



UNIVERSITA' POLITECNICA DELLE MARCHE
FACOLTA' DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria Informatica e dell'Automazione

**Studio e sviluppo di algoritmi per
il mantenimento dell'assetto di mini-droni**

**Study and development of algorithms for
maintaining the attitude of minidron**

Relatore:

Prof. Gianluca Ippoliti

Correlatore:

Prof. Giuseppe Orlando

Tesi di Laurea di:

Albanese Giuseppe

(Mat S1077020)

A.A. 2019/2020

Studio e sviluppo di tecniche di algoritmi per il mantenimento dell'assetto per minidroni

Giuseppe Albanese Mat S1077020

Abstract

La seguente tesi ha come obiettivo quello di proporre un'analisi dettagliato del modello Simulink del minidrone "Mambo" (Parrot S.A.) allo scopo di procedere nella sintesi di un controllore realizzato con il metodo del "Luogo delle radici" che vada a controllare il beccheggio(pitch) del veicolo.

Con l'aiuto del pacchetto Matlab "*Simulink Support Package for Parrot Minidrones*", è stato prima studiato il comportamento della dinamica del sistema per poi passare allo studio e all'implementazione del modello in Simulink.

Lo studio del modello ha permesso di capire il funzionamento del drone e in particolare di analizzare e comprendere le caratteristiche dei vari blocchi e delle variabili di controllo; successivamente è stato progettato il sistema di controllo alternativo a quello di default (PID controller) e sono stati confrontati i dati delle varie tipologie di controllo.

INDICE

Introduzione	3
1 Modello matematico di un quadricottero	4
1.1 Derivazione delle grandezze fisiche fondamentali	4
1.2 Rappresentazione della dinamica rotazionale del sistema in termini lagrangiani.	7
1.3 Equazioni della dinamica	10
2 Analisi modello Mambo Parrot	14
2.1 Caratteristiche tecniche del minidrone	14
2.2 Modellazione Simulink.	15
2.3 Processo.	16
2.4 Generatore di segnali.	24
2.5 Controllo.	26
2.6 Sensori.	32
3 Controllore e Tecniche di controllo	33
3.1 PID Controller.	33
3.2 Analisi PID Mambo.	35
3.3 Tecnica del luogo delle radici.	40
4 Sintesi e Implementazione Controllore con Luogo delle radici	42
4.1 Progettazione Controllore.	42
4.2 Simulazioni del comportamento del controllore	47
5 Conclusioni	64
6 Bibliografia	65

Introduzione

I droni sono ormai diventati “oggetti” di uso comune, sono velivoli radiocomandati (quindi con un pilota che li guida da remoto) utilizzati sia in ambiti professionali, civili e militari, sia a livello consumer. Vengono anche classificati come **sistemi a pilotaggio remoto (SAPR)**. In modo un po' approssimativo potremmo dire che sono l'evoluzione tecnologica degli aeroplani telecomandati, anche se in realtà sono qualcosa di molto più sofisticato e, come vedremo nei prossimi paragrafi, sono oggetti tecnologicamente molto avanzati. Sono utilizzati in diversi settori principalmente vengono utilizzati nel settore edilizio per perizie a terreni o abitazioni. In ambito agricolo per il monitoraggio delle culture, oppure in ambito cinematografico per catturare video ed immagini aeree.

Altre aree di sviluppo dove i droni possono essere impegnati sono quella medico-sanitario e della logistica delle spedizioni a breve raggio, in cui i droni vengono utilizzati per il trasporto di merci pilotando da remoto il veicolo. Un altro settore dove possiamo trovarli è quello militare nel quale vengono utilizzati per operazione di spionaggio o di ricerca dei dispersi grazie allo streaming delle immagini che viene trasmesso di solito da una telecamera collegata al drone che permette a chi lo pilota di vedere e filmare tutto ciò che vede.

Una specifica classe di droni è costituita dai veicoli pale rotanti come “tricotteri”, “quadricotteri”, “eptacotteri”, ecc. Questi si prestano grazie alla loro maneggevolezza (in particolare alla possibilità di mantenersi in volo stazionario), alle misure ridotte, al costante miglioramento dell'autonomia di volo, ad applicazioni sempre più estese.

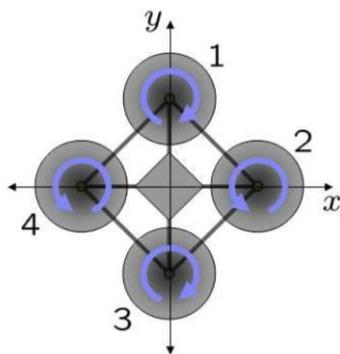


Figura 1: Schema geometrico

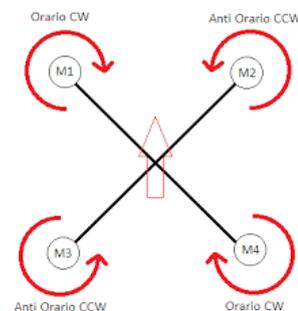


Figura 2: Quadricottero

Questa tesi si concentra sulla progettazione e il controllo di minidroni autonomi, per prima si è studiata la dinamica dei quadricotteri con l'obiettivo di ottenere una rappresentazione fedele del modello matematico poi si è passati alla fase di studio e progettazione del sistema di controllo applicando le opportune tecniche e, con l'aiuto di Simulink, si è implementato il controllore nel drone e sono stati effettuati alcuni test sul suo funzionamento.

Tra le varie aziende costruttrici di droni troviamo l'azienda Parrot S.A. che in collaborazione con il gruppo di sviluppatori di Matlab ha messo a disposizione degli utenti un pacchetto il

“*Support Package for Parrot Minidrones*” che permette di caricare un ambiente di modellazione e sviluppo, il simulatore di volo e tutte le variabili necessarie per lo studio del drone “Mambo”.

All’interno di questo ambiente, è stato possibile modificando alcuni settaggi passare dal modello lineare a quello non-lineare, accedere a tutti i blocchi dedicati all’analisi dei dati provenienti dai sensori di bordo e comprendere il funzionamento degli algoritmi di controllo (implementati mediante PID controller). Sfruttando le caratteristiche di Simulink è stato possibile isolare, all’interno dei vari blocchi, le variabili coinvolte nei processi considerati. È stato possibile studiare un modello “ridotto” relativo esclusivamente ai legami ingresso/uscita di una delle sei variabili di controllo (nel nostro caso l’angolo di beccheggio o pitch).

Attraverso i dati ottenuti è stato possibile sviluppare, attraverso la funzionalità di Matlab “Sisotool”, un controllore con il metodo “Luogo delle radici” che è andato a sostituire il sistema di controllo iniziale direttamente in Simulink.

1 Modello matematico di un quadricottero

La presente sezione fornisce una breve esposizione dei passaggi necessari per giungere ad un modello fisico di un quadricottero che descriva la dinamica del sistema. Il processo di modelling si basa su una serie di assunzioni che agevolano la trattazione e permettono di ottenere un modello semplificato implementabile a livello software:

1. struttura rigida, possibilità di trascurare i contributi dovuti alla flessione dell’intelaiatura;
2. struttura simmetrica, possibilità di valutare una matrice di inerzia diagonale;
3. coincidenza tra centro di gravità (CoG: Centre of Gravity) e origine del sistema di riferimento solidale al veicolo (Body Fixed Frame);
4. rigidità delle eliche;
5. spinta (thrust) e resistenza aerodinamica (drag) proporzionali al quadrato delle velocità angolari delle eliche ω^2 .

1.1 Derivazione delle grandezze fisiche fondamentali

Un quadricottero in termini generali può essere considerato un sistema fisico dotato di 6 gradi di libertà (DoF: Degrees of Freedom):

- direzioni di traslazione (assi X, Y, Z): forniscono lo spostamento lineare del veicolo.
- direzioni di rotazione (angoli ϕ ; θ ; ψ): forniscono lo spostamento angolare del veicolo.

Per poter fornire una corretta rappresentazione nello spazio della dinamica del sistema, capace di comprendere sia movimenti lineari che angolari, è necessario stabilire tre diversi sistemi di riferimento:

1. sistema di riferimento inerziale, scelto in riferimento al piano tangente alla superficie della Terra con origine la posizione dell’osservatore *Earth Fixed Frame*;
2. sistema di riferimento solidale al veicolo con origine il centro di gravità del veicolo *Body Fixed Frame*.
3. sistema di riferimento allineato al quadro di riferimento inerziale con origine coincidente con il centro di gravità del veicolo *Vehicle Frame*.

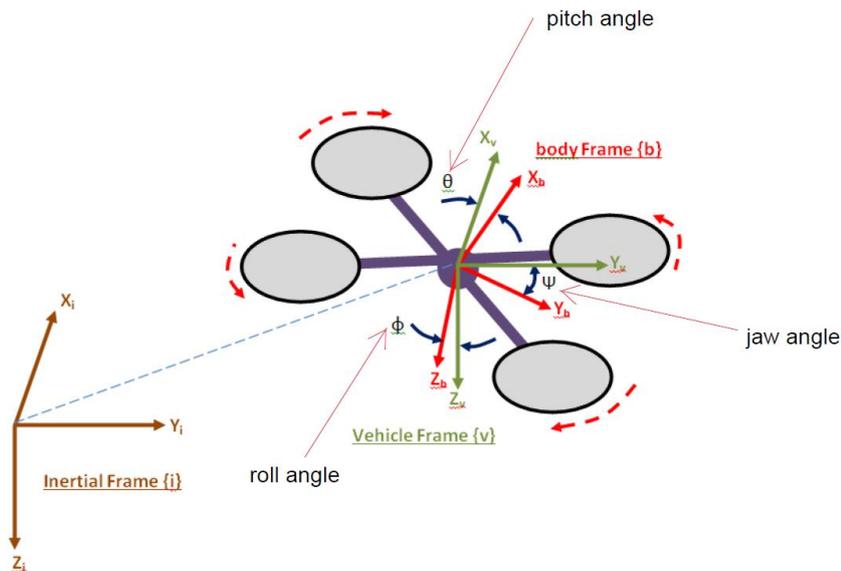


Figure 3: Earth, Body and Vehicle Frames

Mentre lo spostamento lineare è calcolabile direttamente una volta conosciuta la posizione del veicolo per lo spostamento angolare è necessario far riferimento a delle “matrici di rotazione”.

Matrici di rotazione: La rotazione di un corpo rigido nello spazio può essere parametrizzata in più modi come, ad esempio, gli *angoli di Eulero* ($\phi; \theta; \psi$), l'utilizzo dei quaternioni o gli angoli di Tait-Bryan.

Gli *angoli di Eulero* sono una rappresentazione matematica di tre rotazioni sugli assi di riferimento. Nel nostro caso gli assi sono orientati in modo che la direzione delle x positive, siano il fronte del drone, le y positive la parte destra e l'asse delle z è rivolto verso il basso.

Se consideriamo un sistema orientato con la regola della mano destra, possiamo descrivere le 3 rotazioni come:

- Matrice di rotazione attorno all'asse X: $R(X, \phi)$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

- Matrice di rotazione attorno all'asse Y: $R(Y, \theta)$

$$\begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

- Matrice di rotazione attorno all'asse Z: $R(Z, \psi)$

$$\begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Le tre matrici così ottenute sono ortonormali. La matrice di rotazione completa, che considera l'effetto combinato delle tre possibili rotazioni, si ottiene semplicemente dal prodotto (ordinato) tra le diverse matrici di rotazione. Denominando b il vettore delle coordinate rispetto al Body Fixed Frame e r il vettore delle coordinate rispetto al sistema inerziale (coordinate di navigazione), otteniamo:

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (1)$$

Che dopo aver svolti i calcoli diventa:

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} \cos\phi\cos\theta & \cos\theta\sin\phi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\psi\sin\theta & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2)$$

Che possiamo scrivere in maniera compatta:

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = R_i^b * \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3)$$

Con R_i^b matrice di trasferimento del sistema di riferimento inerziale al *Body Fixed Frame*. Permette di passare dalle coordinate espresse nel sistema di riferimento inerziale a quelle di Body Frame e viceversa. Si nota inoltre che tale matrice è ortonormale e le sue colonne sono linearmente indipendenti.

Sfruttando l'ortonormalità delle matrici possiamo ricavare la matrice di trasferimento dal Body Fixed Frame al sistema di riferimento iniziale:

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = (R_i^b)^T * \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = R_b^i * \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} \quad (4)$$

Esplicitando i termini della matrice, otteniamo:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\phi\cos\theta & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\phi\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\psi\sin\theta & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} \quad (5)$$

Una volta ottenute le matrici che consentono il passaggio dal sistema di riferimento solidale al veicolo a quello inerziale, si possono ricavare le altre grandezze fisiche di interesse, ad iniziare

dalle velocità. Le componenti della velocità traslazionale relative al sistema di riferimento solidale al veicolo sono indicate all'interno del vettore \mathbf{V}_b rispettivamente con le lettere u, v, w:

$$\mathbf{V}_b = [u \quad v \quad w]^T \quad (6)$$

Per ottenere le velocità traslazionali rispetto alle coordinate di navigazione è necessario eseguire la trasformazione:

$$\mathbf{v}_i = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_b^i * \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (7)$$

Le componenti della velocità angolare relative al sistema di riferimento solidale al corpo rigido sono misurate solitamente attraverso giroscopi presenti all'interno dell'unità IMU (Inertial Measurement Unit).

Questi sensori forniscono la variazione degli angoli di roll, pitch, yaw, indicati all'interno del vettore velocità angolare (ω_b). Rispettivamente con le lettere p, q, r:

$$\omega_b = [p \quad q \quad r]^T \quad (8)$$

Il passaggio dalle componenti appena descritte alle componenti relative alla variazione degli angoli euleriani (ϕ ; θ ; ψ), che descrivono invece la rotazione del sistema rispetto al sistema di riferimento inerziale, è garantito dalla trasformazione:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \frac{\sin\theta}{\cos\theta} & \cos\phi \frac{\sin\theta}{\cos\theta} \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = Q_b^i \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (9)$$

Come si può vedere nella matrice non compare l'angolo ψ , la sua variazione dipende dalla variazione degli angoli ϕ e θ (pitch e roll) ciò significa che manovre di variazioni di pitch e roll influiscono anche sulla variazione dell'angolo di yaw (ψ).

La trasformazione inversa, che fornisce le variazioni delle velocità angolari rispetto al sistema di riferimento solidale al veicolo, si ottiene applicando la matrice:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\phi \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = Q_i^b \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (10)$$

1.2 Rappresentazione della dinamica rotazionale del sistema in termini lagrangiani

Una volta ricavate le grandezze fisiche fondamentali è possibile avvalersi delle equazioni di Eulero-Lagrange per descrivere la dinamica rotazionale del sistema. Poiché abbiamo presupposto la rigidità della struttura, ovvero un vincolo di tipo olonomo, possiamo utilizzare un sistema di coordinate generalizzate (in particolare gli “angoli di Eulero” introdotti precedentemente e le componenti del vettore posizione valutate rispetto al Body Fixed Frame) per descrivere la rotazione del sistema. Un sistema di coordinate generalizzato, in presenza di forze di tipo conservativo, permette di esprimere le equazioni del moto in termini di funzioni scalari, considerando esclusivamente l’energia cinetica (T) e l’energia potenziale (V). La Lagrangiana (L) di un sistema fisico è definita come la differenza tra l’energia cinetica e l’energia potenziale

$L = T - V$, se questa quantità è nota allora le equazioni della dinamica del sistema si possono scrivere:

$$\Gamma_i = \frac{d}{dt} \left(\frac{\partial}{\partial \dot{q}_i} L \right) - \frac{\partial}{\partial q_i} L \quad (11)$$

Dove ci calcoliamo le forze generalizzate agenti sul sistema e q_i sono le coordinate generalizzate. Per il calcolo dell’energia cinetica in termini Lagrangiani è necessario trasformare le componenti della velocità, espresse in coordinate cartesiane (*Earth Fixed Frame*) in coordinate generalizzate, utilizzando la matrice R_b^i come matrice di cambiamento di base. Delle componenti del vettore velocità così ottenute è necessario quindi calcolare il quadrato del modulo. Dopo aver sviluppato i vari termini e svolti i vari calcoli si ottiene:

$$\begin{aligned} \|\mathbf{v}_i(b_1, b_2, b_3)\|^2 &= (b_2^2 + b_3^2) * (\dot{\psi}^2 \sin^2 \theta - 2 \sin \theta \dot{\phi} \dot{\psi} + \dot{\phi}^2) \\ &+ (b_1^2 + b_3^2) * (\dot{\psi}^2 \sin^2 \phi \cos^2 \theta + 2 \sin \phi \cos \phi \cos \theta \dot{\theta} \dot{\psi} + \cos^2 \phi \dot{\theta}^2) \\ &+ (b_1^2 + b_2^2) * (\dot{\psi}^2 \cos^2 \phi \cos^2 \theta - 2 \sin \phi \cos \phi \cos \theta \dot{\theta} \dot{\psi} + \sin^2 \phi \dot{\theta}^2) \\ &+ 2b_1 b_2 * (\dot{\psi}^2 \sin \phi \sin \theta \cos \theta + \dot{\psi} (\cos \phi \sin \theta \dot{\theta} - \sin \phi \cos \theta \dot{\phi}) - \cos \phi \dot{\phi} \dot{\theta}) \\ &+ 2b_1 b_3 * (\dot{\psi}^2 \cos \phi \sin \theta \cos \theta + \dot{\psi} (-\cos \phi \cos \theta \dot{\phi} - \sin \phi \sin \theta \dot{\theta}) + \sin \phi \dot{\phi} \dot{\theta}) \\ &+ 2b_2 b_3 * (-\dot{\psi}^2 \sin \phi \cos \theta \cos^2 \theta + \dot{\phi} (\sin^2 \phi \cos \theta \dot{\theta} - \cos^2 \phi \cos \theta \dot{\theta}) + \sin \phi \cos \phi \dot{\theta}^2) \end{aligned}$$

Per procedere al calcolo dell’energia cinetica è sufficiente conoscere il contributo di ognuna delle masse infinitesime che compongono il sistema rispetto alla propria posizione espressa attraverso le coordinate generalizzate. L’energia cinetica può quindi essere espressa come:

$$T = \frac{1}{2} \int dm(\mathbf{r}_i(b_1, b_2, b_3)) * \|\mathbf{v}_i(b_1, b_2, b_3)\|^2 \quad (12)$$

L'espressione può essere notevolmente semplificata esprimendo l'energia cinetica attraverso le componenti del tensore momento di inerzia anch'esse espresse nel sistema di coordinate generalizzate:

$$I = \int dm \begin{bmatrix} (b_2^2 + b_3^2) & -b_1b_2 & -b_1b_3 \\ -b_1b_2 & (b_1^2 + b_3^2) & -b_2b_3 \\ -b_1b_3 & -b_2b_3 & (b_1^2 + b_2^2) \end{bmatrix} \quad (13)$$

dove i termini sulla diagonale si riferiscono ai momenti di inerzia rispetto agli assi principali del *Body Fixed Frame*, mentre gli elementi fuori diagonale sono detti prodotti di inerzia e si possono vedere come l'accoppiamento incrociato dell'inerzia degli assi. Poiché abbiamo presupposto la simmetria del veicolo si possono omettere i prodotti di inerzia e quindi la matrice di inerzia può essere riscritta come:

$$I = \int dm \begin{bmatrix} (b_2^2 + b_3^2) & 0 & 0 \\ 0 & (b_1^2 + b_3^2) & 0 \\ 0 & 0 & (b_1^2 + b_2^2) \end{bmatrix} = \begin{bmatrix} I_{b_1,b_1} & 0 & 0 \\ 0 & I_{b_2,b_2} & 0 \\ 0 & 0 & I_{b_3,b_3} \end{bmatrix} \quad (14)$$

Questa proprietà permette di riscrivere T senza prodotti di inerzia:

$$T = \frac{1}{2}I_{b_1,b_1}(\dot{\phi} - \dot{\psi}\sin\theta)^2 + \frac{1}{2}I_{b_2,b_2}(\dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta)^2 + \frac{1}{2}I_{b_3,b_3}(\dot{\theta}\sin\phi - \dot{\psi}\cos\phi\cos\theta)^2 \quad (15)$$

Analogamente l'espressione dell'energia potenziale rispetto al sistema di coordinate generalizzato si ottiene a partire dalla quota verticale nel sistema di coordinate inerziale (z) calcolando il contributo di ogni massa infinitesima:

$$V = \int b_1(-g*\sin\theta)dm(b_1) + \int b_2(g*\sin\phi\cos\theta)dm(b_2) + \int b_3(g*\cos\phi\cos\theta)dm(b_3) \quad (16)$$

A questo punto si può considerare la lagrangiana del sistema $L = T - V$ e scrivere le equazioni relative alla dinamica rotazionale specificando le coordinate e le forze generalizzate:

$$\begin{aligned} \tau_\phi &= \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) - \frac{\partial L}{\partial \phi} \\ \tau_\theta &= \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} \\ \tau_\psi &= \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\psi}} \right) - \frac{\partial L}{\partial \psi} \end{aligned} \quad (17)$$

In particolare, le forze generalizzate si presentano come momenti meccanici (torques) che descrivono l'effetto giroscopico generato dalla rotazione del corpo rigido. Dopo aver sviluppato le derivate e applicato alcune manipolazioni algebriche per semplificare le equazioni, trascurando i termini riguardanti l'energia potenziale (cioè considerando nulla la quota iniziale) e considerando esclusivamente variazioni per piccoli angoli ($\cos x = 1$; $\sin x = 0$) si ottiene:

$$\begin{aligned}\tau_\phi &= I_{b_1, b_1} \ddot{\phi} = (I_{b_2, b_2} - I_{b_3, b_3}) \dot{\theta} \dot{\psi} \\ \tau_\theta &= I_{b_2, b_2} \ddot{\theta} = (I_{b_3, b_3} - I_{b_1, b_1}) \dot{\phi} \dot{\psi} \\ \tau_\psi &= I_{b_3, b_3} \ddot{\psi} = (I_{b_1, b_1} - I_{b_2, b_2}) \dot{\phi} \dot{\theta}\end{aligned}\quad (18)$$

Queste equazioni descrivono il moto rotazionale del sistema in assenza di momenti meccanici esterni.

1.3 Equazioni della dinamica

Nel calcolo delle equazioni generali del sistema dovremo considerare anche i contributi derivati dai momenti meccanici esterni, dovuti per esempio all'azione dei motori. Essendo un sistema fisico con 6 gradi di libertà (3 gradi di libertà traslazionale e 3 gradi di libertà rotazionale) ma essendo dotato di soli quattro motori, un quadricottero è tipico sistema sotto-attuato (under-actuated system), ciò comporta che solo alcuni movimenti potranno essere controllati direttamente tramite l'azione dei motori mentre gli altri movimenti saranno controllati agendo su combinazioni di questi ultimi:

- spinta verticale (Thrust)
- beccheggio (Pitch)
- rollio (Roll)
- imbardata (Yaw)

La spinta verticale è dovuta all'azione sincronizzata dei quattro motori che, attraverso la rotazione delle eliche, convogliano verso il basso una massa di aria tale da generare una differenza di pressione sufficiente a sollevare il quadricottero. Il beccheggio è dovuto al momento generato dalle spinte non uniformi dovute alle diverse velocità di rotazione dei motori anteriori e posteriori. Analogamente il rollio è dovuto al momento generato dalle spinte non uniformi dovute alle diverse velocità di rotazione dei motori posizionati sul lato destro rispetto a quelli posti sul lato sinistro.

L'imbardata si ottiene invece in condizioni di equilibrio per quanto riguarda le spinte dovute all'azione dei motori disposti simmetricamente che annullano reciprocamente i momenti meccanici generati sul veicolo il quale, infatti, non si inclina. Lo Yaw è dovuto alla massa delle eliche che, una volta messe in rotazione, generano un momento angolare che in presenza di un vincolo, rappresentato dalla struttura dell'aeromobile, tende a far ruotare il sistema. In questo caso l'azione dei motori front left e back right (così come quella di front right e backleft) che ruotano nello stesso verso, risulterà accoppiata. Poiché nel progetto si è utilizzato un modello di quadricottero con configurazione "a croce", ovvero con i motori disposti simmetricamente rispetto all'asse principale del veicolo, l'azione dei motori può essere schematizzata in questo modo:

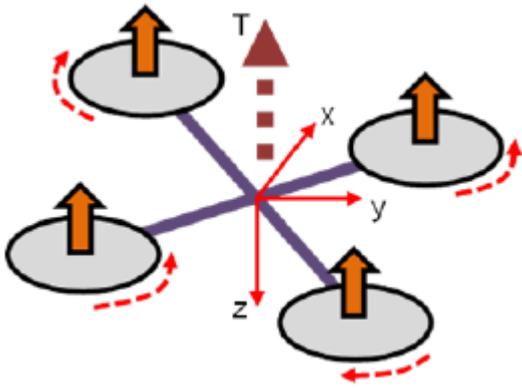


Figura 4: Manovra di Thrust

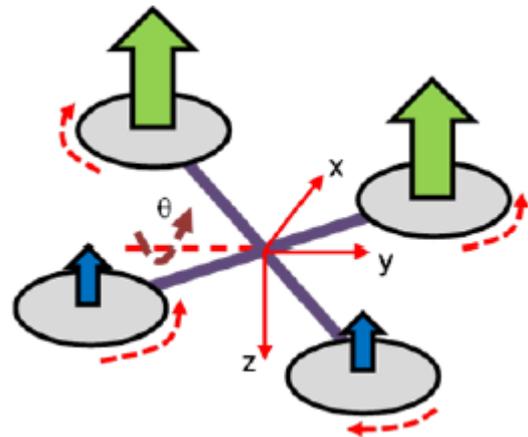


Figura 5: Manovra di Pitch

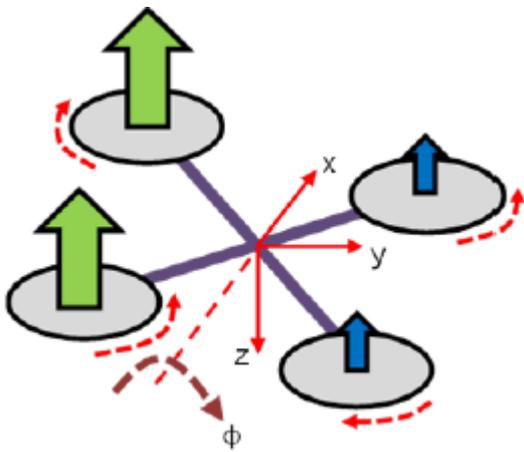


Figura 6: Manovra di Roll

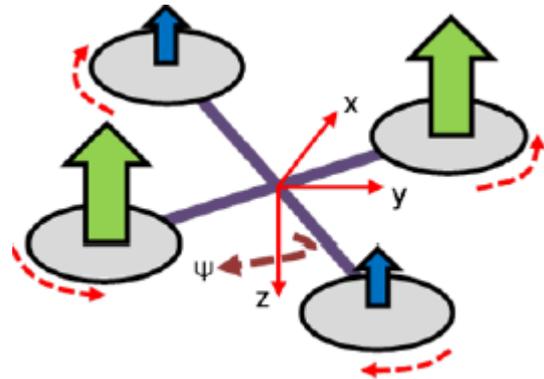


Figura 7: Manovra di Yaw

Alcuni movimenti non sono controllabili direttamente come visto in precedenza, ma lo saranno attraverso azioni accoppiate.

Ad esempio, quelli in direzione longitudinale e quelli in direzione laterale saranno ottenuti inclinando preventivamente il drone verso la direzione richiesta.

Dei quattro propulsori due ruotano in verso orario due in verso antiorario: in particolare su ciascun braccio della croce i due motori opposti ruotano equiversi (1-3 e 2-4).

Su ciascuna diagonale quindi si viene ad avere una coppia agente sull'asse su cui sono posizionati i motori.

Se tutti i motori girano ad un valore di giri identico, non c'è necessità di rotori aggiuntivi per equilibrare le coppie generate, in quanto uguali ed opposte, ottenendo accelerazione angolare nulla intorno agli assi di rotazione. Ne consegue che modificando opportunamente i giri, quindi le forze, si ottiene uno scompenso di forze e/o momenti in grado di far effettuare al veicolo le manovre desiderate. Nel dettaglio, la manovra di salita, la più semplice viene eseguita aumentando la spinta di tutti i rotori allo stesso modo. La condizione di volo a punto fisso, detta anche hovering, è possibile considerarla un caso particolare della manovra di salita in cui

ciascun motore ha tiro pari ad un quarto del peso del drone raggiungendo il bilancio delle forze tra spinta e peso.

Incrementando la potenza di un rotore e decrementando quella dell'opposto si mantengono le coppie complessive globalmente nulle ma si provoca una rotazione attorno dell'asse su cui i motori non cambiano regime: il laterale per il beccheggio (y) e il longitudinale per il rollio (x), ovvero le manovre desiderate. Riguardo al movimento di imbardata, esso è ricavato fornendo potenza ugualmente su una coppia di motori e togliendone allo stesso modo sull'altra.

A questo punto è possibile inserire nelle equazioni che descrivono la dinamica del sistema anche l'azione dei momenti meccanici esterni dovuti all'azione dei motori. Indicando con b il coefficiente di spinta dei motori; con l_x , l_y le proiezioni dei bracci del minidrone rispettivamente sugli assi x e y ; con d il coefficiente di resistenza dell'aria, otteniamo:

$$\begin{aligned}\tau_\phi &= I_{b_1, b_1} \ddot{\phi} = (I_{b_2, b_2} - I_{b_3, b_3}) \dot{\theta} \dot{\psi} + b l_y (-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \tau_\theta &= I_{b_2, b_2} \ddot{\theta} = (I_{b_3, b_3} - I_{b_1, b_1}) \dot{\phi} \dot{\psi} + b l_x (\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \tau_\psi &= I_{b_3, b_3} \ddot{\psi} = (I_{b_1, b_1} - I_{b_2, b_2}) \dot{\phi} \dot{\theta} + d (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)\end{aligned}\quad (19)$$

Il modello fornito sopra dovrebbe essere ulteriormente espanso considerando l'effetto giroscopico risultante dalla rotazione delle eliche in caso di inclinazione del veicolo (variazione angoli ϕ e θ), indicando con J_r il momento di inerzia del rotore, si ha:

$$\begin{aligned}\tau'_\phi &= J_r \dot{\theta} (\Omega_1 - \Omega_2 - \Omega_3 + \Omega_4) \\ \tau'_\theta &= J_r \dot{\phi} (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4)\end{aligned}\quad (20)$$

Per giungere al modello generale del sistema bisogna aggiungere ai momenti meccanici l'azione delle forze agenti sul sistema. In questo caso l'analisi è molto più semplice poiché l'unica forza agente sul sistema è la forza peso che è facilmente espressa in relazione al sistema di riferimento inerziale: $F_p = [0 \quad 0 \quad mg]^T$.

L'azione dovuta ai motori e la spinta verticale atta a contrastare proprio la forza peso. Esprimendo i contributi di tale spinta in relazione al sistema di riferimento inerziale, cioè utilizzando la matrice R_b^i . Come matrice del cambiamento di base otteniamo:

$$\begin{aligned}m\ddot{x} &= (\sin\phi \sin\psi + \cos\phi \sin\theta \cos\psi) b (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ m\ddot{y} &= (-\sin\phi \cos\psi + \cos\phi \sin\theta \cos\psi) b (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ m\ddot{z} &= mg - (\cos\phi \cos\psi) b (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)\end{aligned}\quad (21)$$

Le equazioni generali riguardanti la dinamica del sistema possono infine essere espresse in forma unitaria come:

$$\begin{cases} F_x = (\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi)b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ F_y = (-\sin\phi\cos\psi + \cos\phi\sin\theta\cos\psi)b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ F_z = mg - (\cos\phi\cos\psi)b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \tau_\phi = (I_{b_2,b_2} - I_{b_3,b_3})\dot{\theta}\dot{\psi} + bl_y(-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \tau_\theta = (I_{b_3,b_3} - I_{b_1,b_1})\dot{\phi}\dot{\psi} + bl_x(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \tau_\psi = (I_{b_1,b_1} - I_{b_2,b_2})\dot{\phi}\dot{\theta} + d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{cases} \quad (22)$$

Nella modellazione del quadricottero si sono effettuate alcune assunzioni che hanno permesso di semplificare le equazioni ottenute mantenendo contemporaneamente il modello fedele al reale comportamento del drone:

- Il quadricottero ha una struttura perfettamente simmetrica e la matrice di inerzia è diagonale
- Il quadricottero, ad eccezione delle eliche, è un corpo rigido
- La spinta è proporzionale al quadrato della velocità di rotazione
- Il centro di gravità è fisso e coincide sempre con l'origine del sistema di riferimento solidale con il drone
- Il sistema è tempo invariante, massa e struttura non variano nel tempo
- I quattro motori sono identici, quindi analizzarne uno singolarmente non fa perdere di generalità

Una volta raggiunto un modello matematico della dinamica di un quadricottero adeguato ai nostri scopi è necessario valutare tale modello generale in relazione alle caratteristiche specifiche del minidrone Mambo.

In particolare, nel prossimo capitolo andremo ad analizzare le caratteristiche del Mambo ed il suo modello in Matlab&Simulink andando a studiare ogni singolo blocco del modello Simulink e a spiegare il funzionamento di ognuno di essi.

2 Analisi modello Mambo Parrot

2.1 Caratteristiche tecniche del minidrone

Il drone preso in considerazione per questo elaborato è il Parrot Mambo, dell'azienda francese Parrot. La scelta è ricaduta su questo modello in quanto Matlab fornisce un modello Simulink del drone già implementato e scaricabile gratuitamente dal link <https://it.mathworks.com/hardware-support/parrot-minidrones.html>. Su tale modello è possibile compiere ogni azione di modifica, sia a livello di processo sia a livello di controllore. Alcune caratteristiche tecniche del drone:

- **Sensori di stabilizzazione**

Unità di misura inerziale (IMU): per valutare la velocità, l'inclinazione e il contatto con gli ostacoli (accelerometro a 3 assi e giroscopio a 3 assi).

Sensore a ultrasuoni: montato sul lato inferiore, per la misurazione della quota verticale mediante la determinazione del tempo di propagazione del segnale emesso dal drone e riflesso dal suolo.

Sensore di pressione: per la misurazione indiretta dell'altitudine.

Fotocamera: montata anch'essa sul lato inferiore, da 60 FPS (frame per second) che fornisce i dati necessari al software integrato di elaborazione delle immagini 'Optical Flow'. Questa tecnologia, sfruttando la posizione relativa degli oggetti tra un fotogramma e l'altro è in grado di stimare il movimento orizzontale del drone e quindi la sua velocità.

- **Energia:** Batteria LiPo 660 mAh con 9 minuti di autonomia senza accessori o paraurti a fronte di un tempo di carica di circa 30 minuti

- **Peso:** 63g (senza paraurti o accessori).

- **Dimensioni:** 18 x 18 cm con paraurti, 13.2 x 13.2 cm senza paraurti.



Figura 8: Mambo lato-inferiore



Figura 9: Mambo lato-superiore

2.2 Modellazione Simulink

Simulink è un software per la modellazione, simulazione e analisi di sistemi dinamici, sviluppato dalla compagnia statunitense Mathworks. Questo software è strettamente integrato con Matlab.

Il modello Simulink del minidrone Mambo, fornito all'interno del *Simulink Support Package for Parrot Minidrones*, si compone di diversi blocchi, analizzabili separatamente.

Lanciando il comando `asbQuadcopterStart` sulla command window di Matlab si apriranno diverse finestre che ci permetteranno di visualizzare a schermo il Simulatore di volo del drone, lo schema a blocchi del modello e infine tutte le variabili necessarie vengono caricate nel progetto.

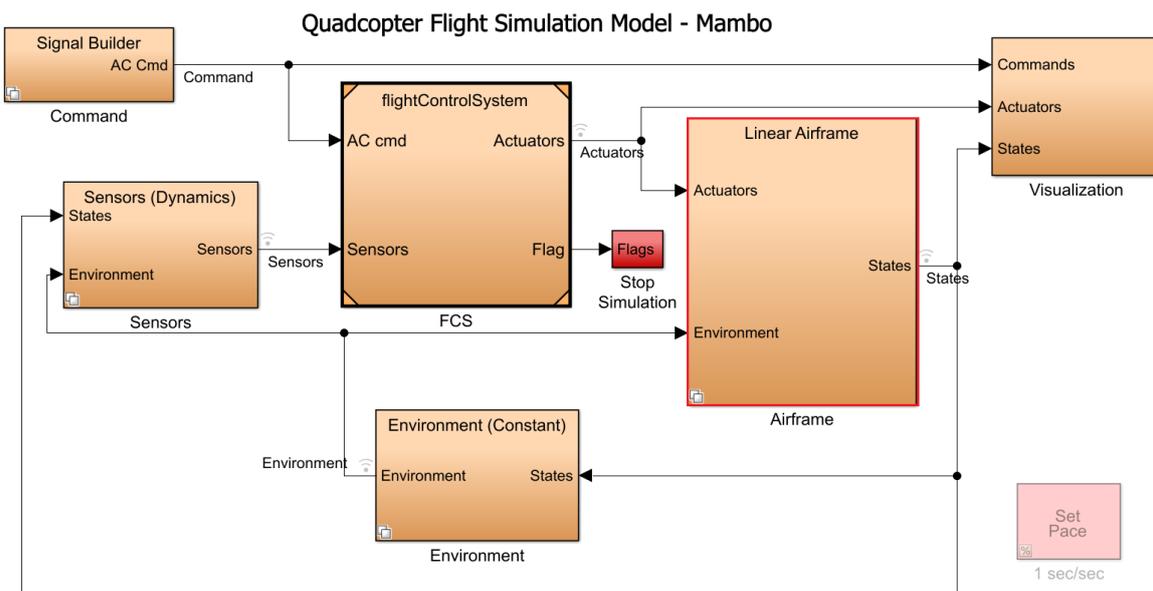


Figura 10: Progetto Completo

La struttura presentata è quella tipica di un sistema di controllo in controreazione, in base alle funzioni svolte si possono distinguere diversi blocchi ognuno dei quali ha un compito specifico nella modellazione e controllo del nostro veicolo.

Linear Airframe : modello del processo;

Signal Builder: generatore dei segnali di riferimento

Flight Control System : controllore del processo;

Environment : blocco contenente le variabili di ambiente;

Sensors : blocco contenente le costanti di trasduzione e le relazioni tra i diversi sensori presenti a bordo;

Visualization : blocco ausiliario che consente la visualizzazione dei valori delle variabili di stato e dei tracciati dei segnali di controllo.

2.3 Processo

Il modello del processo è contenuto all'interno del blocco Lineare Airframe, Simulink ci da la possibilità di scegliere tra Linear Airframe e Non-Linear Airframe andando a modificare la variabile VSS_VEHICLE che se impostata a 0 carica il modello lineare del processo mentre se impostata a 1 carica il modello non lineare del processo.

La variabile è modificabile tramite la command window di Matlab o in alternativa modificando il file .m contenente le variabili di inizializzazione di Matlab.

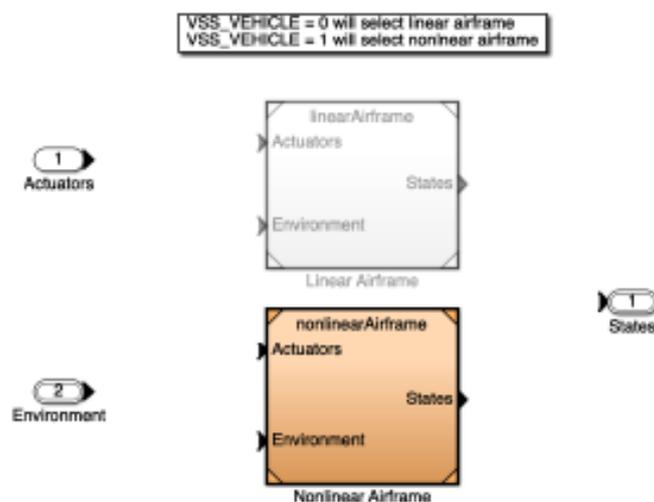


Figura 11: Linear-Airframe/NonLinear-Airframe

Settando la variabile $VSS_VEHICLE = 1$ verrà selezionato il NonLinear-Airframe e aprendo il blocco visto in precedenza si aprirà il modello non lineare composto da 3 grandi componenti:

AC model: è il blocco che contiene la parte della dinamica, in ingresso troviamo le velocità angolari w_b e le velocità lineari v_b espresse rispetto al sistema solidale con il body del drone ed uscenti dal blocco 6DOF. In uscita si hanno le forze e i momenti su tutti e tre gli assi.

6DOF Quaternion: blocco fornite da Matlab prende in ingresso i momenti e le forze. Determina in uscita le grandezze di velocità, accelerazione e angoli.

Bus setup: compone un vettore di bus. Contiene tutte le velocità in entrambi i sistemi di riferimento oltre a posizione espressa in ned , North – East – Down, accelerazioni e angoli di rotazione.

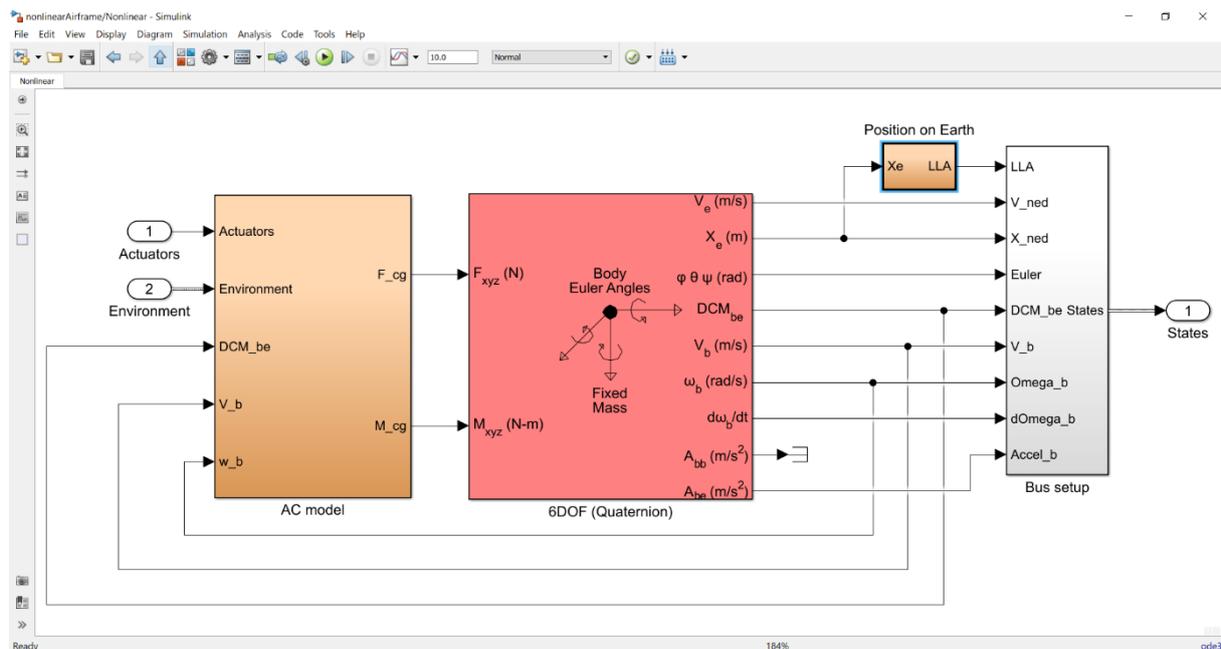


Figura 12: Modello non Lineare

In particolare, possiamo notare come le forze e i momenti meccanici, calcolati rispetto al centro di gravità del veicolo, siano generati dal blocco AC model. Questo blocco ha come variabili di ingresso:

- l'azione di spinta degli attuatori Actuators;
- l'azione delle variabili di ambiente Environment.

- i valori della matrice DCM (Direct Cosine Matrix) necessari per il passaggio dal sistema di riferimento del corpo rigido a quello inerziale.
- la velocità lineare, misurata nel sistema di riferimento del corpo rigido o Body Frame (v_b);
- la velocità angolare, misurata nel sistema di riferimento del corpo rigido o Body Frame (w_b);

Le formule che descrivono l'azione del blocco riprendono e specificano quelle del modello matematico generale. Per le forze avremo che :

$$m\dot{v}_b = -m\omega_b \times v_b + mgR_i^b e_3 + \sum_{N,E,S,W} t_i \quad (23)$$

Mentre per quanto riguarda i momenti meccanici :

$$I\dot{\omega}_b = -\omega_b \times I\omega_b + \sum_{N,E,S,W} (m_i + q_i) \quad (24)$$

dove con le lettere N,E,S,W si indica il posizionamento dei quattro motori.

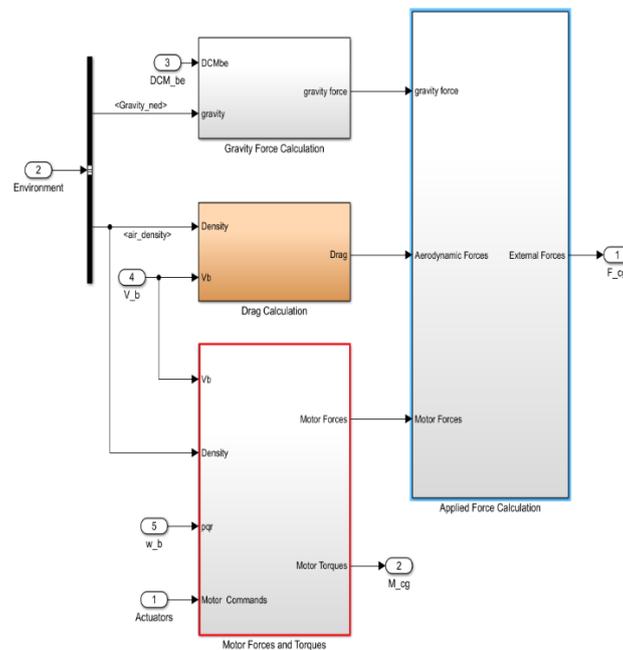


Figura 13:AC model

Nella figura precedente vediamo in dettaglio l'AC model e i blocchi che lo costituiscono, qui possiamo notare:

Gravity Force Calculation : calcola la forza di gravità rispetto al sistema solidale con il drone. Ha come ingresso la matrice di rotazione DCM_be e la costante di gravità.

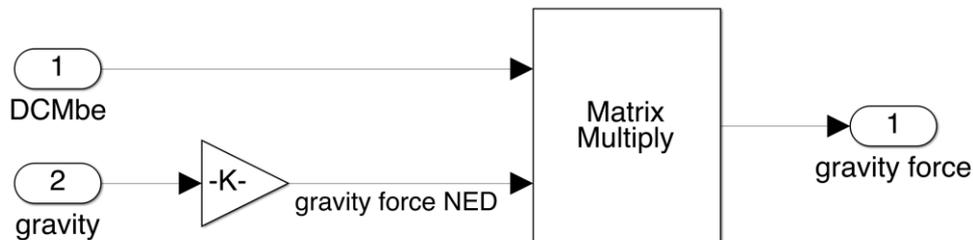


Figura 14 : Gravity Force Calculation

Drag Calculation : Calcola la forza di resistenza che l'aria imprime al drone in movimento. Tale forza è dovuta allo spostamento del drone ed è esprimibile come :

$$D = \frac{1}{2} \rho V^2 S C_D \quad (25)$$

Dove:

ρ è la densità dell'aria

V è la velocità di traslazione del drone

S è la superficie che impatta l'aria durante la traslazione

C_b è il coefficiente di Drag

Nelle figure 15, 16 si vede l'applicazione al nostro problema della formula (25) per il calcolo della forza di drag.

Il vettore S che entra nel blocco moltiplicatore in figura 12 contiene la quantità di superficie che va ad impattare con il fluido (aria). Le prime due componenti di tale vettore (x, y) sono un quarto della terza componente del medesimo vettore S. Ciò accade in quanto si considera il drone come un parallelepipedo tale per cui vi è la seguente relazione tra le superfici (SUP):

$$SUP_{superiore} = SUP_{inferiore} = diamiter = \sum_{i=1}^4 SUP_{laterale_i} \quad (26)$$

Motor Force and Torques Contiene :

- 1) Blocco centrale il quale ha come ingressi le velocità lineari e angolari rispetto al body frame, il vettore Motor Commands il quale contiene lo sforzo di controllo che arriva dal controllore ed infine il vettore D che descrive il posizionamento rispetto al centro di massa dei rotori.
- 2) Blocco MotorToW il quale trasforma lo sforzo di controllo in uno sforzo interpretabile dai motori. (Tale conversione avviene attraverso blocchi di Gain e saturazioni).
- 3) Blocco ForEach che ci permette di analizzare ogni attuatore in particolare ingressi e uscite per il calcolo di momenti e forze.

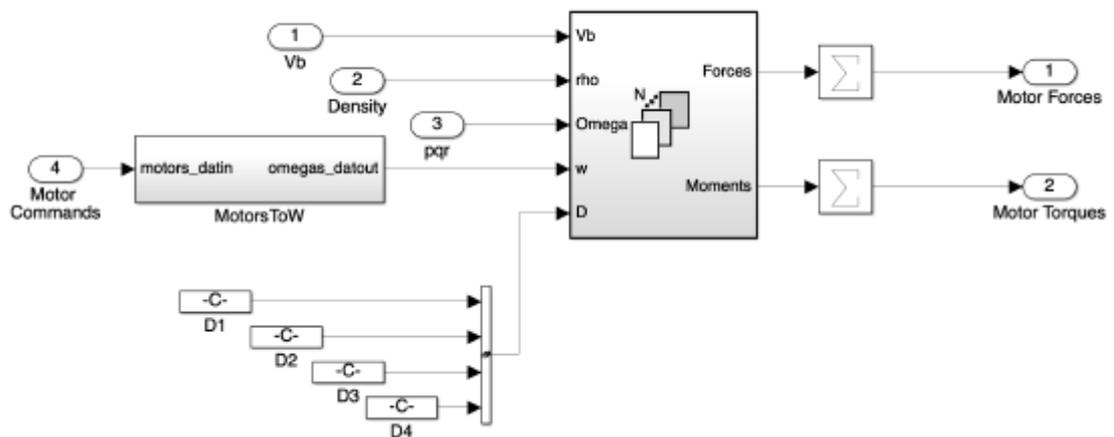


Figura 17 : Motor Force and Torques

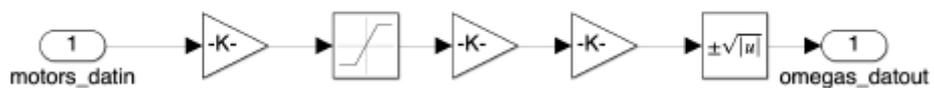


Figura 18 : MotorToW

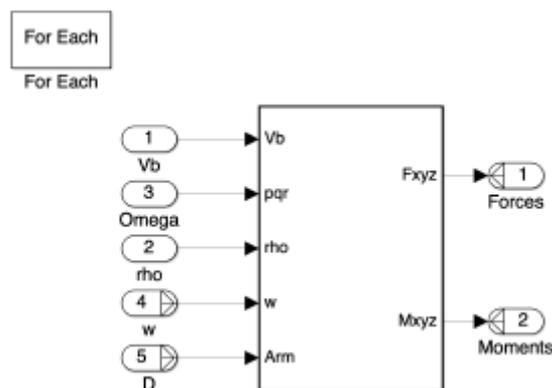


Figura 19 : Dinamica per ogni motore

seguendo il percorso: Motor Forces and Torques/For Each Subsystems/Rotor Dynamics si raggiunge il blocco dedicato al calcolo delle forze e dei momenti generati dall'azione dei motori.

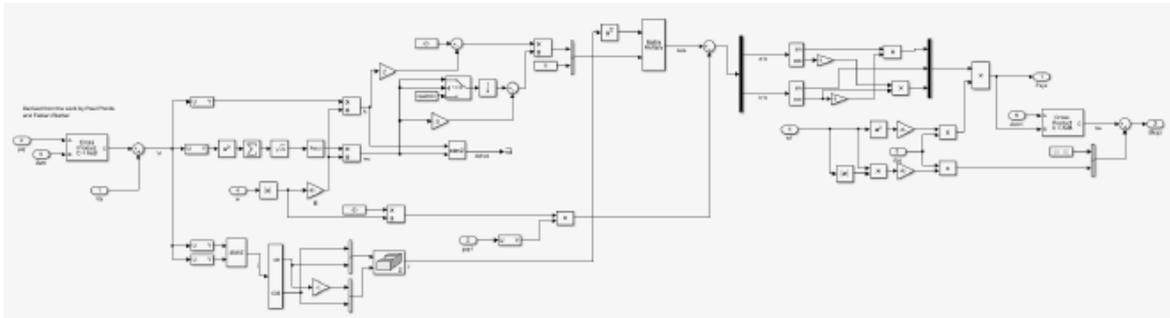


Figura 20 : Modello di calcolo forze e momenti

In uscita a tale modello abbiamo le forze e i momenti meccanici esterni che vengono passati al blocco 6DOF Euler Angles che si occupa di modellare la dinamica del sistema, fornendo le variabili di stato ai blocchi che ne hanno bisogno:

In particolare, avremo le seguenti variabili (Visibili anche in Figura 12) :

- V_e [m/s] : Velocità traslazionale nel sistema di riferimento inerziale (*Earth Fixed Frame*)
- x_e [m] : posizione rispetto all'*Earth Fixed Frame*;
- $(\phi; \theta; \psi)$: angoli di Eulero rispetto al Body Frame;
- DCM_be : coefficienti della matrice Direct Cosine Matrix;
- v_b [m/s] : velocità traslazionale rispetto al Body Frame;
- w_b [rad/s] : velocità angolare rispetto al Body Frame;
- $\frac{dw_b}{dt}$ [rad/s] : Derivata velocità angolare rispetto al Body Frame;

- A_{be} [m/s^2] : accelerazione lineare corpo rigido rispetto al *Earth Fixed Frame*;
- LLA : latitudine, longitudine e altitudine calcolate in coordinate geodetiche.

2.4 Generatore di segnali

All'interno del blocco Command è possibile scegliere qual è il segnale di riferimento in ingresso del nostro modello.

In particolare, andando a modificare la variabile VSS_command è possibile selezionare:

- Il blocco Signal Builder, cioè il costruttore di segnali di default (VSS_command = 0).
- Il blocco Joystick che permette di gestire manualmente il comportamento del veicolo (VSS_command = 1).
- Il blocco Data che consente di selezionare i riferimenti tra quelli presenti in un file .mat (VSS_command = 2).
- Il blocco Spreadsheet Data che consente di selezionare i valori dei riferimenti da un foglio di calcolo (VSS_command = 3).

Selezionando il Signal Builder è possibile fornire in ingresso al sistema segnali di riferimento di posizione (X,Y,Z) che riferimenti di orientamento (Pitch,Roll,Yaw).

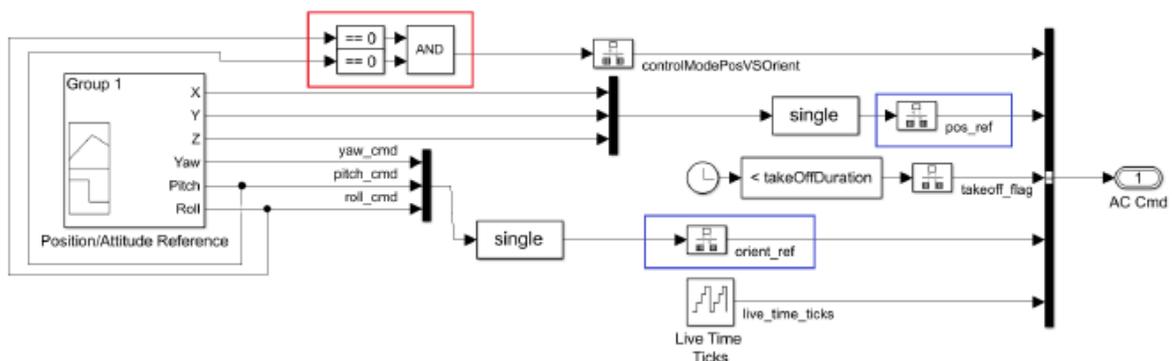


Figura 21 : Blocco Signal Builder

Particolare attenzione, come vedremo anche in seguito, va verso la porta logica AND che mi permette di passare dalla modalità di controllo di posizione (se Pitch e Roll sono uguali a 0) altrimenti rimane in modalità di controllo di orientamento.

All'interno del blocco Position/Attitude reference è possibile regolare l'andamento delle variabili di riferimento relative alla posizione: X, Y, Z (in quest'ultimo caso il riferimento $_e$ negativo poiché l'asse Z nel modello è orientato verso il basso) e all'orientamento in aria del veicolo: angoli di Roll, Pitch, Yaw. L'interfaccia grafica consente di modificare anche l'ampiezza dell'intervallo temporale da considerare e, ovviamente, l'andamento delle diverse variabili nel corso del tempo.

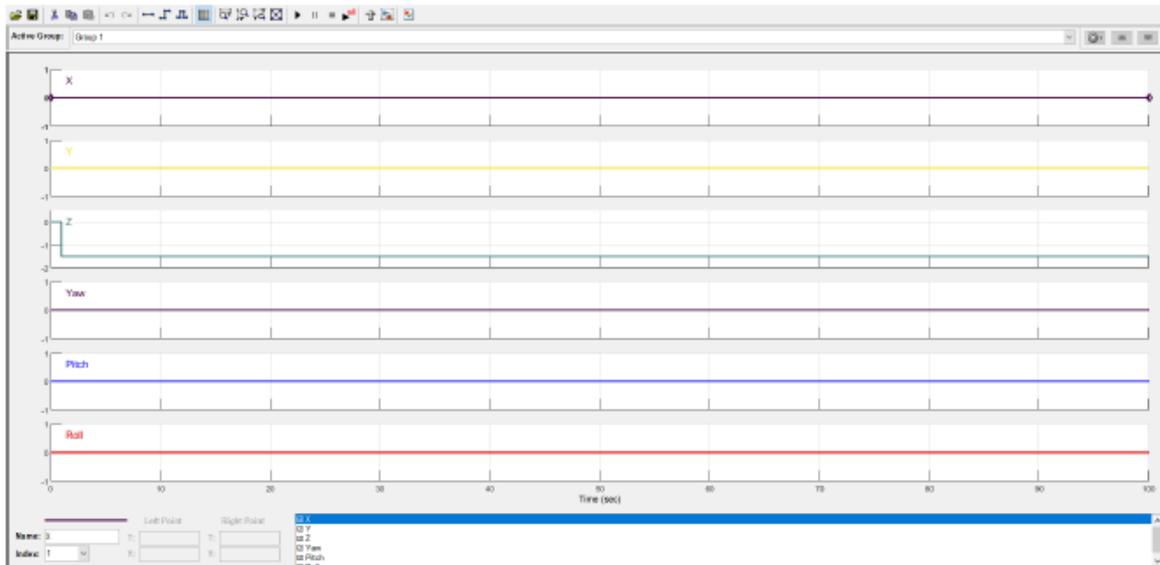


Figura 22 : Segnali di riferimento

Il signal Builder permette di generare qualsiasi tipo di segnale da dare in ingresso al nostro sistema e quindi verificare il comportamento e il controllo di quest'ultimo.

2.5 Controllo

Il blocco che si occupa del controllo del nostro sistema e nel quale abbiamo lavorato per l'implementazione del controllore è il *Flight Control System* (Figura 10). Nel seguente blocco troviamo tutta la parte che si occupa del controllo del drone cioè la parte che in fase finale del progetto, verrà compilata e trasformata in codice binario dal calcolatore e implementata direttamente nel drone fisico.

Esso riceve in ingresso i valori di riferimento del Signal Builder e i valori delle variabili di stato provenienti dai sensori. In uscita invece troviamo i segnali di riferimento sugli attuatori che devono occuparsi del controllo, inoltre c'è un blocco "Flags" che è un blocco di sicurezza che nel caso di perdita di assetto del drone interrompe la simulazione immediatamente.

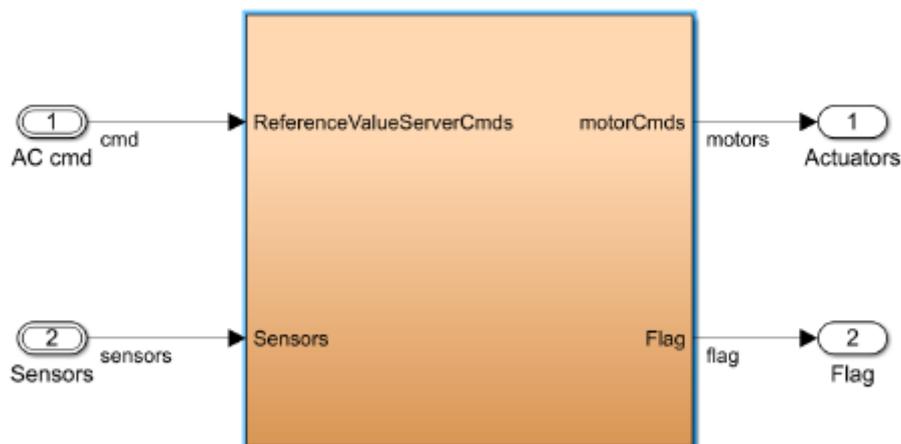


Figura 23 : Flight Control System

Il blocco FCS si suddivide internamente in 3 sotto-blocchi principali (Figura 24):

- Il blocco *State Estimator* riceve ed elabora i segnali provenienti dai diversi sensori presenti sul veicolo e fornisce tali informazioni al blocco *Flight Controller*.
- Il blocco *Flight Controller* effettua l'azione di controllo e fornisce in uscita i segnali di controllo diretti ai motori.
- Il blocco *Crush Predictor Flag* elabora le informazioni fornite direttamente dai sensori e dal blocco *State Estimator*, e determina confrontandole coi parametri base se ci sono eventuali problemi nella sessione di volo.

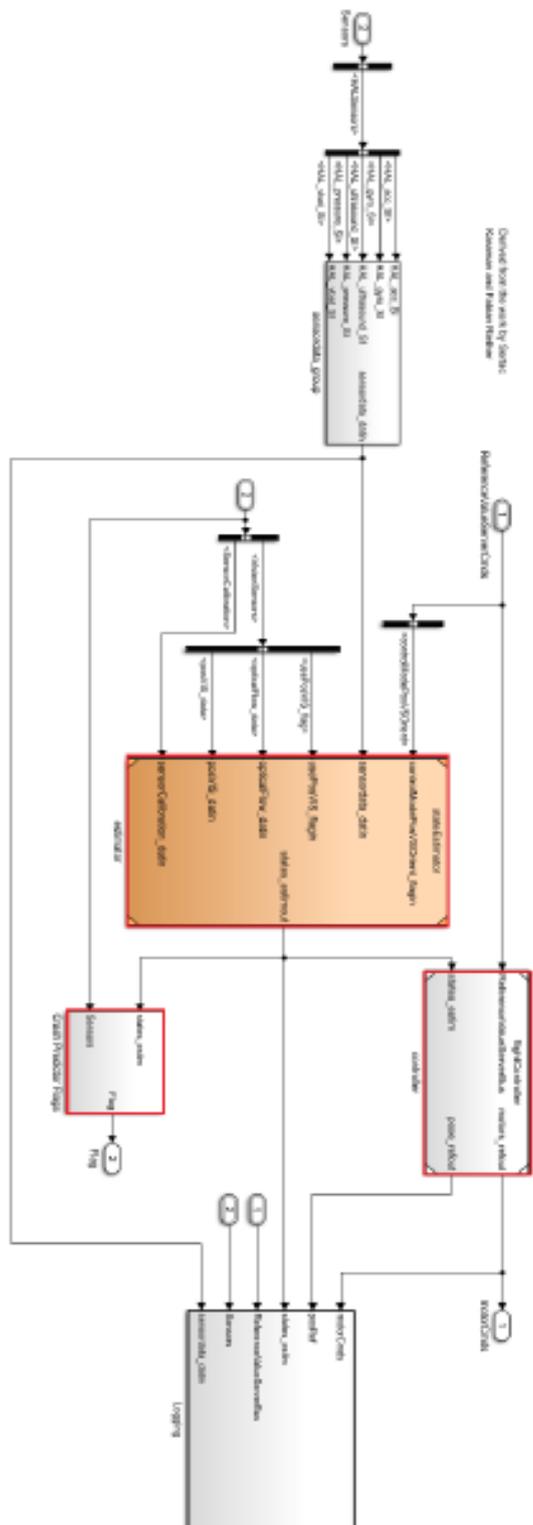


Figura 24 : Struttura FCS

Inoltre, all'interno di FCS sono presenti anche due sottosistemi ausiliari:

- *Sensordata group* : svolge una preelaborazione (ridenominazione delle variabili) sui dati forniti dai vari sensori montati sul veicolo (ad esclusione di quelli ottici);
- *Logging* : fornisce il valore delle variabili in ingresso e uscita dal blocco FCS. Crea dei file di log durante le simulazioni virtuali o reali sul drone. Questi file sono utili per verificare il corretto andamento desiderato post volo e per determinare eventuali cause di arresto.

Blocco State Estimator

è composto da un blocco principale (*Sensor Preprocessing*) che effettua il pre-processamento dei dati provenienti dall' IMU e dall'unità ottica. Insieme a questo blocco ci sono 3 blocchi secondari :

- *Complementary Filter* : elabora i dati provenienti dal sistema di pre-processamento e fornisce una stima dell'orientamento del sistema: in particolare calcola gli angoli di roll, pitch e yaw e le variazioni angolari (p ; q ; r) rispetto al *Earth Fixed Frame*.
- *Estimator Altitude* : elabora i dati provenienti dal sensore di pressione e dal sonar e, incrociandolo con i dati riguardanti l'orientamento del sistema, determina la quota e la velocità del veicolo.
- *Estimator X-Y position* : elabora i dati provenienti dal sensore ottico per determinare posizione e velocità longitudinale e latitudinale del drone.

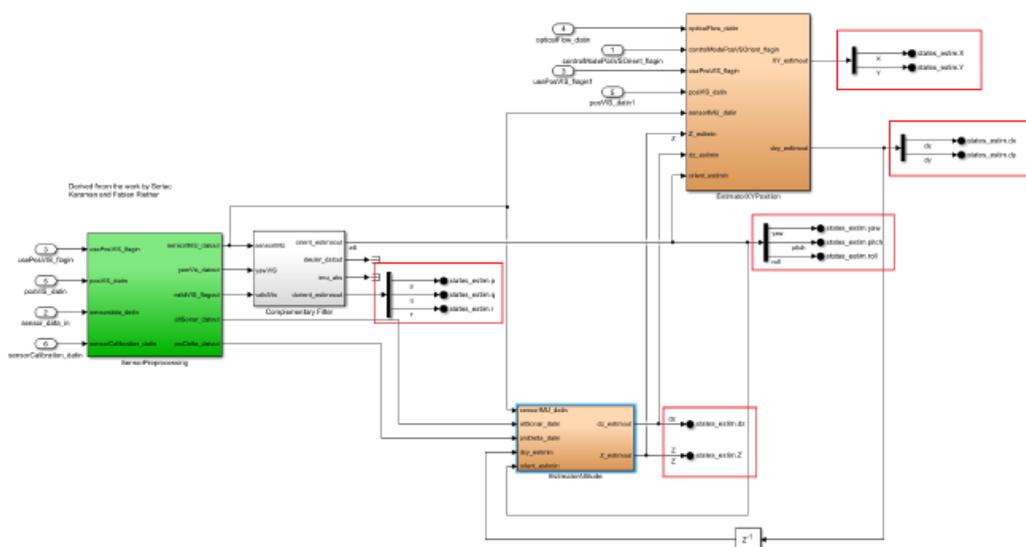


Figura 25 : Blocco State Estimator

Si ottengono 12 variabili (evidenziate in figura 25) rappresentano le variabili di stato del sistema che saranno fornite al blocco Flight Controller.

Blocco Flight Controller

Il blocco flight controller è costituito da 4 blocchi principali (Figura 26) e 3 sotto-blocchi ausiliari i primi si occupano di generare i segnali che andranno definire il comportamento e quindi il controllo del veicolo.

I Sotto-blocchi ausiliari invece sono lo *Switch Ref Att* , che permette di passare da segnale di riferimento rispetto alla posizione a segnale di riferimento rispetto all'orientamento e altri 2 blocchi che vedremo successivamente il *Control Mixer* e *Thrust to Motor Command* che fornisce alcune costanti relative agli attuatori che ci serviranno in fase di progettazione.

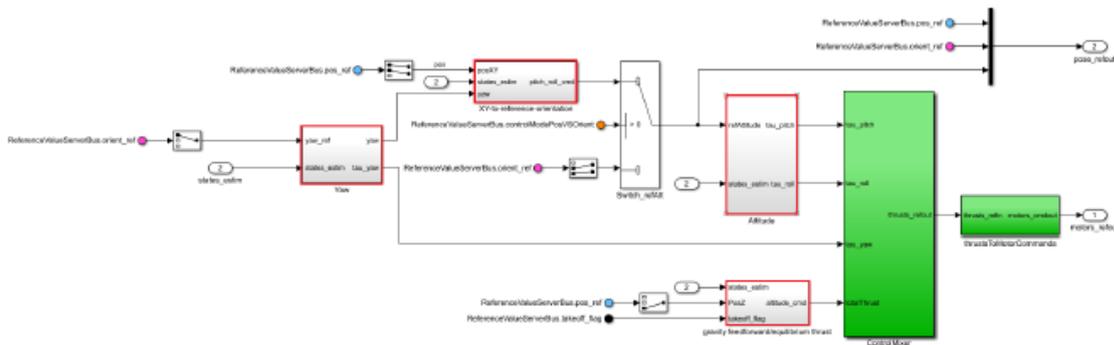


Figura 26 : Flight Controller

I sistemi evidenziati in rosso in figura sono quelli principali che si occuperanno del controllo del nostro veicolo, sono:

1. **XY to reference orientation** : riceve in ingresso il riferimento alla posizione del veicolo (longitudinale e latitudinale), l'angolo di yaw e il valore corrente della posizione e della velocità (longitudinali e latitudinali). In uscita elabora un segnale di controllo indicante il valore che devono assumere gli angoli di pitch e roll per mantenere la posizione di riferimento.
2. **Attitude** : riceve in ingresso il valore corrente degli angoli di pitch e roll, e delle relative velocità angolari, e, a seconda del valore dello switch, il segnale di riferimento relativo alla posizione oppure quello relativo all'orientamento. In uscita fornisce i segnali di controllo per gli angoli di pitch e roll
3. **Yaw** : riceve in ingresso il riferimento relativo all'angolo di Yaw, il valore corrente di tale angolo e la relativa velocità angolare. In uscita fornisce il segnale di controllo per l'angolo di Yaw.

4. **Gravity Feedrorward/Equilibrium Thrust** : riceve in ingresso il segnale di riferimento relativo alla quota (Z), il valore corrente di tale variabile, la velocità verticale del veicolo e il segnale TakeOff Flag che regola l'azione di decollo. In uscita elabora il segnale di controllo relativo all'altitudine del veicolo.

I sistemi sopra elencati sono quelli che si occupano del controllo dell'angolo di beccheggio e rollio (pitch and roll) quindi sono i sistemi che andremo a modificare per effettuare il controllo del nostro drone.

Di seguito analizzeremo ognuno di questi e vedremo quali sono il loro funzionamento e cosa controllano.

XY to reference orientation si compone di una parte superiore in cui, utilizzando l'angolo di yaw, si calcolano le componenti X, Y rispetto al sistema di riferimento inerziale dovute alla rotazione

del veicolo sull'asse Z (nei fatti si applicano le prime due righe della matrice R (Z; ϕ)).

Nella parte inferiore il valore così ottenuti vengono sommati agli errori sulla posizione longitudinale e latitudinale (ottenuti confrontandoli con quelli stimati di X e Y con i loro relativi riferimenti) e poi vengono moltiplicati con le opportune costanti. Nel complesso sommando i due rami si va a formare un controllore PD (Proporzionale Derivativo).

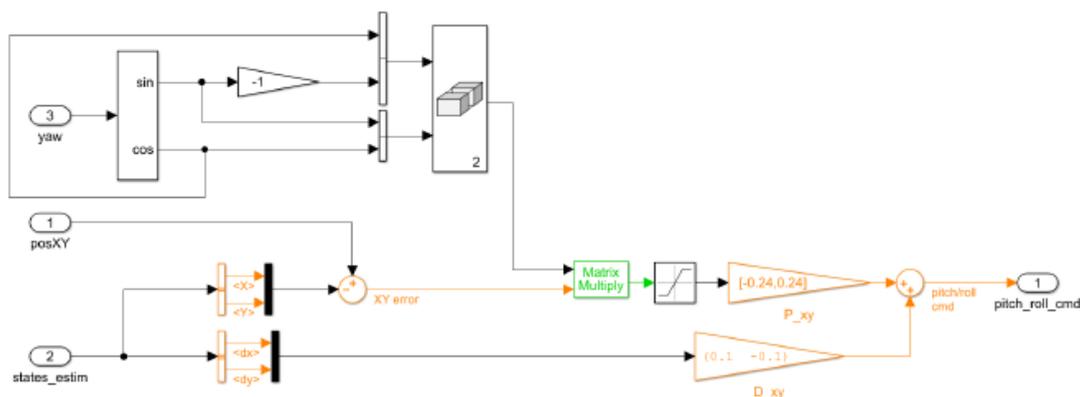


Figura 27 : Blocco X-Y to Reference Orientation

Il Blocco **Attitude** riceve in ingresso a seconda di come è impostato lo switch precedente a esso, o riceve il segnale dal Signal Builder allora in quel caso abbiamo un controllo per orientamento o riceve il segnale dall' XY orientation allora abbiamo un controllo per posizione. Per questo motivo all'interno del blocco attitude il segnale è chiamato *Ref Attitude*.

All'interno di questo blocco entra un doppio segnali corrispondente agli angoli di pitch e roll, che viene confrontato con valori derivanti dallo State Estimator, in modo da ricavare l'errore rispetto al riferimento. Questi valori alimenteranno un blocco proporzionale e un blocco integratore, a questi valori vengono sommati le velocità p e q che prima vengono moltiplicati per opportune costanti poi aggiunti tramite un nodo sommatore. Si ottiene un controllore PID (Proporzionale Integrativo Derivativo).

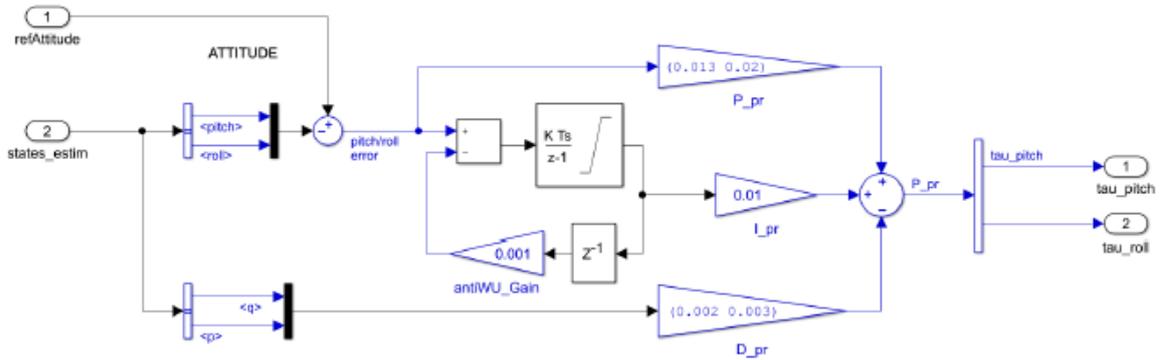


Figura 28 : Blocco Attitude

Il blocco **Yaw** è un controllore Proporzionale Integrativo con una struttura del tipo Derivative Output of Controller dove la parte derivativa non viene eseguita sull'errore ma sull'uscita, in questo caso r . Inoltre, la parte derivativa entra in contro reazione.

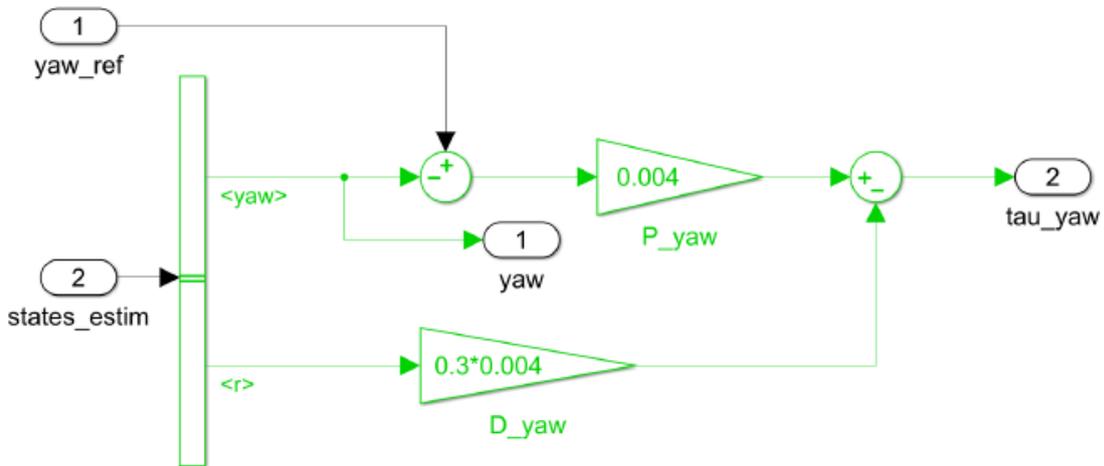


Figura 29 : Blocco Yaw

Gravity Feedforward/Equilibrium Thrust Contiene il controllore che agisce sulla z . Qui troviamo un controllore di tipo PD (Proporzionale – Derivativo). Si nota la presenza di uno switch il quale ha il compito, durante il decollo, di fornire una potenza maggiore agli attuatori e di isolare il controllore. Finito il decollo, il flag takeoff flag torna ad essere NULL e dunque lo switch seleziona il controllore. Utilizza il parametro TakeOffGain.

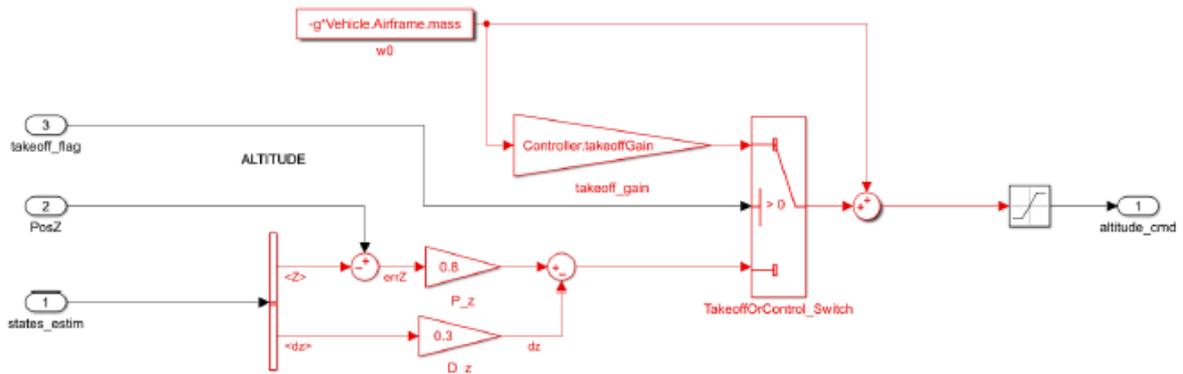


Figura 30 : Blocco Gravity Feedforward/Equilibrium Thrust

2.6 Sensori

Il Blocco Sensor si occupa della gestione e dell'implementazione dei vari sensori di cui è provvisto il drone in particolare, come abbiamo già visto in precedenza, i sensori del drone sono quello inserito sulla telecamera posizionata sotto il drone e l'IMU (Pressure).

- **Camera** : Sottosistema che si occupa di analizzare le immagini provenienti dalla fotocamera e in base, in realtà non è compito di Simulink gestirlo poiché viene gestito direttamente dal firmware fornito da Parrot.
- **IMU Pressure** : Sottosistema che si occupa della gestione di tutti i dati provenienti dagli altri sensori (Accelerometri, Giroscopi, Sensore di Pressione e Sonar).

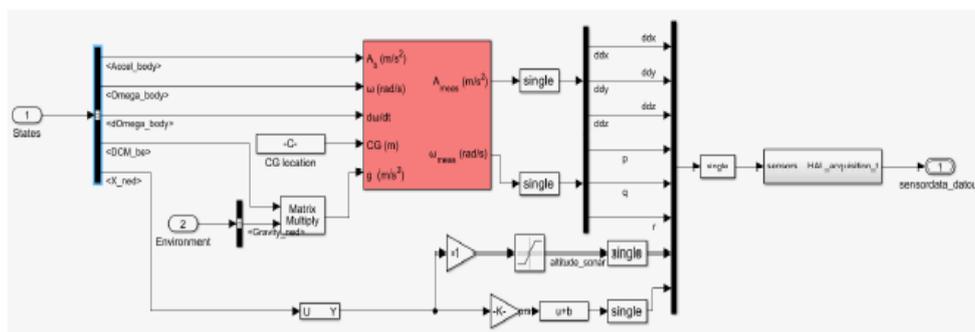


Figura 31 : Blocco IMU Pressure

Questo Ultimo sensore si occupa di trasformare i dati dei sensori forniti dal *Body Frame* in dati utili per l' *Earth Fixed Frame* .

Dati di accelerometri e giroscopi vengono elaborati sulla base di centro di gravità e azione gravitazionale si calcolano accelerazioni angolari e lineari del sistema di rif.

3 Controllore e Tecniche di controllo

3.1 PID Controller

Il controllo PID (Proporzionale Integrativo Derivativo) è uno dei più utilizzati nel settore del controllo industriale poiché facilmente realizzabile e molto affidabile.

È un sistema in retroazione negativa che molto spesso viene utilizzato nella versione PI (cioè una versione in cui il termine derivativo non viene CIRO utilizzato). Grazie a un input che determina il valore iniziale, è in grado di reagire a un eventuale errore, positivo o negativo, tendendo verso il valore 0. La reazione all'errore può essere regolata e questo rende il sistema molto versatile.

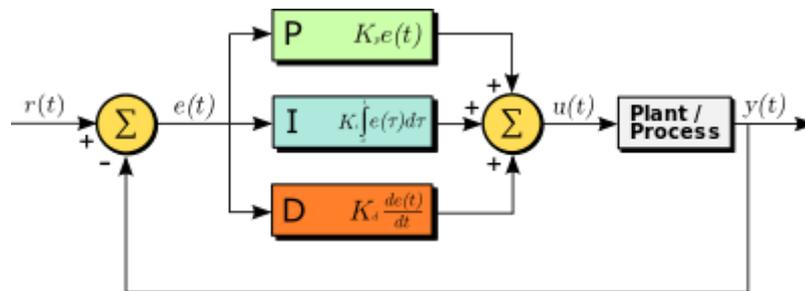


Figura 32 : PID Controller

Il controllore acquisisce in ingresso un valore dal processo e lo confronto con un valore di riferimento. La differenza, il cosiddetto segnale d'errore, viene quindi usato per determinare la variabile d'uscita del controllore, che è la variabile manipolabile dal processo.

Il PID regola l'uscita in base a 3 componenti:

1. Il valore del segnale d'errore (Azione Proporzionale)
2. Il valore passati dal segnale d'errore (Azione Integrale)
3. Quanto velocemente il segnale d'errore varia (Azione Derivativa)

I Controllori di questo tipo sono molto semplici da gestire e da settare, utilizzando le tecniche di Ziegler-Nichols, questi risultano buoni controllori per la stabilizzazione della maggior parte dei processi.

Hanno qualche limitazione sul fatto che non possono essere utilizzati se abbiamo parametri variabili, non sono molto stabili in alcuni casi a causa dell'azione integrativa che genera l'*Integral Windup*, inoltre le tecniche di Ziegler-Nichols per la progettazione non possono essere utilizzate in ogni condizione ma è necessario avere condizioni bene precise per l'uso.

Azione di Controllo di un PID

Le tre azioni di un PID vengono calcolate separatamente e semplicemente sommate algebricamente.

$$\mu = \mu_p + \mu_i + \mu_d \quad (26)$$

Azione Proporzionale

L'azione proporzionale è ottenuta moltiplicando il segnale d'errore "e" con un'opportuna costante.

$$\mu_p = K_p \cdot e \quad (27)$$

È possibile regolare un processo con simile controllore, anche processi instabili, tuttavia non è possibile richiedere che il segnale di errore "e" converga a 0 : questo perché è un'azione di controllo che è possibile solo se "e" è diverso da 0.

Azione Integrale

L'azione integrale è proporzionale all'integrale nel tempo del segnale d'errore "e", moltiplicato per una costante.

$$\mu_i = K_i \cdot \int e(t) dt \quad (28)$$

Questa definizione dell'azione integrale fa sì che il controllore abbia memoria dei valori passati del segnale d'errore; in particolare, il valore dell'azione integrale non è necessariamente nullo se è nullo il segnale d'errore. Questa proprietà dà al PID la capacità di portare il processo esattamente al punto di riferimento richiesto, dove la sola azione proporzionale risulterebbe nulla.

Azione Derivativa

Serve a migliorare le prestazioni di controllo:

$$\mu_d = K_d \cdot \frac{de}{dt} \quad (29)$$

L'idea è compensare rapidamente le variazioni del segnale di errore: se vediamo che “e” sta aumentando, l'azione derivativa cerca di compensare questa deviazione in ragione della sua velocità di cambiamento, senza aspettare che l'errore diventi significativo (azione proporzionale) o che persista per un certo tempo (azione integrale). L'azione derivativa è spesso tralasciata nelle implementazioni dei PID perché li rende troppo sensibili.

3.2 Analisi PID Mambo Parrot

Come abbiamo già detto in precedenza il Parrot “Mambo” implementa un controllo di tipo PID all'interno del modello. Nella sezione seguente andiamo ad analizzare il comportamento del nostro veicolo a seguito del controllo.

Per farlo è stato sufficiente inserire dei blocchi di tipo *Scope* all'interno del sistema ausiliario *Logging* all'interno dei quali è possibile consultare l'andamento delle varie variabili (X,Y,Z,Pitch,Roll,Yaw) coinvolte nel modello.

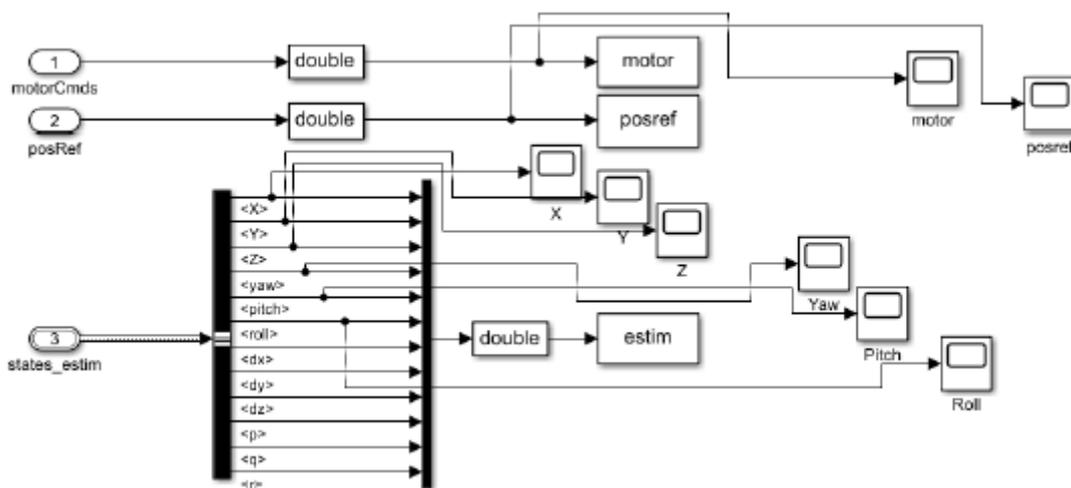


Figura 33 : Blocco Logging con Scope

Nella seguente sezione analizzeremo il modello del nostro sistema e i grafici ottenuti:

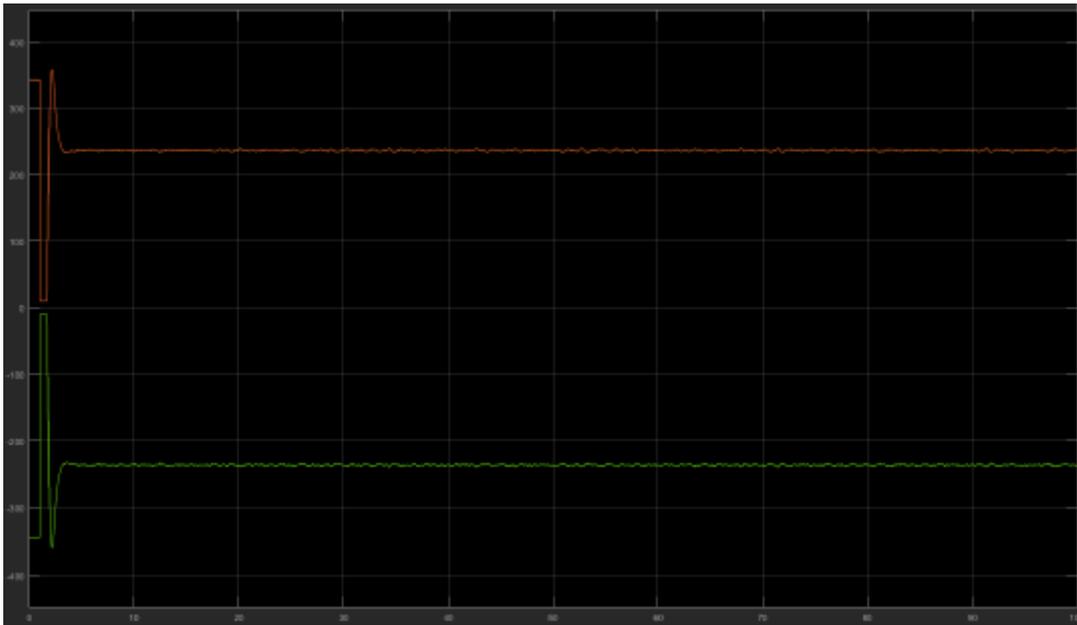


Figura 34 : Tracciato dei motori con Controller PID

Notiamo come i motori ricevano un segnale che li porta prima a una certa soglia per poi stabilizzarsi, questo è necessario affinché il drone raggiunga una certa quota.

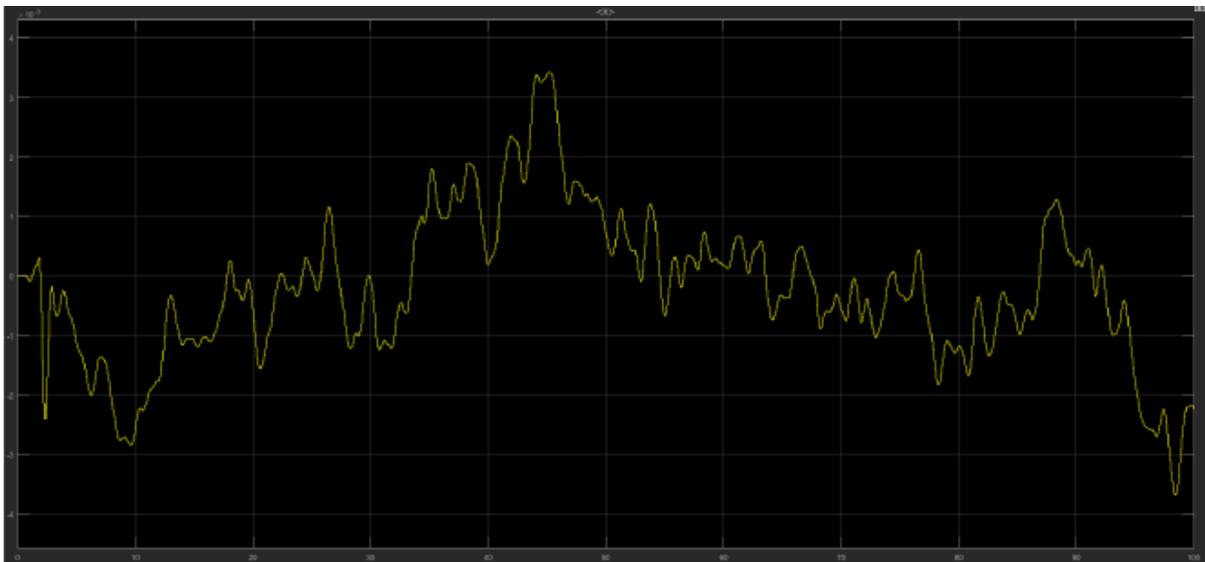


Figura 35 : Tracciato X con Controller PID

Notiamo che il controllo sulla posizione X garantisce delle oscillazioni limitate rispetto al riferimento, nell'ordine di 10^{-3} [m], questo testimonia la buona qualità dell'azione di controllo del PID del modello originale.

Analoghe considerazioni valgono per il controllo sulla posizione Y.

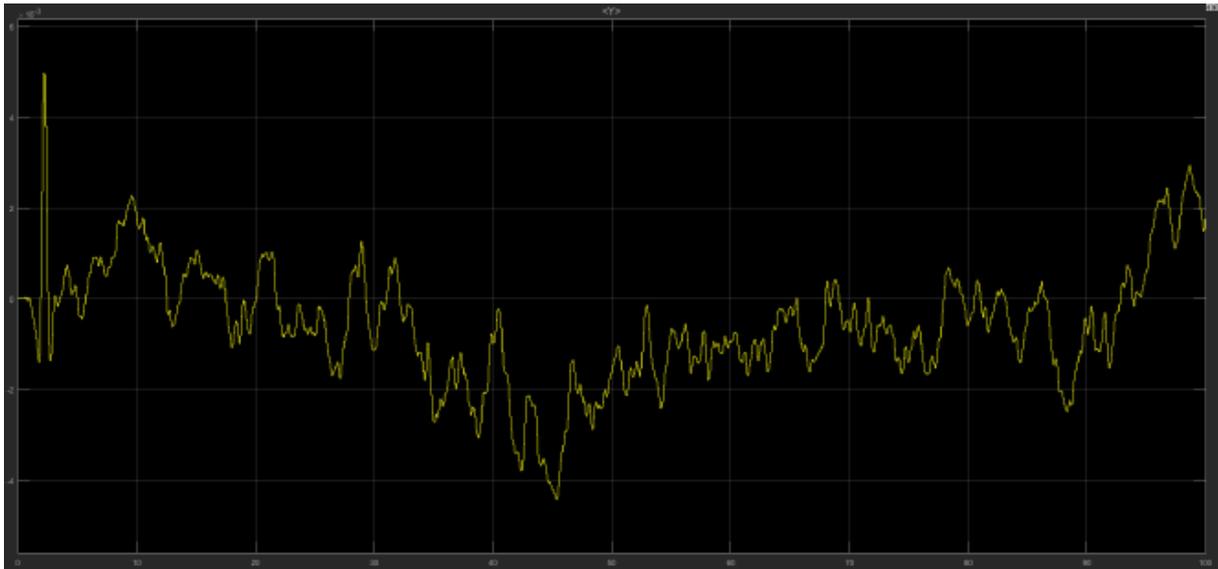


Figura 36 : Tracciato Y con Controller PID

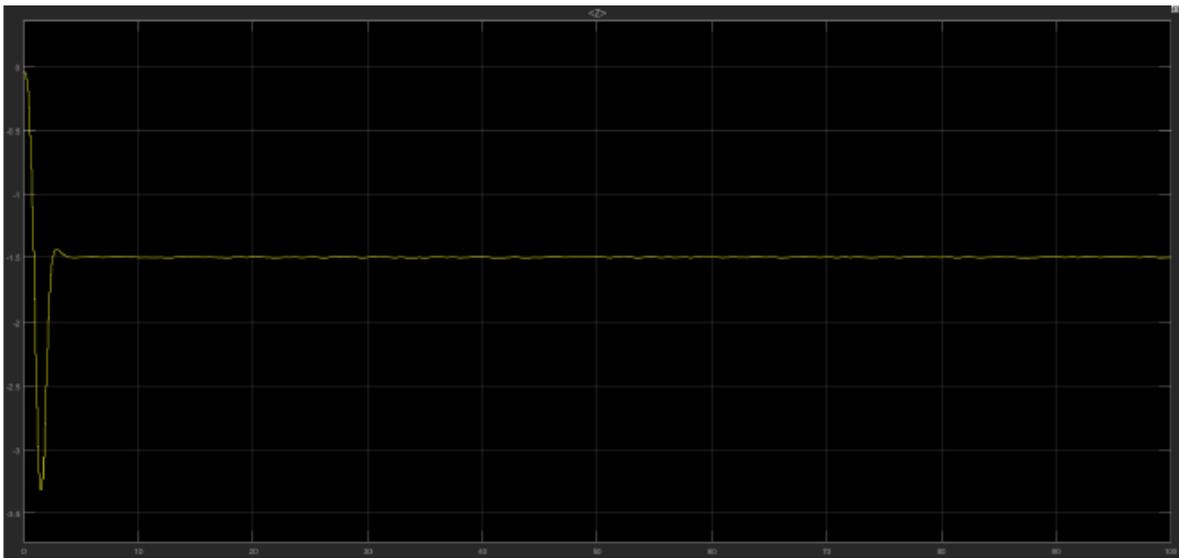


Figura 37 : Tracciato Z con Controller PID

La posizione Z si attesta, dopo una sovraelongazione iniziale, al valore di 1.5 m sul suolo. L'ordinata è negativa in quanto il sistema di riferimento è di tipo NED (North East Down).

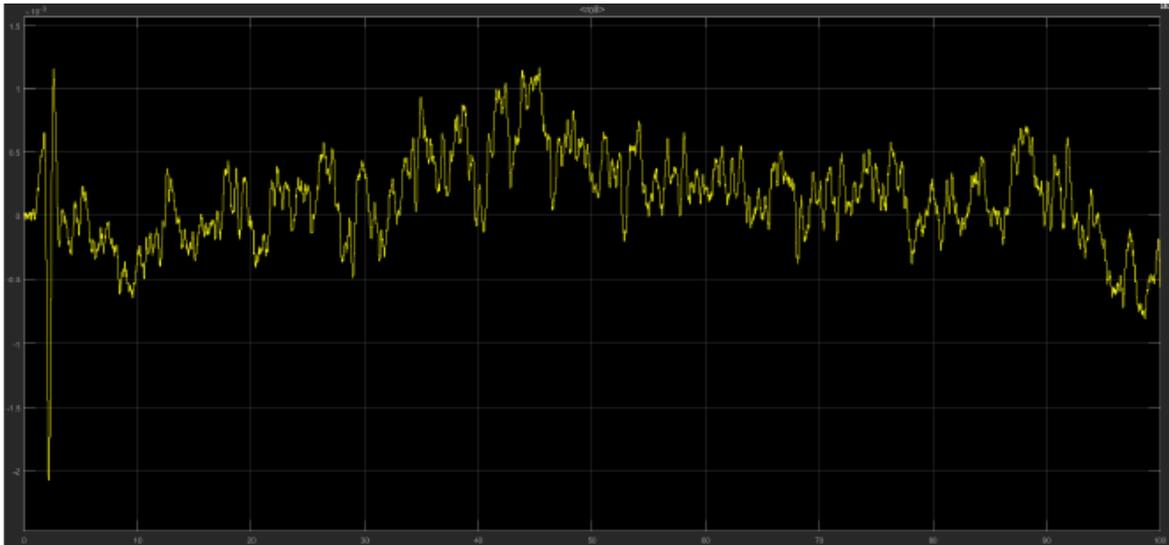


Figura 38 : Tracciato Roll con Controller PID

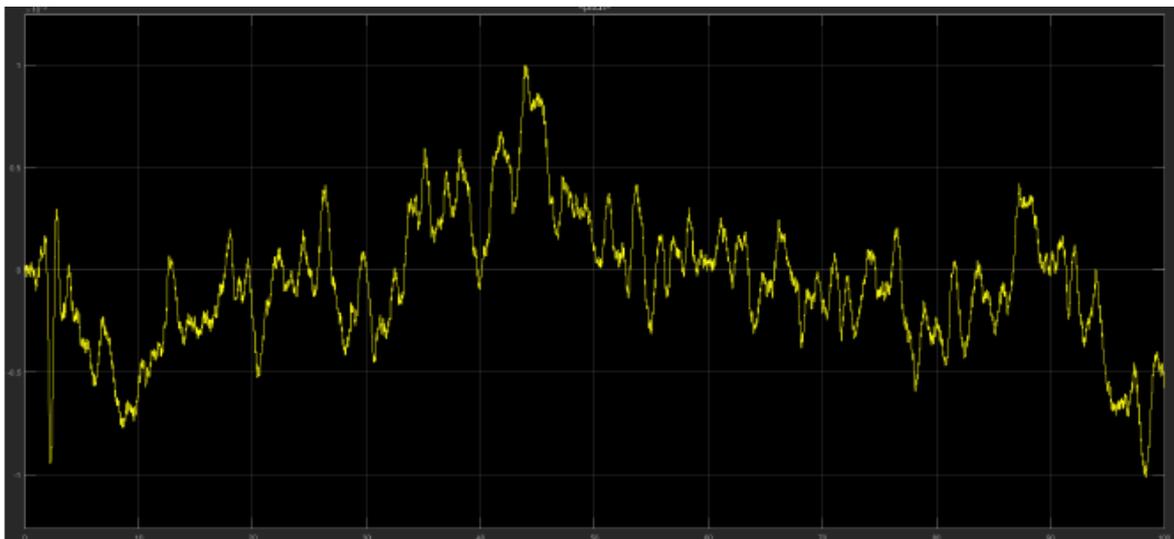


Figura 39 : Tracciato Pitch con Controller PID

Per gli angoli di Pitch e Roll facciamo le stesse considerazione fatte per XY, variazioni dell'ordine di 10^{-3} [m] testimoniano il buon funzionamento del controller originale.

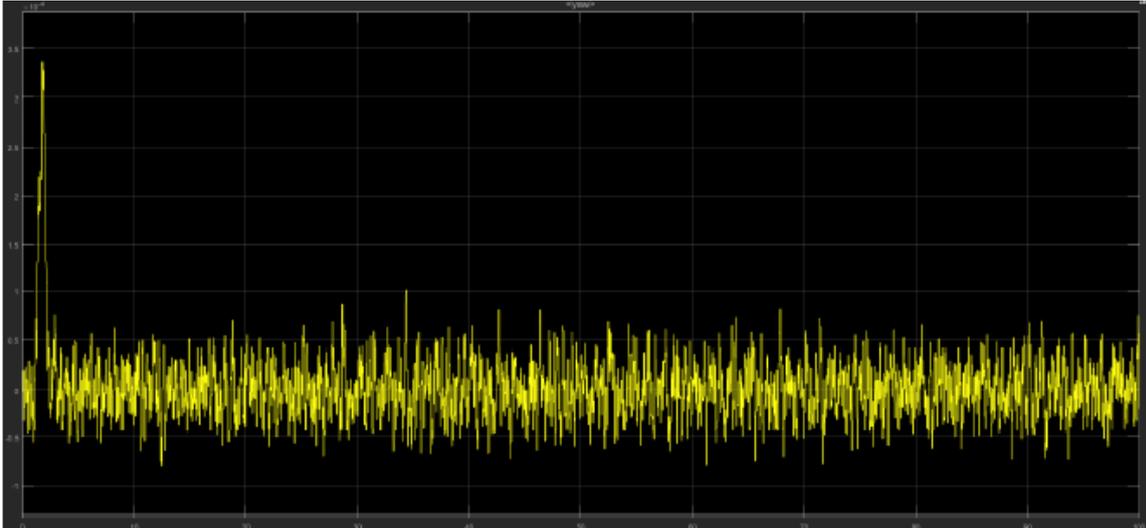


Figura 40 : Tracciato Yaw con controller PID

Il controllo per quanto riguarda lo Yaw come abbiamo detto anche precedentemente è direttamente collegato al controllo di angolo di Pitch e Roll, vediamo che è ancora più preciso, dopo una fase di assestamento iniziale si stabilizza in torno a valori che variano nell'ordine del 10^{-4} [m].

I grafici analizzati valgono sia per modello Lineare del nostro sistema che per quello Non-Lineare cosa che come vedremo successivamente non vale per il controller progettato.

Nel prossimo capitolo analizzeremo la tecnica utilizzata per costruire il controller, quella del “luogo delle radici”.

3.3 Tecnica del luogo delle radici

In analisi complessa il **luogo delle radici** è il luogo geometrico delle radici di una funzione complessa descritto al variare di un suo parametro reale (K'), rappresentato sul piano di Gauss

Si parte dalla funzione di trasferimento del nostro sistema e si scrive come funzione razionale di variabili complesse.

$$F(S) = K' \frac{\prod_{i=1}^m (s - z_i)}{\prod_{i=1}^n (s - p_i)} \quad (30)$$

Dove K' è il guadagno della nostra funzione mentre z_i e p_i sono gli zeri e i poli della nostra funzione.

Dalla (30) è possibile ricavare le radici dell'equazione algebrica ponendo quest'ultima uguale a 0:

$$\prod_{i=1}^n (s - p_i) + K' \prod_{i=1}^m (s - z_i) \quad (31)$$

Le radici di quest'equazione coincidono con i poli a ciclo chiuso della funzione di trasferimento a ciclo chiuso del sistema retroazionato :

$$W(s) = \frac{F(s)}{1 + F(s)} \quad (32)$$

L'utilizzo del luogo delle radici `e utilizzato, nello studio di un sistema di controllo a controreazione, per correlare alcune importanti caratteristiche della funzione di trasferimento a ciclo chiuso ai valori dei parametri che caratterizzano la funzione di trasferimento in catena diretta.

Il procedimento si basa nell'individuazione, esatta o qualitativa a seconda della complessità del caso, del luogo geometrico dei punti del piano complesso descritto dalle radici dell'equazione (43), al variare del parametro moltiplicativo K' .

Tracciamento :

Il metodo del luogo delle radici è un metodo grafico che ci permette quindi di individuare quali sono i poli o gli zeri che non soddisfano le specifiche richieste, e attraverso opportune modifiche permettere a questi zeri o poli non accettabili di entrare in condizioni di essere accettati.

È conveniente scrivere l'equazione (31) come :

$$\prod_{i=1}^n (s - p_i) = -K' \prod_{i=1}^m (s - z_i) \quad (33)$$

Per poi sostituire a quest'ultima la coppia di uguaglianza tra grandezze reali:

$$\prod_{i=1}^n |s - p_i| = |K'| \prod_{i=1}^m |s - z_i|$$

$$\sum_{i=1}^n \angle (s - p_i) = \pi + \angle K' + \sum_{i=1}^m \angle (s - z_i) + 2h\pi$$

La prima di queste viene detta condizione di modulo, la seconda condizione di fase.

Alla prima di tali condizioni corrisponde la proprietà che, in ogni punto del luogo, il valore del modulo di K' è pari al prodotto tra le lunghezze dei vettori che congiungono il punto stesso con i poli della $F(s)$ diviso il prodotto delle grandezze dei vettori che congiungono il punto con gli zeri della $F(s)$.

La seconda condizione ha un utilizzo analogo ma stavolta invece di prodotti compaiono somme e differenze, sta ad indicare che la somma degli argomenti dei vettori orientati dai poli della $F(s)$ verso un punto del luogo meno la somma degli argomenti dei vettori orientati dagli zeri della $F(s)$ verso lo stesso punto del luogo è pari a π o 0 . Il termine $2h\pi$ serve a distinguere se ci troviamo nel piano positivo o quello negativo.

È necessario definire alcune regole per il tracciamento:

- Il luogo delle radici è simmetrico rispetto all'asse reale.
- Per $k \rightarrow 0$ gli n poli della catena chiusa, tendono agli m poli in catena aperta.
- Tutto l'asse reale appartiene al luogo delle radici
- Appartengono al luogo positivo, quelle porzioni di asse che lasciano a destra un numero dispari di poli e/o zeri di $F(s)$, contati con la loro molteplicità. Gli altri punti appartengono al luogo negativo.
- I punti singolari sono radici multiple dell'equazione (31) in particolare sono quei punti che annullano la funzione e la sua derivata.
- Per $k \rightarrow \infty$ l'equazione (31) ha n soluzioni, essendo $n > m$ i rami del luogo partono da n poli. Inoltre, m rami tendono agli zeri mentre gli altri $n-m$ rami tendono $n-m$ semirette centrate nel centro degli asintoti s_o .

Il centro degli asintoti è individuabile attraverso la formula:

$$S_o = \frac{\sum_{i=1}^n P_i - \sum_{i=1}^m (Z_i)}{n-m} \quad (34)$$

4 Sintesi e implementazione controllore con Luogo delle radici

Nel seguente capitolo vedremo come è stato realizzato il controllore, per prima cosa è stato utilizzato il modello lineare del nostro veicolo implementato su Matlab, che ci ha fornito una panoramica semplificata sulla dinamica del modello e delle variabili di stato. Da questo è stato possibile estrapolare il sottosistema relativo all'angolo di pitch, sul quale è stato progettato un controllore che poi è stato inserito all'interno del blocco attitude, il modello lineare permette di trattare separatamente i legami tra le varie variabili.

Sono stati isolati gli ingressi e le uscite relative all'angolo di Pitch e in questo modo si è riuscito a trovare un legame ingresso/uscita dal quale poi abbiamo ricavato la funzione di trasferimento che ci è servita per costruire il controllore con il metodo del "Luogo delle radici", il controllore ha dovuto soddisfare determinate specifiche dette univoche e lasche.

Una volta realizzato il controllore è stato implementato in Simulink e dopo aver effettuato alcuni test, come vedremo nei successivi paragrafi, è stato verificato il suo funzionamento sia per modello lineare che per modelli non lineare.

4.1 Progettazione del Controllore

La creazione del controllore si divide in più fasi :

1. Disaccoppiamento Ingresso/Uscite modello lineare, per ricavare il sottosistema relativo all'angolo di Pitch e successivamente calcolo della funzione di trasferimento.
2. Luogo delle radici e calcolo della $G(s)$ del nostro controllore.
3. Implementazione controllore in Simulink.

Per effettuare il Disaccoppiamento Ingressi/Uscite è possibile utilizzare il modello lineare del nostro sistema (Linear Airframe), all'interno del quale sono contenute le matrici A,B,C,D che descrivono l'intero modello.

1. **A** : Matrice della dinamica degli stati 12×12 , racchiude tutte le variabili del sistema
2. **B** : Matrice degli ingressi del sistema 12×14

3. **C** : Matrice delle uscite del sistema 33x12, descrive le uscite del sistema rispetto agli stati.
4. **D** : Matrice 33x14 che descrive l'azione delle variabili d'ingresso sull'uscita.

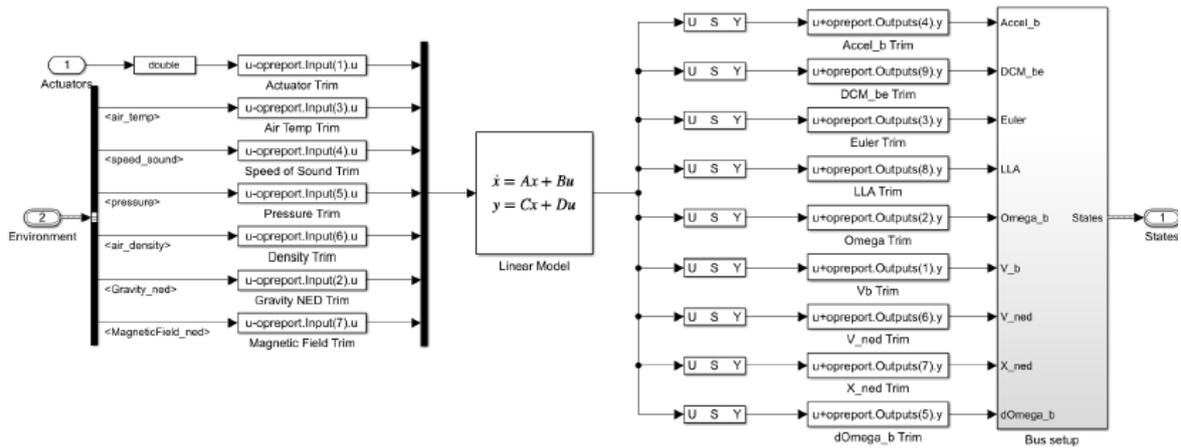


Figura 41 : Modello Lineare del processo

È stato possibile da queste matrici considerare soltanto la parte di sistema che descrive l'angolo di pitch, è stato sufficiente considerare :

- Nella matrice **A** i coefficienti relativi alle variabili di stato θ e q ;
- Nella matrice **B** i coefficienti relativi alle variabili di stato θ e q relative agli Actuators(1,2,3,4).
- Nella matrice **C** i coefficienti relativi all'uscita **Euler2** relativi all'angolo di Pitch.
- Nella matrice **D** i coefficienti relativi all'uscita **Euler2** legati agli attuatori(1,2,3,4).

Il sottosistema ottenuto è il seguente:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -1.692 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0.3133 \end{bmatrix}; \quad C = [1 \quad 0]; \quad D = [0]$$

Per ricavare la funzione di trasferimento dobbiamo ancora ricavare due costanti moltiplicative K1 e K2 che rispettivamente ci permettono di individuare, la prima la partizione del segnale di controllo tra i vari motori e la seconda la spinta dei motori.

La prima è contenuta nel blocco Control Mixer all' interno del FCS/Flight Controller e vale 5.6679, è ottenibile lanciando il comando *Controller.Q2Ts* , che ci fa ottenere in input una matrice che riguarda le costanti di controllo, i valori della 3 colonna di questa matrice riguarda l'angolo di pitch.

La seconda invece si trova all'interno del blocco ThrustToMotorCommands sempre all'interno del FCS e vale 1530,7.

Adesso abbiamo tutto il necessario per trovare la funzione di trasferimento, in Matlab è possibile calcolarla utilizzando la funzione tf.

In particolare, la formula per la fdt è :

$$FDT = C (sI - A)^{-1} B \quad (35)$$

Di seguito il codice Matlab da implementare per il calcolo di P(s).

```
>> A=[0 1;0 -1.619];
B=[0;0.3133];
C=[1 0];
D=[0];
syms s;
sys=ss(A,B,C,D);
Plant=tf(sys);
k1=5.6659;
k2=1530.7;
Ps=Plant*k1*k2

Ps =

      2717
-----
s^2 + 1.619 s
```

Una volta trovata la funzione di trasferimento, siamo passati al calcolo del luogo delle radici.

Il nostro sistema doveva soddisfare alcune specifiche che in parte sono state garantite dalla struttura di $P(s)$, esempio il fatto che il Sistema fosse di Tipo 1 (Cioè con un polo in $s=0$) era già garantita in partenza.

Mentre altre specifiche come l'errore a regime permanente $\tilde{e} \leq 0.01$ e che i poli in catena chiusa abbiano parte reale < -1 sono state soddisfatte in fase di realizzazione del controllore.

Per la prima specifica è stato sufficiente considerare un K abbastanza piccolo ($K \leq 0.037$) ed è stata soddisfatta, mentre per la seconda si è spostato il centro degli asintoti e di conseguenza sono stati calcolati i poli e gli zeri del controllore $G(s)$.

Con l'aiuto del ToolBox Sisotool è stato possibile già in un primo momento verificare il comportamento del nostro Controllore che è :

$$G(s) = 0.037 \frac{(s - 1.28)}{(s - 100)}$$

Sisotool ci fornisce una prima rappresentazione del luogo e la risposta del controllore rispetto a ingresso a gradino.

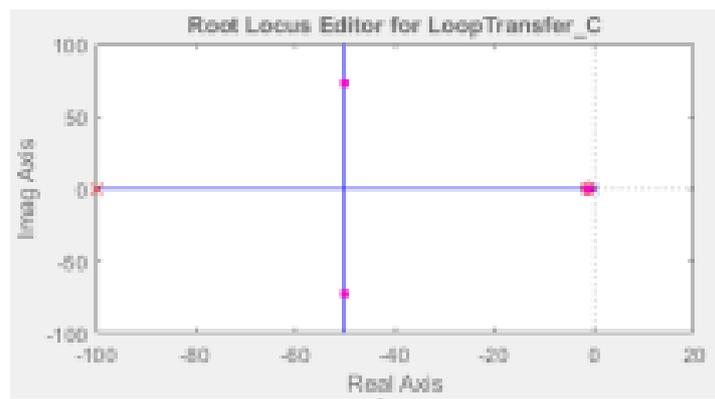


Figura 42 : Luogo delle radici

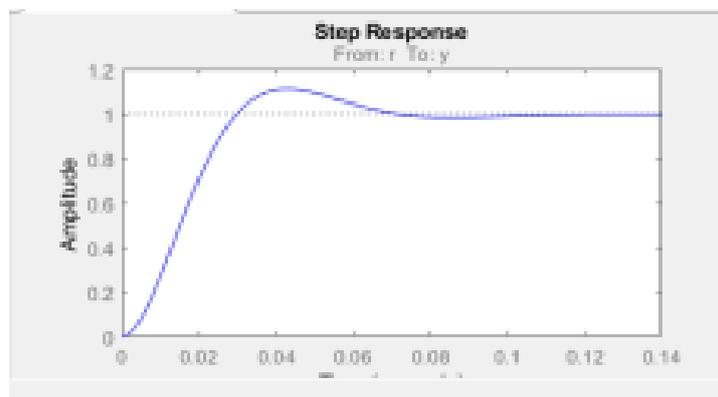


Figura 43 : Risposta al gradino

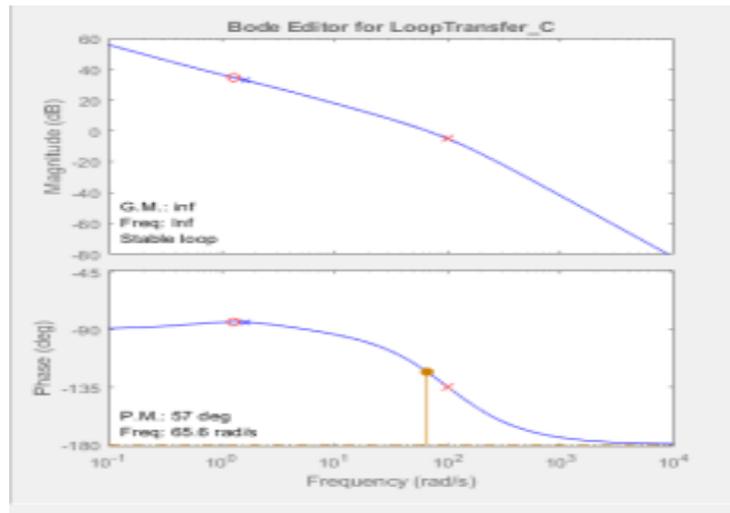


Figura 44 : Diagrammi di Bode

Una volta calcolato il controllore $G(s)$ è stato implementato nel modello Simulink. Inizialmente i sistemi di Pitch e Roll erano controllati entrambi da un controllore PID è stato necessario disaccoppiare il controllo di questi attraverso un demultiplexer e per quanto l'angolo di Roll è sempre il PID a controllarlo mentre l'angolo di Pitch viene controllato dal nostro controllore (Evidenziato in Blu in figura 45).

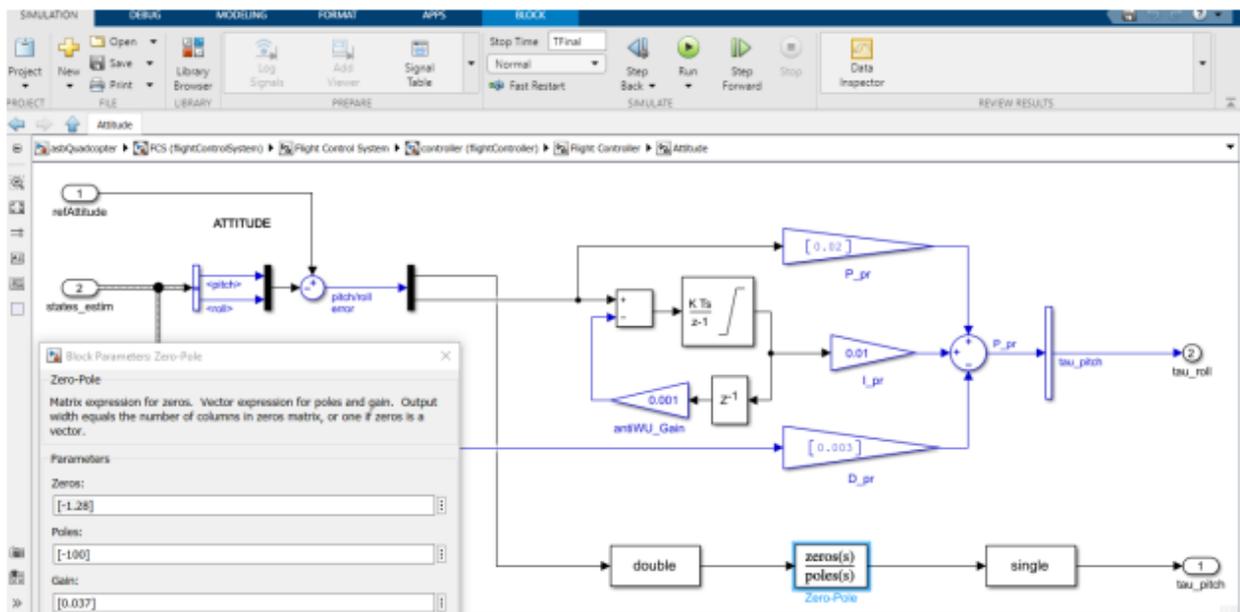


Figura 45 : Implementazione Controllore in Simulink

Da notare come l'ingresso Refattitude provenga dal blocco *Switch refAtt* che in base a come è settato decide se prendere in ingresso il segnale del XY orientation o il segnale dal Signal builder.

Questa scelta non è ininfluyente poiché se il segnale proviene dal blocco XY orientation che è anche lui un controllore si genera un controllo a cascata che genera un segnale variabile nel tempo, mentre se il segnale proviene dal Signal builder il segnale è un segnale costante nel tempo.

È possibile forzare il segnale del Signal Builder attraverso la porta logica AND di cui abbiamo parlato prima (Figura 21) e far scegliere al nostro sistema sempre un riferimento costante per *l'angolo di Pitch*.

4.2 Simulazioni del comportamento del controllore

I seguenti grafici faranno riferimento a un controllo effettuato con Modello Lineare e dando in ingresso al sistema un riferimento variabile.

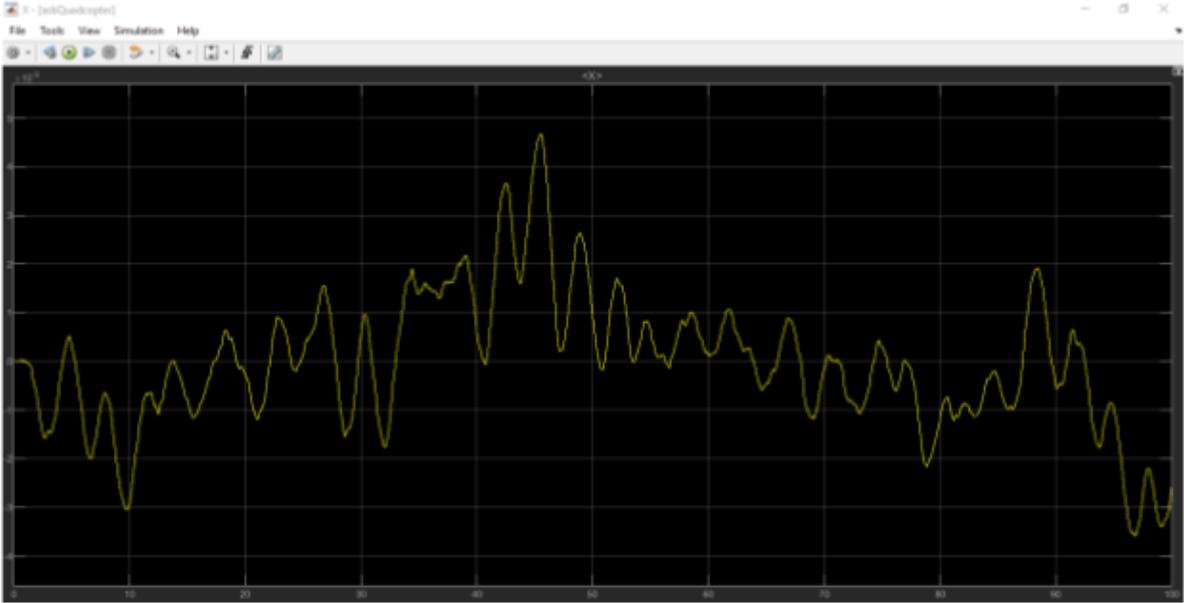


Figura 46 : Tracciato X

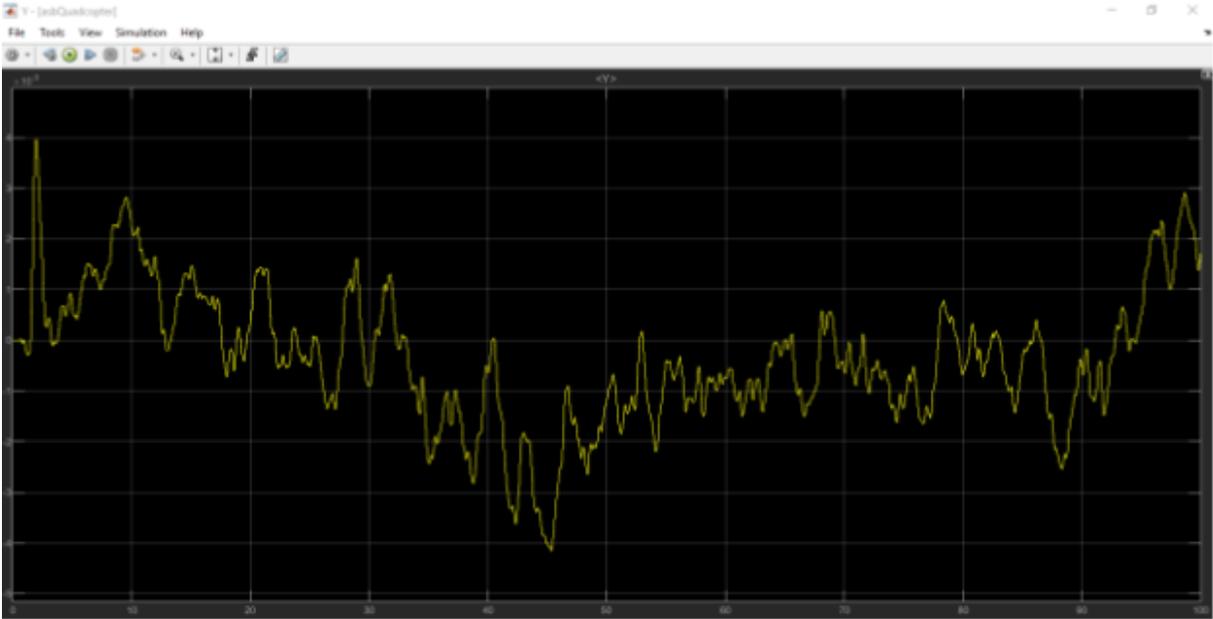


Figura 47 : Tracciato Y

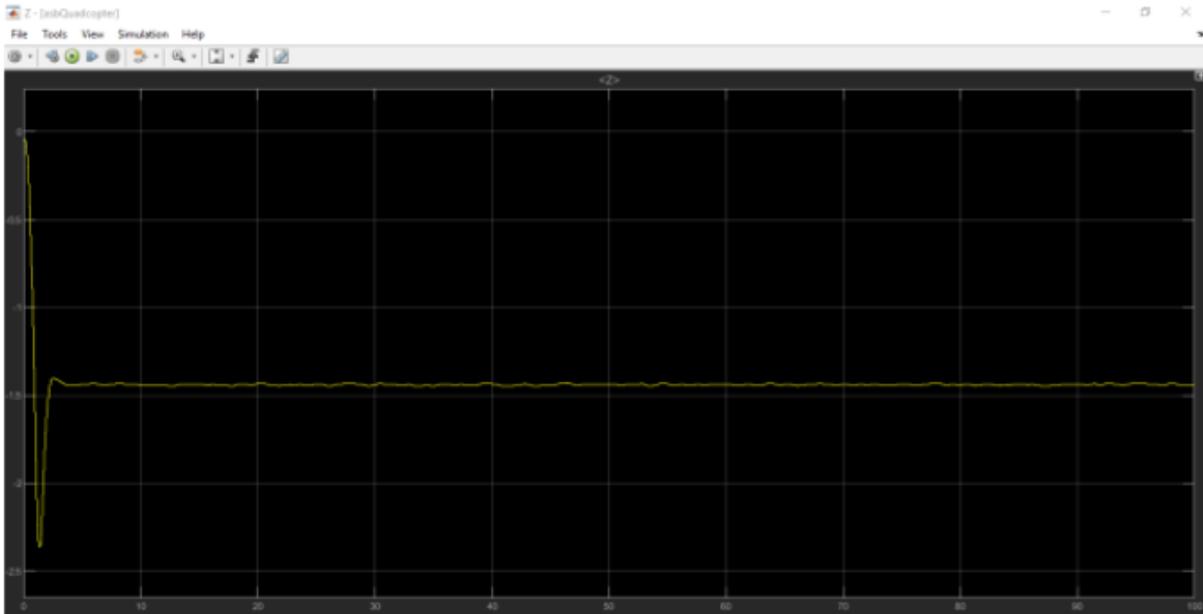


Figura 48 : Tracciato Z

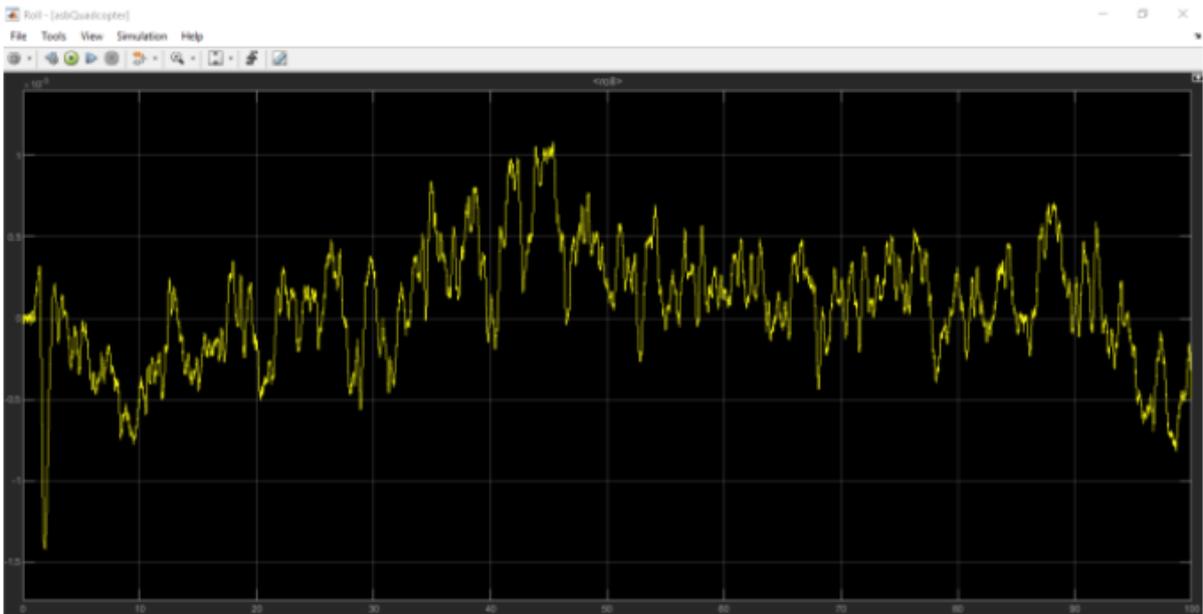


Figura 49 : Tracciato Pitch

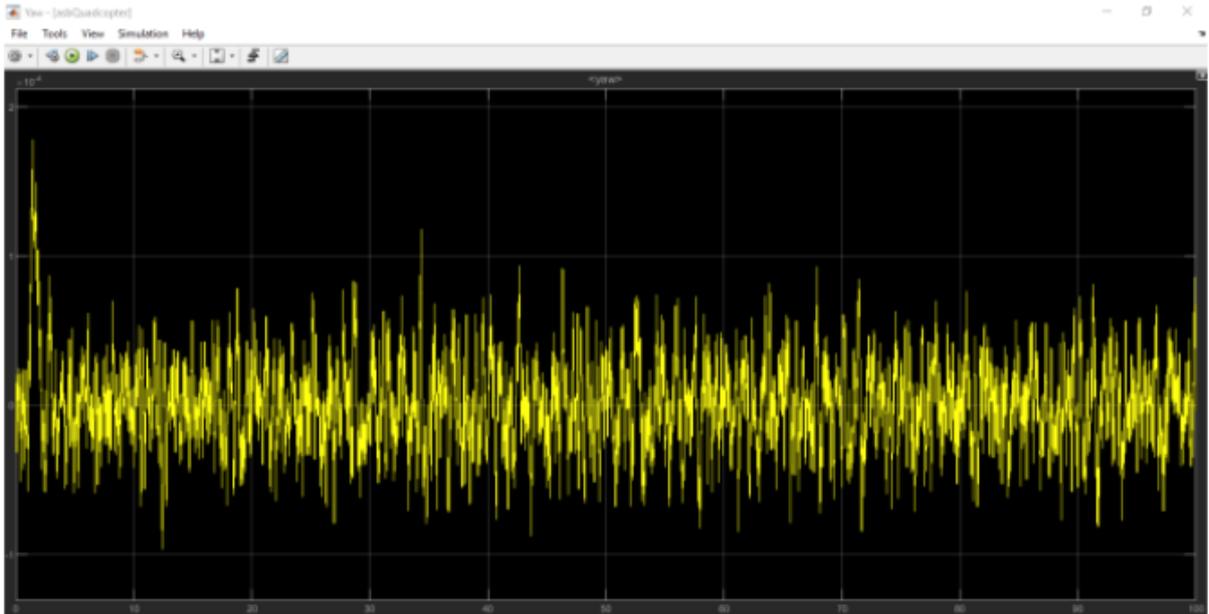


Figura 50 : Tracciato Yaw

I seguenti grafici faranno riferimento al Modello Lineare ma questa volta con un ingresso costante.

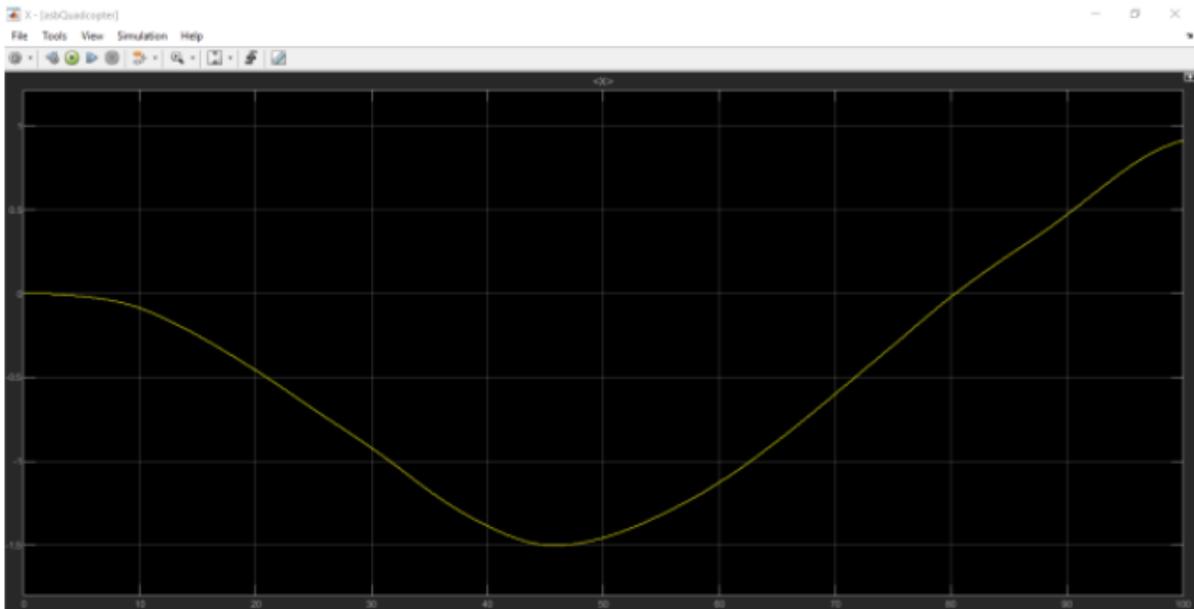


Figura 51 : Tracciato X

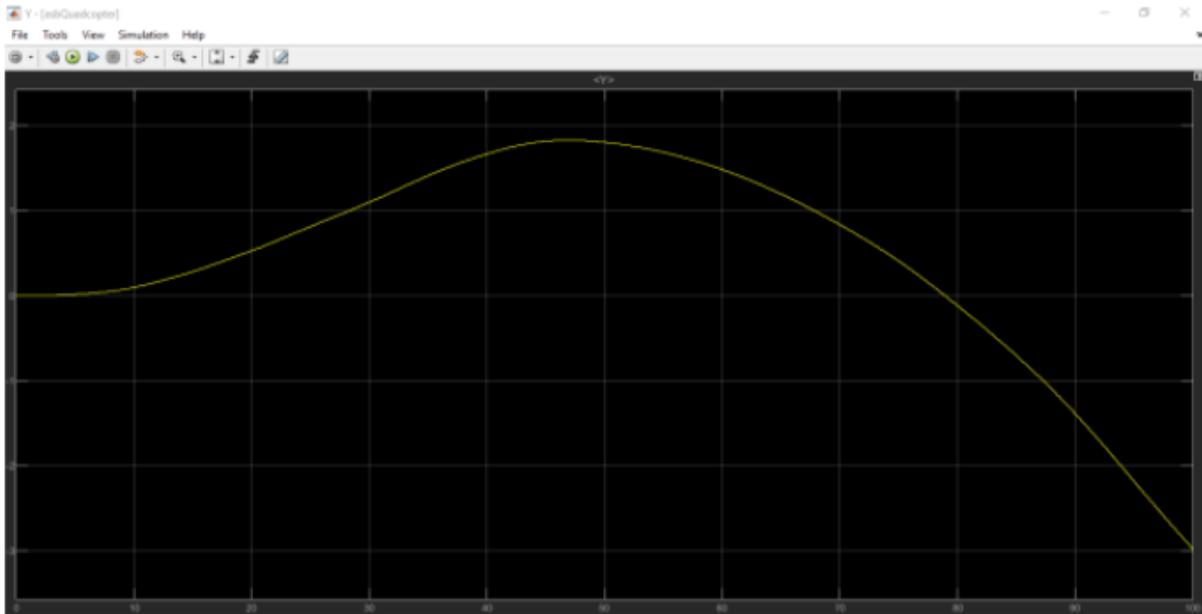


Figura 52 : Tracciato Y

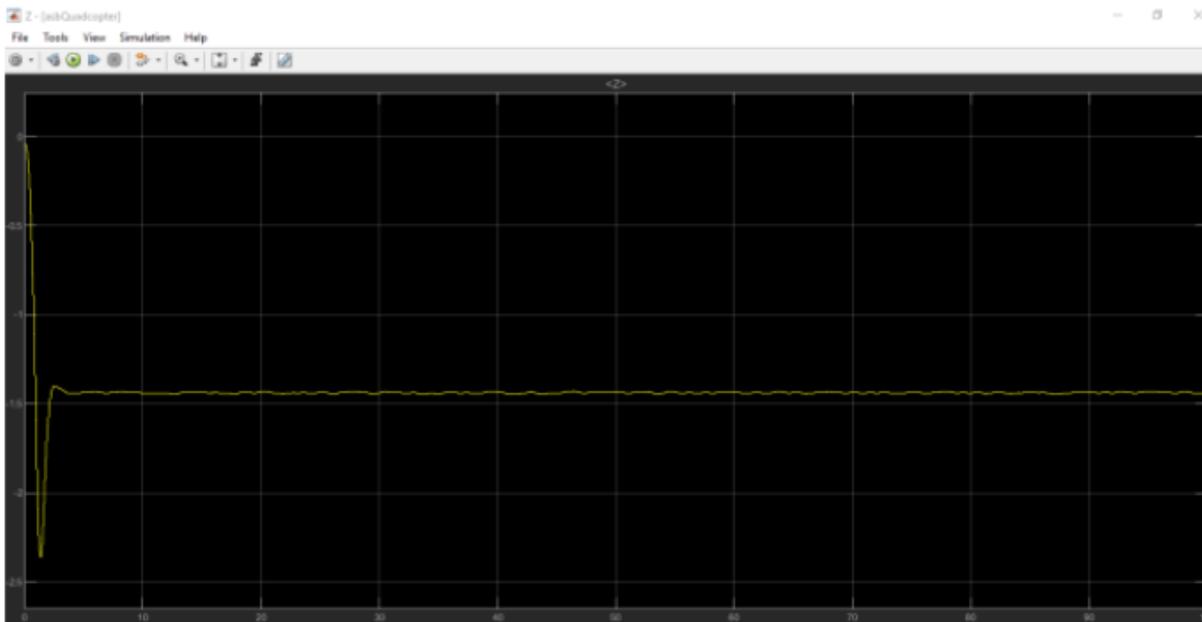


Figura 53 : Tracciato Z

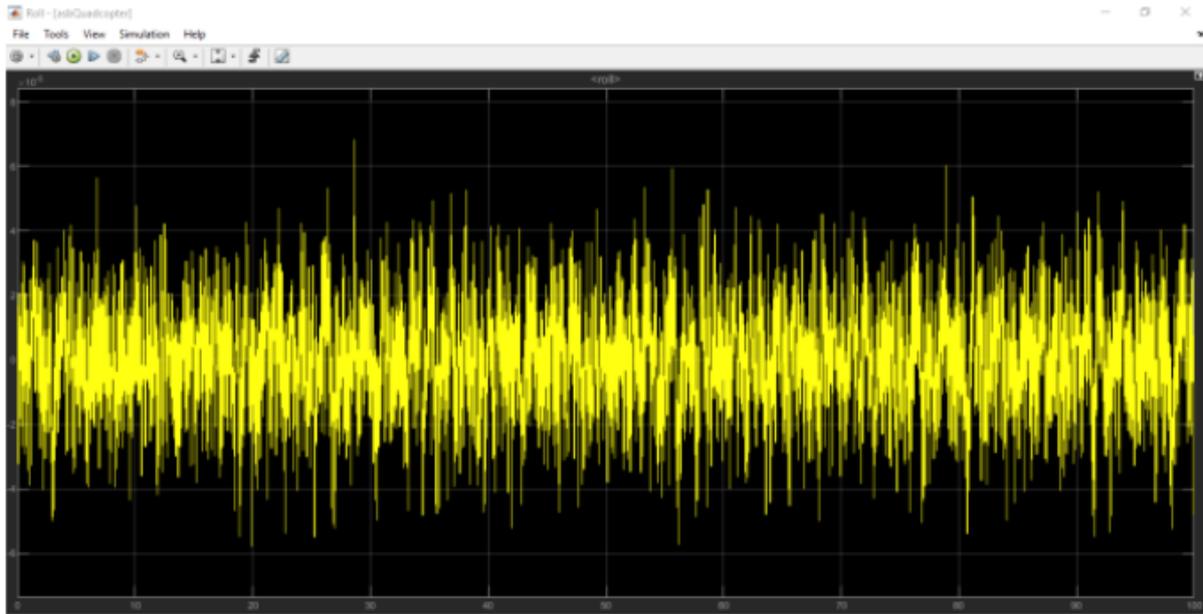


Figura 54 : Tracciato Roll

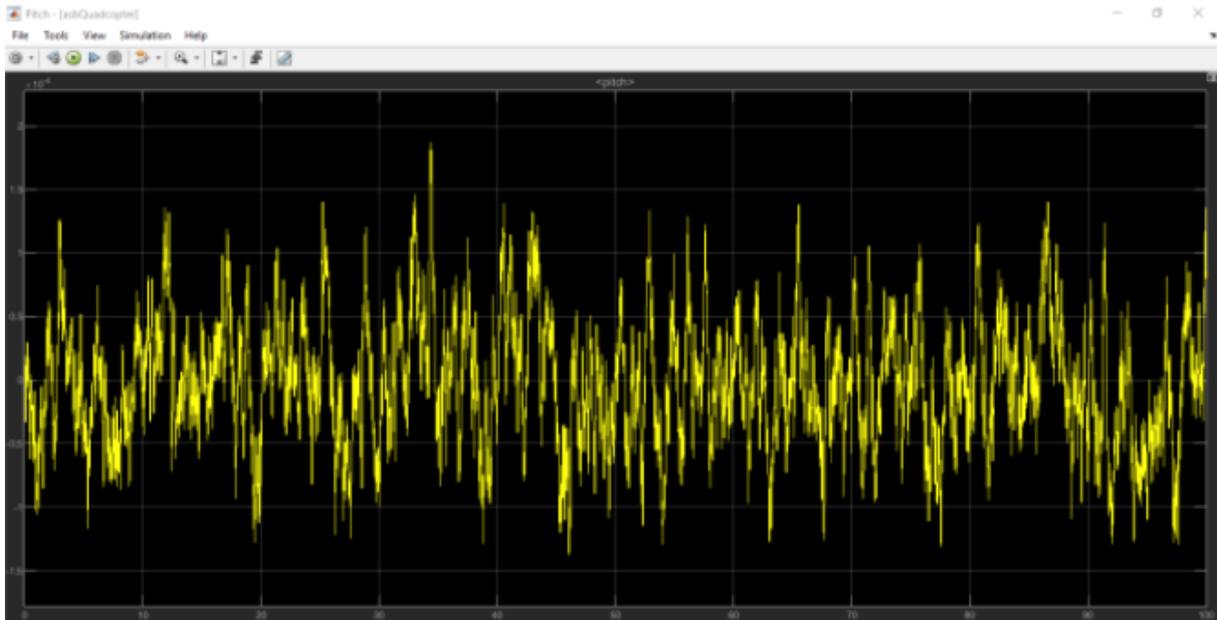


Figura 55 : Tracciato di Pitch

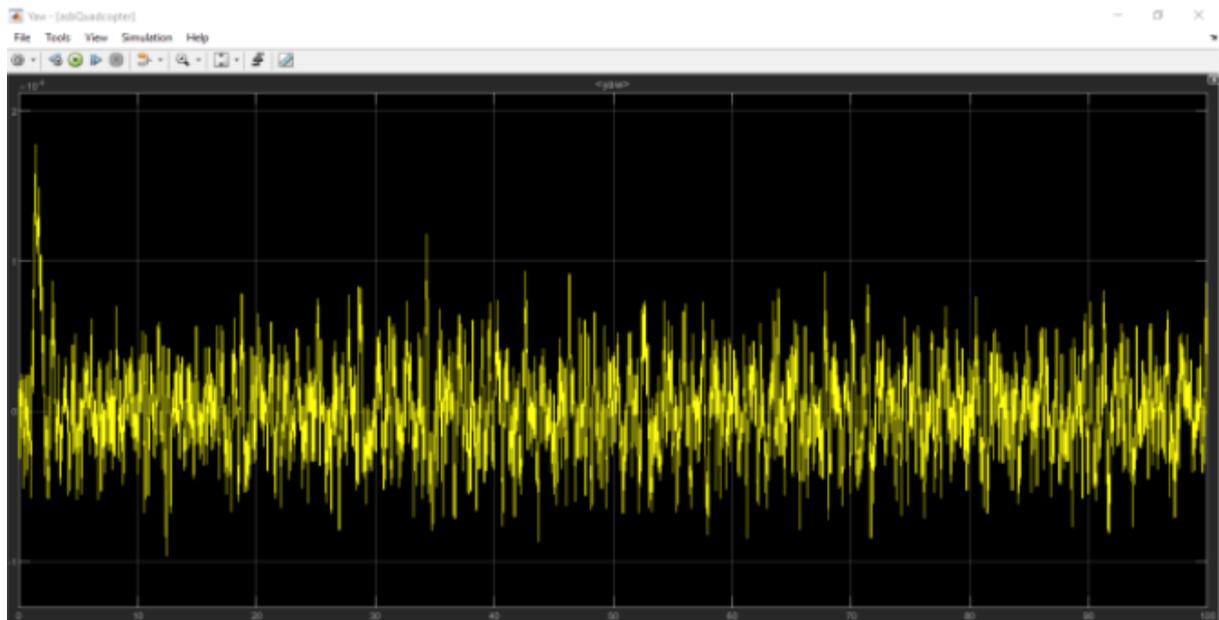


Figura 56 : Tracciato di Yaw

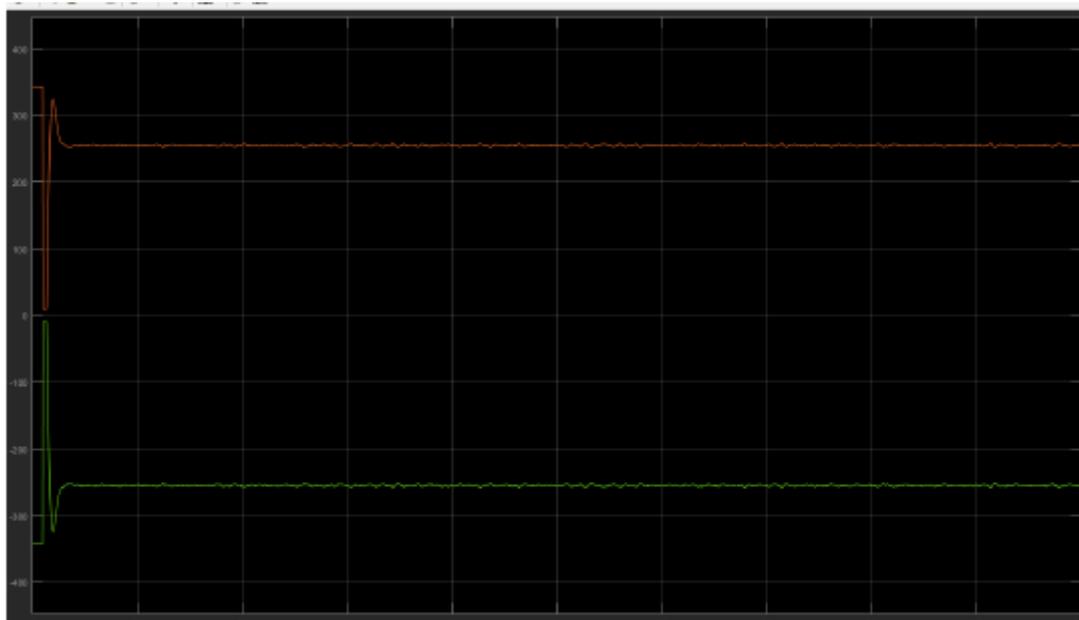


Figura 57 : Tracciato dei motori

Adesso invece passiamo ai grafici relativi al modello non lineare del nostro drone.

Grafici relativi al modello non lineare, con riferimento variabile

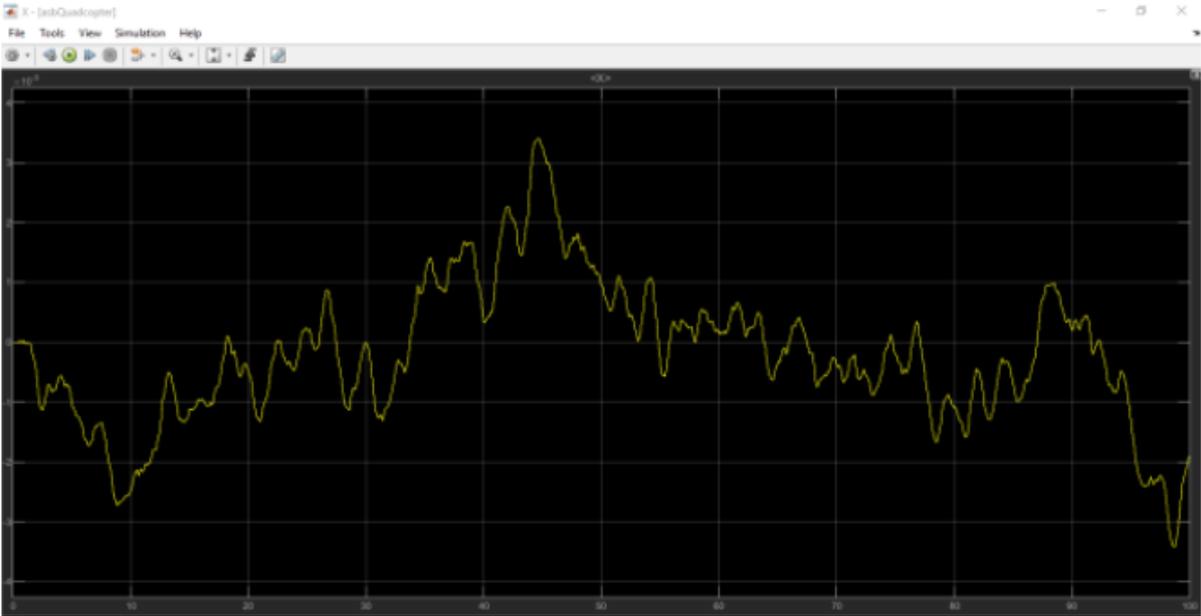


Figura 58 : Tracciato X

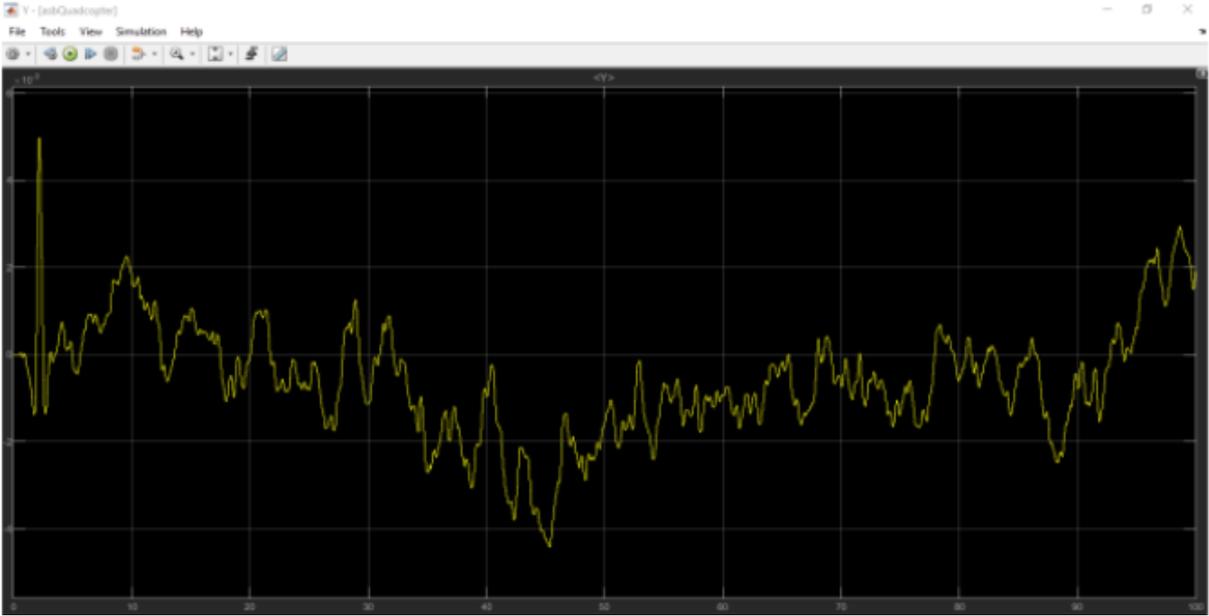


Figura 59 : Tracciato Y

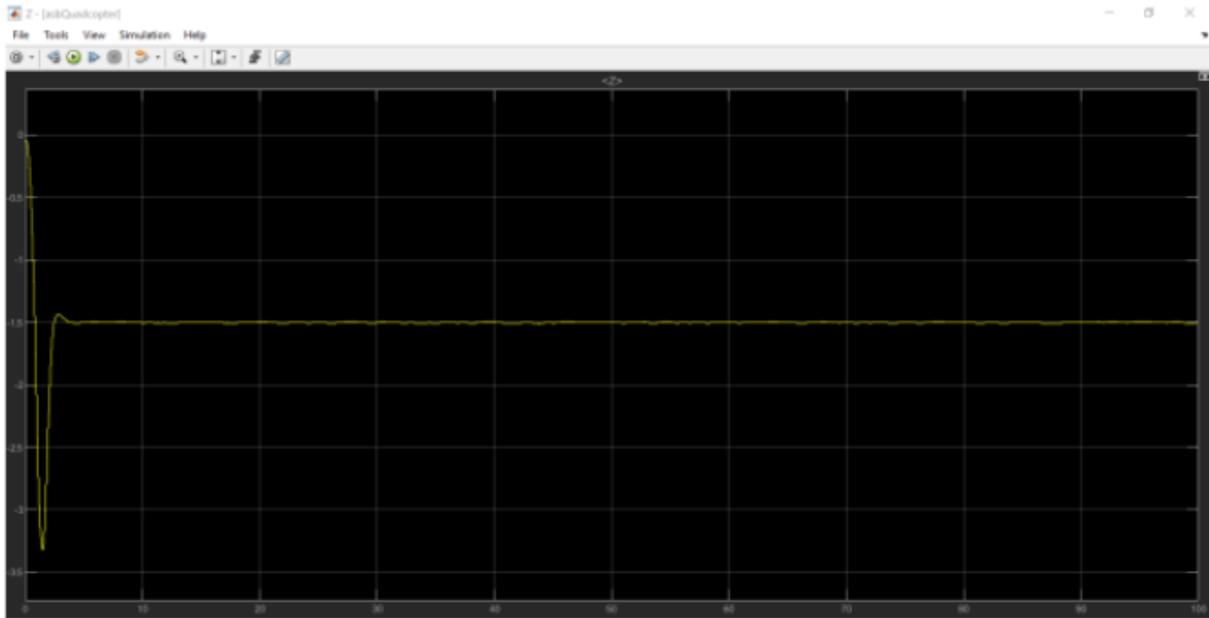


Figura 60 : Tracciato Z

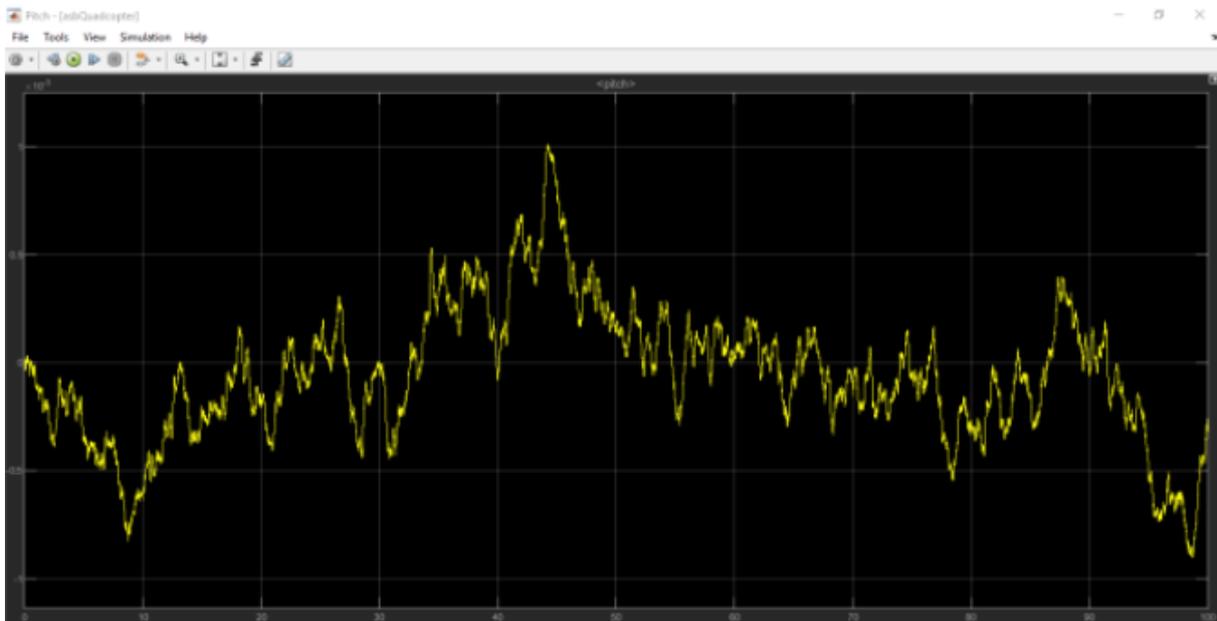


Figura 61 : Tracciato Pitch

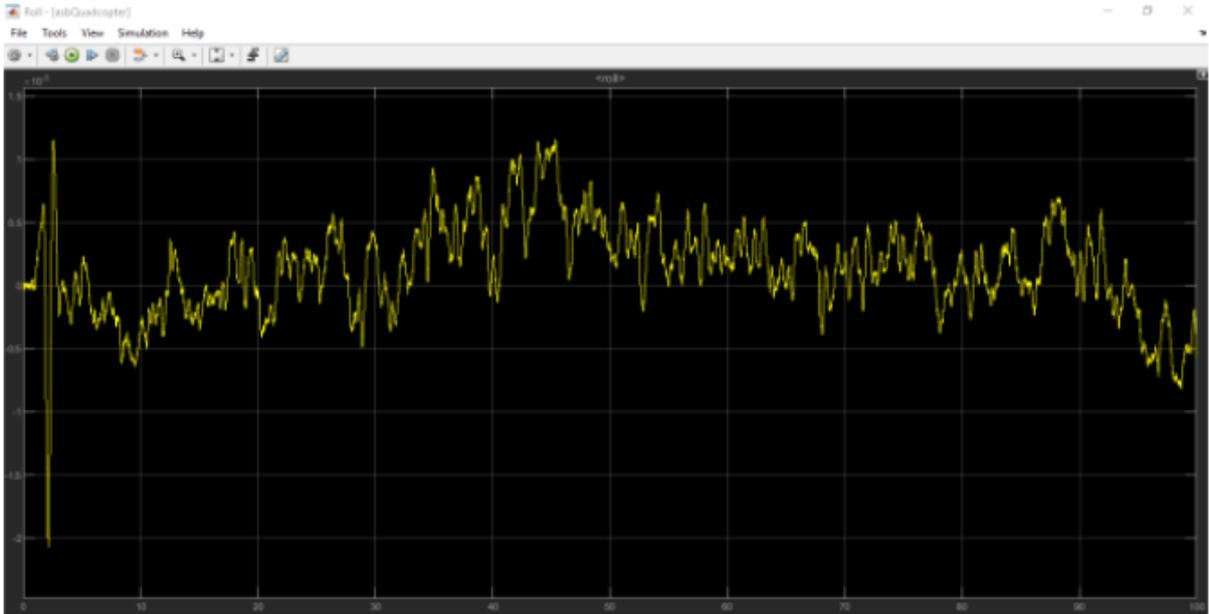


Figura 61 : Tracciato Roll

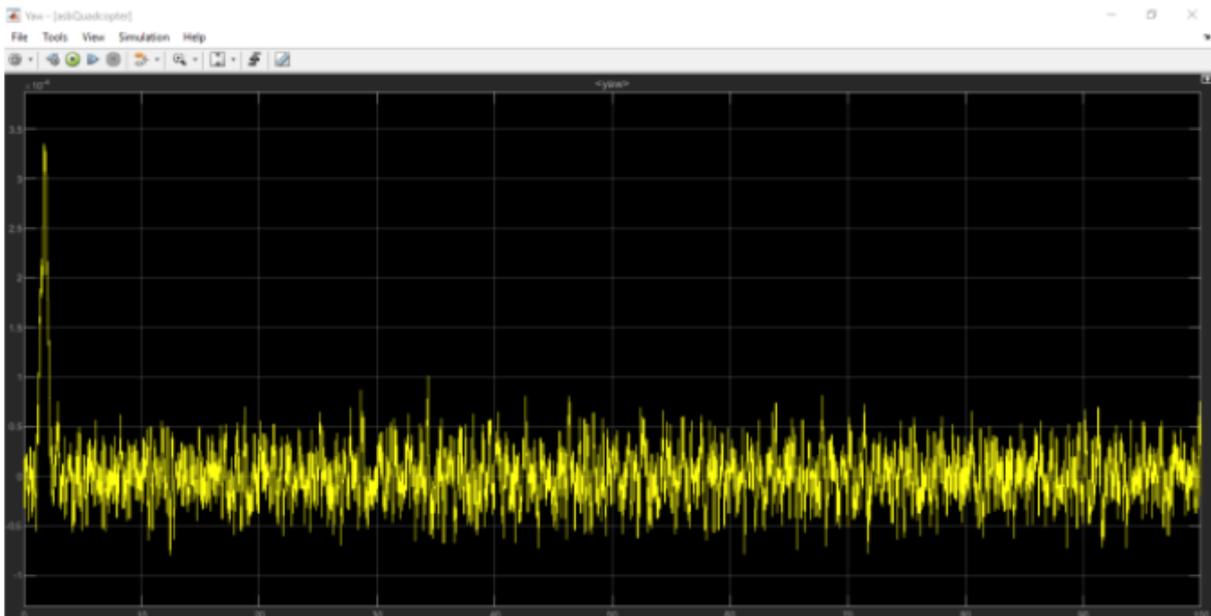


Figura 62 : Tracciato Yaw

Gli ultimi grafici riguardano il modello non-lineare con riferimento costante.

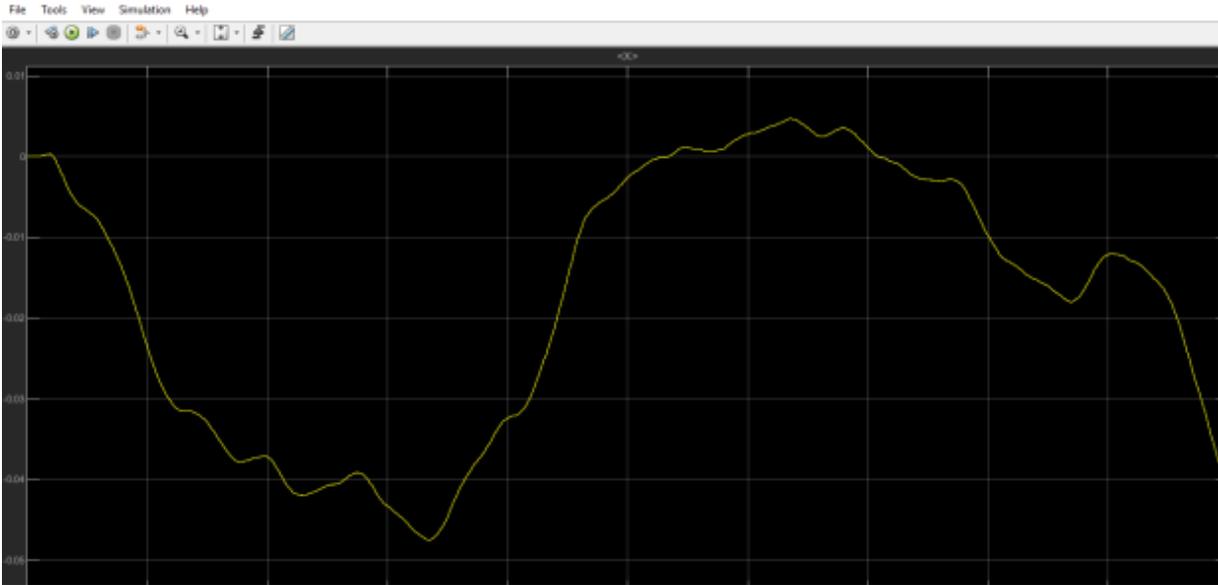


Figura 63 : Tracciato X

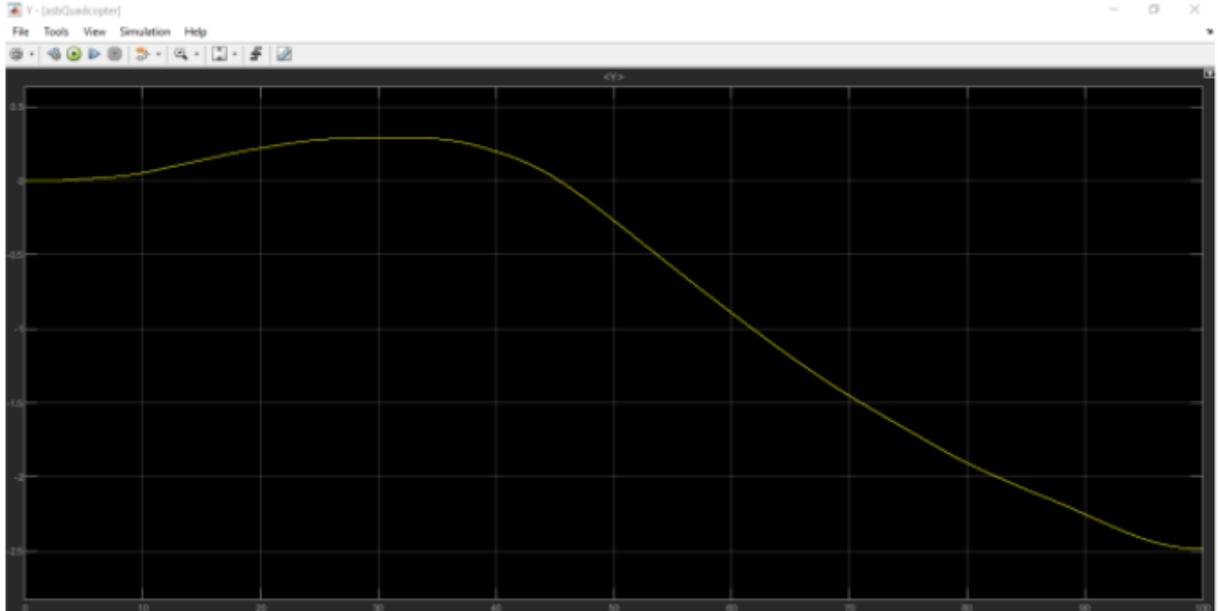


Figura 64 : Tracciato Y

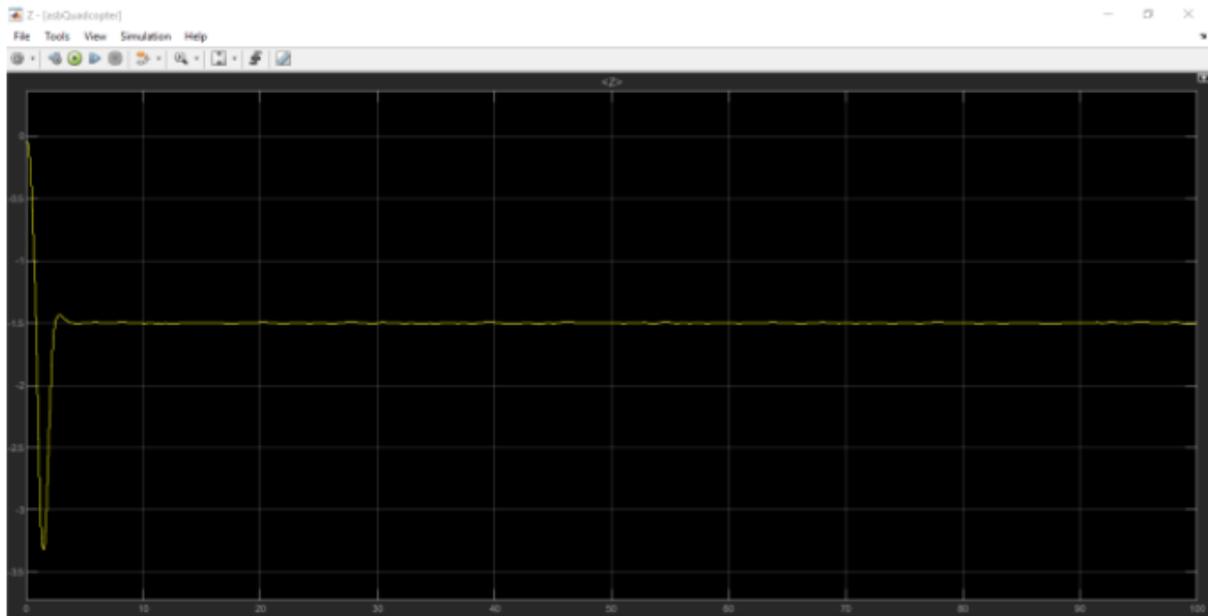


Figura 65 : Tracciato Z

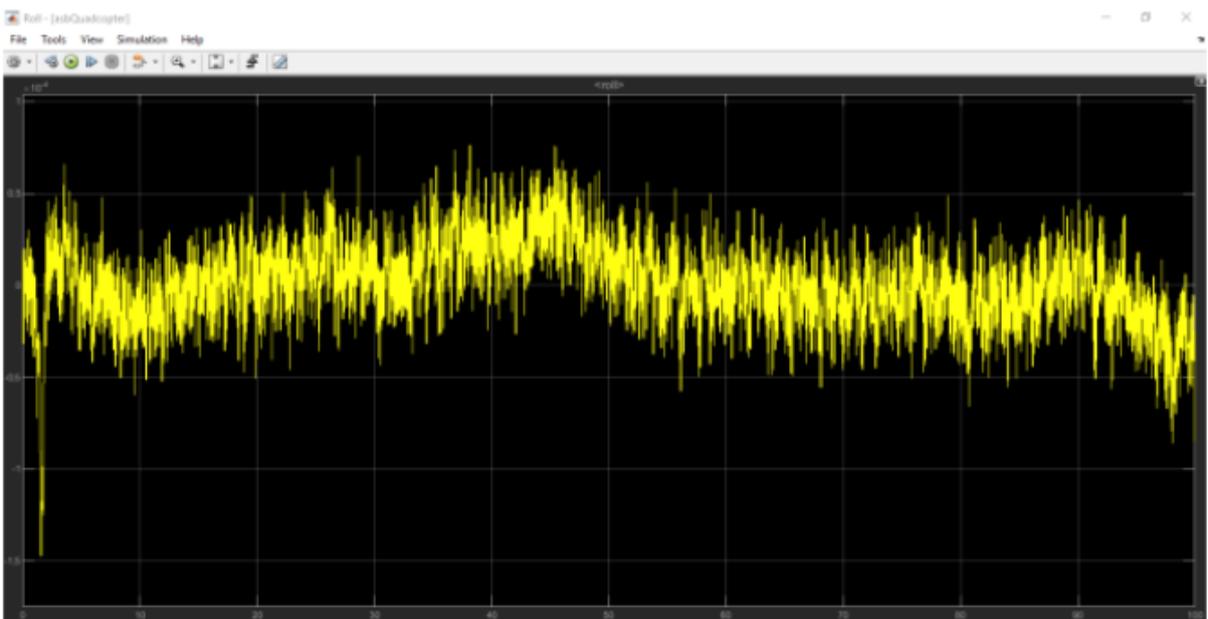


Figura 66 : Tracciato Roll

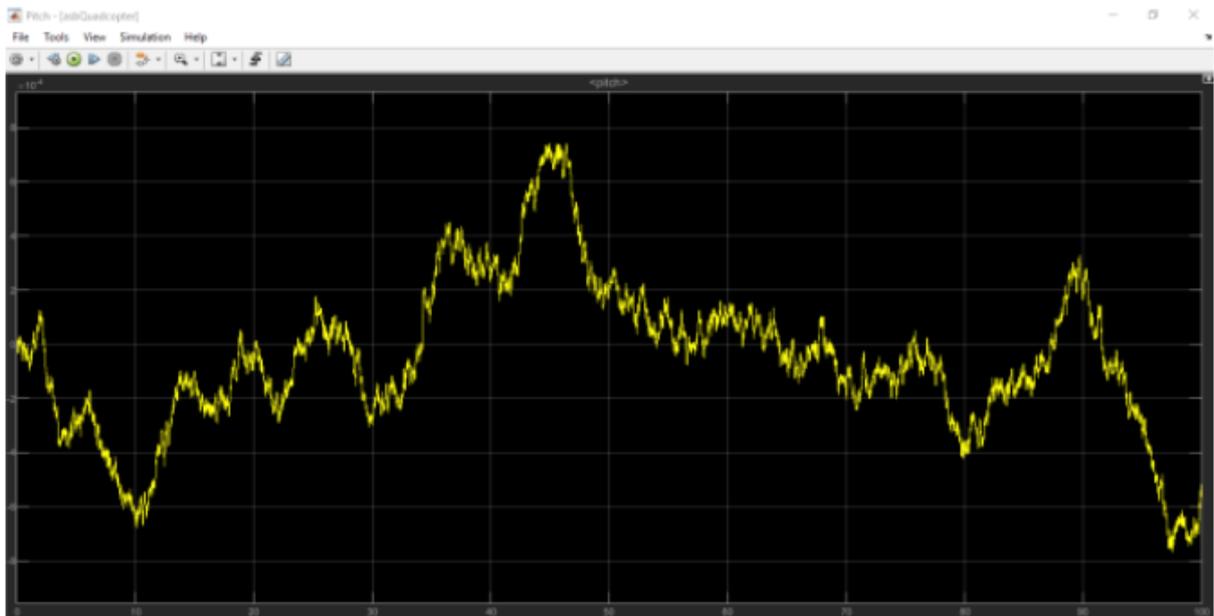


Figura 67 : Tracciato Pitch

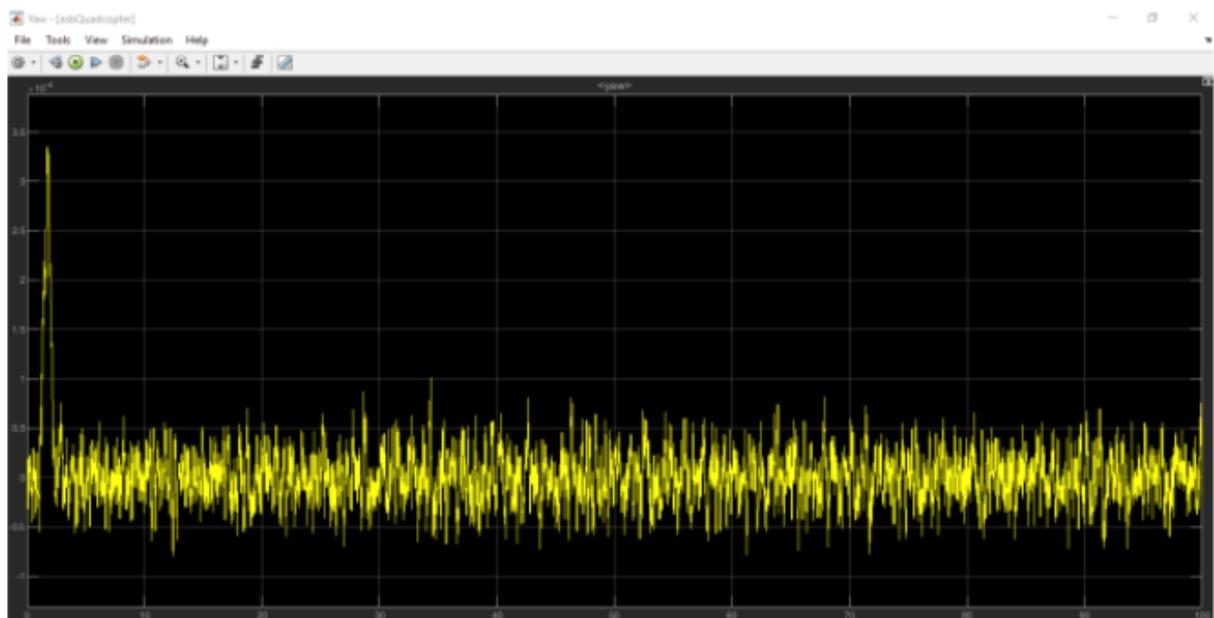


Figura 68 : Tracciato Yaw

5 Conclusioni

Il controllore progettato risulta avere prestazioni simili al controllore PID dapprima implementato, il controllo mantiene le oscillazioni sull'ordine di grandezza del 10^{-3} [m] che sono abbastanza contenute. Il comportamento nel caso lineare è del tutto analogo al comportamento che il sistema adotta in caso di modello non lineare.

Notiamo un piccolo problema rispetto al controllore iniziale riguardante l'angolo di Roll, che insieme all'angolo di Pitch, sono le variabili che controllano la posizione. In particolare, vediamo che all'inizio della simulazione abbiamo oscillazioni dell'angolo di roll leggermente più ampie che poi si assestano successivamente-

Per quanto riguarda il Tracciato Z è uguale in tutti i grafici, ciò è dovuto alla spinta dei motori che come si osserva forniscono al drone una spinta maggiore all'inizio che gli consente di prendere quota e stabilizzarsi, sia in altitudine che per i motori ad un valore costante.

Possiamo concludere che il controllore sull'angolo di Pitch progettato ha prestazioni in alcuni casi analoghe al controllore PID precedentemente implementato e in alcuni casi leggermente migliori con oscillazioni che vanno di $\pm 0.5 \times 10^{-3}$ [m].

Il sistema di controllo realizzato è stato in grado di soddisfare le specifiche richieste e adotta un comportamento del tutto paragonabile al PID iniziale.

BIBLIOGRAFIA

1. S. Bouabdallah Design and Control of Quadrotors with Application to Autonomous Flying, Lausanne, 2007
2. Pounds, Mahony, Corke, (21 February 2010), Modelling and control of a large quadrotor robot, Control Engineering Practice.
3. Isidori A. (1993), Sistemi di controllo, Siderea
4. W.Z.Fum, Implementation of Simulink Controller Design on Iris+Quadcopter, Monterey, CA, 2015.

Ringraziamenti

Vorrei ringraziare Gianluca Ippoliti, relatore di questa tesi, per l'aiuto fornitomi durante tutto il percorso di stesura e di tirocinio.

Vorrei ringraziare la famiglia, in particolare mia madre e mio padre, per avermi permesso di raggiungere questi risultati, senza i loro insegnamenti e senza il loro supporto non ce l'avrei mai fatta.

Un grande ringraziamento va a tutti i miei amici, da quelli di sempre a quelli conosciuti durante il cammino, per aver dimostrato sempre stima e affetto nei miei confronti, aiutandomi in questi anni di università a superare ogni ostacolo.

Un ultimo ma non meno importante ringraziamento va ai miei nonni, per avermi cresciuto e insegnato i veri valori della vita.