



UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria Informatica e dell'Automazione

TESI DI LAUREA TRIENNALE

**Analisi, studio e progettazione di sistemi di controllo
dell'altitudine nei mini-droni Parrot "Mambo"**

Analysis, study and design of altitude control systems in Parrot "Mambo" mini drones

Candidato

Antonio Nakula Brigida

Relatore

Gianluca Ippoliti

Correlatore

Giuseppe Orlando

Anno Accademico 2019-2020

Ringraziamenti

Un grandissimo grazie ai miei genitori, mi avete sempre sostenuto e avete permesso di esaudire il mio sogno. Ringrazio i miei fratelli Fabio e Giorgia per esserci stati sempre, anche quando c'erano chilometri di distanza fra di noi. Un grazie a Martina, amica di sempre e sorella, con te sono riuscito a superare tanti momenti difficili e mi hai aiutato a crescere tanto. Ringrazio Anna per essermi stata vicina e aver reso magici questi ultimi mesi. Infine, un grazie profondo verso i prof, Ippoliti e Orlando, per la disponibilità e la professionalità dimostrata nella fine di questo percorso.

Indice

| | |
|--------------------------------------------------------------------------------|-----------|
| Introduzione | 7 |
| 1 Funzionamento di un quadricottero | 8 |
| 2 Modello fisico-matematico | 10 |
| 3 Modello SIMULINK asbQuadcopter | 18 |
| 3.1 Nonlinear Airframe | 19 |
| 3.2 Flight Control System | 24 |
| 4 Analisi dei controllori PID | 30 |
| 4.1 Andamento temporale delle coordinate di traslazione (x,y,z) | 31 |
| 4.2 Andamento temporale degli angoli di rotazione attorno agli assi | 33 |
| 5 Modello Linearizzato | 35 |
| 5.1 Linearizzazione di un sistema dinamico non lineare tramite serie di Taylor | 35 |
| 5.2 Modello linearizzato del drone e implementazione Simulink | 37 |
| 5.3 Sottosistema SISO linearizzato della dinamica dell'altitudine | 40 |
| 6 Progettazione di un controllore per l'altitudine | 42 |
| 6.1 Analisi in frequenza del sistema e possibili approcci alla progettazione | 42 |
| 6.2 Sintesi per tentativi del controllore con astatismo ai disturbi | 44 |
| 6.3 Sintesi per tentativi del controllore senza astatismo ai disturbi | 51 |
| 7 Conclusioni | 56 |
| 8 Bibliografia | 60 |

Introduzione

Per drone si intende un qualsiasi aeromobile pilotato a remoto. Il nome deriva dall'inglese "*drone*", letteralmente "*fuco*" in italiano, scelto per il caratteristico suono prodotto dalle eliche di alcune tipologie di questi dispositivi. Un drone può essere infatti di varie forme e dimensioni, alcuni presentano ali fisse (come dei comuni aeroplani) e altri presentano eliche per il decollo verticale (chiamati VTOL "*Vertical Take-Off and Landing*").

Con l'avanzamento tecnologico degli ultimi decenni e la miniaturizzazione dei componenti si è riusciti ad avere un controllo remoto sempre più preciso che ha spinto l'uso di questi dispositivi in tantissimi ambiti diversi. Difatti a seconda dell'apparecchiatura montata a bordo è possibile dare al veicolo scopi differenti, spaziando dall'ambito civile al militare.

Negli ultimi anni stanno prendendo piede nel mercato i droni VTOL con pale rotanti denominati a seconda del numero di eliche presenti per il movimento ("*tricottero*", "*quadricottero*", "*esacottero*" ecc...). In questa sede ho intenzione di parlarvi del modello matematico di un drone "*quadcopter*" e del modello simulativo specifico del drone "*Mambo*" prodotto dalla Parrot. Infine, verrà presentato lo studio e la progettazione di un controllore per la regolazione automatica dell'altitudine presentando problematiche e soluzioni.



Minidrone Parrot "Mambo"

1 Funzionamento di un quadricottero

Il quadricottero è un aeromobile che presenta quattro pale rotanti collegate ognuna ad un motore elettrico di tipo di “*brushless*”. Questo tipo di motore non presenta nessun tipo di contatto strisciante per la commutazione periodica delle correnti di avvolgimento e risulta quindi più sicuro e ha bisogno di poca manutenzione. Difatti, a differenza dei tradizionali motori *a collettore*, i motori brushless non producono scintille all’aumentare della velocità ma d’altro canto hanno bisogno di precisi circuiti di controllo, con componenti a semiconduttore, per permettere l’inversione delle correnti negli avvolgimenti.

Nei quadricotteri abbiamo quindi quattro attuatori predisposti al movimento. Purtroppo, il drone presenta 6 gradi di libertà (3 per il movimento di traslazione e 3 per le rotazioni attorno agli assi) e risulta quindi un sistema sotto-attuato. Il movimento del drone è gestito semplicemente dalle velocità di rotazione dei quattro rotori montati a bordo.

La struttura del drone è molto semplice (Fig. 1). Il telaio ha una forma “a croce” e i rotori sono posizionati sulle estremità dei bracci, equidistanti dal centro di massa. Ogni rotore ruota in senso opposto al rotore adiacente in modo tale che due rotori diametralmente opposti ruotino nello stesso verso. A seconda della spinta creata dai vari rotori è possibile quindi creare tutti i movimenti nello spazio. È importante però ricordare che, data la natura di sistema sotto-attuato del quadricottero, alcuni spostamenti fra i vari gradi di libertà tenderanno ad inficiare sugli altri gradi. Per esempio, se vogliamo che il drone trasli nella coordinata x o y non possiamo aspettarci che gli angoli di rotazione rimangano invariati. Per muoversi in questo modo, infatti, bisognerà inclinare il drone (modificare l’angolo di *beccheggio* e di *rollio*) e dare la giusta spinta verso l’alto.

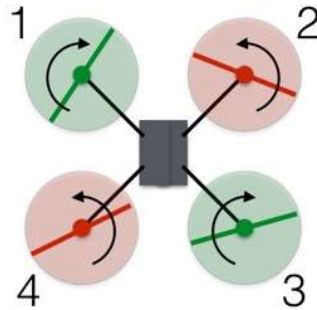


Figura 1

Per il movimento nella coordinata z (verso l'alto o il basso) basterà avere una velocità angolare uniforme nei vari rotori mantenendo invariati i versi di rotazione stabiliti in precedenza. Questo movimento è indipendente dagli altri gradi di libertà e quindi mantiene invariati gli angoli di beccheggio, imbardata e rollio. Se c'è invece una discrepanza fra le velocità angolari dei vari rotori avremo le rotazioni sui vari assi (Fig. 2).

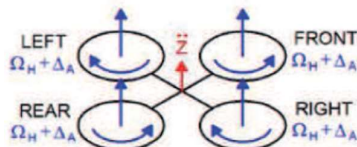


Figura 2.a (coordinata z)

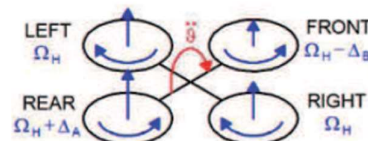


Figura 2.b (beccheggio)

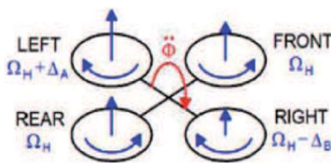


Figura 2.c (rollio)

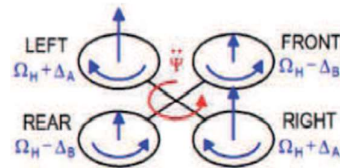


Figura 2.d (imbardata)

Se due rotori con stesso verso di rotazione ruoteranno alla stessa velocità sarà possibile variare l'angolo di imbardata. Per esempio, se ogni coppia di motori diametralmente opposti gireranno con stessa velocità e una delle due coppie con velocità maggiore rispetto all'altra si avrà un movimento nell'angolo di imbardata nel verso opposto rispetto a quello di rotazione dei motori stessi. Tenendo presente la Fig.1 e la Fig.2.d, se i motori 2 e 4 avranno velocità angolare maggiore dei motori 1 e 3, il drone ruoterà in senso orario attorno all'asse z . Invece, per gli spostamenti negli angoli di beccheggio e rollio, servirà una velocità uniforme fra motori adiacenti e con verso di rotazione

opposto. Anche in questo caso, velocità e senso di rotazione rispetto agli assi x e y dipenderà da quale coppia di motori girerà a velocità angolare maggiore.

2 Modello fisico-matematico

Ai fini di una progettazione più accurata dei sistemi di controllo è molto utile, se possibile, la costruzione di un modello fisico-matematico che approssimi il comportamento e la dinamica del sistema in esame. Questo modello sarà utile non solo per la progettazione in sé ma anche per creare ambienti di simulazione implementabili a livello software.

Sistemi di riferimento È importante definire i vari sistemi di riferimento usati in questa trattazione matematica. Definiamo quindi un sistema inerziale alla terra, chiamato *Earth Fixed Frame*, come il sistema $I = \{E_x, E_y, E_z\}$ in cui il versore E_z ha direzione pari all'accelerazione di gravità g ed è quindi diretto verso il "basso". Parliamo poi di un sistema solidale al drone (*Body Fixed Frame*) definito con $A = \{E_1, E_2, E_3\}$ e un sistema inerziale ma centrato nel centro di massa del drone chiamato *Vehicle Frame*. Inoltre, va stabilito che la coordinata x dei sistemi inerziali aumenta nella direzione frontale del drone stesso.

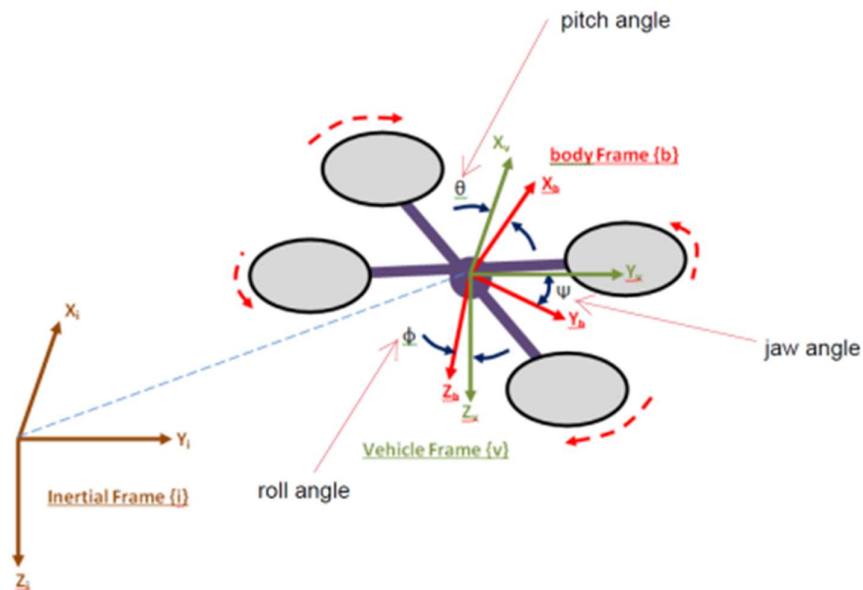


Figura 3

Trasformazione delle coordinate di traslazione Data la natura del sistema e i suoi tre gradi di libertà di rotazione attorno agli assi, c'è bisogno di definire delle matrici di rotazione. Queste matrici esprimono una applicazione lineare di trasformazione nello spazio in cui il sistema di arrivo è dato da una rotazione antioraria del sistema di partenza rispetto ad un asse dato. Ovviamente le matrici di rotazione dipendono dall'angolo di rotazione stesso. Le matrici di rotazione sono matrici ortogonali invertibili, quindi la rotazione inversa sarà data da una trasposta della matrice stessa.

Nel nostro caso abbiamo bisogno di tre matrici, una per ogni asse.

- $R(\varphi)$, rotazione attorno all'asse x con angolo φ
- $R(\theta)$, rotazione attorno all'asse y con angolo θ
- $R(\psi)$, rotazione attorno all'asse z con angolo ψ

Gli angoli φ , θ e ψ esprimono gli *angoli di Eulero* nello spazio.

Vengono riportate di seguito le varie matrici presentate

$$R(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{pmatrix} \quad (1)$$

$$R(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \quad (2)$$

$$R(\psi) = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

La matrice che esprime la rotazione in tutti e tre gli assi è data dal prodotto delle tre matrici (1), (2) e (3), nel nostro caso esprimerà la trasformazione da coordinate del *Body Fixed Frame* in quelle del *Vehicle Frame*.

$$R(\varphi, \theta, \psi) = R(\varphi) * R(\theta) * R(\psi) \quad (4)$$

Facendo gli opportuni calcoli otteniamo:

$$R(\varphi, \theta, \psi) = \mathbf{R}_b^i = \begin{pmatrix} c\varphi c\theta & c\psi s\theta s\varphi - s\psi c\varphi & c\psi s\theta c\varphi + s\psi s\varphi \\ s\varphi c\theta & s\psi s\theta s\varphi + c\psi c\varphi & s\psi s\theta c\varphi - s\varphi c\psi \\ -s\theta & c\theta s\varphi & c\theta c\varphi \end{pmatrix} \quad (5)$$

In cui per "c" si intende l'operazione di coseno e per "s" quella di seno.

Anche in questo caso la rotazione inversa è data dalla trasposta della matrice di rotazione, otteniamo quindi che:

$$\mathbf{R}_b^i = \mathbf{R}_i^b{}^T \quad (6)$$

Trasformazione delle velocità Dobbiamo definire i vettori delle velocità in entrambi i sistemi di riferimento.

Chiamiamo con $\mathbf{b} = (b_1, b_2, b_3)^T$ il vettore di traslazione rispetto al sistema solidale al drone (*Body Fixed Frame*). Una volta fatto questo andiamo a definire il vettore delle velocità relative al *Body* stesso chiamandolo \mathbf{v}_b :

$$\mathbf{v}_b = \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

Il vettore delle velocità lineari rispetto al sistema inerziale (\mathbf{v}_i) invece sarà dato da:

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

Per quanto riguarda le velocità angolari, invece, bisogna fare altre considerazioni. Innanzitutto, definiamo come $\boldsymbol{\omega}_b = (p, q, r)^T$ il vettore delle velocità angolari rispetto al sistema solidale al drone.

Dato il vettore che contiene la variazione degli angoli euleriani, possiamo calcolare il vettore $\boldsymbol{\omega}_b$. In particolare, riutilizzando (1), (2) e (3), si ha che il generico vettore velocità angolare visto dal sistema solidale con il drone (*Body Fixed Frame*) si ottiene dalla seguente relazione:

$$\boldsymbol{\omega}_b = R(\varphi) * R(\theta) * R(\psi) * \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} + R(\varphi) * R(\theta) * \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R(\varphi) * \begin{pmatrix} \dot{\varphi} \\ 0 \\ 0 \end{pmatrix} \quad (8)$$

Da cui si ottiene:

$$\boldsymbol{\omega}_b = \begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 1 & 0 & -s\theta \\ 0 & c\varphi & c\theta s\varphi \\ 0 & -s\varphi & c\theta c\varphi \end{pmatrix} * \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad (9)$$

Calcolando l'inversa, invece, riusciamo ad ottenere una formula per le velocità angolari viste dal sistema di riferimento inerziale (*Vehicle Frame*) partendo da quelle del *Body Fixed Frame*:

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & s\varphi \frac{s\theta}{c\theta} & c\varphi \frac{s\theta}{c\theta} \\ 0 & c\varphi & -s\varphi \\ 0 & \frac{s\varphi}{c\theta} & 1 \end{pmatrix} * \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (10)$$

Dinamica del sistema Prima di iniziare la trattazione e di presentare le equazioni del sistema facciamo delle assunzioni importanti:

1. Il quadricottero ha una struttura perfettamente simmetrica e il tensore di inerzia è diagonale.
2. La struttura del quadricottero è rigida.
3. La struttura è tempo-invariante.
4. Ogni motore del drone è identico agli altri.

Consideriamo ora la dinamica delle coordinate di traslazione. In questa sede verranno citate e trattate le stesse equazioni e leggi usate per il modello simulativo implementato su Simulink. Esponiamo quindi di seguito la Legge di Newton per i corpi rigidi presentate nell'articolo [2]:

$$m\dot{\mathbf{v}}_b = -m\boldsymbol{\omega}_b \times \mathbf{v}_b + mg\mathbf{R}_i^b e_3 + \sum_{N,S,W,E} \mathbf{T}_i \quad (11)$$

Si nota come la forza agente sul drone sia composta da più parti:

- L'accelerazione centripeta $\mathbf{F}_c = m\boldsymbol{\omega}_b \times \mathbf{v}_b$
- L'accelerazione di gravità $\mathbf{F}_g = mg\mathbf{R}_i^b e_3$
- La spinta di ogni i-esimo motore \mathbf{T}_i (Nord, Sud, Ovest, Est)

Facciamo lo stesso per quanto riguarda la dinamica rotazionale del drone. Calcoliamo il momento torcente totale $I\dot{\boldsymbol{\omega}}_b$:

$$I\dot{\boldsymbol{\omega}}_b = -\boldsymbol{\omega}_b \times I\boldsymbol{\omega}_b + \sum_{N,S,W,E} (\mathbf{Q}_i + \mathbf{M}_i) \quad (12)$$

Anche in questo caso abbiamo una somma di più contributi:

- Il prodotto vettoriale tra la velocità angolare $\boldsymbol{\omega}_b$ e il momento angolare $I\boldsymbol{\omega}_b$
- La somma della coppia \mathbf{Q}_i e il momento torcente \mathbf{M}_i di ogni i-esimo motore

Parliamo ora del calcolo della forza di spinta \mathbf{T}_i introducendo, per lo scopo, il fenomeno del *Blade Flapping*. Durante il volo degli aeromobili VTOL, il movimento delle eliche rotanti, fa sì che una parte dell'elica senta un flusso di aria minore rispetto a quello utile per avere una portanza adeguata.

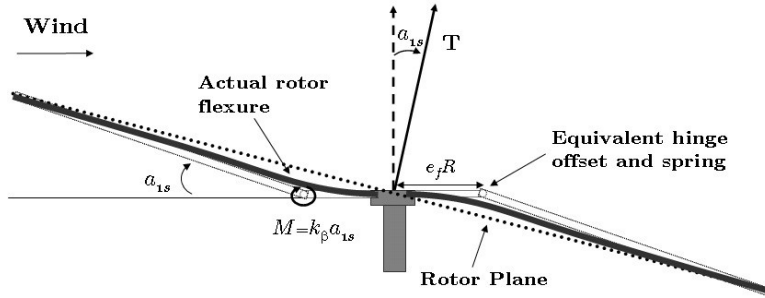


Figura 4

Questo fenomeno (avvertito soprattutto durante movimenti di traslazione) va ad influire sulla corretta salita dell'aeromobile in questione. Il mezzo, infatti, tenderà ad inclinarsi nella direzione in cui è presente la parte di elica che si muove in verso opposto al verso di movimento di traslazione del mezzo stesso.

Si parla di *asimmetria di portanza* (dall'inglese *Dissymmetry of lift*) e può essere contrastato dando la possibilità all'elica di piegarsi di un certo angolo.

Il "flappeggio" dell'elica dà un contributo alla spinta del rotore secondo la relazione (13):

$$\mathbf{T}_i = C_T \rho A r^2 \omega_i^2 \begin{pmatrix} -\sin a_{1i} \\ \cos a_{1i} \sin b_{1i} \\ \cos b_{1i} \sin a_{1i} \end{pmatrix} \quad (13)$$

In cui sono presenti varie caratteristiche fisiche dell'elica e compaiono:

- Il coefficiente di spinta C_T
- La densità dell'aria ρ
- L'area dell'elica A e il quadrato del raggio r
- Il quadrato della velocità angolare del rotore i -esimo ω_i
- Un vettore che dipende dagli angoli a_{1i} e b_{1i} relativi al flappeggio

Calcoliamo ora la coppia e il momento torcente dell' i -esimo motore

$$\mathbf{Q}_i = C_Q \rho A r^3 \omega_i^2 \mathbf{e}_3 \quad (14)$$

$$\mathbf{M}_i = \mathbf{T}_i \times \mathbf{d}_i \quad (15)$$

In cui notiamo che la coppia può avere solo componenti verticali (asse z) e che il momento torcente non è altro che il momento della spinta \mathbf{t}_i a distanza \mathbf{d}_i dal centro di massa.

I vettori di posizione dei motori (N, S, W, E) rispetto al sistema solidale al drone sono:

$$\mathbf{d}_N = (0, d, h)^T \quad (16)$$

$$\mathbf{d}_S = (0, -d, h)^T \quad (17)$$

$$\mathbf{d}_W = (-d, 0, h)^T \quad (18)$$

$$\mathbf{d}_E = (d, 0, h)^T \quad (19)$$

Le equazioni cardinali della dinamica possono infine essere racchiuse nel seguente sistema di equazioni, presentato nel documento [1]:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \mathbf{R}_B^E \left\{ \begin{pmatrix} 0 \\ 0 \\ \sum \mathbf{T}_i \end{pmatrix} - \begin{pmatrix} K_1 \dot{x} \\ K_2 \dot{y} \\ K_3 \dot{z} \end{pmatrix} \right\} \frac{1}{m} - \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \\ (\mathbf{T}_1 - \mathbf{T}_2 - \mathbf{T}_3 + \mathbf{T}_4 + K_4 \dot{\phi}) \frac{l}{I_x} \\ (-\mathbf{T}_1 - \mathbf{T}_2 + \mathbf{T}_3 + \mathbf{T}_4 + K_5 \dot{\theta}) \frac{l}{I_y} \\ (C\mathbf{T}_1 - C\mathbf{T}_2 + C\mathbf{T}_3 - C\mathbf{T}_4 + K_6 \dot{\psi}) \frac{l}{I_z} \end{pmatrix} \quad (20)$$

In cui le costanti K_i con $i = 1, \dots, 4$ rappresentano i coefficienti di resistenza aerodinamica, l la lunghezza delle braccia della struttura (distanza del CdM dal generico motore), C è una costante che lega la forza di spinta al momento e I_x, I_y, I_z sono gli elementi della diagonale del tensore di inerzia.

Blade Flapping Effect Per studiare più a fondo il fenomeno del *flappeggio* delle eliche è importante citare alcune equazioni descritte nel documento [2]:

$$\mathbf{v}_{ri} = \mathbf{v} + \boldsymbol{\omega} \times \mathbf{d}_i \quad (21)$$

$$\mu_{ri} = \frac{\|\mathbf{v}_{r(1,2)i}\|}{\omega_i r} \quad (22)$$

$$\psi_{ri} = \arctan\left(\frac{\mathbf{v}_{r(2)i}}{\mathbf{v}_{r(1)i}}\right) \quad (23)$$

In cui l'equazione (21) indica la velocità del singolo motore, la (22) calcola l'*advance ratio* del motore e la (23) l'angolo compreso fra la direzione di movimento del drone e il nord magnetico terrestre.

Queste equazioni ci sono servono infine per calcolare gli angoli di flappeggio veri e propri, tramite le due ultime equazioni:

$$u_{1si} = \frac{1}{1 - \frac{\mu_{ri}^2}{2}} \mu_{ri} (4\theta_t - 2\lambda_i) \quad (24)$$

$$v_{1si} = \frac{1}{1 + \frac{\mu_{ri}^2}{2}} \frac{4}{3} \left(\frac{C_T}{\sigma} \frac{2}{3} \frac{\mu_{ri} \gamma}{a} + \mu_{ri} \right) \quad (25)$$

In queste relazioni troviamo:

- L'afflusso d'aria all'i-esimo motore λ_i
- Il numero di Lock γ (vedi riferimento [3])

Considerazioni sul coefficiente di spinta Durante la trattazione abbiamo tralasciato un punto importante: il coefficiente di spinta dei motori non può essere costante. Infatti, durante le manovre che coinvolgono beccheggio e rollio, il motore percepisce una diversa velocità verticale che va ad influire sull'angolo di afflusso dell'aria, cambiando il suo valore. In formule abbiamo:

$$\frac{C_T}{\sigma} = \frac{a(\alpha)}{4} \left(\theta_{tip} - \frac{v_i + V_c}{\omega r} \right) \quad (26)$$

In questa equazione è quindi espressa una relazione fra il coefficiente di spinta C_T e la velocità verticale del drone V_c . Compaiono anche:

- Coefficiente di portanza $a(\alpha)$ funzione dell'angolo di attacco α
- L'angolo geometrico della punta dell'elica di rotore θ_{tip}
- La velocità indotta sul rotore v_i
- La solidità del disco σ , pari al rapporto fra l'area dell'elica e l'area del disco di rotore

Queste informazioni possono essere approfondite in [4]

Resistenza Aerodinamica Durante il volo, il drone, sente una resistenza aerodinamica dipendente dalla velocità di volo del veicolo. Questa forza è esprimibile come:

$$D = \frac{1}{2}\rho V^2 S C_D \quad (27)$$

Nell'equazione (27) compaiono:

- La densità dell'aria ρ .
- La velocità di traslazione V del drone.
- La superficie di impatto con l'aria S .
- Il coefficiente di drag C_D .

3 Modello SIMULINK asbQuadcopter

In questa sede verrà presentato e analizzato il modello Simulink® del mini-drone “Mambo” prodotto dall’azienda francese Parrot. Questo modello è stato creato in collaborazione con MathWorks ed è possibile accedervi installando il pacchetto “*Simulink Support Package for Parrot Minidrones*”. Una volta installato il pacchetto, basterà utilizzare il comando *asbQuadcopterStart* su MATLAB per aprire il modello e visualizzarlo in ogni sua parte.

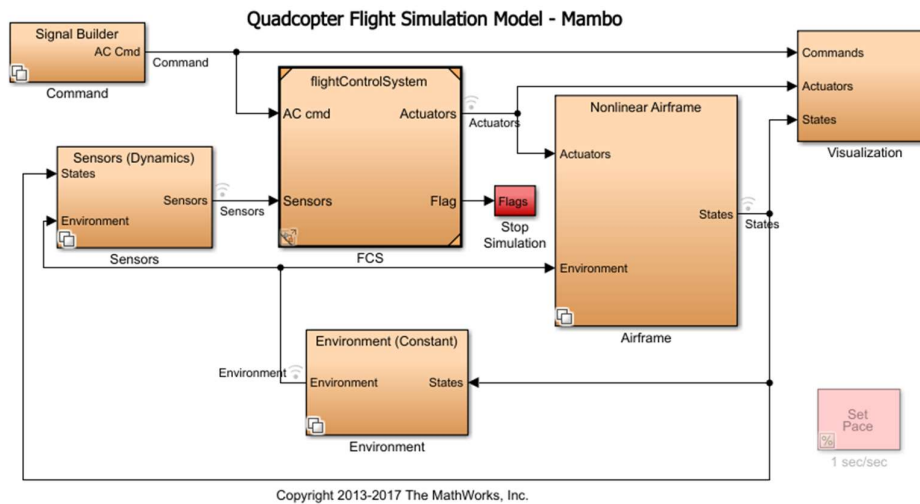


Figura 5

Il sistema è diviso in più blocchi, ognuno con le proprie caratteristiche:

- *Nonlinear AirFrame* in cui viene simulato il comportamento fisico del drone e quindi vi sono descritte e calcolate le equazioni della dinamica interna del processo.
- *Environment* è il blocco adibito al calcolo di tutte le dipendenze con le variabili ambientali.
- *Sensors* in cui viene simulata l’azione di filtraggio dei segnali di uscita del processo e la dinamica dei sensori montati sul drone.
- *flightControlSystem* è il blocco che contiene i vari regolatori e i blocchi di calcolo della stima dello stato.
- *Signal Builder* contiene i segnali di riferimento per la stabilità e il controllo. Questi segnali verranno poi utilizzati dal controllore per permettere la retroazione.
- *Visualization* è il blocco che contiene la simulazione tramite modello 3D del drone e l’estrazione di vari dati del volo.

Nei prossimi paragrafi non verranno visti nel dettaglio tutti i blocchi presentati nella lista ma solo l'implementazione del processo e gli algoritmi di controllo.

3.1 Nonlinear Airframe

Presenteremo ora il modello simulativo di tutta la dinamica interna del processo. Verrà tenuto in considerazione il modello matematico-fisico presentato nel capitolo 2 e quindi il modello non lineare del processo.

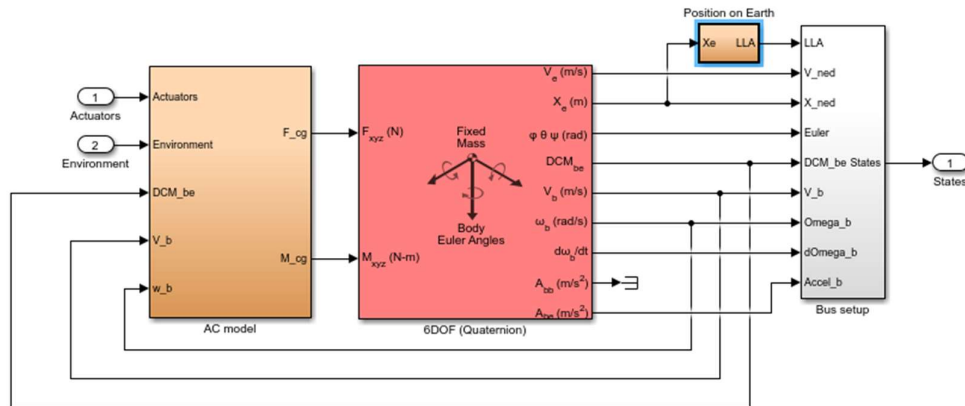


Figura 6

Il blocco, come si può vedere dalla Fig.6, riceve in ingresso i segnali di comando degli attuatori e dell'ambiente, creando in uscita, tramite un bus, un vettore di segnali dello stato.

All'interno troviamo il sottosistema *AC model*. Quest'ultimo calcola le forze e i momenti meccanici prendendo in entrata vari segnali:

- Il comando di spinta ai motori *Actuators*.
- L'azione delle variabili ambientali *Environment*.
- La matrice DCM_{be} (*Direct Cosine Matrix*) utile per il passaggio dalle coordinate del sistema di riferimento solidale al drone a quello inerziale (vedi Capitolo 2).
- La velocità di traslazione v_b relativa al *Body Fixed Frame*.
- La velocità angolare ω_b , anch'essa riferita al sistema solidale al corpo del velivolo.

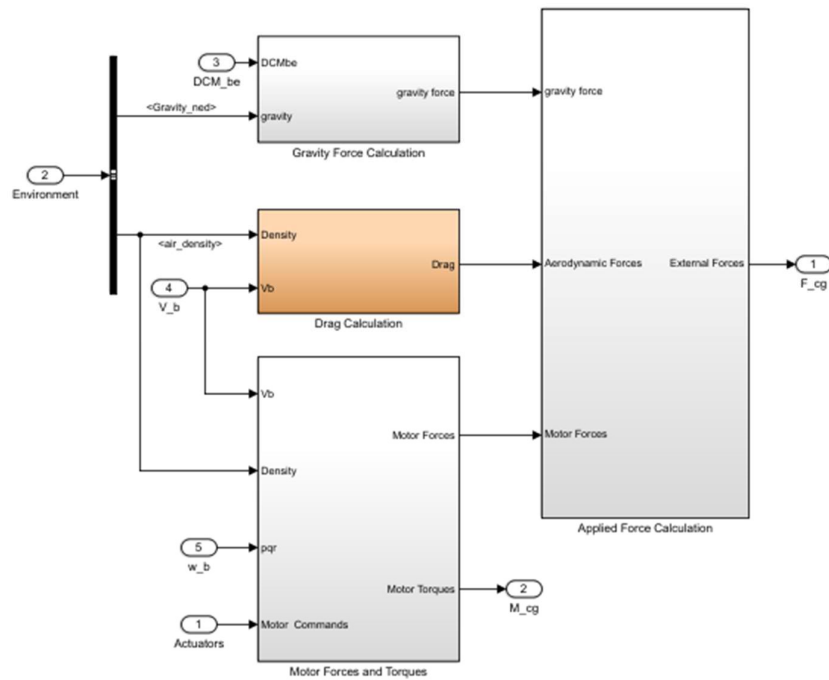


Figura 7

Nella Fig.7 possiamo osservare l'interno del blocco *AC model*, notiamo che è diviso in più parti, ognuno per il calcolo di grandezze differenti. In questa sezione verranno riprese le equazioni presentate nel Capitolo 2 riguardanti la dinamica del sistema.

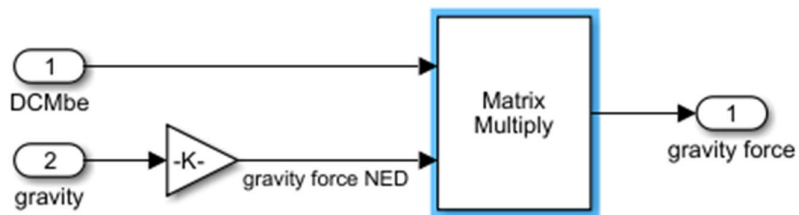


Figura 8

Gravity Force Calculation È il blocco predisposto al calcolo della forza di gravità tramite la matrice di trasformazione DCM_{be} , il suo interno è mostrato il Fig.8

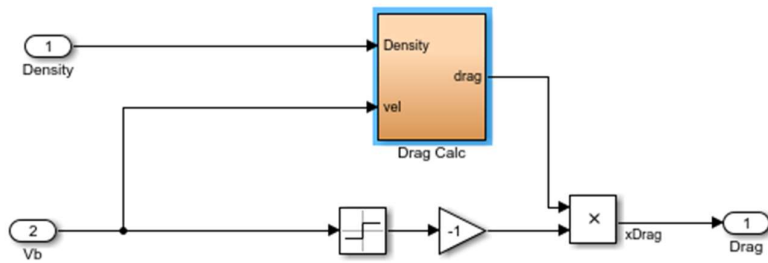


Figura 9

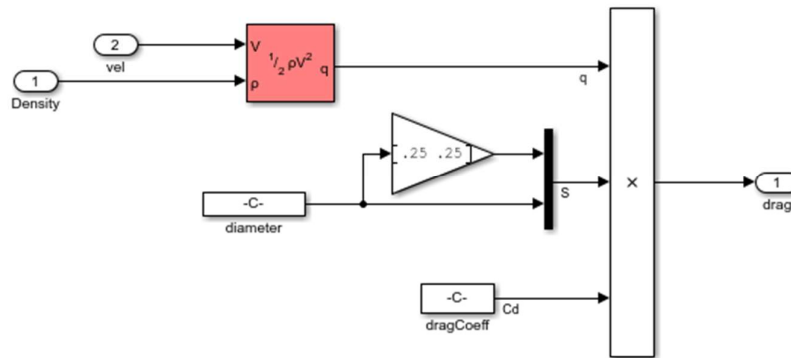


Figura 10

Drag Calculation Qui dentro vengono fatti tutti i calcoli riguardanti la forza di resistenza aerodinamica che il drone percepisce durante il suo volo. Fondamentalmente si fa riferimento all'equazione (27) presente nel Capitolo 2, calcolata in Fig.10.

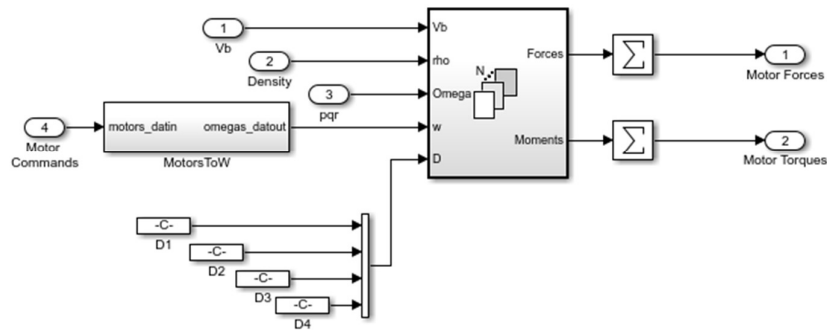


Figura 11

Motor Forces to Torques In Fig.11 vediamo nel dettaglio questo blocco. Riceve in entrata il vettore di comando ai motori, la densità dell'aria, la velocità di traslazione e quella angolare. Il comando ai motori viene convertito, tramite il blocco *MotorsToW* (Fig.12), in un segnale interpretabile dai motori stessi. Il segnale infatti attraversa varie amplificazioni e saturazioni. Quest'ultimo va a finire nel calcolo di tutte le equazioni incontrate nel modello fisico-matematico (Fig.13.a e Fig.13.b). In particolare, vengono utilizzate (11), (12), (13), (14) e (15).

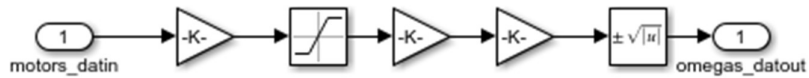


Figura 12

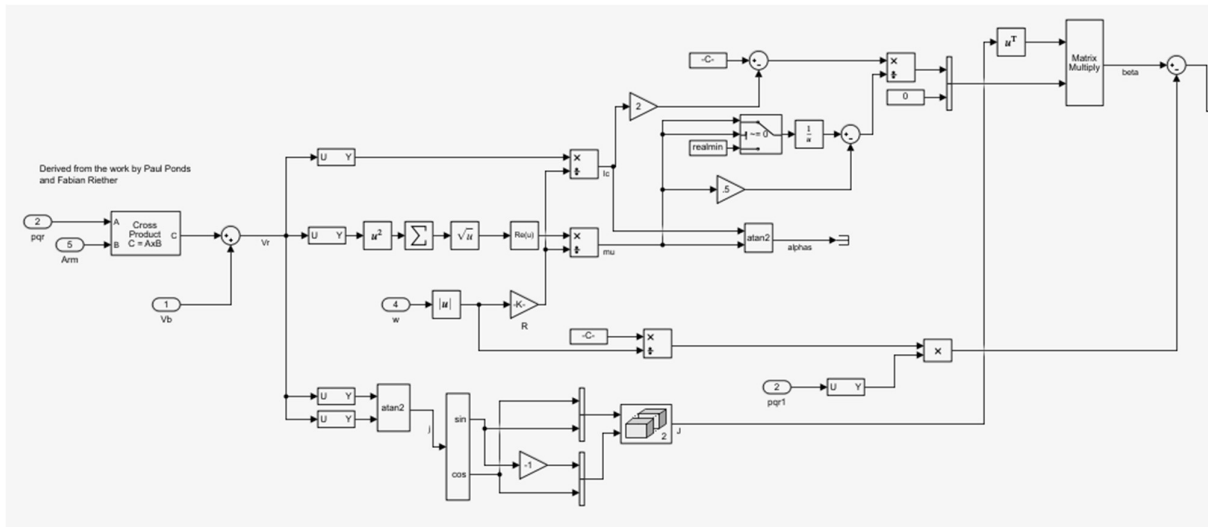


Figura 13.a

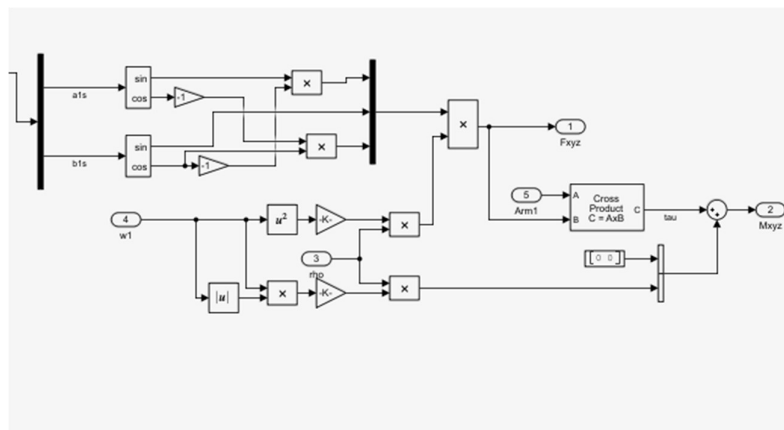


Figura 13.b

Notiamo anche che in questo blocco viene calcolata l'azione del flappeggio sui motori. Si possono notare infatti le equazioni (21), (22), (24) e (25). In uscita avremo invece la forza dei motori e i momenti meccanici.

Applied Force Calculation Infine mostriamo l'ultimo blocco dell'*AC model*, l'*Applied Force Calculation*. Quest'ultimo, come mostrato in Fig.14, va a formare il vettore delle forze esterne.

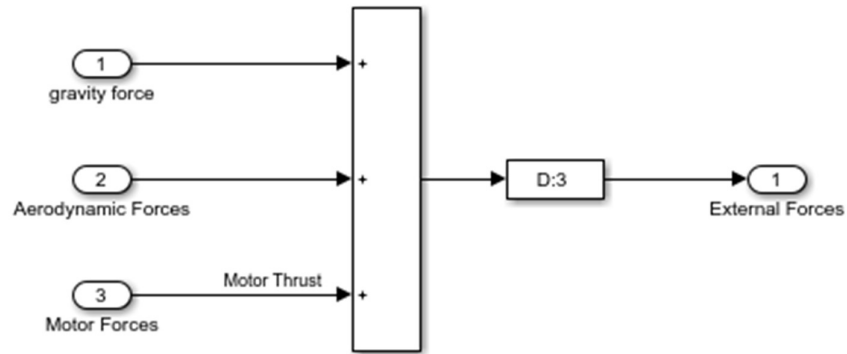


Figura 14

3.2 Flight Control System

Parliamo ora del blocco contenente tutte le operazioni concernenti il controllo dei gradi di libertà del sistema: il *Flight Control System*. Il blocco Simulink si presenta come in Fig.15.

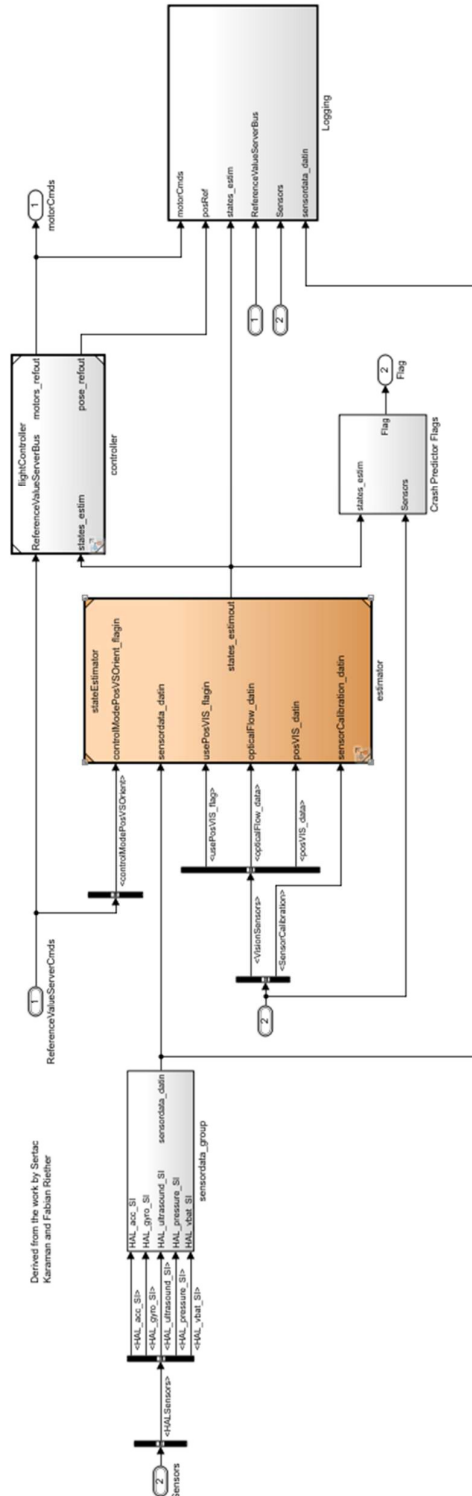


Figura 15

Il blocco riceve in ingresso i valori di riferimento creati nel blocco *Signal Builder* e i dati raccolti dai sensori. È qui infatti (nel Flight Control System) che avviene l'azione di retroazione e quindi di calcolo del segnale di errore fra uscite e riferimento. In uscita avremo invece i segnali di comando dei motori e un segnale (*flag*) utile per terminare la simulazione durante una condizione di *crash* (schianto a terra) del drone stesso.

All'interno troviamo vari blocchi:

- *Sensor Data Group* che preleva i dati dei sensori e li organizza all'interno di strutture dati.
- *Estimator* che stima i valori dello stato del sistema tramite i dati provenienti dai sensori.
- *Crash predictor flags* è il blocco che, tramite dei controlli sulle grandezze, mette in sicurezza il drone. Questo blocco genera il segnale di flag, usato per chiudere la simulazione per evitare situazioni pericolose per il drone.
- *Logging* è un blocco adibito alla visualizzazione dell'andamento temporale delle varie grandezze. Qui vengono creati dei *plot* contenenti i grafici utili all'analisi post simulazione di volo.
- *flightController* contiene tutti i blocchi di controllo veri e propri. All'interno, quindi, sono presenti i regolatori PID dei vari sottoprocessi.

Il blocco più importante è il Flight Controller, il suo modello Simulink si presenta come in Fig.16 e Fig.17

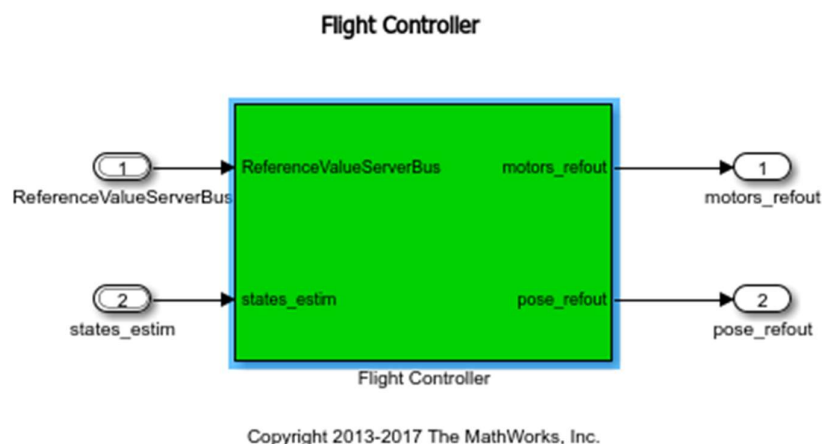


Figura 16

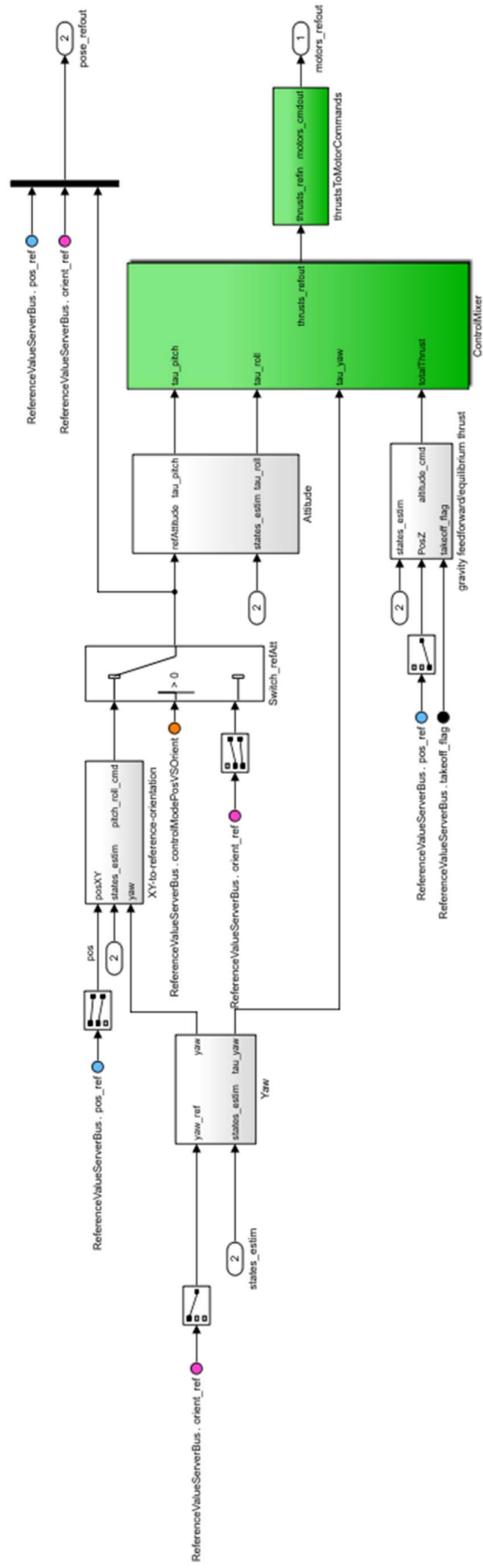


Figura 17

Controllore dell'angolo di imbardata (yaw) In Fig.18 viene riportato il controllore che implementa la legge di controllo per l'angolo di imbardata.

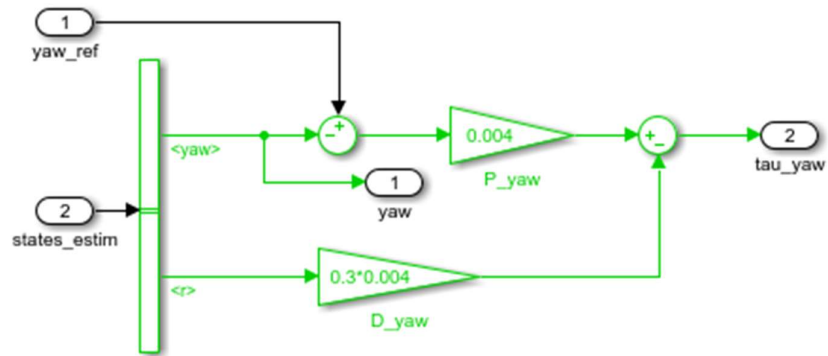


Figura 18

L'algoritmo è di tipo PD, presenta quindi un'azione proporzionale all'errore e una proporzionale alla derivata dell'angolo. Si nota che l'azione derivativa non è calcolata sull'errore dell'angolo ma sull'angolo stesso, infine le due azioni non vengono sommate ma l'azione derivativa viene invece sottratta per creare lo sforzo di controllo. Questa variazione di struttura rispetto al PID tradizionale (chiamato anche *TextBook PID Controller*) viene implementata per diminuire l'effetto di variazioni troppo alte nell'errore. Questo potrebbe accadere, per esempio, durante una variazione a gradino del riferimento.

Controllore delle coordinate x-y Anche per il controllo delle grandezze del piano (X-Y) viene utilizzato un algoritmo di tipo PD (Fig.19).

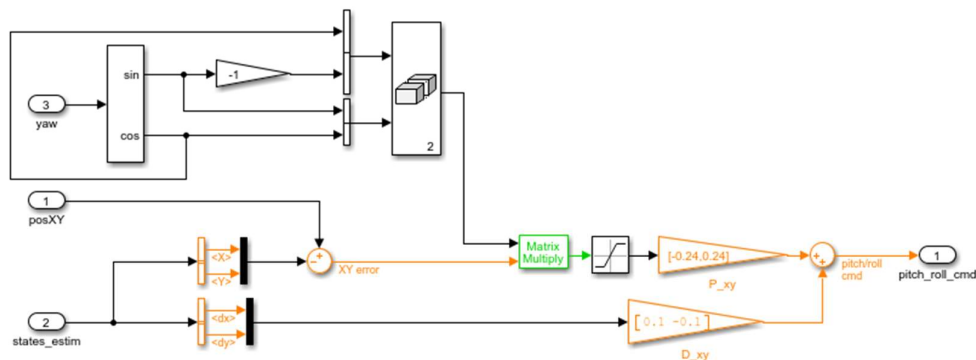


Figura 19

Oltre al riferimento e ai valori di stima delle grandezze, il blocco di controllo riceve in ingresso l'angolo di yaw. Infatti, i valori delle coordinate sul piano dipendono dall'angolo di imbardata. Oltre a questo, si nota che il segnale di comando creato non è mandato direttamente al processo ma è utilizzato per modificare gli angoli di pitch e roll (beccheggio e rollio). Come abbiamo già detto, infatti, essendo il sistema sotto-attuato, non è possibile controllare le coordinate del piano x-y direttamente, bisognerà invece agire sulle altre grandezze.

Controllori degli angoli di beccheggio e rollio (pitch e roll)

I sottosistemi relativi agli angoli di pitch e roll sono molto simili e pertanto viene utilizzata la stessa struttura di controllo (Fig.20)

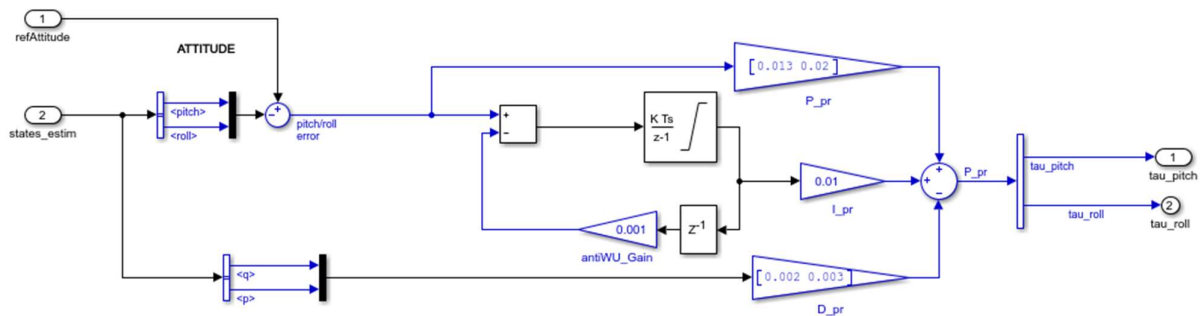


Figura 20

In questo caso viene utilizzato un controllore PID completo (sono presenti tutti e tre le tipologie di azioni). È importante notare che è stato implementato un guadagno (*antiWU_Gain*) per ridurre l'effetto della saturazione nell'azione integrale. I segnali di riferimento sono ottenuti dal controllore delle coordinate x-y e in uscita abbiamo lo sforzo di controllo sui motori per la regolazione dei due angoli.

Controllo dell'altitudine (coordinata z) Per quanto riguarda l'altitudine, invece, il controllore si presenta un po' più complesso. Il drone, durante il decollo, ha bisogno di una spinta maggiore per alzarsi di quanta ne ha bisogno mentre è già in volo. Questo problema viene risolto tramite la struttura in Fig.21

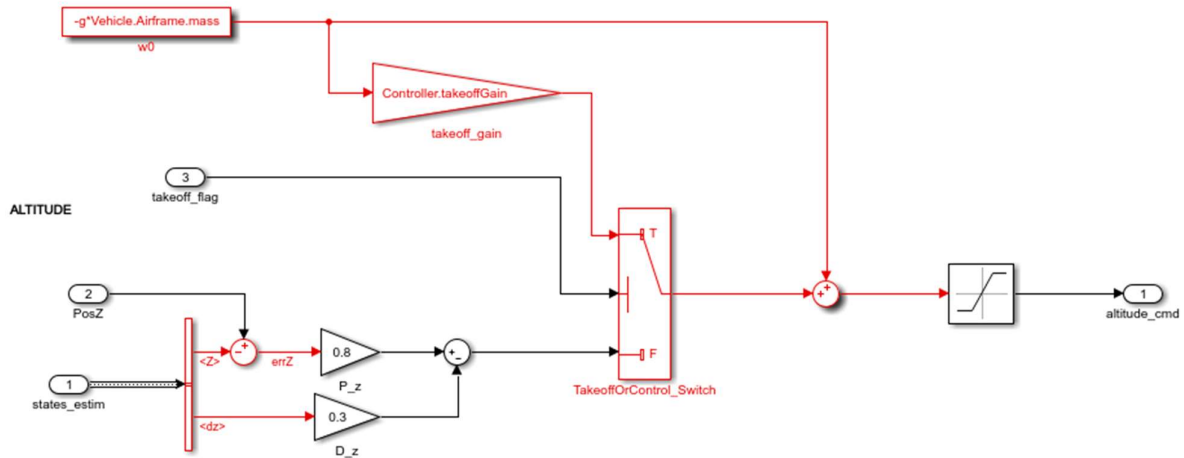


Figura 21

Notiamo che inizialmente, il `TakeOffOrControl_Switch`, lascia passare verso lo sforzo di controllo solo una spinta iniziale, in modo tale da far sollevare il drone da terra. Questa spinta è pari al peso del veicolo moltiplicato per un coefficiente chiamato `takeoffGain`. Una volta che il drone sarà decollato, il valore del `takeoff_flag` sarà maggiore di zero e lo switch andrà a selezionare il controllore PD. Anche in questo caso il controllore avrà una azione derivativa implementata sull'uscita z e non sull'errore `errZ`.

4 Analisi dei controllori PID

Come è stato già mostrato, il modello asbQuadcopter presenta dei regolatori PID per i vari gradi di libertà. In questa sezione si vuole analizzare l'andamento dello stato del sistema ottenuto da questi regolatori implementati da MathWorks.

La dinamica voluta consiste semplicemente in un decollo con stabilizzazione di quota a circa 1.5 m e angoli di rotazione degli assi stabili su zero radianti. All'interno del modello Simulink è possibile manipolare questi riferimenti dal blocco *Signal Builder*. Volendo ottenere i valori detti poc'anzi, i segnali di riferimento dovranno essere impostati in questo modo (Fig.22):

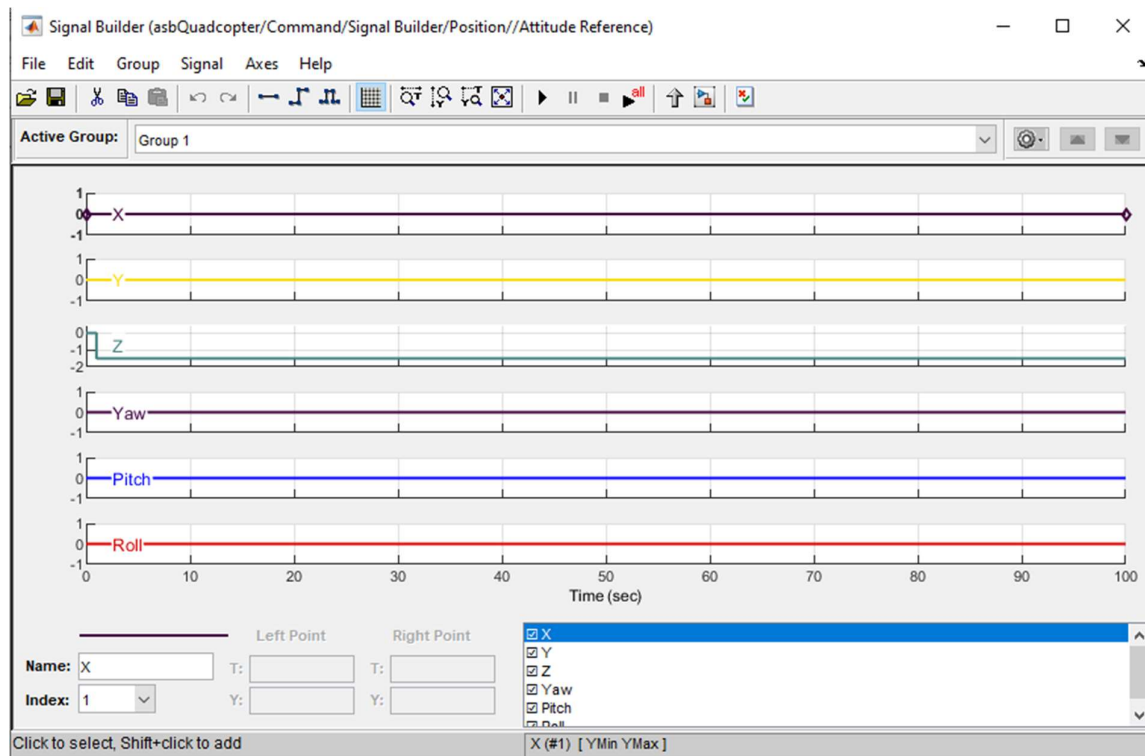


Figura 22

Per visualizzare invece i grafici dei valori temporali dei vari gradi di libertà possiamo utilizzare un blocco contenuto all'interno del Flight Control System. Qui dentro è possibile trovare il blocco *Logging* che presenta tutti i grafici sulle variabili più importanti (Fig.23).

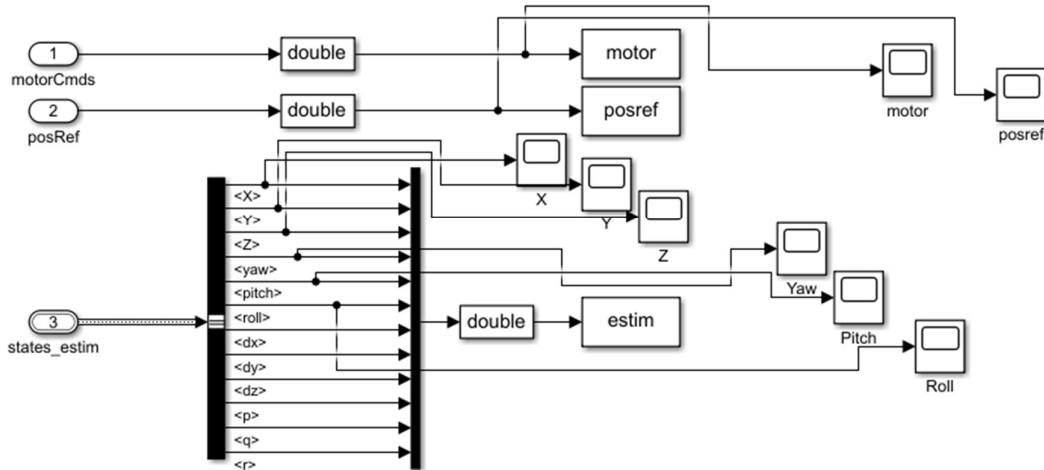


Figura 23

4.1 Andamento temporale delle coordinate di traslazione (x,y,z)

Andiamo a mostrare ora i grafici delle coordinate x, y e z.

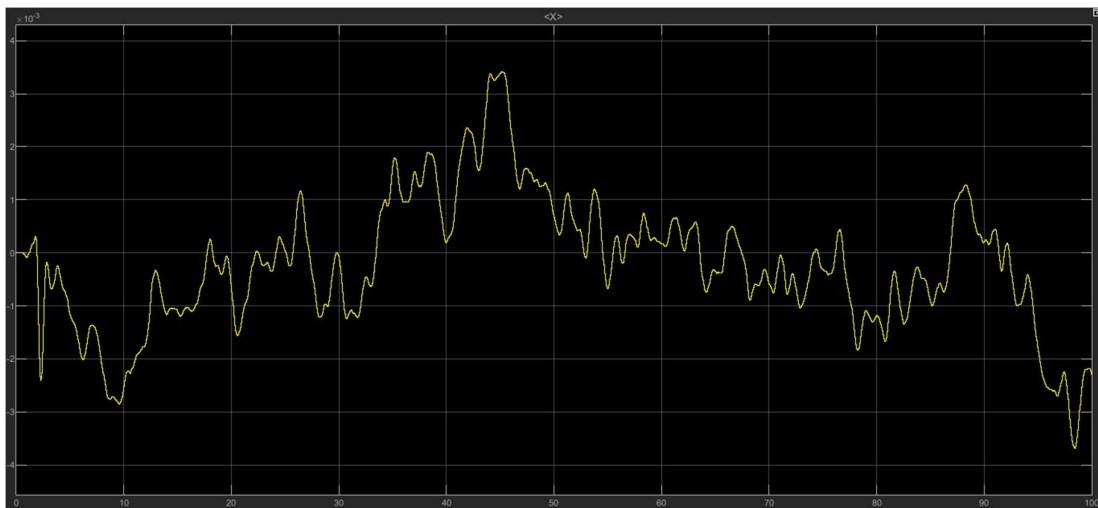


Figura 24

In Fig.24 è mostrato il grafico della coordinata x rispetto al tempo. Si nota come il massimo errore rispetto al valore di riferimento è di circa 4 mm. Naturalmente questo scostamento può considerarsi trascurabile e pertanto l'uscita può considerarsi stabile attorno al valore nullo.

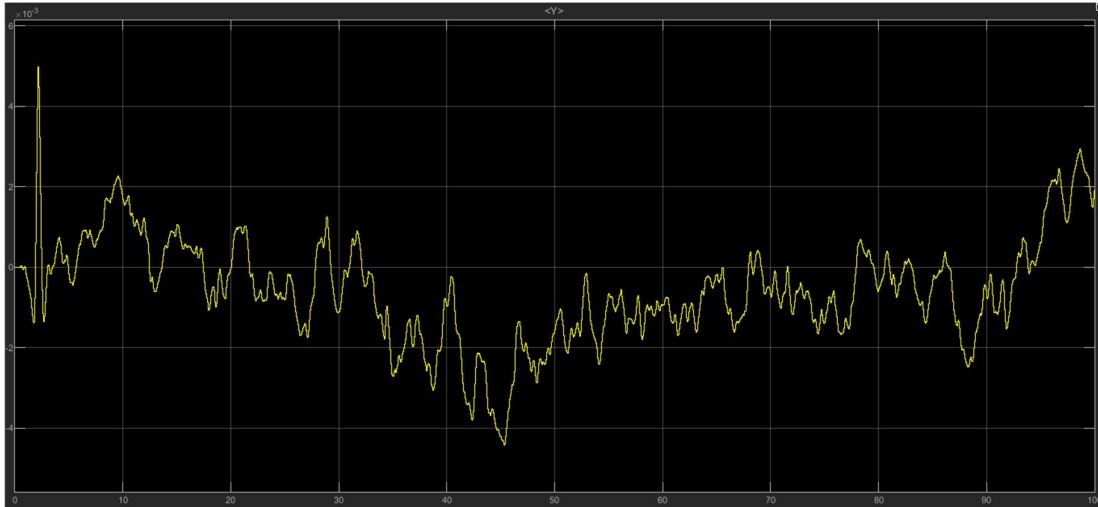


Figura 25

Il grafico dell'andamento della coordinata y è invece mostrato in Fig.25. Anche in questo caso l'errore è dell'ordine del millimetro e l'andamento temporale dell'uscita è soddisfacente.

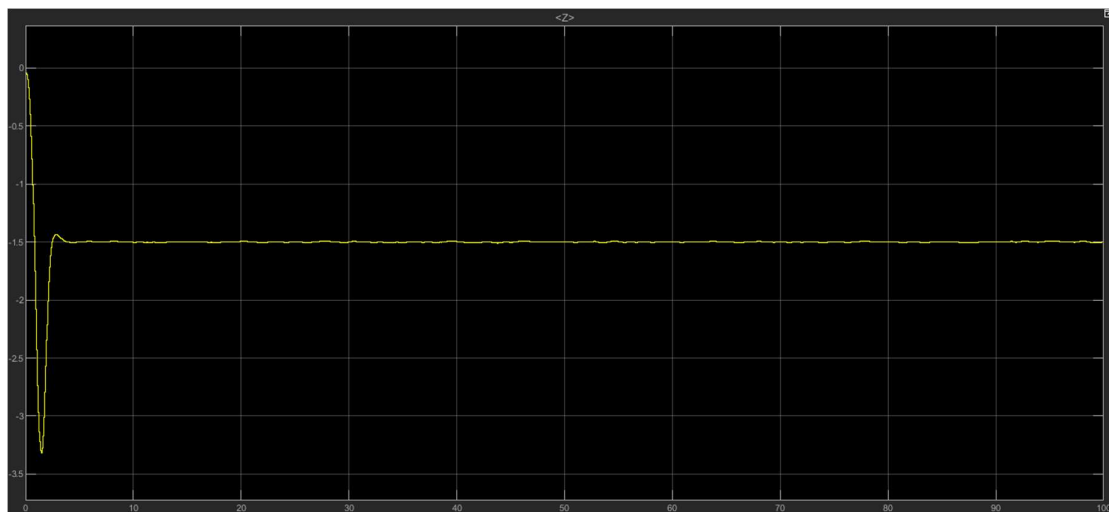


Figura 26

L'andamento della quota z invece è molto diverso (Fig.26). Difatti la quota z , a differenza di x e y , presenta un riferimento non nullo ma pari a -1.5 m. Nel grafico si nota una sovralongazione abbastanza sostenuta (-3.3 m) a circa 1.4 secondi dall'inizio della simulazione. Successivamente l'uscita decresce rapidamente e si assesta al valore di riferimento in circa 4 secondi.

4.2 Andamento temporale degli angoli di rotazione attorno agli assi

Si mostrano i grafici degli angoli di beccheggio (Fig.27), rollio (Fig.28) e imbardata (Fig.29).

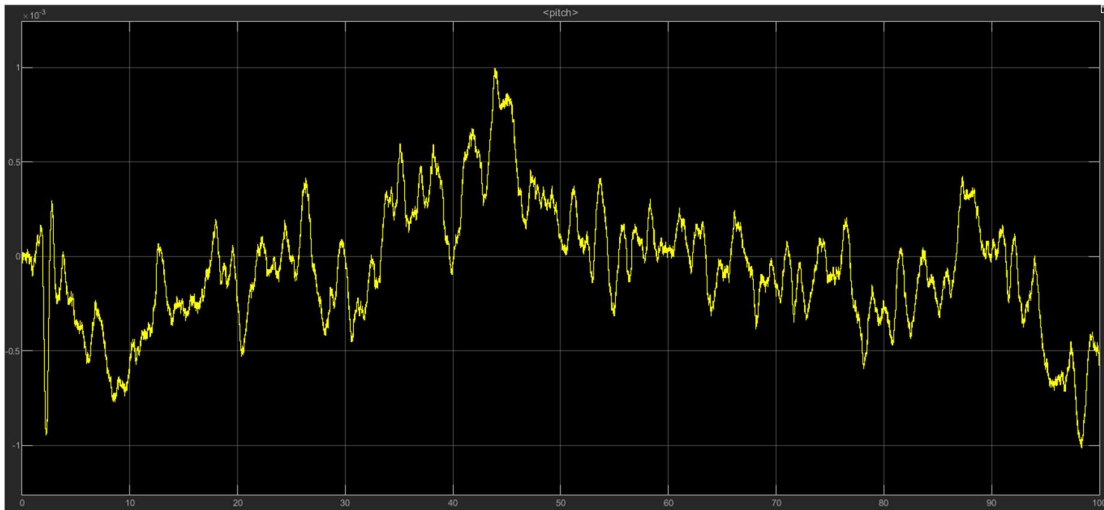


Figura 27

Il pitch rimane stabile attorno all'origine, con un errore dell'ordine di 10^{-3} .

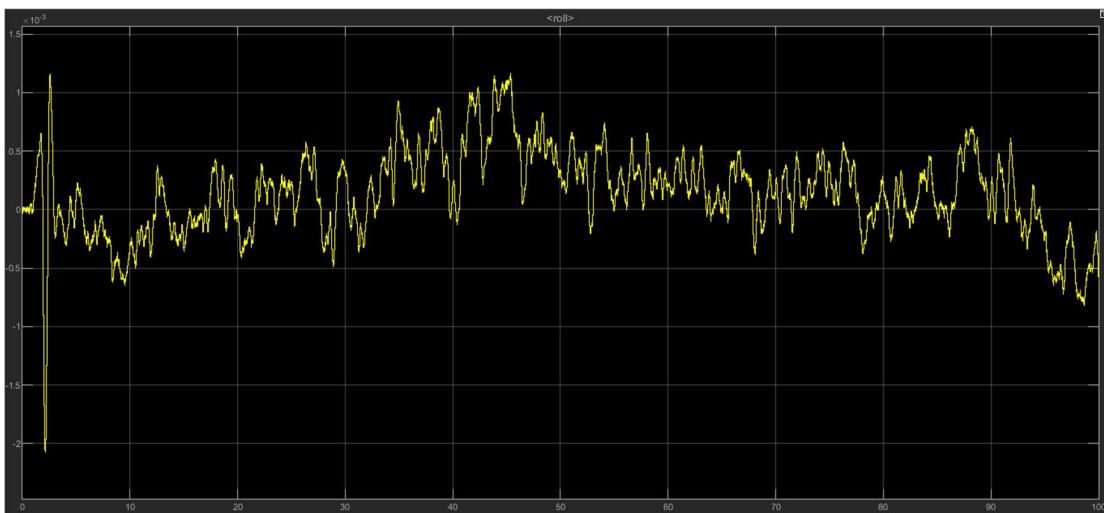


Figura 28

L'andamento dell'angolo di roll è molto simile al precedente e anch'esso presenta un errore di pochi millesimi di radiante.

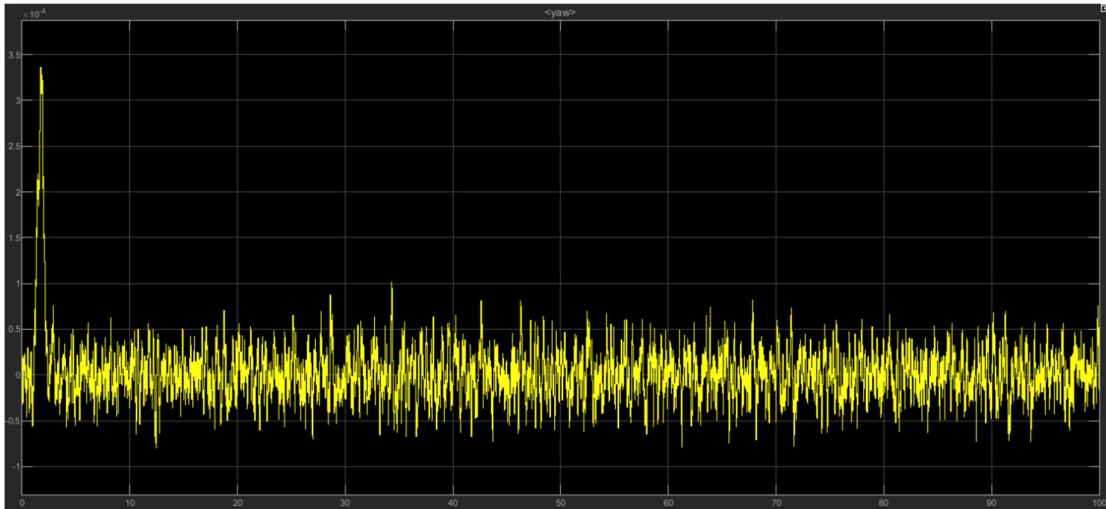


Figura 29

L'angolo di yaw presenta invece un'oscillazione con frequenza più alta e un errore medio più piccolo rispetto ai casi precedenti. Le misure fatte infatti si discostano dal riferimento di un massimo di 3.5×10^{-4} .

5 Modello linearizzato

Il sistema presentato nei capitoli precedenti è un sistema non lineare e pertanto non rispetta il principio di sovrapposizione degli effetti. Trattare con un sistema lineare è molto vantaggioso perché ci permette di utilizzare varie tecniche di controllo ormai consolidate in letteratura. In linea generale, quindi, si cerca di trovare un sistema lineare che vada ad approssimare il comportamento del sistema non lineare attorno ad un certo valore di equilibrio. La tecnica di linearizzazione più usata fa uso dello sviluppo in *serie di Taylor* arrestato al primo ordine.

5.1 Linearizzazione di un sistema dinamico non lineare tramite serie di Taylor

Dato un sistema non lineare MIMO (Multiple input – Multiple output), sia x la variabile di stato, u la variabile di ingresso e y la variabile di uscita, avremo in generale delle equazioni del tipo:

$$\dot{x}(t) = f(x(t), u(t)) \quad (28)$$

$$y(t) = g(x(t), u(t)) \quad (29)$$

Per procedere con la linearizzazione dobbiamo definire un punto di lavoro (o di equilibrio) attorno al quale approssimare l'andamento del nostro sistema. Sia \tilde{u} un ingresso costante, applicandolo al sistema otteniamo lo stato di equilibrio \tilde{x} se tale stato risolve l'equazione:

$$f(\tilde{x}, \tilde{u}) = 0 \quad (30)$$

Si nota che applicando un ingresso costante è possibile avere molteplici punti di equilibrio o addirittura infiniti. Ora, definiamo \tilde{y} come:

$$\tilde{y} = g(\tilde{x}, \tilde{u}) \quad (31)$$

Consideriamo le variazioni rispetto ai punti di equilibrio:

$$u(t) = \tilde{u} + \delta u(t) \quad (32)$$

$$x(t) = \tilde{x} + \delta x(t) \quad (33)$$

$$y(t) = \tilde{y} + \delta y(t) \quad (34)$$

Il sistema diventa:

$$\delta \dot{x}(t) = f(\tilde{x} + \delta x(t), \tilde{u} + \delta u(t)) \quad (35)$$

$$\tilde{y} + \delta y(t) = g(\tilde{x} + \delta x(t), \tilde{u} + \delta u(t)) \quad (36)$$

Con stato iniziale nel tempo t_0 :

$$x_0 = x(t_0) = \tilde{x} + \delta x(t_0) \quad (37)$$

Supponendo che le funzioni f e g siano differenziabili, possiamo svilupparle in serie di Taylor rispetto alle variabili x e u nell'intorno del punto di equilibrio. Otteniamo:

$$\delta \dot{x}(t) = f(\tilde{x}, \tilde{u}) + \left. \frac{\partial f(x,u)}{\partial x} \right|_{\substack{x=\tilde{x} \\ u=\tilde{u}}} \delta x(t) + \left. \frac{\partial f(x,u)}{\partial u} \right|_{\substack{x=\tilde{x} \\ u=\tilde{u}}} \delta u(t) \quad (38)$$

$$\tilde{y} + \delta y(t) = g(\tilde{x}, \tilde{u}) + \left. \frac{\partial g(x,u)}{\partial x} \right|_{\substack{x=\tilde{x} \\ u=\tilde{u}}} \delta x(t) + \left. \frac{\partial g(x,u)}{\partial u} \right|_{\substack{x=\tilde{x} \\ u=\tilde{u}}} \delta u(t) \quad (39)$$

Possiamo racchiudere le equazioni (38) e (39) in una forma più compatta:

$$\delta \dot{x}(t) = \mathbf{A} \delta x(t) + \mathbf{B} \delta u(t) \quad (40)$$

$$\delta y(t) = \mathbf{C} \delta x(t) + \mathbf{D} \delta u(t) \quad (41)$$

$$\delta x(t_0) = \delta x_0 \quad (42)$$

In cui le matrici \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} sono matrici Jacobiane di f e g rispetto alle variabili x e u .

5.2 Modello linearizzato del drone e implementazione Simulink

Il pacchetto *Simulink Support Package for Parrot Minidrones* consente il passaggio dal modello del drone non lineare a quello linearizzato. Ciò è possibile cambiando il valore della variabile *VSS_VEHICLE*. Con *VSS_VEHICLE = 0* avremo il sistema linearizzato, con un valore *VSS_VEHICLE = 1* andremo ad utilizzare il modello non lineare (Fig.30).

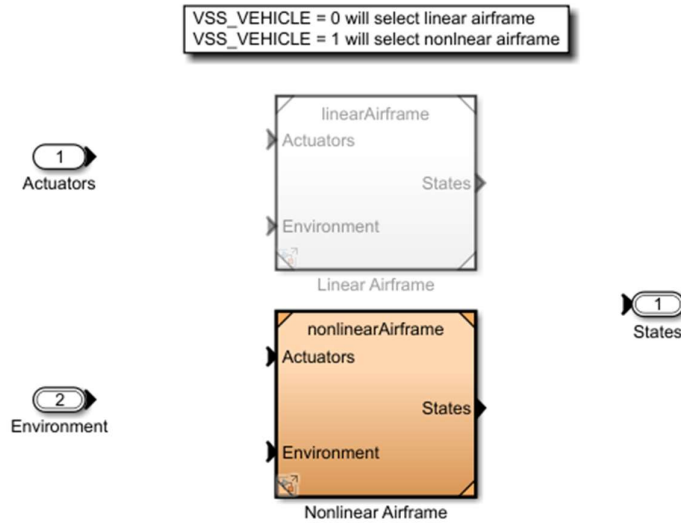


Figura 30

Dopo aver ricompilato il codice avremo a disposizione un nuovo blocco del processo chiamato *Linear Airframe*. All'interno vi troveremo una struttura di questo tipo (Fig.31):

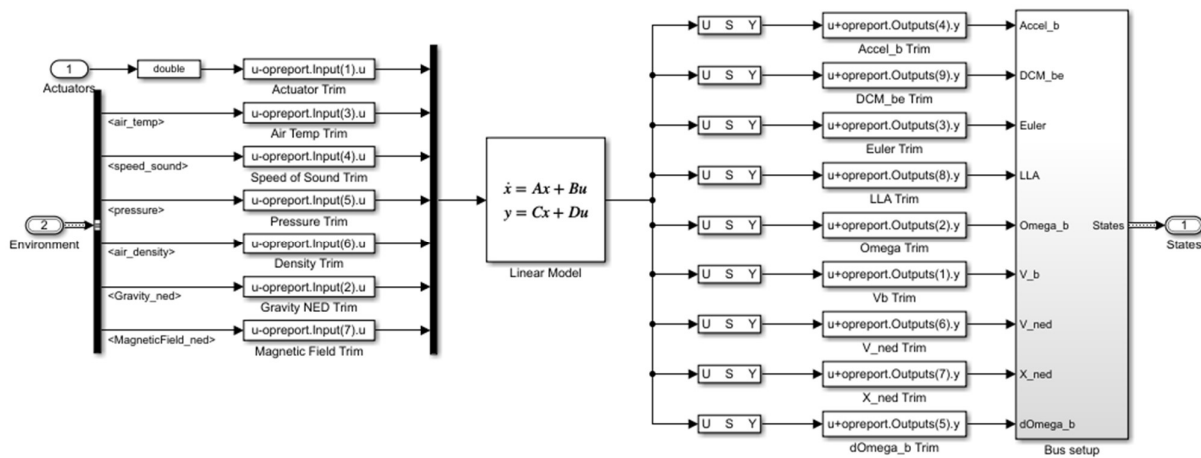


Figura 31

Il modello sarà quindi formato da un sistema dinamico contenente le varie matrici A, B, C, D del modello linearizzato e vari *bias* sovrapposti agli ingressi e alle uscite. Questo viene fatto perché, come visto precedentemente nelle formule (40) e (41), il sistema linearizzato prende in ingresso e manda in uscita solo le variazioni rispetto al punto di lavoro. Bisogna quindi aggiungere, al segnale di uscita δy , il valore dell'uscita all'equilibrio \tilde{y} . Stesso discorso vale con gli ingressi: viene sottratto l'ingresso costante di riferimento \tilde{u} in modo tale da ottenere in entrata al sistema solo la variazione di ingresso δu .

Il modello presentato utilizza delle matrici della dinamica che sono state ricavate tramite la funzione *trimLinearizeOpPoint*.

Queste matrici sono salvate all'interno della struttura *linsys* e sono:

- **A:** matrice 12×12 che rappresenta la dinamica degli stati (Fig.32)
- **B:** matrice 12×14 degli ingressi del sistema (Fig.33)
- **C:** matrice 33×12 della dinamica stato-uscita (Fig.34)
- **D:** matrice 33×14 che pesa l'azione diretta degli ingressi sulle uscite (Fig.35)

La dimensione delle matrici dipende ovviamente dal numero di variabili del sistema (12 variabili di stato, 14 variabili di ingresso e 33 variabili di uscita).

| | | | | | | | | | | | | | | |
|-------------------|-----------|-----------|----------|-----------|-----------|-----------|---|---|---|-----------|-----------|-----------|---|-------|
| phi theta psi(1)' | 0 | 3,66E-17 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | 1 | 3,03E-23 | -5,23E-12 | X | phi |
| phi theta psi(2)' | -3,66E-17 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | 0,00E+00 | 1 | 5,23E-12 | | theta |
| phi theta psi(3)' | -6,83E-13 | -2,03E-28 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | 0,00E+00 | -5,23E-12 | 1 | | psi |
| ub,vb,wb(1)' | 0,00E+00 | -9,81 | 0 | -3,44E-09 | -3,44E-09 | 6,85E-13 | 0 | 0 | 0 | -5,46E-11 | 0,137987 | -3,28E-13 | | ub |
| ub,vb,wb(2)' | 9,81 | -2,85E-22 | 0 | -3,66E-09 | -3,66E-09 | -6,85E-13 | 0 | 0 | 0 | -0,13799 | 5,81E-11 | -3,52E-13 | | vb |
| ub,vb,wb(3)' | 5,45E-11 | 5,45E-11 | 0 | -6,83E-13 | 6,83E-13 | 0 | 0 | 0 | 0 | 3,29E-13 | 3,51E-13 | 0,00E+00 | | wb |
| xe,ye,ze(1)' | 1,72E-24 | 1,35E-19 | 3,29E-13 | 1 | -1,50E-18 | -5,23E-12 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | | xe |
| xe,ye,ze(2)' | -1,35E-19 | 2,02E-37 | 3,51E-13 | 1,50E-18 | 1 | 5,23E-12 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | | ye |
| xe,ye,ze(3)' | -3,29E-13 | -3,51E-13 | 0 | 5,23E-12 | -5,23E-12 | 1 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | | ze |
| p,q,r(1)' | 0,00E+00 | 0,00E+00 | 0 | -5,76E-08 | -5,76E-08 | -2,73E-14 | 0 | 0 | 0 | -2,17153 | 9,14E-10 | 4,33E-13 | | p |
| p,q,r(2)' | 0,00E+00 | 0,00E+00 | 0 | 4,04E-08 | 4,04E-08 | -2,16E-14 | 0 | 0 | 0 | 6,41E-10 | -1,6192 | -5,14E-13 | | q |
| p,q,r(3)' | 0,00E+00 | 0,00E+00 | 0 | -1,44E-19 | -1,28E-19 | 3,62E-19 | 0 | 0 | 0 | 1,22E-13 | 1,12E-13 | -6,88E-09 | | r |

Matrice A - Figura 32

| | | | | | | | | | | | | | | | | |
|-------------------|-----------|-----------|-----------|-----------|-----------|---|---|---|-----------|----------|-----------|---|---|---|---|-----------------------------|
| phi theta psi(1)' | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | X | 'Actuators(1)' |
| phi theta psi(2)' | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | | 'Actuators(2)' |
| phi theta psi(3)' | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | | 'Actuators(3)' |
| ub,vb,wb(1)' | -6,90E-17 | 6,90E-17 | -6,90E-17 | 6,90E-17 | -9,93E-14 | 0 | 0 | 0 | 1 | 1,50E-18 | 5,23E-12 | 0 | 0 | 0 | | 'Actuators(4)' |
| ub,vb,wb(2)' | 6,76E-17 | -6,76E-17 | 6,76E-17 | -6,76E-17 | 9,82E-14 | 0 | 0 | 0 | -1,50E-18 | 1 | -5,23E-12 | 0 | 0 | 0 | | 'AtmosphereBus.air_density' |
| ub,vb,wb(3)' | -9,61E-03 | 9,61E-03 | -9,61E-03 | 9,61E-03 | -8,28547 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 1 | 0 | 0 | 0 | | 'AtmosphereBus.air_temp' |
| xe,ye,ze(1)' | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | | 'AtmosphereBus.pressure' |
| xe,ye,ze(2)' | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | | 'AtmosphereBus.speed_sound' |
| xe,ye,ze(3)' | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | | 'Gravity_ned(1)' |
| p,q,r(1)' | 0,420192 | 0,420192 | -0,42019 | -0,42019 | 2,53E-14 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | | 'Gravity_ned(2)' |
| p,q,r(2)' | 0,313317 | -0,31332 | -0,31332 | 0,313317 | 1,89E-14 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | | 'Gravity_ned(3)' |
| p,q,r(3)' | -1,15E-02 | -1,15E-02 | -1,15E-02 | -1,15E-02 | -7,94E-16 | 0 | 0 | 0 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0 | 0 | 0 | | 'MagneticField_ned(1)' |
| | | | | | | | | | | | | | | | | 'MagneticField_ned(2)' |
| | | | | | | | | | | | | | | | | 'MagneticField_ned(3)' |

Matrice B - Figura 33

| | | | | | | | | | | | | | | |
|------------------|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|----|-----------|----------|-----------|---|-------|
| 'Accel_body(1)' | 0 | -9,81 | 0 | -3,44E-09 | -3,44E-09 | 1,84E-15 | 0 | 0 | 0 | -5,46E-11 | 0,137987 | 7,21E-16 | X | phi |
| 'Accel_body(2)' | 9,81 | -2,85E-22 | 0 | -3,66E-09 | -3,66E-09 | -1,73E-15 | 0 | 0 | 0 | -0,13799 | 5,81E-11 | -7,21E-16 | | theta |
| 'Accel_body(3)' | 5,45E-11 | 5,45E-11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | psi |
| 'DCM_be(1)' | 0 | 5,55E-12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | ub |
| 'DCM_be(2)' | -5,23E-12 | -5,23E-12 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | vb |
| 'DCM_be(3)' | 1,50E-18 | 1 | -5,23E-12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | wb |
| 'DCM_be(4)' | 0 | 8,31E-30 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | xe |
| 'DCM_be(5)' | 5,55E-12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | ye |
| 'DCM_be(6)' | -1 | 1,50E-18 | -5,23E-12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | ze |
| 'DCM_be(7)' | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | pe |
| 'DCM_be(8)' | 1 | -2,90E-23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | qe |
| 'DCM_be(9)' | 5,55E-12 | 5,55E-12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | re |
| 'Euler(1)' | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'Euler(2)' | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'Euler(3)' | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'LLA(1)' | 0 | 0 | 0 | 0 | 0 | 0 | 9,00E-06 | 0 | 0 | 0 | 0 | 0 | | |
| 'LLA(2)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1,21E-05 | 0 | 0 | 0 | 0 | | |
| 'LLA(3)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | | |
| 'Omega_body(1)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | |
| 'Omega_body(2)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | |
| 'Omega_body(3)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| 'V_body(1)' | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'V_body(2)' | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'V_body(3)' | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'V_ned(1)' | 1,72E-24 | 1,35E-19 | 3,29E-13 | 1 | -1,50E-18 | -5,23E-12 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'V_ned(2)' | -1,35E-19 | 2,02E-37 | 3,51E-13 | 1,50E-18 | 1 | 5,23E-12 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'V_ned(3)' | -3,29E-13 | -3,51E-13 | 0 | 5,23E-12 | -5,23E-12 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'X_ned(1)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | |
| 'X_ned(2)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | |
| 'X_ned(3)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | | |
| 'dOmega_body(1)' | 0 | 0 | 0 | -5,76E-08 | -5,76E-08 | -2,73E-14 | 0 | 0 | 0 | -2,17153 | 9,14E-10 | 4,33E-13 | | |
| 'dOmega_body(2)' | 0 | 0 | 0 | 4,04E-08 | 4,04E-08 | -2,16E-14 | 0 | 0 | 0 | 6,41E-10 | -1,6192 | -5,14E-13 | | |
| 'dOmega_body(3)' | 0 | 0 | 0 | -1,44E-19 | -1,28E-19 | 3,62E-19 | 0 | 0 | 0 | 1,22E-13 | 1,12E-13 | -6,88E-09 | | |

Matrice C - Figura 34

| | | | | | | | | | | | | | | | | |
|------------------|-----------|-----------|-----------|-----------|-----------|---|---|---|-----------|----------|-----------|---|---|---|---|-----------------------------|
| 'Accel_body(1)' | -6,90E-17 | 6,90E-17 | -6,90E-17 | 6,90E-17 | -9,93E-14 | 0 | 0 | 0 | 1 | 1,50E-18 | 5,23E-12 | 0 | 0 | 0 | X | 'Actuators(1)' |
| 'Accel_body(2)' | 6,76E-17 | -6,76E-17 | 6,76E-17 | -6,76E-17 | 9,82E-14 | 0 | 0 | 0 | -1,50E-18 | 1 | -5,23E-12 | 0 | 0 | 0 | | 'Actuators(2)' |
| 'Accel_body(3)' | -0,00961 | 0,009607 | -0,00961 | 0,009607 | -8,28547 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | 'Actuators(3)' |
| 'DCM_be(1)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 'Actuators(4)' |
| 'DCM_be(2)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 'AtmosphereBus.air_density' |
| 'DCM_be(3)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 'AtmosphereBus.air_temp' |
| 'DCM_be(4)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 'AtmosphereBus.pressure' |
| 'DCM_be(5)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 'AtmosphereBus.speed_sound' |
| 'DCM_be(6)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 'Gravity_ned(1)' |
| 'DCM_be(7)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 'Gravity_ned(2)' |
| 'DCM_be(8)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 'Gravity_ned(3)' |
| 'DCM_be(9)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 'MagneticField_ned(1)' |
| 'Euler(1)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 'MagneticField_ned(2)' |
| 'Euler(2)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 'MagneticField_ned(3)' |
| 'Euler(3)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'LLA(1)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'LLA(2)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'LLA(3)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'Omega_body(1)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'Omega_body(2)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'Omega_body(3)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'V_body(1)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'V_body(2)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'V_body(3)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'V_ned(1)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'V_ned(2)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'V_ned(3)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'X_ned(1)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'X_ned(2)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'X_ned(3)' | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'dOmega_body(1)' | 0,420192 | 0,420192 | -0,42019 | -0,42019 | 0,00E+00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'dOmega_body(2)' | 0,313317 | -0,31332 | -0,31332 | 0,313317 | 0,00E+00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 'dOmega_body(3)' | -0,01154 | -0,01154 | -0,01154 | -0,01154 | -1,33E-15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |

Matrice D - Figura 35

5.3 Sottosistema SISO linearizzato della dinamica dell'altitudine

Una volta ottenute le matrici della dinamica del modello linearizzato è possibile ricavare il sottosistema riguardante uno dei vari gradi di libertà. Per la dinamica della coordinata z è sufficiente considerare la variabile di stato \mathbf{z}_b e la sua derivata nel tempo \mathbf{w}_b , come ingresso lo sforzo sugli attuatori **Actuators (1)(2)(3)(4)** e come uscita **LLA (3)**, ossia l'altitudine in coordinate geodetiche.

Otteniamo le seguenti sottomatrici:

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} & \mathbf{B} &= \begin{pmatrix} 0 \\ -0.00961 \end{pmatrix} \\ \mathbf{C} &= (-1 \quad 0) & \mathbf{D} &= 0 \end{aligned} \quad (43)$$

Da queste matrici possiamo ricavarci una prima funzione di trasferimento ingresso-uscita:

$$FdT = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = \frac{0.00961}{s^2} \quad (44)$$

Questa funzione non è sufficiente ad esprimere la dinamica considerata. Abbiamo infatti tralasciato dei guadagni presenti fra controllo e processo. Ritorniamo al blocco FlightController presente in FCS. Notiamo che i segnali di controllo, prima di essere mandati al processo, passano in un blocco chiamato *ControlMixer* (Fig.36)

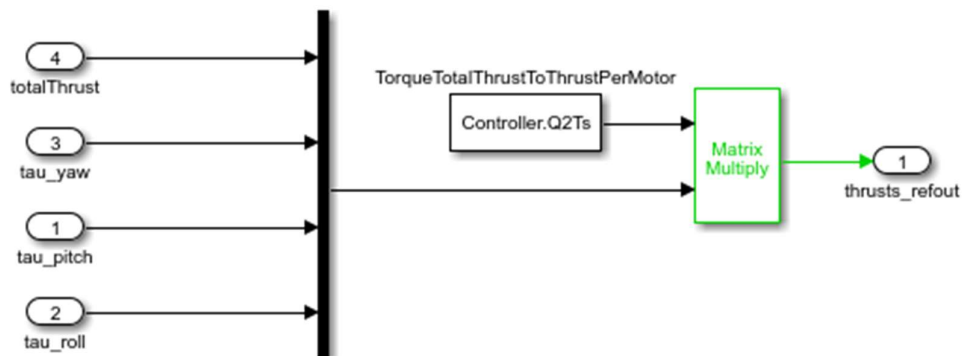


Figura 36

In questo blocco i vari sforzi di controllo vengono immagazzinati in un vettore e moltiplicati per la matrice *Controller.Q2Ts* (Fig.37).

```
>> 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
```

Figura 37

Ogni segnale viene quindi moltiplicato per un dato valore. Il segnale di controllo dell'altitudine viene moltiplicato per

$$K_1 = 0.25 \quad (45)$$

Successivamente, il vettore dei segnali passa nel blocco *thrustToMotorCommands* (Fig.38) che attua un'ultima amplificazione dei segnali e impone dei limiti di saturazione.

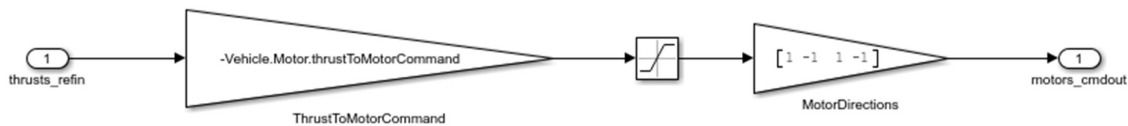


Figura 38

Il valore del coefficiente di guadagno è pari a:

$$K_2 = 1.5307 \times 10^3 \quad (46)$$

Andando ad unire i risultati ottenuti in (44), (45) e (46) ricaviamo la funzione di trasferimento completa della dinamica dell'altitudine:

$$FdT = K_1 * K_2 * [C(sI - A)^{-1}B] = \frac{3.678}{s^2} \quad (47)$$

6 Progettazione di un controllore per l'altitudine

Nel capitolo precedente si è riusciti a ricavare la funzione di trasferimento della dinamica della quota z del drone tramite il modello linearizzato. L'obiettivo ora è quello di progettare un controllore del processo che riesca ad ottenere un andamento dell'uscita simile a quella ricavata tramite PID. La tecnica di controllo che fa uso dei regolatori PID si basa sulla semplice taratura dei coefficienti delle tre azioni (proporzionale, derivativa e integrale). Questa struttura è limitante in certi casi, soprattutto quando abbiamo sistemi molto complessi che hanno bisogno di tecniche di controllo più sofisticate. In questa sede si utilizzerà una delle tecniche più famose in letteratura. Questa tecnica basa la sintesi del controllore sull'analisi in frequenza del sistema tramite diagrammi di Bode.

6.1 Analisi in frequenza del sistema e possibili approcci alla progettazione

Per prima cosa dobbiamo visualizzare il diagramma di Bode del sistema per capire le sue caratteristiche in frequenza (Fig.39). Per tracciare il grafico usiamo la funzione *bode* di Matlab.

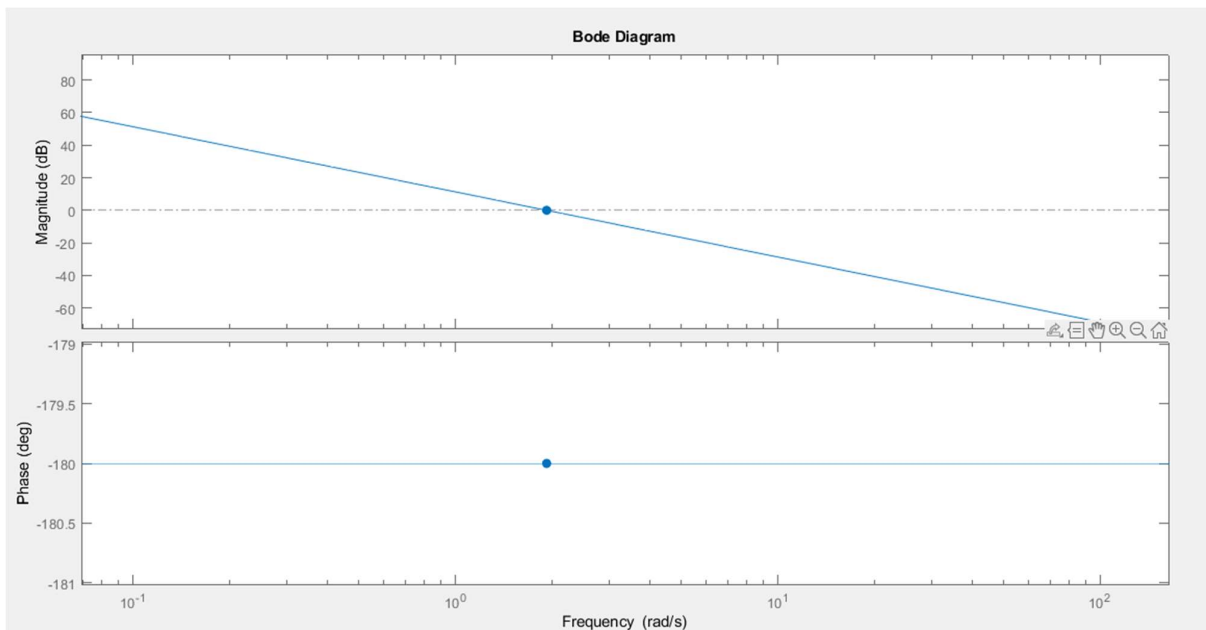


Figura 39

Dal grafico notiamo subito che, a causa dei due poli in $s = 0$, il diagramma del modulo è costituito da una retta con pendenza negativa di 40 dB/decade. Questa retta interseca l'asse a 0 dB alla frequenza 1.92 rad/s, questa è detta *pulsazione di attraversamento* e si indica con ω_t . Il diagramma delle fasi invece è costante a -180° . La funzione *bode* ci permette anche di visualizzare eventuali margini di stabilità del sistema, molto utili per dare un indizio sul comportamento del sistema a ciclo chiuso (Fig.40).

System: P
Gain Margin (dB): 0
At frequency (rad/s): 1.92
Closed loop stable? No

Margine di fase - Figura 40.a

System: P
Phase Margin (deg): 0
Delay Margin (sec): -0
At frequency (rad/s): 1.92
Closed loop stable? No

Margine di guadagno - Figura 40.b

I due margini di stabilità, a causa dei due poli del sistema, sono nulli e pertanto il sistema a ciclo chiuso sarà instabile. Durante la sintesi sarà necessario aumentare questi indici per ottenere un comportamento stabile del sistema in retroazione.

Andiamo ora ad analizzare il sistema a ciclo chiuso della sola dinamica dell'altitudine. Per fare questo è stato creato un modello semplificato visibile in Fig.41.

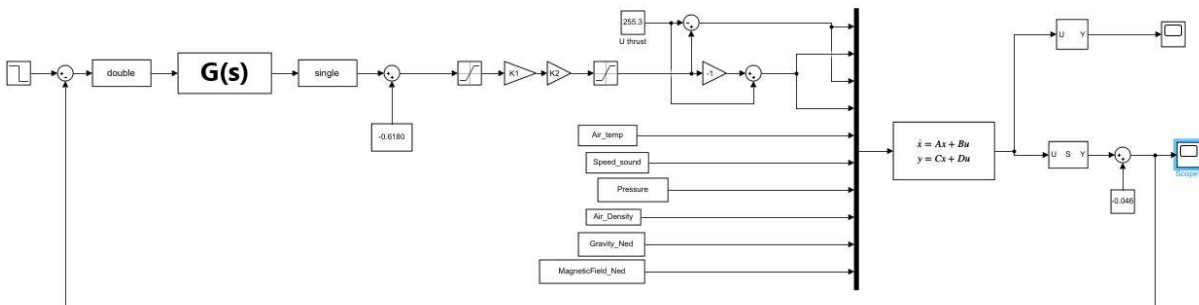


Figura 41

Come è possibile notare in figura è presente, fra controllore e processo, un segnale di disturbo costante. È importante tener presente che questo disturbo genererà in uscita una certa risposta. La risposta del disturbo, quindi, si andrà a sovrapporre sulla risposta al riferimento e questo potrebbe generare un'interferenza rispetto al corretto posizionamento del drone durante il volo.

A questo punto abbiamo due possibili approcci da seguire per la sintesi del controllore:

- Ottenere astatismo completo ai disturbi, eliminando completamente la risposta a regime dei disturbi.
- Attenuare la risposta a regime del disturbo con un aumento di guadagno.

6.2 Sintesi per tentativi del controllore con astatismo ai disturbi

Essendo il disturbo costante e agente fra controllore e processo, per avere un equilibrio astatico, bisognerà inserire un polo in $s = 0$ all'interno del controllore. Questo polo, infatti, andrà ad agire sulla risposta del disturbo facendola tendere asintoticamente verso il valore nullo. Un primo controllore avrà quindi una funzione di trasferimento del tipo:

$$G_1(s) = \frac{K_G}{s}$$

Con funzione di trasferimento in catena aperta F pari a:

$$F_1(s) = \frac{3.678K_G}{s^3}$$

Questo approccio ci permette di utilizzare un guadagno di controllore piccolo. Possiamo scegliere quindi $K_G = 1$.

Andiamo ora ad analizzare il comportamento in frequenza della nuova funzione di trasferimento F_1 (Fig.43).

L'aggiunta del nuovo polo ha peggiorato ancor di più il comportamento a ciclo chiuso. La fase è costante a -270° e il modulo decresce ancor più velocemente (-60dB/decade). Anche il margine di fase ne ha risentito e ora è pari a -90° (Fig.42).

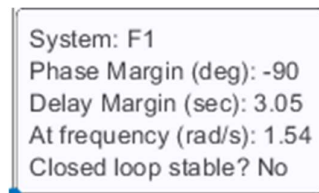


Figura 42

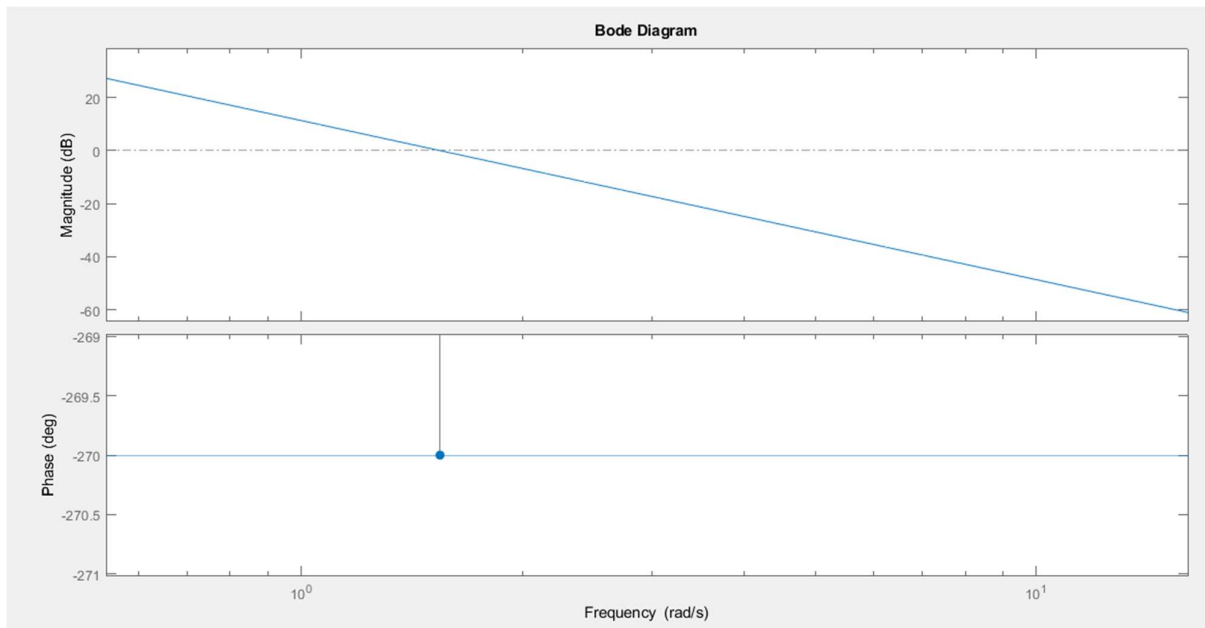


Figura 43

Poniamo ora delle specifiche di progetto standard tramite dei parametri del sistema a ciclo aperto:

- Modulo di risonanza $M_r \leq 2\text{dB}$
- Banda passante $B_3 = 2.5\text{ Hz}$

Che a ciclo aperto corrispondono a:

- Margine di fase $m_\phi \geq 47^\circ$
- Pulsazione di attraversamento $\omega_t = 10\text{ rad/s}$

Il controllore di primo tentativo G_1 non riesce a far rispettare questi parametri. Osservando i diagrammi di Bode in Fig.43 notiamo che, per soddisfare le specifiche, c'è bisogno di aumentare nella frequenza $\omega = 10\text{ rad/s}$ il modulo e la fase di, rispettivamente, 48.7 dB e 137°.

Per attuare questi cambiamenti bisognerà utilizzare una rete correttiva.

Le reti correttive sono delle reti dinamiche del primo ordine che presentano un polo e uno zero. La rete può essere detta anticipatrice o attenuatrice, questo dipende dal suo comportamento in frequenza e dal posizionamento dei poli degli zeri sul piano di Gauss. Se il polo è dominante rispetto allo zero la rete verrà detta attenuatrice, verrà detta anticipatrice se avviene il contrario. La generica rete anticipatrice ha una funzione di trasferimento del tipo:

$$R_a = \frac{1 + \tau_a s}{1 + \frac{\tau_a}{m_a} s} \quad (48)$$

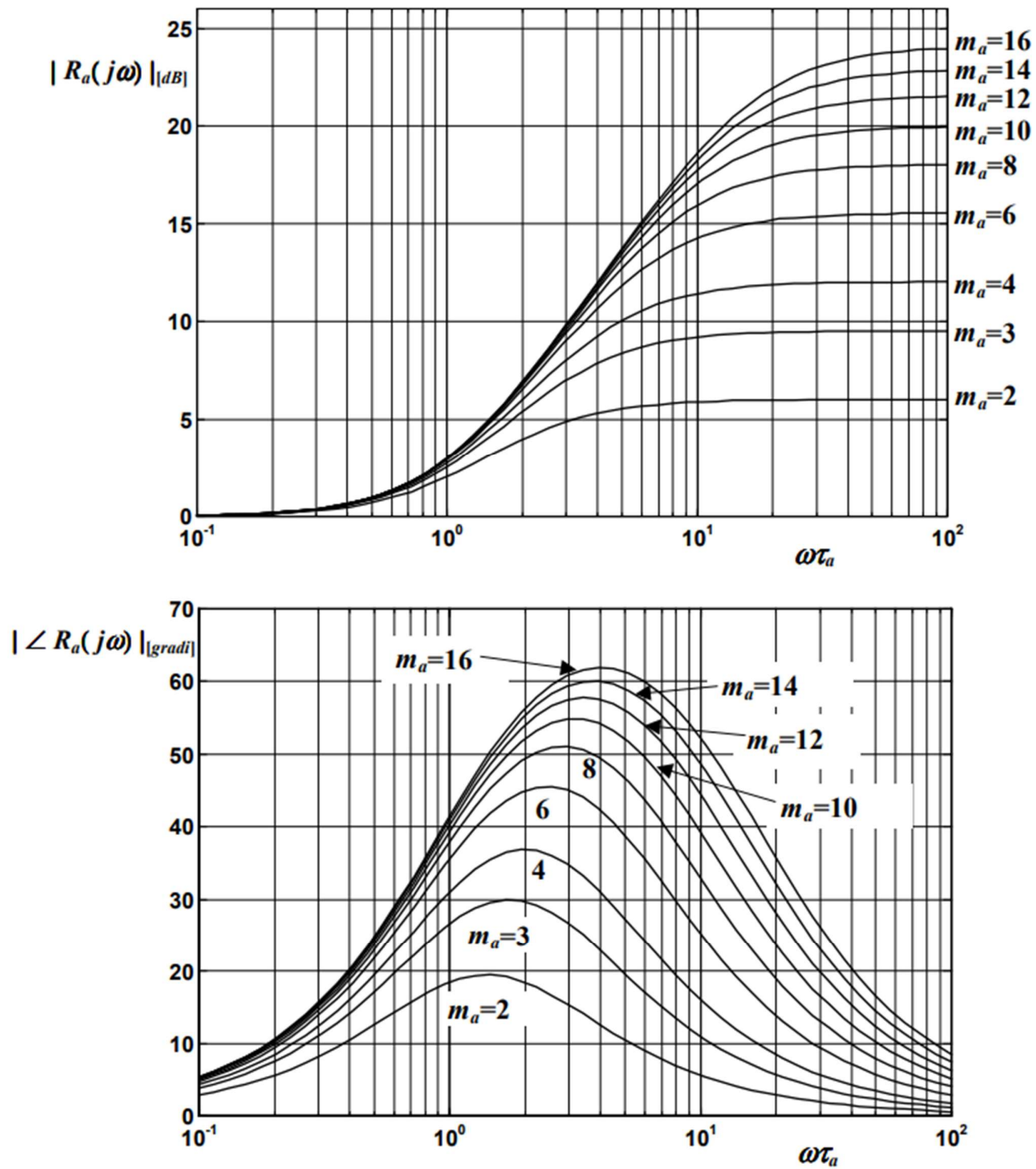


Figure 44

I parametri τ_a e m_a sono scelti tramite i diagrammi universali in Fig.44. I grafici sono normalizzati rispetto a τ_a e pertanto sono in funzione di $\omega\tau_a$. I parametri vengono scelti a seconda dell'aumento voluto di modulo e fase nella pulsazione ω .

Le reti attenuatrici invece hanno una funzione di trasferimento reciproca rispetto a quella delle reti appena viste. La FdT si presenta quindi così:

$$R_i(s) = \frac{1 + \frac{\tau_i}{m_i} s}{1 + \tau_i s} \quad (49)$$

Per la scelta dei parametri τ_i e m_i vengono utilizzati, anche in questo caso, i diagrammi in Fig.44. A differenza del caso precedente i diagrammi vanno letti "con segno opposto". Infatti, i valori sui grafici rappresentano la diminuzione di modulo e fase nella pulsazione desiderata ω .

Nel nostro caso andiamo a progettare tre reti anticipatrici in serie dato che una sola rete non sarebbe riuscita a compensare l'aumento di modulo e fase. Abbiamo scelto due reti R_{a1} e R_{a2} con $\tau_{a1} = \tau_{a2} = 1$ e $m_{a1} = m_{a2} = 16$ e una rete R_{a3} con $\tau_{a3} = 0.4$ e $m_{a3} = 16$.

Otteniamo:

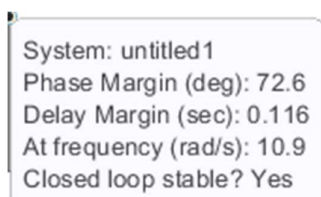
$$R_{a1}(s) = R_{a2}(s) = \frac{16 + 16s}{1 + 16s}$$

$$R_{a3}(s) = \frac{6.4 + 16s}{0.4 + 16s}$$

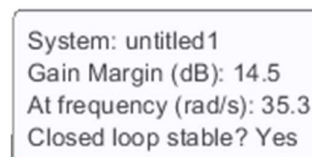
$$\begin{aligned} G_2(s) &= \frac{1}{s} R_{a1}(s) R_{a2}(s) R_{a3}(s) = \\ &= \frac{1638s^3 + 7373s^2 + 9830s + 4096}{0.4s^4 + 28.8s^3 + 614.4s^2 + 4096s} \end{aligned}$$

Riportiamo di seguito (Fig.46) i diagrammi di Bode di $F_2 = G_2 * P$.

Dalla figura si nota che le specifiche di progetto sono rispettate. La pulsazione di attraversamento ω_t ora è circa sui 10 rad/s e il margine di fase è aumentato molto, è pari a $m_\varphi = 72.6^\circ$. Questo nuovo controllore ha generato anche un margine di guadagno positivo di circa 14.5 dB (Fig.45). Il sistema si presenta stabile a ciclo chiuso.



Margine di fase - Figura 45.a



Margine di guadagno - Figura 45.b

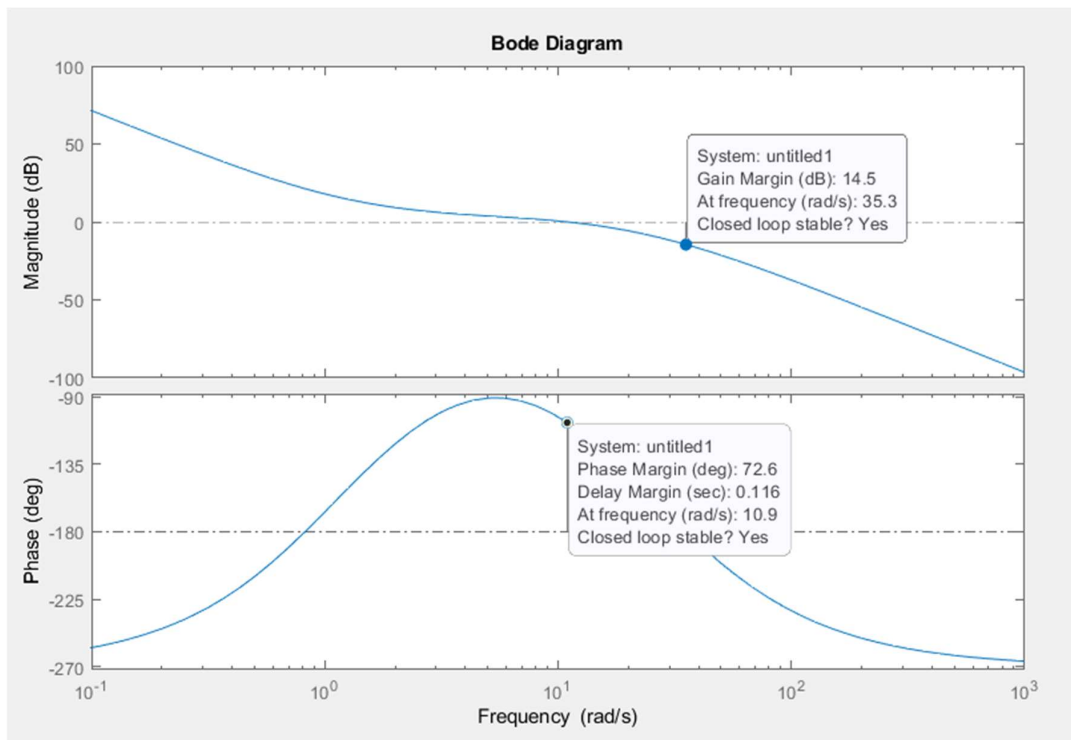


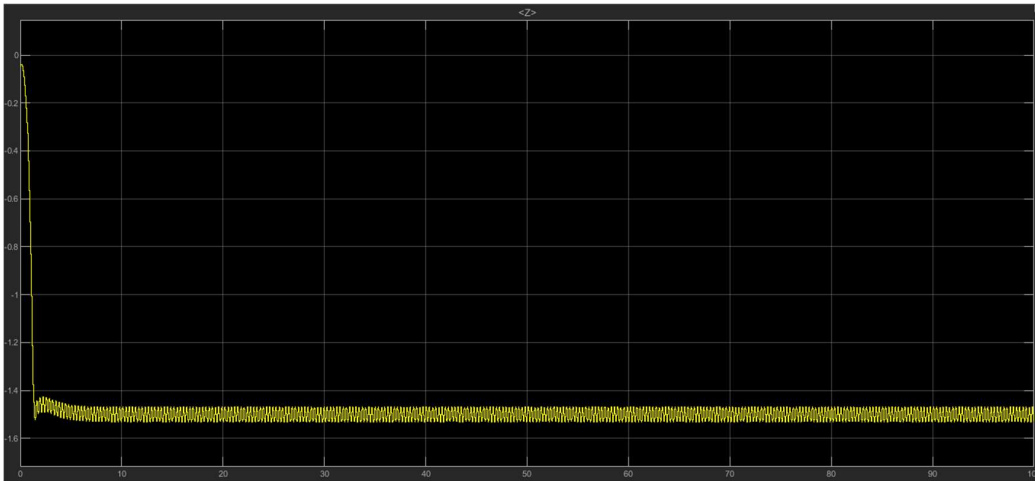
Figura 46

Successivamente abbiamo fatto delle simulazioni per verificare la risposta del sistema con il controllore G_2 in catena diretta.

Purtroppo, la simulazione non è andata come previsto. L'uscita presenta una sovraelongazione e delle oscillazioni piuttosto importanti che portano il drone a precipitare a terra. Questo comportamento è dato dalla spinta iniziale di *takeOff* che viene mandata ai motori in fase di decollo (vedi Fig.21). In default, questa spinta è proporzionale al peso del drone con un coefficiente di proporzionalità pari a $takeOffGain = 0.45$. Abbiamo quindi ritenuto opportuno modificare questo valore. È importante notare che è stato possibile attuare questa modifica perché questo valore non fa parte del modello matematico del drone ma, solo della logica di controllo. Abbiamo scelto un valore del coefficiente di $takeOffGain = 0.15$.

La simulazione, tramite questi cambiamenti, è andata a buon fine.

Mostriamo ora i grafici dell'uscita z (Fig.47 e 48)

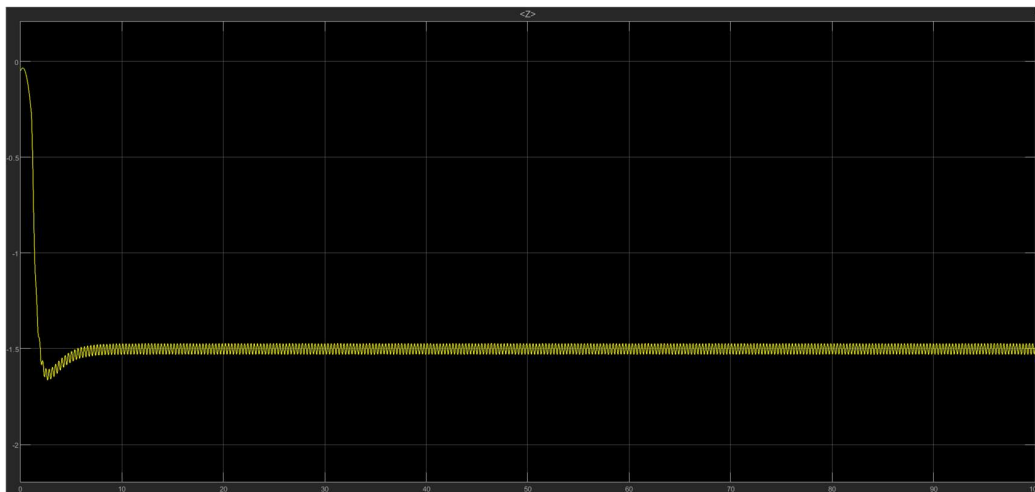


Dalla figura si nota che, rispetto all'uscita avuta tramite i PID integrati nel pacchetto, la sovralongazione è molto più piccola ma è presente una certa oscillazione attorno al valore di riferimento. Questo comportamento può essere attenuato cambiando il valore di guadagno del controllore. Infatti, un guadagno troppo elevato potrebbe generare un'amplificazione non trascurabile delle frequenze alte.

Abbiamo provato a diminuire il valore di K_G ponendolo pari a:

$$K_G = 0.45$$

Uscita del modello non lineare - Figura 47



Uscita del modello lineare - Figura 48

Così facendo è stato modificato il comportamento in frequenza e quindi sono variati i diagrammi di Bode. I nuovi diagrammi ottenuti sono visibili in Fig.49.

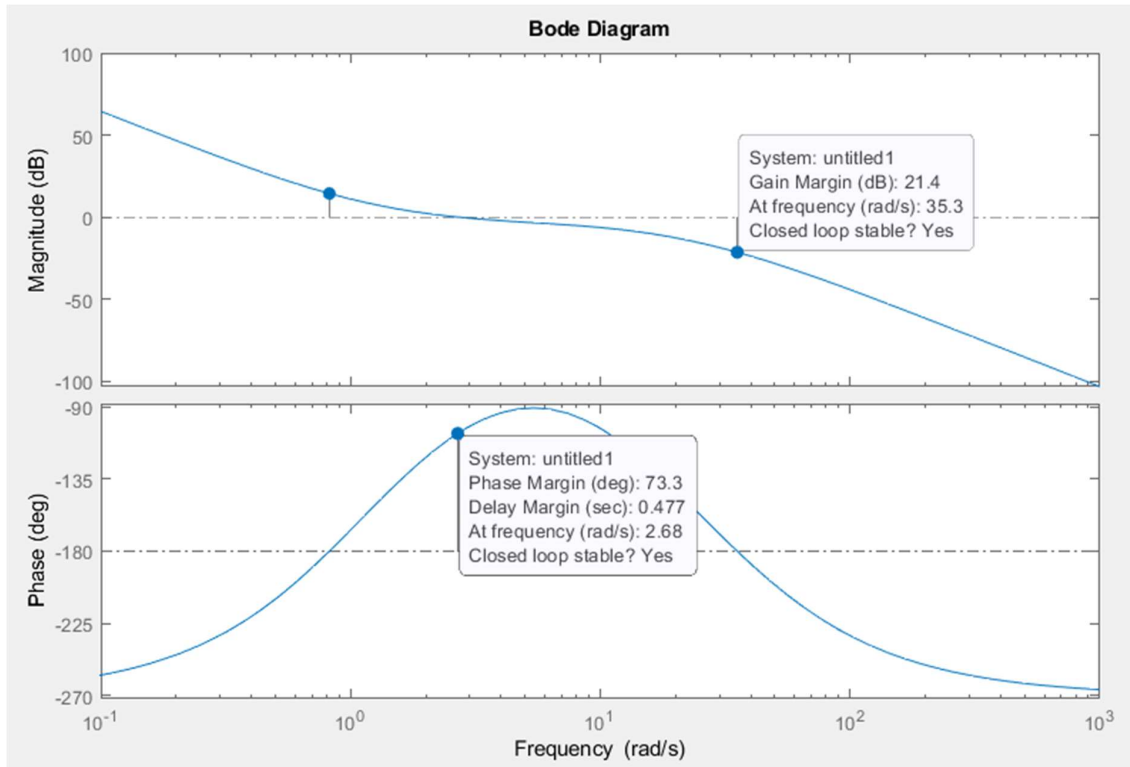
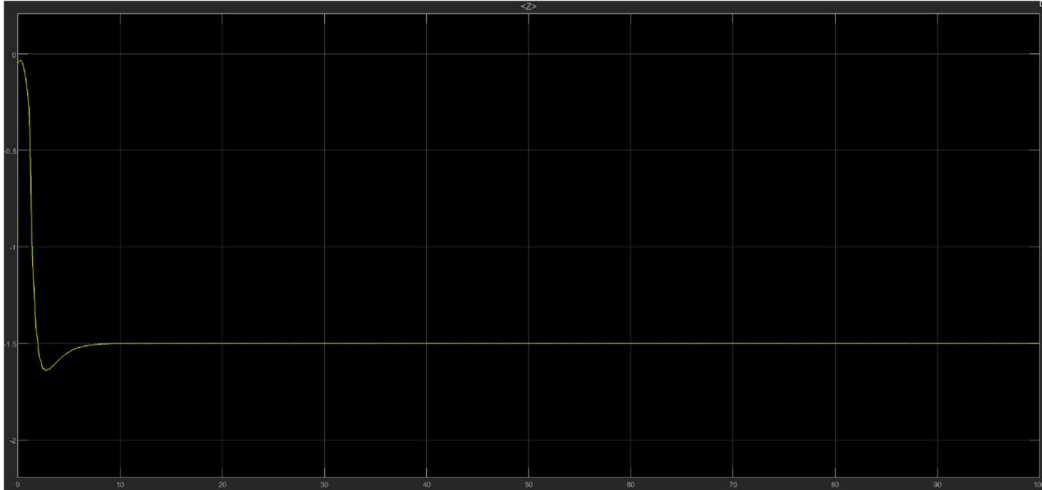


Figura 49

In sostanza, il cambiamento è dato da una traslazione del grafico del modulo verso valori più bassi. Questo ha consentito di aumentare il margine di guadagno positivo e diminuire il suo picco negativo. È stata anche spostata la posizione della pulsazione di attraversamento che ora è posta a $\omega_t = 2.68 \text{ rad/s}$ ma il margine di fase si è mantenuto stabile sul valore già ottenuto. In Fig.50 e Fig.51 mostriamo l'andamento dell'uscita rispetto al controllore ottenuto dopo questi ultimi accorgimenti. Si nota una forte attenuazione delle oscillazioni attorno al valore desiderato.



Uscita del modello non lineare con guadagno 0.45 - Figura 50



Uscita del modello lineare con guadagno 0.45 - Figura 51

6.3 Sintesi per tentativi del controllore senza astatismo ai disturbi

Progettiamo ora un controllore senza ottenere un astatismo completo ai disturbi agenti in catena diretta. L'idea è quella di aumentare il guadagno di controllore in modo tale da far diminuire il valore di regime dell'uscita rispetto al disturbo. Per capire questa scelta facciamo alcune considerazioni sul sistema in questione che schematizziamo in Fig.52.

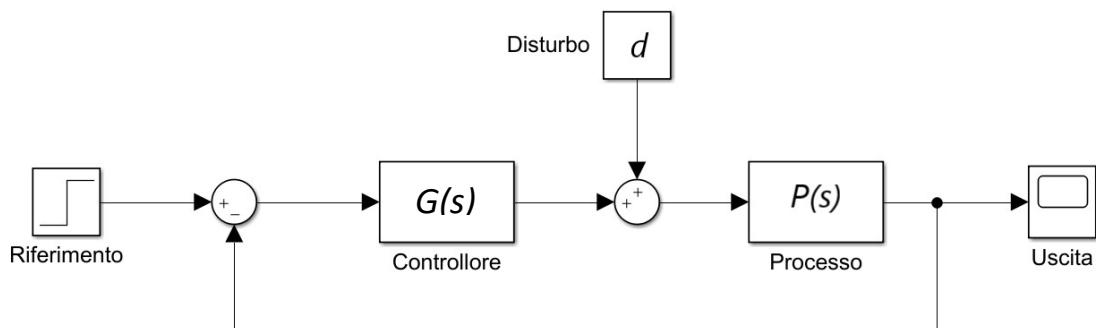


Figura 52

La funzione di trasferimento a ciclo chiuso del sistema disturbo-uscita è data da:

$$W_d = \frac{P(s)}{1 + G(s)P(s)} \quad (50)$$

Usiamo ora il *Teorema del valor finale* per calcolare il valore a regime dell'uscita rispetto al disturbo. Ricordiamo, per lo scopo, che la funzione $P(s)$ ha due poli in $s = 0$ e che la trasformata di Laplace del gradino equivale a $\frac{1}{s}$.

Otteniamo:

$$\widetilde{y}_d = \lim_{s \rightarrow 0} \left(s \frac{1}{s} \frac{P(s)}{1 + G(s)P(s)} \right) = \frac{1}{K_G} \quad (51)$$

In cui K_G è il guadagno statico di controllore. È quindi facile capire che, per attenuare il valore di \widetilde{y}_d bisogna aumentare il valore di K_G . Abbiamo deciso di porre un guadagno pari a:

$$K_G = 25$$

Imponendo quindi, in uscita, una risposta al disturbo pari a circa il 4% del valore del disturbo costante stesso. Poniamo successivamente la funzione di trasferimento del controllore di primo tentativo come semplicemente:

$$G_1(s) = K_G = 25$$

Notiamo l'assenza di poli in $s = 0$.

A questo controllore corrisponde una funzione di trasferimento a ciclo aperto F :

$$F_1(s) = \frac{91.95}{s^2}$$

I diagrammi di Bode ottenuti sono visibili in Fig.53

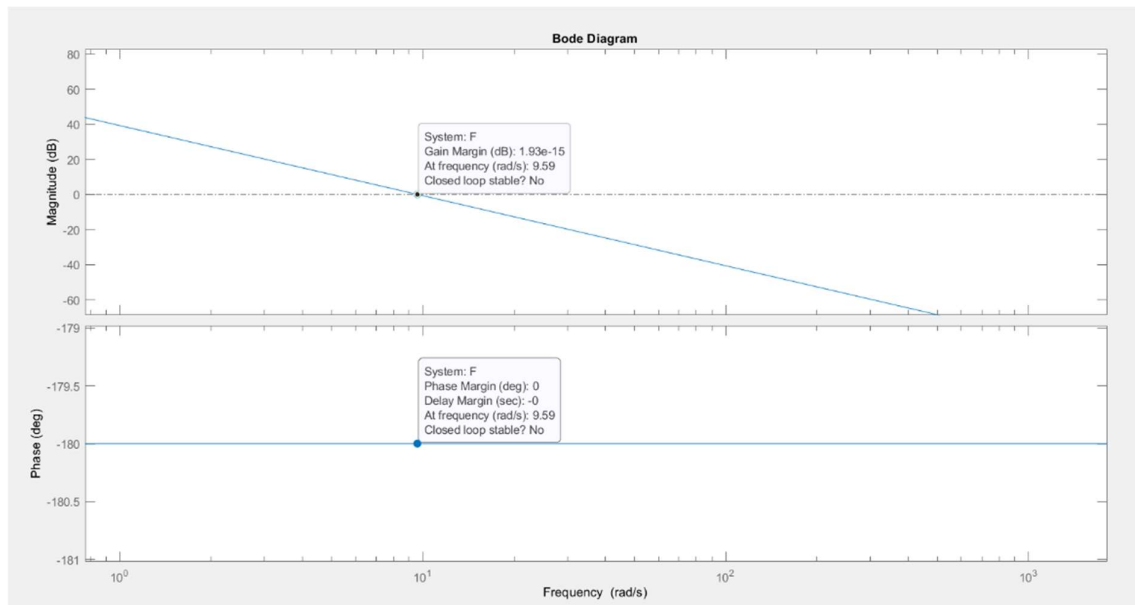


Figura 53

Dai grafici notiamo che, con una fase costante a -180° , non è possibile avere stabilità a ciclo chiuso dato che entrambi i margini (di fase e di guadagno) sono nulli. Dobbiamo progettare delle reti correttrici.

Procediamo ora con l'enunciazione delle specifiche desiderate del sistema. Anche in questo caso utilizziamo delle specifiche "standard", cioè specifiche spesso utilizzate in problemi di questo tipo.

Desideriamo dei parametri a ciclo chiuso del tipo:

- Modulo di risonanza $M_r \leq 2\text{dB}$
- Banda passante $B_3 = 5\text{ Hz}$

Che a ciclo aperto corrispondono a:

- Margine di fase $m_\varphi \geq 47^\circ$
- Pulsazione di attraversamento $\omega_t = 20\text{ rad/s}$

Notiamo che, rispetto al progetto precedente, è stata considerata una banda passante più alta. Questo è stato fatto a causa della forte natura passa-basso del sistema. Visualizzando la Fig.53 è facile notare come per frequenze più basse di $\omega_t = 20\text{ rad/s}$ sarebbe stato complicato progettare una rete correttrice, avendo in quel caso un modulo positivo e una fase negativa. Questo piccolo accorgimento è stato quindi fatto per evitare di impelagarsi nella progettazione di complicate reti a sella. Nel nostro caso, nel punto $\omega = 20\text{ rad/s}$, basterà la sintesi di una sola rete anticipatrice. Infatti, abbiamo bisogno di aumentare il modulo di circa 12.7 dB e la fase di almeno 47° . Utilizzando ancora una volta i diagrammi universali in Fig.44 abbiamo scelto dei parametri di rete pari a $\tau_a = 0.25$ e $m_a = 16$.

Otteniamo:

$$R_a(s) = \frac{1 + 0.25s}{1 + 0.0156s}$$

Cioè un regolatore con funzione di trasferimento:

$$G_2(s) = 25 * R_a(s) = \frac{100s + 400}{0.25s + 16}$$

Verifichiamo le specifiche tramite i diagrammi di Bode (Fig.54)

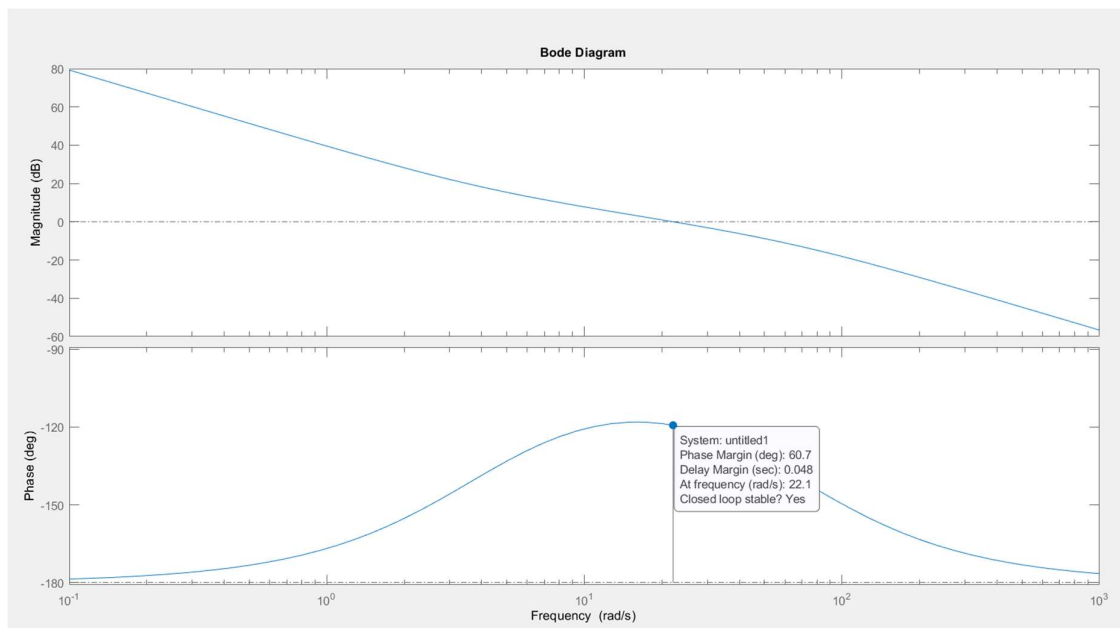


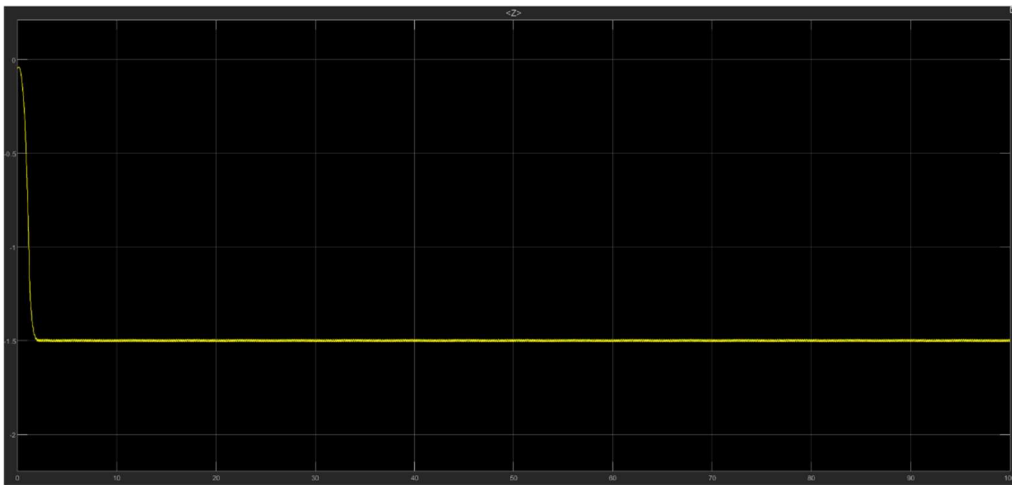
Figura 54

Dalla figura si può notare che il margine di fase è stato migliorato di molto, andando ad ottenere un valore di circa $m_\phi \cong 60^\circ$ nella frequenza $\omega = 22.1 \text{ rad/s}$. Tutte le specifiche sono state rispettate.

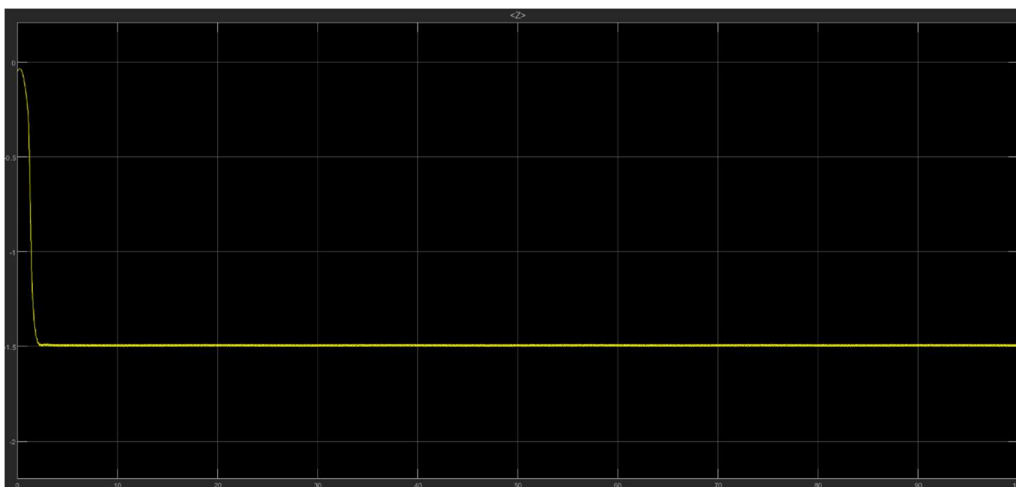
Controlliamo ora il comportamento dell'uscita verificando il tutto tramite delle simulazioni (Fig.55 e Fig.56).

Il sistema di controllo funziona egregiamente, il sistema è stabile e l'uscita insegue il riferimento con un tempo di risposta elevato.

È importante notare che, nella risposta del sistema lineare, l'uscita non corrisponde esattamente al riferimento ma è presente un piccolo errore trascurabile dell'ordine del millimetro. Questo errore non è altro che la risposta del sistema disturbo-uscita, attenuata dal controllore, che si sovrappone all'uscita desiderata. Inoltre, è presente anche una leggera oscillazione attorno al valore di riferimento, questa però può essere considerata trascurabile essendo anch'essa di ampiezza dell'ordine del millimetro.



Uscita del modello non lineare - Figura 55



Uscita del modello lineare - Figura 56

7 Conclusioni

Nel capitolo precedente abbiamo visto due approcci diversi alla progettazione del sistema di controllo. In entrambi casi il sistema si è comportato molto bene con una marcata stabilità e la presenza di oscillazioni trascurabili attorno al valore di riferimento.

Se andiamo però ad osservare nel dettaglio i due controllori realizzati possiamo notare alcune differenze.

Come abbiamo già detto, il controllore senza astatismo, per sua natura progettuale, non può rendere nulla la risposta ai disturbi. L'uscita presenta quindi un *bias* sovrapposto di pochi millimetri dipendente dal guadagno del controllore. Inoltre, dai grafici abbiamo notato che nelle uscite del controllore senza astatismo è presente una piccola oscillazione (Fig.57) non attenuabile se non peggiorando la risposta al disturbo. Si potrebbe infatti diminuire il guadagno del controllore ma, ovviamente, il bias in uscita sarà più ampio.

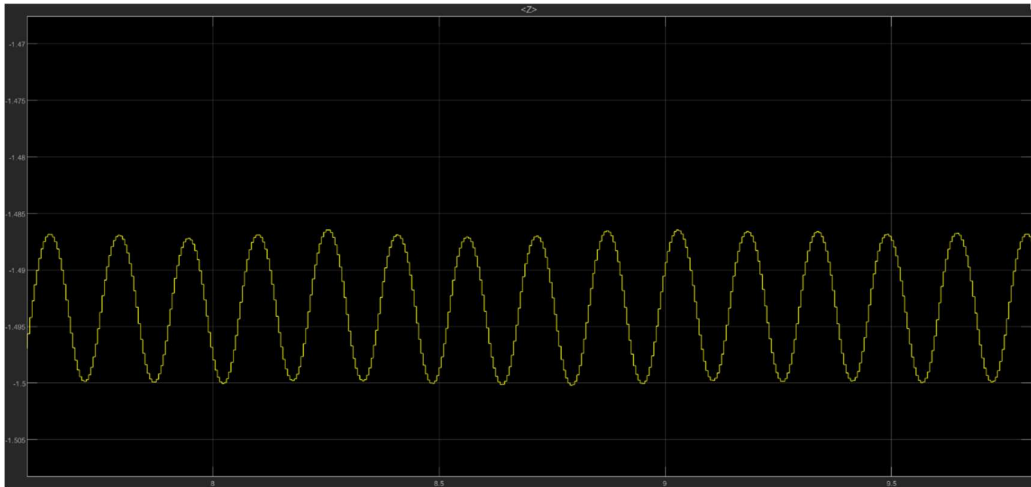


Figura 57

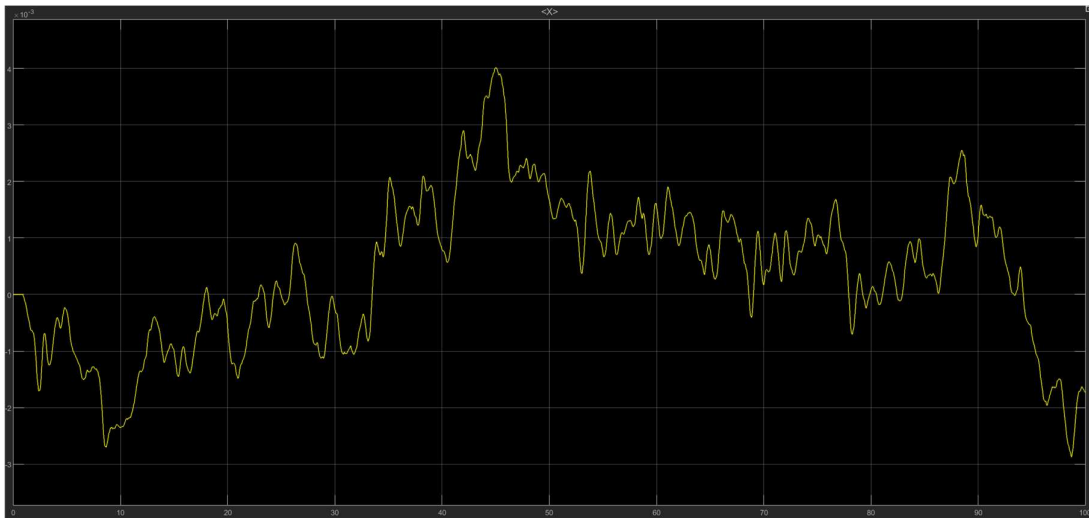
L'oscillazione si presenta comunque trascurabile, con ampiezza di qualche millimetro.

Diversa è la situazione per quanto riguarda il controllore con astatismo ai disturbi. Questo regolatore, nonostante la sua complessità più elevata, non riesce ad ottenere risultati migliori dell'altro. L'uscita non possiede oscillazioni importanti ma, per quanto riguarda il modello lineare, presenta delle sovraelongazioni non trascurabili. Infatti, il picco in uscita è di circa il 9.5% rispetto al valore di regime desiderato. Il regolatore senza poli in $s = 0$, invece, non possiede sovraelongazioni degne di nota e la risposta si assesta al valore di regime immediatamente.

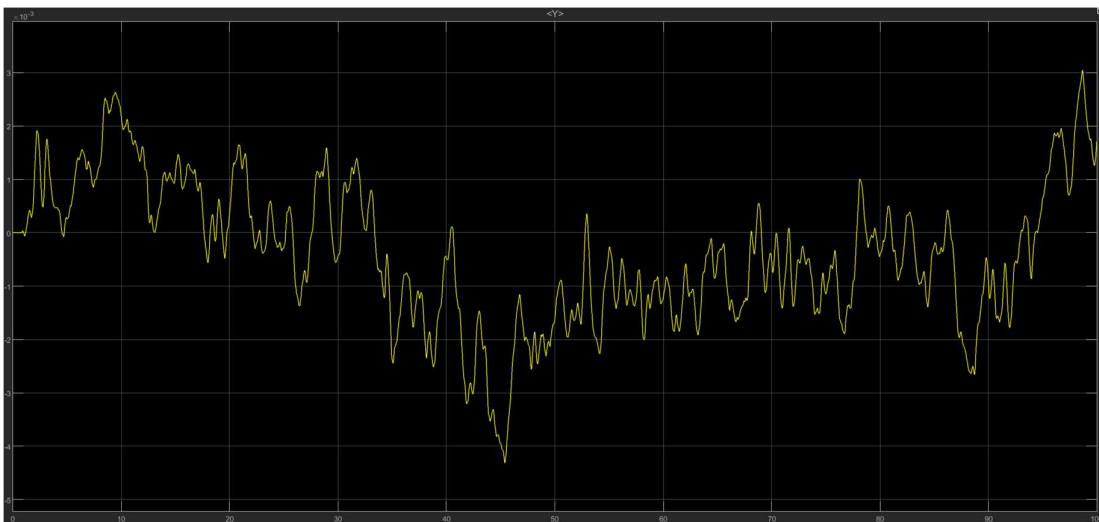
Il sistema di controllo senza astatismo, inoltre, possiede una banda passante più alta e ha un tempo di risposta considerevolmente più breve (circa 2.5 secondi contro i quasi 9 secondi del sistema con astatismo). È anche importante considerare che il regolatore senza astatismo risulta molto meno complesso e questo è un vantaggio dal punto di vista computazionale nell'implementazione algoritmica a tempo discreto.

Presentati i vari vantaggi e le differenze fra i vari controllori progettati, è stato infine scelto il sistema di controllo senza astatismo come regolatore definitivo per l'assetto della quota del drone.

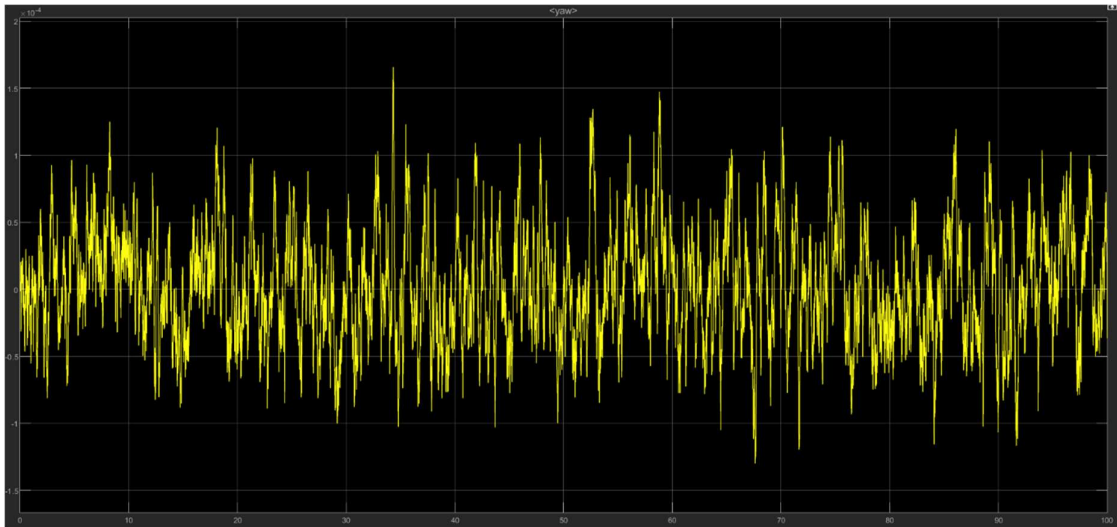
Concludiamo presentando l'andamento dei vari gradi di libertà ottenuto con il controllore scelto nel modello non lineare.



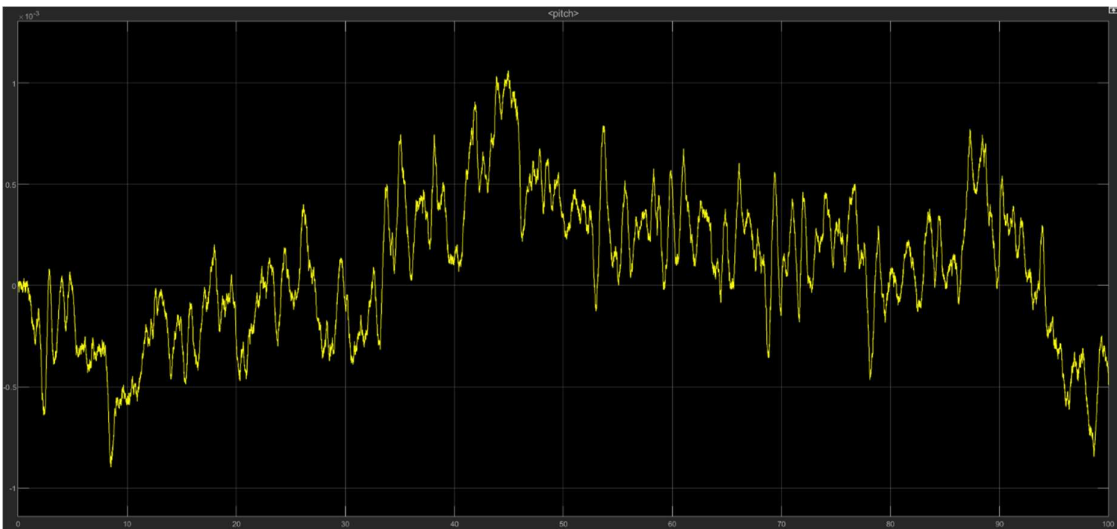
Coordinata X - Figura 58



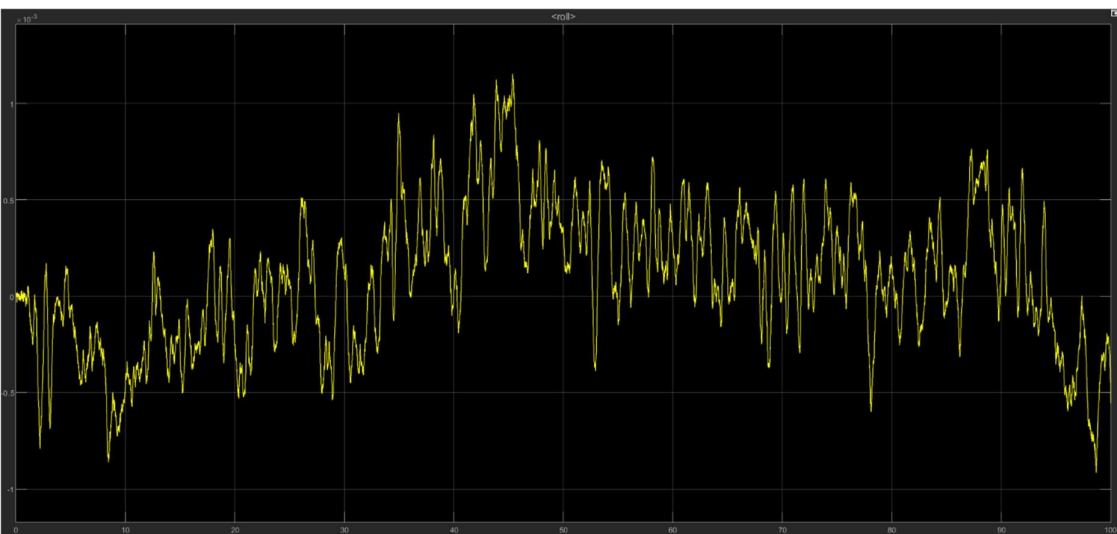
Coordinata Y - Figura 59



Angolo di imbardata - Figura 60



Angolo di beccheggio - Figura 61



Angolo di rollio - Figura 62

8 Riferimenti bibliografici

- [1] Musa, (2018), *Techniques for Quadcopter Modelling & Design: A Review*.
- [2] Pounds, Mahony, Corke (2006), *Modelling and control of a quad-copter robot*.
- [3] Leishman, (2006), *Principles of Helicopter Aerodynamics*.
- [4] Prouty RW. (2002), *Helicopter Performance, Stability and Control*.
- [5] Pantalone, (2015), *Modellazione e simulazione di un quadricottero Multirotores*.
- [6] Corradini, Orlando, (2002), *Fondamenti di Automatica*, Pitagora Editrice.
- [7] Mezzanotti (2019), *Studio e sviluppo di controllori per l'assetto di mini-droni*.
- [8] Gambini (2019), *Studio e sviluppo in Simulink di tecniche di controllo per mini-droni*.