



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di Laurea Triennale
in Ingegneria Informatica e dell'Automazione

**STUDIO E SIMULAZIONE DELLA
DINAMICA DI MINIDRONI**

Study and simulation of minidron's dynamic

Relatore:
Gianluca Ippoliti

Tesi di Laurea di:
Xhulio Limani

Correlatore:
Giuseppe Orlando

A.A. 2018/2019

Sommario

INTRODUZIONE	3
CARATTERISTICHE FISICHE.....	5
FISICA DI UN QUADRICOTTERO.....	7
DINAMICA DEL SISTEMA.....	9
MODELLO NON LINEARE.....	14
MODELLO LINEARIZZATO.....	22
MODELLO SIMULINK	24
CONTROLLORE.....	28
SINTESI IN FREQUENZA	28
PROCESSO $P(s)$	29
PROGETTAZIONE DEL CONTROLLORE	31
CONTROLLORE ORIGINARIO PID SU MODELLO SIMULINK.....	34
CONTROLLORE PID CASO NON LINEARE	35
CONTROLLORE PID CASO LINEARE.....	39
CONTROLLO IN FREQUENZA SU SIMULINK.....	43
CONTROLLORE IN FREQUENZA CASO NON LINEARE.....	44
CONTROLLORE IN FREQUENZA CASO LINEARE	48
COMPORTAMENTO NEL CASO REALE.....	53
CONCLUSIONI.....	55

INTRODUZIONE

Oggi giorno si vede e si parla sempre piu di droni e minidroni. In un futuro non molto prossimo potrebbero diventare quelli che ci portano i pacchi e la posta a casa, quelli che ci portano la colazione a letto la mattina o quelli che ci portano il pasto al ristorante. Ma oltre a droni per il trasporto e per usi ricreativi, ci sono anche: droni per le costruzioni, per le calamità naturali ed esplorazione, per l'agricoltura, supporto medico e droni utilizzati in ambito militare. Quindi vengono usati in molti campi e settori lavorativi, dando così molto spazio alla ricerca e al loro sviluppo grazie a cospicui investimenti.

Ci sono molte tipologie di droni: tricottero, quadricottero, esacottero e octacottero. Per quanto diversi, il principio di funzionamento è simile per tutti i modelli, quindi, si presuppone che tutta la componentistica sia affine. Io in particolare in questo elaborato parlerò dei mini-droni di tipo quadricottero con configurazione a cross. Si è deciso di studiare e sviluppare un sistema di controllo per il mini-drone “Mambo” della casa Parrot, più precisamente ho scelto di sviluppare un controllore per lo yaw, attraverso la sintesi in frequenza, ovvero la rotazione intorno all'asse verticale. Parrot, con la collaborazione di Matlab, mette a disposizione un ambiente di sviluppo dove vengono riportati il modello lineare e non lineare del mini-drone Mambo e anche un simulatore di volo. Inoltre il pacchetto di supporto Simulink consente di progettare e costruire algoritmi di controllo del volo, accedendo ai sensori di bordo, e consente di modellare equazioni di movimento 6-DOF, cioè a 6 gradi di libertà. Questi alla fine possono essere caricati sul mini-drone in modalità wireless tramite Bluetooth Low Energy. Grazie a questo ambiente è stato possibile sostituire il controllore originario del mini-drone con quello da me creato e vedere le differenze, i pregi e i difetti usando il simulatore di volo.

Inizialmente è stato studiato il funzionamento del quadricottero, le sue componenti e il suo comportamento dinamico, solo dopo si è visto come queste sono state modellate con Simulink, cercando di capire le equazioni usate nel modello non lineare.

Successivamente si è passato al modello lineare e allo studio del sistema dinamico. Poiché questo modello si avvaleva di 12 stati, 14 ingressi e 33 uscite, sono andato a ricavarne le sottomatrici che mi interessavano per implementare il controllore di yaw. Una volta ricavate è stato possibile risalire alla funzione di trasferimento del processo per poi finalmente andare a progettare il controllore funzionante nel caso lineare e nel caso non lineare.

Come sopra accennato, ho scelto di applicare una sintesi in frequenza, con una funzione di correzione anticipatrice e mediante lo strumento 'sisotool', offerto da Matlab, sono andato a verificare la stabilità.

Alla fine, una volta soddisfatte le specifiche iniziali da me imposte, si è implementato il controllore sul Simulink.

Il risultato da me ottenuto è stato di poco migliore al controllore originario implementato in casa Parrot.

CARATTERISTICHE FISICHE

In questo paragrafo farò una descrizione del mini-drone Mambo.

Il mini-drone è un quadricottero, ovvero ha quattro eliche rotanti e pesa solo 63 gr.

Tutta la sua elettronica è ubicata in una PCB nella pancia dello scafo.

Iniziamo dai sensori, capovolgendolo si vedono due sensori, uno è un sensore ad ultrasuoni e l'altro è una fotocamera.



Figura 1-Parte inferiore del Mambo.

Il sensore ad ultrasuoni serve per misurare la distanza verticale, ovvero la distanza dal suolo. Invia un impulso sonoro ad alta frequenza e misura quanto tempo impiega quell'impulso a rimbalzare sul pavimento per poi ritornare indietro al sensore per una distanza al massimo di 4 metri circa, dopo di che il suono riflesso risulta essere troppo morbido perchè il sensore possa captarlo.

La fotocamera scatta foto a 60 fotogrammi al secondo e utilizza una tecnica di elaborazione dell'immagine chiamata 'optical flow' per determinare come gli oggetti si muovono tra un fotogramma e l'altro. Grazie a questa tecnica il Mambo può stimare il movimento orizzontale e la velocità.



Figura 2-Parte superiore del Mambo.

All'interno del mini-drone si ha un sensore di pressione che misura indirettamente l'altitudine. Mentre il drone sale di altitudine, la pressione dell'aria scende leggermente, quindi, questo cambiamento di pressione viene usato per stimare come cambia dinamicamente la sua altitudine, cioè se il drone va su o va giù.

Infine si ha un IMU, un sensore di misura inerziale che è fatto da un accelerometro e un giroscopio a tre assi che misurano rispettivamente l'accelerazione lineare e la velocità angolare. Quindi grazie all'IMU e alla conoscenza dell'accelerazione dovuta alla gravità si può stimare l'atteggiamento del Mambo rispetto alla gravità e quanto velocemente esso ruota. Quindi si usano il sensore ad ultrasuoni e quello di pressione per determinare l'altitudine, mentre l'IMU e la fotocamera per determinare il movimento rotazionale e traslazionale.

Si hanno 4 motori brushed coreless, ogni motore ha la propria elica e sono incrociati a x. Le eliche presentano due colori diversi per dare un'idea dell'orientamento del mini-drone al pilota. Questi 4 motori iniettano le forze e le coppie nel sistema, quindi lo scopo è quello di manipolare i motori in modo che il Mambo assuma il comportamento che gli si vuole imporre.

FISICA DI UN QUADRICOTTERO

Come detto precedentemente, abbiamo 4 motori e 6 gradi di libertà, ovvero, su e giù avanti e dietro destra e sinistra ma abbiamo anche tre direzioni di rotazione, roll pitch e yaw.

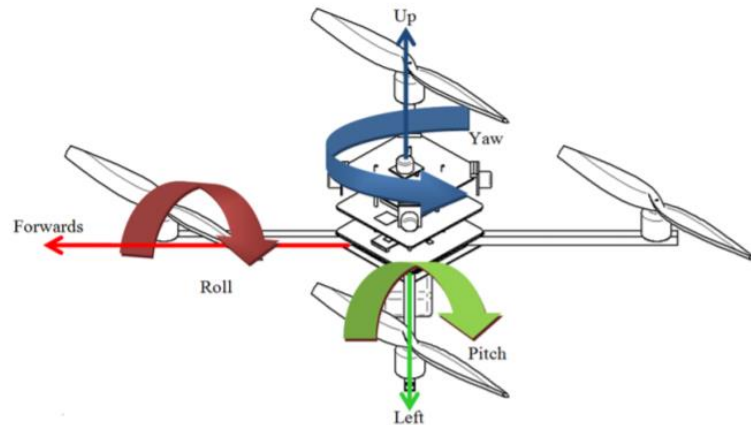


Figura 3- Yaw Pitch e Roll.

Quindi non abbiamo un motore per ogni movimento, questo vuol dire che in un certo momento non riusciamo a comandare un certo spostamento. Per questo si usano sistemi di controllo che accoppiano spinta e rotazione in modo da raggiungere obiettivi generali. Dei 4 motori, due girano in senso orario e gli altri due in senso anti orario, i motori opposti girano nello stesso verso. Per ogni diagonale si ha una coppia la quale agisce sull'asse dove sono posizionati i motori.

Un motore produce la rotazione di un elica, la quale produce una spinta che spinge l'aria verso il basso e questo provoca una forza di reazione. I motori sono posizionati tutti alla stessa distanza dal centro di gravità e hanno tutti la stessa spinta, questi provocano solo la traslazione del mini-drone senza alcuna rotazione, e se la forza di spinta è uguale contraria alla forza di gravità questo fa sì che il Mambo rimanga in volo in una posizione fissa.

Se si modificano opportunamente le forze che applicate ai motori, cioè i giri, si ottiene una variazione di forze o momenti tale da far effettuare al mini-drone la manovra desiderata, questo perché una forza applicata a una distanza dal centro di gravità produce sia un movimento di traslazione che una coppia, oppure un momento rotante attorno al centro di gravità.

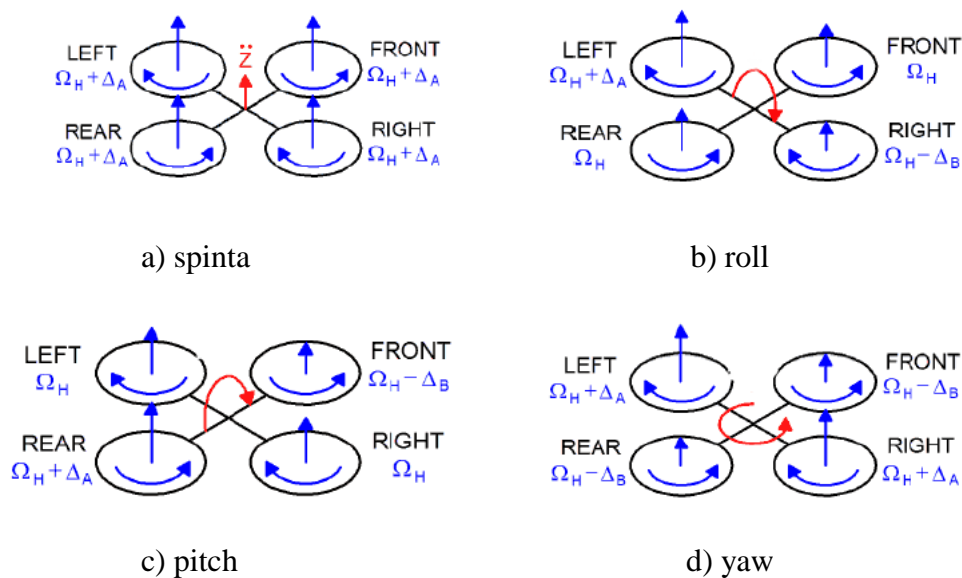
La manovra di salita viene svolta incrementando la spinta dei motori allo stesso modo.

Ci si trova in condizione di hovering quando si vola a un punto fisso e avendo quattro motori si ottiene questa condizione quando ogni motore ha un tiro tale che sia un quarto di peso del Mambo, bilanciando così le forze tra la spinta e il peso.

I movimenti intorno agli assi di roll e pitch possono essere eseguiti senza che venga influenzato l'asse di yaw, agendo sulle coppie di motori diversi, quindi la tecnica che si adotta per roll e pitch è la stessa.

Se da una parte viene diminuita la potenza di un motore e dalla parte opposta viene aumentata la potenza dell'altro motore allora si mantiene la coppia complessivamente nulla, ma si ha una rotazione attorno all'asse degli altri due motori dove non è stata modificata la potenza.

Mentre invece per quanto riguarda lo yaw, la variabile di stato che sono andati a controllare, il suo spostamento è dato fornendo la stessa potenza a una coppia di motori e togliendo potenza all'altra coppia allo stesso modo.



Con Ω_H si indica la velocità di rotazione che serve ad ogni motore per indurre una forza al mini-drone in modo da contrastare la forza di gravità quando si è in hovering.

Con Δ_A e Δ_B si indica una certa quantità.

DINAMICA DEL SISTEMA

Grazie agli angoli di Eulero si ha una descrizione di un corpo rigido nello spazio, in particolare dello sfasamento angolare di un sistema di riferimento XYZ solidale con un corpo rigido con una serie di rotazioni elementari a partire da un sistema di riferimento fisso.

Si ha quindi una parte fissa xyz e una solidale a questa, XYZ. Tutto questo viene detto perché prima di andare a vedere le equazioni che descrivono l'andamento del mini-drone è bisogno introdurre le coordinate di riferimento con la quale si descrivono assetto e posizione.

La terna fissa (xyz) viene detta anche terna inerziale e su questa vale la prima legge di Newton. L'altra invece viene chiamata anche terna mobile oppure più comunemente con il termine inglese frame. Quest'ultima è solidale con il baricentro del quadricottero.

Si indicano con:

Θ : il movimento intorno a y, pitch;

Ψ : il movimento intorno a z, yaw;

Φ : il movimento intorno a x, roll.

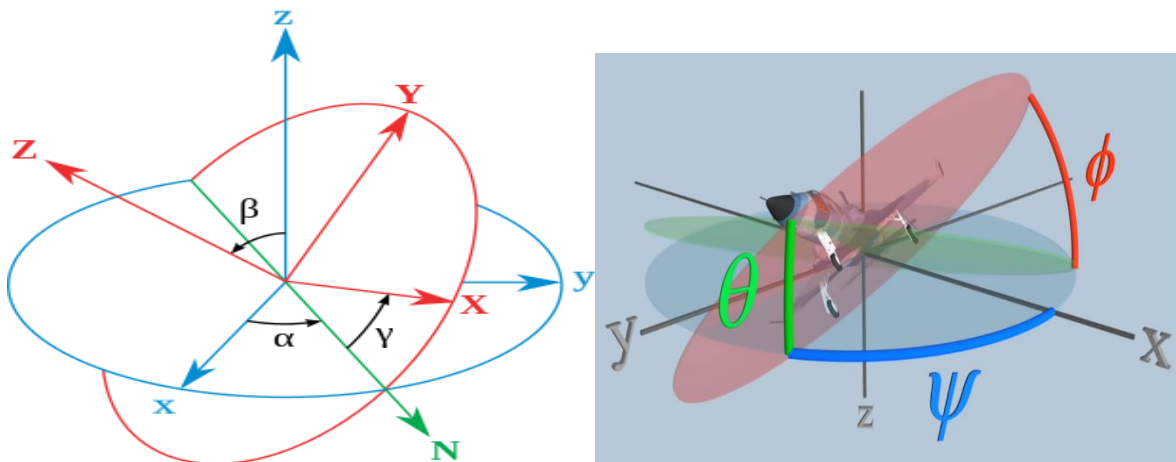


Figura 4- Angoli di Eulero.

Il sistema di riferimento fisso inerziale, è scelto in riferimento al piano tangente passante per la superficie della terra ed è chiamato O_{NED} , invece quella mobile è presa solidale con il baricentro del mini-drone ed è indicata con O_{ABC} .

E' necessario usare entrambi i sistemi di riferimento per risolvere le equazioni del modello.

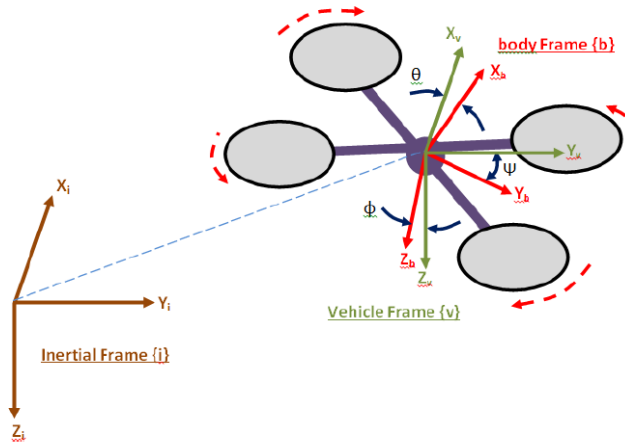


Figura 5- Inertial frame e vehicle frame.

Si usano tre matrici di rotazione per passare da O_{ABC} a O_{NED} ed ognuna corrisponde a un angolo tra roll pitch e yaw.

$$R_x(\phi) = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{vmatrix}; R_y(\theta) = \begin{vmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{vmatrix}; R_z(\psi) = \begin{vmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

Moltiplicando le matrici di rotazione si ottiene la matrice totale di rotazione in R^3 , nota anche come DCM (Direction Cosine Matrix).

$$DCM(\phi, \theta, \psi) = \begin{vmatrix} \cos\theta \cos\psi & \cos\phi \sin\psi + \sin\phi \sin\theta \cos\psi & \sin\phi \sin\psi - \cos\phi \sin\theta \cos\psi \\ -\cos\theta \sin\psi & \cos\phi \cos\psi - \sin\phi \sin\theta \sin\psi & \sin\phi \cos\psi + \cos\phi \sin\theta \sin\psi \\ \sin\theta & -\sin\phi \cos\theta & \cos\phi \cos\theta \end{vmatrix}$$

Però per semplicità da ora in poi chiamerò questa matrice R.

Si ha un'altra matrice, T , per le trasformazioni angolari:

$$T = \begin{vmatrix} 1 & \sin\phi \tan\theta & -\sin\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\sin\theta & \cos\phi/\cos\theta \end{vmatrix}$$

La posizione e l'assetto di un sistema in movimento a sei gradi di libertà sono descritte attraverso dodici equazioni delle quali le variabili di interesse sono 6 per il sistema O_{NED} ed altre 6 per il sistema O_{ABC} .

Si hanno due vettori che richiamano le velocità di traslazione e la velocità angolare, cioè

$$\mathbf{V}_B = [u \ v \ w]^T \quad (1)$$

$$\mathbf{\omega}_B = [p \ q \ r]^T \quad (2)$$

espressi nella terna di riferimento del sistema corpo del drone cioè O_{ABC} .

Per quanto riguarda O_E invece si ha a che fare con le posizioni lineari e gli angoli di roll pitch e yaw:

$$\mathbf{\Gamma}_{NED} = [x \ y \ z]^T \quad (3)$$

$$\mathbf{\Theta}_{NED} = [\phi, \ \theta, \ \psi]^T \quad (4)$$

Ora con l'uso della matrice di rotazione, si può definire:

$$\dot{\mathbf{\Gamma}}_{NED} = \mathbf{R} \cdot \mathbf{V}_B \quad (5)$$

$$\dot{\mathbf{\Theta}}_{NED} = \mathbf{T} \cdot \mathbf{\omega}_B \quad (6)$$

Mettendo il tutto in una forma piu chiara si ha :

$$\begin{cases} \dot{x} = u \cdot \cos\theta \cos\psi + v (\sin\theta \sin\phi \cos\psi - \cos\phi \sin\psi) + w (\sin\theta \cos\phi \cos\psi + \sin\phi \sin\psi) \\ \dot{y} = u \cdot \cos\theta \sin\psi + v (\sin\theta \sin\phi \sin\psi + \cos\phi \cos\psi) + w (\sin\theta \cos\phi \sin\psi - \sin\phi \cos\psi) \\ \dot{z} = -u \sin\theta + v \cos\theta \sin\phi + w \cos\theta \cos\phi \end{cases}$$

$$\begin{cases} \dot{\phi} = p + q \cdot \tan\theta \cdot \sin\phi + r \cdot \tan\theta \cdot \cos\phi \\ \dot{\theta} = q \cdot \cos\theta - r \cdot \sin\theta \\ \dot{\psi} = \frac{\sin\psi}{\cos\theta} \cdot q + \frac{\cos\psi}{\cos\theta} \cdot r \end{cases}$$

Queste rappresentano le prime sei espressioni della terna fissa O_E . Ora possiamo dire che l'angolo ψ dipende dagli altri due (roll e pitch) e questo significa che manovre di roll (ϕ) e pitch (θ) possono causare variazioni angolari di ψ ottenendo cosi anche una manovra di yaw. Se si ipotizza che O_{ABC} coincide con il baricentro del mini-drone e in più gli stessi assi di O_{ABC} coincidono con gli assi principali di inerzia, allora è applicabile la seconda legge di Newton. Quindi si ha :

$$m\ddot{\Gamma}_{NED} = \sum_{i=0}^n F_i = F_{NED} \quad (7)$$

$$I\ddot{\Theta}_{NED} = \sum_{i=0}^n \tau_i = \tau_{Ned} \quad (8)$$

e andando a sostituire la (5) si ha:

$$m\widehat{R\dot{V}_B} = RF_B \quad (9)$$

$$m(R\dot{V}_B + \dot{R}V_B) = RF_B \quad (10)$$

per ottenere poi:

$$mR(\dot{V}_B + \omega_B \times V_B) = RF_B \quad (11)$$

$$m(\dot{V}_B + \omega_B \times V_B) = F_B \quad (12)$$

questo perche la velocità di una terna si può esprimere come velocità lineare della stessa più la sua rotazione.

Alla fine si ottiene la forma generica della equazione di Newton.

Facendo la stessa cosa con i termini angolari si ha :

$$I\dot{\hat{T}}\omega_B = T\tau_B \quad (13)$$

$$IT\dot{\omega}_B + T\omega_B \times (I\omega_B) = T\tau_B \quad (14)$$

$$I\dot{\omega}_B + \omega_B \times (I\omega_B) = \tau_B \quad (15)$$

ottenendo così il risultato della coppia nel riferimento di ABC, che rappresenta l'equazione di Eulero.

Quindi ora sviluppando il tutto sottoforma di prodotto matriciale si può scrivere:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} = F_B$$

dove in termini angolari si ha :

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = m \begin{bmatrix} \dot{p}I_x + qr(I_z - I_y) \\ \dot{q}I_y + pr(I_x - I_z) \\ \dot{r}I_z + pq(I_y - I_x) \end{bmatrix} = \tau_B$$

Finalmente è stato ottenuto tutto l'occorrente per calcolare le informazioni sul sistema in movimento.

MODELLO NON LINEARE

Quanto detto fino a ora non basta, bisogna vedere quali sono le forze e i momenti esterni agenti sul velivolo. Si è studiato nel dettaglio da che cosa sono composti questi elementi.

Si può scomporre la forza F_B in forza peso F_B^g , dovuta all'attrazione gravitazionale della terra, e in forza dei motori F_B^T , che in gergo possiamo definire “spinta”. Per il peso si usa il prodotto tra massa e la gravità. Questo vettore ha una direzione verticale e diretto verso il basso indipendentemente dall'orientamento del corpo.

$$F_B = F_B^g + F_B^T \quad (16)$$

La spinta T dipende dalla velocità di rotazione dei motori:

$$T = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (17)$$

Possiamo sintetizzare le forze esterne quindi in :

$$R \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix}$$

sviluppendole si ricava :

$$\dot{u} = rv - qw - g \sin \theta \quad (18)$$

$$\dot{v} = pw - ru - g \cos \theta \sin \phi \quad (19)$$

$$\dot{w} = qu - pv + g \cos \phi \cos \theta - \frac{T}{m} \quad (20)$$

Per calcolare le coppie invece :

$$M_x = lb(\Omega_2^2 - \Omega_4^2) \quad (21)$$

$$M_y = lb(\Omega_1^2 - \Omega_3^2) \quad (22)$$

$$M_z = d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (23)$$

Le equazioni angolari che descrivono il moto sono:

$$\dot{p} = \frac{lb}{I_x}(\Omega_2^2 - \Omega_4^2) - qr \frac{Iz - Iy}{I_x} \quad (24)$$

$$\dot{q} = \frac{lb}{I_x}(\Omega_1^2 - \Omega_3^2) - pr \frac{Ix - Iz}{Iy} \quad (25)$$

$$\dot{r} = \frac{d}{I_z}(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (26)$$

Quindi alla fine si è arrivati a descrivere la dinamica del quadricottero con le seguenti equazioni:

$$\dot{\Gamma}_{NED} = R^{-1}V_B \quad (27)$$

$$\dot{u} = rv - qw - g\sin\theta \quad (28)$$

$$\dot{v} = pw - ru - g\cos\theta\sin\phi \quad (29)$$

$$\dot{w} = qu - pv + g\cos\phi\cos\theta - \frac{T}{m} \quad (30)$$

$$\dot{\phi} = p + \sin(\phi) + \tan\theta r \quad (31)$$

$$\dot{\theta} = \cos\theta q - \sin\phi r$$

$$\dot{\psi} = \frac{\sin\phi}{\cos\theta}q + \frac{\cos\phi}{\cos\theta}r \quad (33)$$

$$\dot{p} = \frac{lb}{I_x}(\Omega_2^2 - \Omega_4^2) - qr \frac{Iz - Iy}{I_x} \quad (34)$$

$$\dot{q} = \frac{lb}{I_x}(\Omega_1^2 - \Omega_3^2) - pr \frac{Ix - Iz}{Iy} \quad (35)$$

$$\dot{r} = \frac{d}{I_z}(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (36)$$

MODELLO NON LINEARE USATO

Dopo aver visto e studiato come si ricavano le forze e i momenti, siamo andati a spulciare il modello non lineare sul Simulink. In seguito ad alcune difficoltà iniziali causatemi da alcune incongruenze in termini di formule, studiando a dovere il modello non lineare, ho visto che esso è descritto da queste formule:

$$\dot{\xi} = \mathbf{R}\mathbf{v} \quad (37)$$

$$m\dot{\mathbf{v}} = -m\boldsymbol{\omega} \times \mathbf{v} + mg\mathbf{R}^T e_3 + \sum_{N,S,E,W} \mathbf{t}_i \quad (38)$$

$$\dot{\mathbf{R}} = \mathbf{R} \cdot \text{sk}(\boldsymbol{\omega}) \quad (39)$$

$$\mathbf{I}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \sum_{N,S,E,W} [\mathbf{q}_i + \mathbf{m}_i] \quad (40)$$

$$\mathbf{t}_i = C_T \rho A r^2 \omega_i^2 \begin{pmatrix} -\sin a_{1si} \\ \cos a_{1si} \sin b_{1si} \\ -\cos b_{1si} \cos a_{1si} \end{pmatrix} \quad (41)$$

$$\mathbf{q}_i = C_Q \rho A r^3 \omega_i |\omega_i| e_3 \quad (42)$$

$$\mathbf{m}_i = \mathbf{t}_i \times \mathbf{d}_i \quad (43)$$

In queste formule viene indicato con \mathbf{I} la rotazione inerziale del mini-drone, con \mathbf{m} la sua massa, con \mathbf{g} l'accelerazione di gravità, con ρ la densità dell'area, con r il raggio del motore e A la sua area. I vettori \mathbf{t}_i e \mathbf{q}_i sono rispettivamente la spinta e la coppia dei motori, mentre \mathbf{m}_i è il momento dell' i -esimo motore.

Il momento prodotto è dovuto alla spinta che agisce attorno a uno spostamento dal centro di gravità. Inoltre \mathbf{d}_i è lo spostamento del motore dal centro di massa del quadricottero.

Si ha :

$$\mathbf{d}_N = (0 \quad d \quad h) \quad (44)$$

$$\mathbf{d}_S = (0 \quad -d \quad h) \quad (45)$$

$$\mathbf{d}_E = (d \quad 0 \quad h) \quad (46)$$

$$\mathbf{d}_W = (-d \quad 0 \quad h) \quad (47)$$

dove \mathbf{d} e \mathbf{h} sono rispettivamente la lunghezza del braccio del mini-drone e l'altezza dei motori. Con \mathbf{a}_i e \mathbf{b}_i si indicano le prime armoniche degli angoli longitudinali e laterali delle eliche dell' i -esimo motore e con \mathbf{C}_q e \mathbf{C}_t si indicano i coefficienti.

\mathbf{C}_t è legata a V_c , cioè la velocità verticale, dalla seguente relazione:

$$C_T/\sigma = \frac{a(\alpha)}{4} \left[\theta_{\text{tip}} - \frac{v_i + V_c}{\omega r} \right] \quad (48)$$

dove \mathbf{a} è la pendenza di sollevamento, θ_{tip} è l'angolo geometrico dell'elica all'estremità del motore, \mathbf{v}_i è la velocità indotta attraverso il motore, e σ è il rapporto tra l'area della superficie delle eliche e l'area del disco del motore.

Tuttavia la si può riscrivere in un altro modo dato che l'aumento dell'altitudine, il quale è dovuto all'incremento della velocità delle eliche, è relativamente piccolo in relazione al cambiamento angolare.

$$\Delta C_{Ti} = -\frac{a_0}{4} \frac{\sigma}{\omega_i r} (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{d}_i) e_3 \quad (49)$$

$\Delta \mathbf{C}_T$ è il cambiamento dovuto alla variazione del flusso, \mathbf{a}_0 è l'aumento della pendenza intorno a un punto.

Quando i motori traslano orizzontalmente c'è una differenza nell'inclinazione delle eliche posteriori e anteriori che causa l'instabilità del veivolo.

La rotazione delle eliche si ottiene calcolando l'advance ratio, cioè il rapporto tra la velocità delle eliche e quella del motore, e l'angolo di direzione:

$$\mathbf{v}_{ri} = \mathbf{v} + \boldsymbol{\omega} \times \mathbf{d}_i \quad (49)$$

$$\mu_{ri} = \frac{\|\mathbf{v}_{r(1,2)i}\|}{\omega_i r} \quad (50)$$

$$\psi_{ri} = \arctan \left(\frac{\mathbf{v}_{r(2)i}}{\mathbf{v}_{r(1)i}} \right) \quad (51)$$

$\mathbf{v}_{r(n)i}$ è l' n -esimo elemento dell' i -esimo elemento del vettore di velocità e μ_{ri} è l' i -esimo advance ratio. Infine ψ è l'angolo di direzione del movimento.

Gli angoli i -esimi dei movimenti laterali e longitudinali del quadricottero sono:

$$u_{1si} = \frac{1}{1 - \frac{\mu_{ri}^2}{2}} \mu_{ri} (4\theta_t - 2\lambda_{hi}^2) \quad (52)$$

$$v_{1si} = \frac{1}{1 + \frac{\mu_{ri}^2}{2}} \left(\frac{C_T}{\sigma} \frac{8}{9} \frac{\mu_{ri} \gamma}{a} + \frac{C_T}{2\mu_{ri}} \right) \quad (53)$$

dove λ_{hi} è approssimabile come:

$$\lambda_{hi} = \sqrt{C_T/2} \quad (54)$$

mentre invece γ è:

$$\gamma = \frac{\rho a_0 c r^4}{I_b} \quad (55)$$

Per il cambio di coordinate si usano:

$${}^A \mathbf{J}_{B_i} = \begin{pmatrix} \cos \psi_{ri} & -\sin \psi_{ri} \\ \sin \psi_{ri} & \cos \psi_{ri} \end{pmatrix} \quad (56)$$

$$\begin{pmatrix} a_{1si} \\ b_{1si} \end{pmatrix} = {}^A \mathbf{J}_{B_i} \begin{pmatrix} u_{1si} \\ v_{1si} \end{pmatrix} \quad (57)$$

dove infine queste sono sommate alle componenti del corpo rigido :

$$a_{1si} = \dots + \frac{-\frac{16}{\gamma} \left(\frac{q}{\omega} \right) + \left(\frac{p}{\omega} \right)}{1 - \frac{\mu_i^2}{2}} \quad (58)$$

$$b_{1si} = \dots + \frac{-\frac{16}{\gamma} \left(\frac{p}{\omega} \right) + \left(\frac{q}{\omega} \right)}{1 + \frac{\mu_i^2}{2}} \quad (59)$$

Ritroviamo quanto appena descritto all'interno del blocco non lineare, più precisamente nel sotto blocco Rotor Dynamics.

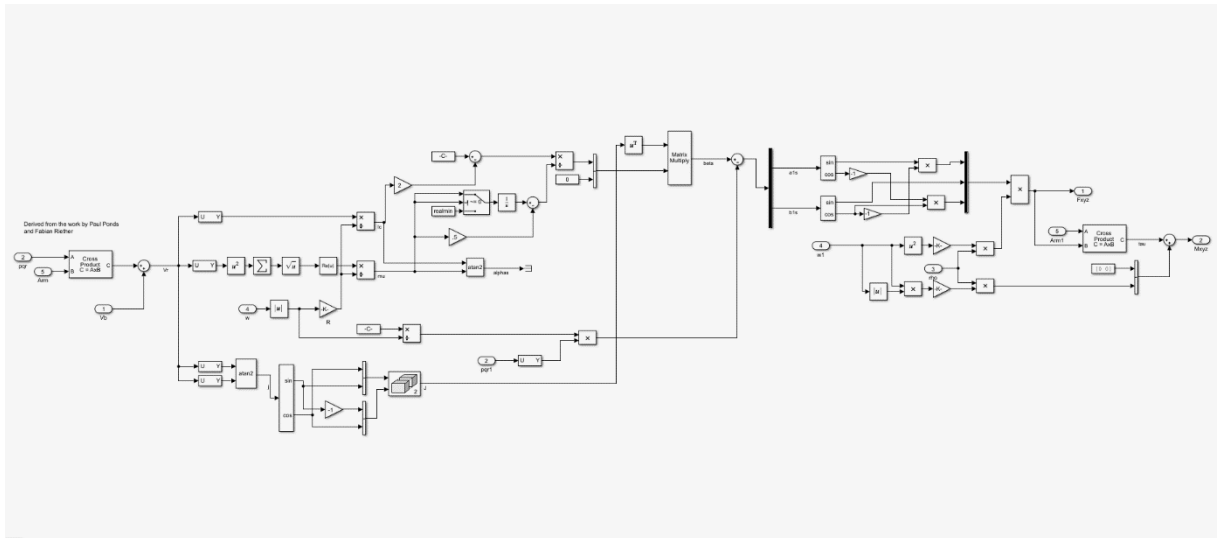


Figura 6 - Diagramma a blocchi del calcolo di forze e momenti in Simulink.

Andando a vedere le sue parti più da vicino e indicando le formule usate, si ha:

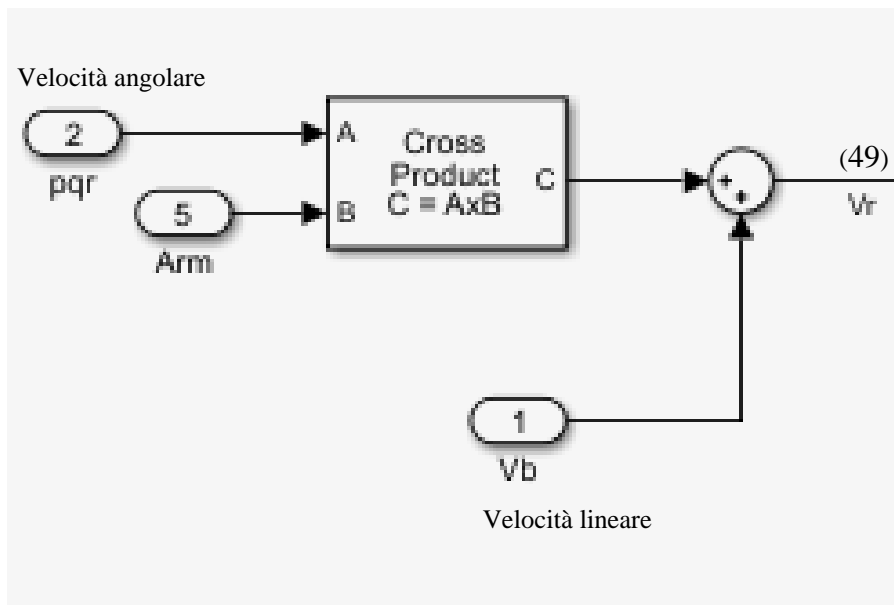


Figura 7- inizio della figura 6.

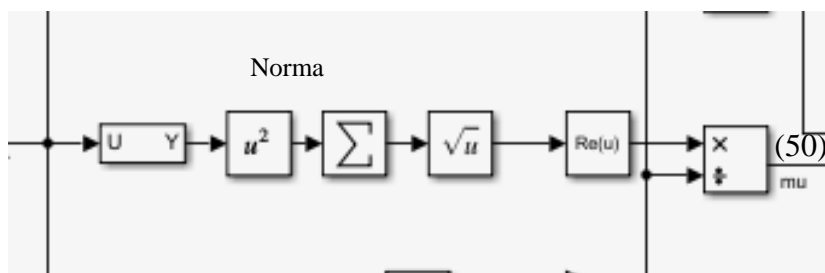


Figura 8- Parte intermedia della figura 6.

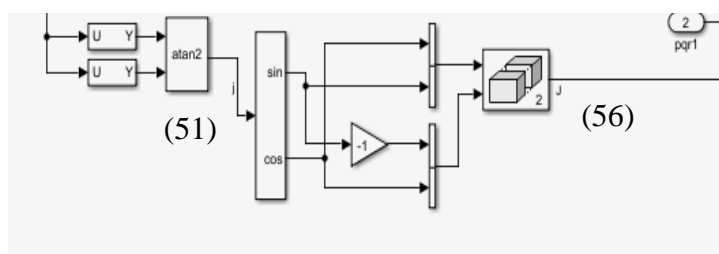


Figura 9- Parte della figura 6. Calcolo dell'angolo di direzione.

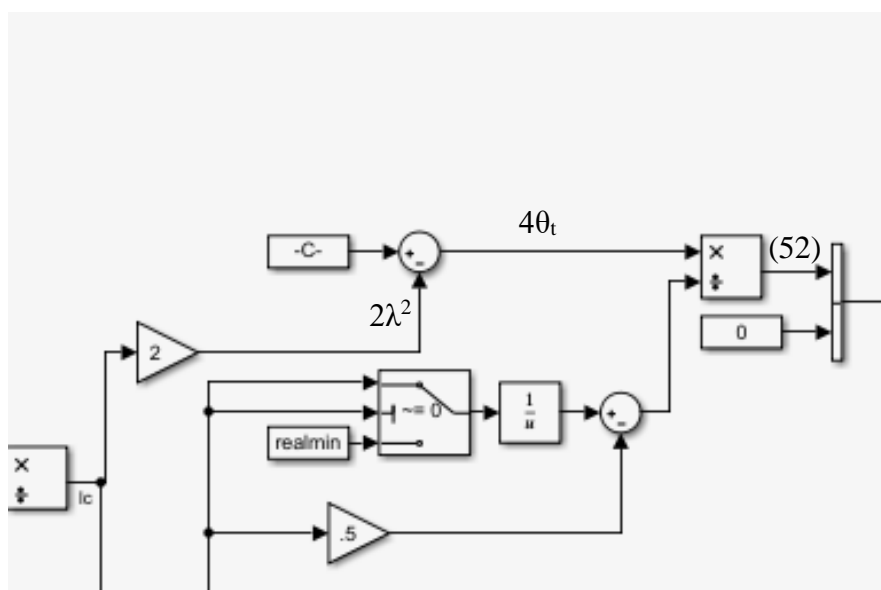


Figura 10- Movimenti longitudinali e laterali.

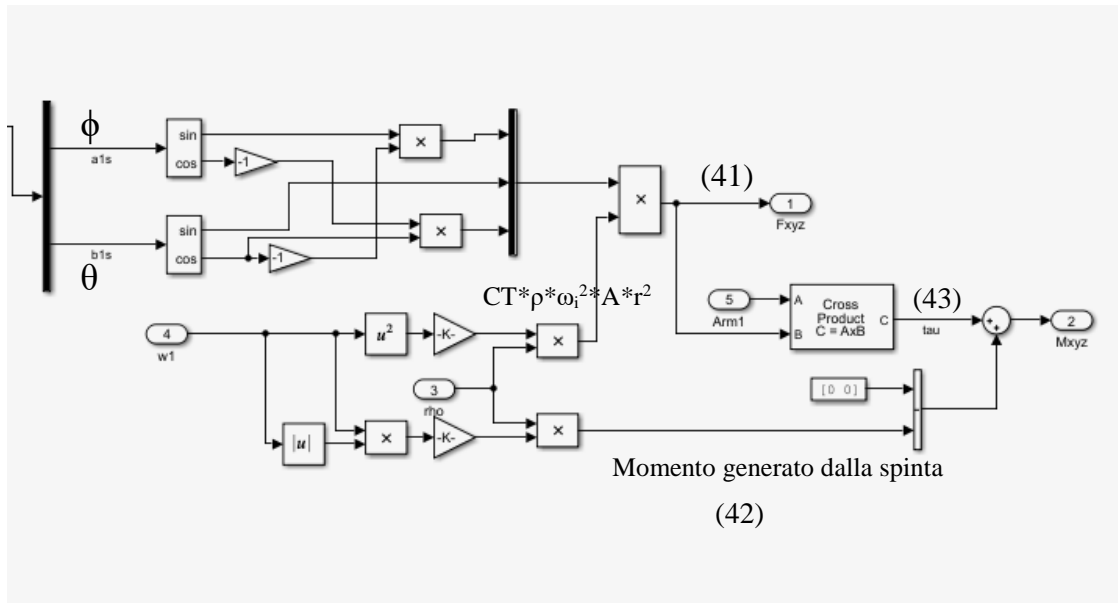


Figura 11-Parte finale della figura 6.

Queste due uscite, forza e momento, vanno poi a finire nel blocco 6DOF (Euler Angles) il quale agisce come descritto nel paragrafo precedente, ottenendo così gli stati.

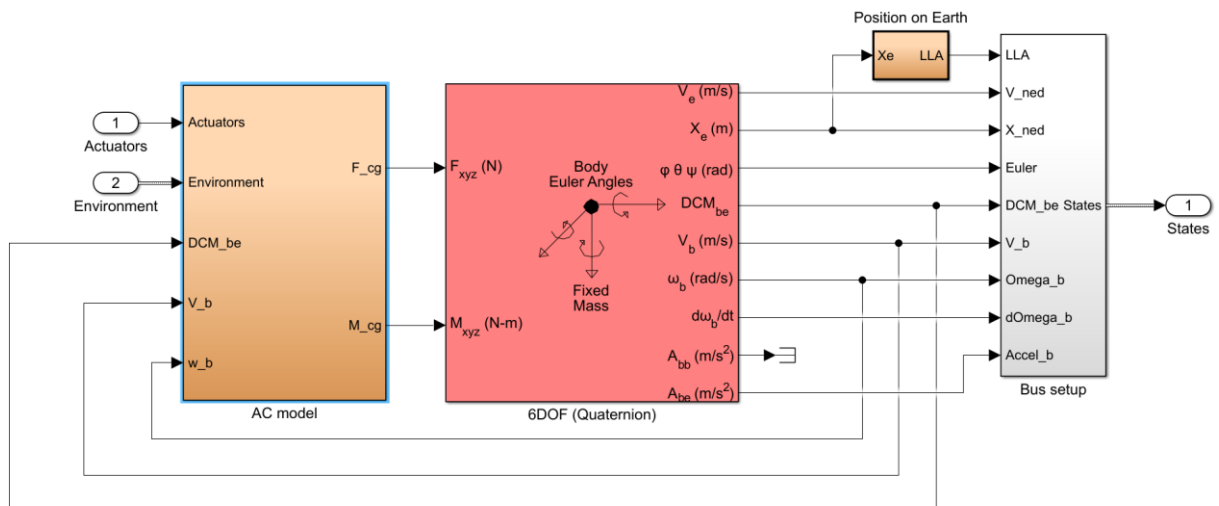


Figura 12-Modello non lineare

MODELLO LINEARIZZATO

Visto la struttura matematica dei sistemi lineari, essi sono più facili da studiare rispetto a quelli non lineari, inoltre, ci sono molte tecniche di controllo efficienti. Tuttavia sono pochi i sistemi fisici descritti da un modello lineare.

Comunque sia è possibile approssimare il comportamento di un sistema generico con quello di un sistema lineare per sfruttare tutte le proprietà dei sistemi lineari, in modo da sviluppare un algoritmo di controllo con tecniche lineari.

Il modello lineare che si introdurrà sarà valido solo negli intorno del punto di lavoro attorno al quale si è scelto di linearizzare il sistema.

Se il sistema non si discosta troppo dal punto scelto, l'approssimazione lineare è soddisfacente e quindi si può progettare un controllore in maniera più agevolata.

Un generico sistema non lineare può essere scritto nella seguente forma:

$$\begin{cases} \dot{x} = f(x(t), u(t), t) \\ y = g(x(t), u(t), t) \end{cases}$$

dove x è il vettore di stato, u degli ingressi e t il tempo.

Si linearizza il sistema attorno ad un punto di lavoro $(x_0(t), u_0(t))$, che soddisfa:

$$\dot{x}_0 = f[x_0(t), u_0(t), t]$$

Con Δx , Δu e Δy si definiscono delle variabili che rappresentano lo scostamento rispettivamente dello stato, dell'ingresso e dell'uscita del sistema rispetto ai punti di lavoro, o meglio, attorno ai valori dove si è deciso di linearizzare, si ha quindi:

$$\begin{aligned} x(t) &= x_0(t) + \Delta x(t) \\ u(t) &= u_0(t) + \Delta u(t) \\ y(t) &= h[x_0(t), u_0(t), t] + \Delta y(t) \end{aligned}$$

Utilizzando gli sviluppi di Taylor fino al primo ordine si possono così scrivere le equazioni:

$$\dot{x} = f[x_0(t), u_0(t), t] + \left. \frac{\partial f}{\partial x} \right|_{x_0(t), u_0(t)} \Delta x + \left. \frac{\partial f}{\partial u} \right|_{x_0(t), u_0(t)} \Delta u$$

$$y = h[x_0(t), u_0(t), t] + \left. \frac{\partial h}{\partial x} \right|_{x_0(t), u_0(t)} \Delta x + \left. \frac{\partial h}{\partial u} \right|_{x_0(t), u_0(t)} \Delta u$$

però sostituendo nella prima equazione :

$$\dot{x}_0 = f[x_0(t), u_0(t), t]$$

e ricordando che :

$$y(t) = h[x_0(t), u_0(t), t] + \Delta y(t)$$

si può scrivere:

$$\Delta \dot{x} = \left. \frac{\partial f}{\partial x} \right|_{x_0(t), u_0(t)} \Delta x + \left. \frac{\partial f}{\partial u} \right|_{x_0(t), u_0(t)} \Delta u$$

$$\Delta y = \left. \frac{\partial h}{\partial x} \right|_{x_0(t), u_0(t)} \Delta x + \left. \frac{\partial h}{\partial u} \right|_{x_0(t), u_0(t)} \Delta u$$

in modo da ottenere il sistema linearizzato e poterlo scrivere finalmente nel seguente modo:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

Dove A, B, C, D sono delle matrici così ottenute:

$$A(t) = \left. \frac{\partial f}{\partial x} \right|_{x_0(t), u_0(t)}, \quad B(t) = \left. \frac{\partial f}{\partial u} \right|_{x_0(t), u_0(t)}$$

$$C(t) = \left. \frac{\partial h}{\partial x} \right|_{x_0(t), u_0(t)}, \quad D(t) = \left. \frac{\partial h}{\partial u} \right|_{x_0(t), u_0(t)}$$

Bisogna però tener conto che è stata fatta un'approssimazione, valida solo in un intorno del punto di lavoro scelto, quindi nel momento in cui lo stato del sistema si allontana troppo dal valore attorno al quale si è deciso di linearizzare, il modello perde la sua validità.

MODELLO SIMULINK

Aperto quadcopter project negli esempi di Aerospace blockset ci troviamo davanti ai blocchi necessari per simulare la dinamica del Mambo.

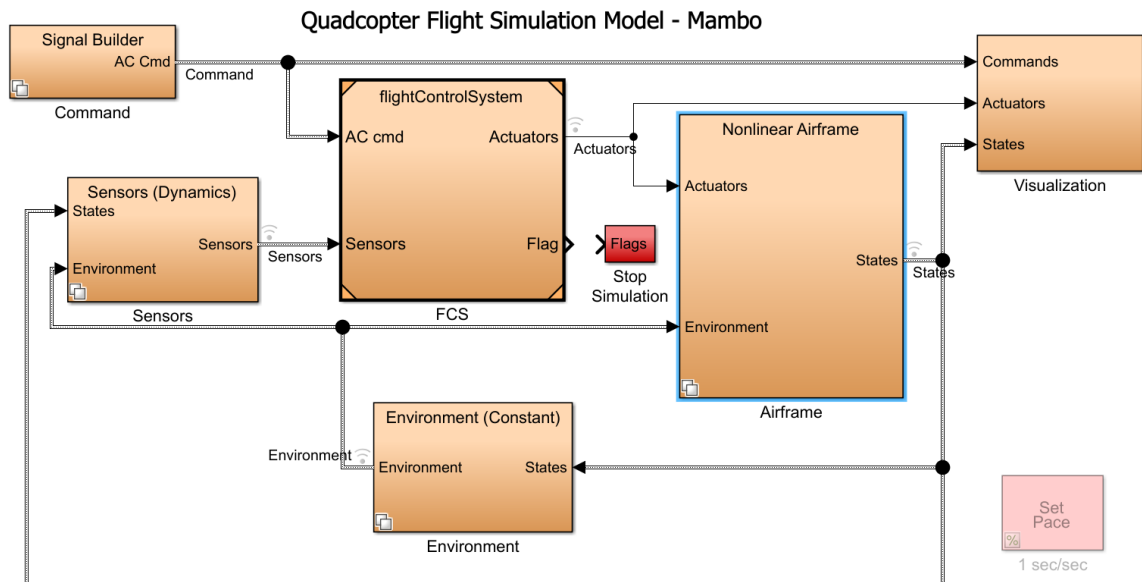


Figura 13-Modello a blocchi iniziale.

Nel blocco Airframe si ha tutta la dinamica lineare e non lineare ed è possibile scegliere quale usare per la simulazione. In Environment è presente la misurazione dell'ambiente e in Sensor

si ha la misurazione fornita dai sensori del mini-drone. Infine nel blocco Command si hanno i riferimenti ed in FCS è presente la logica di controllo.

Quindi come si può intuire, si è andati a lavorare principalmente nel blocco FCFS, più precisamente all'interno del sotto blocco Controller:

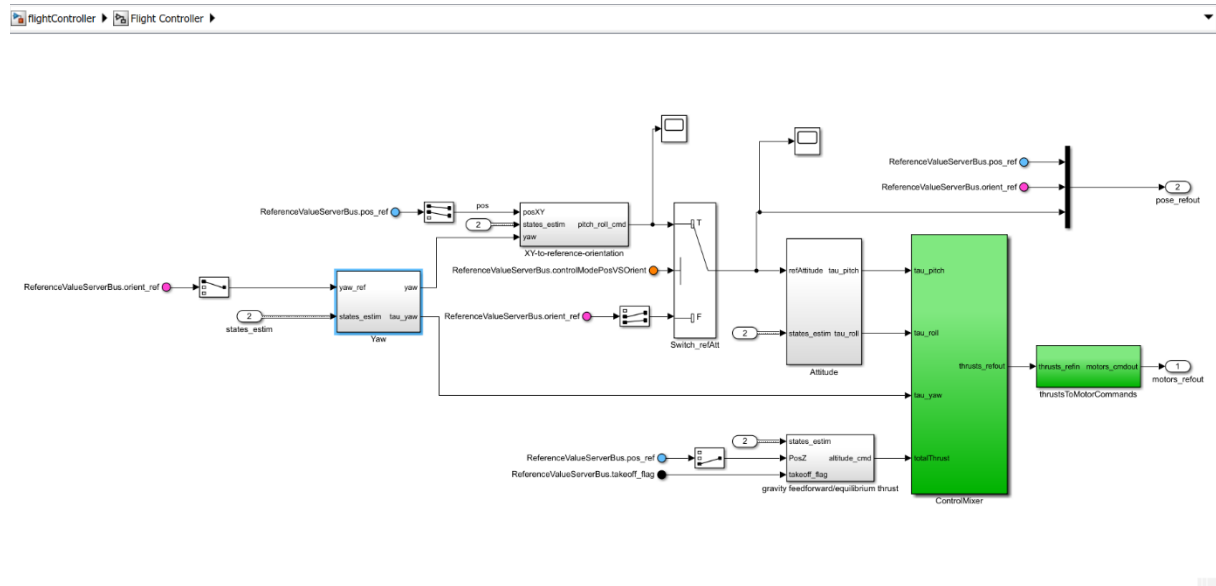


Figura 14-Sotto blocco dove risiedono i controllori dei 6 gradi di libertà.

Sempre all'interno del blocco FCFS vi è presente in particolare il sotto blocco StateEstimator dove mediante l'uso dei sensori precedentemente illustrati, il mini-drone acquisisce i valori delle variabili di stato :

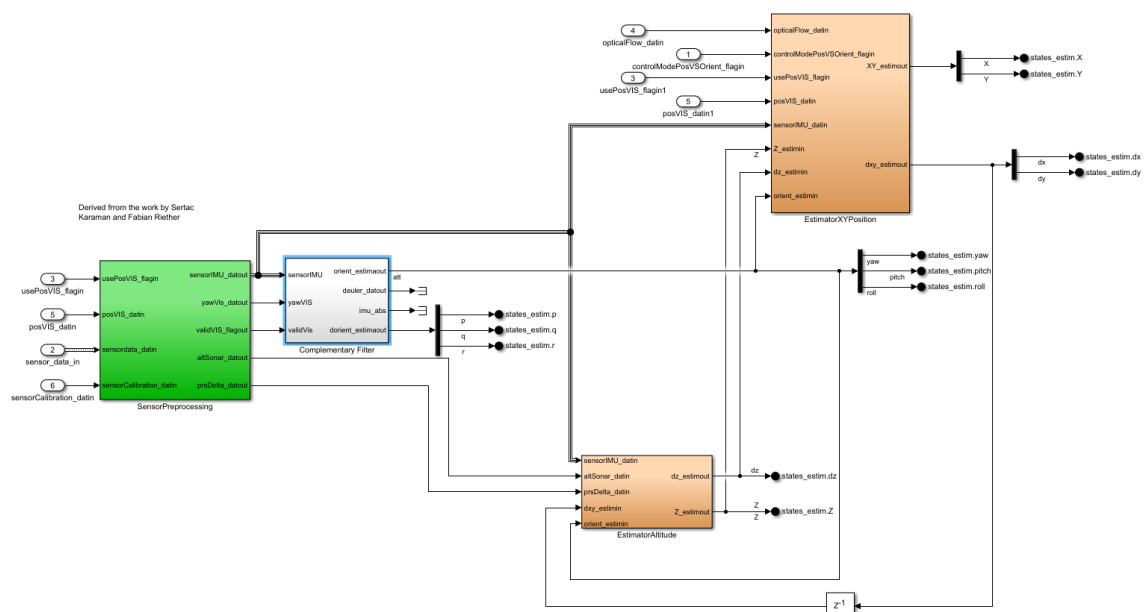


Figura 15-Sotto blocco State estimator.

La linearizzazione e le matrici A B C D sono già presenti in questo esempio all'interno del blocco Linear Airframe, che sta dentro Airframe:

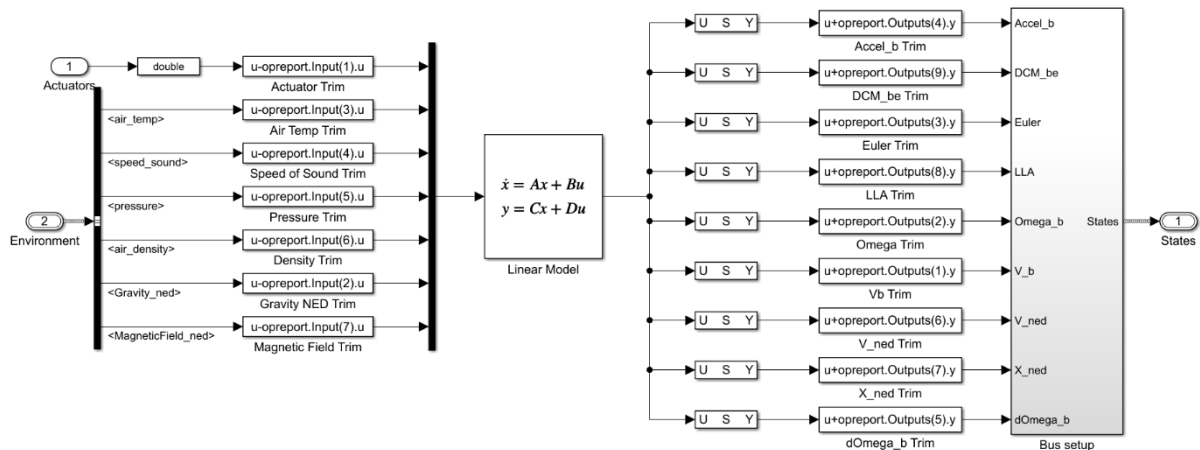


Figura 16-Modello lineare.

Nei workspace di Matlab, dentro “ linsys “ troviamo, inizialmente, le matrici che ci interessano, cioè quelle a tempo continuo. Una volta che si fa partire la simulazione queste vengono sovrascritte e poi diventano a tempo discreto.

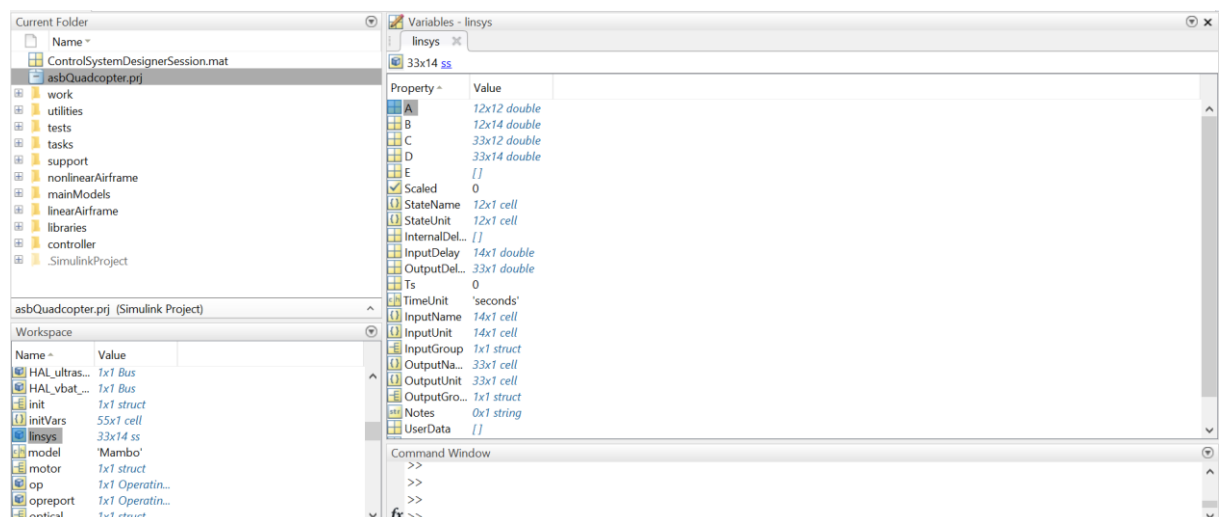
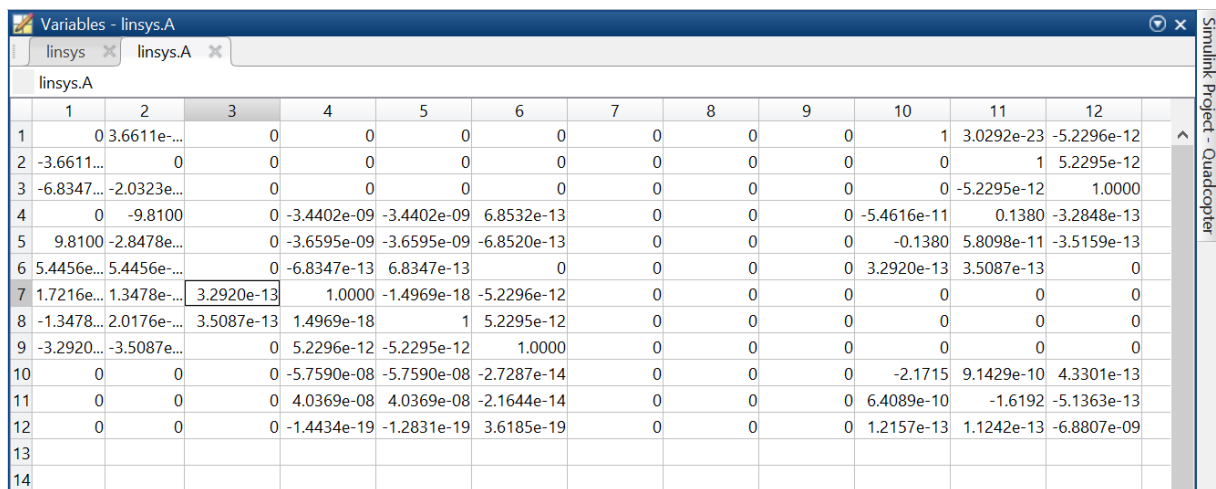


Figura 17-Variabili interessate su Matlab.

Come accennavo all'inizio dell'elaborato, abbiamo a che fare con matrici piuttosto corpose dato che abbiamo 12 stati, 14 ingressi, e 33 uscite.

La matrice A è la matrice dinamica (di stato) del sistema. B è la matrice di ingresso, C quella di uscita e infine D è la matrice di legame diretto ingresso-uscita del sistema.

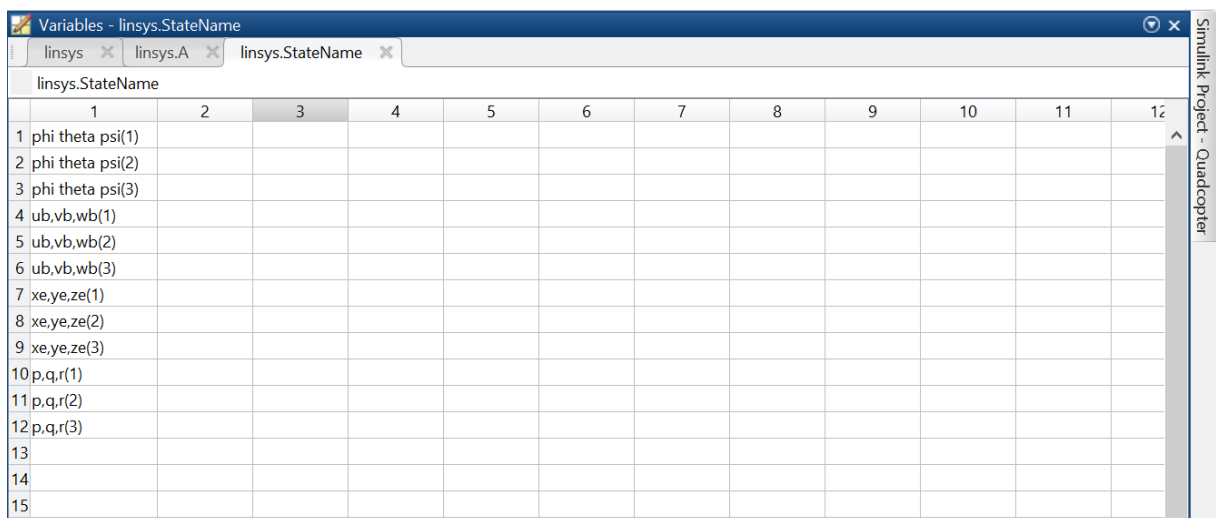
Vediamo che ci sono anche: StateName InputName e OutputName tra le voci di linsys; queste indicano a cosa corrisponde ogni riga della matrice, ad esempio per la matrice A si ha questa situazione:



	1	2	3	4	5	6	7	8	9	10	11	12
1	0	3.6611e-...	0	0	0	0	0	0	0	1	3.0292e-23	-5.2296e-12
2	-3.6611...	0	0	0	0	0	0	0	0	0	1	5.2295e-12
3	-6.8347...	-2.0323e...	0	0	0	0	0	0	0	0	-5.2295e-12	1.0000
4	0	-9.8100	0	-3.4402e-09	-3.4402e-09	6.8532e-13	0	0	0	-5.4616e-11	0.1380	-3.2848e-13
5	9.8100	-2.8478e...	0	-3.6595e-09	-3.6595e-09	-6.8520e-13	0	0	0	-0.1380	5.8098e-11	-3.5159e-13
6	5.4456e...	5.4456e-...	0	-6.8347e-13	6.8347e-13	0	0	0	0	3.2920e-13	3.5087e-13	0
7	1.7216e...	1.3478e-...	3.2920e-13	1.0000	-1.4969e-18	-5.2296e-12	0	0	0	0	0	0
8	-1.3478...	2.0176e-...	3.5087e-13	1.4969e-18	1	5.2295e-12	0	0	0	0	0	0
9	-3.2920...	-3.5087e...	0	5.2296e-12	-5.2295e-12	1.0000	0	0	0	0	0	0
10	0	0	0	-5.7590e-08	-5.7590e-08	-2.7287e-14	0	0	0	-2.1715	9.1429e-10	4.3301e-13
11	0	0	0	4.0369e-08	4.0369e-08	-2.1644e-14	0	0	0	6.4089e-10	-1.6192	-5.1363e-13
12	0	0	0	-1.4434e-19	-1.2831e-19	3.6185e-19	0	0	0	1.2157e-13	1.1242e-13	-6.8807e-09
13												
14												

Figura 18-Matrice A a tempo continuo.

Sapendo che la matrice A è la matrice degli stati, bisogna andare quindi a cliccare su StateName per vedere lo stato corrispondente a ogni riga della matrice.



	1	2	3	4	5	6	7	8	9	10	11	12
1	phi theta psi(1)											
2	phi theta psi(2)											
3	phi theta psi(3)											
4	ub,vb,wb(1)											
5	ub,vb,wb(2)											
6	ub,vb,wb(3)											
7	xe,ye,ze(1)											
8	xe,ye,ze(2)											
9	xe,ye,ze(3)											
10	p,q,r(1)											
11	p,q,r(2)											
12	p,q,r(3)											
13												
14												
15												

Figura 19- Variabili di stato.

Essendo una matrice 12×12 , e sapendo che è la matrice degli stati, sulle colonne si hanno gli stessi riferimenti delle righe. Vale la stessa cosa per le altre matrici.

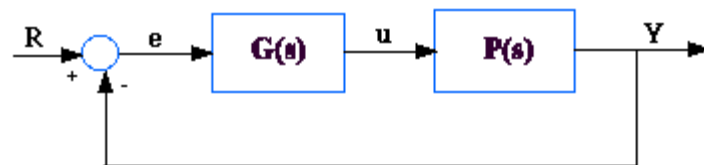
CONTROLLORE

Dopo aver parlato un po' di com'è fatto il modello Simulink e dove si trovano alcune informazioni utili, ora si parlerà del controllore da me realizzato.

L'obiettivo era quello di realizzare un controllore mediante una sintesi in frequenza sullo Yaw, uno dei 6 gradi di libertà.

SINTESI IN FREQUENZA

Considerando un generico sistema in catena chiusa:



per sintesi di un sistema si intende la scelta della funzione di trasferimento $G(s)$ affinché l'uscita soddisfi certe caratteristiche desiderate, in questo caso da me imposte. Queste caratteristiche vengono chiamate specifiche univoche o specifiche lasche, a seconda del tipo di specifica. Quelle univoche possono essere soddisfatte in un solo modo, mentre quelle lasche no, dipendono dalla forma.

Si può progettare il controllore $G(s)$ come meglio si pensa, a patto che esso sia realizzabile e stabile.

Questa è una sintesi per tentativi, o meglio per passi. Prima si soddisfano le caratteristiche univoche, e poi nel caso che le caratteristiche lasche non vengano soddisfatte, si procede a soddisfarle mediante la funzione compensatrice $R(s)$.

Quindi si ha che :

$$G(s) = \frac{K_G}{s^h} R(s) \quad (60)$$

dove $R(s)$, nel mio caso, è una anticipatrice :

$$R_a(s) = \frac{1 + \tau_a s}{1 + \frac{\tau_a}{m_a} s} \quad (61)$$

I parametri K_G e h vanno scelti in modo tale da soddisfare le specifiche sul regime permanente, mentre con $R(s)$ si impongono le specifiche sulla risposta armonica. $R(s)$ viene normalmente presa a guadagno unitario in modo tale che il guadagno di $G(s)$ coincida con K_G .

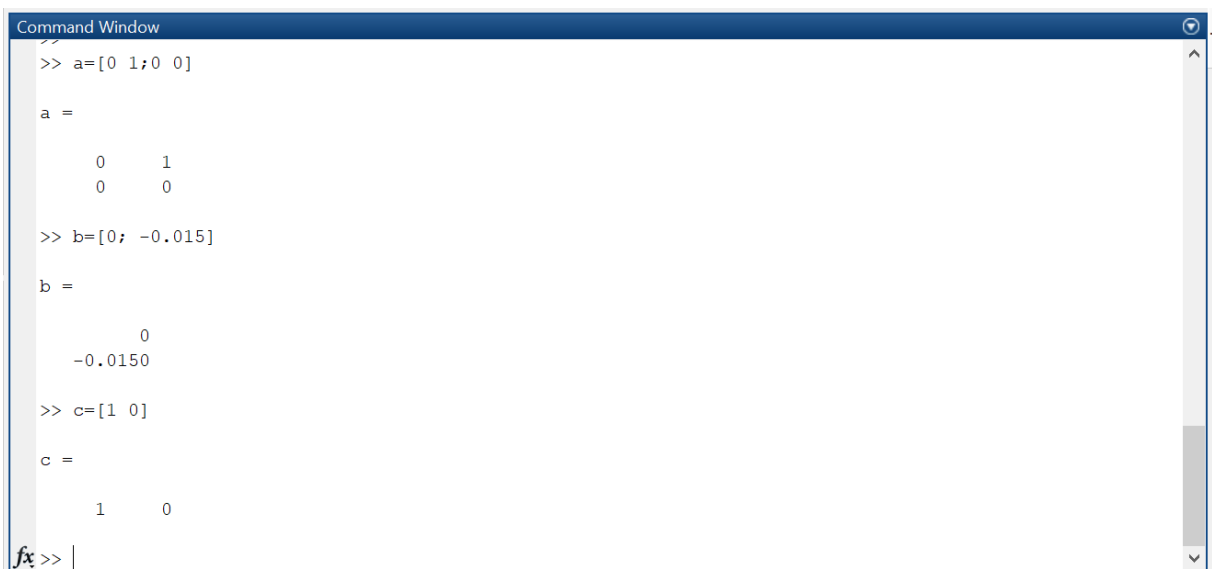
Per la sintesi in frequenza si usano strumenti come: diagrammi di Bode, carta di Nichols e diagrammi universali i quali servono per la funzione compensatrice. Può essere usato anche il diagramma di Nyquist per studiare la stabilità. Questi strumenti sono presenti su Matlab richiamando funzione chiamata “sisotool”.

PROCESSO $P(s)$

Prima di lavorare sul controllore, bisogna ricavarsi la funzione di trasferimento, che descrive l'uscita del sistema, quindi rappresenta il processo $P(s)$ che si andrà a controllare.

Abbiamo già il sistema dinamico, e quindi le matrici A B C D , però queste matrici sono abbastanza corpose dato che contengono tutte le variabili di stato, tutti gli ingressi e tutte le uscite disponibili. Siccome quello che in questo caso interessa è lo yaw, bisogna andare a scorporare le matrici interessate, considerando solo lo yaw e r (velocità angolare) come variabili di stato, i 4 motori come ingresso per poi andare a vedere quali uscite vengono coinvolte.

Il risultato è :



```
Command Window
>> a=[0 1;0 0]

a =

     0     1
     0     0

>> b=[0; -0.015]

b =

     0
    -0.0150

>> c=[1 0]

c =

     1     0

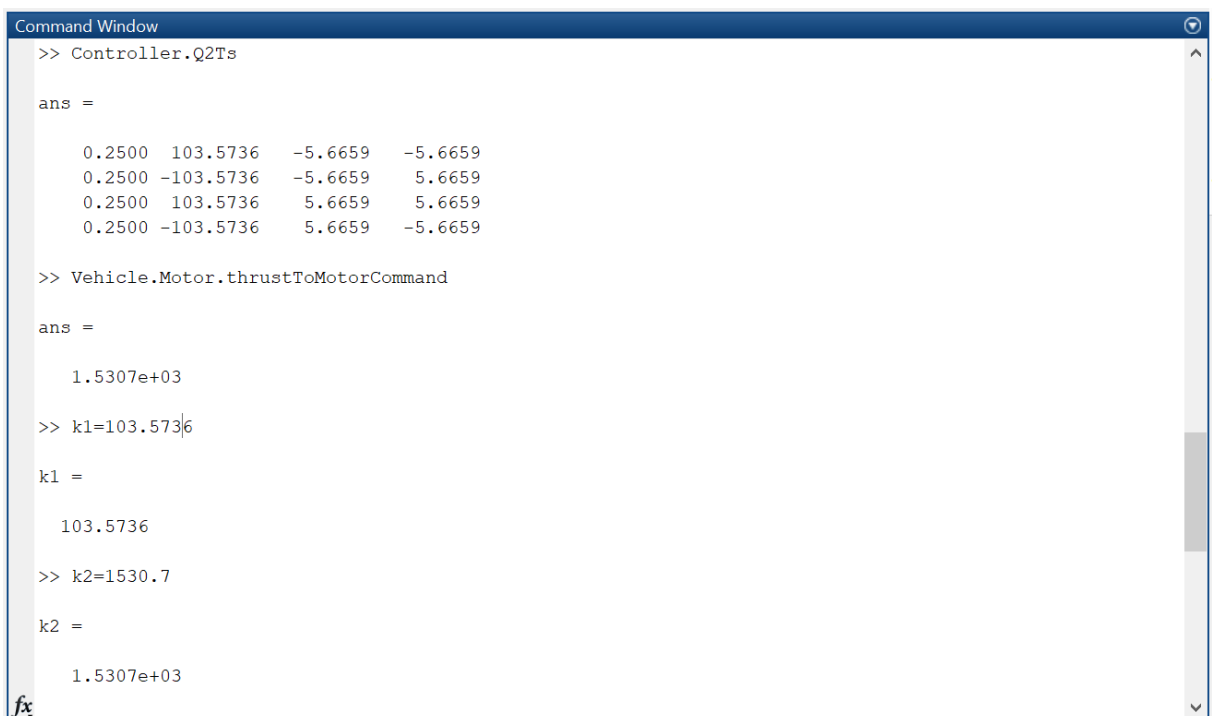
fx >> |
```

Figura 20-Sotto matrici interessate.

La matrice D in questo caso è composta da zeri, quindi non la si considera.

Ora, bisogna considerare anche due coefficienti, k_1 e k_2 , in modo da avere la funzione di trasferimento più precisa:

Controller.Q2Ts, che è una matrice che fa la partizione tra il nostro segnale e i motori, e Vehicle.Motor.thrustToMotorCommand .



```
Command Window
>> Controller.Q2Ts

ans =

    0.2500    103.5736   -5.6659   -5.6659
    0.2500   -103.5736   -5.6659    5.6659
    0.2500    103.5736    5.6659    5.6659
    0.2500   -103.5736    5.6659   -5.6659

>> Vehicle.Motor.thrustToMotorCommand

ans =

    1.5307e+03

>> k1=103.5736

k1 =

    103.5736

>> k2=1530.7

k2 =

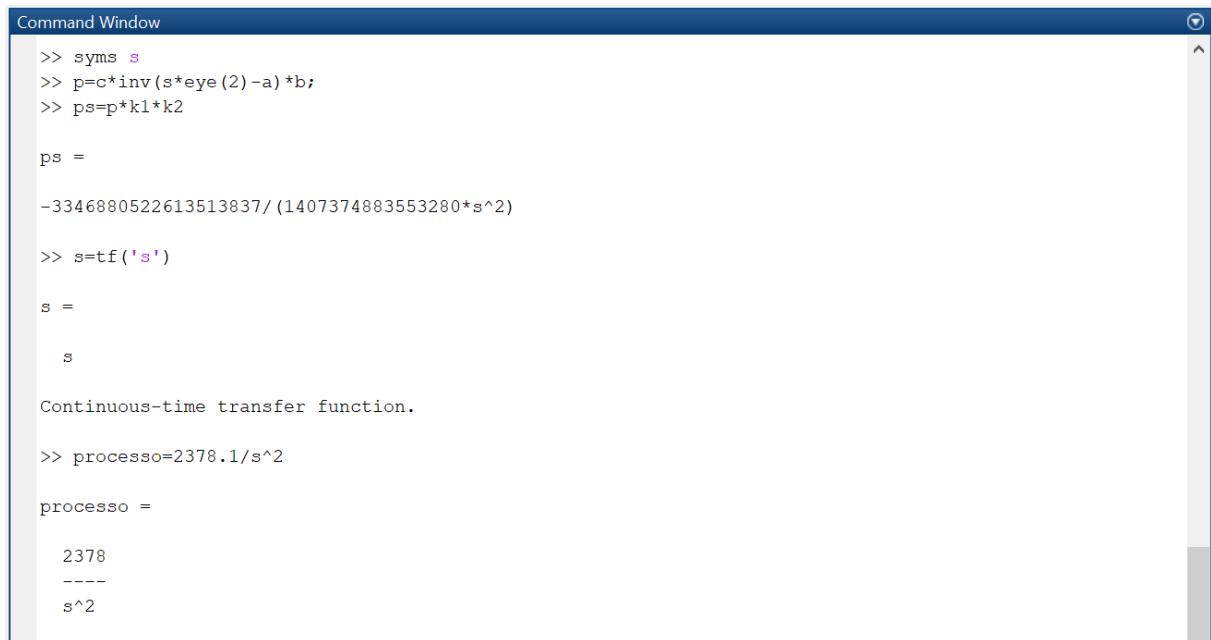
    1.5307e+03

fx
```

Figura 21-Costanti k_1 e k_2

Per k_1 si prende 103.5737 perché la seconda colonna corrisponde allo yaw, mentre la prima colonna corrisponde alla spinta, la terza al pitch e la quarta a roll.

Una volta ottenuti tutti gli elementi necessari, si può calcolare la funzione di trasferimento :



```
Command Window
>> syms s
>> p=c*inv(s*eye(2)-a)*b;
>> ps=p*k1*k2

ps =

-3346880522613513837/(1407374883553280*s^2)

>> s=tf('s')

s =

s

Continuous-time transfer function.

>> processo=2378.1/s^2

processo =

    2378
    ----
    s^2
```

Figura 22- Righe di codice per ottenere la funzione di trasferimento su Matlab

PROGETTAZIONE DEL CONTROLLORE

Si è ottenuto un processo con due poli in zero, quindi astatico rispetto a eventuali disturbi in uscita.

Le caratteristiche imposte sono:

- tipo 2;
- errore ≤ 0.01 ;
- tempo di salita = 0.3 s;
- margine di risonanza $\leq 2\text{db}$.

Traducendole si ottiene:

- un margine di fase $m_{\varphi} = 47^{\circ}$;
- una banda passante $B_3 = 10\text{Hz}$;
- una pulsazione di attraversamento $\omega_t = 40$.

Inizialmente si ha una situazione di questo tipo :

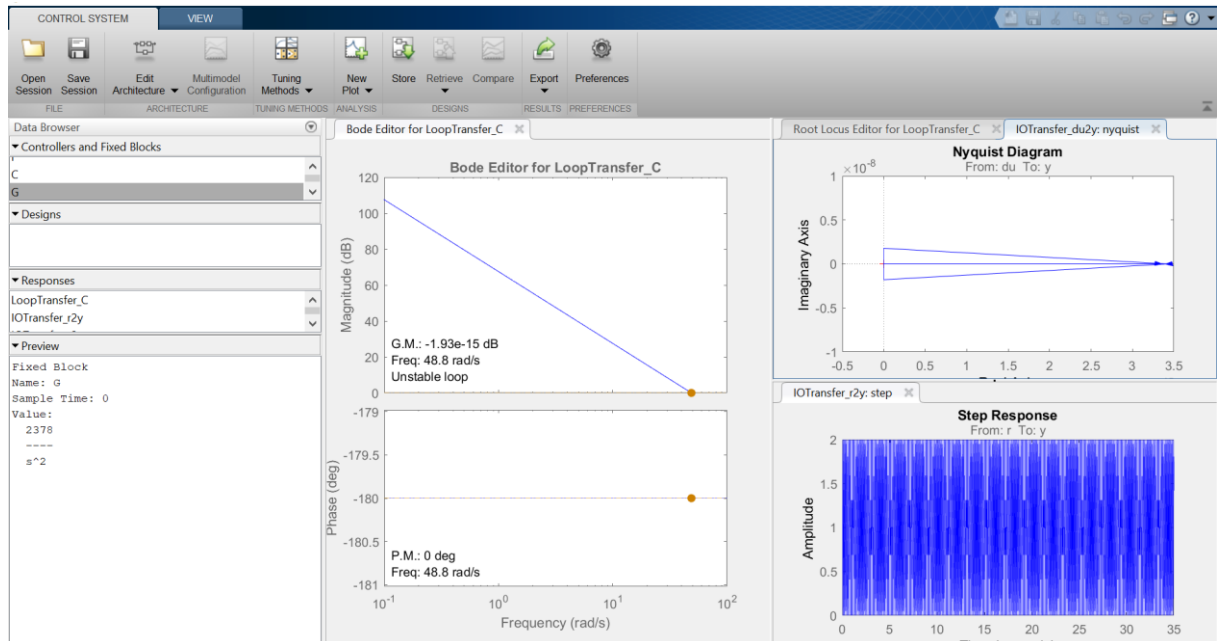


Figura 23- Ambiente Sisotool.

Dopo aver fatto un primo tentativo, impostando un guadagno $K_g = 0.0184$, si è visto che le specifiche lasche erano ancora insoddisfatte, quindi si è passati a un secondo tentativo con una funzione anticipatrice, e si è ottenuto:

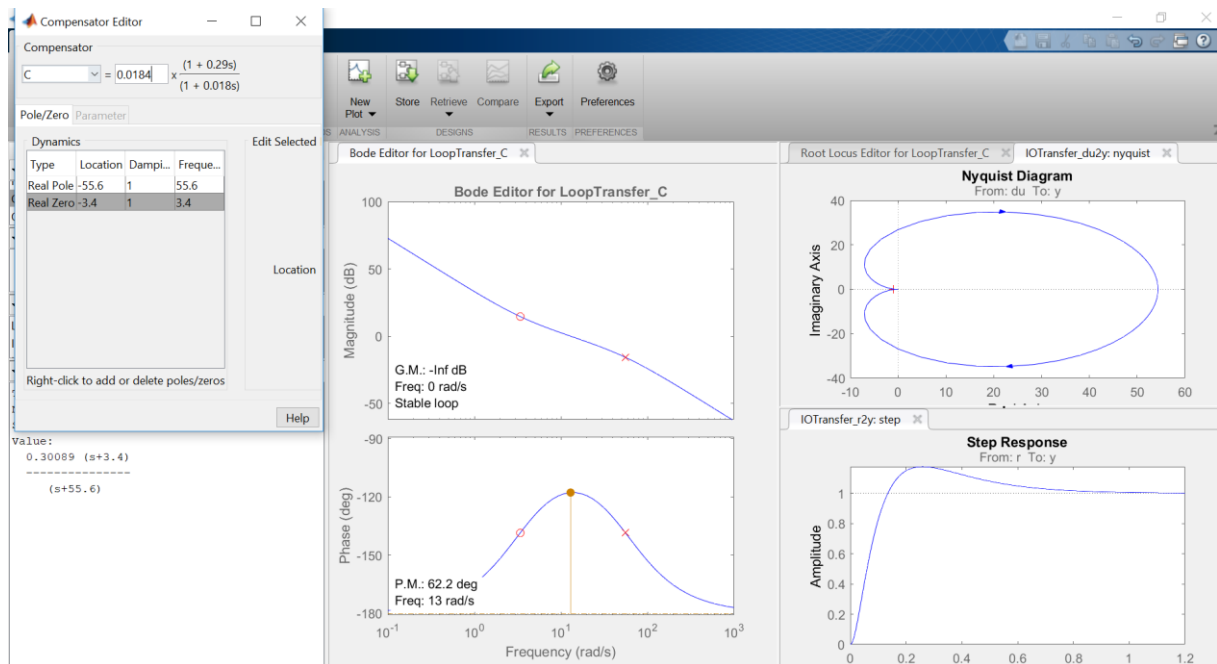


Figura 24- Progettazione del controllore, con il valore del polo e dello zero aggiunto sul Compensator Editor.

soddisfando così le specifiche e ottenendo un uscita stabile.

CONTROLLORE ORIGINARIO PID SU MODELLO SIMULINK

I controllori originari che sono presenti sul modello, sono una cascata di PID, più precisamente nel caso dello yaw si ha a che fare con un controllore PD, così implementato:

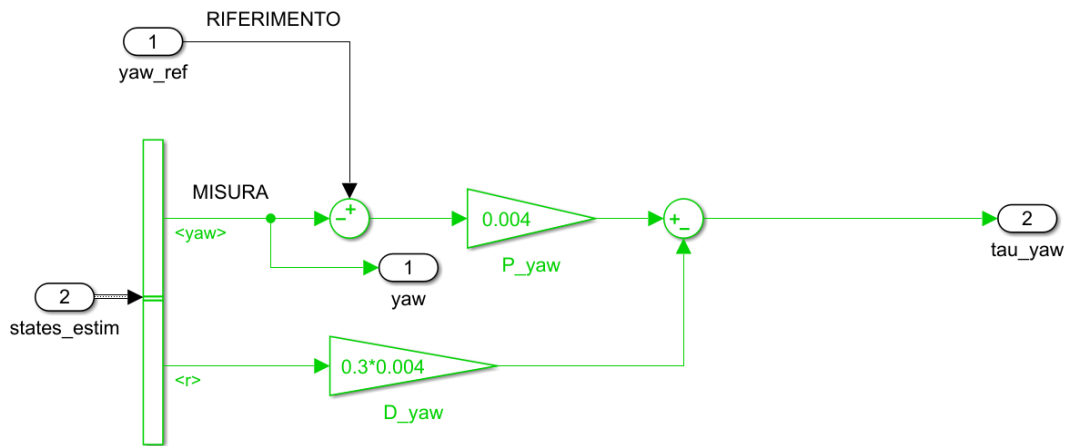


Figura 25- Controllore PD originario dello Yaw.

Da “ states_estim ” si ottengono le variabili di stato e da “ yaw_ref ” si ottiene il riferimento il quale è espresso in radianti ed è pari a 0.5, ovvero circa 28.85°.

La misura, oltre che finire sul nodo sommatore, in modo da correggere poi l'errore con il riferimento, va poi a finire anche sull'uscita “ yaw ” poiché serve al controllo di posizione di X e Y in un altro sotto blocco.

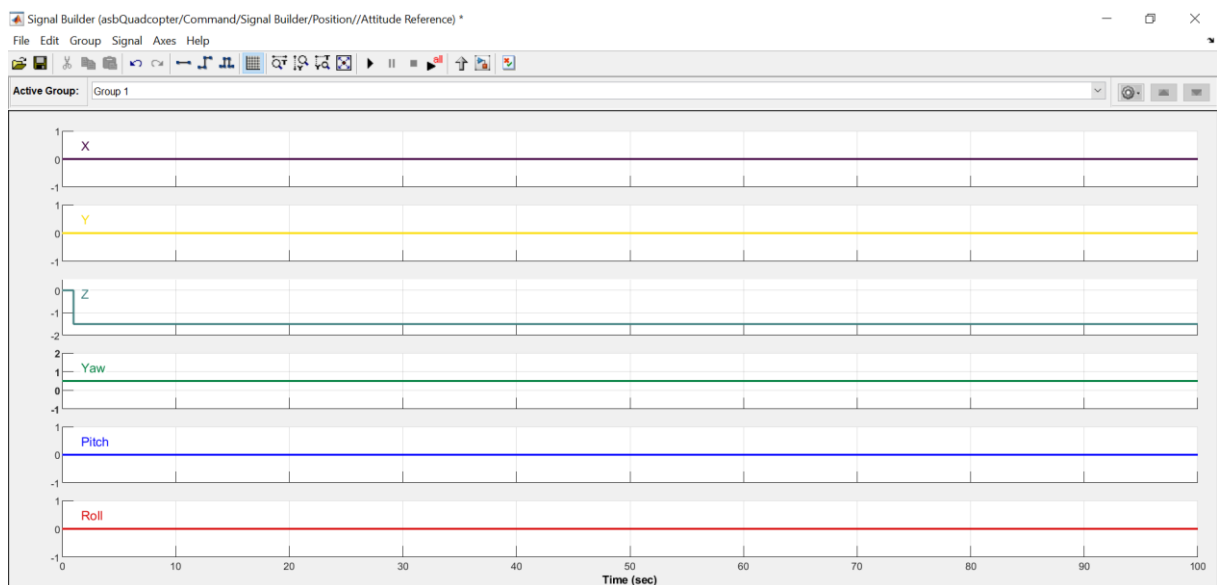


Figura 26-Riferimenti di controllo per i 6 gradi di libertà.

CONTROLLORE PID CASO NON LINEARE

Nel caso non lineare, che è quello impostato in default, lanciando la simulazione si ottengono i seguenti risultati :

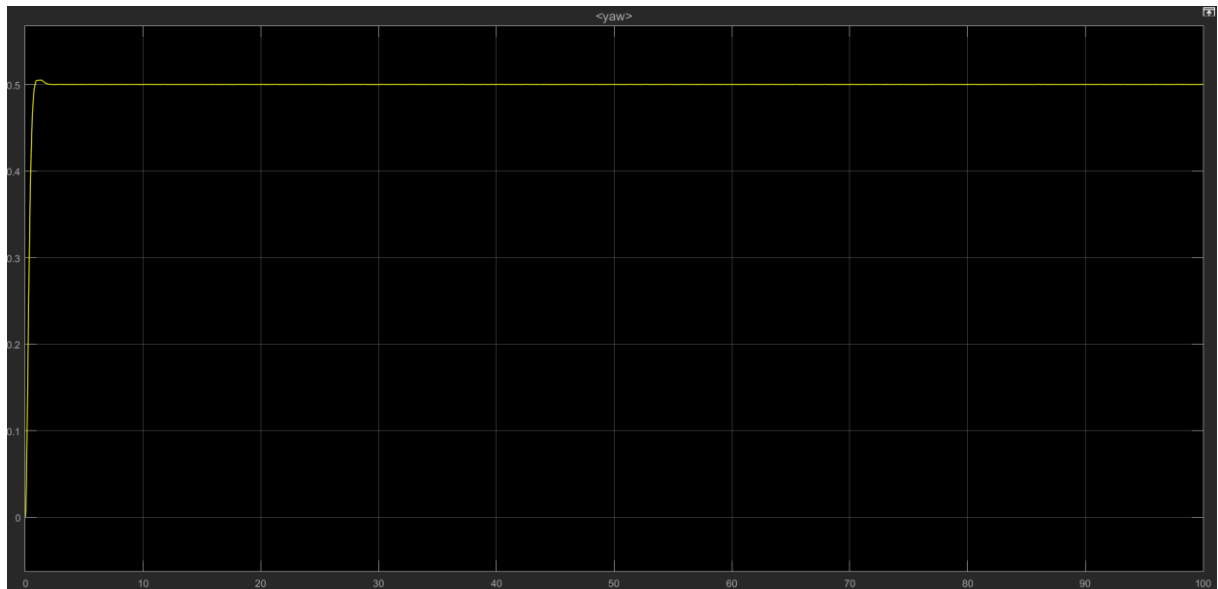


Figura 27-uscita di Yaw.

Il controllore originario ha un tempo di salita molto basso e una sovraelongazione molto piccola. La variabile controllata risulta stabilizzarsi in un tempo molto breve e quindi si può dire che il controllore con i PD funziona molto bene.



Figura 28-Coppia, uscita di G(s).

La figura 28 rappresenta l'uscita del controllore, cioè la spinta necessaria ai motori per raggiungere una certa posizione di riferimento. Questa è pari a zero quando il mini-drone si stabilizza sulla posizione di riferimento, ed è diversa da zero solo in un breve periodo di tempo, ovvero quello necessario al Mambo per raggiungere tale posizione.

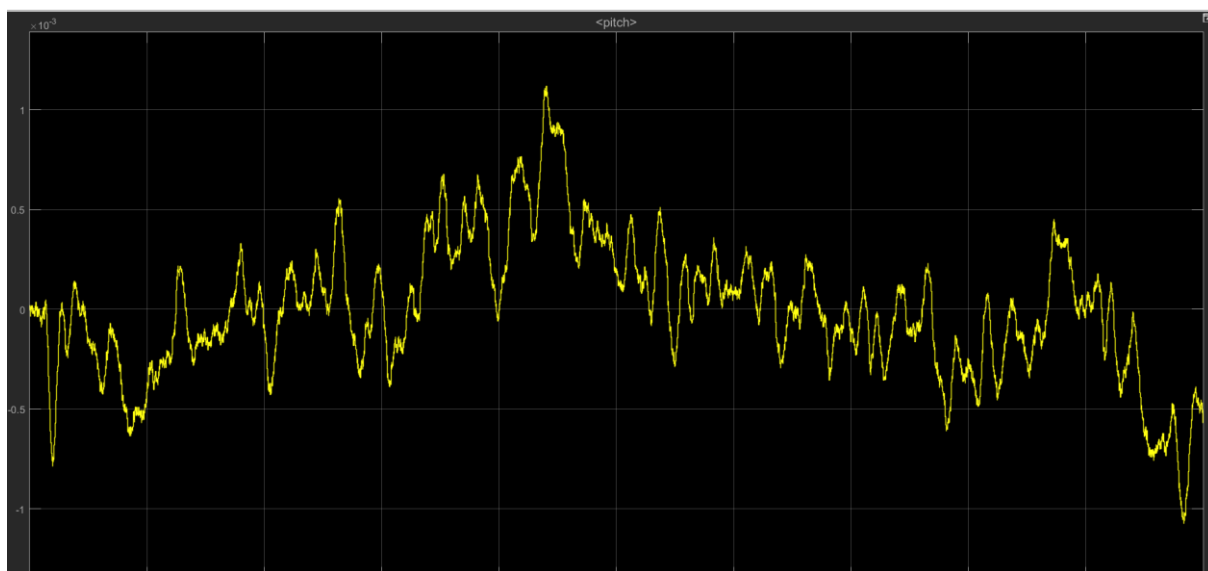


Figura 29-uscita di pitch.

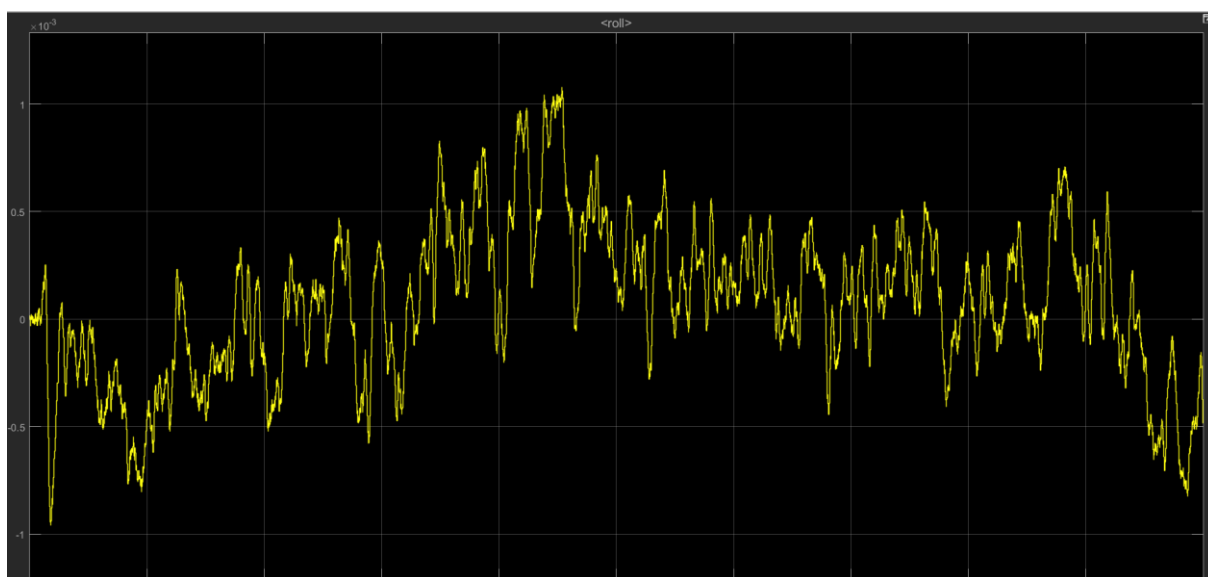


Figura 30-Uscita di roll.

Le oscillazioni delle uscite di pitch e roll, dove il controllo è realizzato mediante un PID in comune, risultano molto basse. Il loro andamento assume valori molto vicini allo zero, dato che sono dell'ordine di 10^{-3} .

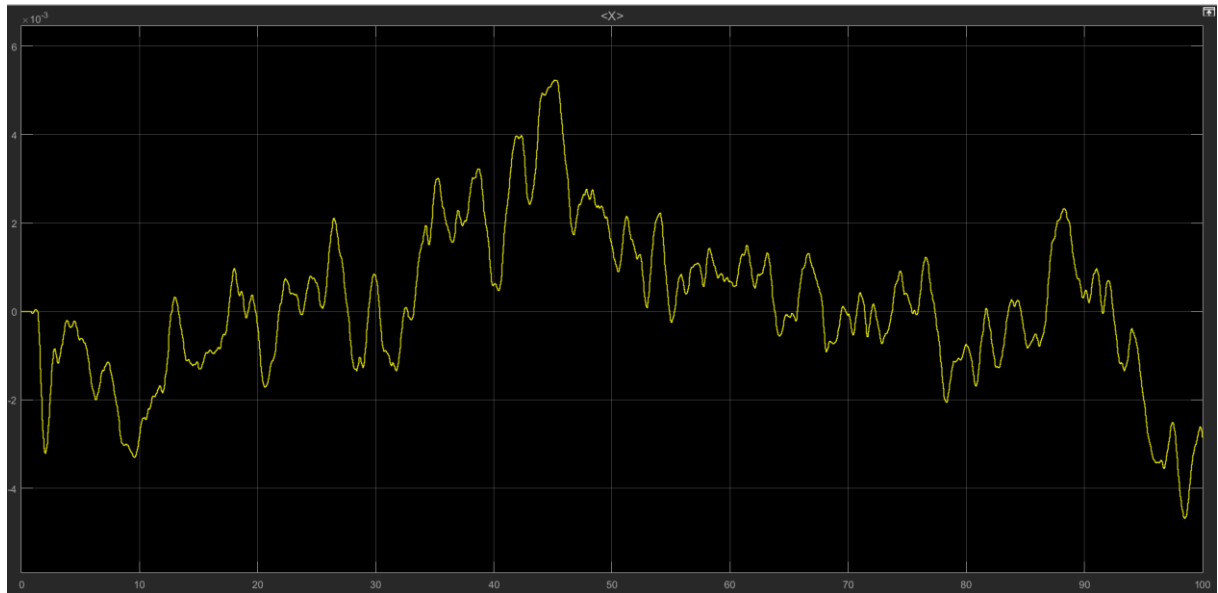


Figura 31-uscita di x.

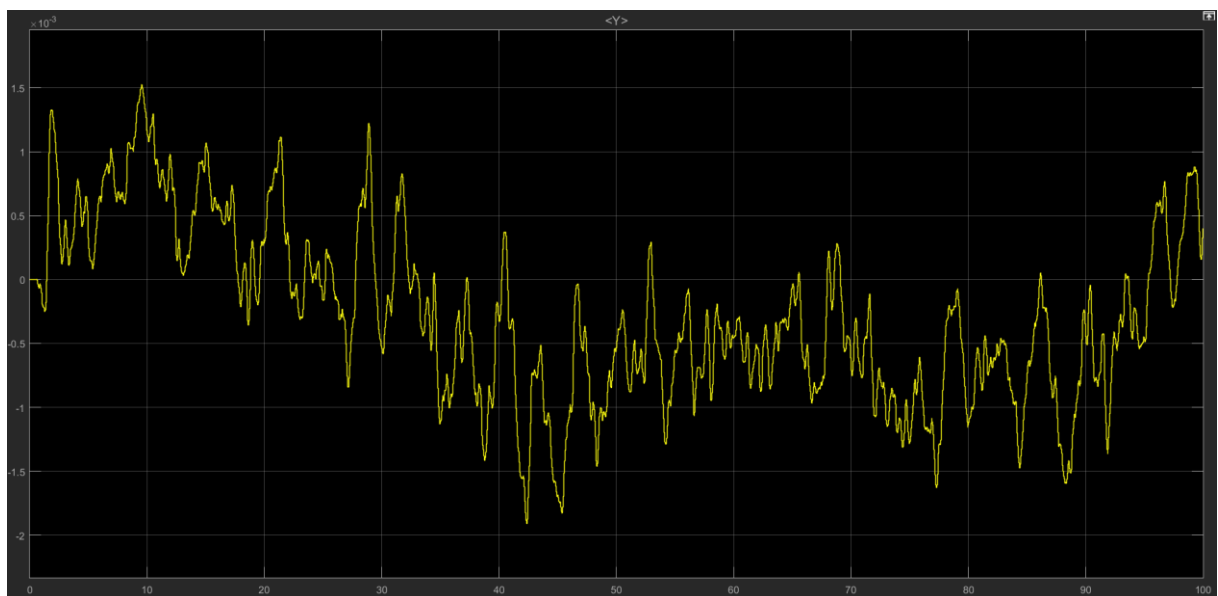


Figura 32-Uscita di y.

Anche qua le oscillazioni sono dell'ordine di 10^{-3} , quindi tendenti allo zero, come il loro riferimento. Questo afferma che il controllore realizzato con un PD dello yaw è stato progettato bene, dato che il controllo sulla posizione X e Y dipende da quello dello yaw.

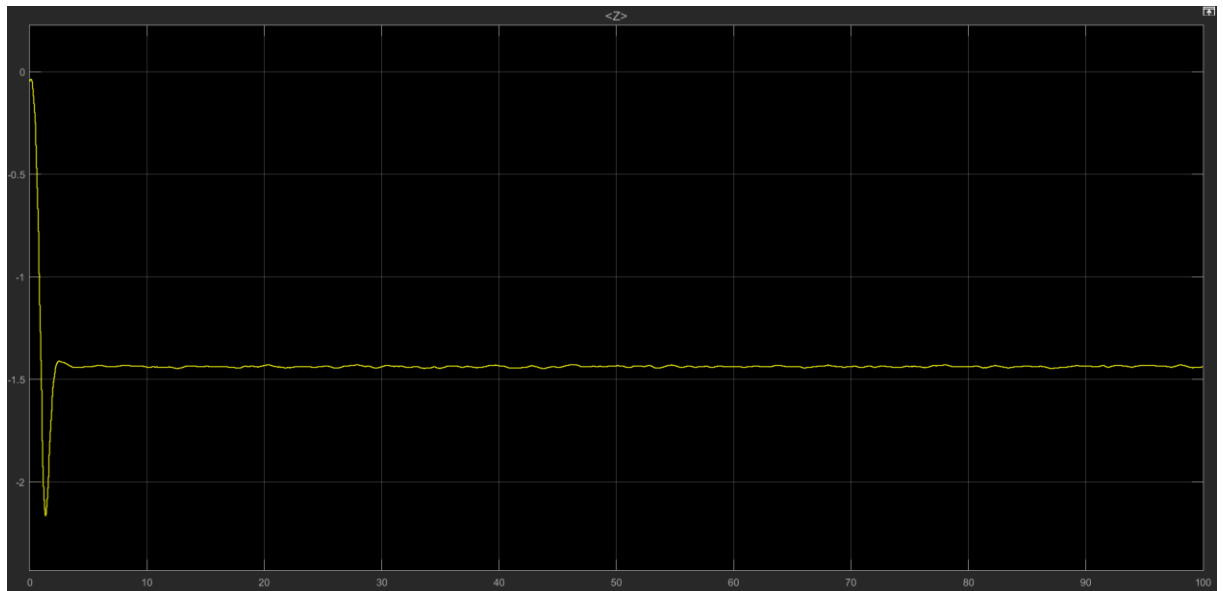


Figura 33-Uscita altitudine.

Il controllo dell'altezza risulta stabile. Dal grafico vediamo che il drone si stabilizza a un valore di poco inferiore a 1.5 m dal suolo, si avvicina al valore voluto dal riferimento imposto.

CONTROLLORE PID CASO LINEARE

Nel caso lineare, ovvero lanciando sul Command window di Matlab il comando “ VSS_VEHICLE=0 ”, facendo partire la simulazione si ottiene lo stesso comportamento del caso non lineare, con valori pressoché uguali come si vede dai grafici sotto riportati:

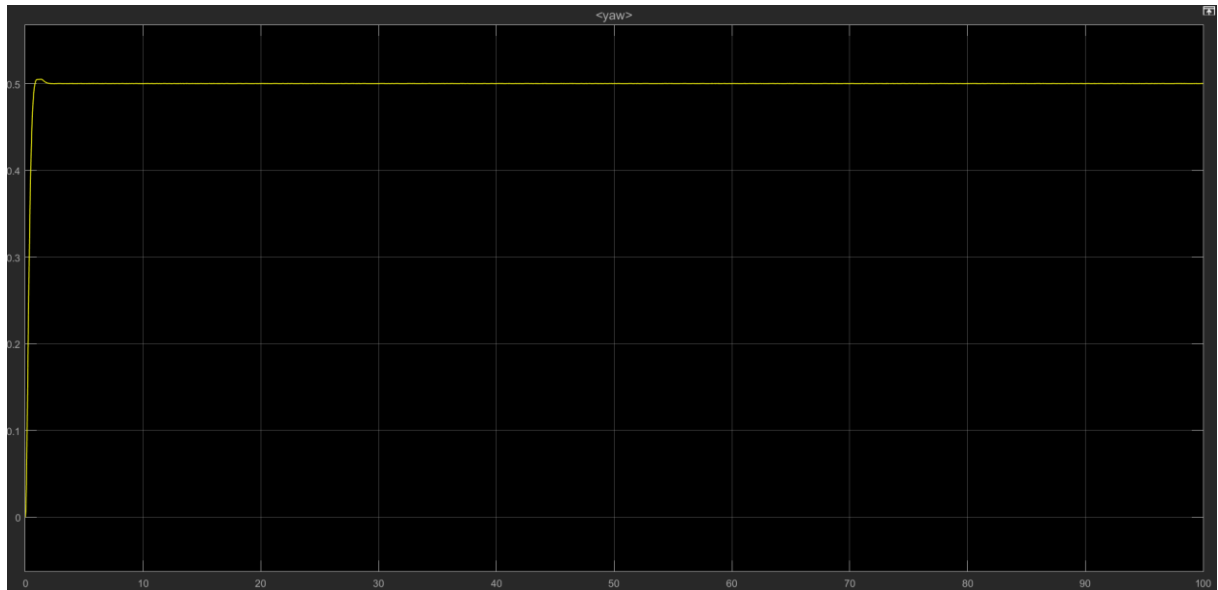


Figura 34-Uscita di yaw.



Figura 35-Coppia di yaw.

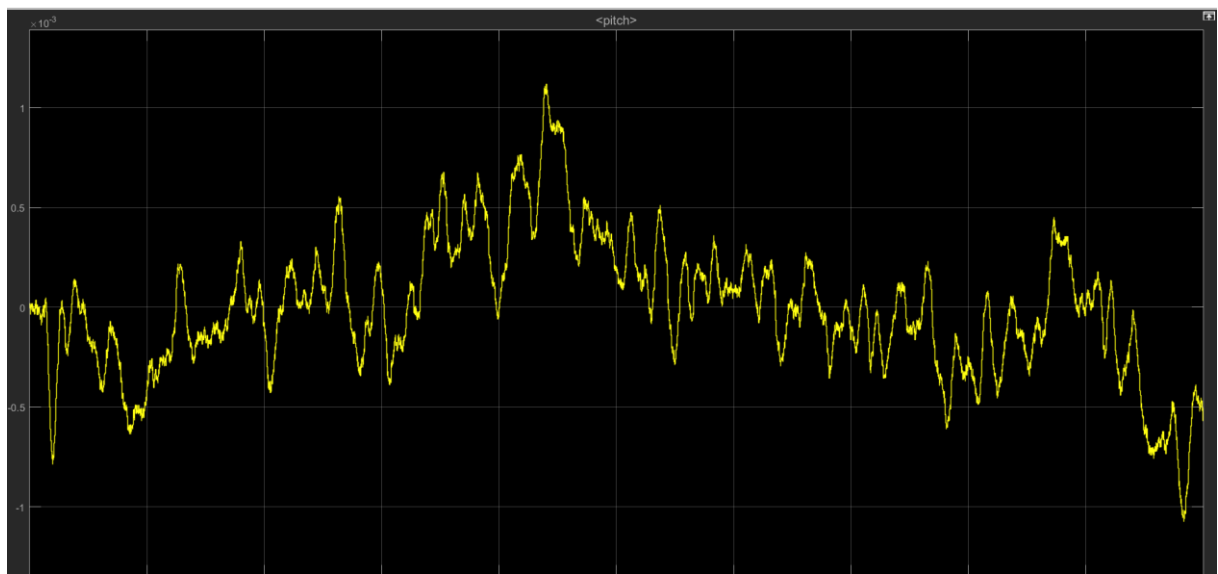


Figura 36-Uscita di pitch.

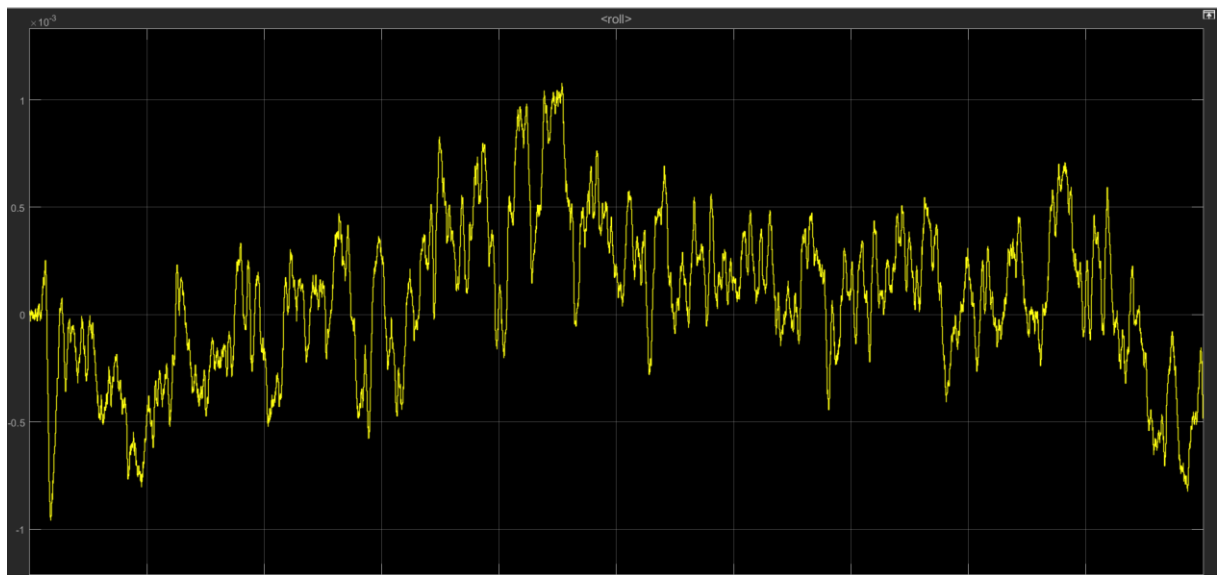


Figura 37-Uscita di roll.

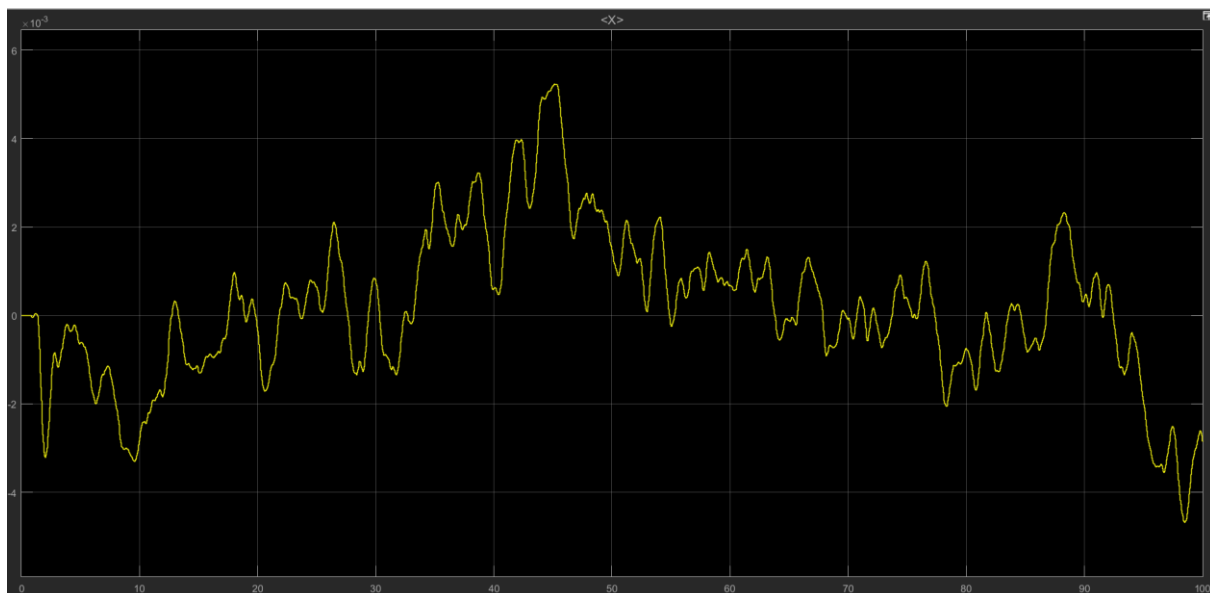


Figura 38-Uscita di x.

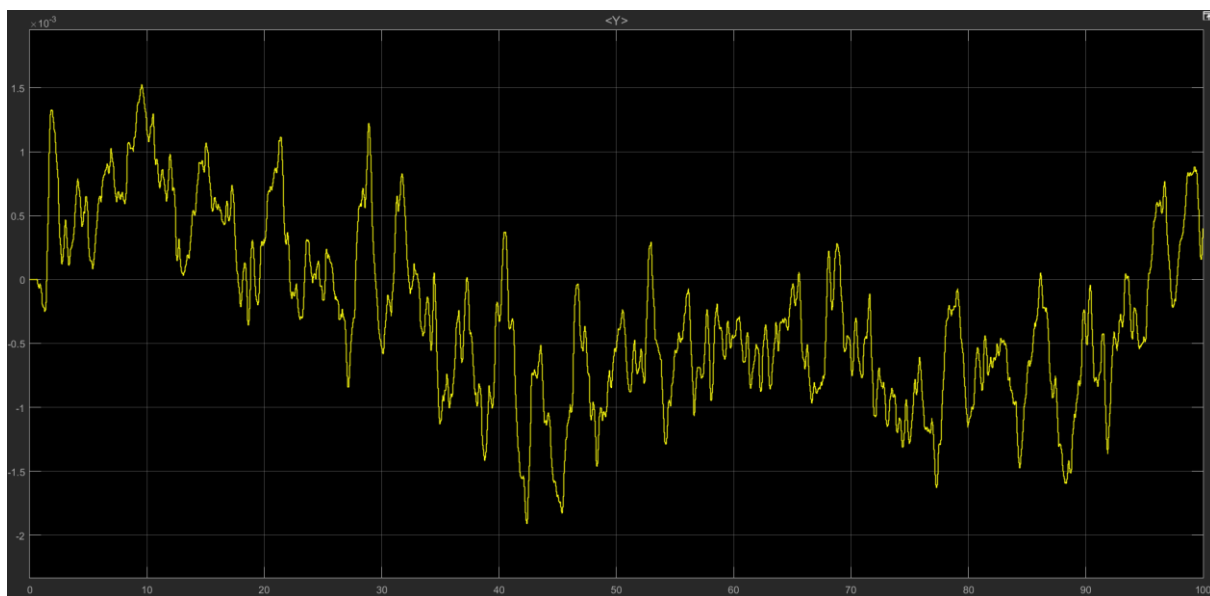


Figura 39- Uscita di y.

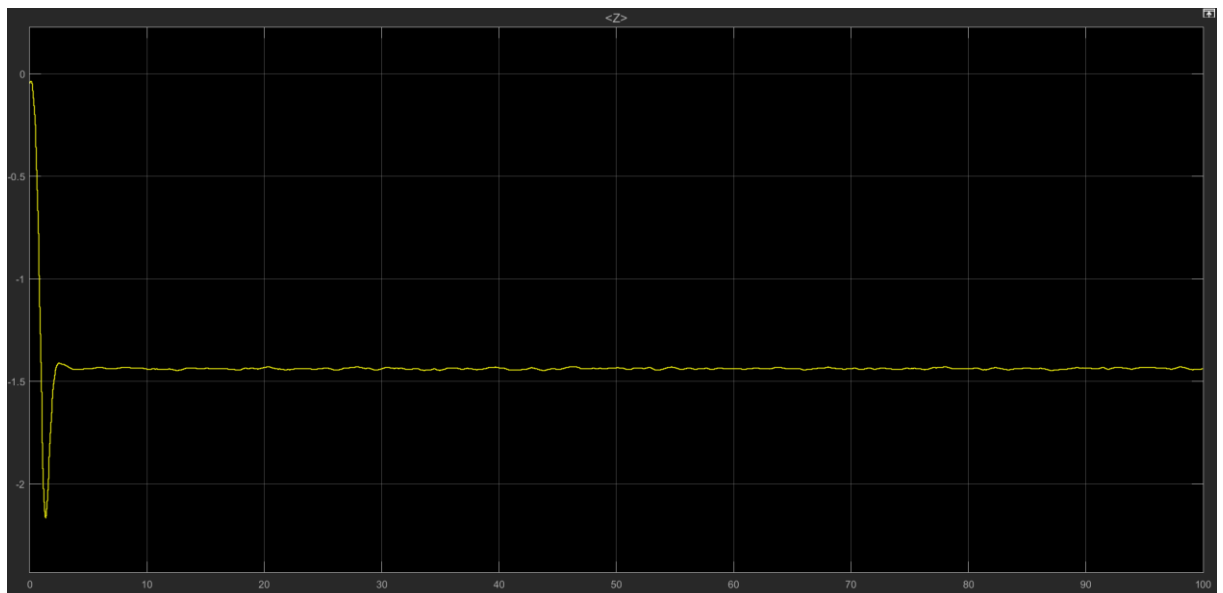


Figura 40-Altitudine.

Per quanto riguarda il controllore in frequenza da me progettato, è stato così creato su Simulink:

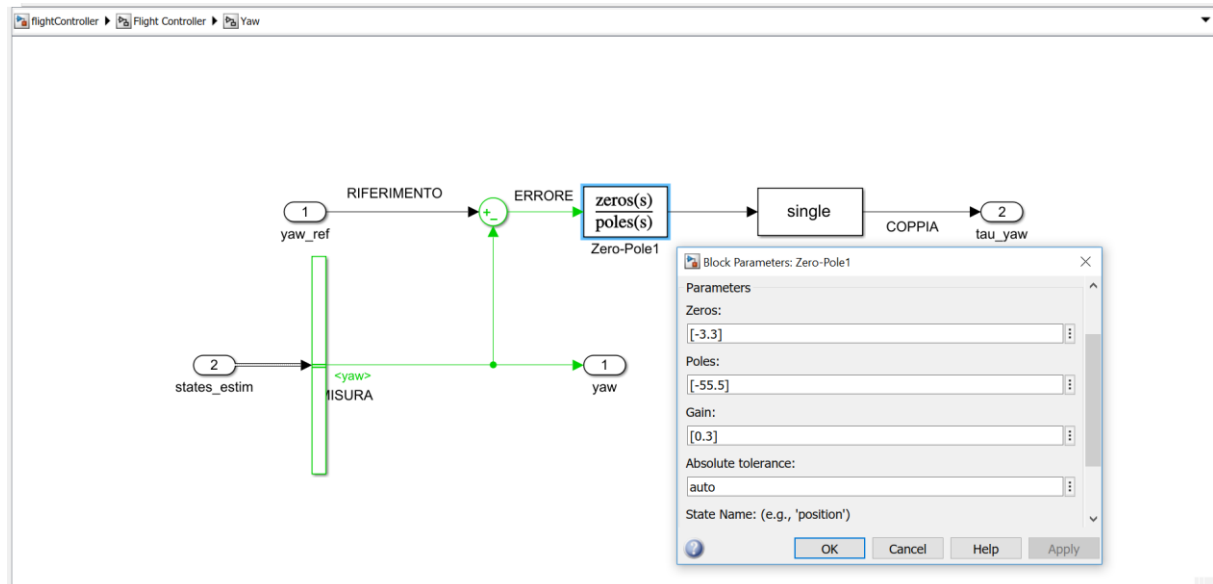


Figura 41-Diagramma a blocchi del nuovo controllore in frequenza con il valore del polo, dello zero e K_G .

dove “yaw_ref” è lo stesso del PID, mentre da “states_estim” ci basta selezionare solo lo stato di yaw. La misura, oltre che finire sul nodo sommatore, in modo da correggere poi l’errore con il riferimento, va poi a finire anche sull’uscita “yaw” poiché serve al controllore della posizione di X e Y in un altro sotto blocco. Il blocco single trasforma l’uscita dal blocco zero-pole in una variabile di tipo single, anziché double. L’uscita del controllore altro non è che la coppia necessaria ai motori per rimanere nella posizione di riferimento. Quest’uscita deve essere uguale a zero quando il Mambo sta nella posizione desiderata di 0.5 radianti, e deve essere diversa da zero nel periodo di tempo che il mini-drone impiega per raggiungere tale valore.

CONTROLLORE IN FREQUENZA CASO NON LINEARE

Lanciando la simulazione, si ottengono i seguenti risultati:

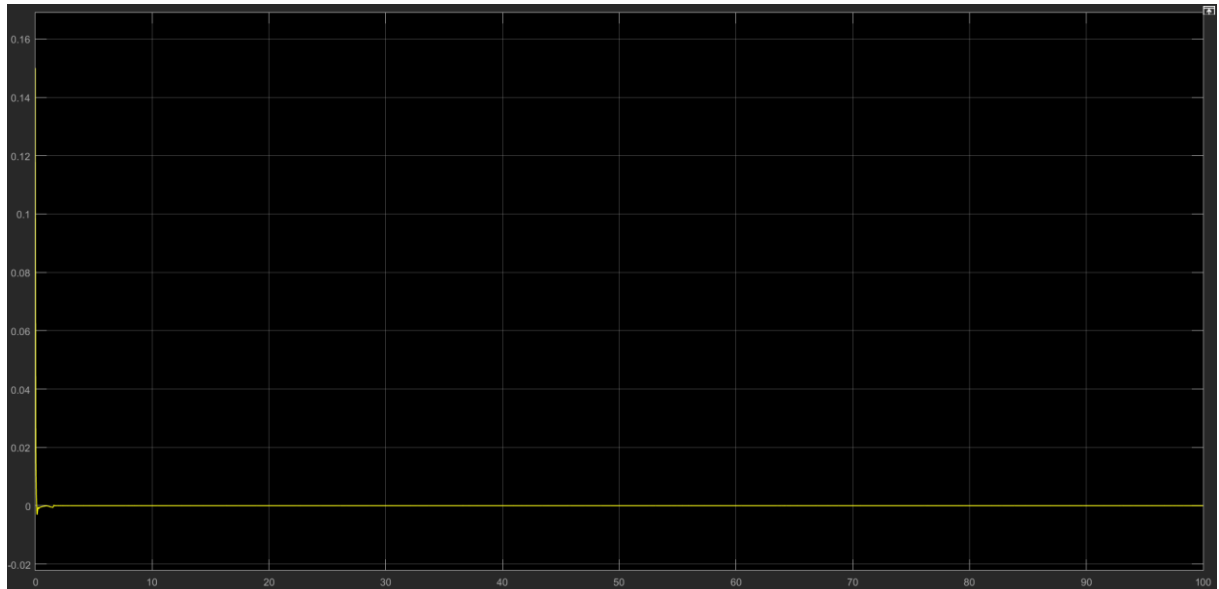


Figura 42- Coppia di yaw.

La figura sovrastante rappresenta la coppia, l'uscita del controllore. Questa è pari a zero quando il mini-drone si stabilizza sulla posizione di riferimento, ed è diversa da zero solo in un breve periodo di tempo, ovvero quello necessario al Mambo per raggiungere tale posizione.

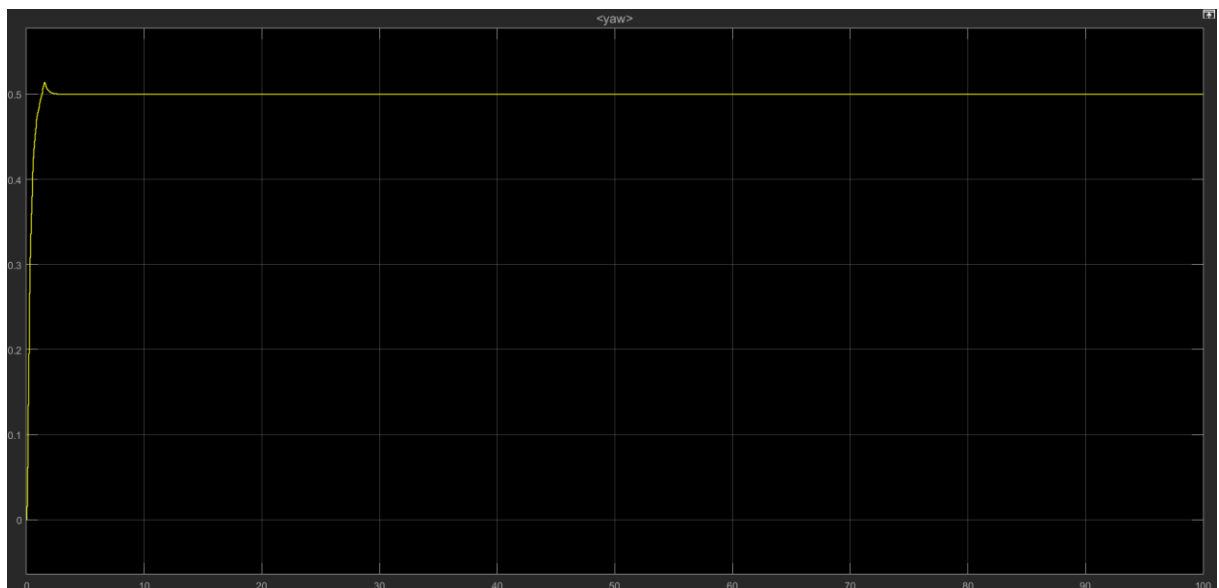


Figura 43- Uscita controllata.

Sopra abbiamo l'andamento della variabile controllata la quale risulta stabile, con un tempo di salita breve e una sovraelongazione bassa, quindi possiamo dire di aver raggiunto l'obiettivo di controllo.

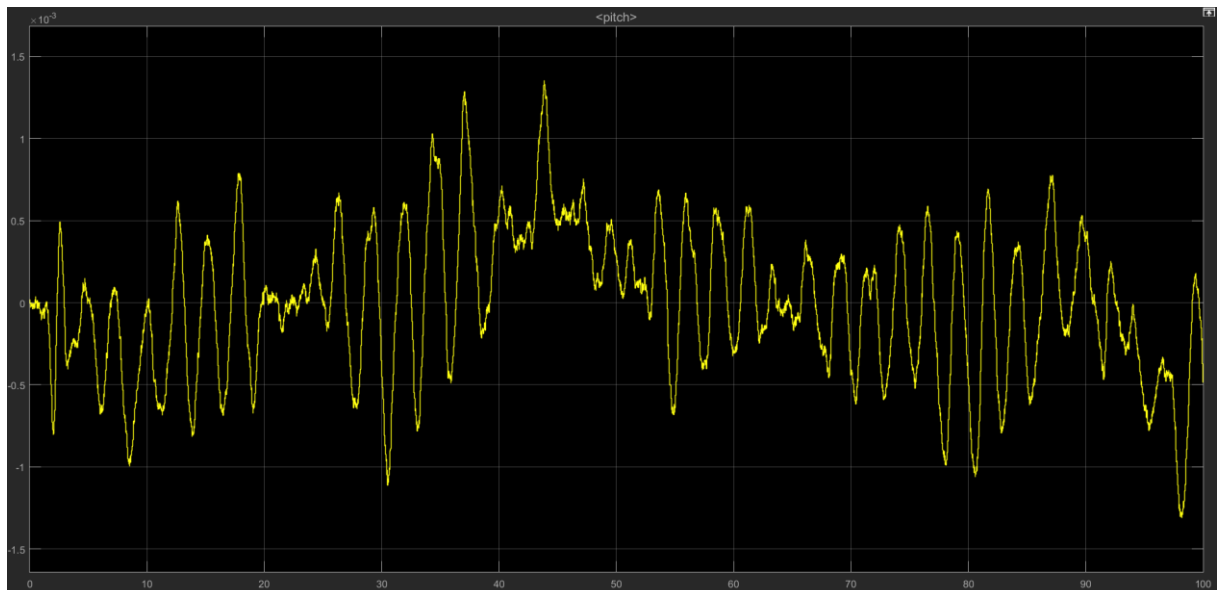


Figura 44-Uscita di pitch.

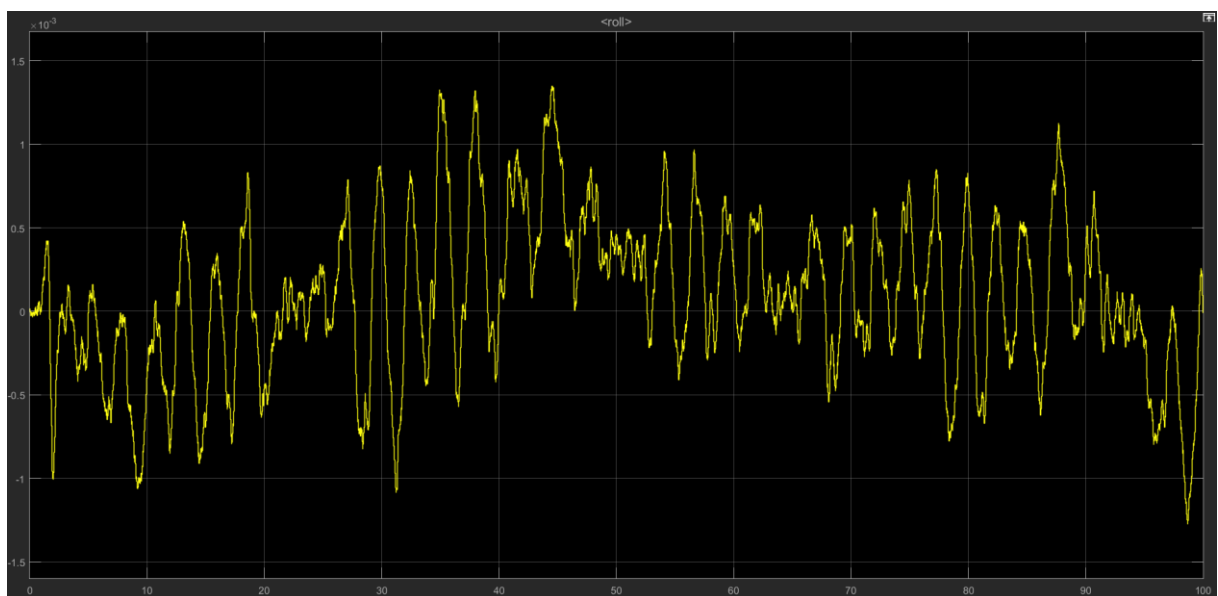


Figura 45- Uscita di roll.

.

Le oscillazioni delle uscite di pitch e roll risultano molto basse, il loro andamento assume valori molto vicini allo zero, dato che sono dell'ordine di 10^{-3} . Quindi il controllore creato non influenza tali variabili di stato.

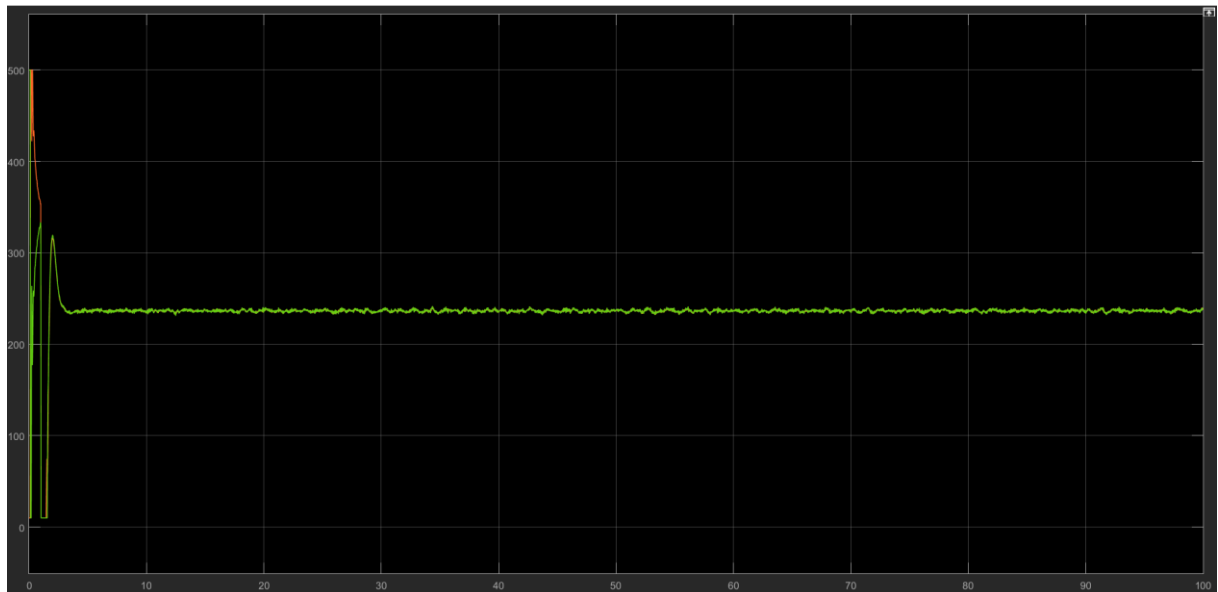


Figura 46- Saturazione dei motori.

Nell'immagine sovrastante viene rappresentata la saturazione dei motori. I motori raggiungono un valore molto alto, quello di saturazione, solo nei primi istanti quando devono avere una forza tale da riuscire ad innalzare il mini-drone, dopo di che, una volta raggiunta l'altezza desiderata, i motori si stabilizzano tutti sullo stesso valore, ottenendo una condizione di hovering.

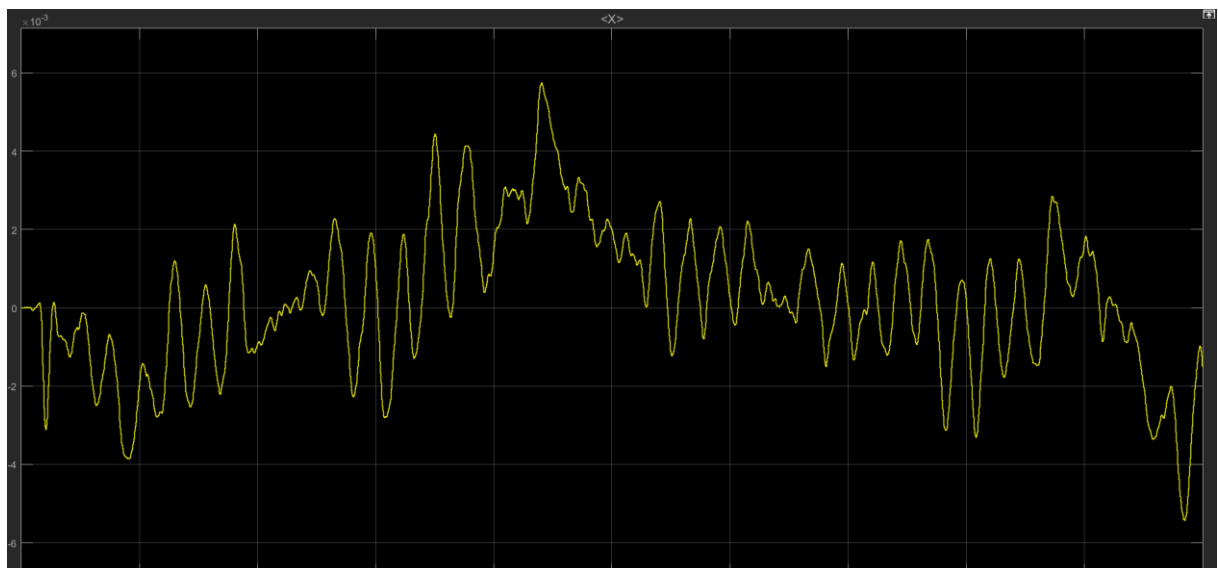


Figura 47-Uscita di x.

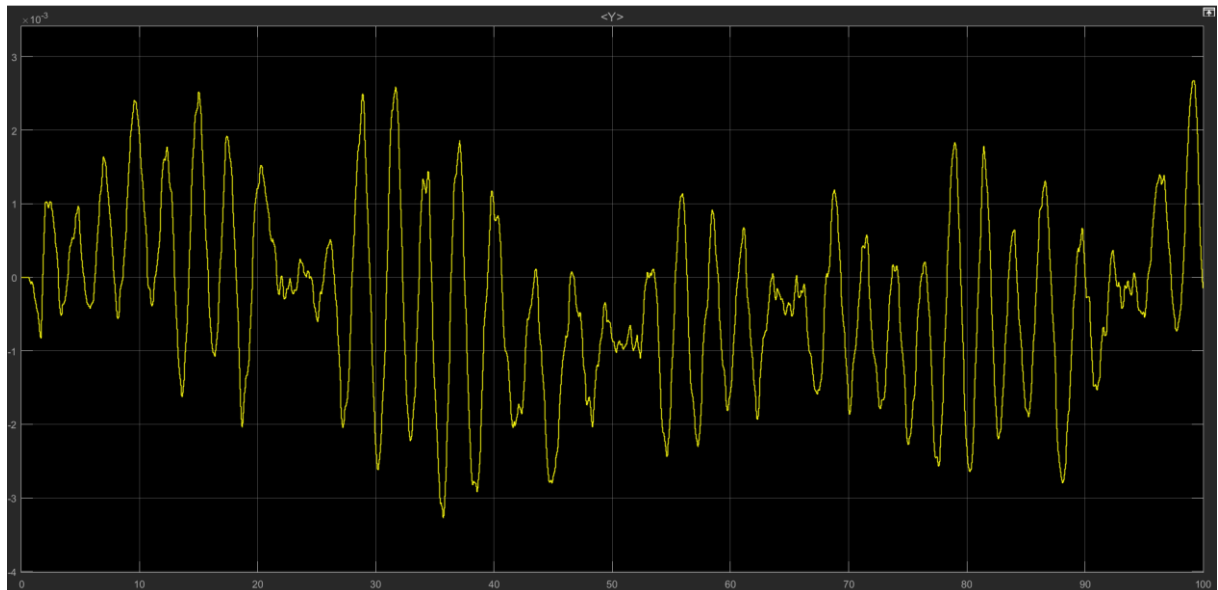


Figura 48-Uscita di y.

Il controllore creato non influenza nemmeno le uscite del controllore di posizione X e Y, perché anche qua le oscillazioni sono dell'ordine di 10^{-3} , quindi tendenti allo zero, come il loro riferimento.

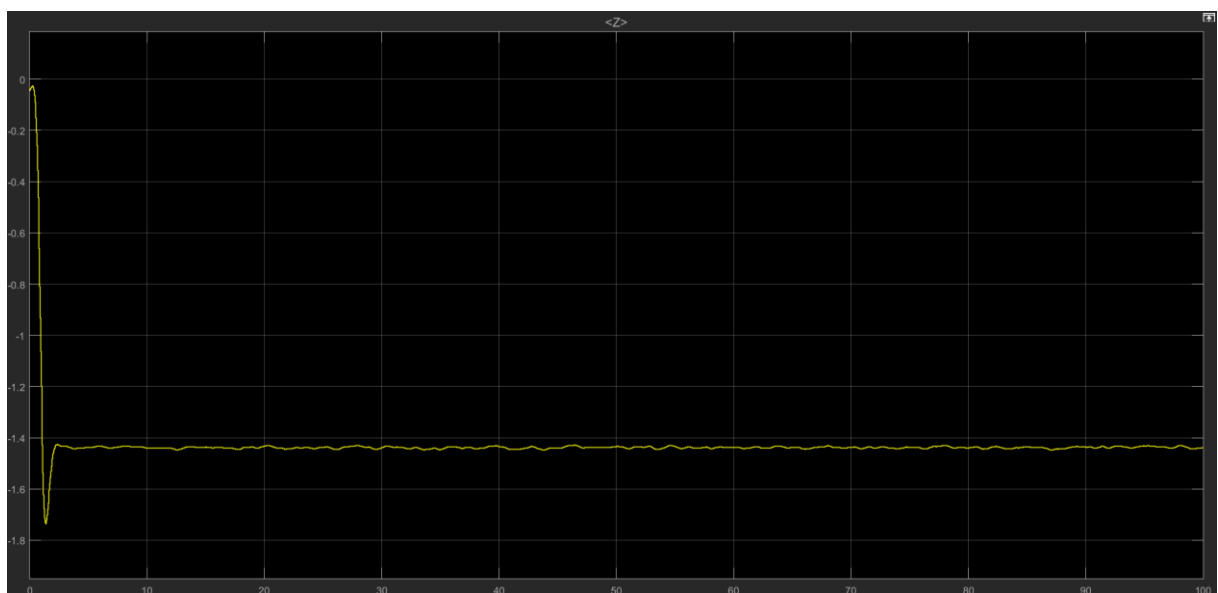


Figura 49- Uscita dell'altitudine.

Anche il controllo dell'altezza risulta stabile, non influenzato negativamente dal controllore dello yaw. Dal grafico vediamo che il drone si stabilizza a circa 1.5 m dal suolo, come voluto dal riferimento imposto.

CONTROLLORE IN FREQUENZA CASO LINEARE

Mandando il mini-drone in simulazione nel caso lineare, si ha il seguente comportamento:

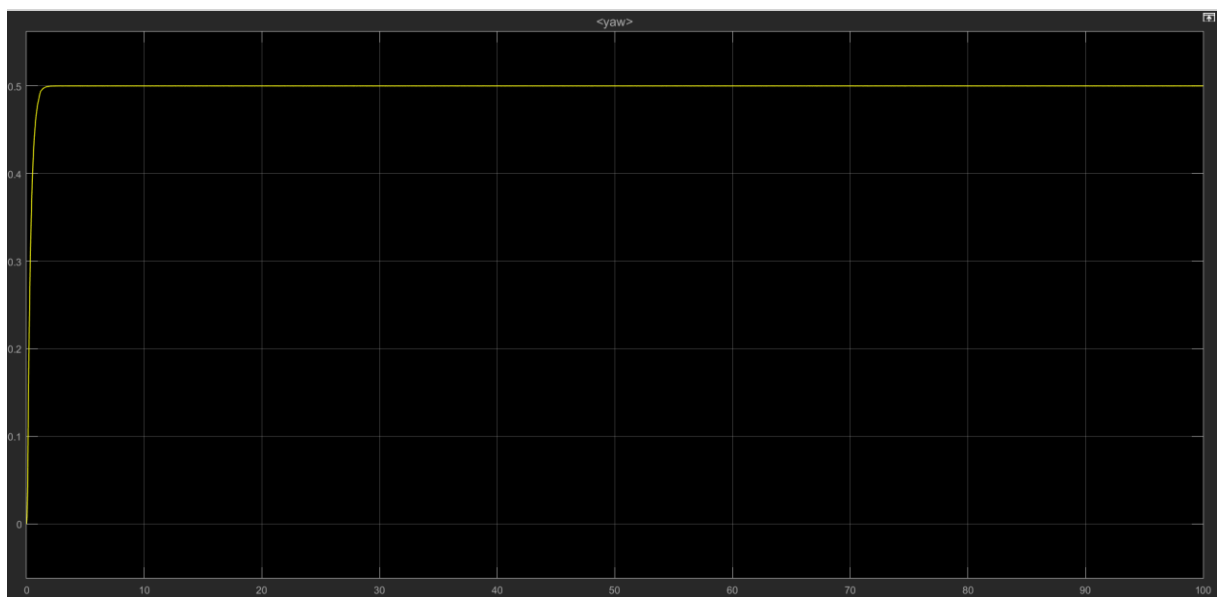


Figura 50- Uscita yaw.

L'obiettivo di controllo risulta raggiunto poiché l'uscita controllata risulta stabile nella posizione imposta. Si nota che nel caso lineare non si ha sovraelongazione, contrariamente di ciò che accade con il controllore originario.



Figura 51- Coppia relativa a yaw.

Come si nota anche in figura 51, la coppia necessaria per portare il Mambo nella posizione di yaw desiderata ha un comportamento conforme a quello voluto, ovvero una volta raggiunta tale posizione, la coppia si stabilizza a zero dato che non ci sono altre variazioni dell'angolo di yaw.

Per quanto riguarda le altre variabili di stato, assumono lo stesso comportamento del caso non lineare, quindi anche in questo caso il controllore creato fa il suo dovere non influenzando negativamente le altre variabili di stato.

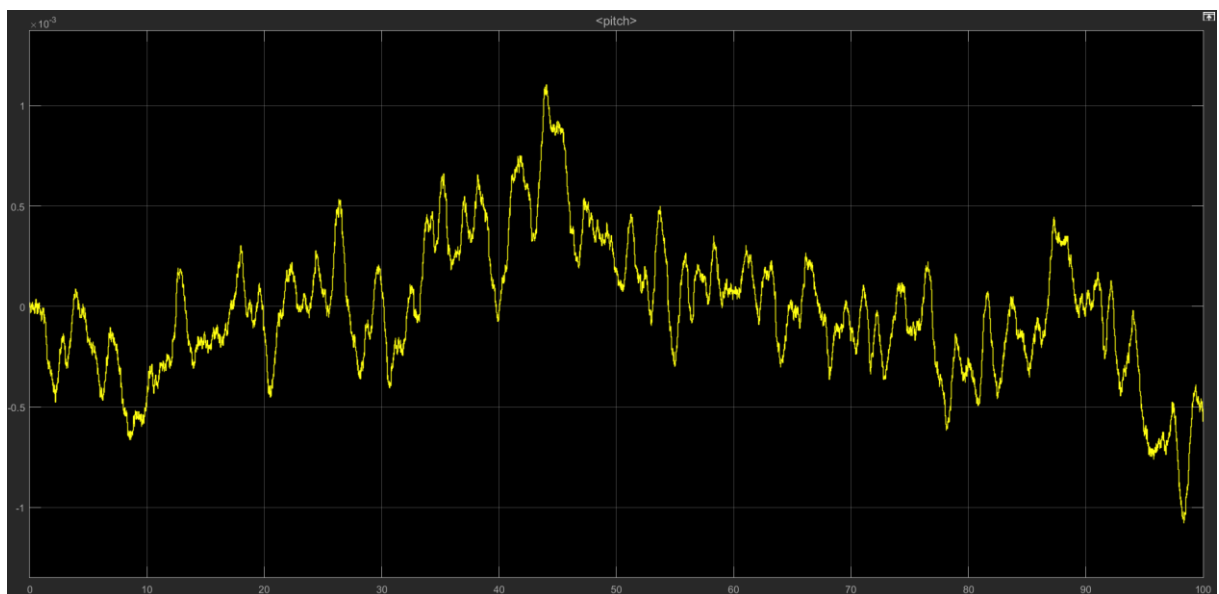


Figura 52-Uscita relativa a pitch.

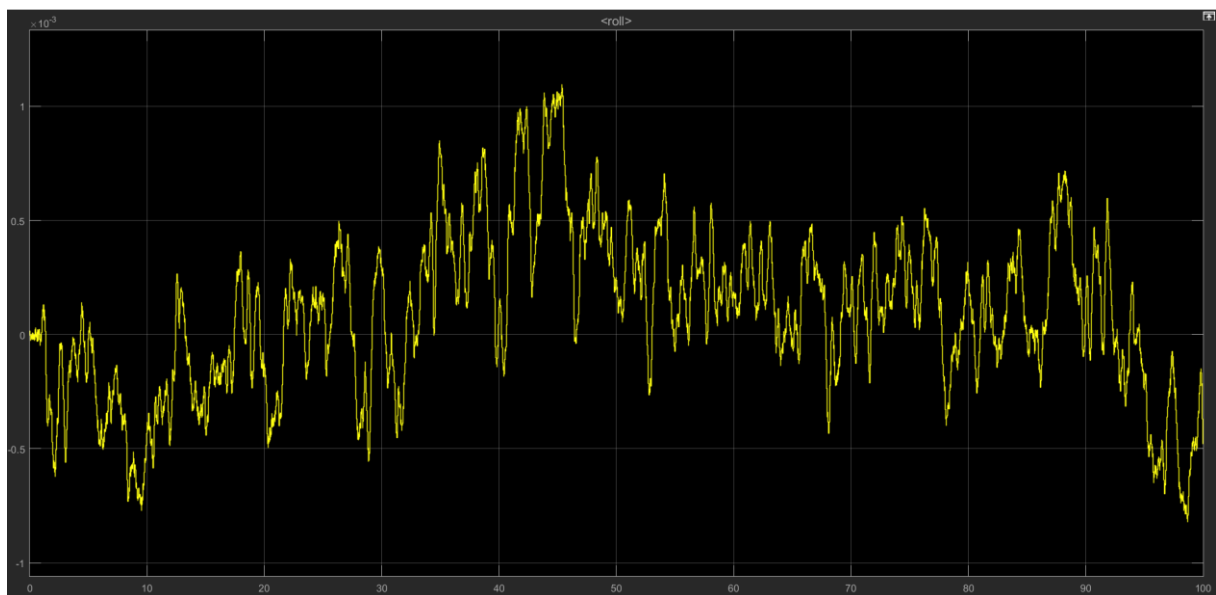


Figura 53-Uscita relativa a roll.

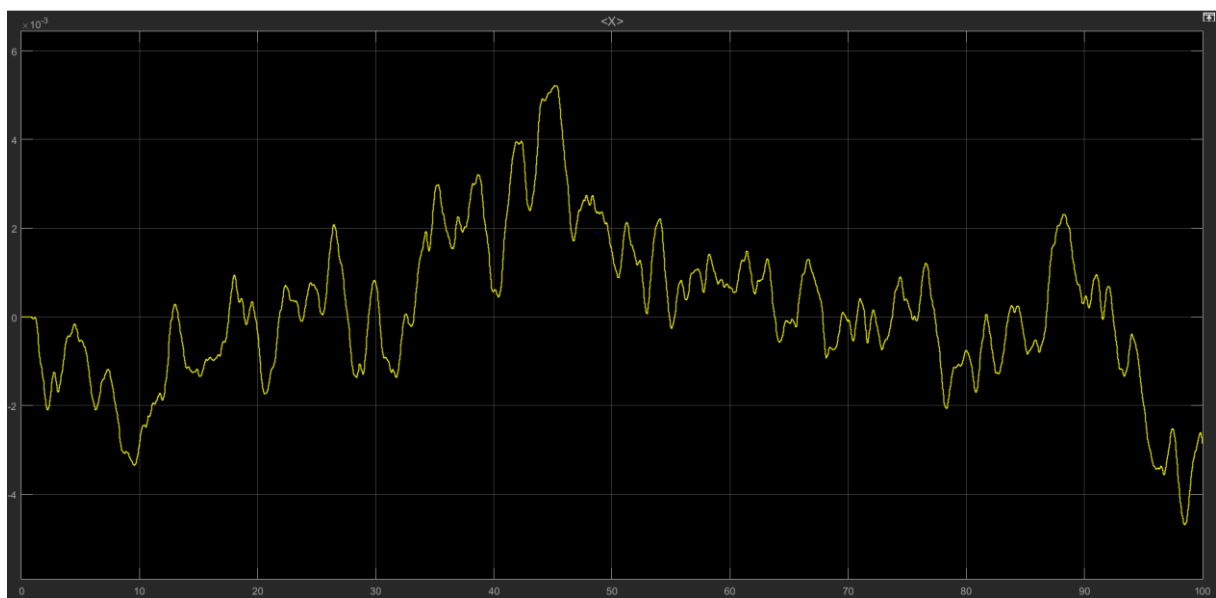


Figura 54-Uscita relativa a x.

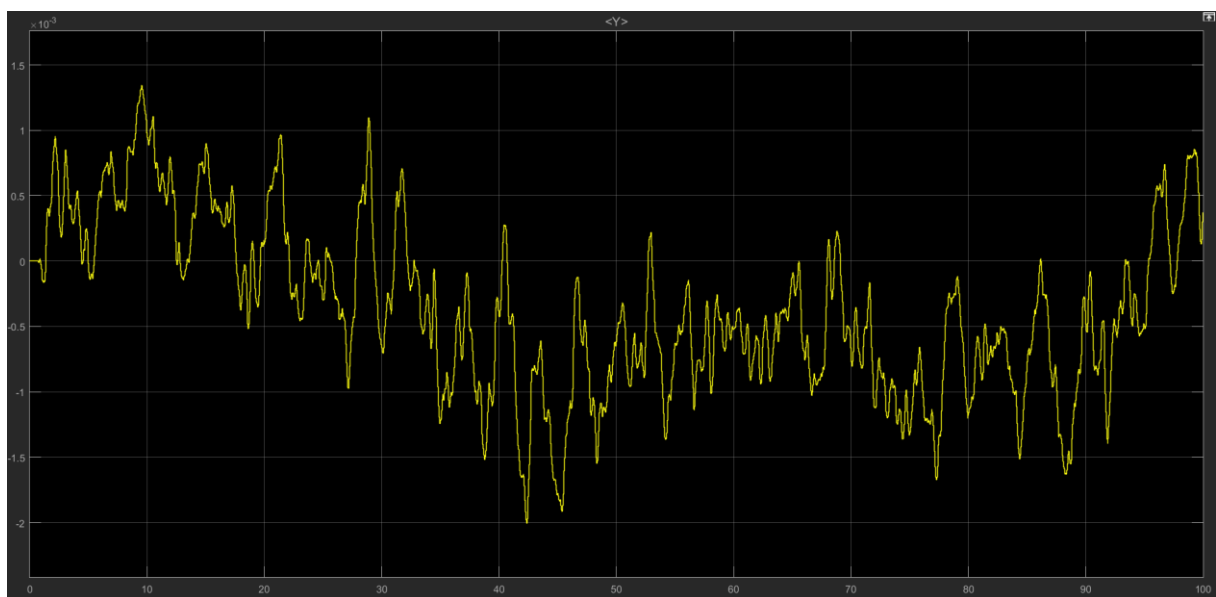


Figura 55- Uscita relativa a y.

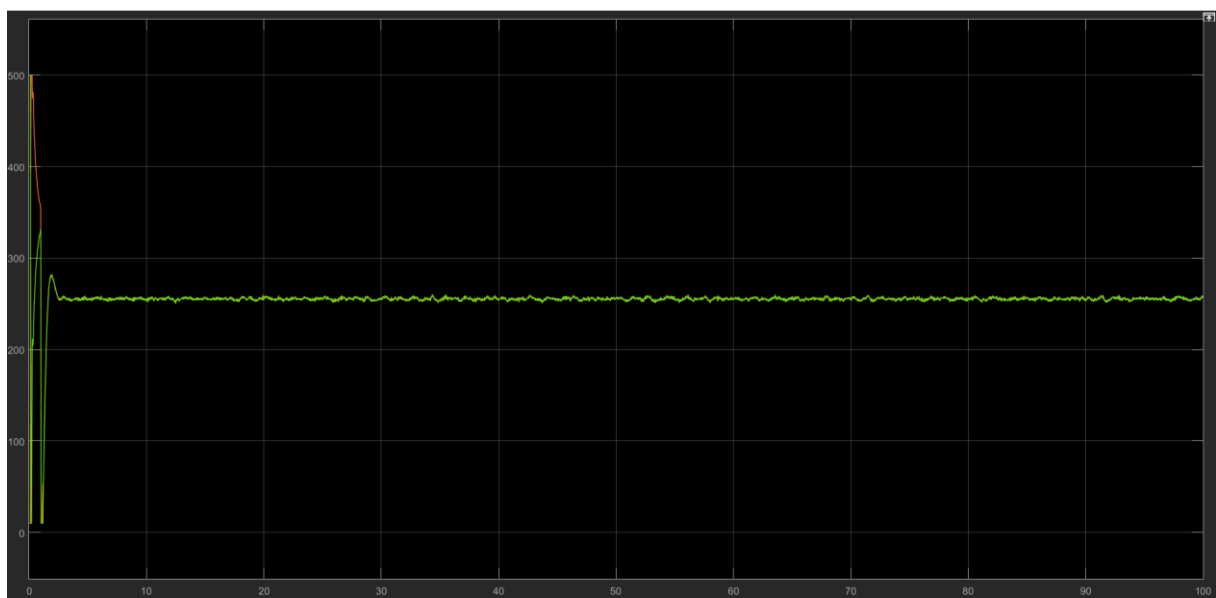


Figura 56-Saturazione dei motori.

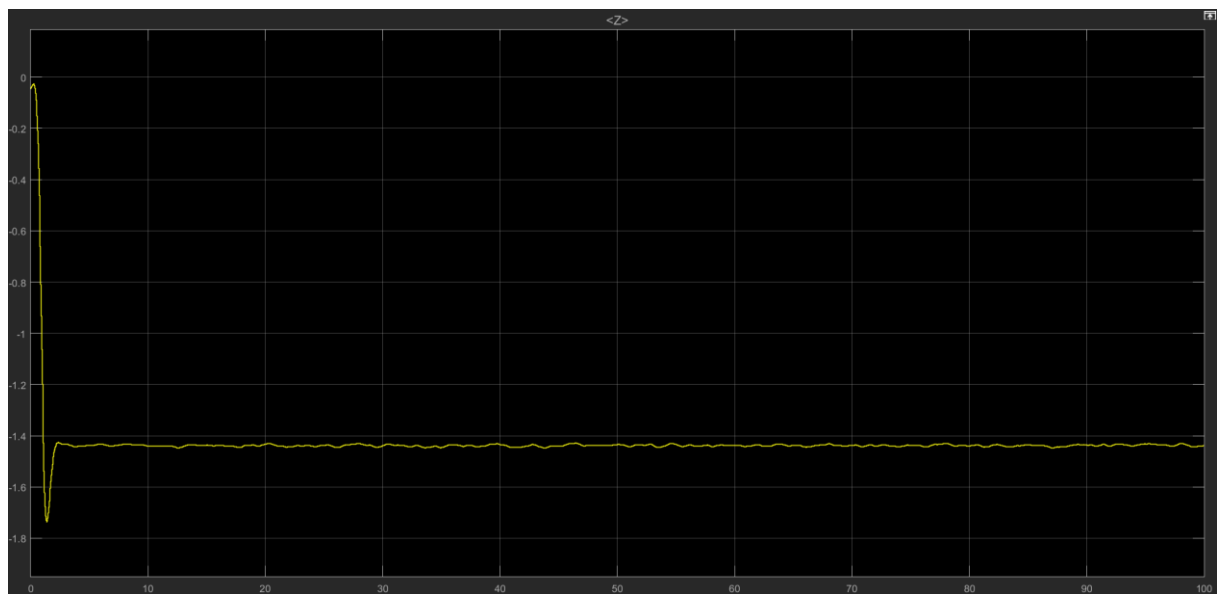


Figura 57-Uscita relativa all'altezza.

COMPORTAMENTO NEL CASO REALE

Con l'ausilio dell'installazione di un pacchetto aggiuntivo, è possibile tramite Bluetooth caricare il modello modificato sul Mambo. Quindi dopo aver collegato il mini-drone tramite Bluetooth al computer, e premuto il pulsante “ build “ sul modello che si trova nella barra superiore tra gli strumenti, viene automaticamente generato un codice C che corrisponde a quello che è stato fatto tramite i diagrammi a blocchi. Si può anche accedere a questo codice, modificandolo se necessario.

Successivamente questo codice viene inviato al Mambo tramite Bluetooth e il firmware viene riprogrammato con le modifiche apportate.

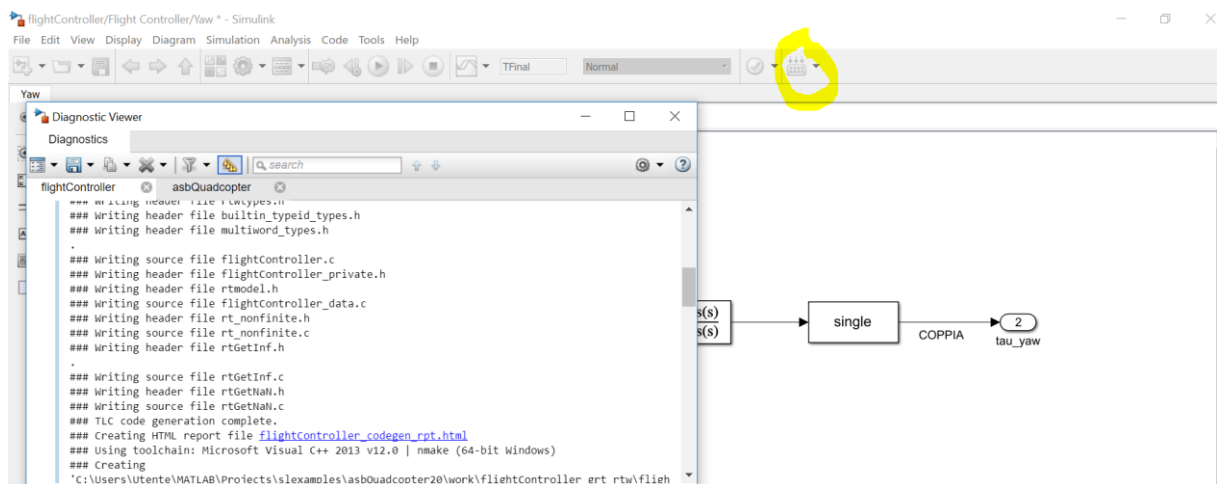


Figura 58- Generazione del codice C e tasto "build" cerchiato in giallo.

Una volta caricato il codice sul mambo, appare una schermata con la quale è possibile avviare o arrestare il mini-drone.

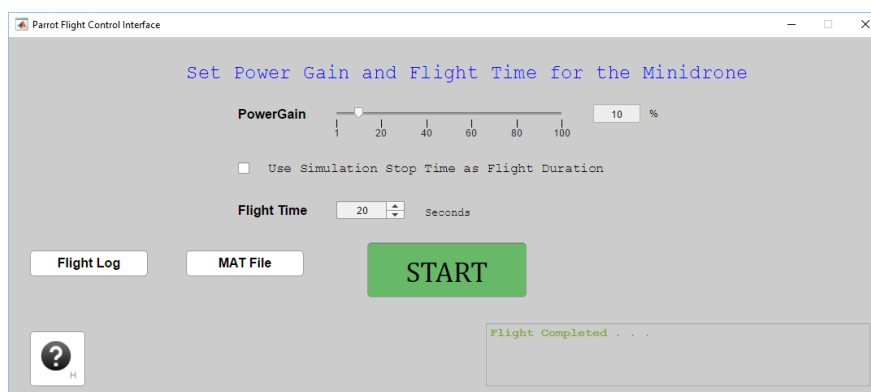


Figura 59- Interfaccia di controllo.

Si è visto che il Mambo si comporta come in fase di simulazione, se si cambia applicando un certo disturbo all'angolo di yaw, ovvero spostando il mini-drone in volo con le mani in modo sicuro, il quadricottero ritorna nella sua posizione di riferimento.



Figura 60- Pochi istanti dopo la partenza.

Come visto dai Grafici, pochi istanti dopo la partenza il drone inizia a posizionarsi nella posizione di riferimento (figura 58) pero poi stabilizzarsi in quella posizione (figura 59).



Figura 61-In condizioni di hovering.

CONCLUSIONI

Dai grafici dell'uscita che viene controllata, ovvero lo yaw, emerge che il controllore in frequenza realizzato, nel caso non lineare, ha una sovraelongazione minore rispetto al controllore originario, tuttavia i due controllori impiegano quasi lo stesso tempo per stabilizzare l'uscita, quindi è difficile dire qual è il migliore, anche perché confrontando gli errori in figura 60 e 61 si vede che l'errore diventa a zero più o meno nello stesso istante, nel caso del controllore in frequenza avviene qualche decimo di secondo prima.

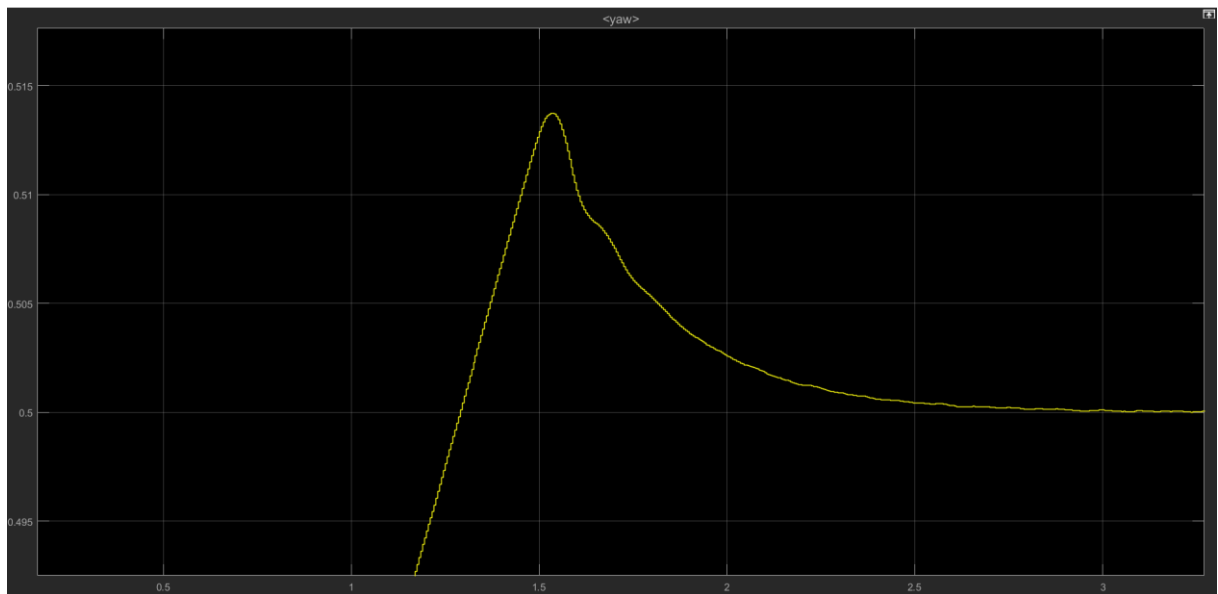


Figura 62- Sovraelongazione con il controllore in frequenza.

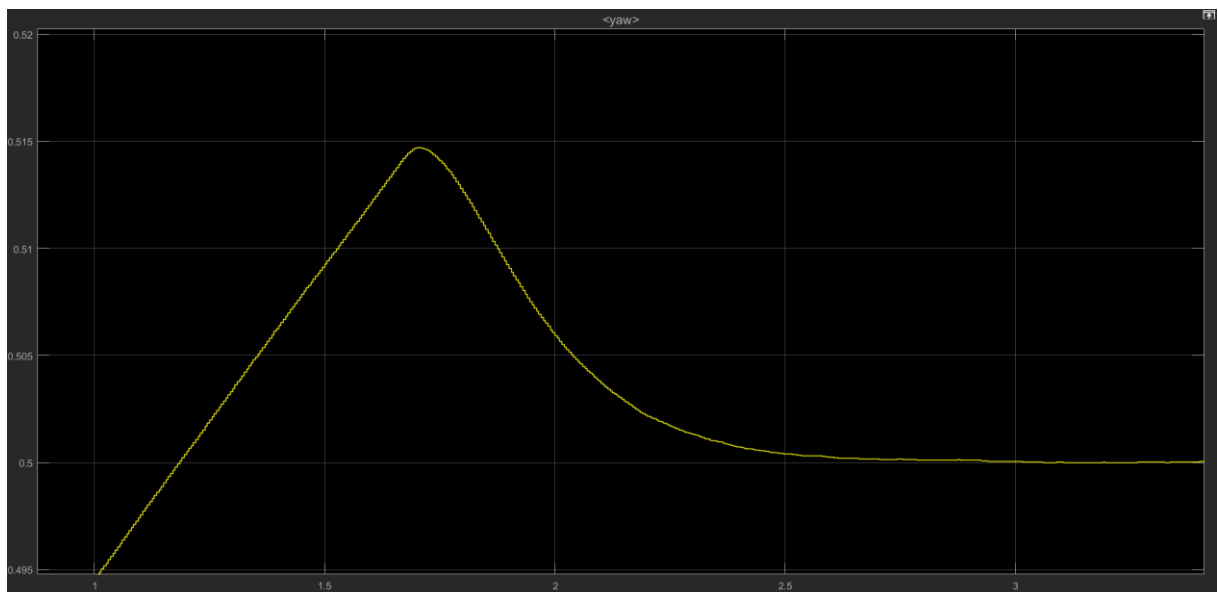


Figura 63- Sovraelongazione caso non lineare con controllore originario.

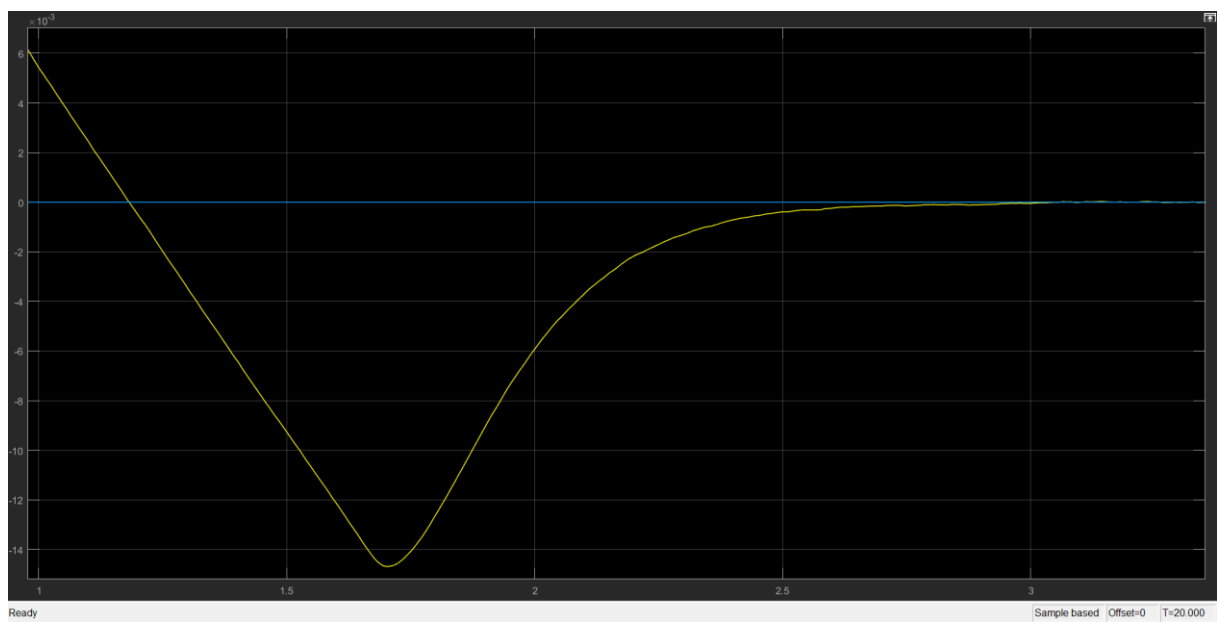


Figura 64- Andamento errore con controllore originario.

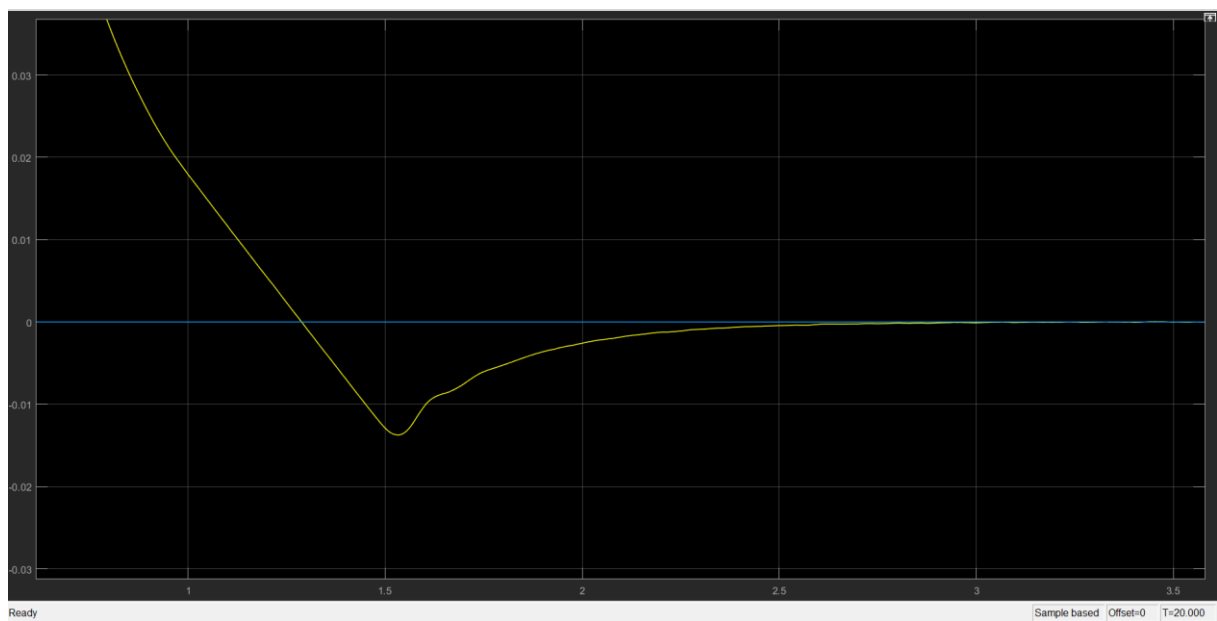


Figura 65- Andamento dell'errore con controllore in frequenza.

BIBLIOGRAFIA

- [1] A.Isidori, Sistemi di controllo
- [2] Ponds, P., Mahony, R., Corke, P. Modelling and control of a large quadrotor robot. Control Engineering Practice. 2010.
- [3] Matlab, <https://es.mathworks.com/help/aeroblks/quadcopter-project.html>
- [4] Samir Bouabdallah, Desing and control of quadrotors with application to autonomous flying
- [5] Francesco Sabatino, Quadrotor control: modeling, nonlinear control design, and simulation
- [6] Wei Zhong, Implementation of Simulink controller design on Iris+ quadrotor