



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Informatica e dell'Automazione

**Studio e sviluppo di tecniche di
controllo per quadricotteri**

Study and development of control
techniques for quadcopters

Relatore: Prof.
Ippoliti Gianluca

Tesi di Laurea di:
Wafa Mohammad

Correlatore: Prof.
Orlando Giuseppe

A.A. 2019/2020

Sommario

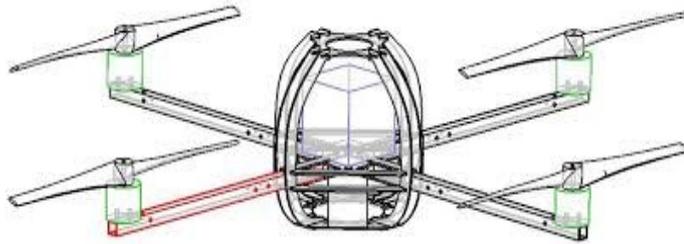
In questo elaborato viene presentato lo studio e lo sviluppo di una tecnica di controllo per un quadricottero, sviluppato per il controllo dell'angolo di rollio. L'elaborato è stato suddiviso in cinque capitoli, i quali comprendono: la struttura fisica e matematica del veicolo, il modello simulato su *Matlab*, l'analisi di tale modello attraverso i controllori originari, la progettazione del controllore ed infine le considerazioni finali sul risultato ottenuto. La tecnica di controllo utilizzata è stata quella della sintesi per tentativi, fatti i calcoli a mano per poi essere riportati su *Matlab*, visualizzando ed analizzando i relativi grafici di Bode, Nyquist e della risposta in frequenza, attraverso il *Sisotool*. L'utilizzo del software *Matlab* ha permesso la semplificazione e la velocità di molti passaggi per lo sviluppo del controllore, in base al tempo ed alla complessità: la linearizzazione, lo schema a blocchi per i vari controllori e la stessa simulazione. Scaricato il pacchetto *Simulink*, è stato possibile lavorare sul drone in forma schematica, a blocchi, e visualizzare il risultato finale attraverso i vari grafici dei comportamenti relativi a tutti e sei i gradi di libertà (X, Y, Z, Pitch, Roll, Yaw) ed attraverso la simulazione di volo permessa dal *Simulink 3-D*.

Indice

Introduzione	- 8 -
Capitolo 1.....	- 10 -
Modello Quadricottero.....	- 10 -
1.1 modello fisico	- 10 -
1.2 modello matematico	- 12 -
Capitolo 2.....	- 18 -
Simulink.....	- 18 -
2.1 modello simulato	- 18 -
2.2 modello del processo	- 20 -
2.3 linearizzazione	- 20 -
2.4 blocco Airframe	- 23 -
2.5 blocco di controllo	- 28 -
2.5.1 PID	- 30 -
2.5.2 blocco XY to reference orientation	- 33 -
2.5.3 blocco Attitude	- 34 -
2.5.4 blocco Yaw	- 35 -
2.5.5 blocco Gravity	- 35 -
2.6 blocco Signal Builder	- 36 -
Capitolo 3.....	- 38 -
Analisi modello PID.....	- 38 -
3.1 analisi del modello lineare (PID)	- 38 -
3.2 analisi del modello non lineare (PID)	- 43 -
Capitolo 4.....	- 46 -
Sviluppo controllore per l'angolo di rollio.....	- 46 -
4.1 sintesi in frequenza per tentativi	- 46 -
4.2 creazione della funzione di trasferimento del processo P(s)	- 48 -
4.3 controllo di primo tentativo	- 51 -
4.3.1 analisi comportamento modello lineare (controllore I tentativo)	- 60 -
4.3.2 analisi comportamento modello non lineare (controllore I tentativo)	- 63 -
4.4 controllore di secondo tentativo	- 66 -
4.4.1 analisi comportamento modello lineare (controllore II tentativo)	- 69 -
4.4.2 analisi comportamento modello non lineare (controllore II tentativo)	- 72 -
Capitolo 5.....	- 75 -
Conclusione.....	- 75 -
Bibliografia	- 78 -



< Quando camminerete sulla terra dopo
aver volato, guarderete il cielo perché
là siete stati e là vorrete tornare. >



Introduzione

La Storia ...

Il sogno dell'uomo è quello di volare e da sempre ha cercato di progettare e costruire modelli per poterlo fare. Tra la fine del Quattrocento e l'inizio del Cinquecento possiamo ritrovare i primi disegni di un ornitottero (dal greco "uccello") con il suo massimo esponente Leonardo da Vinci il quale ragionando sul volo, osservando l'ambiente e la natura circostante da Vinci ebbe un'intuizione capace di discostarlo dalle convinzioni degli scienziati dell'epoca, che vedevano il volo come un processo misterioso e quasi magico. Nel volo non è presente nessun tipo di magia ma è caratterizzato da una semplice e mera meccanica, tale meccanica scoperta ed evoluta tra la fine dell'Ottocento e l'inizio del Novecento.

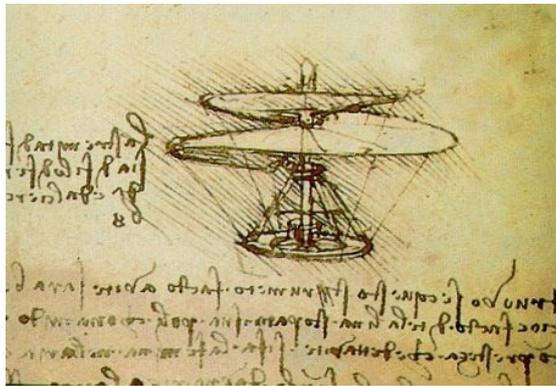


Figura 1: Le macchine di Leonardo: il volo

L'aeroplano ha una data di nascita precisa: il 17 dicembre del 1903, quando i due fratelli statunitensi Orville e Wilbur Wright crearono il primo mezzo in grado di sollevarsi da terra e di volare in modo controllato grazie a un motore e lo fecero alzare in volo vicino alla cittadina di Kitty Hawk.

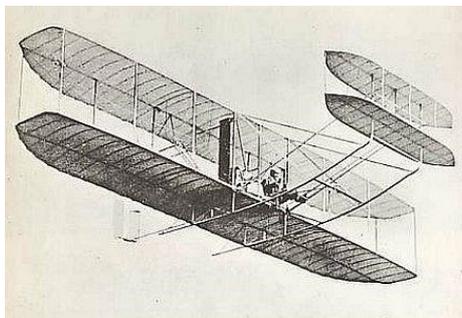


Figura 2: Aeroplano Orville e Wilbur Wright

Dal Novecento fino ad oggi, il mondo dell'ingegneria ha avuto una crescita di sviluppo al livello esponenziale, specialmente nel ramo dell'ingegneria aerospaziale mettendo a disposizione i vari rami dell'ingegneria interessati (nel dettaglio gli sviluppi al livello dell'informatica e dell'automatica) per lo sviluppo e la progettazione di nuovi veicoli.

Sempre sulla base di una ricerca empirica condotta dall'uomo osservando la natura circostante, osservando la specie animale, in dettaglio la specie delle api, l'uomo questa volta provò, con grande successo, a progettare un veicolo aereo senza pilota, progettò un drone (la parola drone in inglese significa "fucò", ovvero il maschio dell'ape).

Secondo il vocabolario Treccani un drone è un velivolo privo di pilota e comandato a distanza, usato generalmente per operazioni di ricognizione e sorveglianza, oltre che di disturbo e inganno nella guerra elettronica. Un aeromobile a pilotaggio remoto, comunemente noto come drone, è un velivolo caratterizzato dall'assenza del pilota umano a bordo. Il suo volo è controllato dal computer a bordo del velivolo, sotto il controllo remoto di un navigatore o pilota, sul terreno o in un altro veicolo.

Inizialmente i droni venivano costruiti per scopi militari, essi venivano e tutt'ora vengono utilizzati come vere e proprie armi di attacco o strumenti per l'esercitazione militare, sotto forma di bombardieri pilotati da remoto. Questi droni con il tempo vengono utilizzati in un'ampia gamma di realtà lavorative: monitoraggio, sicurezza, mappatura, ispezioni, fotografia, cinema, consegna merci e molte altre ancora. Oggi giorno in tutto il mondo vengono utilizzati i cosiddetti APR (aeromobile a pilotaggio remoto) o semplicemente chiamati droni: in Svizzera i droni sono utilizzati principalmente per costruzioni di edifici ("The Flight Assembled Architecture installation"), negli USA vengono usati per il trasporto, in Olanda i droni sono utilizzati per il soccorso medico ed in Francia vengono utilizzati addirittura per l'agricoltura.



Figura 3: Drone agricolo



Figura 4: Drone per trasporto



Figura 5: Drone per costruzione edifici

Capitolo 1

Modello Quadricottero

In questa sezione andremo ad analizzare nel dettaglio la struttura fisica di un *quadricottero*, analizzando le varie leggi fisiche e matematiche che lo regolano e la propria implementazione su Matlab.

1.1 modello fisico

La forza generata per il movimento del veicolo proviene dai motori, i quali hanno un'importanza fondamentale. Un drone può avere diversi motori generando diversi nomi del veicolo: se sono 3 motori si parla di un *tricottero*, 4 motori *quadricottero*, 6 motori *esacottero* oppure un drone con 8 motori è un *ottocottero*. I **motori elettrici** montati sui droni sono solitamente motori **brushless** ovvero “senza spazzole”: questo genere di motori funziona senza bisogno di contatti elettrici striscianti (ossia le “spazzole”) sull'albero motore; in questo modo avranno meno resistenza meccanica, nessun rischio di scintille e minor peso. Ma è grazie all' **Electronic Speed Control (ESC)** che i motori brushless si collegano al **Flight Controller**: l'ESC riceve il comando dal flight controller e lo trasferisce ai motori. Questa piccola scheda elettronica consente quindi alle eliche, di girare in entrambi i sensi e a velocità variabili.

Il drone analizzato in questo elaborato è un *quadricottero* con rispettivamente i propri quattro motori presenti sull'asse i quali possono avere una configurazione a più oppure una configurazione a croce.

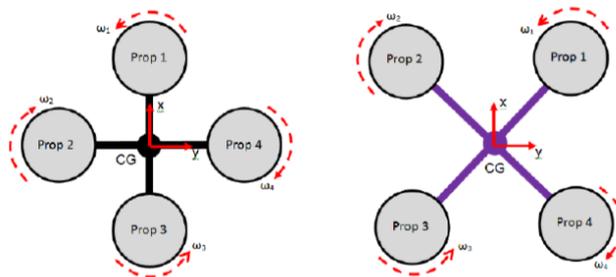


Figura 6: Configurazione Drone

Il drone presenta sei gradi di libertà: riesce a muoversi secondo i tre suddetti angoli di roll, indicato con ϕ , pitch, indicato con θ e yaw, indicato con Ψ , e modificare così il suo assetto, ma riesce anche a muoversi nelle tre direzioni tradizionali dello spazio X, Y e Z.

Il veicolo si presenta con un sistema fisico con sei gradi di libertà (3 gradi di libertà traslazionale e 3 gradi di libertà rotazionale) ma essendo dotato di soli quattro motori, un *quadricottero* è tipico sistema sotto attuato, ciò comporta che solo alcuni movimenti potranno essere controllati direttamente tramite l'azione dei motori:

È un sistema sotto attuato poiché ha sei gradi di libertà: beccheggio, imbardata, rollio, x (movimento nella direzione frontale del veicolo), y (movimento verso il lato sinistro del veicolo) e z (altitudine) ma è controllato utilizzando solo quattro attuatori.

1. **Spinta verticale (thrust):** dovuta all'azione sincronizzata dei quattro motori che, attraverso la rotazione delle eliche, convogliano verso il basso una massa di aria tale da generare una differenza di pressione sufficiente a sollevare il *quadricottero*.
2. **Beccheggio (pitch):** dovuto al momento generato dalle spinte non uniformi dovute alle diverse velocità di rotazione dei motori anteriori e posteriori.
3. **Rollio (roll):** dovuto al momento generato dalle spinte non uniformi dovute alle diverse velocità di rotazione dei motori posizionati sul lato destro rispetto a quelli sul lato sinistro.
4. **Imbardata (yaw):** si ottiene in condizione di equilibrio per quanto riguarda le spinte dovute all'azione dei motori disposti simmetricamente che annullano reciprocamente i momenti meccanici generati sul veicolo il quale, infatti, non si inclina.

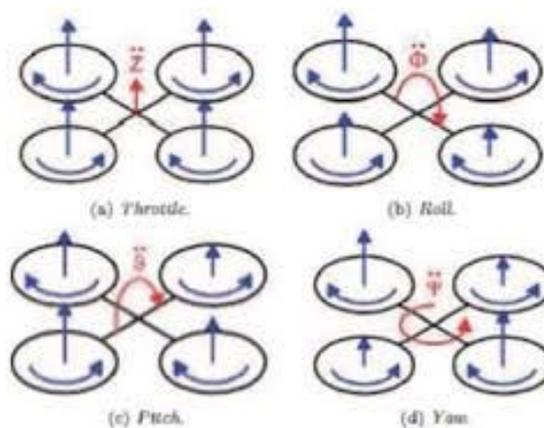


Figura 7: Configurazione Drone a croce

I motori anteriore e posteriore fanno girare le pale in senso orario, mentre quelli di destra e sinistra, in senso antiorario.

1.2 modello matematico

L'analisi della modellazione matematica del sistema focalizza l'attenzione, in particolare, sulle equazioni della cinematica e della dinamica d'assetto per un corpo rigido, e sulle equazioni di controllo.

Le equazioni della cinematica descrivono l'evoluzione nel tempo di un insieme di parametri che definiscono l'assetto del satellite rispetto a un fissato riferimento inerziale. L'orientazione del sistema viene ricavata attraverso la conoscenza delle velocità angolari, indipendentemente dalle coppie esterne agenti su di esso, note le condizioni iniziali. Esistono diverse possibilità per la parametrizzazione dell'assetto: si farà riferimento nel seguito solamente alle descrizioni mediante **angoli di Eulero** e quaternioni, mettendo in luce gli aspetti positivi e negativi del loro utilizzo.

Gli angoli di Eulero sono stati introdotti per descrivere l'orientamento di un corpo rigido nello spazio. Tali angoli descrivono la posizione di un sistema di riferimento XYZ solidale con un corpo rigido attraverso una serie di rotazioni a partire da un sistema di riferimento fisso xyz. I due sistemi di riferimento coincidono con l'origine. Se i piani xy e XY sono distinti, si intersecano in una retta (passante per l'origine) detta linea dei nodi (N). Se i piani coincidono, si definisce la linea dei nodi N come l'asse X. Gli angoli di Eulero sono i tre angoli seguenti:

- α è l'angolo tra l'asse x e la linea dei nodi. Detto angolo di precessione, è definito in $[0, 2\pi)$ oppure in $[-\pi, \pi)$;
- β è l'angolo tra gli assi z e Z. Detto angolo di nutazione, è definito in $[0, \pi)$ oppure in $[-\frac{\pi}{2}, \frac{\pi}{2}]$;
- γ è l'angolo tra la linea dei nodi e l'asse X. Detto angolo di rotazione propria, è definito in $[0, 2\pi)$ oppure in $[-\pi, \pi)$.

In figura si nota il sistema fisso (xyz) evidenziato con il colore blu, e il sistema ruotato (XYZ) rappresentato dal colore rosso. La linea dei nodi, indicata con N, è rappresentata in verde.

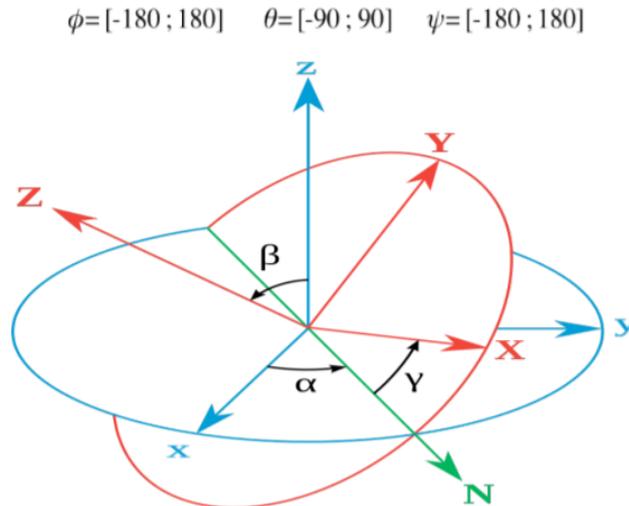


Figura 8: Angoli di Eulero

Su queste rotazioni è possibile paragonare le singole manovre che il drone può effettuare, quindi ad un'inclinazione attorno ad un asse, è possibile associare uno dei angoli di Eulero:

- > alla rotazione attorno all'asse longitudinale del corpo daremo il valore ϕ (roll);
- > alla rotazione attorno all'asse trasversale daremo il valore θ (pitch);
- > alla rotazione attorno all'asse verticale passante per il baricentro il valore Ψ (yaw.)

Intervengono tre matrici di rotazione, ciascuna corrispondente ad un angolo, e quindi ad una manovra di roll, pitch o yaw.

Le matrici sono così definite:

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

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

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

Una matrice totale di rotazione in R^3 , nota anche come DCM (Direction Cosine Matrix), si ottiene moltiplicando le matrici prima illustrate.

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

In base a tale formulazione è possibile ricavare una velocità angolare come un vettore composto dalla somma di tre vettori rispettivamente di modulo, orientati secondo gli assi z, x e y.

La velocità angolare della terna solidale al satellite espressa nelle coordinate di una terna inerziale di riferimento è determinata mediante la seguente:

$$\vec{\omega} = R_x \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_x R_y \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_x R_y R_z \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}$$

eseguendo i prodotti, può essere espressa in forma matriciale come

$$\vec{w} = \begin{bmatrix} 1 & 0 & -\text{sen}(\theta) \\ 0 & \cos(\phi) & \text{sen}(\phi)\cos(\theta) \\ 0 & -\text{sen}(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

Tali velocità angolari saranno \mathbf{p} , \mathbf{q} e \mathbf{r} rispettivamente relativi all'angolo di roll, pitch e yaw. I tre angoli sono strettamente legati, cioè una piccola variazione degli angoli ϕ e θ causano, variano l'angolo ψ , dovuto al fatto che possono causare una rotazione lungo l'asse verticale del veicolo.

Per ottenere le tre velocità traslazionali, denominando \mathbf{b} il vettore delle coordinate rispetto al sistema di riferimento solidale al veicolo con origine il centro di gravità del veicolo e \mathbf{r} il vettore delle coordinate rispetto al sistema inerziale, si ottiene:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \text{sen}\phi \\ 0 & -\text{sen}\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\text{sen}\theta \\ 0 & 1 & 0 \\ \text{sen}\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \text{sen}\psi & 0 \\ -\text{sen}\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Esplicitando i prodotti matriciali, si ottiene:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi & \cos\theta\text{sen}\phi & -\text{sen}\theta \\ -\cos\phi\text{sen}\psi + \text{sen}\phi\text{sen}\theta\cos\psi & \cos\phi\cos\psi + \text{sen}\phi\text{sen}\theta\text{sen}\psi & \text{sen}\phi\cos\theta \\ \text{sen}\phi\text{sen}\psi + \cos\phi\text{sen}\theta\cos\psi & -\text{sen}\phi\cos\psi + \cos\phi\text{sen}\theta\text{sen}\psi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Sfruttando l'ortonormalità delle matrici, è possibile ricavare la matrice di trasferimento dal sistema di riferimento solidale al veicolo a quello inerziale, da cui poi si possono ricavare le altre grandezze fisiche di interesse, come appunto le velocità traslazionali.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi & -\cos\phi\text{sen}\psi + \text{sen}\phi\text{sen}\theta\cos\psi & \text{sen}\phi\text{sen}\psi + \cos\phi\text{sen}\theta\cos\psi \\ \cos\theta\text{sen}\phi & \cos\phi\cos\psi + \text{sen}\phi\text{sen}\theta\text{sen}\psi & -\text{sen}\phi\cos\psi + \cos\phi\text{sen}\theta\text{sen}\psi \\ -\text{sen}\theta & \text{sen}\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Quindi ricapitolando, le trasformazioni rispetto al sistema di riferimento inerziale sono le seguenti:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \frac{\text{sen}\phi \text{sen}\theta}{\text{cos}\theta} & \frac{\text{cos}\phi \text{sen}\theta}{\text{cos}\theta} \\ 0 & \text{cos}\phi & -\text{sen}\phi \\ 0 & \frac{\text{sen}\phi}{\text{cos}\theta} & \frac{\text{cos}\phi}{\text{cos}\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \text{cos}\theta \text{cos}\phi & -\text{cos}\phi \text{sen}\psi + \text{sen}\phi \text{sen}\theta \text{cos}\psi & \text{sen}\phi \text{sen}\psi + \text{cos}\phi \text{sen}\theta \text{cos}\psi \\ \text{cos}\theta \text{sen}\phi & \text{cos}\phi \text{cos}\psi + \text{sen}\phi \text{sen}\theta \text{sen}\psi & -\text{sen}\phi \text{cos}\psi + \text{cos}\phi \text{sen}\theta \text{sen}\psi \\ -\text{sen}\theta & \text{sen}\phi \text{cos}\theta & \text{cos}\phi \text{cos}\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Mentre le trasformazioni inverse, i quali forniscono le variazioni delle velocità angolari e traslazionali rispetto al sistema di riferimento solidale al veicolo risultano essere:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\text{sen}\theta \\ 0 & \text{cos}\phi & \text{sen}\phi \text{cos}\theta \\ 0 & -\text{sen}\phi & \text{cos}\phi \text{cos}\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \text{cos}\theta \text{cos}\phi & \text{cos}\theta \text{sen}\phi & -\text{sen}\theta \\ -\text{cos}\phi \text{sen}\psi + \text{sen}\phi \text{sen}\theta \text{cos}\psi & \text{cos}\phi \text{cos}\psi + \text{sen}\phi \text{sen}\theta \text{sen}\psi & \text{sen}\phi \text{cos}\theta \\ \text{sen}\phi \text{sen}\psi + \text{cos}\phi \text{sen}\theta \text{cos}\psi & -\text{sen}\phi \text{cos}\psi + \text{cos}\phi \text{sen}\theta \text{sen}\psi & \text{cos}\phi \text{cos}\theta \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$

Dopo aver introdotto i vettori relativi alla velocità di traslazione (V) e di rotazione (w) espressi nella terna di riferimento del sistema corpo del drone.

$$V = [u \quad v \quad w]^T$$

$$\omega = [p \quad q \quad r]^T$$

Nel sistema di riferimento degli assi terrestri invece si introducono il vettore lineare di posizione (P) e il vettore che rappresenta l'orientamento nello spazio del velivolo tramite gli angoli di Eulero (A):

$$P = [x \ y \ z]^T$$

$$A = [\phi \ \theta \ \psi]^T$$

Si definisce quindi:

$$\dot{P} = DCM \cdot V$$

$$\dot{A} = ROT \cdot \omega$$

Dove ROT risulta essere:

$$ROT = \begin{bmatrix} 1 & \sin\phi \operatorname{tg}\theta & \cos\phi \operatorname{tg}\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\operatorname{ctg}\theta} & \frac{\cos\phi}{\operatorname{ctg}\theta} \end{bmatrix}$$

Esplicitando i prodotti matriciali si ha :

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

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

Come detto in precedenza, si nota facilmente che le manovre in rollio e/o beccheggio possono automaticamente causare variazioni angolari di ψ ottenendo anche una manovra di imbardata. Queste sei espressioni sono espresse nella terna fissa terrestre, i quali sottospecifiche ipotesi risulta applicabile la seconda legge di Newton.

Capitolo 2

Simulink

In questa sezione si andrà ad analizzare nel dettaglio il software utilizzato per la progettazione del controllore, il modello vero e proprio simulato al computer.

2.1 modello simulato

Per la realizzazione dell'intero progetto è stato utilizzato *Matlab*, un software nel quale presente un ambiente di programmazione per applicazioni scientifiche, di analisi numerica e per la simulazione di sistemi dinamici. La potenza principale di questo software è la vastità di pacchetti e librerie; nell'ambito della modellazione di sistemi dinamici sono presenti i set di funzioni di base *general purpose*, le estensioni *application oriented* (toolboxes) e le varie interfacce grafiche interattive per la modellazione e le simulazioni di sistemi dinamici.

Simulink è un software per la modellazione, simulazione e analisi di sistemi dinamici, sviluppato dalla compagnia statunitense MathWorks.

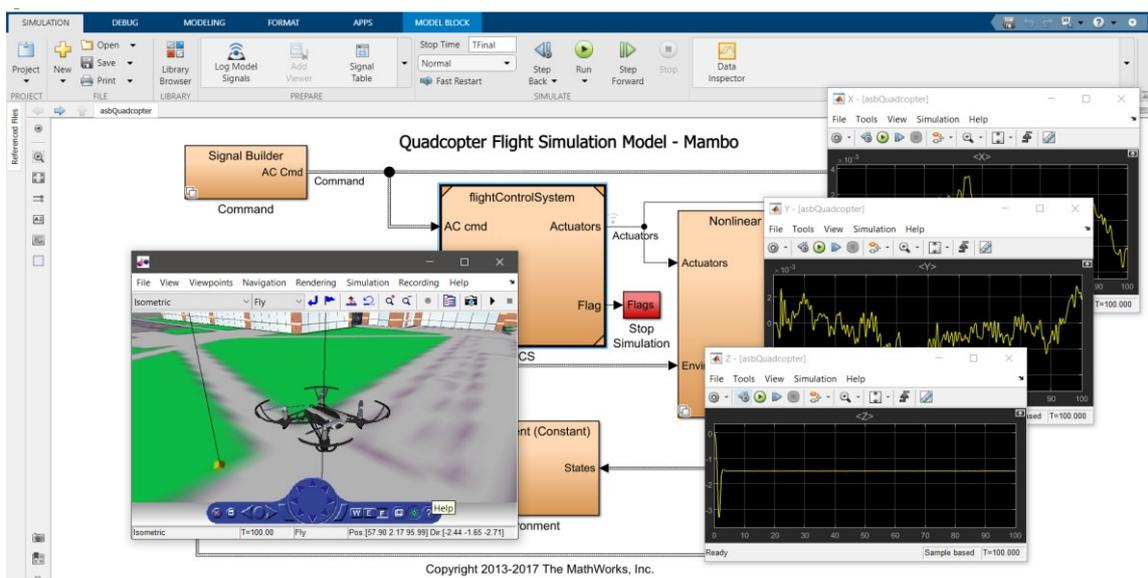
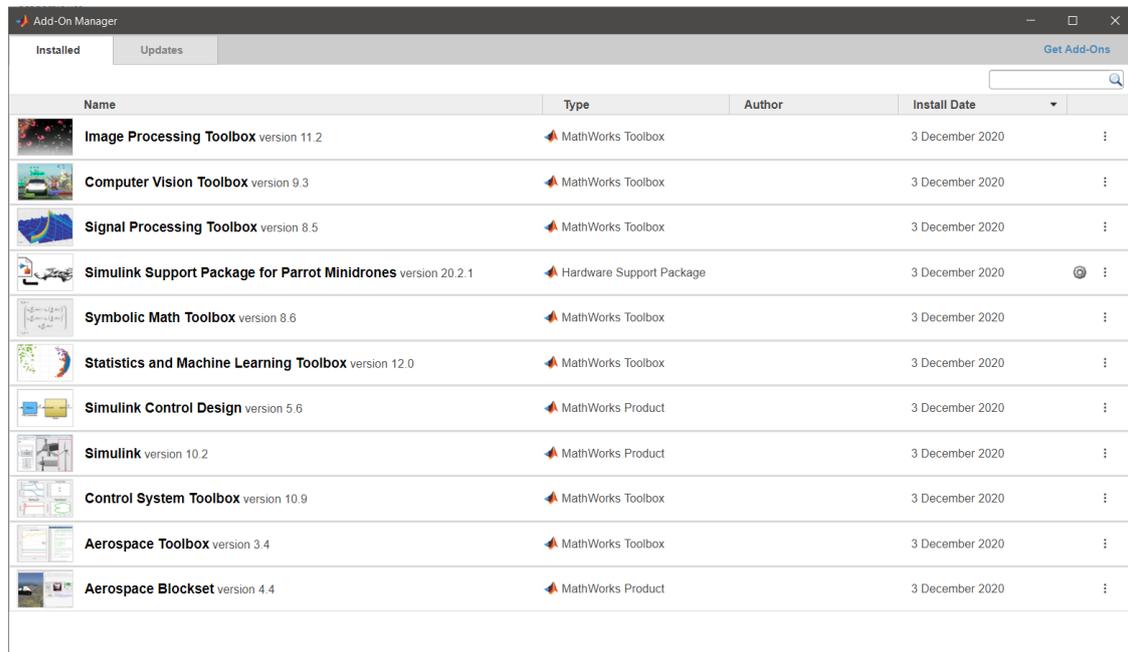


Figura 9: modello simulato in Simulink

Prima d'iniziare, bisogna scaricare i vari pacchetti dallo storage di Matlab per poter utilizzare le varie funzionalità che andremo ad analizzare nella prossima sezione.



Name	Type	Author	Install Date
Image Processing Toolbox version 11.2	MathWorks Toolbox		3 December 2020
Computer Vision Toolbox version 9.3	MathWorks Toolbox		3 December 2020
Signal Processing Toolbox version 8.5	MathWorks Toolbox		3 December 2020
Simulink Support Package for Parrot Minidrones version 20.2.1	Hardware Support Package		3 December 2020
Symbolic Math Toolbox version 8.6	MathWorks Toolbox		3 December 2020
Statistics and Machine Learning Toolbox version 12.0	MathWorks Toolbox		3 December 2020
Simulink Control Design version 5.6	MathWorks Product		3 December 2020
Simulink version 10.2	MathWorks Product		3 December 2020
Control System Toolbox version 10.9	MathWorks Toolbox		3 December 2020
Aerospace Toolbox version 3.4	MathWorks Toolbox		3 December 2020
Aerospace Blockset version 4.4	MathWorks Product		3 December 2020

Figura 10: Pacchetti installati

A questo punto basterà andare ad inserire il seguente codice nel *Command Window* di matlab: `asbQuadcopterStart`.

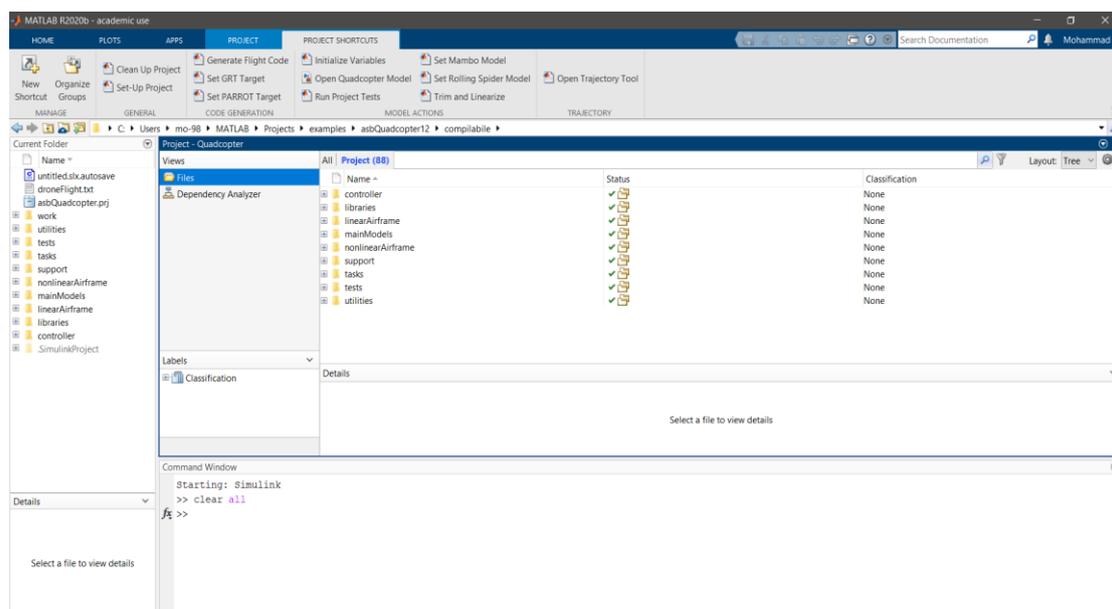


Figura 11: Librerie caricati per l'avvio del progetto

Il software andrà a caricare le varie librerie e cartelle del modello di un *quadricottero*, in particolare quelle relative al *Parrot MAMBO Drone*.

2.2 modello del processo

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

Classico schema di controllo in contrazione dove il processo viene dato dal blocco *Airframe* mentre la parte di controllo proviene dal blocco *FCS*. Gli altri blocchi servono per la generazione di riferimenti, lettura dei valori dei sensori, simulazione dell'ambiente ed infine per la simulazione vera e propria.

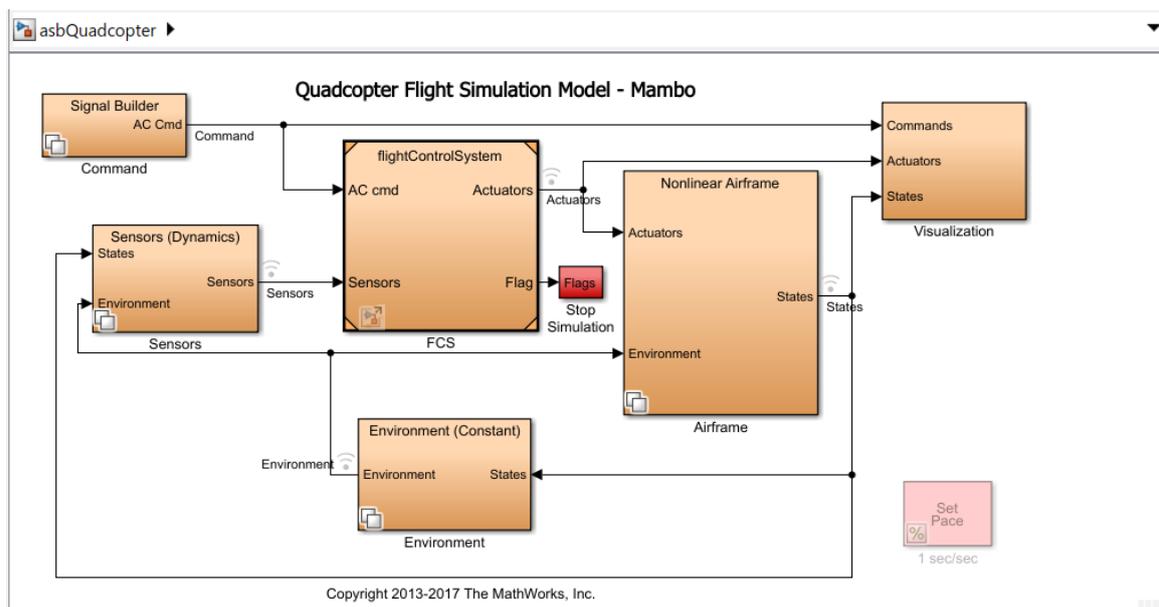


Figura 12: Quadcopter Flight Simulation Model - Mambo

2.3 linearizzazione

Prima di analizzare il primo blocco interessato, il blocco *Airframe*, è stata elaborata una piccola sintesi riguardo il passaggio da un modello non lineare a quello lineare.

Partendo dalle classiche equazioni della dinamica, è possibile approssimare il comportamento del sistema non lineare con quello di un modello lineare.

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) & u &\in R^m \\ y(t) &= g(x(t), u(t)) & x &\in R^n \\ & & y &\in R^p \end{aligned}$$

Si consideri un sistema soggetto ad un ingresso costante $u(t) = \bar{u}$

In corrispondenza di tale ingresso avremo un punto di equilibrio se esiste un valore dello stato \bar{x} tale che $f(\bar{x}, \bar{u}) = 0$

Corrispondentemente avremo $\bar{y} = g(\bar{x}, \bar{u})$

Ricordando che un sistema LTI (Linear Time Invariant system) ammette un unico punto di equilibrio, che coincide con l'origine dello spazio di stato ($x = 0$) mentre i sistemi non lineari possono ammettere zero, uno o più punti equilibrio (anche infiniti).

Si consideri adesso le variazioni rispetto ai valori di equilibrio:

$$u(t) = \bar{u} + \delta u(t)$$

$$x(t) = \bar{x} + \delta x(t)$$

$$y(t) = \bar{y} + \delta y(t)$$

$$x_0 = \bar{x} + \delta x_0$$

Il sistema diventa:

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

Con la condizione iniziale:

$$\bar{x} + \delta x(t_0) = \bar{x} + \delta x_0$$

Supponendo che le funzioni f e g siano sufficientemente regolari, esse possono essere sviluppate in serie di Taylor rispetto a x e u nell'intorno del punto di equilibrio:

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

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



$$\delta\dot{x}(t) = A \delta x(t) + B \delta u(t) \quad \delta x(t_0) = \delta x_0$$

$$\delta y(t) = C \delta x(t) + D \delta u(t)$$

Quindi

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

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

Esaminando lo sviluppo in serie di Taylor ci si rende facilmente conto che l'approssimazione risulta valida a patto che:

- Le variazioni dal punto di equilibrio siano contenute;
- I termini di ordine superiore al primo dello sviluppo in serie siano abbastanza piccoli.

Affinché l'ultima ipotesi sia soddisfatta, le derivate di ordine superiore al primo delle funzioni f e g devono essere abbastanza piccole. Questa condizione equivale a richiedere che le funzioni f e g abbiano un andamento "abbastanza lineare".

Tutti questi passaggi saranno facilitati attraverso delle funzionalità del software utilizzato per questo progetto, come si vedrà nella sezione seguente.

2.4 blocco Airframe

In questo blocco è possibile visualizzare ed analizzare il modello lineare o non-lineare del processo fornito da Simulink che si compone dalle solite quattro matrici utilizzate nell'analisi dei sistemi dinamici lineari: A , B , C e D .

Settando il valore della variabile di ambiente VSS $VEHICLE$ è possibile passare dal modello lineare a quello non lineare e viceversa.

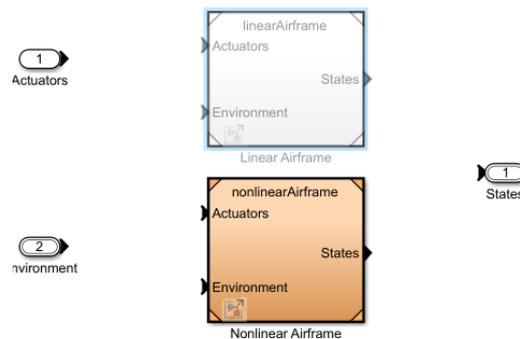


Figura 13: Passaggio dal modello lineare al modello non lineare

Una volta settato il valore VSS $VEHICLE = 0$ e aggiornato il blocco, si ottiene il modello lineare

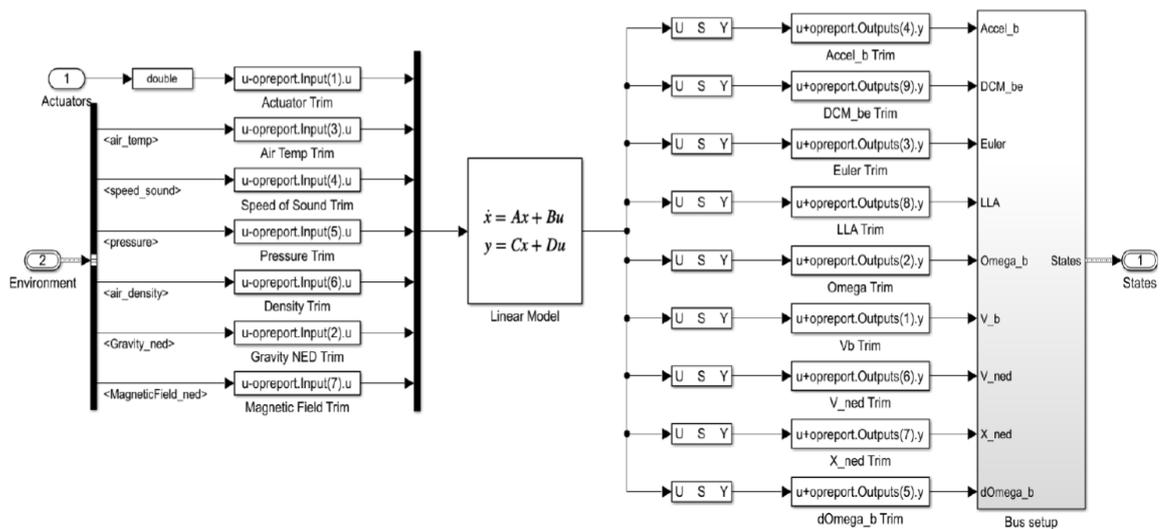


Figura 14: modello lineare

Modello lineare composto dalle solite quattro matrici della dinamica (A, B, C e D). I valori dei coefficienti delle matrici A,B,C,D sono consultabili attraverso del Workspace di Matlab, all'interno del gruppo di variabili denominato *linsys*. Dopo aver caricato i dati attraverso il comando `load ('linearizedAirframe.mat', 'linsys')` .sarà possibile visualizzare le vari matrici con il solito comando:

- `linsys.a`

```
>> linsys.a
ans =
    0    0.0000    0    0    0    0    0    0    0    1.0000    0.0000   -0.0000
  -0.0000    0    0    0    0    0    0    0    0    0    1.0000    0.0000
  -0.0000   -0.0000    0    0    0    0    0    0    0    0    -0.0000    1.0000
    0   -9.8100    0   -0.0000   -0.0000    0.0000    0    0    0   -0.0000    0.1380   -0.0000
    9.8100   -0.0000    0   -0.0000   -0.0000   -0.0000    0    0    0   -0.1380    0.0000   -0.0000
    0.0000    0.0000    0   -0.0000    0.0000    0    0    0    0    0.0000    0.0000    0
    0.0000    0.0000    0.0000    1.0000   -0.0000   -0.0000    0    0    0    0    0    0
  -0.0000    0.0000    0.0000    0.0000    1.0000    0.0000    0    0    0    0    0    0
  -0.0000   -0.0000    0    0.0000   -0.0000    1.0000    0    0    0    0    0    0
    0    0    0   -0.0000   -0.0000   -0.0000    0    0    0   -2.1715    0.0000    0.0000
    0    0    0    0.0000    0.0000   -0.0000    0    0    0    0.0000   -1.6192   -0.0000
    0    0    0   -0.0000   -0.0000    0.0000    0    0    0    0.0000    0.0000   -0.0000
```

Figura 15: matrice A

Matrice A (12 x 12): matrice della dinamica degli stati del sistema, in cui presenta 12 variabili di stato del sistema.

phi
theta
psi
ub
vb
wb
xe
ye
ze
p
q
r

Figura 17: Vettore colonna della matrice A

phi theta psi(1)'
phi theta psi(2)'
phi theta psi(3)'
ub,vb,wb(1)'
ub,vb,wb(2)'
ub,vb,wb(3)'
xe,ye,ze(1)'
xe,ye,ze(2)'
xe,ye,ze(3)'
p,q,r(1)'
p,q,r(2)'
p,q,r(3)'

Figura 16: Vettore riga della matrice A

- linsys.b

```
>> linsys.b
ans =
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   -0.0000    0.0000   -0.0000    0.0000   -0.0000    0    0    0    1.0000    0.0000    0.0000    0    0    0
    0.0000   -0.0000    0.0000   -0.0000    0.0000    0    0    0   -0.0000    1.0000   -0.0000    0    0    0
   -0.0096    0.0096   -0.0096    0.0096   -8.2855    0    0    0    0    0    1.0000    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0.4202    0.4202   -0.4202   -0.4202    0.0000    0    0    0    0    0    0    0    0    0
    0.3133   -0.3133   -0.3133    0.3133    0.0000    0    0    0    0    0    0    0    0    0
   -0.0115   -0.0115   -0.0115   -0.0115   -0.0000    0    0    0    0    0    0    0    0    0
```

Figura 18: matrice B

Matrice B (12 x 14): Matrice degli ingressi del sistema, descrive l'azione degli ingressi sulla dinamica dello stato del sistema.

'Actuators(1)'
'Actuators(2)'
'Actuators(3)'
'Actuators(4)'
'AtmosphereBus.air_density'
'AtmosphereBus.air_temp'
'AtmosphereBus.pressure'
'AtmosphereBus.speed_sound'
'Gravity_ned(1)'
'Gravity_ned(2)'
'Gravity_ned(3)'
'MagneticField_ned(1)'
'MagneticField_ned(2)'
'MagneticField_ned(3)'

Figura 20 vettore colonna della matrice B

phi theta psi(1)'
phi theta psi(2)'
phi theta psi(3)'
ub,vb,wb(1)'
ub,vb,wb(2)'
ub,vb,wb(3)'
xe,ye,ze(1)'
xe,ye,ze(2)'
xe,ye,ze(3)'
p,q,r(1)'
p,q,r(2)'
p,q,r(3)'

Figura 19 vettore riga della matrice B

- linsys.c

```
>> linsys.c
ans =
0 -9.8100 0 -0.0000 -0.0000 0.0000 0 0 0 -0.0000 0.1380 0.0000
9.8100 -0.0000 0 -0.0000 -0.0000 -0.0000 0 0 0 -0.1380 0.0000 -0.0000
0.0000 0.0000 0 0 0 0 0 0 0 0 0 0
0 0.0000 0 0 0 0 0 0 0 0 0 0
-0.0000 -0.0000 -1.0000 0 0 0 0 0 0 0 0 0
0.0000 1.0000 -0.0000 0 0 0 0 0 0 0 0 0
0 0.0000 1.0000 0 0 0 0 0 0 0 0 0
0.0000 0 0 0 0 0 0 0 0 0 0 0
-1.0000 0.0000 -0.0000 0 0 0 0 0 0 0 0 0
0 -1.0000 0 0 0 0 0 0 0 0 0 0
1.0000 -0.0000 0 0 0 0 0 0 0 0 0 0
0.0000 0.0000 0 0 0 0 0 0 0 0 0 0
1.0000 0 0 0 0 0 0 0 0 0 0 0
0 1.0000 0 0 0 0 0 0 0 0 0 0
0 0 1.0000 0 0 0 0 0 0 0 0 0
0 0 0 0.0000 0 0 0.0000 0 0 0 0 0
0 0 0 0 0 0 0 0.0000 0 0 0 0
0 0 0 0 0 0 0 0 0 -1.0000 0 0
0 0 0 0 0 0 0 0 0 1.0000 0 0
0 0 0 0 0 0 0 0 0 0 1.0000 0
0 0 0 0 0 0 0 0 0 0 0 1.0000
0 0 0 0 1.0000 0 0 0 0 0 0 0
0 0 0 0 0 1.0000 0 0 0 0 0 0
0.0000 0.0000 0.0000 1.0000 -0.0000 -0.0000 0 0 0 0 0 0
-0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0 0 0 0 0 0
-0.0000 -0.0000 0 0.0000 -0.0000 1.0000 0 0 0 0 0 0
0 0 0 0 0 0 0 1.0000 0 0 0 0
0 0 0 0 0 0 0 0 1.0000 0 0 0
0 0 0 0 0 0 0 0 0 1.0000 0 0
0 0 0 -0.0000 -0.0000 -0.0000 0 0 0 -2.1715 0.0000 0.0000
0 0 0 0.0000 0.0000 -0.0000 0 0 0 0.0000 -1.6192 -0.0000
0 0 0 -0.0000 -0.0000 0.0000 0 0 0 0.0000 0.0000 -0.0000
```

Figura 21: matrice C

Matrice C (33 x 12): matrice delle uscite del sistema.

phi
theta
psi
ub
vb
wb
xe
ye
ze
pe
qe
re

Figura 23: vettore colonna della matrice C

'(I)ybod_iscA'
'(S)ybod_iscA'
'(E)ybod_iscA'
'(I)DCM_pe(1)'
'(S)DCM_pe(2)'
'(E)DCM_pe(3)'
'(I)DCM_pe(4)'
'(S)DCM_pe(5)'
'(E)DCM_pe(6)'
'(I)DCM_pe(7)'
'(S)DCM_pe(8)'
'(E)DCM_pe(9)'
'(I)enl(1)'
'(S)enl(5)'
'(E)enl(3)'
'(I)AJJ'
'(S)AJJ'
'(E)AJJ'
'(I)ybod_egemO'
'(S)ybod_egemO'
'(E)ybod_egemO'
'(I)ybod_V'
'(S)ybod_V'
'(E)ybod_V'
'(I)ben_V'
'(S)ben_V'
'(E)ben_V'
'(I)ben_X'
'(S)ben_X'
'(E)ben_X'
'(I)ybod_egemOb'
'(S)ybod_egemOb'
'(E)ybod_egemOb'

Figura 22: vettore riga della matrice C

- linsys.d

```

>> linsys.d
ans =
-0.0000  0.0000  -0.0000  0.0000  -0.0000  0  0  0  1.0000  0.0000  0.0000  0  0  0
0.0000  -0.0000  0.0000  -0.0000  0.0000  0  0  0  -0.0000  1.0000  -0.0000  0  0  0
-0.0096  0.0096  -0.0096  0.0096  -0.2855  0  0  0  0  0  1.0000  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0
0.4202  0.4202  -0.4202  -0.4202  0  0  0  0  0  0  0  0  0  0
0.3133  -0.3133  -0.3133  0.3133  0  0  0  0  0  0  0  0  0  0
-0.0115  -0.0115  -0.0115  -0.0115  -0.0000  0  0  0  0  0  0  0  0  0

```

Figura 24: matrice D

Matrice D (33 x 14): descrive l'azione sull'uscita dovuta al valore delle variabili di ingresso.

'Actuators(1)'
'Actuators(2)'
'Actuators(3)'
'Actuators(4)'
'AtmosphereBus.air_density'
'AtmosphereBus.air_temp'
'AtmosphereBus.pressure'
'AtmosphereBus.speed_sound'
'Gravity_ned(1)'
'Gravity_ned(2)'
'Gravity_ned(3)'
'MagneticField_ned(1)'
'MagneticField_ned(2)'
'MagneticField_ned(3)'

Figura 26: vettore colonna della matrice D

'Accel_body(1)'
'Accel_body(2)'
'Accel_body(3)'
'DCM_be(1)'
'DCM_be(2)'
'DCM_be(3)'
'DCM_be(4)'
'DCM_be(5)'
'DCM_be(6)'
'DCM_be(7)'
'DCM_be(8)'
'DCM_be(9)'
'Euler(1)'
'Euler(2)'
'Euler(3)'
'LLA(1)'
'LLA(2)'
'LLA(3)'
'Omega_body(1)'
'Omega_body(2)'
'Omega_body(3)'
'V_body(1)'
'V_body(2)'
'V_body(3)'
'V_ned(1)'
'V_ned(2)'
'V_ned(3)'
'X_ned(1)'
'X_ned(2)'
'X_ned(3)'
'dOmega_body(1)'
'dOmega_body(2)'
'dOmega_body(3)'

Figura 25: vettore riga della matrice D

Per quanto riguarda gli ingressi del modello lineare, oltre al contributo dei motori (*Actuatos*), sono presenti anche variabili legate all'ambiente (*Environment*): temperatura e densità dell'aria, pressione atmosferica, velocità del suono, contributo della gravità e del campo magnetico. Oltre al valore dei coefficienti è possibile ottenere anche il nome delle variabili di stato, degli ingressi e delle uscite aprendo il file *StateName*. Sotto le matrici sono state riportate delle tabelle i quali rappresentano i vettori su cui esse agiscono e i nomi delle variabili coinvolte (riga x colonna).

Mentre per avere lo stesso modello ma non lineare basterà settare il valore della variabile *VSS VEHICLE = 1* ed andare ad aggiornare il blocco.

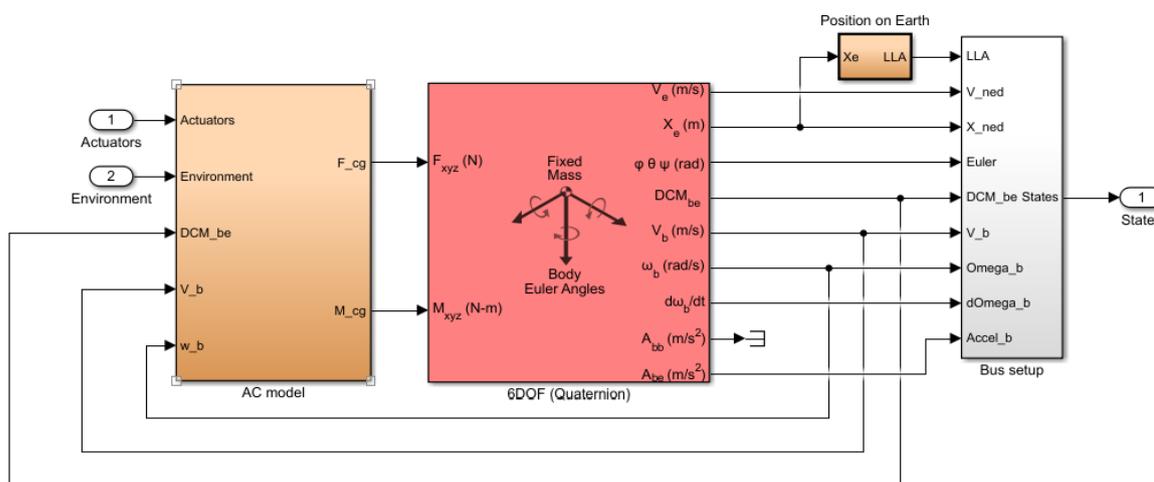


Figura 27: modello non lineare

2.5 blocco di controllo

Il blocco di controllo (*Flight Control System*) riceve come segnali di ingresso i valori dei riferimenti provenienti dal blocco *Signal Builder* e i valori 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.

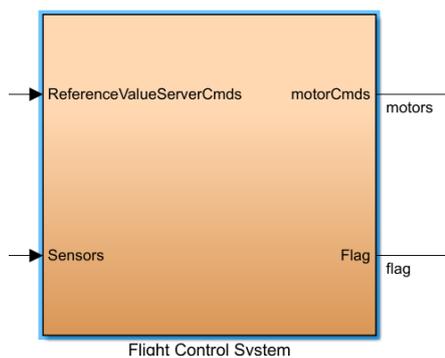


Figura 28: Blocco FCS

Il FCS è suddiviso in altri cinque blocchi:

- State Estimator;
- Flight Controller;
- Crash Predictor Flag;
- Logging;
- Estimator.

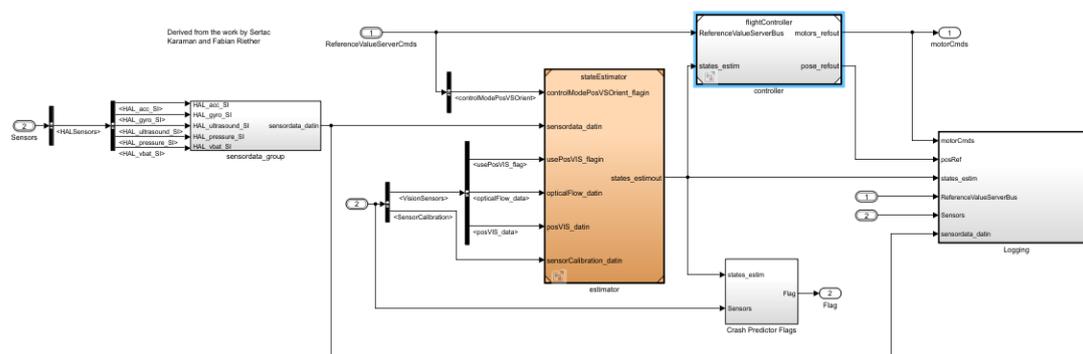


Figura 29: interno del blocco FCS

Per visualizzare i vari controllori presenti si utilizzerà il blocco *Flight Controller* il quale è composto da quattro sottosistemi principali che si occupano di generazione dei segnali di controllo che definiranno il comportamento del veicolo:

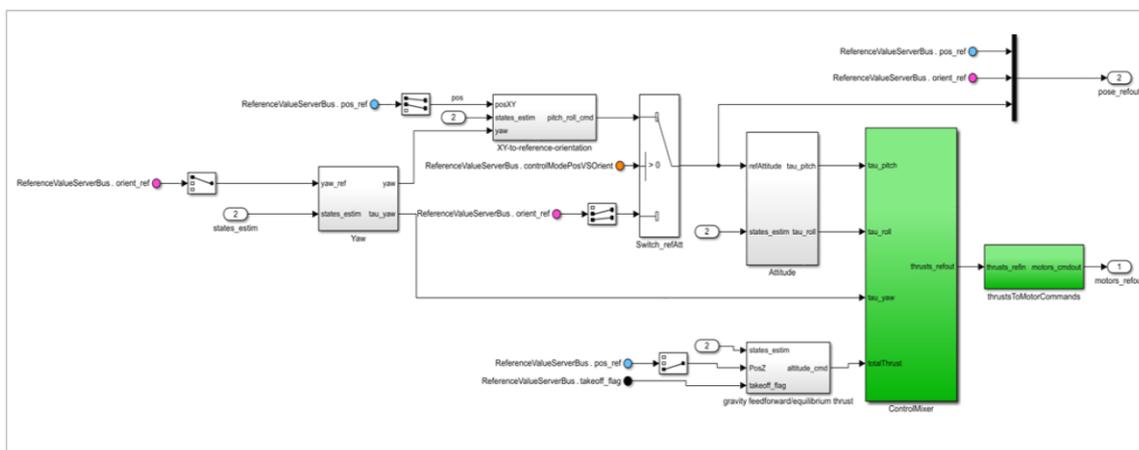


Figura 30: blocco FlightController

1. **XY to reference orientation:** riceve in ingresso la posizione longitudinale e latitudinale del veicolo, l'angolo di yaw e il valore corrente della posizione e della velocità. Questo blocco risulta strettamente legato al blocco Attitude, infatti in uscita elabora un segnale di controllo per il quale gli angoli di pitch e di roll devono assumere per garantire la posizione di riferimento.
2. **Attitude:** riceve in ingresso il valore degli angoli di pitch e roll, e delle relative velocità angolari, p e q , e, a seconda del valore dello switch, il segnale di riferimento relativo alla posizione oppure quello relativo all'orientamento. In uscita fornisce il segnale di controllo per l'angolo di pitch e roll.
3. **Yaw:** riceve in ingresso il riferimento relativo all'angolo di Yaw, quindi anche in questo caso avremo il valore dell'angolo e la propria velocità angolare r . In uscita fornisce il segnale di controllo per l'angolo di Yaw.
4. **Gravity Feedforward/Equilibrium Thrust:** riceve in ingresso il segnale di riferimento relativo alla quota (Z), quindi il valore corrente e la propria velocità questa volta non angolare ma verticale del veicolo e il segnale Take Off Flag che regola l'azione di decollo. In uscita verrà elaborato il segnale di controllo relativo all'altitudine del veicolo.

2.5.1 PID

Prima di andare ad analizzare i vari blocchi di controllo risulta essere opportuno dare una breve spiegazione sul controllore usato frequentemente nel mondo dell'automatica, il PID.

I controllori PID sono algoritmi di regolazione da impiegarsi in sistemi di controllo ad anello chiuso, cioè a reazione negativa o in controreazione, dove l'ingresso di controllo è dato dalla somma di tre componenti: una **Proporzionale**, una **Integrale** ed una **Derivativa**.

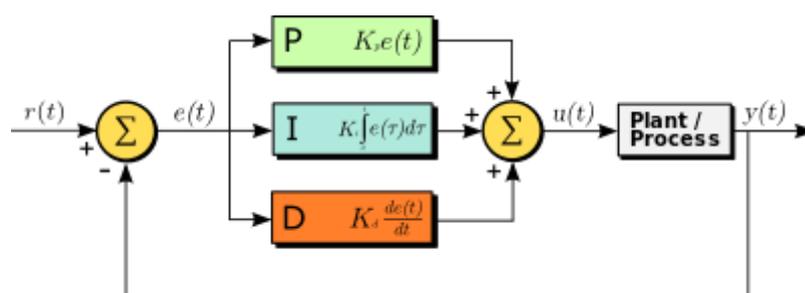


Figura 31: PID controller

Azione Proporzionale (P)

L'uscita è direttamente proporzionale all'ingresso.

Equazione nel dominio del tempo

Funzione di trasferimento

$$m(t) = K_G e'(t)$$

$$G(s) = \frac{M(s)}{E'(s)} = K_G$$

K_G = guadagno del termine proporzionale

Con questo controllore P, avendo assunto che il processo non ha poli nell'origine, il sistema a ciclo chiuso è di tipo 0, pertanto con un ingresso a gradino l'errore non sarà mai nullo ma finito:

$$\tilde{e}_0 = W_e(s) \Big|_{s=0} = \frac{k_d^2}{k_d + K_F} \quad K_F = K_G K_P$$

Aumentando il guadagno del controllore l'errore diminuisce quindi la prestazione di regime "migliora"; l'aumento del guadagno può però rendere il sistema instabile.

Azione Integrativa (I)

L'uscita proporzionale all'area sottesa dalla curva errore-tempo

Equazione nel dominio del tempo

funzione di trasferimento

$$m(t) = K_I \int_0^t e'(\tau) d\tau \quad G(s) = \frac{M(s)}{E'(s)} = \frac{K_I}{s}$$

K_I = guadagno del termine integrale

Questo tipo di controllore introduce un polo nell'origine nella funzione di trasferimento della catena diretta: sistema sicuramente di tipo 1. Inoltre, il sistema è astatico rispetto a segnali di disturbo costanti e non noti che possono agire all'ingresso o all'uscita del processo da controllare. In generale con questo tipo di controllo, migliorano le prestazioni di regime e "peggiora la stabilità".

Azione Derivativa (D)

L'uscita non dipende dall'errore presente o passato, ma dalla velocità alla quale varia l'errore.

Equazione nel dominio del tempo

funzione di trasferimento

$$m(t) = K_D \frac{d}{dt} e'(t) \quad G(s) = \frac{M(s)}{E'(s)} = K_D s$$

K_D = guadagno del termine derivativo.

Questo tipo di controllo non viene mai utilizzato da solo ma assieme a P e PI, perché in generale peggiora le prestazioni di regime ma "migliora la stabilità".

Per poter utilizzare un controllore composto da più controlli (PI, PD, PID) viene utilizzata un'implementazione in parallelo.

PID

Equazione nel dominio del tempo

$$m(t) = K_G e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{d}{dt} e(t)$$

Funzione di trasferimento

$$G(s) = \frac{m(s)}{e(s)} = K_G + \frac{K_I}{s} + K_D s = K_G \left(1 + \frac{K_I}{K_G s} + \frac{K_D s}{K_G} \right) \rightarrow$$

$$G(s) = K_G \left(1 + \frac{1}{\tau_I s} + \tau_D s \right) = K_G \left(\frac{\tau_D \tau_I s^2 + \tau_I s + 1}{\tau_I s} \right)$$

$$\tau_I = \frac{K_G}{K_I} \quad \tau_D = \frac{K_D}{K_G}$$

In questo modo si uniscono le caratteristiche di (P) con quelle di (D) e di (I).

2.5.2 blocco XY to reference orientation

Il seguente diagramma è suddiviso in due parti, in cui nella parte superiore viene utilizzato l'angolo di Yaw, mentre nella parte inferiore vengono sommati agli errori sulla posizione longitudinale e latitudinale, le componenti X, Y calcolate rispetto al sistema di riferimento inerziale dovuto alla rotazione del drone sull'asse Z. L'angolo di Yaw effettua il cambio di variabili dal sistema del corpo centrale a quello inerziale. Nella parte inferiore possiamo notare anche le rispettive velocità lungo l'asse delle ascisse e delle ordinate. Il tutto sarà moltiplicato per le proprie costanti. In sintesi, sommando i risultati dei due rami, si viene a creare un controllore PD (Proporzionale Derivativo). L'angolo di Pitch e di Roll vengono calcolati in modo da permettere il raggiungimento della X e Y di riferimento attraverso un PID, nel quale verranno calcolati i segnali di riferimenti relativi all'angolo di Pitch e Roll; Quindi abbiamo un confronto con il pitch e roll attuali e gli errori derivanti.

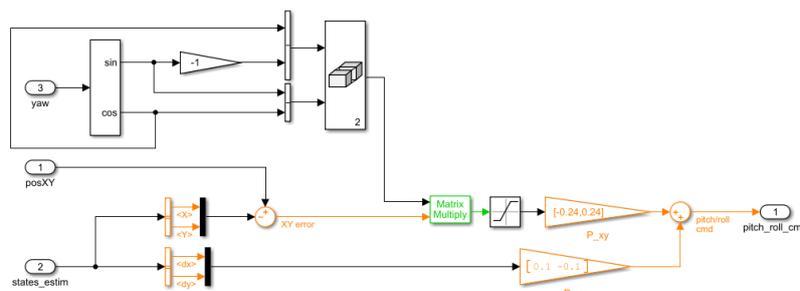


Figura 32: blocco di controllo: XY to reference orientation

Il segnale di riferimento longitudinale e latitudinale sarà regolato dallo stesso controllore.

2.5.3 blocco Attitude

Il controllore nel blocco *Attitude* viene utilizzato sia per la regolazione dell'angolo di Pitch che per l'angolo di Roll. In ingresso avremo anche qua gli angoli di Roll e Pitch con le proprie derivate, cioè le proprie velocità angolari (p e q).

Per ricavare l'errore rispetto al riferimento, viene confrontato i valori attuali degli angoli di Roll e Pitch con il segnale di controllo generato dal *blocco XY to reference orientation* o il segnale di orientamento proveniente dal blocco *Segnal Builder* (il segnale di riferimento all'interno del blocco è denominato genericamente *ref Attitude*). Anche qua, come per il controllore precedente, lo schema in cui l'azione derivativa è messa in controazione.

Nella parte inferiore, le velocità p e q sono moltiplicate per le opportune costanti. Sommando il tutto si ottiene un controllore PID, guardando dall'alto al basso si nota l'azione proporzionale, integrale e derivativa.

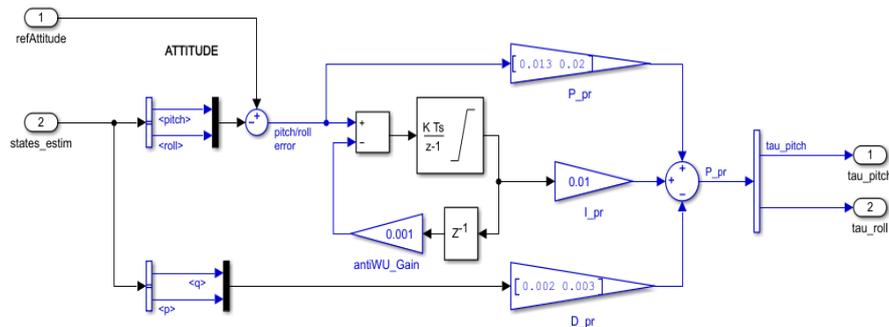


Figura 33: blocco di controllo: Attitude

2.5.4 blocco Yaw

La prima differenza che possiamo notare nel controllore utilizzato per regolare l'angolo di Yaw, al contrario di quelli visti fino ad ora, è proprio l'azione applicata all'uscita e non all'errore, entrando in controreazione. Nella parte superiore abbiamo il valore attuale dell'angolo di Yaw, dove avviene l'azione proporzionale, mentre nella parte inferiore abbiamo la velocità angolare r , moltiplicate tutte e due per le opportune costanti, per poi essere sommate creando un controllore sotto forma di PD.

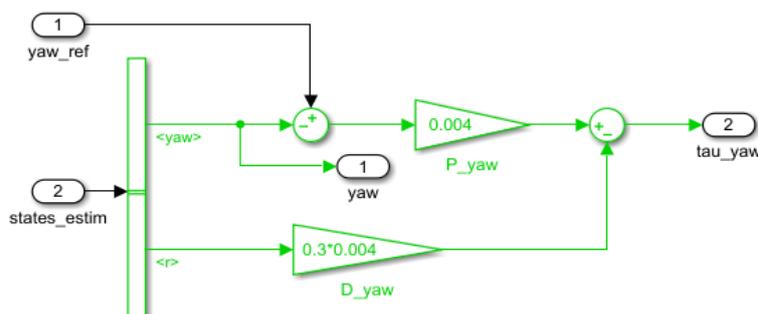


Figura 34: blocco di controllo: Yaw

2.5.5 blocco Gravity

L'ultimo controllore da analizzare è quello relativo alla spinta verticale, quindi il grado di libertà lungo l'asse Z. Anche qui possiamo notare immediatamente l'azione del controllore di tipo PD (Proporzionale-Derivativo), con il valore attuale della variabile Z e la rispettiva derivata, quindi la velocità. Il tutto andrà a finire in uno Switch, *TakeOfControl_Switch*, il quale verrà utilizzato per generare una potenza maggiore per gli attuatori nella fase di decollo. Lo switch utilizzerà una variabile di tipo flag per sapere se il veicolo è in fase di decollo o no. Nel caso in cui il veicolo non si trovasse in fase di decollo, il flag sarà selezionato a NULL e lo switch selezionerà il controllore.

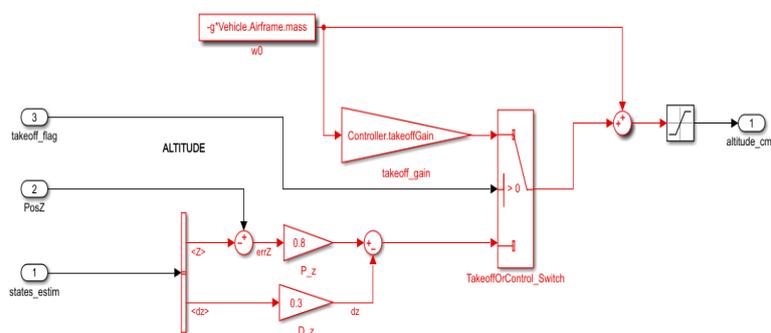


Figura 35: blocco di controllo: Gravity

2.6 blocco Signal Builder

Nel paragrafo precedente si è detto che il controllo degli angoli di roll e pitch ed il controllo della posizione X-Y sono strettamente legati, quindi andiamo ad analizzare il generatore dei segnali.

Dal blocco *Command* è possibile scegliere la fonte dei segnali di riferimento per il sistema di controllo del drone. Anche in questo caso, modificando il valore della variabile *VSScommand* sarà possibile selezionare:

- Il blocco **Signal Builder**: costruttore di segnali di default (*VSScommand* = 0);
- Il blocco **Joystick**: se intendiamo gestire manualmente il comportamento del drone (*VSScommand* = 1);
- Il blocco **Data**: se intendiamo selezionare i valori dei riferimenti presenti in un file (*VSScommand* = 2);
- Il blocco **Spreadsheet Data**: ha la stessa funzionalità del blocco precedente, solo che i valori dei riferimenti vengono selezionati da un foglio di calcolo (*VSScommand* = 3);

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

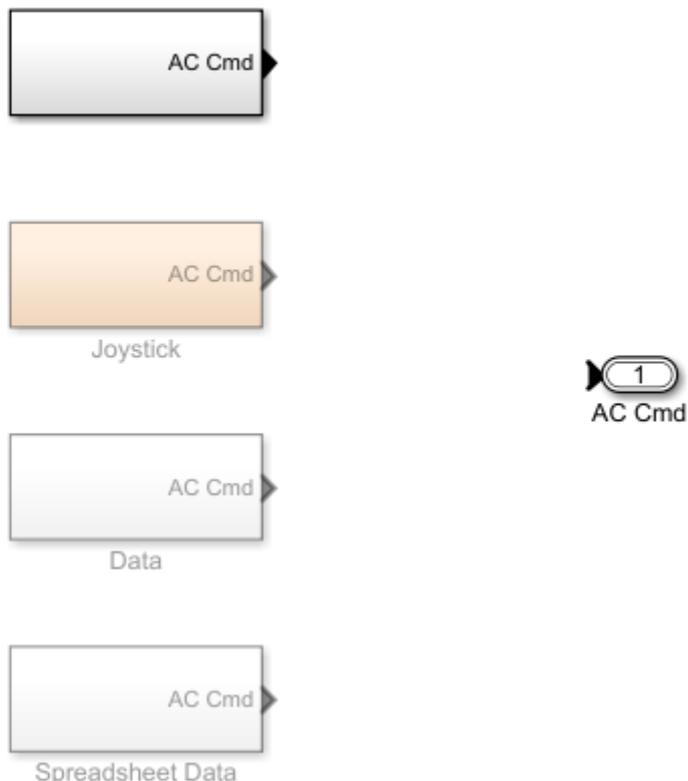


Figura 36: VSS_COMMAND

A seconda della variabile *VSS_COMMAND*, verrà evidenziato il blocco desiderato; per default *VSS_COMMAND* sarà impostata al valore zero, quindi il blocco attivo sarà quello del *Signal Builder*, tale blocco sarà utilizzato in seguito nella progettazione del controllore.

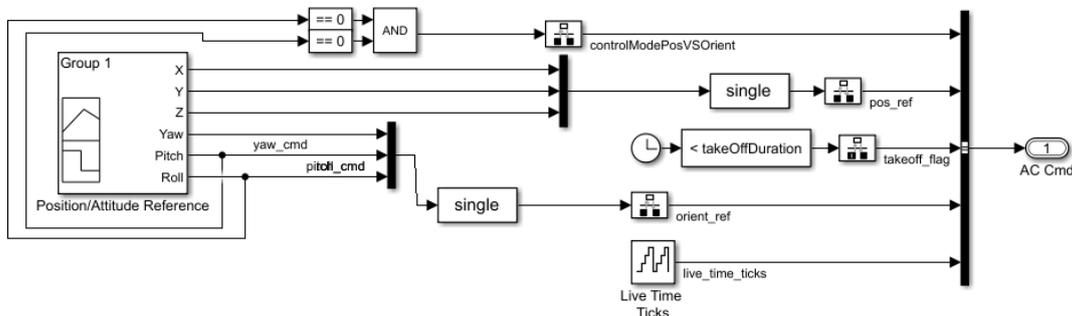


Figura 37: blocco Signal Builder

Nel *Position/Attitude Reference* sono presenti i sei segnali di riferimento, di cui i primi tre segnali (X, Y, Z) danno un segnale di riferimento di posizione, mentre gli ultimi tre segnali (Yaw, Pitch, Roll) forniscono il riferimento di orientamento.

Selezionando lo stesso blocco, Group 1, è possibile visualizzare e regolare l'andamento delle variabili di riferimento relative alla posizione ed all'orientamento in aria del drone.

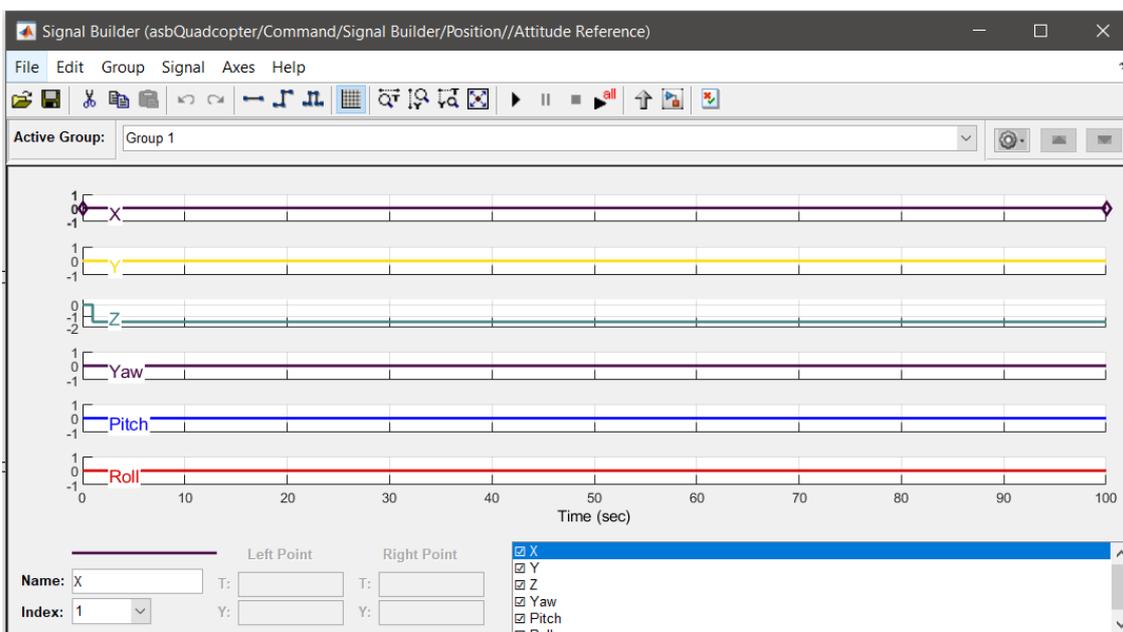


Figura 38: blocco Group all'interno del Signal Builder

Tutte e sei le variabili hanno come condizioni iniziali il valore di zero.

Capitolo 3

Analisi modello PID

In questo capitolo si andrà ad analizzare i vari comportamenti di tutti e sei i gradi di libertà sotto il controllo del PID visto nel capitolo precedente.

3.1 analisi del modello lineare (PID)

Dal blocco *FCS* dovremo andare a selezionare il blocco *Logging*, blocco in cui vengono contenute tutte le variabili coinvolte nel modello.

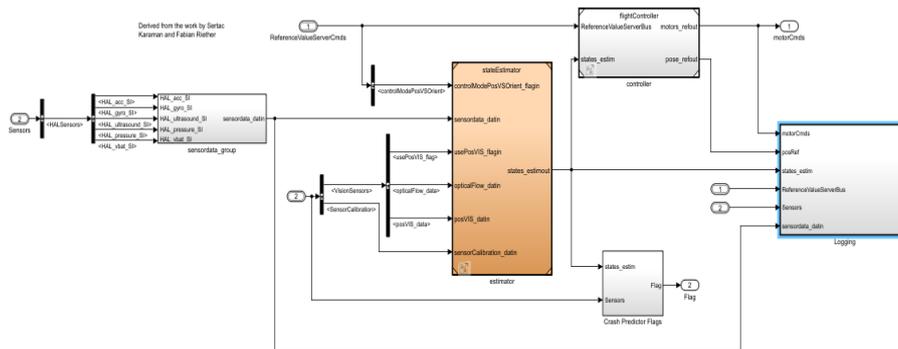


Figura 39: blocco FCS (interno)

Per visualizzare i vari grafici del comportamento delle variabili rispetto al tempo, bisogna andare ad inserire dei blocchi di tipo *Scope* e collegarli ad ogni variabile interessate, come rappresentato nella figura seguente.

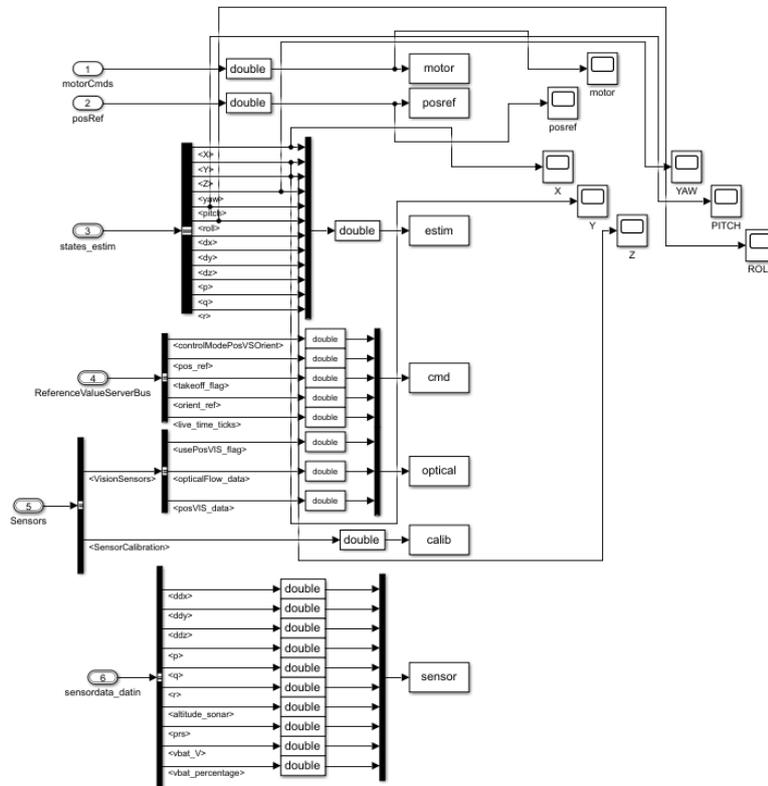


Figura 40: blocco Logging

Da noi interessati sono le sei variabili di stato relative al sistema (X, Y, Z, Pitch, Roll, Yaw) ed il ruolo dei motori.

Una volta inseriti i vari blocchi Scope, selezionandoli si potrà analizzare i seguenti grafici:

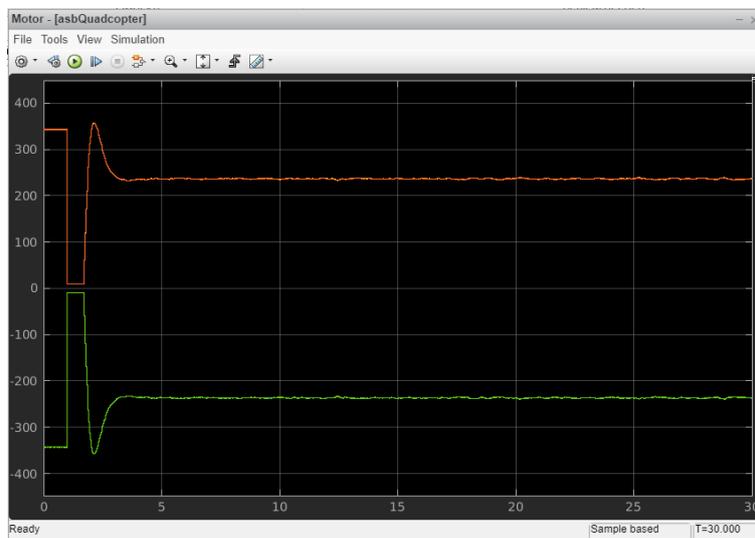


Figura 41: grafico dei motori sotto controllo del PID nel modello lineare

I motori del drone hanno, a due a due, direzioni di rotazione opposte e quindi segni opposti.

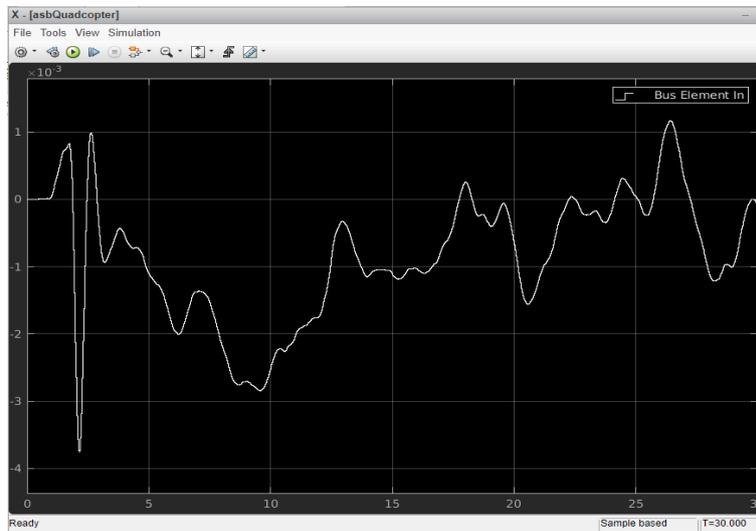


Figura 42: grafico X sotto controllo del PID nel modello lineare

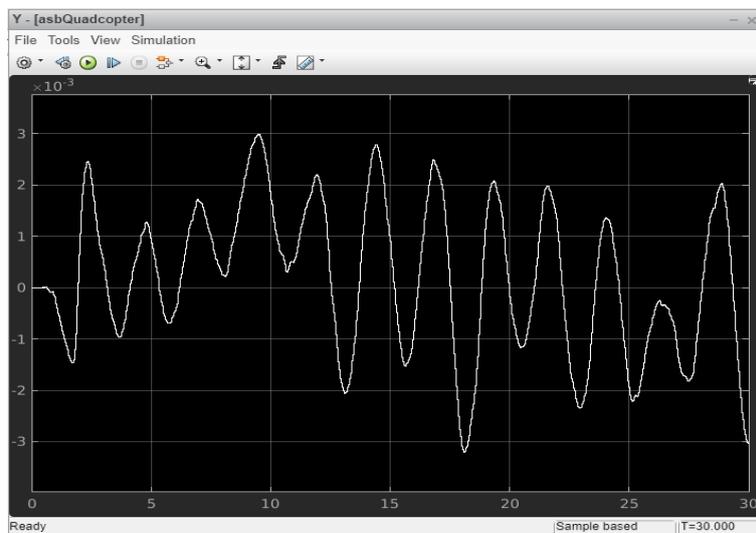


Figura 43: grafico Y sotto controllo del PID nel modello lineare

Per i grafici relativi alle variabili di stato X e Y, si nota facilmente che l'oscillazione risulta essere limitata rispetto al riferimento nell'ordine di $\pm 10^{-3}$, un ordine di grandezza talmente piccolo garantito dalla buona qualità dell'azione di controllo del PID.

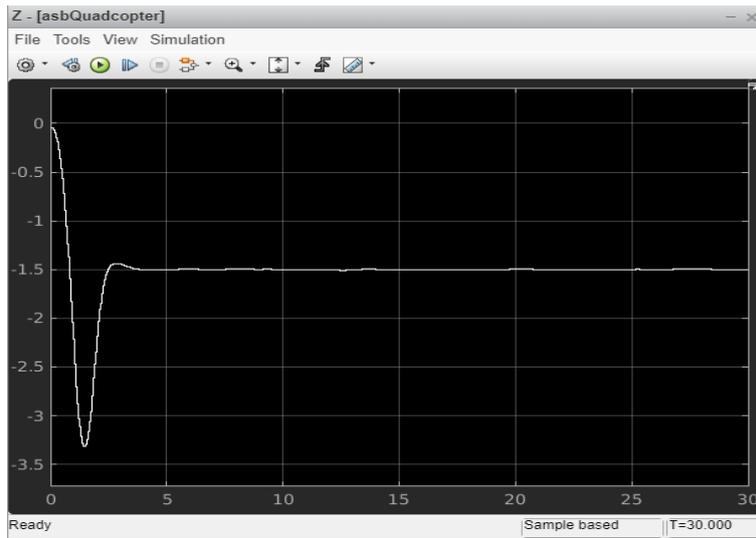


Figura 44: grafico Z sotto controllo del PID nel modello lineare

Dopo una sovraelongazione iniziale, il tracciato si attesta ad un valore di -1.5 m sul suolo. Viene considerato il segno negativo in quanto il sistema di riferimento è di tipo NED (North East Down).

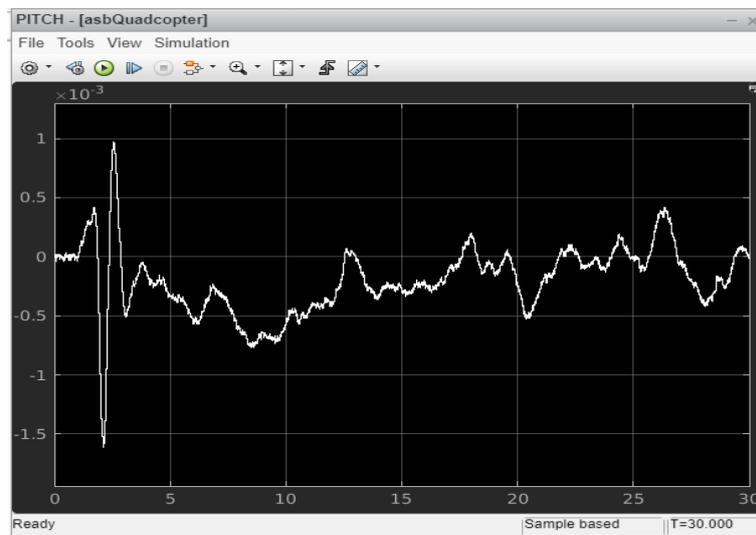


Figura 45: grafico Pitch sotto controllo del PID nel modello lineare

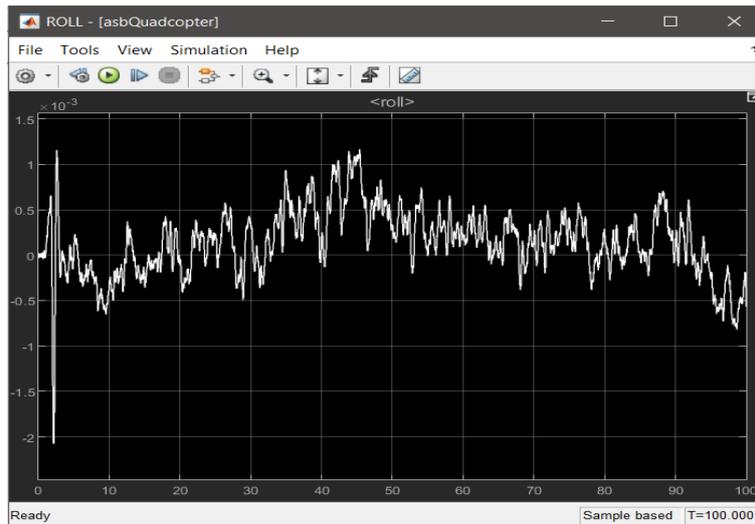


Figura 46: grafico Roll sotto controllo del PID nel modello lineare

Tracciato Roll

Le considerazioni fatte per i grafici degli angoli di Pitch e di Roll sono le stesse fatte per la posizione X e Y.

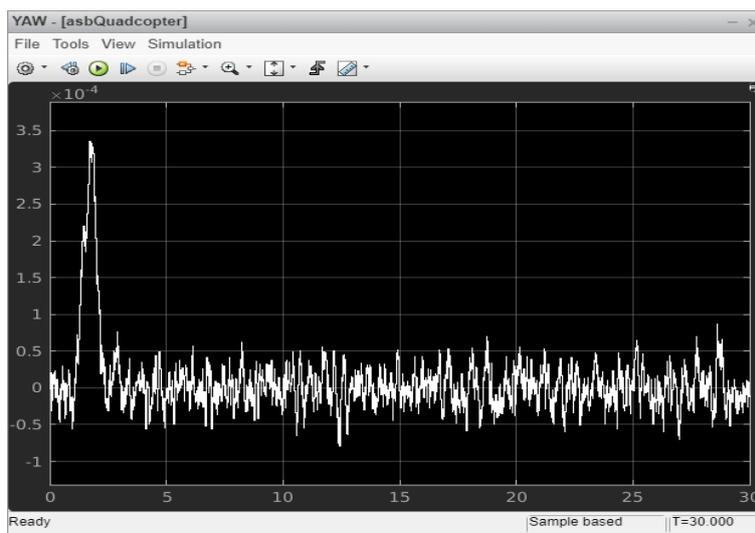


Figura 47: grafico Yaw sotto controllo del PID nel modello lineare

Possiamo fare una simile considerazione anche per il grafico dell'angolo di Yaw, nel quale dopo un picco di salita iniziale, l'oscillazione risultano nell'ordine di $\pm 10^{-4}$ rispetto al riferimento.

3.2 analisi del modello non lineare (PID)

Per default il modello è quello lineare, come detto nei capitoli precedenti, per passare al modello non lineare bisogna impostare la variabile $VSS_VEHICLHE = 1$.

Lo svolgimento per visualizzare i vari grafici è lo stesso per il modello lineare, anche i rispettivi grafici risultano del tutto analogo al modello lineare.

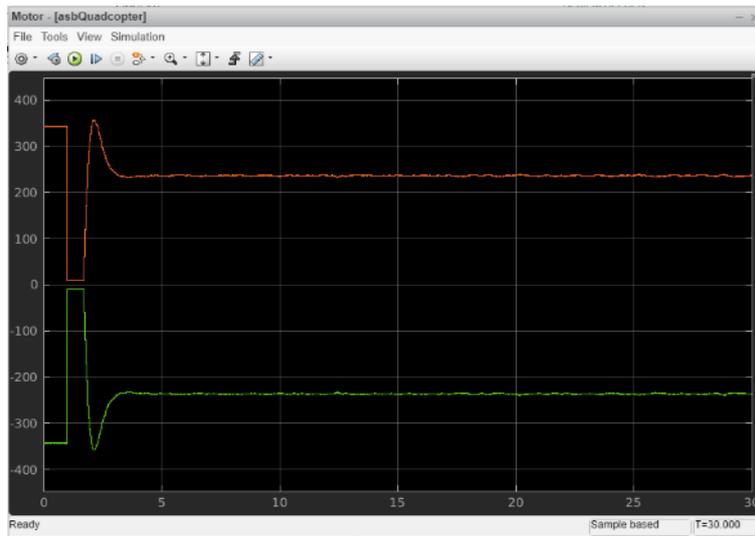


Figura 48: grafico dei motori sotto controllo del PID nel modello non lineare

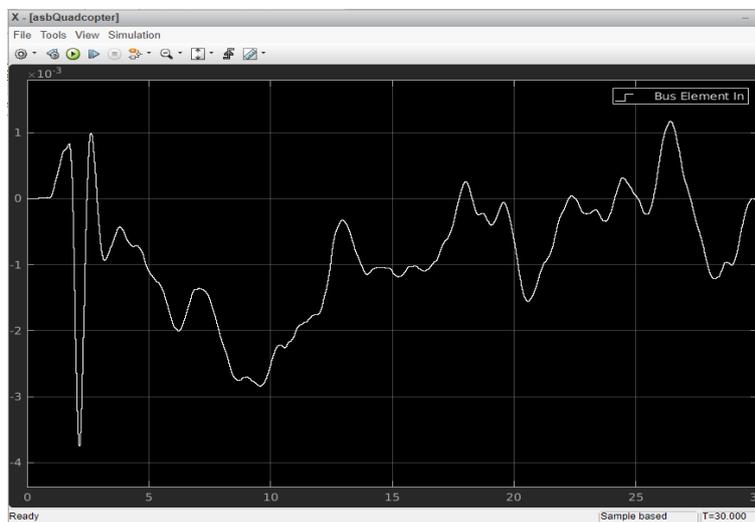


Figura 49: grafico X sotto controllo del PID nel modello non lineare

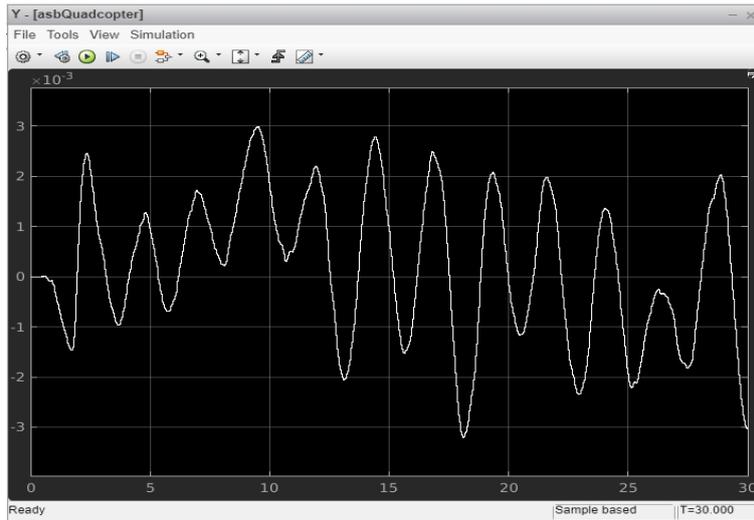


Figura 50: grafico Y sotto controllo del PID nel modello non lineare

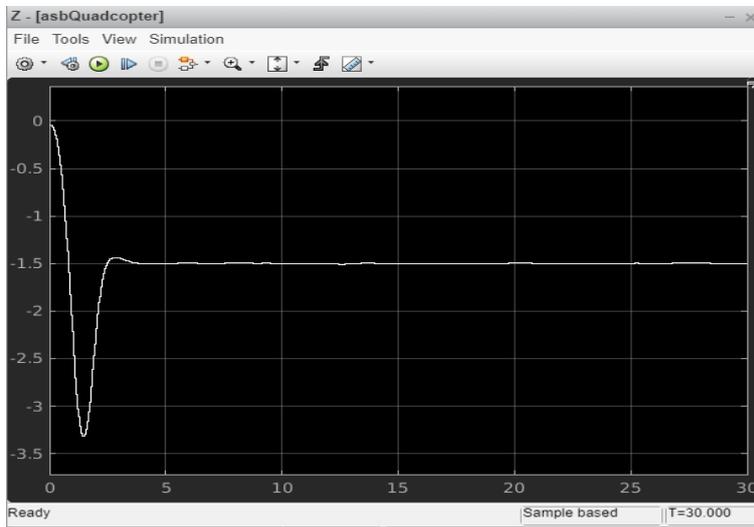


Figura 51: grafico Z sotto controllo del PID nel modello non lineare

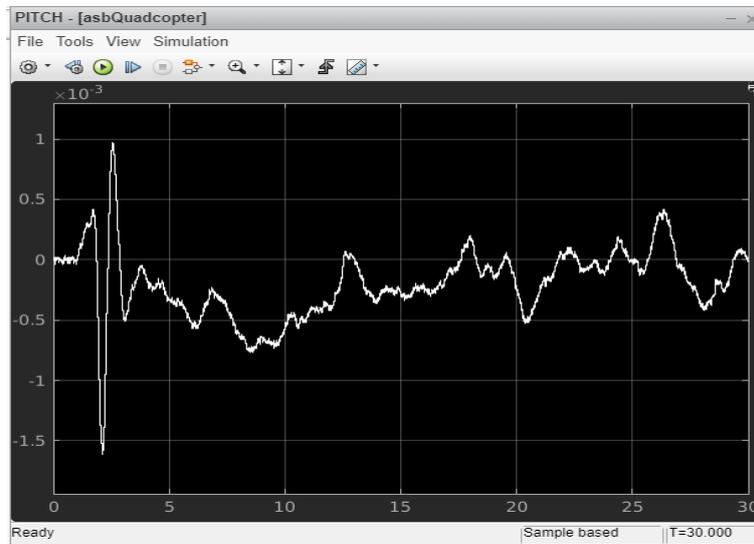


Figura 52: grafico Pitch sotto controllo del PID nel modello non lineare

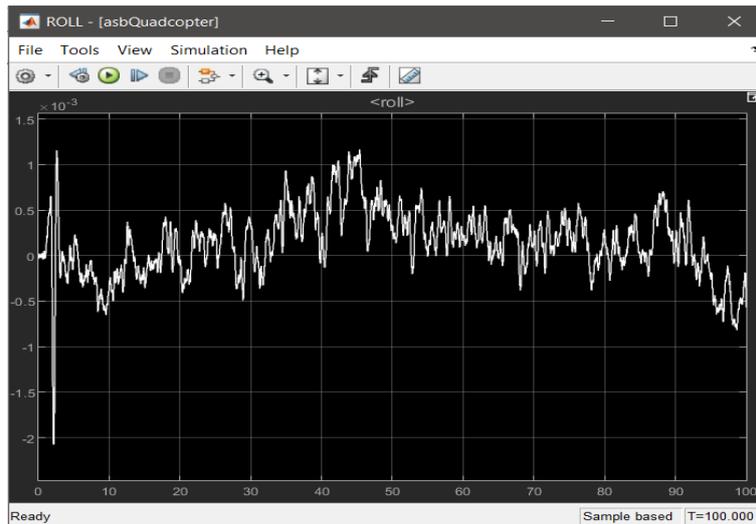


Figura 53: grafico Roll sotto controllo del PID nel modello non lineare

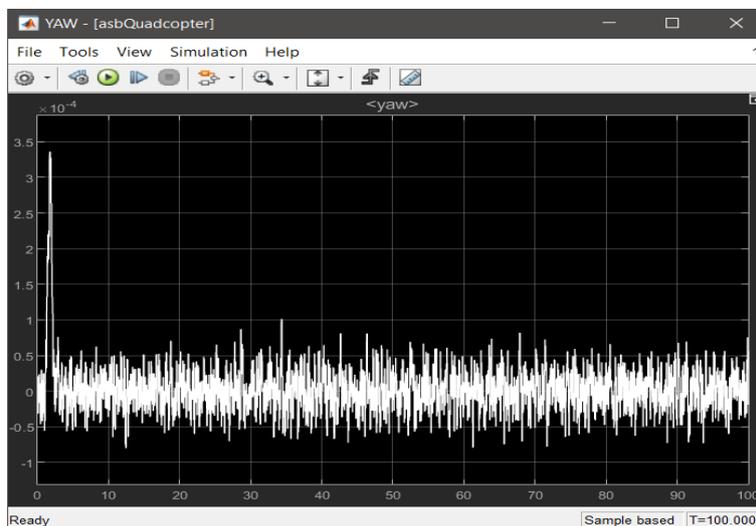


Figura 54: grafico Yaw sotto controllo del PID nel modello non lineare

Quindi anche le considerazioni fatte per i grafici del controllo del modello lineare valgono per i grafici del modello non lineare.

Capitolo 4

Sviluppo controllore per l'angolo di rollio

In questa sezione entriamo nel vivo del progetto, cioè si va a creare un controllore per uno dei sei gradi di libertà, quello scelto da me è l'angolo di Rollio, attraverso la tecnica di controllo per frequenza.

4.1 sintesi in frequenza per tentativi

Principalmente esistono due tecniche utilizzate per la realizzazione di un controllore:

- la sintesi del luogo delle radici.
- la sintesi in frequenza.

Quella utilizzata in questo progetto è la seconda, cioè la tecnica della sintesi in frequenza.

Questa tecnica di progetto è indicata come sintesi per tentativi, perché l'effettivo soddisfacimento di tutte le specifiche avviene spesso per correzioni successive (tentativi) nella definizione delle reti di compensazione, i quali le vedremo in seguito.

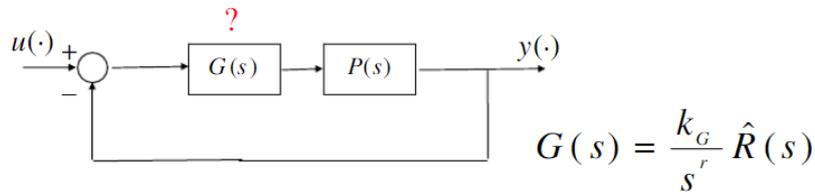
Dopo aver creato il nostro controllore il quale comporrà una funzione di trasferimento, ricordando che tale funzione risulta essere una funzione razionale, ossia un rapporto di polinomi in s , gli zeri e i poli sono rispettivamente le radici del polinomio al numeratore e al denominatore, bisognerà andare a soddisfare delle specifiche sia per la funzione in catena aperta che in catena chiusa.

$$G(s) = \frac{F(s)}{X(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \dots + b_n}{s^n + a_1 s^{n-1} + \dots + a_n}$$

Una radice è un valore che annulla il valore complessivo di un polinomio.

-> Gli zeri sono i valori di s che azzerano la funzione $G(s)$ perché annullano il polinomio al numeratore $F(s)$.

-> I poli della trasformata sono i valori di s che rendono infinita la funzione $G(s)$ perché azzerano il polinomio al denominatore $X(s)$.



La funzione di controllo $G(s)$ è stata progettata in modo che l'uscita del sistema soddisfacesse determinate specifiche:

Univoche: soddisfatte semplicemente con la funzione di controllo $G(s)$.

Lasche: soddisfatte attraverso le funzioni compensatrici $R(s)$.

Le funzioni compensatrici utilizzate in questo progetto sono le seguenti:

- Funzione anticipatrice

$$R_a(s) = \frac{1 + \tau_a s}{1 + \frac{\tau_a}{m_a} s}, \quad \tau_a > 0, \quad m_a > 1$$

- Funzione attenuatrice

$$R_i(s) = \frac{1 + \frac{\tau_i}{m_i} s}{1 + \tau_i s}, \quad \tau_i > 0, \quad m_i > 1$$

utilizzate per modificare il modulo e la fase della nostra funzione di trasferimento come vedremo in seguito.

Le specifiche dovranno essere soddisfatte sia in catena chiusa $W(s)$ che in catena aperta $F(s)$; vediamo alcuni passaggi per la traduzione delle specifiche da catena chiusa a catena aperta.

Sistema ad anello chiuso

Banda passante desiderata
 B_3

trasformato in

Sistema ad anello aperto

Pulsazione di attraversamento
 ω_t^* desiderata
per la $F(j\omega)$

Sistema ad anello chiuso

Modulo alla risonanza
 $M_r \leq M_{r \max}$

trasformato in

Sistema ad anello aperto

Margine di fase
 $m_\varphi \geq m_\varphi^*$
per la $F(j\omega)$

Quindi in sintesi la procedura si riassume nei seguenti quattro punti:

- **Assegnazione** K_G, s^r .
- **Verifica** su $\hat{W}(s) \leftrightarrow \hat{F}(s) = \frac{k_G}{s^r} P(s)$
- **Scelta** di $\hat{R}(s)$
- **Verifica** su $W(s) \leftrightarrow F(s) = G(s)P(s)$

4.2 creazione della funzione di trasferimento del processo P(s)

Una volta caricato le quattro matrici della dinamica, come si è visto nella sezione precedente, si va a creare altre quattro sottomatrici relative alle variabili dell'angolo di Roll.

Le quattro matrici della dinamica saranno le seguenti:

- Matrice della dinamica degli stati del sistema (2 x 2):

$$A = \begin{vmatrix} 0 & 1.000 \\ 0 & -2.175 \end{vmatrix}$$

In tale matrice saranno presenti i coefficienti relativi alle variabili di stato ϕ e p .

- Matrice degli ingressi del sistema (2 x 1):

$$B = \begin{vmatrix} 0 \\ 0.420 \end{vmatrix}$$

In tale matrice saranno presenti i coefficienti relativi alle variabili di stato ϕ e p , rispetto agli ingressi **Actuators (1)(2)(3)(4)**.

- Matrice delle uscite del sistema (1 x 2):

$$C = \begin{vmatrix} 1 & 0 \end{vmatrix}$$

In tale matrice saranno presenti i coefficienti relativi all'uscita **Euler (1)** corrispondente all'angolo di Roll.

- Matrice dell'azione sull'uscita dovuta al valore delle variabili in ingresso:

$$D = | \mathbf{0} |$$

In tale matrice saranno presenti i coefficienti relativi all'uscita **Euler (1)** legata agli ingressi **Actuators (1)(2)(3)(4)**.

L'obiettivo è quello di creare una funzione di trasferimento del processo, relativa alla variabile dell'angolo di Roll, per poi poterci progettare un controllore.

$$F_{dt} = \mathbf{C} (\mathbf{s} \mathbf{I} - \mathbf{A})^{-1} \mathbf{B} + \mathbf{D}$$

La funzione sarà poi moltiplicata alle rispettive costanti, in particolare a due costanti K_1 e K_2 :

Il blocco **Control Mixer** conterrà la matrice relative alle costanti con cui a seconda della variabile di stato da controllare viene gestita la partizione del controllo.

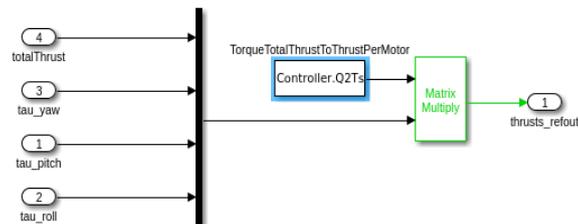


Figura 55: blocco ControlMixer

Attraverso il codice *Controller.Q2Ts*, sarà possibile visualizzare a schermo la matrice quadrata 4 x 4 relativa alla costante K_1 :

```
>> Controller.Q2Ts
```

```
ans =
```

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

Figura 56: matrice relativa al blocco ControlMixer

L'ultima colonna rappresenta la costante relativa all'angolo di Roll.

La seconda costante è contenuta nel blocco **ThurstsToMotorCommands** ed è la costante moltiplicativa la quale previene i motori dalla possibilità di raggiungere i regimi non concessi.

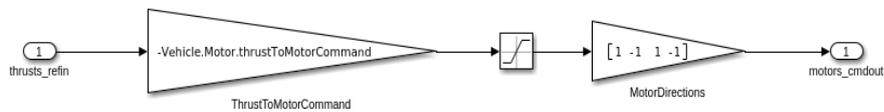


Figura 57: blocco *ThurstsToMotorCommands*

Attraverso il codice *Vehicle.Motor.thrustToMotorComand*, si visualizza a schermo la costante K_2 :

```
>> Vehicle.Motor.thrustToMotorCommand

ans =

1.5307e+03
```

Figura 58: costante K_2

Con i dati a disposizione, attraverso delle funzionalità di matlab, siamo in grado di ricavare la funzione di trasferimento del processo $P(s)$:

```
>> A=[0 1;0 -2.175];
B=[0;0.420];
C=[1 0];
D=[0];
syms s;
sys=ss(A,B,C,D);
Plant=tf(sys);
k1=5.6659;
k2=1530.7;
Ps=Plant*k1*k2

Ps =

3643
-----
s^2 + 2.175 s
```

Figura 59: funzione di trasferimento del processo $P(s)$

4.3 controllo di primo tentativo

Prima di procedere con la progettazione del controllore, vediamo le specifiche da soddisfare, le cosiddette specifiche lasche ed univoche viste nella sezione precedente.

Per quanto riguarda la dinamica di rollio, la risposta dinamica nella velocità angolare p deve essere del primo ordine e, tenendo conto dei requisiti richiesti per garantire delle qualità di volo adeguate, deve avere una frequenza tale da consentire di raggiungere determinati valori dell'angolo di rotazione propria (simbolo angolo) in un fissato intervallo di tempo. Per soddisfare questo requisito, la pulsazione si è stabilita essere $\omega_t = 10 \text{ rad/s}$.

Le specifiche per la funzione in catena chiusa $W(s)$ sono le seguenti:

1. Il sistema deve essere di tipo 1;
2. L'errore a regime permanente $\tilde{e} \leq 0.01$;
3. Margine di risonanza $M_r \leq 2 \text{ dB}$;
4. Banda passante $B_3 = 2 \text{ Hz}$.

Le stesse specifiche si traducono nel seguente modo, per la funzione in catena aperta $F(s)$:

1. $F(s)$ ha un polo in $s=0$;
2. $\tilde{e} = \frac{k_d^2}{k_f} \leq 0.01$
3. margine di fase $M_\varphi \geq 42^\circ$
4. pulsazione di attraversamento $\omega_T = 10 \text{ [rad/s]}$

Per saper se la funzione presa in esame rispetta tali specifiche si fa uso di specifici grafici, in particolare:

> **Diagramma di Bode:**

una rappresentazione grafica della risposta in frequenza di un sistema lineare tempo-invariante e che consiste in due grafici che rappresentano rispettivamente l'ampiezza e la fase della funzione complessa di risposta in frequenza.

> **Diagramma di Nyquist:**

una particolare rappresentazione grafica della funzione di trasferimento di un sistema dinamico lineare stazionario.

> **Risposta in frequenza:**

rappresenta la descrizione della sua uscita utilizzando come variabile la frequenza invece che il tempo.

Il metodo della risposta in frequenza è probabilmente meno intuitivo di altri metodi, ma in effetti presenta alcuni vantaggi non indifferenti, specialmente quando si tratta di creare delle funzioni di trasferimento partendo da un sistema reale con le sue caratteristiche. La risposta in frequenza di un sistema può essere studiata in due modi diversi: tramite il diagramma di Bode o il diagramma di Nyquist. Entrambi i diagrammi visualizzano le medesime informazioni e la differenza consiste solo nel modo in cui queste vengono presentate.

Dopo aver ricavato i vari diagrammi, per la funzione di trasferimento del processo $P(s)$, attraverso il comando $sisotool(P(s))$, è possibile già fare un'analisi riguardo le specifiche richieste:

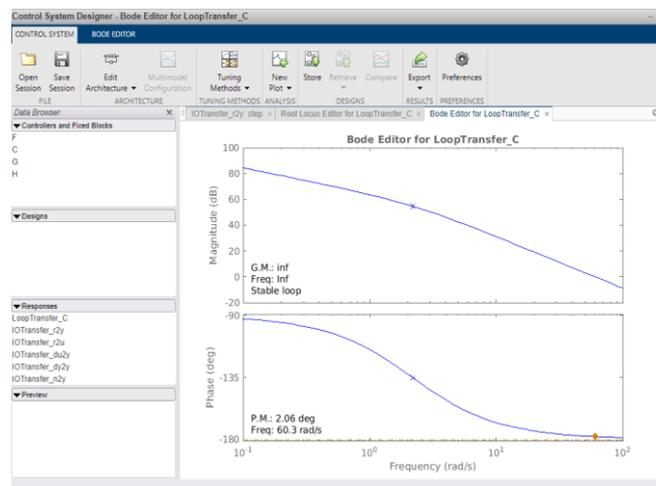


Figura 60: diagramma di Bode di $P(s)$

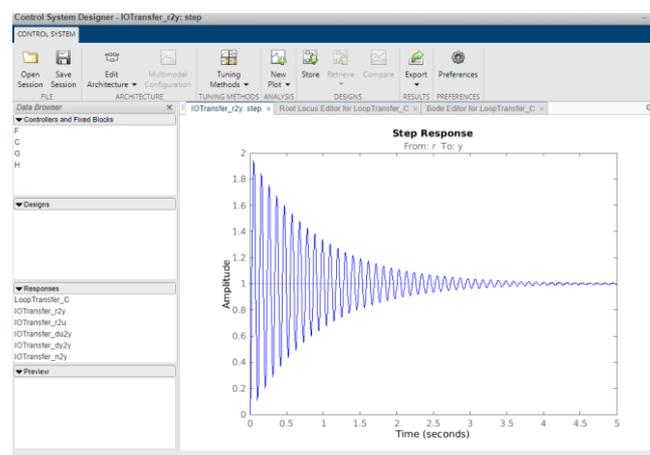


Figura 61: diagramma della risposta in frequenza di $P(s)$

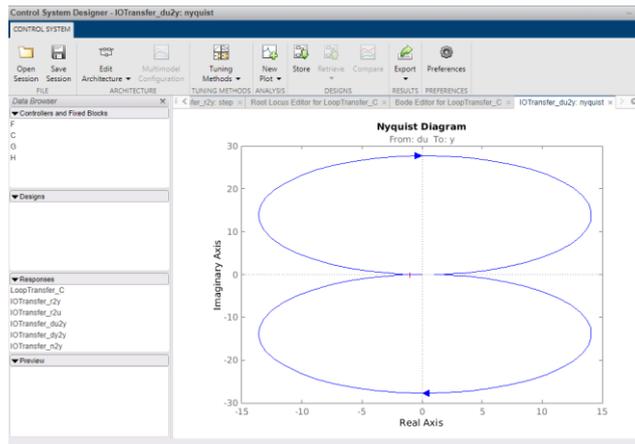


Figura 62: diagramma di Nyquist di $P(s)$

Non vengono soddisfatte le specifiche, tale risposta non è accettabile, quindi avviamoci nella progettazione del controllore di primo tentativo.

Innanzitutto, bisogna ricavarsi il guadagno statico dall'errore di regime, ricordando che il controllo di primo tentativo sarà nella seguente forma:

$$\hat{G} = \frac{k_G}{s^h}$$

Attraverso i seguenti passaggi, si ricava K_g :

$$\tilde{e} = \frac{k_d^2}{k_f} \rightarrow \text{retroazione unitaria } \tilde{e} = \frac{1}{k_f} = \frac{1}{k_g k_p} \rightarrow k_g = \frac{1}{k_f \cdot \tilde{e}}$$

$$K_g = \frac{1}{0.01 \cdot 3643} = 0.0274$$

A questo punto la nostra funzione di trasferimento $F(s)$ sarà composta dalla funzione di trasferimento del processo $P(s)$ moltiplicata alla funzione di controllo $G(s)$.

Inserendo il tutto, otteniamo la nostra funzione frazionaria:

```
>> s=tf('s');
Gs=tf(0.0274);
>> Fs = Gs * Ps |
```

$$F_s = \frac{99.81}{s^2 + 2.175 s}$$

Figura 63: funzione di trasferimento $F(s) = G(s)*P(s)$

Analizziamo i vari diagrammi, visti nel paragrafo precedente, per la nuova funzione $F(s)$.

Sisotool ($F(s)$):

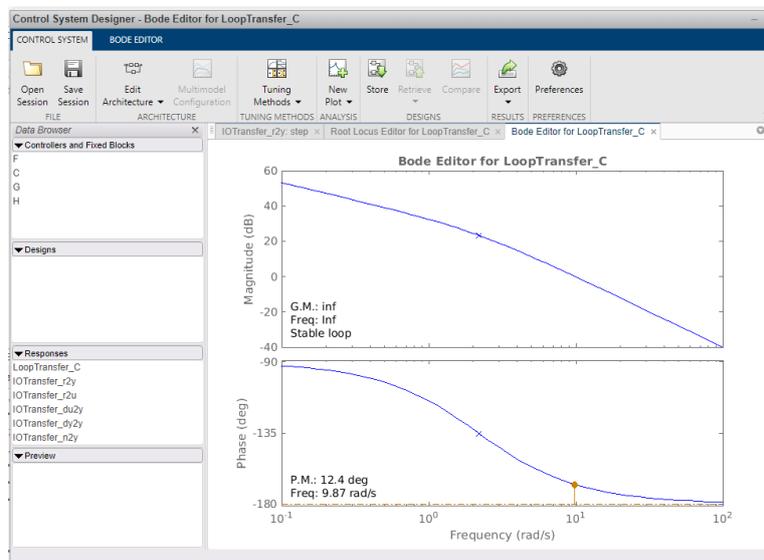


Figura 64: diagramma di Bode di $F(s)$

Si nota fin da subito che la pulsazione di attraversamento risulta rispettata mentre il margine di fase deve essere corretto.

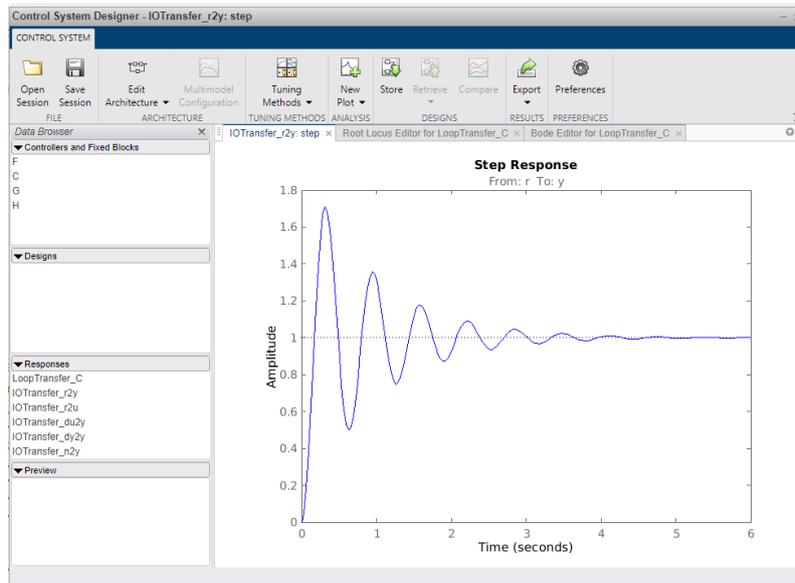


Figura 65: diagramma della risposta in frequenza di $F(s)$

A fronte di un tempo di salita accettabile la sovra elongazione risulta molto elevata, ciò comporta il Tempo di assestamento risulti molto elevato.

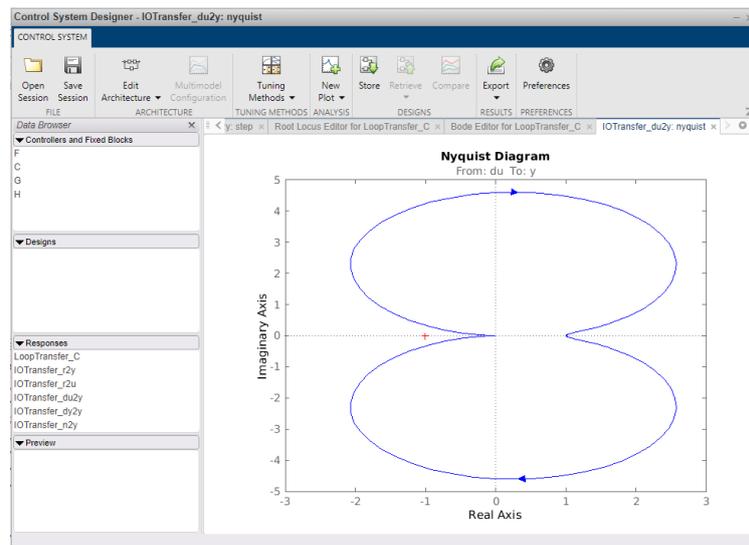


Figura 66: diagramma di Nyquist di $F(s)$

È possibile modificare il controllore $G(s)$ attraverso le funzioni compensatrici $R_a(s)$ e $R_i(s)$:

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

Si andrà ad utilizzare inizialmente la funzione anticipatrice, dato che lo scopo è quello di alzare la fase e lasciare il modulo inalterato.

Scegliendo un $m_a = 10$ e $\tau_a = 0.08$, si ottiene la seguente funzione:

```
>> Fant=(1+0.08*s)/(1+0.008*s)
```

```
Fant =
```

$$\frac{0.08 s + 1}{0.008 s + 1}$$

Figura 67: funzione anticipatrice

La funzione di trasferimento $F(s)$, a questo punto, sarà composta da tre funzioni: $P(s)$, $G(s)$ e $Fant$.

```
>> Fs = Gs * Fant * Ps
```

```
Fs =
```

$$\frac{7.985 s + 99.81}{0.008 s^3 + 1.017 s^2 + 2.175 s}$$

Figura 68: funzione di trasferimento $F(s) = G(s)*Fant*P(s)$

Andando ad analizzare i soliti diagrammi, si nota un grosso miglioramento:

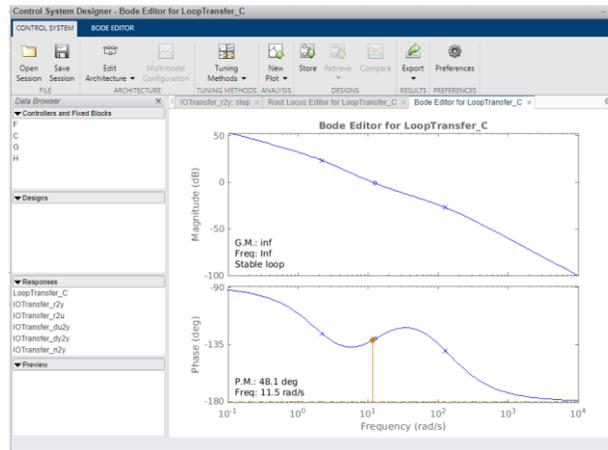


Figura 69: diagramma di Bode di $F(s)$

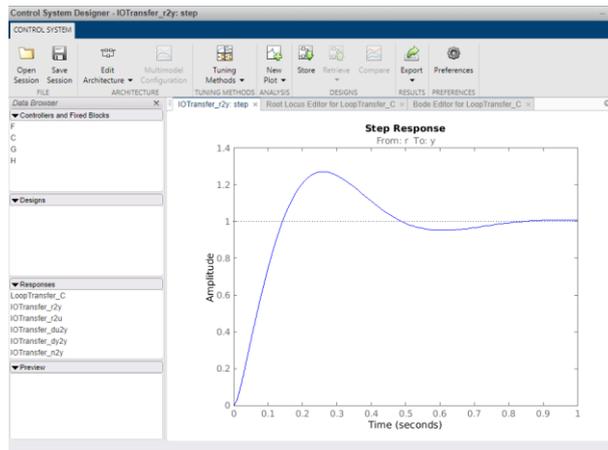


Figura 70: diagramma della risposta in frequenza di $F(s)$

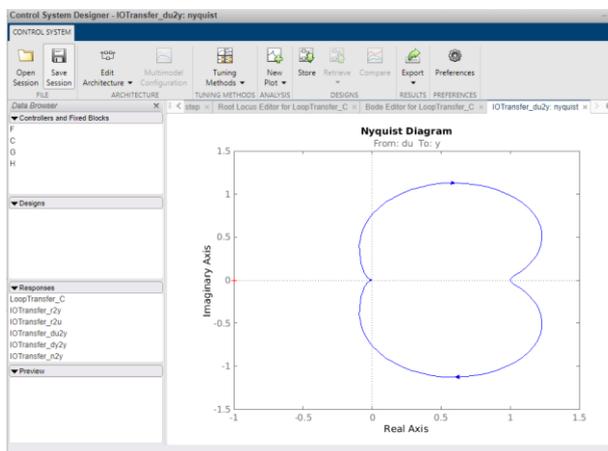


Figura 71: diagramma di Nyquist di $F(s)$

Riesce ad effettuare l'azione di controllo rispetto a segnali di riferimento costanti, ma non riesce ad effettuare l'azione di asservimento rispetto ai segnali di riferimento variabili nel tempo.

Per poter vedere il comportamento di tutte e sei i gradi di libertà, sotto il controllo del controllore progettato, bisogna andarlo ad inserire nel blocco di controllo relativo all'angolo di rollio.

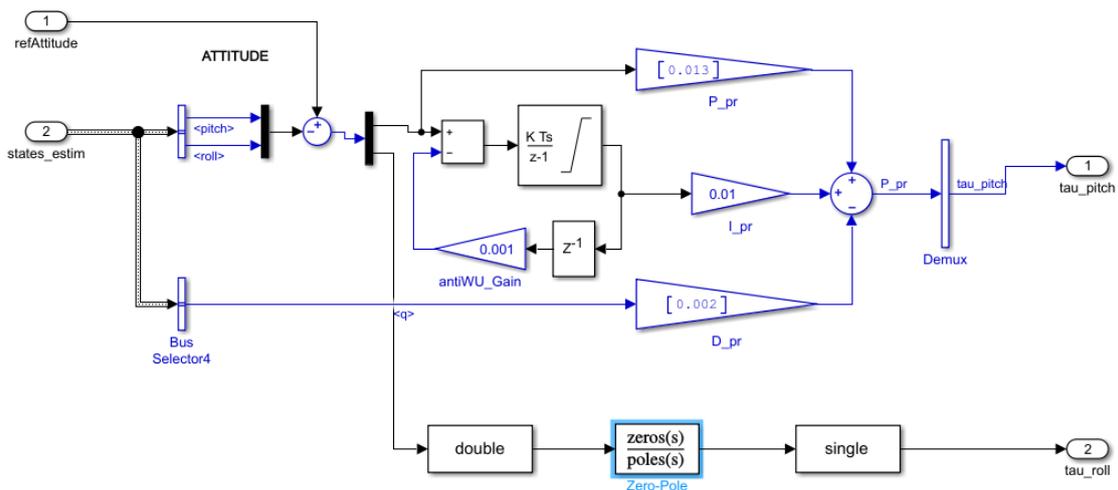


Figura 72: blocco di controllo Attitude modificato con il controllore

Come è stato detto nel capitolo precedente, l'angolo di pitch e di roll sono controllati da un unico controllo, perciò bisogna inizialmente dividerli attraverso un **demultiplexer**. L'angolo di pitch resterà controllato dal controllo precedente mentre l'angolo di roll verrà controllato dal nuovo controllo progettato. Attraverso il blocco Zero-Pole, sarà possibile inserire la funzione di trasferimento del controllore in forma frazionaria.

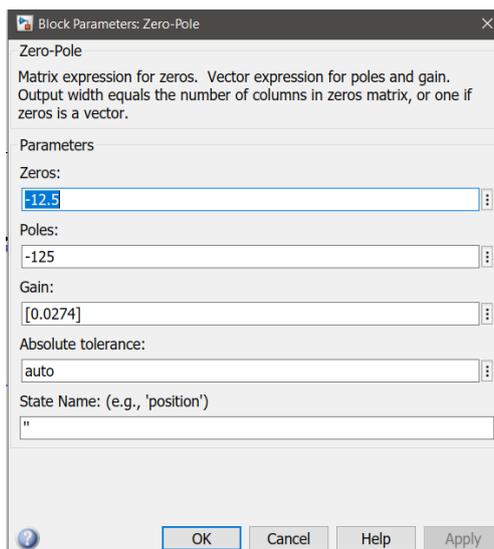


Figura 73: inserimento della funzione di trasferimento

Inserendo due **blocchi cast** per poter invertire il tipo di dato da single a double e viceversa, prima e dopo il **blocco Zero-Pole**.

In questo modo il PID originale continua l'azione di controllo sull'angolo di pitch, mentre il controllore in frequenza si occupa di gestire l'angolo di Roll.

Come è stato detto nella sezione precedente, per evitare un controllo in cascata dato che l'ingresso del riferimento *refAttitude* proviene dal blocco *Switch refAtt* che, sulla base di un segnale di attivazione, sceglie se il riferimento dovrà essere preso dal blocco *XY to reference orientation* oppure dal blocco *Signal Builder*. Modificando appositamente la porta logica che genera il segnale di attivazione per lo *switch* è stato possibile forzare quest'ultimo a scegliere il riferimento costante per l'angolo di roll, escludendo così la possibilità di un'azione di controllo in cascata.

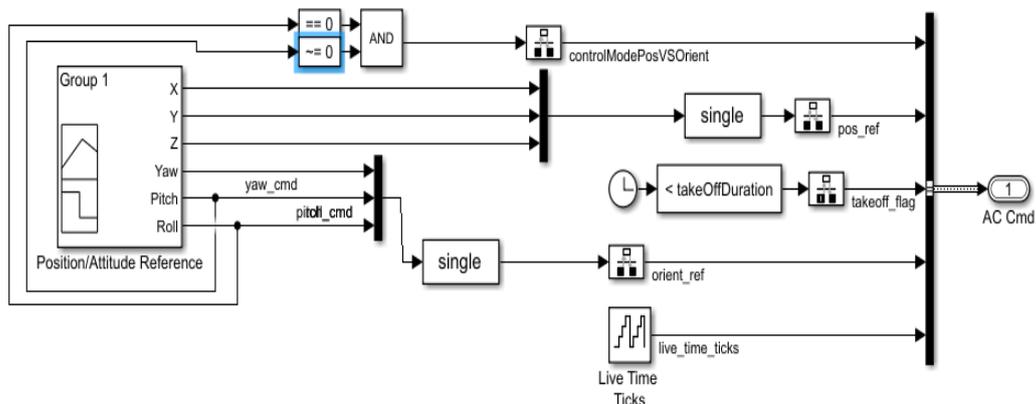


Figura 74: blocco Signal Builder modificato

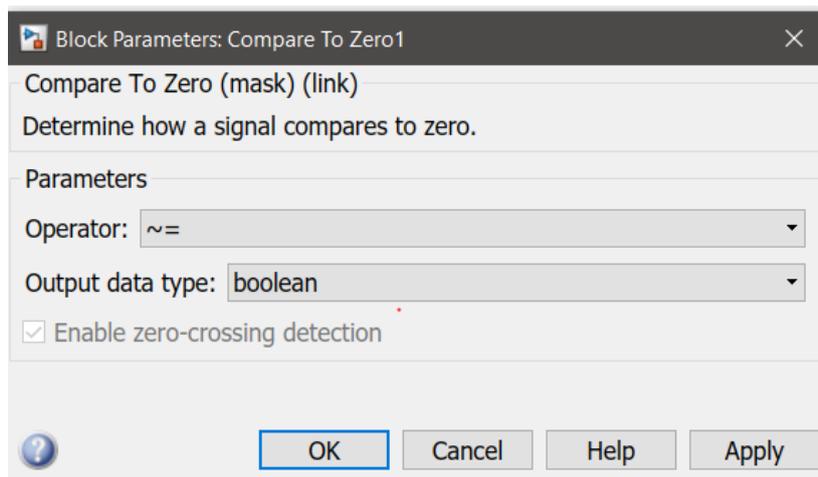


Figura 75: porta logica AND modificata del Signal Builder

Contiene i riferimenti stabili per l'orientamento del *quadricottero* e quindi fornirà un segnale costante.

4.3.1 analisi comportamento modello lineare (controllore I tentativo)

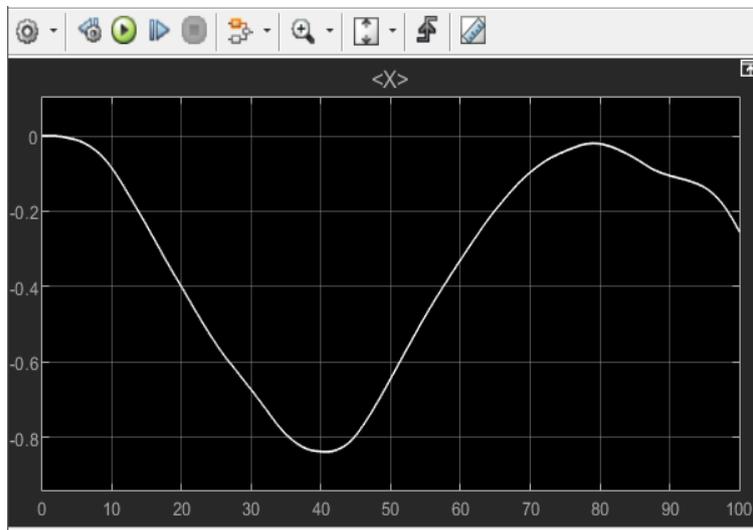


Figura 76: grafico X sotto controllo del controllore di I tentativo nel modello lineare

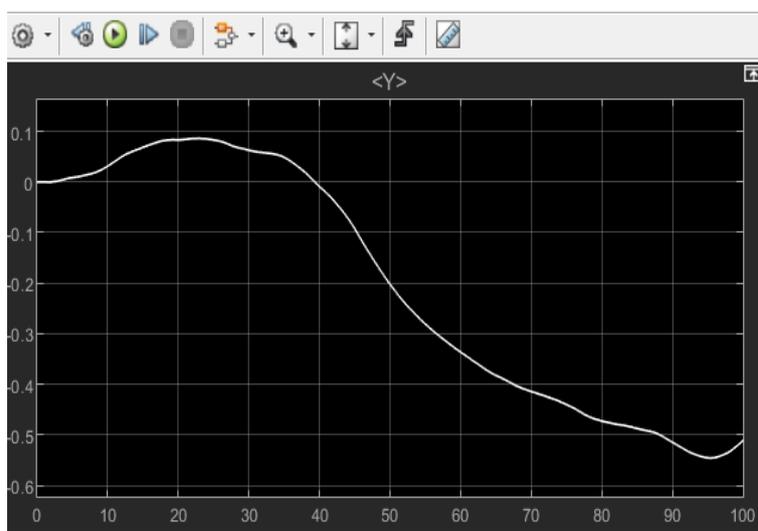


Figura 77: grafico Y sotto controllo del controllore di I tentativo nel modello lineare

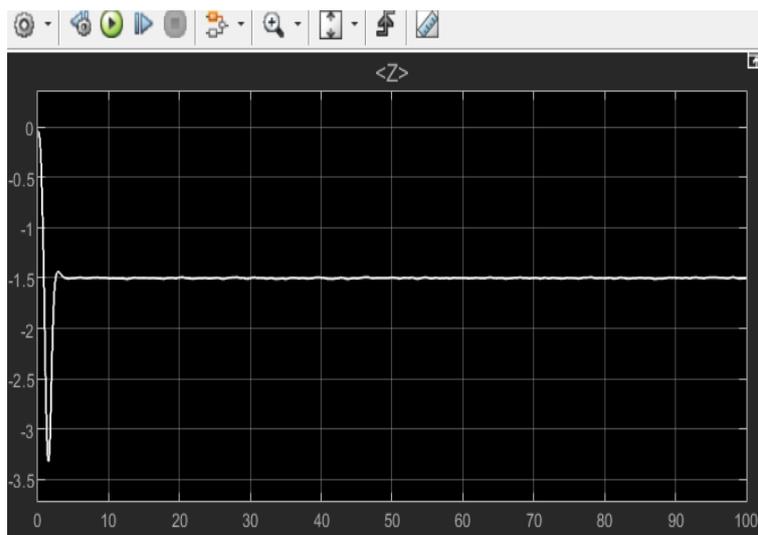


Figura 78: grafico Z sotto controllo del controllore di I tentativo nel modello lineare

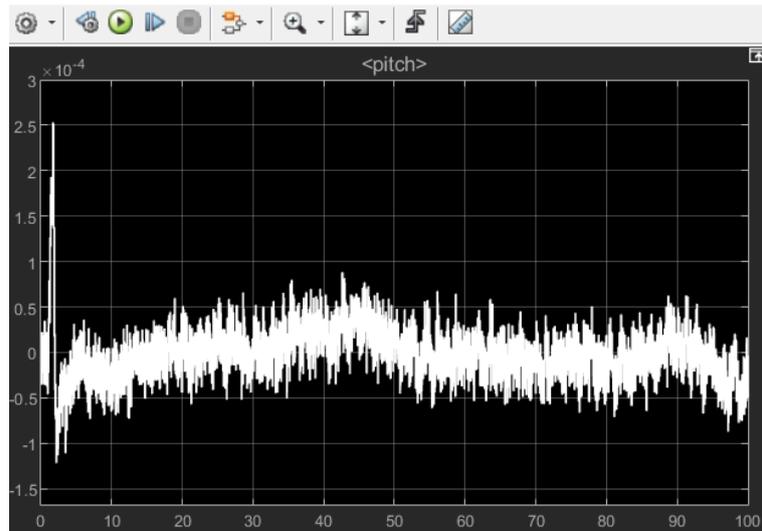


Figura 79: grafico Pitch sotto controllo del controllore di I tentativo nel modello lineare

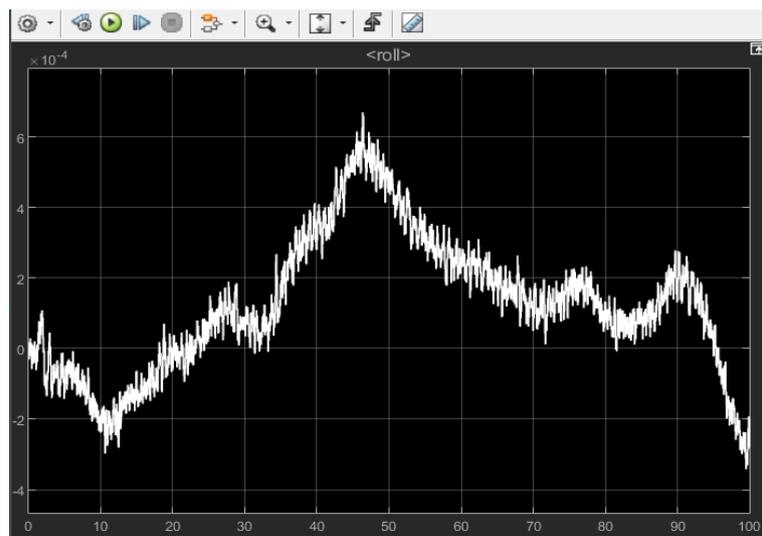


Figura 80: grafico Roll sotto controllo del controllore di I tentativo nel modello lineare

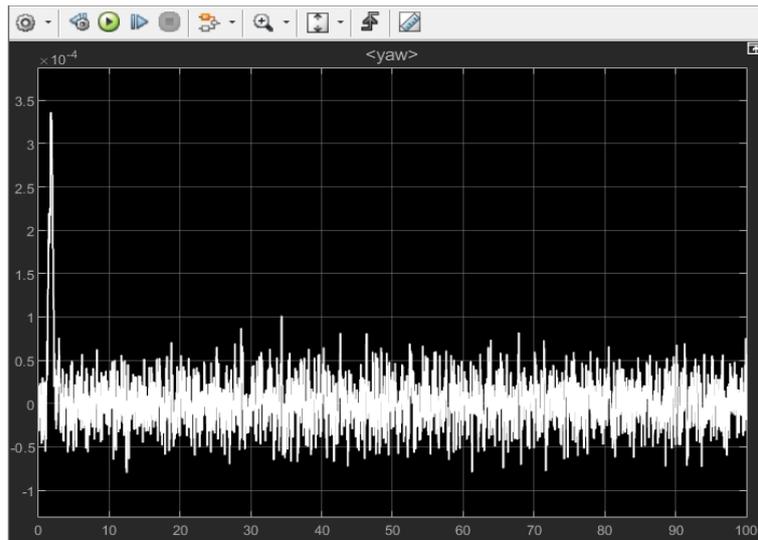


Figura 81: grafico Yaw sotto controllo del controllore di I tentativo nel modello lineare

Si nota fin da subito dai grafici riportati che la posizione X-Y del drone non risulta più controllata.

4.3.2 analisi comportamento modello non lineare (controllore I tentativo)

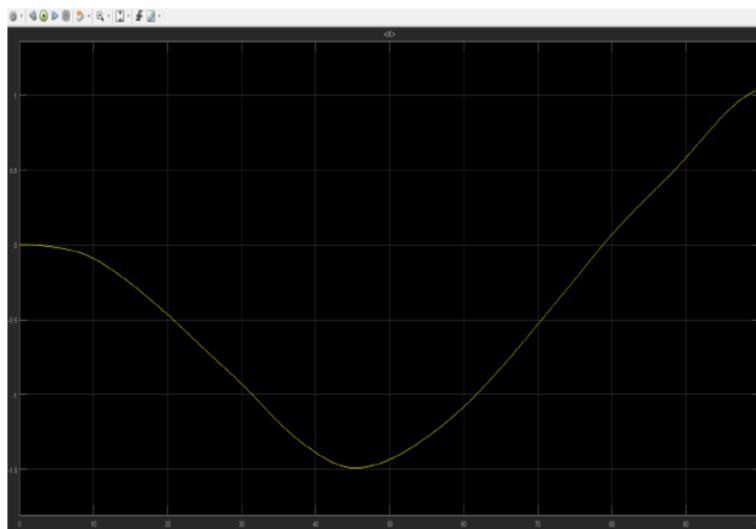


Figura 82: grafico X sotto controllo del controllore di I tentativo nel modello non lineare

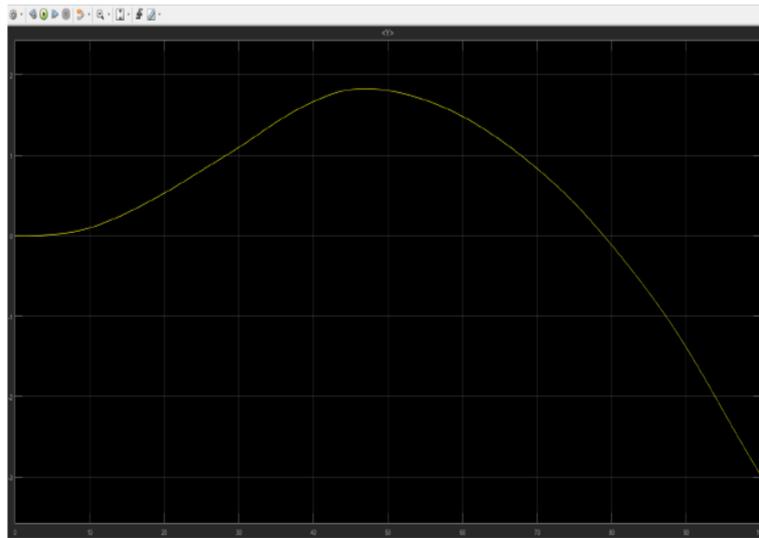


Figura 83: grafico Y sotto controllo del controllore di I tentativo nel modello non lineare

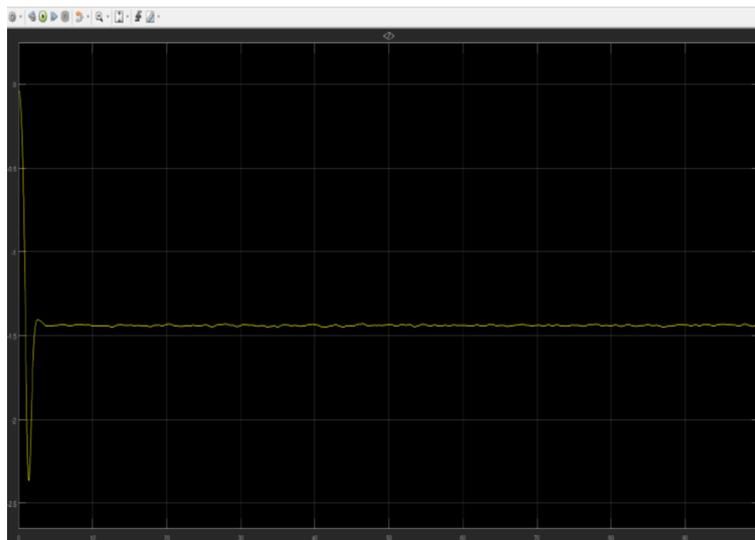


Figura 84: grafico Z sotto controllo del controllore di I tentativo nel modello non lineare

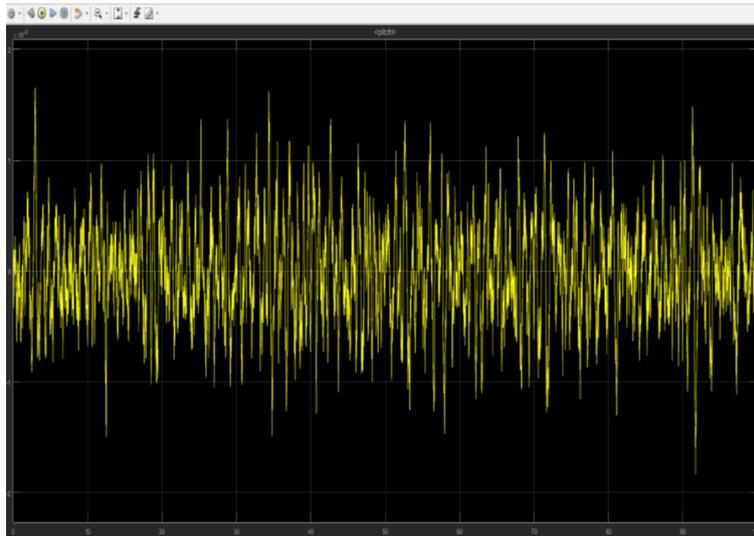


Figura 85: grafico Pitch sotto controllo del controllore di I tentativo nel modello non lineare

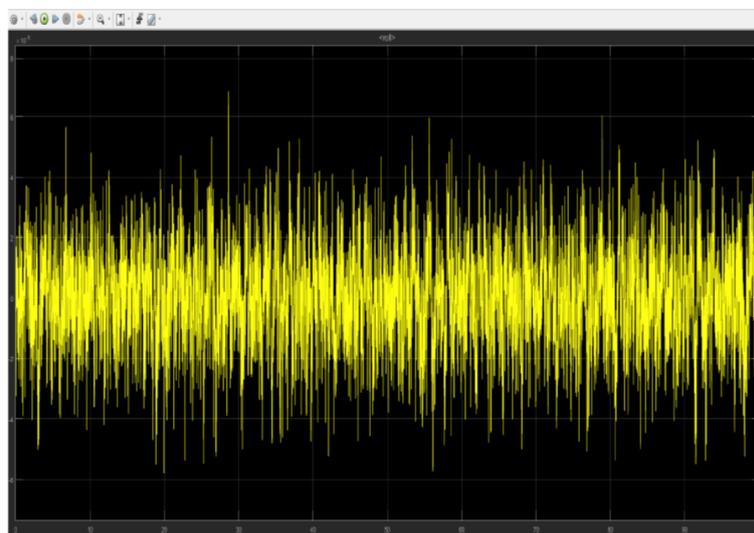


Figura 86: grafico Roll sotto controllo del controllore di I tentativo nel modello non lineare

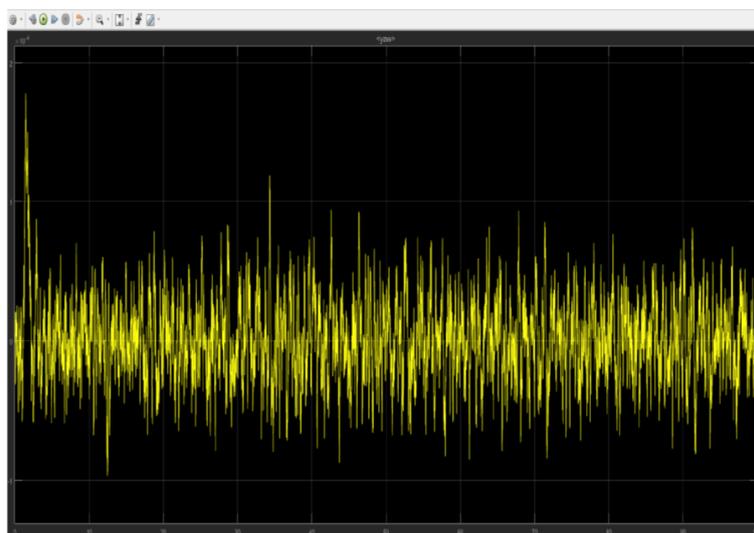


Figura 87: grafico Yaw sotto controllo del controllore di I tentativo nel modello non lineare

Stesse considerazioni fatte per il caso lineare sono accettabili per il caso non lineare, quindi bisogna procedere con la progettazione di un altro controllore.

4.4 controllore di secondo tentativo

Per la modellazione del controllore di secondo tentativo, si può partire dal controllore creato nella sezione precedente, andandolo a modificare attraverso le funzioni compensatrici.

Quindi si va a modificare il modulo e la fase attraverso una funzione anticipatrice e poi attraverso la funzione attenuatrice:

> Funzione Anticipatrice

```
>> Fant=(1+0.2*s)/(1+0.004*s);
>> G1=Gs*Fant;
>> Fs = G1 * Ps

Fs =

          19.96 s + 99.81
-----
0.004 s^3 + 1.009 s^2 + 2.175 s
```

Figura 88: funzione anticipatrice per controllo di secondo tentativo

Analizzando la funzione $F(s)$ ottenuta, si ottengono i seguenti grafici:

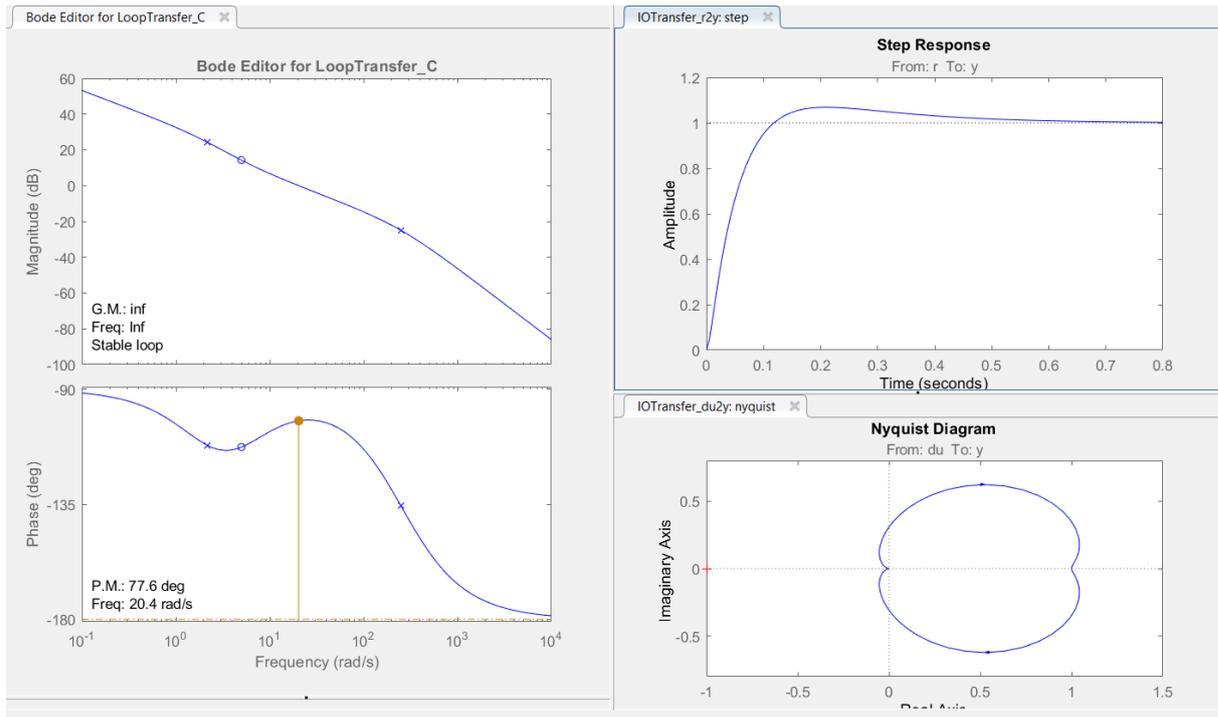


Figura 89: diagrammi $F(s)$ secondo tentativo con F_{att}

Il controllore implementato deve essere più "pronto" a rispondere alle sollecitazioni di quello sviluppato nella sezione precedente.

> Funzione Attenuatrice

```
>> Fatt=(1+5*s)/(1+10*s);
```

```
G2=G1*Fatt;
```

```
Fs = G2 * Ps
```

```
Fs =
```

$$99.81 s^2 + 519 s + 99.81$$

$$0.04 s^4 + 10.09 s^3 + 22.76 s^2 + 2.175 s$$

Figura 90: funzione attenuatrice per controllo di secondo tentativo

I grafici ottenuti sono i seguenti:

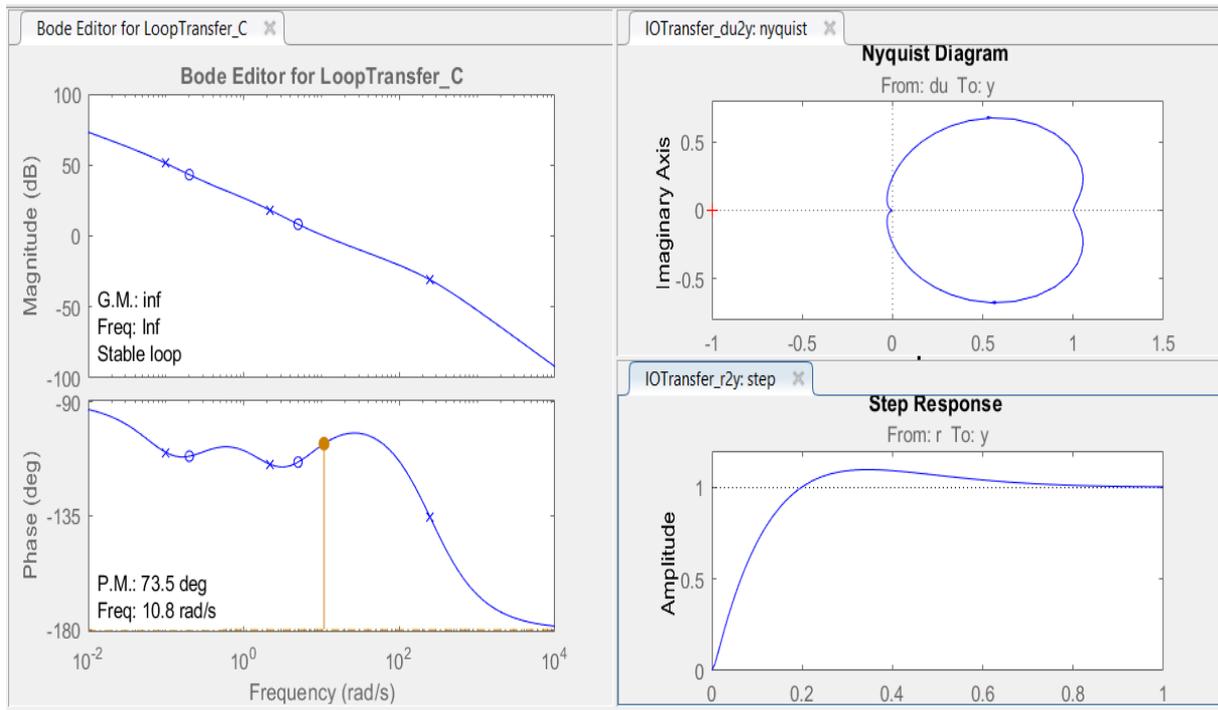


Figura 91: diagrammi $F(s)$ di secondo tentativo con Fatt

Dai grafici ottenuti si raggiunge alle specifiche desiderate, sia al livello di modulo che di fase.

A questo punto il procedimento è lo stesso visto in precedenza, bisogna andare ad aggiungere il nostro controllore nel blocco di controllo relativo all'angolo di roll.

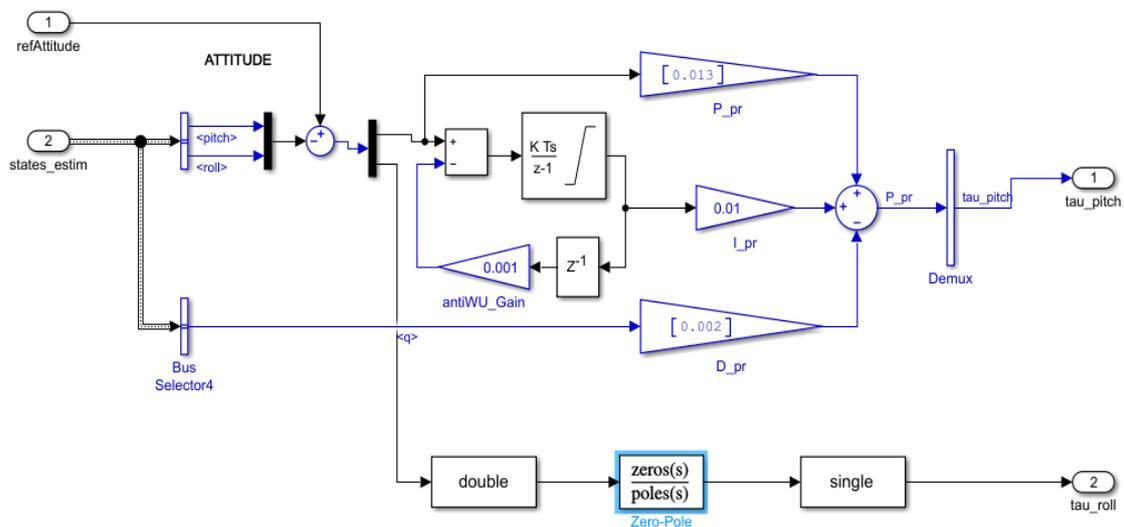


Figura 92: blocco di controllo Attitude modificato con il controllore di secondo tentativo

Inserimento della funzione di trasferimento:

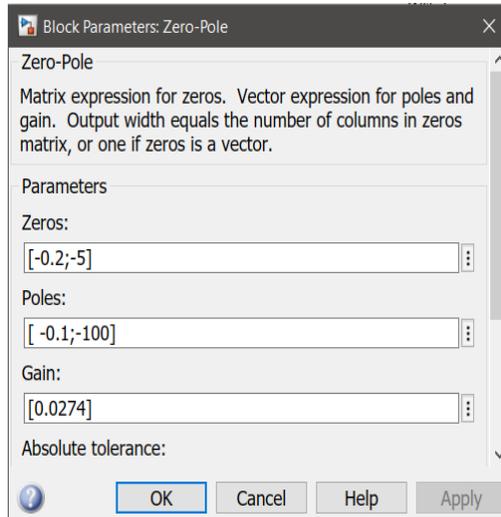


Figura 93: inserimento funzione di trasferimento

4.4.1 analisi comportamento modello lineare (controllore II tentativo)

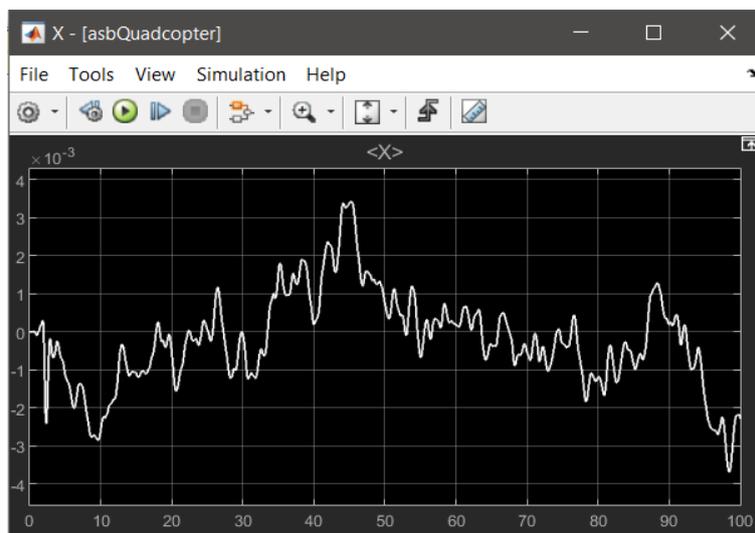


Figura 94: grafico X sotto controllo del controllore di II tentativo nel modello lineare

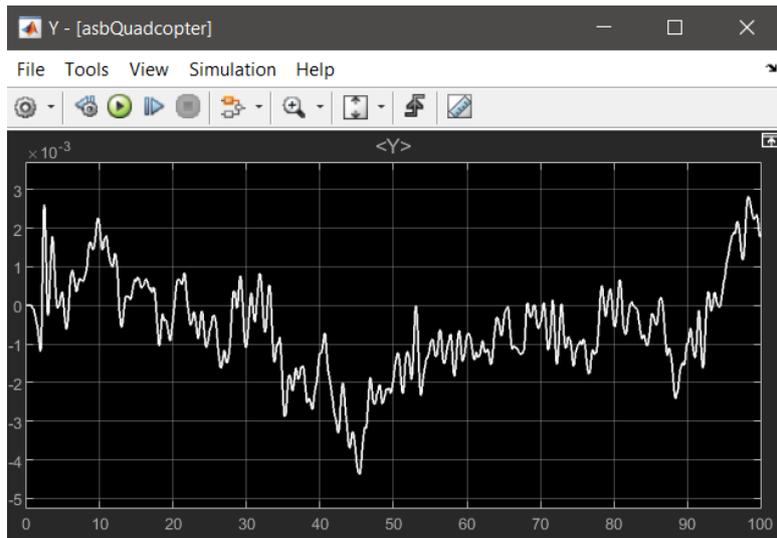


Figura 95: grafico Y sotto controllo del controllore di II tentativo nel modello lineare

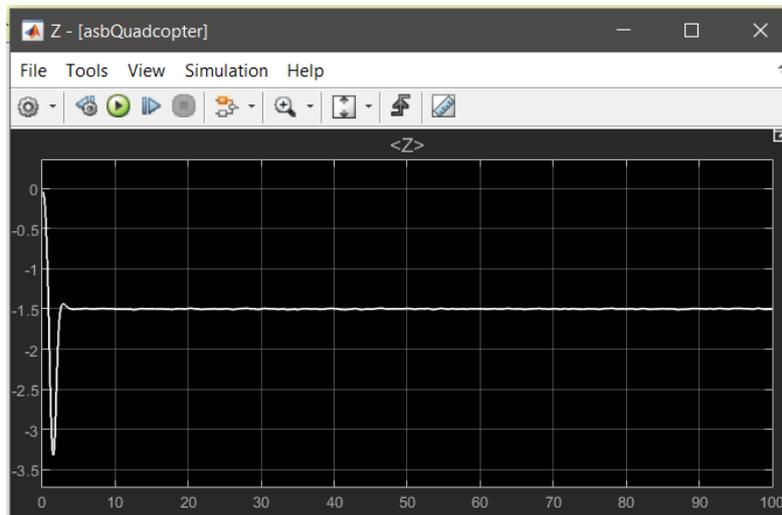


Figura 96: grafico Z sotto controllo del controllore di II tentativo nel modello lineare

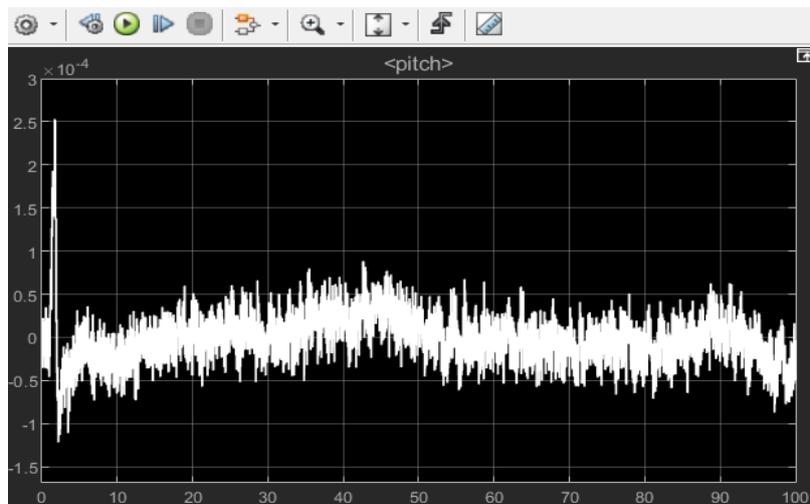


Figura 97: grafico Pitch sotto controllo del controllore di II tentativo nel modello lineare

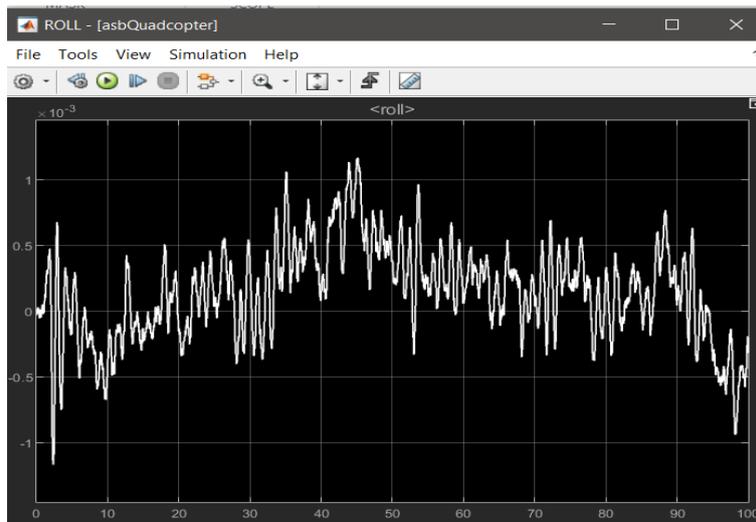


Figura 98: grafico Roll sotto controllo del controllore di II tentativo nel modello lineare

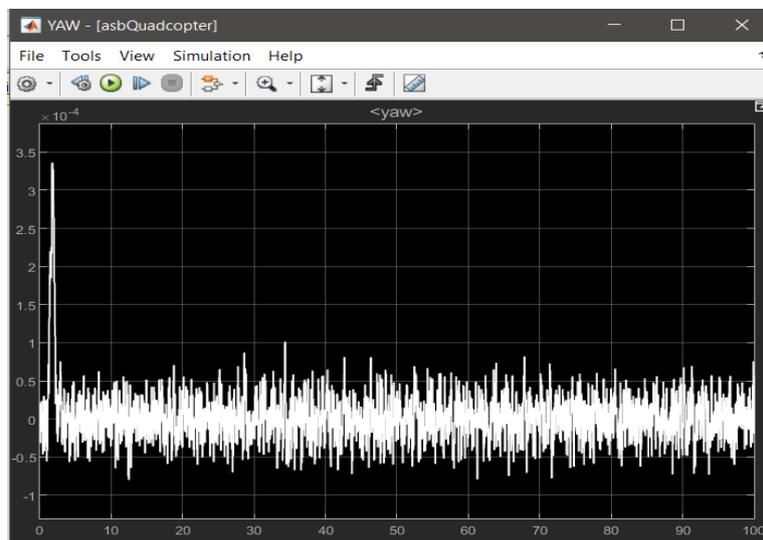


Figura 99: grafico Yaw sotto controllo del controllore di II tentativo nel modello lineare

4.4.2 analisi comportamento modello non lineare (controllore II tentativo)

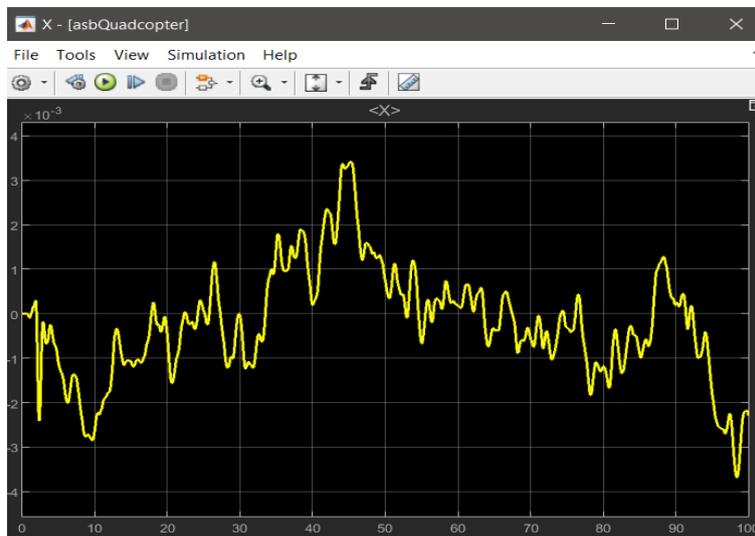


Figura 100: grafico X sotto controllo del controllore di II tentativo nel modello non lineare

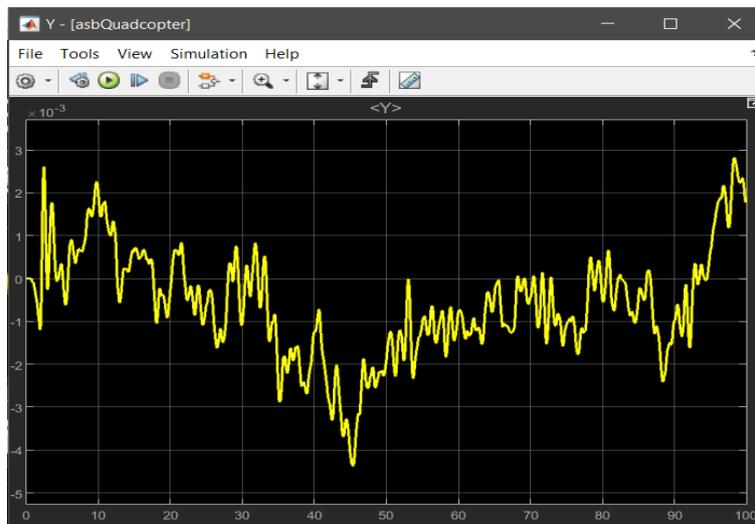


Figura 101: grafico Y sotto controllo del controllore di II tentativo nel modello non lineare

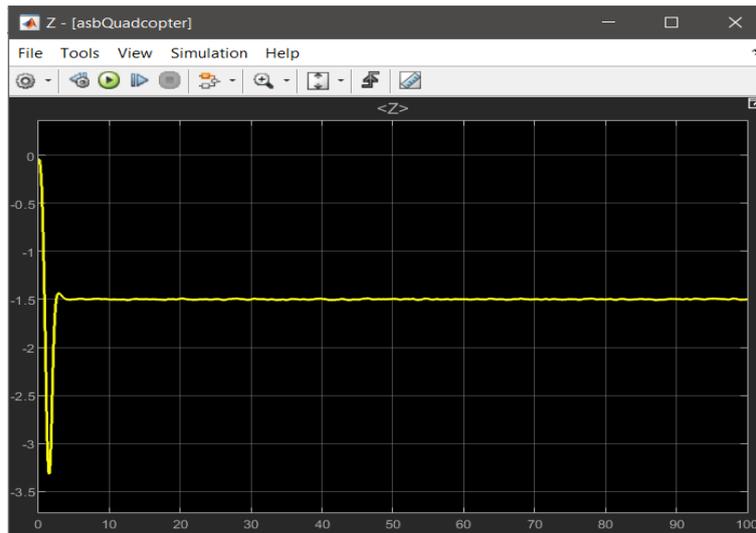


Figura 102: grafico Z sotto controllo del controllore di II tentativo nel modello non lineare

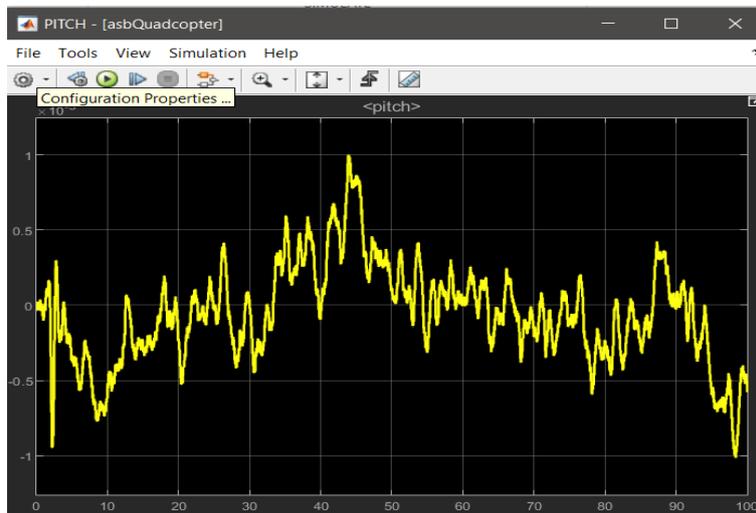


Figura 103: grafico Pitch sotto controllo del controllore di II tentativo nel modello non lineare

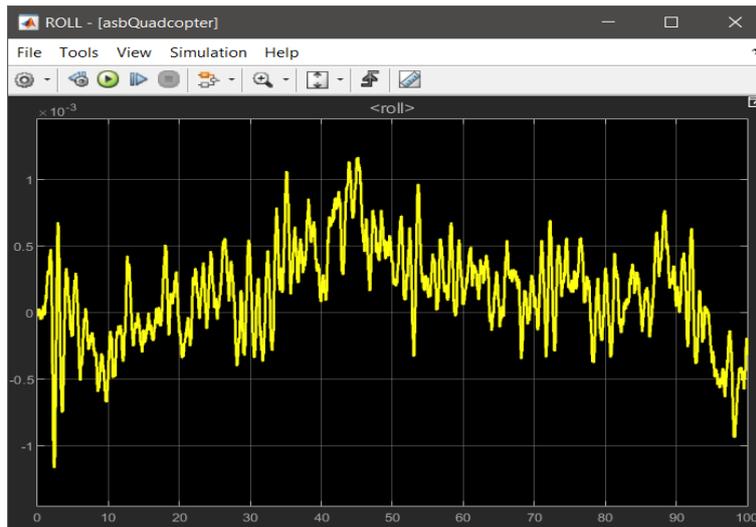


Figura 104: grafico Roll sotto controllo del controllore di II tentativo nel modello non lineare

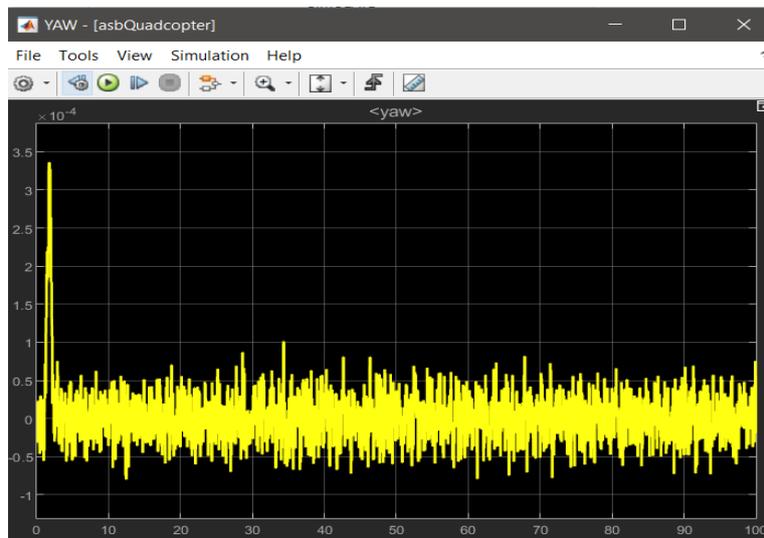


Figura 105: grafico Yaw sotto controllo del controllore di II tentativo nel modello non lineare

Capitolo 5

Conclusione

Il controllore relativo all'angolo di roll sviluppato attraverso la sintesi in frequenza ed il controllore originario dell'angolo di roll relativo al PID, risultano avere stessi comportamenti, in tutti e due i casi le oscillazioni si trovano attorno al valore di riferimento nell'ordine di 10^{-3} [m]. In tutti e due i casi, i comportamenti risultano analoghi sia nel modello lineare che nel modello non lineare.

Avviando la simulazione attraverso il comando *StartSimulation*, è possibile visualizzare a schermo una simulazione in 3D del veicolo, lo stesso veicolo con il controllore progettato. Questa simulazione serve principalmente come risultato finale per la prova di volo, risultato superato con successo.

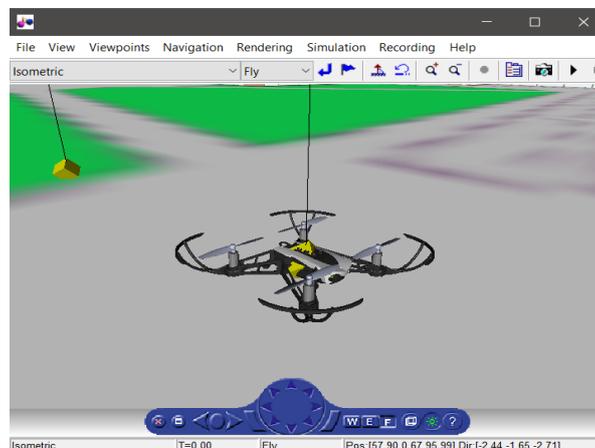


Figura 106: StartSimulation

La prima simulazione visualizzerà il veicolo in equilibrio senza muoversi dato che per default le posizioni sono settate al valore di zero (visto in precedenza nel blocco del *Signal Builder*).

Andando a modificare la posizione X e quella di Y, mettendola circa a 0.2, nel blocco *Group* presente nel blocco principale del *Signal Builder* sarà possibile visualizzare il veicolo in movimento:

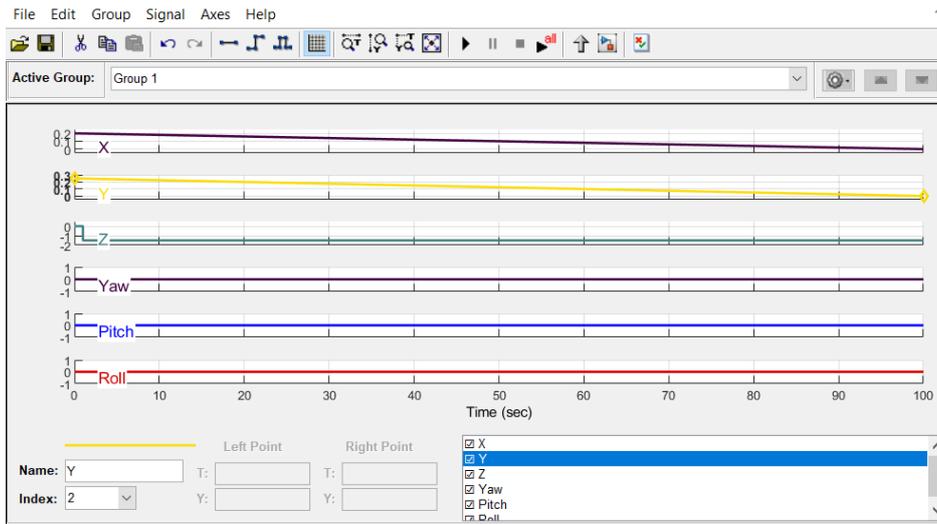


Figura 107: blocco Group: posizioni iniziali X e Y modificate

StartSimulation:

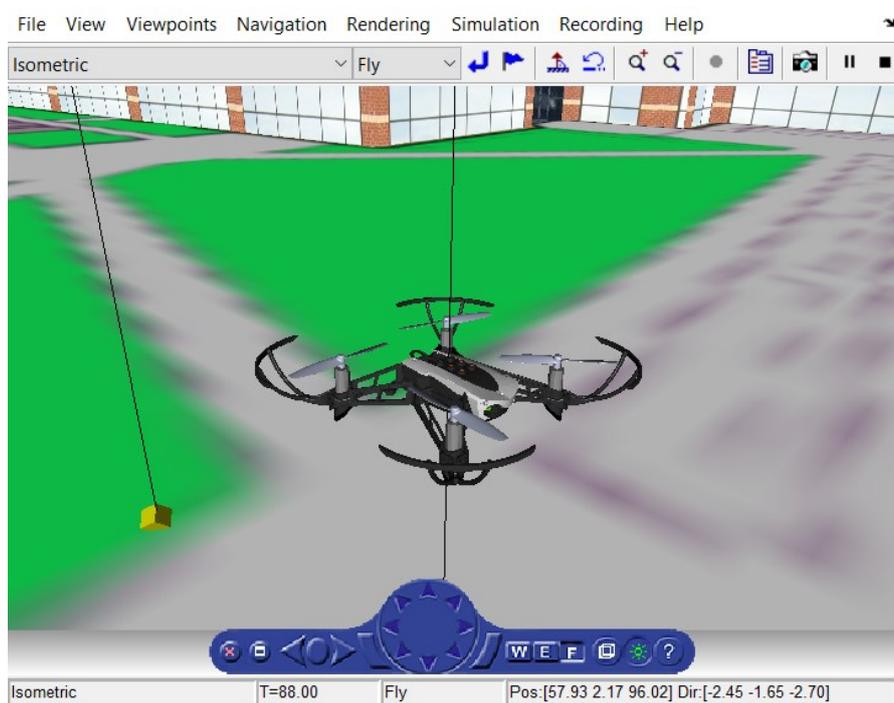


Figura 108: StartSimulation

Ringraziamenti

Vorrei dedicare questo spazio a chi, con dedizione e pazienza, ha contribuito alla realizzazione di questo elaborato.

Innanzitutto, ringrazio il mio relatore prof. Ippoliti Gianluca e il mio correlatore prof. Orlando Giuseppe, sempre pronti a darmi le giuste indicazioni in ogni fase della realizzazione dell'elaborato.

Ringrazio infinitamente mia madre e mio padre, senza i loro insegnamenti e senza il loro supporto, questo lavoro di tesi non esisterebbe nemmeno.

Grazie a tutti i miei colleghi di corso, per avermi sempre incoraggiato fin dall'inizio del percorso universitario.

Bibliografia

1. S. Bouabdallah Design and Control of Quadrotors with Application to Autonomous Flying, Lausanne, 2007.
2. Isidori A. (1993), Sistemi di controllo, Siderea, pp 189-238
3. W.Z.Fum, Implementation of Simulink Controller Design on Iris+ Quadcopter, Monterey, CA, 2015.
4. Pounds, Mahony, Corke, (21 February 2010), Modelling and control of a large quadrotor robot, Control Engineering Practice.
5. MATLAB, <https://es.mathworks.com/help/aeroblks/quadcopterproject.html>.
6. G. Celentano e L. Celentano. Fondamenti di Dinamica dei sistemi, Vol. II. Edises, Campania, Napoli, 1 edition, 2010.
7. MATLAB, <https://it.mathworks.com/help/supportpkg/parrot/ref/color-detection-and-landing-parrot-example.html>