



Università Politecnica delle Marche

FACOLTÀ DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria Informatica e dell'Automazione

TESI DI LAUREA TRIENNALE

**Study and development of models and control
systems for drones**

**Studio e sviluppo di modelli e sistemi di controllo
per droni**

Candidato:

Mattia Del Moro

Matricola 1073238

Relatore:

Gianluca Ippoliti

Correlatore:

Giuseppe Orlando

Contents

1	Introduzione	5
2	Modello del quadricottero	7
2.1	Assunzioni generali	7
2.2	Caratteristiche costruttive Mambo	7
2.3	Sistemi di riferimento	8
2.4	Modello fisico	9
2.5	Modello matematico	11
2.6	Forza di gravità e forza di attrito viscoso	14
2.7	Forze e momenti dei motori	15
2.8	Equazioni del sistema	17
3	Simulink	18
3.1	Non-linear Airframe	20
3.2	Flight Control System (FCS)	24
3.3	Command	33
3.4	Environment	35
3.5	Linear Airframe	36
4	Sintesi controllore	39
4.1	Sintesi in frequenza	42
4.1.1	Riferimento costante	44
4.1.2	Riferimento variabile	49
4.2	Sintesi con luogo delle radici	54
4.2.1	Riferimento costante	55
4.2.2	Riferimento variabile	60
5	Conclusioni	65

ABSTRACTS

Nel presente studio verranno analizzate in primo luogo le leggi fisiche e matematiche che determinano la dinamica di volo di un drone quadricotore. Successivamente verrà effettuata l'analisi dello schema a blocchi del drone, nello specifico sarà preso come riferimento il modello messo a disposizione direttamente in Matlab dal pacchetto 'Simulink Support Package for Parrot Minidrones'.

Una volta effettuata la parte di analisi verranno ricavate le matrici A, B, C, D del sistema complessivo, ed in particolare, tramite un disaccoppiamento del controllo degli angoli di pitch e roll, è stata isolata la parte di volo relativa all'angolo di rollio.

Lo scopo di questo lavoro è infatti quello di sintetizzare un controllore di tale angolo, più performante rispetto a quello implementato di base.

Sono state utilizzate due tecniche di controllo, ovvero la sintesi in frequenza e la sintesi con il luogo delle radici: entrambe le tecniche sono state applicate prima per trovare prima un controllore per l'inseguimento di segnali di ingresso costanti e successivamente per l'inseguimento di segnali di ingresso variabili nel tempo.

Infine si è effettuato un confronto ed una valutazione dei risultati ottenuti con le due tecniche, confrontando tra loro i tracciati dell'angolo di roll ottenuti dalle varie simulazioni di volo.

1 Introduzione

“Un aeromobile a pilotaggio remoto (RPA), noto comunemente come drone, è un apparecchio volante caratterizzato dall’assenza del pilota a bordo. Il suo volo è controllato dal computer a bordo del mezzo aereo oppure tramite il controllo remoto di un navigatore o pilota, sul terreno o in altre posizioni.” [1]



Figure 1: Modello di drone militare.

Nati inizialmente per scopi bellici, negli ultimi anni hanno suscitato sempre più interesse e grazie anche al progredire della tecnologia che ha permesso di ridurre dimensioni e costo, al giorno d’oggi vengono utilizzati in svariati ambiti (soccorso, trasporto, riprese, ludico). Vi è inoltre la possibilità di installare su di essi sensori, strumenti, sistemi di rilevazione e comunicazione specifici per lo scopo di utilizzo.

Vista la crescente commercializzazione e la maggior fruibilità del mezzo, l’ENAC (Ente Nazionale Aviazione Civile) ha deciso di regolamentare i droni in base alle caratteristiche e alle prestazioni globali: peso, raggio d’azione e quota sono soltanto alcuni dei parametri presi in considerazione.

Con l’acronimo APR (o UAV) si fa riferimento ad un’ampia gamma di aeromobili che presentano caratteristiche e tecnologie differenti: all’interno di essa si possono catalogare i vari velivoli per forma, dimensioni, peso e numero di motori (le varie combinazioni di queste caratteristiche vanno ad influire sull’aerodinamicità, la manovrabilità, l’autonomia di volo e il carico massimo trasportabile dal velivolo).

Una specifica classe di droni è costituita dai veicoli a pale rotanti (figura 2) come quadricotteri, esacotteri, octacotteri ecc. Questi particolari droni, che tra loro si differenziano per il numero di motori, si prestano grazie alla loro maneggevolezza (in particolare alla possibilità di mantenersi in condizioni stazionarie, ovvero in hovering), alle misure ridotte, al costante miglioramento dell’autonomia di volo e ad applicazioni sempre più estese.

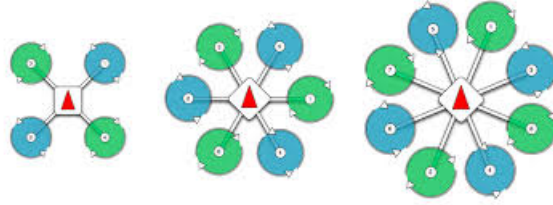
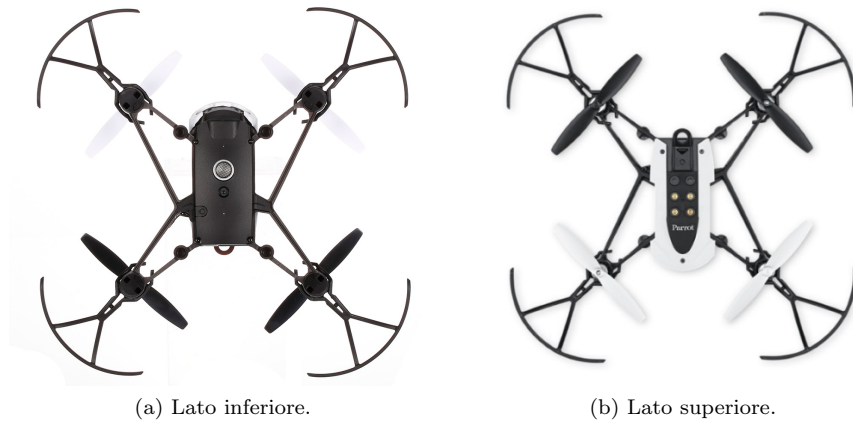


Figure 2: Esempio droni a pale rotanti.

In questo progetto si è affrontato lo studio e parte del controllo del modello matematico e fisico che vi è dietro la dinamica di un quadricottero. In particolare si è fatto riferimento ad un esempio realizzato su Simulink di Matlab (scaricabile dal link: <https://it.mathworks.com/hardware-support/parrot-minidrones.html>), il quale va a descrivere la dinamica del mini drone Parrot Mambo dell'azienda francese Parrot S.A.



(a) Lato inferiore.

(b) Lato superiore.

Figure 3: Mini drone Mambo.

2 Modello del quadricottero

In questa sezione si analizzerà il modello del quadricottero Mambo che viene riportato in Figura 3. Verrà considerato prima il modello fisico e poi quello matematico. Successivamente si darà una rapida descrizione delle forze e dei momenti che delineano il moto del drone per arrivare a definire le sei equazioni che compongono il sistema.

2.1 Assunzioni generali

Il processo di modellazione è basato su una serie di assunzioni che rendono più semplice la trattazione, permettendo di ottenere un modello semplificato ma comunque fedele alla realtà ed implementabile a livello software.

1. Il sistema è tempo invariante (massa e struttura non variano nel tempo);
2. La struttura del drone è rigida (possibilità di trascurare i contributi dovuti alla flessione dell'intelaiatura);
3. La struttura è simmetrica (possibilità di valutare una matrice di inerzia diagonale);
4. I quattro motori sono identici, quindi analizzarne uno singolarmente non fa perdere di generalità al modello complessivo;
5. Il centro di gravità (CoG: Centre of Gravity) e l'origine del sistema di riferimento solidale al veicolo (Body Fixed Frame) coincidono;
6. La spinta (thrust) e la resistenza aerodinamica (drag) vengono considerate proporzionali al quadrato delle velocità angolari delle eliche (w^2).

2.2 Caratteristiche costruttive Mambo

Il minidrone Mambo è un quadricottero, ovvero un aeromobile dotato di quattro eliche rotanti azionate da motori coreless DC brushless. Nella normale configurazione (a croce) due di essi (anteriore destro e posteriore sinistro) ruotano in senso orario, mentre gli altri due (anteriore sinistro e posteriore destro) ruotano in senso anti-orario.

Una rapida descrizione delle caratteristiche costruttive del drone oggetto di studio:

- Sensori di stabilizzazione: Unità di misura inerziale per valutare la velocità, l'inclinazione e il contatto con gli ostacoli (accelerometro a 3 assi e giroscopio a 3 assi), Sensori a ultrasuoni, Sensore di pressione, Sensore telecamera.
- Peso: 63g (senza paraurti o accessori).
- Dimensioni: 18 x 18 cm con paraurti.

- Energia: Batteria LiPo 660 mAh con 8 minuti di autonomia senza accessori o paraurti.

2.3 Sistemi di riferimento

Prima di procedere alla formulazione delle equazioni è necessario determinare tre sistemi di riferimento (figura 4) che saranno poi utilizzati per descrivere la dinamica del sistema, capace di inglobare sia movimenti lineari che movimenti angolari.

- Earth Fixed Frame (x_i, y_i, z_i): sistema di riferimento inerziale, scelto in riferimento al piano tangente alla superficie della Terra con origine nella posizione dell'osservatore;
- Vehicle Frame (x_v, y_v, z_v): allineato al sistema di riferimento inerziale con origine nel centro di gravità del velivolo, anche durante movimenti rotatori. Il movimento traslatorio del quadricottero rispetto al sistema inerziale è descritto dalle grandezze di questo SdR¹;
- Body Fixed Frame (x_b, y_b, z_b): solidale al drone con origine il centro di gravità di esso. Questo sistema di riferimento ruota, insieme al drone, rispetto al Vehicle Frame e quindi le grandezze misurate descrivono il movimento rotatorio del velivolo.

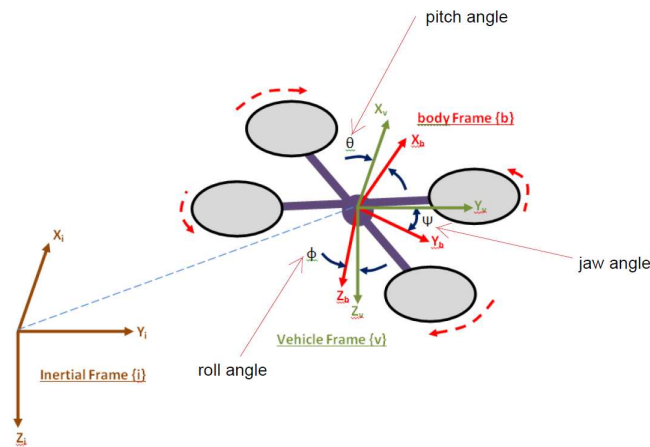


Figure 4: I tre sistemi di riferimento utilizzati.

¹SdR: sistema di riferimento.

2.4 Modello fisico

Per comprendere a pieno il principio generale che sta dietro al funzionamento di un quadricottero, si inizia con lo studiare la struttura di quest'ultimo.

Esso è costituito da un corpo centrale e da quattro bracci, ognuno dei quali termina con un motore brushless che mette in rotazione un'elica. Le eliche opposte, rispetto al braccio, formano una coppia e di conseguenza ne avremo un totale di due: una coppia gira in senso orario e una in senso antiorario.

Un quadricottero, in termini più ampi può essere considerato come un comune sistema sotto-attuato (under-actuated system), poichè dotato di solo 4 motori e ben 6 gradi di libertà (Degrees of Freedom, o DoF): tre gradi di libertà traslazionale, i quali forniscono lo spostamento lineare (assi X, Y, Z) e tre gradi di libertà rotazionale, che forniscono lo spostamento angolare (angoli ϕ , θ , ψ).

I movimenti del drone (figura 5) sono ottenuti variando opportunamente le velocità dei motori e dato che il sistema è sotto-attuato, possiamo controllarne direttamente solo alcuni. Essi sono:

- **ROLL** (Rollio, legato all'angolo ϕ): è il movimento che può far oscillare il velivolo in direzione longitudinale (rotazione attorno l'asse X e spostamento lungo l'asse Y). E' ottenuto lasciando invariata la velocità dei motori anteriore e posteriore, aumentando (diminuendo) di una certa quantità la velocità del motore di destra, e diminuendo (aumentando) la velocità del motore di sinistra in modo che la spinta verso l'alto non cambi. La relazione tra la spinta e la velocità dei rotori ha un andamento quadratico, quindi per piccole variazioni dell'angolo ϕ , le variazioni di velocità dei rotori saranno circa uguali.
- **PITCH** (Beccheggio, legato all'angolo θ): è molto simile al rollio, infatti se quest'ultimo fa oscillare il velivolo longitudinalmente, il beccheggio lo fa oscillare avanti e indietro (rotazione attorno l'asse Y e spostamento lungo l'asse X). Come è facilmente intuibile, per ottenere questo movimento l'idea è la stessa del rollio, ma si usano i motori anteriore e posteriore. Si fanno quindi variare le velocità di tali rotori di una quantità tale da mantenere inalterata la spinta verticale. Anche in questo caso, per piccole variazioni dell'angolo θ , le variazioni di velocità dei rotori sono più o meno coincidenti.
- **YAW** (Imbardata, legato all'angolo ψ): è il movimento che fa ruotare il velivolo su se stesso (rotazione attorno all'asse Z) ed è forse il più complesso dei quattro. Per ottenere questo si aumenta la velocità dei rotori di due motori opposti, e si diminuisce la velocità degli altri due. In questo modo si avranno, per esempio, due rotori che girano velocemente in senso antiorario, e due che girano più lentamente in senso orario. Nel complesso ci sarà dunque un momento che farà girare il drone intorno all'asse verticale in senso orario. Se le velocità sono opportunamente scelte, non ci sarà un cambiamento della spinta verso l'alto. Come nei casi precedenti, per piccole variazioni dell'angolo ψ , le variazioni di velocità dei rotori sono simili.

- THRUST (Spinta verticale): è il movimento più semplice e come già accennato prima, la spinta verso l'alto è legata al quadrato della velocità angolare dei rotori (w^2) da una costante che dipende dall'aria e da come questi sono realizzati. Per aumentare la spinta quindi, si aumenta la velocità di una determinata quantità, uguale per tutti e quattro i motori, in modo da non creare differenze che possano avere conseguenze sulle altre dinamiche di movimento (infatti, ad esempio, se si avesse una potenza diversa in uno o più rotori, la risultante spinta non sarà più rivolta solo verso l'alto, ma entrerebbero in gioco anche le dinamiche descritte nei movimenti precedenti).

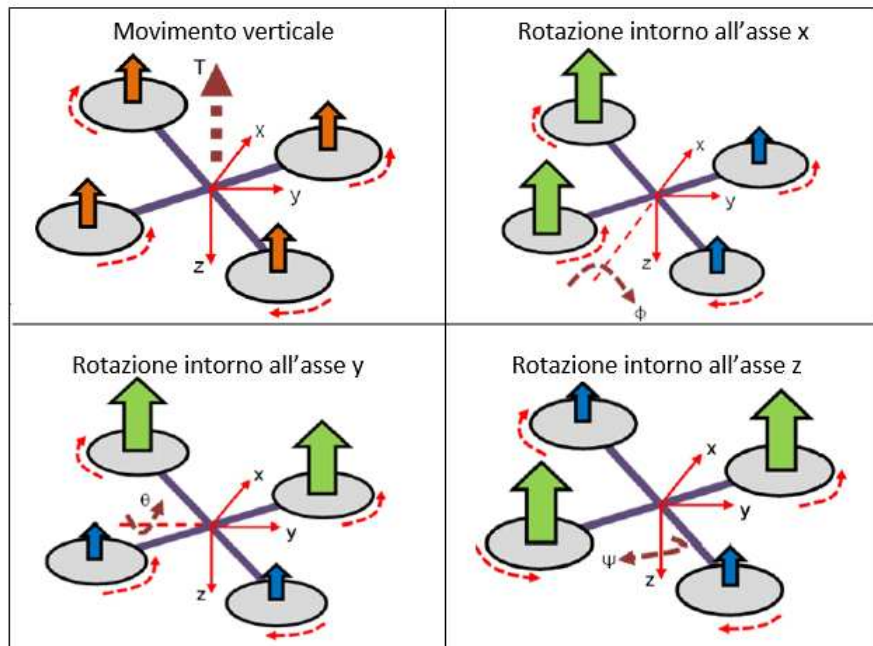


Figure 5: Movimenti controllabili direttamente.

La condizione di volo a punto fisso (hovering), è possibile considerarla un caso particolare della manovra di salita in cui ciascun motore ha tiro pari ad un quarto del peso del drone raggiungendo così il bilanciamento delle forze spinta e peso.

I movimenti non controllabili direttamente dovranno essere generati da una azione combinata tra quelli appena elencati, in particolare i movimenti in direzione longitudinale (spostamento in avanti/indietro) e in direzione laterale (spostamento a sinistra/destra) che saranno ottenuti inclinando il drone nella direzione desiderata.

2.5 Modello matematico

Una volta conosciuta la posizione del velivolo, è possibile ottenere direttamente la misura del suo spostamento lineare rispetto al sistema di riferimento inerziale. Ciò non vale nel caso dello spostamento angolare rispetto al sistema di riferimento inerziale. È necessario fare riferimento a delle "matrici di rotazione" quadrate.

Fissato un asse di rotazione, queste matrici descrivono gli spostamenti angolari ottenibili effettuando rotazioni rispetto agli altri due assi "liberi". Gli angoli ϕ , θ e ψ vengono chiamati angoli di Eulero e descrivono la rotazione del sistema di riferimento solidale al corpo rigido rispetto a quello inerziale.

Possono essere visti ulteriormente come una rappresentazione matematica di tre rotazioni sugli assi di riferimento. In questo caso gli assi sono orientati in modo tale che la direzione delle x positive sia il fronte del drone, le y positive sia la parte destra, mentre l'asse delle z è verso il basso.

Per eseguire il passaggio dal sistema di riferimento solidale con il drone (Body Fixed Frame) a quello inerziale (Earth Fixed Frame), e viceversa, intervengono le matrici di rotazione (nel nostro caso sono 3), corrispondente ciascuna ad un angolo ben preciso (ϕ , θ , ψ), e quindi ad una manovra di roll, pitch o yaw.

Se si considera un sistema orientato con la regola della mano destra, è ammissibile descrivere le tre rotazioni come:

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

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

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

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

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

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

Da notare che le tre matrici di rotazione ottenute sono ortonormali, ovvero $R^T = R^{-1}$. La matrice di rotazione complessiva è data dal prodotto classico (riga per colonna) delle tre singole matrici di rotazioni:

$$R(\phi, \theta, \psi) = R(x, \phi)R(y, \theta)R(z, \psi)$$

da cui si ottiene una matrice totale di rotazione in R^3 , nota anche come Direction Cosine Matrix (DCM) (eq. 4):

$$R_b^i = \begin{bmatrix} \cos\theta\cos\phi & \cos\theta\sin\phi & -\sin\theta \\ -\cos\phi\cos\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix} \quad (4)$$

Introducendo il vettore $b=[b_a, b_b, b_c]^T$ delle coordinate rispetto al Body Fixed Frame ed il vettore $r=[x, y, z]^T$ delle coordinate rispetto al sistema inerziale, in maniera compatta si ottiene:

$$b = \begin{bmatrix} b_a \\ b_b \\ b_c \end{bmatrix} = R_b^i * \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_b^i * r \quad (5)$$

Per la trasformazione inversa, ovvero per il passaggio da coordinate espresse rispetto al sistema di riferimento solidale con il drone a quello inerziale, sfruttando l'ortonormalità si applica la formula inversa:

$$r = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = (R_b^i)^T * \begin{bmatrix} b_a \\ b_b \\ b_c \end{bmatrix} = R_b^i * b \quad (6)$$

Si arriva così a definire la matrice rotazionale per il passaggio di coordinate tra i due sistemi di riferimento (eq. 7):

$$R_b^i = \begin{bmatrix} \cos\theta\cos\phi & -\cos\phi\cos\psi + \text{sen}\phi\text{sen}\theta\cos\psi & \text{sen}\phi\text{sen}\psi + \cos\phi\text{sen}\theta\cos\psi \\ \cos\theta\text{sen}\psi & \cos\phi\cos\psi + \text{sen}\phi\text{sen}\theta\text{sen}\psi & -\text{sen}\phi\cos\psi + \cos\phi\text{sen}\theta\text{sen}\psi \\ -\text{sen}\theta & \text{sen}\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (7)$$

Ottenute queste matrici si può effettuare il calcolo delle grandezze utili per lo studio del sistema, partendo dalla velocità. Si definiscono i seguenti vettori:

$$v_b = [u, v, w]^T \quad (8)$$

ha come componenti le velocità traslazionali lungo gli assi del SdR solidale al drone.

$$w_b = [p, q, r]^T \quad (9)$$

ha come componeti le velocità angolari del SdR solidale al drone.

Le seguenti trasformazioni (10) (11) effettuano, rispettivamente, il passaggio delle componenti appena descritte (8) (9) alle relative componenti che descrivono la rotazione del sistema rispetto al sistema di riferimento inerziale:

$$v_i = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_b^i * \begin{bmatrix} u \\ v \\ w \end{bmatrix} = R_b^i * v_b \quad (10)$$

$$w_i = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \text{sen}\phi\frac{\text{sen}\theta}{\cos\theta} & \cos\phi\frac{\text{sen}\theta}{\cos\theta} \\ 0 & \cos\phi & -\text{sen}\phi \\ 0 & \frac{\text{sen}\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = Q_b^i * \begin{bmatrix} p \\ q \\ r \end{bmatrix} = Q_b^i * w_b \quad (11)$$

Le variazioni delle velocità angolari (w_b) rispetto al sistema di riferimento solidale al quadricottero, sono ottenute definendo la matrice Q_i^b e successivamente applicando la trasformazione inversa (12):

$$w_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\text{sen}\theta \\ 0 & \text{cos}\phi & \text{sen}\phi\text{cos}\theta \\ 0 & -\text{sen}\phi & \text{cos}\phi\text{cos}\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = Q_i^b * \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = Q_i^b * w_i \quad (12)$$

Osservando quest'ultima formula si nota subito che la variazione dell'angolo ψ dipende esclusivamente dalla variazione degli angoli ϕ e θ ; ciò determina che le manovre di pitch o roll potrebbero causare la rotazione del drone lungo l'asse verticale Z, ovvero una manovra di imbardata.

2.6 Forza di gravità e forza di attrito viscoso

Oltre alle dinamiche descritte nei paragrafi precedenti, si deve tenere in considerazione altri due tipi di dinamiche, dovute alle forze fisiche a cui è sottoposto il drone:

1. **FORZA DI GRAVITA'**: dipende dalla massa del corpo ed agisce sempre perpendicolarmente verso il centro della Terra, tuttavia quando il drone presenta un angolo di pitch o roll, l'asse Z del sistema solidale al drone non coincide con l'asse Z del sistema inerziale. Le componenti della forza peso vanno quindi moltiplicate per la matrice di rotazione Q_b^i che permette il passaggio tra i due SdR. La forza di gravità agisce sempre, a prescindere dalle condizioni di moto.

$$\vec{F}_g^b = Q_i^b \vec{F}_g^i = \begin{bmatrix} 1 & 0 & -\text{sen}\theta \\ 0 & \text{cos}\phi & \text{sen}\phi\text{cos}\theta \\ 0 & -\text{sen}\phi & \text{cos}\phi\text{cos}\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = \begin{bmatrix} -mg\text{sen}\theta \\ mg\text{cos}\theta\text{sen}\phi \\ mg\text{cos}\theta\text{cos}\phi \end{bmatrix} \quad (13)$$

2. **FORZA DI ATTRITO VISCOSO (DRAG)**: quando il drone non è in condizioni stazionarie, essendo immerso nell'aria esso è soggetto ad una forza di attrito viscoso la quale si oppone al moto. Tale forza dipende dal quadrato della velocità del corpo, dal mezzo in cui è immerso ed ha stessa direzione ma verso opposto allo spostamento.

$$\vec{F}_D = \frac{1}{2} \rho V_b^2 \vec{v}_b \quad (14)$$

Una doverosa considerazione è che la forza di gravità agisce sempre, ma la stessa cosa non vale per la forza di attrito viscoso che agisce solo in caso di moto del drone.

2.7 Forze e momenti dei motori

I rotori sono individuabili nello spazio attraverso i seguenti vettori, dove NSEW (North, South, East, West) corrisponde alla direzione rispetto al Nord magnetico:

$$dN=(0 \text{ d h});$$

$$dS=(0 \text{ -d h});$$

$$dE=(d \text{ 0 h});$$

$$dW=(-d \text{ 0 h}).$$

La forza di spinta dell'i-esimo motore vale:

$$t_i = C_T \rho A r^2 \omega_i^2 \begin{bmatrix} -sena_i \\ cosa_i senb_i \\ -cosb_i cosa_i \end{bmatrix} \quad (15)$$

Oltre al coefficiente di spinta (C_T), sono presenti i parametri costruttivi dell'elica (area A, raggio r) e la densità del fluido in cui sarà immerso (ρ). I coefficienti a_i e b_i indicano rispettivamente gli angoli formati dagli assi longitudinale e latitudinale del veicolo rispetto alla direzione di avanzamento dello stesso. Come già accennato, un grande contributo viene impresso dal quadrato della velocità di rotazione delle eliche.

La coppia prodotta dall'i-esimo motore vale:

$$q_i = C_Q \rho A r^3 \omega_i |\omega_i| e_3 \quad (16)$$

Anche qui vengono tenuti in considerazione i parametri costruttivi dell'elica e la densità del fluido in cui è immerso il drone. Si nota come, a differenza della spinta, la coppia ha componenti solo sull'asse verticale Z. Questo deriva dalla moltiplicazione per il vettore $e_3 = [001]^T$ (ovvero dal considerare la forza di gravità). Viene utilizzato il valore assoluto della velocità angolare perché moltiplicato per la stessa, preserva il segno, il quale determina il verso di rotazione.

Il momento della spinta è dato dal prodotto vettoriale tra la forza e la distanza dei rotori:

$$m_i = t_i \times d_i \quad (17)$$

Dalla seconda Legge di Newton per i corpi rigidi, si ha che l'accelerazione del corpo vale:

$$m\dot{v} = -m\omega \times v + mgR^T e_3 + \sum_{N,S,E,W} t_i \quad (18)$$

Il primo termine è pari al prodotto vettoriale tra la velocità angolare e la velocità lineare sulla massa;

il secondo termine rappresenta il contributo sul solo asse verticale (ecco perché vi è la moltiplicazione per $e_3 = [001]^T$) dell'accelerazione di gravità g ;

l'ultimo termine è dovuto dalle spinte dei motori sulla massa complessiva: fornisce l'accelerazione che viene impressa al corpo rigido grazie ai rotori.

Per quanto riguarda la dinamica rotazionale, si ha che il momento torcente corrispondente è:

$$I\dot{\omega} = -\omega \times I\omega + \sum_{N,S,E,W} (q_i + m_i) \quad (19)$$

ovvero, è il prodotto vettoriale tra il momento angolare e la velocità, più il contributo dato dalla somma di tutte le coppie e momenti torcenti dei quattro motori.

2.8 Equazioni del sistema

Il modello matematico di quadricottero si può trovare con diversi formalismi, quali quello di Newton-Eulero e quello di Lagrange. È un modello già noto in letteratura, infatti dal documento [5], vediamo che le sei equazioni differenziali che definiscono il moto del drone, una per ogni grado di libertà, possono essere riassunte in:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} R_b^i \left(\begin{bmatrix} 0 \\ 0 \\ \sum T_i \end{bmatrix} - \begin{bmatrix} K_1 \dot{x} \\ K_2 \dot{y} \\ K_3 \dot{z} \end{bmatrix} \right) \frac{1}{m} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \\ (T_1 - T_2 - T_3 + T_4 + K_4 \dot{\phi}) \frac{l}{I_x} \\ (-T_1 - T_2 + T_3 + T_4 + K_5 \dot{\theta}) \frac{l}{I_y} \\ (CT_1 - CT_2 + CT_3 - CT_4 + K_6 \dot{\psi}) \frac{1}{I_z} \end{bmatrix} \quad (20)$$

dove T_i rappresenta la spinta dovuta dall' i -esimo motore, l è la distanza tra il motore ed il centro di gravità del drone, m la massa, K_i rappresenta il coefficiente di drag mentre C è una costante che lega i momenti alle forze ed infine I_i è la componente dell'inerzia lungo l'asse i -esimo.

3 Simulink

In questo capitolo sarà analizzato lo schema di controllo generale, si tratterà sia il caso non lineare sia il caso lineare, descrivendo i vari blocchi di cui sono composti e i vari legami tra input e output di ognuno di essi, arrivando così all'azione di controllo desiderata.

La grande differenza tra i due modelli è che il primo, quello non lineare, garantisce una maggiore fedeltà al caso reale in quanto si basa su di un modello matematico contenente vari termini non lineari, mentre il secondo (lineare) è più importante in fase di progettazione in quanto permette di determinare facilmente la struttura e la modifica del controllore.

Una volta scaricato il Simulink Support Package for Parrot Minidrones, si può accedere al modello minidrone Mambo inserendo nella command window di Matlab il comando `asbQuadcopterStart`. All'avvio del progetto si apre lo schema completo (figura 6).

Si nota una struttura di controllo in controeazione dove il blocco Airframe rappresenta il processo P ed il blocco FCS il controllore G (è proprio questo blocco che andrà poi implementato all'interno del drone). Le altri componenti vengono utilizzate per la lettura dei valori dei sensori, la generazione dei riferimenti, la simulazione dell'ambiente ed infine per la simulazione di volo.

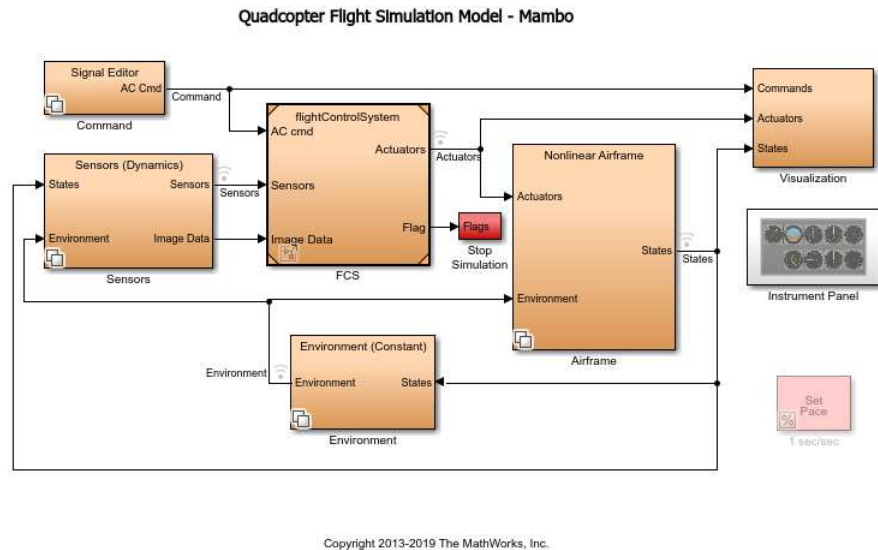


Figure 6: Schema completo Simulink.

Ogni blocco è analizzabile separatamente in base al suo ruolo:

- Command: blocco che genera i segnali di riferimento;
- Flight Control System(FCS): è il controllore vero e proprio del processo;

- Non Linear (Linear) Airframe: rappresenta il modello del processo e può essere cambiato con il modello lineare settando la variabile VSS VEHICLE (figura 7);
- Environment: in questo blocco sono contenute tutte le variabili di ambiente;
- Sensors: contiene le relazioni tra i diversi sensori presenti a bordo e le costanti di trasduzione;
- Visualization: è un blocco ausiliario che permette la visualizzazione dell'andamento dei valori delle variabili di stato e dei segnali di controllo nel tempo.

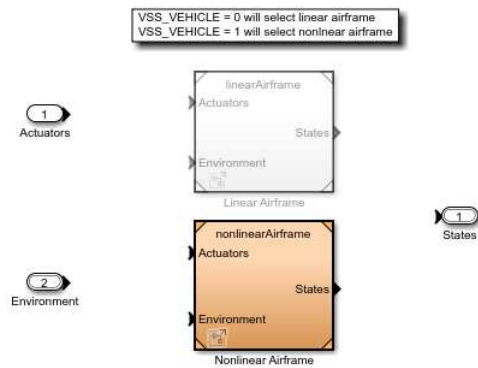


Figure 7: Selezione modello non lineare/lineare.

3.1 Non-linear Airframe

In questa sezione verrà analizzato il modello non lineare. Accedendo al blocco 'Airframe/Nonlinear Airframe' (figura 8) si possono osservare tre blocchi principali:

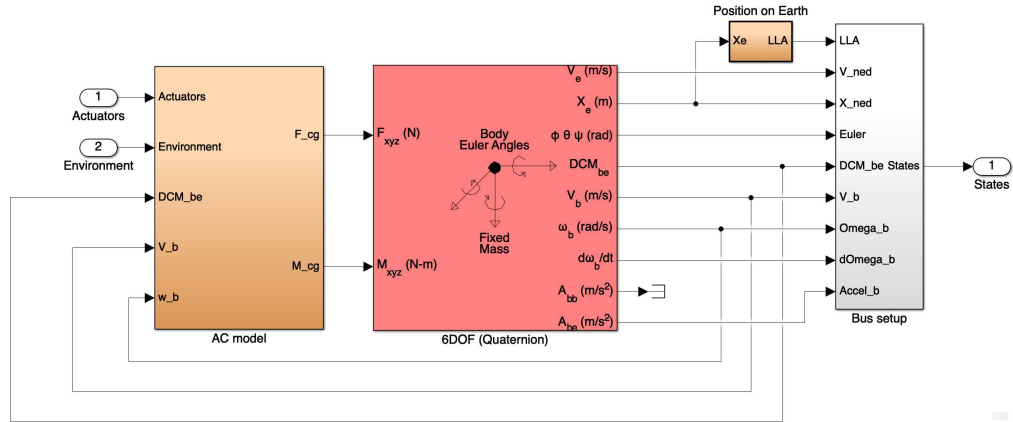


Figure 8: Modello non lineare.

1. AC model: le forze e i momenti meccanici, calcolati rispetto al centro di gravità del drone ed introdotti nel precedente capitolo, sono generati da questo blocco. Contiene la parte della dinamica: in ingresso troviamo le velocità lineari v_b e le velocità angolari w_b , espresse sempre rispetto al sistema solidale con il corpo, l'azione di spinta degli attuatori t_i , l'azione delle variabili di ambiente Environment ed infine i valori della matrice rotazionale necessari per il passaggio dal sistema di riferimento solidale al drone a quello inerziale (R_b^i).

Tale blocco, visibile in figura 9, è composto a sua volta dai blocchi:

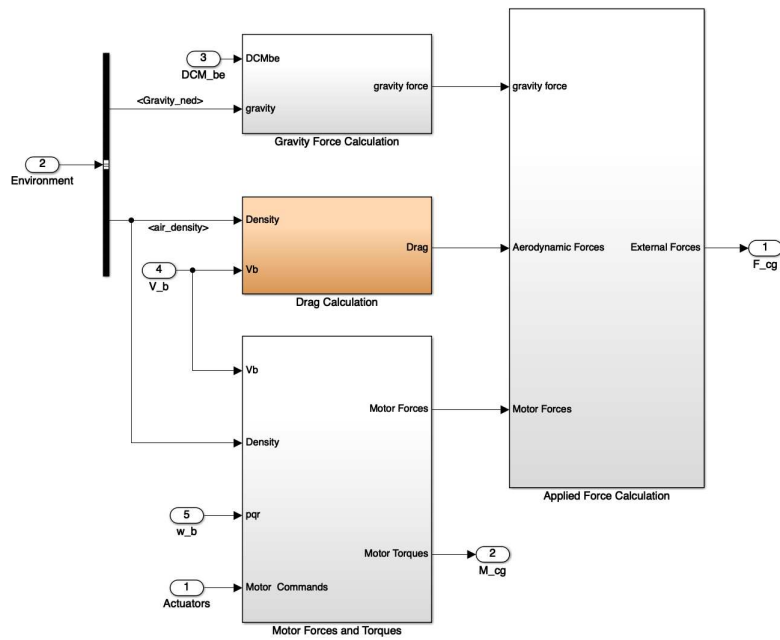


Figure 9: AC model.

- Gravity Force Calculation (figura 10): riceve in ingresso la matrice di rotazione DCM e la costante di gravità e calcola la forza di gravità rispetto al Body Frame. Rappresenta la formula 13 discussa nel capitolo precedente.

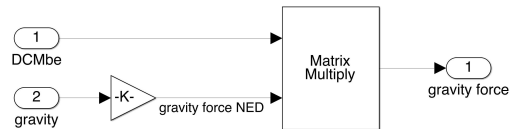


Figure 10: Gravity Force Calculation.

- Drag Calculation (figure 11 e 12): calcola la forza di resistenza che l'aria imprime al drone in moto, è dovuta dallo spostamento e rappresenta la formula 14 discussa precedentemente.

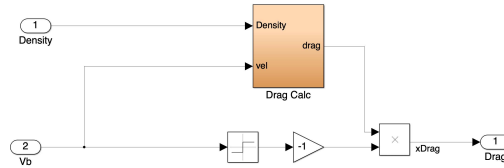


Figure 11: Drag Force Calculation 1.

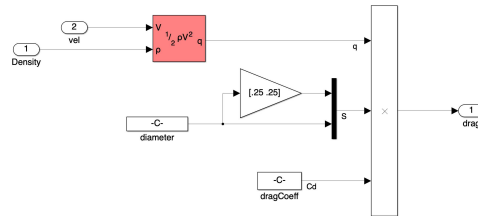


Figure 12: Drag Force Calculation 2.

- Applied Force Calculation (figura 13): contiene un blocco Simulink che genera un vettore avente come componenti la forza di gravità, la forza di drag e le forze generate dai quattro motori.

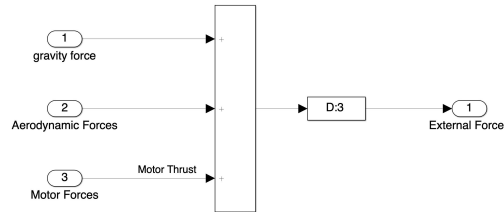


Figure 13: Applied Force Calculation.

- Motor Forces and Torques (figura 14): contiene il modello descritto dalle equazioni della formula 20. Il blocco centrale (figura 15) ha in input la velocità lineare e angolare rispetto al Body Frame, il vettore Motor Commands contenente lo sforzo di controllo che arriva dal controllore ed infine il vettore D, il quale descrive la posizione rispetto al centro di massa dei quattro rotori. Il blocco MotorsToW (figura 16) trasforma lo sforzo di controllo, in un segnale leggibile dai motori tramite dei blocchi gain e saturazioni. Contiene la parte della dinamica corrispondente alle formule 17, 15, 16, 18, 19.

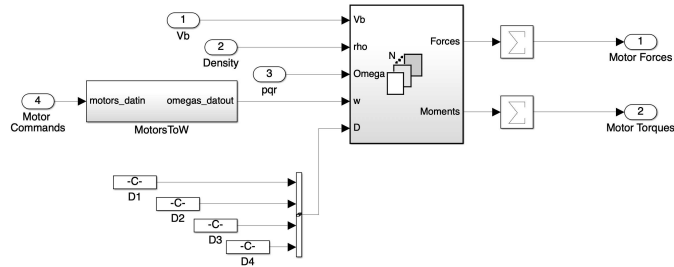


Figure 14: Motor Forces and Torques.

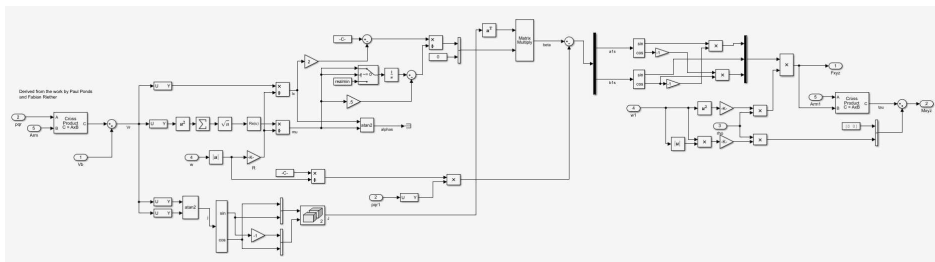


Figure 15: Rotor Dynamics.

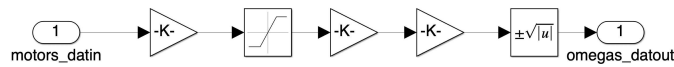


Figure 16: MotorsToW.

2. 6DOF (6 Degrees Of Freedom): blocco fornito direttamente da Matlab. In input riceve le forze e i momenti calcolati in (14) e (16). Successivamente dal sistema (28) determina in output le grandezze di velocità, accelerazioni e angoli.
3. Bus setup: compone un vettore di bus. In esso son contenute tutte le velocità espresse in entrambi i SdR, oltre alla posizione espressa in North-East-Down (NED), accelerazioni e angoli di rotazione.

3.2 Flight Control System (FCS)

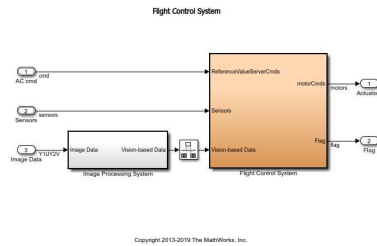


Figure 17: Flight Control System esterno.

Dalla figura 6 è possibile accedere al Flight Control System (FCS), il quale è uno dei componenti principali e su di esso si è svolta l'attività di sviluppo del sistema di controllo (figura 17). In input riceve i valori dei riferimenti provenienti dal blocco Command (nel caso analizzato Signal Builder) ed i valori delle variabili di stato provenienti dai sensori (l'entrata relativa a Image Data è stata tralasciata). In output genera i segnali di riferimento degli attuatori che effettuano l'azione di controllo sui quattro motori e l'uscita per la Flag.

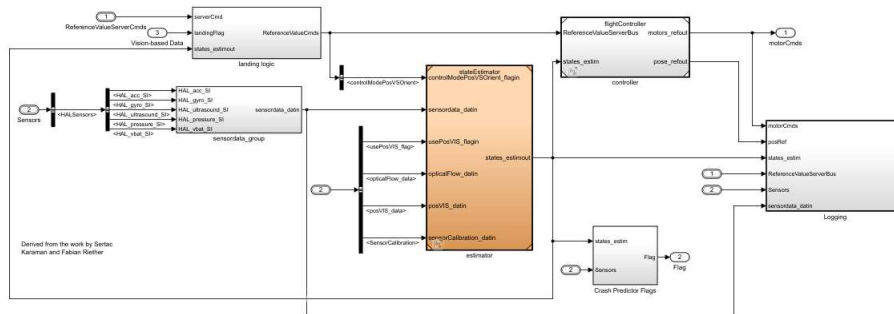


Figure 18: Flight Control System interno.

All'interno di tale blocco, mostrato in figura 18, si trova la parte di controllo vera e propria che inseguito potrà essere compilata e trasformata in codice binario per essere infine caricata sul drone. È composto da 6 sottoblocchi:

1. **Sensordata group:** prende dal bus principale i valori delle variabili dei vari sensori ed effettua una pre-elaborazione che prevede la ridenominazione delle variabili fornite. Come mostrato dalla figura 19 l'uscita di questo blocco va ai blocchi estimator e Logging;

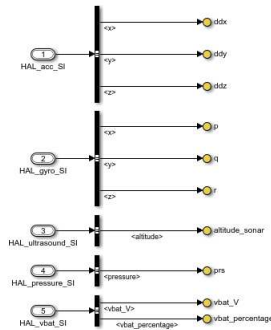


Figure 19: sensordata group.

2. Landing logic: in ingresso riceve la stima degli stati, il riferimento del blocco command e la clausola di arresto (flag). Combinando questi tre input genera il segnale di riferimento vero e proprio (figura 20);

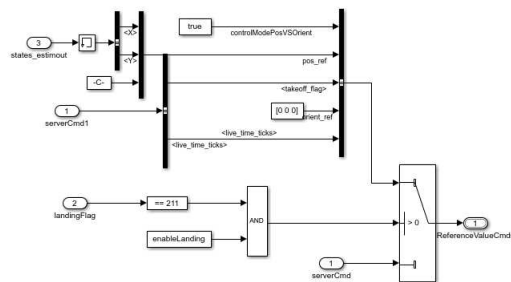


Figure 20: landing logic.

3. estimator: riceve ed elabora i valori dei segnali provenienti dai vari sensori, effettua una stima delle grandezze utilizzate dal controller e restituisce tali valori alla gran parte degli altri blocchi (figura 21);
4. Crash Predictor Flag: osservabile in figura 22 può essere interpretato come una verifica delle condizioni di sicurezza di volo. Riceve le informazioni fornite dai sensori e dal blocco estimator, le confronta con dei parametri assegnati in precedenza e verifica la sicurezza delle condizioni di volo. In caso di movimenti pericolosi o non desiderati si triggera una flag che termina la simulazione e/o disattiva i motori. Vi è la possibilità di personalizzare questo blocco aggiungendo ulteriori condizioni;

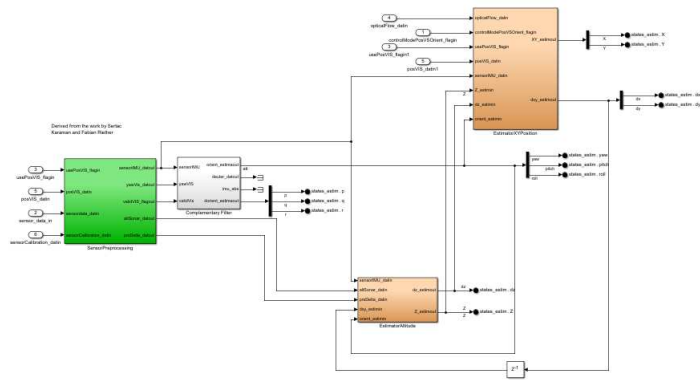


Figure 21: estimator.

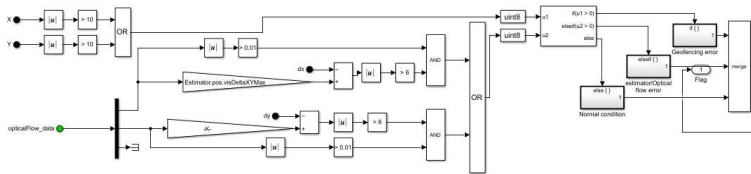


Figure 22: Crash Predictor Flags.

5. Logging: crea un file di log durante la simulazione di volo del drone contenente i valori delle variabili di input/output del blocco FCS. Questo file è successivamente utilizzato per verificare il corretto funzionamento post volo e per determinare eventualmente quali siano le cause delle anomalie. La struttura di questo blocco è visibile in figura 24. Per un'analisi più dettagliata del comportamento dei PID del sistema generale si possono aggiungere dei blocchi 'Scope' alle variabili di interesse per valutarne l'andamento (nella figura 23, per una più chiara visione, è riportata solo la parte superiore di interesse del blocco Logging, il resto non deve essere cancellato);

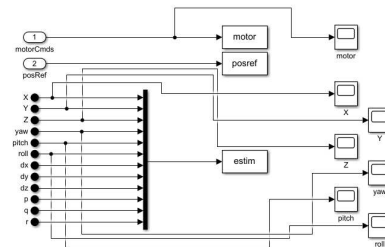


Figure 23: Logging con Scope.

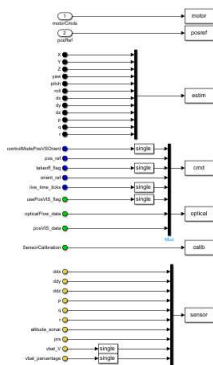


Figure 24: Logging.

- controller: costituisce il controllore vero e proprio del sistema complessivo. Accedendovi si hanno sei sottoblocchi osservabili in figura 25, si osservi che quando lo switch è attivo in posizione 1 (come nel caso in figura), i blocchi XY-to-reference-orientation e Attitude vanno a definire un sistema a cascata, ovvero i segnali di controllo della posizione X-Y influenzano il sistema di controllo degli angoli di pitch e roll. Nel caso in cui lo switch sia invece in posizione 2, saranno le variabili p e q contenute nel blocco Command 'Signal Builder' a definire il controllo sui due angoli.

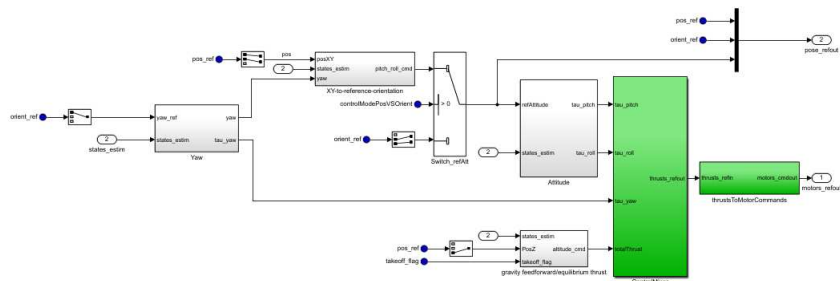


Figure 25: controller.

I sei blocchi sono così costituiti:

- Yaw: riceve in ingresso il valore e la relativa velocità angolare dell'angolo di Yaw. Contiene un controllore PID e dalla figura 26 si nota che esso ha una struttura del tipo Derivative of Output Controller dove appunto l'azione derivativa, che si trova in retroazione, viene eseguita sull'uscita (r) e non sull'errore. In uscita fornisce il segnale per il controllo dell'angolo di Yaw. In figura 27 è riportato l'andamento del PID di controllo dell'angolo di yaw, visualizzabile dal blocco scope posto in Logging;

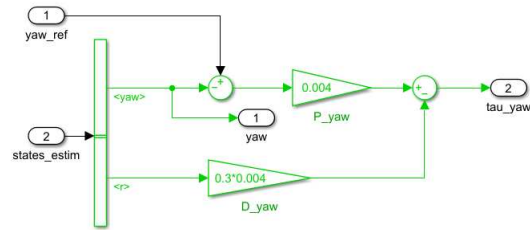


Figure 26: Yaw.

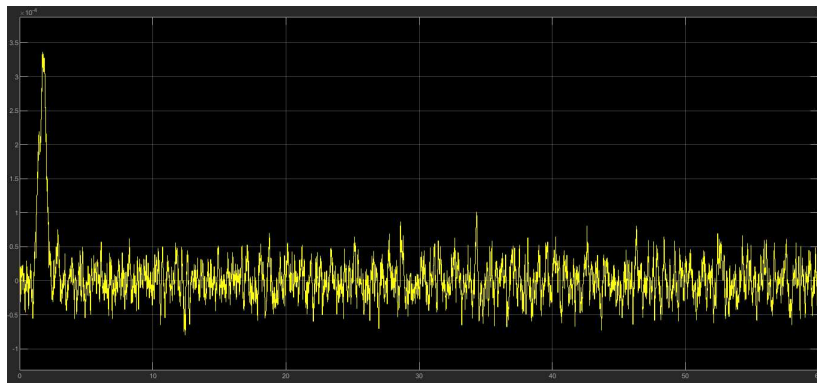


Figure 27: Andamento PID originale per il controllo dell'angolo di yaw.

- XY-to-reference-orientation: riceve in ingresso l'angolo di Yaw, i riferimenti e i valori delle posizioni longitudinali e latitudinali del drone. Ha un controllore PID del tipo Derivative of Output Controller (PD), per il controllo grandezze del piano X-Y. Come si osserva dalla figura 28 tale controllore è condizionato tra le varie cose, dall'angolo di Yaw, infatti come è già stato detto nei capitoli precedenti, una rotazione attorno all'asse Z fa variare anche la posizione X-Y. In uscita genera un segnale di controllo che indica i valori che dovranno essere assunti dagli angoli di pitch e roll per mantenere le condizioni desiderate. In figura 29 e 30 sono riportate, rispettivamente, le risposte dei PID originali per il controllo della posizione X e Y;

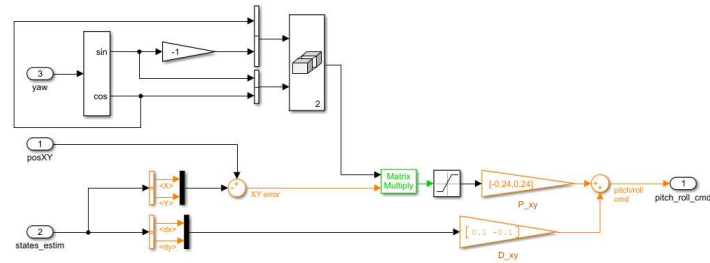


Figure 28: XY-to-reference-orientation.

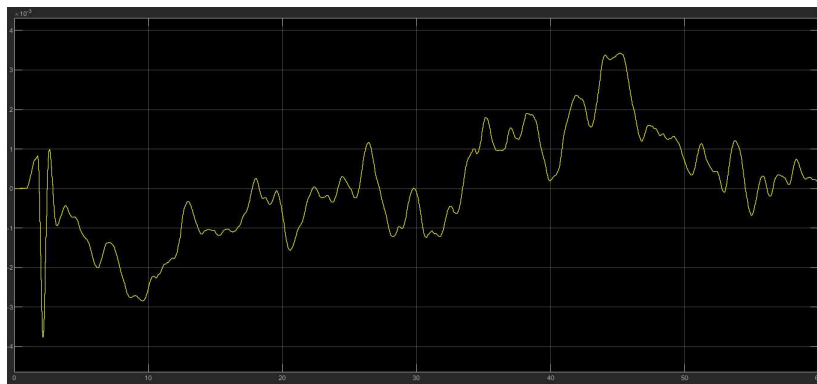


Figure 29: Andamento PID originale per il controllo della X.

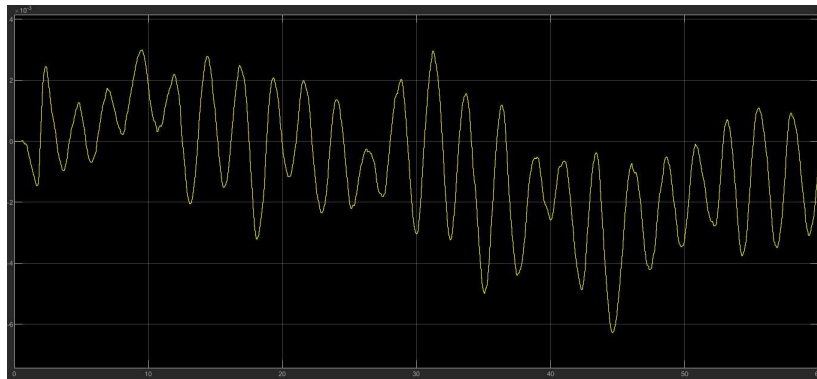


Figure 30: Andamento PID originale per il controllo della Y.

- Attitude: in ingresso prende il valore e le rispettive velocità angolari degli angoli di pitch e roll (figura 31). Può ricevere in input, a seconda della posizione dello switch, il segnale di orientamento generato dal blocco XY to reference orientation piuttosto che il segnale di controllo

proveniente dal blocco Signal Builder.

Questo segnale doppio, relativo agli angoli di pitch e roll, viene relazionato con i valori correnti assunti dagli angoli e provenienti dal blocco 'State Estimator', in modo da ricavarne l'errore rispetto al riferimento. Tale valore sarà passato in input sia ad un blocco proporzionale 'Gain' che ad un blocco integratore. Nel medesimo tempo i segnali relativi alle velocità p e q sono forniti ad un blocco 'Gain' che li moltiplica per un'opportuna costante. Sommando i vari contributi si ottiene un controllore PID (Proporzionale Integratore Derivativo). In uscita fornisce i segnali di controllo dei due angoli considerati. Le figure 32 e 33 mostrano gli andamenti del PID originale per il controllo degli angoli di pitch e roll;

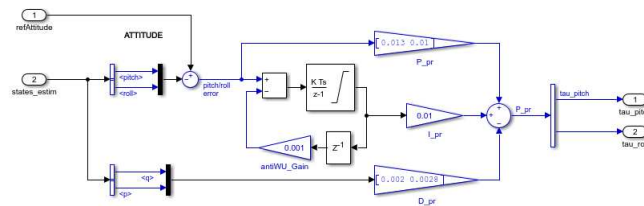


Figure 31: Attitude.

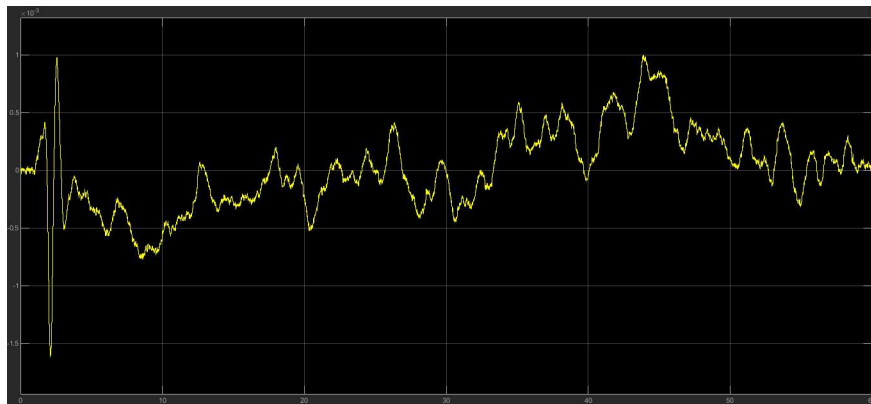


Figure 32: Andamento PID originale per il controllo dell'angolo di pitch.

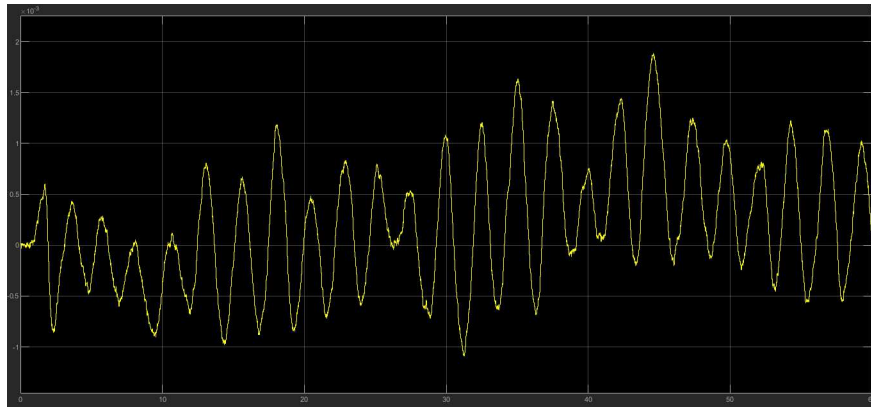


Figure 33: Andamento PID originale per il controllo dell'angolo di roll.

- gravity feedforward /equilibrium thrust: riceve in input il riferimento e l'attuale valore relativo alla quota z, la velocità verticale del veicolo e il segnale Take Of Flag. Quest'ultimo segnale viene attivato, solo in fase di decollo, da uno switch che esclude momentaneamente il controllore dall'azione di controllo. In uscita genera il segnale di controllo relativo all'altezza da terra del drone (figura 34). In figura 35 è raffigurato l'andamento del PID per il controllo dell'altezza;

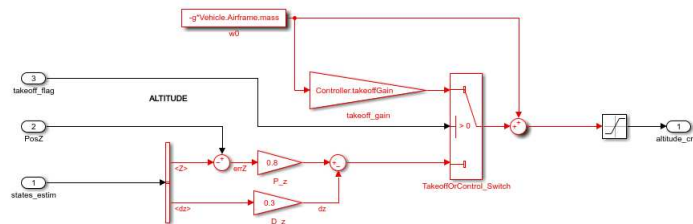


Figure 34: gravity feedforward/equilibrium thrust.

- ControlMixer: i valori uscenti dai controllori dei vari angoli e della quota vengono moltiplicati per una matrice (Controller.Q2Ts) per arrivare ad uno sforzo di controllo grezzo per ciascun motore (figura 36);
- thrustsToMotorCommands: ha in input i valori uscenti dal ControlMixer. Questo blocco ha lo scopo di salvaguardare i motori da sforzi non proporzionati avvalendosi di gain e saturatori. Dalla figura 37 si vede che il blocco gain 'MotorDirections' ha come componenti 1 e -1, questo perché i quattro motori girano a coppie ed in direzione opposta. E' possibile notare anche l'azione di un blocco di saturazione che previene i motori dalla possibilità raggiungere regimi non con-

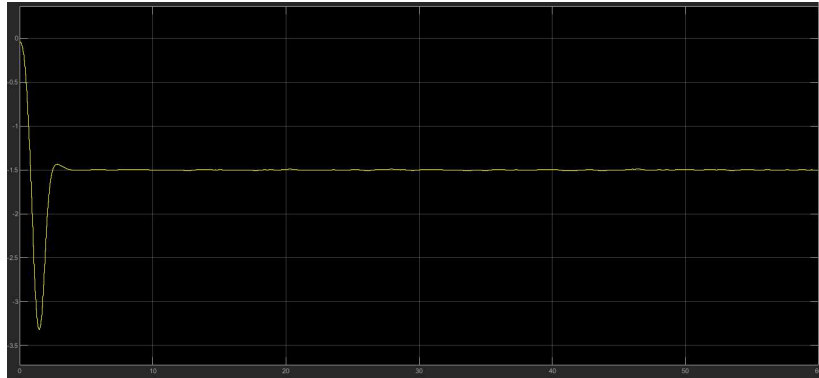


Figure 35: Andamento PID originale per il controllo della Z.

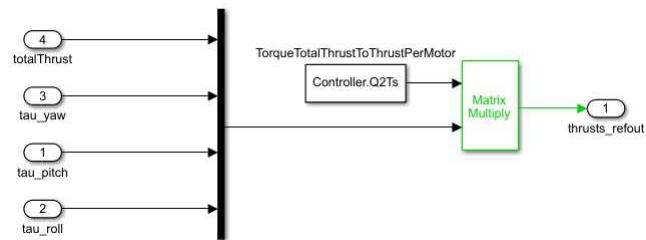


Figure 36: ControlMixer.

cessi. In uscita fornisce le costanti moltiplicative relative all'azione degli attuatori. La figura 38 riporta l'andamento dei motori, si presentano due linee perchè i motori girano a coppie con verso opposto.

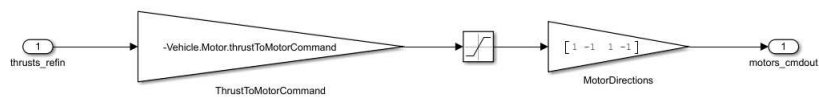


Figure 37: thrustsToMotorCommands.

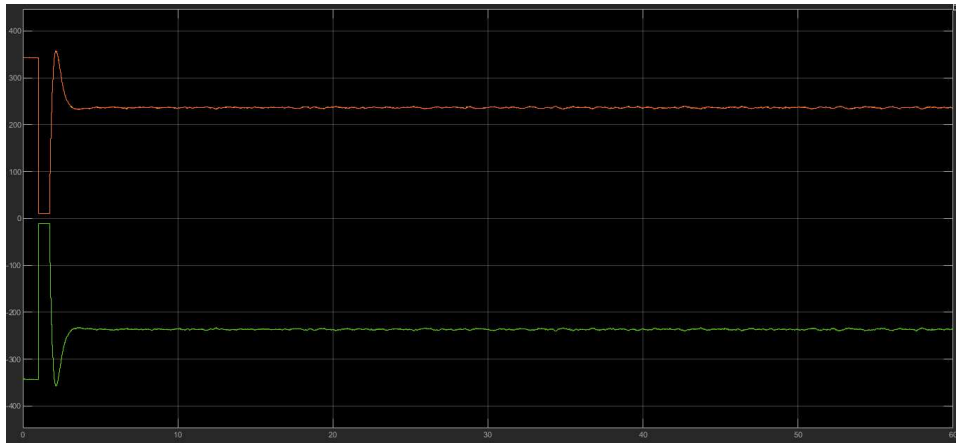


Figure 38: Andamento PID originale per il controllo dei rotori.

3.3 Command

Command permette la scelta della sorgente dei segnali riferimento per il sistema di controllo del drone, settando opportunamente la variabile VSScommand (figura 39).

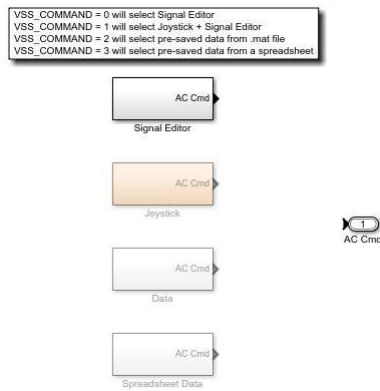


Figure 39: Scelta Command.

- Signal Builder: il sistema riceve in ingresso determinati segnali di default;
- Joystick: permette di comandare manualmente i movimenti del quadricottero ma il blocco Signal Builder rimane comunque attivo, fornendo i riferimenti alle variabili non direttamente controllate;
- Data: permette la selezione dei valori di riferimento tra quelli presenti in un file .mat;

- Spreadsheet Data: permette la selezione dei valori di riferimento da un foglio di calcolo.

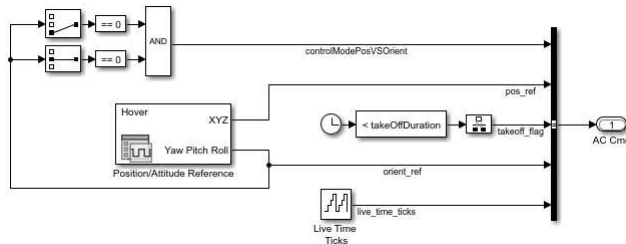


Figure 40: Signal Builder.

Nel caso preso in esame in questo studio verrà utilizzato 'Signal Builder' (figura 40). Analizzando la struttura di quest'ultimo si vede che il riferimento della posizione è ottenuto dai segnali X, Y, Z, mentre il riferimento dell'orientamento è ottenuto dai segnali Roll, Pitch e Yaw.

Qualora i riferimenti di Pitch e Roll siano entrambi nulli, la porta logica AND consente di entrare in modalità di controllo della posizione.

Accedendo a Position/Attitude Reference (figura 41) è possibile settare direttamente l'andamento nel tempo del valore delle variabili di riferimento relative alla posizione (X,Y,Z) e all'orientamento del drone (ϕ, θ, ψ).

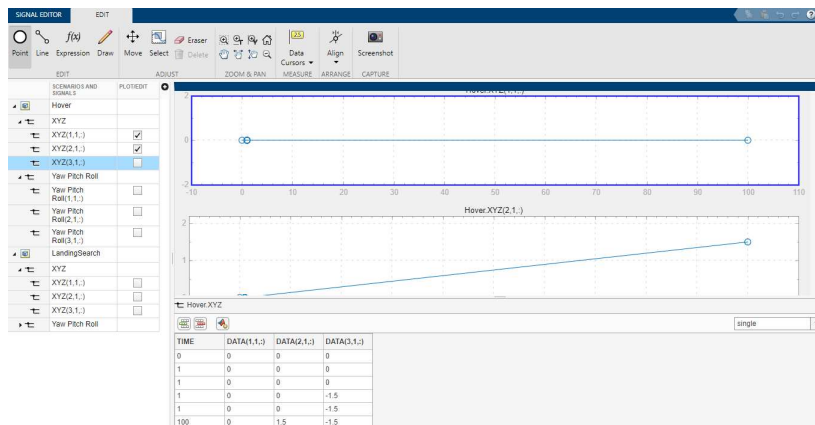


Figure 41: Edit Command.

3.4 Environment

Anche in questo caso si può scegliere tra due tipologie di blocco: costante o variabile (in figura 42 è mostrato il tipo costante). Sono contenuti qui i parametri ambientali, quali:

- gravità;
- temperatura e densità dell'aria;
- velocità del suono;
- pressione;
- campo magnetico.

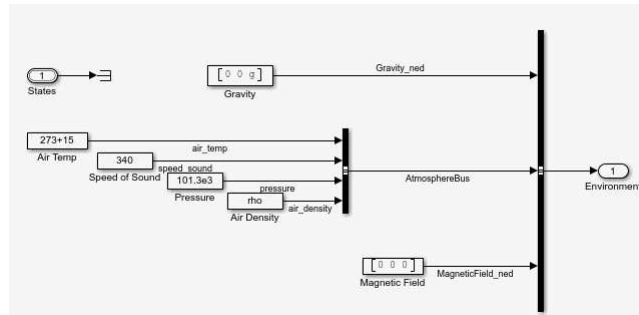


Figure 42: Constant Environment.

3.5 Linear Airframe

Settando il valore della variabile VSS VEHICLE = 0 (figura 7) e aggiornando il sistema si passa al modello lineare del processo mostrato in figura 43. Riceve in ingresso, come nel caso non lineare, Actuators che è il contributo dei quattro motori (descritti in sezione 2.7) e gli elementi citati nella sezione 3.4 relativi alle variabili di ambiente. Esso è composto dalle classiche quattro matrici A, B, C e D utilizzate nell’analisi dei sistemi dinamici lineari. I valori di tali matrici sono ottenibili attraverso il Workspace del progetto, all’interno del gruppo linsys. Accedendo ai file StateName ed Inputname è possibile inoltre conoscere il nome delle variabili di stato, di ingresso e di uscita. Come per gli ingressi, anche le uscite corrispondono alle stesse del caso di modello non lineare².

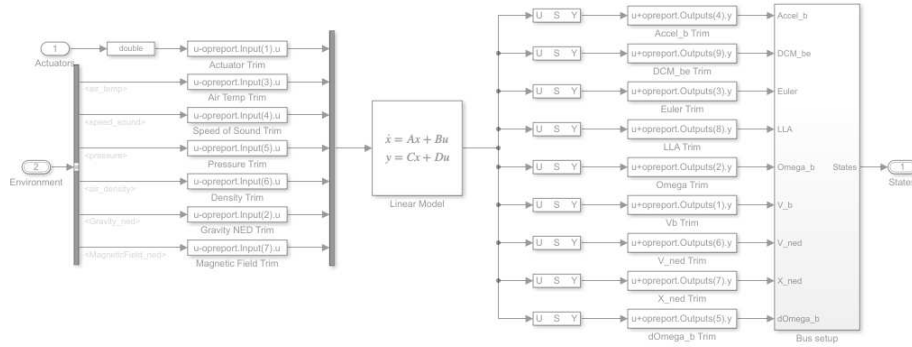


Figure 43: Linear Airframe.

- A: è la matrice della dinamica del sistema. Le variabili di stato sono 12 e quindi essa è una matrice quadrata di dimensioni 12×12 (figura 44);

	1	2	3	4	5	6	7	8	9	10	11	12	
1 phi theta psi(1)	1	0	0	0	0	0	0	0	0	0	0	0	1 phi theta psi(1)
2 phi theta psi(2)	0	1	0	0	0	0	0	0	0	0	0	0	2 phi theta psi(2)
3 phi theta psi(3)	0	0	1	0	0	0	0	0	0	0	0	0	3 phi theta psi(3)
4 ub,vb,wb(1)	-9.8100	0	-3.4402e-09	-3.4402e-09	6.8532e-13	0	0	0	-5.4616e-11	0.1380	-3.2848e-13	0	4 ub,vb,wb(1)
5 ub,vb,wb(2)	1.8478e-22	0	-3.6595e-09	-3.6595e-09	-6.8520e-13	0	0	0	-0.1380	5.8098e-11	-3.5159e-13	0	5 ub,vb,wb(2)
6 ub,vb,wb(3)	1.4456e-11	0	-6.8347e-13	6.8347e-13	0	0	0	0	3.2920e-13	3.5087e-13	0	0	6 ub,vb,wb(3)
7 xe,ye,ze(1)	1.3478e-19	3.2920e-13	1.0000	-1.4969e-18	-5.2295e-12	0	0	0	0	0	0	0	7 xe,ye,ze(1)
8 xe,ye,ze(2)	1.0176e-37	3.5087e-13	1.4969e-18	1	5.2295e-12	0	0	0	0	0	0	0	8 xe,ye,ze(2)
9 xe,ye,ze(3)	1.5087e-13	0	5.2295e-12	-5.2295e-12	1.0000	0	0	0	0	0	0	0	9 xe,ye,ze(3)
10 p,q,r(1)	0	0	-5.7590e-08	-5.7590e-08	-2.7287e-14	0	0	0	-2.1715	9.1429e-10	4.3301e-13	0	10 p,q,r(1)
11 p,q,r(2)	0	0	4.0369e-08	4.0369e-08	-2.1644e-14	0	0	0	6.4089e-10	-1.6192	-5.1363e-13	0	11 p,q,r(2)
12 p,q,r(3)	0	0	-1.4434e-19	-1.2831e-19	3.6185e-19	0	0	0	1.2157e-13	1.1242e-13	-6.8807e-09	0	12 p,q,r(3)

Figure 44: Matrice A con i relativi nomi delle variabili.

- B: rappresenta la matrice degli ingressi del sistema che definisce la relazione tra questi ultimi e la dinamica dello stato del sistema. Essendoci 14 ingressi sarà una matrice rettangolare di dimensioni 12×14 (figura 45);

²La dicitura Trim indica che i valori del modello lineare sono calcolati utilizzando la funzione trimLinearizeOpPoint inclusa nel pacchetto Simulink Control Design.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14		1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	Actuators(1)
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	Actuators(2)
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	Actuators(3)
4	-6.8995e-17	6.8995e-17	-6.8995e-17	6.8995e-17	-9.9331e-14	0	0	0	1.0000	1.4969e-18	5.2296e-12	0	0	0	4	Actuators(4)
5	6.7628e-17	-6.7628e-17	6.7628e-17	-6.7628e-17	9.8151e-14	0	0	0	-1.4969e-18	1.0000	-5.2295e-12	0	0	0	5	AtmosphereBus.air_density
6	-0.0096	0.0096	-0.0096	0.0096	-8.2855	0	0	0	0	0	1.0000	0	0	0	6	AtmosphereBus.air_temp
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	AtmosphereBus.pressure
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	AtmosphereBus.speed_sound
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	Gravity_ned(1)
10	0.4202	0.4202	-0.4202	-0.4202	2.5288e-14	0	0	0	0	0	0	0	0	0	10	Gravity_ned(2)
11	0.3133	-0.3133	-0.3133	0.3133	1.8856e-14	0	0	0	0	0	0	0	0	0	11	Gravity_ned(3)
12	-0.0115	-0.0115	-0.0115	-0.0115	-7.9371e-16	0	0	0	0	0	0	0	0	0	12	MagneticField_ned(1)
															13	MagneticField_ned(2)
															14	MagneticField_ned(3)

Figure 45: Matrice B con i relativi nomi delle variabili.

- C: è la matrice delle uscite del sistema che descrive quest'ultime rispetto al valore delle variabili di stato. Essendo 33 le uscite, è una matrice rettangolare di dimensioni 33×12 (figura 46);

	1	2	3	4	5	6	7	8	9	10	11	12		1
1	0	-9.8100	0	-3.4402e-09	-3.4402e-09	1.8445e-15	0	0	0	-5.4616e-11	0.1380	7.2109e-16	1	phi theta psi(1)
2	9.8100	-2.8478e-22	0	-3.6595e-09	-3.6595e-09	-1.7339e-15	0	0	0	-0.1380	5.8098e-11	-7.2109e-16	2	phi theta psi(2)
3	5.4456e-11	5.4456e-11	0	0	0	0	0	0	0	0	0	0	3	phi theta psi(3)
4	0	5.5511e-12	0	0	0	0	0	0	0	0	0	0	4	ub,yb,wb(1)
5	-5.2296e-12	-5.2295e-12	-1.0000	0	0	0	0	0	0	0	0	0	5	ub,yb,wb(2)
6	1.4969e-18	1.0000	-5.2295e-12	0	0	0	0	0	0	0	0	0	6	ub,yb,wb(3)
7	0	8.3096e-30	1.0000	0	0	0	0	0	0	0	0	0	7	x,eye,ze(1)
8	5.5511e-12	0	0	0	0	0	0	0	0	0	0	0	8	x,eye,ze(2)
9	-1.0000	1.4969e-18	-5.2296e-12	0	0	0	0	0	0	0	0	0	9	x,eye,ze(3)
10	0	-1.0000	0	0	0	0	0	0	0	0	0	0	10	p,q,r(1)
11	1.0000	-2.9029e-23	0	0	0	0	0	0	0	0	0	0	11	p,q,r(2)
12	5.5511e-12	5.5511e-12	0	0	0	0	0	0	0	0	0	0	12	p,q,r(3)
13	1	0	0	0	0	0	0	0	0	0	0	0		
14	0	1	0	0	0	0	0	0	0	0	0	0		
15	0	0	1	0	0	0	0	0	0	0	0	0		
16	0	0	0	0	0	0	9.0026e-06	0	0	0	0	0		
17	0	0	0	0	0	0	0	1.2127e-05	0	0	0	0		
18	0	0	0	0	0	0	0	0	-1.0000	0	0	0		
19	0	0	0	0	0	0	0	0	0	1	0	0		
20	0	0	0	0	0	0	0	0	0	0	1	0		
21	0	0	0	0	0	0	0	0	0	0	0	1		
22	0	0	0	1.0000	0	0	0	0	0	0	0	0		
23	0	0	0	0	1	0	0	0	0	0	0	0		
24	0	0	0	0	0	1.0000	0	0	0	0	0	0		
25	1.7216e-24	1.3478e-19	3.2920e-13	1.0000	-1.4969e-18	-5.2296e-12	0	0	0	0	0	0		
26	-1.3478e-19	2.0176e-37	3.5087e-13	1.4969e-18	1	5.2295e-12	0	0	0	0	0	0		
27	-3.2920e-13	-3.5087e-13	0	5.2296e-12	-5.2295e-12	1.0000	0	0	0	0	0	0		
28	0	0	0	0	0	0	1	0	0	0	0	0		
29	0	0	0	0	0	0	0	1	0	0	0	0		
30	0	0	0	0	0	0	0	0	1	0	0	0		
31	0	0	0	-5.7590e-08	-5.7590e-08	-2.7287e-14	0	0	0	-2.1715	9.1429e-10	4.3301e-13		
32	0	0	0	4.0369e-08	4.0369e-08	-2.1644e-14	0	0	0	6.4089e-10	-1.6192	-5.1363e-13		
33	0	0	0	-1.4434e-19	-1.2831e-19	3.6185e-19	0	0	0	1.2157e-13	1.1242e-13	-6.8807e-09		

Figure 46: Matrice C con i relativi nomi delle variabili.

- D: rappresenta la matrice che descrive il modo in cui le variabili di ingresso sono legate alle variabili di uscita. Quindi questa matrice è rettangolare e di dimensioni 33×14 (figura 47).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	-6.8995e-17	6.8995e-17	-6.8995e-17	6.8995e-17	-9.9331e-14	0	0	0	1.0000	1.4969e-18	5.2296e-12	0	0	0	1 Actuators(1)
2	6.7628e-17	-6.7628e-17	6.7628e-17	-6.7628e-17	9.8151e-14	0	0	0	-1.4969e-18	1.0000	-5.2295e-12	0	0	0	2 Actuators(2)
3	-0.0096	0.0096	-0.0096	0.0096	-8.2855	0	0	0	0	0	1.0000	0	0	0	3 Actuators(3)
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4 Actuators(4)
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5 AtmosphereBus.air_density
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6 AtmosphereBus.air_temp
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7 AtmosphereBus.pressure
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8 AtmosphereBus.speed_sound
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9 Gravity_ned(1)
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10 Gravity_ned(2)
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11 Gravity_ned(3)
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12 MagneticField_ned(1)
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13 MagneticField_ned(2)
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14 MagneticField_ned(3)
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
31	0.4202	0.4202	-0.4202	-0.4202	0	0	0	0	0	0	0	0	0	0	
32	0.3133	-0.3133	-0.3133	0.3133	0	0	0	0	0	0	0	0	0	0	
33	-0.0115	-0.0115	-0.0115	-0.0115	-1.3323e-15	0	0	0	0	0	0	0	0	0	

Figure 47: Matrice D con i relativi nomi delle variabili.

4 Sintesi controllore

In fase di sintesi del controllore si utilizza il modello lineare perchè consente di avere una rappresentazione semplificata della dinamica del sistema ed è inoltre possibile considerare separatamente i legami ingresso/uscita esistenti tra le molteplici variabili del sistema. Dalle matrici A,B,C,D definite nella sezione 3.5 è possibile estrapolare il sottosistema oggetto di questo studio, ovvero il sottosistema per il controllo dell'angolo di rollio, e trattarlo separatamente dal resto del sistema. Le matrici di tale sottosistema sono:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -2.17153 \end{bmatrix} \quad (21)$$

la matrice A è ottenuta considerando i coefficienti corrispondenti alle variabili ϕ e p ;

$$B = \begin{bmatrix} 0 \\ 0.420192 \end{bmatrix} \quad (22)$$

la matrice B è costituita dai coefficienti relativi alle variabili ϕ e p rispetto agli ingressi Actuators(1)(2)(3)(4);

$$C = [1 \quad 0] \quad (23)$$

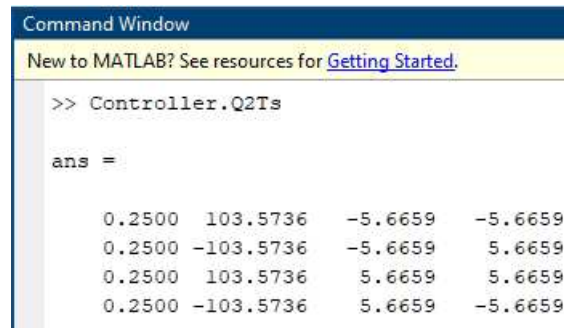
la matrice C viene costruita con i coefficienti corrispondenti alle variabili ϕ e p , rispetto alla variabile di uscita Euler(1)

$$D = [0] \quad (24)$$

infine la matrice D è ricavata dai coefficienti relativi all'uscita Euler(1), legata agli Actuators(1)(2)(3)(4).

Per costruire il legame ingresso/uscita relativo alla variabile di stato controllata è necessario inserire anche due costanti moltiplicative:

1. K_r : effettua la ripartizione del segnale di controllo tra i quattro rotori. La dinamica coinvolta da questa costante è localizzata nel ControlMixer in figura 36. Nel command window di Matlab, digitando il comando 'Controller.Q2Ts', viene mostrata la matrice descrivente le costanti di gestione della ripartizione del segnale di controllo. Le colonne indicano rispettivamente i coefficienti della spinta verticale e del momento di yaw, pitch e roll. Nel caso oggetto di questo studio si andranno a scegliere i coefficienti della quarta colonna.



```

Command Window
New to MATLAB? See resources for Getting Started.

>> Controller.Q2Ts

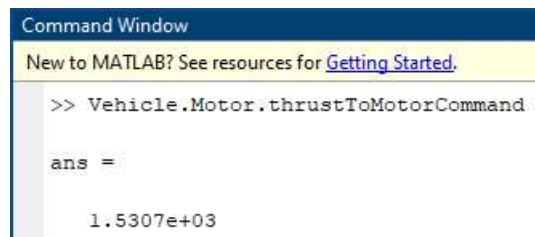
ans =

    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

```

Figure 48: Comandi per il calcolo della costante moltiplicativa K_r .

2. K_t : raffigura il coefficiente di spinta dei motori. Questa costante è presente nel blocco thrustsToMotorCommands introdotto nel precedente capitolo (figura 37). Tale valore si estrapola avvalendosi del comando 'Vehicle.Motor.thrustToMotorCommand' in Matlab.



```

Command Window
New to MATLAB? See resources for Getting Started.

>> Vehicle.Motor.thrustToMotorCommand

ans =

    1.5307e+03

```

Figure 49: Comandi per il calcolo della costante moltiplicativa K_t .

I sei elementi introdotti sino ad ora in questo capitolo (A, B, C, D, K_r , K_t) sono sufficienti per calcolare la funzione di trasferimento (FdT) del sottosistema studiato. Essa è ricavata dapprima definendo le matrici A, B, C, D, in seguito si usufruisce della funzione 'tf' per il calcolo della relativa FdT ed il risultato viene infine moltiplicato per le due costanti K_r e K_t . Quanto appena detto è riassunto nei seguenti comandi in Matlab (figura 50).

Pertanto, la funzione di trasferimento risultante, su cui si concentrerà il resto del lavoro, è:

```

Command Window
New to MATLAB? See resources for Getting Started.

>> A=[0 1;0 -2.17153];
>> B=[0;0.420192];
>> C=[1 0];
>> D=[0];
>> lsys=ss(A,B,C,D);
>> P1=tf(lsys);
>> Kr=5.6659;
>> Kt=1530.7;
>> Ps=P1*Kr*Kt

Ps =

      3644
-----
s^2 + 2.172 s

Continuous-time transfer function.

```

Figure 50: Comandi per il calcolo della Funzione di Trasferimento.

$$FdT = \frac{3644}{s^2 + 2.172s} \quad (25)$$

Nelle successive sezioni verranno analizzate due tecniche di sintesi, studiate nel corso di Controlli Automatici, al fine di progettare un controllore $G(s)$ per il controllo dell'angolo di rollio e della conseguente variazione di posizione lungo l'asse Y. Tale controllore dovrà essere più performante rispetto a quello già implementato di default dal modello utilizzato.

4.1 Sintesi in frequenza

Ottenuta la FdT del sottosistema di interesse è possibile effettuare la sintesi in frequenza del controllore per un riferimento costante o variabile nel tempo. Prima di procedere però, è doveroso definire le specifiche che devono essere soddisfatte dal sistema complessivo in catena chiusa. Esse si dividono in specifiche univoche (le prime due) e specifiche lasche (le altre):

1. sistema di tipo 1;
2. errore a regime permanente $\tilde{e} \leq 0.01$;
3. margine di risonanza $M_r \leq 3$ dB;
4. banda passante $B_3 = 3$ Hz.

A tali specifiche legate al sistema a ciclo chiuso corrispondono delle specifiche che devono essere soddisfatte dal sistema in catena aperta:

1. $F(s)$ ha un polo in $s=0$;
2. $\tilde{e} = k_d^2/k_f \leq 0.01$;³
3. margine di fase $M_\psi \geq 42^\circ$;
4. pulsazione di attraversamento $w_T = 12\text{rad/s}$.

Nel modello originale, il controllo dell'angolo di roll del drone avviene nel blocco Attitude (figura 31) unitamente al controllo dell'angolo di pitch, ovvero sono entrambi gestiti dallo stesso PID. Per poter effettuare il controllo solo sul rollio occorre disaccoppiare tramite un demultiplexer i due segnali, togliere la stima della variabile p e due coefficienti dai blocchi 'Gain', come mostrato nella seguente figura.

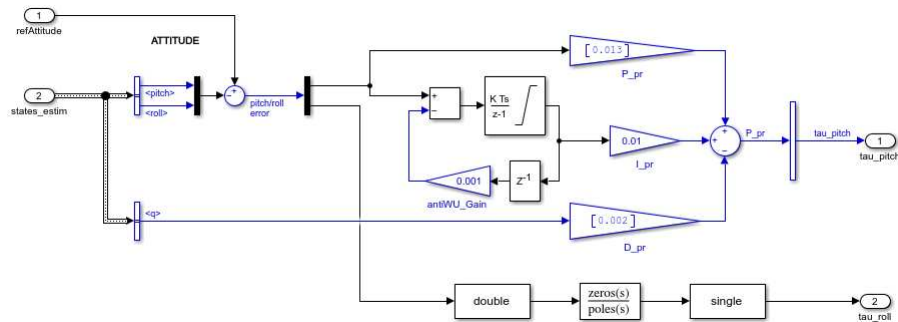


Figure 51: Blocco Attitude con PID disaccoppiati.

³ k_d è il coefficiente di retroazione (nel caso analizzato si tratta di retroazione unitaria), k_f è il prodotto tra il guadagno del processo k_p e quello del controllore k_g .

Questo schema combina l'azione dei due controllori, il PID originale continua l'azione di controllo sull'angolo di beccheggio, mentre il controllore progettato si occupa di gestire l'angolo di rollio.

E' stato inoltre necessario cambiare due parametri di sincronizzazione.

Uno nel blocco landing logging discusso prima: cliccando con il tasto destro del mouse su tale blocco ed andando su Block Parameters bisogna spuntare la casella 'Treat as atomic unit' ed inserire il valore 'Ts' nel campo Sample time (figura 52).

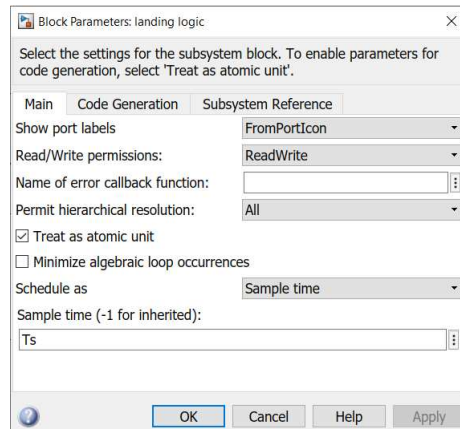


Figure 52: Parametro del blocco landing logic modificato.

L'altro parametro da cambiare riguarda il blocco Flight Control System: cliccando con il tasto destro del mouse su tale blocco ed andando su Block Parameters è necessario mettere il valore -1 nel campo Sample time (figura 53).

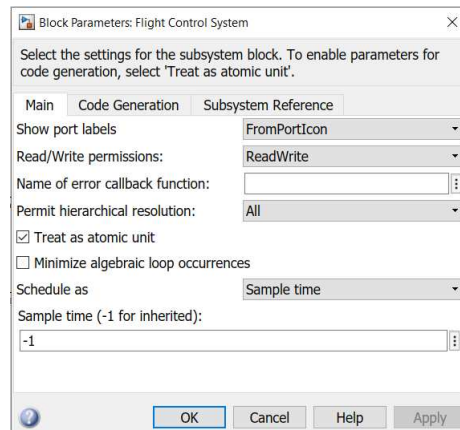


Figure 53: Parametro del blocco Flight Control System modificato.

4.1.1 Riferimento costante

L'ingresso del riferimento refAttitude è passato dallo Switch refAtt il quale, in base al segnale di attivazione determina se il riferimento dovrà essere letto dal blocco Signal Builder (fornisce segnali costanti) piuttosto che dal blocco XY reference orientation (fornisce segnali variabili). Operando propriamente sulla porta logica presente nel blocco Signal Builder, si forza lo Switch a scegliere un riferimento costante per l'angolo di roll (figura 54).

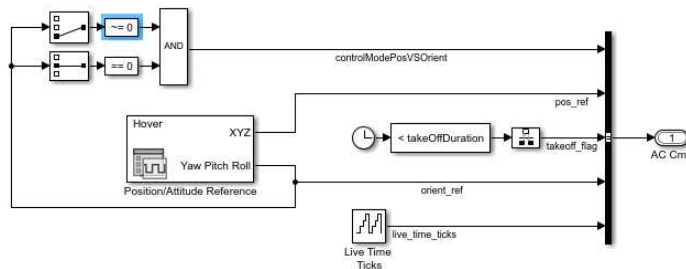


Figure 54: Blocco Signal Builder con porta logica modificata.

Il pacchetto Control System Designer disponibile in Matlab permette di implementare tutti gli strumenti necessari per svolgere l'attività sintesi di un controllore. Richiamandolo con il comando sisotool si possono visualizzare in particolar modo i diagrammi di Bode, Nyquist e la risposta a segnali standard (nel caso in esame si farà riferimento al gradino) di un sistema. Inserendo la FdT (eq. 25) nel blocco G (relativo al processo) del diagramma a blocchi del classico sistema in controreazione è possibile osservare i diagrammi sopra citati, relativi al sottosistema descritto nelle sezioni precedenti:

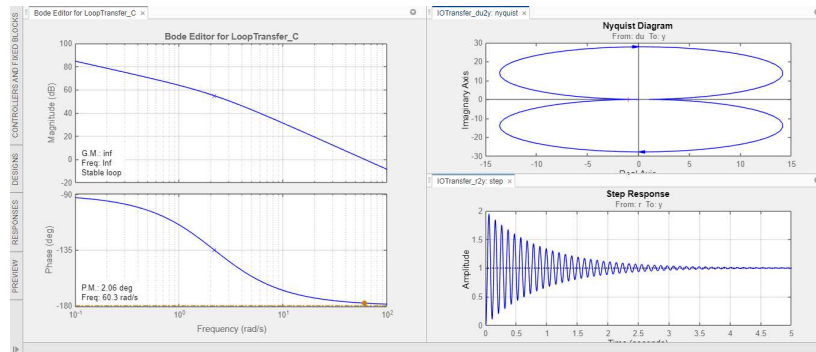


Figure 55: Diagrammi Bode, Nyquist e risposta al gradino del sistema in c.a.

Visivamente risalta subito il fatto che la risposta al gradino di tale sistema non è accettabile e sicuramente non soddisfa le specifiche richieste, ciò lo si vede anche dal margine di fase e pulsazione di attraversamento dei diagrammi di

Bode. Occorre necessariamente progettare un controllore che migliori sensibilmente tale risposta, partendo dal soddisfacimento delle specifiche univoche. La prima specifica risulta già soddisfatta in quanto il sistema risulta essere già di tipo 1 e quindi è sufficiente verificare solo la seconda, ovvero quella sull'errore a regime permanente, per ottenere un controllore di prima istanza \hat{G} . Conoscendo k_p ed imponendo \tilde{e} non è difficile ricavare k_g , ovvero il guadagno del controllore:

$$\tilde{e} = \frac{k_d^2}{k_f} = \frac{1}{k_f} = \frac{1}{k_g \cdot k_p} \quad (26)$$

e quindi

$$k_g = \frac{1}{k_p \cdot \tilde{e}} = \frac{1}{3644 \cdot 0.01} = 0.027 \quad (27)$$

Inserendo ora il guadagno statico appena ottenuto all'interno del blocco controllore (blocco C) del sistema, ed aggiornando l'architettura, si ottengono i seguenti diagrammi:

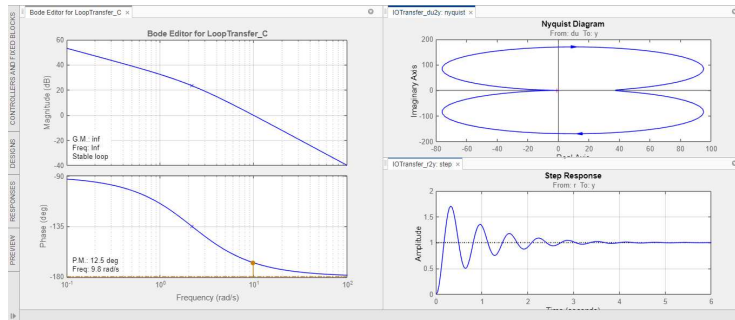


Figure 56: Diagrammi Bode, Nyquist e risposta al gradino del sistema in c.a. dopo il guadano statico.

Si osserva un buon miglioramento della risposta generale, il tempo di salita è accettabile ma vi è un'eccessiva sotto e sovraelongazione ed un transitorio lungo. Inoltre, dalle specifiche lasche, si ha che il margine di fase deve essere aumentato rispetto al valore corrente di almeno 30° ($42^\circ - 12.5^\circ$) e la pulsazione di attraversamento, che si attesa sui 10 rad/s circa, dovrà essere leggermente aumentata. Si rende quindi necessario l'uso di una rete compensatrice che vada

ad aumentare notevolmente il margine di fase M_ψ e di poco il valore del modulo (nei diagrammi di Bode si andranno a scegliere pulsazioni piccole).

$$R_a(s) = \frac{1 + \tau_a \cdot s}{1 + \frac{\tau_a}{m_a} \cdot s} \quad (28)$$

Dai diagrammi universali sono stati scelti i valori di $m_a = 8$ e $w_T \cdot \tau_a = 1$, i quali producono la variazione di modulo e fase che più si avvicina a quella desiderata. Avendo imposto $w_T = 12 \text{ rad/s}$ è stato possibile ricavare $\tau_a = 0.08$. Per valutare la risposta del sistema in catena aperta si deve aggiornare il blocco controllore con il nuovo, cioè $G = \hat{G} \cdot R_a$.

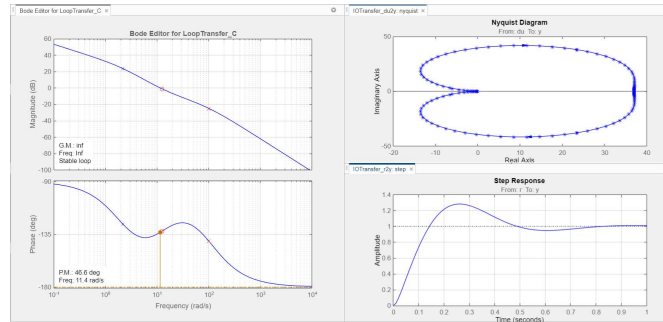


Figure 57: Diagrammi Bode, Nyquist e risposta al gradino del sistema in c.a. dopo la rete anticipatrice.

La pulsazione di attraversamento w_T è circa 12 rad/s mentre il margine di fase M_ψ maggiore di 42° quindi anche le specifiche lasche risultano verificate. Anche il tempo di salita risulta accettabile.

L'ultimo passo da effettuare è il calcolo dei poli e degli zeri della $G(s)$ in Matlab⁴:

- Zeri: [-12.5];
- Poli: [-100].

Inserendo questi valori, assieme al guadagno statico, all'interno del blocco $\frac{zeros(s)}{poles(s)}$, presente in figura 51 ed accedendo ai blocchi 'scope' di figura 23, si visualizzano i seguenti andamenti dei vari PID.

⁴richiamando i comandi pole(G) e zero(G)

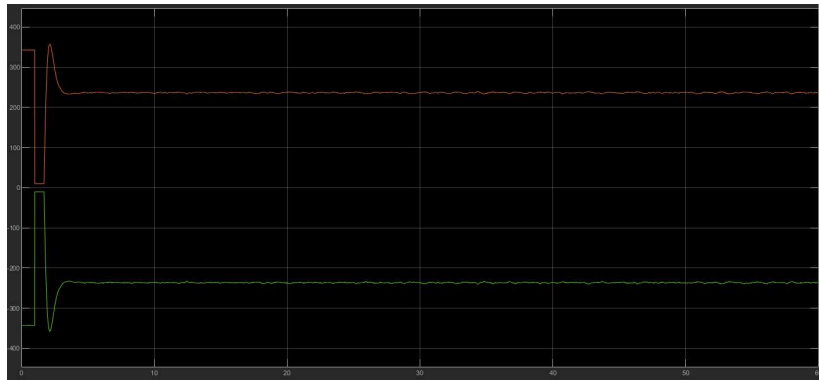


Figure 58: Tracciato motori con controllore G, riferimento costante.

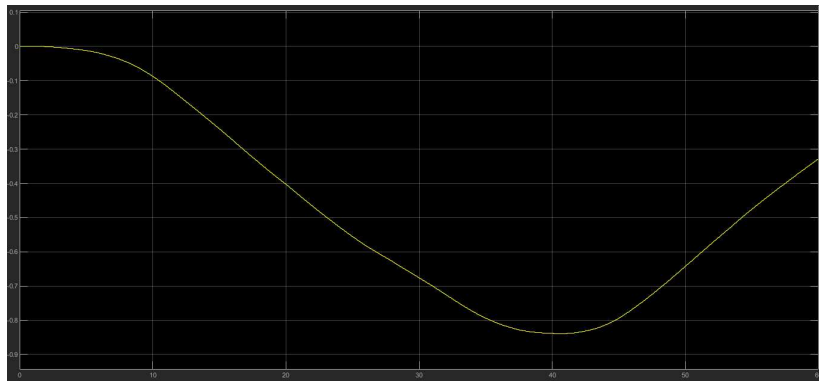


Figure 59: Tracciato asse X con controllore G, riferimento costante.

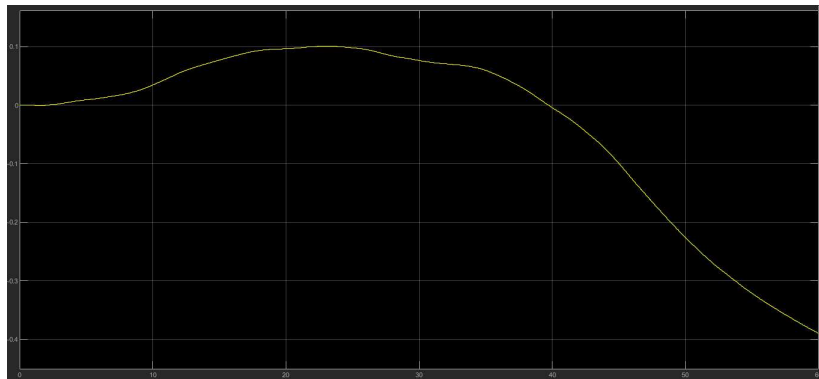


Figure 60: Tracciato asse Y con controllore G, riferimento costante.

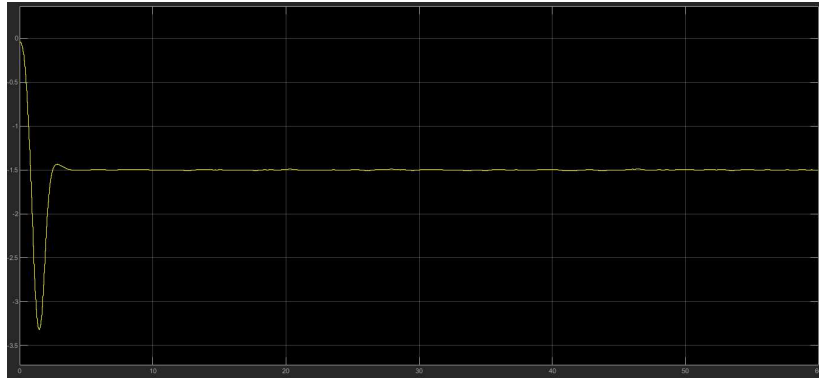


Figure 61: Tracciato asse Z con controllore G, riferimento costante.

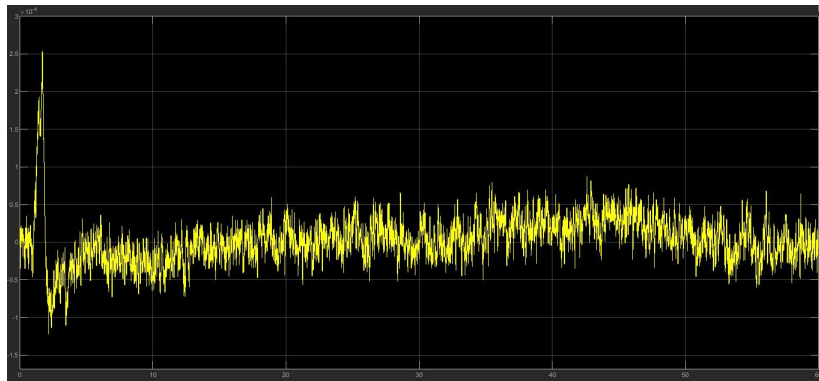


Figure 62: Tracciato angolo di pitch con controllore G, riferimento costante.

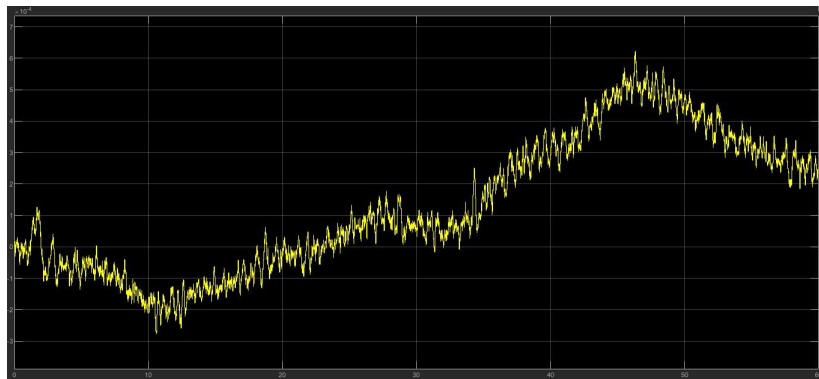


Figure 63: Tracciato angolo di roll con controllore G, riferimento costante.

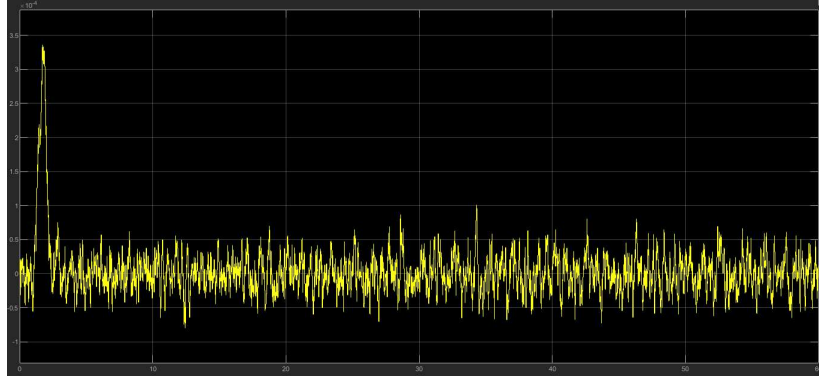


Figure 64: Tracciato angolo di yaw con controllore G, riferimento costante.

4.1.2 Riferimento variabile

Con il controllore ottenuto nella sottosezione precedente, il sistema non è in grado di inseguire un riferimento variabile infatti, dalle simulazioni effettuate, il drone dopo essersi stabilizzato lungo l'asse Z ha oscillazioni sempre più pronunciate lungo l'asse Y fino a triggerare la flag e terminare la simulazione. Per l'implementazione all'interno del modello Symulink di un controllore che soddisfi tale richiesta non è stato necessario modificare la porta logica del blocco Signal Builder, come fatto per il segnale costante, poiché si desidera ricevere il segnale di ingresso al blocco Attitude direttamente dal blocco 'XY to Reference Orientation', il quale produce appunto un segnale variabile nel tempo. Il nuovo sistema dovrà essere più performante rispetto al precedente, ovvero in catena aperta esso dovrà avere tempo di salita minore, sovra elongazione minore e un ragionevole tempo di assestamento: sul piano della sintesi in frequenza si dovrà quindi aumentare il margine di fase m_ψ lasciando pressochè inalterato il modulo. La parte di sintesi per le specifiche univoche che è stata già effettuata nella sezione 4.1.1 risulta valida anche in questo caso, di conseguenza si dovranno progettare solo le eventuali reti compensatrici.⁵ In particolare si è cercata prima una funzione anticipatrice R_a che facesse aumentare di molto il m_ψ (circa 60°) e di poco il modulo. Scegliendo $m_a = 16$ e $w_t\tau_a = 3$, ricordando che $w_t\text{desiderata} = 12$, si ottiene $\tau_a = 0.25$.

$$R_a(s) = \frac{1 + 0.25 \cdot s}{1 + 0.01 \cdot s} \quad (29)$$

⁵per la scelta di m_a e τ_a si faccia riferimento alla figura 56 già compresa di guadagno statico

Inserendo il prodotto tra R_a e \hat{G} nel blocco controllore, come fatto per il riferimento costante, si osservano i seguenti diagrammi:

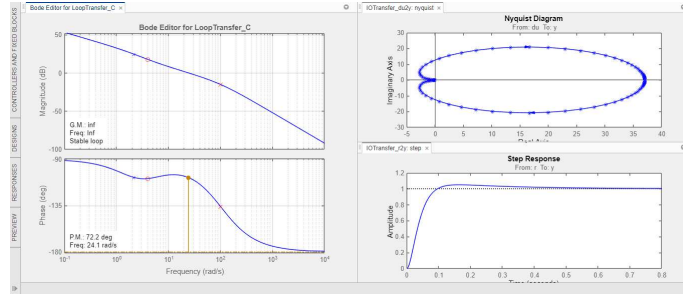


Figure 65: Diagrammi Bode, Nyquist e risposta al gradino del sistema in c.a. dopo la rete anticipatrice.

Il margine di fase m_ψ è aumentato di molto ed ora soddisfa la richiesta precedente. Tuttavia la pulsazione di attraversamento w_t risulta troppo elevata (per le specifiche richieste dovrebbe attestarsi sui 12 rad/s) e quindi si rende necessaria la progettazione di una rete attenuatrice che faccia diminuire in particolar modo il modulo.⁶

$$R_i(s) = \frac{1 + \frac{\tau_i}{m_i} \cdot s}{1 + \tau_i \cdot s} \quad (30)$$

Scegliendo $m_i = 2$ e $w_t \tau_i = 100$ si può facilmente ricavare $\tau_i = 8$ e riscrivere la R_i :

$$R_i(s) = \frac{1 + 4 \cdot s}{1 + 8 \cdot s} \quad (31)$$

Con essa, il sistema complessivo logicamente di R_a e \hat{G} , fornisce i seguenti diagrammi:

⁶si dovranno scegliere valori di $w_t \cdot \tau_i$ molto alti.

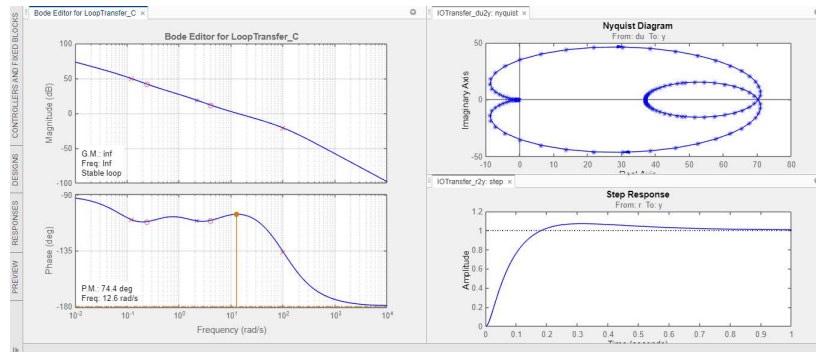


Figure 66: Diagrammi Bode, Nyquist e risposta al gradino del sistema in c.a. dopo la rete attenuatrice.

Poli, zeri e guadagno della funzione di trasferimento di tale controllore sono:

- Guadagno: 0.027;
- Zeri: [-0.25 -4];
- Poli: [-0.125 -100].

Inserendoli insieme al guadagno \hat{G} nel blocco $\frac{zeros(s)}{poles(s)}$ all'interno di Attitude in Symulink, come fatto per il segnale costante, si ottengono complessivamente i seguenti tracciati dei vari PID.

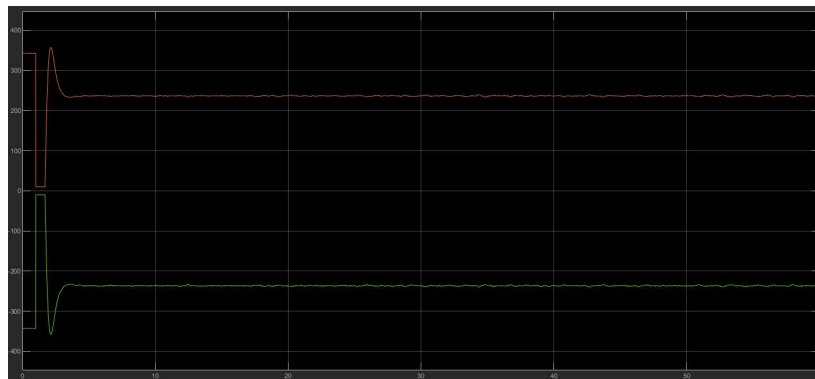


Figure 67: Tracciato motori con controllore G, riferimento variabile.

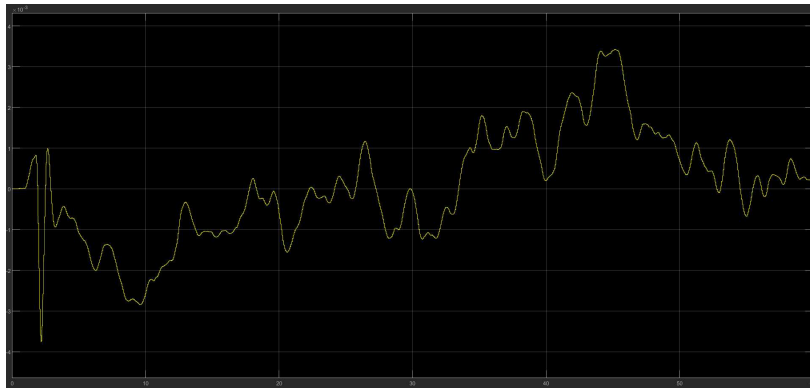


Figure 68: Tracciato asse X con controllore G, riferimento variabile.

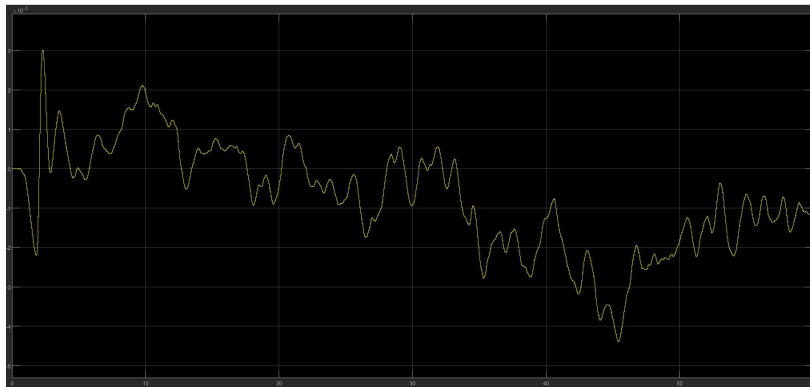


Figure 69: Tracciato asse Y con controllore G, riferimento variabile.

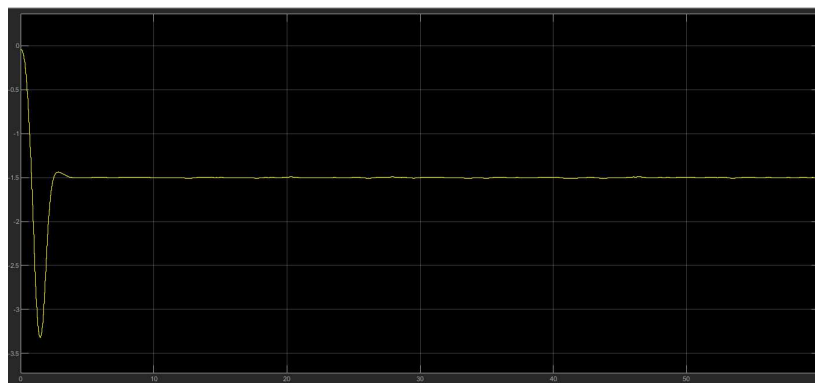


Figure 70: Tracciato asse Z con controllore G, riferimento variabile.

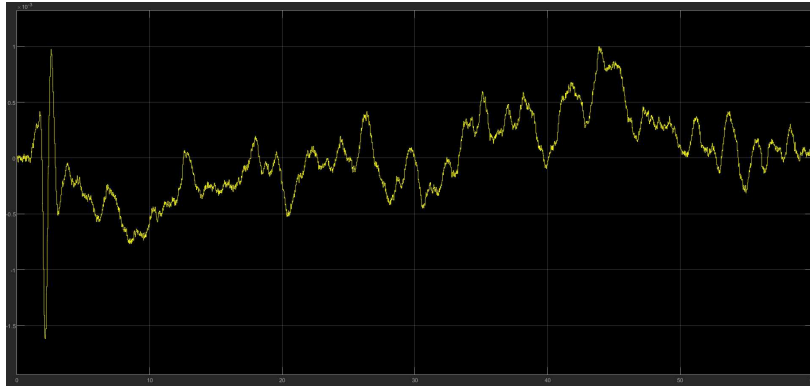


Figure 71: Tracciato angolo di pitch con controllore G, riferimento variabile.

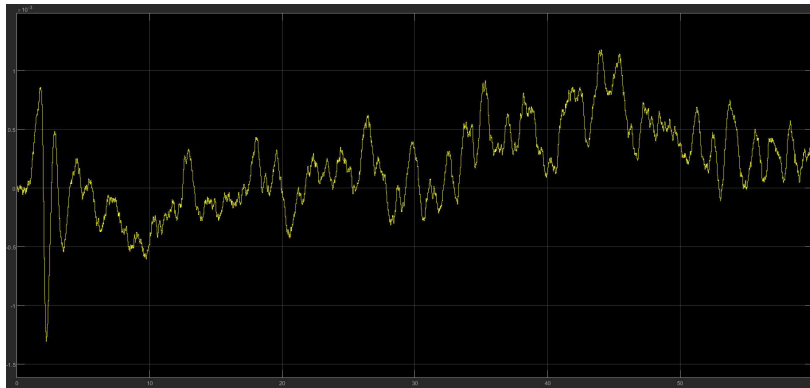


Figure 72: Tracciato angolo di roll con controllore G, riferimento variabile.

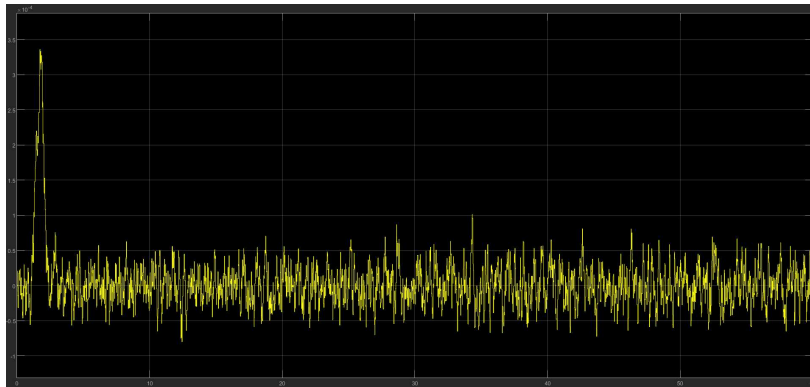


Figure 73: Tracciato angolo di yaw con controllore G, riferimento variabile.

4.2 Sintesi con luogo delle radici

La seconda tecnica utilizzata per la sintesi del controllore, sia per segnali costanti che variabili nel tempo, è stata effettuata utilizzando il luogo delle radici:

anch'esso, come la sintesi in frequenza, correla determinate caratteristiche della FdT del sistema ciclo chiuso con caratteristiche della FdT del sistema in catena aperta. La funzione di trasferimento in c.a. di un sistema è riscrivibile tramite una forma razionale⁷:

$$F(s) = K' \frac{\prod_{i=1}^m (s + z_i)}{\prod_{i=1}^n (s + p_i)} \quad (32)$$

Calcolando ora la FdT a c.c. $W(s)$ è possibile conoscere i poli, essi infatti corrispondono alle n radici del suo denominatore:

$$\prod_{i=1}^n (s - p_i) + K' \prod_{i=1}^m (s - z_i) = 0 \quad (33)$$

Al variare del valore di k' le radici si spostano nel piano complesso definendo il luogo delle radici, il segno di k' definisce invece il luogo positivo o negativo.

Prima di procedere alla sintesi è necessario definire le specifiche che devono essere rispettate dal sistema a ciclo chiuso. Si hanno anche in questo caso due tipi di specifiche, le prime due corrispondono alle specifiche univoche mentre l'ultima è una specifica lasca:

1. sistema di tipo 1;
2. errore a regime permanente $\tilde{e} \leq 0.01$;
3. tutti i poli in catena chiusa abbiano parte reale ≤ 0 (stabilità).

A tali specifiche relative al sistema a c.c. corrispondono determinate specifiche che devono essere soddisfatte dal sistema in catena diretta, ovvero:

1. $F(s)$ ha un polo in $s=0$;
2. $\tilde{e} = k_d^2/k_f \leq 0.01$;
3. tutti i poli in catena aperta abbiano parte reale ≤ 0 (stabilità).

⁷s variabile complessa

Come già spiegato nell'altra tecnica di sintesi, per l'implementazione dei due controllori che verranno successivamente trovati, bisogna effettuare il disaccoppiamento del controllo dei due segnali di pitch e roll nel blocco Attitude (si faccia riferimento alla figura 51).

4.2.1 Riferimento costante

Per forzare lo Switch refAtt a scegliere il riferimento costante proveniente dal blocco Signal Builder è sufficiente, come fatto nella sottosezione 4.1.1, modificare propriamente la porta logica (figura 54).

Avvalendosi dello strumento sisotool si visualizzano questa volta i diagrammi modulo e fase di Bode, la risposta al gradino del sistema ed infine il luogo delle radici. Inserendo la FdT 25 nel blocco G (relativo al processo) del diagramma a blocchi del classico sistema in controreazione è possibile osservare i diagrammi sopra citati, relativi al sottosistema di controllo dell'angolo di roll:

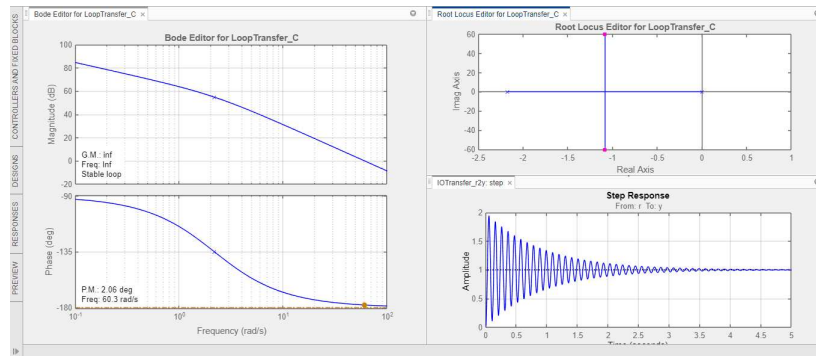


Figure 74: Diagrammi Bode, Luogo delle Radici e risposta al gradino del sistema.

In primo luogo si nota che la risposta al gradino di tale sistema non è accettabile e sicuramente non soddisfa le specifiche desiderate: occorrerà quindi progettare un controllore che migliori sensibilmente la risposta, partendo dal soddisfacimento delle specifiche univoche e proseguendo poi con quella lasca. La prima specifica risulta già soddisfatta in quanto il sistema è di tipo 1 e quindi è sufficiente verificare solo la seconda, per ottenere un controllore di prima istanza \tilde{G} . Conoscendo k_p ed imponendo \tilde{e} non è difficile ricavare k_g , ovvero il guadagno minimo del controllore:

$$\tilde{e} = \frac{k_d^2}{k_f} = \frac{1}{k_f} = \frac{1}{k_g \cdot k_p} \quad (34)$$

e quindi

$$k_g = \frac{1}{k_f \cdot \bar{e}} = \frac{1}{0.01 \cdot 3644} = 0.027 \simeq 0.03 \quad (35)$$

Andando ad inserire il controllore di prima istanza \hat{G} (blocco C) ed aggiornando l'architettura si determinano i diagrammi riportati in figura 75:

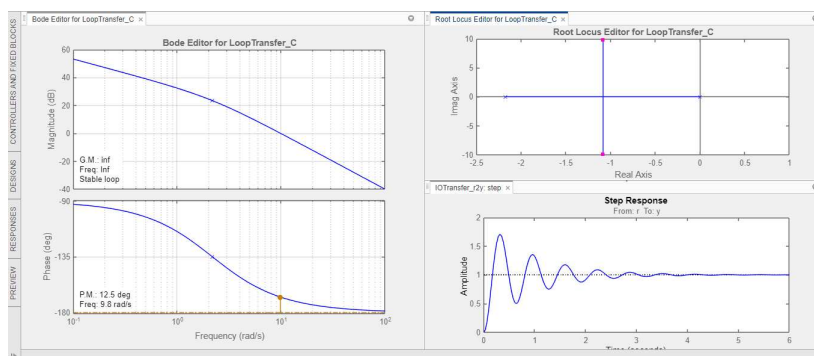


Figure 75: Diagrammi Bode, Luogo delle Radici e risposta al gradino del sistema dopo il guadano statico.

si osserva un buon miglioramento della risposta generale, il tempo di salita è accettabile ma vi sono eccessive sotto e sovraelongazione. Da notare inoltre che la specifica lasca risulta già soddisfatta.

E' comunque necessario proseguire con la sintesi di un controllore adatto allo scopo prefissato che renda più "dolce" la risposta del sistema.

Come primo passo si calcola l' n-m⁸ iniziale del processo P(s) che continuerà a valere anche dopo l'applicazione del controllore G(s):

$$n-m = 2$$

Imponendo che il centro degli asintoti s_0 si trovi in -5 e potendo sfruttare la cancellazione poli/zeri, scegliendo di posizionare lo zero $z_1 = -2.1715$ (è consentito poichè il polo della P(s) si trova dentro la regione descritta dalla specifica lasca), si ottiene:

$$s_0 = \frac{\sum_{i=1}^n p_i - \sum_{i=1}^m z_i}{n - m} = \frac{-2.1715 + p_1 + 2.1715}{2} = -5 \quad (36)$$

⁸n è il numero dei poli di P(s), m è il numero degli zeri di P(s)

Isolando ora p_1 si ha:

$$p_1 = -10$$

Scegliendo quindi un controllore $G(s)$ del seguente tipo:

$$G(s) = 0.03 \frac{(s + 2.1715)}{(s + 10)} \quad (37)$$

ed implementandolo nell'architettura, lo strumento sisotool fornisce i seguenti diagrammi:

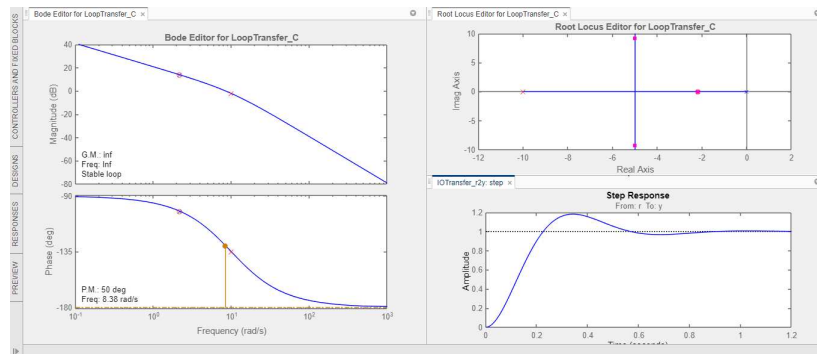


Figure 76: Diagrammi Bode, Luogo delle Radici e risposta al gradino del sistema con il controllore $G(s)$ per riferimento costante.

Il controllore sintetizzato in questo modo è stato in grado di soddisfare le tre specifiche richieste a inizio sezione, inoltre ha una buona sovra elongazione che si aggira attorno al 20%. In particolar modo inserendo $G(s)$ nel blocco $\frac{zeros(s)}{poles(s)}$ di figura 51 vengono forniti i seguenti tracciati della simulazione di volo.

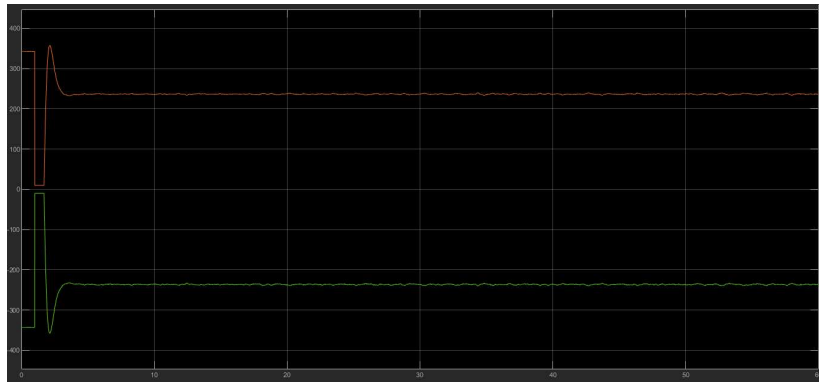


Figure 77: Tracciato motori con controllore G, riferimento costante.

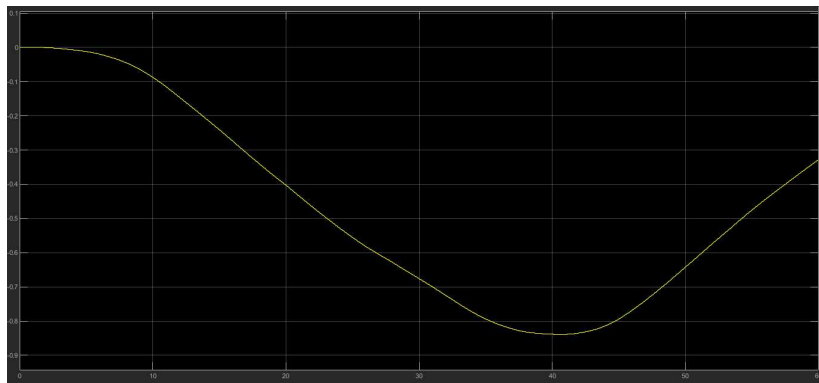


Figure 78: Tracciato asse X con controllore G, riferimento costante.

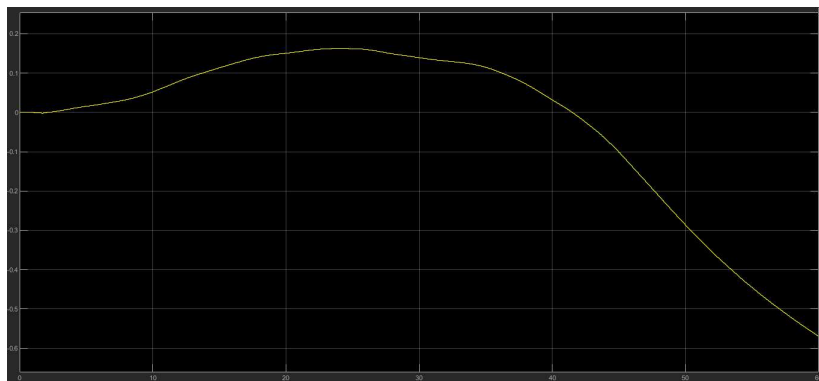


Figure 79: Tracciato asse Y con controllore G, riferimento costante.

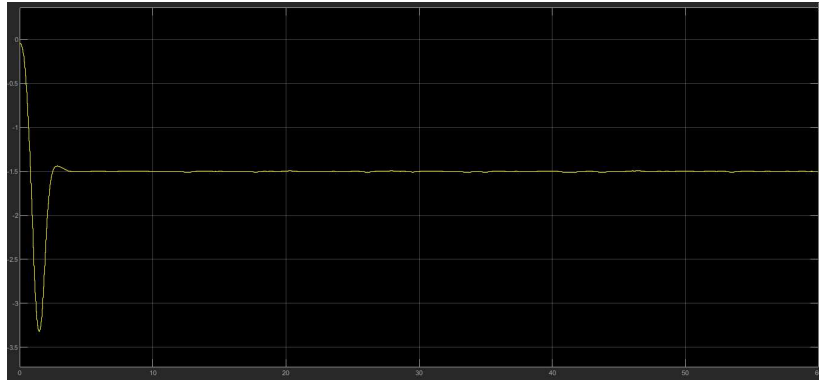


Figure 80: Tracciato asse Z con controllore G, riferimento costante.

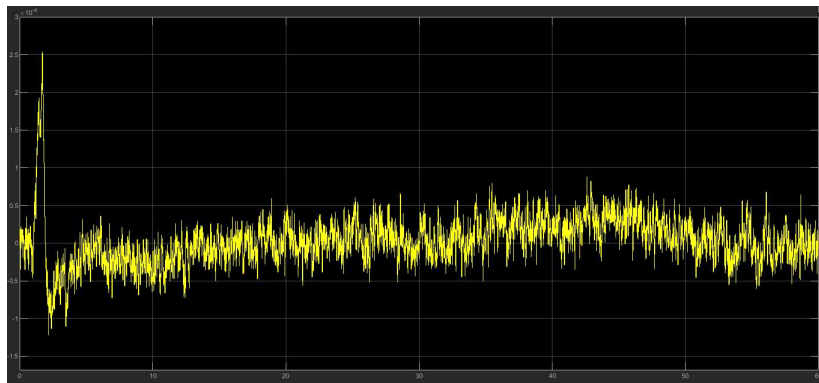


Figure 81: Tracciato angolo di pitch con controllore G, riferimento costante.

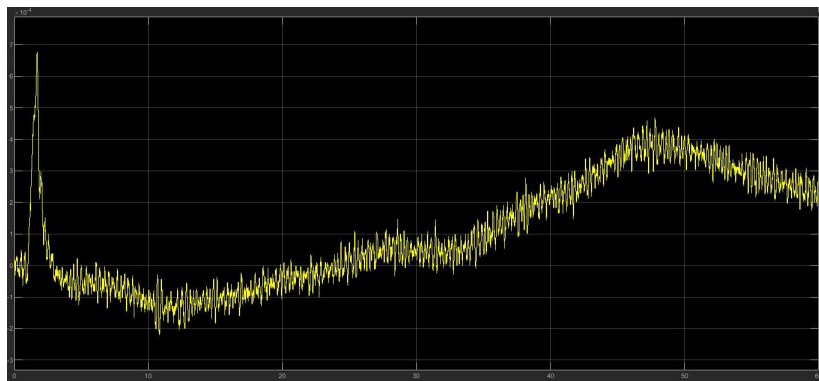


Figure 82: Tracciato angolo di roll con controllore G, riferimento costante.

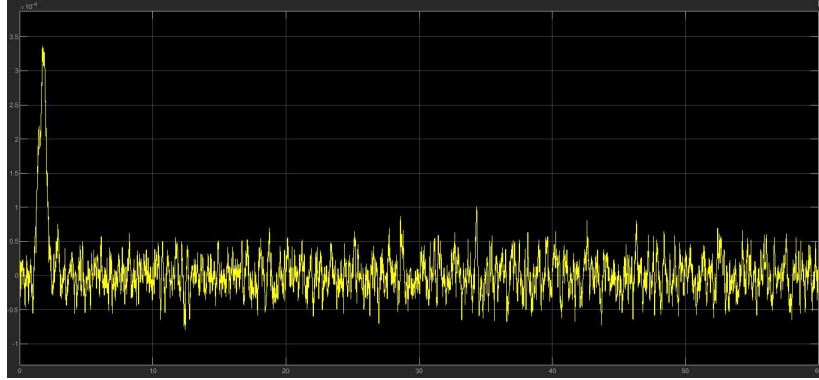


Figure 83: Tracciato angolo di yaw con controllore G, riferimento costante.

4.2.2 Riferimento variabile

Il controllore ottenuto nella sottosezione 4.2.1 non è in grado di effettuare l'azione di asservimento rispetto ad un segnale variabile nel tempo: dalle simulazioni effettuate il drone dopo essersi stabilizzato lungo l'asse Z ha oscillazioni sempre più pronunciate lungo l'asse Y fino a triggerare la flag, terminando così la sua simulazione di volo. Per l'implementazione di un controllore che soddisfi tale richiesta non è stato necessario modificare la porta logica del blocco Signal Builder, come fatto per il segnale costante. La parte di sintesi per le specifiche univoche che è stata già effettuata nella sezione 4.2.1 risulta ancora valida (il guadagno statico rimane lo stesso), di conseguenza si dovranno settare solo gli zeri e i poli del controllore $G(s)$. Dato che il nuovo controllore dovrà essere più sensibile rispetto al precedente il centro degli asintoti verrà spostato verso sinistra nel piano complesso e si utilizzeranno due coppie poli/zero.⁹ Ora imponendo che il centro degli asintoti s_0 si trovi in -30 si ottiene:

$$s_0 = \frac{\sum_{i=1}^n p_i - \sum_{i=1}^m z_i}{n - m} = \frac{-2.1715 + p_1 + p_2 - z_1 - z_2}{2} = -30 \quad (38)$$

Potendo anche in questo caso sfruttare la cancellazione poli/zeri, scegliendo $z_1 = -2.1715$, $z_2 = -0.8285$ e $p_2 = -0.5$ dalla precedente equazione si può facilmente ricavare l'ultimo termine incognito, ovvero p_1 .

⁹vengono utilizzate coppie per mantenere l'n-m iniziale.

$$s_0 = \frac{-2.1715 + p_1 + p_2 - z_1 - z_2}{2} = \frac{-2.1715 + p_1 - 0.5 + 2.1715 + 0.8285}{2} = -30 \quad (39)$$

Da cui:

$$p_1 = -60.3285$$

Quindi il nuovo controllore $G(s)$ sarà così costituito:

$$G(s) = 0.03 \frac{(s + 0.8285)(s + 2.1715)}{(s + 0.5)(s + 60.3285)} \quad (40)$$

Implementandolo assieme al processo $P(s)$ in sisotool, essi forniscono i seguenti diagrammi:

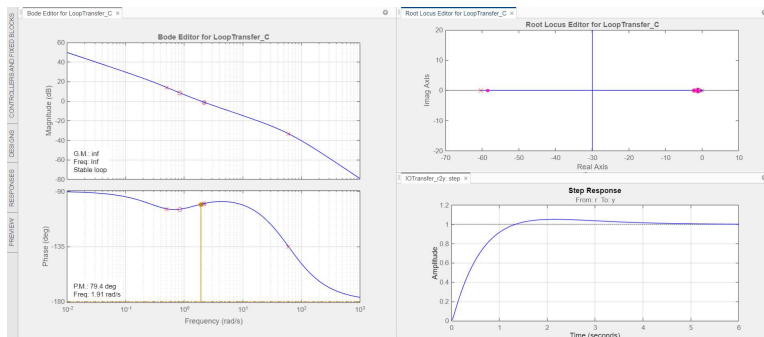


Figure 84: Diagrammi Bode, Luogo delle Radici e risposta al gradino del sistema con il controllore $G(s)$ per riferimento variabile.

Risulta una buona sovra elongazione che si aggira attorno al 6% a fronte di un tempo di salita piuttosto importante di circa 1.4 secondi. I risultati, in ogni caso, soddisfano tutte le specifiche imposte ad inizio sezione e dalle simulazioni di volo effettuate in Simulink si conferma che il controllore è in grado di soddisfare l'azione di asservimento rispetto un segnale variabile nel tempo. Le seguenti figure riportano l'andamento dei PID con il controllore per riferimenti variabili appena sintetizzato.

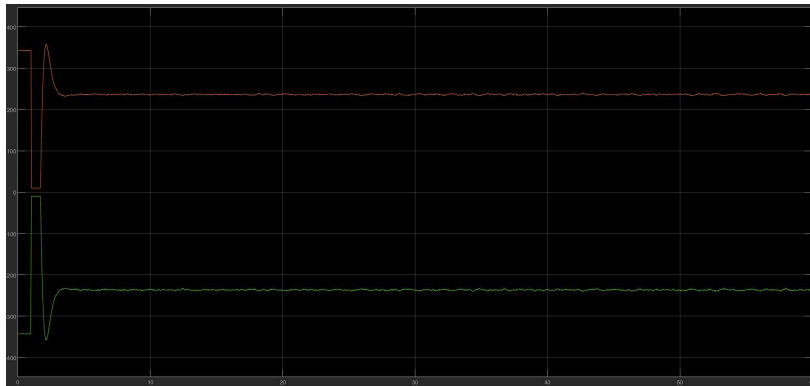


Figure 85: Tracciato motori con controllore G, riferimento variabile.

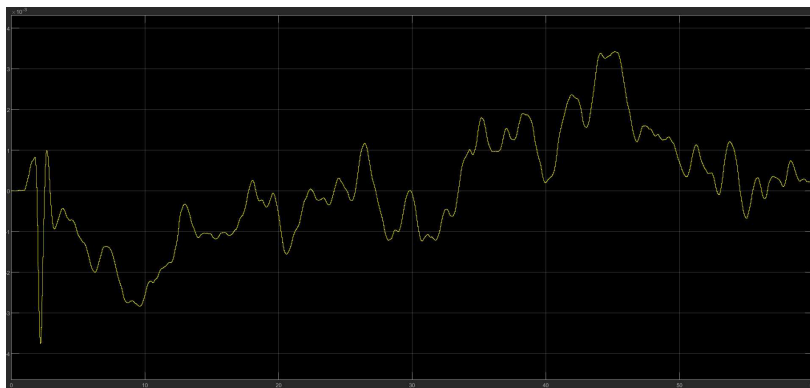


Figure 86: Tracciato asse X con controllore G, riferimento variabile.

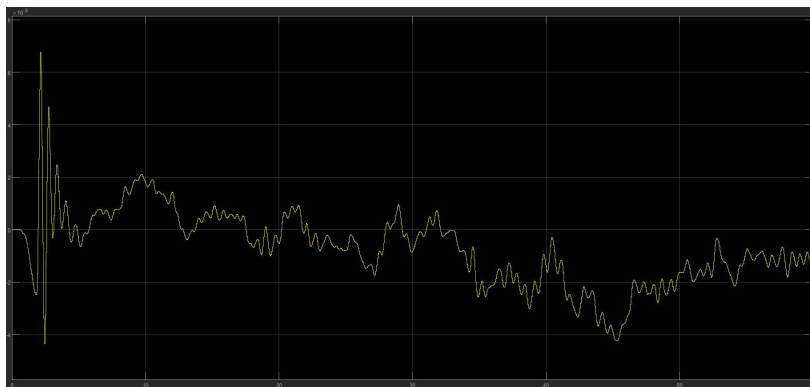


Figure 87: Tracciato asse Y con controllore G, riferimento variabile.

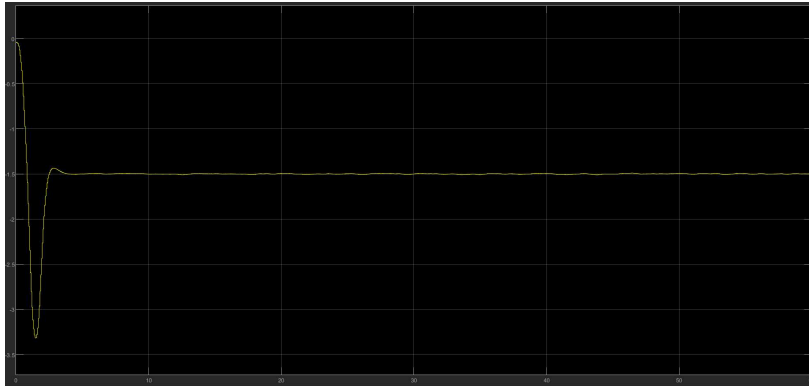


Figure 88: Tracciato asse Z con controllore G, riferimento variabile.

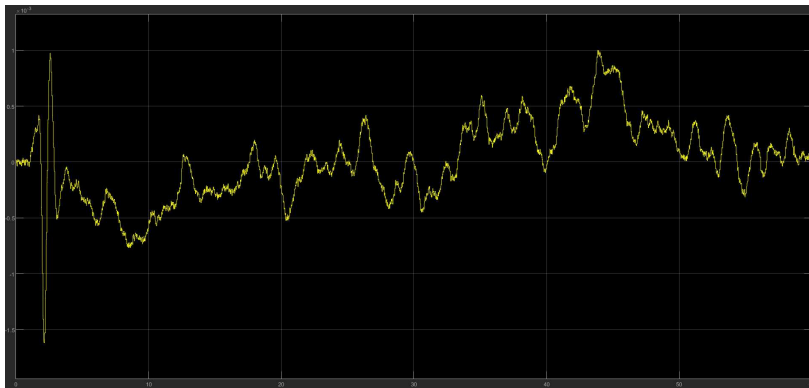


Figure 89: Tracciato angolo di pitch con controllore G, riferimento variabile.

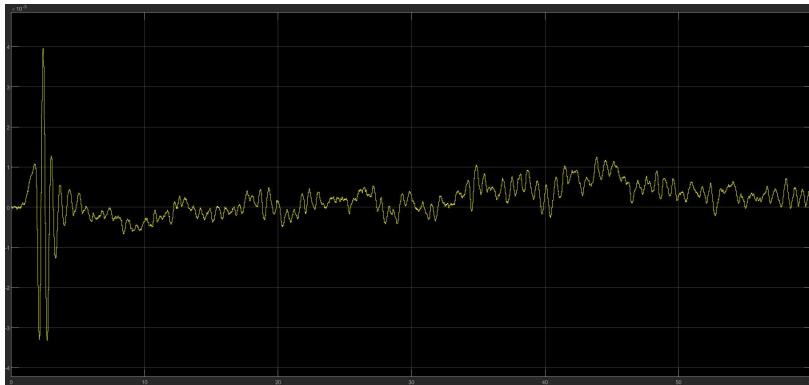


Figure 90: Tracciato angolo di roll con controllore G, riferimento variabile.

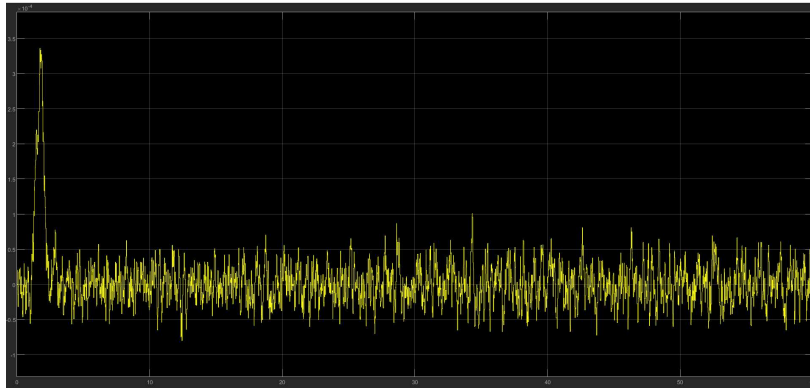


Figure 91: Tracciato angolo di yaw con controllore G, riferimento variabile.

5 Conclusioni

Si è visto come entrambe le tecniche affrontate in questo studio siano state in grado di sintetizzare un controllore per l'angolo di rollio che migliorasse la risposta rispetto al sistema con il PID originale, a fronte sia di ingressi costanti che variabili nel tempo.

- **INGRESSI COSTANTI:**

Come ci si poteva già aspettare, nel caso di ingressi costanti il lavoro di sintesi è risultato meno laborioso rispetto al caso di ingresso di riferimento variabile.

Il diagramma della risposta al gradino del sistema ottenuto con la sintesi in frequenza in figura 57 presenta un tempo di salita di circa 0.14 secondi e una sovra-elongazione massima del 25%.

Invece la risposta del sistema avente il controllore sintetizzato con il luogo delle radici presenta un tempo di salita più elevato che si attesta sui 0.22 secondi, ma una minor sovra-elongazione di circa il 20%. (figura 76)

Dal confronto dei tracciati dei PID relativi all'angolo di roll (figure 63 e 82) una veloce considerazione si può fare sul transitorio che risulta peggiore nel caso della sintesi con il luogo delle radici.

- **INGRESSI VARIABILI:**

Comparando i diagrammi delle risposte al gradino in figura 66 e 84 si nota come il tempo di salita del sistema con il controllore sintetizzato in frequenza è circa 0.2 secondi e risulta nettamente minore rispetto all'altro controllore, il quale riporta un tempo di salita di circa 1.4 secondi. Per quanto riguarda le due sovra-elongazione, esse sono molto simili tra loro ed il loro valore massimo si aggira attorno al 6% del valore di riferimento.

Dal confronto dei tracciati dell'angolo di roll (figure 72 e 90) si osserva che il diagramma ottenuto dalla sintesi in luogo delle radici ha un tempo di assestamento ed uno scostamento massimo dal riferimento maggiore rispetto all'altra tecnica. Ciò è in linea con i risultati ottenuti dallo studio preventivo effettuato in sisotool. (fare attenzione alla scala dell'asse delle ordinate, i due tracciati riportano scale diverse)

Per quanto riguarda il controllo della posizione lungo l'asse Y, effettuato indirettamente attraverso il controllo dell'angolo di rollio, si vede che i tracciati sono molto simili (fare anche qui attenzione alle scale delle ordinate), una piccola differenza è data dal transitorio che nel luogo delle radici risulta leggermente più attenuato e breve. (figure 69 e 87)

Concludendo si può affermare che nel caso specifico del controllo dell'angolo di rollio, la tecnica di sintesi in frequenza ha prodotto un risultato più performante rispetto alla sintesi in luogo delle radici.

References

- [1] https://it.wikipedia.org/wiki/Aeromobile_a_pilotaggio_remoto
- [2] Fum, W. Z. (2015). Implementation of Simulink controller design on Iris+quadrotor.
- [3] Pounds, P., Mahony, R., Corke, P. (2010). Modelling and control of a large quadrotor robot. *Control Engineering Practice*, 18(7), 691-699.
- [4] Musa, S. (2018). Techniques for quadcopter modeling and design: A review. *Journal of unmanned system Technology*, 5(3), 66-75.
- [5] Bouabdallah, S. (2007). Design and control of quadrotors with application to autonomous flying.
- [6] Tesi D. Gambini (2019). Studio e sviluppo in Simulink di tecniche di controllo per minidroni.
- [7] Tesi L. Mezzanotti (2019). Studio e sviluppo di controllori per l'assetto di mini droni.
- [8] Tesi P. Ruggeri (2019). Progetto e sviluppo di un sistema di controllo per un mini drone.