**Filtro di Kalman non lineare**

Dato che le equazioni dinamiche del sistema ed il modello di osservazione sono non lineari, facciamo appello al **filtro di Kalman Esteso** ed al **UKF** (Unscented Kalman Filter). Per quanto riguarda il modello di osservazione (la funzione h) facciamo riferimento alle equazioni:

Per quanto riguarda invece le equazioni dinamiche (la funzione f), la struttura è quella della dinamica a tempo discreto, ove la derivata dello stato è calcolata a partire alle equazioni meccaniche del sistema, in termini di accelerazione ed accelerazione angolare:

Da cui ricaviamo le equazioni dinamiche tempo discreto del sistema:

dove è definito come:

**Filtro di Kalman Esteso (EKF)**

* Immagine che contiene testo, diagramma, Piano, linea

  Descrizione generata automaticamente Come possiamo vedere dalla *Figura 3.1,* il filtro EKF prevede in ingresso le misure dei sensori (con disturbo additivo di tipo Gaussiano) e l’ingresso F anch’esso disturbato.
* Abbiamo aggiunto 2 integratori di tipo discreto così che da poter avere la stima dello stato e la sua varianza all’istante precedente (cioè e ).
* Il tempo di campionamento del filtro è

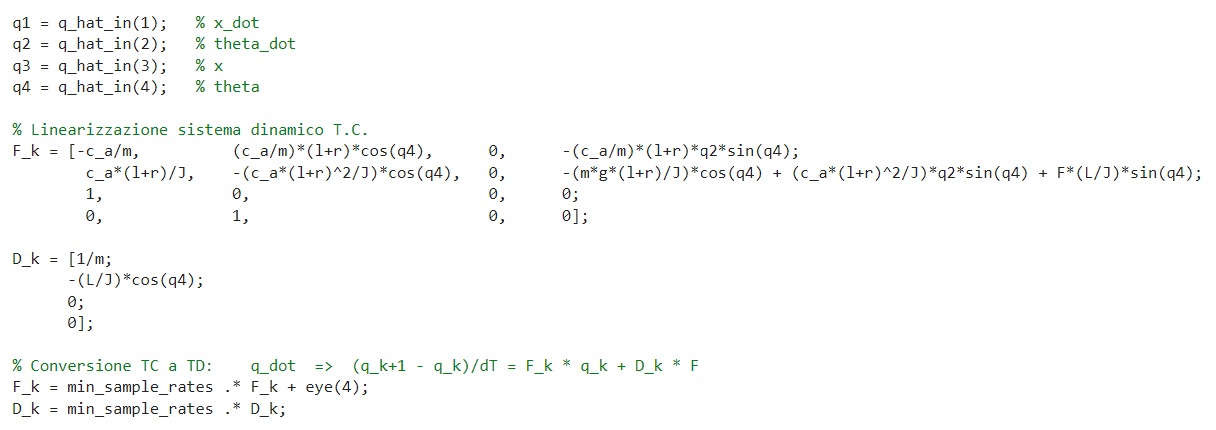
dato dal tempo di campionamento più grande tra i sensori scelti (in questo caso abbiamo che il tempo ).

**Figura 3.1** – Realizzazione EKF tramite Simulink

Infine, l’uscita del filtro è semplicemente la stima dello stato del sistema, cioè , e dove è stato preso come:

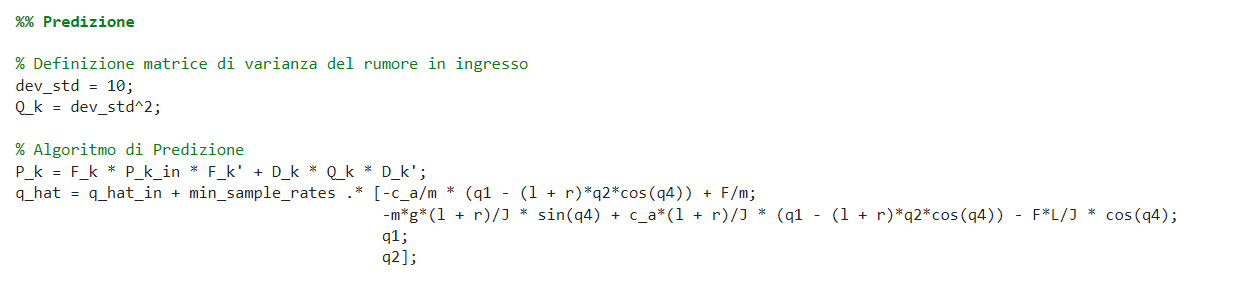
**Predizione EKF**

Ora andiamo a vedere la linearizzazione della dinamica del sistema su cui si basa il filtro EKF; di seguito è riportata la funzione MatLab interna al blocco Simulink della figura *Figura 3.1*:



**Figura 3.2** – Codice per la Linearizzazione della dinamica del sistema (calcolo matrici e )

Da cui segue la parte di predizione del filtro:



**Figura 3.3** – Codice per l’algoritmo di predizione del filtro EKF

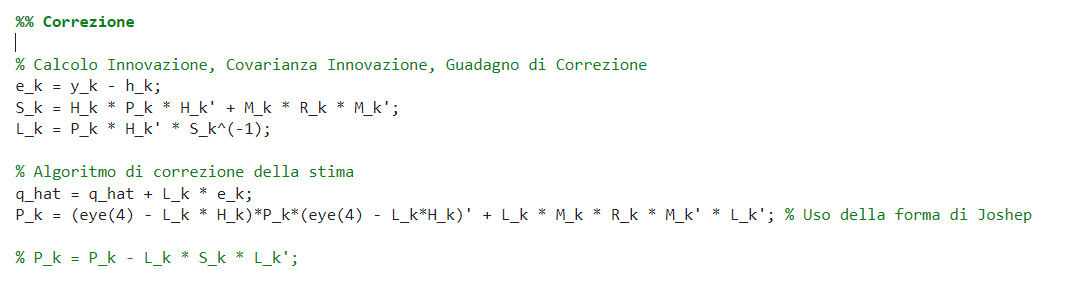
**Correzione EKF**

Ora andiamo a vedere la parte di codice di correzione del filtro, che si trova nello stesso blocco di *Figura 3.1*, dove ci aspettiamo una linearizzazione della dinamica di uscita (sensori) del sistema: Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

**Figura 3.4** – Codice per la Linearizzazione della dinamica di uscita del sistema (calcolo matrici e )

E analogamente alla predizione, vediamo ora la parte di correzione:



**Figura 3.5** – Codice per l’algoritmo di correzione del filtro EKF

**Osservazioni e Conclusioni EKF**

Durante lo sviluppo del filtro EKF su MatLab, ci siamo accorti che la stima dello stato della funivia funzionava correttamente a parte delle piccole oscillazioni per la stima dello stato ; infatti, mettendo come ingresso una forza costante di e con condizioni iniziali tutte nulle, troviamo i seguenti dati in di simulazione:

**Figura 3.6** – Differenza tra stima EKF e lo stato vero del sistema

Immagine che contiene testo, linea, diagramma, Diagramma

Descrizione generata automaticamente

Il comportamento della stima di può essere spiegato dal fatto che il rumore dei sensori diventa rilevante quando abbiamo piccole oscillazioni attorno al punto di equilibrio del sistema, da cui ne deriva un errore maggiore nei picchi dell’oscillazioni (dove ha piccole variazioni confrontabili con i rumori dei sensori).

Inoltre, per verificare se c’è accordo tra il modello utilizzato per il filtro e il sistema vero, abbiamo controllato se le innovazioni delle 3 misure dei sensori fossero dei rumori bianchi:

**Figura 3.7** – Innovazione delle misure dei sensori

Immagine che contiene testo, schermata, linea, Diagramma

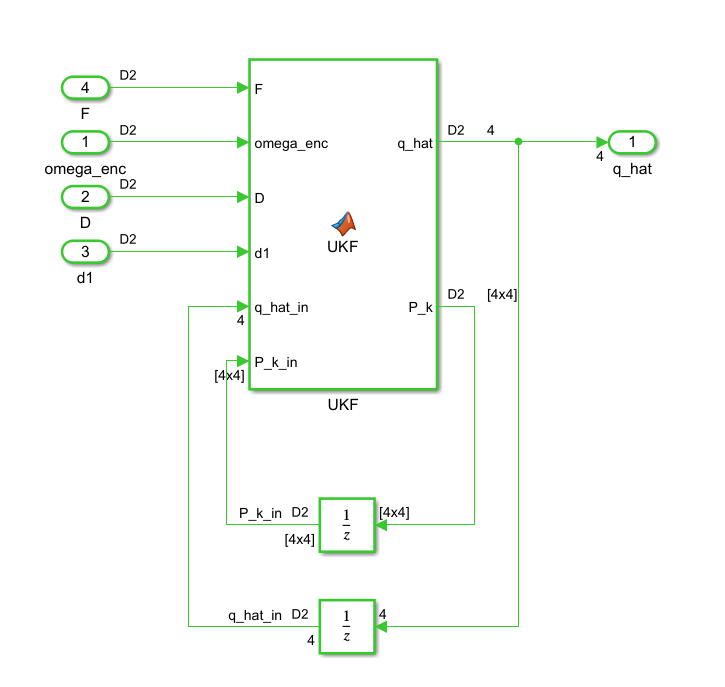
Descrizione generata automaticamente

E come possiamo vedere da Figura 3.7, la condizione di rumore bianco per tutte e tre le misure è soddisfatta.

**Filtro di Kalman Unscented (UKF)**

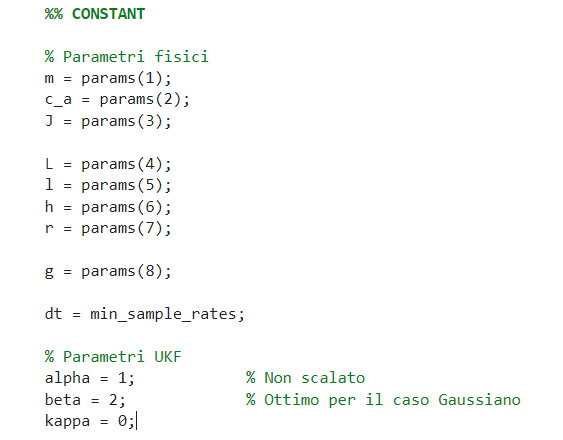
* Come possiamo vedere dalla *Figura 3.8,* il filtro EKF prevede in ingresso le misure dei sensori (con disturbo additivo di tipo Gaussiano) e l’ingresso F anch’esso disturbato.
* Abbiamo aggiunto 2 integratori di tipo discreto così che da poter avere la stima dello stato e la sua varianza all’istante precedente (cioè e ).
* Il tempo di campionamento del filtro è

dato dal tempo di campionamento più grande tra i sensori scelti (in questo caso abbiamo che il tempo ).



**Figura 3.8** – Realizzazione UKF tramite Simulink

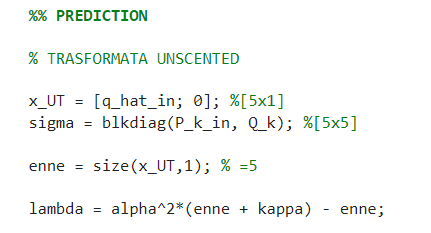
Prima di andare a vedere la predizione e correzione per il filtro, vediamo la parte iniziale del codice interno al blocco di figura *Figura 3.8*, dove definiamo le costanti e i parametri dell’UKF:



**Figura 3.9** – Definizione costanti fisiche e parametri del filtro , e

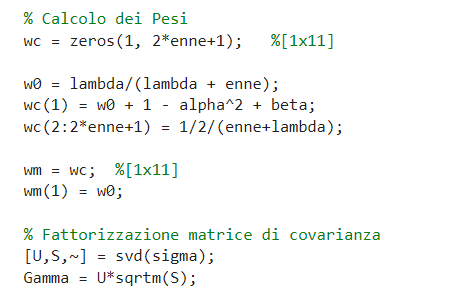
**Predizione UKF**

Partiamo con il vedere la parte di codice in cui applichiamo la trasformata unscented per la predizione dello stato, e quindi l’uso della trasformata per la funzione cioè la dinamica del sistema:

****

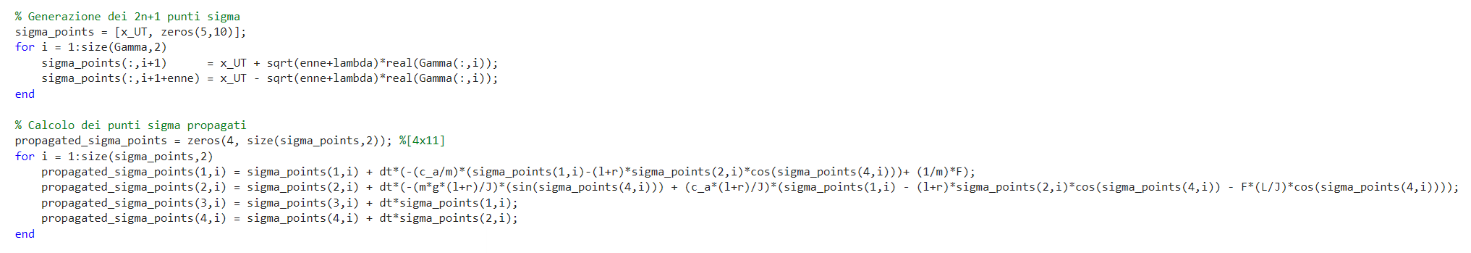
**Figura 3.10** – Definizione parametri UT (variabili di stato + rumore ) e costanti e

Il passaggio successivo è il calcolo dei pesi e la fattorizzazione della matrice di covarianza:



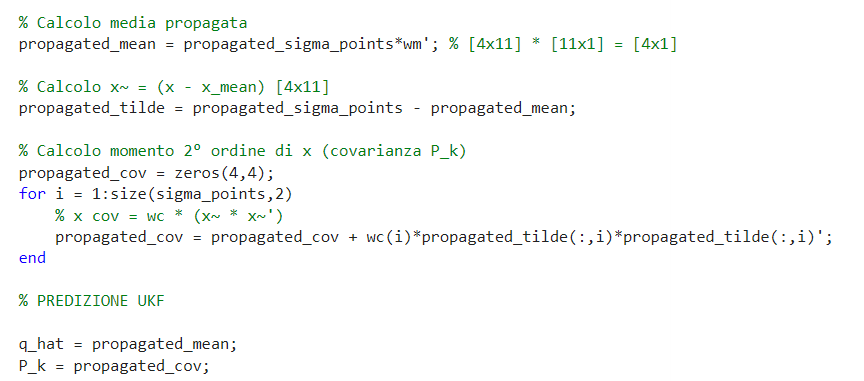
**Figura 3.10** – Calcolo pesi , e fattorizzazione SVD per ricavare la matrice

Da cui segue il calcolo dei punti sigma e l’utilizzo della per il calcolo dei punti sigma propagati (alla dinamica):



**Figura 3.11** – Calcolo punti sigma utilizzati per trovare quelli propagati nella funzione

Infine, ci basta calcolare i momenti del 1° e 2° ordine (media e covarianza dello stato, la cross-covarianza non ci interessa) attraverso i punti sigma propagati:



**Figura 3.12** – Predizione dello stato con relativo calcolo della media propagata e covarianza propagata

**Correzione UKF**

**EKF con Smoother**

In questa parte andremo a vedere una variante del filtro EKF con stima regolarizzata a posteriori della simulazione (dato che la stima regolarizzata non può essere fatta in real time, ma solo a posteriori dell’esperimento). Il codice usato per la regolarizzazione della stima è il seguente:

Immagine che contiene testo, schermata, Carattere, documento

Descrizione generata automaticamente

**Figura 3.8** – Codice per la Regolarizzazione della stima

Come risultato ci aspettiamo di trovare una stima migliore rispetto a quella dell’EKF visto in precedenza, dato che facciamo una stima dello stato con tutte le misure dell’uscita della simulazione (quindi a parità di simulazione, ho maggiori informazioni).

Infatti, il risultato ottenuto per la stima di e , comparato con il precedente EKF, è il seguente:

Immagine che contiene testo, linea, Diagramma, schermata

Descrizione generata automaticamente

**Figura 3.9** – Confronto EKF e EKF con Smoother nel caso di ingresso a 400N e condizioni iniziali nulle (10 secondi)

Come anticipato, la stima con lo smoother si avvicina maggiormente ai valori veri degli stati del sistema: da *Figura 3.9* si vede bene dalla stima di , e per apprezzare meglio il confronto anche per lo stato ci conviene vedere i primi secondi della simulazione:

Immagine che contiene linea, testo, diagramma, Diagramma

Descrizione generata automaticamente

**Figura 3.10** – Confronto EKF e EKF con Smoother nel caso di ingresso a 400N e condizioni iniziali nulle (3 secondI)