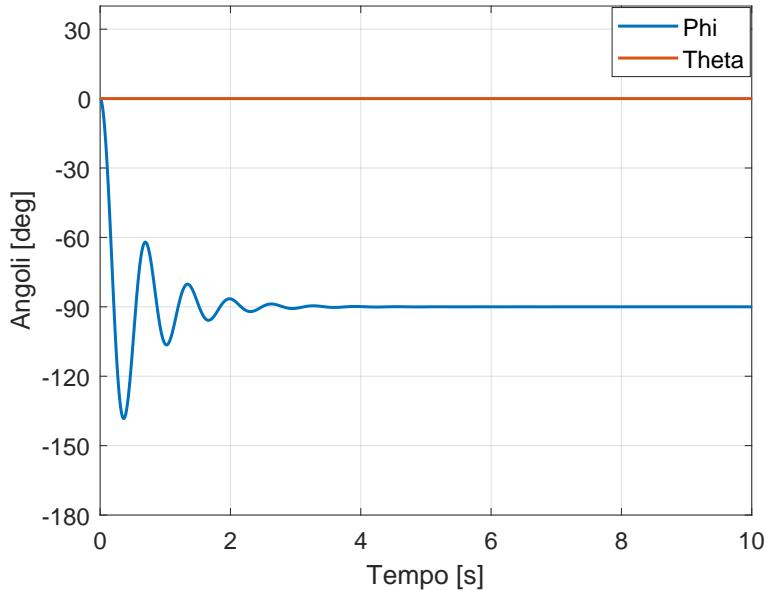


Figura 4.2: Simulazione del sistema con $D_\phi = 0,5 \text{ kg/s}$ e $D_\theta = 0,025 \text{ kg/s}$, $f_R = 0$, $x_0 = [0 \ 0 \ 0 \ 0]^T$.

In questo caso il sistema parte sempre dalle stesse condizioni iniziali cadendo verso il basso, assestandosi intorno alla posizione di equilibrio $\phi = -\pi/2$.

4.3 Terza simulazione

Per validare il comportamento del robot quando si fornisce una forza f_R è possibile ripetere gli esperimenti appena visti (con smorzamento e senza) per studiarne l'andamento.

4.3.1 Caso senza smorzamento

In Fig. 4.4 è riportato l'andamento nel caso in cui è presente un **thrust** f_R pari a 1,5 volte la **forza peso del drone** ($m_R g$), dunque in questo caso l'angolo ϕ non diminuisce, bensì aumenta e dato che non è presente smorzamento, come nella Sezione 4.1 il sistema continua ad oscillare.

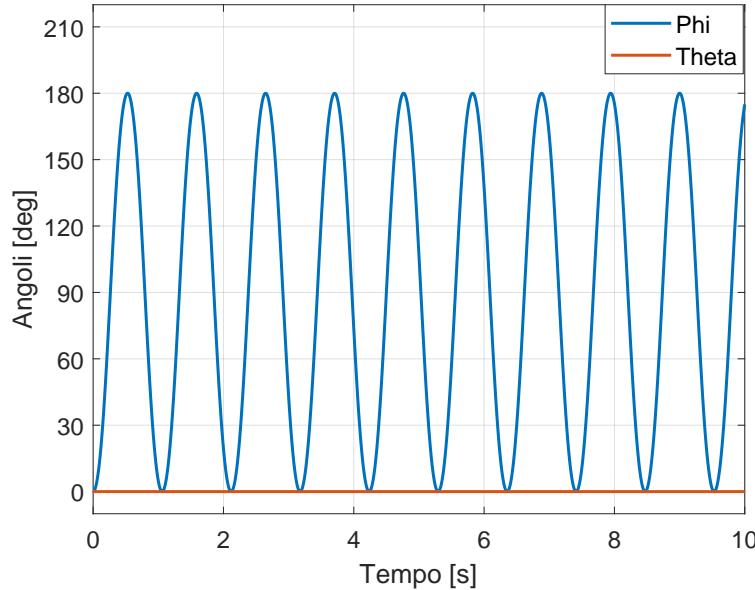


Figura 4.3: Simulazione del sistema con $D_\phi = 0 \text{ kg/s}$ e $D_\theta = 0 \text{ kg/s}$, $f_R = 1,5m_R g$, N , $x_0 = [0 \ 0 \ 0 \ 0]^T$.

4.3.2 Caso con smorzamento

Usando la spinta delle eliche come nel caso precedente possiamo verificare che nella Fig. 4.4 si ha un andamento simile come nella Sezione 4.2 però in questo caso, come giustamente ci si aspettava l'angolo ϕ a regime raggiunge la posizione di equilibrio pari a $\pi/2$.

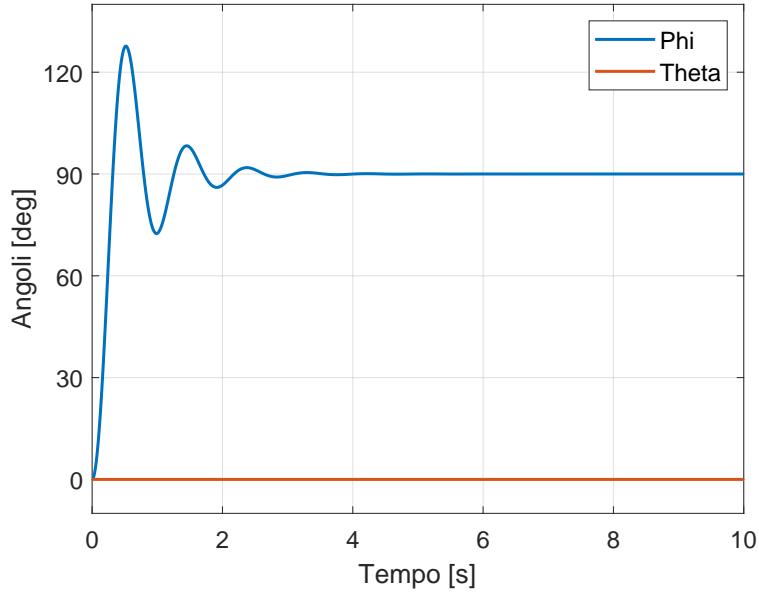


Figura 4.4: Simulazione del sistema con $D_\phi = 0,5 \text{ kg/s}$ e $D_\theta = 0,025 \text{ kg/s}$, $f_R = 1,5m_Rg N$, $x_0 = [0 \ 0 \ 0 \ 0]^T$.

E' stato dunque verificato il fatto che l'angolo ϕ agisce coerentemente con il modello reale. Nelle simulazioni appena riportate anche l'angolo θ sembra essere coerente (la **forza peso** ed il **thrust** generano una coppia a braccio nullo), però bisogna studiarne il comportamento in maniera più approfondita nel caso in cui si applica una **coppia esterna τ_R** .

4.4 Quarta simulazione

Nella Fig. 4.5 è sono riportati gli andamenti del sistema nel caso in cui $x_0 = [-\pi/2 \ 0 \ 0 \ 1]^T$.

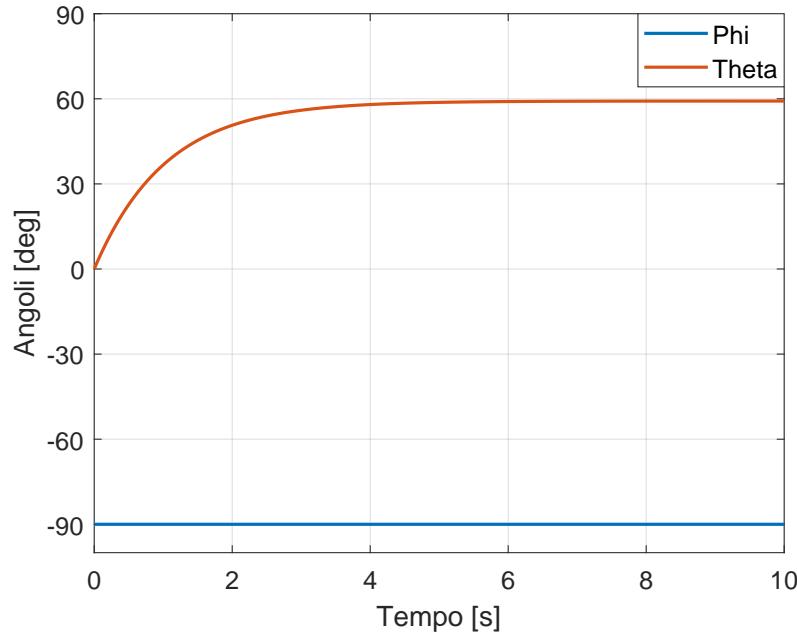


Figura 4.5: Simulazione del sistema con $D_\phi = 0,5 \text{ kg/s}$ e $D_\theta = 0,025 \text{ kg/s}$, $f_R = 0$ N , $\tau_R = 0$, $x_0 = [-\pi/2 \ 0 \ 0 \ 1]^T$

L'angolo θ cresce dato che è presente una **velocità angolare** iniziale e successivamente si assesta (a causa dello smorzamento) intorno a $\pi/3$, mentre l'angolo ϕ rimane in posizione di equilibrio.

4.5 Quinta simulazione

Se in ingresso al sistema si fornisce una coppia τ_R costante si ha una rotazione rispetto all'asse Z_B che porta l'angolo θ ad aumentare continuamente.

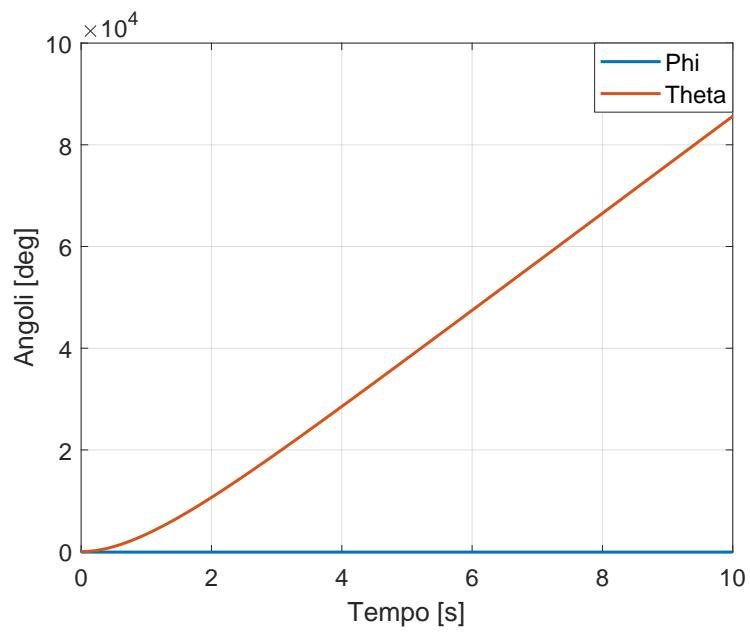


Figura 4.6: Simulazione del sistema con $D_\phi = 0,5 \text{ kg/s}$ e $D_\theta = 0,025 \text{ kg/s}$, $f_R = 0$
 N , $\tau_R = 0,1$, $x_0 = [-\pi/2 \ 0 \ 0 \ 0]^T$

5 Validazione dei sensori

Oltre a verificare la corretta implementazione del modello è opportuno analizzare se anche i sensori forniscono delle accurate misurazioni prima di procedere con la stima dello stato.

5.1 Sensore DOA

Il seguente grafico Fig. 5.1 riporta nella parte superiore l'andamento dell'angolo ϕ e nella parte inferiore l'angolo misurato a_2 .

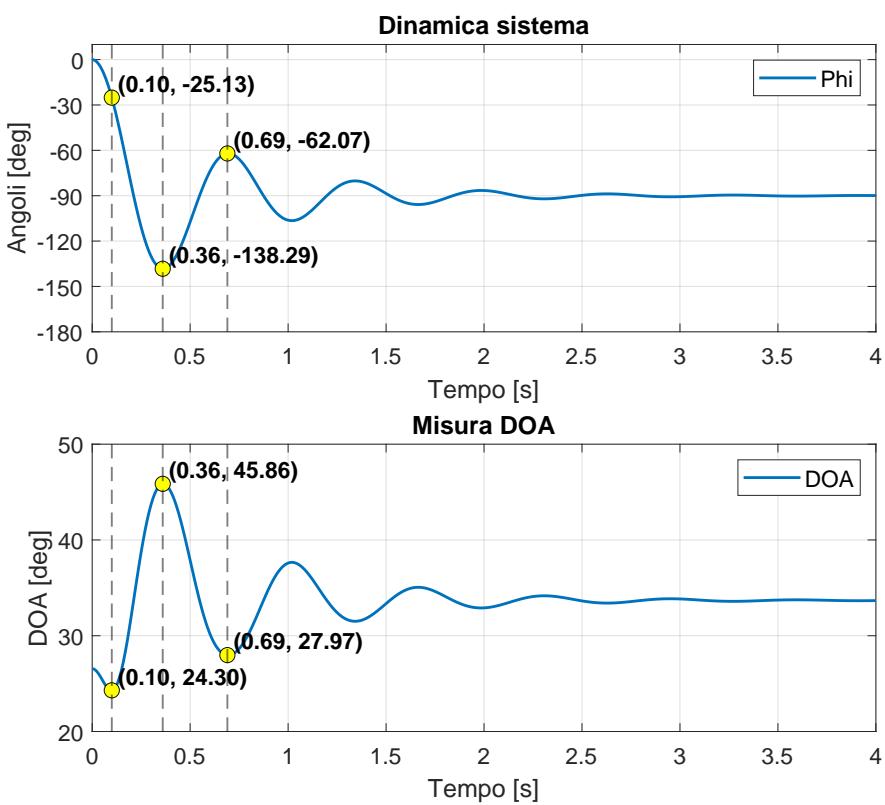


Figura 5.1: Confronto tra andamento del sistema simulato e misurazioni del sensore DOA

Il sistema parte con una condizione iniziale $x_0 = [0 \ 0 \ 0 \ 0]^T$ con entrambi gli ingressi nulli, si ha che ϕ parte da zero ed arriva a -90° con una piccola sovraelargazione. Di conseguenza il sensore inizialmente parte da un valore poco inferiore di 30° per poi iniziare a decrescere fino all'istante $t = 0, 10\text{s}$ dove raggiunge il valore minimo di $24, 30^\circ$, dopodiché torna a crescere raggiungendo il valore massimo in $t = 0, 36\text{s}$ dove vale $45, 86^\circ$. Si può notare che a causa dell'oscillazione intorno al punto di equilibrio il sensore misura correttamente dei valori che oscillano tra il valore minimo e massimo appena visti per poi assestarsi progressivamente intorno al valore di regime. Per una maggior comprensione è riportata anche l'immagine in Fig. 5.2 dove si distinguono meglio gli angoli trattati, in particolar modo il punto **verde** è il punto di **minimo** del grafico in Fig. 5.1, forma l'angolo più piccolo tra il punto sulla circonferenza (la traiettoria che compie il drone essendo vincolato dalla barra) e l'asse verticale tratteggiato (la distanza e l'altezza dove è posto il sensore fisso). Il punto **viola** è il punto di massimo ($45, 86^\circ$), mentre il punto **rosso** è il valore che il sensore misura a regime ($33, 69^\circ$).

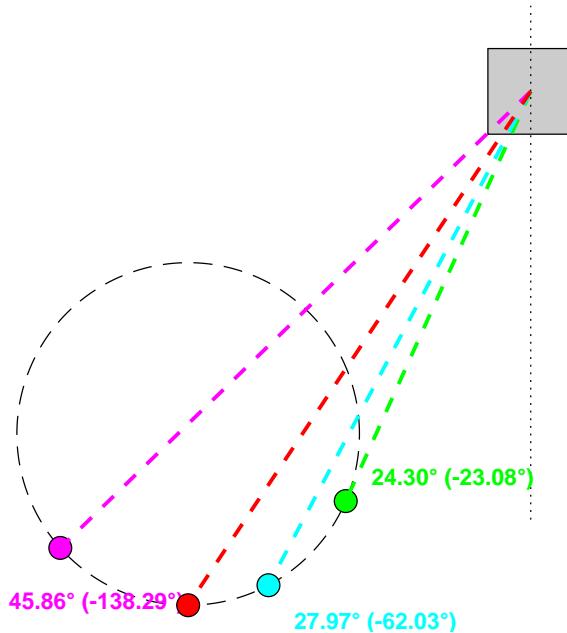


Figura 5.2: Confronto tra andamento del sistema simulato e misurazioni del sensore DOA

5.2 Accelerometri

Per studiare il corretto funzionamento degli accelerometri è stata svolta la simulazione in Fig. 5.3 dove si è partiti dalla seguente condizione iniziale: $x_0 = [0 \ 0 \ 0 \ 3]^T$, con $\tau_R = 0$ e $f_R = m_R g$. Il drone parte nella configurazione **classica** (eliche ri-

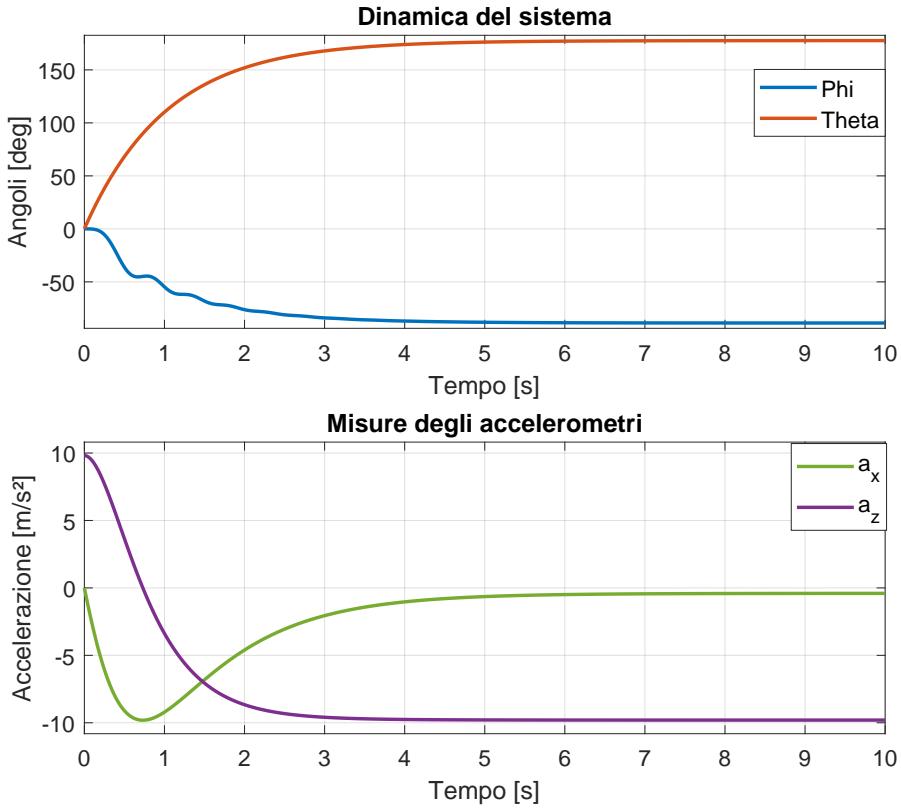


Figura 5.3: Confronto tra andamento del sistema simulato e misurazioni degli accelerometri

volte verso l'alto), dopodichè a causa della condizione iniziale posta sulla velocità angolare $\dot{\theta}$ ruota fino a raggiungere circa i 180° . L'angolo ϕ invece con delle piccole oscillazioni, dovute alla combinazione della rotazione intorno a θ ed al **thrust** a regime raggiunge una configurazione di -90° . Nel grafico sottostante si può vedere come gli accelerometri catturano perfettamente questi andamenti, infatti all'inizio l'accelerazione sull'**asse x** è di **0m/s²** mentre sull'**asse z** è **9,81m/s²**.

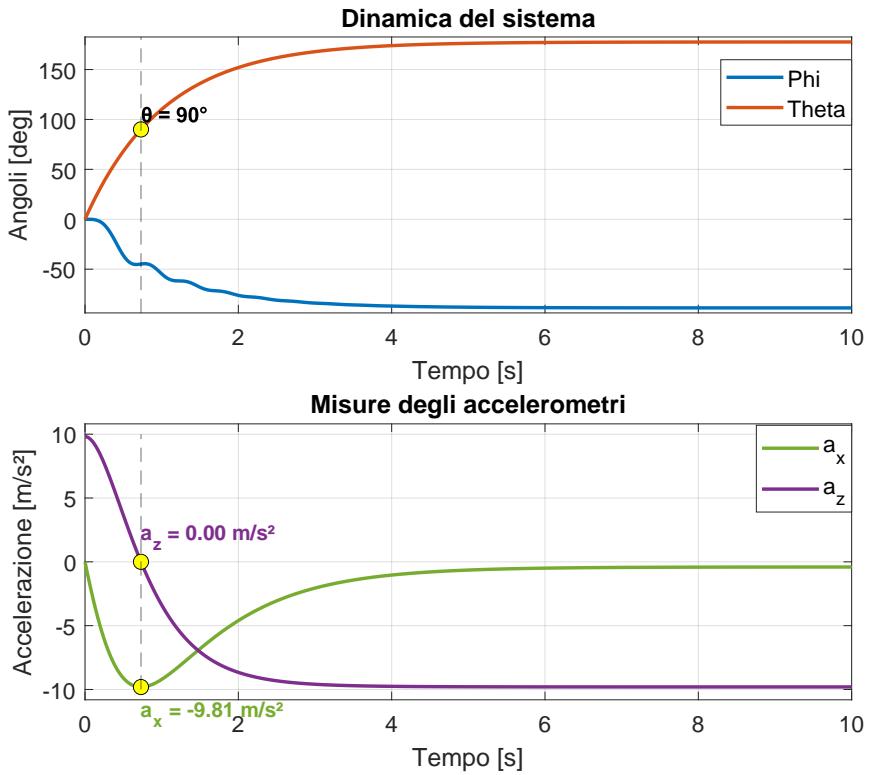


Figura 5.4: Confronto tra andamento del sistema e misurazioni degli accelerometri

Nel momento in cui l'angolo θ è ruotato di 90° come si può vedere in Fig. 5.4 l'accelerazione $a_x = -9,81 m/s^2$ e l'accelerazione $a_z = 0 m/s^2$ coerentemente con quanto ci si aspettava. In ultima analisi quando θ raggiunge la posizione di 180° l'accelerazione lungo x torna a crescere fino ad assestarsi a 0° , mentre lungo l'asse z continua decrescere fino a $-9,81 m/s^2$.

6 Stima dello stato

Dopo aver verificato che il modello dinamico del sistema e dei sensori è stato implementato fedelmente rispetto alla realtà, è possibile applicare tecniche di stima dello stato, queste, permettono di combinare la conoscenza del modello dinamico affetto dai disturbi di processo con le misure dei sensori a cui si sovrappone del rumore, in modo da ottenere una stima dello stato più accurata. L'algoritmo cardine per la stima dello stato di sistemi **lineari** è il **Filtro di Kalman**. Da esso discendono altre tipologie di filtri come ad esempio l'**Extended Kalman Filter (EKF)** e l'**Unscented Kalman Filter (UKF)**, i quali sono una generalizzazione di quello classico che estendono la loro efficacia anche nel caso di sistemi dinamici **non lineari**. La stima dello stato del drone è proprio un problema che presenta delle **non linearità**, per questo motivo sono state implementate queste ultime due tipologie di filtro.

6.1 EKF

L'Extended Kalman Filter sfrutta la teoria della **linearizzazione** combinandola con l'implementazione del filtro classico, un algoritmo ricorsivo che alterna l'esecuzione di due funzioni:

- **Predizione**
- **Correzione**

6.1.1 Predizione

Nella fase di predizione viene aggiornata la stima dello stato e la covarianza ad essa associata basandosi sulla conoscenza del modello del sistema $\mathbf{f}_k(\mathbf{x}_k, \mathbf{w}_k)$. L'aggiornamento della stima dello stato e della matrice di covarianza avviene attraverso le seguenti due formule:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{f}_k(\mathbf{x}_{k|k}, \mathbf{w}_k) \quad (6.1)$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^T + \mathbf{D}_k \mathbf{Q}_k \mathbf{D}_k^T \quad (6.2)$$

dove:

- $\mathbf{P}_{k+1|k}$: la matrice di covarianza dell'errore di stima al passo $k+1$ basandosi sulle misure fino all'istante k

- F_k : la matrice linearizzata della funzione f_k rispetto allo stato x_k valutata in $x_k = \hat{x}_{k|k}$ e $w_k = 0$ al passo k
- D_k : la matrice linearizzata della funzione f_k rispetto al disturbo di processo w_k valutata in $x_k = \hat{x}_{k|k}$ e $w_k = 0$ al passo k
- Q_k : la matrice di covarianza del disturbo di processo al passo k

6.1.2 Correzione

Nella fase di correzione viene aggiornato lo stato stimato e la matrice di covarianza dell'errore di stima utilizzando le ultime misure ricevute dai sensori sfruttando le seguenti equazioni:

$$e_k = y_k - h(\hat{x}_{k|k-1}, 0) \quad (6.3)$$

con:

- e_k : l'innovazione al passo k
- y_k : il vettore di misura dei sensori al passo k
- $h(\hat{x}_{k|k-1}, 0)$: il modello del sensore che propaga lo stato predetto nella fase di predizione $\hat{x}_{k|k-1}$

$$S_k = H_k P_{k|k-1} H_k^T + M_k R_k M_k^T \quad (6.4)$$

- S_k : matrice di covarianza dell'innovazione al passo k
- H_k : matrice linearizzata della funzione h_k rispetto allo stato x_k valutata in $x_k = \hat{x}_{k|k-1}$ e $v_k = 0$ al passo k
- $P_{k|k-1}$: matrice di covarianza della predizione al passo k basandosi sulle misure fino al passo k-1
- M_k : matrice linearizzata della funzione h_k rispetto allo stato v_k valutata in $x_k = \hat{x}_{k|k-1}$ e $v_k = 0$ al passo k

6.2 Applicazione EKF

Per applicare l'Extended Kalman Filter per la stima del drone il primo passo da compiere dato che l'algoritmo lavora in tempo discreto è quello di scrivere discretizzare il modello dinamico.

6.2.1 Discretizzazione del sistema

Per portare il sistema in tempo discreto è stato usato il metodo di **Eulero in avanti**:

$$x_{k+1} = x_k + T_c f(t_k, x_k) \quad (6.5)$$

dove T_c è il periodo di campionamento e $f(t_k, x_k)$ è la derivata dello stato. Si ottiene che il sistema discretizzato è:

$$x_{k+1} = \begin{bmatrix} \dot{\phi}_k + T_c \dot{\phi}_k \\ \dot{\phi}_k + T_c \left(-\frac{g \cos(\phi_k)}{(l+w_{lk})} - \frac{D_\varphi \dot{\phi}_k}{(m_R+w_{mk})(l+w_{lk})^2} \right) \\ \theta_k + T_c \dot{\theta}_k \\ \dot{\theta}_k - T_c \frac{D_\psi \dot{\theta}_k}{(J_R+w_{jk})} \end{bmatrix} + T_c \begin{bmatrix} 0 \\ \frac{\cos(\phi_k+\theta_k)(u_{1k}+w_{u1k})}{(m_R+w_{mk})(l+w_{lk})} \\ 0 \\ \frac{(u_{2k}+w_{u2k})}{(J_R+w_{jk})} \end{bmatrix} \quad (6.6)$$

I disturbi di modello sono tutti raccolti nella matrice:

$$Q_k = \begin{bmatrix} \sigma_{f_R}^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\tau_R}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{m_R}^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{J_R}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_l^2 \end{bmatrix} \quad (6.7)$$

6.2.2 Ingressi del sistema

Per studiare nel modo più approfondito possibile l'implementazione del filtro sono stati scelti i seguenti ingressi per eccitare tutte le dinamiche del sistema. Per l'ingresso di **thrust** è stato fornito un valore costante pari al doppio della **forza peso**:

$$f_R = 2m_r g \quad (6.8)$$

mentre per la coppia di rotazione τ_R si è scelto il segnale sinusoidale:

$$\tau_R = 0.3 \sin(4\pi t) \quad (6.9)$$

In aggiunta è presente del disturbo di attuazione $[\sigma_{f_R} \ \sigma_{\tau_R}]^T$ come già visto nel capitolo (2).

6.2.3 Condizioni iniziali

Le condizioni iniziali **nominali** scelte sono: $x_0 = [\pi/2 \ 0 \ 0 \ 0]^T$ più un fattore di incertezza $\delta x = [\sigma_\phi \ \sigma_{\dot{\phi}} \ \sigma_\theta \ \sigma_{\dot{\theta}}]^T$ dovuto al fatto che in realtà non si riesce

perfettamente a far partire il sistema in queste condizioni per via di errori numerici e di misura. Di conseguenza si può definire la matrice di covarianza a priori:

$$P_0 = \begin{bmatrix} \sigma_\phi^2 & 0 & 0 & 0 \\ 0 & \sigma_{\dot{\phi}}^2 & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{\theta}}^2 \end{bmatrix} \quad (6.10)$$

6.2.4 Sensori

Le altre componenti fondamentali per la stima dello stato sono i **sensori**. Le equazioni che ne descrivono il comportamento sono le medesime viste nel capitolo (2):

$$y_k = \begin{bmatrix} y_{DOA} \\ y_{ax} \\ y_{az} \end{bmatrix} = \begin{bmatrix} \text{atan2}\left(\frac{S_{\text{sens}} - l\cos(\phi)}{L_{\text{sens}} - l\sin(\phi)}\right) \\ -g\sin(\theta) \\ g\cos(\theta) \end{bmatrix} \quad (6.11)$$

con la matrice di covarianza:

$$R_k = \begin{bmatrix} \sigma_{DoA}^2 & 0 & 0 \\ 0 & \sigma_{ax}^2 & 0 \\ 0 & 0 & \sigma_{az}^2 \end{bmatrix} \quad (6.12)$$

Essi sono importanti per la scelta del tempo di campionamento T_c , infatti l'algoritmo lavora alla stessa frequenza del sensore più veloce, in questo caso gli accelerometri. Nella fase di correzione la differenza di frequenza è stata gestita utilizzando due funzioni: una che corregge utilizzando **tutti e tre i sensori** se la misura del sensore DOA al passo corrente è già presente e non è stata ancora utilizzata, l'altra che corregge la stima sfruttando **solamente i dati dagli accelerometri**.

6.2.5 Jacobiani

I jacobiani delle matrici definite nella sezione (6.1) sono:

$$F_k = \begin{bmatrix} 1 & T_c & 0 & 0 \\ \frac{-T_c \sin(\phi+\theta)f_R}{m_r l} + \frac{T_c g \sin(\phi)}{l} & 1 - \frac{T_c D_\phi}{m_r l^2} & \frac{-T_c \sin(\phi+\theta)f_R}{m_r l} & 0 \\ 0 & 0 & 1 & T_c \\ 0 & 0 & 0 & 1 - \frac{D_\psi T_c}{J_R} \end{bmatrix} \quad (6.13)$$

$$D_k = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{T_c \cos(\phi+\theta)}{m_r l} & 0 & \frac{-T_c \cos(\phi+\theta)f_R}{l m_r^2} + \frac{T_c D_\phi x_2}{l^2 m_r^2} & 0 & \frac{-T_c \cos(\phi+\theta)f_R}{m_r l^2} + \frac{T_c g \cos(\phi)}{l^2} + \frac{2 T_c D_\phi \dot{\phi}}{m_r l^3} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{T_c}{J_r} & 0 & \frac{-T_c u_2}{J_r^2} + \frac{T_c D_\phi x_4}{J_r^2} & 0 \end{bmatrix} \quad (6.14)$$

$$H_k = \begin{bmatrix} \frac{L_{\text{sens}}l \sin(\phi) + S_{\text{sens}}l \cos(\phi) - l^2}{L_{\text{sens}}^2 + S_{\text{sens}}^2 + l^2 - 2L_{\text{sens}}l \sin(\phi) - 2S_{\text{sens}}l \cos(\phi)} & 0 & 0 \\ 0 & 0 & -g \cos(\theta) \\ 0 & 0 & -g \sin(\theta) \end{bmatrix} \quad (6.15)$$

$$M_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.16)$$

6.2.6 Simulazioni

Dopo aver definito tutti i parametri si può procedere con la simulazione, come durata si è deciso di impostare $T_s = 40s$. Dai grafici in Fig. 6.1 e Fig. 6.2 è possibile vedere come dopo un certo periodo di tempo le curve che rappresentano le posizioni e le velocità degli angoli stimati iniziano a sovrapporsi alle curve delle posizioni e velocità reali. Questo è un andamento che ci si aspettava di ottenere perché all'inizio della simulazione le informazioni a priori fornite al filtro presentano delle incertezze che solo nei cicli successivi vengono corrette tramite le misure dai sensori.

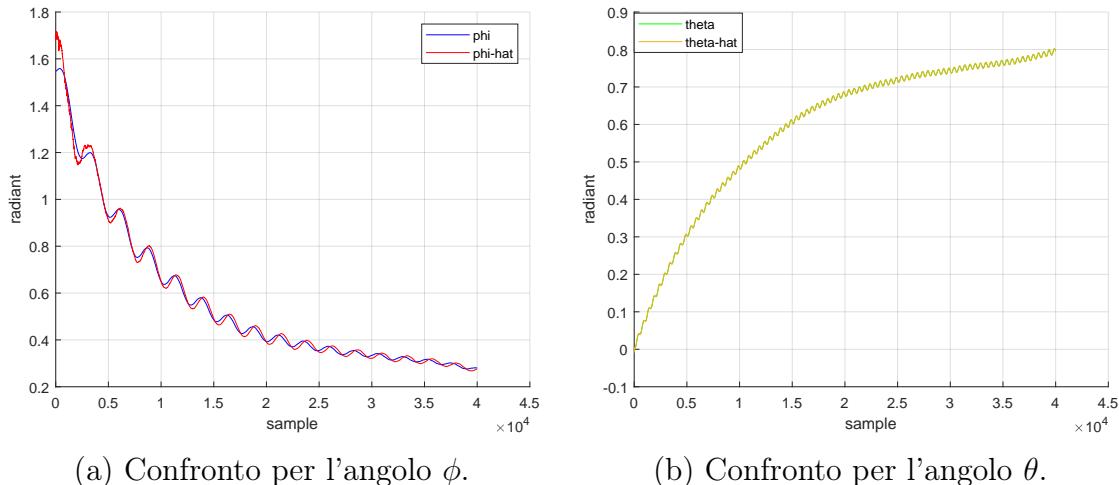


Figura 6.1: Confronto degli angoli ϕ e θ reali e stimati.

Analizzando anche i grafici in Fig. 6.3 si può notare che gli errori (sia di posizioni che di velocità) all'inizio sono alti, dopo pochi istanti di tempo convergono a valori prossimi allo zero. Le curve di ϕ e $\dot{\phi}$ presentano dei picchi maggiori rispetto a quelle di θ e $\dot{\theta}$ perché la correzione su questo angolo avviene meno frequentemente causando un accumulo di errore, ciò comporta che al momento della correzione si ha un guadagno maggiore in corrispondenza di questo sensore.

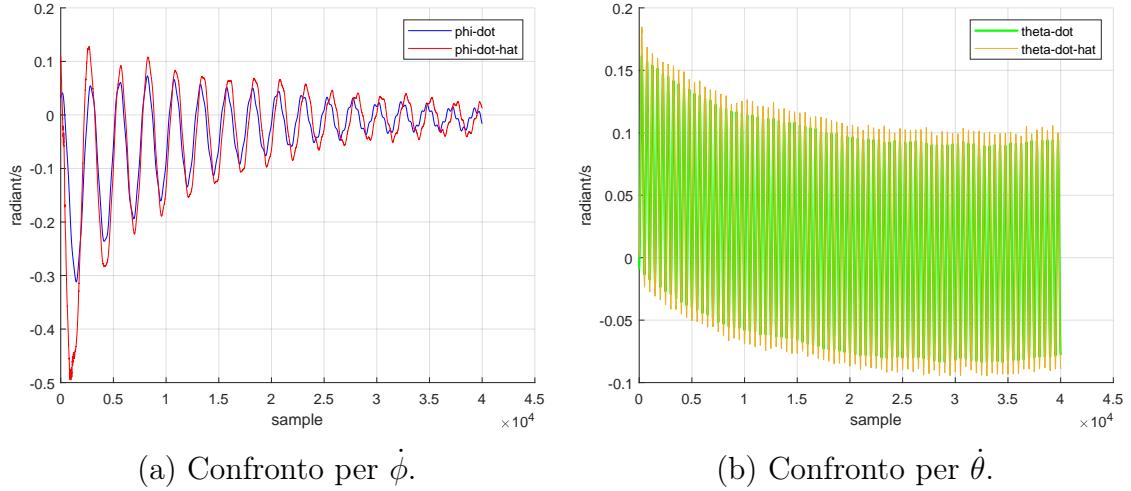


Figura 6.2: Confronto velocità $\dot{\phi}$ e $\dot{\theta}$ reali e stimate.

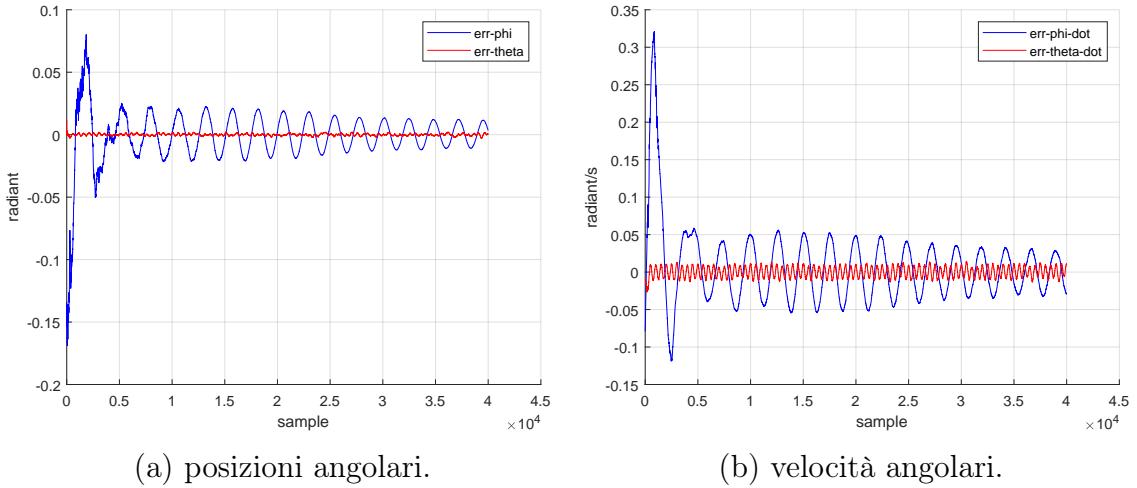


Figura 6.3: Errore di stima delle posizioni e delle velocità degli angoli.

Oltre a valutare l'andamento degli errori è utile analizzare l'innovazione, infatti essa è un ottimo strumento diagnostico, è possibile valutare la bontà del filtro analizzando le sue caratteristiche statistiche:

- la media
- l'autocorrelazione
- la consistenza

6.2.7 Test di bianchezza

La **media** dell'innovazione deve essere nulla, altrimenti significa che il filtro sbaglia sistematicamente le letture, c'è un **bias**. In questo caso i sensori sono stati modellati nel modo corretto in quanto la media dell'innovazione è nulla per tutti e tre i sensori:

$$\begin{aligned}\hat{e}_{DOA} &= 0 \\ \hat{e}_{acc_x} &= 0 \\ \hat{e}_{acc_z} &= 0\end{aligned}$$

Guardando invece il grafico dell'**autocorrelazione** si ha che solo il campione a $lag = 0$ è pari a 1 mentre tutti gli altri valori sono all'interno dell'intervallo di confidenza del 95%. Questo risultato è coerente con le aspettative perché dimostra che l'innovazione non è correlata con se stessa nel tempo, non c'è un percorso che si ripete dovuto ad errori sistematici.

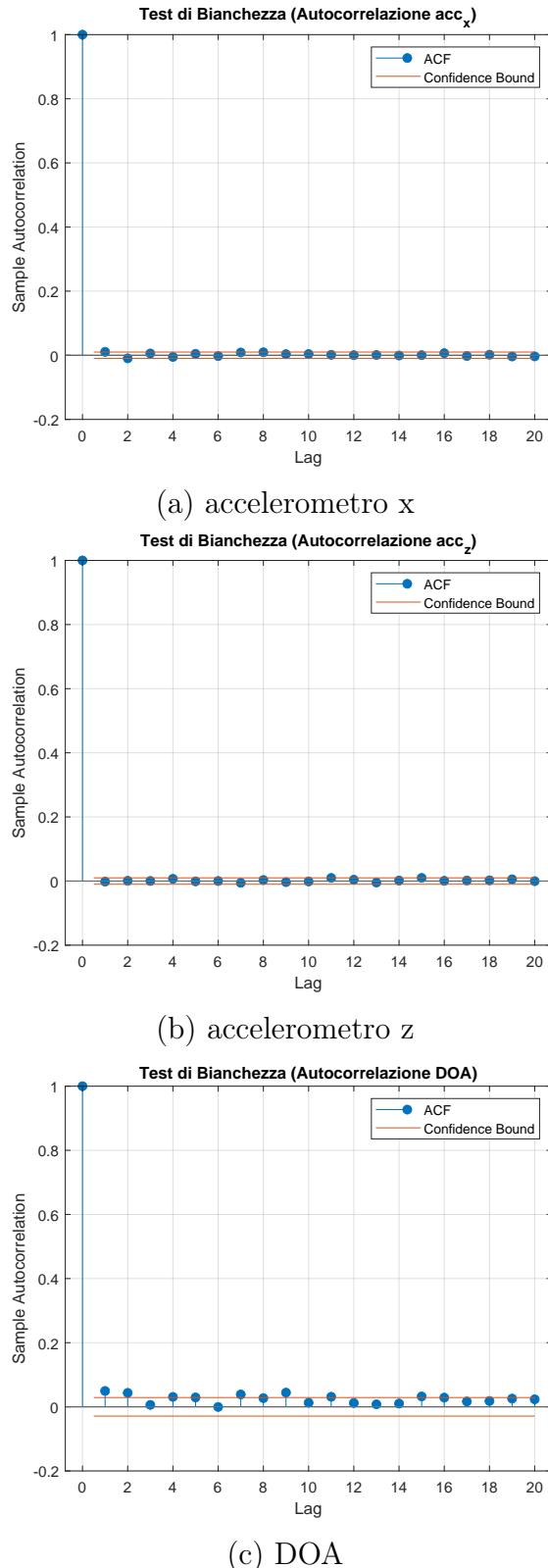


Figura 6.4: Autocorrelazione dell’innovazione.

L’ultimo test è quello di consistenza, il **Normalized Innovation Squared(NIS)**, il quale verifica che la covarianza dell’innovazione S_k atteso è concorde con l’innova-

vazione che effettivamente si ha, in formule:

$$\epsilon_{NIS}(k) = e_k^T S_k^{-1} e_k \quad (6.17)$$

Sempre per il discorso che si hanno due sensori che lavorano a frequenze diverse si sono eseguiti 2 test: uno in cui venivano usati tutti e tre i sensori ($m=3$) e l'altro dove venivano usati solo gli accelerometri ($m=2$). Nel caso generale di m sensori ci si aspettava che $\epsilon_{NIS} \simeq m$ dato che ogni sensore dovrebbe avere un rapporto in media tra l'innovazione e la matrice di covarianza pari a 1. In questo esperimento i valori trovati sono: $\epsilon_{NIS} = 2.83$ per $m=3$ e $\epsilon_{NIS} = 1.90$ per $m=2$, che a meno di qualche errore numerico è esattamente il valore previsto.

6.3 UKF

L'Unscented Kalman Filter è il secondo metodo per sfruttare il Filtro di Kalman per sistemi non lineari. A differenza dell'EKF in cui si linearizzavano le funzioni, nell'UKF si **approssima la distribuzione di probabilità dello stato**, non viene propagata la semplice media \hat{x} , ma vengono generati $N=2n+1$ (n è la dimensione dello stato) **sigma-points** $x_i \in \mathbb{R}^n$ con $i=-n, \dots, 0, \dots, n$ tali che la loro media campionaria e covarianza campionaria pesate attraverso degli opportuni pesi w_i coincidano con la media e la covarianza di X :

$$\bar{x} = \sum_{i=-N}^N w_i x_i \quad (6.18)$$

$$\Sigma_x = \sum_{i=-N}^N w_i (x_i - \bar{x})(x_i - \bar{x})^T \quad (6.19)$$

Per ricavare le approssimazioni delle caratteristiche statistiche di $Y = g(X)$: \bar{y} , Σ_y , Σ_{xy} viene usata la **Trasformata Unscented(UT)**:

$$[\bar{y}, \Sigma_y, \Sigma_{xy}] = UT[\bar{x}, \Sigma_x, g(\cdot)] \quad (6.20)$$

dove vengono passati i sigma points x_i attraverso $y_i = g(x_i)$ calcolando poi la media campionaria e la covarianza campionaria. Per determinare i singoli pesi w_i vengono definiti i seguenti parametri che descrivono come vengono generati geometricamente i sigma-points attorno alla media:

- $\alpha \in (0, 1]$: dice quanto i punti sono vicini o lontani dalla media, più α tende ad uno 1 e più i punti sono lontani, è opportuno scegliere questo valore in maniera corretta per catturare i comportamenti non lineari del sistema.
- $\beta \geq 0$: è un valore che tiene conto della conoscenza a priori, nel nostro caso dato che siamo in una situazione gaussiana $\beta = 2$.
- $\kappa \geq 0$ è un termine che influenza i momenti di ordine superiore al secondo.

Per rispettare le condizioni su media e covarianza viene scelto il peso centrale in corrispondenza di \bar{x} e tutti gli altri in maniera simmetrica:

$$\lambda = \alpha^2(n + k) - n \quad (6.21)$$

$$w_0 = \frac{\lambda}{n + \lambda} \quad (6.22)$$

$$w_i = \frac{1}{2(n + \lambda)} \quad \forall i \neq 0 \quad (6.23)$$

$$w'_0 = w_0 + 1 - \alpha^2 + \beta \quad (6.24)$$

$$w'_i = w_i \quad \forall i \neq 0 \quad (6.25)$$

$$a_i = \sqrt{n + \lambda} \quad \forall i \quad (6.26)$$

E' stato definito il peso w'_i perché nel calcolo della covarianza il peso centrale viene pesato in maniera differente rispetto al calcolo della media. La matrice di covarianza viene fattorizzata come:

$$\Sigma_x = \Gamma\Gamma^T = \sum_{i=1}^N \gamma\gamma^T \quad (6.27)$$

ed a questo punto è possibile calcolare i sigma-points:

$$x_i = \begin{cases} \bar{x} + a_i\gamma_i & i > 0 \\ \bar{x} & i = 0 \\ \bar{x} - a_i\gamma_i & i < 0 \end{cases} \quad (6.28)$$

per propagarli attraverso la funzione $y_i = g(x_i)$. Infine si calcolano i momenti fino al secondo ordine di y:

$$\bar{y} = \sum_{i=-N}^N \omega_i y_i \quad (6.29)$$

$$\Sigma_y = \sum_{i=-N}^N \omega'_i (y_i - \bar{y})(y_i - \bar{y})^T \quad (6.30)$$

$$\Sigma_{xy} = \sum_{i=-N}^N \omega'_i (x_i - \bar{x})(y_i - \bar{y})^T \quad (6.31)$$

6.3.1 Applicazione UKF

In questo paragrafo, dopo aver visto la teoria viene descritto il modo in cui è stato applicato l'algoritmo nel caso in esame. I modelli, gli ingressi, le condizioni iniziali e le incertezze a priori sono le stesse che sono state usate per l'EKF Cap. 6.2.1 in modo da poter confrontare le due diverse soluzioni.

6.3.2 Predizione

Dopo aver calcolato i sigma-points, questi ultimi vengono passati al modello dinamico per calcolare 2N stime al passo k+1. Il valore dello stato predetto è la media campionaria delle 2N stime pesate. In modo analogo usando la covarianza campionaria si calcola la matrice di covarianza.

$$(\bar{x}_{k+1|k}, P_{k+1|k}, \sim) = UT \left(\begin{bmatrix} x_k \\ \bar{0} \end{bmatrix}, \begin{bmatrix} P_k & 0 \\ 0 & Q \end{bmatrix}, f(.,.) \right) \quad (6.32)$$

6.3.3 Correzione

In questa fase viene utilizzato lo stimatore BLUE per integrare la misura y_k alla stima $\hat{x}_{k|k-1}$ ed ottenere $\hat{x}_{k|k}$, per fare ciò viene effettuata la stima statica passando per la UT. Dato che ci troviamo nel caso di rumore additivo a media nulla non correlato v_k , è possibile semplificare il procedimento applicando la trasformata solo al modello del sensore ed applicare successivamente il disturbo:

$$h(x_k, v_k) = h'(x_k) + v_k \quad (6.33)$$

definendo:

$$z_k = h'(x_k) \Rightarrow y_k = z_k + v_k \quad (6.34)$$

da cui:

$$E[y_k] = E[z_k + v_k] = E[z_k] + E[v_k] = E[z_k] \quad (6.35)$$

$$\Sigma_y = \Sigma_z + \Sigma_v = \Sigma_z + R \quad (6.36)$$

ne consegue che basta studiare i momenti di $z = h'(x)$:

$$(\bar{z}, \Sigma_z, \Sigma_{zx}) = UT(\hat{x}_{k|k-1}, P_{k|k-1}, h'(.)) \quad (6.37)$$

con:

$$\bar{y} = \bar{z} \quad (6.38)$$

$$\Sigma_y = \Sigma_z + R \quad (6.39)$$

$$\Sigma_{yx} = \Sigma_{zx} \quad (6.40)$$

ed effettuare infine la correzione con la stima BLUE:

$$e_k = y_k - \bar{y}_k \quad (6.41)$$

$$L_k = \Sigma_{xy} \Sigma_y^{-1} \quad (6.42)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k e_k \quad (6.43)$$

$$P_{k|k} = P_{k|k-1} - L_k \Sigma_y L_k^T \quad (6.44)$$

6.3.4 Simulazioni

Come per l'EKF vengono riportati i risultati delle simulazioni per valutarne la bontà del filtro.

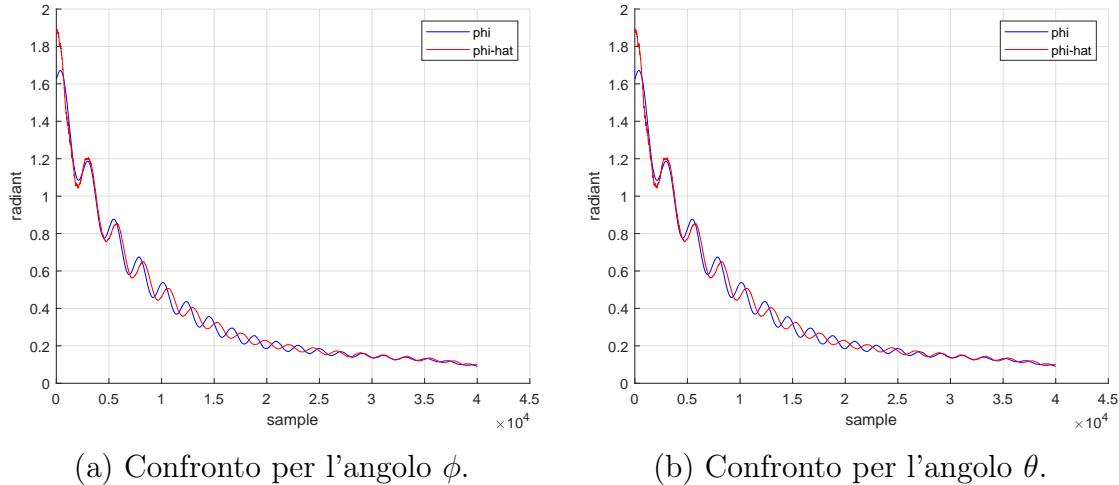


Figura 6.5: Confronto degli angoli ϕ e θ reali e stimati.

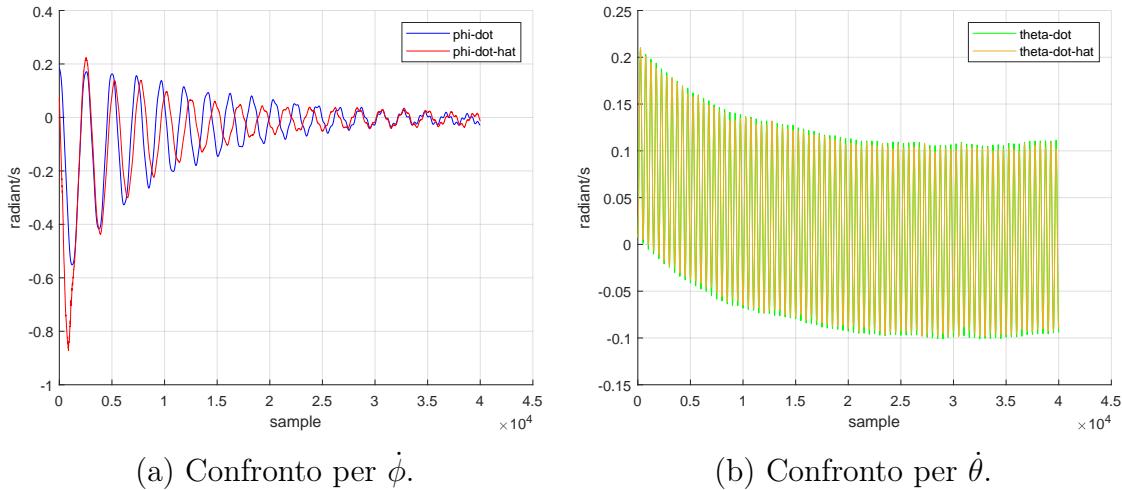


Figura 6.6: Confronto velocità $\dot{\phi}$ e $\dot{\theta}$ reali e stimate.

anche in questo caso il filtro sembra funzionare molto bene, all'inizio la stima non segue perfettamente il valore reale, ma dopo qualche istante di tempo hanno lo stesso andamento. Per completezza è utile osservare anche i grafici dell'errore di stima.

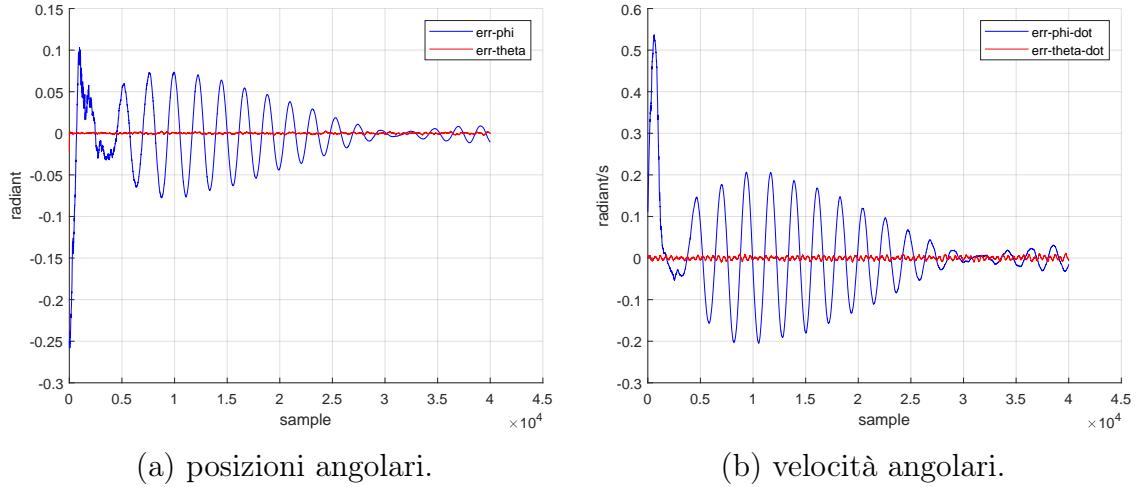


Figura 6.7: Errore di stima delle posizioni e delle velocità degli angoli.

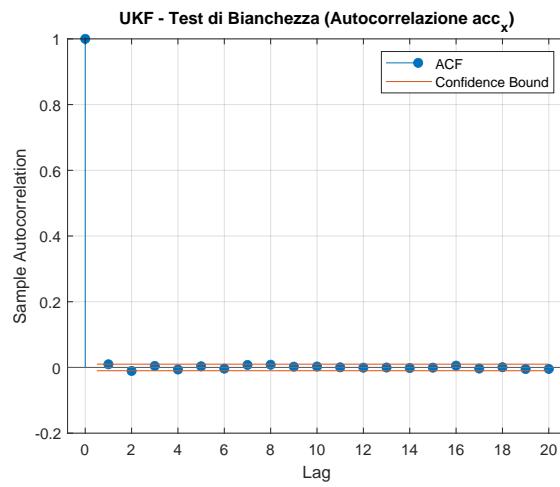
Come ci si poteva aspettare le curve in Fig. 7.3 all'inizio presentano delle oscillazioni più accentuate e successivamente convergono a zero. Esattamente come avveniva per l'EKF, anche in questo caso la curva che descrive l'andamento dell'errore di stima di ϕ e $\dot{\phi}$ ha dei picchi maggiori, questo comportamento è in linea con quanto ci si aspettava per le medesime ragioni discusse nel Cap. 6.2.6.

6.3.5 Test di bianchezza

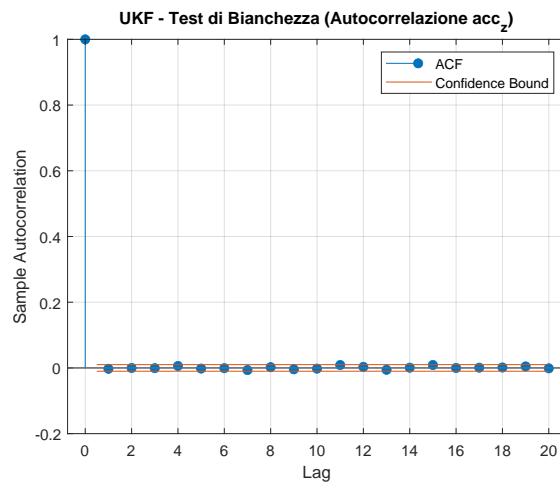
Per assicurarci che il filtro è stato implementato adeguatamente è necessario effettuare il test di bianchezza. Per quanto riguarda la media dell'innovazione anche in questo caso si ha che le medie delle innovazioni sono tutte nulle:

$$\begin{aligned}\hat{e}_{DOA} &= 0 \\ \hat{e}_{acc_x} &= 0 \\ \hat{e}_{acc_z} &= 0\end{aligned}$$

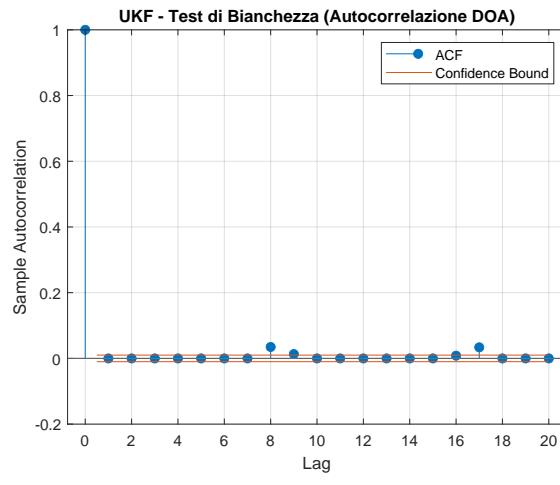
Discutendo anche le autocorrelazioni si ha anche qui una verifica positiva: tutte le misure risultano non correlate tra loro in quanto non fuoriescono dalle bande di confidenza, eccetto un paio di valori per il sensore DOA, questo non vuol dire che il test è fallito perché scegliendo un intervallo di confidenza del 95% ci si può aspettare che il 5% dei valori per pura casualità superino comunque le bande.



(a) accelerometro x



(b) accelerometro z



(c) DOA

Figura 6.8: Autocorrelazione dell'innovazione.

L'ultimo test da effettuare è quello del NIS. Per l'UKF si hanno i seguenti risultati:

$$\epsilon_{NIS} = 2.91 \quad \text{per } m = 3 \quad (6.45)$$

$$\epsilon_{NIS} = 1.99 \quad \text{per } m = 2 \quad (6.46)$$

che sono entrambi corretti, anzi, nel caso di $m=2$ si ha un valore veramente vicino a 2.

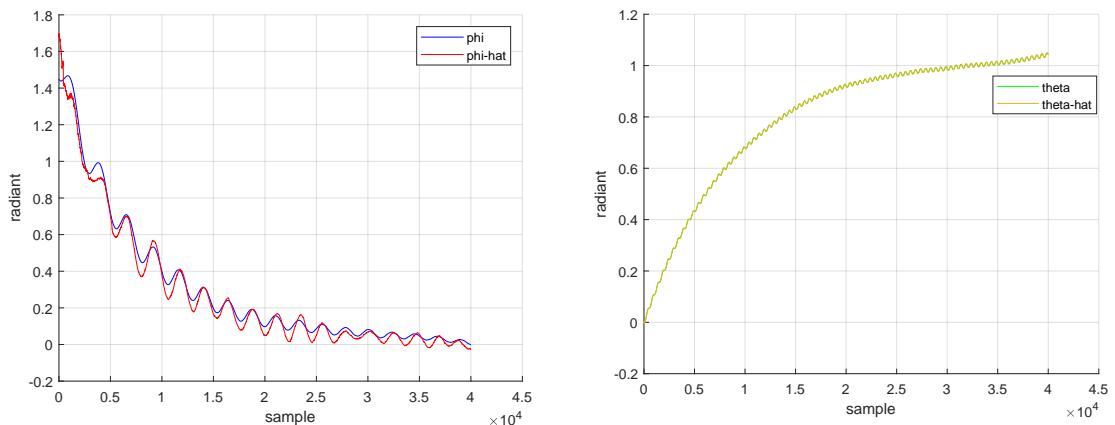
6.4 Distanza di Mahalanobis

Per rendere le simulazioni più "realistiche" è possibile implementare dei metodi per generare dei picchi di rumore nelle letture dei sensori, questi prendono il nome di **outlier**, si tratta di un problema comune di tutti i sensori che deve essere gestito, per questo viene usata la **distanza di Mahalanobis** che è una metrica statistica uguale al NIS che aiuta a decidere se usare la lettura per la correzione oppure rifiutarla. In questo progetto vengono generati outlier da tutti i sensori in modo indipendente circa una volta ogni cento letture. Proviamo ad analizzare due casi:

- presenza di outlier senza usare la distanza di Mahalanobis
- presenza di outlier usando la distanza di Mahalanobis

6.4.1 Senza usare la distanza di Mahalanobis

In questo scenario ci si aspetta che la stima inevitabilmente peggiori perché una volta ogni cento la correzione con il singolo sensore è sbagliata. Analizziamo i risultati:



(a) Confronto per l'angolo ϕ .

(b) Confronto per l'angolo θ .

Figura 6.9: Andamenti delle posizioni angolari vere e stimate senza usare la distanza di Mahalanobis

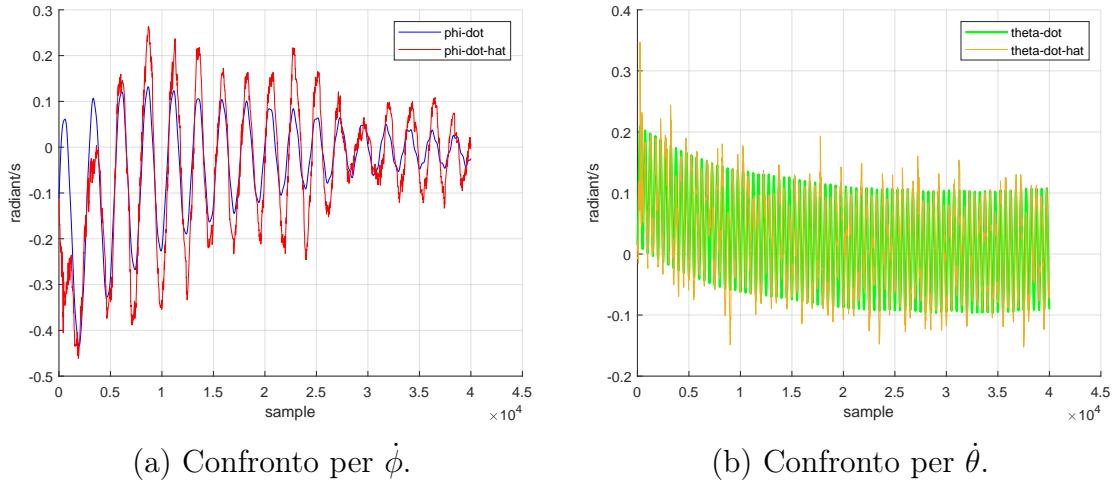


Figura 6.10: Andamenti delle velocità angolari vere e stimate senza usare la distanza di Mahalanobis.

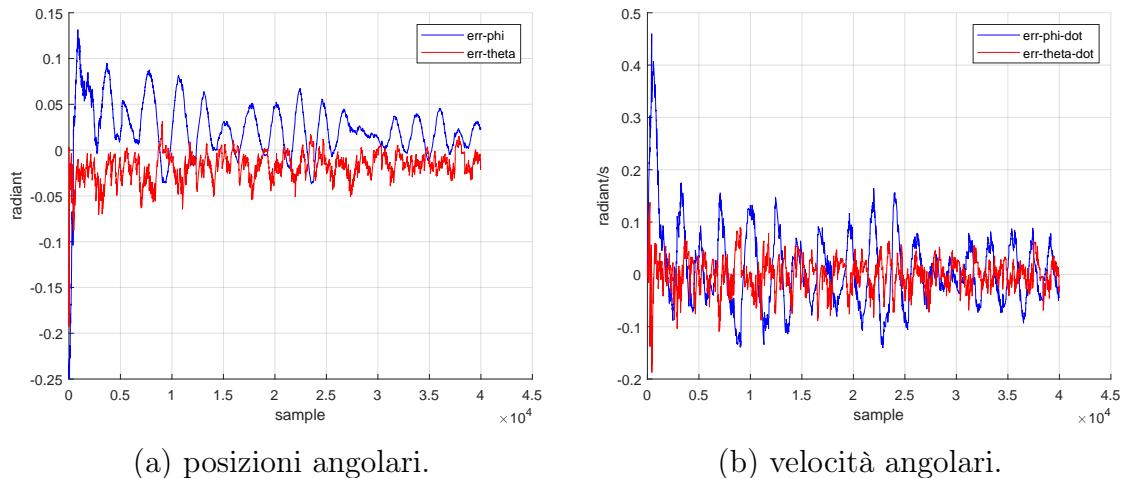


Figura 6.11: Errore di stima delle posizioni e delle velocità degli angoli senza usare la distanza di Mahalanobis.

Nella Fig 6.11a risulta evidente vedere come l'errore di posizione è cresciuto ed inoltre non è a media nulla, questo è un classico sintomo che certifica la presenza di errori in fase di correzione che portano il filtro ad impiegare molto tempo per tornare a convergere sul vero andamento.

6.4.2 Usando la distanza di Mahalanobis

Dai grafici notiamo come la stima sia nettamente migliore rispetto al caso precedente.

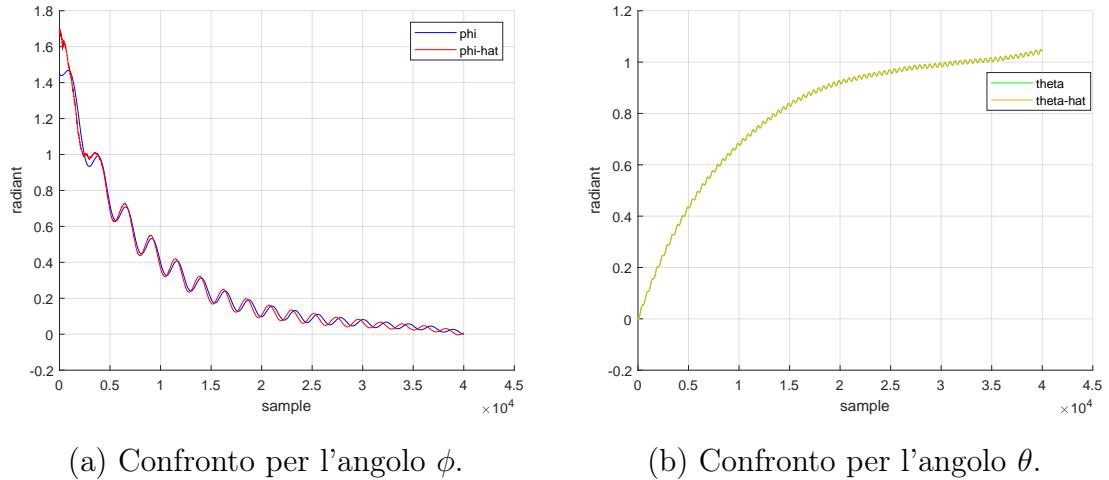


Figura 6.12: Andamenti delle posizioni angolari vere e stimate senza usare la distanza di Mahalanobis

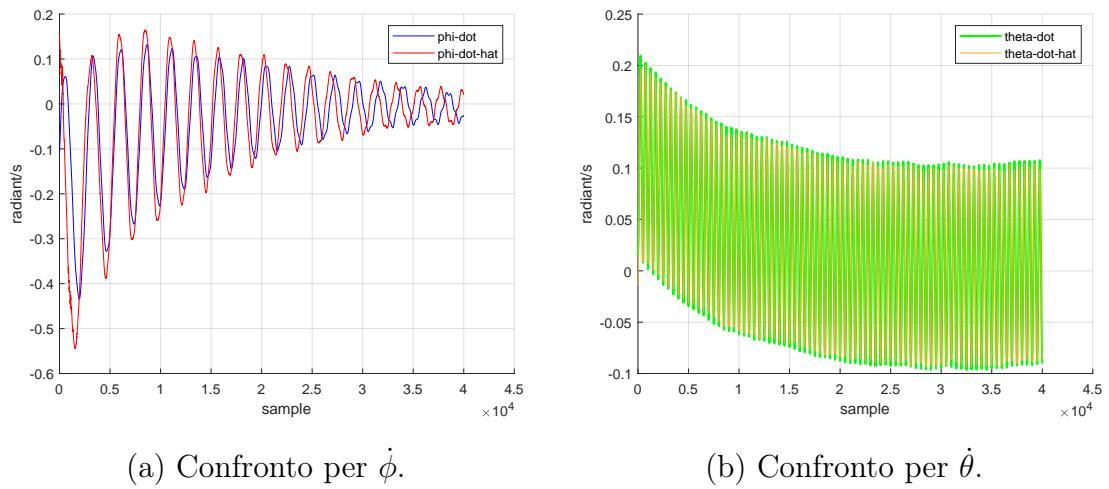


Figura 6.13: Andamenti delle velocità angolari vere e stimate senza usare la distanza di Mahalanobis.

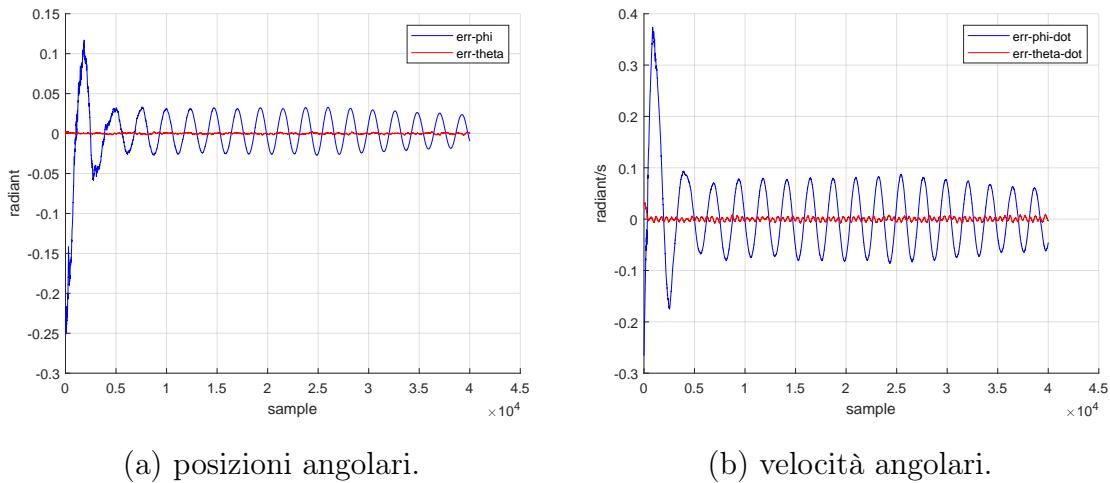


Figura 6.14: Errore di tima delle posizioni e delle velocità degli angoli senza usare la distanza di Mahalanobis.

In questo esperimento gli outlier identificati sono **1197**. Facendo un po' di calcoli si può verificare che la condizione sulle misure riesce a catturare efficacemente i valori indesiderati. Dato che la simulazione dura **40s**, vengono eseguiti 40.000 cicli (lavorando alla frequenza degli accelerometri:1000Hz).

1. Definizione dei campioni totali:

$$N_{\text{tot}} = 40 \text{ s} \times 1000 \text{ Hz} = 40.000$$

$$N_{\text{3sens}} = 40 \text{ s} \times 120 \text{ Hz} = 4.800$$

$$N_{\text{2sens}} = 40.000 - 4.800 = 35.200$$

2. Probabilità di scarto nel caso a 2 sensori (Solo Accelerometri):

$$P_{2s}(\text{outlier real}) = 1 - (0.99 \times 0.99) = 1 - 0.9801 = 0.0199 \quad (1, 99\%)$$

$$P_{2s}(\text{falso pos.}) = 0.9801 \times 0.01 = 0.0098 \quad (0, 98\%)$$

$$P_{2S}(\text{scarto tot}) = 0.0199 + 0.0098 = 0.0297 \quad (2, 97\%)$$

$$Scarti_{2S} = 35.200 \times 0.0297 \approx 1.045$$

3. Probabilità di scarto nel caso a 3 sensori (Con DOA):

$$P_{3s}(\text{outlier reale}) = 1 - (0.99)^3 = 1 - 0.9703 = 0.0297 \quad (2, 97\%)$$

$$P_{3s}(\text{falso pos.}) = 0.9703 \times 0.01 = 0.0097 \quad (0, 97\%)$$

$$P_{3s}(\text{scarto tot}) = 0.0297 + 0.0097 = 0.0394 \quad (3, 94\%)$$

$$\text{Scarti}_{3s} = 4.800 \times 0.0394 \approx 189$$

4. Totale complessivo atteso:

$$\begin{aligned} \text{Scarti}_{\text{tot}} &= \text{Scarti}_{2s} + \text{Scarti}_{3s} \\ &= 1.045 + 189 \\ &= \mathbf{1.234} \end{aligned}$$

Il numero di outlier identificati è coerente con le aspettative.

6.5 Conclusioni

L'EKF e l'UKF sono entrambe delle soluzioni valide per la stima dello stato di sistemi non lineari, il secondo approccio però è chiaramente migliore in applicazioni dove il sistema presenta delle forti non linearità, l'EKF in questo caso potrebbe anche divergere dato che la linearizzazione introdurrebbe troppi errori nel calcolo della stima. Il vantaggio che questo filtro ha rispetto all'UKF è che in contesti in cui le non linearità possono essere gestite correttamente ed i jacobiani sono pressocché costanti, dal punto di vista computazionale l'algoritmo è più efficiente in quanto non ha bisogno di propagare i $2n+1$ sigma-points ad ogni iterazione.

7 Regolarizzazione

Fino ad ora è stato trattato il problema della predizione ($k > t$) e del filtraggio ($k = t$), però è utile anche trattare il problema della regolarizzazione: $k > t$. Uno dei più importanti algoritmi di regolarizzazione è quello ideato da **Rauch-Tung-Striebel(RTS)**, tale algoritmo rappresenta lo stato dell'arte per la stima **off-line**, in quanto sfrutta l'**intero set di misurazioni disponibili (passate e future)** per raffinare la stima dello stato e ridurre la covarianza dell'errore rispetto al solo filtraggio in avanti. Nel filtraggio in avanti è possibile usare uno dei Filtri di Kalman discussi nel Cap 6, in questo caso è stato usato l'EKF implementato nella Sez. 6.2.1. La regolarizzazione consiste nel prendere l'ultimo valore predetto dal filtro $\hat{x}_{k|N}$ e l'ultima matrice di covarianza $P_{k|N}$ con $k=1,\dots,N$ e calcolare i valori all'istante precedente, da $k=N-1$ a $k=1$:

$$C_k = P_{k|k} F_{k+1|k}^T P_{k+1|k}^{-1} \quad (7.1)$$

$$\hat{x}_{k|N} = \hat{x}_{k|k} + C_k (\hat{x}_{k+1|N} - \hat{x}_{k+1|k}) \quad (7.2)$$

$$P_{k|N} = P_{k|k} + C_k (P_{k+1|N} - P_{k+1|k}) \quad (7.3)$$

7.0.1 Analisi dei risultati

E' possibile notare nei grafici sottostanti, in particolare nel grafico in Fig. 7.3a come gli andamenti del regolarizzatore siano più smussati rispetto a quello dell'EKF. Per

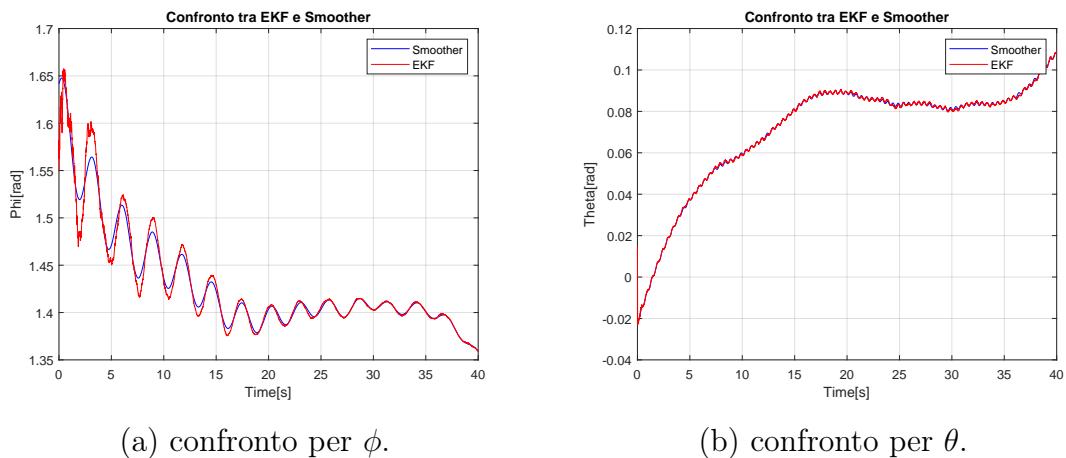


Figura 7.1: Confronto tra stima regolarizzata e stima EKF delle posizioni angolari.

il grafico in Fig. 7.3b è riportato qui di seguito uno zoom delle curve dato che osservando l'intera simulazione non si riesce ad apprezzare graficamente la differenza tra i due approcci. Vedendo il grafico ingrandito Fig 7.2 è immediato notare come

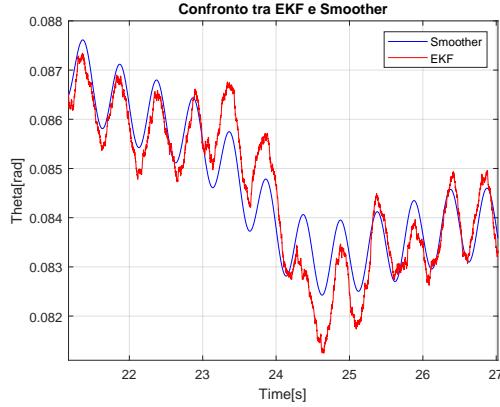


Figura 7.2: Zoom del grafico di confronto tra l'EKF ed il regolarizzatore della stima di θ

anche per l'angolo θ la curva RTS è più regolare. Si ottengono analoghe osservazioni se vengono analizzati i grafici delle velocità angolari:

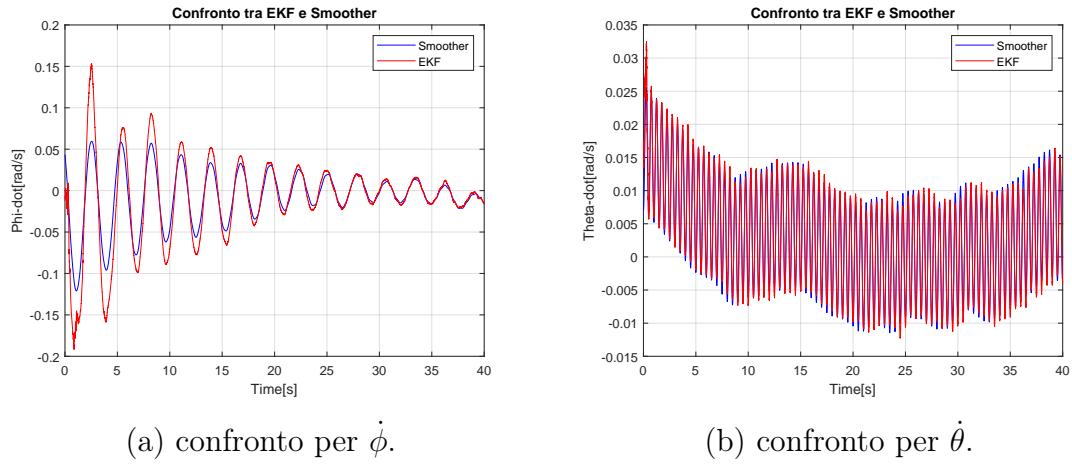


Figura 7.3: Confronto tra stima regolarizzata e stima EKF delle velocità angolari.