

Corso di Laurea in Ingegneria e Scienze Informatiche

Studio e applicazione del filtro di Kalman e sue varianti per il tracciamento di corpi celesti

Tesi di laurea in:
METODI NUMERICI

Relatore
Prof.ssa Damiana Lazzaro

Candidato
Marco Buda

Sessione Unica
Anno Accademico 2024-2025

Sommario

Il filtro di Kalman è un algoritmo estremamente importante per l'elaborazione di dati. In questa tesi se ne studiano le origini matematiche e se ne sperimenta concretamente l'utilizzo, implementando un algoritmo innovativo di tracciamento di corpi celesti nel sistema solare a partire da osservazioni ottiche sia simulate che reali. Si ottengono risultati decisamente più accurati rispetto alle sole osservazioni.

Indice

Sommario	iii
Introduzione	1
1 Fondamenti teorici	3
1.1 I sistemi dinamici	3
1.2 Il filtro di Kalman	5
1.3 Costruzione del filtro lineare	6
1.4 Varianti non lineari	13
1.4.1 Extended Kalman Filter	13
1.4.2 Unscented Kalman Filter	14
1.4.3 Metodo Monte Carlo	17
1.5 Sviluppi recenti	18
1.6 Limitazioni	19
2 Modello astronomico	21
2.1 Coordinate celesti	21
2.2 Fotometria e il sistema H-G	23
2.3 Orbite ellittiche	23
3 Adattamento del filtro di Kalman al caso di studio: formulazione e risultati	27
3.1 Approccio diretto	27
3.2 Sfide sul tracciamento di angoli	29
3.3 Scelta dei vettori di stato e di misurazione	30
3.4 Equazioni del sistema	32
3.5 Matrici di covarianza	34
3.6 Simulazione	36
3.7 Applicazione a dati reali	40
Conclusioni	49

	51
Bibliografia	51

Introduzione

Ogni volta che si effettua una misurazione della realtà, il valore ottenuto è soggetto ad imprecisione. Questo vale per i sensori più semplici, ma anche per gli strumenti scientifici più sofisticati. Fin da quando questo limite è stato riconosciuto, è emersa l'esigenza di ricavare informazioni più accurate a partire dai dati osservati, attraverso un processo di filtraggio.

Non è difficile immaginare che una conoscenza più approfondita delle leggi che governano l'evoluzione di un sistema consente di ottenere stime via via più accurate man mano che si raccolgono nuove misurazioni. Questa intuizione è stata formalizzata da Rudolf Emil Kálmán e Richard Snowden Bucy nell'algoritmo oggi conosciuto come filtro di Kalman. Nonostante siano passati più di 60 anni dalla sua introduzione, il filtraggio dei dati resta un processo cruciale nell'ingegneria applicata e la soluzione di Kálmán e Bucy resta alla base di molte tecniche moderne.

Il modello matematico assume rilevanza informatica nel momento in cui viene implementato nel codice e deve garantire stabilità numerica durante l'esecuzione. L'obiettivo di questa tesi è, dunque, individuare un caso di applicazione reale che permetta di sperimentare concretamente l'utilizzo del filtro, con ricadute significative per la ricerca, in particolare nel settore astronomico.

A tal fine, si è scelto di tracciare la posizione di corpi celesti nel sistema solare, con particolare attenzione a NEA (near-Earth asteroids) e PHA (potentially hazardous asteroids), a partire da osservazioni ottiche effettuate da osservatori in tutto il mondo e raccolte dal Minor Planet Center. Questo studio richiede una buona conoscenza dei modelli del moto orbitale dei corpi e dei fenomeni ottici che ne influenzano la luminosità degli asteroidi, ma consente di ottenere stime accurate della posizione tridimensionale, che possono rivelarsi utili per analisi successive

riguardanti la pericolosità potenziale per la Terra.

Il modello si scoprirà essere altamente non lineare, rendendo necessario l'utilizzo della variante Unscented Kalman Filter (UKF), che rappresenta un'estensione più complessa rispetto alla formulazione originale. Inoltre, la complessità del modello richiederà la soluzione numerica di equazioni, aprendo così la possibilità di applicare metodi avanzati per il calcolo degli zeri di funzioni, oltre alla sperimentazione e all'eventuale sviluppo di nuove tecniche numeriche.

Struttura della tesi

- Capitolo 1:
 - Definizione formale di sistema dinamico e del problema risolto dal filtro di Kalman.
 - Costruzione algebrica, raccogliendo in modo più chiaro possibile i passaggi che portano alle formule finali.
 - Esplorazione delle varianti non lineari, con i relativi pregi e difetti, includendo alcune delle innovazioni più recenti.
- Capitolo 2:
 - Presentazione del modello astronomico, ossia l'insieme delle formule di astrometria e astrodinamica che governano i fenomeni del caso di applicazione studiato, da cui si ricaveranno le equazioni del sistema nella forma richiesta dal filtro, insieme alle matrici e ai parametri necessari.
- Capitolo 3:
 - Discussione dell'implementazione di una simulazione del sistema e dei risultati ottenuti da essa, con successivo passaggio ai dati forniti dalle osservazioni reali e confronto.
 - Infine, valutazione del vantaggio in termini di precisione portato dal processo di filtraggio, rispetto a un approccio diretto per il calcolo delle posizioni.

Capitolo 1

Fondamenti teorici

In questo capitolo si definiscono i concetti alla base degli argomenti di questa tesi, insieme ai modelli matematici che li descrivono. Si presenta, inoltre, una costruzione algebrica del filtro di Kalman e si offre una panoramica delle sue varianti.

1.1 I sistemi dinamici

Un sistema dinamico è un qualunque sistema che evolve nel tempo secondo una determinata legge.

A seconda dell'obiettivo, si individuano le proprietà del sistema che interessa esaminare (es.: posizione e velocità di un oggetto), dette variabili di stato, e le si raccoglie in un vettore, generalmente $x \in \mathbb{R}^n$, andando a definire una cosiddetta rappresentazione in spazio di stato.

Nella forma generale, si considera il tempo come continuo ($t \in \mathbb{R}$), per cui si parla di modelli continui, espressi con equazioni differenziali:

$$\frac{d}{dt}(x(t)) = f(t, x(t))$$

Tuttavia, si può decidere di considerare lo stato soltanto in determinati istanti t_k . Ne risultano modelli discreti, su cui si concentrerà questa tesi, i quali possono essere espressi con relazioni di ricorrenza:

$$x_k = f(k, x_{k-1})$$

In genere si sceglie di esplicitare la presenza di ingressi $u \in \mathbb{R}^l$ nel sistema (*control input*, es.: forza di gravità, propulsione...), assunti deterministici, e di misurazioni $z \in \mathbb{R}^m$ (*measurement*), effettuate sulle variabili osservate.

Dunque il modello viene espresso con le equazioni:

$$\begin{aligned} x_k &= f(k, x_{k-1}, u_{k-1}) \\ z_k &= h(k, x_k) \end{aligned}$$

La prima è detta “equazione di stato” e la seconda “trasformazione di uscita”.

Modellizzare la realtà comporta spesso che non si conosca la vera legge di evoluzione f_{reale} , ma solo una sua approssimazione, quindi soggetta a disturbi o imprecisioni (*process noise*, $w \in \mathbb{R}^n$), da cui l'importanza di effettuare misurazioni periodiche per monitorare la reale evoluzione dello stato.

Poiché le misurazioni stesse sono soggette a disturbi (*measurement noise*, $v \in \mathbb{R}^m$), il modello completo diventa:

$$x_k = f(k, x_{k-1}, u_{k-1}) + w_{k-1} \quad (1.1)$$

$$z_k = h(k, x_k) + v_k \quad (1.2)$$

Un caso particolare riguarda i modelli lineari, cioè esprimibili nella seguente forma matriciale:

$$x_k = A_k x_{k-1} + B_k u_{k-1} + w_{k-1} \quad (1.3)$$

$$z_k = H_k x_k + v_k \quad (1.4)$$

In questo caso, $A_k \in \mathbb{R}^{n \times n}$ è detta “matrice dinamica” e determina l'evoluzione dello stato dal passo $k - 1$ al passo k in assenza di ingressi e disturbi, $B_k \in \mathbb{R}^{l \times n}$

è detta “matrice di ingresso” e determina il contributo degli ingressi che agiscono dal passo $k - 1$ al passo k e $H_k \in \mathbb{R}^{m \times n}$ è detta “matrice di uscita” e determina le variabili osservate in base alle variabili da stimare.

In generale, i modelli sono tempo-varianti (o non stazionari), da cui k al pedice di A_k , B_k e H_k , ma si evidenziano i modelli LTI, ossia lineari tempo-invarianti (o lineari stazionari) in cui le matrici A , B e H sono costanti nel tempo.

Dalla combinazione delle imprecisioni nel modello e nelle misurazioni, nasce la necessità di algoritmi che raccolgano ed interpretino i dati osservati per determinare una stima (*state estimate*, \hat{x}) dello stato reale (*ground truth*, x). Questi algoritmi sono detti, appunto, stimatori.

1.2 Il filtro di Kalman

Il filtro di Kalman [1] [2] [3] è uno stimatore lineare ricorsivo che minimizza l’errore quadratico medio.

L’aspetto ricorsivo implica, innanzitutto, che il filtro opera in tempo discreto e che ogni stima \hat{x}_k è determinata in base alla stima precedente \hat{x}_{k-1} e alla misurazione corrente z_k , senza richiedere l’utilizzo esplicito di $\hat{x}_0, \hat{x}_1, \dots, \hat{x}_{k-2}$ o z_1, \dots, z_{k-1} . Questo costituisce uno dei principali vantaggi del filtro, in quanto riduce sia la complessità temporale che quella spaziale, senza compromettere l’ottimalità. Come anticipato, l’ottimalità è definita dal fatto che, ad ogni passo k , l’algoritmo produce la stima \hat{x}_k che minimizza la quantità $\mathbb{E}[\|e_k\|^2]$, con $e_k = \hat{x}_k - x_k$.

Il filtro è applicabile a un qualsiasi modello lineare a tempo discreto, anche tempo-variante, ossia descritto dalle equazioni 1.3 e 1.4.

Sono posti, però, vincoli di non correlazione sulle variabili aleatorie:

$$\begin{aligned}
\text{Cov}(w_k, v_j) &= 0_{n \times m}, & \forall k, \forall j \\
\text{Cov}(w_k, u_j) &= 0_{n \times l}, & \forall k, \forall 0 \leq j \leq k-1 \\
\text{Cov}(v_k, u_j) &= 0_{m \times l}, & \forall k, \forall 0 \leq j \leq k-2 \\
\text{Cov}(w_k, w_j) &= 0_{n \times n}, & \forall k, \forall j \neq k \\
\text{Cov}(v_k, v_j) &= 0_{m \times m}, & \forall k, \forall j \neq k \\
\text{Cov}(w_k, x_0) &= 0_{n \times n}, & \forall k \\
\text{Cov}(v_k, x_0) &= 0_{m \times n}, & \forall k
\end{aligned}$$

Infine, nelle formulazioni standard del filtro, si richiede che w e v siano privi di *bias*, ossia che $\mathbb{E}[w_k] = \underline{0}$ e $\mathbb{E}[v_k] = \underline{0}$, ad ogni passo k .

1.3 Costruzione del filtro lineare

In letteratura sono riportate diverse derivazioni e dimostrazioni di ottimalità del filtro di Kalman [4, pp. 107-113] [5]. Si presenta qui una costruzione algebrica il più elementare possibile.

Il primo passo è descrivere l'espressione che calcoli la stima \hat{x}_k . Al passo $k \geq 1$ sono disponibili le informazioni riguardo \hat{x}_{k-1} , u_{k-1} e z_k , per cui l'espressione lineare avrà la forma generica:

$$\hat{x}_k = \mathcal{A}_k \hat{x}_{k-1} + \mathcal{B}_k u_{k-1} + \mathcal{K}_k z_k \quad (1.5)$$

Per quanto riguarda il passo $k = 0$, la scelta della stima iniziale \hat{x}_0 è libera, purché sia deterministica. Se si hanno informazioni sulla distribuzione di x_0 , è consigliabile scegliere $\hat{x}_0 = \mathbb{E}[x_0]$.

Una condizione implicita sul filtro richiede che anche l'errore sulle stime gene-

rate e_k sia privo di *bias*, ossia, ad ogni passo k :

$$\mathbb{E}[e_k] = \mathbb{E}[\hat{x}_k - x_k] = \underline{0}$$

Da questa condizione si otterranno informazioni sulle matrici incognite \mathcal{A} , \mathcal{B} e \mathcal{K} . In effetti, considerando per $k \geq 1$ e sostituendo \hat{x}_k con la sua definizione (eq. 1.5), si ottiene:

$$\mathbb{E}[\mathcal{A}_k \hat{x}_{k-1} + \mathcal{B}_k u_{k-1} + \mathcal{K}_k z_k - x_k] = \underline{0}$$

Successivamente, sostituendo z_k con la sua definizione (eq. 1.4):

$$\mathbb{E}[\mathcal{A}_k \hat{x}_{k-1} + \mathcal{B}_k u_{k-1} + \mathcal{K}_k (H_k x_k + v_k) - x_k] = \underline{0}$$

Sostituendo x_k con la relazione di ricorrenza (eq. 1.3) e manipolando i termini:

$$\begin{aligned} & \mathbb{E}[\mathcal{A}_k \hat{x}_{k-1} + \mathcal{B}_k u_{k-1} + \mathcal{K}_k (H_k (A_k x_{k-1} + B_k u_{k-1} + w_{k-1}) + v_k) + \\ & \quad - (A_k x_{k-1} + B_k u_{k-1} + w_{k-1})] = \underline{0} \\ \Rightarrow & \mathbb{E}[\mathcal{A}_k \hat{x}_{k-1} + \mathcal{B}_k u_{k-1} + \mathcal{K}_k H_k A_k x_{k-1} + \mathcal{K}_k H_k B_k u_{k-1} + \mathcal{K}_k H_k w_{k-1} + \\ & \quad + \mathcal{K}_k v_k - A_k x_{k-1} - B_k u_{k-1} - w_{k-1}] = \underline{0} \\ \Rightarrow & \mathbb{E}[\mathcal{A}_k \hat{x}_{k-1} + \mathcal{B}_k u_{k-1} + \mathcal{K}_k H_k A_k x_{k-1} + \mathcal{K}_k H_k B_k u_{k-1} + \mathcal{K}_k H_k w_{k-1} + \\ & \quad + \mathcal{K}_k v_k - A_k x_{k-1} - B_k u_{k-1} - w_{k-1} - \mathcal{A}_k x_{k-1} + \mathcal{A}_k x_{k-1}] = \underline{0} \\ \Rightarrow & \mathbb{E}[\mathcal{A}_k (\hat{x}_{k-1} - x_{k-1}) + (\mathcal{K}_k H_k A_k - A_k + \mathcal{A}_k) x_{k-1} + \\ & \quad + (\mathcal{K}_k H_k B_k - B_k + \mathcal{B}_k) u_{k-1} + (\mathcal{K}_k H_k - I) w_{k-1} + \mathcal{K}_k v_k] = \underline{0} \\ \Rightarrow & \mathcal{A}_k \mathbb{E}[\hat{x}_{k-1} - x_{k-1}] + (\mathcal{K}_k H_k A_k - A_k + \mathcal{A}_k) \mathbb{E}[x_{k-1}] + \\ & \quad + (\mathcal{K}_k H_k B_k - B_k + \mathcal{B}_k) \mathbb{E}[u_{k-1}] + (\mathcal{K}_k H_k - I) \mathbb{E}[w_{k-1}] + \mathcal{K}_k \mathbb{E}[v_k] = \underline{0} \end{aligned}$$

Sfruttando l'ipotesi che le quantità $\hat{x}_{k-1} - x_{k-1}$, w_{k-1} e v_k siano prive di *bias*:

$$(\mathcal{K}_k H_k A_k - A_k + \mathcal{A}_k) \mathbb{E}[x_{k-1}] + (\mathcal{K}_k H_k B_k - B_k + \mathcal{B}_k) \mathbb{E}[u_{k-1}] = \underline{0}$$

Non potendo fare assunzioni su x e u , ne segue che:

$$\mathcal{K}_k H_k A_k - A_k + \mathcal{A}_k = 0_{n \times n} \quad , \quad \mathcal{K}_k H_k B_k - B_k + \mathcal{B}_k = 0_{l \times n}$$

$$\Rightarrow \mathcal{A}_k = (I - \mathcal{K}_k H_k) A_k, \quad \mathcal{B}_k = (I - \mathcal{K}_k H_k) B_k$$

Risulta pratico definire la seguente quantità, per $k \geq 1$, come una stima *a priori*, ossia che non tenga conto della misurazione z_k :

$$\hat{x}_k^- = A_k \hat{x}_{k-1} + B_k u_{k-1}$$

Così facendo, si vanno a distinguere due fasi all'interno di ogni passo: una fase di predizione, o *time update*, in cui si calcola \hat{x}_k^- in base alle conoscenze sul modello, e una fase di correzione, o *measurement update*, in cui si combina \hat{x}_k^- con la misurazione z_k per ottenere una stima ottimale \hat{x}_k .



L'espressione per ricavare \hat{x}_k , ossia la stima *a posteriori*, diventa ora:

$$\begin{aligned} \hat{x}_k &= (I - \mathcal{K}_k H_k) A_k \hat{x}_{k-1} + (I - \mathcal{K}_k H_k) B_k u_{k-1} + \mathcal{K}_k z_k = \\ &= (I - \mathcal{K}_k H_k) (A_k \hat{x}_{k-1} + B_k u_{k-1}) + \mathcal{K}_k z_k = (I - \mathcal{K}_k H_k) \hat{x}_k^- + \mathcal{K}_k z_k = \\ &= \hat{x}_k^- + \mathcal{K}_k (z_k - H_k \hat{x}_k^-) \end{aligned}$$

Proseguendo, si definisca $e_k^- = \hat{x}_k^- - x_k$ come l'errore sulla stima *a priori* e si consideri:

$$\begin{aligned} e_k^- &= \hat{x}_k^- - x_k = (A_k \hat{x}_{k-1} + B_k u_{k-1}) - (A_k x_{k-1} + B_k u_{k-1} + w_{k-1}) = \\ &= A_k (\hat{x}_{k-1} - x_{k-1}) - w_{k-1} = A_k e_{k-1} - w_{k-1} \end{aligned}$$

Si osservi che anche e_k^- è privo di *bias*:

$$\mathbb{E}[e_k^-] = \mathbb{E}[A_k e_{k-1} - w_{k-1}] = A_k \mathbb{E}[e_{k-1}] - E[w_{k-1}] = \underline{0}$$

Si definiscano, per $k \geq 1$, le auto-covarianze degli errori $P_k^- = \text{Cov}(e_k^-, e_k^-)$ e $P_k = \text{Cov}(e_k, e_k)$. Si ha:

$$\begin{aligned}
 P_k^- &= \mathbb{E} [(e_k^- - \mathbb{E}[e_k^-])(e_k^- - \mathbb{E}[e_k^-])^T] = \mathbb{E} [e_k^- (e_k^-)^T] = \\
 &= \mathbb{E} [(A_k e_{k-1} - w_{k-1})(A_k e_{k-1} - w_{k-1})^T] = \\
 &= \mathbb{E} [(A_k e_{k-1} - w_{k-1}) (e_{k-1}^T A_k^T - w_{k-1}^T)] = \\
 &= A_k \mathbb{E} [e_{k-1} (e_{k-1})^T] A_k^T - A_k \mathbb{E} [e_{k-1} (w_{k-1})^T] + \\
 &\quad - \mathbb{E} [w_{k-1} (e_{k-1})^T] A_k^T + \mathbb{E} [w_{k-1} (w_{k-1})^T] = \\
 &= A_k \mathbb{E} [(e_{k-1} - \underline{0})(e_{k-1} - \underline{0})^T] A_k^T - A_k \mathbb{E} [(e_{k-1} - \underline{0})(w_{k-1} - \underline{0})^T] + \\
 &\quad - \mathbb{E} [(w_{k-1} - \underline{0})(e_{k-1} - \underline{0})^T] A_k^T + \mathbb{E} [(w_{k-1} - \underline{0})(w_{k-1} - \underline{0})^T] = \\
 &= A_k \mathbb{E} [(e_{k-1} - \mathbb{E}[e_{k-1}])(e_{k-1} - \mathbb{E}[e_{k-1}])^T] A_k^T + \\
 &\quad - A_k \mathbb{E} [(e_{k-1} - \mathbb{E}[e_{k-1}])(w_{k-1} - \mathbb{E}[w_{k-1}])^T] + \\
 &\quad - \mathbb{E} [(w_{k-1} - \mathbb{E}[w_{k-1}])(e_{k-1} - \mathbb{E}[e_{k-1}])^T] A_k^T + \\
 &\quad + \mathbb{E} [(w_{k-1} - \mathbb{E}[w_{k-1}])(w_{k-1} - \mathbb{E}[w_{k-1}])^T] = \\
 &= A_k \text{Cov}(e_{k-1}, e_{k-1}) A_k^T - A_k \text{Cov}(e_{k-1}, w_{k-1}) + \\
 &\quad - \text{Cov}(w_{k-1}, e_{k-1}) A_k^T + \text{Cov}(w_{k-1}, w_{k-1})
 \end{aligned}$$

Si osservi che i termini centrali si annullano se e_{k-1} e w_{k-1} sono non correlati. In effetti, analizzando ricorsivamente l'errore e_{k-1} , si trova che le uniche variabili aleatorie da cui esso dipende sono u_0, \dots, u_{k-2} , w_0, \dots, w_{k-2} , v_1, \dots, v_{k-1} e x_0 , ossia variabili con cui w_{k-1} è non correlato per ipotesi.

Ricordando la definizione di P_{k-1} e definendo Q_{k-1} come la auto-covarianza di w_{k-1} , si trova, dunque, l'espressione:

$$P_k^- = A_k P_{k-1} A_k^T + Q_{k-1}$$

La scelta iniziale di P_0 è pressoché libera. Se si hanno informazioni sulla distribuzione di x_0 , è consigliabile utilizzare una stima della sua auto-covarianza. In ogni caso, è necessario avere $P_0 \neq 0_{n \times n}$ e semidefinita positiva.

Per trovare un'espressione per P_k , si consideri inizialmente:

$$\begin{aligned}\hat{x}_k &= (I - \mathcal{K}_k H_k) \hat{x}_k^- + \mathcal{K}_k z_k \\ \Rightarrow e_k &= \hat{x}_k - x_k = (I - \mathcal{K}_k H_k) \hat{x}_k^- + \mathcal{K}_k z_k - x_k\end{aligned}$$

Sostituendo z_k con la sua definizione (eq. 1.4):

$$\begin{aligned}e_k &= (I - \mathcal{K}_k H_k) \hat{x}_k^- + \mathcal{K}_k (H_k x_k + v_k) - x_k = \\ &= (I - \mathcal{K}_k H_k) \hat{x}_k^- + \mathcal{K}_k H_k x_k + \mathcal{K}_k v_k - x_k = \\ &= (I - \mathcal{K}_k H_k) \hat{x}_k^- - (I - \mathcal{K}_k H_k) x_k + \mathcal{K}_k v_k = \\ &= (I - \mathcal{K}_k H_k) (\hat{x}_k^- - x_k) + \mathcal{K}_k v_k = (I - \mathcal{K}_k H_k) e_k^- + \mathcal{K}_k v_k\end{aligned}$$

Dunque, si ha:

$$\begin{aligned}P_k &= \mathbb{E} [(e_k - \mathbb{E}[e_k])(e_k - \mathbb{E}[e_k])^T] = \mathbb{E} [e_k (e_k)^T] = \\ &= \mathbb{E} \left[((I - \mathcal{K}_k H_k) e_k^- + \mathcal{K}_k v_k) ((I - \mathcal{K}_k H_k) e_k^- + \mathcal{K}_k v_k)^T \right] = \\ &= \mathbb{E} \left[((I - \mathcal{K}_k H_k) e_k^- + \mathcal{K}_k v_k) ((e_k^-)^T (I - \mathcal{K}_k H_k)^T + v_k^T \mathcal{K}_k^T) \right] = \\ &= (I - \mathcal{K}_k H_k) \mathbb{E} [e_k^- (e_k^-)^T] (I - \mathcal{K}_k H_k)^T + (I - \mathcal{K}_k H_k) \mathbb{E} [e_k^- (v_k)^T] \mathcal{K}_k^T + \\ &\quad + \mathcal{K}_k \mathbb{E} [v_k (e_k^-)^T] (I - \mathcal{K}_k H_k)^T + \mathcal{K}_k \mathbb{E} [v_k (v_k)^T] \mathcal{K}_k^T\end{aligned}$$

Anche in questo caso, l'espressione si riduce alle covarianze. I termini centrali si annullano, poiché le uniche variabili aleatorie ad influenzare e_k^- sono u_0, \dots, u_{k-2} , w_0, \dots, w_{k-1} , v_1, \dots, v_{k-1} e x_0 , ossia variabili con cui v_k è non correlato per ipotesi. Ricordando la definizione di P_k^- e definendo R_k come la auto-covarianza di v_k , si trova l'espressione provvisoria:

$$P_k = (I - \mathcal{K}_k H_k) P_k^- (I - \mathcal{K}_k H_k)^T + \mathcal{K}_k R_k \mathcal{K}_k^T$$

Si osservi che minimizzare l'errore quadratico medio $\mathbb{E}[\|e_k\|^2]$ equivale a minimizzare la traccia di P_k . In effetti:

$$\mathbb{E}[\|e_k\|^2] = \mathbb{E}[e_k^T e_k] = \mathbb{E}[\text{tr}(e_k^T e_k)] = \mathbb{E}[\text{tr}(e_k (e_k)^T)] = \text{tr}(\mathbb{E}[e_k (e_k)^T]) = \text{tr}(P_k)$$

Dunque, si calcola:

$$\begin{aligned}
\text{tr}(P_k) &= \text{tr}\left((I - \mathcal{K}_k H_k) P_k^- (I - \mathcal{K}_k H_k)^T + \mathcal{K}_k R_k \mathcal{K}_k^T\right) = \\
&= \text{tr}\left((I - \mathcal{K}_k H_k) P_k^- (I - H_k^T \mathcal{K}_k^T) + \mathcal{K}_k R_k \mathcal{K}_k^T\right) = \\
&= \text{tr}\left(P_k^- - P_k^- H_k^T \mathcal{K}_k^T - \mathcal{K}_k H_k P_k^- + \mathcal{K}_k H_k P_k^- H_k^T \mathcal{K}_k^T + \mathcal{K}_k R_k \mathcal{K}_k^T\right)
\end{aligned}$$

Essendo P_k^- simmetrica, vale $P_k^- H_k^T \mathcal{K}_k^T = (P_k^-)^T H_k^T \mathcal{K}_k^T = (\mathcal{K}_k H_k P_k^-)^T$, per cui:

$$\text{tr}(P_k) = \text{tr}(P_k^-) - 2 \text{tr}(\mathcal{K}_k H_k P_k^-) + \text{tr}(\mathcal{K}_k H_k P_k^- H_k^T \mathcal{K}_k^T) + \text{tr}(\mathcal{K}_k R_k \mathcal{K}_k^T)$$

Per ricercare \mathcal{K}_k che minimizza $\text{tr}(P_k)$ si pone:

$$\begin{aligned}
\frac{\partial \text{tr}(P_k)}{\partial \mathcal{K}_k} &= 0_{n \times n} \\
\Rightarrow \frac{\partial \text{tr}(P_k^-)}{\partial \mathcal{K}_k} - 2 \frac{\partial \text{tr}(\mathcal{K}_k H_k P_k^-)}{\partial \mathcal{K}_k} + \frac{\partial \text{tr}(\mathcal{K}_k H_k P_k^- H_k^T \mathcal{K}_k^T)}{\partial \mathcal{K}_k} + \\
&\quad + \frac{\partial \text{tr}(\mathcal{K}_k R_k \mathcal{K}_k^T)}{\partial \mathcal{K}_k} = 0_{n \times n}
\end{aligned}$$

Si osserva che $\text{tr}(P_k^-)$ è costante in K_k , per cui il primo termine si annulla. Per calcolare i termini restanti, si utilizzino le seguenti identità, con la seconda valida se N è simmetrica:

$$\frac{\partial \text{tr}(MN)}{\partial M} = N^T, \quad \frac{\partial \text{tr}(MNM^T)}{\partial M} = 2MN$$

Si ottiene:

$$\begin{aligned}
-2(H_k P_k^-)^T + 2\mathcal{K}_k (H_k P_k^- H_k^T) + 2\mathcal{K}_k R_k &= 0_{n \times n} \\
\Rightarrow -2P_k^- H_k^T + 2\mathcal{K}_k H_k P_k^- H_k^T + 2\mathcal{K}_k R_k &= 0_{n \times n} \\
\Rightarrow \mathcal{K}_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}
\end{aligned}$$

Esaminando la matrice Hessiana di $\text{tr}(P_k)$, la quale risulta essere semidefinita positiva, si può verificare che il valore trovato rappresenta un minimo globale.

La quantità \mathcal{K}_k è detta matrice dei guadagni di Kalman (*Kalman gain*) e può essere pensata come un indice di affidabilità delle misurazioni rispetto al modello teorico. In effetti, per $R_k \rightarrow 0_{m \times m}$ si ha $\mathcal{K}_k \rightarrow H^{-1}$, per cui $\hat{x}_k \rightarrow z_k$, mentre per $P_k^- \rightarrow 0_{n \times n}$ si ha $\mathcal{K}_k \rightarrow 0_{n \times m}$, per cui $\hat{x}_k \rightarrow \hat{x}_k^-$.

Sostituendo il valore trovato, l'espressione per P_k diventa:

$$\begin{aligned}
 P_k &= P_k^- - P_k^- H_k^T \mathcal{K}_k^T - \mathcal{K}_k H_k P_k^- + \mathcal{K}_k H_k P_k^- H_k^T \mathcal{K}_k^T + \mathcal{K}_k R_k \mathcal{K}_k^T = \\
 &= (I - \mathcal{K}_k H_k) P_k^- - P_k^- H_k^T \mathcal{K}_k^T + \mathcal{K}_k (H_k P_k^- H_k^T + R_k) \mathcal{K}_k^T = \\
 &= (I - \mathcal{K}_k H_k) P_k^- - P_k^- H_k^T \mathcal{K}_k^T + \\
 &\quad + P_k^- H_k^T \underbrace{(H_k P_k^- H_k^T + R_k)^{-1} (H_k P_k^- H_k^T + R_k)}_I \mathcal{K}_k^T = \\
 &= (I - \mathcal{K}_k H_k) P_k^- - P_k^- H_k^T \mathcal{K}_k^T + P_k^- H_k^T \mathcal{K}_k^T = \\
 &= (I - \mathcal{K}_k H_k) P_k^-
 \end{aligned}$$

Riassumendo, l'algoritmo si basa sulle seguenti formule:

Inizializzazioni:

$$\hat{x}_0 = \mathbb{E}[x_0]$$

$$P_0 = \text{Cov}(x_0, x_0)$$

Fase di predizione:

$$\hat{x}_k^- = A_k \hat{x}_{k-1} + B u_{k-1}$$

$$P_k^- = A_k P_{k-1} A_k^T + Q_{k-1}$$

Fase di correzione:

$$\mathcal{K}_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + \mathcal{K}_k (z_k - H_k \hat{x}_k^-)$$

$$P_k = (I - \mathcal{K}_k H_k) P_k^-$$

1.4 Varianti non lineari

Sulla base della formulazione originale di Kálmán — oggi conosciuta come “Standard” Kalman Filter o KF — sono state sviluppate numerose varianti del filtro, con l’obiettivo principale di estenderne il campo di applicazione, specie a modelli non lineari. Si presenta qui una descrizione delle varianti più conosciute, successivamente schematizzate in fig. 1.1.

1.4.1 Extended Kalman Filter

EKF [2] nasce dalle prime applicazioni delle intuizioni di Kálmán e Bucy al programma Apollo [6]. Si basa sull’utilizzo del polinomio di Taylor di grado 1 (ossia la linearizzazione tramite la derivata prima) per ottenere approssimazioni lineari alle equazioni del sistema. Questo permette di applicare la moderna teoria dei filtri di Kalman anche in contesti non lineari, come quelli affrontati nelle missioni spaziali Apollo.

Partendo dalla formula generale del polinomio in una variabile reale:

$$f(x) = f(a) + f'(a)(x - a) + o(x - a)$$

e applicandola alla legge di evoluzione f di un sistema dinamico, si ottiene:

$$f(k, x_{k-1}, u_{k-1}) \approx f(k, \hat{x}_{k-1}, u_{k-1}) + J_x f(k, \hat{x}_{k-1}, u_{k-1})(x_{k-1} - \hat{x}_{k-1})$$

Qui, $f'(a)$ è sostituita da $J_x f(k, \hat{x}_{k-1}, u_{k-1})$, ossia dalla matrice jacobiana delle derivate parziali di f rispetto ad x calcolate in $k, \hat{x}_{k-1}, u_{k-1}$.

Da qui in poi verrà indicata con A_k , poiché verrà utilizzata nelle formule del filtro in modo identico alla matrice dinamica dei sistemi dinamici lineari.

$$(A_k)_{ij} = \left. \frac{\partial f_i}{\partial x_j} \right|_{k, \hat{x}_{k-1}, u_{k-1}}$$

Lo stesso ragionamento vale per la funzione di misurazione h :

$$h(k, x_k) \approx h(k, \hat{x}_k) + J_x h(k, \hat{x}_k)(x_{k-1} - \hat{x}_{k-1})$$

$$H_k = J_x h(k, \hat{x}_k) \Rightarrow (H_k)_{ij} = \left. \frac{\partial h_i}{\partial x_j} \right|_{k, \hat{x}_k}$$

Dunque, il sistema è descritto dalle seguenti approssimazioni lineari:

$$x_k \approx f(k, \hat{x}_{k-1}, u_{k-1}) + A_k(x_{k-1} - \hat{x}_{k-1}) + w_{k-1}$$

$$z_k \approx h(k, \hat{x}_k) + H_k(x_{k-1} - \hat{x}_{k-1}) + v_k$$

Una volta determinate le matrici jacobiane A_k e H_k , EKF fa uso delle stesse formule di KF per il calcolo di P_k^- , P_k e K_k . Dove possibile, ossia per il calcolo delle stime, utilizza, invece, le funzioni f e h direttamente:

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1})$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-))$$

Il filtro risultante è poco più complesso di KF, ma con un evidente vantaggio in termini di flessibilità. Tuttavia, il processo di linearizzazione è efficace solo per piccoli errori e può essere soggetto a rapida divergenza.

Un logico sviluppo di queste idee prevede di includere anche i termini di secondo grado nelle espansioni in serie di Taylor. Si parla in questo caso di Second-Order Extended Kalman Filter, abbreviato EKF2 o SO-EKF.

1.4.2 Unscented Kalman Filter

UKF [7] è il più noto dei cosiddetti filtri Sigma-Point. Questi filtri selezionano deterministicamente un insieme ridotto di punti chiamati “Sigma-Point” ($2n + 1$ per UKF, con n dimensione dello stato) attorno alla stima precedente \hat{x}_{k-1} e studiano come questi punti si trasformano attraverso la funzione non lineare, per ricostruire una stima più precisa della media e covarianza senza dover linearizzare la funzione

con il polinomio di Taylor come per EKF.

UKF utilizza tre parametri che influenzano la scelta dei Sigma-Point:

- α , con $0 < \alpha \ll 1$, regola la dispersione dei Sigma-Point attorno alla stima media. È un parametro di scala che influenza quanto i punti si distribuiscono intorno al valore centrale.
- β è usato per incorporare informazioni sulla distribuzione dello stato (per variabili gaussiane il valore ottimale è 2).
- κ è un parametro di sovrappeso scalare, solitamente impostato a 0, che può aiutare a stabilizzare la selezione dei punti.

Una volta definiti tali parametri, si calcola il parametro intermedio $\lambda = \alpha^2(n + \kappa) - n$, che serve per costruire la matrice $(\lambda + n)P_{k-1}$ con P_{k-1} la matrice di covarianza dello stato all'istante precedente. Per trovare i Sigma-Point, si esegue una decomposizione (tipicamente una decomposizione di Cholesky) di questa matrice, ottenendo la matrice $U = \sqrt{(\lambda + n)P_{k-1}}$.

Ora, indicando con $U_{[i]}$ la i -esima riga di U , i punti selezionati sono dati da:

$$\mathbf{x}_{k-1}^{(i)} = \begin{cases} \hat{x}_{k-1} & \text{se } i = 0 \\ \hat{x}_{k-1} + U_{[i]} & \text{se } i = 1, \dots, n \\ \hat{x}_{k-1} - U_{[i-n]} & \text{se } i = n + 1, \dots, 2n \end{cases}$$

A ciascuno viene anche associato un peso fisso sul calcolo del valore medio e uno sul calcolo della covarianza:

$$W^{(\text{mean}, i)} = \begin{cases} \frac{\lambda}{\lambda + n} & \text{se } i = 0 \\ \frac{1}{2(\lambda + n)} & \text{se } i = 1, \dots, 2n \end{cases}$$

$$W^{(\text{cov}, i)} = \begin{cases} \frac{\lambda}{\lambda + n} + (1 - \alpha^2 + \beta) & \text{se } i = 0 \\ \frac{1}{2(\lambda + n)} & \text{se } i = 1, \dots, 2n \end{cases}$$

Dunque, viene applicata la trasformazione non lineare, per determinare la stima *a priori* e la sua covarianza come somme pesate:

$$\begin{aligned}\mathbf{x}_k^{(i)} &= f(k, \mathbf{x}_{k-1}^{(i)}, u_{k-1}) \quad \forall i = 0, \dots, 2n \\ \hat{x}_k^- &= \sum_{i=0}^{2n} W^{(\text{mean}, i)} \mathbf{x}_k^{(i)} \\ P_k^- &= \sum_{i=0}^{2n} W^{(\text{cov}, i)} \left(\mathbf{x}_k^{(i)} - \hat{x}_k^- \right) \left(\mathbf{x}_k^{(i)} - \hat{x}_k^- \right)^T\end{aligned}$$

Similmente, nella fase di correzione viene applicata la trasformazione h per trovare una previsione della misurazione \hat{z}_k^- (si mantiene la notazione all'apice per indicare che il calcolo viene effettuato prima della misurazione, seppure non sia necessario migliorare successivamente questa stima). Ne viene poi calcolata sia la auto-covarianza che la covarianza incrociata con la stima *a priori*:

$$\begin{aligned}\mathbf{z}_k^{(i)} &= h(k, \mathbf{x}_{k-1}^{(i)}) \quad \forall i = 0, \dots, 2n \\ \hat{z}_k^- &= \sum_{i=0}^{2n} W^{(\text{mean}, i)} \mathbf{z}_k^{(i)} \\ \text{Cov}(\hat{z}_k^-, \hat{z}_k^-) &= \sum_{i=0}^{2n} W^{(\text{cov}, i)} \left(\mathbf{z}_k^{(i)} - \hat{z}_k^- \right) \left(\mathbf{z}_k^{(i)} - \hat{z}_k^- \right)^T \\ \text{Cov}(\hat{x}_k^-, \hat{z}_k^-) &= \sum_{i=0}^{2n} W^{(\text{cov}, i)} \left(\mathbf{x}_k^{(i)} - \hat{x}_k^- \right) \left(\mathbf{z}_k^{(i)} - \hat{z}_k^- \right)^T\end{aligned}$$

A questo punto, la matrice dei guadagni di Kalman viene calcolata come:

$$\mathcal{K}_k = \text{Cov}(\hat{x}_k^-, \hat{z}_k^-) \text{Cov}(\hat{z}_k^-, \hat{z}_k^-)^{-1}$$

Infine, la stima *a posteriori* e la relativa auto-covarianza sono date da:

$$\begin{aligned}\hat{x}_k &= \hat{x}_k^- + \mathcal{K}_k (z_k - \hat{z}_k^-) \\ P_k &= P_k^- - \mathcal{K}_k \text{Cov}(\hat{z}_k^-, \hat{z}_k^-) \mathcal{K}_k^T\end{aligned}$$

UKF permette di ottenere stime di sistemi non lineari con una precisione equi-

valente a quella di un polinomio di Taylor di grado 3. Questo significa che l'approssimazione della trasformazione non lineare fatta da UKF è accurata fino al terzo ordine di derivata della funzione non lineare, ma con la stessa complessità computazionale di EKF base.

Sono disponibili diverse implementazioni di questo algoritmo. In questa tesi si farà uso della classe `filterpy.kalman.UnscentedKalmanFilter`, con i relativi Sigma-Point `filterpy.kalman.sigma_points.MerweScaledSigmaPoints`.

Altre varianti di tipo Sigma-Point includono i filtri Quadrature KF (QKF), fra cui il più conosciuto Gauss-Hermite Kalman Filter (GHKF), basati sulle formule di integrazione numerica (o quadratura numerica) di Gauss, e i Cubature KF (CKF), che utilizzano la “regola di cubatura”, ossia l'applicazione delle stesse regole di quadratura ad uno spazio in coordinate sferiche.

1.4.3 Metodo Monte Carlo

Un'ulteriore strategia per lo studio della distribuzione di una variabile aleatoria sottoposta ad una trasformazione non lineare è il classico metodo Monte Carlo. Diversamente dai metodi Sigma-Point, si considerano campioni cospicui di punti, consentendo di raggiungere precisione arbitraria a costo di un maggiore onere computazionale, il quale, però, può essere facilmente distribuito con tecniche di parallelizzazione. Da queste idee nascono Ensemble KF (EnKF) e i Particle Filters (PF).

Ensemble Kalman Filter (EnKF): basato su un campione (*ensemble*) di stati che rappresentano la distribuzione di probabilità, mantiene la supposizione di distribuzioni Gaussiane iniziali. EnKF sostituisce la matrice di covarianza con la covarianza campionaria calcolata sull'*ensemble*, evitando così di dover gestire esplicitamente matrici di covarianza grandi e complesse. Ogni membro dell'*ensemble* evolve indipendentemente e viene aggiornato in seguito all'osservazione, assicurando efficienza computazionale anche in spazi di stato di dimensioni molto elevate.

Particle Filter (PF): estremamente flessibile e robusto anche per distribuzioni non Gaussiane, usa un insieme di particelle (campioni) per rappresentare la di-

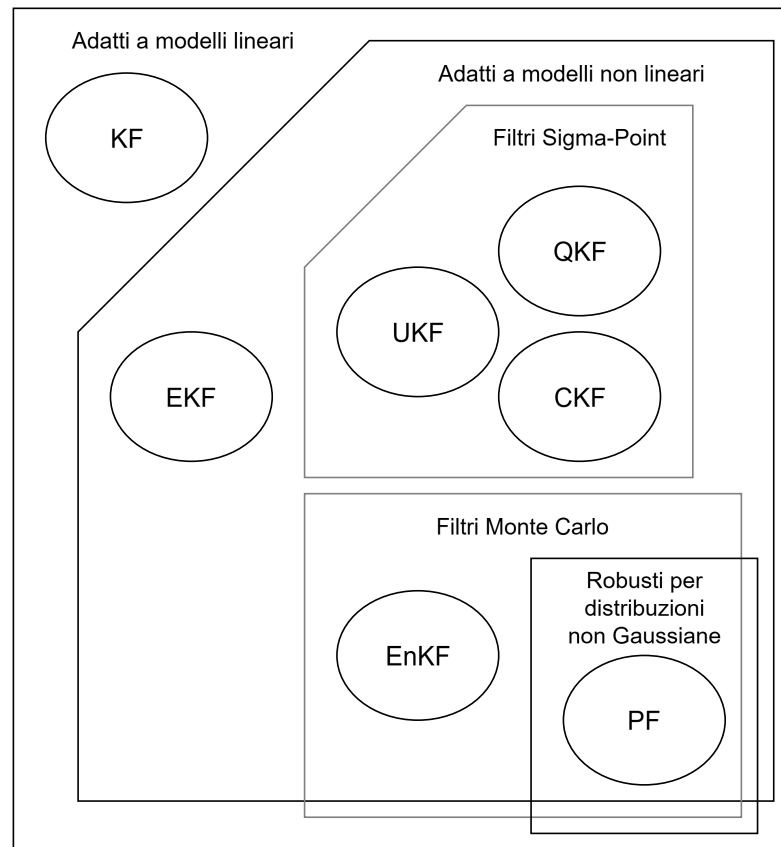


Figura 1.1: Diagramma di classificazione delle varianti del filtro di Kalman.

sistribuzione dello stato, con pesi che riflettono la probabilità. È però più costoso computazionalmente e meno efficiente rispetto a EnKF.

1.5 Sviluppi recenti

Lo studio del filtro di Kalman resta un settore vivo, a più di 60 anni dalla sua concezione. Si continuano a identificare nuovi settori di applicazione, nuove tecniche di miglioramento dell'accuratezza, della stabilità numerica e della performance computazionale. Nuove varianti, sia generiche che specializzate, si aggiungono a quelle già consolidate. Solo negli ultimi mesi si evidenziano nuove tecniche di “Iteratively Saturated Kalman Filtering” (ISKF) [8] e la possibilità di affiancare una

rete neurale profonda (DNN) ad un filtro classico nei cosiddetti “AI-Aided Kalman Filters” [9].

ISKF è una nuova variante del filtro di Kalman progettata per rendere il filtro più robusto nei confronti di misurazioni anomale o *outlier*. Questo approccio modifica iterativamente il passo di aggiornamento del filtro standard applicando una funzione di saturazione per compensare errori e rumori anomali nei dati di misura e nei rumori di processo. Derivato come un metodo di gradiente scalato per risolvere un problema convesso di stima robusta, ISKF mantiene la semplicità e l’efficienza computazionale del filtro di Kalman tradizionale, spesso richiedendo solo una o due iterazioni per migliorare significativamente la stima in presenza di *outlier*. Inoltre, possiede una variante in stato stazionario che ne migliora l’efficienza per applicazioni in tempo reale, senza richiedere complessi calcoli matriciali ad ogni passo.

I filtri AI-Aided ibridano i filtri di Kalman classici con reti neurali profonde (DNN) per superare i limiti imposti dai modelli di stato-spazio semplificati o inaccurati. Le DNN possono apprendere rappresentazioni più complesse e non lineari del sistema dinamico direttamente dai dati, permettendo così un miglioramento nel tracking e nella stima dello stato in scenari complessi. Le tecniche di AI-aided KF si dividono in:

- Approcci orientati al compito (task-oriented), che integrano reti DNN esterne o parallele al filtro per migliorare la stima.
- Approcci orientati al modello di stato-spazio (SS model-oriented), in cui le DNN aiutano nell’identificazione del modello matematico parziale del sistema, migliorando il filtro basato su model-based tracking.

1.6 Limitazioni

Sebbene oggi assuma un ruolo fondamentale nell’ingegneria aerospaziale e in diversi altri settori, il filtro di Kalman presenta alcune limitazioni intrinseche:

- La prima riguarda i sistemi non lineari con distribuzioni di rumore non gaussiane: in questi casi, l’ottimalità dell’algoritmo non è garantita, perché le

assunzioni fondamentali del filtro (linearità e gaussianità) vengono violate. Questo porta a stime spesso sub-ottimali, soprattutto quando i modelli di sistema si discostano dalla realtà e le distribuzioni di errore sono complesse.

- La seconda limitazione è legata alla dipendenza dall'accuratezza del modello matematico che descrive il sistema reale. Il filtro utilizza questo modello per stimare l'evoluzione dello stato nel tempo, ma se il modello è impreciso o incompleto, la stima risulterà inevitabilmente meno affidabile. Questo è particolarmente critico in sistemi complessi, dove non sempre è possibile costruire modelli sufficientemente dettagliati o precisi. Inoltre, l'adattamento e la regolazione dei parametri del filtro richiedono spesso procedure empiriche e possono introdurre ulteriori fonti di errore o instabilità.

Capitolo 2

Modello astronomico

In questo capitolo si presentano le formule e i modelli teorici che governano le osservazioni di corpi celesti e il tracciamento del loro moto. Si delinea un sistema dinamico altamente non lineare, che dimostra la necessità di tecniche e algoritmi avanzati per lo studio di questo caso d'uso reale.

2.1 Coordinate celesti

Le osservazioni astrometriche fanno uso di una coppia di angoli (α, δ) per identificare la posizione dei corpi nella sfera celeste, o, più precisamente, per identificare la direzione orientata dall'osservatore verso la posizione apparente del corpo considerato. I due angoli sono definiti con riferimento al piano equatoriale e al punto vernale dell'epoca J2000.0 (mezzogiorno del 1° gennaio 2000, calendario gregoriano e tempo terrestre). Nella pratica si considerano assi orientati secondo il sistema ICRS, adottato dall'Unione Astronomica Internazionale dal 1° gennaio 1998, spostando l'origine dal baricentro del sistema solare alla posizione dell'osservatore (osservazione topocentrica). In questo contesto, l'ascensione retta $\alpha \in [0, 2\pi)$ rappresenta la rotazione attorno all'asse z , mentre la declinazione $\delta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ rappresenta la distanza angolare dal piano xy .

Dunque, indicando con Δ la distanza apparente tra l'osservatore e il corpo al momento dell'osservazione, la posizione apparente del corpo osservato, in coordinate cartesiane topocentriche, è data da:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R_z(\alpha) R_y(-\delta) \begin{pmatrix} \Delta \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \Delta \cos(\delta) \cos(\alpha) \\ \Delta \cos(\delta) \sin(\alpha) \\ \Delta \sin(\delta) \end{pmatrix} \quad (2.1)$$

La conversione a coordinate baricentriche richiede semplicemente di sommare la posizione dell'osservatore rispetto al baricentro del sistema solare.

Si definiscono le distanze apparenti R tra l'osservatore e il Sole al momento dell'osservazione e d (solitamente indicata con r , ma da non confondersi con la distanza dal fuoco occupato dell'orbita) tra il corpo osservato e il Sole nel momento in cui la luce osservata viene riflessa sulla superficie del corpo. Si definiscono gli angoli di separazione $\theta \in [0, \pi]$, detta *elongazione*, tra il corpo e il Sole come appaiono all'osservatore al momento dell'osservazione e $\phi \in [0, \pi]$, detto angolo di fase, tra l'osservatore e il Sole come appaiono dal corpo nel momento in cui la luce osservata viene riflessa. Seppure si tratti di una definizione analoga, l'angolo di fase viene definito, in genere, come l'angolo tra la luce incidente e la luce riflessa verso l'osservatore, per via del suo impatto sul modello ottico descritto nella sezione 2.2.

Si è parlato finora di quantità apparenti. Questa precisazione è necessaria per via della velocità finita della luce e ad altri fenomeni come aberrazioni, rifrazioni ed effetti relativistici. Da questo punto in poi si considereranno le quantità apparenti come coincidenti alle controparti “reali” (geometriche), con la consapevolezza che ciò introdurrà imprecisioni nel modello, le quali dovranno essere considerate nell'applicazione del filtro di Kalman. Queste imprecisioni sono accettabili per il tracciamento di asteroidi, ma possono diventare problematiche per oggetti più veloci.

Con questa semplificazione diventa possibile considerare il triangolo tra il corpo osservato, l'osservatore e il Sole e applicare il teorema dei seni e il teorema dei coseni

per determinare relazioni fra gli angoli e le distanze. In particolare:

$$\frac{d}{\sin(\theta)} = \frac{R}{\sin(\phi)} \quad (2.2)$$

$$\frac{\Delta}{\sin(\pi - \theta - \phi)} = \frac{R}{\sin(\phi)} \Rightarrow \frac{\Delta}{\sin(\theta + \phi)} = \frac{R}{\sin(\phi)} \quad (2.3)$$

$$r^2 + \Delta^2 - 2r\Delta \cos(\phi) = R^2 \Rightarrow \phi = \arccos\left(\frac{r^2 + \Delta^2 - R^2}{2r\Delta}\right) \quad (2.4)$$

2.2 Fotometria e il sistema H-G

Nello studio degli asteroidi, l'intensità della luce osservata è espressa dalla magnitudine apparente e può essere utilizzata per stimare la distanza del corpo. Intuitivamente, questa quantità dipende dall'angolo di fase ($\phi \approx 0$ indica un corpo perfettamente illuminato, mentre $\phi \approx 1$ comporta che non venga riflessa luce verso l'osservatore), dalle distanze d e Δ e da alcune caratteristiche fisiche del corpo. Tuttavia non esiste un modello macroscopico esatto.

Una delle prime applicazioni delle nozioni di fotometria allo studio degli asteroidi ha prodotto il modello H-G [10] [11], usato tuttora quando non si dispone di informazioni sufficienti per impiegare modelli più precisi. Gli asteroidi vengono caratterizzati da due parametri: H , detta magnitudine assoluta, che equivale alla magnitudine apparente per $d = \Delta = 1$ AU e $\phi = 0$ e G , detto *slope parameter*, che quantifica un picco di luminosità che spesso si verifica nella curva di fase vicino all'opposizione. Se si esprimono le distanze in unità astronomiche, la magnitudine apparente è descritta da:

$$V = H + 5 \log_{10}(d\Delta) - 2.5 \log_{10}(\Phi(\phi)) \quad (2.5)$$

$$\Phi(\phi) = (1 - G) \exp\left(-3.33\left(\tan\left(\frac{\phi}{2}\right)\right)^{0.63}\right) + G \exp\left(-1.87\left(\tan\left(\frac{\phi}{2}\right)\right)^{1.22}\right) \quad (2.6)$$

2.3 Orbite ellittiche

Il moto dei pianeti, degli asteroidi e delle comete è ben approssimato da orbite ellittiche, con un fuoco occupato dal baricentro del sistema solare.

Tali orbite possono essere caratterizzate tramite i parametri orbitali kepleriani:

- il semiasse maggiore a ;
- l'eccentricità e , dove $e = 0$ indica un'orbita circolare, mentre $e \rightarrow 1$ descrive un'orbita progressivamente più schiacciata;
- l'inclinazione $i \in [0, \pi]$ del piano dell'orbita, in genere espressa rispetto al piano dell'orbita della Terra, ma in questo caso espressa rispetto al piano xy del sistema di riferimento, con $i < \frac{\pi}{2}$ indicante un'orbita progradata e $i > \frac{\pi}{2}$ indicante un'orbita retrogradata;
- la longitudine del nodo ascendente Ω ;
- l'argomento del periapside ω .

La posizione del corpo nell'orbita è identificata dall'anomalia. Più precisamente, la distanza angolare dal periapside è detta anomalia vera e indicata con ν .

Si definisce la distanza radiale r dal fuoco occupato e l'argomento di latitudine $u = \omega + \nu$, cosicché le coordinate cartesiane del corpo siano date da:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R_z(\Omega)R_x(i)R_z(u) \begin{pmatrix} r \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} r(\cos(\Omega)\cos(u) - \sin(\Omega)\sin(u)\cos(i)) \\ r(\sin(\Omega)\cos(u) + \cos(\Omega)\sin(u)\cos(i)) \\ r\sin(u)\sin(i) \end{pmatrix} \quad (2.7)$$

Per studiare l'evoluzione della posizione nel tempo si introducono i concetti di anomalia media M e anomalia eccentrica E .

La prima avanza linearmente nel tempo in base al moto medio n e al tempo di periapside τ :

$$M = n(t - \tau) \quad (2.8)$$

Il moto medio è definito dal periodo dell'orbita come $n = \frac{2\pi}{T}$ ed è ricavabile dal semiasse maggiore e dal parametro gravitazionale $\mu = GM$ del corpo primario (in questo caso G è la costante di gravitazione universale e M è la massa del corpo primario):

$$n = \sqrt{\frac{\mu}{a^3}} \quad (2.9)$$

Una volta determinata l'anomalia media, si risale all'anomalia eccentrica esprimendo gli angoli in radianti e risolvendo l'equazione di Keplero tramite metodi numerici:

$$M = E - e \sin(E) \quad (2.10)$$

In questo caso si è scelto di utilizzare il metodo di Newton, ponendo $f(x) = x - e \sin(x) - M$, $f'(x) = 1 - e \cos(x)$ e $x_0 = M$.

Dall'anomalia eccentrica è possibile passare all'anomalia vera. Una formula numericamente stabile richiede di calcolare il valore intermedio β :

$$\beta = \frac{e}{1 + \sqrt{1 - e^2}} \quad (2.11)$$

$$\nu = E + 2 \arctan \left(\frac{\beta \sin(E)}{1 - \beta \cos(E)} \right) \quad (2.12)$$

Infine, la distanza radiale è legata all'anomalia eccentrica da:

$$r = a(1 - e \cos(E)) \quad (2.13)$$

Capitolo 3

Adattamento del filtro di Kalman al caso di studio: formulazione e risultati

In questo capitolo si illustra come sono state messe in pratica le nozioni teoriche in modo da sperimentare concretamente l'utilizzo del filtro di Kalman, come sono state adattate le formule astronomiche per ottenere le equazioni del sistema, come sono state ricavate le matrici di covarianza e le sfide affrontate durante il percorso. Si presentano, infine, i risultati ottenuti dapprima su dati simulati e successivamente su dati reali.

3.1 Approccio diretto

Prima ancora di considerare strategie di filtraggio o di tracciamento del moto orbitale, si potrebbe pensare di risalire a una posizione direttamente dai dati osservati. In effetti, se si conoscesse la distanza del corpo Δ , la si potrebbe sfruttare, insieme agli angoli α, δ , nella formula 2.1. Tuttavia, risalire a Δ dalla magnitudine visuale V richiede particolare attenzione.

Idealmente, si vorrebbe sfruttare direttamente la formula 2.5, ma non si dispone della distanza dal Sole d e dell'angolo di fase ϕ . Una possibile soluzione prevede

di risolvere in ϕ la seguente equazione tramite metodi numerici:

$$R^2 \sin(\theta) \sin(\theta + \phi) - 10^{0.2(V-H)} \sin^2(\phi) \sqrt{\Phi(\phi)} = 0 \quad (3.1)$$

Tale equazione deriva dai seguenti passaggi:

$$\begin{aligned} V &= H + 5 \log_{10}(d\Delta) - 2.5 \log_{10}(\Phi(\phi)) \\ &\rightarrow 0.2(V - H) + \log_{10}((\Phi(\phi))^{0.5}) = \log_{10}(d\Delta) \\ &\rightarrow d\Delta = 10^{0.2(V-H)} \sqrt{\Phi(\phi)} \end{aligned}$$

Utilizzando le equazioni 2.2 e 2.3, si trova che:

$$d\Delta = \frac{R^2 \sin(\theta) \sin(\theta + \phi)}{\sin^2(\phi)}$$

Sostituendo e moltiplicando entrambi i lati per $\sin^2(\phi)$ si ottiene l'espressione desiderata. Si osservi che per $\phi = 0$ e $\phi = \pi$ sarebbe impossibile effettuare osservazioni valide, per cui si può assumere $\sin(\phi) \neq 0$. Lo stesso vale per l'elongazione θ .

Il vantaggio di questa specifica scrittura sta nel fatto che l'espressione ha valore positivo finito per $\phi = 0$, ovvero $R^2 \sin^2(\theta)$, e ha valore negativo finito per $\phi \rightarrow \pi$, ovvero $-R^2 \sin^2(\theta)$, il che permette di utilizzare il metodo di Brent nell'intervallo $[0, \pi]$.

Una volta determinata una radice, la distanza è data da:

$$\Delta = \frac{R \sin(\theta + \phi)}{\sin(\phi)} \quad (3.2)$$

Si osservi che in generale può esistere più di una radice, specie se l'elongazione θ è piccola; tuttavia questo non si verifica nei dataset considerati.

In ogni caso, questo processo è estremamente sensibile all'errore sui dati iniziali e la magnitudine risulta essere una misura piuttosto imprecisa. Tutto ciò evidenzia ulteriormente la necessità di strategie più sofisticate.

Questo approccio iniziale resta comunque un utile punto di riferimento e verrà utilizzato per quantificare l'efficacia del filtro di Kalman in questo contesto.

Per il calcolo di R e θ si può sfruttare un servizio di effemeridi che forni-

sca le posizioni dell'osservatore e del Sole. In questo caso si è utilizzato il servizio JPL, sia attraverso le funzioni `astropy.coordinates.get_body_barycentric` e `astropy.coordinates.EarthLocation.get_gcrs_posvel`, che dalla piattaforma web <https://ssd.jpl.nasa.gov/horizons/app.html>.

Indicando con \hat{R} il versore dall'osservatore verso il sole e con $\hat{\Delta}$ il versore dall'osservatore verso il corpo, l'elongazione è data da:

$$\theta = \arccos(\hat{\Delta} \cdot \hat{R}) \quad (3.3)$$

Con $\hat{\Delta}$ dato da:

$$\hat{\Delta} = \begin{pmatrix} \cos(\delta) \cos(\alpha) \\ \cos(\delta) \sin(\alpha) \\ \sin(\delta) \end{pmatrix} \quad (3.4)$$

Si procede ora con l'impostazione del problema sotto forma di sistema dinamico, come richiesto dal filtro di Kalman.

3.2 Sfide sul tracciamento di angoli

A causa della natura del problema, è necessario memorizzare le quantità angolari sia nel vettore di stato che in quello di misurazione. Tuttavia, farlo senza le dovute precauzione causerebbe forti discontinuità nelle equazioni che descrivono il sistema. Si noti, ad esempio, che all'avanzare dell'anomalia, questa passa da $\nu \approx 2\pi$ a $\nu \approx 0$. In aggiunta, una stima $\hat{\nu} \approx 2\pi$ affiancata ad una misurazione $\nu \approx 0$ è in realtà da considerarsi valida, ma dal punto di vista del filtro apparirebbe come una grossa discrepanza.

Per affrontare questo problema, si decide di non memorizzare direttamente gli angoli, ma bensì di rappresentarli tramite le loro componenti sulla circonferenza unitaria, ossia la coppia $(\cos(\cdot), \sin(\cdot))$. Questo approccio è analogo alla rappresentazione di rotazioni in tre dimensioni attraverso l'uso di quaternioni e permette di rimuovere le discontinuità di cui si è parlato, poiché le coordinate sulla circonferenza unitaria sono continue e periodiche, evitando così problemi dovuti al salto tra valori di angoli equivalenti (come 2π e 0).

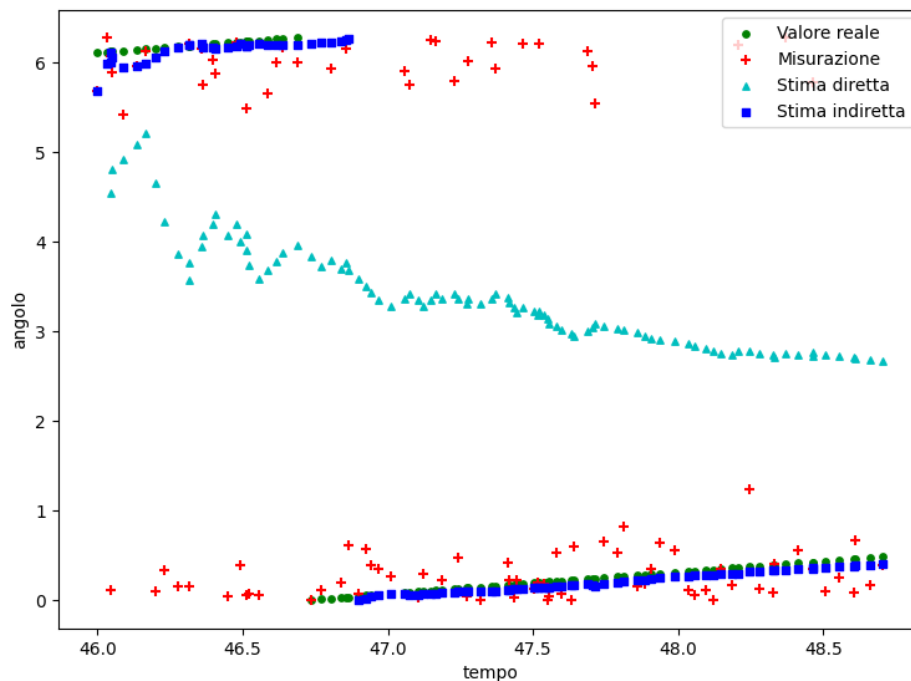


Figura 3.1: Confronto dei metodi per la stima di angoli.

In figura 3.1 si mostra l'efficacia di questa soluzione applicata alla simulazione di un corpo in moto circolare uniforme.

Tuttavia restano delle limitazioni, dovute al fatto che questa trasformazione è pressoché lineare solo per piccoli errori sugli angoli. Ne segue, ad esempio, che la media fra due coppie di componenti possa non corrispondere esattamente alla media dei due rispettivi angoli, oppure possa non risiedere precisamente sulla circonferenza unitaria.

3.3 Scelta dei vettori di stato e di misurazione

Un passo fondamentale è la scelta delle quantità da includere nel vettore di stato. I parametri orbitali a, e, i, Ω, ω sono pressoché costanti, ma averne stime accurate

nel tempo può essere vantaggioso, per cui si sceglie di tracciarli. Questa scelta permette anche di partire da stime iniziali approssimative e lasciare al filtro il compito di individuare i valori più probabili.

Il moto medio n può sempre essere calcolato a partire dal semiasse maggiore, ma un approccio che consenta piccole variazioni dovute a fattori esterni può migliorare la qualità della stima.

I parametri H e G sono costanti, ma soggetti ad incertezza, per cui si cerca di trovare stime più accurate.

La decisione più complessa riguarda la posizione del corpo. L'obiettivo finale è tracciarne la posizione cartesiana, per cui un'opzione naturale sarebbe quella di aggiungere al vettore le componenti x, y, z . Tuttavia, in questa tesi si è scelto di tracciare l'anomalia media M e, solo in un secondo momento, risalire alla posizione cartesiana. Questo approccio consente di utilizzare una legge di evoluzione molto più semplice e numericamente stabile. Sarebbe comunque consigliabile uno studio approfondito per mettere a confronto le diverse alternative.

In conclusione, esprimendo le quantità angolari nelle loro componenti, come descritto nella sezione 3.2, il vettore di stato diventa:

$$x = \begin{pmatrix} \cos(M) \\ \sin(M) \\ a \\ e \\ \cos(i) \\ \sin(i) \\ \cos(\Omega) \\ \sin(\Omega) \\ \cos(\omega) \\ \sin(\omega) \\ n \\ H \\ G \end{pmatrix} \quad (3.5)$$

La scelta del vettore di misurazione è più diretta, poiché le quantità misurate sono α , δ e V :

$$z = \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \\ \cos(\delta) \\ \sin(\delta) \\ V \end{pmatrix} \quad (3.6)$$

3.4 Equazioni del sistema

La legge di evoluzione f agisce esclusivamente sull'anomalia media, secondo la formula 2.8:

$$x_{k-1} = \begin{pmatrix} \cos(M_{k-1}) \\ \sin(M_{k-1}) \\ \dots \\ n \\ \dots \end{pmatrix} \rightarrow M_{k-1} = \arctan\left(\frac{\sin(M_{k-1})}{\cos(M_{k-1})}\right)$$

$$\rightarrow M_k = M_{k-1} + n(t_k - t_{k-1}) \rightarrow x_k = \begin{pmatrix} \cos(M_k) \\ \sin(M_k) \\ \dots \end{pmatrix}$$

Questo passaggio è rispecchiato dal seguente codice. La funzione `propagate` implementa, appunto, la legge di evoluzione f , ricevendo in input il vettore di stato al passo precedente `before_vec` o x_{k-1} sotto forma di array di numpy e l'intervallo di tempo `dt` o $t_k - t_{k-1}$. Restituisce poi il vettore di stato al passo attuale x_k , anch'esso sotto forma di array di numpy.

```

1 def propagate(before_vec: np.ndarray, dt: float) -> np.ndarray:
2     cos_mm_before, sin_mm_before, a, e, cos_i, sin_i, cos_om, sin_om, cos_w, sin_w
3     , n, hh, gg = before_vec
4     mm_before = angle_from_components(cos_mm_before, sin_mm_before)
5     mm_after = advance_mean_anomaly(mm_before, n, dt)
6     return np.array([*angle_components(mm_after), a, e, cos_i, sin_i, cos_om,
7                        sin_om, cos_w, sin_w, n, hh, gg])

```

Questo codice estrae le quantità dallo stato precedente x_{k-1} , poi usa la funzione `angle_from_components` per trovare l'anomalia media allo stato precedente M_{k-1} come l'arcotangente del quoziente `sin_mm_before / cos_mm_before`. Dunque, la funzione `advance_mean_anomaly` fa avanzare l'anomalia media da M_{k-1} a M_k se-

condo la formula citata. Il risultato viene inserito nel nuovo vettore di stato insieme alle altre quantità, le quali non sono state modificate.

La trasformazione di uscita h prevede una fase iniziale di calcolo della posizione cartesiana, del tutto analoga alla trasformazione finale delle stime in posizioni cartesiane. Questa trasformazione è implementata dalla funzione `finalize_transform` riportata di seguito, che dal vettore di stato `state_vec` o x_k , passato come array di numpy, costruisce la posizione cartesiana come un array di numpy di lunghezza 3, i cui elementi corrispondono alle coordinate xyz.

```

1 def finalize_transform(state_vec: np.ndarray) -> np.ndarray:
2     cos_mm, sin_mm, a, e, cos_i, sin_i, cos_om, sin_om, cos_w, sin_w, n, hh, gg =
        state_vec
3     mm = angle_from_components(cos_mm, sin_mm)
4     ee = eccentric_anomaly_from_mean_anomaly(mm, e)
5     v = true_anomaly_from_eccentric_anomaly(ee, e)
6     cos_v, sin_v = angle_components(v)
7     r = distance_from_eccentric_anomaly(ee, a, e)
8     return position_from_orbital_angle_components(cos_om, sin_om, cos_w, sin_w,
        cos_v, sin_v, cos_i, sin_i, r)

```

In questo codice, dall'anomalia media M o `mm` si passa all'anomalia eccentrica E o `ee` tramite la funzione `eccentric_anomaly_from_mean_anomaly`, che realizza la formula 2.10, poi dall'anomalia eccentrica E all'anomalia vera ν tramite `true_anomaly_from_eccentric_anomaly`, che realizza la formula 2.12, e alla distanza radiale r tramite `distance_from_eccentric_anomaly`, che realizza la formula 2.13. Dunque, la funzione `position_from_orbital_angle_components` si occupa di calcolare la posizione secondo la formula 2.7.

Come anticipato, questa trasformazione è utilizzata all'interno del codice che implementa h , ossia nella funzione `measure`, che riceve in input tre array di numpy: un'ipotetico vettore di stato `state_vec`, la posizione dell'osservatore `obs_pos` e la posizione del Sole `sun_pos`. Anche in questo caso si fa uso di un servizio di effemeridi per ottenere queste posizioni. Viene restituito il vettore di misurazione corrispondente come un array di numpy di lunghezza 5 contenente coseno e seno dell'ascensione retta α e della declinazione δ e la magnitudine visuale V .

```

1 def measure(state_vec: np.ndarray, obs_pos: np.ndarray,
2     sun_pos: np.ndarray) -> np.ndarray:

```

```

3     tgt_pos = finalize_transform(state_vec)
4     hh, gg = state_vec[11], state_vec[12]
5     tgt_obs_pos = tgt_pos - obs_pos
6     tgt_obs_dist = norm(tgt_obs_pos)
7     tgt_obs_ray = tgt_obs_pos / tgt_obs_dist
8     cos_ra, sin_ra, cos_dec, sin_dec = direction_to_ra_dec_components(tgt_obs_ray)
9     tgt_sun_dist = norm(tgt_pos - sun_pos)
10    obs_sun_dist = norm(obs_pos - sun_pos)
11    phi = phase_from_distances(tgt_sun_dist, tgt_obs_dist, obs_sun_dist)
12    vv = visual_magnitude_from_absolute(hh, tgt_sun_dist, tgt_obs_dist, phi, gg)
13    return np.array([cos_ra, sin_ra, cos_dec, sin_dec, vv])

```

In questo codice, in seguito alla chiamata a `finalize_transform`, si passa dalla posizione baricentrica `tgt_pos` alla posizione topocentrica `tgt_obs_pos` e si determinano i valori osservati. Per fare ciò, si calcolano innanzitutto `tgt_obs_ray` o $\hat{\Delta}$ come vettore normalizzato della posizione topocentrica e `tgt_obs_dist` o Δ come il suo modulo. Si trovano, poi, le componenti di ascensione retta e declinazione, sfruttando la funzione `direction_to_ra_dec_components`, che applica l'inverso della formula 3.4:

$$\sin(\delta) = z \quad (3.7)$$

$$\cos(\delta) = \sqrt{1 - \sin^2(\delta)} \quad (3.8)$$

$$\cos(\alpha) = \frac{x}{\cos(\delta)} \quad (3.9)$$

$$\sin(\alpha) = \frac{y}{\cos(\delta)} \quad (3.10)$$

Per i casi speciali $\delta = \pm\pi$, si pone arbitrariamente $\alpha = 0$. Infine, si calcolano le altre distanze `tgt_sun_dist` o d e `obs_sun_dist` o R , l'angolo di fase con la funzione `phase_from_distances`, che applica il risultato del teorema dei seni nella formula 2.4, e da esso la magnitudine visuale V grazie alla funzione `visual_magnitude_from_absolute`, che implementa la formula 2.5.

3.5 Matrici di covarianza

La precisione del filtro di Kalman dipende fortemente dall'accuratezza delle matrici di covarianza Q e R utilizzate. Ovviamente non se ne conoscono i valori reali, ma analisi sia formali che statistiche possono fornire stime pressoché ottimali.

3.5. MATRICI DI COVARIANZA

Per modelli semplici, Q ed R sono spesso matrici diagonali, ma non è questo il caso per il sistema considerato. In effetti, se si considera una generica quantità angolare θ con distribuzione di von Mises avente valore medio μ e parametro di concentrazione κ , approssimata da una distribuzione normale lungo la retta tangente avente varianza $\sigma^2 = \frac{1}{\kappa}$, si trova che la coppia $(\cos(\theta), \sin(\theta))$ ha la seguente matrice di auto-covarianza:

$$\begin{pmatrix} \sigma^2 \sin^2(\mu) & -\sigma^2 \sin(\mu) \cos(\mu) \\ -\sigma^2 \sin(\mu) \cos(\mu) & \sigma^2 \cos^2(\mu) \end{pmatrix} \quad (3.11)$$

Dunque, in prima approssimazione, Q è data da (vengono mostrate separatamente le prime 6 righe poi le restanti 7 per esigenze di impaginazione):

$$Q = \begin{pmatrix} \sigma_M^2 \sin^2(M) & -\sigma_M^2 \sin(M) \cos(M) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\sigma_M^2 \sin(M) \cos(M) & \sigma_M^2 \cos^2(M) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_a^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_e^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_i^2 \sin^2(i) & -\sigma_i^2 \sin(i) \cos(i) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\sigma_i^2 \sin(i) \cos(i) & \sigma_i^2 \cos^2(i) & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \quad (3.12)$$

$$Q = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \sigma_\Omega^2 \sin^2(\Omega) & -\sigma_\Omega^2 \sin(\Omega) \cos(\Omega) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\sigma_\Omega^2 \sin(\Omega) \cos(\Omega) & \sigma_\Omega^2 \cos^2(\Omega) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\omega^2 \sin^2(\omega) & -\sigma_\omega^2 \sin(\omega) \cos(\omega) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\sigma_\omega^2 \sin(\omega) \cos(\omega) & \sigma_\omega^2 \cos^2(\omega) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_n^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_H^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_G^2 \end{pmatrix} \quad (3.13)$$

La matrice R richiede ulteriori passaggi. Se si considera il versore di osservazione $\hat{\Delta}$ con distribuzione di Fisher avente parametro di concentrazione $\kappa_{\alpha,\delta}$, allora gli angoli α e δ sono approssimati da distribuzioni di von Mises con parametri di concentrazione $\kappa_{\alpha,\delta} \cos^2(\delta)$ e $\kappa_{\alpha,\delta}$ rispettivamente. Dunque, in prima approssimazione, R è data da:

$$R = \begin{pmatrix} \sigma_{\alpha,\delta}^2 \frac{\sin^2(\alpha)}{\cos^2(\delta)} & -\sigma_{\alpha,\delta}^2 \frac{\sin(\alpha) \cos(\alpha)}{\cos^2(\delta)} & 0 & 0 & 0 \\ -\sigma_{\alpha,\delta}^2 \frac{\sin(\alpha) \cos(\alpha)}{\cos^2(\delta)} & \sigma_{\alpha,\delta}^2 \frac{\cos^2(\alpha)}{\cos^2(\delta)} & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\alpha,\delta}^2 \sin^2(\delta) & -\sigma_{\alpha,\delta}^2 \sin(\delta) \cos(\delta) & 0 \\ 0 & 0 & -\sigma_{\alpha,\delta}^2 \sin(\delta) \cos(\delta) & \sigma_{\alpha,\delta}^2 \cos^2(\delta) & 0 \\ 0 & 0 & 0 & 0 & \sigma_V^2 \end{pmatrix} \quad (3.14)$$

3.6 Simulazione

Inizialmente si è scelto di verificare l'efficacia del filtro utilizzando dati simulati. In primo luogo, il file `simulation/ideal_model_test.py` si occupa di simulare il moto orbitale, sia del corpo osservato, che del Sole e dell'osservatore, attraverso la seguente funzione `real_position`, che si avvale delle formule descritte nella sezione 2.3 in modo simile alla funzione `finalize_transform`. Prende in input 8 parametri: t (tempo in cui calcolare la posizione), a (semiasse maggiore), e (eccentricità), i (inclinazione), Ω o `om` (longitudine del nodo ascendente), ω o `w` (argomento del periapside), n (moto medio), τ o `tp` (tempo di periapside). Anche in questo caso, la posizione cartesiana restituita è rappresentata come un array di numpy di lunghezza 3.

```
1 def real_position(t: float, a: float, e: float, i: float, om: float, w: float,
2                   n: float, tp: float) -> np.ndarray:
3     mm = wrap_angle(n * (t - tp))
4     ee = eccentric_anomaly_from_mean_anomaly(mm, e)
5     v = true_anomaly_from_eccentric_anomaly(ee, e)
6     r = distance_from_eccentric_anomaly(ee, a, e)
7     return position_from_orbital_angles(om, w + v, i, r)
```

L'unica differenza in questo codice rispetto alla funzione `finalize_transform` sta nel calcolo dell'anomalia media, che in questo caso sfrutta direttamente la formula 2.8, assicurandosi di mantenere l'angolo nell'intervallo $[0, 2\pi)$.

Per garantire dati realistici, sono stati usati parametri orbitali reali, seppur successivamente considerati costanti, ottenuti dalla piattaforma JPL Horizons, in particolare accurati all'epoca J2000.0 tempo dinamico baricentrico. Per il corpo osservato si è considerato Bennu (1999 RQ36) e per l'osservatore si è considerato il baricentro Terra-Luna, garantendo sufficiente diversificazione nei dati.

Dalle posizioni dei tre soggetti, si determinano, poi, i valori osservati, come nella funzione `measure`, ma applicando disturbi di misurazione. Gli angoli α, δ sono ottenuti dalla direzione $\hat{\Delta}$, dopo che essa è stata perturbata secondo una distribuzione di Fisher (`scipy.stats.vonmises_fisher`). La magnitudine visuale V è ricavata dall'angolo di fase, a sua volta ricavato trigonometricamente, e perturbata secondo una distribuzione normale (`numpy.random.normal`).

Questo è implementato dalla funzione `observe`, che riceve in input le tre posizioni come array di numpy, i parametri H e G che caratterizzano l'asteroide, le varianze sulla direzione $\sigma_{\alpha,\delta}^2$ e sulla magnitudine visuale σ_V^2 e il generatore di valori casuali da utilizzare. Restituisce quindi i valori di ascensione retta, declinazione e magnitudine visuale, come sarebbero osservati nella realtà.

```

1 def observe(tgt_pos: np.ndarray, obs_pos: np.ndarray, sun_pos: np.ndarray,
2             hh: float, gg: float, dir_var: float, vv_var: float,
3             rng: np.random.Generator) -> tuple[float, float, float]:
4     tgt_obs_pos = tgt_pos - obs_pos
5     tgt_obs_dist = norm(tgt_obs_pos)
6     tgt_obs_ray = tgt_obs_pos / tgt_obs_dist
7     tgt_obs_ray = sp.stats.vonmises_fisher(mu=tgt_obs_ray, kappa=(1.0 / dir_var),
8     seed=rng).rvs(1)[0]
9     ra, dec = direction_to_ra_dec(tgt_obs_ray)
10    tgt_sun_dist = norm(tgt_pos - sun_pos)
11    obs_sun_dist = norm(obs_pos - sun_pos)
12    phi = phase_from_distances(tgt_sun_dist, tgt_obs_dist, obs_sun_dist)
13    vv = visual_magnitude_from_absolute(hh, tgt_sun_dist, tgt_obs_dist, phi, gg)
14    vv = rng.normal(loc=vv, scale=np.sqrt(vv_var))
15    return ra, dec, vv

```

La differenza dal codice della funzione `measure` sta nell'uso delle funzioni di statistica. Alla funzione `vonmises_fisher` viene passato il parametro di concentrazione $\kappa_{\alpha,\delta} = (\sigma_{\alpha,\delta}^2)^{-1}$, mentre alla funzione `normal` viene passata la deviazione standard $\sigma_V = \sqrt{\sigma_V^2}$.

Ovviamente questa simulazione non modella la velocità finita della luce.

Una volta ottenute le osservazioni, separate da un lasso di tempo variabile, si crea un'istanza UKF con $\alpha = 10^{-3}$, $\beta = 2$, $\kappa = 0$ e vengono fornite la legge di evoluzione f e la trasformazione di uscita h ricavate nella sezione 3.4 e le matrici di covarianza R e Q della sezione 3.5.

```

1 points = MerweScaledSigmaPoints(13, alpha=1E-3, beta=2, kappa=0)
2 ukf = UnscentedKalmanFilter(dim_x=13, dim_z=5, dt=None, hx=measure, fx=
3     propagate, points=points)
4
5 x0 = initial_state(prepared[0], obs_pos[0], sun_pos[0], a0, e0, i0, n0, hh0,
6     gg0, mm0_hint, om0_hint)
7 pp0 = state_autocovariance_matrix(x0, 1E-3, 1E-3, 1E-3, 1E-3, 1E-3, 1E-3, 1E-3,
8     1E-3, 1E-3, 1E-3)
9 prev_t = times[0]
10 ukf.x = x0
11 ukf.P = pp0

```

```
9
10     estimates = [finalize_transform(x0)]
11     for t, obs, o, s in zip(times[1:], prepared[1:], obs_pos[1:], sun_pos[1:]):
12         dt = t - prev_t
13         ukf.Q = state_autocovariance_matrix(ukf.x, 1E-8, 1E-8, 1E-8, 1E-8, 1E-8, 1
14             E-8, 1E-8, 1E-8, 1E-8)
15         ukf.predict(dt=dt)
16         ukf.R = measurement_autocovariance_matrix(obs, dir_var, vv_var)
17         ukf.update(z=obs, obs_pos=o, sun_pos=s)
18         estimates.append(finalize_transform(ukf.x))
19         prev_t = t
```

Qui vengono inizializzati \hat{x}_0 e P_0 , dove \hat{x}_0 viene ricavato dalla prima osservazione simulata. Durante l'esecuzione si tiene traccia del tempo al passo precedente così da poter calcolare gli intervalli di tempo tra le osservazioni. Inoltre si mantiene una lista `estimates` delle stime prodotte dal filtro e trasformate in posizioni cartesiane. Ad ogni passo viene impostata la matrice Q , per poi chiamare il metodo di libreria `ukf.predict` che effettua la fase di predizione, per poi impostare la matrice R e chiamare `ukf.update` per la fase di correzione.

Le stime sono poi confrontate con le posizioni ottenute dall'approccio diretto discusso nella sezione 3.1. In figura 3.2 si riportano i risultati sui primi 20 passi di simulazione, poi sui primi 150, con una rappresentazione 3D delle posizioni e un grafico di andamento dell'errore, calcolato come $\left\| \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} - \begin{pmatrix} x \\ y \\ z \end{pmatrix} \right\|$. Si evidenzia una graduale convergenza.

3.6. SIMULAZIONE

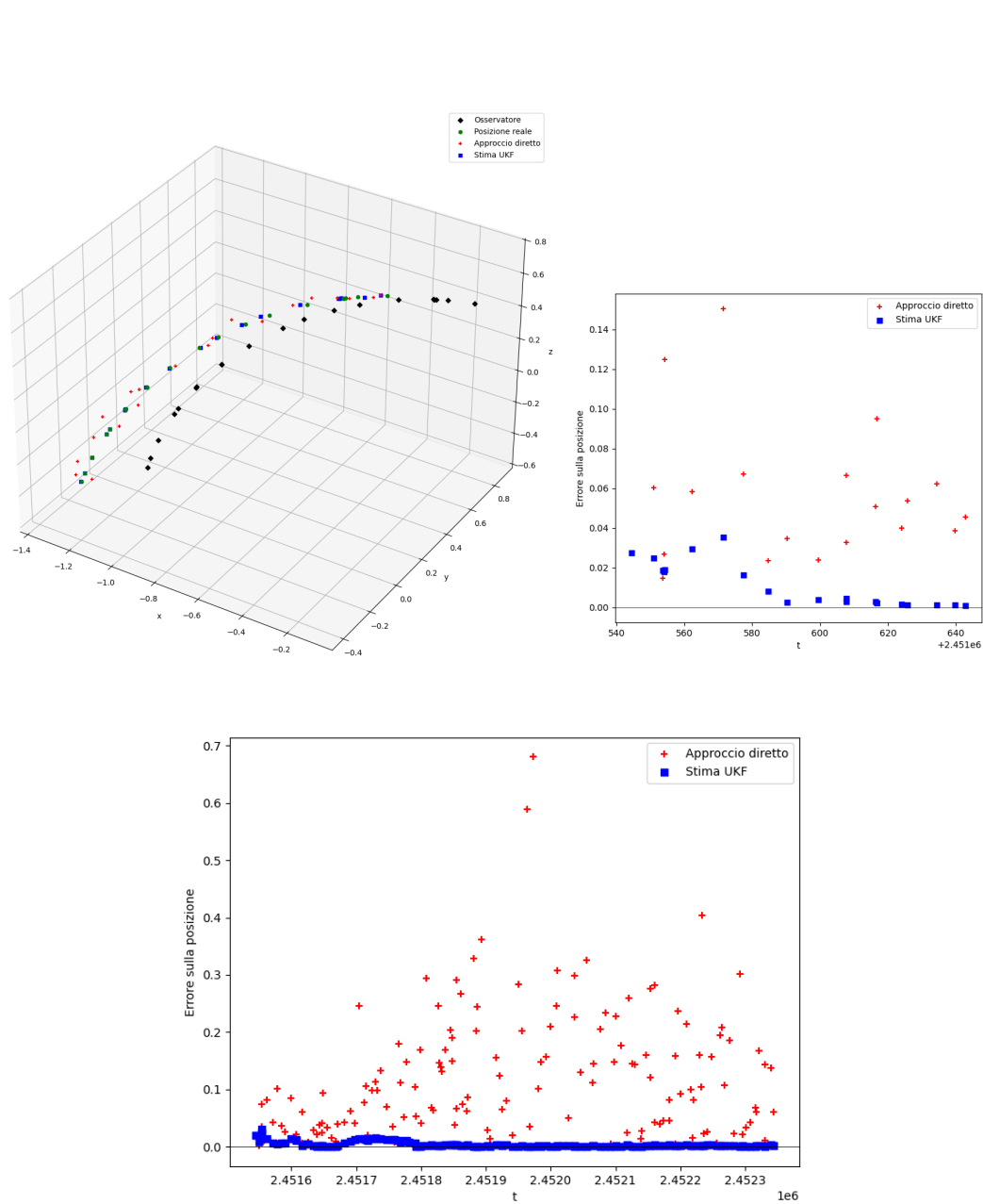


Figura 3.2: Visualizzazione delle stime sui dati simulati (primi 20 passi, sopra, e primi 150 passi, sotto), confrontate con l'approccio diretto.

3.7 Applicazione a dati reali

Il Minor Planet Center raccoglie dati su osservazioni ottiche astrometriche di corpi minori, comete e satelliti naturali, effettuate da osservatori in tutto il mondo. I dati sono forniti senza alcun tipo di correzione, come specificato nelle linee guida (<https://minorplanetcenter.net/iau/info/Astrometry.html#corrs>).

Moduli Python esistenti come `astroquery` consentono di richiedere automaticamente le osservazioni; tuttavia si possono scaricare archivi più completi direttamente dalla piattaforma web https://www.minorplanetcenter.net/db_search.

Questi archivi sono file di testo in classico stile Fortran a 80 colonne e seguono il formato descritto alla pagina <https://www.minorplanetcenter.net/iau/info/OpticalObs.html>. In particolare, contengono le seguenti informazioni di nostro interesse:

- l'ascensione retta, espressa come angolo orario, minuti e secondi
- la declinazione, espressa in gradi, minuti e secondi
- la magnitudine osservata
- la banda utilizzata, ossia un'indicazione della fascia di frequenze elettromagnetiche considerate
- il codice dell'osservatorio

Il file `parse.py` si occupa dell'interpretazione dei dati e della trasformazione in oggetti della classe di utility `MinorPlanetObservation`. Questo passaggio richiede la conversione degli angoli in radianti e della magnitudine osservata in magnitudine visuale. In effetti, i modelli fotometrici si basano sulla magnitudine relativa alla luce visibile, per cui può essere necessario utilizzare la tabella di conversione fornita alla pagina <https://www.minorplanetcenter.net/iau/info/BandConversion.txt>.

Infine, il codice dell'osservatorio viene confrontato con la lista alla pagina <https://www.minorplanetcenter.net/iau/lists/ObsCodes.html>, per risalire alle coordinate geodetiche dell'osservatore. Anche in questo caso, un servizio di effemeridi può essere utilizzato per risalire alla posizione baricentrica dell'osservatore in un determinato istante date le coordinate geodetiche.

```

1 OBSERVATION_FORMAT: Final[str] = '(A5, A7, A1, A1, A1, I4, 1X, I2, 1X, F9.6, I2, 1
  X, I2, 1X, F6.3, I3, 1X, I2, 1X, F5.2, 9X, F5.2, A1, 6X, A3)'
2 observation_reader = FortranRecordReader(OBSERVATION_FORMAT)
3
4 def parse_observation_line(body: MinorPlanet, line: str) -> MinorPlanetObservation
  :
5     _packed_num, _packed_desig, _discov_ast, _note1, _note2, year, month,
        day_decimal, ra_hour, ra_minute, ra_second, dec_degree, dec_minute,
        dec_second, mag, band, observatory_code = observation_reader.read(line)
6     observatory = observatories[observatory_code]
7     epoch = decimal_day_date_to_time(year, month, day_decimal, observatory.
        location)
8     return MinorPlanetObservation.with_band(target_body=body, epoch=epoch,
        observatory=observatory, ra=np.radians(15 * ra_hour + ra_minute / 4 +
        ra_second / 240), dec=np.radians(dec_degree + dec_minute / 60 + dec_second
        / 3600), observed_magnitude=mag, band=band)

```

Qui viene utilizzata il metodo `read` della classe `FortranRecordReader`, per estrarre i dati dalle righe che rispettano il formato fornito. L'osservatorio viene ritrovato nell'elenco caricato da file e la data viene convertita in un oggetto `astropy.time.Time`. Infine vengono effettuate le conversioni per istanziare l'oggetto contenitore.

Raccolte le osservazioni, si trasformano gli oggetti `MinorPlanetObservation` in vettori di misurazione e si procede, come per i dati simulati, con un'istanza UKF con $\alpha = 10^{-3}$, $\beta = 2$, $\kappa = 0$. In figura 3.3 si riportano i risultati sulle prime 130 osservazioni dell'asteroide Bennu (1999 RQ36). Anche in questo caso si osserva una generale convergenza, seppur decisamente più lenta rispetto alla simulazione. Inoltre, si è osservato che il processo non risulta perfettamente stabile; su altre porzioni di dati, lievi discostamenti improvvisi portano a rapida divergenza. Tutto ciò potrebbe essere dovuto a diversi motivi, possibilmente legati a problemi di stabilità numerica, per cui studi ulteriori sono necessari. In ogni caso, si è mostrato come il filtro permetta di ottenere risultati dalle 10 alle 100 volte più accurati rispetto all'approccio diretto, nonostante le diverse semplificazioni che sono state fatte, dando prova concreta dell'utilità del filtro di Kalman in questo contesto reale.

Nel complesso, il processo è stato sperimentato su 4 dataset, relativi a 4 asteroidi classificati come Potentially Hazardous Asteroids (PHA) dal Minor Planet Center, seppure il rischio di effettiva collisione con la Terra sia stimato at-

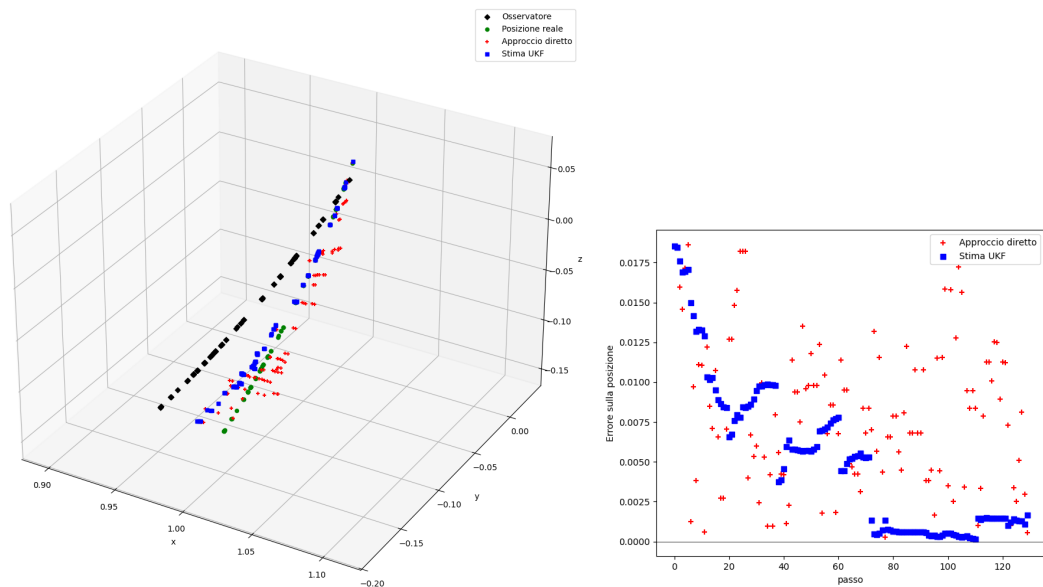


Figura 3.3: Visualizzazione delle stime sui dati reali dell'asteroide Bennu (prime 130 osservazioni), confrontate con l'approccio diretto.

tualmente come estremamente ridotto. L'elenco completo dei PHA riconosciuti dal MPC è disponibile alla pagina <https://www.minorplanetcenter.net/iau/Dangerous.html>. Nello specifico, per i test eseguiti in questa tesi sono stati presi in considerazione per i dataset corrispondenti ai seguenti asteroidi:

- Bennu: un asteroide con orbita di tipo Apollo (semiasse maggiore $a > 1$ AU) e un diametro medio di circa 490 metri. Oggi si hanno molte informazioni sulle caratteristiche fisiche e sull'orbita dell'asteroide, principalmente grazie alla missione OSIRIS-REx della NASA, che lo ha sorvolato e studiato con grande precisione. Questi dati includono rilevamenti topografici tramite lidar e fotografie ad alta risoluzione, utili per creare modelli 3D dettagliati della forma e della superficie. Sono disponibili informazioni su caratteristiche geologiche, composizione superficiale, velocità di rotazione, e dati termici. Per questo studio sono state considerate solo le osservazioni terrestri, a partire dalla scoperta nel 1999, che potranno essere poi confrontate con le misurazioni più precise per ottenere modelli completi. Ad oggi si prevede una

3.7. APPLICAZIONE A DATI REALI

serie di 157 possibili impatti tra l'anno 2178 e il 2290, con una probabilità cumulativa intorno allo 0.05% e un valore cumulativo di -1.40 sulla scala Palermo.

- Mjolnir: un piccolo asteroide con orbita di tipo Apollo che incrocia quella terrestre e classificato come potenzialmente pericoloso (PHA) dalla NASA, seppure non siano attualmente previsti possibili impatti. Mjolnir ha dimensioni stimate fra 82 e 366 metri, con un periodo di rotazione di 11.6 ore. Il dataset di Mjolnir comprende dati orbitali e fisici raccolti da osservazioni radar e ottiche terrestri. Le stime precise dell'orbita permettono di monitorare la sua traiettoria e di prevedere eventuali prossimità alla Terra. Mjolnir è studiato anche per la sua rotazione e per caratteristiche fisiche come la riflettività superficiale e la struttura. Tali dati sono utilizzati in modelli di monitoraggio di eventualmente comportamenti dinamici complessi dovuti a perturbazioni gravitazionali o effetti Yarkovsky (forza causata dall'irraggiamento termico). Questo dataset è meno ricco di immagini ad alta risoluzione rispetto a Bennu, ma essenziale per la sorveglianza e valutazione del rischio.

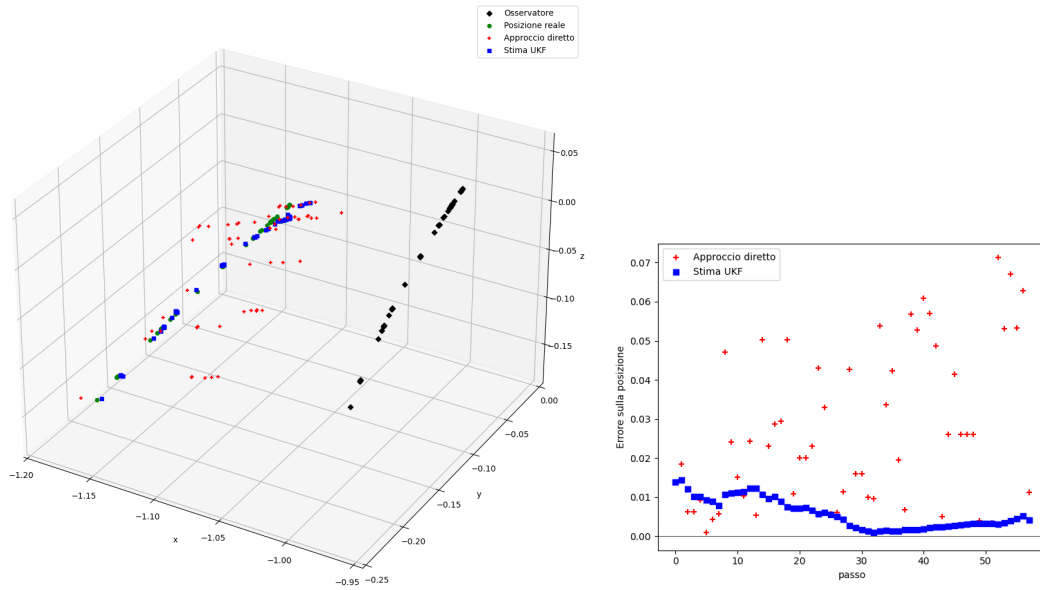


Figura 3.4: Visualizzazione delle stime sui dati reali dell'asteroide Mjolnir (prime 60 osservazioni), confrontate con l'approccio diretto.

- 1950 DA: un asteroide di circa 1.1–1.3 km di diametro appartenente al gruppo Apollo e collocato in cima alle liste dei PHA. Si stima, infatti, un possibile impatto nell'anno 2880, con una probabilità intorno allo 0.04% e un valore di -0.92 sulla scala Palermo, superiore a quello di Bennu per via delle dimensioni maggiori. Il dataset di 1950 DA include misure ottenute da radar planetari e telescopi che ne permettono la ricostruzione orbitale dettagliata e la caratterizzazione fisica. Significativo è il fatto che 1950 DA ruoti molto velocemente, indicando una possibile coesione interna elevata o struttura particolare. Viene osservato sia per i possibili impatti futuri con la Terra, che per analisi che considerano gli effetti gravitazionali e della radiazione solare sul suo moto. I dati comprendono parametri orbitali, estimazioni di densità e massa, e modelli di forma approssimati, usati in simulazioni su lunga scala temporale.

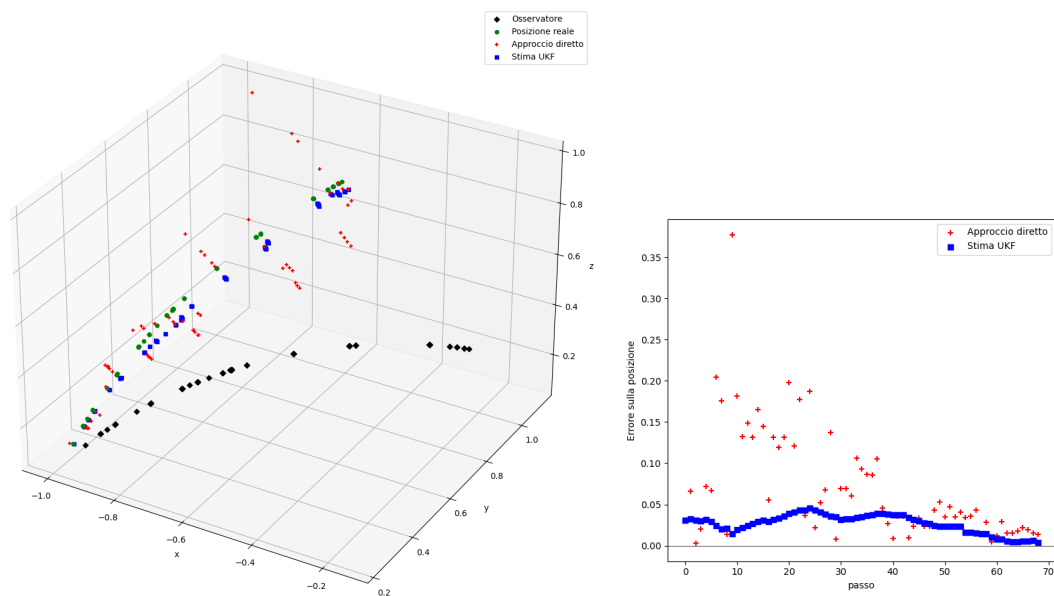


Figura 3.5: Visualizzazione delle stime sui dati reali dell'asteroide 1950 DA (prime 70 osservazioni), confrontate con l'approccio diretto.

- Castalia: un asteroide di circa 1.4 km appartenente al gruppo Apollo, primo a essere modellato tramite immagini radar. Ha una forma a “nociola” o binaria a contatto, formata da due componenti unite dalla gravità mutua.

3.7. APPLICAZIONE A DATI REALI

Questa forma inusuale rende più difficile utilizzare i modelli tradizionali per rappresentare la relazione tra l'angolo di fase e la magnitudine. Si tratta di un asteroide potenzialmente pericoloso con orbita ben caratterizzata e osservato ampiamente tramite radar e fotometria. Il dataset include modelli tridimensionali basati su dati radar, informazioni sulla composizione, e proprietà dinamiche della coppia di corpi connessi gravitazionalmente (binario a contatto). Questi dati aiutano a capire la formazione e l'evoluzione degli asteroidi binari, la loro rotazione sincronizzata e l'interazione gravitazionale interna. Inoltre, sono disponibili dati osservativi dettagliati che supportano simulazioni numeriche e studi di sorveglianza orbitale per la valutazione del rischio associato.

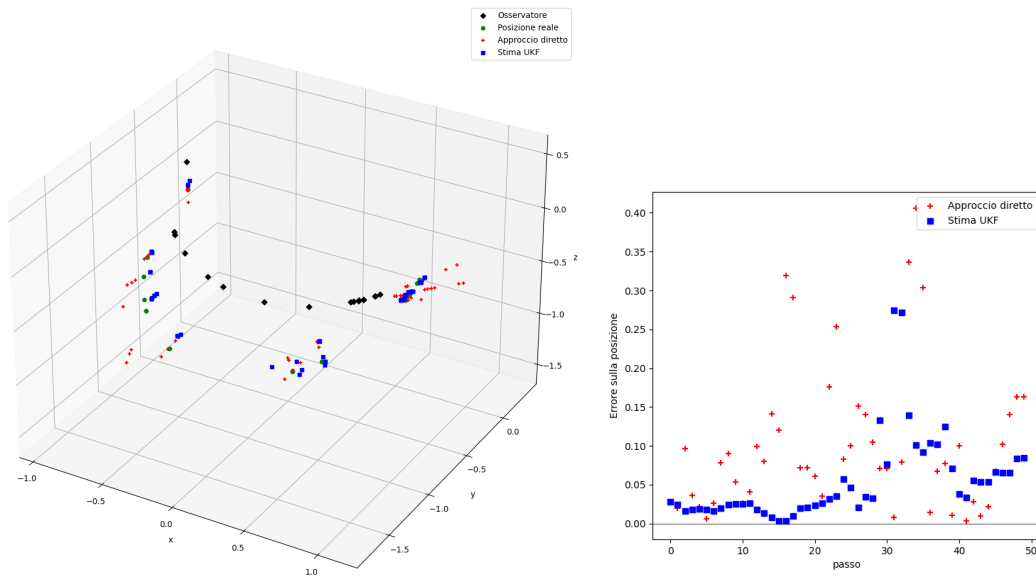


Figura 3.6: Visualizzazione delle stime sui dati reali dell'asteroide Castalia (prime 50 osservazioni), confrontate con l'approccio diretto. Si osservano cenni di divergenza.

Per meglio interpretare i risultati ottenuti, si effettua un'analisi statistica sugli errori. Si tenta di caratterizzare la distribuzione degli errori attraverso un riepilogo a cinque numeri, ossia l'insieme del valore minimo, del primo quartile (25% degli errori saranno minori di esso), della mediana (50%), del terzo quartile (75%) e del valore massimo. Nella tabella 3.1 si riporta l'analisi sui test effettuati, espressi in unità astronomiche. Si tratta di una rappresentazione limitata, in quanto non si evince il miglioramento delle stime nel tempo, ma permette di osservare concretamente l'efficacia del filtraggio.

	Minimo	Q1 (25%)	Mediana	Q2 (75%)	Massimo
Bennu (diretto)	2.60e-4	4.30e-3	7.92e-3	1.13e-2	1.86e-2
Bennu (UKF)	1.61e-4	6.32e-4	5.19e-3	8.18e-3	1.85e-2
Mjolnir (diretto)	8.92e-4	9.87e-3	2.30e-2	4.29e-2	7.13e-2
Mjolnir (UKF)	9.84e-4	2.50e-3	4.45e-3	9.12e-3	1.45e-2
1950 DA (diretto)	2.90e-3	2.32e-2	4.32e-2	1.06e-1	3.77e-1
1950 DA (UKF)	3.98e-3	2.01e-2	2.96e-2	3.53e-2	4.57e-2
Castalia (diretto)	3.89e-3	3.18e-2	7.81e-2	1.30e-1	4.06e-1
Castalia (UKF)	3.44e-3	2.04e-2	3.30e-2	6.87e-2	2.75e-1

Tabella 3.1: Caratterizzazione degli errori sulle posizioni attraverso un riepilogo a cinque numeri.

In generale, l'approccio UKF porta a valori tipicamente inferiori delle statistiche di errore rispetto al metodo diretto per quasi tutti gli asteroidi, suggerendo una migliore efficacia nel ridurre l'errore di posizione.

Per quanto riguarda Bennu, UKF migliora sia il minimo che la mediana e mantiene il massimo molto simile al diretto. Questo è facilmente spiegato dal fatto che lo stato iniziale è particolarmente impreciso.

Relativamente a Mjolnir, la mediana dell'errore con UKF è circa la metà del diretto e anche il massimo diminuisce notevolmente.

In riferimento a 1950 DA, si nota una netta riduzione del massimo e dei quartili per UKF rispetto al diretto, evidenziando la sua efficacia anche in condizioni di errore elevato.

Infine per Castalia, l'UKF riduce di molto sia la mediana che il massimo rispetto al diretto, specialmente nelle code della distribuzione degli errori.

L'analisi quantitativa evidenzia che UKF è sistematicamente vantaggioso rispetto all'approccio diretto, offrendo una significativa riduzione degli errori di posizione per tutti gli oggetti analizzati, sia nei valori mediani che nei valori estremi.

Conclusioni

Il progetto ha consentito di sviluppare una conoscenza approfondita delle varianti più importanti del filtro, sia dal punto di vista del loro utilizzo, sia con riferimento ai passaggi intermedi di cui si occupano le implementazioni e che normalmente sono “nascosti” all’utente. Questo ha permesso di rifinire le funzioni e i parametri forniti per ottenere risultati soddisfacenti.

Il processo di simulazione ha contribuito a una migliore comprensione del modello astronomico, così da poter superare le sfide nell’applicazione ai dati reali. È stata anche un’occasione per ricercare le distribuzioni di von Mises e di Fisher, fondamentali nella statistica direzionale. Allo stesso modo, l’utilizzo del metodo di Brent per il calcolo degli zeri ha ampliato il catalogo costituito durante gli studi. In generale, si è fatto ampio uso delle librerie `numpy` e `scipy`, importanti per la computazione efficiente e l’analisi dei dati.

In termini di risultati, si è confermata la possibilità di tracciare il movimento di un corpo celeste in modo accurato, con stime dalle 10 alle 100 volte più precise rispetto al solo utilizzo delle osservazioni.

Possibili approfondimenti

- Mettere a confronto le varie possibilità riguardo i vettori di stato e di misurazione, ad esempio considerare le componenti cartesiane x, y, z piuttosto che l’anomalia media.
- Trovare approssimazioni migliori delle matrici di covarianza Q ed R .
- Esaminare attentamente le caratteristiche delle porzioni di dati che portano alla divergenza.

- Espandere il modello del sistema per considerare la velocità finita della luce e altri fenomeni connessi.

Bibliografia

- [1] R. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, pp. 35–45, Mar. 1960.
- [2] G. Welch and G. Bishop, “An introduction to the kalman filter,” Tech. Rep. 95-041, University of North Carolina at Chapel Hill, 1995.
- [3] Y. Pei, S. Biswas, D. Fussell, and K. Pingali, “An elementary introduction to kalman filtering,” *Communications of the ACM*, vol. 62, p. 122–133, Oct. 2019.
- [4] A. Gelb and T. A. S. Corporation, *Applied Optimal Estimation*. The MIT Press, 1974.
- [5] H. Masnadi-Shirazi, A. Masnadi-Shirazi, and M. Dastgheib, “A step by step mathematical derivation and tutorial on kalman filters.” ArXiv Preprint ArXiv:1910.03558, Oct. 2019.
- [6] G. Smith, S. Schmidt, and L. McGee, “Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle,” Tech. Rep. R-135, National Aeronautics and Space Administration, 1962.
- [7] E. Wan and R. Van Der Merwe, “The unscented kalman filter for nonlinear estimation,” in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, 2000.
- [8] A. Yang and S. Boyd, “Iteratively saturated kalman filtering,” 2025.

- [9] N. Shlezinger, G. Revach, A. Ghosh, S. Chatterjee, S. Tang, T. Imbiriba, J. Dunik, O. Straka, P. Closas, and Y. C. Eldar, “Ai-aided kalman filters,” 2025.
- [10] E. Bowell, B. Hapke, D. Domingue, K. Lumme, J. Peltoniemi, and A. Harris, “Application of photometric models to asteroids,” in *Asteroids II*, pp. 524–556, The University of Arizona Press, 1989.
- [11] R. Dymock, “The H and G magnitude system for asteroids,” *Journal of the British Astronomical Association*, vol. 117, pp. 342–343, Nov. 2007.

Ringraziamenti

Questa ricerca si è avvalsa di dati e servizi forniti dal Minor Planet Center dell'Unione Astronomica Internazionale, nonché dal Jet Propulsion Laboratory del California Institute of Technology.

Il software è stato sviluppato utilizzando le librerie Python `fortranformat`, `matplotlib`, `numpy`, `scipy`, `scipy-stubs`, `astropy`, `jplephem`, `certifi`, `requests`, `filterpy`.