



UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Informatica e dell' Automazione

TESI DI LAUREA TRIENNALE

PROGETTO E SVILUPPO DI TECNICHE DI CONTROLLO PER UN MINIDRONE

Design and development of control techniques for a mini drone

Candidato:

Flavia Gioiello

Matricola 1087154

Relatore:

Prof. Gianluca Ippoliti

Correlatore:

Prof. Giuseppe Orlando

ANNO ACCADEMICO 2020-2021

"The wing structure of the hornet, in relation to its weight, is not suitable for flight, but he does not know this and flies anyway"

Albert Einstein

Ringraziamenti

Desidero ringraziare innanzitutto il relatore di questa tesi, il Professor Gianluca Ippoliti e il correlatore, il Professor Giuseppe Orlando, per la loro grande disponibilità e attenzione in questi mesi difficili, dovuti alla particolare condizione storica.

Ringrazio la mia famiglia che con il loro sostegno morale ed economico hanno permesso di raggiungere questo traguardo.

Un grazie speciale va alla mia seconda famiglia, compagni di corso e amici sinceri, Hermes Karalliu, Valeria Calvarese, Alessia Castagnolo e Federica Paganica, con cui ho condiviso tre anni bellissimi e su cui ho potuto sempre contare.

Grazie alla mia coinquilina Giusy Cinquepalmi con cui ho trascorso spensieratamente le mie lunghe giornate anconetane.

Un grazie di cuore al mio ragazzo, Lorenzo Onori, che mi è stato sempre vicino, sopportando le mie ansie e i miei sbalzi di umore.

Grazie a tutti.

Sommario

Il presente elaborato offre una modellazione matematica dei fenomeni alla base del comportamento di un drone quadricottero e un'analisi dettagliata del modello *Simulink*, presente all'interno del pacchetto "*Simulink Support Package for Parrot Minidrones*", con lo scopo di progettare un controllore per il rollio, attraverso due tecniche di controllo. Infine i due controllori implementati e il controllore di "*default*" vengono confrontati attraverso varie considerazioni e con l'ausilio di indici di prestazione.

Indice

Elenco delle figure	8
Glossario	10
Lista dei simboli e Acronimi	12
Introduzione	17
1 I quadricotteri	19
1.1 Caratteristiche generali	19
1.1.1 Parrot Mambo	21
1.2 Modello matematico	22
1.2.1 Semplificazioni iniziali	22
1.2.2 Sistemi di riferimento	23
1.2.3 Angoli di Eulero	24
1.2.4 Matrici di rotazione	25
1.2.5 Equazioni generali del sistema	27
2 Modello in Simulink	33
2.1 Signal Editor	34
2.2 Sensors	35
2.3 Flight Control System	36
2.3.1 Controllore per il Pitch e il Roll	39
2.3.2 Controllore per l'angolo di Yaw	41
2.3.3 Controllore per l'altitudine	41
2.4 Environment	42
2.5 Airframe	42
2.5.1 Modello non lineare	43
2.5.2 Modello lineare	47
2.6 Visualization	50
2.7 Instrument Panel	51
3 Progettazione dei controllori	53
3.1 Controllore in frequenza	54
3.2 Controllore con luogo delle radici	59

4	Simulazioni e confronti	63
4.1	Analisi dei tracciati	64
4.2	Indici di prestazione	70
	Conclusione	75
A	Teoria del controllo	77
A.1	Sintesi in frequenza	77
A.2	Sintesi in s	80
A.3	Regolatori PID	81
	Bibliografia	83

Elenco delle figure

1	Progetto della vite aerea di Leonardo da Vinci	17
2	Un drone usato per le consegne	17
3	Un drone usato per irrigare i campi	17
1.1	A sinistra la configurazione a più, a destra quella a croce	19
1.2	Manovre della configurazione a croce	21
1.3	Drone Mambo della Parrot SA	21
1.4	Earth Fixed Frame e Body Fixed Frame	23
1.5	Angoli di Eulero	24
1.6	Effetto del Blade Flapping	30
2.1	Progetto Simulink dal pacchetto per mini droni	34
2.2	Generatore di segnali di default	35
2.3	Blocco del sensore IMU	36
2.4	Blocco "camera"	36
2.5	Blocco Flight Control System	37
2.6	Blocco Landing logic	37
2.7	Crash Predictor Flags	38
2.8	Blocco State Estimator	38
2.9	Blocco Flight Controller	39
2.10	Blocco XY-to-reference-orientation	40
2.11	Blocco Attitude	40
2.12	Blocco Yaw	41
2.13	Blocco Gravity Feedforward/Equilibrium Thrust	41
2.14	Blocco Environment	42
2.15	Blocco NonLinear Airframe	43
2.16	Blocco AC model	43
2.17	Blocco Drag Calculation	44
2.18	Blocco Drag Calc	44
2.19	Blocco Gravity Force Calculation	45
2.20	Blocco Motors To W	45
2.21	Blocco Rotor Dynamics	46
2.22	Blocco Rotor Dynamics (continuo)	46
2.23	Blocco Applied Force Calculation	46
2.24	Blocco Bus setup	47
2.25	Blocco Linear Airframe	48
2.26	Matrice A	48

2.27	Matrice B	49
2.28	Matrice C	49
2.29	Matrice D	50
2.30	Simulazione tramite Simulink 3D	50
2.31	Pannello di controllo	51
3.1	Diagrammi di Bode, Nichols, Nyquist e risposta a gradino di $P(s)$	56
3.2	Risposta al gradino di $\hat{F}(s)$	57
3.3	Diagramma di Bode di $\hat{F}(s)$	57
3.4	Diagramma di Bode di $F_a(s)$	58
3.5	Diagrammi di Bode, di Nichols e risposta al gradino di $F(s)$	59
3.6	Luogo delle radici di $P(s)$	60
3.7	Luogo delle radici di $\hat{F}(s)$	60
3.8	Risposta al gradino di $F(s)$	61
4.1	Errore di sample times multipli	63
4.2	Modifiche del blocco Attitude	64
4.3	Aggiunta di blocchi Scope per l'analisi dei tracciati	65
4.4	Tracciati X a confronto	66
4.5	Tracciati Y a confronto	67
4.6	Tracciati Z a confronto	68
4.7	Tracciati Roll a confronto	69
4.8	Blocco IAE in Simulink	70
4.9	Blocco ISE in Simulink	71
4.10	Blocco ITAE in Simulink	71
4.11	Blocco aggiunto per il calcolo degli indici	72
4.12	Blocco INDICI	72
A.1	Diagrammi della rete anticipatrice	79

Glossario

C

Controreazione Sistema in cui parte del segnale di uscita viene riportato all'entrata; è usato per limitare i problemi dei sistemi in catena diretta e fornisce al controllore l'andamento effettivo dell'uscita, permettendo, così, di modificarlo.. 33

D

Demultiplexer Il demultiplexer è un circuito digitale combinatorio che divide una linea di ingresso dati in n linee di selezione.. 64

Drag Con il termine Drag si intende la forza di resistenza di un fluido che si oppone al moto dell'oggetto. 23

F

Filtro di Kalman Il filtro di Kalman è un filtro che permette di stimare lo stato di un sistema a partire dai dati misurati soggetti a rumore. 39

I

Inerzia L'inerzia è la tendenza di un corpo a resistere alle variazioni di stato. 23

Integral Wind Up Questo fenomeno è dovuto ai vincoli di saturazione degli attuatori. Succede che quando l'uscita del controllore impone valori superiori ai vincoli degli attuatori, l'uscita non cresce, seppur l'errore non è nullo, portando ad un aumento incessante del termine integratore. 41

L

Legge fondamentale della Dinamica rotazionale Equivalente al Secondo principio della dinamica o Seconda legge di Newton, afferma che il momento risultante delle forze agenti su un corpo rigido in rotazione attorno a un asse fisso è direttamente proporzionale all'accelerazione angolare per il momento di inerzia.. 27, 28

M

Momento Torcente Il momento torcente è il prodotto vettoriale tra la forza applicata e il braccio. 20

Moto di precessione Il moto di precessione è il moto generato dalla rotazione dell'asse di rotazione di un oggetto, caratterizzato da una rotazione lenta ed uniforme attorno all'asse verticale z (un esempio classico è il moto della trottola). 31

Motori Brushless I motori Brushless sono una tipologia di motore a commutazione elettronica senza contatti striscianti (ovvero la commutazione avviene per mezzo di circuiti elettrici, grazie all'utilizzo di transistori di potenza). 19

P

Pitch Il movimento di Pitch, in italiano *Beccheggio*, è il movimento di rotazione di un oggetto intorno al proprio asse trasversale. 20, 24, 27, 34, 39, 41, 64

Primo principio della dinamica o Prima legge di Newton Il primo principio della dinamica afferma che "*un oggetto non soggetto a forze o soggetto a forze la cui risultante è zero, permane nel suo stato di quiete o si muove di moto rettilineo uniforme se era in moto*". 11

R

Roll Il movimento di Roll, in italiano *Rollio*, è il movimento di rotazione di un oggetto intorno al proprio asse longitudinale. 20, 24, 27, 34, 39, 41, 64, 71

S

Secondo principio della dinamica o Seconda legge di Newton La forza totale agente su un corpo rigido è direttamente proporzionale all'accelerazione per la massa del corpo, e condivide la direzione e il verso dell'accelerazione. 10, 27

Sistema Inerziale Un sistema inerziale è un sistema di riferimento dove vale il Primo principio della dinamica o Prima legge di Newton. 12–14, 24, 26, 27

Sistema Tempo Invariante In un sistema a tempo invariante i parametri non dipendono direttamente dal tempo, o ancora l'uscita generata da un segnale di ingresso ritardata di un certo fattore, è uguale a quella generata dal segnale originario, ritardata a sua volta del medesimo fattore. 23

T

Thrust Il Thrust, in italiano *spinta verticale*, è il movimento traslazionale lungo l'asse z. 20, 23

Y

Yaw Il movimento di Yaw, in italiano *Imbardata*, è il movimento di rotazione di un oggetto intorno al proprio asse verticale. 20, 24, 27, 34, 39–41

Lista dei simboli e Acronimi

Simboli

- A Sezione di area superiore del drone. 30
- A_r Area del rotore. 30
- D Coefficiente di resistenza. 30
- F_b Forze sul sistema solidale con il drone. 27
- F_d Forza di attrito viscoso. 30
- F_i Forze sul Sistema Inerziale. 27
- I Tensore d'Inerzia. 27
- P Forza peso. 28
- R Raggio del rotore. 30
- T Spinta dei motori. 28
- T_i Spinta dell' i -esimo motore. 30
- α Angolo di salita. 31
- $\ddot{\phi}$ Accelerazione angolare lungo l'asse x del Sistema Inerziale. 27
- $\ddot{\psi}$ Accelerazione angolare lungo l'asse z del Sistema Inerziale. 27
- $\ddot{\theta}$ Accelerazione angolare lungo l'asse y del Sistema Inerziale. 27
- \ddot{x}_i Accelerazione lineare lungo l'asse x del Sistema Inerziale. 27
- \ddot{y}_i Accelerazione lineare lungo l'asse y del Sistema Inerziale. 27
- \ddot{z}_i Accelerazione lineare lungo l'asse z del Sistema Inerziale. 27
- $\dot{\phi}$ Velocità angolare lungo l'asse x del Sistema Inerziale. 26
- $\dot{\psi}$ Velocità angolare lungo l'asse z del Sistema Inerziale. 26
- $\dot{\theta}$ Velocità angolare lungo l'asse y del Sistema Inerziale. 26
- \dot{p} Accelerazione angolare lungo l'asse x nel sistema solidale con il drone. 28

\dot{q} Accelerazione angolare lungo l'asse y nel sistema solidale con il drone. 28
 \dot{r} Accelerazione angolare lungo l'asse r nel sistema solidale con il drone. 28
 \dot{u} Accelerazione lineare lungo l'asse x nel sistema solidale con il drone. 28
 \dot{v} Accelerazione lineare lungo l'asse y nel sistema solidale con il drone. 28
 \dot{w} Accelerazione lineare lungo l'asse z nel sistema solidale con il drone. 28
 \dot{x}_i Velocità lineare lungo l'asse x del Sistema Inerziale. 26
 \dot{y}_i Velocità lineare lungo l'asse y del Sistema Inerziale. 26
 \dot{z}_i Velocità lineare lungo l'asse z del Sistema Inerziale. 26
 λ Inflow Ratio. 31
 μ_i Advance Ratio. 31
 ω_i Velocità angolare dell' i -esimo motore. 30, 31
 ϕ Angolo di *rotazione propria* del Sistema Inerziale. 24, 25, 27, 53
 ψ Angolo di *precessione* del Sistema Inerziale. 24, 25, 27
 ρ Densità dell'aria. 30
 τ_b Momento torcente sul sistema solidale con il drone. 28
 τ_i Momento torcente sul Sistema Inerziale. 27
 θ Angolo di *nutazione* del Sistema Inerziale. 24, 25, 27
 a_1 Angolo di flapping longitudinale. 30
 b Coefficiente di spinta dei motori. 28
 b_1 Angolo di flapping laterale. 30
 d Coefficiente di resistenza dell'aria. 29
 g Accelerazione di gravità. 28
 l Proiezione dei bracci del drone sugli assi. 29
 m Massa del drone. 27, 28
 p Velocità angolare lungo l'asse x nel sistema solidale con il drone. 25
 q Velocità angolare lungo l'asse y nel sistema solidale con il drone. 25, 53
 r Velocità angolare lungo l'asse z nel sistema solidale con il drone. 25, 31
 u Velocità lineare lungo l'asse x nel sistema solidale con il drone. 25

v Velocità lineare lungo l'asse y nel sistema solidale con il drone. 25
 w Velocità lineare lungo l'asse z nel sistema solidale con il drone. 25
 x_b Posizione lungo l'asse x del Sistema solidale con il drone. 24
 x_i Posizione lungo l'asse x del Sistema Inerziale. 24, 25
 y_b Posizione lungo l'asse y del Sistema solidale con il drone. 24
 y_i Posizione lungo l'asse y del Sistema Inerziale. 24, 25
 z_b Posizione lungo l'asse z del Sistema solidale con il drone. 24
 z_i Posizione lungo l'asse z del Sistema Inerziale. 24, 25

A

ABC Aircraft Body Center. 24
APR Aerogiri a Pilotaggio Remoto. 18
ARM Advanced RISC Machines. 22

C

CoG Center of Gravity. 23, 24, 30, 31, 45

D

DCM Direction Cosine Matrix. 26, 35, 44
DoF Degrees of Freedom. 19, 24

F

FPS Frame per second. 22

I

IAE Integral of the Absolute value of Error. 70
IMU Inertial Measurement Unit. 22, 35
ISE Integral of Square Error. 70
ITAE Integral of the Time-weighted Absolute Error. 70

M

mAh MilliAmpere per ora. 22
MHz MegaHertz. 22

N

NED North-East-Down. 24

P

PCB Printed Circuit Board. 22

PID Controllore Proporzionale-Integrale-Derivativo. 33, 39

S

SONAR Sound Navigation and Ranging. 35

Introduzione

Il primo progetto di velivolo è attribuibile a *Leonardo Da Vinci*, sotto il nome di "Vite aerea", immaginata come una vite senza fine (vedi Figura 1¹). Sebbene non vi sono tracce di un'effettiva realizzazione del prototipo e sebbene, a causa delle sue caratteristiche, non avrebbe potuto funzionare, rimane una brillante intuizione, anticipando di secoli i primi prototipi di velivolo, nati in ambito bellico nel 1849 e largamente utilizzati durante le Guerre Mondiali. [4]

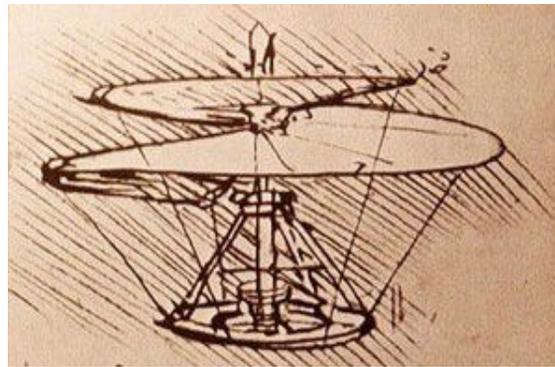


Figura 1. Progetto della vite aerea di Leonardo da Vinci



Figura 2. Un drone usato per le consegne



Figura 3. Un drone usato per irrigare i campi

Attualmente diverse tipologie di drone, velivoli motorizzati con comando remoto, vengono utilizzate per i più svariati scopi dalla sorveglianza, all'agricoltura,

¹Manoscritto B, foglio 83 v., 1489, Manoscritti di Francia, Parigi, Institut de France.

oppure usati per spedizioni, per gare agonistiche, nella fotografia o anche per meri scopi ludici (esempi nelle Figure 2² e 3³).

Il termine *drone* fu introdotto per la prima volta nel 1920, usato come bersaglio volante durante le esercitazioni di tiro dei cannoni delle navi da guerra o anche associato al nome *fuco* a causa del suo ronzio mentre è in funzione. In generale questo termine viene utilizzato per indicare tutte le tipologie di *Aerogiri a Pilotaggio Remoto (APR)*, differenziandosi per forme, raggio operativo, quota di volo, peso, numero di motori e dimensioni.[3]

²Immagine presa da: Project Wing: i droni corrieri di Google arriveranno nel 2017

³Immagine presa da: Drones in Agriculture – An overall analysis

Capitolo 1

I quadricotteri

Prima di analizzare le varie tecniche di controllo e la conseguente implementazione, è opportuno soffermarsi sulle caratteristiche dei droni quadricotteri, oggetto di studio in questo elaborato, e sulla loro dinamica.

1.1 Caratteristiche generali

La tipologia di drone più diffusa è il quadricottero (dal greco *ptéron*: ala, “quattro ali”). I droni quadricotteri appartengono alla famiglia degli aerogiri a pilotaggio remoto e sono chiamati così per i loro 4 rotori. Presentano un corpo centrale e quattro bracci che sorreggono i quattro Motori Brushless, ognuno con una propria elica, di colori diversi per differenziare il lato a cui è rivolto il drone durante il volo.

I quattro attuatori possono presentare due diverse tipologie di configurazione: la configurazione *a croce* oppure *a più*, come in Figura 1.1.

Sebbene la configurazione *a più* sia più simile a un aeroplano, il che rende più

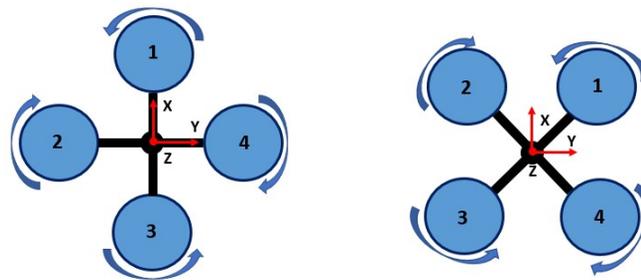


Figura 1.1. A sinistra la configurazione a più, a destra quella a croce

facile l'orientamento durante il volo, la configurazione *a croce* è la più utilizzata per la sua grande stabilità.

Indipendentemente dalla configurazione scelta, il sistema risulta sotto-attuato in quanto ha solo quattro attuatori per sei gradi di libertà (DoF), 3 traslazionali

e 3 rotazionali. Non avendo, quindi, abbastanza attuatori per ogni movimento, alcune direzioni sono incontrollabili direttamente. Ad esempio per muoversi in orizzontale, il drone deve prima ruotare leggermente, accoppiando rotazione e spinta verticale; dunque le manovre direttamente controllabili sono le tre rotazionali (Yaw, Pitch, Roll) più il Thrust.

I motori producono spinta grazie all'azione delle eliche (ognuna produce una forza di un quarto di quella di gravità) che, producendo forza verso il basso, causano una forza di reazione verso l'alto; se questa forza è applicata nel punto di gravità, la risultante è una pura traslazione senza rotazione e se è esattamente in modulo uguale alla forza di gravità, il drone fluttuerà sul posto (manovra chiamata anche "hovering"). D'altro canto, se la forza è applicata a distanza dal centro di gravità, vengono prodotti movimenti sia traslazionali che rotazionali, a causa della creazione di momenti.¹

Però, oltre alle forze verticali, i motori generano un Momento Torcente, dovuto alla rotazione dell'elica per produrre spinta, causando una rotazione. L'azione di questi momenti viene compensata facendo muovere le eliche opposte nella stessa direzione e le eliche adiacenti con direzione opposta.

I due tipi di configurazione controllano le varie manovre in modi simili, perciò, per semplicità verrà studiata solo il controllo delle manovre della configurazione a croce (di maggior interesse nel presente elaborato).

Yaw. Per generare un angolo di Yaw, si accelerano i due motori che girano nello stesso verso di un certo fattore e si rallentano gli altri due per lo stesso fattore; in questo modo la forza risultante rimane la stessa e il drone continua a librare ma cambiano i momenti di rotazione, aumentando quelli di direzione uguali, così il drone ruota. Quindi la manovra di Yaw non ha effetti sul Thrust o sul Pitch e Roll.²

Roll. Per il Roll si aumenta la forza generata dai due motori adiacenti con direzione opposta sul lato destro o sinistro (in base al senso di rotazione) e viene diminuita l'altra coppia, in modo da generare un Momento Torcente, che faccia ruotare il drone senza influire sullo Yaw, essendo coppie con verso opposto.

Pitch. Analogamente al Roll, vengono accelerati i motori adiacenti frontali o posteriori e diminuiti gli altri due.

Thrust. Per il Thrust, vengono aumentati o diminuiti tutti i quattro motori simultaneamente. [6]

Lo schema illustrativo delle varie manovre è rappresentato nella Figura 1.2.

¹Se la rotazione non è controbilanciata da altre forze, la rotazione cambia direzione alla forza di reazione che non si oppone più a quella di gravità con la conseguente caduta del drone al suolo.

²Se le coppie di motori con la stessa direzione fossero adiacenti, si verrebbe a creare uno sbilanciamento di forze e le varie manovre non risulterebbero più disaccoppiate; ecco spiegata la scelta sopra illustrata.

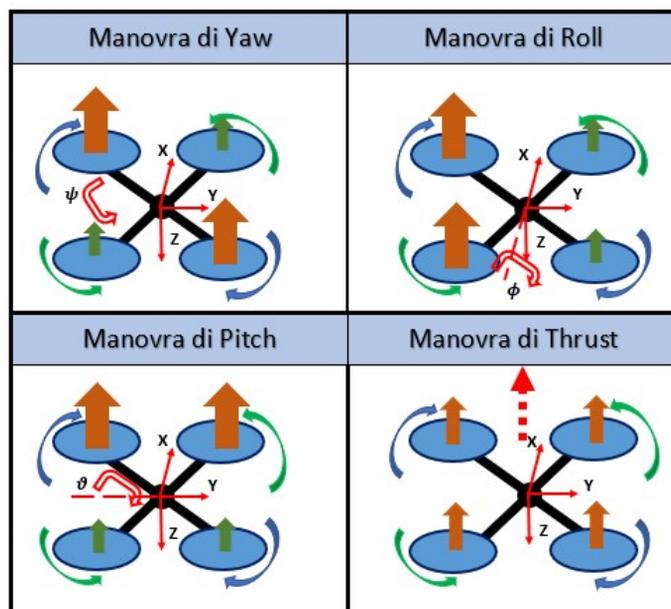


Figura 1.2. Manovre della configurazione a croce

1.1.1 Parrot Mambo

In particolare il drone preso come modello del presente studio è il *Parrot Mambo* della società francese *Parrot*, in Figura 1.3.



Figura 1.3. Drone Mambo della Parrot SA

È un quadricottero con configurazione *a croce*, costituito da quattro eliche rotanti azionate da motori di tipo *coreless DC brushed*³, con peso complessivo di 63.5

³I motori *coreless DC brushed*, rispetto ai motori tradizionali, non presentano il nucleo in ferro nel rotore ma l'avvolgimento in rame è realizzato in una configurazione a cilindro. Il risultato è

grammi. Viene alimentato da una batteria ricaricabile LiPo (accumulatore Litio-Polimero) da 660 mAh, con autonomia dichiarata di 10 minuti senza accessori e ricarica di 30 minuti. È dotato di quattro sensori:

- **Sensore degli ultrasuoni**, usato per misurare le distanze verticali e dalle superfici, attraverso un impulso sonoro ad alta frequenza; misurando il tempo impiegato dall'impulso per rimbalzare sulla superficie e poi tornare indietro, è possibile calcolarne la distanza.
- **Fotocamera**, da 60 FPS, stima il movimento orizzontale e la velocità attraverso la tecnica di processamento chiamata *Flusso ottico*.⁴
- **Sensore della pressione**, misura l'altitudine in base alle variazioni di pressione (salendo di altitudine, la pressione diminuisce e viceversa).
- **IMU (Unità di Misura Inerziale)**, misura l'accelerazione lineare e angolare attraverso l'uso di rispettivamente tre accelerometri assiali e tre giroscopi.

Inoltre ha una scheda di tipo Printed Circuit Board (PCB) con processore ARM A9 a 800 MHz con Linux come Sistema Operativo. Presenta sia una connettività wireless di tipo Bluetooth 4.0 che consente di interfacciarsi al drone tramite l'applicazione per smartphone "*Freeflight Mini*", sia attraverso una porta micro-USB che consente il collegamento ad un computer.[8]

1.2 Modello matematico

⁵ L'obiettivo di questa sezione è quello di trovare un modello matematico in grado di descrivere correttamente la dinamica di un quadricottero. Questa fase risulta necessaria per una buona analisi e implementazione dei vari controllori.

1.2.1 Semplificazioni iniziali

Tra tutti i modelli studiati negli ultimi anni, il più comune è quello di *Hamel et al.*, modello non lineare che consiste nel considerare solo la dinamica di un corpo rigido soggetto a forze e momenti, ma non la sua aerodinamica. Il modello citato considera alcune assunzioni in modo da semplificarne lo studio, mantenendolo comunque un valido modello rispetto al reale comportamento del drone:

che sono più leggeri e hanno minori momenti d'inerzia, ovvero una risposta più rapida. Inoltre essendo brushed, presentano spazzole per il contatto.

⁴Il *Flusso ottico* descrive il movimento degli oggetti tra un frame e il successivo attraverso l'uso di campi vettoriali della variazione di luminosità; non rispecchia, però, il campo di moto in quanto si possono avere variazioni di luminosità anche se l'oggetto rimane fermo.

⁵Per questa sezione si consiglia fortemente di leggere la lista dei simboli per ulteriori chiarimenti sulle notazioni usate.

1. il quadricottero viene rappresentato come un corpo rigido con massa e, quindi, Inerzia, sottoposto alla forza di gravità e ai vari momenti;
2. il quadricottero ha una struttura simmetrica e quindi la matrice di Inerzia è diagonale;
3. il Centro di Gravità (CoG) coincide con l'origine del sistema di riferimento solidale al quadricottero (*Body Fixed Frame*);
4. Thrust e la resistenza dell'aria (Drag) sono proporzionali al quadrato della velocità angolare;
5. il sistema è un Sistema Tempo Invariante;
6. si possono assumere i quattro motori uguali tra loro senza perderne di generalità
7. gli assi del sistema solidale al quadricottero coincidono con gli assi principali di inerzia.

Naturalmente questo modello semplificato non riesce a rappresentare l'intera dinamica complessa di un quadricottero ⁶, in quanto trascura l'effetto giroscopico (trascurabile a causa delle ridotte velocità) e, in particolar modo, l'effetto del *Blade Flapping*. Questi ultimi saranno brevemente accennati nella parte conclusiva del suddetto capitolo, insieme ad altre opportune considerazioni.

1.2.2 Sistemi di riferimento

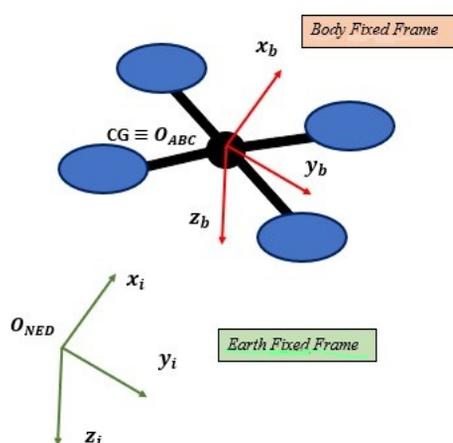


Figura 1.4. Earth Fixed Frame e Body Fixed Frame

⁶D'altronde nessun modello matematico che si rispetti riesce a modellare l'intera dinamica dei sistemi reali e tutti i modelli funzionano solo se vengono rispettate certe ipotesi di lavoro.

È bene partire andando a specificare i sistemi di riferimento necessari per trovare il modello matematico.

Il primo sistema di riferimento è un Sistema Inerziale, scelto in riferimento al piano tangente alla superficie terrestre in ogni punto con origine degli assi la posizione dell'osservatore (*Earth Fixed Frame*). La terna usata per questo sistema è la terna North-East-Down (NED), con il versore x_i che punta verso Nord, il versore y_i verso Est e quello z_i verso il centro della terra.

Il secondo sistema è il sistema solidale al quadricottero con origine nel CoG (*Body Fixed Frame*). La terna usata è la cosiddetta terna Aircraft Body Center (ABC), dove il versore x_b punta la parte frontale del drone, il versore y_b punta la destra del drone, quello z_b punta verso il basso. I due sistemi di riferimento sono rappresentati nella Figura 1.4.

1.2.3 Angoli di Eulero

Come accennato in precedenza, il quadricottero ha 6 gradi di libertà (Degrees of Freedom (DoF)): 3 di traslazione X, Y, Z, che rappresentano lo spostamento lineare dell'oggetto e 3 rotazionali Roll, Pitch, Yaw, che rappresentano lo spostamento angolare.

In particolare gli sfasamenti angolari, dovuti agli spostamenti rotazionali, sono descritti dagli *Angoli di Eulero*, introdotti da Leonhard Euler per descrivere l'orientamento di un corpo rigido rispetto a un sistema inerziale.

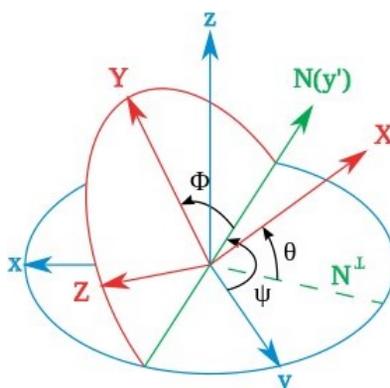


Figura 1.5. Angoli di Eulero

- ϕ : Angolo di *rotazione propria* del Sistema Inerziale, generato dalla rotazione della terna NED, intorno all'asse X e definito in $[-\pi, \pi]$. Corrisponde alla manovra di Roll.
- θ : Angolo di *nutazione* del Sistema Inerziale, generato dalla rotazione intorno all'asse Y e definito in $[-\pi/2, \pi/2]$. Corrisponde alla manovra di Pitch.
- ψ : Angolo di *processione* del Sistema Inerziale, generato dalla rotazione intorno all'asse Z e definito in $[-\pi, \pi]$. Corrisponde alla manovra di Yaw.

Gli angoli vengono rappresentati nella Figura 1.5.

Essendoci 6 gradi di libertà e 2 sistemi di riferimento, si possono definire 12 variabili totali di interesse:

Earth Fixed Frame

Vettore lineare di posizione:

$$P=[x_i \ y_i \ z_i]^T$$

Vettore degli angoli di Eulero:

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

Body Fixed Frame

Velocità lineari:

$$V_b=[u \ v \ w]^T$$

Velocità angolari:

$$\Omega_b=[p \ q \ r]^T.$$

1.2.4 Matrici di rotazione

È possibile passare dal sistema di riferimento mobile a quello inerziale tramite tre matrici di trasformazione, ognuna corrispondente a un angolo di Eulero, chiamate "*Matrici di Rotazione*". Queste matrici sono quadrate e ortogonali⁷ e rappresentano gli spostamenti angolari di un asse fisso rispetto agli altri due. Le tre matrici sono così definite:

Matrice di rotazione intorno all'asse X

$$R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (1.1)$$

Matrice di rotazione intorno all'asse Y

$$R_Y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (1.2)$$

Matrice di rotazione intorno all'asse Z

$$R_Z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

La matrice complessiva è data dal prodotto delle matrici 1.1, 1.2 e 1.3:

$$R_i^b = R_X(\phi) \cdot R_Y(\theta) \cdot R_Z(\psi) =$$

⁷In una matrice quadrata le righe sono in numero pari alle colonne; una matrice si dice ortogonale se è quadrata, se moltiplicata per la sua trasposta dà la matrice identità e se la trasposta coincide con la matrice inversa.

$$= \begin{bmatrix} \cos \theta \cos \psi & -\cos \theta \sin \psi & \sin \theta \\ \cos \phi \sin \psi + \cos \psi \sin \phi \sin \theta & \cos \phi \cos \psi - \sin \phi \sin \theta \sin \psi & -\cos \theta \sin \phi \\ \sin \phi \sin \psi - \cos \phi \cos \psi \sin \theta & \cos \psi \sin \phi + \cos \phi \sin \theta \sin \psi & \cos \phi \cos \theta \end{bmatrix} \quad (1.4)$$

Tale matrice permette il trasferimento dalle componenti lineari del sistema inerziale a quelle del *Body Fixed Frame*.

Per il trasferimento inverso basta fare la trasporta della matrice 1.4:

$$R_b^i = (R_i^b)^T = \begin{bmatrix} \cos \theta \cos \psi & \cos \phi \sin \psi + \cos \psi \sin \phi \sin \theta & \sin \phi \sin \psi - \cos \phi \cos \psi \sin \theta \\ -\cos \theta \sin \psi & \cos \phi \cos \psi - \sin \phi \sin \theta \sin \psi & \cos \psi \sin \phi + \cos \phi \sin \theta \sin \psi \\ \sin \theta & -\cos \theta \sin \phi & \cos \phi \cos \theta \end{bmatrix} \quad (1.5)$$

Tale matrice è anche chiamata matrice Direction Cosine Matrix (DCM).

Quindi

$$V_b = R_i^b \cdot \dot{P}$$

$$\dot{P} = R_b^i \cdot V_b \quad (1.6)$$

dove \dot{P} è il vettore delle velocità lineari del Sistema Inerziale, $\dot{P} = [\dot{x}_i \ \dot{y}_i \ \dot{z}_i]^T$.

Analogamente è possibile trovare le matrici di trasferimento delle componenti angolari:

$$T_i^b = R_X(\phi) \cdot R_Y(\theta) \cdot R_Z(\psi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + R_X(\phi) \cdot R_Y(\theta) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_X(\phi) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

$$T_i^b = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (1.7)$$

La corrispondente inversa:

$$T_b^i = \begin{bmatrix} 1 & \sin \phi \frac{\sin \theta}{\cos \theta} & \cos \phi \frac{\sin \theta}{\cos \theta} \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \frac{1}{\cos \theta} & \cos \theta \frac{1}{\cos \theta} \end{bmatrix} \quad (1.8)$$

Quindi

$$\Omega_b = T_i^b \cdot \dot{E}$$

$$\dot{E} = T_b^i \cdot \Omega_b \quad (1.9)$$

dove \dot{E} è il vettore delle velocità angolari del Sistema Inerziale, $\dot{E} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$.

1.2.5 Equazioni generali del sistema

Essendoci 12 variabili, si hanno 12 equazioni che descrivono la dinamica del sistema.

Le prime 6 equazioni sono direttamente date dalle relazioni di Eulero 1.6 e 1.9, andando ad esplicitare i prodotti matriciali:

$$\begin{cases} \dot{x} = u \cdot \cos \theta \cos \psi + v(\cos \psi \sin \theta \sin \phi - \cos \phi \sin \psi) + w(\cos \phi \cos \psi \sin \theta + \sin \phi \sin \psi) \\ \dot{y} = u \cdot \cos \theta \sin \psi + v(\sin \theta \sin \phi \sin \psi + \cos \phi \cos \psi) + w(\cos \phi \sin \theta \sin \psi - \cos \psi \sin \phi) \\ \dot{z} = -u \cdot \sin \theta + v \cdot \cos \theta \sin \phi + w \cdot \cos \phi \cos \theta \end{cases} \quad (1.10)$$

$$\begin{cases} \dot{\phi} = p + q \cdot \sin \phi \tan \theta + r \cdot \cos \phi \tan \theta \\ \dot{\theta} = q \cdot \cos \theta - r \cdot \sin \phi \\ \dot{\psi} = q \cdot \frac{\sin \psi}{\cos \theta} + r \cdot \frac{\cos \psi}{\cos \theta} \end{cases} \quad (1.11)$$

Si può notare come le variazioni degli angoli ϕ e θ influiscono sull'angolo ψ , ovvero le manovre di Roll e Pitch causano rotazioni intorno all'asse verticale, cioè Yaw.

Le altre 6 equazioni si trovano applicando le leggi della dinamica, in particolare il Secondo principio della dinamica o Seconda legge di Newton e la Legge fondamentale della Dinamica rotazionale, sotto le ipotesi 3. e 7. descritte a pagina 22:

$$m \cdot \ddot{P} = \sum F_i$$

$$I \cdot \ddot{E} = \sum \tau_i \quad (1.12)$$

dove \ddot{P} è il vettore delle accelerazioni lineari del Sistema Inerziale,

$$\ddot{P} = [\ddot{x}_i \ \ddot{y}_i \ \ddot{z}_i]^T,$$

e dove \ddot{E} è il vettore delle accelerazioni angolari del Sistema Inerziale,

$$\ddot{E} = [\ddot{\phi} \ \ddot{\theta} \ \ddot{\psi}]^T.$$

Riprendendo l'equazione 1.6 e considerando che:

$$\ddot{P} = \frac{d}{dt}[\dot{P}]$$

$$\ddot{E} = \frac{d}{dt}[\dot{E}] \quad (1.13)$$

la 1.12 si può scrivere:

$$m \cdot \frac{d}{dt}[R_b^i \cdot V_b] = R_b^i \cdot \sum F_b \quad (1.14)$$

dove F_b =Forze sul sistema solidale con il drone.

Sviluppando le derivate si ha:

$$m \cdot \left(\frac{d}{dt}[R_b^i] \cdot V_b + R_b^i \cdot \dot{V}_b \right) = R_b^i \cdot \sum F_b \quad (1.15)$$

dove \dot{V}_b è il vettore delle accelerazioni lineari nel sistema solidale con il drone, $\dot{V}_b=[\dot{u} \ \dot{v} \ \dot{w}]$.

Ricordando che la velocità lineare di un sistema di riferimento ortogonale in rotazione è uguale al prodotto vettoriale tra la sua velocità angolare e la rotazione stessa, cioè:

$$\frac{d}{dt}[R_b^i] = \Omega_b \times R_b^i \quad (1.16)$$

Si può ricavare:

$$m \cdot R_b^i \cdot (\Omega_b \times V_b + \dot{V}_b) = R_b^i \cdot \sum F_b \quad (1.17)$$

Semplificando si ottiene:

$$m \cdot (\Omega_b \times V_b + \dot{V}_b) = \sum F_b \quad (1.18)$$

Con procedimento analogo, la Legge fondamentale della Dinamica rotazionale si può scrivere:

$$I \cdot (\Omega_b \times \Omega_b + \dot{\Omega}_b) = \sum \tau_b \quad (1.19)$$

dove $\dot{\Omega}_b$ è il vettore delle accelerazioni angolari nel sistema solidale con il drone, $\dot{\Omega}_b=[\dot{p} \ \dot{q} \ \dot{r}]$

e τ_b =Momento torcente sul sistema solidale con il drone.

Esprimendo la 1.18 e la 1.19 sotto forma di sistema di equazioni si ottiene:

$$\begin{cases} F_x = m \cdot (\dot{u} + qw - rv) \\ F_y = m \cdot (\dot{v} + ru - pw) \\ F_z = m \cdot (\dot{w} + pv - qu) \end{cases} \quad (1.20)$$

$$\begin{cases} \tau_x = \dot{p}I_x + qr(I_z - I_y) \\ \tau_y = \dot{q}I_y + pr(I_x - I_z) \\ \tau_z = \dot{r}I_z + pq(I_y - I_x) \end{cases} \quad (1.21)$$

Il prossimo passo è quello di capire quali forze agiscono sul drone, ovvero la forza peso e la spinta dei motori.

Per la forza peso si usa la classica formula $P=mg$.

Per la spinta si considera il seguente modello semplificato:

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

dove b rappresenta il coefficiente di spinta dei motori e $\omega_1, \omega_2, \omega_3$ e ω_4 rappresentano le velocità di rotazione di ciascun motore.

Considerando che la spinta dei motori risulterà sempre perpendicolare al piano contenente il drone e che la forza peso avrà direzioni diverse in base alle rotazioni (tenendo presente come riferimento il sistema solidale al drone), si può scrivere l'equazione 1.20 come:

$$R_b^i \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (1.22)$$

Il segno meno è spiegato dall'orientamento scelto per l'asse z nei sistemi di riferimento.

Riorganizzando e sviluppando i prodotti, si ottiene:

$$\begin{cases} \dot{u} = rv - qw - g \sin \theta \\ \dot{v} = pw - ru - g \cos \theta \sin \phi \\ \dot{w} = qu - pv + g \cos \phi \cos \theta - \frac{T}{m} \end{cases} \quad (1.23)$$

Per rimodulare l'equazione 1.21, si considera l'influenza di ciascuna manovra intorno ad un asse e, avendo i motori dello stesso asse velocità diverse, si crea una coppia che produce i momenti così definiti:

$$\tau_x = lb(\omega_2^2 - \omega_4^2)$$

$$\tau_y = lb(\omega_1^2 - \omega_3^2)$$

$$\tau_z = ld(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)$$

dove l è la proiezione dei bracci del drone sugli assi e d è il coefficiente di resistenza dell'aria.

Sostituendo e sviluppando si ottiene:

$$\begin{cases} \dot{p} = \frac{lb}{I_x}(\Omega_2^2 - \Omega_4^2) - qr \frac{I_z - I_y}{I_x} \\ \dot{q} = \frac{lb}{I_y}(\Omega_1^2 - \Omega_3^2) - pr \frac{I_x - I_z}{I_y} \\ \dot{r} = \frac{ld}{I_z}(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) - pq \frac{I_y - I_x}{I_z} \end{cases} \quad (1.24)$$

Le 12 equazioni così trovate descrivono un modello non lineare e semplificato della dinamica del quadricottero, modello che ha avuto la maggior fortuna negli anni, in quanto riesce a bilanciare la complessità con l'attinenza alla realtà.

Per completezza in questa ultima parte del capitolo vengono accennati alcuni degli effetti non considerati nel modello semplificato e il loro contributo alla dinamica del sistema.

[4][5]

Forza di attrito.

Trascurata nella modellazione, in quanto per basse velocità ha un contributo poco significativo rispetto alle altre forze in gioco, la forza di attrito viscoso rappresenta

la resistenza dell'aria al moto del drone:

$$F_d = \frac{1}{2} \rho D A V_b^2 \quad (1.25)$$

dove, F_d è la Forza di attrito viscoso;
 ρ rappresenta la densità dell'aria;
 D il coefficiente di resistenza;
 A la sezione di area superiore del drone.

Blade Flapping.

È causato dal contrasto dell'aria che genera differenti velocità della lama dell'elica che avanza rispetto a quella in ritirata, causando uno squilibrio e oscillazioni; inoltre, l'ala viene piegata di un certo angolo (vedi Figura 1.6⁸) e se non è ben allineata con il CoG, si genera un momento che va a degradare le prestazioni dei controllori.

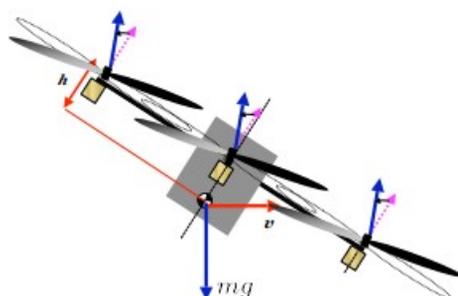


Figura 1.6. Effetto del Blade Flapping

Il fenomeno è trascurato per velivoli di ridotte dimensioni poiché non comporta variazioni sostanziali della spinta dei motori.[2]

Il suo contributo modifica il calcolo della spinta nella seguente equazione:

$$T_i = b \rho A_r R^2 \omega_i^2 \begin{bmatrix} -\sin a_i \\ \cos a_i \sin b_i \\ -\cos b_i \cos a_i \end{bmatrix} \quad (1.26)$$

dove T_i è la spinta dell' i -esimo motore;
 A_r è l'area del rotore;
 R è il raggio del rotore;
 ω_i è velocità angolare dell' i -esimo motore;
 a_1 è l'angolo di flapping longitudinale e b_1 l'angolo di flapping laterale.

Rolling Moment.

⁸Immagine presa da: "Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering di Huang".

Si genera questo fenomeno, strettamente collegato al blade flapping, quando la forza non è applicata nel CoG, causando una rotazione. Si introduce l'advance ratio per il calcolo del contributo di questo effetto, definito come:

$$\mu_i = \frac{\|v_{(1,2)i}\|}{\omega_i r} \quad (1.27)$$

dove μ_i è l'Advance Ratio, $v_{(1,2)i}$ rappresenta la velocità effettiva del drone, ω_i e r sono già state definite in precedenza.

L'angolo di inclinazione che si viene a creare è :

$$\eta = \arctan\left(\frac{v_{(2)i}}{v_{(1)i}}\right) \quad (1.28)$$

Gli angoli di orientamento che si formano sono descritti dalle equazioni:

$$a_i = \frac{1}{1 - \frac{\mu_i^2}{2}} \mu_i (4\theta - 2\lambda^2) \quad (1.29)$$

$$b_i = \frac{1}{1 + \frac{\mu_i^2}{2}} \left(\frac{8\gamma b \mu_i}{9\alpha\sigma} + \frac{b}{2\mu_i} \right) \quad (1.30)$$

dove θ rappresenta l'angolo di pitch rispetto alla direzione di avanzamento, λ è Inflow Ratio ovvero il flusso d'aria $\approx \sqrt{\frac{b}{2}}$, γ è un coefficiente legato alla geometria del drone $= \frac{\rho \alpha r^4}{I_b}$, α è l'angolo di salita, σ è il rapporto tra superficie spazzata dalle eliche e superficie del rotore.

Per il cambio di coordinate tra i vari sistemi si usa la matrice:

$$\mathbf{J} = \begin{pmatrix} \cos \eta & -\sin \eta \\ \sin \eta & \cos \eta \end{pmatrix} \quad (1.31)$$

[11]

Fenomeni aerodinamici e effetto giroscopico.

Fenomeni aerodinamici trascurati sono le perturbazioni legate al vento e l'effetto suolo, cioè, come il suolo interferisce con il flusso d'aria intorno all'oggetto, tendendo a far ritardare l'atterraggio per la pressione creatasi tra l'oggetto e il suolo stesso; anche questi effetti sono trascurabili per le ridotte dimensioni del drone in questione. Inoltre è trascurato l'effetto giroscopico che genera un Moto di precessione, causato quando una forza che agisce sul piano dell'asse intorno al quale un corpo sta ruotando, porta l'asse stesso a cambiare orientamento.[7]

Capitolo 2

Modello in Simulink

Lo scopo di questo capitolo è analizzare il modello a blocchi *Simulink*¹ all'interno del "*Simulink Support Package for Parrot Minidrones*", che, oltre alla dinamica del drone della Parrot (con una modellazione analoga a quella descritta nel Capitolo precedente), implementa dei controllori di tipo PID e ne permette la simulazione tramite una visualizzazione grafica 3D.

Come primo punto è stato necessario installare l'ambiente di sviluppo Matlab e l'integrazione di Simulink e tutte le integrazioni necessarie per il corretto funzionamento del pacchetto, attraverso il sito della MathWorks. In seguito, se installati correttamente tutti i toolbox richiesti, è possibile aprire il progetto con il comando *asbQuadcopterStart*.

Il sistema presenta la classica struttura in Controreazione ed è diviso in più blocchetti (Figura 2.1):

- **Signal Editor:** generatore dei segnali di riferimento in ingresso;
- **Sensors:** contenente le informazioni provenienti dai sensori del drone e le costanti di trasduzione;
- **Flight Control System:** parte di controllo del sistema;
- **Environment:** contenente le variabili di ambiente;
- **Airframe:** modello del processo;
- **Visualization:** utile per visualizzare i tracciati dei segnali di controllo e per simulare in grafica 3D il comportamento del drone;
- **Instrument Panel:** pannello di controllo.

[6][9]

¹Simulink è un software, integrato con MATLAB, per la modellazione, simulazione e analisi di sistemi dinamici, sviluppato dalla MathWorks.

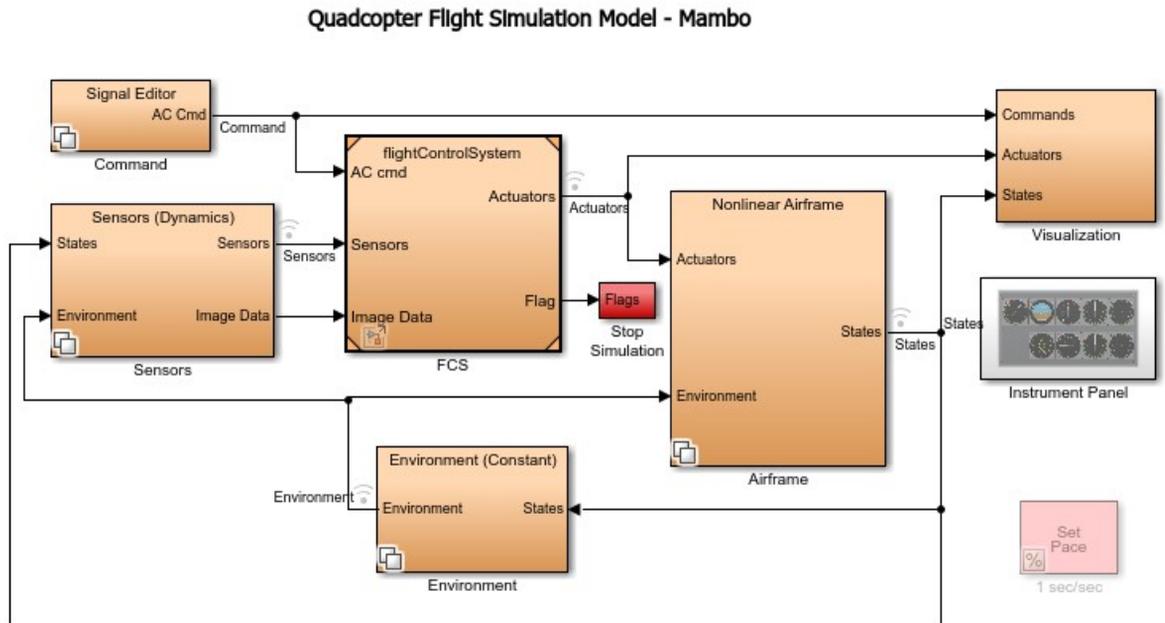


Figura 2.1. Progetto Simulink dal pacchetto per mini droni

2.1 Signal Editor

Questo blocco è responsabile della generazione dei segnali di input dati al sistema, consentendo di modificarne la fonte tramite la variabile $VSS_COMMAND$ direttamente dal *Workspace*.

Se $VSS_COMMAND=0$ si selezionerà il costruttore di segnali di default, di maggior interesse per gli scopi di questo lavoro. La struttura del blocco è rappresentata nella Figura 2.2. Si può notare che i segnali di riferimento X,Y,Z del blocco *Position/Attitude Reference* (in blu in figura) costituiscono le informazioni per la posizione (pos_ref), mentre $Yaw, Pitch, Roll$ i riferimenti per l'orientamento nello spazio ($orient_ref$). Si possono cambiare gli scenari e l'andamento dei segnali di ingresso attraverso l'interfaccia del signal editor, all'interno del blocco *Position/Attitude Reference*. Lo scenario usato di default è quello "*Hover*", ovvero del mantenimento del drone a una certa altezza. Inoltre il blocco in verde indica che quando i segnali di $Pitch$ e $Roll$ sono uguali a zero viene controllata la posizione, in caso contrario viene controllato l'orientamento. Inoltre il signal editor fornisce il segnale di clock di riferimento per l'intero sistema.

Se $VSS_COMMAND=1$ si potrà controllare manualmente i riferimenti del drone attraverso la modalità *Joystick* ma rimane comunque attivo il signal editor per le variabili non controllate direttamente.

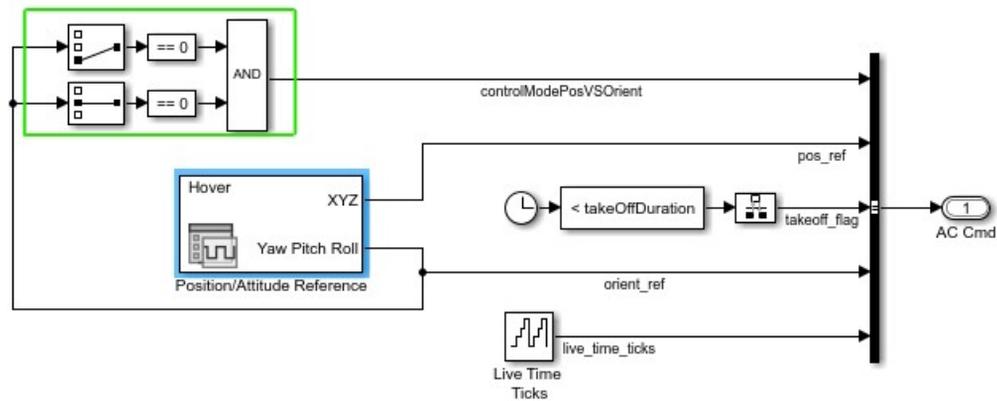


Figura 2.2. Generatore di segnali di default

Se $VSS_COMMAND=2$ i riferimenti delle variabili verranno presi direttamente da un file .mat contenente i suddetti segnali.

Se $VSS_COMMAND=3$ consente di prendere i riferimenti da un foglio elettronico.

2.2 Sensors

Il blocco dei sensori modella la dinamica dei diversi sensori presenti sul drone, elabora i dati provenienti dall'ambiente, li traduce in segnali di altra natura e, infine, li salva nelle variabili del gruppo *sensorVars*. In particolare usa i sensori IMU per le misure delle accelerazioni traslazionali e angolari, la fotocamera per la stima del flusso ottico e il SONAR per la stima dell'altitudine.

Inoltre, è possibile selezionare il modello del sensore con la variabile $VSS_SENSORS$. Se $VSS_SENSORS=0$ i sensori usati non terranno conto del disturbo in retroazione, mentre selezionando $VSS_SENSORS=1$ i sensori saranno dinamici e si terranno in considerazione tali disturbi, poi corretti nel controllore. Quest'ultima è l'opzione di default in quanto più rispecchia la realtà.

All'interno del blocco *Sensor (Dynamics)* ci sono altri due sottoinsiemi:

IMU_Pressure, come in Figura 2.3, che elabora i dati provenienti dai giroscopi, dagli accelerometri, dal sensore di pressione e dal Sonar. In particolare, il blocco di colore rosso, si occupa di trasformare i dati dei sensori, relativi al sistema di riferimento solidale con il drone, nei rispettivi valori nel sistema inerziale, con l'ausilio della Direction Cosine Matrix (contenuta nell'ingresso DCM_be) e con il riferimento della posizione del centro di gravità (*CG location*).

Camera, in Figura 2.4, elabora i dati dalla fotocamera con la tecnica del flusso

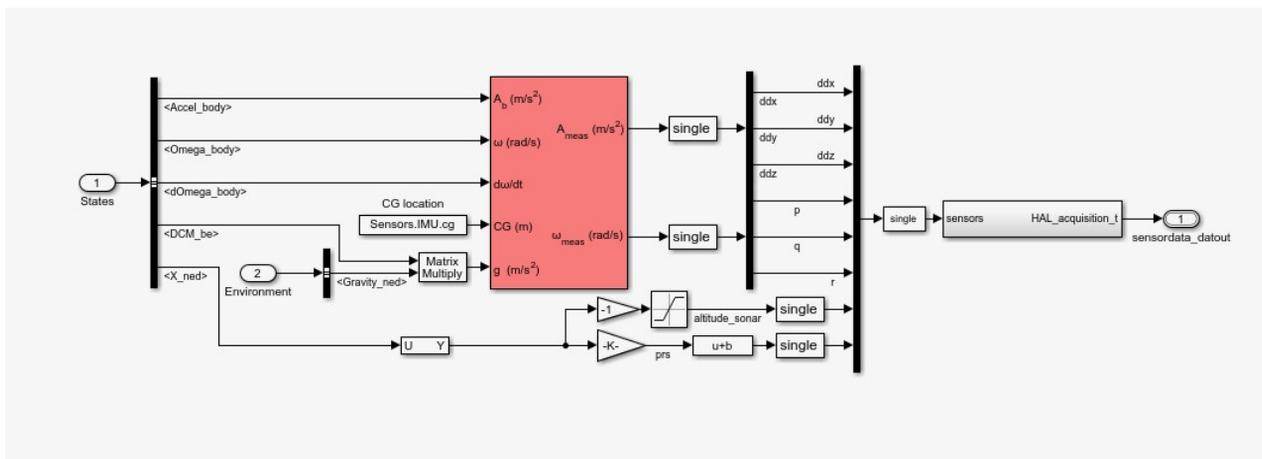


Figura 2.3. Blocco del sensore IMU

ottico. Questo modello è solo una simulazione di quello inaccessibile all'interno del firmware del drone ed è molto lontano dal quello reale. [6][9]

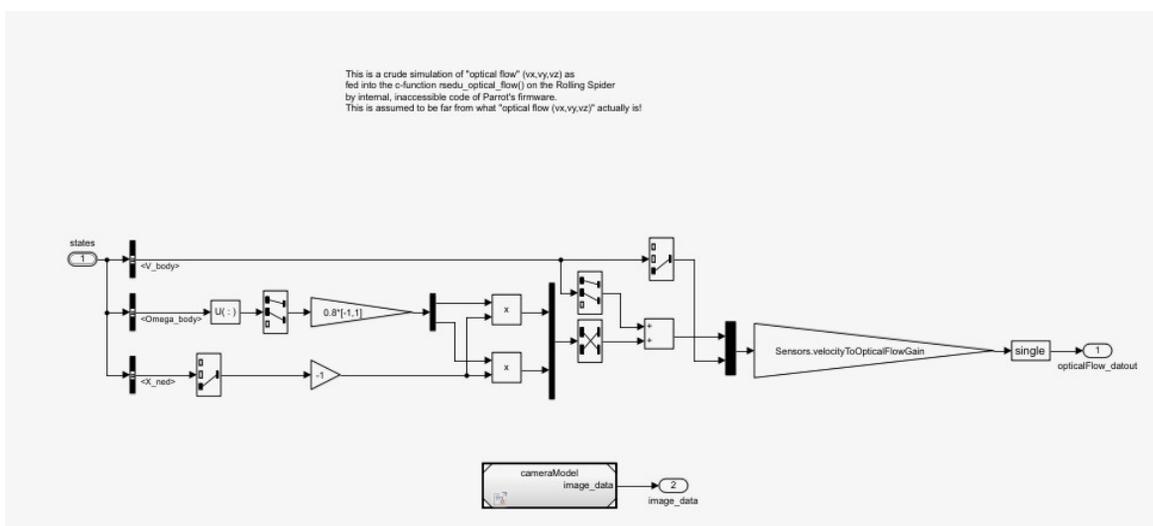


Figura 2.4. Blocco "camera"

2.3 Flight Control System

Questo blocco rappresenta tutta la parte di controllo del sistema, ovvero una delle parti fondamentali, dove si incentra principalmente l'attività descritta in seguito. Ha come segnali di ingresso i riferimenti generati dal signal editor e le variabili di stato in retroazione, date dai sensori. Mentre in uscita fornisce i segnali per il controllo degli attuatori che generano azioni diverse in base alle manovre da fare, e un flag, ovvero predittore di scontri.

Andando ad aprire il Flight Control System, Figura 2.5, è possibile notare che è

diviso in più sottosistemi che hanno compiti diversi.
 In particolare il blocco *sensordata_group* si occupa meramente di dividere i dati

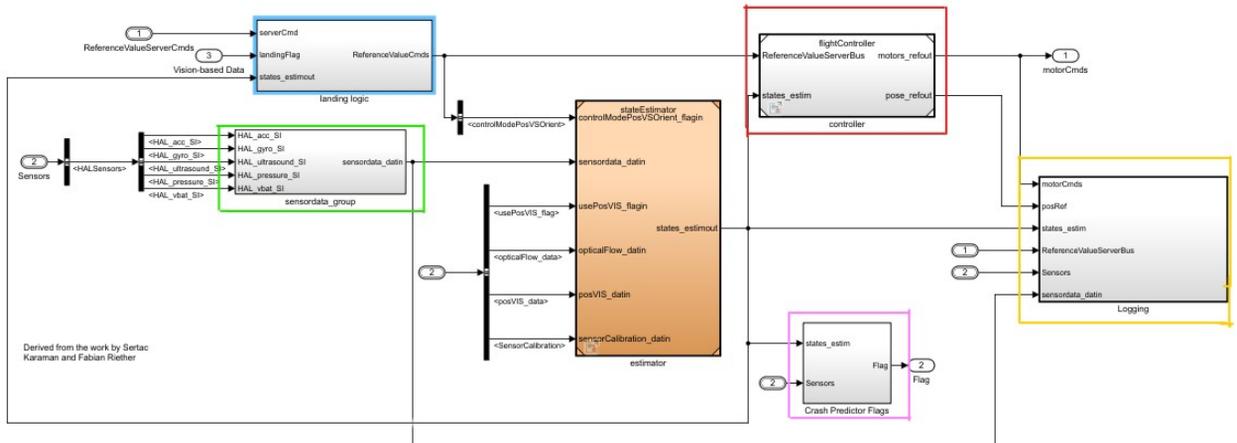


Figura 2.5. Blocco Flight Control System

nel bus provenienti dai sensori senza apportare alcune modifiche particolari.

Il *landing logic* sovrascrive i comandi dei riferimenti provenienti dal signal editor con i comandi di atterraggio, se viene attivato il corrispondente flag, Figura 2.6.

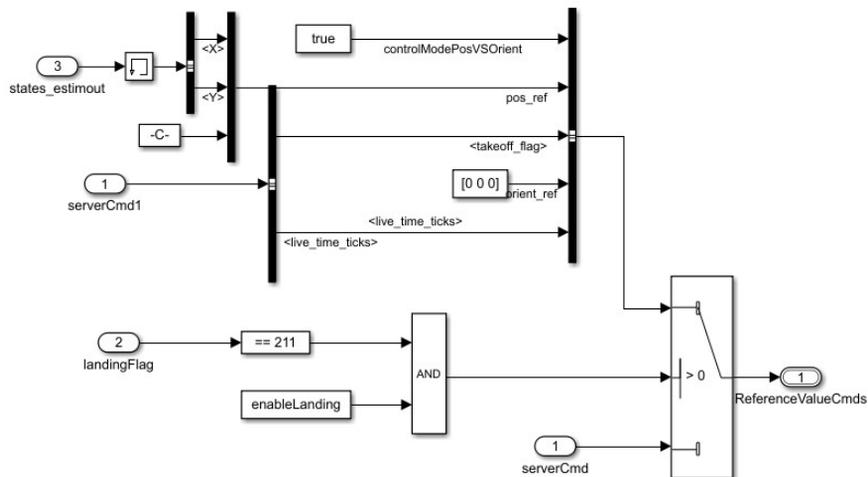


Figura 2.6. Blocco Landing logic

Il *Crash Predictor Flags*, in Figura 2.7, confronta i dati provenienti dai sensori e dallo stimatore e li confronta con dati prestabiliti, interrompendo la simulazione in sicurezza se il controllore non riesce a svolgere bene l'azione di controllo e se ci sono delle oscillazioni elevate che portano il drone ad avere un comportamento

instabile e pericoloso oppure se l'altitudine è troppo elevata, rischiando di scontrarsi rovinosamente con altri oggetti o cadere al suolo.

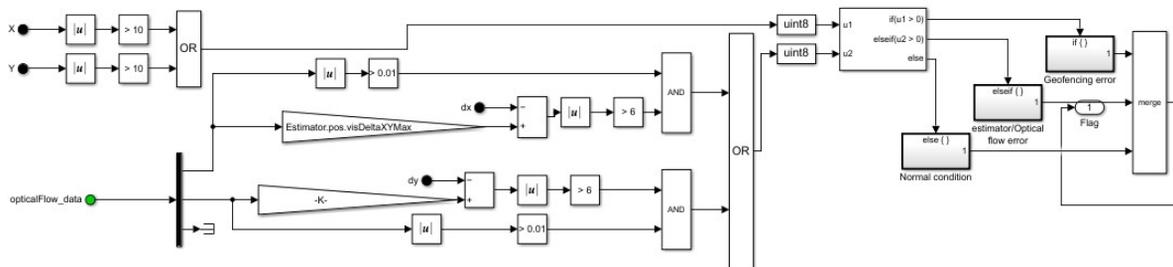


Figura 2.7. Crash Predictor Flags

Il blocco *Logging* memorizza in un file *.mat* alcuni dati come i riferimenti delle posizioni e i comandi degli attuatori, utili per verificare l'andamento delle variabili e le cause di arresto.

Lo *stateEstimator* (Figura 2.8) genera le 12 grandezze di stato, stimate rielaborando i dati provenienti dai sensori. Si divide, a sua volta in altri sottoinsiemi:

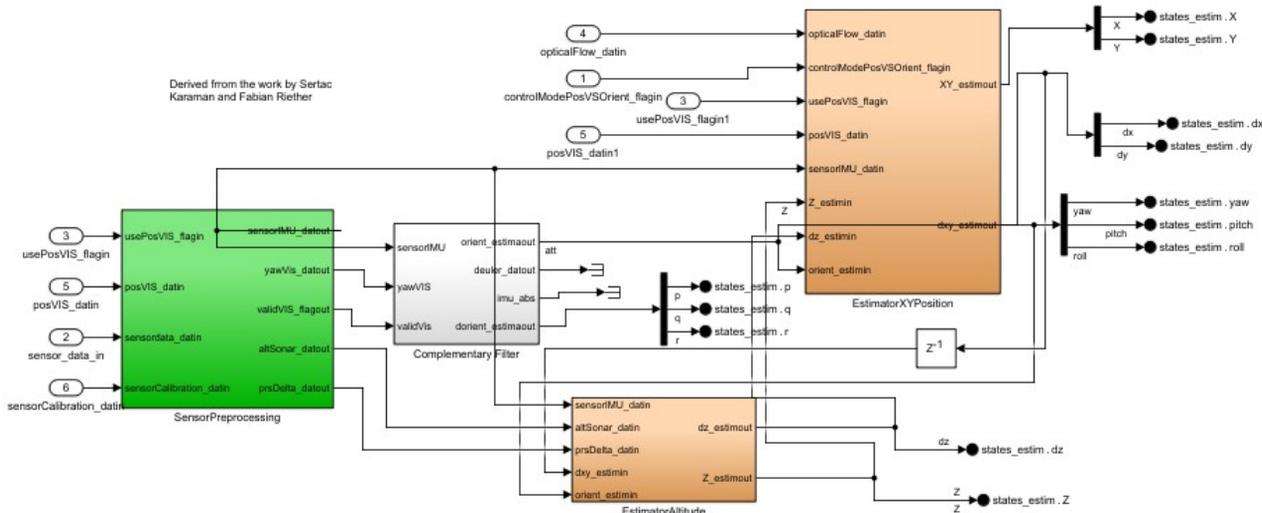


Figura 2.8. Blocco State Estimator

- *Sensor Preprocessing*, dove i dati provenienti dall'IMU vengono calibrati sottraendo i bias precedentemente calcolati. Infine vengono filtrati con un filtro

passa-basso per rimuovere il rumore ad alte frequenze. Inoltre viene controllato se i dati dal sensore fotocamera hanno una valida posizione settando un flag di validità.

- *Complementary Filter*, prende in ingresso i dati del preprocessing e stima Roll, Pitch e Yaw, usando un filtro complementare, ovvero combina i segnali lenti degli accelerometri con quelli più veloci dei giroscopi.
- *Estimator Altitude* stima l'altitudine attraverso un Filtro di Kalman, confrontando i dati delle misure con una predizione del comportamento del sistema.
- *Estimator XY Position* similmente allo stimatore di altitudine, stima le posizioni e le velocità con un Filtro di Kalman.

Infine il blocco *Flight Controller* è il responsabile dell'intera azione di controllo del sistema. Prende in ingresso i segnali di riferimento e li confronta con quelli stimati in modo da ottenere l'errore del sistema (ingresso dei controllori di tipo PID). Prima di andare a descrivere i controllori, è opportuno parlare dei blocchi

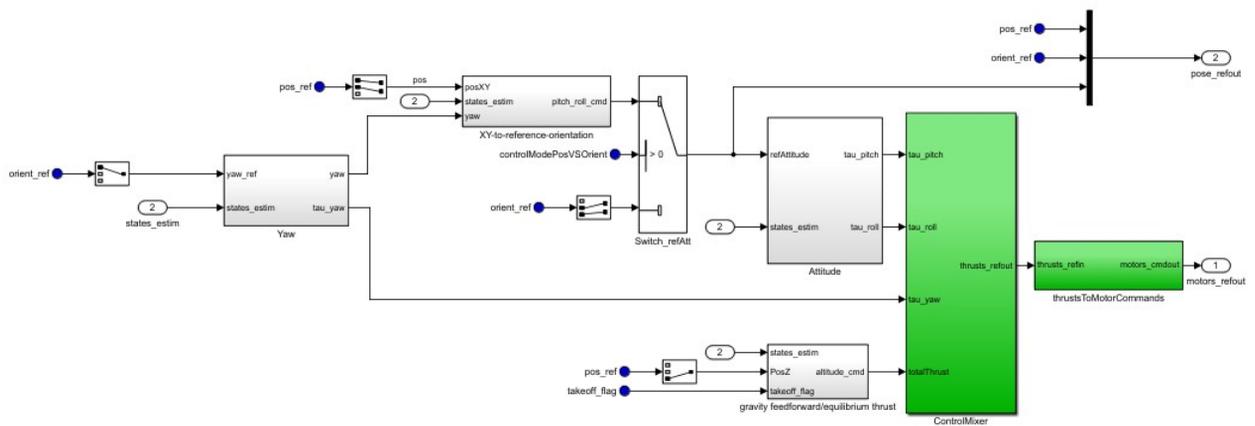


Figura 2.9. Blocco Flight Controller

Switch_refAtt, *ControlMixer* e *thrustsToMotorCommands* (Figura 2.9).

Il primo blocco consente di cambiare il riferimento dalla posizione a quello di orientamento; il secondo blocco prende in input i segnali uscenti dai vari controllori, ovvero le forze e i momenti da applicare ai motori, convertendo la spinta in termini di velocità e moltiplicando tutti i valori per la matrice $Q2Ts$. Infine il blocco *thrustsToMotorCommands* moltiplica l'uscita del *ControlMixer* con delle costanti di saturazione, che proteggono i motori da valori troppo elevati, e con un vettore che indica il senso di rotazione delle eliche.

2.3.1 Controllore per il Pitch e il Roll

Il controllo usato per il Pitch e per il Roll è un controllo a cascata (quando lo *Switch_refAtt* prende il riferimento della posizione), formato da un primo controllore implementato nel blocco *XY-to-reference-orientation* e un secondo controllore

nel blocco *Attitude*. Questa particolare scelta è spiegata dal fatto che il quadricottero è sotto-attuato. Quando lo switch prende come riferimento l'orientamento dato dal *Signal Editor*, l'unico controllore diventa quello *Attitude*.

Blocco *XY-to-reference-orientation*. In Figura 2.10, il blocco prende in ingresso l'angolo di Yaw, i valori di X e Y, le velocità e i corrispondenti valori di riferimento e sviluppa un comando di controllo che indica il valore che devono assumere il Roll e il Pitch. Il segnale di Yaw viene usato per calcolare le posizioni

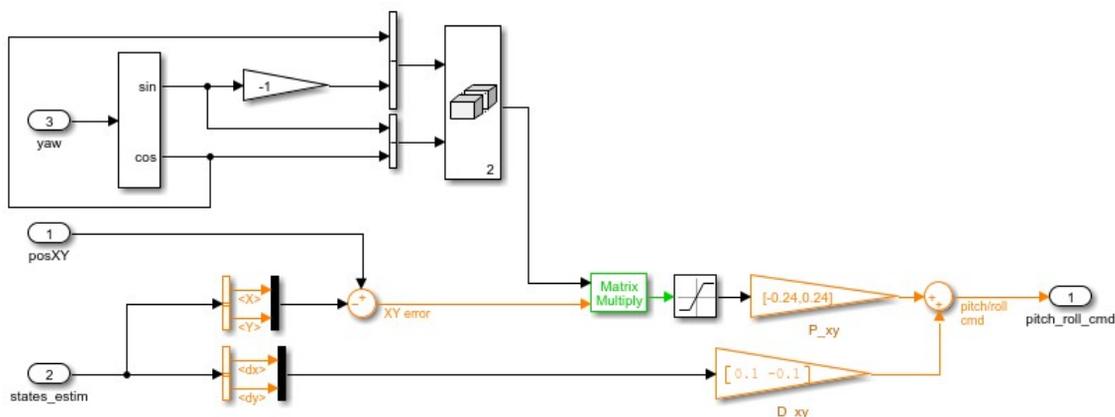


Figura 2.10. Blocco XY-to-reference-orientation

X e Y rispetto al sistema di riferimento inerziale, ottenute usando la matrice di rotazione rispetto a Z. I valori ottenuti vengono poi moltiplicati per l'errore sulle posizioni, calcolato sottraendo il riferimento con i valori stimati. Infine, il valore risultante e la stima delle velocità vengono moltiplicati per le costanti del controllore Proporzionale-Derivativo, dove la parte derivativa è eseguita sulle velocità e non sull'errore.

Blocco *Attitude*. Questo blocco, in Figura 2.11, prende in ingresso i valori di

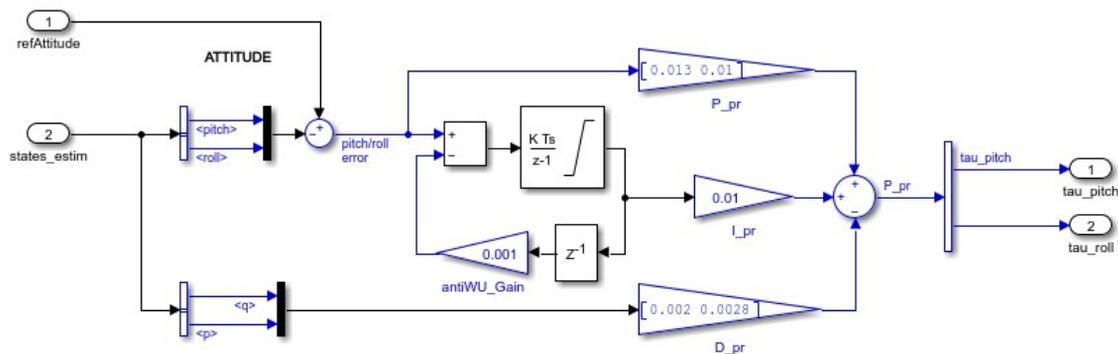


Figura 2.11. Blocco Attitude

riferimento per il Pitch e il Roll o dal blocco *XY-to-reference-orientation* o direttamente dal *Signal Editor*, in base alla posizione dello switch che seleziona il primo ingresso se i valori nel *Signal Editor* sono zero, il secondo altrimenti; prende, inoltre, i valori stimati dei due angoli e delle loro velocità angolari, confrontandoli con il riferimento, ottenendo, così, l'errore. In seguito, tale segnale viene controllato attraverso un'azione Proporzionale-Integrativa, e, infine, viene sommato alle velocità angolari, moltiplicate per il blocco Derivativo. Si può notare che viene aggiunto un polo al sistema, per evitare il cosiddetto Integral Wind Up.

2.3.2 Controllore per l'angolo di Yaw

Analogamente ai controllori precedenti, il blocco *Yaw* (Figura 2.12) riceve in ingresso il riferimento per l'angolo e la sua stima. Il controllore è di tipo Proporzionale-Derivativo e al segnale di uscita viene aggiunto la velocità moltiplicata al blocco derivativo, dando in uscita il segnale di controllo per lo Yaw.

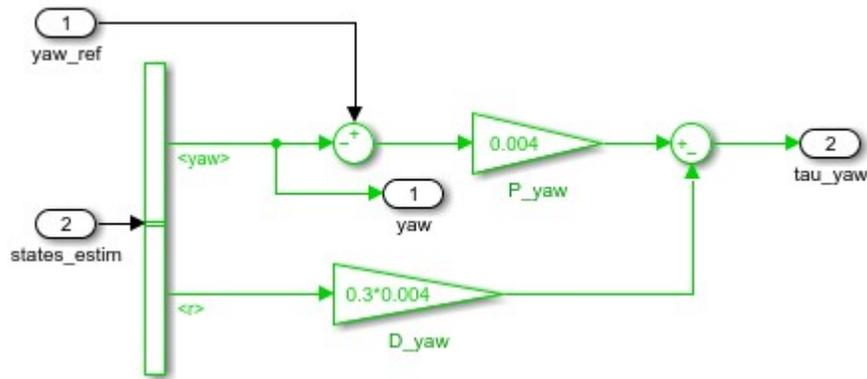


Figura 2.12. Blocco Yaw

2.3.3 Controllore per l'altitudine

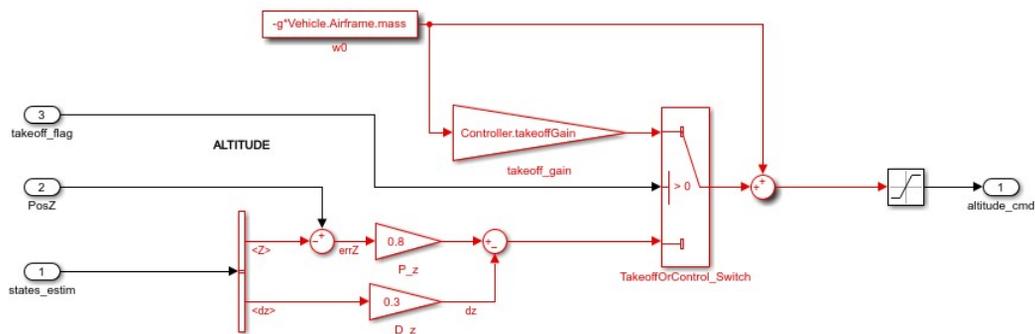


Figura 2.13. Blocco Gravity Feedforward/Equilibrium Thrust

Il blocco *gravity feedforward/equilibrium thrust*, in Figura 2.13, prende in ingresso la posizione verticale di riferimento, la sua stima e la stima della velocità lineare lungo z e dà in uscita il segnale di controllo per l'altitudine. Inoltre, è presente uno switch, regolato dal Take Off flag. Quando il flag è positivo, avviene la modalità di decollo che regola attraverso un gain, la spinta dei motori. Finito il decollo viene selezionato il sistema di controllo PD.

2.4 Environment

Il blocco *Environment* modella le variabili di ambiente, ovvero il vettore gravità, la pressione atmosferica, la temperatura dell'aria, la densità dell'aria, la velocità del suono e il campo magnetico, utili per il modello del processo e per i sensori. Inoltre, è possibile attraverso il comando *VSS_ENVIRONMENT*, cambiare il modello delle variabili di ambiente. Se *VSS_ENVIRONMENT=0*, le variabili sono costanti (opzione di default, in Figura 2.14). Se *VSS_ENVIRONMENT=1*, le variabili variano in base alla posizione.

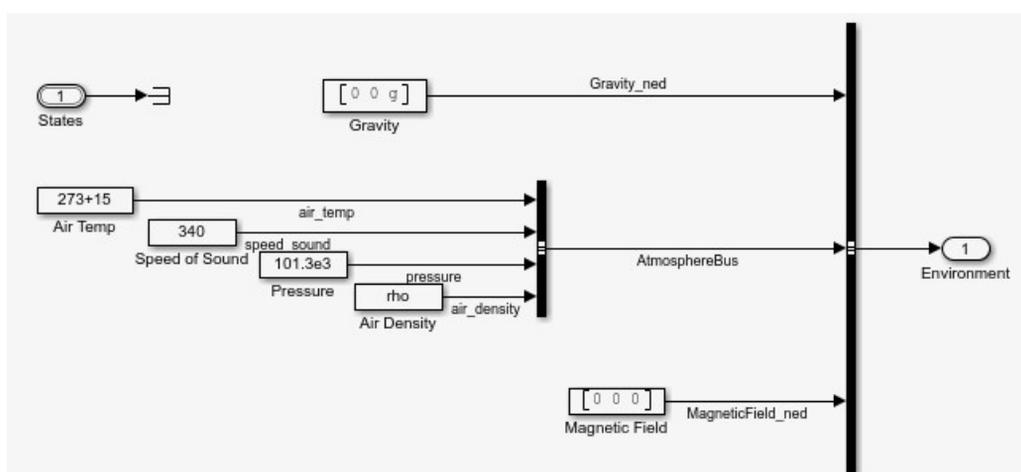


Figura 2.14. Blocco Environment

2.5 Airframe

Il blocco *Airframe* contiene tutte le equazioni matematiche che vanno a definire il modello del processo ed, attraverso la variabile *VSS_VEHICLE*, è possibile scegliere il modello di default non lineare (imponendo la variabile = 1), oppure il modello lineare (variabile = 0). Sia il modello lineare che il non lineare prendono in ingresso le informazioni provenienti dal controllore e dal blocco *Environment* e generano le variabili di stato e le uscite del sistema.

2.5.1 Modello non lineare

Il modello non lineare in Figura 2.15 è diviso in più sottosistemi con funzioni diverse.

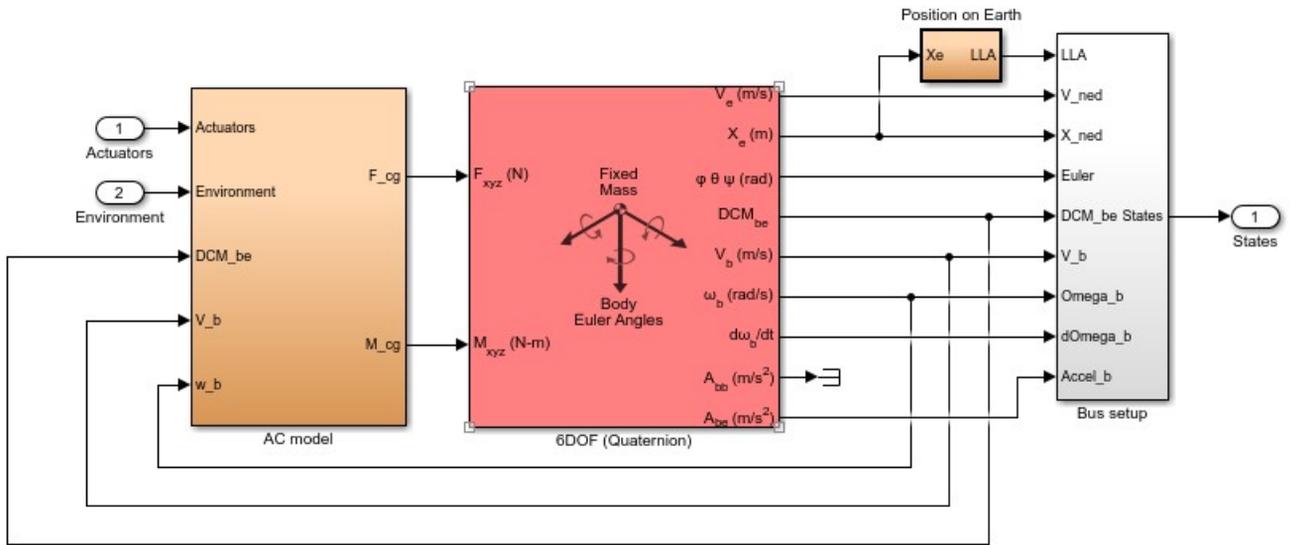


Figura 2.15. Blocco NonLinear Airframe

AC model. Questo blocco (Figura 2.16) contiene il modello matematico del sistema, ampiamente descritto nel capitolo precedente. Prende come segnali di

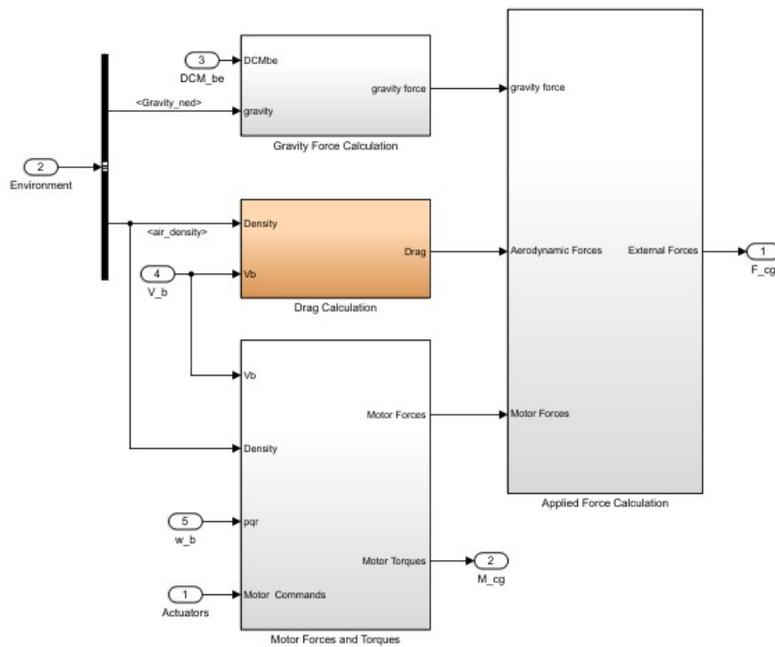


Figura 2.16. Blocco AC model

ingresso l'azione di controllo, le variabili di ambiente e, le velocità lineari e angolari nel sistema di riferimento solidale con il drone e la matrice DCM, provenienti dal blocco *6DOF*. Usando queste informazioni, procede al calcolo delle forze e dei momenti sui 3 assi. Il modello matematico usato tiene in considerazione alcuni effetti descritti nelle ultime pagine del capitolo precedente, in particolare il contributo della forza di attrito dell'aria e l'effetto del Blade Flapping.

La resistenza dell'aria è calcolata nel blocco *Drag Calculation* e nelle Figure 2.17 e 2.18 ne è rappresentata l'implementazione. Il blocco prende in ingresso le ve-

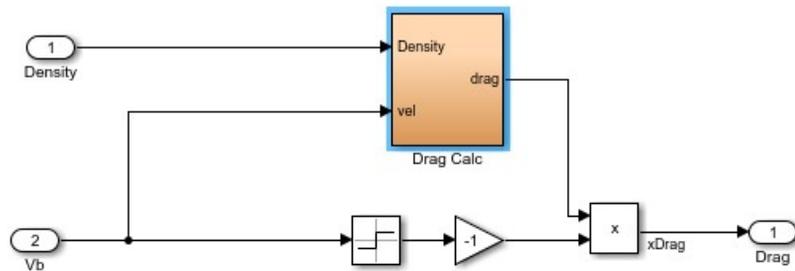


Figura 2.17. Blocco Drag Calculation

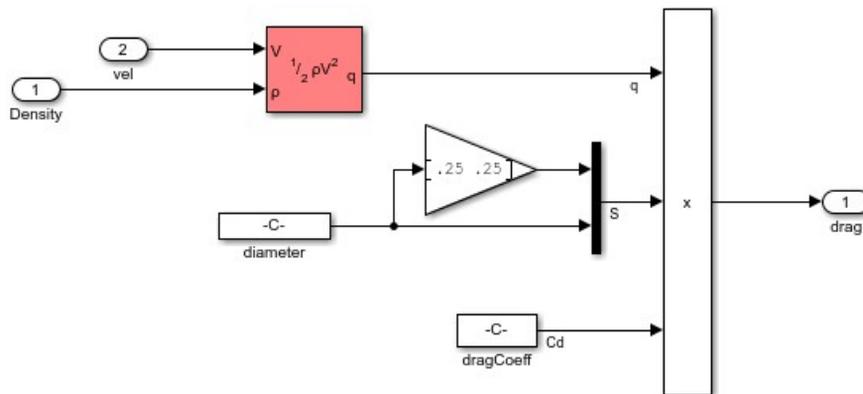


Figura 2.18. Blocco Drag Calc

locità lineari e la densità dell'aria e le moltiplica, secondo l'equazione 1.25, per il coefficiente di drag e la sezione di superficie che impatta con l'aria e dà come risultato la forza di drag, specificando anche il verso della forza con il blocchetto Sign.

La forza di gravità viene calcolata nel blocco *Gravity Force Calculation* in Figura 2.19. Prende in ingresso la costante di gravità e la DCM e calcola la forza di gravità rispetto al Body Fixed Frame.

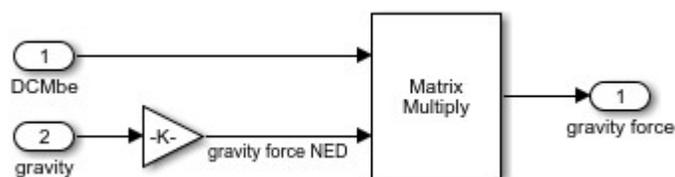


Figura 2.19. Blocco Gravity Force Calculation

I calcoli rappresentanti la forza dovuta dai motori e il calcolo dei momenti sono implementati all'interno del blocco *Motor Forces and Torques*. In particolare il blocco *MotorsToW*, in Figura 2.20, trasforma il segnale proveniente dal controllore nella velocità ω del singolo motore.

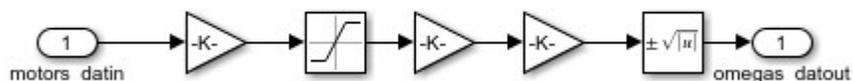


Figura 2.20. Blocco Motors To W

Nel blocco *ForEach/Rotor Dynamics* è presente la dinamica per ogni rotore. Nella Figura 2.21 vengono calcolate le componenti delle velocità (in verde), secondo la formula $v_{ri} = V_b + \Omega_b \times d_i$, dove v_{ri} è la velocità dell' i -esimo rotore rispetto all'asse r -esimo, mentre d_i è la distanza dell' i -esimo rotore dal CoG. Successivamente viene calcolata l'advance ratio con l'equazione 1.27 (in rosso) e l'angolo dell'equazione 1.29 (in blu), tramite la matrice 1.31 (in giallo). Nella Figura 2.22 viene calcolata la spinta dei motori secondo la formula 1.26 e il momento risultante.

Infine nel blocco *Applied Force Calculation* in Figura 2.23, vengono sommate le tre forze calcolate, quella di gravità, l'attrito dell'aria e la spinta dei motori, così da trovare la risultante delle forze esterne dell'equazione 1.18.

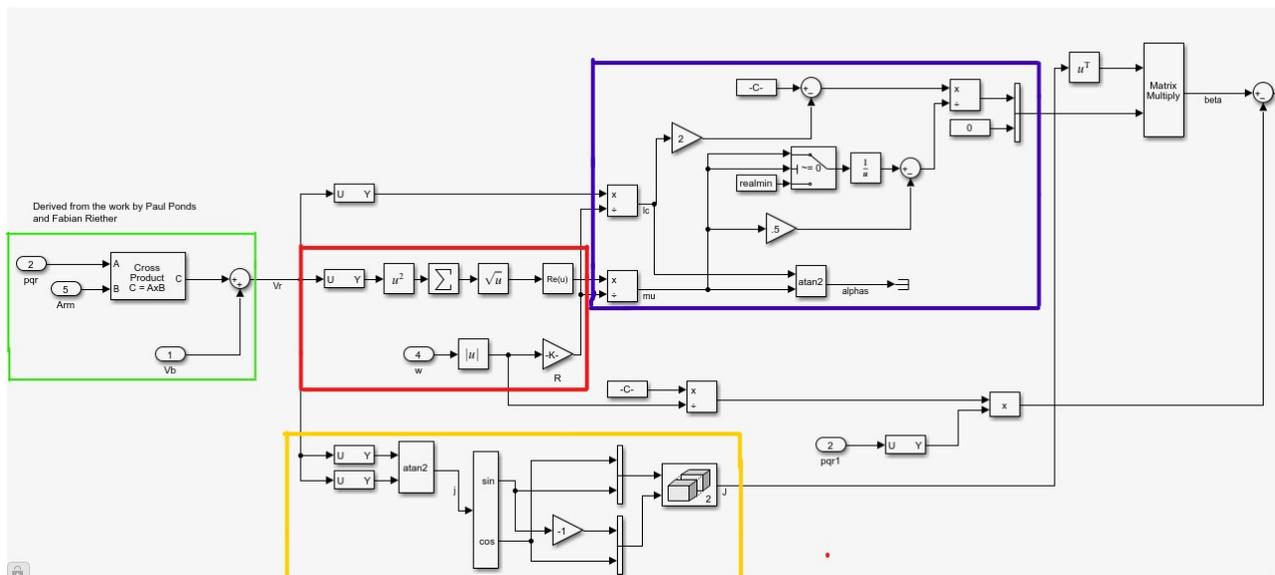


Figura 2.21. Blocco Rotor Dynamics

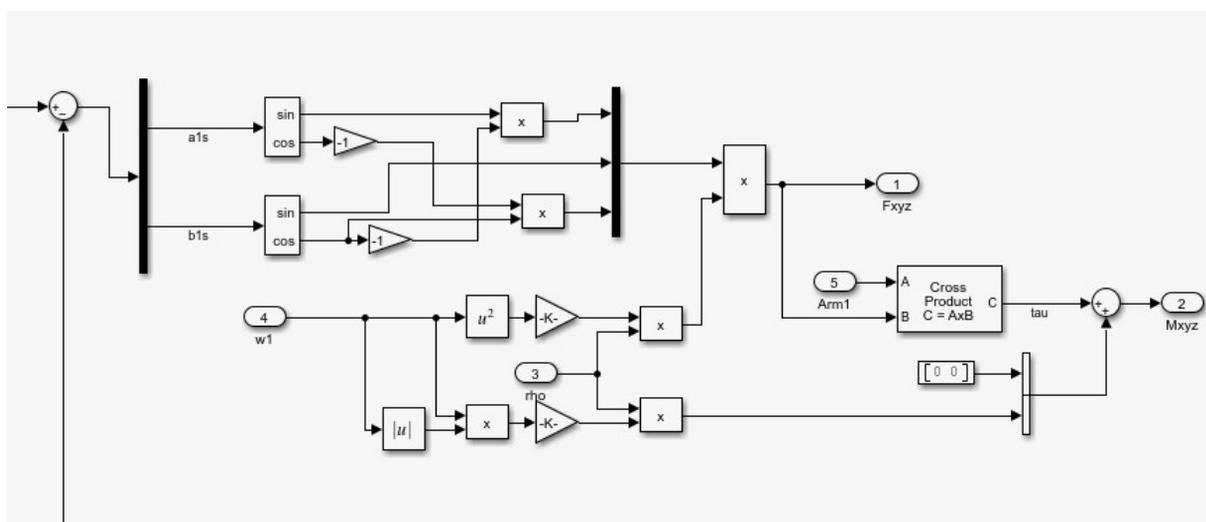


Figura 2.22. Blocco Rotor Dynamics (continuo)

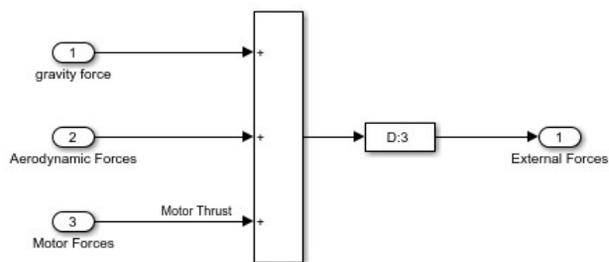


Figura 2.23. Blocco Applied Force Calculation

6OD. Prende in ingresso le forze e i momenti calcolati nel blocco *AC model* e, attraverso le equazioni 1.23 e 1.24 (con l'aggiunta delle forze nominate prima), calcola le uscite del sistema, ovvero le velocità traslazionali, angolari, le posizioni e le accelerazioni sia rispetto al sistema inerziale che a quello solidale al drone. A queste ultime si aggiungono le coordinate di longitudine, latitudine e altitudine calcolate nel blocco *Position on Earth*.

Bus setup. Crea un bus degli stati calcolati nei blocchi precedenti (Figura 2.24).

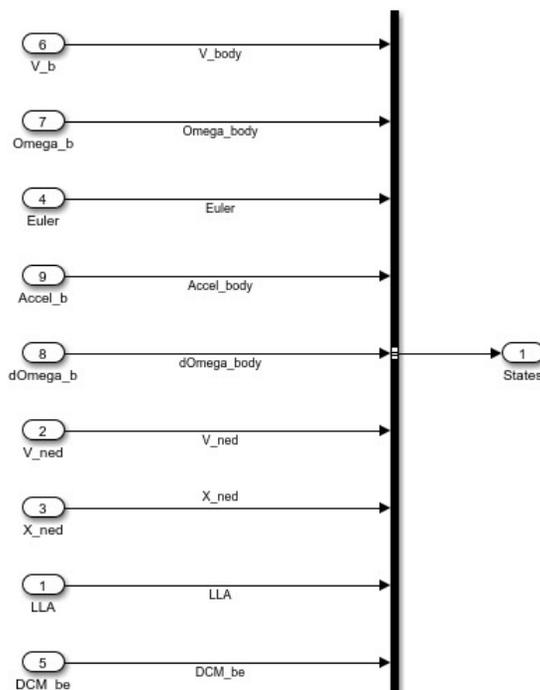


Figura 2.24. Blocco Bus setup

2.5.2 Modello lineare

Prendendo in ingresso le variabili di ambiente e il segnale proveniente dal controllore, il blocco utilizza il modello lineare:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (2.1)$$

generando così le variabili di stato e le uscite del sistema, poi riunite nel Bus setup come già visto per il modello non lineare (Figura 2.25).

I valori sono stati calcolati con la funzione *trimLinearizeOpPoint* che si occupa di linearizzare il modello non lineare vicino ad un punto di equilibrio.

Le matrici A, B, C e D sono chiamate matrici della dinamica e i loro valori sono consultabili nel *Workspace* sotto il nome di *linsys*:

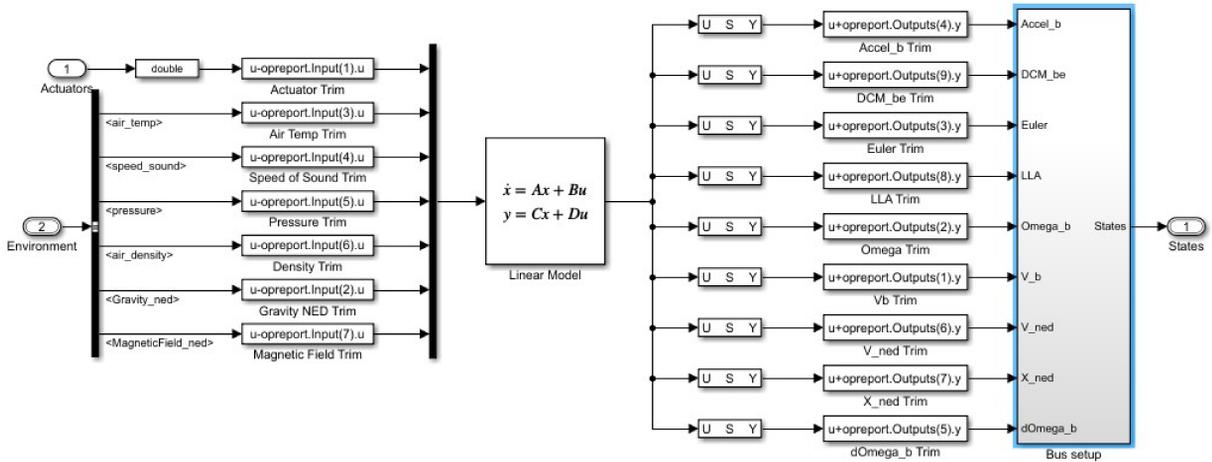


Figura 2.25. Blocco Linear Airframe

- **A** è la matrice della dinamica delle variabili di stato del sistema. Essendoci 12 variabili di stato (posizioni e angoli di Eulero rispetto al Earth Fixed Frame e velocità traslazionali e angolari rispetto al Body Frame), la matrice ha dimensione 12x12, con i valori estrapolati in Figura 2.26, dove la prima riga e la prima colonna rappresentano i nomi degli stati corrispondenti).

State\State	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)
phi theta psi(1)	0	3.6611e-17	0	0	0	0	0	0	0	1	3.0292e-23	-5.2296e-12
phi theta psi(2)	-3.6611e-17	0	0	0	0	0	0	0	0	0	1	5.2295e-12
phi theta psi(3)	-6.8347e-13	-2.0323e-28	0	0	0	0	0	0	0	0	-5.2295e-12	1
ub,vb,wb(1)	0	-9.81	0	-3.4402e-09	-3.4402e-09	6.8532e-13	0	0	0	-5.4616e-11	0.13799	-3.2848e-13
ub,vb,wb(2)	9.81	-2.8478e-22	0	-3.6595e-09	-3.6595e-09	-6.852e-13	0	0	0	-0.13799	5.8098e-11	-3.5159e-13
ub,vb,wb(3)	5.4456e-11	5.4456e-11	0	-6.8347e-13	6.8347e-13	0	0	0	0	3.292e-13	3.5087e-13	0
xe,ye,ze(1)	1.7216e-24	1.3478e-19	3.292e-13	1	-1.4969e-18	-5.2296e-12	0	0	0	0	0	0
xe,ye,ze(2)	-1.3478e-19	2.0176e-37	3.5087e-13	1.4969e-18	1	5.2295e-12	0	0	0	0	0	0
xe,ye,ze(3)	-3.292e-13	-3.5087e-13	0	5.2296e-12	-5.2295e-12	1	0	0	0	0	0	0
p,q,r(1)	0	0	0	-5.759e-08	-5.759e-08	-2.7287e-14	0	0	0	-2.1715	9.1429e-10	4.3301e-13
p,q,r(2)	0	0	0	4.0369e-08	4.0369e-08	-2.1644e-14	0	0	0	6.4089e-10	-1.6192	-5.1363e-13
p,q,r(3)	0	0	0	-1.4434e-19	-1.2831e-19	3.6185e-19	0	0	0	1.2157e-13	1.1242e-13	-6.8807e-09

Figura 2.26. Matrice A

- **B** è la matrice degli ingressi che influiscono sulle variabili di stato. Gli ingressi corrispondono alle variabili di ambiente e agli attuatori(4 attuatori, densità dell'aria, pressione, temperatura, velocità del suono, gravità e campo magnetico sui 3 assi), per un totale di 14. Quindi la dimensione della matrice è 12x14, come in Figura 2.27.

2.5 – Airframe

State <input style="background-color: yellow; color: black; font-weight: bold; font-size: small;"/> Input	Actuators (1)	Actuators (2)	Actuators (3)	Actuators (4)	Atmosph ereBus.air _density	Atmosph ereBus.air _r_temp	Atmosph ereBus.p ressure	Atmosph ereBus.p ed_sound	Gravity_n ed(1)	Gravity_n ed(2)	Gravity_n ed(3)	Magnetic Field_ne d(1)	Magnetic Field_ne d(2)	Magnetic Field_ne d(3)
phi theta psi(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
phi theta psi(2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
phi theta psi(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ub,vb,wb(1)	-6,90E-13	6,90E-13	-6,90E-13	6,90E-13	-9,93E-10	0	0	0	1	1,50E-14	5,23E-08	0	0	0
ub,vb,wb(2)	6,76E-13	-6,76E-13	6,76E-13	-6,76E-13	9,82E-10	0	0	0	-1,50E-14	1	-5,23E-08	0	0	0
ub,vb,wb(3)	0.0096071	0.0096071	-0.0096071	0.0096071	-82.855	0	0	0	0	0	1	0	0	0
xe,ye,ze(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
xe,ye,ze(2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
xe,ye,ze(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p,q,r(1)	0.42019	0.42019	-0.42019	-0.42019	2,53E-10	0	0	0	0	0	0	0	0	0
p,q,r(2)	0.31332	-0.31332	-0.31332	0.31332	1,89E-10	0	0	0	0	0	0	0	0	0
p,q,r(3)	-0.011544	-0.011544	-0.011544	-0.011544	-7,94E-12	0	0	0	0	0	0	0	0	0

Figura 2.27. Matrice B

- **C** rappresenta la matrice delle uscite rispetto alle variabili di stato. Le uscite sono 33 (sono le variabili uscenti dal blocco Airframe, riunite nel bus setup) e la dimensione della matrice è 33x12 (Figura 2.28).

Outputs <input style="background-color: yellow; color: black; font-weight: bold; font-size: small;"/> State	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)	
Accel_body(1)	0	0	0	0	-3,44E-05	-3,44E-05	1,84E-11	0	0	0	-5,46E-07	0,13799	7,21E-12
Accel_body(2)	9,81	-2,85E-18	0	0	-3,66E-05	-3,66E-05	-1,73E-11	0	0	0	-0,13799	5,81E-07	-7,21E-12
Accel_body(3)	5,45E-07	5,45E-07	0	0	0	0	0	0	0	0	0	0	0
DCM_be(1)	0	5,55E-08	0	0	0	0	0	0	0	0	0	0	0
DCM_be(2)	-5,23E-08	-5,23E-08	-1	0	0	0	0	0	0	0	0	0	0
DCM_be(3)	1,50E-14	1	-5,23E-08	0	0	0	0	0	0	0	0	0	0
DCM_be(4)	0	8,31E-26	1	0	0	0	0	0	0	0	0	0	0
DCM_be(5)	5,55E-08	0	0	0	0	0	0	0	0	0	0	0	0
DCM_be(6)	-1	1,50E-14	-5,23E-08	0	0	0	0	0	0	0	0	0	0
DCM_be(7)	0	-1	0	0	0	0	0	0	0	0	0	0	0
DCM_be(8)	1	-2,90E-19	0	0	0	0	0	0	0	0	0	0	0
DCM_be(9)	5,55E-08	5,55E-08	0	0	0	0	0	0	0	0	0	0	0
Euler(1)	1	0	0	0	0	0	0	0	0	0	0	0	0
Euler(2)	0	1	0	0	0	0	0	0	0	0	0	0	0
Euler(3)	0	0	1	0	0	0	0	0	0	0	0	0	0
LLA(1)	0	0	0	0	0	0	9,00E-02	0	0	0	0	0	0
LLA(2)	0	0	0	0	0	0	0	1,21E-01	0	0	0	0	0
LLA(3)	0	0	0	0	0	0	0	0	-1	0	0	0	0
Omega_body(1)	0	0	0	0	0	0	0	0	0	0	1	0	0
Omega_body(2)	0	0	0	0	0	0	0	0	0	0	0	1	0
Omega_body(3)	0	0	0	0	0	0	0	0	0	0	0	0	1
V_body(1)	0	0	0	1	0	0	0	0	0	0	0	0	0
V_body(2)	0	0	0	0	1	0	0	0	0	0	0	0	0
V_body(3)	0	0	0	0	0	1	0	0	0	0	0	0	0
V_ned(1)	1,72E-20	1,35E-15	3,29E-10	1	-1,50E-14	-5,23E-08	0	0	0	0	0	0	0
V_ned(2)	-1,35E-15	2,02E-33	3,51E-09	1,50E-14	1	5,23E-08	0	0	0	0	0	0	0
V_ned(3)	-3,29E-10	-3,51E-09	0	5,23E-08	-5,23E-08	1	0	0	0	0	0	0	0
X_ned(1)	0	0	0	0	0	0	1	0	0	0	0	0	0
X_ned(2)	0	0	0	0	0	0	0	1	0	0	0	0	0
X_ned(3)	0	0	0	0	0	0	0	0	1	0	0	0	0
dOmega_body(1)	0	0	0	-5,76E-05	-5,76E-05	-2,73E-10	0	0	0	-21,715	9,14E-06	4,33E-09	
dOmega_body(2)	0	0	0	4,04E-04	4,04E-04	-2,16E-10	0	0	0	6,41E-06	-16,192	-5,14E-09	
dOmega_body(3)	0	0	0	-1,44E-15	-1,28E-15	3,62E-15	0	0	0	1,22E-09	1,12E-09	-6,88E-05	

Figura 2.28. Matrice C

- **D** rappresenta la matrice di come gli ingressi influiscono sulle uscite, quindi ha dimensione 33x14 (Figura 2.29).

Componente	Actuator1 (1)	Actuator1 (2)	Actuator1 (3)	Actuator1 (4)	Atmospheric.air_density	Atmospheric.air_temp	Atmospheric.pressure	Atmospheric.speed_sound	gravity_ned(1)	gravity_ned(2)	gravity_ned(3)	Magnetics.ned(1)	Magnetics.ned(2)	Magnetics.ned(3)
Arcal_body(1)	-6.96E-13	6.96E-13	-6.96E-13	6.96E-13	0.93E-10	0	0	0	1	1.50E-14	-5.23E-08	0	0	0
Arcal_body(2)	6.76E-13	-6.76E-13	6.76E-13	-6.76E-13	0.82E-10	0	0	0	-1.50E-14	1	-5.23E-08	0	0	0
Arcal_body(3)	-0.0096071	0.0096071	-0.0096071	0.0096071	-8.2E-05	0	0	0	0	0	1	0	0	0
DCM_bat(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DCM_bat(2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DCM_bat(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DCM_bat(4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DCM_bat(5)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DCM_bat(6)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DCM_bat(7)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DCM_bat(8)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DCM_bat(9)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Euler(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Euler(2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Euler(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LLA(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LLA(2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LLA(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Omega_body(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Omega_body(2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Omega_body(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V_body(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V_body(2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V_body(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V_ned(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V_ned(2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V_ned(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
X_ned(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
X_ned(2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
X_ned(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
dOmega_body(1)	0.42019	0.42019	-0.42019	-0.42019	0	0	0	0	0	0	0	0	0	0
dOmega_body(2)	0.31332	-0.31332	-0.31332	0.31332	0	0	0	0	0	0	0	0	0	0
dOmega_body(3)	-0.011544	-0.011544	-0.011544	-0.011544	1.33E-11	0	0	0	0	0	0	0	0	0

Figura 2.29. Matrice D

2.6 Visualization

Si occupa della visualizzazione dei segnali di uscita, ingresso e stato del sistema. Si può scegliere la modalità di visualizzazione attraverso la variabile $VSS_VISUALIZATION$. Se $VSS_VISUALIZATION=0$ la visualizzazione si effettuerà attraverso dei grafici (implementati con i blocchetti di tipo Scope). $VSS_VISUALIZATION=1$ i segnali saranno salvati in una variabile nel *Workspace*. Se $VSS_VISUALIZATION=2$ verranno visualizzati con una animazione di FlightGear (un simulatore di volo non incluso nel software). Infine, se $VSS_VISUALIZATION=3$ verrà avviata la simulazione attraverso Simulink 3D (esempio in Figura 2.30).

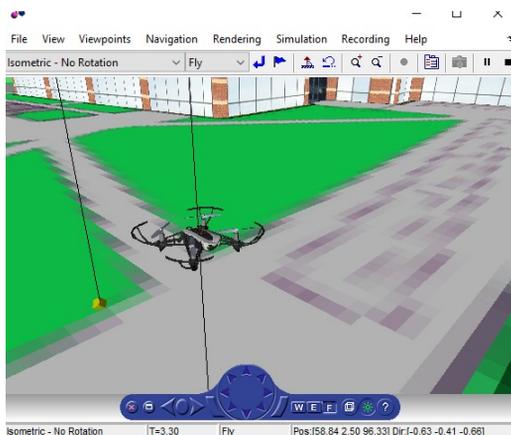


Figura 2.30. Simulazione tramite Simulink 3D

2.7 Instrument Panel

Durante la simulazione è possibile controllare alcuni valori attraverso il Pannello di controllo, in Figura 2.31. Nel pannello sono presenti degli strumenti di volo, ovvero un indicatore di velocità, un altimetro, un indicatore di orientamento relativo alla linea di orizzonte della Terra, gli indicatori di potenza dei quattro motori, un girodirezionale e un indicatore di salita o discesa.



Figura 2.31. Pannello di controllo

Capitolo 3

Progettazione dei controllori

Dopo aver analizzato a fondo il modello matematico che è presente nei blocchetti del progetto di Simulink, è possibile avviare la fase di sintesi dei controllori. In particolare in questo elaborato saranno implementati due controllori per la manovra di rollio, uno ricavato con il metodo del luogo delle radici e l'altro attraverso la sintesi in frequenza.

È necessario evidenziare che in questa fase si farà riferimento al modello lineare del sistema, utile per la sintesi dei controllori, mentre per la simulazione e la verifica si userà il modello non lineare, che meglio descrive la dinamica del drone.

Per procedere alla sintesi dei controllori, è necessario in primo luogo ricavare la funzione di trasferimento del processo, così definita:

$$FdT = C \cdot (sI - A)^{-1}B + D \quad (3.1)$$

dove I è la matrice d'identità, $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ e A , B e C le matrici della dinamica.

Le matrici A , B , C e D , però rappresentano l'intera dinamica del sistema, mentre scopo di questo lavoro è di andare a controllare soltanto la manovra di rollio, quindi è fondamentale estrarre le sotto-matrici di interesse relative al rollio. Per quanto riguarda gli ingressi, sono stati estratti i valori corrispondenti alla dicitura **Actuators**¹, mentre per le uscite la dicitura **Euler(3)**; infine, per le variabili di stato sono stati considerati i valori relativi alle variabili ϕ e q . Si ottengono così le seguenti matrici:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -2.171 \end{bmatrix} \quad (3.2)$$

$$B = \begin{bmatrix} 0 \\ 0.42 \end{bmatrix} \quad (3.3)$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (3.4)$$

¹il segno indica il verso di rotazione e non è rilevante in questa analisi.

$$D = [0] \quad (3.5)$$

Si può notare che, essendo $D=0$, il sistema è strettamente proprio, ovvero l'uscita dipende esclusivamente dagli stati.

Adesso è possibile sostituire le matrici 3.2, 3.3, 3.4 e 3.5, all'equazione 3.1. Infine bisogna moltiplicare la FdT ottenuta per due costanti:

K_1 , costante che indica come i vari segnali di controllo per le quattro manovre sono ripartiti tra i quattro motori, contenuta nel blocco *FCS/FlightController/ControlMixer*; l'ultima colonna della matrice *Controller.Q2Ts²* è quella di riferimento per il rollio (anche qui vale il discorso fatto in precedenza sul segno).

$$K_1 = 5,6659$$

K_2 , coefficiente di spinta dei motori, presente nel blocco *FCS/FlightController/thrustsToMotorCommand*, nel vettore *Vehicle.Motor.thrustToMotorCommand*.

$$K_2 = 1530.7$$

Si ottiene così la seguente funzione di trasferimento:

$$P(s) = \frac{3644}{s^2 + 2.171s} \quad (3.6)$$

Per il calcolo di $P(s)$ si è usato l'ausilio di uno script di MATLAB:

```

1 sys=ss(A,B,C,D); %crea lo spazio di stato
2 FdT=tf(sys); %calcola la fdt di uno spazio di stato
3 k1=5.6659;
4 k2=1530.7;
5 Ps=FdT*k1*k2
6 save('Ps.mat','Ps') %salva Ps in un file .mat

```

Dopo aver trovato la $P(s)$, è possibile procedere con le tecniche di controllo della sintesi in frequenza e in s .

Per maggiori chiarimenti sulla teoria delle tecniche di controllo, si rimanda all'appendice del documento.

3.1 Controllore in frequenza

Come prima fase si vanno ad analizzare le specifiche univoche e lasche che il sistema deve rispettare.

In particolare le specifiche da soddisfare sono:

²La prima colonna è relativa al Thrust, la seconda allo Yaw, la terza al Pitch e l'ultima al Roll.

1. Sistema di TIPO 1;
2. Errore a regime permanente $|\tilde{e}_1| \leq 0.01$;
3. Margine di risonanza $M_r \leq 2dB$;
4. Banda passante $B_3 \simeq 2Hz$.

Le specifiche, però, sono in catena chiusa, quindi bisogna tradurle in catena aperta.

1. Il sistema è di tipo 1 se la $F(s)$, data dal prodotto tra il controllore $G(s)$ per il processo $P(s)$, ha un polo in $s = 0$.
2. $\tilde{e}_1 = \frac{k_d^2}{k_f} \leq 0.01$, con $k_d = 1$, guadagno in retroazione.
3. A un margine di risonanza $M_r \leq 2dB$, corrisponde un margine di fase $M_\phi \geq 47^\circ$, calcolato attraverso l'intersezione sulla carta di Nichols tra il luogo 2 e 0 ($-133^\circ + 180^\circ = 47^\circ$).
4. La pulsazione di attraversamento si calcola facendo $\omega_t = 3 \div 5[B_3]_{Hz} = 4 \times 2 = 8 \text{ rad/s}$.

Per le specifiche 3. e 4. bisogna considerare che il modulo di $F(s)$ in $\omega_t = 8 \text{ rad/s}$ deve essere uguale a zero, e la fase maggiore o uguale di -133 gradi.

Prima di progettare il controllore di primo tentativo, si analizza il comportamento del sistema in catena aperta, attraverso i diagrammi di Bode, di Nyquist, di Nichols e la risposta al gradino, grazie all'ausilio dei seguenti comandi in Matlab per tracciare i diagrammi sopraccitati:

```

1 load('Ps.mat') %carica il file .mat contenente P(s)
2 s=tf('s'); %dichiara s come funzione di trasferimento
3 ws=feedback(Ps,1); %catena chiusa
4 bode(Ps);
5 grid; %griglia
6 nyquist(Ps);
7 nichols(Ps);
8 ngrid; %griglia dei luoghi
9 step(ws)

```

Analizzando il diagramma di Nyquist in Figura 3.1, è possibile applicarne il criterio, che afferma che un sistema in catena chiusa è stabile se e solo se il diagramma polare di $F(j\omega)$, per $\omega \in (-\infty, +\infty)$, circonda il punto $(-\frac{1}{K_d}; J_0)$ in senso antiorario un numero di volte uguale al numero di poli a parte reale positiva di $F(s)$. Essendo tutti i poli a parte reale negativa o nulla, il sistema è sicuramente stabile, ma è ulteriormente confermato anche dal criterio di Nyquist.

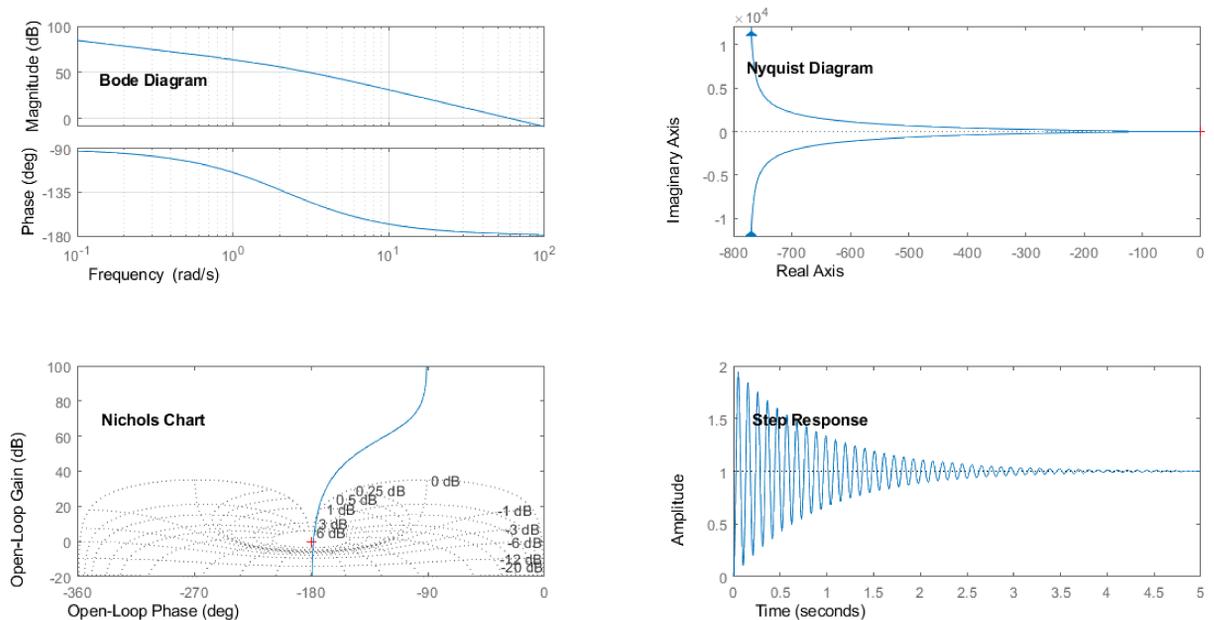


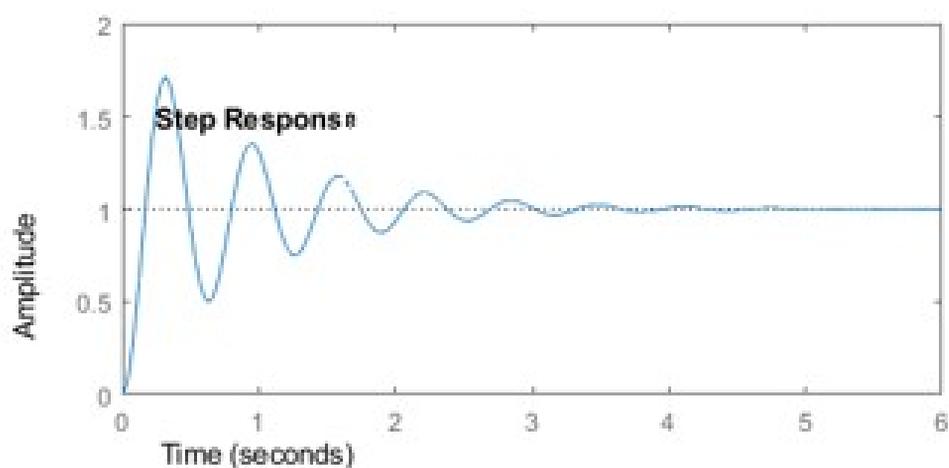
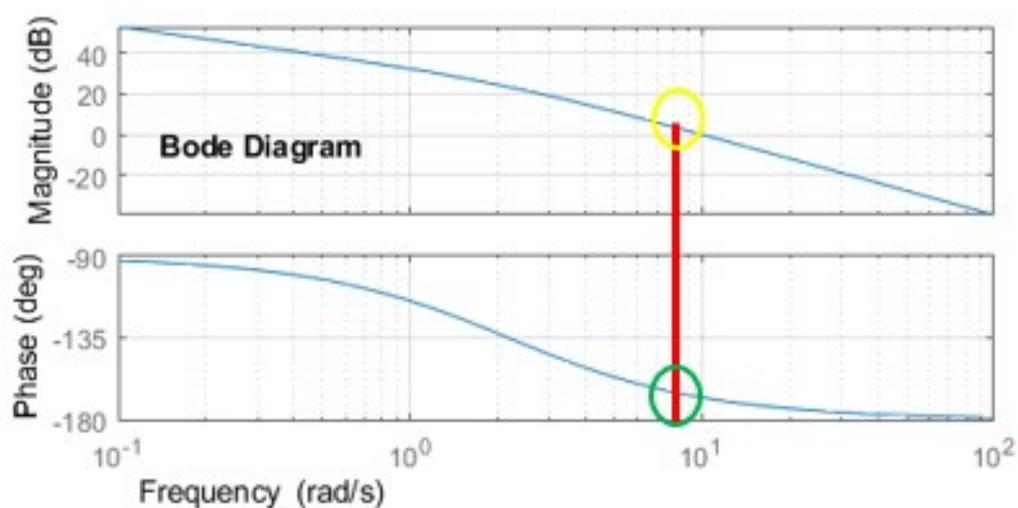
Figura 3.1. Diagrammi di Bode, Nichols, Nyquist e risposta a gradino di $P(s)$

Inoltre la prima specifica è soddisfatta da $P(s)$ che presenta un polo in $s=0$, quindi il sistema è di tipo 1, mentre le altre specifiche non sono soddisfatte, come è possibile notare dai diagrammi di Bode. Infatti in $\omega_t = 8 \text{ rad/s}$, il modulo è maggiore di zero e la fase molto minore di -133° . Inoltre la risposta al gradino, seppur segue l'ingresso a regime, presenta eccessive oscillazioni. Il diagramma di Nichols servirà successivamente per la verifica delle specifiche in catena chiusa.

Si procede, quindi, a progettare il controllore di primo tentativo, soddisfacendo, così le prime due specifiche. Come già evidenziato, non c'è bisogno di aggiungere poli nell'origine in quanto già presente un polo nel processo da controllare. È necessario, invece, cambiare il guadagno del sistema andando a moltiplicare $P(s)$ per un $K_g = 0,0274$, in modo da soddisfare la seconda specifica.

Si ottiene, in questo modo il controllore di primo tentativo $\hat{G}(s) = 0.0274$.

Analizzando la risposta a gradino in Figura 3.2 si può notare un miglioramento della risposta, con l'errore a regime permanente come richiesto, ma non è ancora soddisfacente.

Figura 3.2. Risposta al gradino di $\hat{F}(s)$ Figura 3.3. Diagramma di Bode di $\hat{F}(s)$

Inoltre, dal diagramma di Bode in Figura 3.3, si evince che il modulo in $\omega_t = 8$ è circa uguale a 4 dB, mentre la fase a -165 gradi.

Bisogna, quindi, aumentare la fase di almeno 32 gradi e diminuire il modulo di 4dB, attraverso l'utilizzo delle reti compensatrici³.

³Si consiglia per completezza di leggere la teoria nell'appendice del documento.

In primo luogo si aggiunge una rete anticipatrice, opportunamente scelta in modo che aumenti la fase e non aumenti eccessivamente il modulo. I valori sono stati scelti con l'aiuto dei diagrammi di Bode delle reti compensatrici.

$$R_a(s) = \frac{1 + \frac{s}{4}}{1 + \frac{s}{64}} \quad (3.7)$$

Moltiplicando $\hat{F}(s)$ per la rete anticipatrice $R_a(s)$ si ottiene la nuova funzione $F_a(s)$ con il diagramma di Bode in Figura 3.4. Si può notare che il modulo è aumentato

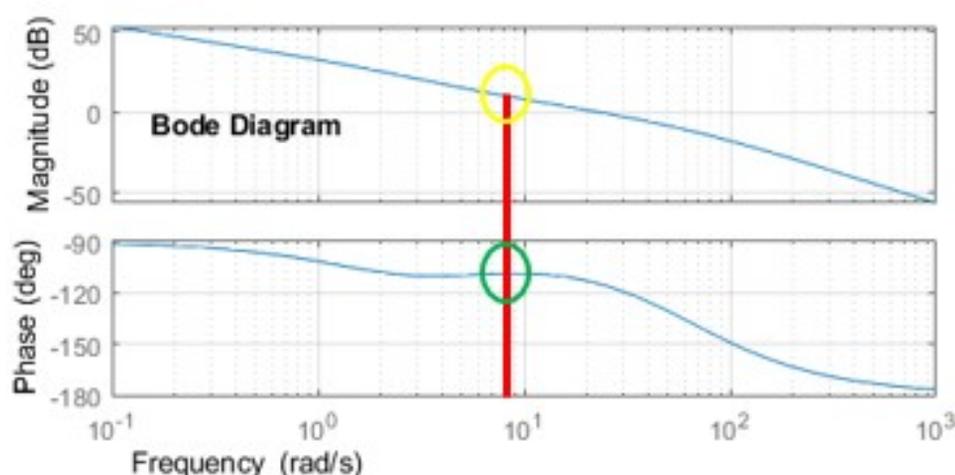


Figura 3.4. Diagramma di Bode di $F_a(s)$

fino ad arrivare a 10 dB e la fase risulta uguale a -108 gradi.

C'è bisogno, quindi, di diminuire il modulo di 10 dB, in modo da soddisfare le specifiche, ma non diminuire troppo la fase.

Si inserisce, a tal scopo, la seguente rete attenuatrice:

$$R_i(s) = \frac{1 + \frac{s}{0.24}}{1 + \frac{s}{0.08}} \quad (3.8)$$

Moltiplicando la $F_a(s)$ per la rete $R_i(s)$, si ottiene, infine, la $F(s)$; andando ad analizzare, nuovamente, i diagrammi in Figura 3.5, risulta evidente che le specifiche sono soddisfatte. Nello specifico, il modulo in $\omega_t = 8$ è circa uguale a zero e la fase a -110 gradi. Per una ulteriore verifica viene analizzato il diagramma di Nichols, dove si nota che il grafico di $F(s)$ non incontra mai il luogo a 2 dB, soddisfacendo così la specifica sul margine di risonanza. Inoltre, il grafico incontra il luogo a

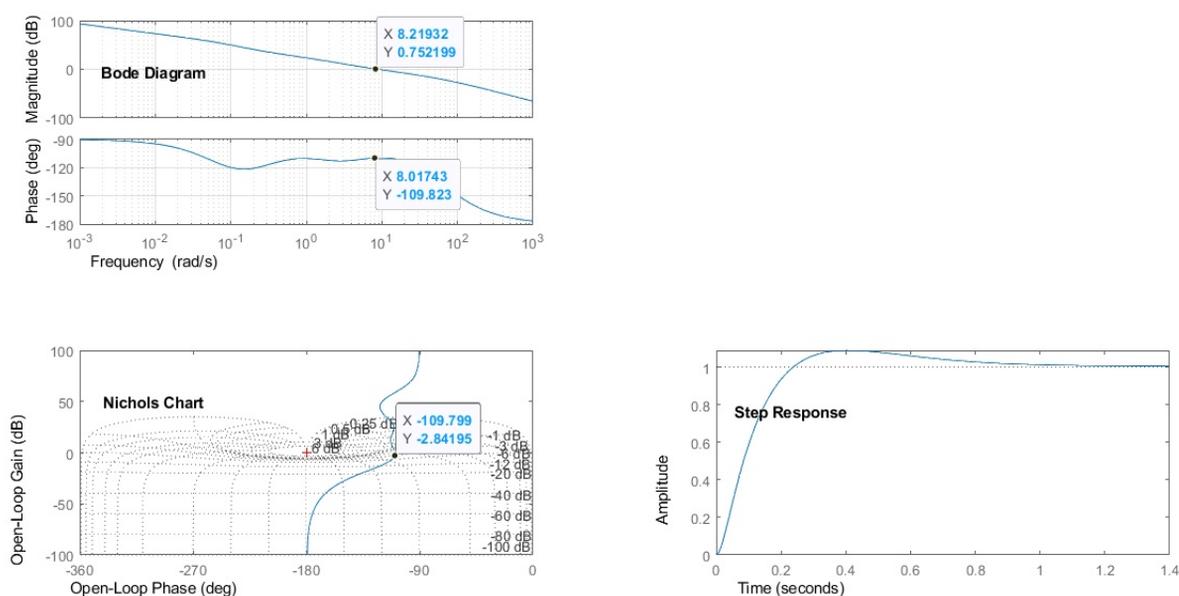


Figura 3.5. Diagrammi di Bode, di Nichols e risposta al gradino di $F(s)$

-3 dB in -110 gradi e -3 dB; andando a trovare la pulsazione corrispondente a quel modulo e quella fase nel diagramma di Bode e dividendola per 2π , la banda passante risulta essere circa uguale a 2Hz.

In conclusione il sistema in catena chiusa è stabile e non ha eccessive oscillazioni, con un tempo di assestamento e una sovranelongazione non troppo elevate, risultando nel complesso abbastanza pronto.

3.2 Controllore con luogo delle radici

Anche per la sintesi in s , è necessario partire con la valutazione delle specifiche da soddisfare:

1. Errore a regime permanente $|\tilde{e}_1| \leq 0.01$;
2. Poli a parte reale minore uguale di -3.

Come primo passo, si inizia tracciando il luogo delle radici del processo, attraverso il comando su MATLAB "`rlocus(Ps)`". Dal grafico in Figura 3.6 si evidenzia che il sistema è stabile in quanto tutti i poli sono a parte reale minore uguale di zero, però le specifiche non sono soddisfatte, poiché le radici non si trovano nella regione desiderata. Nel progettare il controllore bisogna tener conto della fisica realizzabilità, ovvero il numero di zeri deve essere minore al numero di poli; inoltre per non cambiare la forma e il numero degli asintoti, rendendo il tutto più complesso, si consiglia di inserire tanti poli quanti zeri.

È buona norma scegliere lo zero in modo che sia il più vicino possibile al limite della zona desiderata per non avere comportamenti irregolari nel transitorio;

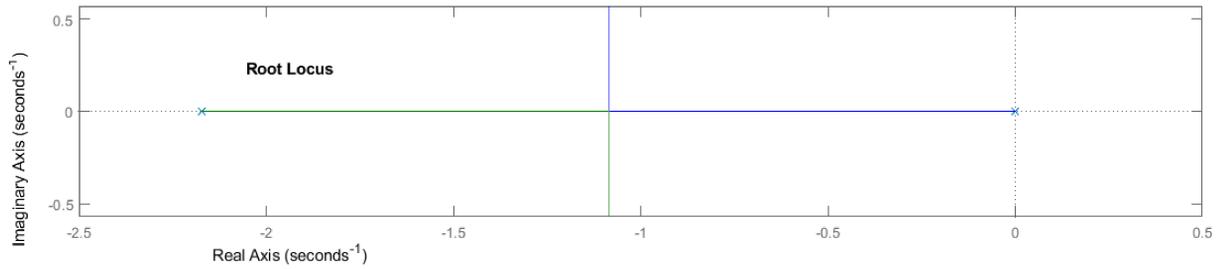


Figura 3.6. Luogo delle radici di P(s)

mentre il centro degli asintoti viene spostato con i poli il più lontano possibile dalla zona non desiderata.

Rese note queste considerazioni si è scelto di progettare il seguente controllore:

$$\hat{G}(s) = \frac{s + 4}{s + 100} \quad (3.9)$$

Moltiplicandolo per la P(s), si può tracciare il luogo delle radici, in Figura 3.7.

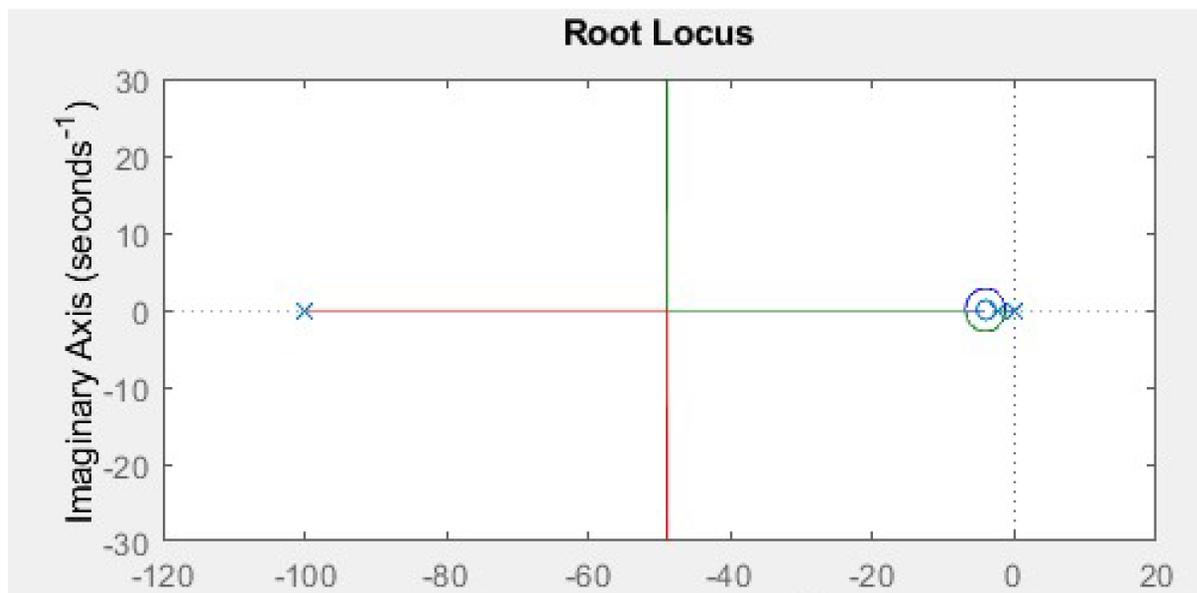


Figura 3.7. Luogo delle radici di $\hat{F}(s)$

Si può notare che le radici intersecano -3 in un punto K_1 , il cui valore si può ricavare con il criterio di Routh o più banalmente con il comando `rlocfind(Fhat,-3)` su MATLAB.

La seconda specifica viene, così, soddisfatta in quanto tutte le radici si trovano nella zona desiderata per un $K_1 \geq 0.0662$.

Per soddisfare la prima specifica, invece, si deve avere un guadagno $K_2 \geq 0.6860$.

Il guadagno finale, che viene moltiplicato per la $\hat{F}(s)$ è $K=0.6860$; si consiglia sempre di prendere il guadagno minimo per non rischiare che un valore troppo elevato porti altri tipi di problematiche.

Valutando la risposta a gradino in Figura 3.8, si nota una risposta pronta e un transitorio stabile.

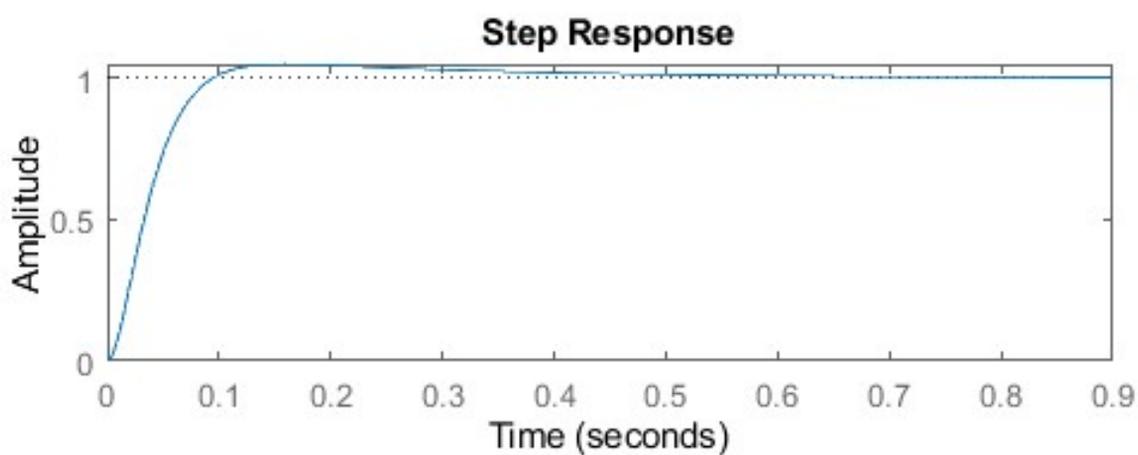


Figura 3.8. Risposta al gradino di $F(s)$

Capitolo 4

Simulazioni e confronti

Come ultima fase di questo lavoro, sono stati implementati i due controllori trovati nel capitolo precedente sul progetto "Simulink Support Package for Parrot Minidrones", andando a sostituire il controllore PID di default; i tre controllori, poi, sono stati confrontati attraverso i tracciati delle variabili e attraverso degli indici di prestazione.

Prima di sostituire i controllori nel modello, però, è opportuno discretizzarli, in quanto andrebbero a causare errori di sample times multipli (Figura 4.1), essendo questi controllori a tempo continuo mentre gli altri controllori PID delle altre manovre sono a tempo discreto con un tempo di campionamento $T_s=0.005$.



Figura 4.1. Errore di sample times multipli

Per trasformare il controllore da tempo continuo a tempo discreto si è utilizzato il seguente codice su MATLAB:

```
1 Ts = 0.005 %tempo di campionamento
2 Gz = c2d(Gs, Ts) %discretizzazione
3 %calcolo poli, zeri e guadagno da inserire nel
  blocchetto Simulink
4 poles = pole(Gz)
5 zeros = zero(Gz)
6 gain = dcgain(Gz)
```

Adesso è possibile andare a sostituire i controllori trovati nel blocco *flightControlSystem/Attitude*, facendo delle opportune modifiche, come in Figura 4.2.

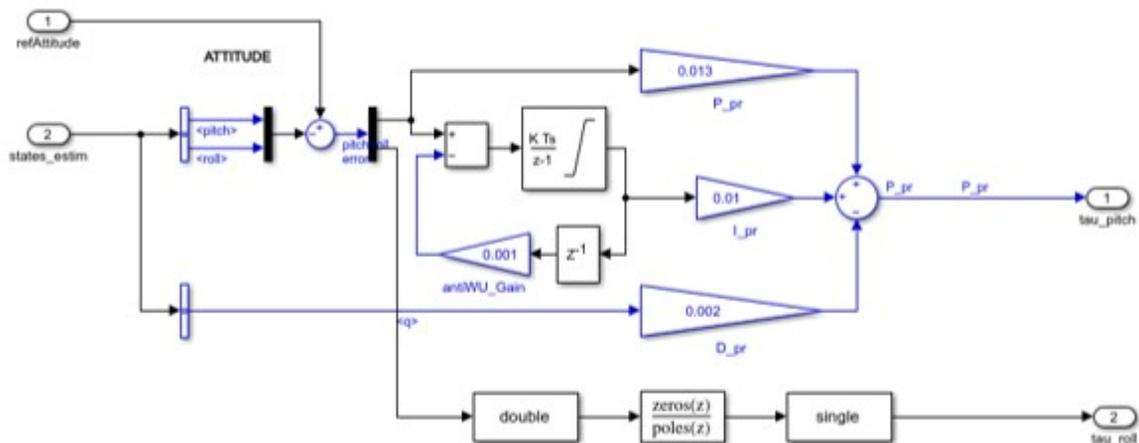


Figura 4.2. Modifiche del blocco Attitude

Innanzitutto con l'ausilio di un Demultiplexer sono stati disaccoppiati i segnali di Pitch e Roll, in seguito è stato modificato il controllore PID, in modo da controllare solo il Pitch, e aggiunti i controllori trovati, prima con la tecnica in frequenza e poi con il luogo delle radici, con il blocco zero-poli anticipato da un cast to double e seguito da un cast to single; il cast to double trasforma la precisione singola, con cui lavora il sistema, in una precisione doppia, con cui, invece, lavora il controllore, in seguito ritrasformata in precisione singola con il cast to single.

Il modello usato in questa fase di verifica, come già detto in precedenza, è quello non lineare, andando a impostare la $VSS_VEHICLE=1$ e il riferimento è quello costante dato dal Signal Editor.

Dopo aver compilato il progetto, pigiando il pulsante Run, inizia la simulazione di volo, osservabile grazie al Toolbox di visualizzazione in 3D. Testando entrambi i controllori si nota che il drone riesce a salire abbastanza rapidamente e a mantenere la posizione con precisione, rendendo così i controllori validi¹.

Dopo un primo controllo visivo del corretto comportamento del drone, si è passato a delle verifiche più precise, attraverso, prima, l'analisi dei tracciati, poi, con degli indici di prestazioni.

4.1 Analisi dei tracciati

Per analizzare i tracciati delle variabili in uscita dal sistema, sono stati aggiunti dei blocchetti di tipo Scope all'interno del blocco *Flight Control System/Logging*, in

¹Se il controllore non riesce a svolgere bene l'azione di controllo, Simulink dà errore di Crash Predictor Flag.

Figura 4.3.

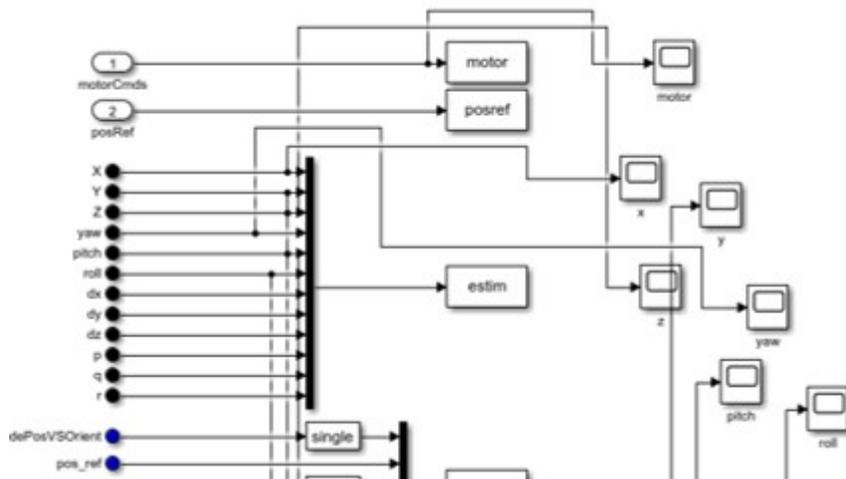


Figura 4.3. Aggiunta di blocchi Scope per l'analisi dei tracciati

Verranno messi a confronto i tre controllori:

- il controllore **PID** di default;
- il controllore **in frequenza**;
- il controllore con la tecnica del **luogo delle radici**.

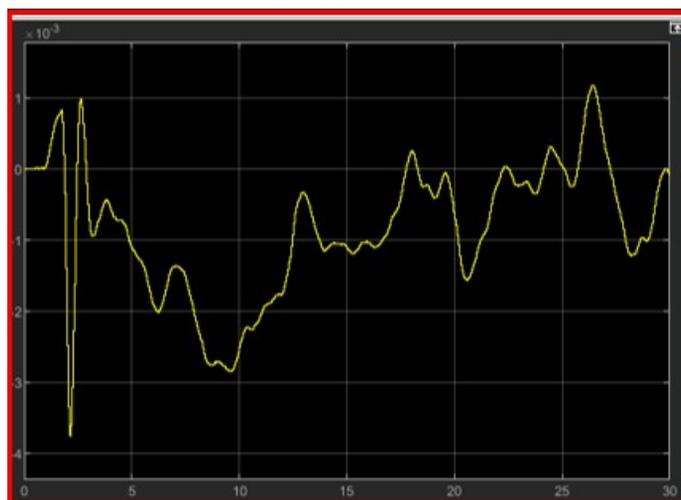
Dai tracciati X in Figura 4.4, si può notare che le oscillazioni sono nell'ordine dei 10^{-3} m per tutti e tre i controllori.

Dai tracciati Y in Figura 4.5, si può notare che le oscillazioni sono nell'ordine dei 10^{-3} m per i controllori PID originale (4.5(a)) e luogo delle radici (4.5(c)), mentre il controllore in frequenza (4.5(b)) ha oscillazioni dell'ordine di 10^{-2} m quindi quest'ultimo è meno pronto a rispondere alle sollecitazioni del sistema, mentre gli altri due rimangono buoni controllori.

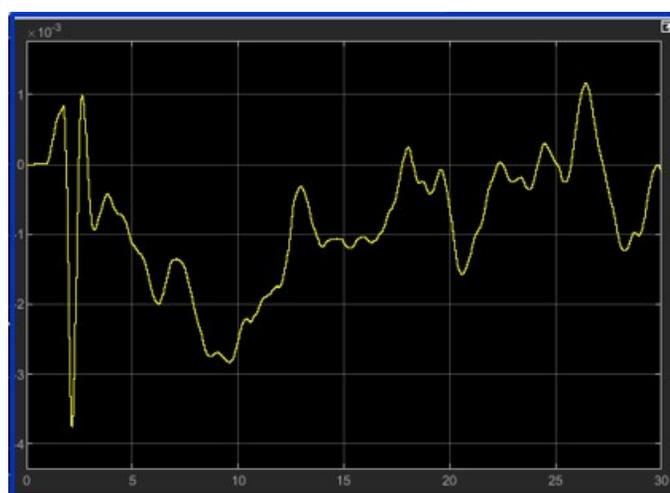
Tutti i controllori si assestano, dopo una sovraelongazione iniziale, alla posizione di 1,5 m dal suolo², come si evince dal tracciato Z in Figura 4.6.

Per i tracciati del Roll in Figura 4.7, si possono fare le stesse considerazioni fatte per i tracciati Y.

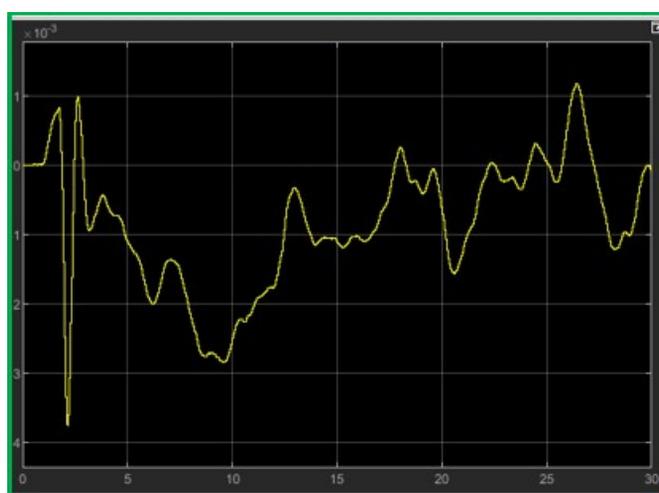
²Si ricorda che l'asse Z punta verso il basso nel sistema di riferimento NED.



(a) PID di default

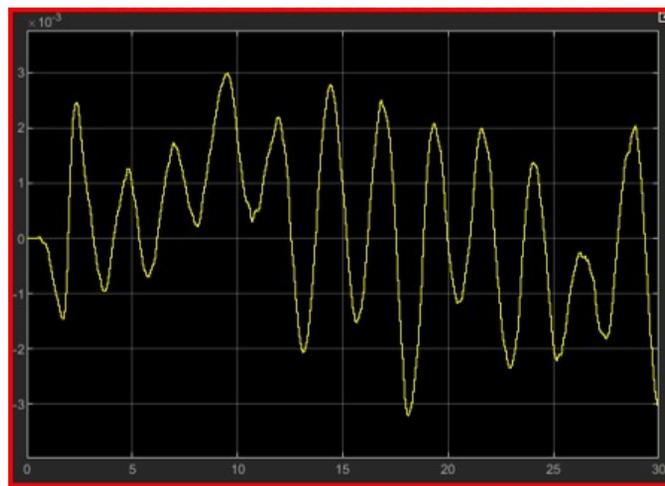


(b) Controllore in frequenza

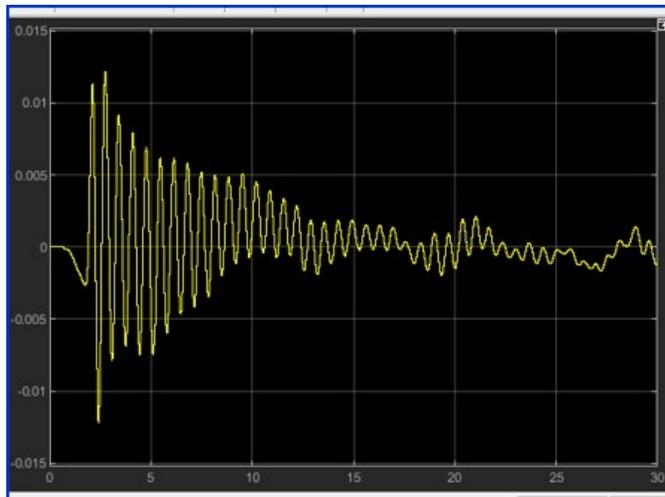


(c) Controllore con luogo delle radici

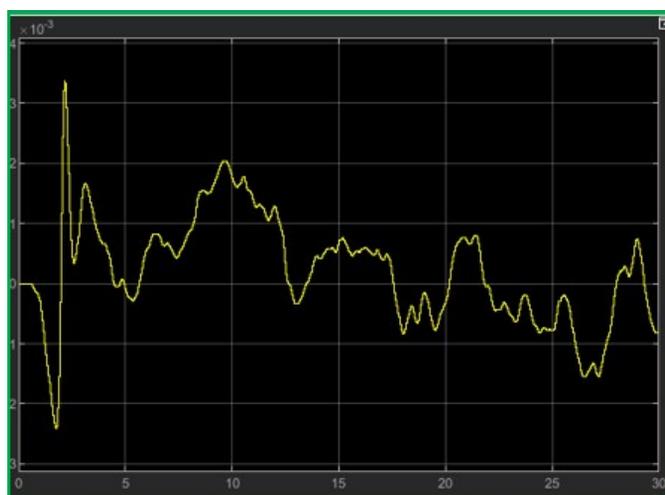
Figura 4.4. Tracciati X a confronto



(a) PID di default

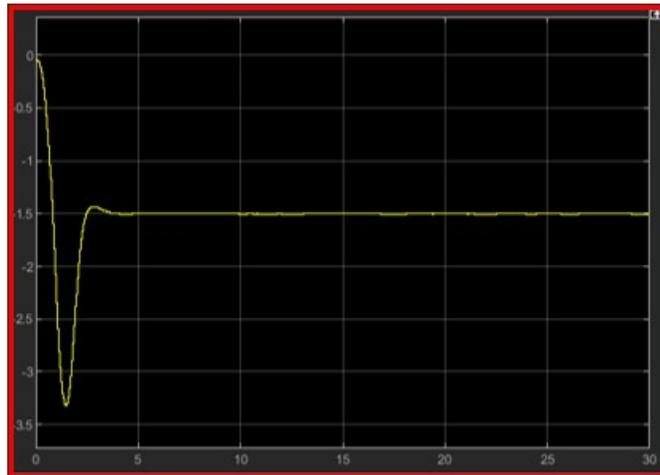


(b) Controllore in frequenza

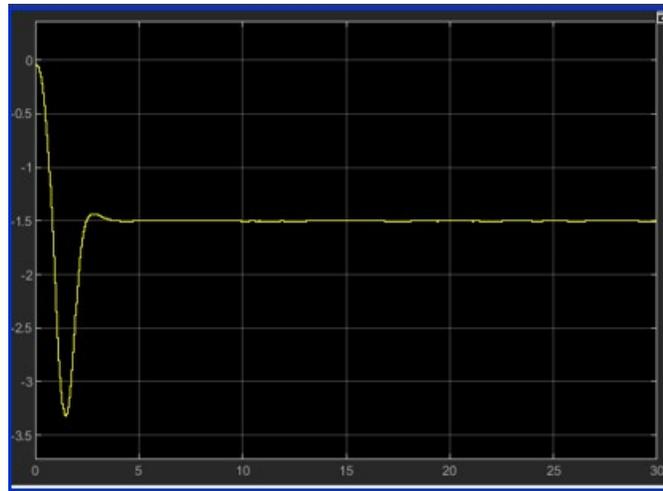


(c) Controllore con luogo delle radici

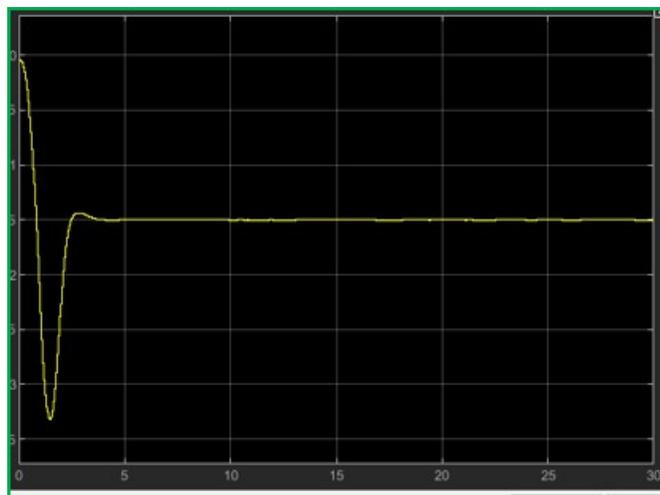
Figura 4.5. Tracciati Y a confronto



(a) PID di default

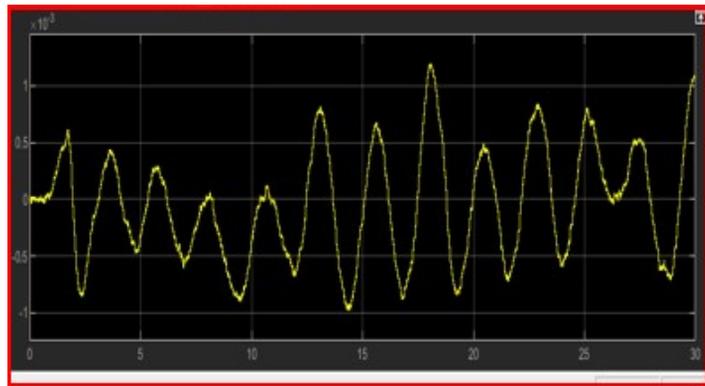


(b) Controllore in frequenza

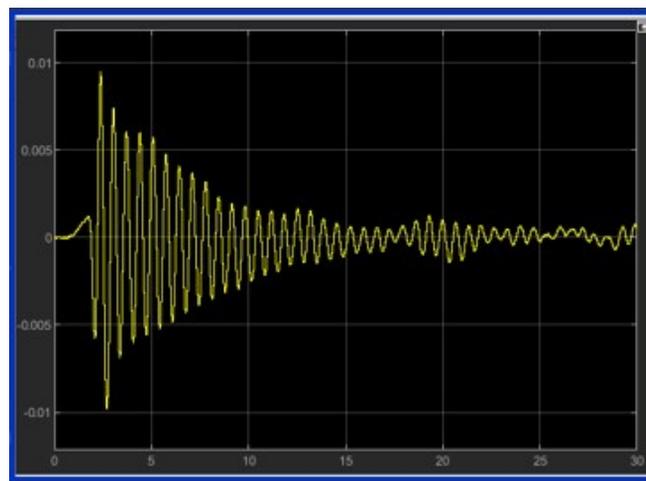


(c) Controllore con luogo delle radici

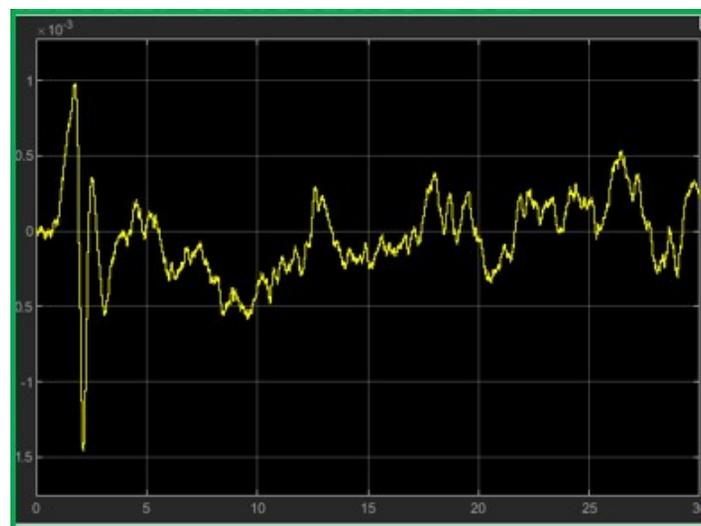
Figura 4.6. Tracciati Z a confronto



(a) PID di default



(b) Controllore in frequenza



(c) Controllore con luogo delle radici

Figura 4.7. Tracciati Roll a confronto

4.2 Indici di prestazione

È possibile analizzare la risposta e, quindi, la prestazione del sistema, grazie a indici di prestazione integrali che valutano la differenza tra l'uscita del sistema e il riferimento nel corso del transitorio. Suddetti indici, quindi lavorano nell'intero arco temporale sull'errore del sistema definito come:

$$\epsilon(t) = y_{SP}(t) - y(t) \quad (4.1)$$

ovvero la differenza tra l'uscita effettiva e il set point. In particolare, gli indici usati per mettere a confronto i tre controllori sono:

- Integral of the Absolute value of Error (IAE), misura la prestazione del sistema integrando il valore assoluto dell'errore, trascurando, però, i piccoli errori.

$$IAE = \int_0^t |\epsilon(\tau)| d\tau \quad (4.2)$$

- Integral of Square Error (ISE), integra il quadrato dell'errore e rileva in modo efficiente i grandi errori ma tollera i piccoli errori in un periodo di tempo. Ha una risposta più rapida rispetto all'IAE ma tollera maggiori oscillazioni.

$$ISE = \int_0^t \epsilon(\tau)^2 d\tau \quad (4.3)$$

- Integral of the Time-weighted Absolute Error (ITAE), integra il valore assoluto dell'errore pesandolo con il tempo. Anche piccoli errori hanno peso e gli errori dopo un lungo tempo pesano di più di quelli iniziali.

$$ITAE = \int_0^t \tau |\epsilon(\tau)| d\tau \quad (4.4)$$

[10] Per il calcolo di tali indici si è fatto uso della libreria *indexlib*, che implementa IAE, ISE, ITAE nei blocchi Simulink, rispettivamente in Figura 4.8, 4.9, 4.10.

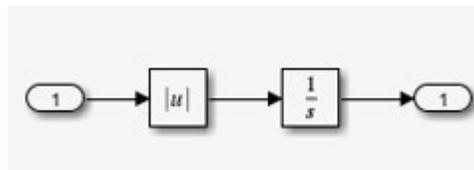


Figura 4.8. Blocco IAE in Simulink

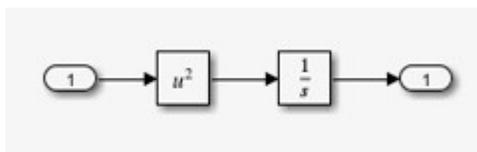


Figura 4.9. Blocco ISE in Simulink

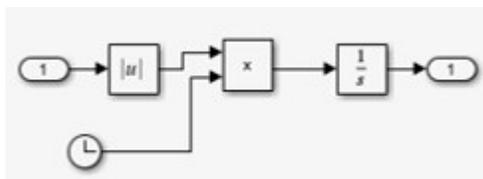


Figura 4.10. Blocco ITAE in Simulink

Al sistema originario, è stato aggiunto il blocco, in Figura 4.11, che calcola gli indici, dove è possibile visualizzarli attraverso dei blocchetti di tipo *Display*. Il blocco è formato dal sottosistema in Figura 4.12, che seleziona attraverso un *Bus Selector* i dati contenuti nei bus *Command* e *States*. In questo caso particolare si vuole analizzare l'errore per l'angolo di Roll, quindi si va a selezionare il riferimento del Roll sul *Command* e l'uscita effettiva corrispondente all'angolo ϕ . Andando a fare la differenza tra i due, si ottiene l'errore in ingresso ai blocchi degli indici sopra raffigurati.

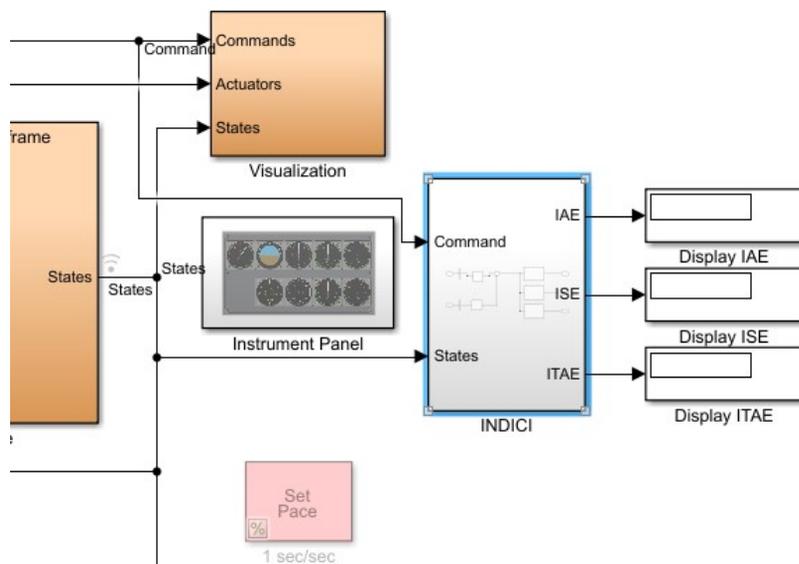


Figura 4.11. Blocco aggiunto per il calcolo degli indici

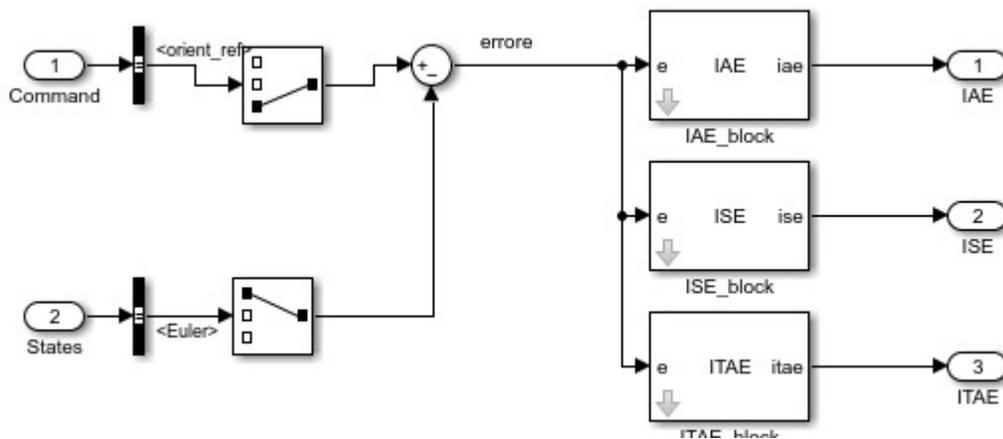


Figura 4.12. Blocco INDICI

Mettendo a confronto i vari controllori nella tabella e tenendo in considerazione il fatto che bisogna minimizzare il valore di questi indici, in modo da minimizzare l'errore, si può notare come il controllore in s sia nettamente migliore degli altri due, mentre il controllore in frequenza risulta leggermente peggiore di quello originale.

Tabella 4.1. Indici di prestazione

	PID di default	Controllore in frequenza	Luogo delle radici
IAE	0.01182	0.03648	0.004246
ISE	0.000006528	0.0001233	0.000001584
ITAE	0.1944	0.3071	0.05476

Conclusione

L'obiettivo dell'elaborato era quello di trovare delle alternative al controllore di default per la manovra del rollio. Oltre alle verifiche fatte nelle pagine precedenti sul modello non lineare e a riferimento costante, sono stati fatti altri test su riferimenti non costanti oppure annullando l'azione del controllore a cascata (modificando la porta logica del blocco Signal Editor per forzarlo a un riferimento costante), non riportati nel presente elaborato. Le varie verifiche hanno mostrato tutti risultati analoghi, indicando un miglioramento delle prestazioni per il controllore progettato con il luogo delle radici, mentre il controllore in frequenza mostra prestazioni più scarse, ma comunque ottimali. Nel complesso i controllori riescono a svolgere bene la loro azione di controllo e hanno un comportamento equiparabile a quello del PID di default.

In lavori futuri si può pensare di andare a progettare altri controllori per tutte le manovre rimanenti, in modo da sostituire quelli di default e migliorare la risposta complessiva dell'intero sistema.

Appendice A

Teoria del controllo

In questa appendice vengono approfondite alcune generalità teoriche, messe in atto per la sintesi dei controllori. In particolare vengono approfondite le tre tecniche di controllo nominate in queste pagine, ovvero la sintesi in frequenza, con luogo delle radici e il controllo PID.

A.1 Sintesi in frequenza

La sintesi in frequenza consiste nello scegliere la funzione di trasferimento del controllore $G(s)$ in modo che il sistema in catena chiusa abbia le proprietà assegnate in fase di progettazione. È un tipo di sintesi per tentativi dove prima si cerca di soddisfare le specifiche univoche, (tipo del sistema, errore a regime permanente, astatismo rispetto a disturbi. . .) le quali possono essere soddisfatte in un solo modo; in seguito vengono soddisfatte le specifiche lasche (forma di $F(s)$, margine di risonanza, banda passante, posizionamento dei poli. . .), le quali permettono maggior libertà di scelta del controllore. In particolare, si avvale di due strumenti:

- **Il diagramma di Bode** che rappresenta l'ampiezza e la fase della risposta in frequenza. La scala utilizzata è quella semilogaritmica divisa in decadi, in quanto permette la compressione della scala e favorisce il tracciamento qualitativo. Si procede con il passaggio della funzione di trasferimento in forma normale o forma di Bode, mettendo così in evidenza il guadagno statico $K_F = \lim_{s \rightarrow 0} s^m F(s)$, dove m è il numero di poli in $s=0$ della $F(s)$. Viene calcolata l'ampiezza $|F(j\omega)| = 20 \log_{10} |F(j\omega)|_{[dB]}$. Per facilitarne il calcolo è possibile individuare in qualsiasi funzione quattro possibili termini elementari, ovvero il termine costante (K_F), il termine monomio ($j\omega^m$), il termine binomio ($1 + j\omega\tau$) e il termine trinomio ($1 - \frac{\omega^2}{\omega_n^2} + 2j\omega \frac{\xi}{\omega_n}$) e trovare i loro diagrammi qualitativi, per poi correggerli con dei diagrammi di correzione per il termine trinomio e binomio. Esistono delle regole per il tracciamento qualitativo che affermano:

1. La pendenza iniziale del modulo dipende solo dai poli nell'origine ed è $20n$ dove n è la loro molteplicità.

2. La pendenza finale è uguale alla differenza tra il numero di poli e il numero di zeri, dove ogni polo dà pendenza -20 e ogni zero $+20$.
 3. Il valore iniziale della fase dipende solo dal termine costante e il termine monomio.
 4. Per il valore finale della fase si considera che ogni termine dà contributo $+90^\circ$ o -90° in base se positivo o negativo e se è un polo o uno zero.
- **Il diagramma di Nichols** che rappresenta in un'unica curva le coppie modulo-fase della risposta in frequenza. Viene comunemente usato insieme alla carta di Nichols ovvero il luogo dei punti a modulo e fase costante, utile per il calcolo del margine di fase e guadagno.

Analizzando le specifiche univoche, si procede con l'elaborazione di un controllore di primo tentativo del tipo $\hat{G}(s) = \frac{K_G}{s^r}$; in seguito si tracciano i diagrammi di Bode e si controlla se le altre specifiche sono soddisfatte. Se non risultano soddisfatte le specifiche lasche si modifica il controllore aggiungendo l'azione di delle *Funzioni compensatrici elementari* chiamate anche *reti compensatrici*, di guadagno unitario e senza poli nell'origine. Le reti compensatrici sono di due tipi:

- **Funzione Anticipatrice** che aumenta fase e modulo ed è del tipo

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

dove ω_a e m_a sono opportunamente scelti con l'aiuto dei grafici in Figura A.1.

- **Funzione Attenuatrice** che diminuisce fase e modulo ed è del tipo

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

Il grafico usato è il simmetrico rispetto l'asse delle ascisse di quello della rete anticipatrice.

Infine si controlla che tutte le specifiche siano soddisfatte, in particolare attraverso l'uso della carta di Nichols, vengono analizzati i margini di stabilità.

Il margine di guadagno rappresenta la vicinanza del sistema all'instabilità, ovvero di quanto è possibile aumentare il guadagno senza che il sistema diventi instabile. Si calcola trovando la differenza tra 0 dB e la curva nel punto in cui la fase vale -180° .

Il margine di fase che rappresenta la fase in cui il guadagno vale 0 dB (a cui corrisponde la pulsazione di attraversamento) sommata a 180° .

Questi, però, sono margini in catena aperta, mentre in catena chiusa si ha la banda passante e il margine di risonanza.

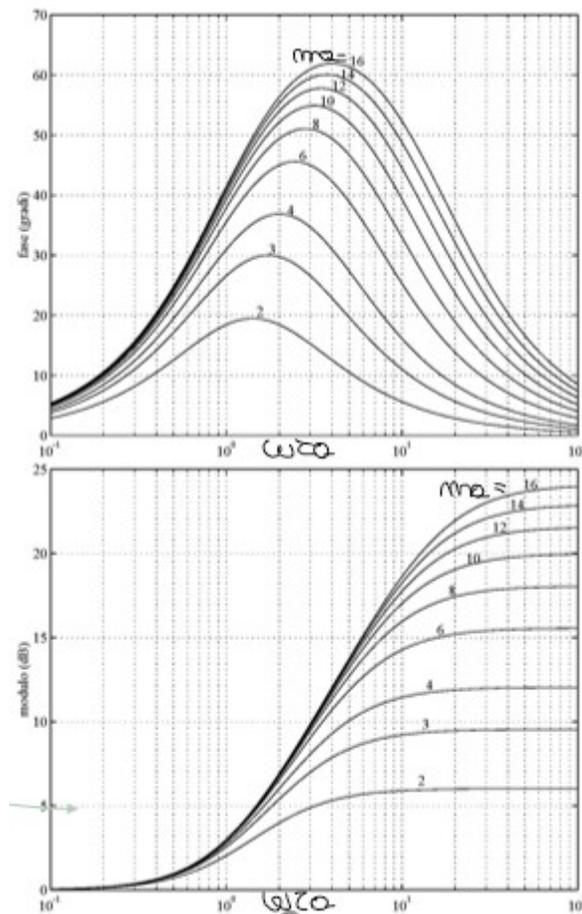


Figura A.1. Diagrammi della rete anticipatrice

La banda passante è la pulsazione dove il modulo si attenua di -3 dB e si trova facendo l'intersezione tra la funzione e il luogo a -3 dB.

Il margine di risonanza che rappresenta il picco dove il modulo della funzione in catena chiusa è massimo e si calcola trovando il luogo dei punti tangente alla curva.

A.2 Sintesi in s

Similmente alla sintesi in frequenza, anche la sintesi in s è un tipo di sintesi per tentativi, dove la funzione compensatrice è progettata tramite la tecnica del luogo delle radici. Mentre la sintesi in frequenza può essere usata solo per sistemi a fase minima (cioè stabili), la sintesi in s può essere usata anche quando i poli sono a parte reale positiva.

Data una funzione di trasferimento

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

il luogo delle radici è il luogo dei punti percorsi dalle radici dell'equazione

$$f(s, k) = \prod_{i=1}^n (s - p_i) + K \prod_{i=1}^m (s - z_i) = 0 \quad (\text{A.2})$$

al variare di $K \in \mathbb{R}$.

Se $K > 0$ si parla di luogo positivo, se $K < 0$ luogo negativo, se $K = 0$ i poli in catena chiusa coincidono con i poli in catena aperta.

Per il tracciamento del luogo delle radici esistono due condizioni:

- **Condizione di modulo** per determinare il valore di K:

$$|K| = \frac{\prod_{i=1}^n |s - p_i|}{\prod_{i=1}^m |s - z_i|} \quad (\text{A.3})$$

- **Condizione di fase** che determina la forma del luogo:

$$\sum_{i=1}^n \angle(s - p_i) - \sum_{i=1}^m \angle(s - z_i) = \begin{cases} (2h + 1)\pi & \text{se } K > 0 \\ 2h\pi & \text{se } K < 0 \end{cases} \quad (\text{A.4})$$

Esistono, inoltre, delle regole utili per il tracciamento qualitativo del luogo delle radici:

1. Per $K=0$ i poli in catena chiusa coincidono con quelli in catena aperta.
2. Il luogo delle radici è simmetrico rispetto l'asse reale poiché le radici o sono reali o complesse coniugate.
3. Tutto l'asse reale appartiene al luogo; in particolare appartengono al luogo positivo tutte le porzioni di asse reale che lasciano alla propria destra un numero dispari di poli o zeri contati con le loro molteplicità.
4. Nei punti singolari entrano 2μ rami alternativamente convergenti e le tangenti al luogo nei punti singolari dividono l'angolo giro in 2μ parti uguali. I punti singolari sono le radici con molteplicità μ che soddisfano le seguenti condizioni:

$$\begin{cases} f(s, k) = 0 \\ \frac{\partial f(s, k)}{\partial s} = 0 \end{cases} \quad (\text{A.5})$$

5. Per $K \rightarrow \infty$ m rami convergono sugli zeri; gli altri $n-m$ rami divergono all'infinito lungo $n-m$ asintoti (Φ_+ e Φ_-) che partono da un punto sull'asse reale, chiamato centro degli asintoti (s_0), descritti dalle seguenti formule:

$$s_0 = \frac{\sum_{i=1}^n p_i - \sum_{i=1}^m z_i}{n - m} \quad (\text{A.6})$$

$$\begin{cases} \Phi_+ = \frac{(2h+1)\pi}{n-m} & \text{luogo positivo} \\ \Phi_- = \frac{2h\pi}{n-m} & \text{luogo negativo} \end{cases} \quad h = 0, 1, \dots, n - m - 1 \quad (\text{A.7})$$

Per la sintesi bisogna tenere in considerazione solo il luogo positivo e controllare se soddisfa le specifiche. Se non le soddisfa bisogna spostare il centro degli asintoti nella regione desiderata, mettendo poli e zeri in modo da non cambiare la forma del luogo. È possibile cancellare, con il controllore, poli e zeri per rendere la funzione di trasferimento meno complessa, a patto che siano nella regione desiderata; infatti, cancellando poli o zeri all'esterno della zona di specifica, lo spazio di stato diventa non completamente osservabile o raggiungibile e, inoltre, non vengono cancellati perfettamente fisicamente, quindi si viene a creare un ramo instabile.[1]

A.3 Regolatori PID

Sono dei regolatori industriali analogici dove l'uscita fornita varia gradualmente. Sono strutture di controllo in retroazione a ingresso costante, dove si ha una struttura fissa e bisogna scegliere solo alcuni parametri (i guadagni dei vari controllori). La loro azione è divisa tra tre controllori diversi:

- **Controllore proporzionale (P)** che fornisce un'uscita proporzionale all'errore di ingresso.

$$m_p(t) = K_G e(t)$$

Questo controllore garantisce un errore a regime costante ma non nullo; inoltre aumentando K_G diminuisce l'errore ma possono esserci problemi al transitorio.

- **Controllore integrale (I)** che fornisce un'uscita proporzionale all'area sottesa dalla curva errore-tempo.

$$m_i(t) = K_I \int_0^t e(\tau) d\tau$$

Questo controllore introduce un polo nell'origine e, quindi, un ritardo di fase di 90° e ha problemi di saturazione. In generale migliora le prestazioni a regime ma peggiora la stabilità.

- **Controllore derivativo (D)** dove l'uscita dipende dalla velocità alla quale varia l'errore.

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

Introduce uno zero nell'origine, per cui non è fisicamente realizzabile da solo, ma viene usato insieme ad altri controllori. Migliora la stabilità ma peggiora le prestazioni a regime.

In generale i tre controllori vengono combinati in parallelo formando i cosiddetti controllori PID:

$$m(t) = m_P(t) + m_I(t) + m_D(t)$$

e scegliendo opportunamente i valori dei K_G , K_I e K_D si possono migliorare le prestazioni del sistema.[1]

Bibliografia

- [1] Alberto Isidori. *Sistemi di controllo*. Siderea, 1996.
- [2] Haomiao Huang, Gabriel M Hoffmann, Steven L Waslander, and Claire J Tomlin. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In *2009 IEEE international conference on robotics and automation*, pages 3277–3282. IEEE, 2009.
- [3] Cesare Zumpano. Storia dei droni. URL=<https://www.techuniverse.it/drone/>.
- [4] G. P. Carratelli e M. Del Duca. *Controllo di un Quadcopter*. Univesità Federico II, Napoli, 2012.
- [5] Matteo Pantalone. *MODELLAZIONE E SIMULAZIONE DI UN QUADRICOTTERO MULTIROTORE*. UNIVERSITÀ DI BOLOGNA, 2015.
- [6] Brian Douglas. *Drone Simulation and Control*. MATLAB Tech Talks, URL=<https://it.mathworks.com/videos/series/drone-simulation-and-control.html>.
- [7] Andrea Cogliervina. *Controllo di un quadricottero per trasporto di carichi ignoti*. UNIVERSITÀ DEGLI STUDI DI PADOVA, 2015.
- [8] Riccardo Galli. Parrot mambo minidrone, recensione. URL=<https://www.riccardogalli.net/2018/01/24/parrot-mambo-minidrone-recensione/>.
- [9] MathWorks. *Quadcopter Project*. URL=<https://it.mathworks.com/help/aeroblks/quadcopter-project.html>.
- [10] Davide Manca. Sintesi di controllori in retroazione, politecnico di milano. URL=<https://pselab.chem.polimi.it/wp-content/uploads/2017/06/Lez-06-Sintesi-di-controllori-in-retroazione.pdf>.
- [11] Paul Pounds, Robert Mahony, and Peter Corke. Modelling and control of a large quadrotor robot. *Control Engineering Practice*, 18(7):691–699, 2010.