



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA E DELL' AUTOMAZIONE

Studio e sviluppo di sistemi di controllo per diversi gradi di libertà di un drone

**- Study and development of control systems for different degrees of
freedom of a drone -**

Candidato:
Gianmarco Di Renzo

Relatore:
Prof. Gianluca Ippoliti

Correlatore:
Prof. Giuseppe Orlando

Anno Accademico 2021-2022



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA E DELL' AUTOMAZIONE

Studio e sviluppo di sistemi di controllo per diversi gradi di libertà di un drone

**- Study and development of control systems for different degrees of
freedom of a drone -**

Candidato:
Gianmarco Di Renzo

Relatore:
Prof. Gianluca Ippoliti

Correlatore:
Prof. Giuseppe Orlando

Anno Accademico 2021-2022

UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA IN INGEGNERIA INFORMATICA E DELL' AUTOMAZIONE
Via Brezze Bianche – 60131 Ancona (AN), Italy

Mi è doveroso dedicare questo spazio del mio elaborato alle persone che hanno contribuito, con il loro instancabile supporto, alla realizzazione dello stesso. Un sentito grazie al mio relatore Ippoliti per la sua infinita disponibilità e tempestività ad ogni mia richiesta. Grazie per avermi fornito ogni materiale utile alla stesura dell'elaborato. Senza il supporto morale di mia madre, mio padre e mia sorella non sarei mai potuto arrivare fin qui. Grazie per esserci sempre stati soprattutto nei momenti di sconforto e grazie per avermi fatto intraprendere questo percorso di studi, spero in futuro di potermi sdebitare. Un grazie speciale a Gianluca, ormai coinquilino e collega da tanti anni, che ha condiviso con me gioie e fatiche di questo percorso universitario trascorso insieme. Cari amici, non potrò mai ringraziarvi abbastanza per essere stati sempre presenti ogni volta che ho avuto bisogno di voi. Non avrei mai raggiunto questo traguardo senza il vostro sostegno. Grazie di cuore!

A chi ha sempre creduto in me... Grazie!

Sommario

il presente elaborato si propone come obiettivo lo studio di un drone multirottore a 4 rotori (quadricottero), lo sviluppo di una sua modellazione matematica e la sintesi di controllori per il controllo di due dei suoi sei gradi di libertà, esaminandone poi i risultati in simulazione. Si è utilizzato ed analizzato il modello matematico della dinamica di un generico velivolo per poi caratterizzarlo con i parametri del drone in questione. Si passa poi all'implementazione di tale modello all'interno di Simulink, in cui vengono sviluppati e implementati controllori per il controllo della dinamica di Pitch e Roll di un quadricottero attraverso l'impiego di tecniche di controllo lineare, al fine di migliorarne le prestazioni nella risposta ai riferimenti impartiti.

Indice

1	Introduzione	1
2	Descrizione del modello matematico	3
2.1	Aspetti generali	3
2.1.1	Derivazione delle grandezze fisiche fondamentali	4
2.1.2	Matrici di rotazione	5
2.1.3	Roll Pitch Yaw	6
2.1.4	Spinta	7
2.1.5	Forze e Momenti agenti sul quadricottero	7
2.1.6	Manovra di Hovering	10
3	Modellazione di un quadricottero in Simulink	11
3.1	Matlab e SimuLink	11
3.2	Modello <i>Simulink</i>	11
3.3	Modello del processo	13
3.3.1	Modello non lineare	14
3.3.2	Modello lineare	18
3.4	Generatori di segnali	19
3.5	Blocco di controllo	21
3.5.1	Controllo dello Yaw	25
3.5.2	Sistema di Controllo Pitch e Roll	25
3.5.3	Controllo dell'altitudine Z	26
4	Sintesi dei controllori	29
4.0.1	Equazioni matematiche	29
4.1	Linearizzazione del modello	32
4.1.1	Matrici del sistema Pitch	34
4.1.2	Matrici del sistema Roll	37
4.2	Controllori PID	41
4.2.1	Analisi dei PID	41
4.2.2	Analisi delle prestazioni dei PID originali	43
4.3	Tecniche di controllo lineare	46
4.3.1	Sintesi in frequenza	47
4.3.2	Luogo delle radici	49
4.4	Realizzazione del controllore Pitch con la sintesi in frequenza	50
4.4.1	Realizzazione del controllore Pitch con il luogo delle radici	56

Indice

4.5	Realizzazione del controllore Roll con la sintesi in frequenza	61
4.5.1	Realizzazione del controllore Roll con il luogo delle radici . . .	66
4.6	Realizzazione di controllori per la dinamica del Pitch e per la dinamica del Roll senza implementazione del PID	71
4.6.1	Sintesi in frequenza	71
4.6.2	Luogo delle radici	72
5	Conclusioni	75

Elenco delle figure

2.1	Tipologie di configurazione	3
2.2	Earth Fixed Frame, Body Fixed Frame, Vehicle Frame	4
2.3	Yaw Pitch Roll	6
2.4	Tipologie di assetto	7
2.5	Spinta	7
2.6	Forze agenti sul drone	9
2.7	Manovra di hovering	10
3.1	Progetto <i>asbQuadcopter</i>	12
3.2	animazione 3D	12
3.3	Blocco <i>AirFrame</i>	13
3.4	Blocco <i>nonlinear AirFrame</i>	14
3.5	Blocco <i>Bus_setup</i>	15
3.6	Blocco <i>Position_on_Earth</i>	15
3.7	Blocco <i>AC model</i>	16
3.8	Blocco <i>Applied Force Calculation</i>	17
3.9	Blocco <i>Gravity Force Calculation</i>	17
3.10	Blocco <i>Drag Calculation</i>	17
3.11	Blocco <i>Motor Forces and Torques</i>	18
3.12	Blocco <i>Rotor_Dynamics</i>	18
3.13	Blocco Linear	19
3.14	Blocco <i>Command</i>	20
3.15	Blocco <i>Signal Builder</i>	20
3.16	Blocco <i>FCS</i>	21
3.17	Blocco <i>Flight Controll System</i>	22
3.18	Blocco <i>State Estimator</i>	23
3.19	Blocco <i>Flight Controller</i>	24
3.20	Blocco <i>Yaw</i>	25
3.21	Blocco <i>XY to reference orientation</i>	25
3.22	Blocco <i>Attitude</i>	26
3.23	Blocco <i>Gravity Feedforward/Equilibrium Thrust</i>	26
4.1	Matrice A della dinamica (Pitch)	34
4.2	Matrice B degli ingressi (Pitch)	34
4.3	Matrice C delle uscite (Pitch)	34
4.4	Matrice D (Pitch)	35

Elenco delle figure

4.5	Varibili di stato, ingresso e uscite	35
4.6	Costanti moltiplicative	36
4.7	Funzione di trasferimento della dinamica del pitch	37
4.8	Matrice A della dinamica (Roll)	38
4.9	Matrice B degli ingressi (Roll)	38
4.10	Matrice C delle uscite (Roll)	38
4.11	Matrice D (Roll)	39
4.12	Varibili di stato, ingresso e uscite	39
4.13	Costanti moltiplicative	40
4.14	Funzione di trasferimento della dinamica del Roll	41
4.15	Struttura del PID	41
4.16	Segnali di riferimento impartiti	43
4.17	PID originali - X	44
4.18	PID originali - Y	44
4.19	PID originali - Z	45
4.20	PID originali - Pitch	45
4.21	PID originali - Roll	46
4.22	PID originali - Yaw	46
4.23	Diagrammi universali	48
4.24	Diagrammi di bode	51
4.25	Diagrammi di bode - rete anticipatrice	52
4.26	Diagrammi di bode - rete attenuatrice	53
4.27	Carta di Nichols	54
4.28	<i>SystemControlDesigner</i>	54
4.29	Blocco Attitude - Sintesi in Frequenza	55
4.30	Blocco Parameters : Zero-Pole	55
4.31	Sintesi in frequenza - Pitch	56
4.32	Luogo delle Radici - Controllore di primo tentativo	57
4.33	Luogo delle radici-Controllore finale	59
4.34	Luogo delle radici-Controllore finale	59
4.35	Poli del sistema a ciclo chiuso	59
4.36	Blocco Parameters : Zero-Pole	60
4.37	Luogo delle radici- Pitch	60
4.38	Diagrammi di bode Roll	62
4.39	Carta di Nichols	64
4.40	<i>SystemControlDesigner</i>	64
4.41	Blocco Attitude - Sintesi in Frequenza	65
4.42	Blocco Parameters : Zero-Pole	65
4.43	Sintesi in frequenza - Roll	66
4.44	Luogo delle Radici - Controllore di primo tentativo	67
4.45	Luogo delle radici-Controllore finale	69
4.46	Luogo delle radici-Controllore finale	69

4.47 Poli del sistema a ciclo chiuso	69
4.48 Blocco Parameters : Zero-Pole	70
4.49 Luogo delle radici- Pitch	70
4.50 Blocco Attitude - Sintesi in Frequenza	71
4.51 Blocco Parameters : Zero-Pole Sintesi in Frequenza	72
4.52 Quadricottero	72
4.53 Blocco Attitude - Luogo delle Radici	73
4.54 Blocco Parameters : Zero-Pole Luogo delle radici	73

Elenco delle tabelle

4.1	Parametri specifiche lasche	50
4.2	Parametri specifiche lasche Roll	61

Capitolo 1

Introduzione

Nella Prima parte della tesi, vengono approfonditi gli aspetti generali dei quadricotteri, con particolare riferimento alla loro struttura costitutiva ed ai concetti relativi ai fenomeni fisici che ne determinano la dinamica di volo, seguita poi dalla modellazione di un modello matematico di un quadricottero. La seconda parte della tesi introduce all'ambiente Simulink di MATLAB ed analizza nel dettaglio il progetto asbQuadcopter messo a disposizione da MathWork per la modellazione della dinamica completa di un quadricottero. La terza parte della tesi, dopo una breve introduzione sui controllori PID, sulle tecniche della sintesi in frequenza e del luogo delle radici, considera di tutti e sei i gradi di libertà solamente il pitch ed il Roll e ricorre alle tecniche di controllo per la realizzazione di controllori che migliorino l'assetto e la quota rispetto ai PID originali, approfondendo separatamente il grado di libertà Pitch e il grado di libertà Roll, facendoli lavorare contemporaneamente in maniera indipendente su un unico modello Simulink senza far riferimento al PID di default.

Capitolo 2

Descrizione del modello matematico

2.1 Aspetti generali

Il quadricottero è un multirottore dotato di quattro motori, posti alle estremità delle braccia che vanno a formare una croce. I motori adiacenti ruotano nella direzione opposta. Si tratta di un sistema robotico senza pilota a decollo e atterraggio verticale. Utilizzano un sistema di controllo che si basa sulla lettura dei dati provenienti dalla IMU (Inertial Measurement Unit) grazie al quale è possibile un volo stabilizzato. Nella Figura 2.1 sono rappresentate le configurazioni possibili per un quadricottero, cioè a '+' o a 'X'.

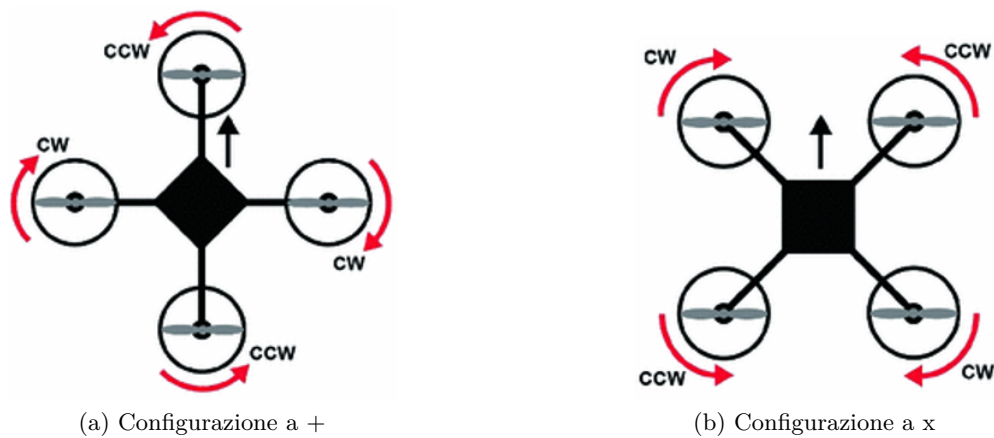


Figura 2.1: Tipologie di configurazione

- Configurazione a +: i 4 rotori sono allineati agli assi x e y del corpo e la direzione di volo corrisponde all'asse creato dai motori CW (in testa) e CCW (in coda)
- Configurazione a X: i 4 rotori sono disallineati di 45° rispetto agli assi x e y del corpo e la direzione di volo corrisponde all'asse creato tra i motori CW e l'asse creato tra i motori CCW.

2.1.1 Derivazione delle grandezze fisiche fondamentali

Un quadricottero in termini generali pu' essere considerato un sistema fisico dotato di 6 gradi di libert'a (DoF:Degrees of Freedom):

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

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

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

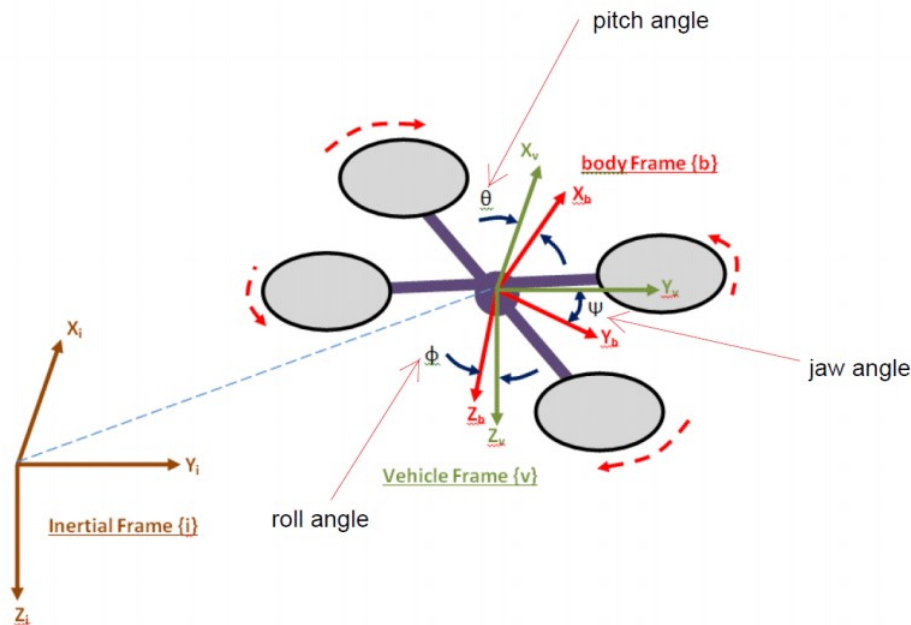


Figura 2.2: Earth Fixed Frame, Body Fixed Frame, Vehicle Frame

Mentre la misurazione dello spostamento lineare del quadricottero rispetto al sistema di riferimento inerziale ‘e ottenibile direttamente, una volta conosciuta la posizione del veicolo; per ottenere lo spostamento angolare rispetto al sistema di riferimento inerziale ‘e necessario fare riferimento a delle ”matrici di rotazione”.

2.1.2 Matrici di rotazione

Queste particolari matrici quadrate descrivono, fissato un asse di rotazione, gli spostamenti angolari ottenibili effettuando rotazioni rispetto agli altri due assi ”liberi”. Gli angoli di Eulero descrivono la rotazione del sistema di riferimento solidale al corpo rigido rispetto a quello inerziale. Si otterranno quindi tre diverse matrici di rotazione :

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{pmatrix} \quad (2.1)$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (2.2)$$

$$R_z(\psi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.3)$$

Una rotazione attorno a tutti e 3 gli assi è data dal prodotto delle 3 matrici precedenti:

$$\begin{aligned} R_{xyz}(\phi, \theta, \psi) &= R_x(\phi)R_y(\theta)R_z(\psi) = \\ &= \begin{pmatrix} \cos(\phi)\cos(\psi) & \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ \sin(\psi)\cos(\phi) & \cos(\theta)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi) & \sin(\theta)\cos(\phi)\sin(\psi) - \sin(\phi)\sin(\psi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{pmatrix} \end{aligned} \quad (2.4)$$

La matrice risultante dal prodotto delle tre matrici sopra definite è la matrice DCM (*Direction Cosine Matrix*) Figura 2.4. Analogamente alla matrice DCM, anche per le trasformazioni angolari è presente una matrice, detta ROT:

$$ROT = \begin{pmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{pmatrix} \quad (2.5)$$

Con l’ausilio delle matrici 2.4 e 2.5 possiamo definire le Eq. 2.6 e 2.7 che rappresentano la conversione dal body frame all’inertial frame del vettore della velocità rispettivamente lineari ed angolari.

$$\dot{P}_E = DCM \cdot V_B \quad (2.6)$$

$$\dot{A}_E = ROT \cdot \omega_B \quad (2.7)$$

Esplicitando i prodotti matrciali si ha:

$$\begin{cases} \dot{x} = u \cos(\theta) \cos(\psi) + v(\sin(\theta) \sin(\phi) \cos(\psi) - \cos(\phi) \sin(\psi)) + w(\sin(\theta) \cos(\phi) \cos(\psi) + \sin(\phi) \sin(\psi)) \\ \dot{y} = u \cos(\theta) \sin(\psi) + v(\sin(\theta) \sin(\phi) \sin(\psi) - \cos(\phi) \cos(\psi)) + w(\sin(\theta) \cos(\phi) \sin(\psi) - \sin(\phi) \cos(\psi)) \\ \dot{z} = -u \sin(\theta) + v(\cos(\theta) \sin(\phi)) + w(\cos(\theta) \cos(\phi)) \\ \dot{\phi} = p + q \tan(\theta) \sin(\phi) + r \tan(\theta) \cos(\phi) \\ \dot{\theta} = q \cos(\phi) - r \sin(\phi) \\ \dot{\psi} = q \frac{\sin(\phi)}{\cos(\theta)} + r \frac{\cos(\phi)}{\cos(\theta)} \end{cases} \quad (2.8)$$

2.1.3 Roll Pitch Yaw



Figura 2.3: Yaw Pitch Roll

Come si evince dalla Figura 2.3 Il rollio (Roll) è il movimento che può far oscillare il quadricottero da destra a sinistra e viceversa. E' ottenuto lasciando invariata la velocità dei motori anteriore e posteriore, e 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 piccoli angoli desiderati, le variazioni di velocità dei rotori saranno circa uguali. Il beccheggio (Pitch) è molto simile al rollio, infatti se quest'ultimo fa ondeggiare il quadricottero da destra a sinistra, il beccheggio lo fa oscillare tra fronte e retro. Come è facile intuire, per ottenere questo movimento l'idea è la stessa del rollio, ma si usano i motori anteriore e posteriore. Quindi si cambiano le velocità dei rotori anteriore e posteriore di una quantità tale da mantenere inalterata la spinta. Per angoli piccoli le due variazioni sono uguali. L'imbardata (Yaw) è probabilmente il movimento più complesso dei tre. L'idea fondamentale è cambiare il momento rispetto all'asse z, immaginato uscente dal piano dei rotori, e mantenere costante la spinta verso l'alto. Per ottenere questo, si aumenta la velocità dei rotori di due motori opposti, e si diminuisce la velocità dei due restanti rotori. In questo modo si avranno, per

esempio, due rotori che girano velocemente in senso orario, e due che girano più lentamente in senso antiorario. Nel complesso ci sarà dunque un momento che fa girare il drone intorno all'asse z. Se le velocità sono opportunamente scelte, non ci sarà un cambiamento della spinta verso l'alto. Come nei casi precedenti, per piccole variazioni desiderate dell'angolo di imbardata, le variazioni di velocità dei rotori sono uguali.

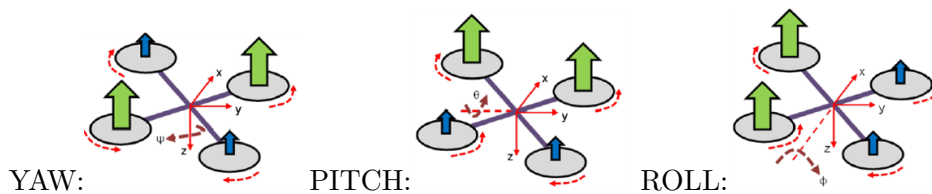


Figura 2.4: Tipologie di assetto

2.1.4 Spinta

L'ultimo movimento fondamentale che può essere controllato è la spinta verso l'alto. Come già accennato, la spinta è legata al quadrato della velocità angolare dei rotori da una costante che dipende dall'aria e da come sono fatti i rotori. Per aumentare la spinta verso l'alto, si aumentano le velocità di tutti i rotori di una stessa quantità, in modo da non creare differenze che possano modificare l'assetto. La figura 2.5 mostra il concetto.

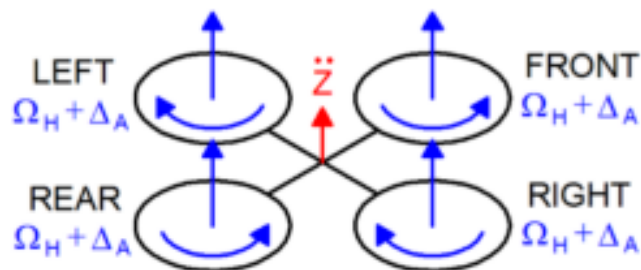


Figura 2.5: Spinta

2.1.5 Forze e Momenti agenti sul quadricottero

Alla formulazione precedente devono essere applicate le forze e i momenti esterni agenti sul velivolo. Ciò che deve essere analizzato nel dettaglio è il modo di cui questi elementi si compongono. Riguardo alle forze, possiamo scomporle in forza peso, dovuta all'attrazione gravitazionale della Terra, e in forza dei motori, che in gergo possiamo definire "spinta". Per il peso si utilizza la formulazione classica: il prodotto massa moltiplicata per la gravità. La direzione di questo vettore, ovviamente, sarà sempre verticale e diretta verso il basso indipendentemente dall'orientamento del

corpo. Per la spinta dei motori si considera una modellazione semplificata dipendente dal quadrato delle velocità di rotazione di ciascun rotore:

$$T = b (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$$

La costante b rappresenta il coefficiente di spinta dei motori ed è un dato del nostro problema. Essendo il tutto calcolato in assi corpo, la direzione della spinta risulterà sempre perpendicolare al piano contenente il drone. Per il peso, il vettore avrà componenti diverse variabili invece con le possibili rotazioni del quadricottero, in particolare con la matrice DCM. Le forze esterne quindi sono sintetizzabili in:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = F_{ext} = DCM \cdot \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}$$

L'analisi delle coppie generate è stata effettuata guardando l'influenza di ciascun tipo di manovra intorno ad un asse a partire dalla forza generata come per la spinta verticale. Moltiplicando le forze per il braccio agente, ovvero la distanza tra punto di applicazione della forza e baricentro corrispondente per approssimazione al semiasse della scocca, si ottengono momenti agenti su diversi assi. Anche in questo caso è evidente la dipendenza quadratica del valore delle coppie al variare dei giri.

$$\begin{aligned} u_2 &= b (\omega_4^2 - \omega_2^2) \\ u_3 &= b (\omega_3^2 - \omega_1^2) \\ u_4 &= d (\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \end{aligned}$$

La costante presa in considerazione questa volta è anche b , costante di coppia della motorizzazione scelta, ottenibile allo stesso modo di d . In quanto ogni manovra agisce con una rotazione attorno ad un asse, parimenti ad un momento, il vettore che rappresenta i momenti esterni può essere scritto in funzione delle manovre stesse.

$$\begin{aligned} M_x &= l \cdot u_2 = l \cdot b (\omega_4^2 - \omega_2^2) \\ M_y &= l \cdot u_3 = l \cdot b (\omega_3^2 - \omega_1^2) \\ M_z &= l \cdot u_4 = l \cdot d (\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \end{aligned}$$

Tipicamente, le forze create durante il funzionamento del drone sono disposte come illustrato di seguito:

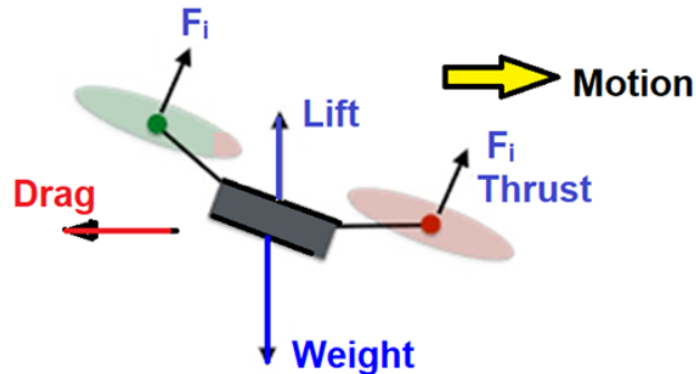


Figura 2.6: Forze agenti sul drone

- **Weight**
A causa della massa del drone, la forza della massa corporea agisce sempre nella direzione della gravità. Maggiore è il peso del drone, più potenza è necessaria per sollevare e spostare il drone.
Peso del drone = $m \cdot g$, dove m è la massa del drone e g è l'accelerazione di gravità.
- **Thrust**
La forza che agisce sul drone nella direzione del movimento è chiamata spinta. Tuttavia, per la dinamica dei droni, è normale per il piano del rotore. Durante il volo stazionario, la spinta è puramente verticale. Se la spinta è inclinata, il drone si inclinerà in avanti o all'indietro. Questa forza è essenziale per muovere il drone nella direzione desiderata alla stessa velocità. Per ottenere il movimento desiderato, a due eliche è stata assegnata l'alta velocità.
- **Lift**
La forza verticale che agisce sul drone è chiamata Lift. Questa forza è dovuta alle differenze di pressione attraverso il drone (in direzione verticale). Quindi, la velocità, le dimensioni e la forma della pala dell'elica determinano la quantità di forza di sollevamento. Il Lift è essenziale per sollevare il corpo contro la gravità. Per creare questa forza, tutte e quattro le eliche corrono ad alta velocità per sollevare il drone. Il Peso è bilanciato dal Lift.
- **Drag**
La forza che agisce sul drone nella direzione opposta al movimento a causa della resistenza dell'aria è chiamata resistenza. Ciò può essere dovuto alla differenza di pressione e alla viscosità dell'aria. Per ridurre la resistenza, viene selezionata la forma aerodinamica del drone. Il Drag è bilanciato dalla spinta.

Se durante il volo costante e livellato, la spinta viene aumentata, l'aereo inizierà ad accelerare nella direzione della spinta (inizierà a guadagnare velocità). L'aumento della velocità comporterà un aumento della resistenza.

Nella situazione inversa, quando durante il volo costante e livellato la spinta viene ridotta, l'aereo inizierà ad accelerare nella direzione della resistenza (la velocità inizierà a diminuire). La diminuzione della velocità comporterà una diminuzione della resistenza.

2.1.6 Manovra di Hovering

Il volo a punto fisso, oppure hover, rappresenta lo stazionamento in volo a velocità nulla di un velivolo. Come prima citato, il bilancio delle forze è dato semplicemente da: - forza peso perpendicolare al piano appartenente al quadricottero diretta verso il basso; - spinta data dai quattro motori diretta allo stesso modo della forza peso ma verso l'alto. Non dimentichiamo che i motori sono controrotanti a due a due: questo significa che la spinta dovuta alla rotazione delle pale delle eliche può, oltre ad eguagliare il peso, generare momenti intorno agli assi di rotazione del mezzo. Per l'annullamento delle coppie occorrerà rendere nulli i termini u_2 , u_3 , u_4 . L'unica opzione è far girare i motori alla stessa velocità di rotazione. Nella simulazione, di 10s, si otterrà dalla posizione di partenza, uno stazionamento così descritto. Questo rappresenta il modello più semplice da simulare. Tutte le condizioni iniziali quali sono angoli di assetto, velocità di rotazione e traslazione sono nulle. ragged2e

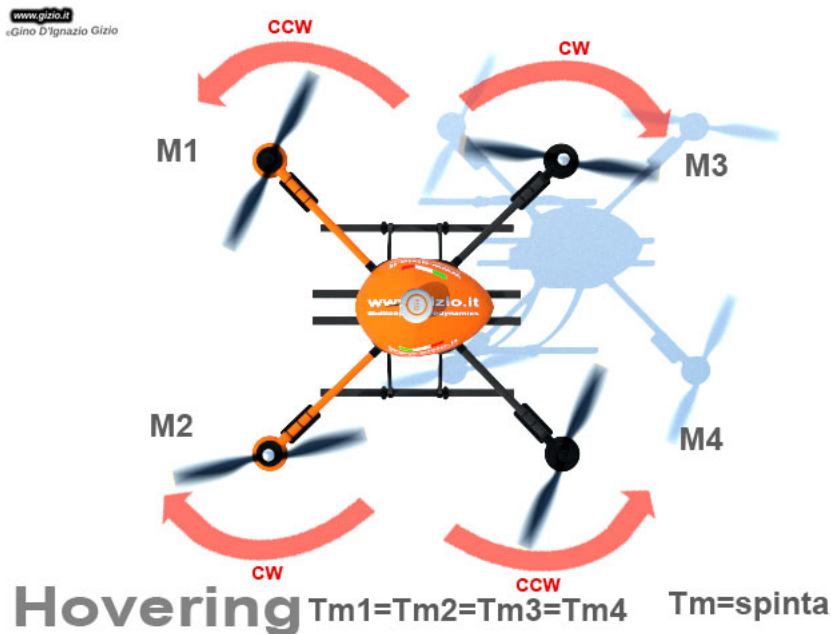


Figura 2.7: Manovra di hovering

Capitolo 3

Modellazione di un quadricottero in Simulink

Nel presente capitolo ci si propone di presentare il progetto *asbQuadcopter* di Simulink per la modellazione di un quadricottero ed analizzarne tutti i blocchi che lo costituiscono.

3.1 Matlab e SimuLink

L'ambiente Matlab permette di lavorare in ambienti matematici complessi gestendo i dati come se questi fossero matrici. La presenza di strumenti specializzati, tramite librerie, per ogni tipo di approccio ingegneristico ne rende lo strumento ideale per la risoluzione di calcoli numerici. Simulink è un'estensione di Matlab che tramite l'adozione di interfaccia per l'utente tramite schema a blocchi modella e simula come Matlab stesso sistemi di ogni tipo. Si è scelto perciò di adottarli per la risoluzione delle equazioni del moto da noi utilizzate e per simulare manovre di base effettuando un confronto tra i due strumenti messi a disposizione. Si tratta di un sistema che presenta forte dipendenza dalle funzioni trigonometriche quindi è sicuramente di tipo non lineare.

3.2 Modello *Simulink*

Il modello Simulink del minidrone Mambo, fornito all'interno del **Simulink Support Package for Parrot Minidrones**, si compone di diversi blocchi, analizzabili separatamente. La struttura presentata è quella tipica di un sistema di controllo in controeazione, in base alle funzioni svolte si possono distinguere:

- *Signal Builder*: generatore dei segnali di riferimento
- *Flight Control System*: controllore del processo
- *Non Linear Airframe*: modello del processo
- *Environment*: blocco contenente le variabili di ambiente

Capitolo 3 Modellazione di un quadricottero in Simulink

- *Sensor*: blocco contenente le costanti di trasduzione e le relazioni tra i diversi sensori presenti a bordo
- *Visualization*: blocco ausiliario che consente la visualizzazione dei valori delle variabili di stato e dei tracciati dei segnali di controllo

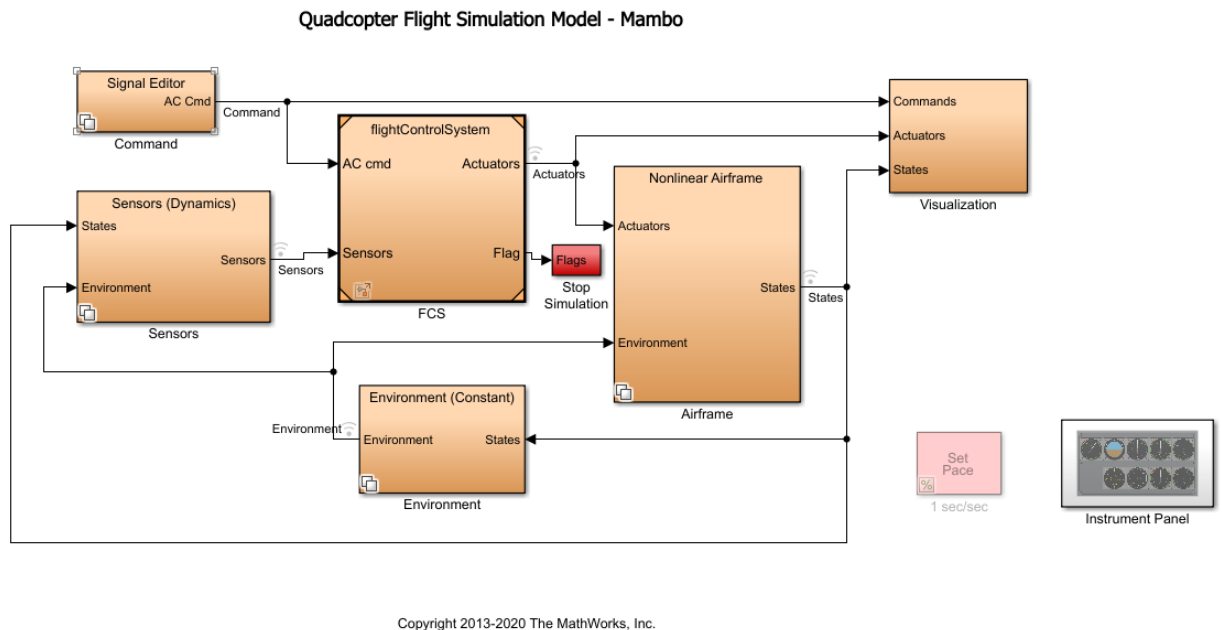


Figura 3.1: Progetto *asbQuadcopter*

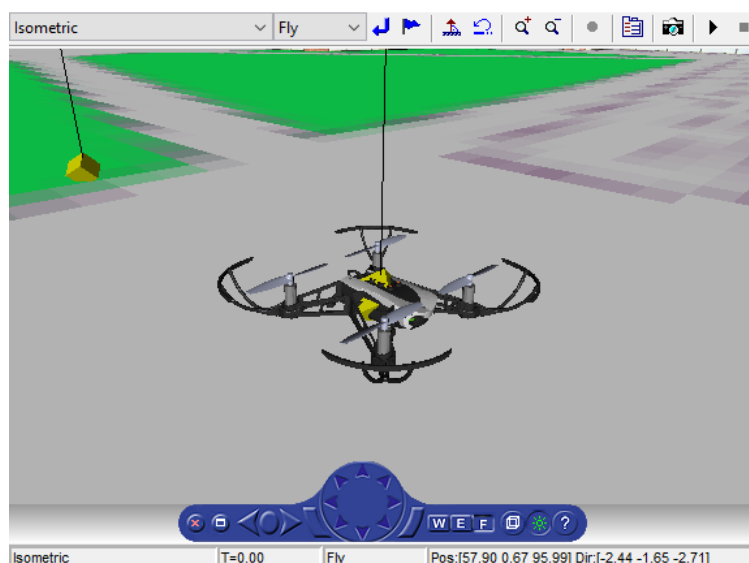


Figura 3.2: animazione 3D

Dopo la creazione del progetto, si apriranno due finestre contenenti rispettivamente il progetto *Simulink* complessivo riportato in Figura 3.1 e l'animazione 3D utilizzata per visualizzare le prestazioni del drone in simulazione riportata in Figura 3.2.

3.3 Modello del processo

Come anticipato prima il processo da controllare si trova nel blocco *AirFrame*, all'interno del quale ci sono due sottoblocchi *linearAirframe* e *nonlinearAirframe*, come si vede dalla Figura che rappresentano rispettivamente il modello lineare e non lineare del quadricottero. È possibile scegliere che tipo di modello utilizzare tramite la variabile *VSS_VEHICLE*:

- = 0 sarà selezionato il modello lineare
- = 1 sarà selezionato il modello non lineare

essa è inizializzata di default ad 1 e caricata sul Workspace automaticamente durante l'avvio del progetto *asbQuadcopter* assieme a tutte le altre variabili necessarie per il suo funzionamento.

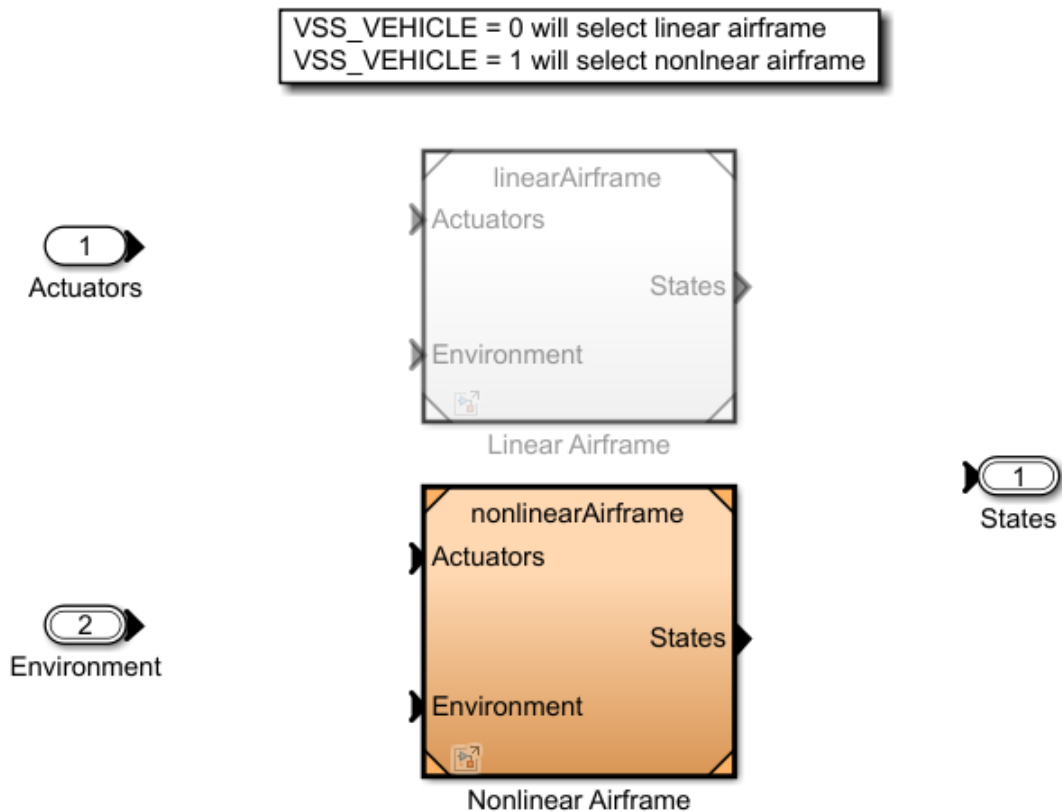


Figura 3.3: Blocco *AirFrame*

3.3.1 Modello non lineare

Entrando all'interno del Nonlinear Airframe possiamo vedere la configurazione del modello non lineare, il quale, come mostrato in figura Figura 3.4, è suddiviso in 4 blocchi principali:

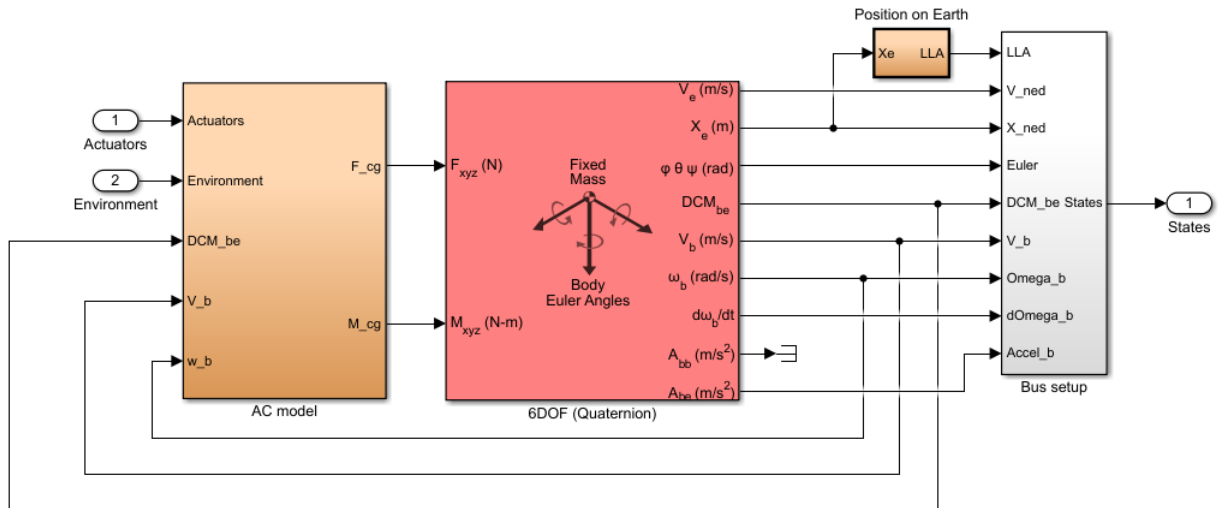


Figura 3.4: Blocco *nonlinear AirFrame*

- *AC model* è il blocco, contenente tutti risultati della modellazione matematica sviluppata nel capitolo 2. Che calcola forze e momenti angolari, applicati al drone rispetto a tutti e 3 i suoi assi, partendo dall'azione di spinta degli attuatori, dalle variabili d'ambiente, dalla matrice DCM_{be} e dalle variabili V_b e ω_b del Body Frame.
- *6DOF (Quaternion)* che prende in ingresso le forze e i momenti applicati al corpo, e restituisce i valori aggiornati di:
 - V_e : velocità lineare nella terna inerziale
 - X_e : posizione nella terna inerziale
 - ϕ, θ, ψ : angoli di assetto
 - DCM_{be} : matrice per la conversione di coordinate dall'Inertial Frame al Body Frame
 - V_b : velocità lineare nella terna Body Frame
 - ω_b : velocità angolare nella terna Body Frame
 - $\frac{d\omega_b}{dt}$: accelerazione angolare rispetto al Body Frame
 - A_{bb} : accelerazione traslazionale lungo gli assi del Body Frame espressa rispetto al Body Frame stesso

- A_{be} : accelerazione traslazionale lungo gli assi del Body Frame espressa rispetto all’Inertial Frame
- *Bus setup* è il blocco che, date in ingresso tutte le componenti che caratterizzano lo stato del sistema, raggruppa quest’ultime all’interno di un unico vettore che viene retroazionato al controllore così da rendere noto a quest’ultimo lo stato del quadricottero in volo.
- *Position on Earth* è il blocco che, date la posizione X_e del quadricottero nello spazio rispetto all’Inertial Frame e l’altezza di riferimento che intercorre tra la superficie effettiva della Terra ed il suo sistema di riferimento piatto (FEF), stima il valore di latitudine, longitudine ed altitudine geodetiche (LLA).



Figura 3.5: Blocco *Bus_setup*

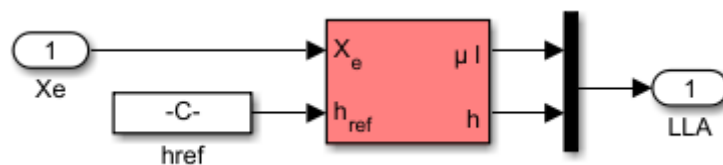


Figura 3.6: Blocco *Position_on_Earth*

Entriamo all'interno del blocco *AC model* come notiamo dalla Figura 3.7 sono presenti altri 4 blocchi:

- *Gravity Force Calculation* : calcola i 3 contributi della forza di gravità, prendendo in ingresso la matrice DCM_{be} e moltiplicando tale matrice per il prodotto tra la costante di gravità g e la massa del drone, come riportato in Figura 3.9 .
- *Drag Calculation* : calcola i 3 contributi dell'attrito viscoso riportato in figura 3.10b, questo dipende dal coefficiente di drag C_d , dalla densità ρ dell'aria, dall'area A della superficie del drone che impatta il fluido durante il volo e dalla sua velocità lineare V_b .
- *Motor Forces and Torques* : calcola tutte le forze traslazionali e rotazionali applicate al corpo del drone rispetto al Body Frame. Figura 3.11
- *Applied Force Calculation* : calcola il contributo totale alla dinamica del drone dato da tutte le forze lineari che agiscono su di esso ovvero dai 3 contributi dati da gravità, attrito viscoso e forze generate dai motori. Figura 3.8 . Questo blocco all'interno contiene un sotto-blocco For-Each attraverso il quale vengono calcolate tutte le forze e momenti esercitate da ciascun motore del drone. Vedi Figura 3.12

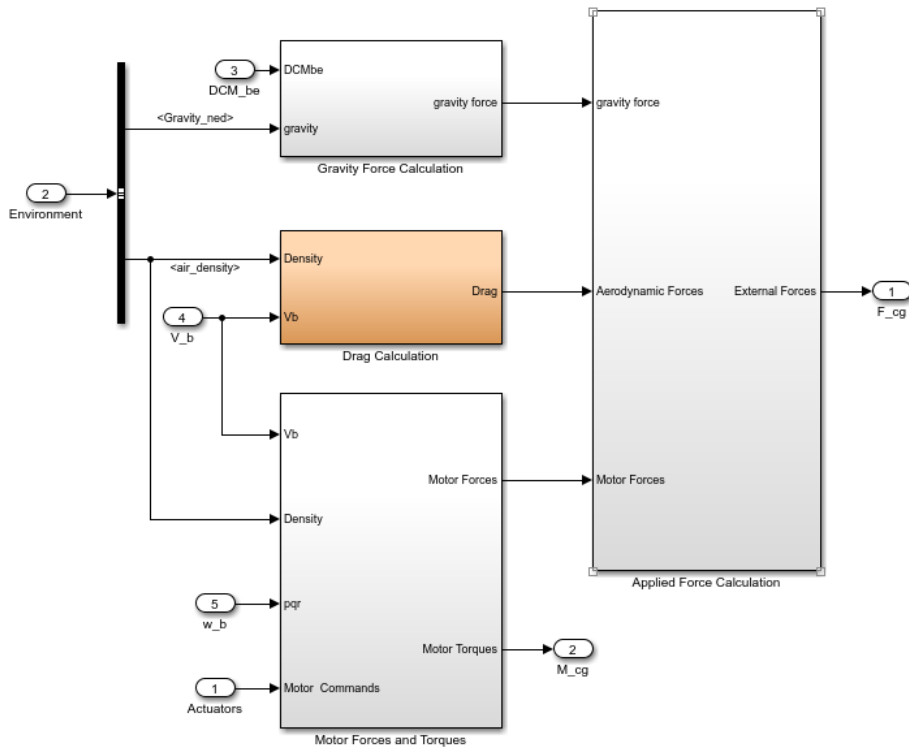


Figura 3.7: Blocco *AC model*

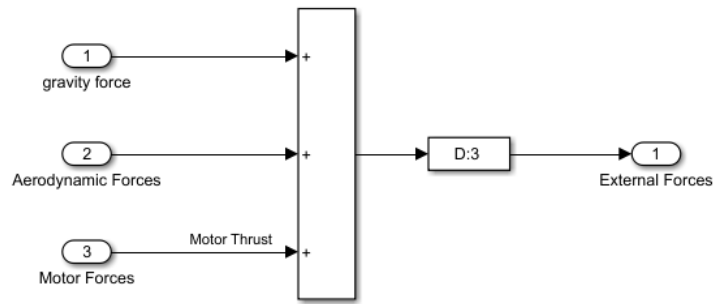


Figura 3.8: Blocco *Applied Force Calculation*

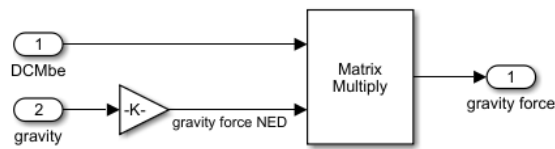


Figura 3.9: Blocco *Gravity Force Calculation*

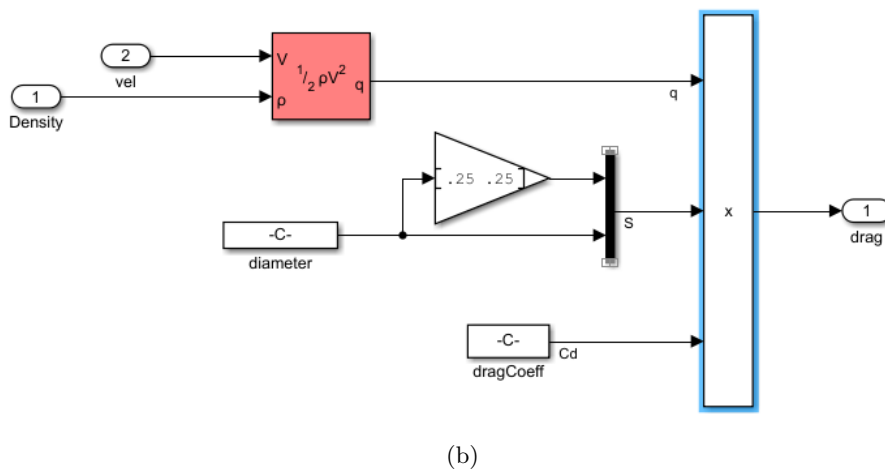
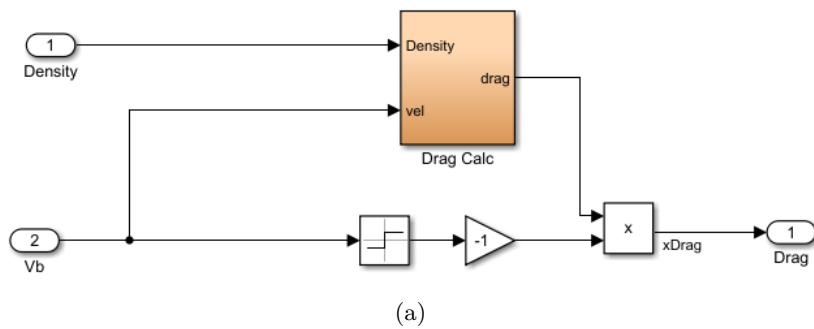


Figura 3.10: Blocco *Drag Calculation*

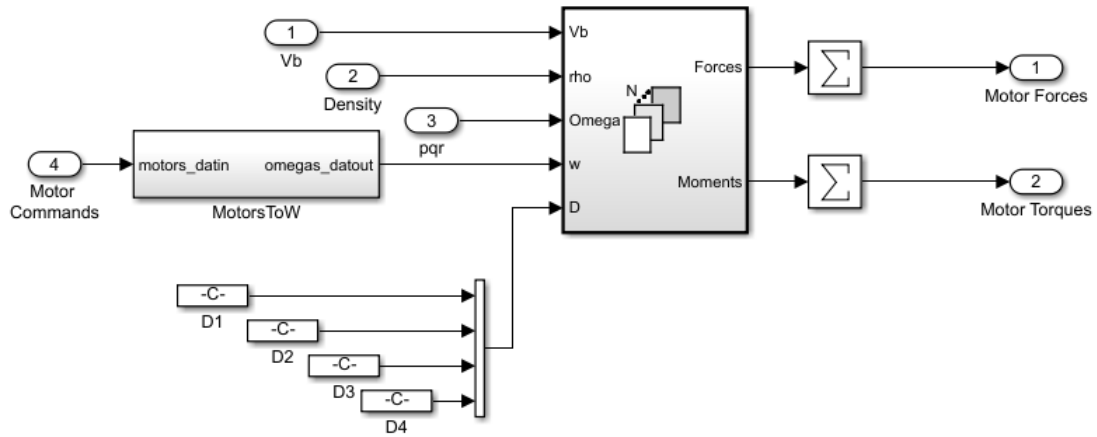


Figura 3.11: Blocco *Motor Forces and Torques*

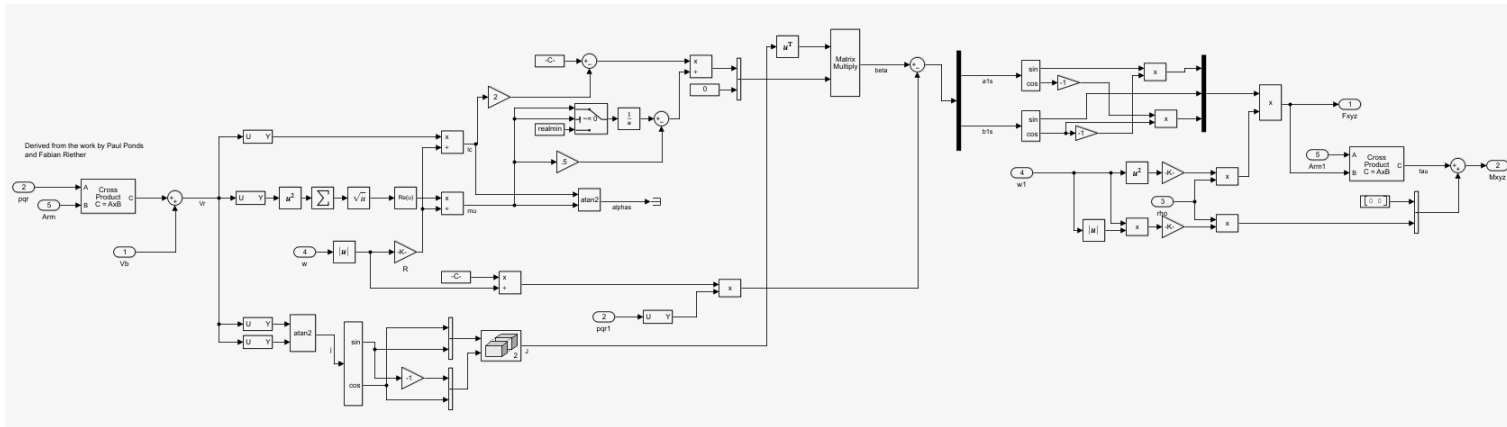


Figura 3.12: Blocco *Rotor_Dynamics*

3.3.2 Modello lineare

All'interno del blocco *Linear Airframe*, riportato nella Figura 3.13, è presente una versione linearizzata di tale modello, rappresentata tramite l'espressione in spazio di stato del sistema.

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (3.1)$$

Nel capitolo 4 viene mostrato come determinare le matrici A, B, C e D aprendo la struttura *linsys*, caricata sul Workspace di MATLAB all'apertura del progetto *asbQuadcopter* dal file *linearizedAirframe.mat* contenuto all'interno della cartella *linearAirframe* e successivamente si progettano dei controllori tramite sintesi in frequenza e con luogo delle radici (tecniche di controllo lineare) da applicare per il controllo del modello non lineare del quadricottero.

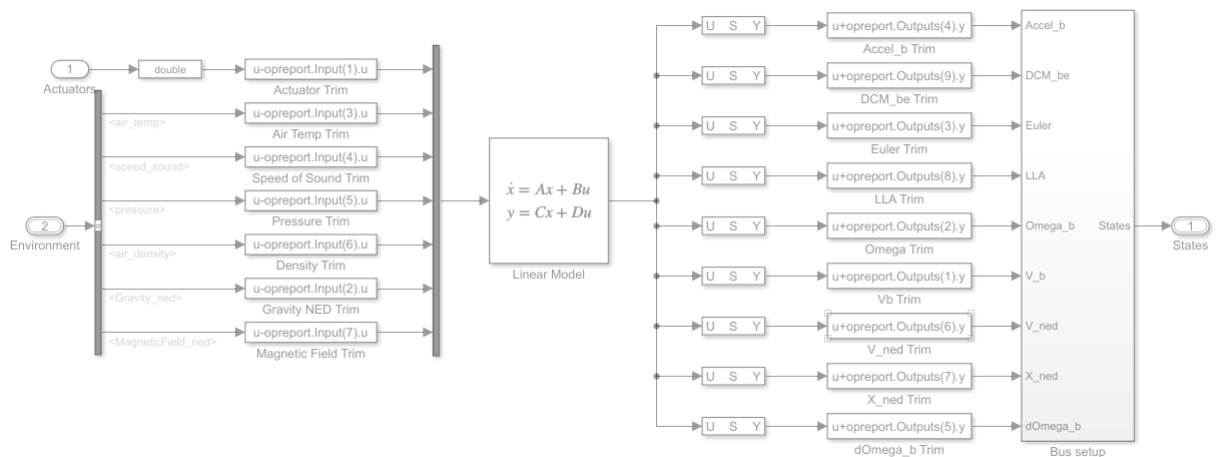


Figura 3.13: Blocco Linear

3.4 Generatori di segnali

Il blocco *Command* consente di scegliere la fonte dei segnali riferimento per il sistema di controllo del drone. Modificando il valore della variabile *VSSCommand* è possibile selezionare:

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

Selezionando l'opzione *Signal Builder* è possibile analizzare la struttura interna del sistema. Si può notare come i segnali X;Y;Z vadano a fornire il riferimento di posizione mentre i segnali *Yaw,Pitch,Roll* costituiscano il riferimento in orientamento (evidenziati in blu nell'immagine). Particolare attenzione va prestata all'analisi della porta logica AND (in rosso in figura) essa consente, qualora i riferimenti di Pitch e Roll siano entrambi uguali a zero, di entrare in modalità di controllo della posizione; in caso contrario si passa in modalità controllo dell'orientamento. Vedi Figura 3.15

VSS_COMMAND = 0 will select Signal Editor
 VSS_COMMAND = 1 will select Joystick + Signal Editor
 VSS_COMMAND = 2 will select pre-saved data from .mat file
 VSS_COMMAND = 3 will select pre-saved data from a spreadsheet

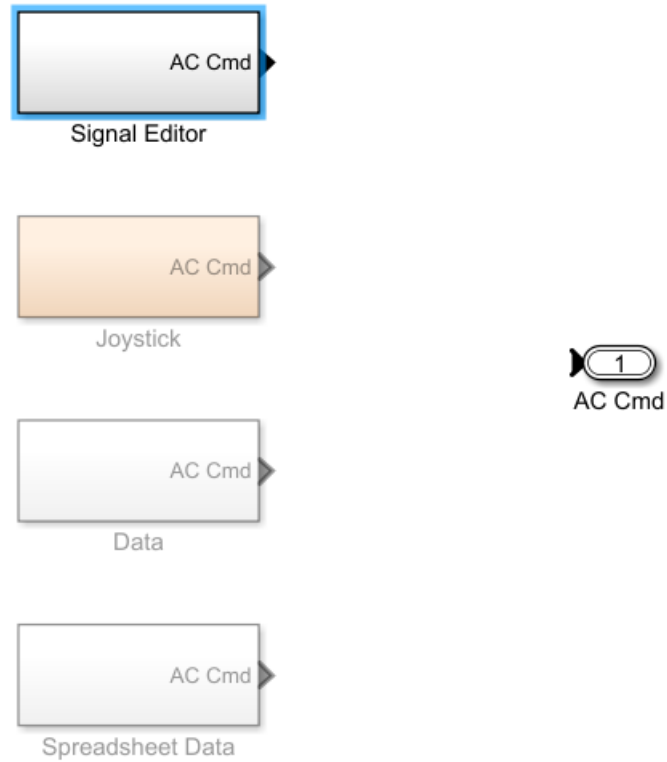


Figura 3.14: Blocco *Command*

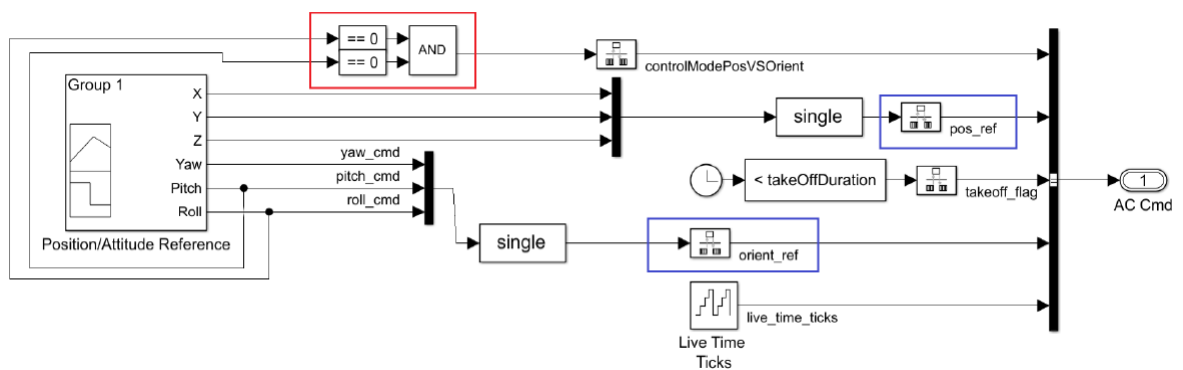


Figura 3.15: Blocco *Signal Builder*

Per quanto riguarda l'uscita del blocco *Signal Builder* troviamo il vettore *AC Cmd* composto da:

- *controlModelPosVSOrient* : come spiegato sopra se i riferimenti di Pitch e Roll sono entrambi uguali a 0 impone al drone di seguire dei riferimenti di Pitch e

Roll tali da fargli rispettare anche i riferimenti impartiti per le coordinate X e Y

- *pos_ref* : riferimenti per le coordinate X, Y e Z
- *takeoff_flag* : variabile booleana che rimane a 'true' se il valore del clock di sistema risulta essere inferiore alla variabile *takeOffDuration* del Workspace, quando passa a 'false' invia un segnale al controllore del drone per far seguire a quest'ultimo i riferimenti della Z generati dal blocco *Position/Attitude Reference*.
- *orient_ref* : riferimenti per Yaw, Pitch e Roll
- *live_time_ticks* : il quale, a una determinata frequenza specificata, emette in uscita una sequenza di valori crescenti che partono da 0 ed arrivano al valore massimo rappresentabile attraverso il numero specificato di bit a disposizione, per poi ripartire da 0

3.5 Blocco di controllo

Il blocco FCS (*Flight Control System*), vedi Figura 3.16, è uno dei componenti fondamentali del modello Simulink e su di esso si è concentrata l'attività di sviluppo del sistema di controllo. Esso riceve come segnali di ingresso i valori dei riferimenti provenienti dal blocco Signal Builder e i valori delle variabili di stato provenienti dai sensori in retroazione. In uscita troviamo i segnali di riferimento degli attuatori che effettueranno il controllo sui motori del modello non lineare. L'uscita Flags è un'uscita di sicurezza che, in caso di perdita di assetto da parte del drone, fornisce un segnale di controllo che interrompe la simulazione.

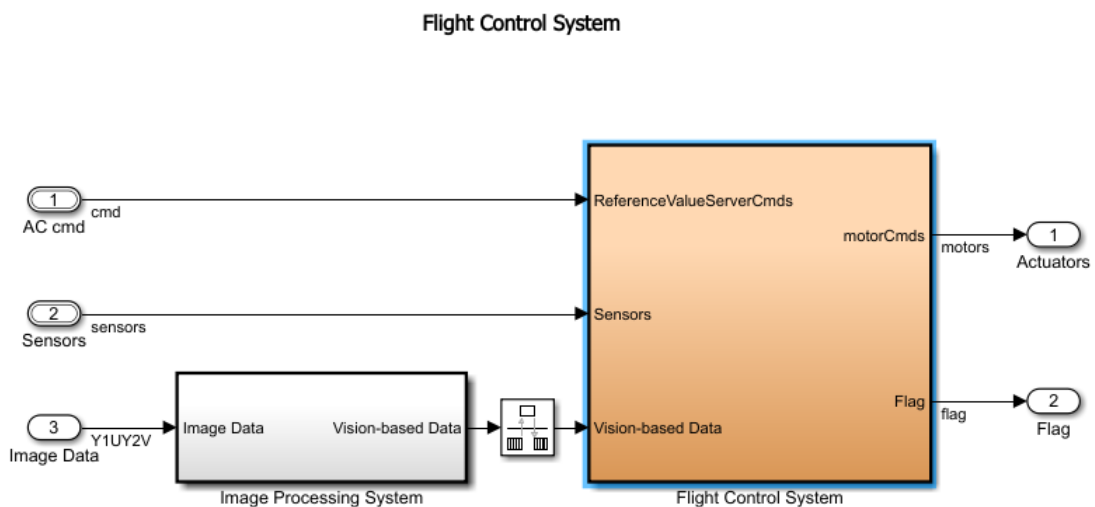


Figura 3.16: Blocco *FCS*

il blocco FCS è piuttosto complesso in quanto si suddivide internamente in tre sottosistemi principali (evidenziati in figura 3.17) che svolgono funzioni diverse:

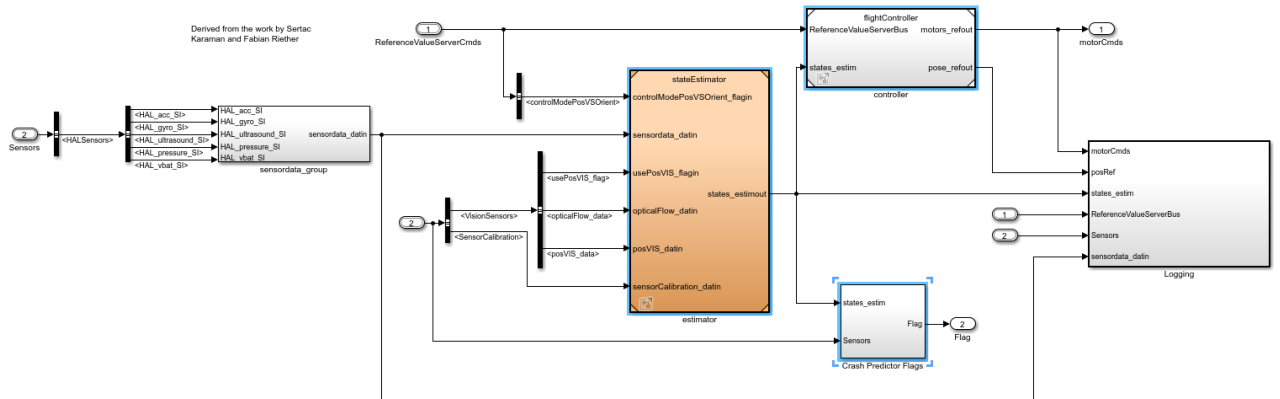


Figura 3.17: Blocco *Flight Control System*

- il blocco *State Estimator* : riceve ed elabora i segnali provenienti dai diversi sensori presenti sul veicolo e fornisce tali informazioni al blocco *Flight Controller*;
- il blocco *Flight Controller* : effettua propriamente l'azione di controllo fornendo in uscita i segnali di controllo diretti ai motori.
- il blocco *Crash Predictor Flag* : elabora le informazioni fornite direttamente dai sensori e dal blocco *State Estimator* e determina, confrontandole con dei parametri pre-assegnati, se sussistono le "normali" condizioni di volo.

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

Sensordata group svolge una pre-elaborazione (ridenominazione delle variabili) sui dati forniti dai vari sensori montati sul veicolo (ad esclusione di quelli ottici); il sottosistema *Logging* fornisce il valore delle variabili in ingresso e uscita dal blocco FCS. In particolare il blocco *State Estimator* è composto internamente da un blocco ausiliario *Sensor Preprocessing* che effettua il preprocessing dei dati provenienti dall'IMU e dall'unità ottica (esclusivamente per il segnale stima ottica dello Jaw) e da tre sottosistemi principali:

1. *Complementary Filter*: elabora i dati provenienti dal sistema di preprocessing e fornisce una stima dell'orientamento del sistema: in particolare calcola gli angoli di roll, pitch e yaw e le variazioni angolari (p ; q ; r) rispetto al *Earth Fixed Frame*.

2. *Estimator Altitude*: elabora i dati provenienti dal sensore di pressione e dal sonar e, incrociandoli con i dati riguardanti l'orientamento del sistema, determina la quota e la velocità del veicolo.
3. *Estimator XY Position*: elabora i dati provenienti dal sensore ottico per determinare la posizione e la velocità longitudinale e latitudinale del veicolo.

Le 12 variabili così ottenute (evidenziate in figura) rappresentano le variabili di stato del sistema che saranno fornite al blocco *Flight Controller*.

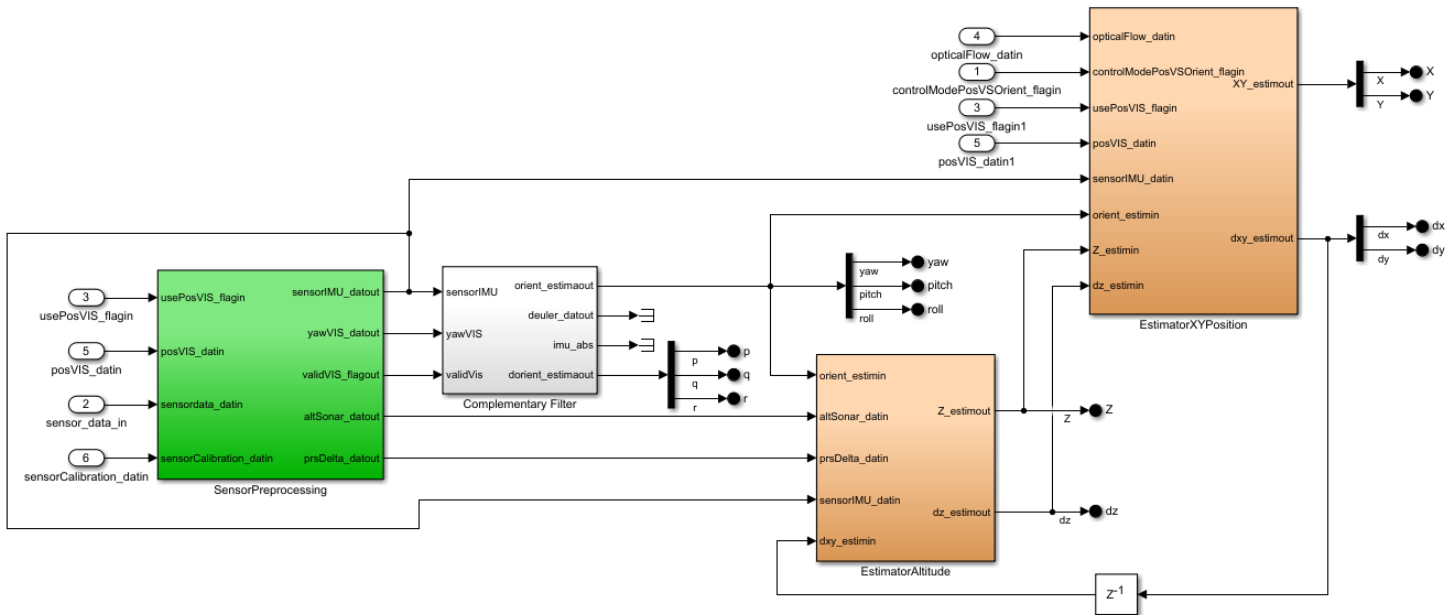


Figura 3.18: Blocco *State Estimator*

Il blocco *Flight Controller* è composto da quattro sottosistemi principali e da tre blocchi ausiliari, dei quali *Switch Ref Att* consente il passaggio dal segnale di riferimento rispetto alla posizione, al segnale di riferimento rispetto all'orientamento del veicolo (utilizzando come segnale di attivazione dello switch quello proveniente dalla porta logica del blocco *Signal Builder*); mentre gli altri due blocchi ausiliari *Control Mixer* e *Thrust to Motor Command* forniscono le costanti moltiplicative relative all'azione degli attuatori.

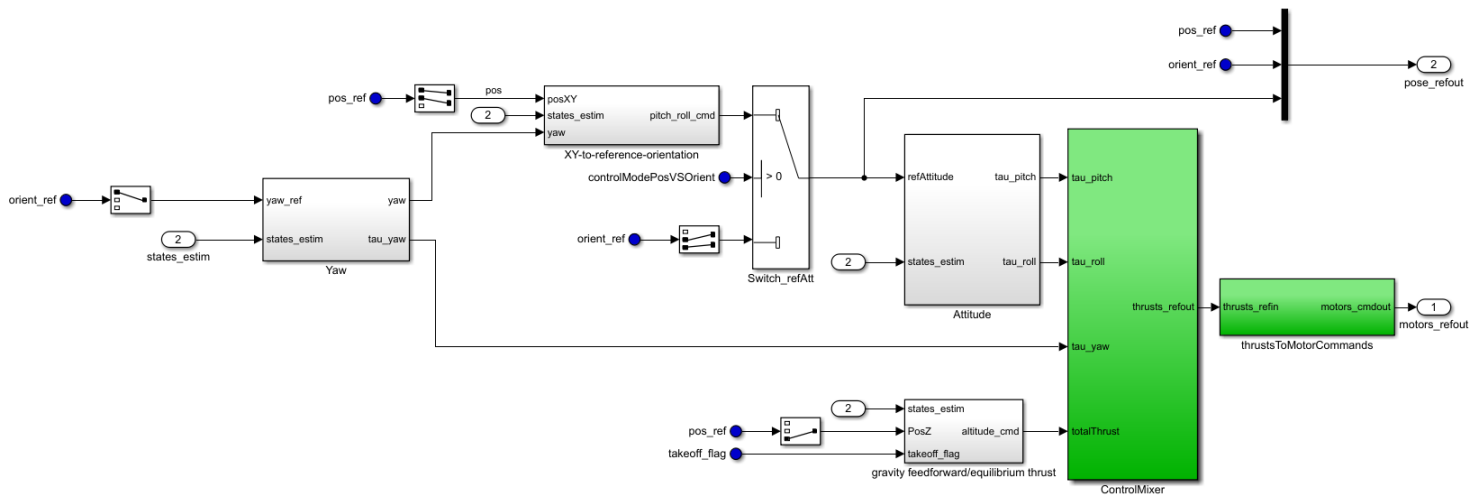


Figura 3.19: Blocco *Flight Controller*

I quattro sottosistemi principali si occupano direttamente della generazione dei segnali di controllo che definiranno il comportamento del veicolo:

- *XY to reference orientation*: riceve in ingresso il riferimento alla posizione del veicolo (longitudinale e latitudinale), l'angolo di yaw e il valore corrente della posizione e della velocità (longitudinali e latitudinali). In uscita elabora un segnale di controllo indicante il valore che devono assumere gli angoli di pitch e roll per mantenere la posizione di riferimento.
- *Attitude*: riceve in ingresso il valore corrente degli angoli di pitch e roll, e delle relative velocità angolari, e, a seconda del valore dello switch, il segnale di riferimento relativo alla posizione oppure quello relativo all'orientamento. In uscita fornisce i segnali di controllo per gli angoli di pitch e roll.
- *Yaw*: riceve in ingresso il riferimento relativo all'angolo di Yaw, il valore corrente di tale angolo e la relativa velocità angolare. In uscita fornisce il segnale di controllo per l'angolo di Yaw.
- *Gravity Feedrorward/Equilibrium Thrust*: riceve in ingresso il segnale di riferimento relativo alla quota (Z), il valore corrente di tale variabile, la velocità verticale del veicolo e il segnale *Take Off Flag* che regola l'azione di decollo. In uscita elabora il segnale di controllo relativo all'altitudine del veicolo.

Possiamo notare che i blocchi XY to reference orientation e Attitude compongono, quando lo switch attivo in posizione 1, un sistema di controllo a cascata, in quanto i segnali di controllo relativi alla posizione latitudinale e longitudinale vanno ad alimentare il sistema di controllo degli angoli di pitch e yaw, questo tipo di controllo indiretto è inevitabile in quanto un quadricottero è un sistema sotto-attuato. In caso

di switch attivo in posizione 2, il segnale di riferimento sarà invece costituito dalle variabili p, q settate all'interno del blocco Signal Builder.

3.5.1 Controllo dello Yaw

Riceve in ingresso il riferimento relativo all'angolo di Yaw, il valore corrente di tale angolo. In uscita fornisce il segnale di controllo per l'angolo di Yaw.

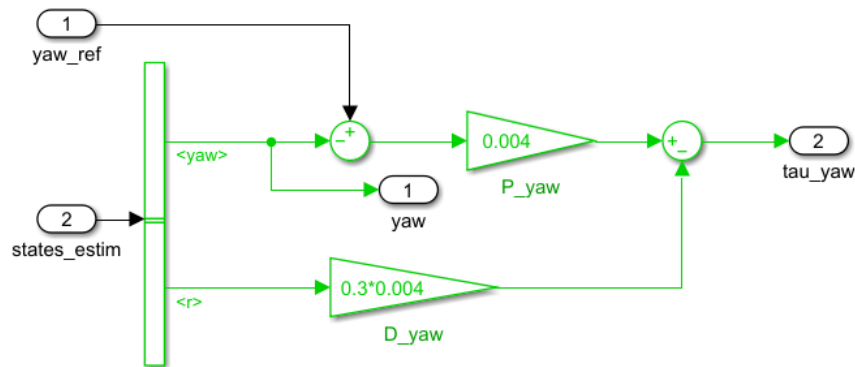


Figura 3.20: Blocco *Yaw*

3.5.2 Sistema di Controllo Pitch e Roll

Il blocco XY to reference orientation si compone di una parte superiore in cui, utilizzando l'angolo di yaw, si calcolano le componenti X, Y rispetto al sistema di riferimento inerziale dovute alla rotazione del veicolo sull'asse Z (nei fatti si applicano le prime due righe della matrice $R(Z;\phi)$). Nella parte inferiore il valori così ottenuti vengono sommati agli errori sulla posizione longitudinale e latitudinale (ottenuti comparando i valori correnti di X e Y con i relativi riferimenti) e quindi moltiplicati rispettivamente per due costanti. Gli ingressi relativi alle velocità dx e dy sono anch'essi moltiplicati rispettivamente per costanti opportune. Nel complesso, sommando i risultati dei due rami, si viene a creare un controllore PD (Proporzionale Derivativo).

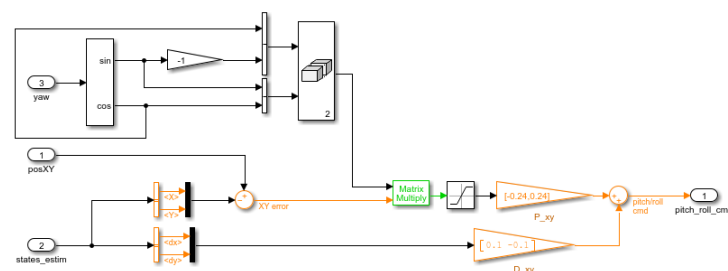


Figura 3.21: Blocco *XY to reference orientation*

Riceve in ingresso il segnale di riferimento relativo alla posizione oppure quello relativo all'orientamento e il valore corrente degli angoli di pitch e roll. In uscita fornisce i segnali di controllo per gli angoli di pitch e roll. Come possiamo notare dalla Figura 3.22 il segnale di riferimento $refAttitude$ è chiamato in questo modo per l'introduzione dello switch in questo modo il segnale di riferimento può essere o un segnale di controllo generato dal blocco *XY to reference orientation* o il segnale di orientamento proveniente dal blocco *Signal Builder*. Questo segnale relativo agli angoli di pitch e roll, viene confrontato con i valori correnti di tali angoli in modo da ricavare l'errore rispetto al riferimento. Questo valore sarà preso in ingresso sia dal blocco proporzionale che dal blocco integratore. Mentre nella parte inferiore della Figura 3.22 notiamo come i segnali relativi alle velocità p e q sono moltiplicati per una costante. Sommando i vari contributi si ottiene un controllore PID.

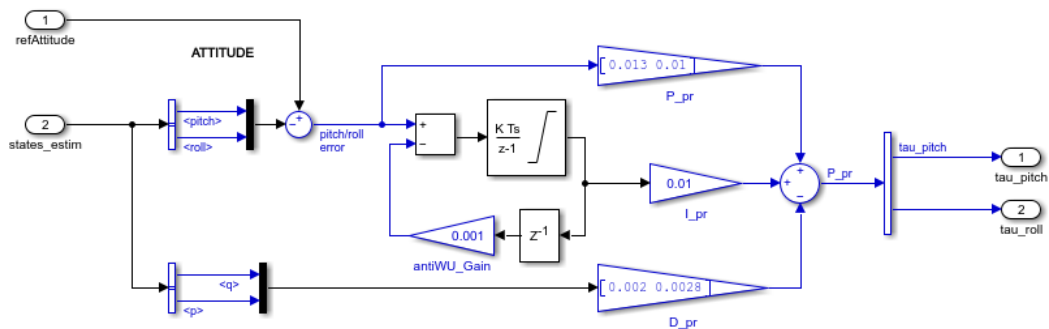


Figura 3.22: Blocco *Attitude*

3.5.3 Controllo dell'altitudine Z

Riceve in ingresso il valore corrente della quota, il segnale di riferimento relativo alla quota, il segnale *TakeOff_Flag* che regola l'azione di decollo. In uscita fornisce il segnale di controllo dell'altitudine del veicolo.

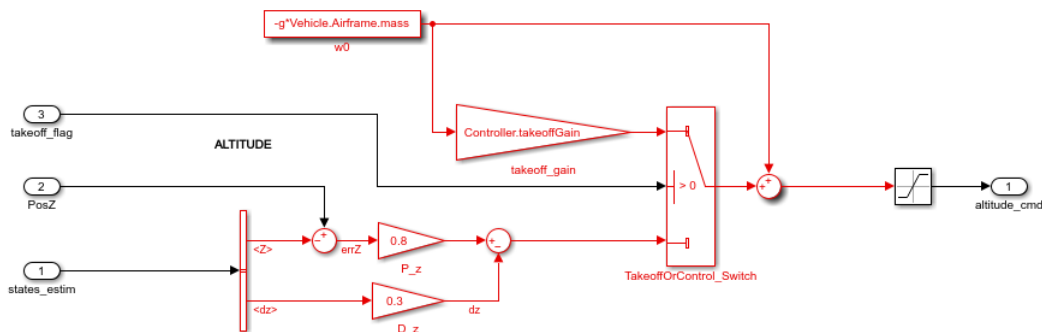


Figura 3.23: Blocco *Gravity Feedforward/Equilibrium Thrust*

Capitolo 4

Sintesi dei controllori

Avendo evidenziato nel capitolo precedente gli aspetti costitutivi principali del progetto asbQuadcopter, si procede ora con la realizzazione di controllori tramite sintesi in frequenza e con luogo delle radici per il controllo della dinamica del pitch e del roll. In particolare, si parte con la linearizzazione del modello matematico, poichè, nonostante la sua accuratezza nel descrivere la dinamica di un quadricottero ed i fenomeni fisici che ne influenzano il comportamento in volo, questo risulta essere un modello non lineare, e quindi sul quale non è possibile applicare tecniche di controllo lineare quali sintesi in frequenza e con luogo delle radici. A questo segue l'analisi delle prestazioni di volo ottenute tramite l'impiego dei controllori PID originali del progetto asbQuadcopter, con relativa introduzione ai principi di funzionamento di questi controllori. Si procede infine con l'effettiva realizzazione di controllori tramite le due tecniche di sintesi sopracitate, con loro successiva implementazione all'interno del progetto Simulink ed analisi delle prestazioni ottenute in simulazione.

4.0.1 Equazioni matematiche

Definite le trasformazioni che permettono di esprimere l'assetto del frame nella terna inerziale, è possibile applicare la seconda legge di Newton facendo prima delle ipotesi:

- L'origine della terna Body coincide con quella Inertial
- Ogni rotore è strutturalmente identico agli altri
- Il modello è tempo-invariante
- Il drone ha una struttura perfettamente simmetrica per cui la matrice di inerzia è diagonale :

$$J_{CM} = \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix} \quad (4.1)$$

Adesso dall'Equazione di Newton si ha:

$$\begin{pmatrix} F_B \\ \tau_B \end{pmatrix} = \begin{pmatrix} mI_{3x3} & 0 \\ 0 & J_{CM} \end{pmatrix} \cdot \begin{pmatrix} \dot{V}_B \\ \dot{\omega}_B \end{pmatrix} + \begin{pmatrix} \omega_B x(mV_B) \\ \omega_B x(J_{CM}\omega_B) \end{pmatrix} \quad (4.2)$$

Dove la matrice $I_{3 \times 3}$ indica la matrice identità di ordine 3 e J_{CM} è il tensore di inerzia riportato sopra Eq. 4.1. A questo punto possiamo riscrivere l'equazione 4.2 in forma matriciale definendo:

- il vettore delle velocità generalizzato nel Body (ν)
- la matrice di rotazione generalizzata (J_{Θ})
- il vettore generalizzato delle forze e dei momenti (Λ)
- il vettore generalizzato della posizione e dell'assetto (ξ)

$$\nu = \begin{bmatrix} V_b & \omega_b \end{bmatrix}^T = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \quad (4.3)$$

$$\Lambda = \begin{bmatrix} F_b & \tau_b \end{bmatrix}^T = \begin{bmatrix} F_x & F_y & F_z & \tau_x & \tau_y & \tau_z \end{bmatrix}^T \quad (4.4)$$

$$\xi = \begin{bmatrix} P_E & A_E \end{bmatrix}^T = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T \quad (4.5)$$

$$J_{\Theta} = \begin{bmatrix} DCM & 0_{3 \times 3} \\ 0_{3 \times 3} & ROT \end{bmatrix}^T \quad (4.6)$$

Si ottiene quindi :

$$M_B \dot{\nu} + C_B \nu = \Lambda \quad (4.7)$$

Dove M_B è detta «System inertia matrix» e C_B è la «Coriolis-centripetal matrix» il cui valore è :

$$M_B = \begin{bmatrix} mI_{3 \times 3} & 0 \\ 0 & J_{CM} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & J_x & 0 & 0 \\ 0 & 0 & 0 & 0 & J_y & 0 \\ 0 & 0 & 0 & 0 & 0 & J_z \end{bmatrix} \quad (4.8)$$

$$C_B = \begin{bmatrix} 0_{3 \times 3} - mS(V_B) \\ 0_{3 \times 3} - S(J_{CM}\omega_B) \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & mw & -mv \\ 0 & m & 0 & -mw & 0 & mu \\ 0 & 0 & m & mv & -mu & 0 \\ 0 & 0 & 0 & 0 & J_z r & -J_y q \\ 0 & 0 & 0 & -J_z r & 0 & J_x p \\ 0 & 0 & 0 & J_y q & -J_x p & 0 \end{bmatrix} \quad (4.9)$$

Dove S è l'operatore antisimmetrico, ovvero dato un vettore $k = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}^T$ si ha che

$$S(k) = -S^T(k) = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \quad (4.10)$$

Abbiamo ottenuto un risultato generico che riguarda tutti i corpi rigidi soggetti alle ipotesi precedenti. Dato che stiamo trattando il caso specifico di un quadricottero, vediamo i contributi che compongono il vettore Λ .

- Vettore gravitazionale : dovuto all'accelerazione di gravità $g = 9.81[m/s^2]$, non influenza il momento totale del corpo ma solo le forze a cui è soggetto e in particolare, riferendosi alla terna inerziale, all'asse z
- Effetto giroscopio : è un fenomeno fisico dovuto alla conservazione del momento angolare e generato quindi dal movimento delle eliche.
- Forze e momenti : riguarda gli effetti degli input riguardanti i movimenti principali. Definiamo l la distanza tra il centro di un rotore e il centro del quadricottero, b il coefficiente di spinta del motore e d drag.

$$\text{Si ottiene : } \begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = lb(-\Omega_2^2 + \Omega_4^2) \\ U_3 = lb(-\Omega_1^2 + \Omega_3^2) \\ U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{cases}$$

L'effetto del giroscopio è tanto più forte quanto più la velocità di rotazione è grande, e in questo caso non ci sono grandi velocità, l'effetto è trascurabile. Si può dimostrare che trascurandolo, il modello può essere quindi scritto con il seguente sistema di equazioni nel sistema di riferimento del drone.

$$\begin{cases} \dot{u} = (vr - wq) + g\sin(\theta) \\ \dot{v} = (wp - ur) - g\cos(\theta)\sin(\phi) \\ \dot{w} = (uq - vq) - g\cos(\theta)\sin(\phi) + \frac{U_1}{m} \\ \dot{p} = \frac{J_y - J_z}{J_x}qr + \frac{U_2}{J_x} \\ \dot{q} = \frac{J_z - J_x}{J_y}qr + \frac{U_3}{J_y} \\ \dot{r} = \frac{J_x - J_y}{J_z}qr + \frac{U_4}{J_z} \end{cases} \quad (4.11)$$

Se ora volessimo scrivere il modello con riferimento al sistema terra E, ricordando le equazioni 2.6 e 2.7, otteniamo il seguente sistema di equazioni.

$$\begin{cases} \ddot{x} = (\sin(\psi)\sin(\phi) + \cos(\psi)\sin(\theta)\cos(\phi))\frac{U_1}{m} \\ \ddot{y} = (-\cos(\psi)\sin(\phi) + \sin(\psi)\sin(\theta)\cos(\phi))\frac{U_1}{m} \\ \ddot{z} = -g + (\cos(\theta)\cos(\phi))\frac{U_1}{m} \\ \dot{\phi} = p + \sin(\phi)\tan(\theta)q + \cos(\phi)\tan(\theta)r \\ \dot{\theta} = \cos(\phi)q + \cos(\theta)\sin(\phi)r \\ \dot{\psi} = -\sin(\phi)q + \cos(\phi)\cos(\theta)r \end{cases} \quad (4.12)$$

Nonostante il modello sia già stato approssimato, perché si è trascurato l'effetto giroscopico, esso risulta chiaramente non lineare.

L'equazione sopra riportata Eq. 4.12 insieme all' Eq. 2.8, esprimono tutte le equazioni necessarie.

4.1 Linearizzazione del modello

In questa sezione si tratterà la linearizzazione del modello precedentemente definito non lineare. Consideriamo il quadricottero come un sistema non lineare e stazionario, la cui dinamica è descritta da un sistema di equazioni :

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= g(x(t), u(t))\end{aligned}\tag{4.13}$$

Per la linearizzazione consideriamo delle piccole oscillazione intorno a uno stato d'equilibrio per il drone e successivamente si applica lo sviluppo di Taylor del modello troncando la serie ai primi termini del primo ordine.

Definiamo \bar{x} come stato d'equilibrio per un ingresso costante \bar{u} . Dalla definizione di stato d'equilibrio si ha che uno stato \bar{x} si dice d'equilibrio per l'ingresso costante \bar{u} se valgono le seguenti condizioni:

$$\begin{aligned}x(t) &= \bar{x} \quad \forall t \geq t_0 \\ u(t) &= \bar{u} \quad \forall t \geq t_0 \\ \dot{x}(t) &= f(\bar{x}, \bar{u}) = 0 \quad \forall t \geq t_0 \\ y(t) &= g(\bar{x}, \bar{u}) = \bar{y} \quad \forall t \geq t_0\end{aligned}\tag{4.14}$$

A questo punto consideriamo $x(t)$, $y(t)$ e $u(t)$ con delle piccole oscillazione attorno a \bar{x} , \bar{y} e \bar{u} :

$$\begin{aligned}x(t_0) &= \bar{x} + \delta x(t_0) \\ x(t) &= \bar{x} + \delta x(t) \\ u(t) &= \bar{u} + \delta u(t) \\ y(t) &= \bar{y} + \delta y(t)\end{aligned}\tag{4.15}$$

Otteniamo quindi riprendendo l'equazione 4.13 :

$$\begin{aligned}\dot{x}(t) &= \dot{\bar{x}} + \delta \dot{x}(t) = f(\bar{x} + \delta x(t), \bar{u} + \delta u(t)) \\ y(t) &= \bar{y} + \delta y(t) = g(\bar{x} + \delta x(t), \bar{u} + \delta u(t))\end{aligned}\tag{4.16}$$

Procedendo con con lo sviluppo di Taylor delle due equazioni troncato ai termini del primo ordine si ottiene:

$$\delta\dot{x}(t) = f(\bar{x}, \bar{u}) + \left. \frac{\partial f(x, u)}{\partial x} \right|_{x=\bar{x}, u=\bar{u}} \delta x(t) + \left. \frac{\partial f(x, u)}{\partial u} \right|_{x=\bar{x}, u=\bar{u}} \delta u(t) \quad (4.17)$$

$$\bar{y} + \delta y(t) = g(\bar{x}, \bar{u}) + \left. \frac{\partial g(x, u)}{\partial x} \right|_{x=\bar{x}, u=\bar{u}} \delta x(t) + \left. \frac{\partial g(x, u)}{\partial u} \right|_{x=\bar{x}, u=\bar{u}} \delta u(t) \quad (4.18)$$

Come notiamo nell'Eq. 4.17 il termine \bar{x} essendo uno stato di equilibrio costante si annulla. A questo punto è possibile fare delle semplificazioni nelle equazioni 4.17 e 4.18 riprendo le equazioni 4.14 nello specifico le ultime due equazioni ovvero $f(\bar{x}, \bar{u}) = 0$ e $g(\bar{x}, \bar{u}) = \bar{y}$ così da ottenere:

$$\delta\dot{x}(t) = \left. \frac{\partial f(x, u)}{\partial x} \right|_{x=\bar{x}, u=\bar{u}} \delta x(t) + \left. \frac{\partial f(x, u)}{\partial u} \right|_{x=\bar{x}, u=\bar{u}} \delta u(t) \quad (4.19)$$

$$\delta y(t) = \left. \frac{\partial g(x, u)}{\partial x} \right|_{x=\bar{x}, u=\bar{u}} \delta x(t) + \left. \frac{\partial g(x, u)}{\partial u} \right|_{x=\bar{x}, u=\bar{u}} \delta u(t) \quad (4.20)$$

Da cui, applicando le condizioni iniziali quello che otteniamo è il seguente sistema linearizzato:

$$\begin{aligned} \delta\dot{x}(t) &= A\delta x(t) + B\delta u(t) \\ \delta y(t) &= C\delta x(t) + D\delta u(t) \end{aligned} \quad (4.21)$$

Dove:

$$A = \left. \frac{\partial f(x, u)}{\partial x} \right|_{x=\bar{x}, u=\bar{u}} \quad (4.22)$$

$$B = \left. \frac{\partial f(x, u)}{\partial u} \right|_{x=\bar{x}, u=\bar{u}} \quad (4.23)$$

$$C = \left. \frac{\partial g(x, u)}{\partial x} \right|_{x=\bar{x}, u=\bar{u}} \quad (4.24)$$

$$D = \left. \frac{\partial g(x, u)}{\partial u} \right|_{x=\bar{x}, u=\bar{u}} \quad (4.25)$$

Con :

- A : matrice della dinamica del sistema
- B : matrice degli ingressi del sistema
- C : matrice delle uscite del sistema
- D : che descrive il contributo degli ingressi sulle uscite del sistema

Le matrici A, B, C e D del modello lineare sono state ricavate dalla struttura dati *linsys* caricata sul Workspace di MATLAB dal file *linearizedAirframe.mat* contenuto all'interno della cartella *linearAirframe* del progetto Simulink. Per i nostri scopi di

Capitolo 4 Sintesi dei controllori

controllo, consideriamo le componenti relative alla dinamica del pitch e del Roll. Oltre al valore dei coefficienti delle matrici, aprendo i file *StateName*, *InputName*, *OutputName* troviamo il nome delle variabili di stato, degli ingressi e delle uscite.

4.1.1 Matrici del sistema Pitch

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	3.6611e-17	0	0	0	0	0	0	0	1	3.0292e-23	-5.2296e-12
2	-3.6611e-17	0	0	0	0	0	0	0	0	0	1	5.2295e-12
3	-6.8347e-13	-2.0323e-28	0	0	0	0	0	0	0	0	-5.2295e-12	1.0000
4	0	-9.8100	0	-3.4402e-09	-3.4402e-09	6.8532e-13	0	0	0	-5.4616e-11	0.1380	-3.2848e-13
5	9.8100	-2.8478e-22	0	-3.6595e-09	-3.6595e-09	-6.8520e-13	0	0	0	-0.1380	5.8098e-11	-3.5159e-13
6	5.4456e-11	5.4456e-11	0	-6.8347e-13	6.8347e-13	0	0	0	0	3.2920e-13	3.5087e-13	0
7	1.7216e-24	1.3478e-19	3.2920e-13	1.0000	-1.4969e-18	-5.2296e-12	0	0	0	0	0	0
8	-1.3478e-19	2.0176e-37	3.5087e-13	1.4969e-18	1	5.2295e-12	0	0	0	0	0	0
9	-3.2920e-13	-3.5087e-13	0	5.2296e-12	-5.2295e-12	1.0000	0	0	0	0	0	0
10	0	0	0	-5.7590e-08	-5.7590e-08	-2.7287e-14	0	0	0	-2.1715	9.1429e-10	4.3301e-13
11	0	0	0	4.0369e-08	4.0369e-08	-2.1644e-14	0	0	0	6.4089e-10	-1.6192	-5.1363e-13
12	0	0	0	-1.4434e-19	-1.2831e-19	3.6185e-19	0	0	0	1.2157e-13	1.1242e-13	-6.8807e-09
13												

Figura 4.1: Matrice A della dinamica (Pitch)

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
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
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
6	-0.0096	0.0096	-0.0096	0.0096	-8.2855	0	0	0	0	0	1.0000	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0.4202	0.4202	-0.4202	-0.4202	2.5288e-14	0	0	0	0	0	0	0
11	0.3133	-0.3133	-0.3133	0.3133	1.8856e-14	0	0	0	0	0	0	0
12	-0.0115	-0.0115	-0.0115	-0.0115	-7.9371e-16	0	0	0	0	0	0	0
13												

Figura 4.2: Matrice B degli ingressi (Pitch)

	1	2	3	4	5	6	7	8	9	10	11	12
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
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
3	5.4456e-11	5.4456e-11	0	0	0	0	0	0	0	0	0	0
4	0	5.5511e-12	0	0	0	0	0	0	0	0	0	0
5	-5.2296e-12	-5.2295e-12	-1.0000	0	0	0	0	0	0	0	0	0
6	1.4969e-18	1.0000	-5.2295e-12	0	0	0	0	0	0	0	0	0
7	0	8.3096e-30	1.0000	0	0	0	0	0	0	0	0	0
8	5.5511e-12	0	0	0	0	0	0	0	0	0	0	0
9	-1.0000	1.4969e-18	-5.2296e-12	0	0	0	0	0	0	0	0	0
10	0	-1.0000	0	0	0	0	0	0	0	0	0	0
11	1.0000	-2.9029e-23	0	0	0	0	0	0	0	0	0	0
12	5.5511e-12	5.5511e-12	0	0	0	0	0	0	0	0	0	0
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

Figura 4.3: Matrice C delle uscite (Pitch)

4.1 Linearizzazione del modello

linsys.D												
	1	2	3	4	5	6	7	8	9	10	11	12
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
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
3	-0.0096	0.0096	-0.0096	0.0096	-8.2855	0	0	0	0	0	1.0000	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0

Figura 4.4: Matrice D (Pitch)

linsys.InputName			
	1	2	3
1	Actuators(1)		
2	Actuators(2)		
3	Actuators(3)		
4	Actuators(4)		
5	Atmospher...		
6	Atmospher...		
7	Atmospher...		
8	Atmospher...		
9	Gravity_ned...		
10	Gravity_ned...		
11	Gravity_ned...		
12	MagneticFi...		
13	MagneticFi...		
14	MagneticFi...		
15			

(a) Ingressi

linsys.StateName			
	1	2	3
1	phi theta ps...		
2	phi theta ps...		
3	phi theta ps...		
4	ub,vb,wb(1)		
5	ub,vb,wb(2)		
6	ub,vb,wb(3)		
7	xe,ye,ze(1)		
8	xe,ye,ze(2)		
9	xe,ye,ze(3)		
10	p,q,r(1)		
11	p,q,r(2)		
12	p,q,r(3)		

(b) Stati

linsys.OutputName			
	1	2	3
1	Accel_body...		
2	Accel_body...		
3	Accel_body...		
4	DCM_be(1)		
5	DCM_be(2)		
6	DCM_be(3)		
7	DCM_be(4)		
8	DCM_be(5)		
9	DCM_be(6)		
10	DCM_be(7)		
11	DCM_be(8)		
12	DCM_be(9)		
13	Euler(1)		
14	Euler(2)		
15	Euler(3)		

(c) Uscite

Figura 4.5: Variabili di stato, ingresso e uscite

Per quanto riguarda l'angolo di pitch è stato sufficiente considerare per ogni singola matrice le seguenti componenti:

- $A : [\theta, \theta], [\theta, q], [q, \theta], [q, q]$
- $B : [\theta, Actuators(1)(2)(3)(4)], [q, Actuators(1)(2)(3)(4)]$
- $C : [Euler(2), \theta], [Euler(2), q]$
- $D : [Euler(2), Actuators(1)(2)(3)(4)]$

Dove θ , q , $Euler(2)$ e $Actuators(1)(2)(3)(4)$ sono rispettivamente la posizione angolare di pitch rispetto all'inertial frame, la velocità angolare di pitch rispetto al body frame, l'angolo di Eulero per il controllo del pitch e la spinta applicata dai 4 attuatori (rotori) del drone.

A questo punto possiamo costruire le matrici del sistema riprendendo i valori delle Figure 4.1, 4.2, 4.3, 4.4 :

$$A = \begin{pmatrix} 0 & 1 \\ 0 & -1.619 \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ 0.3133 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 0 \end{pmatrix} \quad (4.26)$$

Prima di poter calcolare la nostra funzione di trasferimento, data dalla matrice A, B, C, D, bisogna calcolare le due costanti moltiplicative K1 e K2 che rispettivamente rappresentano i pesi attraverso i quali lo sforzo di controllo viene spartito tra i 4 attuatori e il coefficiente di spinta dei motori. Per ottenere le due costanti K1 e K2 basta scrivere nella linea di comando di Matlab:

- `Controller.Q2Ts`
- `Vehicle.Motor.thrustToMotorCommand`

Nella Figura 4.6 per quanto riguarda la costante K1 vengono mostrate i pesi per il controllo di altitudine, yaw, pitch e roll. Nel nostro caso siamo interessati ai pesi per il controllo del pitch evidenziati di blu.

```
Command Window
>> Controller.Q2Ts

ans =
      Z      Yaw      Pitch      Roll
  0.2500  103.5736  -5.6659  -5.6659
  0.2500 -103.5736  -5.6659   5.6659
  0.2500  103.5736   5.6659   5.6659
  0.2500 -103.5736   5.6659  -5.6659

>> Vehicle.Motor.thrustToMotorCommand

ans =
  1.5307e+03
```

Figura 4.6: Costanti moltiplicative

Dalla Figura 4.6 ricaviamo $K1 = 5.6659$ e $K2 = 1.5307 \cdot 10^3$. Adesso possiamo calcolare la funzione di trasferimento della dinamica del pitch attraverso le righe di codice riportate in Figura 4.7 :

```

Command Window

>> A=[0 1; 0 -1.619];
>> B=[0;0.3133];
>> C=[1 0 ];
>> D=0;
>> sys=ss(A,B,C,D);
>> plant=tf(sys);
>> K1=5.6659;
>> K2=1.5307e+03;
>> Ps=K1*K2*plant

Ps =

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

Continuous-time transfer function.
fx

```

Figura 4.7: Funzione di trasferimento della dinamica del pitch

Otteniamo che $P(s)$ è ricavata dall' Eq. 4.27 di seguito riportata

$$P(s) = K1 \cdot K2 \cdot C(sI - A)^{-1}B = \frac{2717}{s^2 + 1.619s} \quad (4.27)$$

4.1.2 Matrici del sistema Roll

Come nel caso del Pitch, le matrici A, B, C, D del modello linearizzato sono già state calcolate e memorizzate all'interno della struttura dati 'linsys' caricata sul Workspace di MATLAB all'apertura del progetto asbQuadcopter dal file linearizedAirframe.mat contenuto all'interno della cartella linearAirframe. Delle matrici A, B, C, D complessive del sistema, per quelli che sono i nostri scopi di controllo, risulta rilevante considerare solamente le componenti relative alla dinamica del roll, le quali sono state evidenziate nelle 4.8, 4.9, 4.10, 4.11. Nello specifico, all'interno delle matrici, tali componenti sono state selezionate basandosi sugli indici che hanno le variabili di stato, di ingresso e di uscita del sistema relative al roll nei rispettivi vettori degli stati, degli ingressi e delle uscite, presenti su MATLAB rispettivamente con i nomi di 'StateName', 'InputName' e 'OutputName' all'interno della struttura dati 'linsys' e riportati in 4.12a, 4.12b, 4.12c.

Capitolo 4 Sintesi dei controllori

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	3.6611e-17	0	0	0	0	0	0	0	0	3.0292e-23	-5.2296e-12	
2	-3.6611e-17	1	0	0	0	0	0	0	0	0	1	5.2295e-12	
3	-6.8347e-13	-2.0323e-28	1	0	0	0	0	0	0	0	-5.2295e-12	1.0000	
4	0	-9.8100	0	1	-3.4402e-09	-3.4402e-09	6.8532e-13	0	0	0	-5.4616e-11	0.1380	-3.2848e-13
5	9.8100	-2.8478e-22	0	-3.6595e-09	-3.6595e-09	-6.8520e-13	0	0	0	-0.1380	5.8098e-11	-3.5159e-13	
6	5.4456e-11	5.4456e-11	0	-6.8347e-13	6.8347e-13	0	0	0	0	3.2920e-13	3.5087e-13	0	
7	1.7216e-24	1.3478e-19	3.2920e-13	1.0000	-1.4969e-18	-5.2296e-12	0	0	0	0	0	0	0
8	-1.3478e-19	2.0176e-37	3.5087e-13	1.4969e-18	1	5.2295e-12	0	0	0	0	0	0	0
9	-3.2920e-13	-3.5087e-13	0	5.2296e-12	-5.2295e-12	1.0000	0	0	0	0	0	0	0
10	0	0	0	-5.7590e-08	-5.7590e-08	-2.7287e-14	0	0	0	-2.1715	9.1429e-10	4.3301e-13	
11	0	0	0	4.0369e-08	4.0369e-08	-2.1644e-14	0	0	0	6.4089e-10	-1.6192	-5.1363e-13	
12	0	0	0	-1.4434e-19	-1.2831e-19	3.6185e-19	0	0	0	1.2157e-13	1.1242e-13	-6.8807e-09	
13													

Figura 4.8: Matrice A della dinamica (Roll)

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
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	
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	
6	-0.0096	0.0096	-0.0096	0.0096	-8.2855	0	0	0	0	0	1.0000	0	
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0.4202	0.4202	-0.4202	-0.4202	2.5288e-14	0	0	0	0	0	0	0	0
11	0.3133	-0.3133	-0.3133	0.3133	1.8856e-14	0	0	0	0	0	0	0	0
12	-0.0115	-0.0115	-0.0115	-0.0115	-7.9371e-16	0	0	0	0	0	0	0	0
13													

Figura 4.9: Matrice B degli ingressi (Roll)

	1	2	3	4	5	6	7	8	9	10	11	12	13
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	
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	
3	5.4456e-11	5.4456e-11	0	0	0	0	0	0	0	0	0	0	0
4	0	5.5511e-12	0	0	0	0	0	0	0	0	0	0	0
5	-5.2296e-12	-5.2295e-12	-1.0000	0	0	0	0	0	0	0	0	0	0
6	1.4969e-18	1.0000	-5.2295e-12	0	0	0	0	0	0	0	0	0	0
7	0	8.3096e-30	1.0000	0	0	0	0	0	0	0	0	0	0
8	5.5511e-12	0	0	0	0	0	0	0	0	0	0	0	0
9	-1.0000	1.4969e-18	-5.2296e-12	0	0	0	0	0	0	0	0	0	0
10	0	-1.0000	0	0	0	0	0	0	0	0	0	0	0
11	1.0000	-2.9029e-23	0	0	0	0	0	0	0	0	0	0	0
12	5.5511e-12	5.5511e-12	0	0	0	0	0	0	0	0	0	0	0
13	1	0	0	0	0	0	0	0	0	0	0	0	0
14	0	1	0	0	0	0	0	0	0	0	0	0	0
15	0	0	1	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	9.0026e-06	0	0	0	0	0	0

Figura 4.10: Matrice C delle uscite (Roll)

4.1 Linearizzazione del modello

linsys.D													
	1	2	3	4	5	6	7	8	9	10	11	12	13
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	
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	
3	-0.0096	0.0096	-0.0096	0.0096	-8.2855	0	0	0	0	0	1.0000	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	
15	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	

Figura 4.11: Matrice D (Roll)

linsys.InputName	linsys.StateName	linsys.OutputName		
1	1	2	3	
1 Actuators(1)	1 phi theta psi(1)			1 Accel_body(1)
2 Actuators(2)	2 phi theta psi(2)			2 Accel_body(2)
3 Actuators(3)	3 phi theta psi(3)			3 Accel_body(3)
4 Actuators(4)	4 ub,vb,wb(1)			4 DCM_be(1)
5 AtmosphereBus.air_density	5 ub,vb,wb(2)			5 DCM_be(2)
6 AtmosphereBus.air_temp	6 ub,vb,wb(3)			6 DCM_be(3)
7 AtmosphereBus.pressure	7 xe,ye,ze(1)			7 DCM_be(4)
8 AtmosphereBus.speed_sound	8 xe,ye,ze(2)			8 DCM_be(5)
9 Gravity_ned(1)	9 xe,ye,ze(3)			9 DCM_be(6)
10 Gravity_ned(2)	10 p,q,r(1)			10 DCM_be(7)
11 Gravity_ned(3)	11 p,q,r(2)			11 DCM_be(8)
12 MagneticField_ned(1)	12 p,q,r(3)			12 DCM_be(9)
13 MagneticField_ned(2)				13 Euler(1)
14 MagneticField_ned(3)				14 Euler(2)
15				15 Euler(3)
				16 LLA(1)
				17 LLA(2)
				18 LLA(3)
				19 Omega_body(1)
				20 Omega_body(2)
				21 Omega_body(3)
				22 Omega_body(4)

(a) Ingressi

(b) Stati

(c) Uscite

Figura 4.12: Variabili di stato, ingresso e uscite

Le componenti di tali vettori rilevanti per la dinamica del roll sono:

- ϕ e p nel vettore degli stati, rappresentanti rispettivamente la posizione e la velocità angolare del roll.
- $Actuators(1)(2)(3)(4)$ nel vettore degli ingressi, rappresentati la spinta esercitata dai 4 attuatori del drone.
- $Euler(1)$ nel vettore delle uscite, rappresentante l'angolo di Eulero per il controllo del roll.

Delle matrici A, B, C, D sono stati considerati quindi solamente gli elementi corrispondenti ai seguenti indici.

- A: $[\phi, \phi], [\phi, p], [p, \phi], [p, p]$
- B: $[\phi, Actuators(1)(2)(3)(4)], [p, Actuators(1)(2)(3)(4)]$
- C: $[Euler(1), \phi], [Euler(1), p]$
- D: $[Euler(1), Actuators(1)(2)(3)(4)]$

Utilizzando i valori presenti nelle 4.8, 4.9, 4.10, 4.11 si hanno le seguenti matrici del sistema per la dinamica del roll.

$$A = \begin{pmatrix} 0 & 1 \\ 0 & -2.172 \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ 0.4202 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 0 \end{pmatrix} \quad (4.28)$$

Per il calcolo della funzione di trasferimento ad esse associate, è importante ricordare i contributi all'uscita del sistema del blocco *TorqueTotalToThrustPerMotor* e del blocco *ThrustToMotorCommand*, dalle quali si ottengono rispettivamente le due costanti moltiplicative K1 e K2, di cui la prima rappresenta i pesi attraverso i quali lo sforzo di controllo viene spartito tra i 4 attuatori, mentre la seconda rappresenta il coefficiente di spinta dei motori. Da MATLAB si ottiene che:

```

Command Window
>> Controller.Q2Ts
ans =

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

>> Vehicle.Motor.thrustToMotorCommand
ans =

    1.5307e+03
    
```

Figura 4.13: Costanti moltiplicative

Dalla seconda variabile si ottiene direttamente $K2 = 1.5307 \cdot 10^3$, mentre dalla prima, considerato che le colonne di *Controller.Q2Ts* in ordine da sinistra a destra rappresentano i pesi per il controllo rispettivamente di altitudine, yaw, pitch e roll, si ottiene $K1 = 5.6659$.


```

Command Window
>> A = [0 1;0 -2.172];
>> B = [0;0.4202];
>> C = [1 0];
>> D = [0];
>> sys = ss(A, B, C, D);
>> plant = tf(sys);
>> K1 = 5.6659;
>> K2 = 1.5307e+03;
>> Ps = K1 * K2 * plant

Ps =

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

Continuous-time transfer function.

```

Figura 4.14: Funzione di trasferimento della dinamica del Roll

La funzione di trasferimento rappresentativa della dinamica del roll ha di conseguenza la seguente espressione.

$$P(s) = K1 \cdot K2 \cdot C(sI - A)^{-1}B = \frac{3644}{s \cdot (s + 2.172)} \quad (4.29)$$

4.2 Controllori PID

In questa sezione analizzeremo il sistema di controllo con PID originali del progetto *asbQuacopterStart* facendo una breve analisi della struttura dei PID in generale e successivamente visualizzeremo le prestazioni del sistema con i PID di default.

4.2.1 Analisi dei PID

Un controllore PID è in generale composto da 3 componenti, una proporzionale, una integrale ed una derivativa, il cui effetto complessivo sullo sforzo di controllo generato dipende proporzionalmente da altrettanti coefficienti che vengono stimati attraverso delle tecniche atte a minimizzare l'errore commesso dal sistema rispetto ai riferimenti impartiti, come ad esempio i metodi di Ziegler-Nichols.

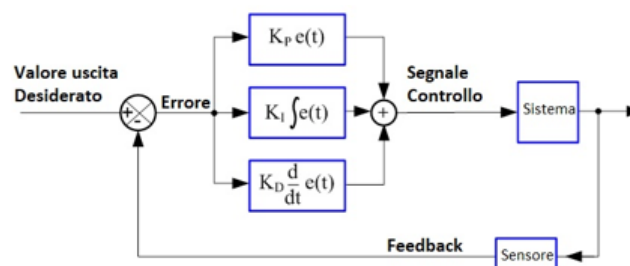


Figura 4.15: Struttura del PID

Nel dettaglio, dato uno sforzo di controllo $m(t) = m_P(t) + m_I(t) + m_D(t)$ scomposto nelle 3 componenti derivanti ciascuna da una parte del controllore PID, si ha:

Azione proporzionale: $m_P(t) = K_P e(t)$

All'aumentare del valore del coefficiente dell'azione proporzionale K_P il sistema diventa sempre più "pronto" a reagire ad eventuali ingressi, riducendo quindi il suo tempo di risposta. In casi particolarmente semplici quest'azione riesce anche da sola a stabilizzare processi instabili, e qualora si abbia un segnale con espressione polinomiale in ingresso ad un sistema che a ciclo chiuso risulta essere di tipo pari al grado di tale segnale allora si potrebbe anche riuscire a rendere costante l'errore commesso rispetto al riferimento, ma non ad annullarlo completamente.

Azione integrale:

$$m_I(t) = K_I \int_0^t e(\tau) d\tau \quad (4.30)$$

L'integrale dell'errore rispetto al tempo permette di tenere conto dell'errore commesso rispetto ai riferimenti anche in istanti di tempo passati, il cui peso sull'azione di controllo complessiva viene scelto tarando il coefficiente dell'azione integrale K_I . Questo permette di produrre eventualmente uno sforzo di controllo anche quando l'errore risulta essere nullo, riducendo progressivamente l'errore accumulato anche in istanti di tempo precedenti fino ad annullarlo a regime permanente. Se vista nel dominio di Laplace, la parte integrale introduce un polo nell'origine nel controllore, rendendo quindi il sistema astatico rispetto a disturbi costanti applicati in catena diretta e migliorando la risposta a regime permanente, a fronte però di un peggioramento del transitorio dovuto al ritardo di fase di 90° .

Azione derivativa:

$$m_D(t) = K_D \frac{de(t)}{dt} \quad (4.31)$$

La derivata dell'errore rispetto al tempo permette al controllore di stimarne il valore "futuro", opponendosi quindi a tutte le brusche variazioni del suo valore in proporzione al valore del coefficiente dell'azione derivativa K_D . Se vista nel dominio di Laplace, la parte derivativa introduce uno zero nell'origine nel controllore,

comportando quindi un anticipo di fase di 90° ed aumentando il margine di fase, con conseguente miglioramento del tempo di salita e della sovraelongazione. Essendo tuttavia che la funzione di trasferimento di un sistema fisicamente realizzabile deve essere necessariamente propria, l'azione derivativa di un PID va sempre affiancata ad una parte integrale, o comunque ad un polo lontano in alta frequenza che non vada ad interferire con l'azione dello zero.

Come mostrato in Figura 4.15, la funzione di trasferimento complessiva del controllore PID data dall'azione delle sue 3 componenti è quindi la seguente:

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (4.32)$$

4.2.2 Analisi delle prestazioni dei PID originali

Per analizzare le prestazioni dei controllori PID originali del progetto asbQuadcopter definiamo degli ingressi di riferimento rispetto ai quali valutare l'adeguatezza dell'azione di controllo fornita da tali controllori. Consideriamo quindi il segnale di riferimento raffigurato in Figura 4.16, rappresentante rispettivamente la traiettoria di hovering.

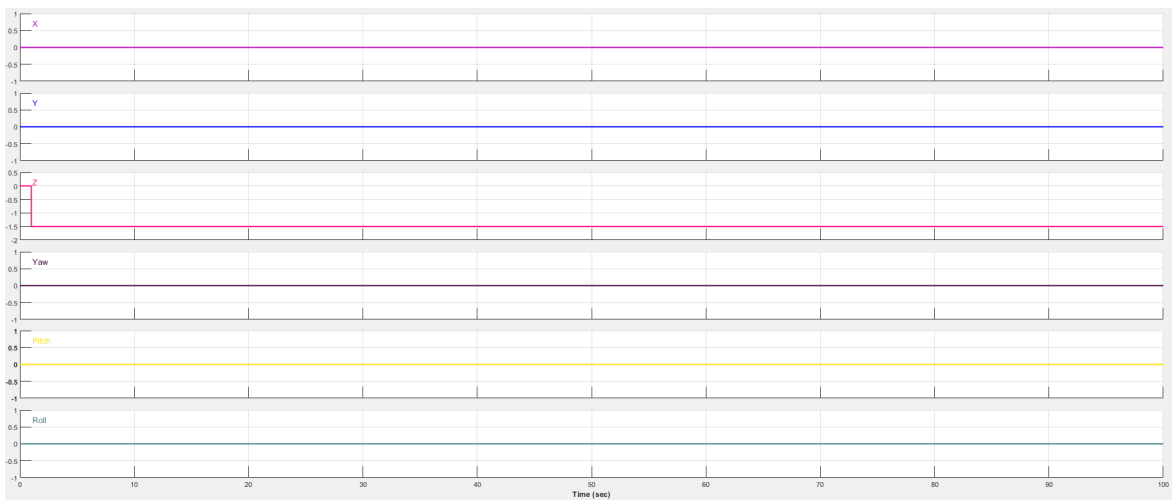


Figura 4.16: Segnali di riferimento impartiti

Si riportano di seguito i grafici delle prestazioni del quadricottero, per i riferimenti impartiti, per ciascuno dei 6 gradi di libertà, dove per ognuno di questi vengono rappresentati in giallo il suo valore effettivo e in blu il riferimento ad esso impartito. Le simulazioni effettuate con i riferimenti impartiti hanno durata di 30 s.

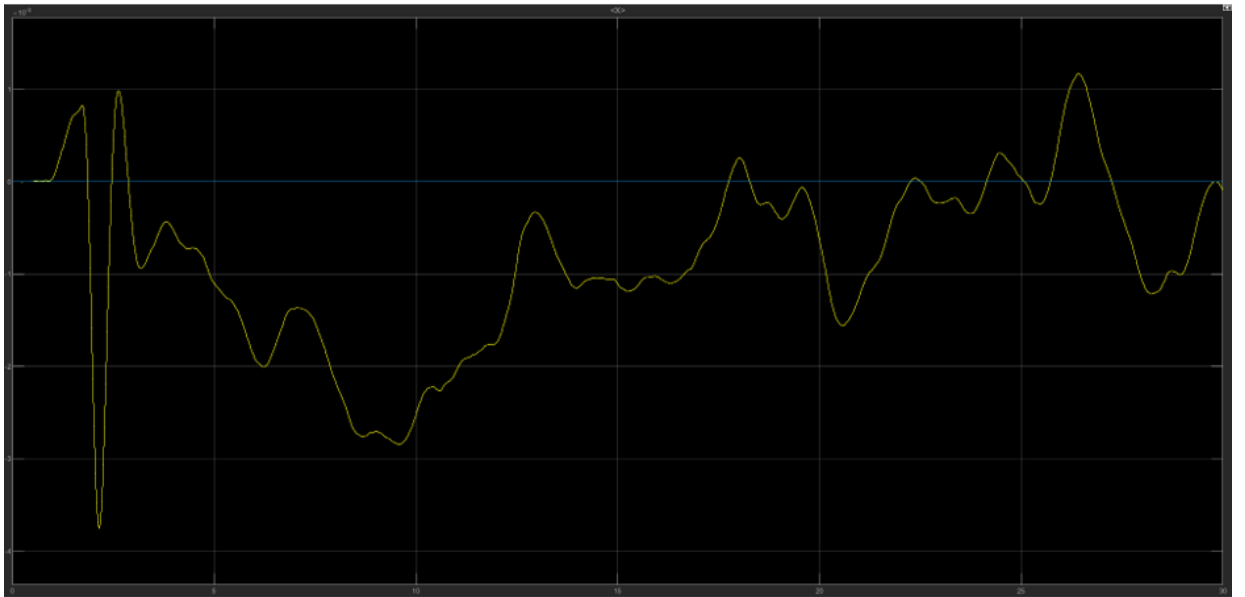


Figura 4.17: PID originali - X

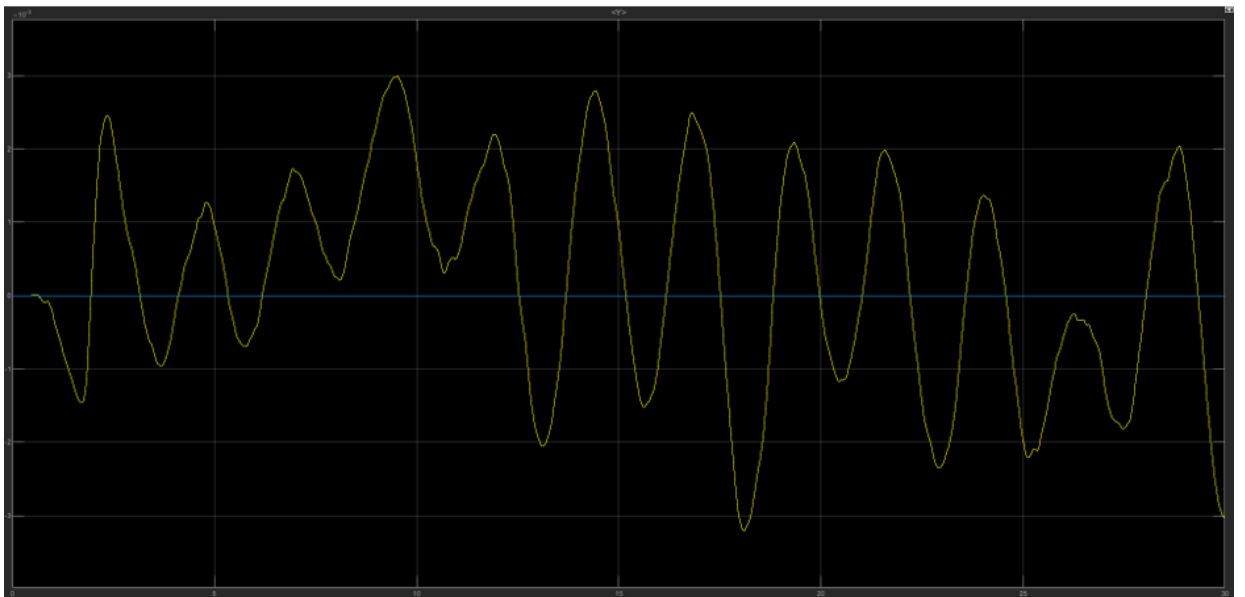


Figura 4.18: PID originali - Y

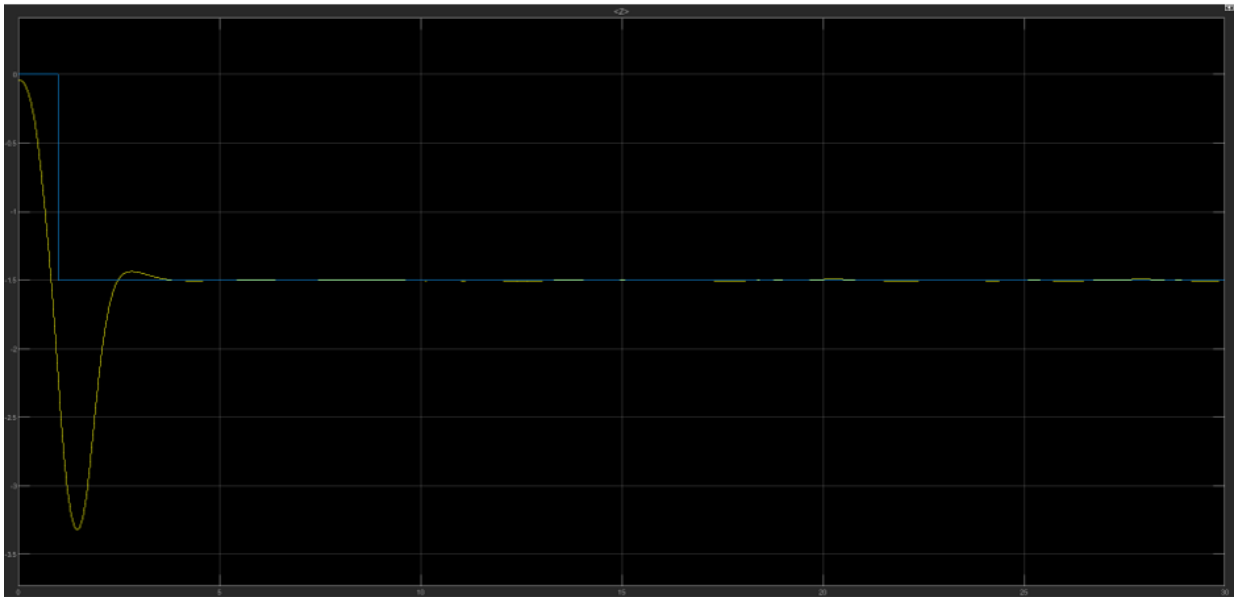


Figura 4.19: PID originali - Z

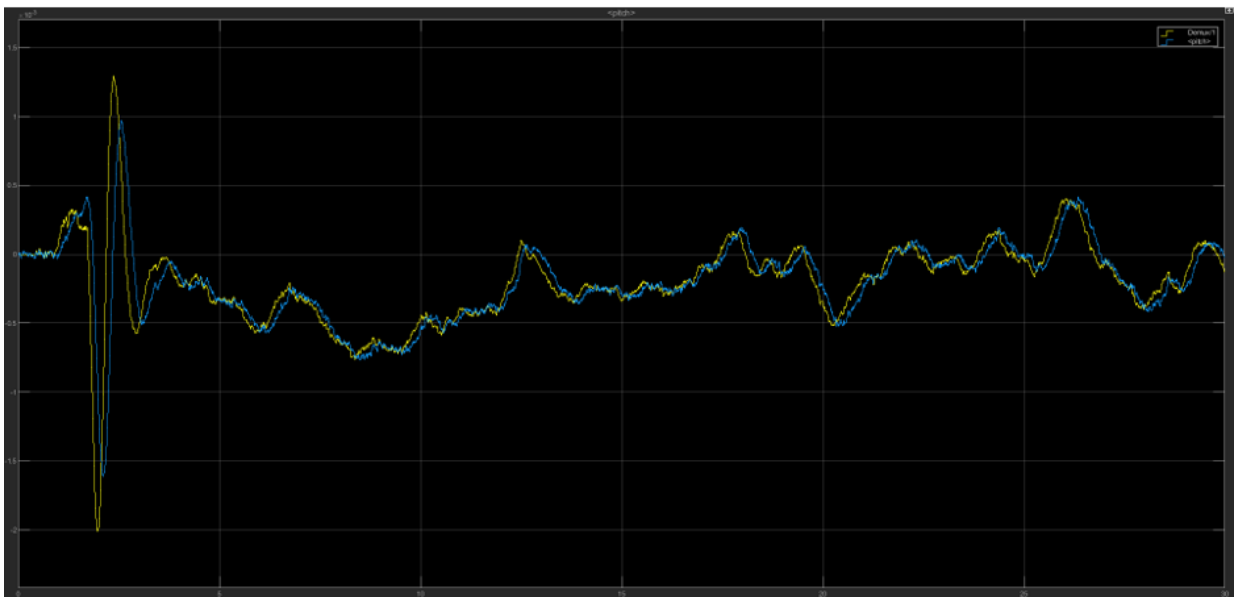


Figura 4.20: PID originali - Pitch

Volendo fare alcune considerazioni notiamo che, rispetto ad entrambi i riferimenti scelti, i gradi di libertà del roll e della Y (controllata sempre attraverso il roll) presentano delle oscillazioni non trascurabili che il sistema di controllo non è in grado di smorzare, motivo per il quale il quadricottero riesce difficilmente a seguire i riferimenti impartiti. Nel proseguo del presente elaborato ci si propone quindi di realizzare dei controllori che tengano conto di questi aspetti, andando eventualmente a ridurre le oscillazioni e migliorare la fedeltà di risposta.

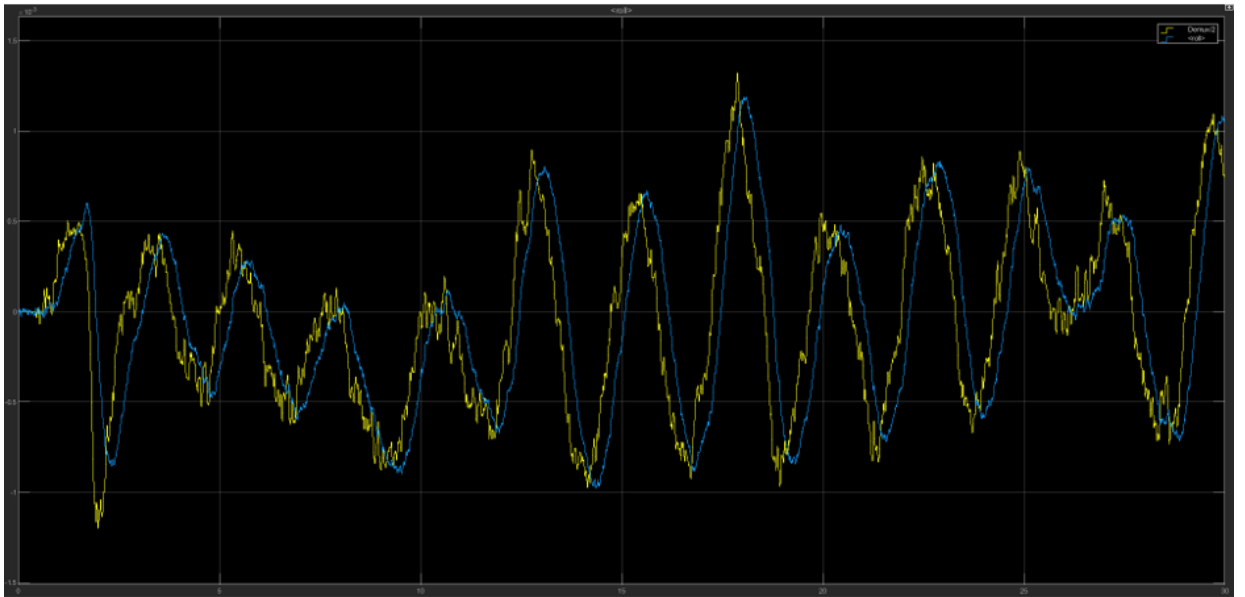


Figura 4.21: PID originali - Roll

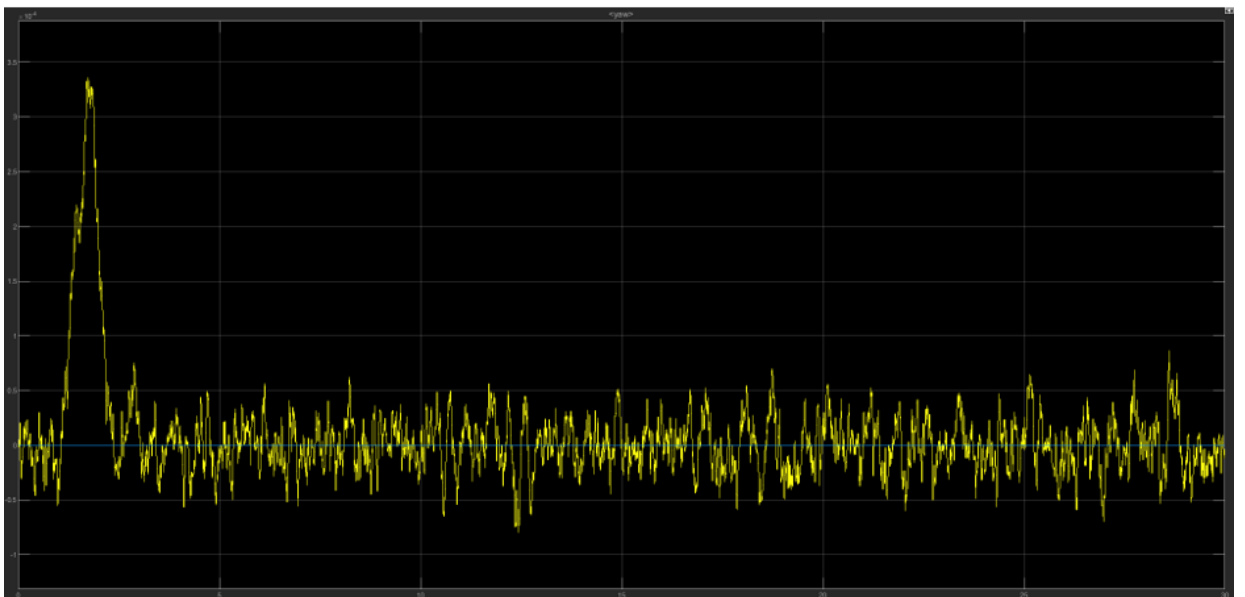


Figura 4.22: PID originali - Yaw

4.3 Tecniche di controllo lineare

Per la progettazione di controllori per la dinamica del pitch vengono utilizzate le due tecniche di controllo lineare ovvero sintesi in frequenza e luogo delle radici. In questa sezione viene descritta l'implementazione generale di entrambe le tecniche.

4.3.1 Sintesi in frequenza

Con questa tecnica si va a studiare la funzione di trasferimento nel dominio della frequenza, inizialmente abbiamo la funzione nel dominio di Laplace in cui è presente la variabile complessa s andando a sostituire $s = j\omega$ si ottiene la funzione di trasferimento nel dominio della frequenza. Successivamente con determinati strumenti come i diagrammi di Bode e di Nyquist è possibile studiare la stabilità del sistema. Nella sintesi in frequenza come per la sintesi con il luogo delle radici abbiamo delle specifiche da soddisfare, in questa sintesi troviamo:

- specifiche uniche : analizzare l'errore a regime permanente $e_k(t)$ rispetto a ingressi polinomiali di tipo k e astatismo rispetto ai disturbi
- specifiche lasche : vengono analizzati i parametri in frequenza in catena chiusa M_r e B_3 , i quali sono legati ai parametri in frequenza in catena aperta m_φ e ω_t e ai parametri nel tempo in catena chiusa \hat{s} e t_s , dove:
 - M_r : margine di risonanza, si definisce come la differenza tra il massimo valore raggiunto dal modulo di $W(j\omega)$ ed il valore di regime. In generale, si desidera avere un picco di risonanza abbastanza piccolo, in quanto tanto maggiore è il picco e tanto minore risulterà essere lo smorzamento della risposta del sistema
 - B_3 : banda passante, ovvero la pulsazione ω in corrispondenza della quale il valore $|W(j\omega)|_{dB}$ diminuisce di $3dB$ rispetto a $|W(j0)|_{dB}$. Un aumento della banda passante si traduce generalmente in un aumento della prontezza di risposta del sistema
 - m_φ : margine di fase, ovvero il valore della fase pari a $\angle F(j\omega t) + 180^\circ$
 - ω_t : pulsazione di attraversamento, ovvero la pulsazione in corrispondenza della quale il modulo di $|F(j\omega t)|_{dB}$ risulta essere pari a $0dB$
 - \hat{s} : sovralongazione è la differenza tra il massimo valore in uscita e il valore di riferimento
 - t_s : tempo di salita, si intende il tempo necessario al sistema per variare dal 10% al 90% del valore di regime. L'uscita deve raggiungere tale valore senza compiere oscillazioni

Di seguito sono riportate le relazioni tra i vari parametri:

$$\frac{1 + \hat{s}}{M_r} \approx 0.85 \quad (4.33)$$

$$[B_3]_{\frac{rad}{s}} \cdot [t_s]_s \approx 3 \text{ rad} \quad (4.34)$$

$$[\omega_t]_{\frac{rad}{s}} \approx 3 \div 5 [B_3]_{Hz} \quad (4.35)$$

Per soddisfare le specifiche univoche verrà realizzato in controllore di primo tentativo $\hat{G}(s)$ nella forma seguente :

$$\hat{G}(s) = \frac{K_G}{s^k} \quad (4.36)$$

Mentre per il soddisfacimento delle specifiche lasche vengono introdotti i diagrammi universali, Figura 4.23, per la costruzione delle reti compensatrici :

- rete attenuatrice : comporta un decremento di modulo e fase
- rete anticipatrice : comporta un aumento di modulo e fase

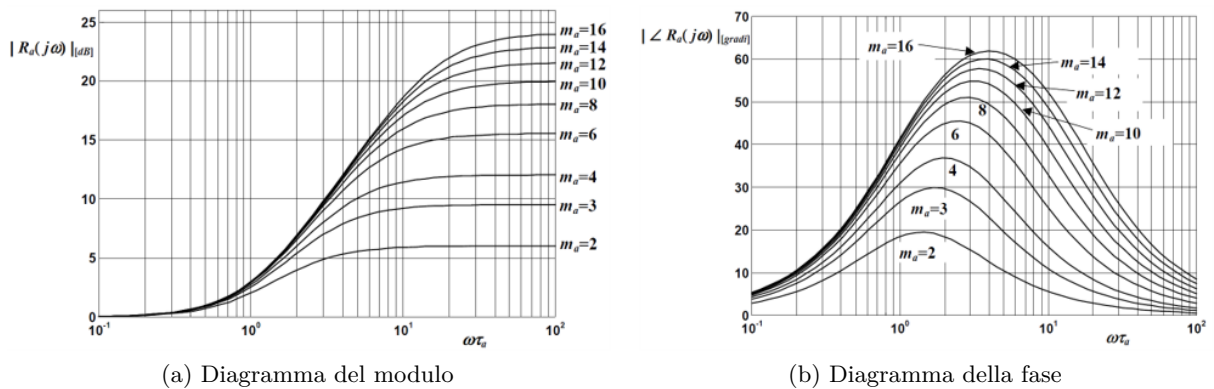


Figura 4.23: Diagrammi universali

Si noti che, per non compromettere l'efficacia del controllore di primo tentativo $\hat{G}(s)$ nell'esaudire le specifiche univoche, è bene che la funzione compensatrice $R(s)$ abbia guadagno statico nullo e che non presenti né poli né zeri nell'origine del piano complesso, bensì che il suo numeratore e denominatore siano composti solamente da:

- *termini binomi* della forma $1 + j\omega\tau$, dove τ rappresenta la costante di tempo del polo/zero associato a tale termine binomio.
- *termini trinomi* della formula $1 - \frac{\omega^2}{\omega_n^2} + 2j\omega\frac{\zeta}{\omega_n}$, dove ω_n e ζ rappresentano rispettivamente la pulsazione naturale (o di rottura) ed il coefficiente di smorzamento della coppia di poli/zeri coniugati associati a tale termine trinomio.

Terminata la sintesi del controllore, risulta utile utilizzare la carta di Nichols per verificare che il sistema in catena chiusa rispetti effettivamente le specifiche richieste.

4.3.2 Luogo delle radici

La sintesi tramite luogo delle radici è il metodo più rigoroso per valutare la stabilità di un sistema reazionato, partendo dalla conoscenza dei poli e degli zeri della funzione di trasferimento ad anello aperto $F(s)$. Inizialmente appunto abbiamo la funzione di trasferimento in catena aperta:

$$F(s) = \frac{K \prod_{i=1}^m (s - z_i)}{\prod_{i=1}^n (s - p_i)} \quad (4.37)$$

dove m indica il numero di zeri e n indica il numero di poli, ricordando che per un sistema reazionato, la funzione di trasferimento ad anello chiuso $W(s)$ è uguale a :

$$W(s) = \frac{F(s)}{1 + F(s)} = \frac{K \prod_{i=1}^m (s - z_i)}{\prod_{i=1}^n (s - p_i) + K \prod_{i=1}^m (s - z_i)} \quad (4.38)$$

Osservando il polinomio caratteristico $f(s, K)$:

$$f(s, K) = \prod_{i=1}^n (s - p_i) + K \prod_{i=1}^m (s - z_i) \quad (4.39)$$

si nota che:

- i poli della $W(s)$ dipendono dal guadagno K , dal numeratore $N(s)$ e dal denominatore $D(s)$ del processo $P(s)$
- nel caso $K = 0$ i poli della $W(s)$ sono esattamente i poli della $P(s)$

Si definisce luogo delle radici il luogo dei punti percorsi dalle radici dell'equazione $f(s, K) = 0$ nel piano di Gauss al variare di $k \in (-\infty, +\infty)$.

$$f(s, K) = \prod_{i=1}^n (s - p_i) + K \prod_{i=1}^m (s - z_i) = 0 \quad (4.40)$$

Per il tracciamento delle radici del polinomio caratteristico, Eq. 4.40, nel piano di Gauss si utilizzano delle regole fondamentali:

1. Il luogo delle radici è simmetrico rispetto all'asse reale
2. Il luogo delle radici è suddiviso in luogo positivo dove $K > 0$ e negativo dove $K < 0$
3. Tutto l'asse reale appartiene al luogo. In particolare appartengono al luogo positivo le porzioni di asse reale che lasciano alla propria destra un numero dispari di poli e/o zeri di $F(s)$ contati con la loro molteplicità e appartengono al luogo negativo quelle che lasciano alla propria destra un numero pari di poli e/o zeri

4. Per $k \rightarrow +\infty$ si ha che m rami convergono sugli zeri e gli altri $n - m$ rami divergono all'infinito lungo $n - m$ semirette, detti asintoti del luogo, che partono da un punto sull'asse reale, detto centro degli asintoti S_0

$$S_0 = \frac{\sum_{i=1}^n p_i - \sum_{i=1}^m z_i}{n - m} \quad (4.41)$$

5. I punti singolari sono le radici multiple dell'equazione $f(s, K) = 0$, tali che:

$$\begin{cases} f(s, K) = 0 \\ \left. \frac{\partial f(s, K)}{\partial s} \right|_{s=\bar{s}} = 0 \end{cases} \quad (4.42)$$

Nella sezione 4.5 sarà illustrato, come utilizzare il luogo delle radici per la progettazione di un controllore.

4.4 Realizzazione del controllore Pitch con la sintesi in frequenza

Andiamo ora ad utilizzare effettivamente le tecniche lineari di sintesi introdotte per la progettazione di controllori per la dinamica del Pitch e poi per la dinamica del Roll, partendo dalla tecnica di sintesi in frequenza.

Si parte con la definizione delle specifiche univoche e lasche che il sistema a ciclo chiuso deve soddisfare:

- **Specifiche univoche**
 - sistema di tipo 1
 - errore a regime permanente $|\bar{e}_1| \leq 0.025$
- **Specifiche lasche**
 - $M_r \leq 2dB$
 - $B_3 \approx 0.8Hz$

Parametri in c.c	Parametri c.a.	Parametri nel tempo in c.c.
$M_r \leq 2dB$	$m_\varphi \geq 47^\circ$	$\hat{s} \leq 0.7 = 70\%$
$B_3 \approx 0.8Hz$	$\omega_t \approx 3.2 \frac{rad}{s}$	$t_s = 0.596s$

Tabella 4.1: Parametri specifiche lasche

Partiamo dal soddisfare le specifiche univoche:

- il sistema deve essere di tipo 1, questo implica che $F(s) = G(s)P(s)$ deve necessariamente avere un polo in $s = 0$. Dall'Eq. 4.27 notiamo che $P(s)$ ha già un polo in $s = 0$ e quindi il controllore di primo tentativo è $\hat{G}(s) = K_G$.

4.4 Realizzazione del controllore Pitch con la sintesi in frequenza

- errore a regime permanente $|\tilde{e}_1| \leq 0.025$ ovvero l'errore relativo ad un ingresso polinomiale di grado 1 commesso da un sistema di tipo 1 e questo risulta essere uguale a $|\tilde{e}_1| = \frac{k_d^2}{K_F}$ dove k_d è il coefficiente di retroazione, nel nostro caso unitario ($k_d^2 = 1$). Otteniamo $|\tilde{e}_1| = \frac{1}{K_F} = \frac{1}{K_G K_P} \leq 0.025$ da cui ricaviamo $K_G \geq 0.0238$.

Scegliamo arbitrariamente $K_G = 0.0286$ ottenendo il controllore di primo tentativo $\hat{G}(s) = 0.0286$ quindi si ha che la funzione di trasferimento in catena diretta di primo tentativo $\hat{F}(s)$ risulta essere uguale a:

$$\hat{F}(s) = \hat{G}(s)P(s) = \frac{48}{s(1 + \frac{s}{1.619})} \quad (4.43)$$

Analizziamo ora i diagrammi di bode della nostra $\hat{F}(s)$, riportati in Figura 4.24, per poi implementare eventuali funzioni compensatrici per il soddisfacimento delle specifiche lasche.

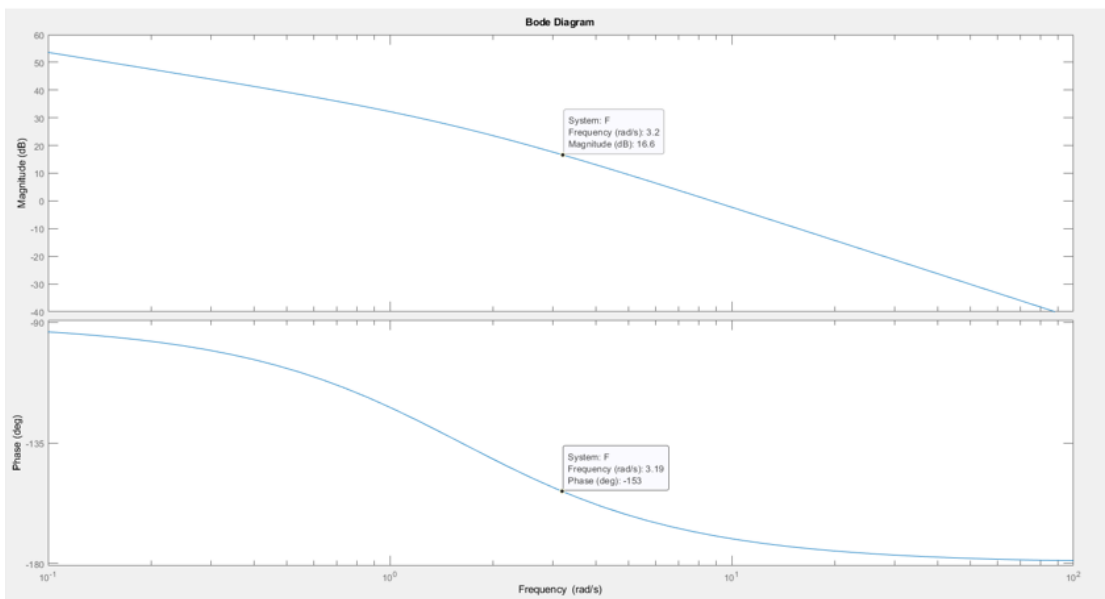


Figura 4.24: Diagrammi di bode

Come vediamo in corrispondenza della pulsazione di attraversamento desiderata $\omega_t = 3.2 \text{ rad/s}$ si hanno i seguenti valori di modulo e fase:

$$\begin{aligned} |\hat{F}(j\omega_t)|_{dB} &= 16.6 \text{ dB} \\ \angle F(j\omega_t) &= -153^\circ \end{aligned} \quad (4.44)$$

mentre i valori desiderati di $F(s)$ in corrispondenza della pulsazione di attraversa-

mento sono :

$$\begin{aligned} |\hat{F}(j\omega_t)|_{dB} &= 0dB \\ \angle F(j\omega_t) &\geq -180^\circ + m_\varphi = -133^\circ \end{aligned} \quad (4.45)$$

Per soddisfare le specifiche dell'Eq 4.45 dobbiamo utilizzare due funzioni compensatrici, applichiamo prima una funzione anticipatrice $R_a(s)$ per aumentare la fase in modo che $\angle F(j\omega_t) \geq -133^\circ$ senza aumentare troppo il modulo e successivamente applichiamo una funzione attenuatrice $R_i(s)$ per diminuire il modulo della $\angle F(j\omega_t)$ fino a $0dB$ senza però diminuire troppo la fase.

Implementiamo come appena detto la rete anticipatrice, ricorriamo ai diagrammi universali riportati in Figura 4.23 e scegliendo i seguenti valori:

$$m_a = 16, \quad \omega_t \tau_a = 1 \quad \Rightarrow \quad \omega_a = 3.2, \quad m_a \omega_a = 51.2 \quad (4.46)$$

Otteniamo la seguente funzione anticipatrice

$$R_a(s) = \frac{1 + \frac{s}{\omega_a}}{1 + \frac{s}{m_a \omega_a}} = \frac{1 + \frac{s}{3.2}}{1 + \frac{s}{51.2}} \quad (4.47)$$

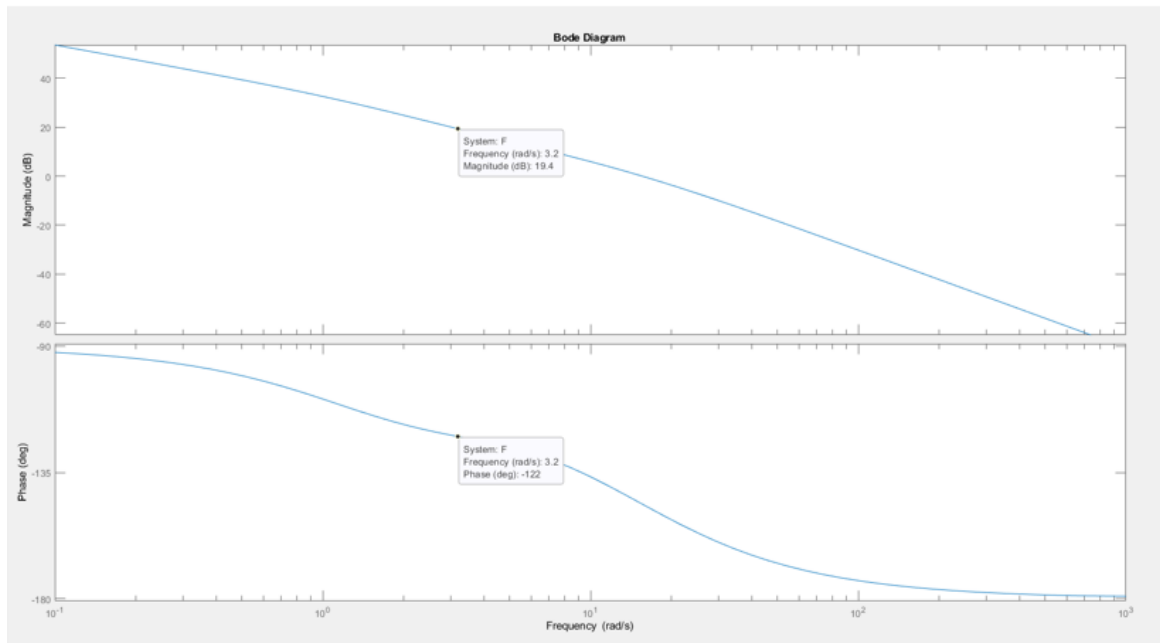


Figura 4.25: Diagrammi di bode - rete anticipatrice

Otteniamo i seguenti valori :

$$\begin{aligned} |\hat{F}(j\omega_t)|_{dB} &= 19.4dB \\ \angle F(j\omega_t) &= -122^\circ \end{aligned} \quad (4.48)$$

4.4 Realizzazione del controllore Pitch con la sintesi in frequenza

Adesso utilizziamo la funzione attenuatrice per ottenere il modulo desiderato, scegliamo come valori:

$$m_i = 10, \quad \omega_t \tau_i = 100 \quad \Rightarrow \quad \omega_i = 0.32, \quad m_i \omega_i = 0.032 \quad (4.49)$$

Otteniamo la seguente funzione attenuatrice

$$R_i(s) = \frac{1 + \frac{s}{m_i \omega_i}}{1 + \frac{s}{\omega_i}} = \frac{1 + \frac{s}{0.032}}{1 + \frac{s}{0.32}} \quad (4.50)$$

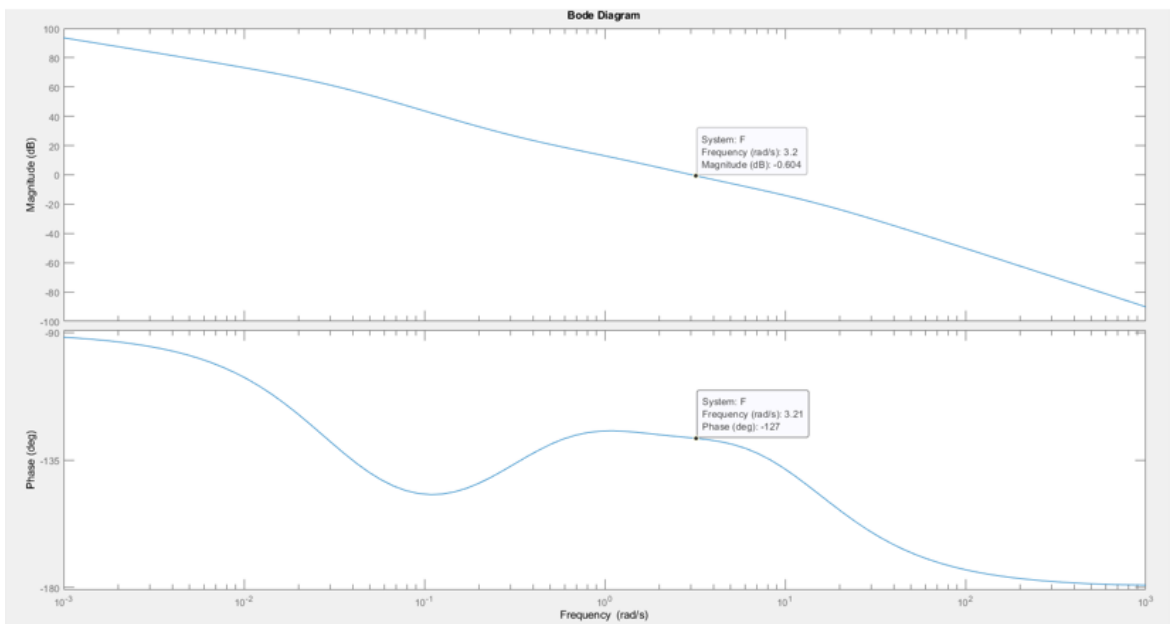


Figura 4.26: Diagrammi di bode - rete attenuatrice

Otteniamo i seguenti valori :

$$\begin{aligned} |\hat{F}(j\omega_t)|_{dB} &= -0.6dB \\ \angle F(j\omega_t) &= -127^\circ \end{aligned} \quad (4.51)$$

Concludiamo dicendo che le specifiche vengono soddisfatte da un controllore del tipo:

$$G(s) = K_G R_a(s) R_i(s) = \frac{0.0286(1 + \frac{s}{3.2})(1 + \frac{s}{0.32})}{(1 + \frac{s}{51.2})(1 + \frac{s}{0.032})} \quad (4.52)$$

Vediamo anche come dalla Carta di Nichols della $F(s)$, riportato in Figura 4.27, le specifiche lasche sono soddisfatte.

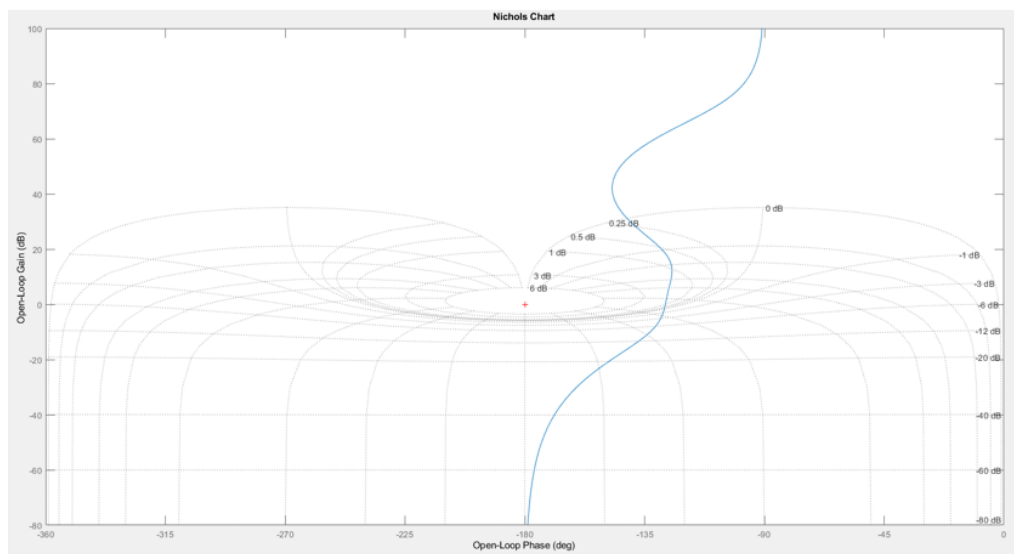


Figura 4.27: Carta di Nichols

Analizziamo quindi attraverso lo strumento *sisotool* di MATLAB la funzione di trasferimento in catena diretta $F(s) = G(s)P(s)$ ottenuta tramite la sintesi di tale controllore. Notiamo dalla Figura 4.28 come il margine di fase pari a 52.8° sia ben al di sopra dei 47° richiesti dalle specifiche. Dalla risposta al gradino, otteniamo nel dominio del tempo una sovraelongazione \hat{s} pari al 21%, anch'essa ben inferiore al 70% delle specifiche e un tempo di salita $t_s = 0.4$ circa uguale a quello delle specifiche.

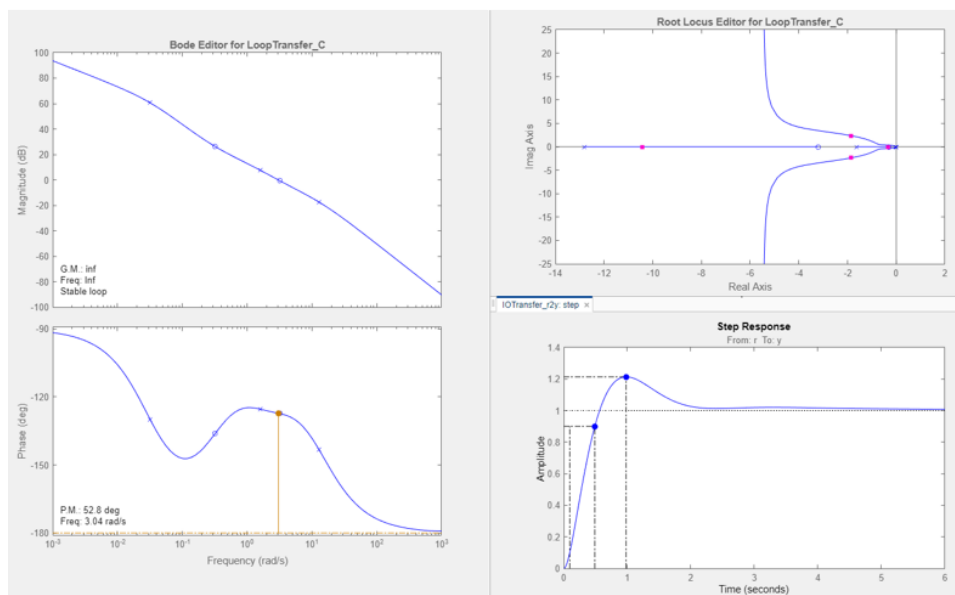


Figura 4.28: *SystemControlDesigner*

Concluso il primo controllore ci interessa ora verificare quali siano le sue prestazioni nel controllo dell'assetto del quadricottero in simulazione rispetto ai riferimenti impartiti Figura 4.16. Riprendiamo il progetto *asbQuadcopterStart* nello specifico

4.4 Realizzazione del controllore Pitch con la sintesi in frequenza

entriamo nel blocco *Flight Control System/Controller/ Attitude* rispetto alla Figura 3.22 viene modificata la parte del controllo del pitch mentre il controllo del roll rimane gestito dal PID di default.

All'interno della Figura 4.29 è presente anche un blocco *scope* che permette di visualizzare a schermo il segnale d'errore del pitch, il segnale del valore effettivo dell'angolo di pitch confrontato con il segnale di riferimento del pitch.

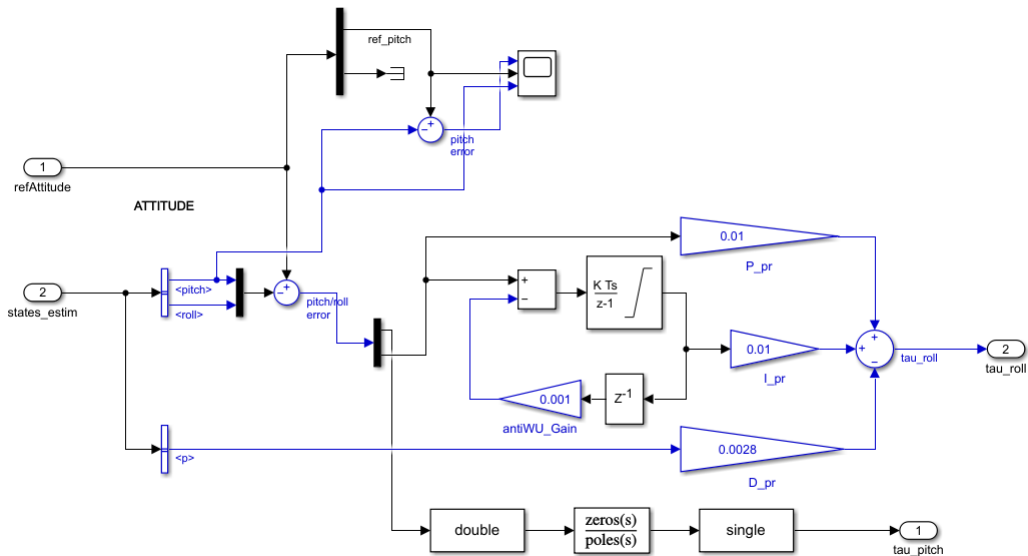


Figura 4.29: Blocco Attitude - Sintesi in Frequenza

Nella Figura 4.51a vengono inseriti poli, zeri e coefficiente di guadagno del controllore implementato precedentemente per la dinamica del pitch.(Eq. 4.52)

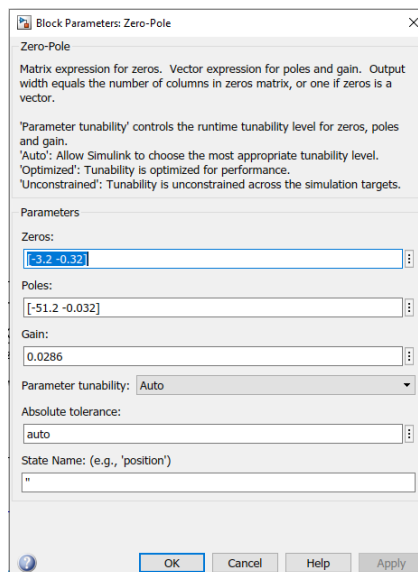


Figura 4.30: Blocco Parameters : Zero-Pole

Di seguito vengono riportati i grafici (Figura 4.31) delle prestazioni del controllore precedentemente realizzato, viene riportato sia l'errore della dinamica del pitch dovuto alla differenza tra riferimento e valore effettivo e viene riportato anche grafico del pitch confrontato con il suo riferimento.

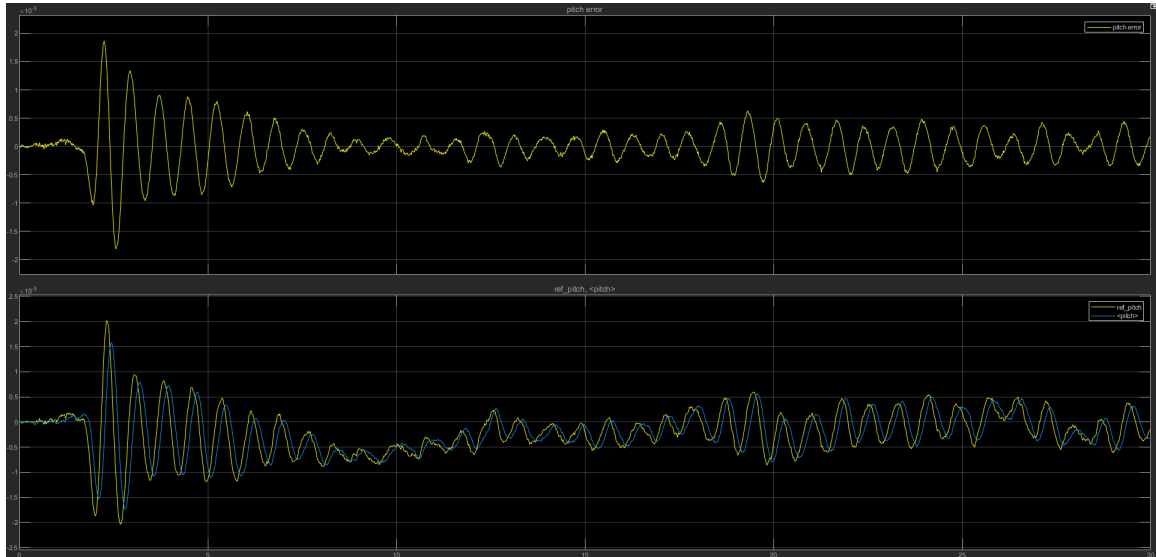


Figura 4.31: Sintesi in frequenza - Pitch

Facciamo alcune considerazioni osserviamo che il controllore realizzato per i riferimenti impartiti in Figura 4.16:

- Per quanto riguarda l'errore di picco siamo riusciti a diminuirlo di molto rispetto ai PID originali
- l'errore a regime permanente risulta essere nullo
- il tempo di salita potrebbe essere migliorato per seguire meglio il segnale
- non ha un tempo di assestamento ottimale ma è soddisfacente

Tutto sommato riesce a seguire efficacemente il riferimento, ci riteniamo quindi soddisfatti.

4.4.1 Realizzazione del controllore Pitch con il luogo delle radici

In questa sezione andremo ad implementare la tecnica di sintesi con il luogo delle radici descritta prima per la progettazione di controllori per la dinamica del pitch. Si parte, come per la sintesi in frequenza, dalle specifiche che il sistema a ciclo chiuso deve soddisfare:

- Specifiche univoche
 - sistema di tipo 1

4.4 Realizzazione del controllore Pitch con la sintesi in frequenza

– errore a regime permanente $|\tilde{e}_1| \leq 0.025$

- Specifiche lasche

– tutti i poli p in catena chiusa devono avere $R_e(p) \leq -0.5$

Per le specifiche univoche lo svolgimento è lo stesso applicato precedentemente per la sintesi in frequenza. Quindi abbiamo come controllore di primo tentativo $\hat{G}(s) = K_G$, mentre per quanto riguarda l'errore $\tilde{e}_1 \leq 0.025$ risulta essere uguale a $|\tilde{e}_1| = \frac{k_d^2}{K_F}$ dove k_d è il coefficiente di retroazione, nel nostro caso unitario ($k_d^2 = 1$). Otteniamo quindi $|\tilde{e}_1| = \frac{1}{K_F} \leq 0.025$ da cui ricaviamo $K_G = K \cdot R(0) \geq 0.0238$, dove $R(0)$ è la rete compensatrice $R(s)$ considerata in $s = 0$.

Prima di definire $R(s)$ utilizziamo lo strumento *rlocus* per tracciare il luogo delle radici in Matlab.

Dal controllore di primo tentativo $\hat{G}(s)$ otteniamo che la funzione di trasferimento in catena diretta di primo tentativo $\hat{F}(s)$ è uguale a :

$$\hat{F}(s) = \hat{G}(s) \cdot P(s) = \frac{2717 \cdot K_G}{s(s + 1.619)} \quad (4.53)$$

Dopo aver definito $\hat{F}(s)$ digitando nella linea di comando di Matlab *rlocus*($\hat{F}(s)$) verrà tracciato il luogo delle radici della funzione di trasferimento in catena diretta

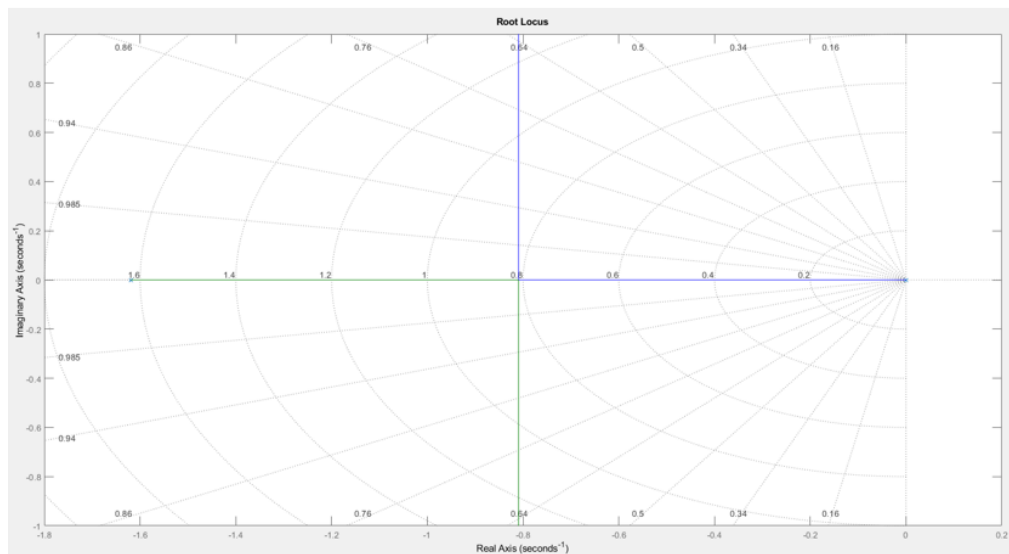


Figura 4.32: Luogo delle Radici - Controllore di primo tentativo

Come notiamo dalla Figura 4.32 le specifiche lasche non vengo soddisfatte ovvero non tutti i poli in catena chiusa risultano essere ≤ -0.5 e quindi necessario introdurre una rete compensatrice che porti il sistema in catena chiusa ad assumere il comportamento desiderato.

Notiamo subito che la nostra funzione ha 2 poli, $p_1 = 0$ e $p_2 = -1.619$ e nessuno zero questo comporta che, indicando con n il numero di poli e con m il numero degli

zeri, lo scarto sia pari a $n - m = 2$, quindi con l'introduzione della rete compesatrice dobbiamo mantenere $n - m$ il più piccolo possibile ovvero uguale a 2.

Per soddisfare le specifiche lasche dobbiamo necessariamente spostare il centro degli asintoti e quindi è necessario introdurre una coppia polo-zero e per mantenere $n - m = 2$ bisogna aggiungere una seconda coppia polo-zero.

Per queste ragioni la nostra rete compesatrice avrà la seguente forma:

$$R(s) = \frac{(s - z_1)(s - z_2)}{(s - p_1)(s - p_2)} \quad (4.54)$$

Scegliamo a questo punto la posizione del centro degli asintoti in modo arbitrario ovvero $S_0 < -24$, ricordando che :

$$S_0 = \frac{\sum_{i=1}^n p_i - \sum_{i=1}^m z_i}{n - m} \quad (4.55)$$

si ottiene

$$S_0 = \frac{-0 - 1.619 - p_1 - p_2 + z_1 + z_2}{2} \quad (4.56)$$

In modo opportuno scegliamo poli e zeri della rete compesatrice $R(s)$ in modo da soddisfare l'Eq. 4.56 :

$$p_1 = -0.03 \quad p_2 = -50 \quad z_1 = -0.5 \quad z_2 = -2.5 \quad (4.57)$$

Otteniamo quindi una rete compesatrice:

$$R(s) = \frac{(s + 0.5)(s + 2.5)}{(s + 0.03)(s + 50)} \quad (4.58)$$

Ora che abbiamo progettato la $R(s)$ del controllore, riprendendo la specifica $K_G = K \cdot R(0) \geq 0.0238$, essendo $R(0) = 0.8333$ si ha che:

$$K \geq \frac{0.0238}{R(0)} = 0.028 \quad (4.59)$$

Scegliamo arbitrariamente $K = 0.028$ da cui otteniamo il controllore complessivo $G(s)$:

$$G(s) = K \cdot R(s) = \frac{0.028(s + 0.5)(s + 2.5)}{(s + 0.03)(s + 50)} \quad (4.60)$$

Riutilizziamo *'rlocus* per visualizzare il luogo delle radici della funzione di trasferimento composta dal controllore finale $G(s)$

4.4 Realizzazione del controllore Pitch con la sintesi in frequenza

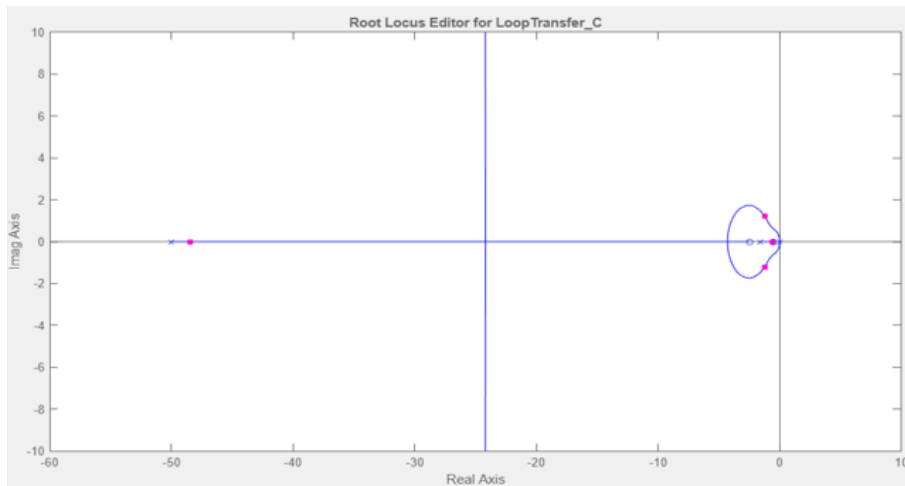


Figura 4.33: Luogo delle radici-Controllore finale

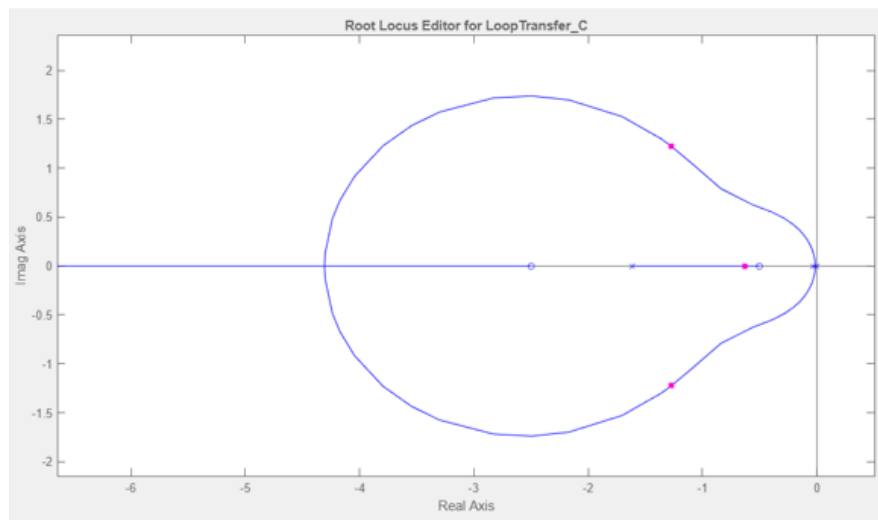


Figura 4.34: Luogo delle radici-Controllore finale

Adesso controlliamo i poli a parte reale in catena chiusa se soddisfano le specifiche, utilizzando lo strumento *pole*, Figura 4.35, su Matlab :

```

Command Window
>> s=tf('s');
>> num=(2717*0.028*(s+0.5)*(s+2.5));
>> den=(s*(s+1.619)*(s+50)*(s+0.03)+(2717*0.028*(s+0.5)*(s+2.5));
>> W=num/den;
>> pole(W)

ans =

-48.4751 + 0.00001i
-1.2722 + 1.22371i
-1.2722 - 1.22371i
-0.6296 + 0.00001i

f >>
    
```

Figura 4.35: Poli del sistema a ciclo chiuso

Inseriamo il controllore appena realizzato nel progetto *asbQuadcopterStart* precisamente nel blocco *Attitude*, come fatto per la sintesi in frequenza, nel blocco *zeros-poles*. Riportiamo i valori degli zeri, dei poli e del coefficiente di guadagno stabiliti precedentemente e facciamo partire la simulazione. (Figura 4.36)

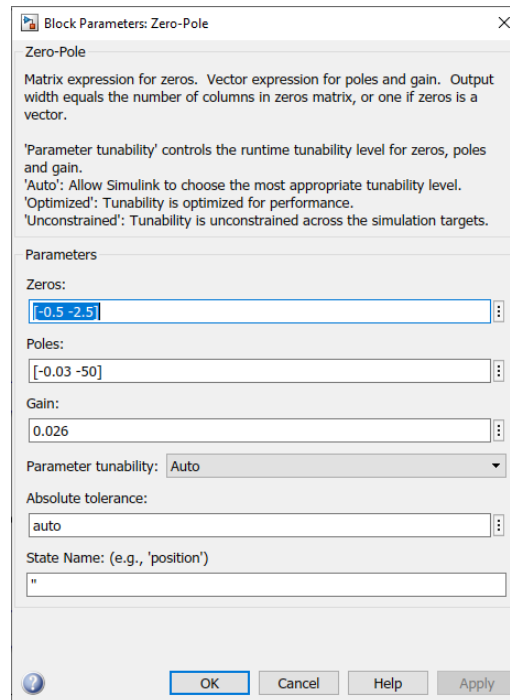


Figura 4.36: Blocco Parameters : Zero-Pole

Nel blocco scope possiamo notare la Figura 4.37 riguardo le prestazioni del controllore realizzato rispetto all'angolo di pitch.

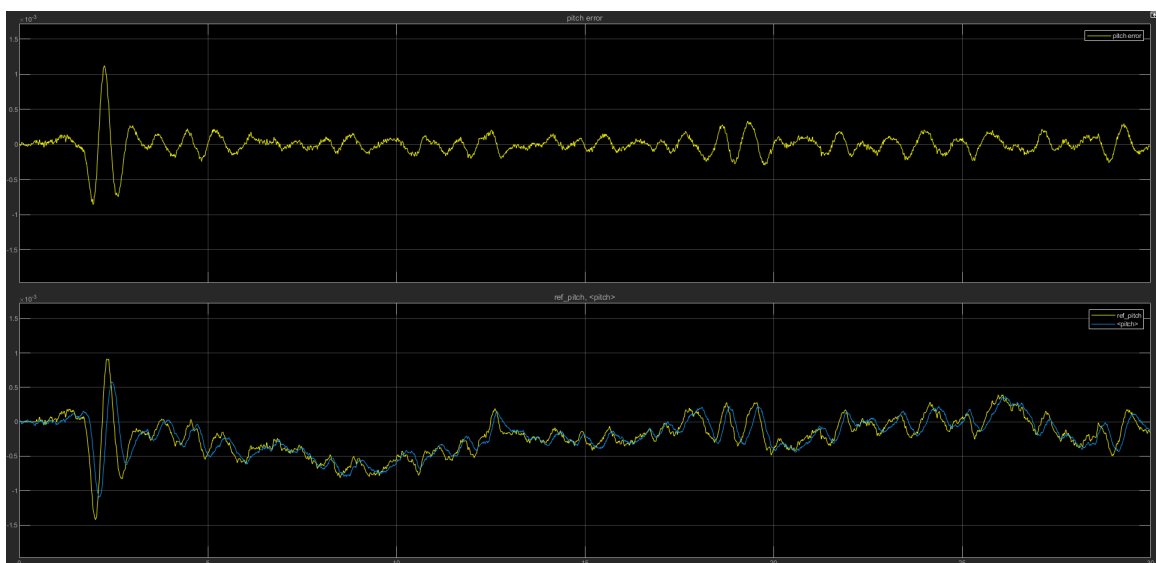


Figura 4.37: Luogo delle radici- Pitch

4.5 Realizzazione del controllore Roll con la sintesi in frequenza

Con il controllore realizzato tramite la sintesi con luogo delle radici possiamo notare che :

- l'errore a regime permanente è nullo quindi rispetta le specifiche
- l'errore di picco è stato diminuito rispetto ai PID originali del progetto
- considerato il fattore di scala di 10^{-3} per i valori sull'asse delle ordinate, il ridotto tempo di salita porta a delle buone prestazioni rispetto al riferimento

Ci possiamo ritenere soddisfatti.

4.5 Realizzazione del controllore Roll con la sintesi in frequenza

In questa sezione andremo ad implementare la tecnica di sintesi in frequenza descritta prima per la progettazione di controllori per la dinamica del Roll.

Si parte con la definizione delle specifiche univoche e lasche che il sistema a ciclo chiuso deve soddisfare:

- **Specifiche univoche**
 - sistema di tipo 1
 - errore a regime permanente $|\tilde{e}_1| \leq 0.025$
- **Specifiche lasche**
 - $M_r \leq 2dB$
 - $B_3 \approx 0.5Hz$

Parametri in c.c	Parametri c.a.	Parametri nel tempo in c.c.
$M_r \leq 2dB$	$m_\varphi \geq 47^\circ$	$\hat{s} \leq 0.7 = 70\%$
$B_3 \approx 0.5Hz$	$\omega_t \approx 2 \frac{rad}{s}$	$t_s = 0.955s$

Tabella 4.2: Parametri specifiche lasche Roll

Iniziamo a questo punto ad esaudire le specifiche univoche: per la specifica relativa al tipo del sistema, essa richiede che la funzione di trasferimento in catena diretta $F(s) = G(s)P(s)$ abbia già un polo in $s = 0$. Notiamo però dall'espressione del processo $P(s)$ dell'Eq.4.29 che questo presenta già di per sé un polo nell'origine, motivo per il quale scegliamo di non introdurne un secondo anche nell'espressione della $G(s)$. Questo porta ad un controllore di primo tentativo della forma $\hat{G}(s) = K_G$. Riguardo invece la seconda specifica univoca, l'errore a regime permanente $|\tilde{e}_1| \leq 0.025$, ovvero l'errore relativo ad un ingresso polinomiale di grado 1 commesso da un

sistema di tipo 1 e questo risulta essere uguale a $|\tilde{e}_1| = \frac{k_d^2}{K_F}$ dove k_d è il coefficiente di retroazione, nel nostro caso unitario ($k_d^2 = 1$). Otteniamo $|\tilde{e}_1| = \frac{1}{K_F} = \frac{1}{K_G K_P} \leq 0.025$ da cui ricaviamo $K_G \geq 0.0238$.

Scegliamo arbitrariamente $K_G = 0.025$ ottenendo il controllore di primo tentativo $\hat{G}(s) = 0.025$ quindi si ha che la funzione di trasferimento in catena diretta di primo tentativo $\hat{F}(s)$ risulta essere uguale a:

$$\hat{F}(s) = \hat{G}(s)P(s) = \frac{91.1}{s(s + 2.172)} \quad (4.61)$$

Analizziamo ora i diagrammi di bode della nostra $\hat{F}(s)$, riportati in Figura 4.38, per poi implementare eventuali funzioni compensatrici per il soddisfacimento delle specifiche lasche.

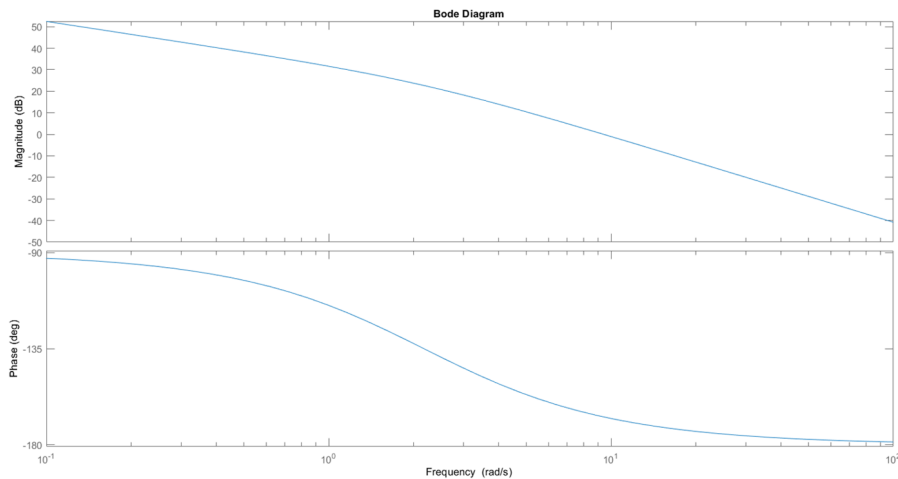


Figura 4.38: Diagrammi di bode Roll

Come vediamo in corrispondenza della pulsazione di attraversamento desiderata $\omega_t = 2 \text{ rad/s}$ si hanno i seguenti valori di modulo e fase:

$$\begin{aligned} |\hat{F}(j\omega_t)|_{dB} &= 23.8dB \\ \angle F(j\omega_t) &= -133^\circ \end{aligned} \quad (4.62)$$

mentre i valori desiderati di $F(s)$ in corrispondenza della pulsazione di attraversamento sono :

$$\begin{aligned} |\hat{F}(j\omega_t)|_{dB} &= 0dB \\ \angle F(j\omega_t) &\geq -180^\circ + m_\varphi = -133^\circ \end{aligned} \quad (4.63)$$

Si osserva che, benché la specifica relativa $\angle F(j\omega_t) \geq -133^\circ$ risulti rispettata, per

4.5 Realizzazione del controllore Roll con la sintesi in frequenza

soddisfare le specifiche dell'Eq 4.63 , sarà sicuramente necessario progettare una rete attenuatrice $R_i(s)$ per portare il modulo della $F(j\omega_t)$ pari a 0dB. Essendo tuttavia che un rete attenuatrice comporta sempre, per quanto piccola, anche una diminuzione della fase, si prevede l'esigenza di dover progettare anche una rete anticipatrice $R_a(s)$ al fine di controbilanciare il ritardo di fase indesiderato introdotto dalla $R_i(s)$. Implementiamo come appena detto la rete anticipatrice, ricorriamo ai diagrammi universali riportati in Figura 4.23 e scegliendo i seguenti valori:

$$m_a = 18, \quad \omega_t \tau_a = 0.5 \quad \Rightarrow \quad \omega_a = 4, \quad m_a \omega_a = 72 \quad (4.64)$$

Otteniamo la seguente funzione anticipatrice:

$$R_a(s) = \frac{1 + \frac{s}{\omega_a}}{1 + \frac{s}{m_a \omega_a}} = \frac{1 + \frac{s}{4}}{1 + \frac{s}{72}} \quad (4.65)$$

Otteniamo i seguenti valori :

$$\begin{aligned} |\hat{F}(j\omega_t)|_{dB} &= 24.7dB \\ \angle F(j\omega_t) &= -108^\circ \end{aligned} \quad (4.66)$$

Procediamo analogamente anche per la rete attenuatrice, per la quale si sceglie:

$$m_i = 15, \quad \omega_t \tau_i = 400 \quad \Rightarrow \quad \omega_i = 0.005, \quad m_i \omega_i = 0.075 \quad (4.67)$$

Otteniamo la seguente funzione attenuatrice:

$$R_i(s) = \frac{1 + \frac{s}{m_i \omega_i}}{1 + \frac{s}{\omega_i}} = \frac{1 + \frac{s}{0.075}}{1 + \frac{s}{0.005}} \quad (4.68)$$

Otteniamo i seguenti valori :

$$\begin{aligned} |\hat{F}(j\omega_t)|_{dB} &= 1.23dB \\ \angle F(j\omega_t) &= -110^\circ \end{aligned} \quad (4.69)$$

Concludiamo dicendo che le specifiche vengono soddisfatte da un controllore del tipo :

$$G(s) = K_G R_a(s) R_i(s) = \frac{0.03(s+4)(s+0.075)}{(s+72)(s+0.005)} \quad (4.70)$$

Vediamo anche come dalla Carta di Nichols della $F(s)$, riportato in Figura 4.39, le specifiche lasche sono soddisfatte.

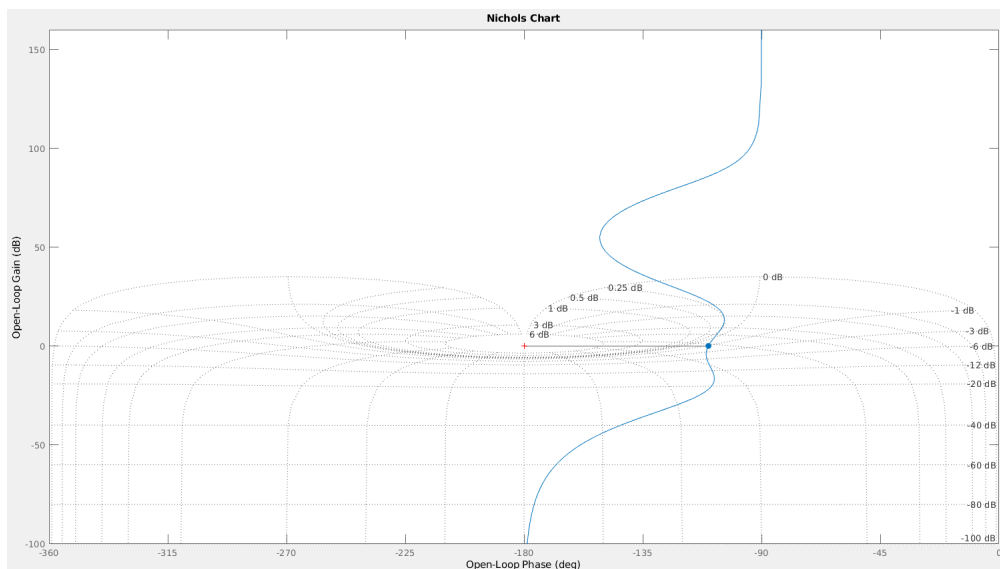


Figura 4.39: Carta di Nichols

Analizziamo quindi attraverso lo strumento *sisotool* di MATLAB la funzione di trasferimento in catena diretta $F(s) = G(s)P(s)$ ottenuta tramite la sintesi di tale controllore. Notiamo dalla Figura 4.40 come il margine di fase pari a 69.8° sia ben al di sopra dei 47° richiesti dalle specifiche. Dalla risposta al gradino, otteniamo nel dominio del tempo una sovravelongazione \hat{s} pari al 7%, anch'essa ben inferiore al 70% delle specifiche e un tempo di salita $t_s = 0.984$ circa uguale a quello delle specifiche.

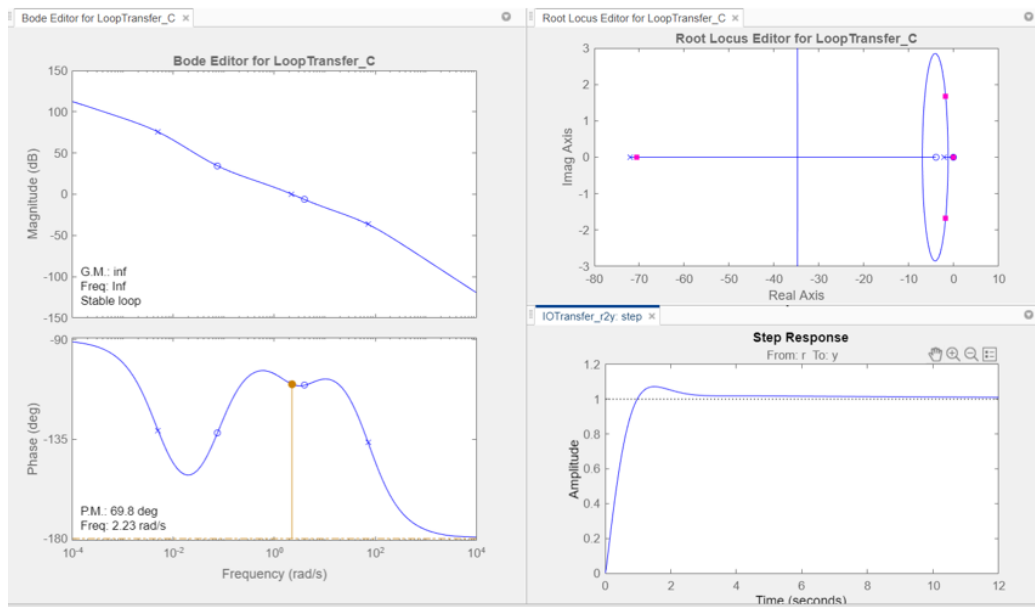


Figura 4.40: *SystemControlDesigner*

Concluso il primo controllore ci interessa ora verificare quali siano le sue prestazioni nel controllo dell'assetto del quadricottero in simulazione rispetto ai riferimenti

4.5 Realizzazione del controllore Roll con la sintesi in frequenza

impartiti Figura 4.16. Riprendiamo il progetto *asbQuadcopterStart* nello specifico entriamo nel blocco *Flight Control System/Controller/ Attitude* rispetto alla Figura 3.22 viene modificata la parte del controllo del roll mentre il controllo del pitch rimane gestito dal PID di default.

All'interno della Figura 4.41 è presente anche un blocco *scope* che permette di visualizzare a schermo il segnale d'errore del roll, il segnale del valore effettivo dell'angolo di roll confrontato con il segnale di riferimento del roll.

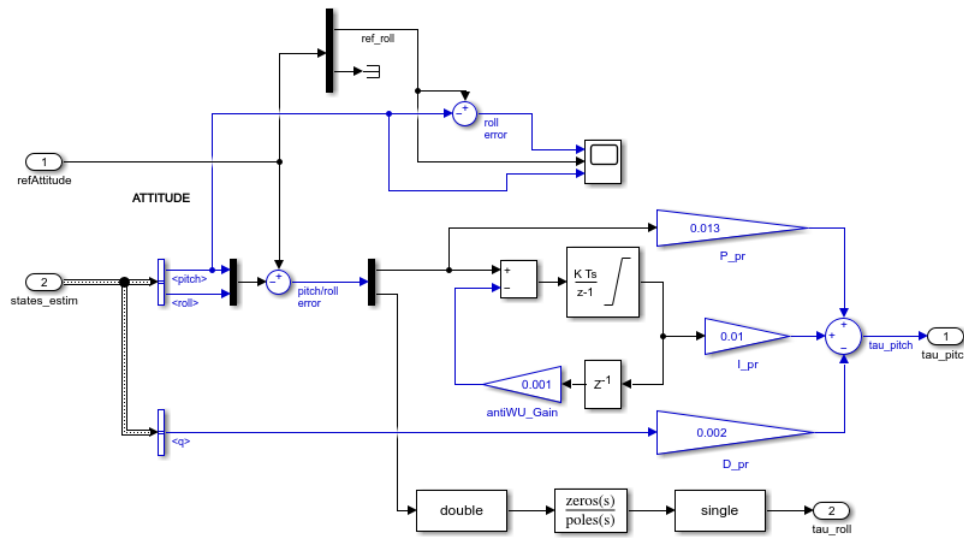


Figura 4.41: Blocco Attitude - Sintesi in Frequenza

Nella Figura 4.42 vengono inseriti poli, zeri e coefficiente di guadagno del controllore implementato precedentemente per la dinamica del roll.(Eq. 4.70)

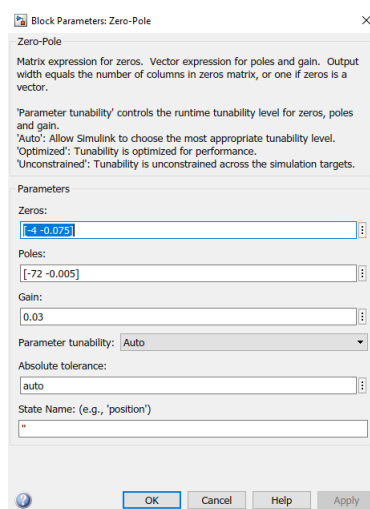


Figura 4.42: Blocco Parameters : Zero-Pole

Di seguito vengono riportati i grafici (Figura 4.43) delle prestazioni del controllore precedentemente realizzato, viene riportato sia l'errore della dinamica del roll dovuto alla differenza tra riferimento e valore effettivo e viene riportato anche il grafico del roll confrontato con il suo riferimento.

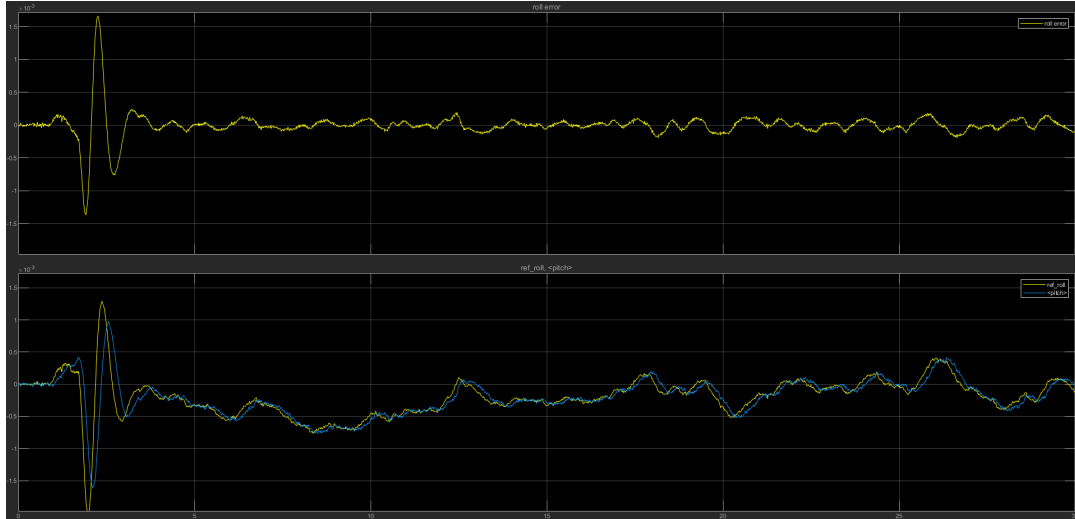


Figura 4.43: Sintesi in frequenza - Roll

Possiamo osservare che il controllore realizzato per riferimenti costanti:

- riesce a seguire efficacemente il riferimento variabile.
- presenta una sovra-elongazione \hat{S} molto bassa, che però non è richiesta, mentre il tempo di salita t_s deve essere migliorato per seguire meglio in segnale.

Tutto sommato riesce a seguire efficacemente il riferimento, ci riteniamo quindi soddisfatti.

4.5.1 Realizzazione del controllore Roll con il luogo delle radici

In questa sezione andremo ad implementare la tecnica di sintesi con il luogo delle radici descritta prima per la progettazione di controllori per la dinamica del Roll. Si parte, come per la sintesi in frequenza, dalle specifiche che il sistema a ciclo chiuso deve soddisfare:

- Specifiche univoche
 - sistema di tipo 1
 - errore a regime permanente $|\tilde{e}_1| \leq 0.025$
- Specifiche lasche
 - tutti i poli p in catena chiusa devono avere $R_e(p) \leq -1$

4.5 Realizzazione del controllore Roll con la sintesi in frequenza

Per le specifiche univoche lo svolgimento è lo stesso applicato precedentemente per la sintesi in frequenza. Quindi abbiamo come controllore di primo tentativo $\hat{G}(s) = K_G$, mentre per quanto riguarda l'errore $\tilde{e}_1 \leq 0.025$ risulta essere uguale a $|\tilde{e}_1| = \frac{k_d^2}{K_F}$ dove k_d è il coefficiente di retroazione, nel nostro caso unitario ($k_d^2 = 1$). Otteniamo quindi $|\tilde{e}_1| = \frac{1}{K_F} \leq 0.025$ da cui ricaviamo $K_G = K \cdot R(0) \geq 0.0238$, dove $R(0)$ è la rete compensatrice $R(s)$ considerata in $s = 0$.

Prima di definire $R(s)$ utilizziamo lo strumento *rlocus* per tracciare il luogo delle radici in Matlab.

Dal controllore di primo tentativo $\hat{G}(s)$ otteniamo che la funzione di trasferimento in catena diretta di primo tentativo $\hat{F}(s)$ è uguale a :

$$\hat{F}(s) = \hat{G}(s) \cdot P(s) = \frac{3644 \cdot K_G}{s(s + 2.172)} \quad (4.71)$$

Dopo aver definito $\hat{F}(s)$ digitando nella linea di comando di Matlab *rlocus*($\hat{F}(s)$) verrà tracciato il luogo delle radici della funzione di trasferimento in catena diretta

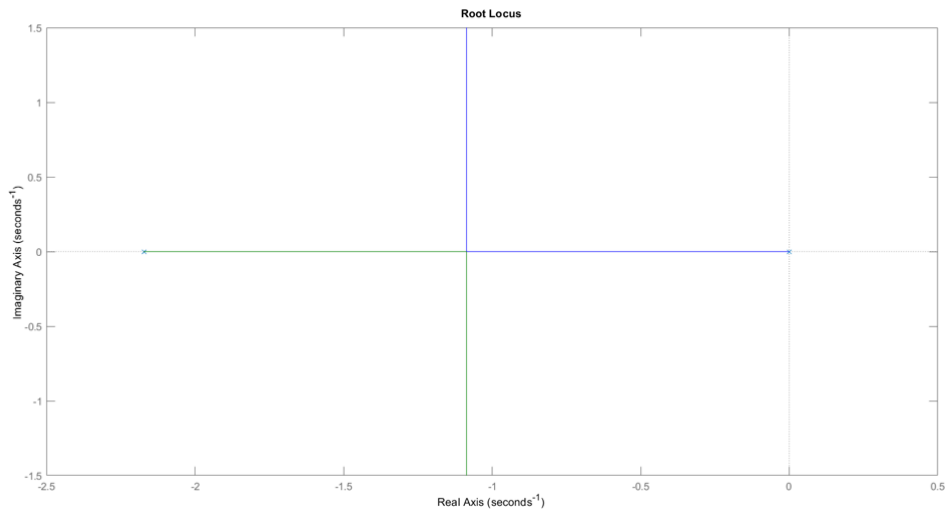


Figura 4.44: Luogo delle Radici - Controllore di primo tentativo

Come notiamo dalla Figura 4.44 le specifiche lasche non vengo soddisfatte ovvero non tutti i poli in catena chiusa risultano essere ≤ -1 e quindi necessario introdurre una rete compensatrice che porti il sistema in catena chiusa ad assumere il comportamento desiderato.

Notiamo subito che la nostra funzione ha 2 poli, $p_1 = 0$ e $p_2 = -2.172$ e nessuno zero questo comporta che, indicando con n il numero di poli e con m il numero degli zeri, lo scarto sia pari a $n - m = 2$, quindi con l'introduzione della rete compatrice dobbiamo mantenere $n - m$ il più piccolo possibile ovvero uguale a 2.

Per soddisfare le specifiche lasche dobbiamo necessariamente spostare il centro degli

asintoti e quindi è necessario introdurre una coppia polo-zero e per mantenere $n - m = 2$ bisogna aggiungere una seconda coppia polo-zero.

Per queste ragioni la nostra rete compensatrice avrà la seguente forma:

$$R(s) = \frac{(s - z_1)(s - z_2)}{(s - p_1)(s - p_2)} \quad (4.72)$$

e scegliamo i valori di z_1 , z_2 , p_1 e p_2 in base al posizionamento desiderato del centro degli asintoti S_0 sul piano complesso. Scegliendo di localizzare quest'ultimo in un punto s appartenente al semipiano complesso $R_e(s) < 20$ ed essendo esso sempre giacente sull'asse reale, si ottiene per S_0 la seguente condizione da verificare:

$$S_0 = \frac{\sum_{i=1}^n p_i - \sum_{i=1}^m z_i}{n - m} \quad (4.73)$$

si ottiene:

$$S_0 = \frac{-0 - 2.172 - p_1 - p_2 + z_1 + z_2}{2} < -20 \quad (4.74)$$

In modo opportuno scegliamo poli e zeri della rete compensatrice $R(s)$ in modo da soddisfare l'Eq. 4.74 :

$$p_1 = -0.03 \quad p_2 = -45 \quad z_1 = -0.65 \quad z_2 = -2.5 \quad (4.75)$$

Otteniamo quindi una rete compensatrice:

$$R(s) = \frac{(s + 0.65)(s + 2.5)}{(s + 0.03)(s + 45)} \quad (4.76)$$

Ora che abbiamo progettato la $R(s)$ del controllore, riprendendo la specifica $K_G = K \cdot R(0) \geq 0.0238$, essendo $R(0) = 1.2037$ si ha che:

$$K \geq \frac{0.0238}{R(0)} = 0.019807 \quad (4.77)$$

Scegliamo arbitrariamente $K = 0.021$ da cui otteniamo il controllore complessivo $G(s)$:

$$G(s) = K \cdot R(s) = \frac{0.021(s + 0.65)(s + 2.5)}{(s + 0.03)(s + 45)} \quad (4.78)$$

Riutilizziamo *'rlocus* per visualizzare il luogo delle radici della funzione di trasferimento composta dal controllore finale $G(s)$

4.5 Realizzazione del controllore Roll con la sintesi in frequenza

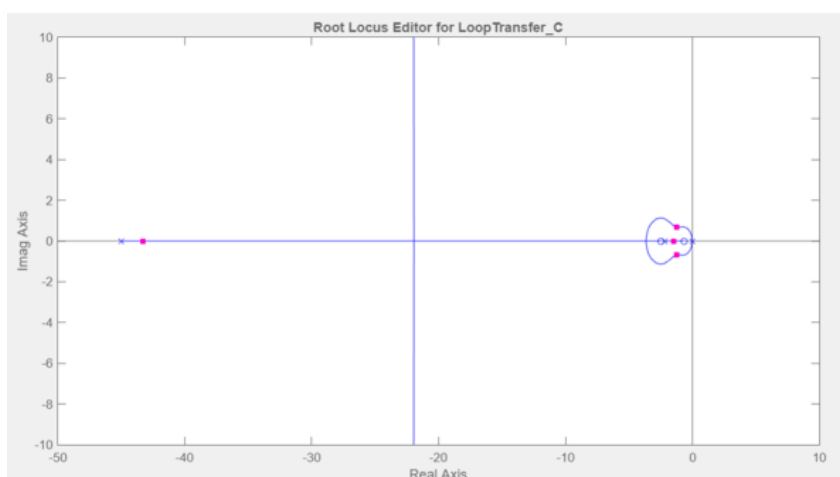


Figura 4.45: Luogo delle radici-Controllore finale

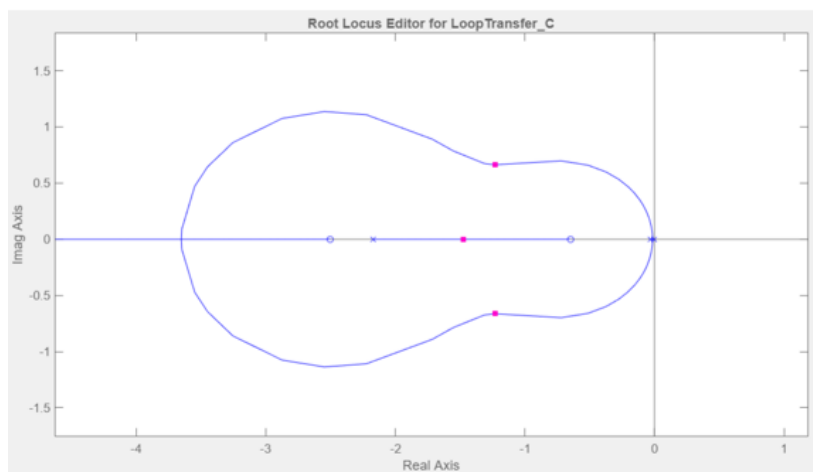


Figura 4.46: Luogo delle radici-Controllore finale

Adesso controlliamo i poli a parte reale in catena chiusa se soddisfano le specifiche, utilizzando lo strumento *pole*, Figura 4.47, su Matlab :

```

Command Window
>> s = tf('s');
>> num = 76.524*(s+2.5)*(s+0.65);
>> den = s*(s+2.172)*(s+45)*(s+0.03) + 76.524*(s+2.5)*(s+0.65);
>> Ws = num/den;
>> pole(Ws)

ans =

-43.2708 + 0.0000i
-1.4726 + 0.0000i
-1.2293 + 0.6636i
-1.2293 - 0.6636i
    
```

Figura 4.47: Poli del sistema a ciclo chiuso

Inseriamo il controllore appena realizzato nel progetto *asbQuadcopterStart* precisamente nel blocco *Attitude*, come fatto per la sintesi in frequenza, nel blocco *zeros-poles*. Riportiamo i valori degli zeri, dei poli e del coefficiente di guadagno stabiliti precedentemente e facciamo partire la simulazione. (Figura 4.48)

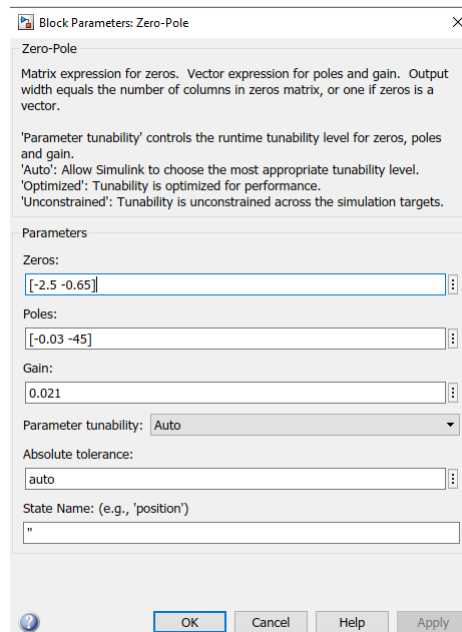


Figura 4.48: Blocco Parameters : Zero-Pole

Nel blocco scope possiamo notare la Figura 4.49 riguardo le prestazioni del controllore realizzato rispetto all'angolo di roll.

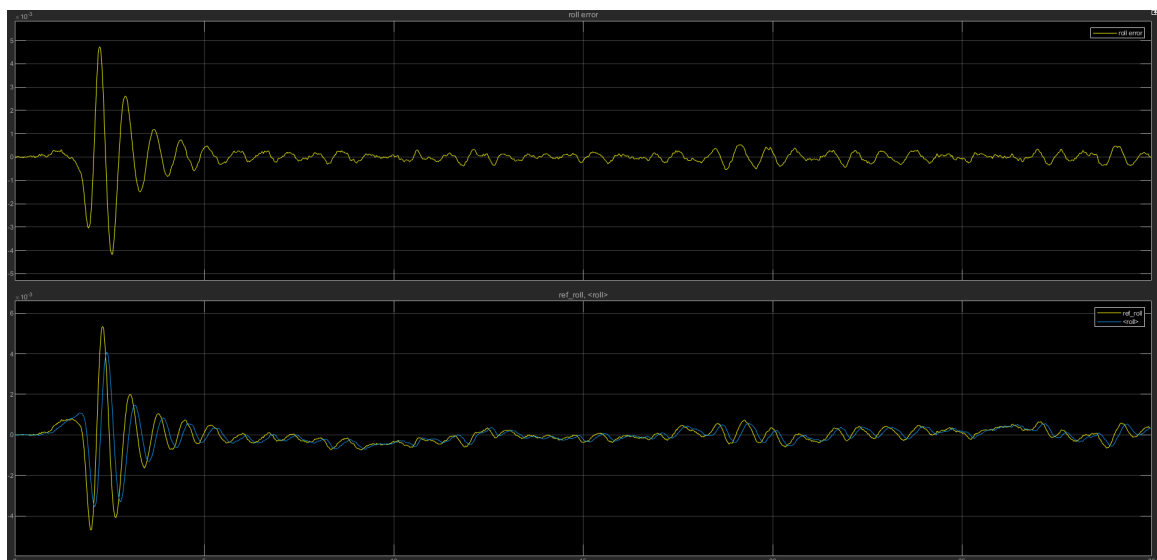


Figura 4.49: Luogo delle radici- Pitch

Con il controllore realizzato tramite la sintesi con luogo delle radici possiamo notare che : l'errore di picco è stato diminuito rispetto ai PID originali del progetto.

Ci possiamo ritenere soddisfatti.

4.6 Realizzazione di controllori per la dinamica del Pitch e per la dinamica del Roll senza implementazione del PID

Si procede ora con la sintesi dei controllori riguardo il grado di libertà Pitch e il grado di libertà Roll, facendoli lavorare contemporaneamente in maniera indipendente su un unico modello *Simulink*.

Per analizzare al meglio, anche qui si è deciso di realizzare 2 controllori:

- Sintesi in frequenza per riferimenti costanti
- Sintesi con luogo delle radici per riferimenti costanti

4.6.1 Sintesi in frequenza

In questa sezione andremo ad implementare la tecnica di sintesi in frequenza descritta prima per la progettazione di controllori per la dinamica del pitch e per la dinamica del roll. Per la realizzazione di questo controllore, utilizzeremo le specifiche ed i controllori realizzati nelle sezioni precedenti 4.4 4.5.

Riprendiamo il progetto *asbQuadcopterStart* nello specifico entriamo nel blocco *Flight Control System/Controller/ Attitude* rispetto alla Figura 3.22, in questo caso viene modificata sia la parte del controllo del Pitch sia la parte del controllo del Roll, così tale da non includere la parte gestita dal PID di default.

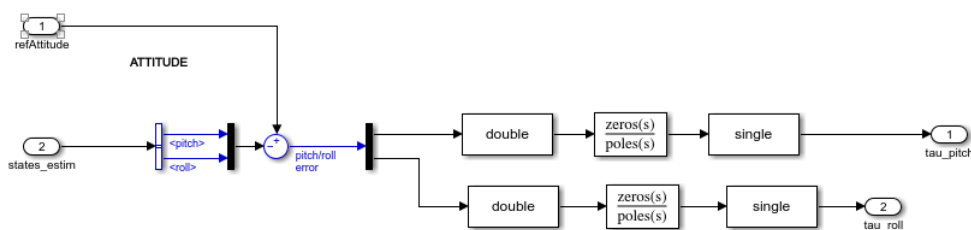


Figura 4.50: Blocco Attitude - Sintesi in Frequenza

Nella Figura 4.51 vengono inseriti poli, zeri e coefficiente di guadagno del controllore implementato precedentemente per la dinamica del pitch e del roll.

Possiamo concludere dicendo che il drone segue in maniera efficace il riferimento con entrambi i controllori Pitch e Roll, anche senza far riferimento al PID di default.

Ci riteniamo quindi soddisfatti.

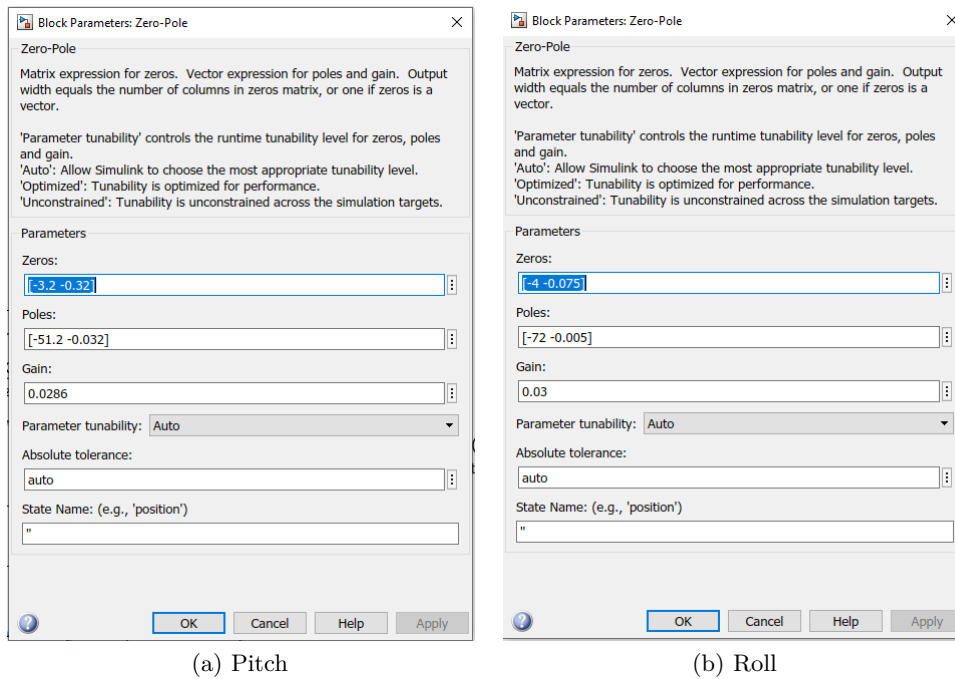


Figura 4.51: Blocco Parameters : Zero-Pole Sintesi in Frequenza

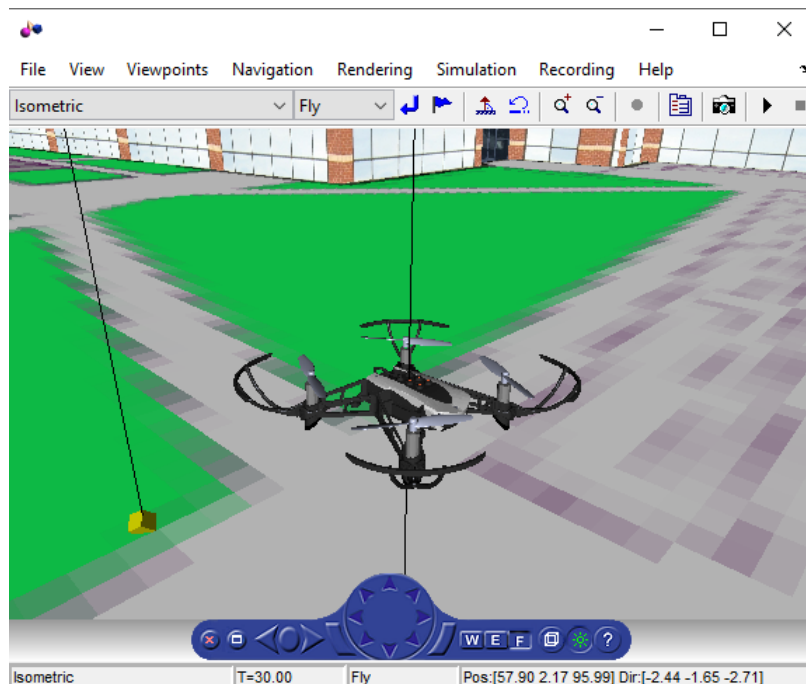


Figura 4.52: Quadricottero

4.6.2 Luogo delle radici

In questa sezione andremo ad implementare la tecnica di sintesi con il luogo delle radici descritta prima per la progettazione di controllori per la dinamica del pitch

4.6 Realizzazione di controllori per la dinamica del Pitch e per la dinamica del Roll senza implementazione

e del roll. Anche in questo caso, come per la sintesi in frequenza, utilizzeremo le specifiche ed i controllori realizzati nelle sezioni precedenti 4.4 4.5.

Inseriamo il controllore appena realizzato nel progetto *asbQuadcopterStart* precisamente nel blocco *Attitude*, come fatto per la sintesi in frequenza, nel blocco *zeros-poles*. Riportiamo i valori degli zeri, dei poli e del coefficiente di guadagno stabiliti precedentemente e facciamo partire la simulazione. (Figura 4.54)

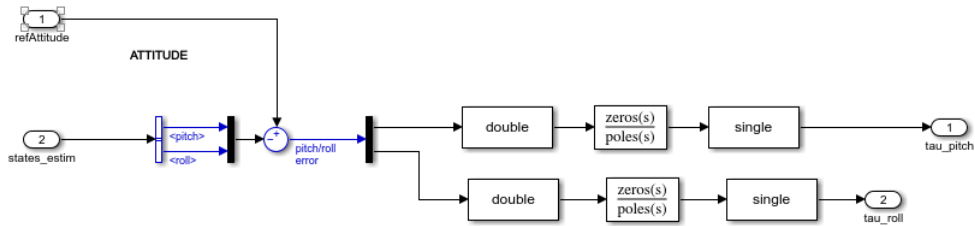


Figura 4.53: Blocco Attitude - Luogo delle Radici

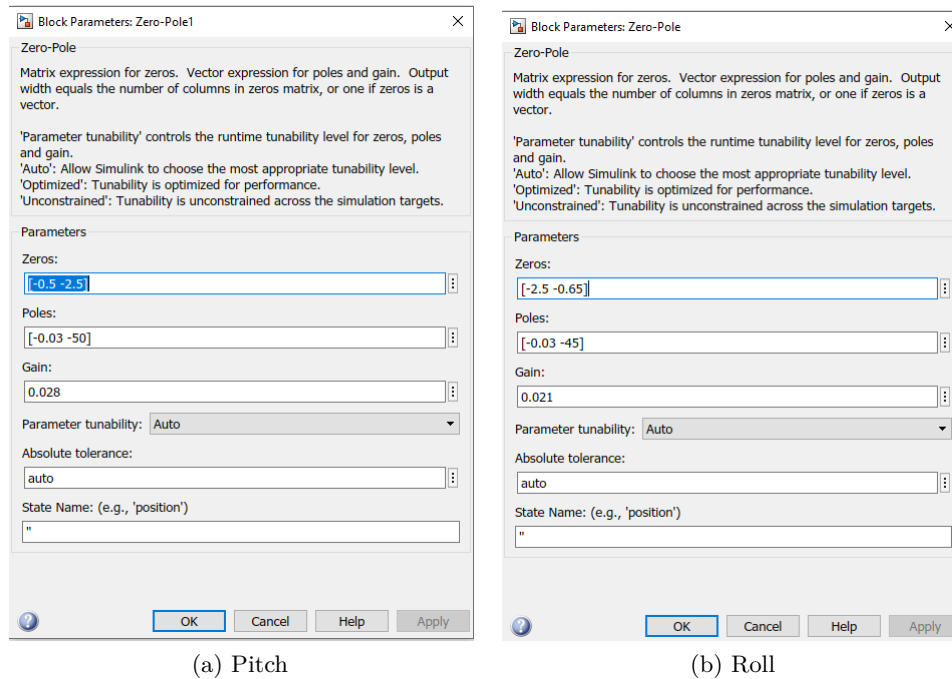


Figura 4.54: Blocco Parameters : Zero-Pole Luogo delle radici

A questo punto vedremo che la simulazione andrà a buon fine e che tutte le specifiche ed i requisiti siano stati correttamente soddisfatti.

Possiamo concludere dicendo che il drone riesce a seguire in maniera efficace il riferimento con entrambi i controllori Pitch e Roll, anche senza far riferimento al PID di default.

Ci riteniamo quindi soddisfatti del seguente elaborato.

Capitolo 5

Conclusioni

Nel presente elaborato è stato affrontato il controllo di sistemi come il quadricottero, dove inizialmente si è analizzato il suo funzionamento, in particolare i movimenti e le caratteristiche di volo. Si è partiti innanzitutto dalla determinazione di un modello matematico non lineare molto accurato per la rappresentazione della sua dinamica, del quale è poi stata effettuata una linearizzazione che ha posto le basi per la progettazione di sei controllori, ognuno concepito per garantire al sistema una buona fedeltà di risposta rispetto a determinati tipi di ingresso.

La realizzazione di questi controllori è avvenuta attraverso l'impiego di tecniche di controllo lineare quali sintesi in frequenza e sintesi tramite luogo delle radici, utilizzate per la realizzazione dei sei controllori, al fine di confrontare i risultati ottenuti. Nel complesso possiamo affermare che i sistemi di controllo realizzati con entrambe le tecniche si sono mostrati in grado di soddisfare le specifiche richieste per tutti i controllori realizzati, inoltre i risultati delle simulazioni sono soddisfacenti ottenendo un comportamento assimilabile al controllore del PID originale, anche nel caso dei 2 controllori dove la dinamica del pitch e la dinamica del roll vengono usati in maniera indipendente senza includere il PID di default.

Bibliografia

- [1] Carlo Cosentino. Linearizzazione e stabilità dei sistemi non lineari. 2016.
- [2] Livio Mariano e John Straetmans. Drone modeling control and validation. 2021.
- [3] Tommaso Gordini e Lorenzo Pantieri. L'arte di scrivere con latex. 2021.
- [4] Davide Meneghel. Regolatori pid: tecniche di sintesi e problematiche implementative. 2010.
- [5] Malcolm D Shuster et al. A survey of attitude representations. *Navigation*, 8(9):439–517, 1993.
- [6] Oleg A Yakimenko. *Engineering computations and modeling in MATLAB®/Simulink®*. American Institute of Aeronautics and Astronautics, 2011.

[1, 3, 2, 4, 6, 5]