



UNIVERSITA' POLITECNICA DELLE MARCHE
FACOLTA' DI INGEGNERIA

Corso di Laurea triennale "INGEGNERIA INFORMATICA E
DELL'AUTOMAZIONE"

**Studio e sviluppo di tecniche di controllo per minidroni in
ambiente Simulink**

**Study and development of control technique for minidrones in Simulink
environment**

Relatore:
Prof. Ippoliti Gianluca

Tesi di Laurea di:
Di Santo Davide

Correlatore:
Prof. Orlando Giuseppe

A.A. 2020 /2021

Sommario

Il seguente elaborato ha come obiettivo quello di fornire un'analisi del modello Simulink del minidrone Mambo Parrot S.A. e di progettare due controllori tramite sintesi in frequenza e sintesi con Luogo delle radici per il controllo di uno dei suoi sei gradi di libertà (roll), esaminandone i risultati sperimentali e confrontandoli con i controllori PID originali.

In prima battuta saranno analizzate le caratteristiche tecniche e la dinamica del sistema attraverso le relazioni fisiche del quadricottero, dalle quali si ricaverà il modello matematico.

In seguito, verranno descritti i blocchi presenti nel pacchetto Simulink e le loro funzionalità.

Successivamente verrà implementato il modello matematico nell'ambiente Simulink all'interno del *Simulink Support Package for Parrot Minidrones*.

Infine, si procederà con lo sviluppo e l'implementazione in Simulink di controllori alternativi rispetto a quelli originali in modo tale da migliorare le prestazioni del minidrone rientrando nei parametri stabiliti.

Indice

1. Introduzione.....	5
2. Caratteristiche tecniche del minidrone Mambo.....	6
3. Modello fisico, dinamica del veicolo e modellazione matematica.....	8
3.1 Gradi di libertà e controllo.....	8
3.2 Thrust, Pitch, Roll, Yaw.....	9
3.3 Sistemi di riferimento.....	10
3.4 Matrici di rotazione.....	11
3.5 Dinamica rotazionale.....	13
3.6 Momenti esterni.....	15
3.7 Forze agenti sul sistema.....	16
3.8 Equazioni generali del sistema.....	16
4. Analisi del modello in Simulink.....	18
4.1 Signal Builder.....	19
4.2 Flight Control System.....	21
4.3 Controllo dei processi.....	24
4.3.1 <i>Non linear Airframe.....</i>	<i>24</i>
4.3.2 <i>Linear Airframe.....</i>	<i>26</i>
5. Sintesi dei controllori.....	26
5.1 Riferimento costante e variabile.....	27
5.2 PID originali.....	28
5.3 Linearizzazione del modello non lineare.....	32
5.4 Tecniche di controllo lineare.....	38
5.4.1 <i>Sintesi in frequenza.....</i>	<i>38</i>
5.4.2 <i>Sintesi con luogo delle radici.....</i>	<i>40</i>
5.5 Sintesi dei controllori in frequenza.....	43
5.5.1 <i>Controllore per riferimenti costanti.....</i>	<i>43</i>

5.5.2	<i>Controllore per riferimenti variabili.....</i>	49
5.6	Sintesi tramite luogo delle radici.....	53
5.6.1	<i>Controllore di primo tentativo.....</i>	53
5.6.2	<i>Controllore di secondo tentativo.....</i>	59
6.	Conclusioni.....	66
7.	Bibliografia.....	67

1. Introduzione

Nella prima parte dell'elaborato ci sarà una breve descrizione sui droni rivolta soprattutto alla loro struttura costitutiva e ai fenomeni fisici che ne determinano la dinamica di volo. Successivamente verrà trattata la modellazione matematica che tiene conto degli aspetti fisici e strutturali. Il modello utilizzato per lo sviluppo dei controlli automatici è il *Parrot Mambo minidrone*.

Nella seconda parte verranno analizzati nel dettaglio i blocchi logici che compongono il progetto *absQuadcopter* presente nell'ambiente *Simulink*. Vengono trattati nello specifico i blocchi principali e responsabili del controllo dei gradi di libertà del drone.

Nella terza ed ultima parte ci sarà la progettazione, tramite tecniche di controllo lineare come sintesi in frequenza e sintesi mediante luogo delle radici, di controllori per il controllo dell'angolo di roll. Si analizzeranno le problematiche e le relative soluzioni insorte durante le simulazioni di volo in ambiente *Simulink*.

Infine, verranno tratte le conclusioni e paragonati i comportamenti tra i controllori originali e quelli progettati.

2. Caratteristiche tecniche del minidrone

Mambo

Il minidrone Mambo è un veicolo aeromobile a pilotaggio remoto formato da quattro eliche rotanti azionate da altrettanti motori coreless.

Esso è dotato di una batteria Li-Po da 550 mAh che a fronte di un tempo di carica di 30 minuti garantisce un funzionamento di circa 9 minuti.

Il peso del mezzo (senza accessori o paraurti) è di 63.5 g; le dimensioni sono pari a 13.2x13.2x4.1 cm, le quali aumentano a 18x18x4.1 cm con il montaggio dei paraurti.



Fig.1 Parrot Mambo

Il minidrone Mambo presenta una configurazione “a croce” con due eliche (anteriore destro e posteriore sinistro) che ruotano in senso orario (clock) e due eliche (anteriore sinistro e posteriore destro) che ruotano in senso antiorario (anti-clock).

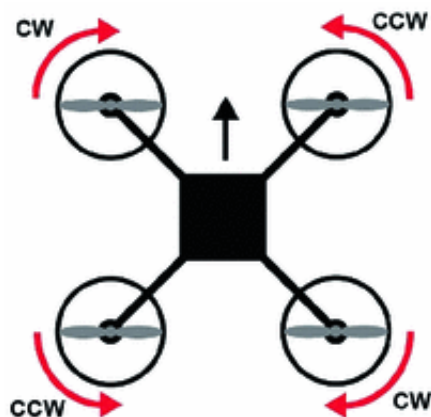


Fig.2 Configurazione “a croce”

L'aeromobile presenta inoltre delle componenti, sia hardware che software, che fungono da base per l'orientamento e per riuscire a muoversi liberamente nello spazio.

Inoltre, esse sono fondamentali per soddisfare tutte le azioni assegnate. Le componenti si distinguono in:

- sensori, per la determinazione della quota verticale del mezzo e più in generale dell'orientamento e posizione del veicolo nello spazio;
- *Inertial Measurement Unit (IMU)*, per la misura dell'accelerazione lineare e della variazione angolare rispetto al sistema di riferimento solidale col veicolo tramite un accelerometro ed un giroscopio;
- camera posta sul lato inferiore (60 FPS) del mezzo, per fornire dati al software "Optical Flow". Tale software sfrutta la posizione relativa degli oggetti tra un frame e l'altro ed è in grado di stimare la velocità orizzontale del minidrone.



Fig.3 Lato inferiore



Fig.4 Lato superiore

È presente anche una porta micro-USB che permette di interfacciare il drone ad un computer.

La gestione del sistema è affidata ad una scheda prestampata PCB (Printed Circuit Board) dotata di un processore ARM A9 a 800 Mhz con sistema operativo Linux.

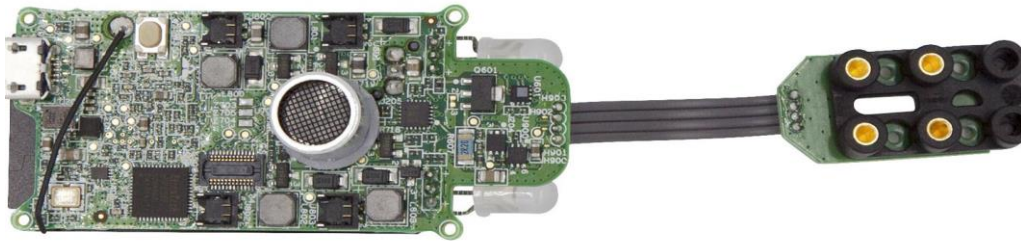


Fig.5 Scheda Madre Mambo

3. Modello fisico e dinamica del veicolo

3.1 Gradi di libertà e controllo

Un minidrone può essere considerato come un sistema fisico dotato di 6 gradi di libertà:

- 3 gradi di libertà traslazionali che favoriscono lo spostamento lineare del mezzo (assi X, Y, Z);
- 3 gradi di libertà rotazionali che favoriscono lo spostamento angolare del mezzo (angoli ϕ, θ, ψ).

Si tratta di un sistema sotto-attuato, ossia un sistema di cui non si possono controllare direttamente tutti e 6 i gradi di libertà.

Per effettuare un qualsiasi spostamento di coordinate dentro lo spazio \mathbb{R}^3 il drone si affida solamente al 'dosaggio' di potenza che verrà fornita a ognuno dei suoi 4 motori.

Esso, se sbilanciato, comporterà la generazione di un momento torcente rispetto ad un asse passante per il centro di massa del drone, dal quale si innescherà una rotazione attorno all'asse stesso. Tramite questo funzionamento, il drone riuscirà a spostarsi lungo i 6 gradi di libertà tramite i suoi 4 motori.

3.2 Thrust, Pitch, Roll, Yaw

I movimenti che tale minidrone è in grado di effettuare spostandosi lungo i 6 gradi di libertà sono:

- spinta verticale (thrust);
- beccheggio (pitch);
- rollio (roll);
- imbardata (yaw).

La spinta verticale è data dall'azione dei quattro motori che, in sincronia, attraverso la rotazione delle eliche, spostano verso il basso una massa di aria che permette di creare una differenza di pressione tale da far innalzare il drone.

Il beccheggio, invece, è dato da spostamenti positivi dell'angolo θ attorno all'asse y dovuti alle diverse velocità di rotazione dei motori anteriori e posteriori.

Per quanto riguarda il rollio, esso è dovuto a spostamenti positivi intorno all'asse x dell'angolo ϕ . È il risultato delle diverse velocità di rotazione dei motori posti sul lato destro e sinistro del quadricottero.

Infine, l'imbardata è un movimento di rotazione attorno all'asse z ottenuto grazie alla differenza di potenza nella coppia di motori presenti nella stessa diagonale.

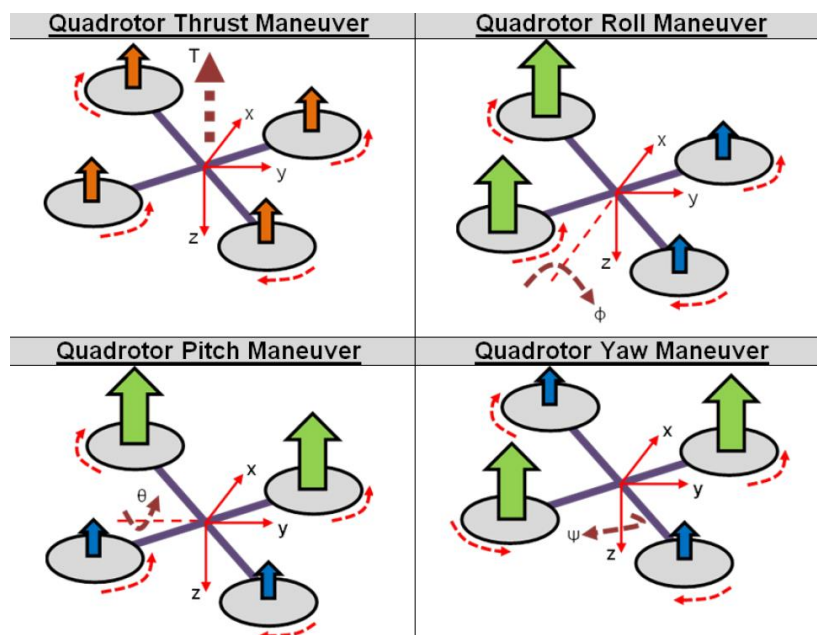


Fig.6 Azione dei motori nei 4 movimenti descritti

3.3 Sistemi di riferimento

Si può passare ora a definire i sistemi di riferimento, di cui è necessario far affidamento, per rappresentare al meglio la dinamica del sistema e comprendere sia i movimenti lineari che quelli angolari del drone.

- sistema di riferimento inerziale o *Earth Fixed Frame*: è il riferimento preso in base al piano tangente alla superficie della Terra con origine la posizione dell'osservatore;
- sistema di riferimento solidale con il centro di gravità del drone (*Body Fixed Frame*).

Assumiamo che centro di gravità (*CoG: Centre of Gravity*) e *Body Fixed Frame* coincidano.

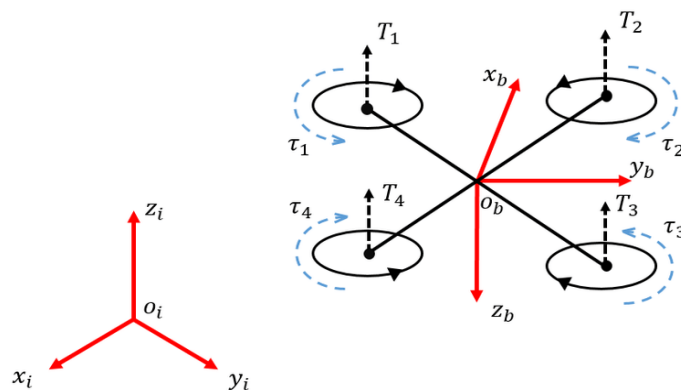


Fig.7 Inertial frame e Body Fixed Frame

Definiamo ora rispetto al sistema di riferimento inerziale $\xi = (x \ y \ z)^T$ il vettore posizione del drone e $\sigma = (\phi \ \theta \ \psi)^T$ come vettore degli angoli che descrivono l'orientamento del *body frame* del quadricottero rispetto all'*inertial frame*.

Analogamente definiamo, rispetto al sistema di riferimento solidale col velivolo, il vettore delle velocità lineari $v = (u \ v \ w)^T$ e il vettore delle velocità angolari $\omega = (p \ q \ r)^T$. Utilizziamo delle matrici di rotazione attorno agli assi per passare da un sistema di riferimento all'altro.

3.4 Matrici di rotazione

Le matrici di rotazione sono particolari matrici che descrivono, fissato un asse, gli spostamenti angolari ottenibili compiendo delle rotazioni rispetto agli altri due assi liberi.

Nel caso in esame vengono utilizzate per ottenere lo spostamento angolare rispetto all'*inertial frame*.

Dato il vettore $\sigma = (\phi \ \theta \ \psi)^T$ si ottengono tre matrici di rotazione:

- matrice di rotazione attorno all'asse X:

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

- matrice di rotazione attorno all'asse Y:

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

- matrice di rotazione attorno all'asse Z:

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

Denominiamo \mathbf{c} il vettore delle coordinate rispetto al *Body Frame* e \mathbf{s} le coordinate di navigazione, quali coordinate rispetto all'*Inertial Frame*. La matrice completa è pari al prodotto delle matrici di rotazione nell'ordine $R(X, \phi)$, $R(Y, \theta)$, $R(Z, \psi)$.

Abbiamo quindi:

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi & \cos\theta\sin\phi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

che compattiamo in:

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = R_i^b * \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.4.1)$$

con R_i^b matrice di trasferimento dall'*Inertial Frame* al *Body Frame*. Le tre matrici ottenute in precedenza sono ortonormali, ovvero $R^T = R^{-1}$.

Sfruttando l'orto normalità possiamo effettuare la trasformazione inversa dal *Body Frame* all'*Inertial Frame*:

$$\mathbf{s} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = (R_i^b)^T * \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = R_b^i * \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad (3.4.2)$$

Ottenute le matrici che permettono di effettuare queste trasformazioni, ricaviamo anche le altre grandezze fisiche, quali ad esempio la velocità.

Il vettore $\mathbf{v}_b = [u \ v \ w]^T$ indica le componenti della velocità traslazionale relative al *Body Frame*, mentre quelle relative al sistema inerziale, attuando la trasformazione tramite la matrice completa di rotazione, risulta essere pari a

$$\mathbf{v}_i = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_b^i * \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (3.4.3)$$

Per le trasformazioni relative alla velocità angolare tra *Inertial Frame* e *Body Frame* utilizziamo una matrice completa di rotazione del tipo

$$T_b^i = \begin{bmatrix} 1 & \frac{\sin\phi \sin\theta}{\cos\theta} & \frac{\cos\phi \sin\theta}{\cos\theta} \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \quad (3.4.4)$$

Prendiamo $\boldsymbol{\omega}_b = [p \ q \ r]^T$ come velocità angolare del sistema di riferimento solidale con il drone calcolata dall'unità *IMU* e, attraverso la matrice di rotazione appena introdotta, andiamo a trasformarla rispetto al sistema inerziale, quindi:

$$\boldsymbol{\omega}_i = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T_b^i * \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.4.5)$$

Essendo anche T_b^i ortonormale sarà invertibile. Possiamo perciò calcolare la velocità angolare rispetto al *Body Frame* a partire da $\boldsymbol{\omega}_i$.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\text{sen}\theta \\ 0 & \text{cos}\phi & \text{sen}\phi\text{cos}\theta \\ 0 & -\text{sen}\phi & \text{cos}\phi\text{cos}\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T_i^b * \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (3.4.6)$$

3.5 Dinamica rotazionale

Assumiamo che il quadricottero preso in esame abbia una struttura rigida. È possibile utilizzare un sistema di coordinate generalizzate (angoli di Eulero) per descrivere le rotazioni del sistema, tramite l'utilizzo delle equazioni di Eulero-Lagrange che, in presenza di forze conservative nel sistema, permettono di enunciare le equazioni del moto mediante funzioni scalari.

La Lagrangiana (L) di un sistema fisico non è altro che la differenza tra l'energia cinetica (K) e l'energia potenziale (E), uniche componenti prese in considerazione. Se L è una quantità nota allora possiamo scrivere le equazioni del sistema come:

$$\Gamma_i = \frac{d}{dt} \left(\frac{\delta}{\delta \dot{q}_i} L \right) - \frac{\delta}{\delta q_i} L \quad (3.5.1)$$

con Γ_i pari alle forze generalizzate che agiscono sul sistema e q_i pari alle coordinate generalizzate.

Per poter esprimere l'energia cinetica K in termine lagrangiani, bisogna trasformare le componenti della velocità relative al *Body Frame* in coordinate generalizzate utilizzando la matrice di cambiamento di base R_i^b .

Avremo quindi la funzione dell'energia cinetica espressa come:

$$K = \frac{1}{2} \int dm (s_i(c_1, c_2, c_3)) * \|\mathbf{v}_i(c_1, c_2, c_3)\|^2 \quad (3.5.2)$$

Esponiamo ora l'energia cinetica tramite le componenti del tensore momento di inerzia espresse sempre nel sistema di coordinate generalizzate.

In termini matriciali otteniamo:

$$I = \int dm \begin{bmatrix} (c_2^2 + c_3^2) & -c_1c_2 & -c_1c_3 \\ -c_1c_2 & (c_1^2 + c_3^2) & -c_2c_3 \\ -c_1c_3 & -c_2c_3 & (c_1^2 + c_2^2) \end{bmatrix} \quad (3.5.3)$$

Si suppone la simmetria del mezzo. I termini sulla diagonale della matrice espressa sono legati ai momenti di inerzia rispetto agli assi principali del *Body Frame*.

Siamo perciò in grado di esprimere il seguente momento di inerzia, eliminando i termini fuori diagonale, i cosiddetti prodotti di inerzia (relativi all'accoppiamento incrociato dell'inerzia degli assi):

$$I = \int dm \begin{bmatrix} (c_2^2 + c_3^2) & 0 & 0 \\ 0 & (c_1^2 + c_3^2) & 0 \\ 0 & 0 & (c_1^2 + c_2^2) \end{bmatrix} = \begin{bmatrix} I_{c_1,c_1} & 0 & 0 \\ 0 & I_{c_2,c_2} & 0 \\ 0 & 0 & I_{c_3,c_3} \end{bmatrix} \quad (3.5.4)$$

La formula dell'energia cinetica sarà la seguente:

$$K = \frac{1}{2}I_{c_1,c_1}(\dot{\phi} - \dot{\psi}\sin\theta)^2 + \frac{1}{2}I_{c_2,c_2}(\theta\cos\phi + \dot{\psi}\sin\phi\cos\theta)^2 + \frac{1}{2}I_{c_3,c_3}(\dot{\theta}\sin\phi - \dot{\psi}\cos\phi\cos\theta)^2 \quad (3.5.5)$$

L'energia potenziale rispetto al sistema di coordinate generalizzate si ottiene calcolando il contributo di ogni massa infinitesima, partendo dalla quota verticale nel sistema di coordinate inerziale:

$$E = \int c_1(-g * \sin\theta)dm(c_1) + \int c_2(g * \sin\phi\cos\theta)dm(c_2) + \int c_3(g * \cos\phi\cos\theta)dm(c_3) \quad (3.5.6)$$

Ora, considerando la Lagrangiana del sistema, dopo aver sviluppato le derivate e aver fatto alcune operazioni algebriche per semplificare le equazioni ed esprimere le velocità angolari come variazioni di angoli euleriani, ponendo la quota iniziale nulla, con conseguente annullamento dell'energia potenziale, e considerando variazioni per piccoli angoli ($\cos x=1$, $\sin x=0$) avremmo che:

$$\begin{aligned}
\tau_\phi &= I_{c_1, c_1} \ddot{\phi} = (I_{c_2, c_2} - I_{c_3, c_3}) \dot{\theta} \dot{\psi} \\
\tau_\theta &= I_{c_2, c_2} \ddot{\theta} = (I_{c_3, c_3} - I_{c_1, c_1}) \dot{\phi} \dot{\psi} \\
\tau_\psi &= I_{c_3, c_3} \ddot{\psi} = (I_{c_1, c_1} - I_{c_2, c_2}) \dot{\phi} \dot{\theta}
\end{aligned} \tag{3.5.7}$$

In assenza di momenti esterni queste sono le equazioni che descrivono il moto rotazionale del sistema.

3.6 Momenti esterni

Nel calcolo della dinamica del nostro sistema bisogna definire anche i momenti meccanici esterni, relativi alla spinta dei motori (thrust) e alla forza di attrito viscoso (drag). Essi andranno sommati ai momenti risultanti ricavati dal precedente calcolo. Nello specifico risultano essere pari a:

$$M_{td} = \begin{pmatrix} l_x b (\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) \\ l_y b (-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2) \\ -d (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{pmatrix} \tag{3.6.1}$$

dove l_x e l_y sono le proiezioni dei bracci del quadricottero sugli assi x e y, b è il coefficiente di spinta dei motori e d è il coefficiente di resistenza dell'aria.

A questo contributo non aggiungeremo il momento relativo all'effetto giroscopico risultante delle eliche, poiché assumiamo di operare con il sistema di controllo in condizioni più possibili simili a quelle stazionarie.

Di seguito le equazioni riguardanti i momenti totali:

$$\begin{aligned}
\tau_\phi &= I_{c_1, c_1} \ddot{\phi} = (I_{c_2, c_2} - I_{c_3, c_3}) \dot{\theta} \dot{\psi} + l_y b (-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2) \\
\tau_\theta &= I_{c_2, c_2} \ddot{\theta} = (I_{c_3, c_3} - I_{c_1, c_1}) \dot{\phi} \dot{\psi} + l_x b (\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) \\
\tau_\psi &= I_{c_3, c_3} \ddot{\psi} = (I_{c_1, c_1} - I_{c_2, c_2}) \dot{\phi} \dot{\theta} + d (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)
\end{aligned} \tag{3.6.2}$$

3.7 Forze agenti sul sistema

L'unica forza agente sul sistema risulta essere la forza peso. Affinchè ci sia equilibrio nel sistema essa viene contrastata dall'attività dei motori che, attraverso la loro spinta verticale, permettono la stabilità.

La forza peso è esprimibile rispetto all'*Inertial Frame* come:

$$F_p = [0 \quad 0 \quad mg]^T$$

Ora, utilizzando la matrice di cambiamento di base R_b^i enunciamo, rispetto al sistema di riferimento inerziale, i contributi relativi alla spinta verticale.

$$\begin{aligned} F_x &= (\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi)b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ F_y &= (-\sin\phi\cos\psi + \cos\phi\sin\theta\cos\psi)b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ F_z &= mg - (\cos\phi\cos\psi)b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{aligned} \quad (3.7.1)$$

3.8 Equazioni generali del sistema

Il sistema di equazioni generali formato dalle forze risultanti e dai momenti è:

$$\left\{ \begin{array}{l} F_x = (\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi)b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ F_y = (-\sin\phi\cos\psi + \cos\phi\sin\theta\cos\psi)b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ F_z = mg - (\cos\phi\cos\psi)b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ \tau_\phi = (I_{c_3,c_3} - I_{c_3,c_1})\dot{\theta}\dot{\psi} + l_y b(-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2) \\ \tau_\theta = (I_{c_3,c_3} - I_{c_1,c_1})\dot{\phi}\dot{\psi} + l_x b(\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) \\ \tau_\psi = (I_{c_1,c_1} - I_{c_2,c_2})\dot{\phi}\dot{\theta} + d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{array} \right\} \quad (3.8.1)$$

Possiamo semplificare il tutto facendo coincidere gli assi principali del *Body Frame* con gli assi principali dell'*Inertial Frame* e applicando la seconda legge di Newton:

$$\Sigma F_i = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \quad (3.8.2)$$

$$\Sigma \tau_i = I \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \quad (3.8.3)$$

Applicando le relazioni che permettono di trasformare le componenti dal sistema di riferimento inerziale rispetto a quello solidale col mezzo ed esplicitando le derivate otteniamo:

$$\Sigma R_i^b F_b = m \left(\dot{R}_b^i \begin{bmatrix} u \\ v \\ w \end{bmatrix} + R_b^i \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \right) \quad (3.8.4)$$

$$\Sigma T_i^b \tau_b = I \left(\dot{T}_b^i \begin{bmatrix} p \\ q \\ r \end{bmatrix} + T_b^i \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \right) \quad (3.8.5)$$

La velocità lineare di una terna ortogonale in rotazione può essere espressa come il prodotto vettoriale della velocità angolare della stessa per la sua rotazione.

Con $\dot{R}_b^i = \omega_b \times R_b^i$ e $\dot{T}_b^i = \omega_b \times T_b^i$ semplificando i termini comuni relativi alle matrici di rotazione abbiamo:

$$m \left(\omega_b \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \right) = \Sigma F_b \quad (3.8.6)$$

$$I \left(\omega_b \times \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \right) = \Sigma \tau_b \quad (3.8.7)$$

Con $\omega_b = [p \quad q \quad r]^T$ sviluppando i prodotti vettoriali otteniamo:

$$\Sigma F_b = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (3.8.8)$$

$$\Sigma \tau_b = \begin{bmatrix} \dot{p}I_{c_1,c_1} + qr(I_{c_3,c_3} - I_{c_2,c_2}) \\ \dot{q}I_{c_2,c_2} + pr(I_{c_1,c_1} - I_{c_3,c_3}) \\ \dot{r}I_{c_3,c_3} + pq(I_{c_2,c_2} - I_{c_1,c_1}) \end{bmatrix} \quad (3.8.9)$$

Per il calcolo delle forze esterne otteniamo:

$$R_i^b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix} \quad (3.8.10)$$

Sommando i termini relativi al calcolo delle forze esterne appena calcolato e l'equazione (3.8.8) abbiamo:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} m\dot{u} \\ m\dot{v} \\ m\dot{w} \end{bmatrix} = \begin{bmatrix} mqv - mrv - \text{sen}\theta * mg \\ mru - mpw + \text{sen}\phi\text{cos}\theta * mg \\ mpv - mqu + \text{cos}\phi\text{cos}\theta * mg + b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix} \quad (3.8.11)$$

In relazione ai momenti, dato che la spinta dei motori avviene in maniera solidale col mezzo, sommando i contributi otteniamo:

$$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} \dot{p}I_{c_1,c_1} \\ \dot{q}I_{c_2,c_2} \\ \dot{r}I_{c_3,c_3} \end{bmatrix} = \begin{bmatrix} bl(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) - qr(I_{c_3,c_3} - I_{c_2,c_2}) \\ bl(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) - pr(I_{c_1,c_1} - I_{c_3,c_3}) \\ d(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) - pq(I_{c_2,c_2} - I_{c_3,c_3}) \end{bmatrix} \quad (3.8.12)$$

Attraverso queste formule abbiamo una modellazione matematica per il nostro drone.

4. Analisi del modello in Simulink

Il modello Simulink del minidrone Mambo è composto da diversi blocchi che possono essere analizzati separatamente. Esso ha la struttura tipica dei sistemi di controllo in controreazione, di cui si distinguono:

- *Signal Builder;*
- *Flight Control System;*
- *Non Linear Airframe;*
- *Environment;*
- *Sensors;*
- *Visualization.*

Il primo blocco è il generatore di segnali di riferimento; il secondo e terzo sono rispettivamente controllore e modello del processo; il blocco *Environment* contiene le variabili di ambiente, il *Sensors* presenta le costanti di trasduzione e le relazioni tra i diversi sensori presenti sul mezzo e infine l'ultimo blocco consente la visualizzazione dei valori delle variabili di stato e dei tracciati dei segnali di controllo.

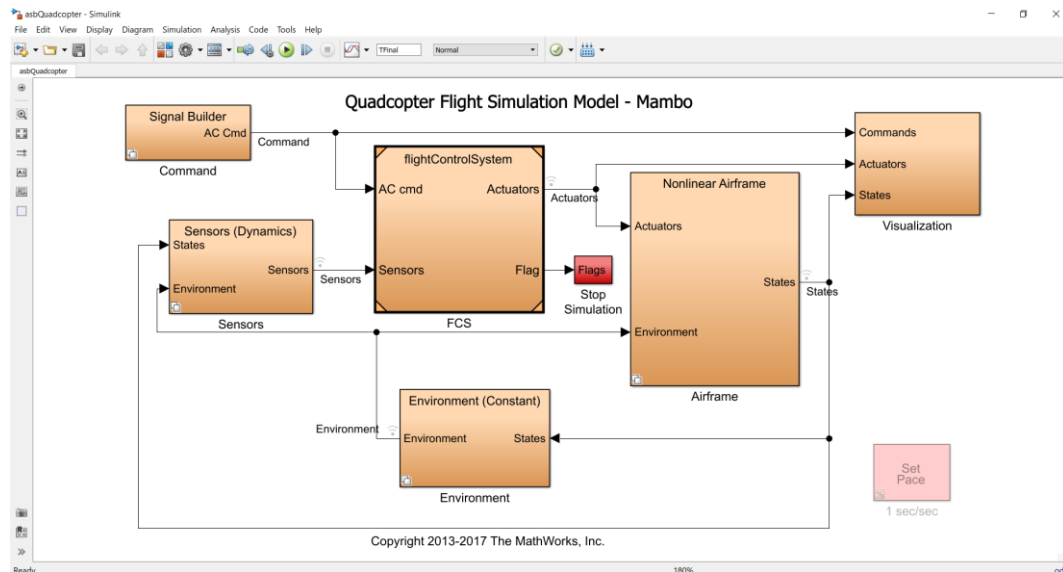


Fig.8 Quadcopter Flight Simulation Model-Mambo

4.1 Signal Builder

Nel blocco *Command* possiamo scegliere a quale fonte dei segnali di riferimento affidarci per controllare il drone. Questo avviene tramite il settaggio del comando VSS_COMMAND sul *Workspace* di MATLAB dal progetto *absQuadcopter*. Le scelte saranno:

- *Signal Editor*;
- *Joystick*;
- *Data*;
- *Spreadsheet Data*.

Settando il primo blocco utilizzeremo i riferimenti predefiniti da MATLAB e i riferimenti definiti dall'utente; con il secondo blocco potremmo gestire manualmente, tramite l'ausilio di un joystick, l'andamento del veicolo; per

concludere con *Data e Spreadsheet Data* andremo ad utilizzare rispettivamente i riferimenti memorizzati all'interno di un MAT file (.mat) e di un foglio di calcolo Excel (.xlsx).

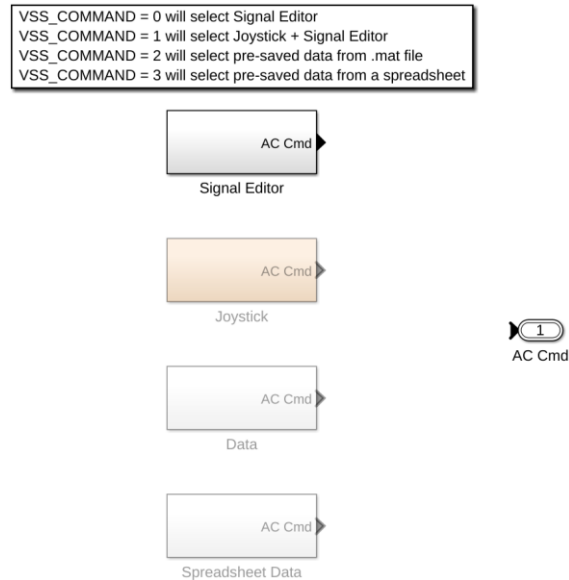


Fig.9 Blocco Command

Il blocco di nostro interesse che sarà utile successivamente per il controllo del drone è il blocco Signal Editor.

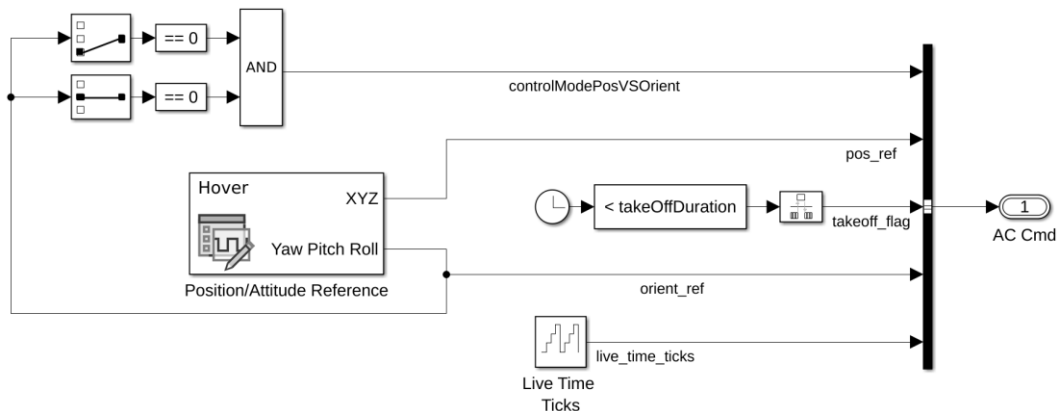


Fig.10 Signal Editor

Gli elementi principali che lo compongono sono:

- *position/Attitude Reference*: generatore dei segnali di riferimento per i 6 gradi di libertà;
- *controlModePosvsOrient*: variabile booleana che se settata 'true' consente ai riferimenti di pitch e roll di entrare in modalità di controllo della posizione

segundo i riferimenti impartiti dalle variabili X e Y; al contrario, se settata 'false', i riferimenti di pitch e roll entrano in modalità di controllo dell'orientamento seguendo i riferimenti impartiti dal blocco *Position/Attitude Reference* anche se la posizione X-Y non risultasse controllata;

- *pos_ref*: riferimenti per X, Y e Z;
- *orient_ref*: riferimenti per pitch, roll e yaw.

Si preannuncia che per le simulazioni il blocco *Position/Attitude Reference* sarà sostituito con un blocco *Signal Builder* che permette di definire segnali personalizzati.

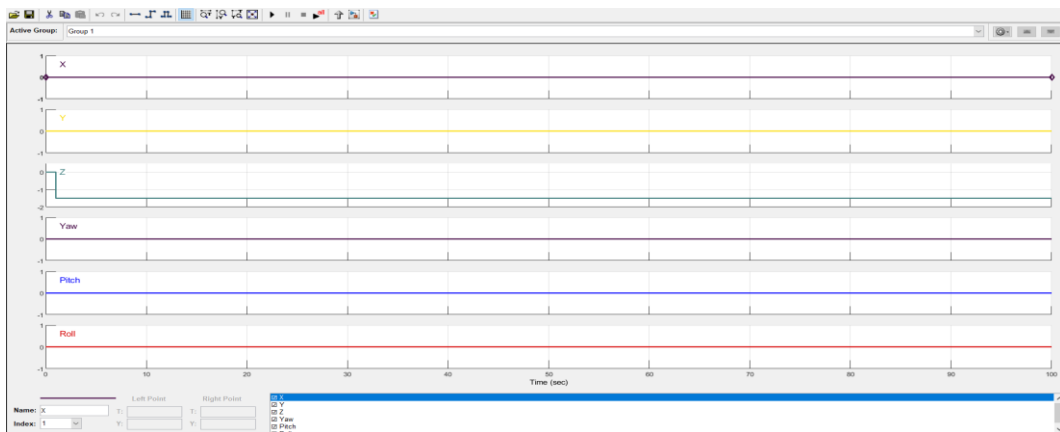


Fig.11 Signal Builder

4.2 Flight Control System

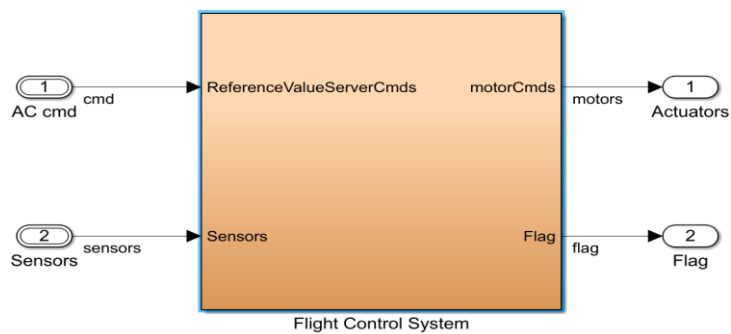


Fig.12 Flight Control System

Il seguente blocco è uno dei più importanti presente sul modello *Simulink* del quadricottero, poiché responsabile dell'attività di sviluppo del sistema di controllo. I segnali d'ingresso del seguente blocco provengono dal *Signal Builder* per quanto riguarda i valori di riferimento e dal blocco *Sensors* per i valori delle variabili di stato.

Come uscite notiamo il segnale *Flag* relativo all'uscita di sicurezza che, nel caso in cui si presenti una perdita di assetto, provvede a fornire un segnale di controllo che interrompe la simulazione.

Infine, l'uscita *Actuators* riguarda i segnali di riferimento degli attuatori che controlleranno i motori del modello non lineare.

L'FCS è formato da 3 sottosistemi principali:

- *State Estimator*: elabora i segnali ricevuti dai differenti sensori presenti sul mezzo e li fornisce al *Flight Controller*;
- *Flight Controller*: realizza l'azione di controllo dando in uscita i segnali di controllo diretti ai motori;
- *Crash Predictor Flag*: si tratta di un blocco che elabora le informazioni provenienti dai sensori e dallo *State Estimator* per constatare se ci sono le condizioni di volo normali.

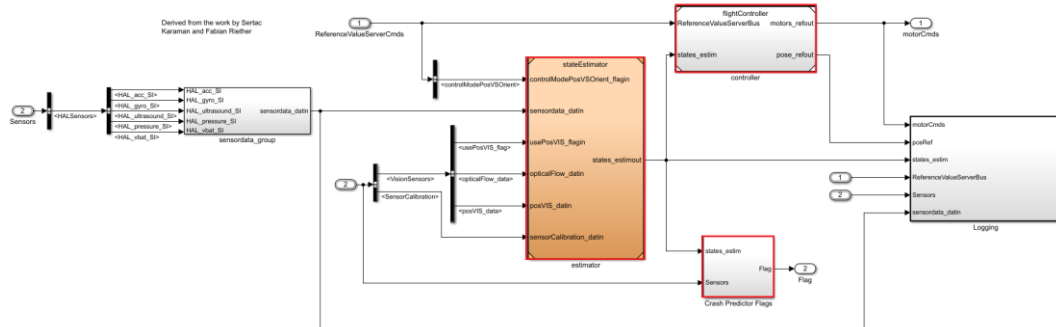


Fig.13 Flight Control System

Nell'approfondire la logica di controllo 'apriamo' il blocco *Flight Controller*.

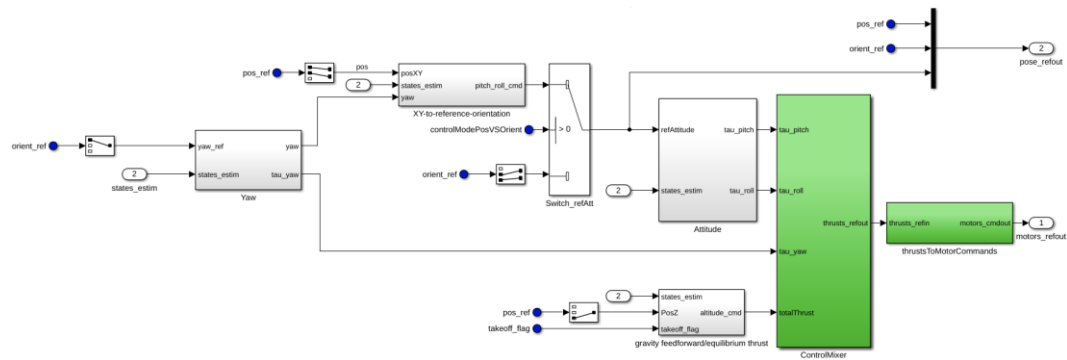


Fig.14 *Flight Controller*

I controllori veri e propri riguardano i blocchi: *Yaw*, *XY-to-reference-orientation*, *gravity feedforward/equilibrium thrust* e *Attitude*.

Nel primo vi è l'implementazione del controllore PID per il controllo dello yaw; nel secondo per il controllo delle coordinate X e Y e in *gravity feedforward/equilibrium thrust* per il controllo della dinamica dell'altitudine del veicolo.

Per quanto concerne il blocco *Attitude*, al suo interno viene implementato il controllore PID per il controllo della dinamica del pitch e del roll del quadricottero. Quest'ultimo riceve in ingresso, in base al settaggio degli switch, il segnale di controllo generato dal blocco *XY-to-reference-orientation* oppure il segnale di orientamento proveniente dal *Signal Builder*. Tale segnale sarà chiamato *refAttitude* e verrà confrontato con i valori correnti riguardanti gli angoli di pitch e roll provenienti dallo *State Estimator*, in modo da ricavarne l'errore rispetto al riferimento.

I valori conseguenti alimenteranno un blocco proporzionale e uno integratore. Inoltre, i segnali delle velocità p e q , i quali vengono forniti ad un blocco che li moltiplica per una costante, a loro volta andranno ad alimentare un blocco di tipo derivativo.

Sommando tutti i contributi avremo un controllore PID. Questo è il blocco nel quale andremo ad agire più in avanti nella progettazione di controllori per il controllo della dinamica di roll.

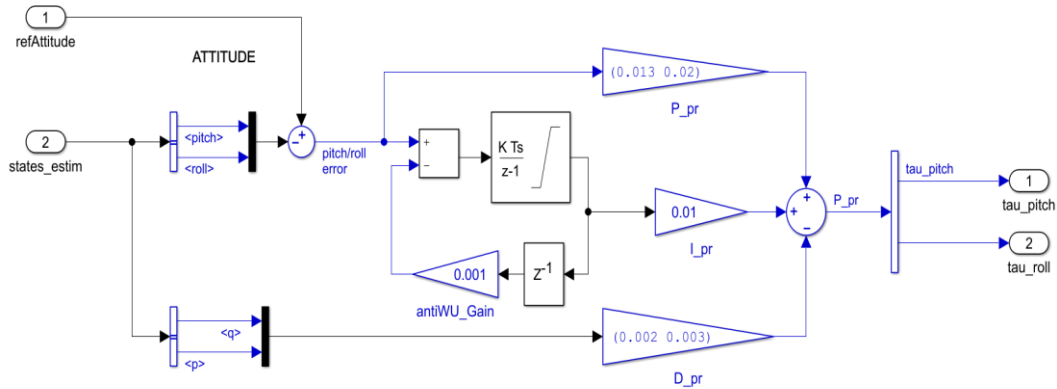


Fig.15 Blocco Attitude

4.3 Controllo dei processi

4.3.1 Non linear Airframe

Nel blocco *Airframe* ci sono due sottosistemi, *Linear Airframe* e *Non Linear Airframe*, che rappresentano il modello lineare e non lineare del drone.

Possiamo scegliere a quale dei due modelli fare affidamento settando nel *Workspace* di MATLAB la funzione `VSS_VEHICLE`.

Ponendola pari a 1 (opzione di default) utilizzeremo il modello non lineare del sistema. Questo blocco è formato da 4 sottoblocchi:

- *Ac Model*;
- *6DOF (Quaternion)*;
- *Bus setup*;
- *Position on Earth*.

Il primo blocco l'*Ac Model* contiene tutti i risultati della modellazione matematica trovati nel capitolo 3. Esso inoltre determina le forze e i momenti torcenti applicati al drone nei tre assi a partire dalla conoscenza del thrust esercitato dagli attuatori, delle matrici di conversione, dalle variabili d'ambiente e dalle velocità lineari e angolari rispetto al *Body Frame* del mezzo.

Il blocco *6DOF (Quaternion)* è presente nel sistema dopo l'installazione della Toolbox di MATLAB '*Aerospace Blockset*' che serve a calcolare posizioni, velocità e accelerazioni caratterizzanti il moto del drone. Esse, in base alle forze e i momenti applicati, sono:

- V_e : velocità traslazionale rispetto al sistema di riferimento inerziale;
- X_e : posizione nello spazio rispetto al sistema di riferimento inerziale;
- V_b : velocità traslazionale rispetto al *Body Frame*;
- ω_b : velocità angolare rispetto al *Body Frame*;
- ϕ, θ, ψ : angoli che descrivono l'orientamento del *Body Frame* del drone rispetto all'*Inertial Frame* tramite delle rotazioni attorno agli assi x, y, z (angoli di Eulero);
- DCM_{be} : coefficienti della matrice di conversione di coordinate dall'*Inertial frame* al *Body Frame*;
- $\frac{d\omega_b}{dt}$: accelerazione angolare rispetto al *Body Frame*;
- A_{bb} : accelerazione traslazionale lungo gli assi del *Body Frame* rispetto a sé stesso;
- A_{be} : accelerazione traslazionale lungo gli assi del *Body Frame* rispetto all'*Inertial Frame*.

Il blocco *Bus Setup* raggruppa tutte le componenti che caratterizzano lo stato del sistema in un unico vettore, in modo tale che possa essere trasferito in feedback al controllore per il controllo dello stato del drone in volo.

Nell'ultimo blocco, il *Position on Earth*, prese le posizioni X_e del drone nello spazio rispetto al sistema di riferimento inerziale e l'altezza di riferimento tra la superficie della Terra e il suo riferimento piatto, viene stimata la longitudine, la latitudine e l'altitudine espresse rispetto al sistema di riferimento che tiene conto della curvatura della Terra.

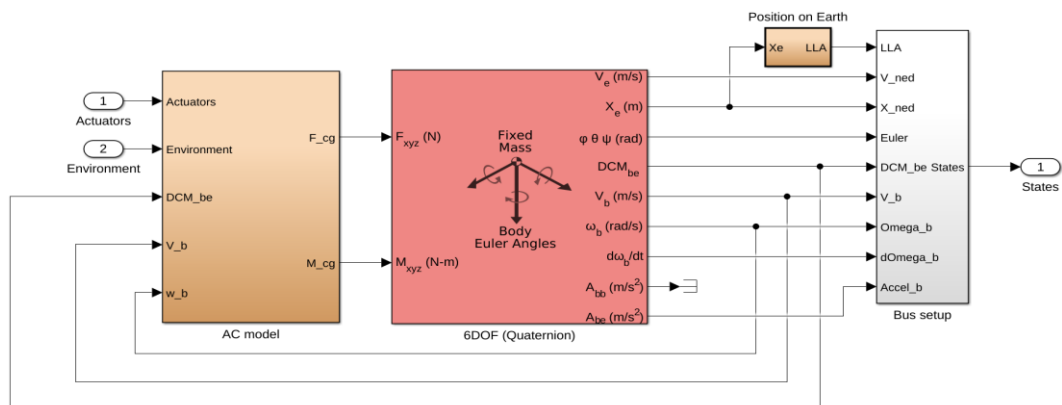


Fig.16 Non Linear Airframe

4.3.2 Linear Airframe

La dinamica rappresentata dal *Non Linear Airframe* ha come vantaggio quello di essere molto precisa e come svantaggio quello di essere difficile da controllare.

Per questo, nel modello *Simulink* del progetto, è presente anche il blocco *Linear Airframe* che ci mostra una versione linearizzata del modello.

In seguito, mostreremo come vengono ricavate le matrici A, B, C e D e come si progettano dei controllori tramite tecniche di controllo lineare (sintesi in frequenza e tramite luogo delle radici) per il controllo del modello non lineare del drone.

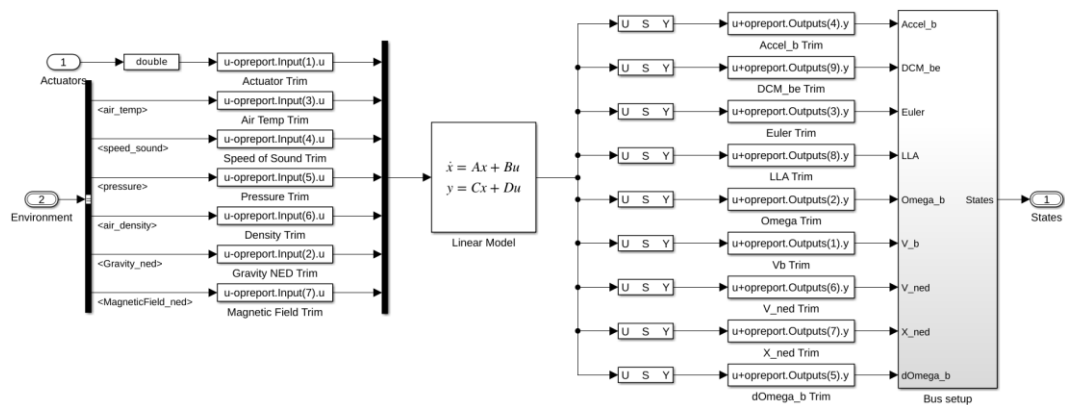


Fig.17 Linear Airframe

Gli ingressi presenti sono relativi ai motori (*Actuators*), a variabili relative all'ambiente come temperatura, pressione atmosferica, contributo della gravità e del campo magnetico, velocità del suono, densità dell'aria.

Le uscite sono identiche a quelle del sistema non lineare.

5. Sintesi dei controllori

In questo capitolo verrà trattata la sintesi dei controllori adottati per il controllo del movimento di roll.

Sarà analizzato il loro funzionamento, in un intervallo di 100s, rispetto ai riferimenti costanti delle posizioni X-Y e rispetto ai riferimenti variabili delle posizioni X-Y formati da un movimento 'step' rispettivamente dall'istante 30s e dall'istante 50s.

5.1 Riferimento costante e variabile

Nella sezione precedente si era preannunciato l'inserimento di un blocco *Signal Builder* all'interno del blocco primario *Command*.

Nel nuovo blocco andremo ad inserire, oltre ai segnali di riferimento costanti già presenti, dei segnali di riferimento variabili per il controllo della posizione X-Y.

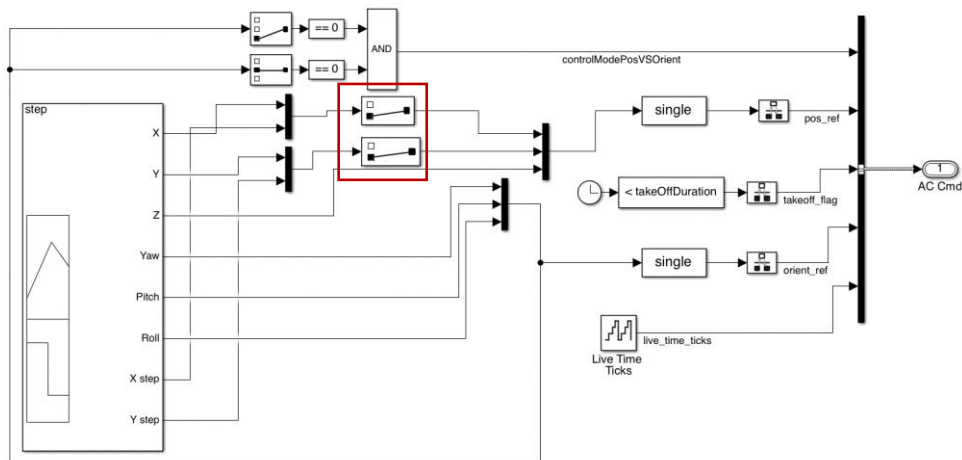


Fig.18 Blocco Command con Signal Builder

Essi saranno degli 'step', ovvero dei movimenti in X e Y dai quali andremo a valutare l'efficienza dei controllori progettati con i due metodi di controllo lineare. Per scegliere quale riferimento utilizzare basterà settare gli *switch* evidenziati in modo tale da cambiare riferimento in base alle esigenze.

Attraverso l'interfaccia *Signal Builder* possiamo modificare, oltre al riferimento relativo alla posizione (X, Y, Z), anche quello relativo all'orientamento in aria del drone (Yaw, Pitch, Roll), cambiando l'andamento delle diverse variabili nel tempo.

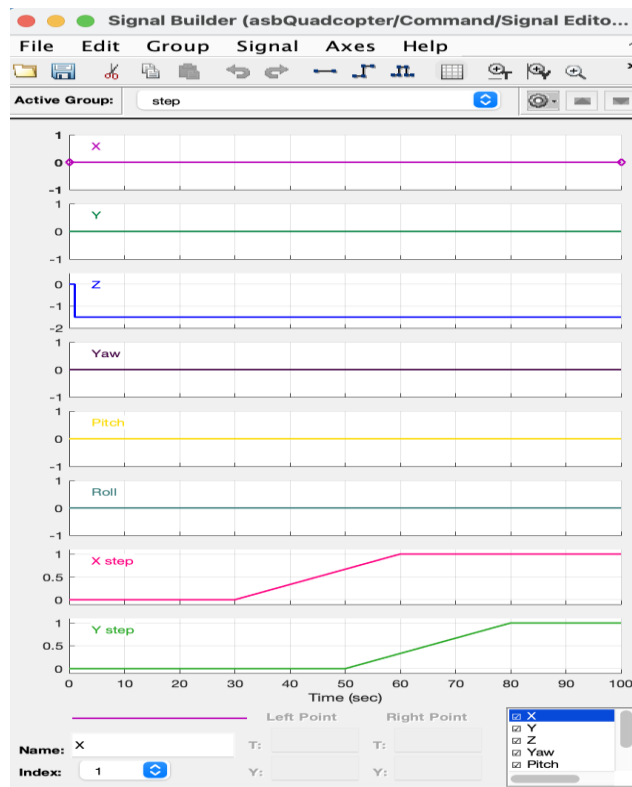


Fig.19 Signal Builder

Il drone si muoverà in $t=30s$ di 1 metro in X fino al tempo $t=60s$; contemporaneamente si muoverà di un metro in Y dal tempo $t=50s$ a $t=80s$. La simulazione totale ha durata di 100s.

5.2 PID originali

Per visualizzare il comportamento del sistema tramite controllori utilizziamo dei blocchi di tipo *Scopes* dentro il blocco *Logging*, il quale contiene l'andamento di tutte le variabili del modello.

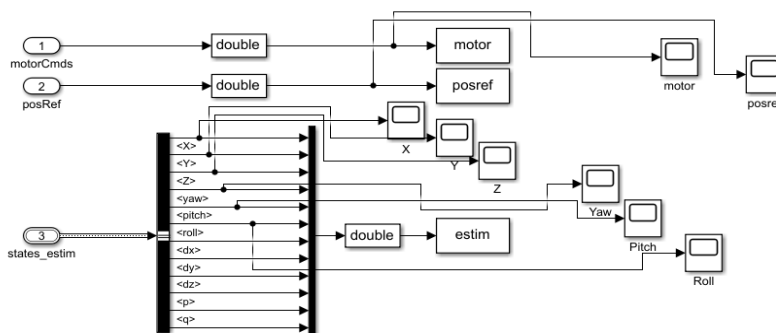


Fig.20 Blocco Logging con Scope

Portiamo all'attenzione i grafici relativi alle posizioni X-Y e il grafico relativo all'andamento del Roll.

In particolare, vedremo se risulteranno controllate le posizioni X-Y e andremo a confrontare il riferimento del Roll con le prestazioni del controllore.

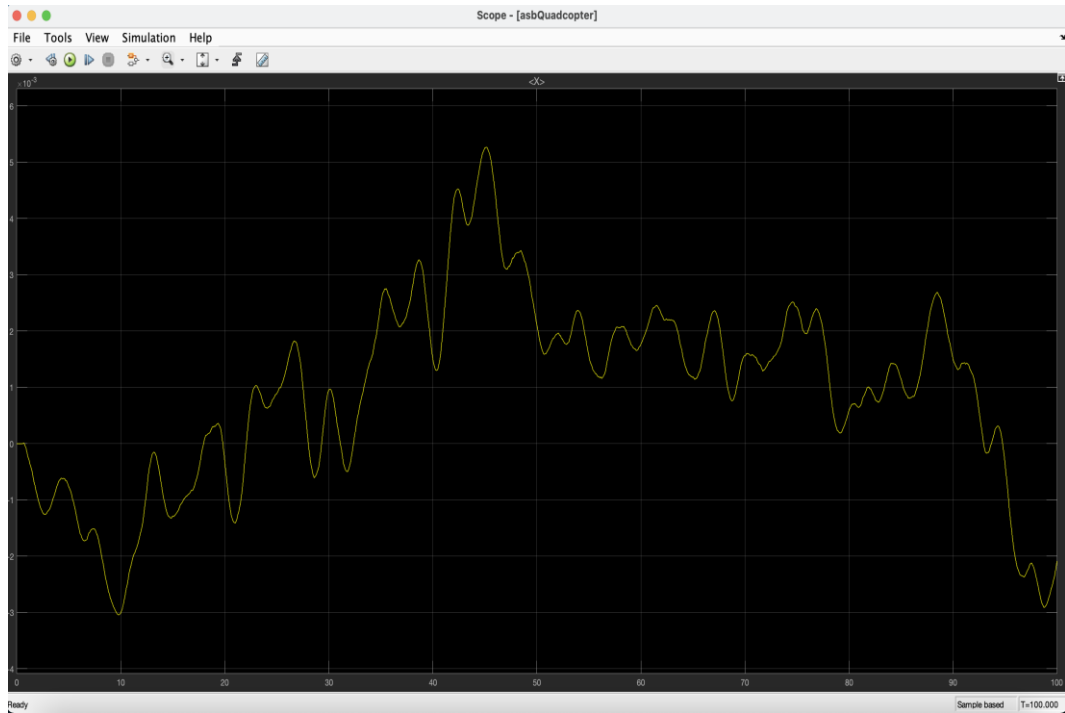


Fig.21 Scope con controllore originale PID variabile X, riferimento costante

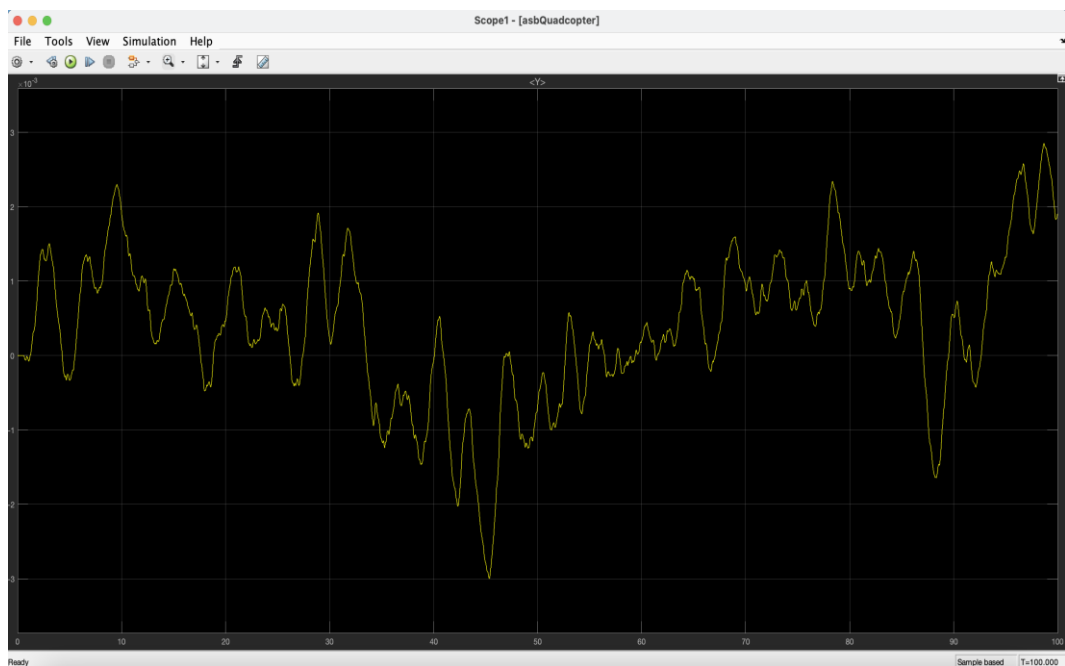


Fig.22 Scope con controllore originale PID variabile Y, riferimento costante

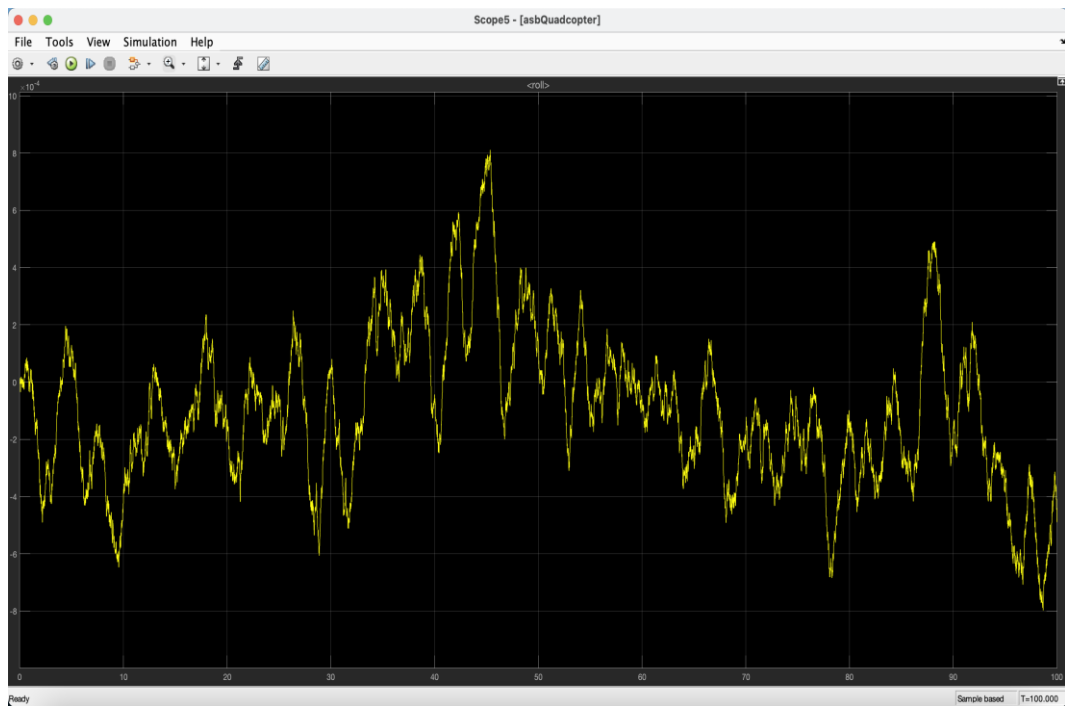


Fig.23 Scope con controllore originale PID variabile Roll, riferimento costante

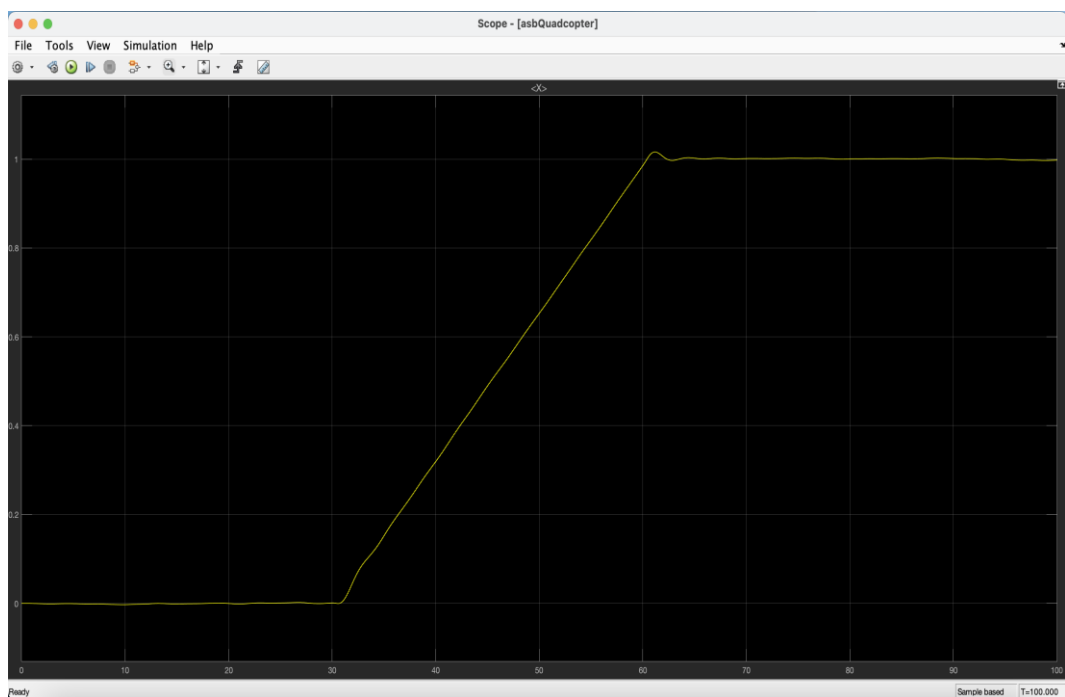


Fig.24 Scope con controllore originale PID variabile X, riferimento variabile

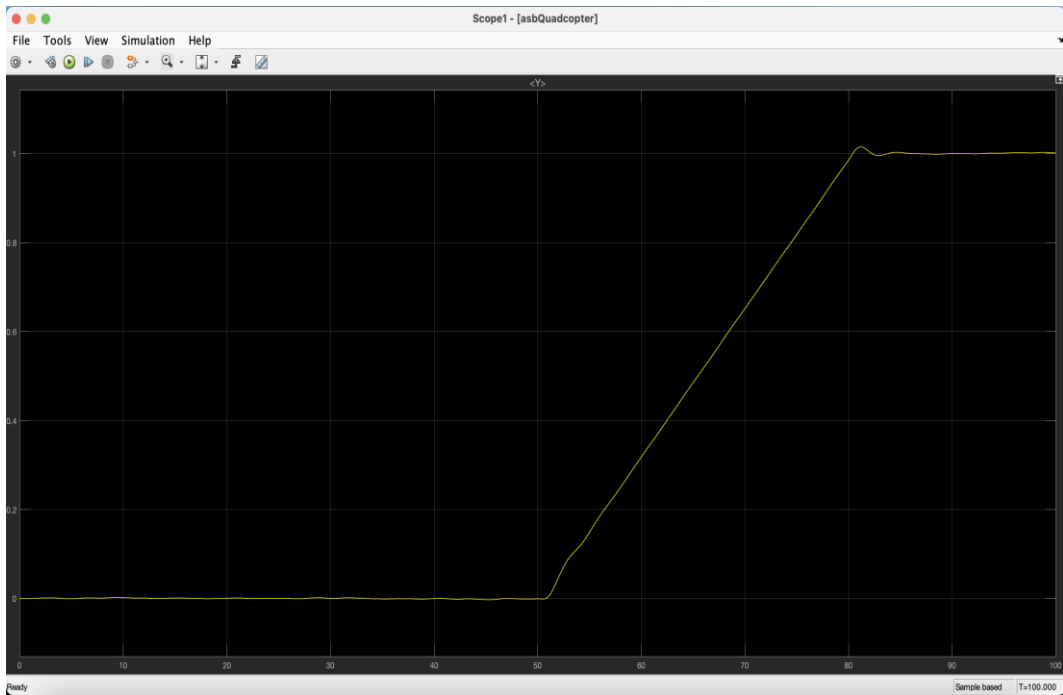


Fig.25 Scope con controllore originale PID variabile Y, riferimento variabile

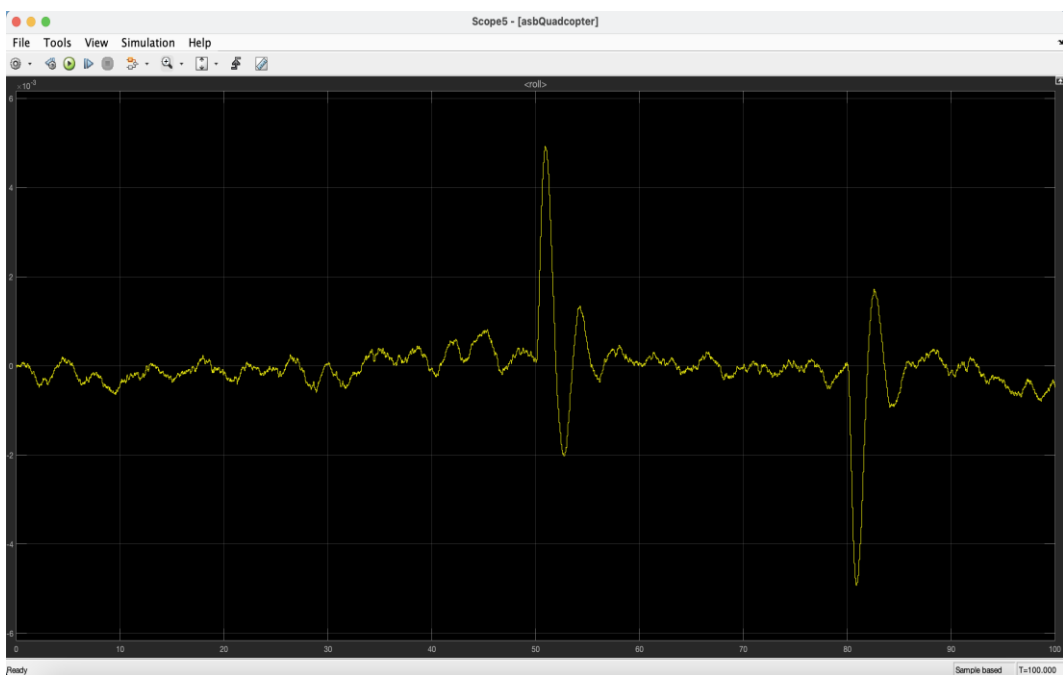


Fig.26 Scope con controllore originale PID variabile Y, riferimento variabile

Da questi grafici possiamo notare come il controllo della posizione X-Y e del Roll con i PID originali è ottimale dato che le oscillazioni rispetto al riferimento sono nell'ordine di $\pm 10^{-3} [m]$ e quindi trascurabili.

5.3 Linearizzazione del modello non lineare

Notiamo come il modello non lineare esposto nella sezione precedente risulta essere più accurato e preciso rispetto al caso reale, ma al tempo stesso presenta al suo interno degli elementi di non linearità che ne rendono difficoltoso lo studio.

Per questo cerchiamo di ottenere un modello linearizzato che si comporti in maniera quanto più simile al modello non lineare in modo tale da permettere di adottare tecniche di controllo lineare. Preso un sistema non lineare e stazionario:

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

con $x(t)$ variabili di stato, $u(t)$ ingressi e $y(t)$ le uscite, descriviamo il suo comportamento intorno un particolare punto di lavoro $(x_0(t), y_0(t))$ tramite un sistema lineare che approssimi quello originario.

Considerando piccole oscillazioni intorno al punto stesso avremo:

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

Fissato il punto di lavoro che soddisfi le condizioni di equilibrio nominale rispetto ad un ingresso costante del tipo $u(t) = \bar{u} \quad \forall t \geq t_0$ con $\dot{x}(t) = f(\bar{x}, \bar{u}) = 0$ e $y(t) = g(\bar{x}, \bar{u}) = \bar{y} \quad \forall t \geq t_0$ descriviamo le variabili di stato, ingresso e uscita rispetto ai valori di equilibrio (\bar{x}, \bar{u}) :

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

con equazioni del sistema dinamico:

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

Sviluppando in serie di Taylor rispetto a x e u negli intornoi $x = \bar{x}$ e $u = \bar{u}$ e troncando la serie al primo ordine avremo:

$$\delta\dot{x}(t) = \frac{\delta f(x, u)}{\delta x} \delta x(t) + \frac{\delta g(x, u)}{\delta u} \delta u(t)$$

$$\delta y(t) = \frac{\delta g(x, u)}{\delta x} \delta x(t) + \frac{\delta f(x, u)}{\delta u} \delta u(t)$$

dove i termini $\frac{\delta f(x, u)}{\delta x}$, $\frac{\delta g(x, u)}{\delta u}$, $\frac{\delta g(x, u)}{\delta x}$, $\frac{\delta f(x, u)}{\delta u}$ corrispondono rispettivamente alle matrici A, B, C, D del sistema lineare che approssima quello originale.

Queste matrici possiamo trovarle nel blocco *Airframe* relativo al modello lineare.

Esse sono così composte:

- matrice A: ha una dimensione 12×12 dove sono presenti le variabili di stato del sistema;
- matrice B: ha una dimensione 12×14 dove ci sono gli ingressi del sistema;
- matrice C: ha una dimensione 33×12 dove sono presenti le uscite del sistema;
- matrice D: ha una dimensione 33×14 dove viene descritta l'azione sull'uscita dovuta dalle variabili di ingresso.

I coefficienti relativi a tali variabili possiamo consultarli tramite il *Workspace* di MATLAB accedendo alla struttura dati *linsys*. In tale programma possiamo consultare anche il nome di tali variabili tramite i file *StateName*, *InputName* e *OutputName*.

I valori ricavati saranno relativi all'angolo di Roll da controllare e più nello specifico utilizzeremo:

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

dove ϕ è la posizione angolare di Roll rispetto all'*Inertial Frame*, p è la velocità angolare di roll rispetto al *Body Frame*, *Actuators(1)(2)(3)(4)* è la spinta

applicata ai 4 rotori del drone e $Euler(1)$ è l'angolo di Eulero per il controllo del Roll.

Nelle seguenti immagini sono stati evidenziati le celle riguardanti i valori cercati.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	phi theta p...												
2	phi theta p...												
3	phi theta p...												
4	ub,vb,wb(1)												
5	ub,vb,wb(2)												
6	ub,vb,wb(3)												
7	xe,ye,ze(1)												
8	xe,ye,ze(2)												
9	xe,ye,ze(3)												
10	p,q,r(1)												
11	p,q,r(2)												
12	p,q,r(3)												
13													
14													
15													
16													

Fig.27 File StateName

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Actuators(1)												
2	Actuators(2)												
3	Actuators(3)												
4	Actuators(4)												
5	Atmospher...												
6	Atmospher...												
7	Atmospher...												
8	Atmospher...												
9	Gravity_ne...												
10	Gravity_ne...												
11	Gravity_ne...												
12	MagneticFi...												
13	MagneticFi...												
14	MagneticFi...												
15													
16													

Fig.28 File InputName

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

Fig.29 File OutputName

linsys.A													
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	3.6611e-17	0	0	0	0	0	0	0	1	3.0292e-23	-5.2296e-12	
2	-3.6611e-17	0	0	0	0	0	0	0	0	0	1	5.2295e-12	
3	-6.8347e-13	-2.0323e-28	0	0	0	0	0	0	0	0	-5.2295e-12	1.0000	
4	0	-9.8100	0	-3.4402e-09	-3.4402e-09	6.8532e-13	0	0	0	-5.4616e-11	0.1380	-3.2848e-13	
5	9.8100	-2.8478e-22	0	-3.6595e-09	-3.6595e-09	-6.8520e-13	0	0	0	-0.1380	5.8098e-11	-3.5159e-13	
6	5.4456e-11	5.4456e-11	0	-6.8347e-13	6.8347e-13	0	0	0	0	3.2920e-13	3.5087e-13	0	
7	1.7216e-24	1.3478e-19	3.2920e-13	1.0000	-1.4969e-18	-5.2296e-12	0	0	0	0	0	0	
8	-1.3478e-19	2.0176e-37	3.5087e-13	1.4969e-18	1	5.2295e-12	0	0	0	0	0	0	
9	-3.2920e-13	-3.5087e-13	0	5.2296e-12	-5.2295e-12	1.0000	0	0	0	0	0	0	
10	0	0	0	-5.7590e-08	-5.7590e-08	-2.7287e-14	0	0	0	-2.1715	9.1429e-10	4.3301e-13	
11	0	0	0	4.0369e-08	4.0369e-08	-2.1644e-14	0	0	0	6.4089e-10	-1.6192	-5.1363e-13	
12	0	0	0	-1.4434e-19	-1.2831e-19	3.6185e-19	0	0	0	1.2157e-13	1.1242e-13	-6.8807e-09	
13													
14													
15													
16													

Fig.30 Matrice A

linsys.B													
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	-6.8995e-17	6.8995e-17	-6.8995e-17	6.8995e-17	-9.9331e-14	0	0	0	1.0000	1.4969e-18	5.2296e-12	0	0
5	6.7628e-17	-6.7628e-17	6.7628e-17	-6.7628e-17	9.8151e-14	0	0	0	-1.4969e-18	1.0000	-5.2295e-12	0	0
6	-0.0096	0.0096	-0.0096	0.0096	-8.2855	0	0	0	0	0	1.0000	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0.4202	0.4202	-0.4202	-0.4202	2.5288e-14	0	0	0	0	0	0	0	0
11	0.3133	-0.3133	-0.3133	0.3133	1.8856e-14	0	0	0	0	0	0	0	0
12	-0.0115	-0.0115	-0.0115	-0.0115	-7.9371e-16	0	0	0	0	0	0	0	0
13													
14													
15													
16													

Fig.31 Matrice B

linsys.C													
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	-9.8100	0	-3.4402e-09	-3.4402e-09	1.8445e-15	0	0	0	-5.4616e-11	0.1380	7.2109e-16	0
2	9.8100	-2.8478e-22	0	-3.6595e-09	-3.6595e-09	-1.7339e-15	0	0	0	-0.1380	5.8098e-11	-7.2109e-16	0
3	5.4456e-11	5.4456e-11	0	0	0	0	0	0	0	0	0	0	0
4	0	5.5511e-12	0	0	0	0	0	0	0	0	0	0	0
5	-5.2296e-12	-5.2295e-12	-1.0000	0	0	0	0	0	0	0	0	0	0
6	1.4969e-18	1.0000	-5.2295e-12	0	0	0	0	0	0	0	0	0	0
7	0	8.3096e-30	1.0000	0	0	0	0	0	0	0	0	0	0
8	5.5511e-12	0	0	0	0	0	0	0	0	0	0	0	0
9	-1.0000	1.4969e-18	-5.2296e-12	0	0	0	0	0	0	0	0	0	0
10	0	-1.0000	0	0	0	0	0	0	0	0	0	0	0
11	1.0000	-2.9029e-23	0	0	0	0	0	0	0	0	0	0	0
12	5.5511e-12	5.5511e-12	0	0	0	0	0	0	0	0	0	0	0
13	1	0	0	0	0	0	0	0	0	0	0	0	0
14	0	1	0	0	0	0	0	0	0	0	0	0	0
15	0	0	1	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	9.0026e-06	0	0	0	0	0	0

Fig.32 Matrice C

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	-6.8995e-17	6.8995e-17	-6.8995e-17	6.8995e-17	-9.9331e-14	0	0	0	1.0000	1.4969e-18	5.2296e-12	0	0
2	6.7628e-17	-6.7628e-17	6.7628e-17	-6.7628e-17	9.8151e-14	0	0	0	-1.4969e-18	1.0000	-5.2295e-12	0	0
3	-0.0096	0.0096	-0.0096	0.0096	-8.2855	0	0	0	0	0	1.0000	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 33 Matrice D

Utilizzando i valori evidenziati nelle immagini riportate avremo come matrici della dinamica di roll:

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

Queste ultime verranno utilizzate per il calcolo della funzione di trasferimento insieme a due coefficienti K1 e K2, relativi rispettivamente a sforzo di controllo ripartito tra i 4 pesi di controllo (altitudine, yaw, pitch e roll) e a spinta dei motori. Essi sono presenti nei blocchi *TorqueTotalToThrustPerMotor* e *ThrustToMotorCommand* del Flight System Control.

```

Command Window
>> Controller.Q2Ts

ans =

      z      yaw      pitch      roll
0.2500  103.5736  -5.6659  -5.6659
0.2500 -103.5736  -5.6659   5.6659
0.2500  103.5736   5.6659   5.6659
0.2500 -103.5736   5.6659  -5.6659

>> Vehicle.Motor.thrustToMotorCommand

ans =

1.5307e+03

```

Fig.34 Costanti moltiplicative K1 e K2

Da MATLAB abbiamo:

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

Ps =

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

Continuous-time transfer function.
```

Fig.35 Command Window di MATLAB

La funzione di trasferimento ha la seguente forma:

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

5.4 Tecniche di controllo lineare

Prima di passare alla sintesi dei controllori verranno presentate brevemente, a livello teorico, le tecniche di controllo utilizzate, ovvero la sintesi in frequenza e la sintesi mediante luogo delle radici.

5.4.1 Sintesi in frequenza

La tecnica della sintesi in frequenza viene utilizzata per studiare la stabilità del sistema e determinare in che modo il sistema influisca su modulo e fase sull'ingresso di riferimento dato per generare la relativa uscita.

Essa viene svolta nel dominio della frequenza dove vengono in nostro aiuto strumenti come i diagrammi di Bode e di Nyquist.

Per effettuare la sintesi di un controllore in frequenza dobbiamo soddisfare due tipologie di specifiche: *univoche* e *lasche*.

Le prime sono chiamate così poiché c'è solo un modo per soddisfarle e sono relative all'errore a regime permanente $\tilde{e}_1(t)$ rispetto ad ingressi a rampa e all'astatismo rispetto a disturbi costanti che agiscono in catena diretta.

Le seconde possono essere soddisfatte in più modi e riguardano i parametri in frequenza in catena chiusa margine di risonanza M_r e banda passante B_3 , che hanno influenza diretta anche sui parametri in frequenza in catena aperta, quali margine di fase m_φ e pulsazione di attraversamento ω_t , e sui parametri nel tempo in catena chiusa, quali sovraelongazione \hat{s} e tempo di salita t_s .

Nello specifico:

- il margine di risonanza M_r rappresenta la differenza tra il valore massimo raggiunto dal modulo della funzione di trasferimento in catena chiusa $W(j\omega)$ e il suo valore in $\omega = 0$;
- la banda passante B_3 è la pulsazione ω dove il modulo della funzione di trasferimento in catena chiusa $W(j\omega)$ diminuisce di 3dB rispetto a $W(j0)$;
- la pulsazione di attraversamento ω_t è la pulsazione in corrispondenza della quale il modulo della funzione di trasferimento in catena diretta $F(j\omega)$ è pari a 0 dB;
- il margine di fase m_φ è il valore della fase pari a $\angle F(j\omega_t) + 180^\circ$;
- la sovraelongazione \hat{s} rappresenta, nel transitorio, di quanto l'uscita 'sorpassa' il riferimento;
- il tempo di salita t_s è il tempo impiegato dall'uscita del sistema per raggiungere il valore di riferimento.

Le relazioni algebriche che legano i parametri tra loro sono:

$$\frac{1 + \hat{s}}{M_r} \approx 0.85$$

$$[B_3]_{rad/s} \cdot [t_s]_s \approx 3 \text{ rad}$$

$$[\omega_t]_{rad/s} \approx 3 \div 5 [B_3]_{Hz}$$

Dopo aver esaurito le specifiche univoche otteniamo un controllore di primo tentativo $\hat{G}(s)$ del tipo:

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

utilizzando successivamente i diagrammi universali di modulo e fase verranno costruite delle reti anticipatrici ed attenuatrici in modo da tale da progettare una rete compensatrice $R(s)$ che generi il controllore complessivo $G(s)$:

$$G(s) = \frac{K_G}{s^k} R(s)$$

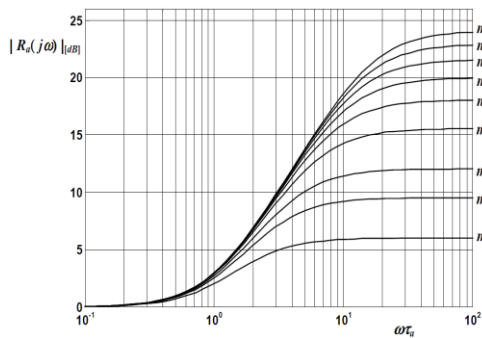


Fig.36 Diagramma del modulo

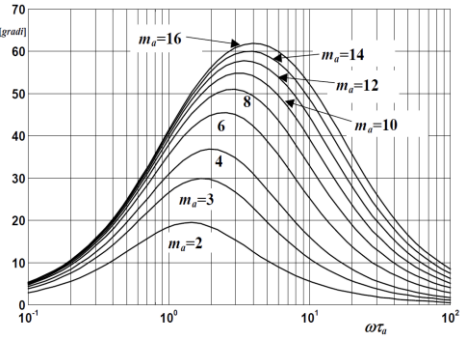


Fig.37 Diagramma della fase

5.4.2 Sintesi con luogo delle radici

Il luogo delle radici è una tecnica di controllo lineare tramite la quale possiamo studiare il posizionamento dei poli della funzione di trasferimento in catena chiusa $W(s)$ del sistema, spostati successivamente sul piano complesso in base al valore che assume il guadagno K della funzione di trasferimento in catena diretta $F(s)$. Ponendo la funzione di trasferimento in catena diretta in forma razionale avremo:

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

con n e m numero di poli e zeri di $F(s)$.

Per un sistema a retroazione unitaria da questa formula otteniamo che $W(s)$ è pari a:

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

Il luogo delle radici sarà espresso tramite le radici del polinomio caratteristico, che rappresenta il luogo geometrico dei punti del piano complesso percorso dai poli della funzione di trasferimento in catena chiusa $W(s)$ del sistema al variare del coefficiente di guadagno K ($K \in (-\infty, +\infty)$).

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

Per la sintesi basta solo tracciare qualitativamente la posizione delle radici del polinomio caratteristico appena espresso.

Per effettuare il tracciamento bisogna seguire alcune regole fondamentali:

- i valori delle radici sono sempre o puramente reali o complesse coniugate per tutti i valori di $K \in (-\infty, +\infty)$ poiché il luogo delle radici è simmetrico rispetto all'asse $\text{Re}[s]$ del piano complesso;
- tutto l'asse reale del piano complesso appartiene al luogo delle radici;
- con $K > 0$ e $K < 0$ indichiamo il luogo positivo e il luogo negativo come suddivisione del luogo delle radici totale;
- al luogo positivo appartengono quelle porzioni di asse che lasciano alla propria destra un numero dispari di poli e/o zeri di $F(s)$, le restanti porzioni appartengono al luogo negativo;
- il luogo delle radici è formato da n rami dove ciascuno rappresenta il movimento di un singolo polo di $W(s)$ al variare di K sul piano complesso;
- con $K \rightarrow 0$ i poli in catena chiusa di $W(s)$ coincidono con i poli in catena aperta $F(s)$;
- con $K \rightarrow \infty$ abbiamo m rami del luogo che convergono sugli zeri di $F(s)$ e altri $n-m$ che divergono a ∞ lungo $n-m$ semirette chiamate asintoti del luogo che partono tutte dal centro degli asintoti S_0 che ha espressione:

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

Definiamo punti singolari del luogo delle radici quei valori \bar{s} tali da essere radici multiple dell'equazione $f(s, K) = 0$, ossia:

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

- Condizione di modulo:

$$|K| = \frac{\prod_{i=1}^n |s - p_i|}{\prod_{i=1}^m |s - z_i|}$$

Per ogni punto s giacente sul luogo delle radici, il modulo di K ad esso associato è pari al rapporto tra il prodotto dei moduli di tutti i vettori $s - p_i$ ed il prodotto dei moduli di tutti i vettori $s - z_i$.

I moduli rappresentano i vettori che sul piano complesso congiungono rispettivamente l' i -esimo polo o zero al punto del luogo s considerato.

- Condizione di fase:

$$\sum_{i=1}^n \angle(s - p_i) - \sum_{i=1}^m \angle(s - z_i) = \begin{cases} (2h + 1)\pi & \text{per } K > 0 \\ 2h\pi & \text{per } K < 0 \end{cases} \text{ con } h \in \mathbb{Z}$$

le condizioni sono concettualmente analoghe espresse in termini di fase piuttosto che in modulo.

La tecnica del luogo delle radici rispetto alla sintesi in frequenza, che può essere applicata solo in sistemi a fase minima, può essere applicata a qualsiasi tipo di sistema purché si conosca esattamente il modello matematico del processo $P(s)$ da controllare, requisito che nella sintesi in frequenza è sostituito dalla conoscenza dei diagrammi di Bode in modulo e fase.

5.5 Sintesi dei controllori in frequenza

5.5.1 Controllore per riferimenti costanti

Andiamo dapprima a definire le specifiche univoche e lasche da rispettare. Successivamente saranno progettate le reti compensatrici che soddisfano tali specifiche di sistema.

Specifiche univoche

- sistema di tipo 1
- errore a regime permanente $|\tilde{e}_1(t)| \leq 0.03$

Specifiche lasche

- $M_r \leq 2 \text{ dB} \Leftrightarrow m_\varphi \geq 48^\circ \Leftrightarrow \hat{s} \leq 0.7 = 70\%$
- $B_3 \approx 1 \text{ Hz} \Leftrightarrow \omega_t \approx 4 \text{ rad/s} \Leftrightarrow t_s \approx 0.5 \text{ s}$

La prima specifica univoca da esaudire richiede che la funzione di trasferimento in catena diretta $F(s) = G(s)P(s)$ abbia almeno un polo in $s = 0$.

Dall'espressione $P(s)$ precedentemente trovata notiamo come questa specifica è soddisfatta.

Il controllore di primo tentativo, perciò, ha la seguente forma:

$$\hat{G}(s) = K_G$$

La seconda specifica univoca chiede che l'errore a regime permanente $|\tilde{e}_1(t)| \leq 0.03$. Esso per un ingresso polinomiale di grado k fatto da un sistema di tipo k (nel nostro caso $k=1$) è uguale a:

$$\tilde{e}_k(t) = \frac{k_d^2}{K_F}$$

dove k_d è il coefficiente di retroazione. Nel nostro caso è uguale a 1 dato che il sistema è a retroazione unitaria.

Quindi avremo:

$$|\tilde{e}_1(t)| = \left| \frac{1}{K_F} \right| = \frac{1}{K_G K_F} \leq 0.03 \Leftrightarrow K_G \geq 0.018$$

arbitrariamente scegliamo $K_G = 0.02$. La funzione di trasferimento in catena diretta sarà:

$$\hat{F}(s) = \hat{G}(s)P(s) = \frac{72.88}{s(s + 2.172)}$$

Ora aiutandoci con i diagrammi universali di fase e modulo andiamo a progettare la rete anticipatrice $R_a(s)$ e la rete attenuatrice $R_i(s)$. Esse ci permettono di ‘correggere’ la funzione di trasferimento e di soddisfare le specifiche lasche. Utilizziamo arbitrariamente le seguenti specifiche nella progettazione del controllore di primo tentativo:

$R_a(s)$:

$$\omega_a = 4 \quad m_a = 16$$

$$t_s \approx 0.48 \text{ s}$$

$R_i(s)$:

$$\omega_i = 0.005 \quad m_i = 5$$

$$\hat{s} \approx 8\%$$

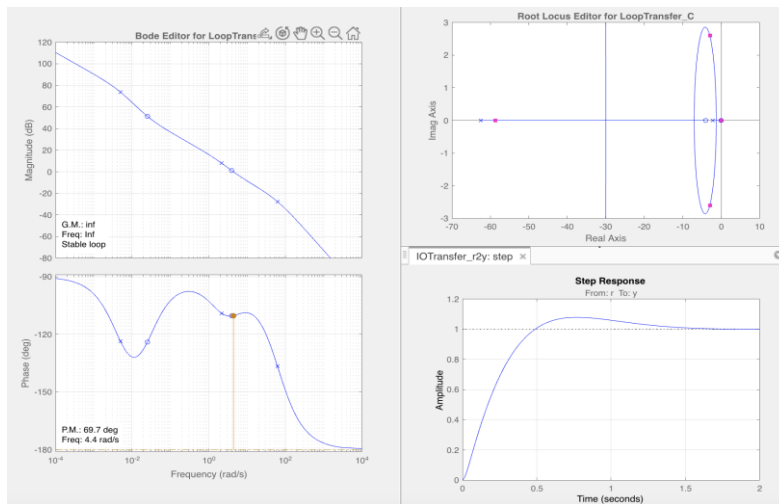


Fig.38 Grafici ControlSystemDesign della $F(s)$ di primo tentativo

Dai diagrammi di Bode di modulo e fase notiamo come il margine di fase m_ϕ è pari a 69.7° , in linea con i vincoli imposti. Analogamente anche la pulsazione di attraversamento ω_t , che si attesta intorno ai 4.4 rad/s.

Soddisfatte queste specifiche andiamo ad implementare il controllore appena progettato in ambiente *Simulink*.

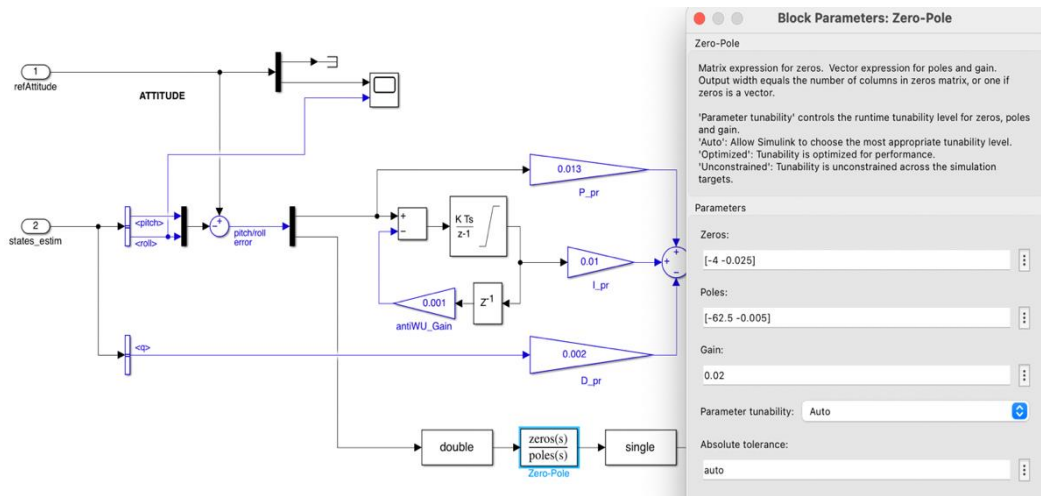


Fig.39 Blocco Attitude con implementazione del controllore

All'interno del progetto *absQuadcopter*, e più nello specifico nel blocco *Attitude* precedentemente descritto, andiamo a modificare la parte relativa alla dinamica di roll inserendo dei blocchi *Cast to Double* e *Cast to Single* a destra e sinistra del blocco *Zero-Pole*.

I due blocchi *Cast* sono dei convertitori di valori numerici, i quali garantiscono che il 'data type' degli ingressi e delle uscite del controllore sia persistente con quello dei dati che vengono scambiati lungo il bus di sistema con le restanti componenti del progetto.

È stato aggiunto un blocco *Scope* per visualizzare il grafico del roll con il suo riferimento.

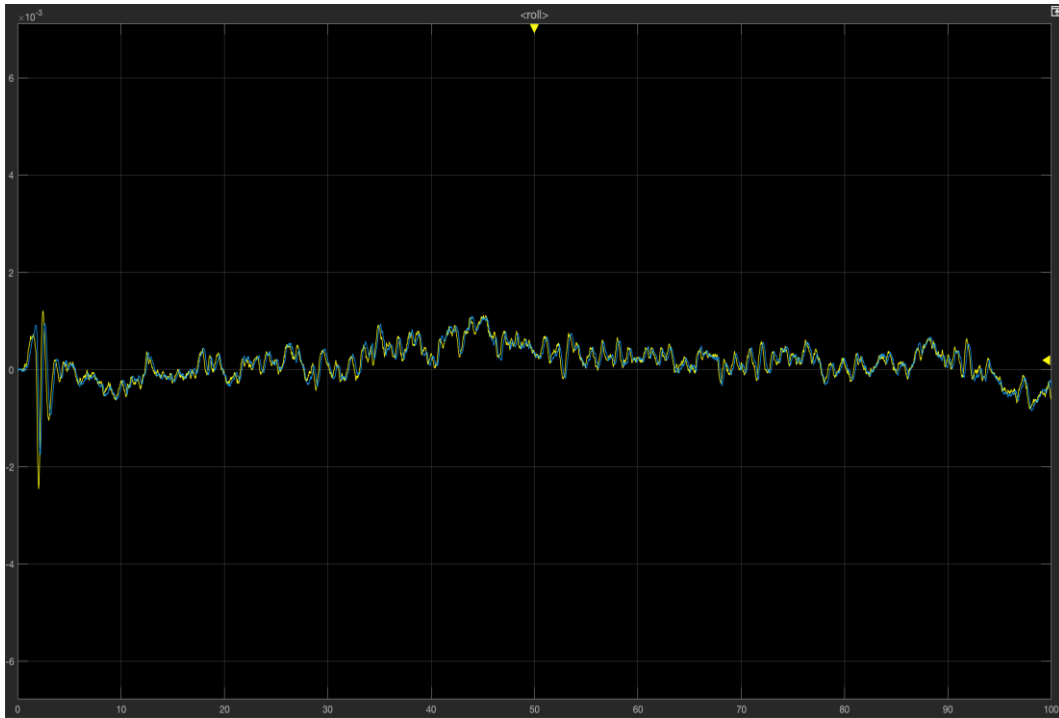


Fig.40 Grafico simulazione del controllore in frequenza di primo tentativo, ingressi costanti, tracciato del Roll con il suo riferimento

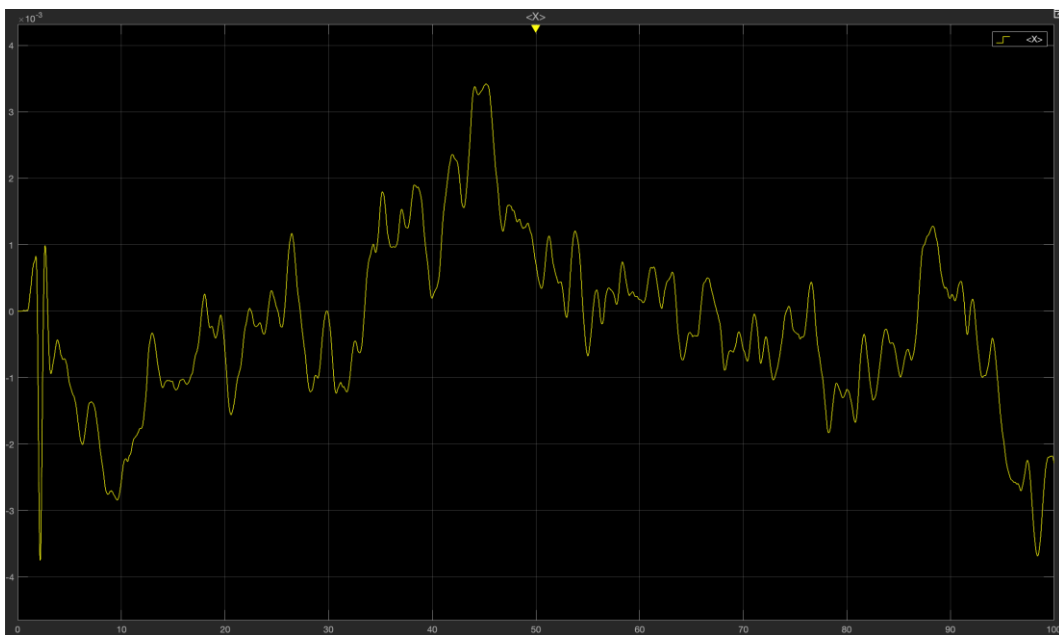


Fig.41 Grafico simulazione del controllore in frequenza di primo tentativo, ingressi costanti, tracciato della X

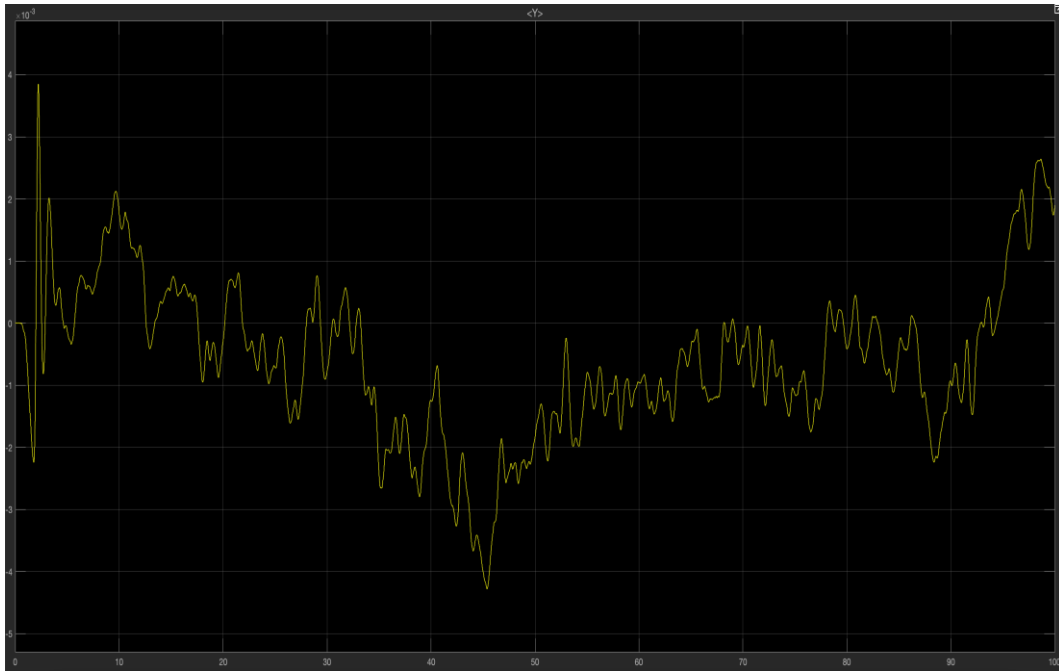


Fig.42 Grafico simulazione del controllore in frequenza di primo tentativo, ingressi costanti, tracciato della Y

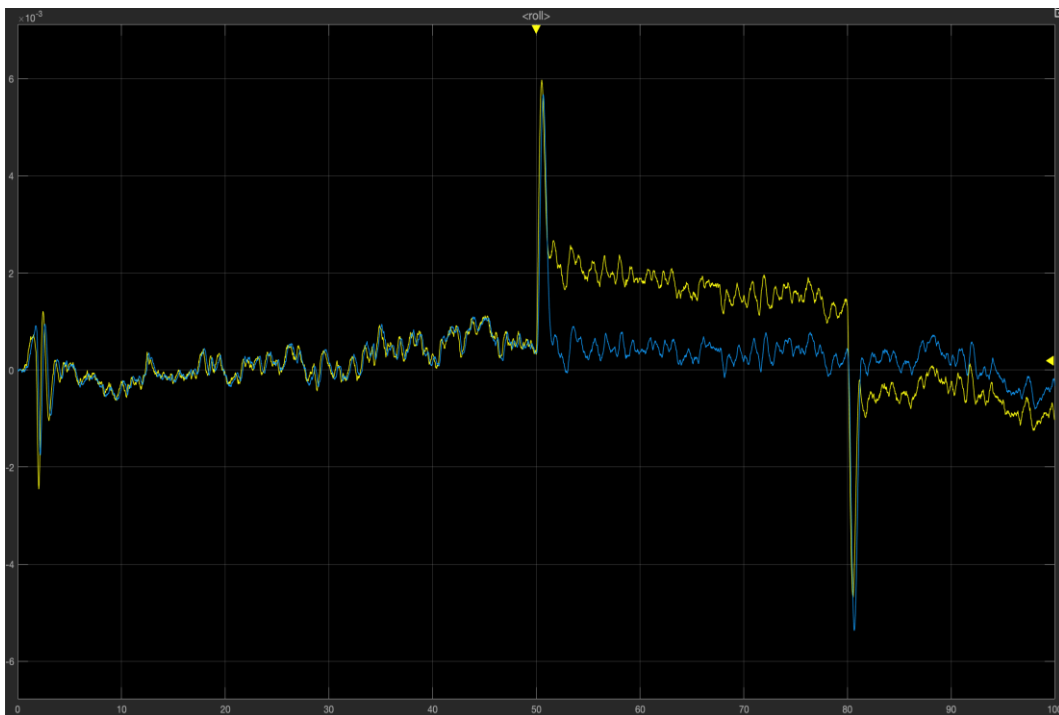


Fig.43 Grafico simulazione del controllore in frequenza di primo tentativo, ingressi variabili, tracciato del Roll con il suo riferimento

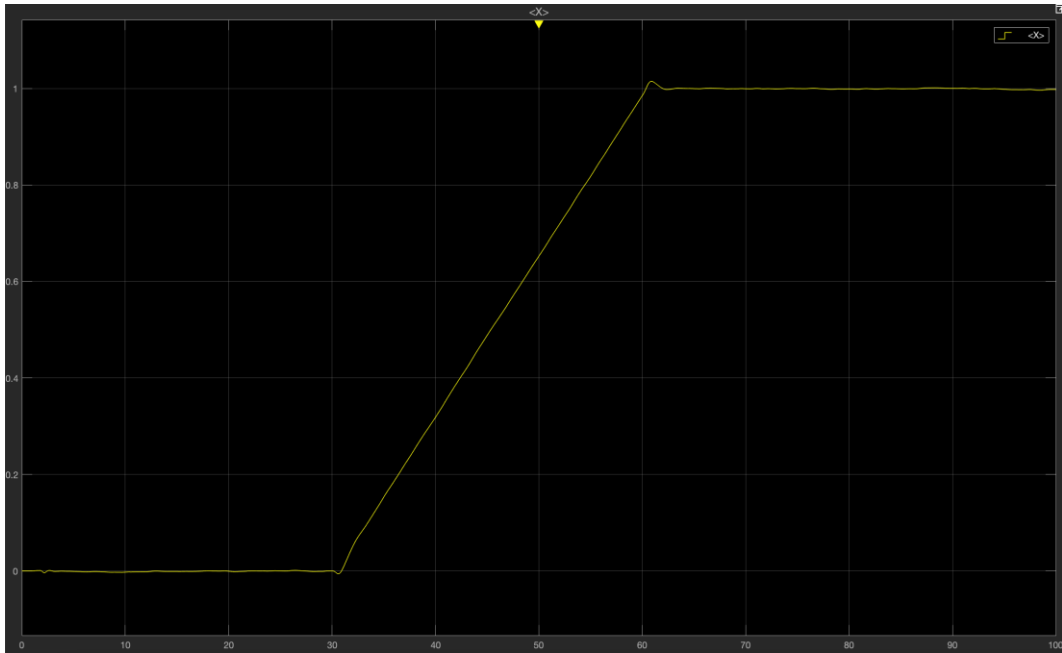


Fig.44 Grafico simulazione del controllore in frequenza di primo tentativo, ingressi variabili, tracciato della X

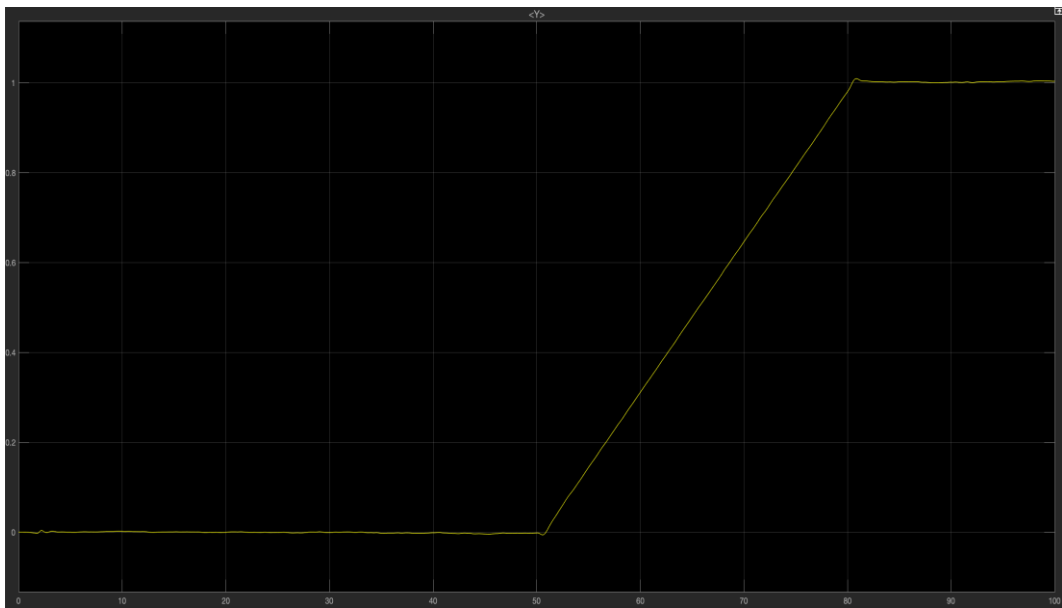


Fig.45 Grafico simulazione del controllore in frequenza di primo tentativo, ingressi variabili, tracciato della Y

Possiamo notare come con questo controllore abbiamo ottimi valori di tempo di salita e sovraelongazione.

Per i riferimenti costanti il controllore riesce a seguire ottimamente il riferimento (fig.40) e la posizione X-Y risulta controllata (fig 41,42).

Diverso è il risultato per i riferimenti variabili dove la posizione X-Y risulta essere controllata ottimamente (fig.e 44, 45), ma il riferimento dell'angolo di roll non viene seguito in maniera efficace e costante (fig.43).

Questo può essere causato dal fatto che il sistema non sia abbastanza 'pronto' nel reagire agli stimoli esterni.

Tale caratteristica risulta fondamentale nei riferimenti variabili.

Per questo progettiamo un controllore di secondo tentativo che risolva la suddetta problematica.

5.5.2 Controllore per riferimenti variabili

Per progettare il secondo controllore adatto ai riferimenti variabili sfruttiamo i risultati trovati per le specifiche univoche utilizzando quindi $K_G = 0.02$.

Per quanto riguarda le specifiche lasche scegliamo dei nuovi valori che ci permettono di realizzare nuove reti compensatrici:

$R_a(s)$:

$$\omega_a = 2.5 \quad m_a = 30$$

$$t_s \approx 0.6 \text{ s}$$

$R_i(s)$:

$$\omega_i = 0.08 \quad m_i = 10$$

$$\hat{s} \approx 14\%$$

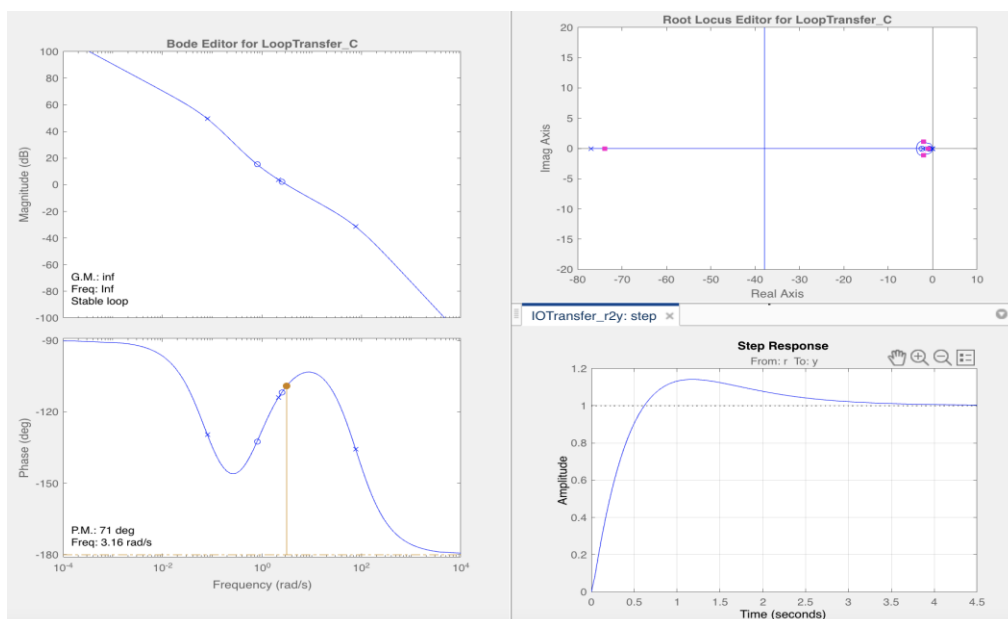


Fig.46 Grafici ControlSystemDesign della $F(s)$ di secondo tentativo

Notiamo come il margine di fase m_φ e la pulsazione di attraversamento ω_t sono pari a 71° e $3,16$ rad/s, rientrando nelle specifiche. Sovraelongazione \hat{s} e tempo di salita t_s sono di poco aumentati e hanno valore rispettivamente di 14% e 0.6 s.

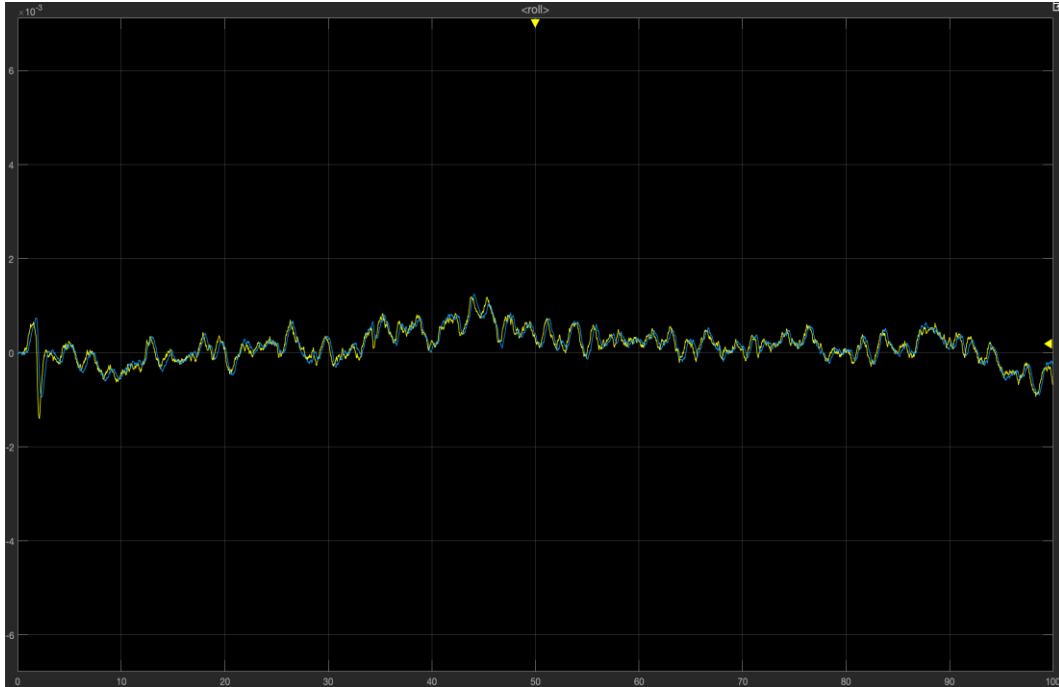


Fig.47 Grafico simulazione del controllore in frequenza di secondo tentativo, ingressi costanti, tracciato del Roll con il suo riferimento

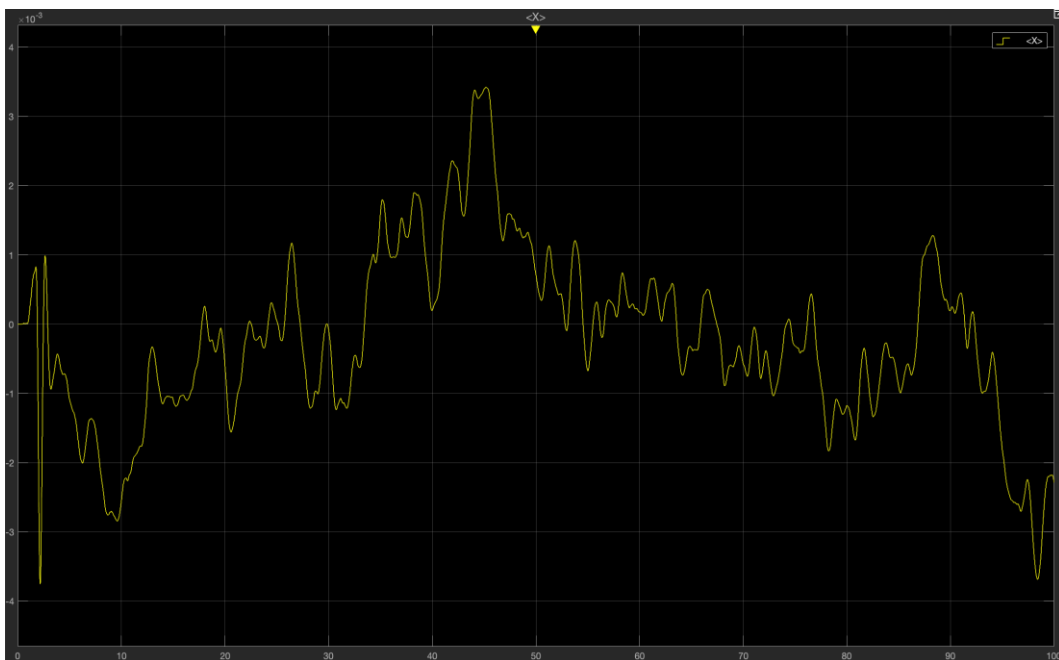


Fig.48 Grafico simulazione del controllore in frequenza di secondo tentativo, ingressi costanti, tracciato della X

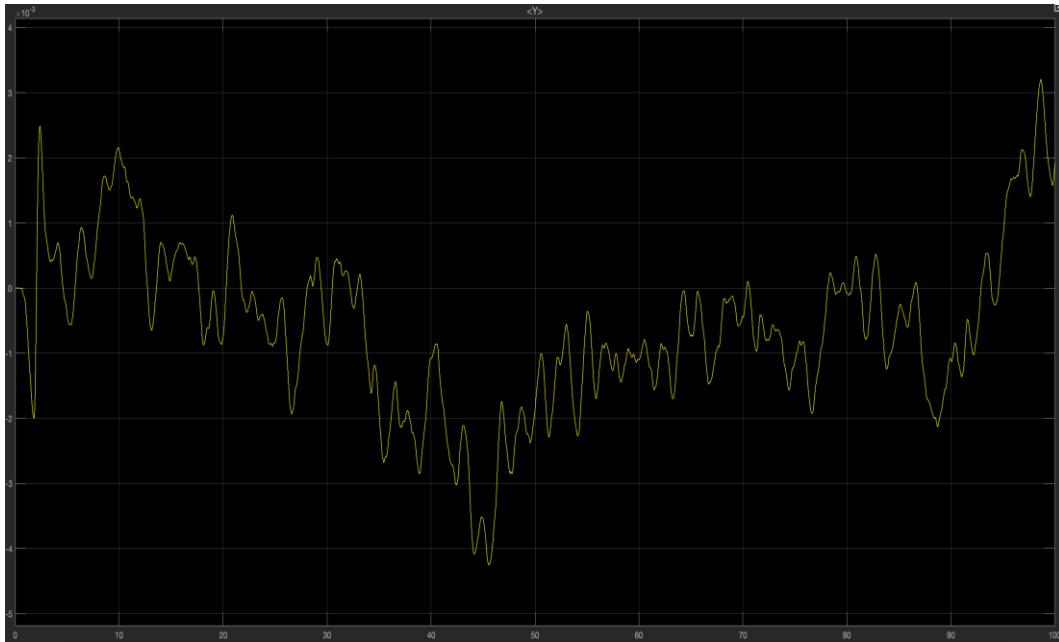


Fig.49 Grafico simulazione del controllore in frequenza di secondo tentativo, ingressi costanti, tracciato della Y

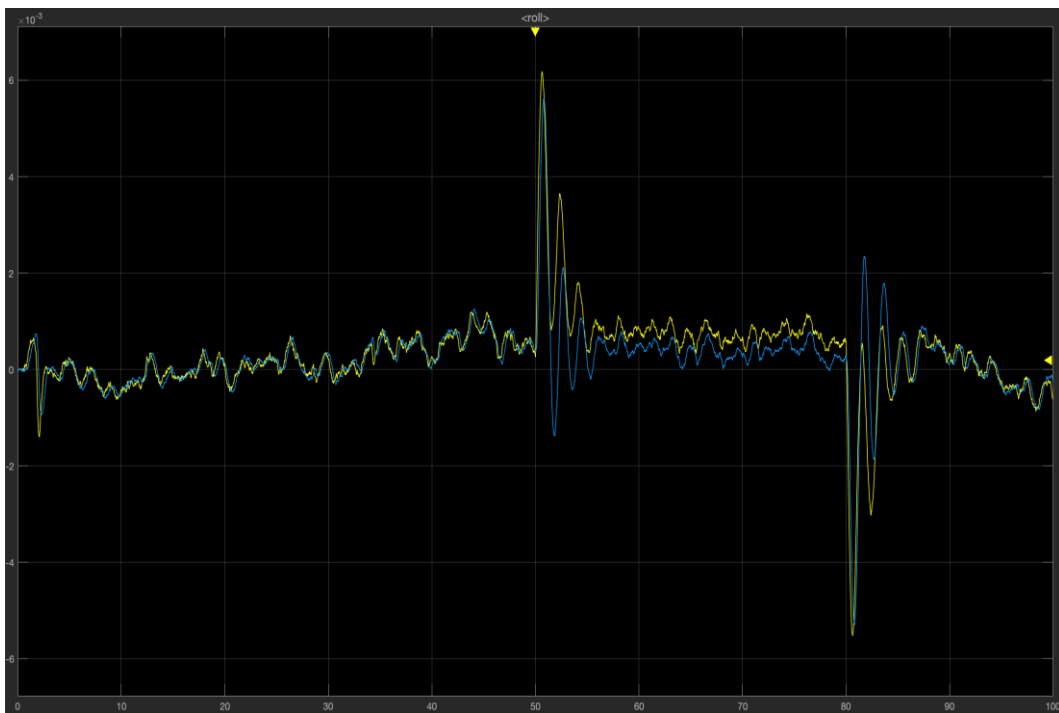


Fig.50 Grafico simulazione del controllore in frequenza di secondo tentativo, ingressi variabili, tracciato del Roll con il suo riferimento

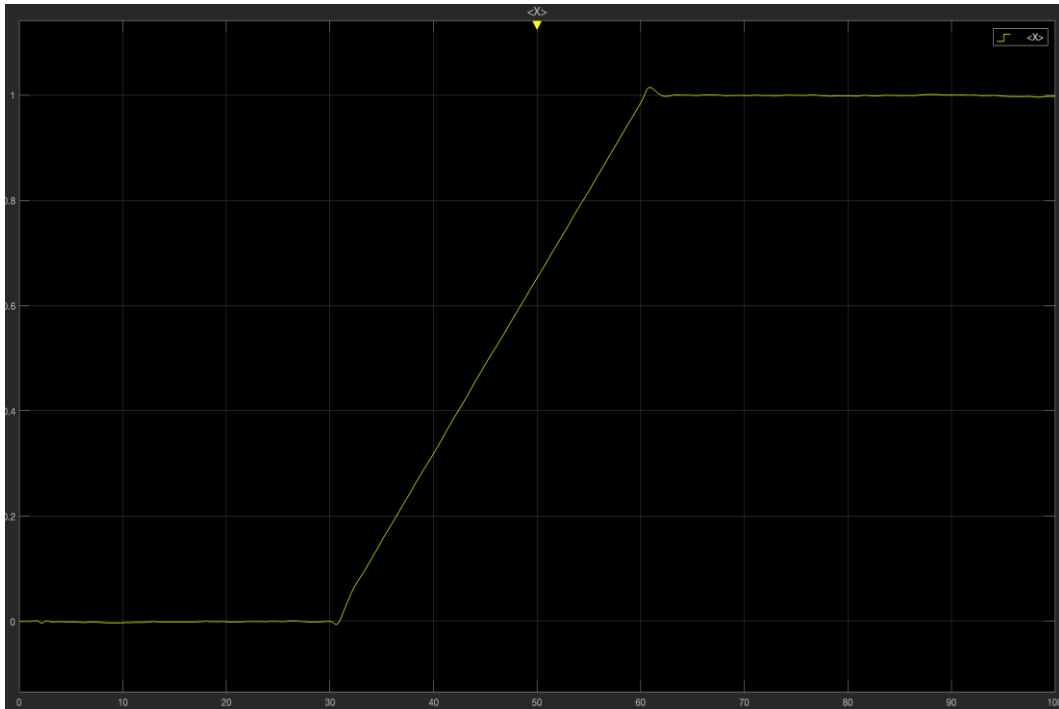


Fig.51 Grafico simulazione del controllore in frequenza di secondo tentativo, ingressi variabili, tracciato della X

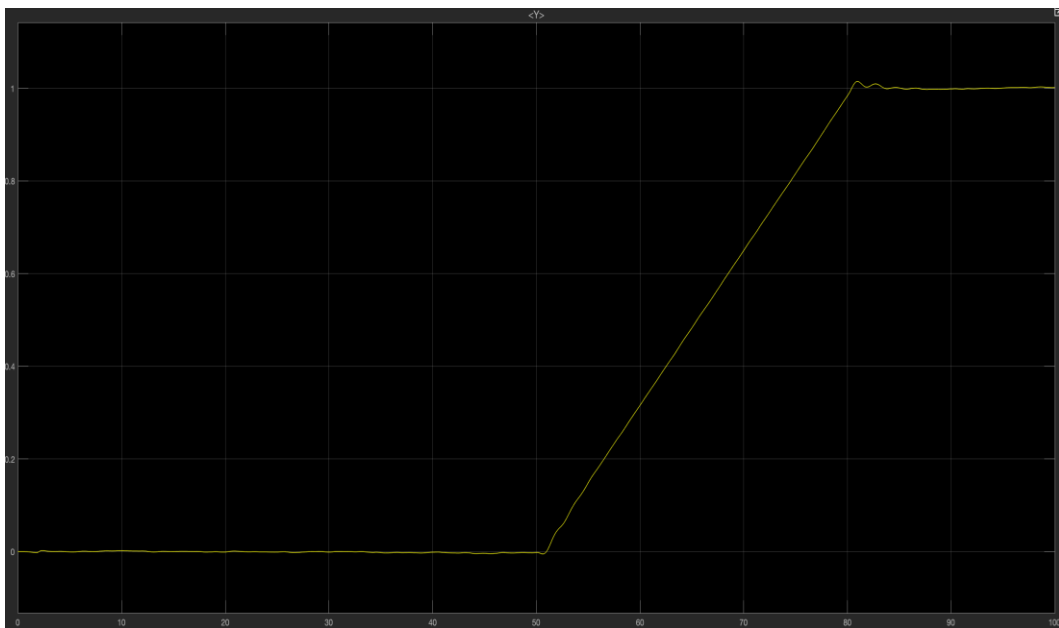


Fig.52 Grafico simulazione del controllore in frequenza di secondo tentativo, ingressi variabili, tracciato della Y

Dai grafici notiamo come le posizioni sia per i riferimenti costanti che per quelli variabili risultano controllate (fig.e 48, 49, 51, 52).

Per il controllo dell'angolo di roll abbiamo che, per i riferimenti costanti, presenta risultati simili al controllore di primo tentativo (fig.47).

Invece, per i riferimenti variabili, notiamo un miglioramento dato che il controllore riesce a seguire ottimamente il riferimento del roll (fig.50).

5.6 Sintesi tramite luogo delle radici

Utilizziamo adesso questa tecnica di controllo lineare già introdotta in precedenza per progettare dei controllori per il controllo dell'angolo di roll.

L'obiettivo è quello di avere dei risultati simili a quelli ottenuti con la precedente tecnica.

5.6.1 Controllore di primo tentativo

Per la sintesi mediante luogo delle radici utilizziamo le medesime specifiche univoche presenti per la sintesi in frequenza.

Le specifiche lasche invece riguardano la parte reale dei poli del sistema in catena chiusa $W(s)$ per la quale viene definito un limite superiore. Nello specifico:

Specifiche univoche

- sistema di tipo 1
- errore a regime permanente $|\tilde{e}_1(t)| \leq 0.03$

Specifiche lasche

- i poli in catena chiusa devono avere $Re(s) \leq -1$

Per le specifiche univoche utilizziamo le considerazioni esposte per la sintesi in frequenza nella quale, con la condizione relativa al sistema di tipo 1, utilizzeremo un controllore del tipo $\hat{G}(s) = K$.

Per l'errore a regime permanente rispetto a ingressi polinomiali di primo grado abbiamo che:

$$|\tilde{e}_1(t)| = \left| \frac{1}{K_F} \right| = \frac{1}{K_G K_F} \leq 0.03 \Leftrightarrow K_G = \lim_{s \rightarrow 0} G(s) = K \cdot R(0) \geq 0.019868$$

Dove $R(0)$ è la rete compensatrice $R(s)$ considerata in $s=0$ tale che

$$G(s) = K \cdot R(s)$$

La funzione $R(s)$, relativa al soddisfacimento delle specifiche lasche, verrà definita a breve.

Per adesso consideriamo solo $\hat{G}(s) = K$, da cui:

$$\hat{F}(s) = \hat{G}(s)P(s) = \frac{3644K}{s(s + 2.172)}$$

Per quanto riguarda le specifiche lasche notiamo come il processo da controllare $P(s)$ presenta già uno scarto poli-zeri pari a due e la $G(s)$ che andremo a progettare dovrà essere caratterizzato da una funzione di trasferimento propria ($n \geq m$).

Sintetizziamo perciò un controllore che mantenga uno scarto poli-zeri uguale a quello della funzione $P(s)$ in modo tale da non complicare maggiormente il modello del sistema.

Definiamo una rete compensatrice $R(s)$ con 2 zeri e 2 poli del tipo:

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

Scegliamo adesso i valori dei poli e degli zeri in base al posizionamento desiderato del centro degli asintoti S_0 nel piano complesso.

Scegliamo una posizione di S_0 in un punto s con $Re(s) < -15$ in modo tale da verificare la seguente condizione:

$$S_0 = \frac{\sum_{i=1}^n p_i - \sum_{i=1}^m z_i}{n - m} = \frac{-0 - 2.172 - p_1 - p_2 + z_1 + z_2}{2} < -15$$

Ora posizioniamo in maniera arbitraria i poli e gli zeri della rete compensatrice $R(s)$ nella posizione più opportuna per soddisfare le specifiche stilate. Determiniamo:

$$z_1 = -0.6 \quad z_2 = -2.5 \quad p_1 = -0.03 \quad p_2 = -42$$

Il centro degli asintoti è posto in $S_0 = -20.551$ e la rete compensatrice è pari a:

$$R(s) = \frac{(s + 0.6)(s + 2.5)}{(s + 0.03)(s + 42)}$$

Per trovare la costante di guadagno K poniamo $s=0$ in $R(s)$ e abbiamo:

$$R(0) = 1.1904 \rightarrow K \geq \frac{0.019868}{R(0)} \rightarrow K = 0.018$$

Scegliamo arbitrariamente $K = 0.02$ e otteniamo:

$$G(s) = K \cdot R(s) = \frac{0.02(s + 0.6)(s + 2.5)}{(s + 0.03)(s + 42)}$$

Per visualizzare il luogo delle radici della funzione $F(s)=G(s)P(s)$ utilizziamo sul *Workspace* di MATLAB la funzione *ControlSystemDesign*.

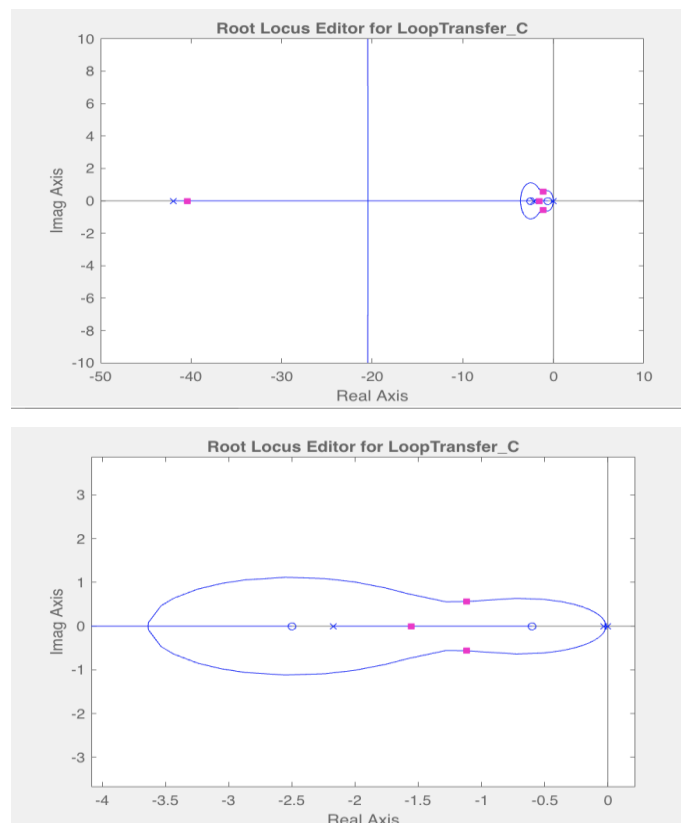


Fig.53 Grafici ControlSystemDesign dei luoghi delle radici della funzione $F(s)$ di primo tentativo

```

>> pole(ws)

ans =

-40.4134 + 0.0000i
-1.5537 + 0.0000i
-1.1174 + 0.5643i
-1.1174 - 0.5643i

```

Fig.54 Poli della funzione in catena aperta $W(s)$

Nel calcolo dei poli della funzione in catena aperta $W(s)$ possiamo notare di avere due poli puramente reali (modi aperiodici), di cui il primo pari a -40.4134 , il quale contributo tende ad annullarsi più velocemente, e due poli complessi, coniugati (modi aperiodici) dove il modulo delle parti immaginarie e reali della coppia è legata alla frequenza delle oscillazioni e alla velocità con cui queste tendono ad annullarsi.

Ecco i grafici che mostrano l'andamento e l'efficienza del controllore appena introdotto.

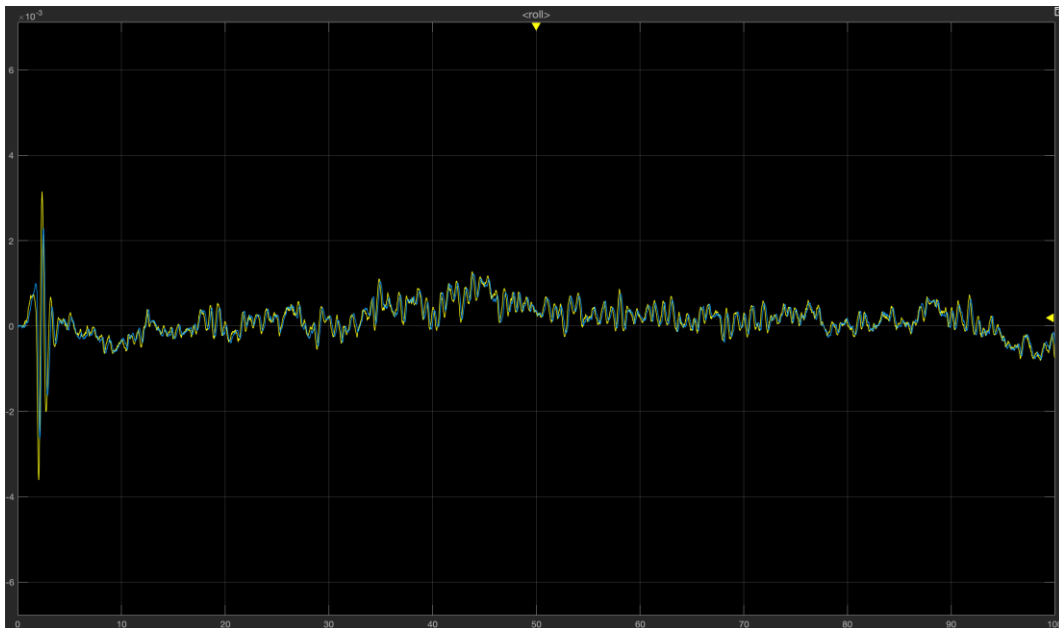


Fig.55 Grafico simulazione del controllore con luogo delle radici di primo tentativo, ingressi costanti, tracciato del Roll con il suo riferimento

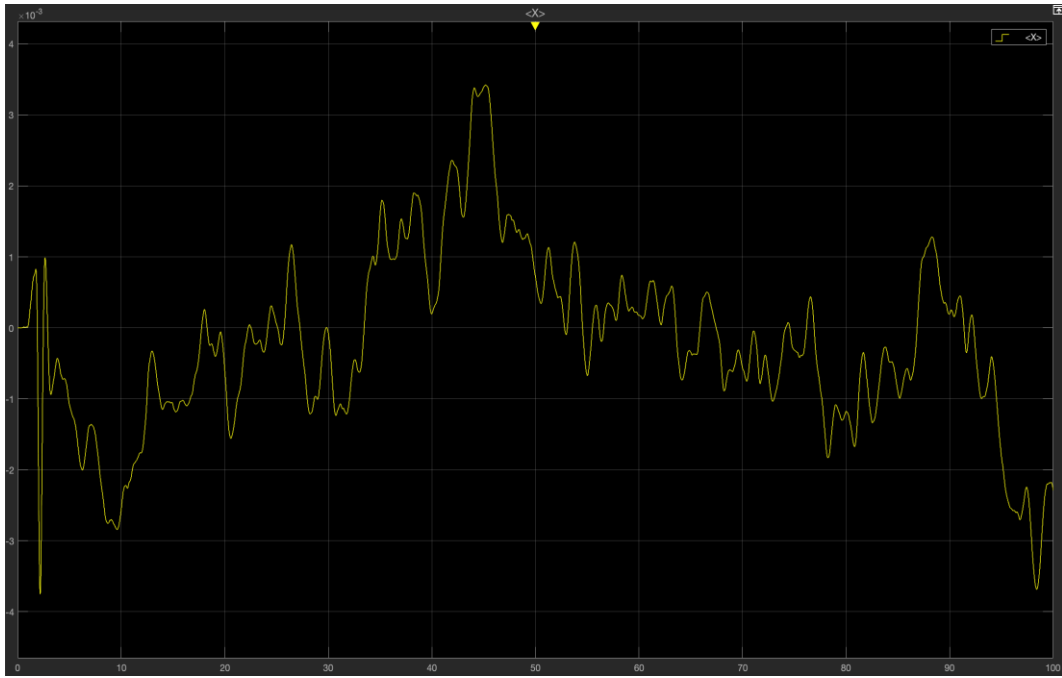


Fig.56 Grafico simulazione del controllore con luogo delle radici di primo tentativo, ingressi costanti, tracciato della X

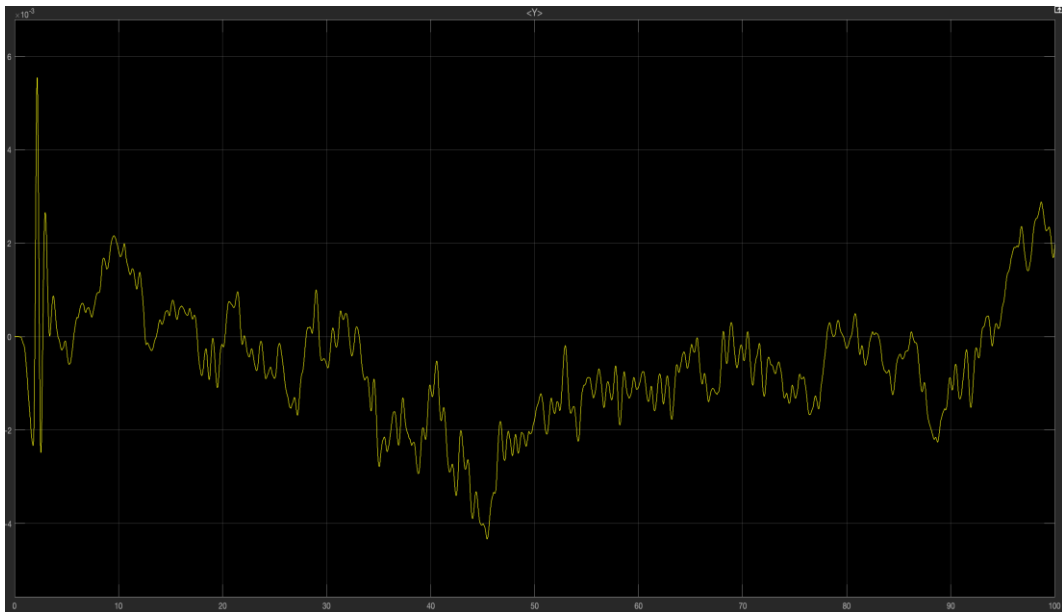


Fig.57 Grafico simulazione del controllore con luogo delle radici di primo tentativo, ingressi costanti, tracciato della Y

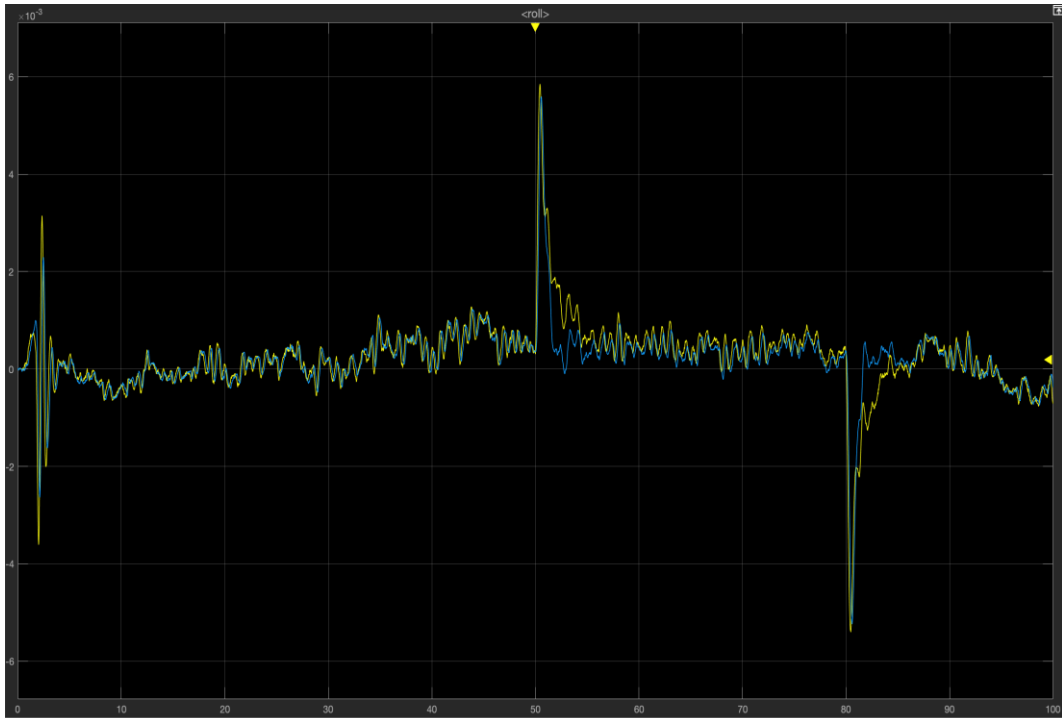


Fig.58 Grafico simulazione del controllore con luogo delle radici di primo tentativo, ingressi variabili, tracciato del Roll con il suo riferimento

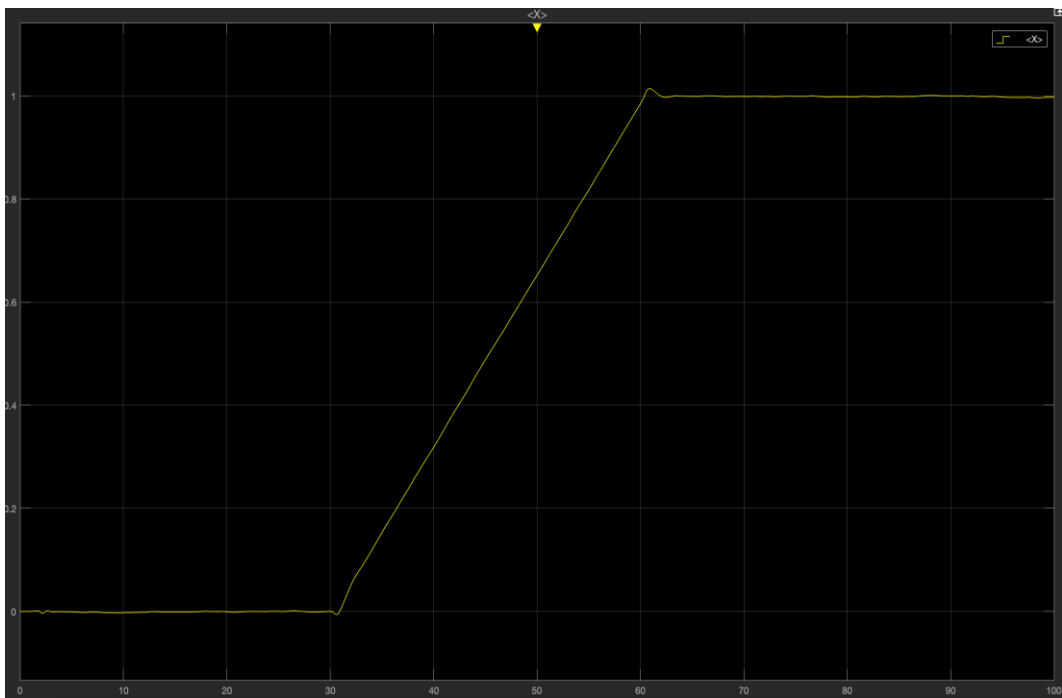


Fig.59 Grafico simulazione del controllore con luogo delle radici di primo tentativo, ingressi variabili, tracciato della X

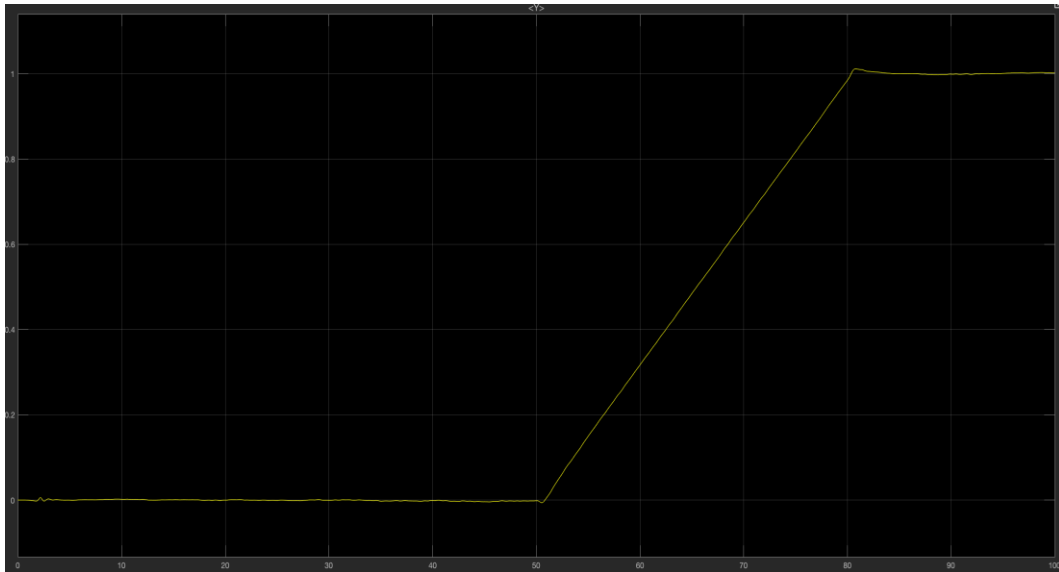


Fig.60 Grafico simulazione del controllore con luogo delle radici di primo tentativo, ingressi variabili, tracciato della Y

Dai grafici notiamo come questo tipo di controllore ha un controllo ottimale della posizione X-Y sia per i riferimenti costanti che per quelli variabili.

Inoltre, segue in maniera efficace il riferimento dell'angolo di roll sia per i riferimenti costanti che per i riferimenti variabili.

L'unico problema è dato dalle troppe oscillazioni che possiamo notare in fig.55 e fig.58.

La causa principale è da ricondurre al valore della sovralongazione, che si attesta intorno al 17,8%, la quale comporta delle oscillazioni non trascurabili.

Studiamo ora un controllore che 'sacrifichi' parte del tempo di salita in favore di una diminuzione della sovralongazione, in modo tale da alleviare le oscillazioni.

5.6.2 Controllore di secondo tentativo

Per sintetizzare un controllore di secondo tentativo tramite luogo delle radici possiamo utilizzare i risultati riguardanti le specifiche univoche del primo controllore.

Per le specifiche lasche invece utilizziamo delle condizioni più stringenti.

Specifiche lasche:

- i poli in catena chiusa devono avere $Re(s) \leq -0.5$

Definiamo ora una rete compensatrice $R(s)$ con 2 poli e 2 zeri e impostiamo in maniera arbitraria il centro degli asintoti S_0 in:

$$S_0 = \frac{\sum_{i=1}^n p_i - \sum_{i=1}^m z_i}{n - m} = \frac{-0 - 2.172 - p_1 - p_2 + z_1 + z_2}{2} < -20$$

Scegliamo:

$$z_1 = -0.4 \quad z_2 = -2.7 \quad p_1 = -0.02 \quad p_2 = -55$$

con $S_0 = -27.046$ e la rete compensatrice $R(s)$:

$$R(s) = \frac{(s + 0.4)(s + 2.7)}{(s + 0.02)(s + 55)}$$

Troviamo la costante di guadagno K ponendo $s=0$:

$$R(0) = 0.98181 \rightarrow K \geq \frac{0.019868}{R(0)} \rightarrow K = 0.02023$$

Scegliamo arbitrariamente $K = 0.022$ e otteniamo:

$$G(s) = K \cdot R(s) = \frac{0.022(s + 0.4)(s + 2.7)}{(s + 0.02)(s + 55)}$$

Per visualizzare il luogo delle radici della funzione $F(s)=G(s)P(s)$ utilizziamo sul *Workspace* di MATLAB la funzione *ControlSystemDesign*.

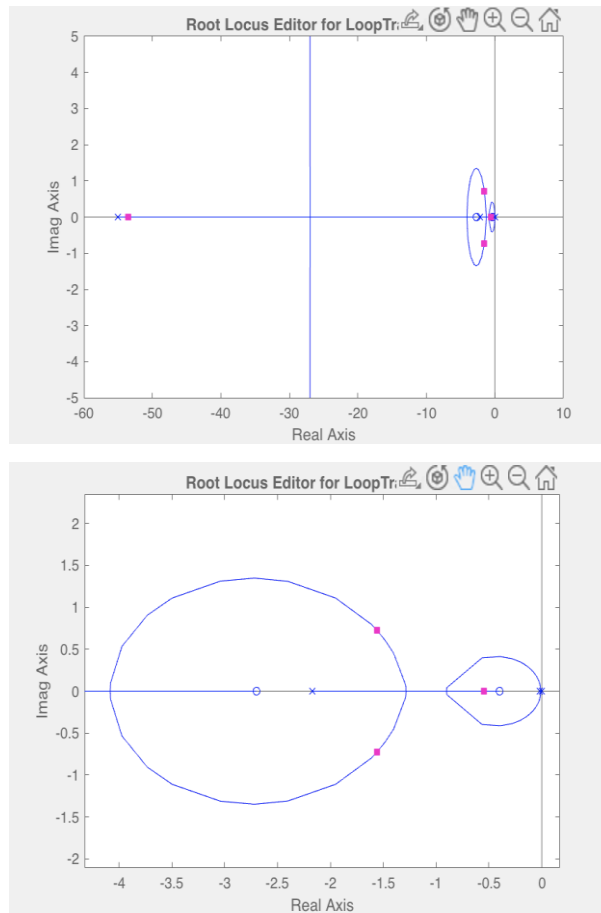


Fig.61 Grafici ControlSystemDesign dei luoghi delle radici della funzione $F(s)$ di secondo tentativo

```
>> pole(Ws)

ans =

-53.5282 + 0.0000i
-1.5576 + 0.7226i
-1.5576 - 0.7226i
-0.5486 + 0.0000i
```

Fig.62 Poli della funzione in catena aperta $W(s)$

In catena aperta abbiamo due poli puramente reali, uno pari a -0.5486, che è il polo dominante ed ha una dinamica molto lenta, e l'altro pari a -53.5282, che controbilancia il primo e permette di rendere il sistema stabile, avendo una dinamica molto veloce.

I due poli complessi coniugati rispetto a quelli del primo controllore ci permettono di smorzare più le oscillazioni, poiché l'ampiezza delle stesse decresce esponenzialmente in modo più rapido.

Ecco i grafici che mostrano l'efficienza del controllore:

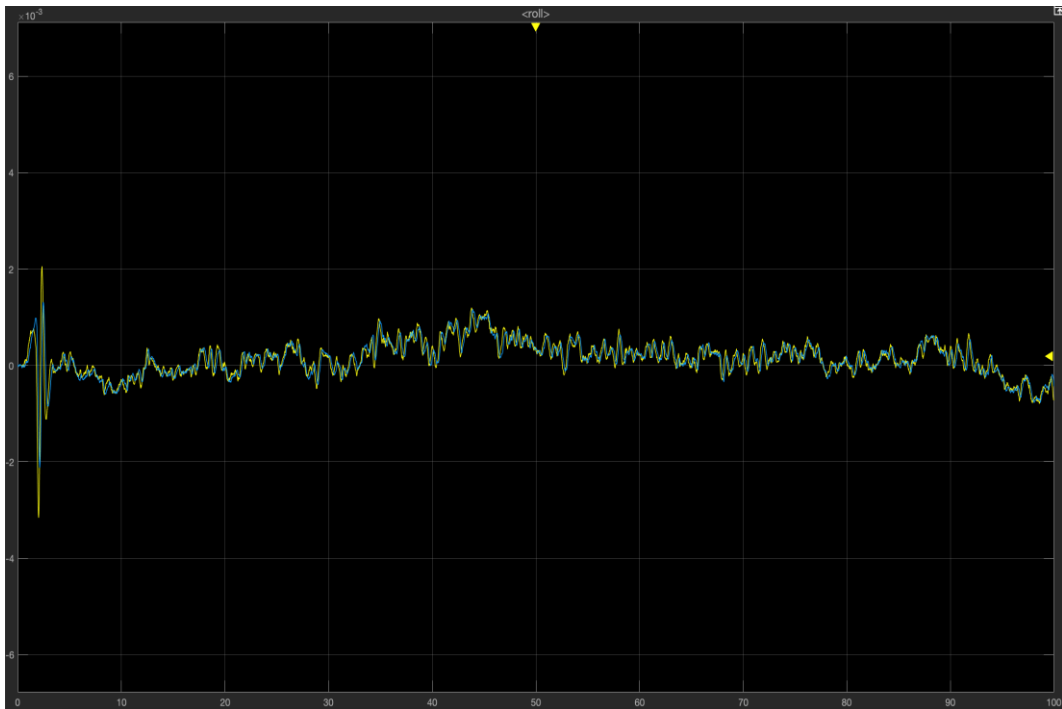


Fig.63 Grafico simulazione del controllore con luogo delle radici di secondo tentativo, ingressi costanti, tracciato del Roll con il suo riferimento

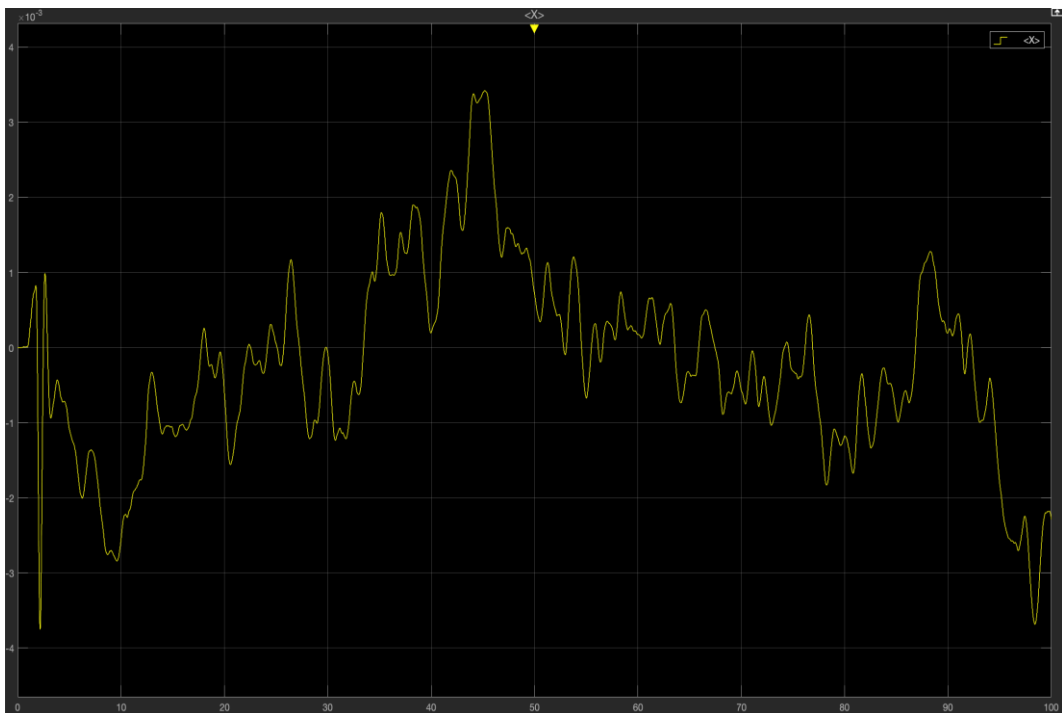


Fig.64 Grafico simulazione del controllore con luogo delle radici di secondo tentativo, ingressi costanti, tracciato della X

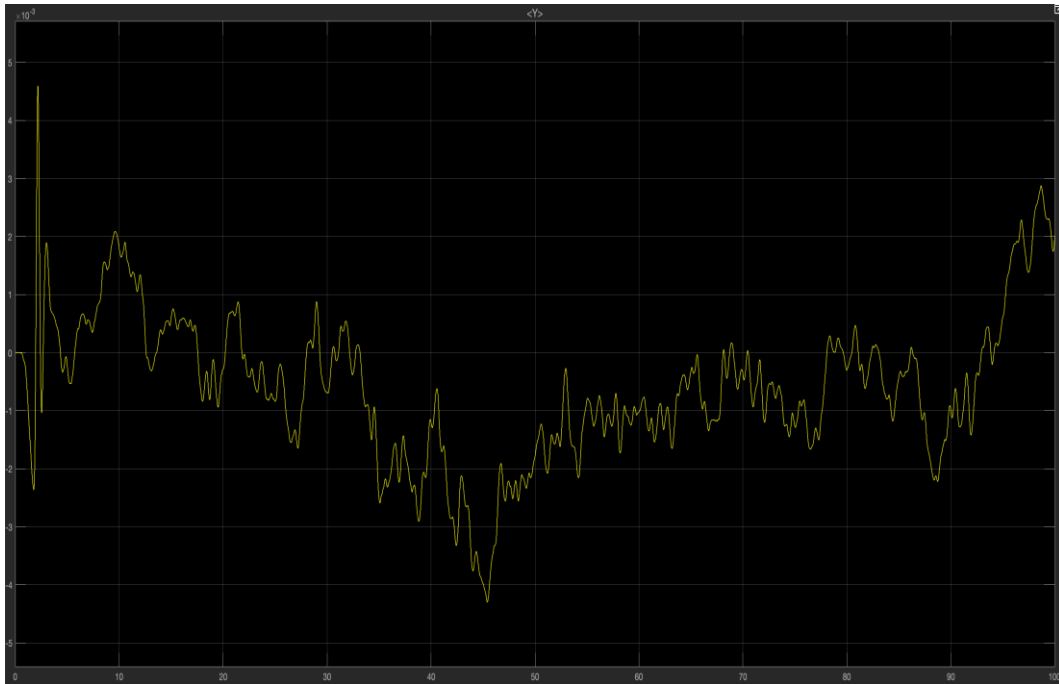


Fig.65 Grafico simulazione del controllore con luogo delle radici di secondo tentativo, ingressi costanti, tracciato della Y

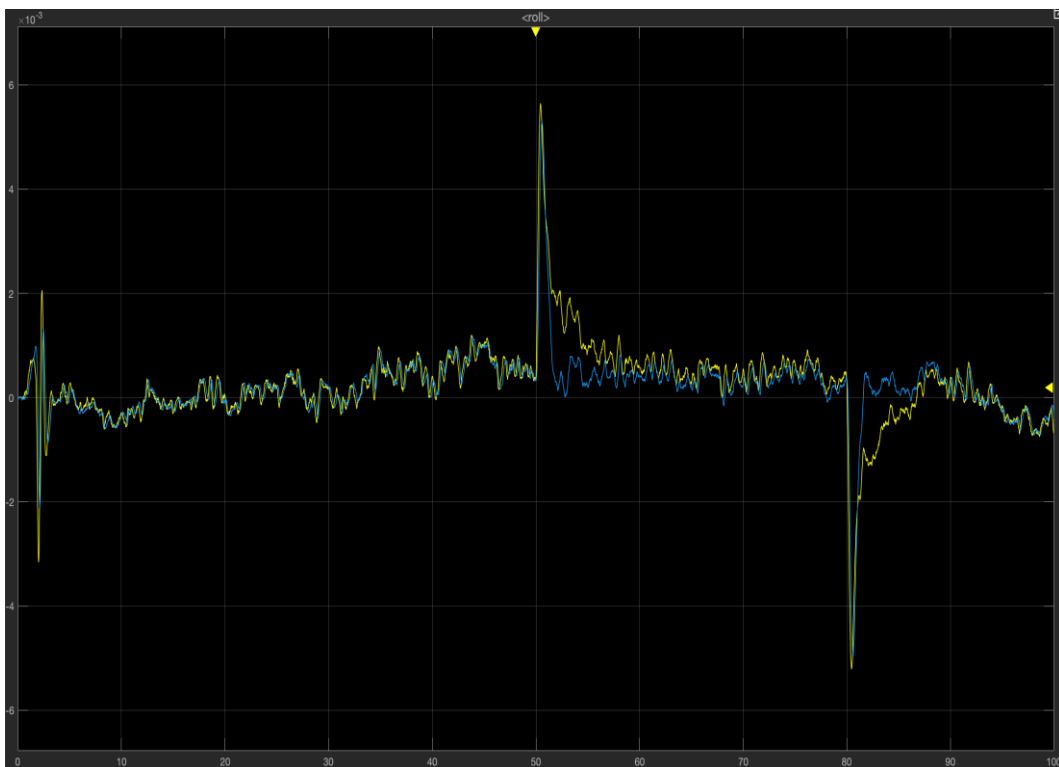


Fig.66 Grafico simulazione del controllore con luogo delle radici di secondo tentativo, ingressi variabili, tracciato del Roll con il suo riferimento

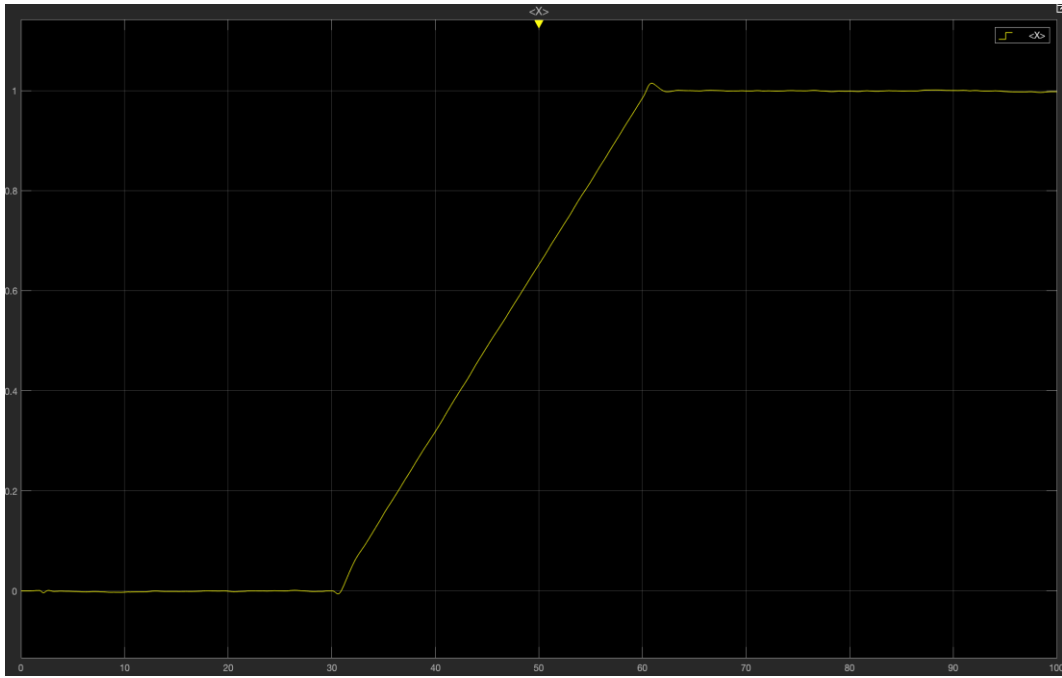


Fig.67 Grafico simulazione del controllore con luogo delle radici di secondo tentativo, ingressi variabili, tracciato della X

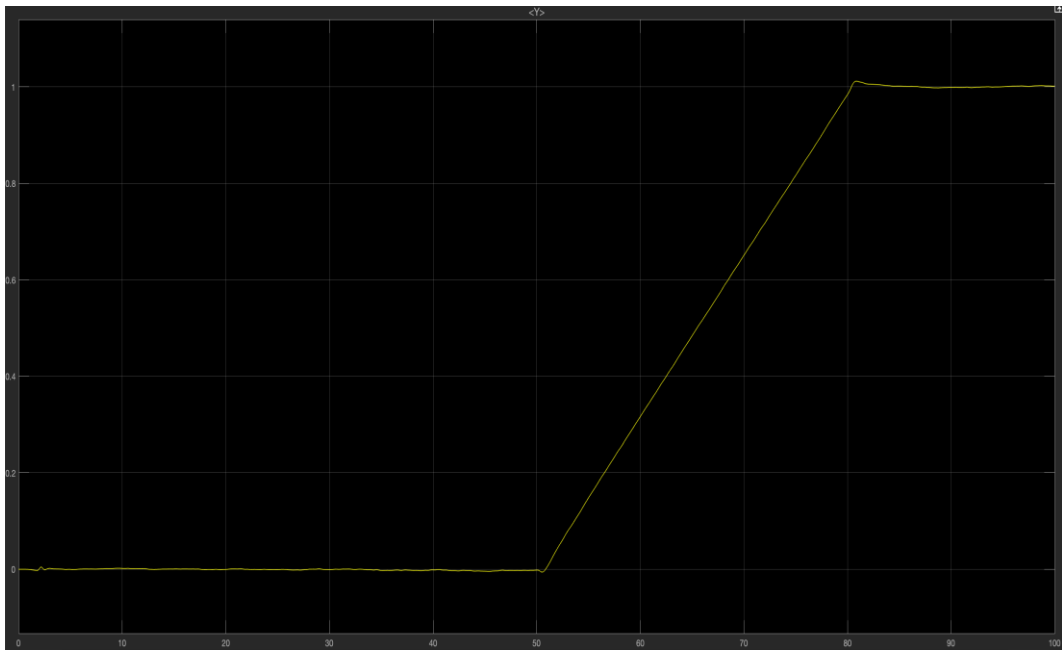


Fig.67 Grafico simulazione del controllore con luogo delle radici di secondo tentativo, ingressi variabili, tracciato della Y

Dai grafici notiamo come anche il controllore di secondo tentativo è in grado di controllare efficacemente le posizioni X-Y, sia per i riferimenti costanti che per quelli variabili.

Inoltre, notiamo da fig.63 e fig.66 che il controllore riesce a seguire il riferimento in tutti e due i casi con un'attenuazione delle oscillazioni a fronte di una sovralongazione minore e pari a 14% e tempo di salita leggermente superiore pari a 1.1s.

6. Conclusioni

Nel presente elaborato ci siamo posti l'obiettivo di progettare dei controllori tramite sintesi in frequenza e tramite sintesi con luogo delle radici per il controllo dell'angolo di roll di un minidrone.

Dai risultati ottenuti notiamo come la posizione X-Y risulta controllata e stabile in entrambi i riferimenti e con tutte e due le tecniche di controllo lineare utilizzate.

Per quanto riguarda il controllo dell'angolo di roll, con la tecnica di sintesi in frequenza, abbiamo sintetizzato un controllore che riuscisse a seguire ottimamente il riferimento.

Con sintesi mediante luogo delle radici è stato progettato un controllore che seguisse efficacemente il riferimento e limitasse le oscillazioni.

Nel complesso possiamo affermare che i controllori sintetizzati soddisfano le specifiche richieste e soprattutto hanno un comportamento simile ai PID controller originali.

7. Bibliografia

- S. Bouabdallah *Design and Control of Quadrotors with Application to Autonomous Flying*, Lausanne, 2007
- P.Ponds, R.Mahony, P.Corke, *Modelling and Control of a large quadrotor root. Control Engeneering Practice*, 2010
- H.Goldstein, *Classical Mechanics*, Addison-Wesley, 1970 (settima edizione)
- W.Z.Fum, *Implementation of Simulink Controller Design on Iris+Quadcopter*, Monterey, CA, 2015
- Daniele Gambini, *Studio e sviluppo in Simulink di tecniche di controllo per minidroni*, 2019
- Francesco Paoli Leonardi, *Studio e sviluppo di tecniche di controllo avanzate per droni*, 2020