Politecnico di Torino



Corso di Laurea Magistrale in Ingegneria Aerospaziale

A.a. 2025/2026

Sessione di laurea di Luglio

Modellazione, Guida e Controllo di un Drone Tailsitter

Relatori:

Angelo Lerro

Giorgio Antonio Orlando

Candidato:

Simone Edoardo Neri

Sommario

Il presente lavoro di tesi si propone di sviluppare un modello di simulazione, guida e controllo per un velivolo a decollo verticale di tipo tailsitter, con architettura di controllo globale priva di commutazione discreta tra le modalità di volo. Il velivolo, progettato per una massa di 7 kg e una velocità di crociera di 220 km/h, integra quattro motori elettrici in configurazione X con un'ala fissa a profilo NACA 0012 e superfici di controllo di tipo elevone.

Il modello dinamico a sei gradi di libertà, non lineare, è stato implementato in ambiente MATLAB/Simulink adottando la rappresentazione in quaternioni per l'assetto, al fine di evitare singolarità cinematiche e *gimbal lock*. L'architettura di controllo è organizzata in una cascata di controllori: un anello esterno di posizione (PID), un generatore di assetto desiderato con transizione continua tramite interpolazione sigmoide, un controllore di assetto proporzionale basato sull'errore quaternionico, e un controllore di velocità angolare PI con ramo di *feed-forward* e protezione anti-windup. L'allocazione dei comandi ai motori e agli elevoni avviene mediante un meccanismo adattivo basato sulla massima autorità erogabile dagli elevoni.

Il sistema di guida genera missioni complete a partire da una libreria di primitive di volo a continuità di classe $\mathcal{C}^3$, raccordate da una macchina a stati che orchestra l'intera sequenza operativa. Lo scenario di riferimento adottato per la validazione è una missione di recupero e rilascio di un dispositivo medico, articolata in tre *leg*: il velivolo decolla dalla base, recupera il dispositivo presso un primo sito operativo, lo rilascia presso un secondo sito e rientra infine alla base di partenza.

I risultati delle simulazioni dimostrano la capacità del sistema di effettuare transizioni fluide tra hovering e crociera mantenendo la stabilità e il tracciamento delle traiettorie desiderate. Lungo l'intera missione l'inseguimento della traiettoria risulta accurato, con errore di posizione prossimo allo zero nelle fasi di crociera e di stazionamento e picchi transitori contenuti, dell'ordine di alcuni metri, in corrispondenza delle manovre dinamiche. L'impegno degli attuatori si mantiene entro l'inviluppo realizzabile, confermando la validità dell'architettura di controllo unificata proposta per la classe di missioni considerata.

Indice

Elenco delle figure xi

Elenco delle tabelle xv

1 Introduzione 1
- 1.1 Velivoli a Decollo e Atterraggio Verticale: Stato dell'Arte 1
- 1.2 Obiettivi della Ricerca 3
- 1.3 Metodologia 4
- 1.4 Fasi di Volo 6
- 1.5 Struttura della Tesi 6
- 1.6 Notazioni e Convenzioni 7

2 Specifiche di Progetto e Design del Velivolo 8
- 2.1 Requisiti Prestazionali 8
- 2.2 Dimensionamento Geometrico 9
  - 2.2.1 Fusoliera 9
  - 2.2.2 Ala 9
  - 2.2.3 Elevoni 11
  - 2.2.4 Bracci Motore 11
  - 2.2.5 Pod Propulsivi 13
- 2.3 Configurazione Propulsiva 13
  - 2.3.1 Motore Elettrico: T-Motor MN5008 KV340 13
  - 2.3.2 Disposizione Motori (X-Configuration) 14
  - 2.3.3 Eliche 16
- 2.4 Modellazione Geometrica in OpenVSP 16
  - 2.4.1 Processo di Modellazione 16
  - 2.4.2 Allocazione Masse Interne 16
- 2.5 Proprietà di Massa e Inerzia 19
- 2.6 Identificazione del Modello Aerodinamico 19
  - 2.6.1 Coefficienti Longitudinali dell'Ala: Reti Neurali su Dati Sperimentali NACA 0012 20

2.6.2 Coefficienti Costanti Estratti da VSPAero 20
2.7 Sommario delle Specifiche 21

3 Inizializzazione del Modello: Script MATLAB di Configurazione 23
3.1 Introduzione 23
3.2 Architettura Generale degli Script 23
3.3 Parametri Ambientali e di Solver 24
3.4 Tolleranze Numeriche 25
3.5 Parametri del *Plant* 25
3.5.1 Massa, Inerzia e Propulsione 25
3.5.2 Caricamento delle Reti Neurali Aerodinamiche 26
3.5.3 Derivate di Stabilità e Modello di Contatto 26
3.6 Parametri di Missione 26
3.6.1 Blending Hover/Cruise 27
3.6.2 Buffer di Approach 27
3.7 Guadagni del Sistema di Controllo 27
3.7.1 Position Controller 27
3.7.2 Attitude Controller (cascata esterna) 27
3.7.3 Rate Loop (cascata interna) 27
3.8 Specifica della Missione Operativa 28
3.8.1 Topologia della Missione 28
3.8.2 Costruzione della `mission_spec` 28
3.8.3 Condizioni Iniziali 29
3.9 Sintesi e Vantaggi dell'architettura 29

4 Modello Dinamico a Sei Gradi di Libertà 30
4.1 Introduzione 30
4.2 Convenzioni di Riferimento e Vettore di Stato 30
4.2.1 Sistemi di Coordinate 30
4.2.2 Rappresentazione dell'Assetto mediante Quaternioni 31
4.2.3 Vettore di Stato 31
4.3 Architettura Generale del Modello Simulink 32
4.4 Il Blocco *Plant* 33
4.5 Sottosistema Forces & Moments 35
4.5.1 Gravità (`fcn_gravity_body`) 37
4.5.2 Propulsione (`fcn_mixer`) 37
4.5.3 Aerodinamica (`fcn_aerodynamics`) 38
4.5.4 Contatto con il Terreno (`fcn_ground`) 42
4.6 Sottosistema Equations of Motion 44
4.6.1 Dinamica Traslazionale (`fcn_trans_dynamics`) 44
4.6.2 Dinamica Rotazionale (`fcn_rot_dynamics`) 45

4.6.3 Cinematica dell'Assetto (`fcn_quat_kinematics`) 46
4.6.4 Normalizzazione dei Quaternioni (`fcn_quat_normalize`) . . 47
4.6.5 Cinematica della Posizione (`fcn_pos_kinematics`) 47

5 Allocazione dei Controlli 49
5.1 Introduzione 49
5.1.1 Formulazione del Problema di Allocazione 49
5.2 Architettura del Sottosistema Simulink 51
5.3 Blending Adattivo: `fcn_blending` 53
5.3.1 Pressione Dinamica e Coefficienti di Efficacia 53
5.3.2 Cutoff a Bassa Velocità 55
5.3.3 Tetto Fisico degli Elevoni e Allocazione 56
5.3.4 Conservazione del Momento Totale 56
5.3.5 Guadagni di Conversione Momento–Deflessione 58
5.4 Allocazione ai Motori 59
5.4.1 Inversione della Matrice di Allocazione 59
5.4.2 Saturazione delle Velocità Angolari 60
5.5 Allocazione agli Elevoni 60
5.5.1 Calcolo delle Deflessioni Virtuali 60
5.5.2 Combinazione in Deflessioni Fisiche 62
5.5.3 Saturazione ai Limiti Meccanici 62
5.6 Riepilogo del Flusso dei Segnali 62

6 Controllo di Assetto 63
6.1 Introduzione 63
6.2 Architettura del Sottosistema Simulink 64
6.3 Errore di Assetto in Quaternioni 64
6.3.1 Selezione del Percorso Più Breve 65
6.4 Anello Esterno: Controllore di Assetto 65
6.4.1 Equivalenza con un Controllore PD sull'Assetto 66
6.5 Anello Interno: Controllore di Velocità Angolare 67
6.5.1 Errore di Velocità Angolare 68
6.5.2 Legge di Controllo PI con Feed-Forward 68
6.5.3 Feed-Forward Inerziale 69
6.5.4 Anti-Windup Condizionale 71
6.5.5 Saturazione 72
6.6 Valori dei Guadagni 72
6.7 Separazione delle Bande Passanti 73
6.7.1 Banda Passante del Rate Loop 73
6.7.2 Banda Passante dell'Attitude Loop 74
6.7.3 Calcolo Numerico e Rapporto di Separazione 75

7 Generazione del Comando di Assetto 77
7.1 Introduzione . . . 77
7.2 Architettura del Sottosistema Simulink . . . 78
7.3 Convenzioni e Mappatura degli Assi . . . 80
7.4 Calcolo della Velocità di Riferimento . . . 80
7.5 Costruzione dell'Assetto di Hover . . . 80
7.5.1 Determinazione dell'Asse di Spinta . . . 81
7.5.2 Determinazione dell'Asse Laterale via Yaw Desiderato . . . 81
7.6 Costruzione dell'Assetto di Crociera: Cascata di Rotazioni e Trim . 81
7.6.1 Rotazione $\mathbf{R}_a$: Allineamento alla Direzione di Volo . . . 82
7.6.2 Rotazione $\mathbf{R}_b$: Allineamento del Vettore di Spinta . . . 82
7.6.3 Rotazione $\mathbf{R}_c$: Angolo di Attacco di Trim . . . 83
7.6.4 Composizione della Cascata e Estrazione del Quaternione . . 83
7.7 Solver di Trim: Newton–Raphson Smorzato . . . 84
7.7.1 Residui e Matrice Jacobiana . . . 84
7.7.2 Schema Iterativo . . . 84
7.7.3 Filtro Passa-Basso sull'Uscita . . . 85
7.8 Blending Hover/Cruise e Output Finale . . . 85
7.8.1 Sigmoide di Pesatura . . . 85
7.8.2 Interpolazione SLERP . . . 86
7.8.3 Output di Spinta . . . 86
7.9 Conversione Matrice di Rotazione → Quaternione (Algoritmo di Shepperd) . . . 86
7.10 Sintesi del Flusso e Riepilogo delle Variabili Persistenti . . . 87

8 Controllo di Posizione 89
8.1 Introduzione . . . 89
8.2 Architettura del Sottosistema Simulink . . . 89
8.3 Formulazione Matematica . . . 91
8.3.1 Errori e Conversione di Frame . . . 91
8.3.2 Anello PID nel Frame Inerziale . . . 91
8.3.3 Termine di Feed-Forward . . . 92
8.3.4 Saturazione di Accelerazione . . . 92
8.3.5 Compensazione di Gravità e Generazione della Forza . . . 92
8.4 Proprietà di Stabilità e Convergenza . . . 93
8.5 Sintesi . . . 93

9 Guida e Navigazione 94
9.1 Introduzione . . . 94
9.2 Architettura del Sottosistema Simulink . . . 95
9.2.1 Configurazione della Missione e Comandi Elementari . . . 95

9.3 Specifica della Missione e Variabili di Stato . . . 96
9.3.1 Struttura della `mission_spec` . . . 96
9.3.2 Variabili di Stato della Macchina . . . 96
9.4 Le Primitive di Volo . . . 97
9.5 Profili Septici: Smoothness Garantita . . . 99
9.5.1 La Funzione *Septic Norm* . . . 99
9.5.2 Caratterizzazione del Massimo della Derivata . . . 99
9.6 Descrizione Dettagliata delle Primitive . . . 100
9.6.1 Decollo (`takeoff`) . . . 100
9.6.2 Transizione in Avanti (`fw_transition`) . . . 101
9.6.3 Crociera (`cruise`) . . . 101
9.6.4 Virata (`turn`) . . . 101
9.6.5 Decelerazione (`decel`) . . . 102
9.6.6 Approach (`approach`) . . . 102
9.6.7 Hover Pre-Landing, Landing, Hover Delivery . . . 103
9.7 Phase Planner e Logica di Transizione . . . 103
9.8 Generazione dell'Output di Riferimento . . . 105

10 Risultati e Analisi delle Simulazioni 106
10.1 Introduzione . . . 106
10.2 Scenario di Missione Simulato . . . 107
10.3 Inseguimento della Traiettoria . . . 107
10.4 Analisi dell'Errore di Inseguimento . . . 111
10.5 Le Transizioni Hover–Crociera . . . 113
10.6 Le Virate Coordinate . . . 118
10.7 Impegno degli Attuatori . . . 122
10.8 Atterraggio Verticale . . . 123
10.9 Sintesi dei Risultati . . . 125

11 Conclusioni 126
11.1 Sintesi del Lavoro Svolto . . . 126
11.2 Principali Risultati . . . 127
11.3 Limiti del Lavoro . . . 127
11.4 Sviluppi Futuri . . . 128

A Parametri del Modello e dei Controllori 129
A.1 Introduzione . . . 129
A.2 Parametri Fisici del Velivolo . . . 130
A.3 Propulsione . . . 131
A.4 Aerodinamica e Superfici di Controllo . . . 132
A.5 Modello di Contatto col Suolo . . . 132

A.6 Controllo di Posizione 133
A.7 Controllo di Assetto e Rate Loop 133
A.8 Allocazione e Blending 134
A.9 Navigazione 134
A.10 Parametri di Simulazione 134
A.11 Note sui Parametri 134

Bibliografia 136

Elenco delle figure

1.1 Tassonomia semplificata delle architetture UAV con capacità VTOL. Il percorso evidenziato indica la configurazione quad-motor tailsitter con elevoni adottata nel presente lavoro. 2

1.2 Render assonometrico del tailsitter quad-motore. 5

2.1 Vista frontale (y–z) del veicolo con la numerazione dei motori, gli assi del body frame e i versi di rotazione. I motori 1 e 3 ruotano in senso orario (CW, blu); i motori 2 e 4 in senso antiorario (CCW, rosso): l'alternanza diametrale annulla la coppia reattiva netta in hover. 15

2.2 Viste del modello OpenVSP del tailsitter (isometrica e ortogonali). . 17

4.1 Vista complessiva del modello Simulink del tailsitter: dal modello di *Guidance* (sinistra) al *plant* (destra), passando per i cinque livelli del sistema di controllo. 32

4.2 Interfaccia del blocco Plant: ingressi di comando e uscite di stato. . 34

4.3 Struttura interna del blocco Plant: sottosistemi Forces & Moments e Equations of Motion. 35

4.4 Sottosistema Forces & Moments. 36

4.5 Sottosistema Equations of Motion: blocchi Dynamics e Kinematics con integratori. 44

4.6 Blocco Simulink della dinamica traslazionale. 45

4.7 Blocco Simulink della dinamica rotazionale. 46

4.8 Blocco Simulink della cinematica dell'assetto: integrazione dei quaternioni con normalizzazione. 47

4.9 Blocco Simulink della cinematica della posizione: trasformazione body-to-inertial e integrazione. 48

5.1 Interfaccia esterna del blocco Control Allocation nel modello Simulink. 51

5.2 Sottosistema Simulink dell'allocazione dei controlli: blocco di blending (a sinistra, in azzurro), allocazione motore (in alto a destra, in giallo) e allocazione elevone (in basso a destra, in rosso). 52
5.3 Blocco di blending: calcolo della pressione dinamica reale, dei tetti fisici degli elevoni e ripartizione dei momenti tra motori ed elevoni. . 53
5.4 Blocco di allocazione ai motori: inversione della matrice di allocazione e saturazione delle velocità angolari al quadrato. 59
5.5 Blocco di allocazione elevoni: calcolo delle deflessioni virtuali (δ_e, δ_a), combinazione nelle deflessioni fisiche (δ_{eL}, δ_{eR}) e saturazione ai limiti meccanici. 61

6.1 Interfaccia del blocco Attitude Control: ingressi di stato e uscita di coppia desiderata. 64
6.2 Struttura interna del sottosistema Attitude Control: il blocco Attitude Controller genera $\boldsymbol{\omega}_{\text{des}}$, il blocco Rate PID produce la coppia $\boldsymbol{\tau}_{\text{des}}$. 64
6.3 Dettaglio del blocco Rate PI con feed-forward: ramo di feed-forward (in alto), ramo proporzionale, ramo integrale con anti-windup condizionale e saturazione (in basso). 68
6.4 Sottoblocco del feed-forward inerziale: filtro a primo ordine su ciascun asse, differenza con il segnale originale e moltiplicazione per $\mathbf{I}/\tau_{\text{rate,cl}}$. 70

7.1 Interfaccia esterna del blocco *Attitude Command*. 78
7.2 Struttura interna del sottosistema *Attitude Command*: una singola MATLAB Function `fcn_coordinated_flight` aggrega l'intera logica di generazione del comando di assetto e ne espone gli output principali e di debug. 79

8.1 Interfaccia esterna del blocco *Position Control*. 90
8.2 Struttura interna del blocco *Position Control*: anello PID nel frame inerziale con feed-forward, saturazione di accelerazione, compensazione di gravità e forza aero-propulsiva richiesta. 90

9.1 Interfaccia esterna del sottosistema *Guidance*: i riferimenti di traiettoria ($\boldsymbol{p}_{\text{des}}$, $\boldsymbol{v}_{\text{des}}$, $\boldsymbol{a}_{\text{ff}}$, ψ_{des}) sono prodotti in funzione del tempo, dello stato logico della missione e della specifica di missione `mission_spec`. L'identificatore di fase corrente (`flight_mode`) è esposto come output e impiegato altrove nel modello per il logging e la visualizzazione per fase. 95

10.1 Traiettoria tridimensionale dell'intera missione: confronto fra la traiettoria misurata e quella di riferimento. La scala cromatica rappresenta il modulo della velocità relativa V_{air}. 108
10.2 Vista in pianta (piano Nord–Est) della traiettoria di missione. La coincidenza dei marcatori di partenza e di arrivo conferma il rientro alla base. 108
10.3 Profilo di quota in funzione della distanza orizzontale percorsa. Le discese verticali fino a quota nulla corrispondono agli atterraggi presso i siti operativi e la base. 109
10.4 Evoluzione temporale delle componenti di posizione: confronto fra valore misurato e riferimento. 110
10.5 Componenti della velocità inerziale nel frame NED durante l'intera missione. 110
10.6 Errore di posizione istantaneo $\boldsymbol{p} - \boldsymbol{p}_{\text{des}}$, scomposto nelle tre componenti NED. 111
10.7 Norma dell'errore di posizione istantaneo $\|\boldsymbol{e}_p\|$ durante l'intera missione. 112
10.8 Angolo di beccheggio θ durante l'intera missione: confronto fra valore misurato e riferimento. Le escursioni fra 90° e 4° corrispondono alle transizioni hover–crociera. 113
10.9 Modulo della velocità relativa V_{air} durante l'intera missione. La linea tratteggiata indica la velocità di transizione V_{trans}. 113
10.10 Dettaglio dell'angolo di beccheggio θ su una finestra comprendente una transizione in avanti e la successiva decelerazione. 114
10.11 Modulo della velocità relativa V_{air} durante una transizione in avanti e la successiva decelerazione. 114
10.12 Dettaglio dell'angolo di beccheggio θ durante una transizione in avanti. 115
10.13 Dettaglio dell'angolo di beccheggio θ durante una decelerazione con richiamo all'assetto di hover. 115
10.14 Rappresentazione tridimensionale dell'assetto del velivolo durante una transizione in avanti. 116
10.15 Vista laterale della transizione in avanti: la rotazione di beccheggio avviene a quota sostanzialmente costante. 116
10.16 Rappresentazione tridimensionale dell'assetto del velivolo durante una decelerazione con richiamo all'assetto di hover. 117
10.17 Vista laterale della transizione all'indietro: la rotazione di beccheggio avviene a quota sostanzialmente costante. 117
10.18 Angoli di Eulero durante due virate coordinate consecutive: confronto fra valore misurato e riferimento. 118
10.19 Dettaglio in pianta di una virata coordinata: confronto fra traiettoria misurata e arco di riferimento. 119

10.20Errore di posizione istantaneo durante due virate coordinate consecutive, scomposto nelle tre componenti NED. 120
10.21Norma dell'errore di posizione durante due virate coordinate consecutive. 120
10.22Rappresentazione tridimensionale dell'assetto del velivolo durante una virata coordinata. 121
10.23Spinta totale comandata T_{des} durante l'intera missione. La linea tratteggiata indica il livello di equilibrio mg. 122
10.24Velocità di rotazione dei quattro motori durante l'intera missione. . 122
10.25Deflessione dei due elevoni $\delta_{e,L}$ e $\delta_{e,R}$ durante l'intera missione. Le linee tratteggiate indicano i limiti di saturazione meccanica. 123
10.26Rappresentazione tridimensionale dell'assetto del velivolo durante un atterraggio verticale. 124

Elenco delle tabelle

2.1 Requisiti prestazionali del tailsitter. 8
2.2 Parametri geometrici della fusoliera. 9
2.3 Parametri geometrici dell'ala. 10
2.4 Parametri degli elevoni. 11
2.5 Parametri dei bracci motore. 12
2.6 Parametri dei pod propulsivi. 13
2.7 Specifiche tecniche del motore T-Motor MN5008 KV340. 13
2.8 Posizioni e sensi di rotazione dei motori. 15
2.9 Componenti interni e relativi volumi di alloggiamento. 18
2.10 Distribuzione della massa del veicolo. 19
2.11 Sommario delle specifiche di design. 22

3.1 Script di inizializzazione e relativa funzione. 24
3.2 Tolleranze numeriche utilizzate nel modello. 25
3.3 Topologia della missione di riferimento. 28

4.1 Derivate di stabilità aerodinamica e relativo significato fisico. 41

6.1 Guadagni del controllore di assetto e del rate loop. Valori unificati su tutto l'inviluppo di volo. 73
6.2 Bande passanti e rapporto di separazione (set di guadagni unificati con feed-forward). 76

7.1 Sequenza di operazioni del generatore del comando di assetto. . . . 87

9.1 Primitive di volo della macchina di guidance. 98
9.2 Transizioni nominali del phase planner. 104

A.1 Parametri fisici del velivolo tailsitter. 130
A.2 Parametri della propulsione (T-Motor MN5008 KV340 + elica 17×5.8" tripala). 131
A.3 Coefficienti aerodinamici e parametri delle superfici di controllo. . . 132

A.4 Parametri del modello di contatto col suolo. 132
A.5 Guadagni del controllore PID di posizione e parametri di saturazione. 133
A.6 Guadagni del controllore di assetto, rate loop e feed-forward. 133
A.7 Parametri dell'allocazione adattiva. 134
A.8 Parametri della macchina a stati di navigazione. 134
A.9 Parametri di integrazione e simulazione. 134

Capitolo 1

Introduzione

1.1 Velivoli a Decollo e Atterraggio Verticale: Stato dell'Arte

L'ultimo decennio ha assistito a una crescita esponenziale nell'impiego di sistemi aerei senza pilota (*Unmanned Aerial Vehicles*, UAV) in ambito civile e militare [4, 6]. Applicazioni quali la fotogrammetria aerea, il monitoraggio ambientale, la ricognizione tattica, le operazioni di ricerca e soccorso e la logistica di ultimo miglio hanno stimolato lo sviluppo di architetture sempre più sofisticate, capaci di combinare le prestazioni di volo in avanti tipiche delle piattaforme ad ala fissa con la flessibilità operativa del decollo e atterraggio verticale (VTOL) [51]. La convergenza di queste esigenze ha dato origine a una classe eterogenea di velivoli ibridi, il cui panorama tassonomico è sintetizzato nella Figura 1.1.

Le architetture VTOL attualmente impiegate si distinguono in base alla strategia con cui si realizza la transizione tra volo stazionario e volo in avanti. I multirotori (quadricotteri, esacotteri, ottacotteri) offrono eccellente manovrabilità e semplicità di controllo in hover, ma la loro efficienza aerodinamica è intrinsecamente limitata dall'assenza di superfici portanti fisse. All'estremo opposto, le piattaforme ad ala fissa convenzionale garantiscono efficienza in crociera e autonomie dell'ordine delle ore, ma richiedono infrastrutture di decollo e atterraggio che ne riducono la flessibilità di dispiegamento.

Per colmare il divario tra queste due categorie, sono state sviluppate diverse architetture ibride. I quadplane aggiungono rotori verticali dedicati a una cellula ad ala fissa, ottenendo capacità VTOL a costo di massa e resistenza aerodinamica aggiuntive dovute ai componenti non operativi in crociera. Le configurazioni tiltrotor e tiltwing ruotano rispettivamente i rotori o l'intera ala per ridirigere la spinta da verticale a orizzontale: la transizione è aerodinamicamente efficiente, ma la complessità meccanica dei meccanismi di tilting e la massa dei relativi

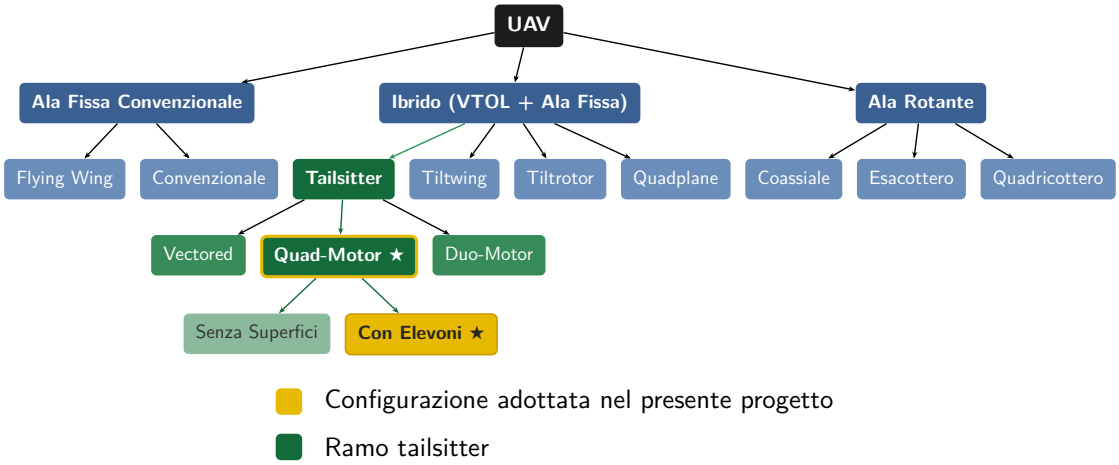


Figura 1.1: Tassonomia semplificata delle architetture UAV con capacità VTOL. Il percorso evidenziato indica la configurazione quad-motor tailsitter con elevoni adottata nel presente lavoro.

servomeccanismi costituiscono penalizzazioni significative in termini di affidabilità e costo.

Il tailsitter rappresenta un approccio radicalmente differente: il veicolo decolla e atterra in assetto verticale, appoggiato sulla coda, e transita al volo orizzontale ruotando l'intero corpo rigido attorno all'asse laterale [5, 18, 48]. Nella sua forma più essenziale, il tailsitter impiega lo stesso sistema propulsivo a orientamento fisso per entrambe le fasi di volo, eliminando la necessità di servomeccanismi di tilting e rotori ausiliari. I vantaggi principali di questa architettura sono:

1. **Semplicità meccanica:** il cambio di modalità di volo è una conseguenza della rotazione strutturale, senza componenti mobili aggiuntivi.
2. **Efficienza in crociera:** la configurazione *wing-borne* sfrutta l'aerodinamica di ala fissa, raggiungendo autonomie significativamente superiori rispetto ai multirotori.
3. **Rapporto spinta/peso:** l'impiego dello stesso sistema propulsivo per hover e crociera elimina la massa di rotori ausiliari o meccanismi di tilting, ottenendo un rapporto spinta/peso complessivo più favorevole a parità di MTOW.
4. **Indipendenza da infrastrutture:** il decollo e l'atterraggio verticali non richiedono piste, catapulte o reti di recupero, consentendo il dispiegamento in ambienti operativi non preparati.

All'interno della famiglia dei tailsitter si distinguono diverse sotto-categorie, che differiscono per numero di rotori e strategia di controllo. Le configurazioni

duo-motor (due rotori con elevoni, come il WingtraOne) e *quad-motor* (quattro rotori in configurazione X, con o senza superfici di controllo) mantengono i rotori fissi rispetto alla cellula. Esistono inoltre configurazioni *vectored* che introducono la possibilità di orientare i rotori indipendentemente dalla fusoliera: queste varianti sacrificano parte della semplicità meccanica tipica del tailsitter in favore di una maggiore autorità di controllo, specialmente durante le fasi di transizione.

La configurazione quad-motor con elevoni a rotori fissi, oggetto del presente lavoro, combina l'autorità di controllo in hover propria del quadricottero con la capacità di generare forze e momenti aerodinamici in crociera tramite le superfici di controllo, mantenendo al contempo la semplicità meccanica che caratterizza il concetto di tailsitter.

Tuttavia, il tailsitter a rotori fissi introduce sfide di controllo particolarmente complesse. La transizione tra hover e volo in avanti richiede una rotazione di circa 90° dell'intero veicolo, con una contestuale variazione del regime aerodinamico da dominanza di spinta a dominanza di portanza. La letteratura tradizionale affronta questa problematica mediante commutazione (*switching*) tra controllori monomodali [13, 24], approccio che introduce discontinuità e può portare a comportamenti indesiderati nelle fasi intermedie. Lavori recenti hanno proposto controllori unificati basati su formulazioni globali dell'assetto, dimostrando la fattibilità di architetture di controllo prive di commutazione modale sia per configurazioni *flying-wing* sia per configurazioni quad-motor con elevoni [37, 44, 49].

1.2 Obiettivi della Ricerca

Lo scopo della presente tesi è lo sviluppo e la validazione numerica di un'architettura di controllo unificata per un tailsitter quadrirotore con elevoni, capace di gestire l'intero inviluppo di volo, dall'hover stazionario alla crociera ad alta velocità, senza ricorso a commutazione di controllori. Gli obiettivi specifici comprendono:

- Lo sviluppo di un modello dinamico non lineare a 6 gradi di libertà (6-DOF), formulato in quaternioni per evitare le singolarità tipiche degli angoli di Eulero durante le rotazioni di grande ampiezza.

- La progettazione di un'architettura di controllo gerarchica.

- La validazione dell'architettura proposta attraverso simulazioni nonlineari in MATLAB/Simulink su scenari di missione completa, comprendenti decollo verticale, transizione, crociera e atterraggio.

1.3 Metodologia

La ricerca si articola in quattro macrofasi metodologiche. La modellazione geometrica del veicolo è condotta in OpenVSP, strumento open-source sviluppato dalla NASA per la parametrizzazione e l'analisi geometrica di veicoli aerospaziali, dal quale si estraggono le proprietà geometriche, inerziali e di drag parassita necessarie alle fasi successive. I coefficienti aerodinamici dell'ala (C_L, C_D, C_m) sono identificati separatamente mediante reti neurali *shallow* addestrate su dati sperimentali del profilo NACA 0012, fornendo predittori continui e derivabili validi su un intervallo esteso di angoli di attacco (incluso lo stallo); le rimanenti derivate di stabilità sono invece estratte da VSPAero e considerate costanti su tutta la missione. Lo sviluppo del modello dinamico avviene in MATLAB/Simulink, integrando le equazioni del moto di Newton–Euler in forma quaternionica. La sintesi del controllore è articolata su livelli gerarchici a banda passante crescente verso l'interno: pianificazione di traiettoria, controllo di posizione, generazione del comando di assetto, controllo di assetto a cascata e allocazione dei controlli. La validazione avviene attraverso simulazione non lineare a 6-DOF su missioni complete multi-leg, con analisi di robustezza e verifica della fattibilità dei comandi attuatore.

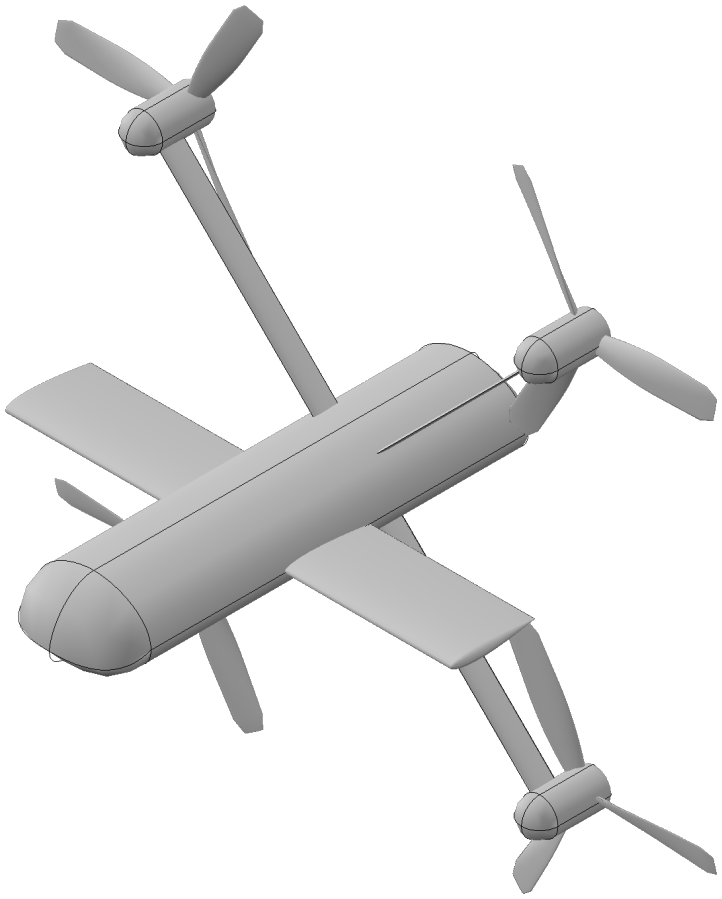


Figura 1.2: Render assonometrico del tailsitter quad-motore.

1.4 Fasi di Volo

Il tailsitter opera attraverso quattro regimi di volo principali, ciascuno caratterizzato da un equilibrio aerodinamico differente, ma tutti gestiti dalla medesima legge di controllo unificata, senza commutazione modale:

1. **Hover:** il veicolo si trova in assetto verticale, con la spinta dei quattro motori bilanciata contro la forza gravitazionale. Il controllo di rollio, beccheggio e imbardata è affidato al differenziale di spinta tra i motori; gli elevoni risultano in pratica inefficaci a causa della bassa pressione dinamica.

2. **Transizione *forward*:** rotazione progressiva attorno all'asse laterale (y_B) dall'assetto verticale all'assetto di crociera. Con l'incremento della velocità, la portanza alare assume gradualmente il sostentamento del veicolo e gli elevoni acquisiscono autorità di controllo. La transizione è gestita da un *blending* sigmoidale dell'assetto desiderato fra una soluzione di hover e una di crociera, e da un'allocazione adattiva dei controlli che ripartisce momenti di rollio e beccheggio fra motori ed elevoni in funzione della pressione dinamica reale.

3. **Crociera:** volo in avanti in regime *wing-borne*, con controllo affidato alla deflessione degli elevoni in beccheggio e rollio, modulazione della spinta sul controllo longitudinale e differenziale di spinta sul momento di imbardata. L'assetto di riferimento è il trim livellato, calcolato in tempo reale risolvendo un sistema di equilibrio aerodinamico via Newton–Raphson su angolo di attacco e spinta.

4. **Transizione *backward*:** fase di richiamo dalla crociera all'hover, ottenuta come traiettoria inversa del medesimo *blending* sigmoidale. Lo stesso solver di assetto coordinato, riapplicato con la velocità di riferimento decrescente, genera una rampa continua di quaternione desiderato dall'assetto di crociera a quello di hover, eliminando la necessità di una macchina di stato dedicata o di un *override* esplicito.

1.5 Struttura della Tesi

La tesi è organizzata in modo da seguire il processo di sviluppo del velivolo dalla concezione geometrica fino al sistema di controllo completo, procedendo dal *centro* del modello (la piattaforma fisica) verso l'esterno (il pianificatore di missione). Il Capitolo 2 presenta le specifiche di progetto e il design del velivolo, dal dimensionamento aerodinamico alla scelta dei componenti propulsivi, descrivendo in dettaglio il lavoro di modellazione geometrica in OpenVSP e l'identificazione dei coefficienti aerodinamici. Il Capitolo 3 documenta gli script MATLAB di

inizializzazione che popolano la struttura dei parametri condivisa con il modello Simulink. Il Capitolo 4 formalizza il modello dinamico a sei gradi di libertà, descrivendo i sottosistemi di forze, momenti, equazioni del moto e cinematica.

I Capitoli 5–8 sviluppano il sistema di controllo procedendo verso l'esterno: l'allocazione adattiva dei controlli (Capitolo 5), il controllore di assetto a cascata con feed-forward inerziale sul *rate loop* (Capitolo 6), il generatore del comando di assetto basato sul *coordinated flight* di Ritz e D'Andrea (Capitolo 7), e infine il controllore di posizione PID nel frame inerziale (Capitolo 8). Il Capitolo 9 descrive il sistema di guida e navigazione, con particolare riguardo al sistema che orchestra le missioni complete attraverso primitive di volo settiche di classe $\mathcal{C}^3$. Il Capitolo 10 riporta i risultati delle simulazioni di validazione. L'appendice A raccoglie il riepilogo completo dei parametri del modello e dei controllori.

Avvertenza sulla lettura degli schemi Simulink. Nei capitoli che illustrano i sottosistemi del modello, ciascun blocco è presentato secondo lo schema *vista esterna* $\rightarrow$ *vista interna*: si introduce dapprima la sua interfaccia, ovvero le porte di ingresso e uscita visibili al livello superiore, per poi entrare gradualmente nel dettaglio della logica interna. È opportuno precisare fin da subito che le linee esterne di un sottosistema, come visualizzate nella rispettiva figura, non corrispondono necessariamente al numero complessivo dei segnali effettivamente consumati o prodotti dal blocco: rappresentano soltanto i segnali principali, quelli che si è ritenuto opportuno mantenere espliciti per favorire la leggibilità del modello al livello superiore. Tutti gli altri segnali (riferimenti ausiliari, parametri di stato della macchina di navigazione, variabili di scoping, vento relativo) sono propagati internamente attraverso coppie di blocchi `Goto`/`From` distribuite nel modello. Si tratta di una scelta puramente di *layout* che riduce la densità del cablaggio nelle figure principali senza alterare la struttura logica del sistema.

1.6 Notazioni e Convenzioni

Nel corso della tesi si adottano le seguenti notazioni: vettori in grassetto ($\boldsymbol{v}$), con coordinate nel frame inerziale $\mathcal{F}_I$ o nel frame corpo $\mathcal{F}_B$; matrici in maiuscolo grassetto ($\mathbf{R}$, $\mathbf{M}$); quaternioni $\mathbf{q} = [q_0,\, q_1,\, q_2,\, q_3]^T = [q_0,\, \boldsymbol{q}_v]^T$. La velocità angolare del corpo è indicata con $\boldsymbol{\omega}_B$, espressa nel frame corpo. I pedici $_I$ e $_B$ distinguono rispettivamente le quantità riferite al frame inerziale e quelle riferite al frame corpo; i pedici ref, meas e cmd indicano riferimenti desiderati, misurazioni e comandi di controllo.

Capitolo 2

Specifiche di Progetto e Design del Velivolo

2.1 Requisiti Prestazionali

Il veicolo è stato concepito secondo un insieme di requisiti tecnici e operazionali che guidano le scelte progettuali successive. La Tabella 2.1 sintetizza i principali obiettivi di prestazione.

Tabella 2.1: Requisiti prestazionali del tailsitter.

Parametro	**Valore**	**Simbolo**
Capacità di hover	Sì	—
Velocità di crociera[a]	$220\,\mathrm{km\,h^{-1}}$	—
Altitudine operativa (AGL)	$200\,\mathrm{m}$	—
Massa al decollo (MTOW)	$7\,\mathrm{kg}$	—
Tempo di transizione hover–crociera[b]	$30\,\mathrm{s}$	—

[a] Velocità target di riferimento in regime stazionario di crociera.
[b] Durata nominale della manovra di transizione hover→crociera.

La velocità di crociera di $220\,\mathrm{km\,h^{-1}}$ è coerente con applicazioni di logistica sanitaria e di trasporto di dispositivi medici, in cui la riduzione del tempo di transito è un parametro critico, ma anche con missioni di ricognizione tattica e mappatura ad alta risoluzione, dove la riduzione del tempo di sorvolo e l'estensione dell'area coperta per sortita sono parametri determinanti. Il requisito di hover stazionario consente operazioni di ispezione puntuale e di decollo/atterraggio in spazi confinati.

2.2 Dimensionamento Geometrico

A partire dai requisiti prestazionali, le dimensioni caratteristiche del veicolo sono state definite attraverso un processo iterativo di dimensionamento concettuale, bilanciando vincoli aerodinamici e di payload interno. I paragrafi seguenti descrivono le scelte progettuali per ciascun componente e le motivazioni che le hanno guidate.

2.2.1 Fusoliera

La fusoliera costituisce l'elemento strutturale portante del veicolo e ospita l'avionica, la batteria e il payload. Le dimensioni sono state determinate dal volume necessario per alloggiare i componenti interni, mantenendo al contempo un profilo a bassa resistenza aerodinamica.

Tabella 2.2: Parametri geometrici della fusoliera.

Parametro	**Valore**	**Simbolo**
Lunghezza	0.62 m	L_{fus}
Larghezza	0.15 m	—
Altezza	0.15 m	—
Massa strutturale	0.6 kg	—

La sezione trasversale della fusoliera è asimmetrica rispetto al piano orizzontale: il ventre è quasi piatto per facilitare l'integrazione dei volumi interni, mentre il dorso presenta un profilo a cupola. In OpenVSP questa geometria è stata ottenuta mediante un body of revolution con sezioni trasversali (`XSec`) parametriche personalizzate.

2.2.2 Ala

L'ala è il componente preposto alla generazione della forza di sostentamento aerodinamica in volo di crociera. Il dimensionamento è partito dalla scelta del profilo, individuato nel NACA 0012. La superficie alare è stata quindi ricavata imponendo l'equilibrio verticale in crociera ($L = W$); fissato un valore di allungamento, ne sono state derivate apertura e corda media. I parametri risultanti sono riportati in Tab. 2.3.

Il NACA 0012, oltre alla simmetria, presenta momento di beccheggio sostanzialmente nullo a incidenza nulla ($C_{m,0} \approx 0$), il che semplifica il trimmaggio in crociera. L'ala è posizionata con il bordo di attacco a 0.22 m dal muso, in modo da collocare il centro aerodinamico (approssimativamente al 25% della corda media,

Tabella 2.3: Parametri geometrici dell'ala.

Parametro	**Valore**	**Simbolo**
Profilo	NACA 0012	—
Corda media aerodinamica	0.14 m	$\bar{c}$
Apertura alare	0.73 m	b
Superficie alare	0.1022 m^2	S_w
Aspect Ratio	5.21	AR
Posizione bordo di attacco (da *nose*)	0.22 m	x_{LE}
Massa (struttura in carbonio)	0.225 kg	m_w
Angolo di freccia	0°	Λ
Diedro	0°	Γ
Twist geometrico	0°	ε

$x_{\mathrm{AC}} \approx 0.22 + 0.035 = 0.255\,\mathrm{m}$) in posizione favorevole rispetto al baricentro ai fini della stabilità statica longitudinale.

2.2.3 Elevoni

Gli elevoni (elevator-ailerons) sono superfici di controllo ibride integrate nel bordo di uscita dell'ala. In hover aiutano nella stabilizzazione del veicolo, anche se con poca autorità; in crociera, la deflessione differenziale genera il momento di rollio, mentre la deflessione simmetrica genera momento di beccheggio.

Tabella 2.4: Parametri degli elevoni.

Parametro	**Valore**	**Simbolo**
Rapporto corda elevone/corda ala (L/C)	0.30	—
Estensione in apertura: inizio (η_{start})	0.30	—
Estensione in apertura: fine (η_{end})	0.945	—
Deflessione massima	$\pm 30°$	δ_{max}
Dinamica servocomando	$100\,°\,\mathrm{s}^{-1}$	rate massimo

L'estensione in apertura tra $\eta = 0.30$ e $\eta = 0.945$ lascia libera la zona di radice dell'ala (dove il flusso è influenzato dalla fusoliera e dai bracci motore) e la punta alare (per evitare interazioni con i vortici marginali), concentrando l'efficacia di controllo nella regione a massimo braccio aerodinamico.

Il controllo di beccheggio si ottiene per deflessione simmetrica ($\delta_L = \delta_R = \delta_{\mathrm{sym}}$), che genera un momento:

$$M_y = q_\infty \cdot S_w \cdot c \cdot C_{m,\delta} \cdot \delta_{\mathrm{sym}} \tag{2.1}$$

Il controllo di rollio si ottiene per deflessione differenziale ($\delta_L = -\delta_R = \delta_{\mathrm{asym}}$):

$$M_x = q_\infty \cdot S_w \cdot b \cdot C_{l,\delta} \cdot \delta_{\mathrm{asym}} \tag{2.2}$$

dove $q_\infty = \frac{1}{2}\rho V^2$ è la pressione dinamica.

2.2.4 Bracci Motore

I quattro bracci motore collegano i pod propulsivi alla fusoliera. Lo span totale dei bracci (dalla punta di un pod al pod opposto) è di 0.90 m, cosicché ogni braccio ha una semi-estensione di 0.45 m dal centro della fusoliera. I bracci sono modellati in OpenVSP come superfici alari a sezione sottile.

La posizione longitudinale dei bracci a $x = 0.413\,\mathrm{m}$ è stata scelta in modo che i motori risultino posizionati leggermente a valle del bordo di fuga dell'ala, per

Tabella 2.5: Parametri dei bracci motore.

Parametro	**Valore**	**Simbolo**
Posizione longitudinale (da *nose*)	0.413 m	x_{arm}
Span totale (coppia superiore + inferiore)	0.90 m	—
Semi-estensione (per braccio)	0.45 m	—
Corda	0.04 m	—
Massa per braccio	0.1 kg	—
Numero di bracci	4	—

semplicità costruttiva. La distanza $d_y = d_z \approx 0.325\,\text{m}$ tra ciascun motore e il baricentro definisce il braccio di leva efficace per la generazione dei momenti di controllo.

2.2.5 Pod Propulsivi

Ciascun pod è un corpo di rivoluzione che alloggia un motore e il relativo ESC (*Electronic Speed Controller*). La forma è compatta e affusolata, con estremità arrotondate (*round cap*) per minimizzare la resistenza aerodinamica.

Tabella 2.6: Parametri dei pod propulsivi.

Parametro	**Valore**	**Simbolo**
Diametro sezione (`XSec`)	0.06 m	—
Lunghezza	0.10 m	—
Forma estremità	Round cap	—
Massa (motore + ESC + struttura)	0.250 kg	—
Numero di pod	4	—

2.3 Configurazione Propulsiva

2.3.1 Motore Elettrico: T-Motor MN5008 KV340

Il sistema propulsivo impiega quattro motori brushless T-Motor MN5008 KV340 [31], selezionati per il rapporto spinta/massa, l'affidabilità e la disponibilità di dati di caratterizzazione. La Tabella 2.7 riporta le specifiche tecniche.

Tabella 2.7: Specifiche tecniche del motore T-Motor MN5008 KV340.

Parametro	**Valore**	**Simbolo**
Costante di velocità (KV)	340 RPM/V	—
RPM massimi (a $V_{\text{batt}} = 22.2\,\text{V}$, 6S LiPo)	6482 RPM	—
Velocità angolare massima	$679\,\text{rad}\,\text{s}^{-1}$	—
Spinta massima (P17×5.8" tripala)	47.1 N	—
Fattore tripala	1.35	—
Coppia massima	0.920 N m	—
Resistenza interna	$0.087\,\Omega$	—

La spinta prodotta dal singolo motore con elica è modellata come funzione quadratica della velocità angolare:

$$T = k_T \cdot \Omega^2 \tag{2.3}$$

dove k_T è il coefficiente di spinta. La spinta massima per motore è ricavata dal datasheet del produttore, che riporta $T_{\max,2\text{pale}} = 34.9\,\text{N}$ in configurazione bipala al regime massimo $\Omega_{\max} = 679\,\text{rad}\,\text{s}^{-1}$ (6482 RPM con batteria 6S a 22.2 V). La configurazione effettivamente impiegata sul velivolo è invece tripala (elica P17×5.8" in fibra di carbonio), che incrementa la spinta erogata con un fattore correttivo di circa 1.35 (rapporto sperimentale tipico tra elica tripala e bipala a parità di RPM e diametro):

$$T_{\max} = T_{\max,2\text{pale}} \cdot 1.35 = 34.9 \cdot 1.35 = 47.1\,\text{N}, \tag{2.4}$$

da cui:

$$k_T = \frac{T_{\max}}{\Omega_{\max}^2} = \frac{47.1}{679^2} = 1.022 \times 10^{-4}\,\text{N}/(\text{rad/s})^2. \tag{2.5}$$

La coppia reattiva è legata alla spinta tramite il rapporto coppia–spinta dell'elica κ, anch'esso ricavato dal datasheet bipala come $\kappa = Q_{\max,2\text{pale}}/T_{\max,2\text{pale}} = 0.69/34.9 \approx 0.0198$:

$$Q = \kappa \cdot T, \qquad k_Q = \kappa \cdot k_T = 2.021 \times 10^{-6}\,\text{Nm}/(\text{rad/s})^2. \tag{2.6}$$

L'invarianza di κ tra configurazione bipala e tripala è una proprietà del singolo profilo d'elica: la rotazione di una pala genera, a parità di velocità angolare, una coppia proporzionale alla spinta che essa stessa produce, indipendentemente dalla presenza di altre pale. La generalizzazione del fattore 1.35 alla coppia totale, automatica nel passaggio bipala → tripala, è già incorporata nella definizione di k_Q tramite il prodotto $\kappa \cdot k_T$, dove k_T contiene il fattore 1.35 del totale di spinta.

2.3.2 Disposizione Motori (X-Configuration)

I quattro motori sono disposti in configurazione X nel piano trasversale (y–z del frame corpo), con alternanza di senso di rotazione per il bilanciamento dei momenti reattivi. La Figura 2.2b mostra la vista frontale del veicolo con la numerazione dei motori e i versi di rotazione, mentre la Tabella 2.8 ne formalizza posizione e senso.

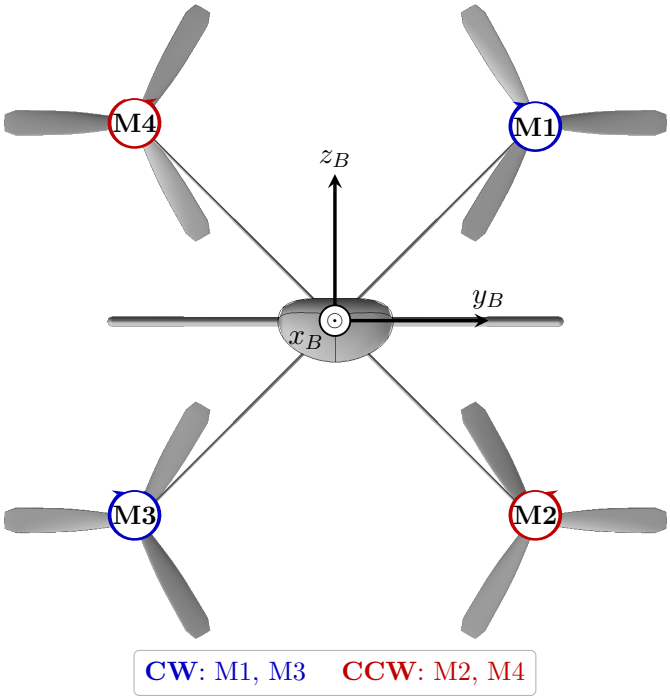


Figura 2.1: Vista frontale (y–z) del veicolo con la numerazione dei motori, gli assi del body frame e i versi di rotazione. I motori 1 e 3 ruotano in senso orario (CW, blu); i motori 2 e 4 in senso antiorario (CCW, rosso): l'alternanza diametrale annulla la coppia reattiva netta in hover.

Tabella 2.8: Posizioni e sensi di rotazione dei motori.

Motore	y **(m)**	z **(m)**	**Rotazione**	**Senso**
1	$+d$	$+d$	Orizzontale	CW
2	$+d$	$-d$	Orizzontale	CCW
3	$-d$	$-d$	Orizzontale	CW
4	$-d$	$+d$	Orizzontale	CCW

dove $d = d_y = d_z = 0.325\,\text{m}$.

In hover (assetto verticale, asse x_B verticale), i momenti di controllo generati dal differenziale di spinta sono:

$$\boldsymbol{M}_{\text{ctrl}} = \begin{bmatrix} -Q_1 + Q_2 - Q_3 + Q_4 \\ (T_1 - T_2 - T_3 + T_4) \cdot d \\ (-T_1 - T_2 + T_3 + T_4) \cdot d \end{bmatrix} \tag{2.7}$$

dove la prima componente è il momento di rollio (asse x_B) dovuto ai momenti reattivi, la seconda è il momento di beccheggio (asse y_B) e la terza il momento di imbardata (asse z_B), entrambi generati dal differenziale di spinta.

2.3.3 Eliche

Le eliche utilizzate sono T-Motor P17×5.8" in configurazione tripala. Il fattore di spinta tripala rispetto alla bipala equivalente è di 1.35, consentendo di ottenere la spinta necessaria a regimi di rotazione contenuti.

2.4 Modellazione Geometrica in OpenVSP

2.4.1 Processo di Modellazione

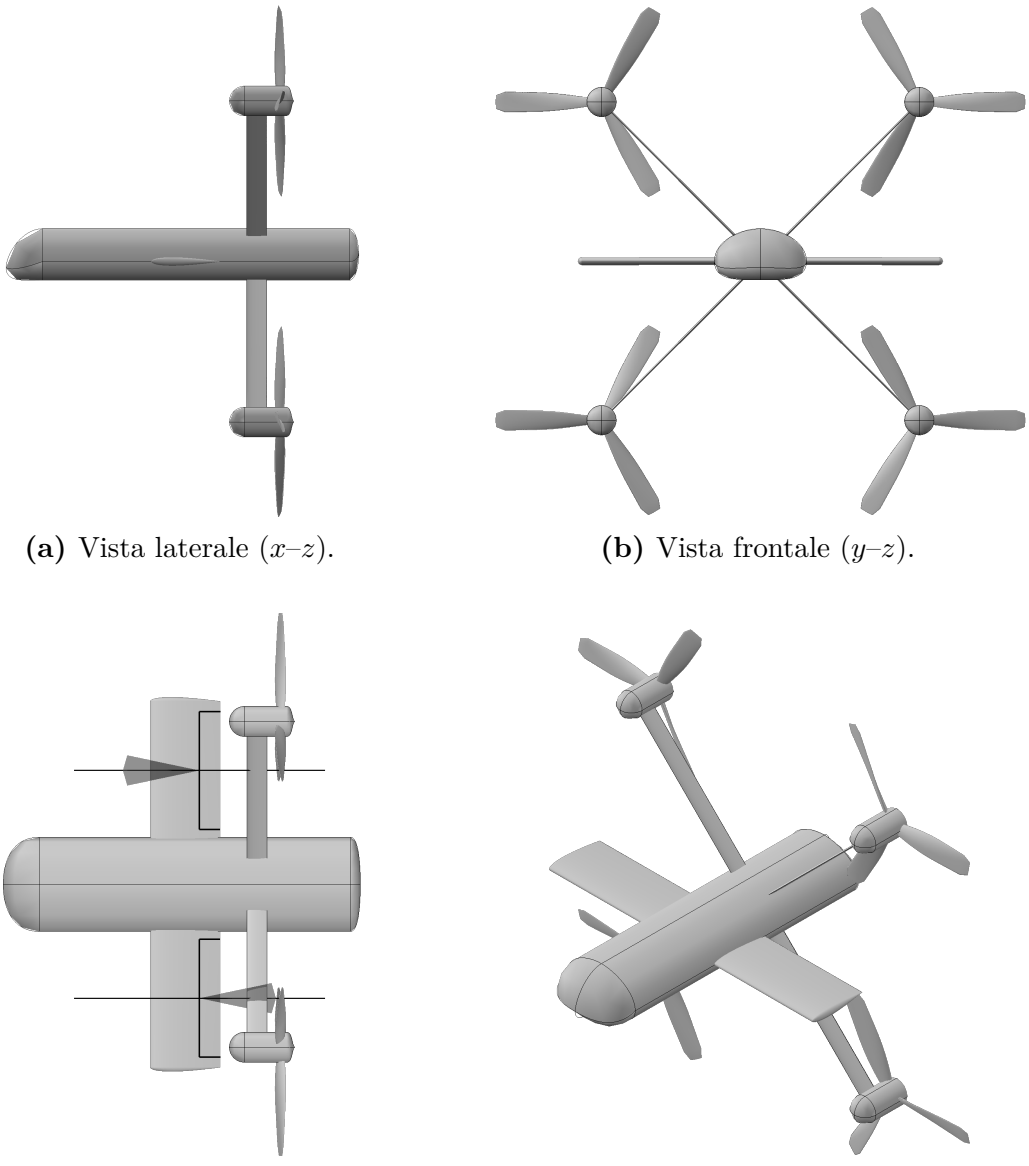
OpenVSP (Vehicle Sketch Pad) è una suite open-source sviluppata dalla NASA per la parametrizzazione geometrica di veicoli aerospaziali [17, 28, 34]. Nel presente progetto, OpenVSP è stato impiegato per formalizzare la geometria del tailsitter, consentendo iterazioni rapide di design e l'estrazione diretta delle proprietà inerziali e di massa.

Il processo di modellazione ha seguito la sequenza: definizione della fusoliera come corpo di rivoluzione con sezioni parametriche asimmetriche; costruzione della superficie alare utilizzando il profilo NACA 0012; posizionamento dei bracci motore come superfici a sezione sottile; creazione dei pod propulsivi come corpi di rivoluzione compatti; definizione delle superfici di controllo (elevoni) con i parametri L/C, η_{start} e η_{end} indicati nella Tabella 2.4.

Le viste del modello risultante sono presentate nella Figura 2.2.

2.4.2 Allocazione Masse Interne

In OpenVSP è possibile assegnare la massa a ciascun componente geometrico, ottenendo automaticamente il calcolo del baricentro complessivo e della matrice di inerzia. La distribuzione delle masse interne è stata organizzata associando a ciascun volume della fusoliera un componente con massa e posizione definite. La Tabella 2.9 riporta i componenti interni e le relative dimensioni dei volumi di alloggiamento, modellati come box rettangolari all'interno della fusoliera.
I volumi sono stati dimensionati con tolleranze compatibili con la sezione interna della fusoliera ($0.15\,\text{m} \times 0.15\,\text{m}$), lasciando margine per il passaggio del cablaggio e la ventilazione dei componenti elettronici.



(a) Vista laterale (x–z).

(b) Vista frontale (y–z).

(c) Vista dall'alto (x–y).

(d) Vista isometrica.

Figura 2.2: Viste del modello OpenVSP del tailsitter (isometrica e ortogonali).

Tabella 2.9: Componenti interni e relativi volumi di alloggiamento.

Componente	**Massa (kg)**	**Lungh. (m)**	**Largh. (m)**	**Alt. (m)**
Batteria + payload anteriore	2.30	0.15	0.13	0.07
Elettronica (FC, ESC, sensori)	0.85	0.05	0.13	0.07
Payload posteriore	0.85	0.13	0.13	0.07

2.5 Proprietà di Massa e Inerzia

La distribuzione complessiva della massa risulta dalla somma dei contributi strutturali, propulsivi e di payload, come riportato nella Tabella 2.10.

Tabella 2.10: Distribuzione della massa del veicolo.

Componente	**Massa (kg)**
Fusoliera	0.60
Ala (struttura in carbonio)	0.225
Bracci motore ($4 \times 0.1\,\text{kg}$)	0.40
Pod propulsivi ($4 \times 0.250\,\text{kg}$)	1.00
Batteria + payload anteriore	2.30
Elettronica	0.85
Payload posteriore	0.85
Varie (servi, cablaggio, minuteria)	0.775
Massa totale (MTOW)	**7.00**

La matrice di inerzia nel frame corpo, calcolata in OpenVSP mediante integrazione numerica sulla geometria 3D, assume la forma diagonale grazie alla simmetria laterale e alla distribuzione simmetrica dei motori in configurazione X:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} = \begin{bmatrix} 0.326 & 0 & 0 \\ 0 & 0.500 & 0 \\ 0 & 0 & 0.500 \end{bmatrix} \text{kg} \cdot \text{m}^2 \tag{2.8}$$

Il momento d'inerzia I_{xx} risulta inferiore poiché l'asse x_B (longitudinale) presenta una minore distribuzione di massa rispetto agli assi trasversali.

La posizione del centro di gravità, calcolata rispetto al muso della fusoliera, è:

$$\boldsymbol{r}_{\text{CG}} = \begin{bmatrix} 0.2333 \\ 0 \\ 0 \end{bmatrix} \text{m} \tag{2.9}$$

La ridistribuzione delle masse di payload (anteriore 2.30 kg, posteriore 0.85 kg) sposta il baricentro a monte del centro aerodinamico ($x_{\text{AC}} \approx 0.255\,\text{m}$) di $\Delta x_{\text{CG}} = x_{\text{CG}} - x_{\text{AC}} = -0.0217\,\text{m}$, configurazione che garantisce un margine positivo di stabilità longitudinale statica (cfr. Sezione 2.6 e Capitolo 4, equazione (4.20)).

2.6 Identificazione del Modello Aerodinamico

Il modello aerodinamico del velivolo, che sarà formalizzato dal punto di vista implementativo nel Capitolo 4, è costituito da due componenti con metodologie di

identificazione distinte: i coefficienti longitudinali dell'ala ($C_{L,\text{wing}}$, $C_{D,\text{wing}}$, $C_{m,\text{wing}}$), funzione non lineare dell'angolo di attacco α, e le restanti derivate di stabilità (latero-direzionali, di smorzamento, di efficacia delle superfici di controllo, di resistenza parassita non-alare), assunte costanti su tutto l'inviluppo di volo. La distinzione metodologica riflette il diverso grado di non linearità dei due insiemi di coefficienti: i tre coefficienti longitudinali dell'ala variano in modo fortemente non lineare con α, attraversando lo stallo a $\alpha \approx \pm 12°$ e proseguendo nel *post-stallo*; le altre derivate, viceversa, sono prossimamente lineari nell'intervallo di assetti che le sollecita e possono essere ragionevolmente fissate a un valore nominale di trim.

2.6.1 Coefficienti Longitudinali dell'Ala: Reti Neurali su Dati Sperimentali NACA 0012

Il profilo NACA 0012 è uno dei profili più estesamente caratterizzati in letteratura [1, 9, 39]. Per la sua simmetria e per l'ampiezza del database disponibile, costituisce una scelta naturale per il tailsitter: il velivolo attraversa infatti un intervallo di angoli di attacco di $\pm 90°$ durante le transizioni, e nessun modello *thin-airfoil* o lineare risulta valido su un intervallo di tale ampiezza.

I coefficienti $C_{L,\text{wing}}(\alpha)$, $C_{D,\text{wing}}(\alpha)$, $C_{m,\text{wing}}(\alpha)$ sono identificati mediante reti neurali *shallow* (singolo strato nascosto) con funzione di attivazione `tansig` [16, 19]:

$$C_{(\cdot),\text{wing}}(\alpha) = \boldsymbol{\theta}^{\top} \begin{bmatrix} 1 \\ \tanh\big(\mathbf{V}\,\alpha + \mathbf{c}\big) \end{bmatrix} \tag{2.10}$$

dove $\boldsymbol{\theta}$ raccoglie i pesi dello strato di uscita (incluso il bias), $\mathbf{V}$ i pesi dello strato nascosto e $\mathbf{c}$ i bias dei neuroni nascosti. La derivata $\partial C/\partial\alpha$ è necessaria per il solver di trim di Newton–Raphson nel generatore del comando di assetto (Capitolo 7), è disponibile in forma chiusa come

$$\frac{\partial C_{(\cdot),\text{wing}}}{\partial \alpha}(\alpha) = \boldsymbol{\theta}_{2:\text{end}}^{\top} \Big[\Big(1 - \tanh^2(\mathbf{V}\,\alpha + \mathbf{c})\Big) \odot \mathbf{V} \Big], \tag{2.11}$$

con $\odot$ prodotto elemento per elemento. L'esistenza di una derivata analitica, e non semplicemente di una interpolazione *look-up*, costituisce una proprietà essenziale per la convergenza locale di ordine due del solver di trim.

I tre set di pesi addestrati sono memorizzati in tre file (`Lift.mat`, `Drag.mat`, `Pitch.mat`) e caricati al momento dell'inizializzazione del modello, come descritto nel Capitolo 3.

2.6.2 Coefficienti Costanti Estratti da VSPAero

Le restanti grandezze aerodinamiche del modello sono calcolate una sola volta in fase di progettazione mediante VSPAero [26], il modulo di analisi a pannelli di

OpenVSP, e successivamente *congelate* come parametri costanti per tutta la durata della simulazione. Si distinguono tre famiglie:

1. **Drag parassita totale del velivolo:** $C_{D,0,\text{tot}} = 0.15$, valore stimato sul modello completo (fusoliera, bracci, pod, ala) in condizioni di crociera. Da questo si sottrae il valore restituito dalla rete neurale a $\alpha = 0$ per ottenere il drag parassita dei *soli* componenti non-alari, $C_{D,0,\text{nw}} = C_{D,0,\text{tot}} - C_{D,\text{wing}}(0)$, da sommare al drag dell'ala (Sezione 4.5.3).

2. **Derivate di efficacia degli elevoni:** C_{L,δ_e}, C_{m,δ_e} (deflessione simmetrica), C_{l,δ_a}, C_{n,δ_a} (deflessione antisimmetrica).

3. **Derivate latero-direzionali e di smorzamento:** $C_{Y,\beta}$, $C_{l,\beta}$, $C_{n,\beta}$, $C_{l,p}$, $C_{m,q}$, $C_{n,r}$, $C_{l,r}$, $C_{n,p}$.

I valori numerici sono raccolti nella Tabella 4.1 del Capitolo 4, dove se ne discute il significato fisico e la convenzione di segno.

Sintesi del modello aerodinamico complessivo. La conseguente architettura del modello aerodinamico può essere riassunta come segue. La portanza, il drag e il momento di beccheggio sono ottenuti sommando il contributo non lineare dell'ala (predetto dalle reti neurali) e il contributo lineare delle deflessioni degli elevoni, delle velocità angolari adimensionali e dell'offset del baricentro rispetto al centro aerodinamico. La forza laterale, il momento di rollio e il momento di imbardata sono invece interamente lineari in β, $\hat{p}$, $\hat{r}$ e δ_a. Il modello cattura quindi le non linearità essenziali del tailsitter (stallo, regime post-stallo).

2.7 Sommario delle Specifiche

La Tabella 2.11 raccoglie le specifiche di progetto complessive del veicolo.

Questa configurazione rappresenta il compromesso progettuale tra autonomia, velocità di crociera, capacità di hover e semplicità costruttiva, coerente con gli obiettivi della ricerca su architetture di controllo unificato per tailsitter.

Tabella 2.11: Sommario delle specifiche di design.

Categoria	**Parametro**	**Valore**
Masse	MTOW	$7\,\mathrm{kg}$
	Centro di gravità (x)	$0.2333\,\mathrm{m}$
	Carico alare	$68.5\,\mathrm{N\,m^{-2}}$
Ala	Superficie	$0.1022\,\mathrm{m^2}$
	Apertura	$0.73\,\mathrm{m}$
	Profilo	NACA 0012
	AR	5.21
	Posizione LE	$0.22\,\mathrm{m}$
Propulsione	Motori	$4\times$ T-Motor MN5008
	$\Omega_{\max}$	$679\,\mathrm{rad\,s^{-1}}$
	$T_{\max}$ (per motore)	$47.1\,\mathrm{N}$
	k_T	$1.022\times10^{-4}\,\mathrm{N/(rad/s)^2}$
Controllo	Elevoni (L/C)	0.30
	$\delta_{\max}$	$\pm30^\circ$
Inerzia	I_{xx}	$0.326\,\mathrm{kg\,m^2}$
	I_{yy}	$0.500\,\mathrm{kg\,m^2}$
	I_{zz}	$0.500\,\mathrm{kg\,m^2}$

Capitolo 3

Inizializzazione del Modello: Script MATLAB di Configurazione

3.1 Introduzione

Il modello Simulink descritto nei capitoli successivi si appoggia, per la totalità delle costanti fisiche, dei coefficienti aerodinamici, dei guadagni del controllore e dei parametri di missione, su una struttura dati condivisa, denominata `P`, popolata da una serie di script MATLAB eseguiti in cascata prima dell'avvio della simulazione [27, 43]. La scelta di un modello *data-driven*, in cui tutti i parametri risiedono nel *workspace* di MATLAB e non sono *hard-coded* all'interno dei blocchi Simulink, è una pratica raccomandata in ambiente aerospaziale [10]: garantisce tracciabilità delle modifiche, separazione fra modello (Simulink) e configurazione (MATLAB) e riproducibilità delle simulazioni.

Il presente capitolo descrive la struttura degli script di inizializzazione, seguendo l'ordine logico con cui vengono eseguiti: la configurazione ambientale (gravità, densità dell'aria, passo di integrazione), le tolleranze numeriche, i parametri del *plant* (massa, inerzia, propulsione, aerodinamica), i parametri di missione (velocità di crociera, quote, tempi), i guadagni del controllore di posizione e di assetto, e infine le specifiche di traiettoria della missione operativa.

3.2 Architettura Generale degli Script

L'inizializzazione è organizzata in un *master script* di *setup* (non riprodotto qui per brevità) che invoca, nell'ordine elencato in Tabella 3.1, le funzioni di configurazione.

Ognuna restituisce una sotto-struttura che viene poi fusa nella struttura complessiva `P`, accessibile globalmente al modello Simulink come *Parameter Data*.

Tabella 3.1: Script di inizializzazione e relativa funzione.

Ordine	Funzione	Ruolo
1	`env_params`	Costanti ambientali e di solver
2	`tolerances`	Soglie numeriche per il modello
3	`plant_params`	Parametri fisici del veicolo
4	`mission_params`	Parametri di missione
5	`controller_gains`	Guadagni del sistema di controllo
6	`nav_mode_8_delivery_mission`	Specifica della missione operativa

Il flusso è interamente sequenziale e privo di dipendenze incrociate fra script: ognuno dipende soltanto dagli output di quelli che lo precedono. In particolare, `controller_gains` non legge alcun parametro di `plant_params`, perché i guadagni sono stati progettati con un approccio basato su feed-forward inerziale (Capitolo 6) che disaccoppia la sintesi del controllore dai dettagli del modello del veicolo.

3.3 Parametri Ambientali e di Solver

La funzione `env_params` restituisce le costanti ambientali e quelle che governano l'integrazione numerica:

$$g = 9.81\,\mathrm{m\,s^{-2}}, \tag{3.1}$$

$$\rho = 1.112\,\mathrm{kg\,m^{-3}} \quad \text{(densità di riferimento)}, \tag{3.2}$$

$$dt_{\mathrm{ctrl}} = 0.001\,\mathrm{s}. \tag{3.3}$$

La densità dell'aria è considerata costante: il velivolo opera in un intervallo di quota troppo ristretto (0–300 m AGL nello scenario operativo di riferimento) perché una modellazione di atmosfera ISA produca variazioni significative dei coefficienti aerodinamici. Il valore adottato $\rho = 1.112\,\mathrm{kg\,m^{-3}}$ corrisponde all'atmosfera standard ISA a circa 1000 m AMSL ed è leggermente inferiore al valore ISA a quota operativa nominale ($\approx 1.200\,\mathrm{kg\,m^{-3}}$ a 200 m AGL). La scelta è motivata dalla variabilità reale delle condizioni meteorologiche: in giornate di bassa pressione barometrica, anche a quote operative basse, la densità effettiva può scendere sotto il valore ISA standard e avvicinarsi a quella di quote più elevate. Adottare $\rho = 1.112\,\mathrm{kg\,m^{-3}}$ rende dunque la simulazione rappresentativa di un *caso conservativo*, che copre un inviluppo realistico di condizioni atmosferiche in cui il veicolo può operare a bassa quota. Il passo del controllore dt_{ctrl} coincide con il passo del solutore Simulink, configurato a `ode4` (Runge–Kutta del quarto ordine a passo fisso).

3.4 Tolleranze Numeriche

La funzione `tolerances` raccoglie le soglie numeriche utilizzate dal modello per gestire i regimi degeneri (norme prossime a zero, velocità inferiori a una soglia minima di riferimento, ecc.). Pur trattandosi di costanti minute, la loro corretta calibrazione è essenziale per la robustezza del modello: una soglia troppo grande introduce *deadband* indesiderate, una troppo piccola amplifica il rumore numerico in prossimità della singolarità.

Tabella 3.2: Tolleranze numeriche utilizzate nel modello.

Tolleranza	**Valore**	**Significato**
$D_{\min,\text{cruise}}$	10^{-6} m	Distanza minima per estrarre il versore di crociera
$D_{\min,\text{fw}}$	10^{-6} m	Idem nella fase di *forward transition*
$dz_{\min}$	10^{-3} m	Salita/discesa trascurabile (skip della rampa septica di quota)
$d_{\text{arrived,xy}}$	0.1 m	Soglia di arrivo planare per il passaggio a *hover pre-land*
$v_{\min,xy}$	10^{-3} m/s	Modulo planare minimo per definire il versore tangente in virata
$v_{\min,\psi}$	0.3 m/s	Soglia di velocità sotto la quale lo yaw desiderato non viene ricalcolato
$\|\cdot\|_{\min,\text{basis}}$	10^{-8}	Soglia per la ri-ortonormalizzazione delle terne body
$\|\cdot\|_{\min,\text{quat}}$	10^{-8}	Soglia per la rinormalizzazione del quaternione
$F_{\text{th,min}}$	0.5 N	Sotto cui $\boldsymbol{F}_{\text{tot},I}$ è considerato degenere per estrarre $\hat{b}_x$
$V_{\text{dyn,min}}$	5 m/s	Soglia di pressione dinamica del solver di trim
$V_{\text{real,takeover}}$	5 m/s	Soglia sopra cui il modulo reale sostituisce V_{ref}

3.5 Parametri del *Plant*

La funzione `plant_params` popola la struttura con tutte le costanti fisiche e aerodinamiche del veicolo descritte nel Capitolo 2, e con i pesi delle reti neurali aerodinamiche identificati sui dati sperimentali del NACA 0012.

3.5.1 Massa, Inerzia e Propulsione

Le grandezze inerziali sono caricate come scalari e poi assemblate nella matrice di inerzia diagonale:

$$\mathbf{I} = \text{diag}(I_{xx},\, I_{yy},\, I_{zz}) = \text{diag}(0.326,\, 0.500,\, 0.500)\ \text{kg}\,\text{m}^2, \tag{3.4}$$

con $\mathbf{I}^{-1}$ pre-calcolata e memorizzata per ridurre il costo computazionale all'interno di ogni passo del solutore. I parametri di propulsione k_T, k_Q sono derivati dal datasheet del motore T-Motor MN5008 KV340 e dalle caratteristiche dell'elica P17×5.8" tripala (Capitolo 2). La matrice di mixing $\mathbf{M}_{\text{mix}}$ in configurazione X è assemblata in modo esplicito a partire da k_T, k_Q e dal braccio $d = 0.325\,\text{m}$, e la sua inversa $\mathbf{M}_{\text{mix}}^{-1}$ è anch'essa pre-calcolata per evitare inversioni a tempo di esecuzione nel sottosistema di allocazione dei motori (Capitolo 5).

3.5.2 Caricamento delle Reti Neurali Aerodinamiche

Tre file `.mat`, prodotti durante la fase di addestramento delle reti su dati sperimentali NACA 0012 (Sezione 2.6), sono caricati nel workspace:

- `Lift.mat` — pesi $\boldsymbol{\theta}_L$, $\mathbf{V}_L$, $\mathbf{c}_L$ della rete che predice $C_{L,\text{wing}}(\alpha)$.
- `Drag.mat` — pesi $\boldsymbol{\theta}_D$, $\mathbf{V}_D$, $\mathbf{c}_D$ della rete che predice $C_{D,\text{wing}}(\alpha)$.
- `Pitch.mat` — pesi $\boldsymbol{\theta}_m$, $\mathbf{V}_m$, $\mathbf{c}_m$ della rete che predice $C_{m,\text{wing}}(\alpha)$.

I pesi sono successivamente copiati nei campi corrispondenti della struttura `P` e diventano così accessibili dalle MATLAB Function del blocco Aerodynamics nel modello Simulink. Il drag parassita dei componenti non-alari $C_{D,0,\text{nw}}$ è calcolato in fase di inizializzazione come differenza tra il drag parassita totale stimato in VSPAero ($C_{D,0,\text{tot}} = 0.15$) e il valore restituito dalla rete neurale a $\alpha = 0$:

$$C_{D,0,\text{nw}} = C_{D,0,\text{tot}} - C_{D,\text{wing}}(0), \tag{3.5}$$

in modo da evitare il doppio conteggio del contributo alare.

3.5.3 Derivate di Stabilità e Modello di Contatto

Le restanti derivate di stabilità (Tabella 4.1), l'efficacia degli elevoni, l'angolo massimo di deflessione $\delta_{\max} = 30°$ e i parametri del modello di contatto con il suolo ($k_G = 50000\,\text{N m}^{-1}$, $b_G = 2000\,\text{N s m}^{-1}$, $\mu = 0.5$) sono assegnati come campi scalari della struttura `P`. Il loro significato fisico è discusso nel Capitolo 4 in sede di descrizione del modello aerodinamico e del contatto.

3.6 Parametri di Missione

La funzione `mission_params` raccoglie i parametri trasversali della missione operativa: velocità di riferimento alla fine della back-transition ($V_{\text{hover-ready}} = 8\,\text{m s}^{-1}$), saturazioni di accelerazione planare e verticale, decelerazione massima nel profilo di decelerazione unificato ($a_{\text{max,decel}} = 0.5\,\text{m s}^{-2}$, con margine $\alpha_{\text{margin}} = 1.3$) e raggio di arrivo ($r_{\text{arrived}} = 2\,\text{m}$).

3.6.1 Blending Hover/Cruise

Il parametro $V_{\text{trans}} = 32\,\text{m}\,\text{s}^{-1}$ e il guadagno $k_\sigma = 0.3$ definiscono la sigmoide di pesatura

$$\sigma(V_{\text{air}}) = \frac{1}{1 + \exp\Big(-k_\sigma\,(V_{\text{air}} - V_{\text{trans}})\Big)}, \tag{3.6}$$

utilizzata dal generatore del comando di assetto per interpolare con *SLERP* fra l'assetto di hover e quello di crociera (Capitolo 7).

3.6.2 Buffer di Approach

Il parametro $d_{\text{approach,buffer}} = 30\,\text{m}$ definisce la distanza geometrica della fase di approach: la rampa septica di velocità $V_{\text{he}} \to 0$ si svolge sull'ultimo segmento di 30 m prima del target, garantendo un raccordo $\mathcal{C}^1$ fra la decelerazione di rotta e l'arrivo in hover (Capitolo 9).

3.7 Guadagni del Sistema di Controllo

La funzione `controller_gains` popola i guadagni della cascata di controllo, suddivisi per sottosistema:

3.7.1 Position Controller

$$\mathbf{K}_{p,\text{pos}} = \text{diag}(1.2,\ 1.2,\ 3.5), \qquad \mathbf{K}_{d,\text{pos}} = \text{diag}(1.3,\ 1.3,\ 4.5). \tag{3.7}$$

L'asimmetria tra gli assi orizzontali (N, E) e l'asse verticale (D) riflette la diversa banda passante richiesta sui due canali, come discusso nel Capitolo 8. Le saturazioni di accelerazione sono fissate ad $a^{\pm}_{\text{max},xy} = 15\,\text{m}\,\text{s}^{-2}$ e $a^{\pm}_{\text{max},z} = 13\,\text{m}\,\text{s}^{-2}$.

3.7.2 Attitude Controller (cascata esterna)

I guadagni proporzionali sull'errore di assetto sono asimmetrici:

$$\mathbf{K}_{p,\text{att}} = \text{diag}(5.0,\ 5.0,\ 3.0), \tag{3.8}$$

con il valore inferiore sul canale di imbardata (z_B).

3.7.3 Rate Loop (cascata interna)

$$\mathbf{K}_{p,\text{rate}} = \text{diag}(10,\ 4,\ 12), \qquad \mathbf{K}_{i,\text{rate}} = \text{diag}(5,\ 1,\ 6). \tag{3.9}$$

La costante di tempo del filtro sul feed-forward inerziale è $\tau_{\text{rate,cl}} = 0.05\,\text{s}$. Le saturazioni di coppia sono $\boldsymbol{\tau}_{\max} = (11.2,\ 11.2,\ 5)\ \text{N}\,\text{m}$, e l'*anti-windup* è configurato in modalità condizionale, come descritto in dettaglio nel Capitolo 6.

3.8 Specifica della Missione Operativa

La funzione `nav_mode_8_delivery_mission` restituisce due strutture: `N`, che raccoglie i parametri di alto livello della missione, e `IC`, che contiene le condizioni iniziali del veicolo. `N` viene anche utilizzata per costruire la struttura `mission_spec`, che è l'oggetto effettivamente consumato dalla macchina a stati di guidance (Capitolo 9).

3.8.1 Topologia della Missione

La missione operativa di riferimento è una missione a tre *leg* di recupero e rilascio di un dispositivo medico, con *touchdown* ai siti intermedi, descritta nella Tabella 3.3.

Tabella 3.3: Topologia della missione di riferimento.

Leg	Descrizione
Leg 1	Base $\rightarrow$ Sito A. Decollo, cruise a $h_1 = 100\,\mathrm{m}$, virata 30° destra (W2), salita a $h_2 = 200\,\mathrm{m}$, virata 60° destra (W3), cruise, decelerazione, approach, atterraggio al Sito A e recupero del dispositivo medico.
Leg 2	Sito A $\rightarrow$ Sito B. Decollo, cruise a h_2, virata 90° sinistra (W5), discesa a $h_3 = 150\,\mathrm{m}$, virata 90° sinistra (W6), cruise, decelerazione, approach, atterraggio al Sito B e rilascio del dispositivo medico.
Leg 3	Sito B $\rightarrow$ Base. Decollo, cruise a h_3, virata $\sim 163.5°$ sinistra (W8), discesa a h_1, cruise, decelerazione, approach, atterraggio alla base e conclusione della missione.

I waypoint di pre-virata e le quote sono progettati in modo che il velivolo mantenga le velocità nominali ($V_{\mathrm{cruise}} = 61\,\mathrm{m\,s^{-1}}$, $V_{\mathrm{climb,takeoff}} = 5\,\mathrm{m\,s^{-1}}$, $V_{\mathrm{descent,landing}} = 3\,\mathrm{m\,s^{-1}}$) entro l'inviluppo di accelerazione configurato. Le tempistiche di hover (iniziale, pre-landing, operativo a terra) sono fissate rispettivamente a 4, 2 e 10 secondi.

3.8.2 Costruzione della `mission_spec`

La struttura `mission_spec` è una rappresentazione esplicita della missione adatta al consumo da parte della macchina a stati. È organizzata come array di *leg*, ciascuna contenente la posizione di partenza e quella di arrivo, la quota di crociera iniziale, l'array di waypoint (coordinate NED), l'array di virate (angolo e direzione, $+1$ destra, -1 sinistra) e la quota finale. Gli array di waypoint e di virate hanno dimensione costante (2×3 e 2×2 rispettivamente).

3.8.3 Condizioni Iniziali

Il veicolo è inizializzato in assetto di hover alla base, con velocità e velocità angolari nulle:

$$\boldsymbol{p}_0 = (0,0,0)^T, \quad \boldsymbol{v}_{B,0} = \mathbf{0}, \quad \boldsymbol{\omega}_0 = \mathbf{0}, \quad \mathbf{q}_0 = \left(\cos\tfrac{\pi}{4},\, 0,\, \sin\tfrac{\pi}{4},\, 0\right)^T, \tag{3.10}$$

dove $\mathbf{q}_0$ corrisponde a una rotazione di $+90°$ attorno all'asse y_B (asse x_B verticale, ovvero *nose-up*), che è la postura naturale del tailsitter al decollo da terra.

3.9 Sintesi e Vantaggi dell'architettura

Il modello presentato nei capitoli successivi è interamente parametrizzato sulla struttura `P` prodotta dagli script descritti in questo capitolo. La separazione fra modello e configurazione consente di alterare qualunque caratteristica del veicolo (massa, inerzia, coefficienti aerodinamici, parametri di propulsione, guadagni del controllore, specifica di missione) senza modificare il modello Simulink, e si rivela particolarmente preziosa durante il *tuning* dei controllori e nella preparazione di scenari di test parametrici, in cui una singola simulazione può differire da quella di riferimento per un solo campo della struttura.

Capitolo 4

Modello Dinamico a Sei Gradi di Libertà

4.1 Introduzione

Il modello dinamico del velivolo costituisce il nucleo dell'intero sistema di simulazione e controllo: dalla sua accuratezza dipendono la fedeltà delle simulazioni, la qualità delle leggi di controllo sintetizzate e la validità dei risultati della tesi. Il presente capitolo sviluppa il modello completo a sei gradi di libertà (6-DOF) del tailsitter, dalla sua architettura implementativa in Simulink fino alla formulazione matematica di ciascun sottosistema, basandosi sui formalismi classici della meccanica del volo non lineare [6, 8, 12, 23, 32, 35, 38, 46, 47].

La trattazione segue un approccio *top-down*: si presenta dapprima la struttura complessiva del modello Simulink, identificando i blocchi funzionali e i flussi di segnale che li collegano, per poi analizzare in dettaglio la formulazione matematica implementata in ciascun blocco. Ogni funzione MATLAB che realizza un sottosistema è accompagnata dalla derivazione analitica corrispondente, consentendo di verificare la coerenza tra modello matematico e implementazione.

4.2 Convenzioni di Riferimento e Vettore di Stato

4.2.1 Sistemi di Coordinate

La modellazione richiede due sistemi di riferimento [12, 23, 47]:

1. **Riferimento inerziale NED** ($\mathcal{F}_I$): sistema cartesiano destrorso fisso a terra, con assi orientati verso Nord, Est e verso il basso (*Down*). L'origine coincide con il punto di decollo.

2. **Riferimento del corpo** ($\mathcal{F}_B$): sistema cartesiano destrorso solidale al veicolo, con origine nel baricentro, asse x_B diretto lungo la direzione di volo primaria (*forward*), asse y_B lungo l'ala destra e asse z_B verso il basso.

La trasformazione dal riferimento del corpo a quello inerziale è rappresentata dalla matrice di rotazione $\mathbf{R}(\mathbf{q})$, funzione dell'assetto del veicolo espresso in quaternioni.

4.2.2 Rappresentazione dell'Assetto mediante Quaternioni

L'assetto è parametrizzato mediante quaternioni unitari [11, 21, 45]:

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \in \mathbb{R}^4, \quad \|\mathbf{q}\| = 1 \tag{4.1}$$

dove q_0 è la componente scalare e $\boldsymbol{q}_v = [q_1,\, q_2,\, q_3]^T$ è la parte vettoriale. I quaternioni sono scelti in luogo degli angoli di Eulero per evitare la singolarità di *gimbal lock* a $\theta = \pm 90°$, condizione attraversata ad ogni transizione hover–crociera del tailsitter.

La matrice di rotazione da $\mathcal{F}_B$ a $\mathcal{F}_I$ corrispondente al quaternione $\mathbf{q}$ è:

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix} \tag{4.2}$$

Questa espressione sfrutta il vincolo $\|\mathbf{q}\| = 1$ per semplificare i termini diagonali. La trasformazione inversa (da $\mathcal{F}_I$ a $\mathcal{F}_B$) è data dalla trasposta $\mathbf{R}^T$, poiché $\mathbf{R} \in SO(3)$.

4.2.3 Vettore di Stato

Il vettore di stato completo del sistema è costituito da 13 variabili:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v}_B \\ \mathbf{q} \\ \boldsymbol{\omega}_B \end{bmatrix} \in \mathbb{R}^{13} \tag{4.3}$$

dove $\boldsymbol{p} = [p_N,\, p_E,\, p_D]^T$ è la posizione nel riferimento NED, $\boldsymbol{v}_B = [u,\, v,\, w]^T$ è la velocità lineare nel riferimento del corpo, $\mathbf{q}$ è il quaternione di assetto e $\boldsymbol{\omega}_B = [p,\, q,\, r]^T$ è la velocità angolare nel riferimento del corpo. Le variabili sono soggette al vincolo $\|\mathbf{q}\| = 1$, che riduce i gradi di libertà effettivi a 12.

4.3 Architettura Generale del Modello Simulink

Prima di entrare nel dettaglio del modello dinamico, è opportuno introdurre la struttura complessiva dell'implementazione Simulink, che sarà ripresa e approfondita progressivamente nei capitoli seguenti. Il modello è implementato secondo un'architettura gerarchica modulare in cui ciascun sottosistema rappresenta una funzione logica autosufficiente del sistema di controllo e simulazione. La Figura 4.1 mostra la vista complessiva del modello al livello superiore.

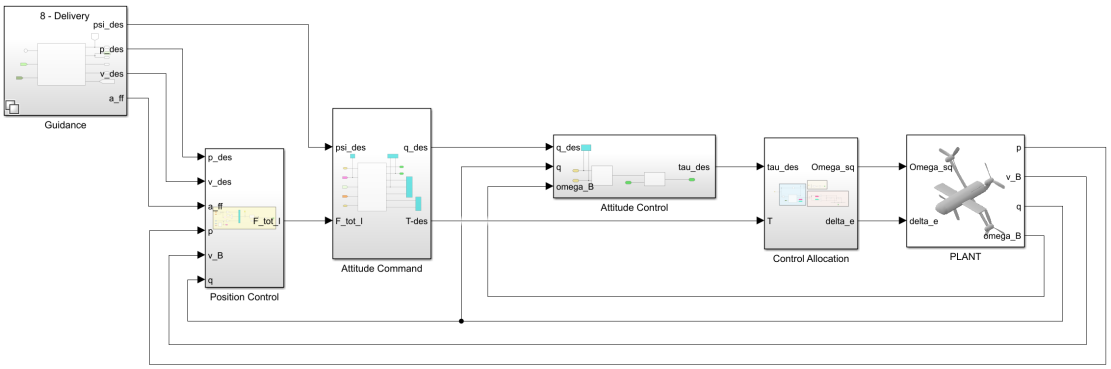


Figura 4.1: Vista complessiva del modello Simulink del tailsitter: dal modello di *Guidance* (sinistra) al *plant* (destra), passando per i cinque livelli del sistema di controllo.

Procedendo da sinistra a destra nello schema si identificano sei sottosistemi principali, in cascata:

1. **Guidance** (Capitolo 9). Macchina a stati di pianificazione di missione, che produce i riferimenti di traiettoria ($\boldsymbol{p}_{\text{des}}$, $\boldsymbol{v}_{\text{des}}$, $\boldsymbol{a}_{\text{ff}}$, ψ_{des}) in funzione del tempo e dello stato corrente.

2. **Position Control** (Capitolo 8). Anello PD con *feed-forward* che traduce l'errore di posizione e velocità nel vettore di forza di spinta desiderato $\boldsymbol{F}_{\text{thrust},I}$.

3. **Attitude Command** (Capitolo 7). Generatore del riferimento di assetto: data $\boldsymbol{F}_{\text{thrust},I}$, produce il quaternione desiderato $\mathbf{q}_{\text{des}}$ e il modulo di spinta T_{des}, secondo la legge di volo coordinato di Ritz e D'Andrea.

4. **Attitude Control** (Capitolo 6). Controllore di assetto a cascata che, dato l'errore di assetto in quaternioni, produce il vettore di coppia desiderata $\boldsymbol{\tau}_{\text{des}}$.

5. **Control Allocation** (Capitolo 5). Allocatore adattivo che traduce (T_{des}, $\boldsymbol{\tau}_{\text{des}}$) nei comandi fisici degli attuatori: velocità angolari al quadrato dei motori $\boldsymbol{\Omega}^2$ e deflessioni degli elevoni δ_e.

6. **Plant** (oggetto del presente capitolo). Modello dinamico 6-DOF del veicolo, che riceve $\boldsymbol{\Omega}^2$ e δ_e e restituisce il vettore di stato $\boldsymbol{x} = (\boldsymbol{p},\, \boldsymbol{v}_B,\, \mathbf{q},\, \boldsymbol{\omega}_B)$.

L'ordine espositivo adottato in questa tesi è inverso rispetto al flusso causale del sistema di controllo: si parte dal nucleo, il *plant*, e si procede verso l'esterno, attraverso gli anelli di controllo, fino al pianificatore di missione. La scelta rispecchia il processo costruttivo del simulatore, sviluppato incrementalmente dall'interno verso l'esterno: ogni strato è stato implementato e validato sopra quelli già operativi, e la stessa logica si presta naturalmente all'esposizione, perché consente a ciascun capitolo di appoggiarsi sui modelli e sulle convenzioni introdotti in quello precedente senza ricorrere a riferimenti in avanti.

Avvertenza sulle interfacce dei sottosistemi. Le porte di ingresso e uscita visibili sui blocchi al livello superiore (Figura 4.1) non coincidono con il numero complessivo dei segnali effettivamente consumati o prodotti da ciascun sottosistema: rappresentano solo i flussi principali, quelli che si è ritenuto opportuno mantenere espliciti per chiarezza visiva. Diversi segnali ausiliari sono propagati *internamente* attraverso coppie di blocchi `Goto`/`From` distribuite nel modello. Si tratta di una scelta puramente di *layout* per ridurre la densità del cablaggio, senza alcun effetto sulla struttura logica del sistema.

Solver e passo di integrazione. Il modello utilizza un solutore Runge–Kutta a passo fisso del quarto ordine (`ode4`) con passo di integrazione $dt = 0.001\,\mathrm{s}$. La scelta del passo fisso è motivata dalla volontà di mantenere la simulazione deterministica e riproducibile, condizione necessaria per il debug incrementale delle leggi di controllo e per la pianificazione di una generazione di codice embedded (*Simulink Coder* [42]) preservando il medesimo regime temporale del solutore.

4.4 Il Blocco *Plant*

Il blocco *Plant* riceve in ingresso le velocità angolari elevate al quadrato dei quattro motori $(\Omega_1, \ldots, \Omega_4)$ e le deflessioni dei due elevoni $(\delta_{eL},\, \delta_{eR})$, e produce in uscita il vettore di stato completo. La sua interfaccia esterna è mostrata in Figura 4.2.

All'interno del blocco Plant (Figura 4.3), la struttura è articolata in due sottosistemi principali:

1. **Forces & Moments:** calcola tutte le forzanti esterne (gravità, propulsione, aerodinamica, contatto con il terreno) a partire dallo stato corrente e dai comandi degli attuatori.

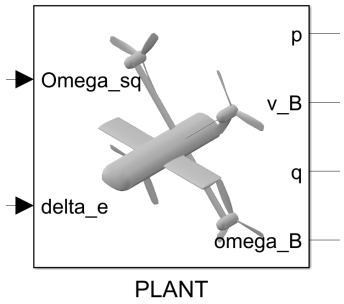


Figura 4.2: Interfaccia del blocco Plant: ingressi di comando e uscite di stato.

2. **Equations of Motion (EOM):** integra le equazioni differenziali della dinamica traslazionale e rotazionale e della cinematica dell'assetto e della posizione.

Questa separazione rispecchia la struttura causale del problema: le forzanti dipendono dallo stato corrente e dai comandi, mentre le equazioni del moto propagano lo stato nel tempo.

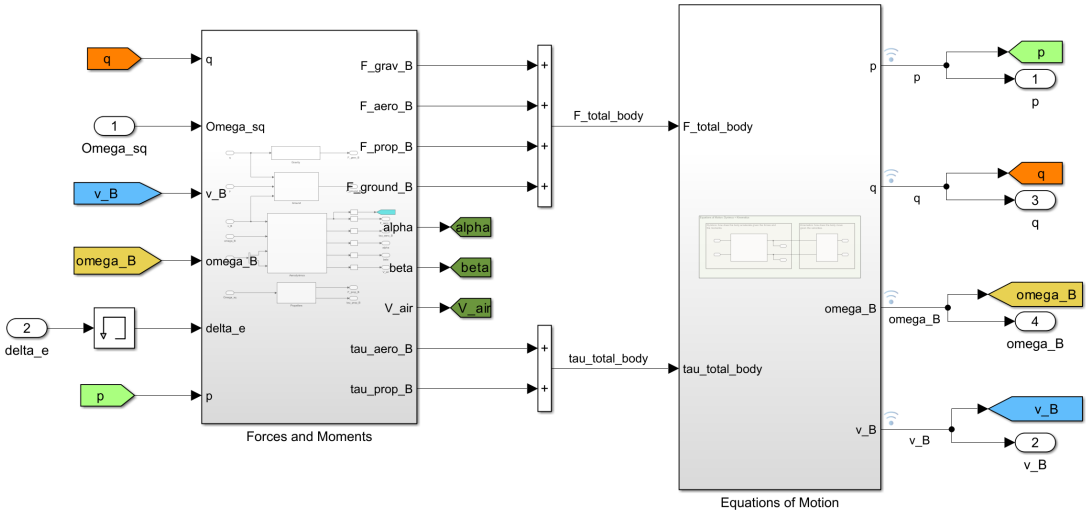


Figura 4.3: Struttura interna del blocco Plant: sottosistemi Forces & Moments e Equations of Motion.

Le sezioni seguenti descrivono in dettaglio la formulazione matematica di ciascun blocco, nell'ordine in cui i segnali vengono elaborati durante un singolo passo di simulazione.

4.5 Sottosistema Forces & Moments

Il sottosistema che viene presentato in Figura 4.4 aggrega quattro contributi distinti, ciascuno implementato in una funzione MATLAB dedicata. Le forze e i momenti risultanti sono sommati per ottenere i vettori totali nel riferimento del corpo:

$$\boldsymbol{F}_{\text{tot},B} = \boldsymbol{F}_{\text{grav},B} + \boldsymbol{F}_{\text{prop},B} + \boldsymbol{F}_{\text{aero},B} + \boldsymbol{F}_{\text{ground},B} \tag{4.4}$$

$$\boldsymbol{\tau}_{\text{tot},B} = \boldsymbol{\tau}_{\text{prop},B} + \boldsymbol{\tau}_{\text{aero},B} \tag{4.5}$$

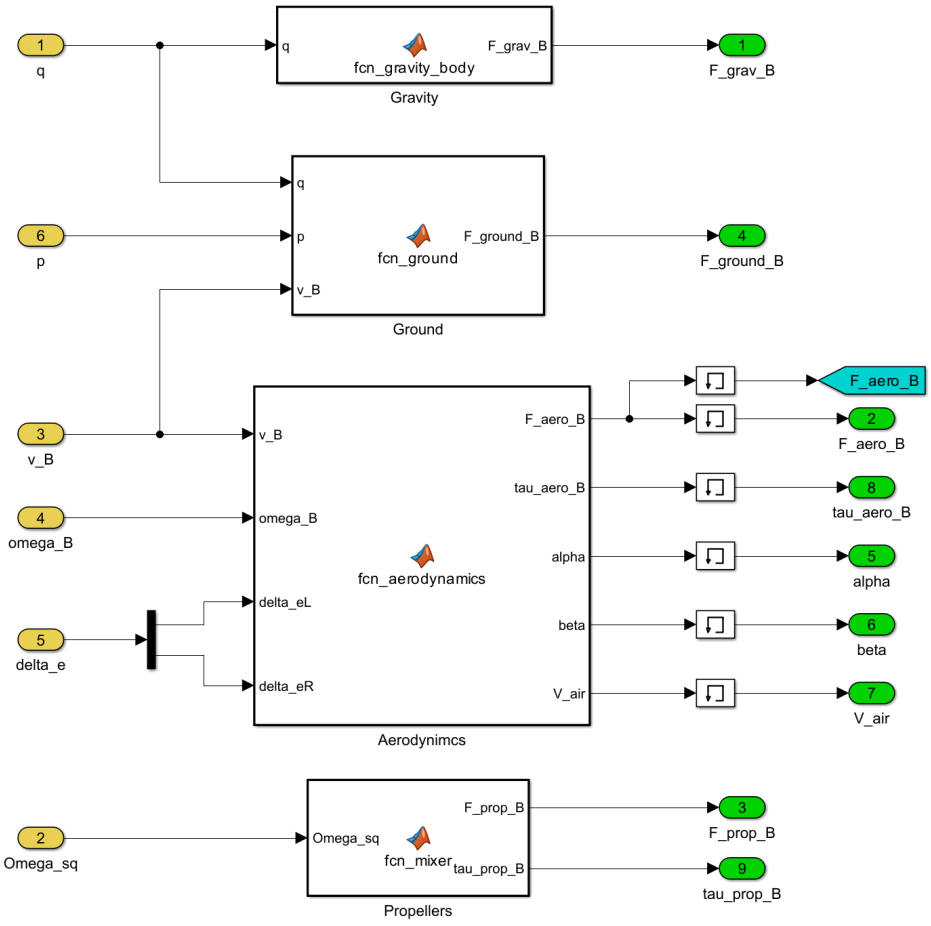


Figura 4.4: Sottosistema Forces & Moments.

4.5.1 Gravità (`fcn_gravity_body`)

La forza gravitazionale nel riferimento inerziale NED è diretta lungo l'asse z_I (verso il basso):

$$\boldsymbol{F}_{\text{grav},I} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{4.6}$$

dove $m = 7\,\text{kg}$ e $g = 9.81\,\text{m}\,\text{s}^{-2}$. Per esprimerla nel riferimento del corpo si applica la rotazione inversa:

$$\boldsymbol{F}_{\text{grav},B} = \mathbf{R}^T(\mathbf{q})\,\boldsymbol{F}_{\text{grav},I} \tag{4.7}$$

Poiché $\boldsymbol{F}_{\text{grav},I}$ ha componenti non nulle solo lungo z_I, il prodotto si riduce all'estrazione della terza colonna di $\mathbf{R}^T$, ovvero della terza riga di $\mathbf{R}$. Indicando con R_{3j} gli elementi della terza riga della matrice di rotazione (4.2):

$$\boldsymbol{F}_{\text{grav},B} = mg \begin{bmatrix} R_{31} \\ R_{32} \\ R_{33} \end{bmatrix} = mg \begin{bmatrix} 2(q_1 q_3 - q_0 q_2) \\ 2(q_2 q_3 + q_0 q_1) \\ 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \tag{4.8}$$

Come verifica, è possibile analizzare la forza di gravità percepita nel riferimento del corpo per due condizioni di volo specifiche:

- **Assetto di crociera:** considerando $\mathbf{q} = [1,\, 0,\, 0,\, 0]^T$ (identità, con asse x_B orizzontale e z_B verso il basso), si ottiene $\boldsymbol{F}_{\text{grav},B} = [0,\, 0,\, mg]^T = [0,\, 0,\, 68.67]^T$ N. Questo risultato è coerente con il peso diretto verso il basso lungo $+z_B$.
- **Assetto di hover:** imponendo un angolo di beccheggio $\theta = 90°$ (ovvero con il *nose* verso l'alto), il quaternione risulta $\mathbf{q} = [\cos 45°,\, 0,\, \sin 45°,\, 0]^T = [\frac{\sqrt{2}}{2},\, 0,\, \frac{\sqrt{2}}{2},\, 0]^T$. Si ottiene quindi $\boldsymbol{F}_{\text{grav},B} = [-mg,\, 0,\, 0]^T$: il peso è diretto interamente lungo $-x_B$, ovvero verso la coda, come atteso per un veicolo sospeso verticalmente.

4.5.2 Propulsione (`fcn_mixer`)

Ciascun motore genera una spinta T_i e una coppia reattiva Q_i proporzionali al quadrato della velocità angolare del rotore:

$$T_i = k_T\,\Omega_i^2, \qquad Q_i = k_Q\,\Omega_i^2, \qquad i = 1, \ldots, 4 \tag{4.9}$$

dove $k_T = 1.022 \times 10^{-4}\,\text{N}/(\text{rad}/\text{s})^2$ e $k_Q = 2.021 \times 10^{-6}\,\text{Nm}/(\text{rad}/\text{s})^2$ sono i coefficienti determinati nel Capitolo 2. La spinta è diretta lungo $+x_B$ (asse forward del corpo) per tutti i motori.

La disposizione dei motori in configurazione X (Tabella 2.8) definisce la matrice di allocazione $\mathbf{M}_{\text{mix}}$ che mappa il vettore $\boldsymbol{\Omega}^2 = [\Omega_1^2,\, \Omega_2^2,\, \Omega_3^2,\, \Omega_4^2]^T$ nel vettore delle forze e dei momenti propulsivi:

$$\begin{bmatrix} F_x \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \underbrace{\begin{bmatrix} k_T & k_T & k_T & k_T \\ -k_Q & +k_Q & -k_Q & +k_Q \\ +d\,k_T & -d\,k_T & -d\,k_T & +d\,k_T \\ -d\,k_T & -d\,k_T & +d\,k_T & +d\,k_T \end{bmatrix}}_{\mathbf{M}_{\text{mix}}} \boldsymbol{\Omega}^2 \tag{4.10}$$

dove $d = 0.325\,\text{m}$ è la distanza di ciascun motore dal baricentro.

La struttura della matrice riflette la fisica del sistema. La prima riga esprime la spinta totale lungo x_B, somma delle quattro spinte individuali. La seconda riga genera il momento di rollio τ_x tramite le coppie reattive dei motori: i motori CW (1 e 3) producono coppia reattiva $-k_Q\Omega^2$ e i motori CCW (2 e 4) producono $+k_Q\Omega^2$, per cui una variazione differenziale della velocità dei rotori genera un momento netto attorno a x_B. La terza riga genera il momento di beccheggio τ_y tramite il differenziale di spinta tra motori posizionati a $+d_z$ e $-d_z$ rispetto al baricentro. La quarta riga genera il momento di imbardata τ_z tramite il differenziale di spinta tra motori posizionati a $+d_y$ e $-d_y$.

Il vettore risultante è scomposto in forza propulsiva $\boldsymbol{F}_{\text{prop},B} = [F_x,\, 0,\, 0]^T$ e momento propulsivo $\boldsymbol{\tau}_{\text{prop},B} = [\tau_x,\, \tau_y,\, \tau_z]^T$.

4.5.3 Aerodinamica (`fcn_aerodynamics`)

Il modello aerodinamico calcola le forze e i momenti generati dall'interazione tra il veicolo e il flusso d'aria. La trattazione è condotta in ipotesi quasi-stazionaria: i coefficienti aerodinamici sono funzione istantanea delle condizioni di volo correnti, senza modellazione di effetti di ritardo o isteresi. Il calcolo si articola in cinque passi.

Passo 1, Mescolanza degli elevoni. I comandi in ingresso sono le deflessioni dei due elevoni, δ_{eL} e δ_{eR} (in radianti). Da questi si derivano due variabili di controllo virtuali:

$$\delta_e = \tfrac{1}{2}(\delta_{eL} + \delta_{eR}) \qquad \text{(deflessione simmetrica, beccheggio)} \tag{4.11}$$

$$\delta_a = \tfrac{1}{2}(\delta_{eL} - \delta_{eR}) \qquad \text{(deflessione asimmetrica, rollio)} \tag{4.12}$$

Passo 2, Angoli aerodinamici e pressione dinamica. Le componenti della velocità nel riferimento del corpo sono u, v, w. L'angolo di attacco α, l'angolo di sideslip β e la velocità totale V_{air} sono:

$$V_{\text{air}} = \sqrt{u^2 + v^2 + w^2}, \qquad \alpha = \operatorname{atan2}(w,\, u), \qquad \beta = \arcsin\!\left(\frac{v}{V_{\text{safe}}}\right) \tag{4.13}$$

dove $V_{\text{safe}} = \max(V_{\text{air}},\, V_{\min})$ con $V_{\min} = 0.5\,\text{m}\,\text{s}^{-1}$ è introdotta per evitare divisioni per zero a velocità prossime a zero (regime di hover). La pressione dinamica è calcolata con la velocità reale V_{air}:

$$\bar{q} = \tfrac{1}{2}\rho\, V_{\text{air}}^2 \tag{4.14}$$

dove $\rho = 1.112\,\text{kg}\,\text{m}^{-3}$ è la densità di riferimento dell'aria. L'utilizzo di V_{air} (e non di V_{safe}) garantisce che $\bar{q} \to 0$ in hover, annullando correttamente le forze aerodinamiche quando la velocità è trascurabile.

Le velocità angolari adimensionali, necessarie per le derivate di smorzamento, sono:

$$\hat{p} = \frac{p\,b}{2\,V_{\text{safe}}}, \qquad \hat{q} = \frac{q\,c}{2\,V_{\text{safe}}}, \qquad \hat{r} = \frac{r\,b}{2\,V_{\text{safe}}} \tag{4.15}$$

dove $b = 0.73\,\text{m}$ è l'apertura alare e $c = 0.14\,\text{m}$ è la corda media aerodinamica.

Passo 3, Coefficienti aerodinamici. I coefficienti aerodinamici sono costruiti secondo il modello classico a derivate di stabilità, nel quale ogni coefficiente di forza o momento è espresso come combinazione lineare di variabili di stato e comandi di controllo. Le derivate di stabilità quantificano la sensibilità di ciascun coefficiente a una variazione unitaria della variabile corrispondente e rappresentano, di fatto, una linearizzazione locale del comportamento aerodinamico del veicolo.

I coefficienti di portanza, resistenza e momento di beccheggio dell'ala sono approssimati da reti neurali *shallow* (un singolo strato nascosto) con funzione di attivazione `tansig` $\sigma(x) = \frac{2}{1+e^{-2x}} - 1$, i cui pesi sono stati addestrati su dati sperimentali. La forma analitica di ciascuna rete è:

$$C_{(\cdot),\text{wing}}(\alpha) = \boldsymbol{\theta}^{\top} \begin{bmatrix} 1 \\ \sigma\big(\mathbf{V}\,\alpha + \mathbf{c}\big) \end{bmatrix} \tag{4.16}$$

dove $\boldsymbol{\theta}$, $\mathbf{V}$ e $\mathbf{c}$ sono rispettivamente i pesi dello strato di uscita (incluso il bias), i pesi dello strato nascosto e i bias dello strato nascosto. Si ottengono così tre predittori non lineari:

$$C_{L,\text{wing}}(\alpha), \qquad C_{D,\text{wing}}(\alpha), \qquad C_{m,\text{wing}}(\alpha) \tag{4.17}$$

Questa rappresentazione cattura il comportamento non lineare dell'ala (incluso lo stallo) in modo continuo e derivabile, proprietà vantaggiosa per l'integrazione numerica. I contributi delle superfici di controllo, dello smorzamento e dell'accoppiamento sono invece modellati tramite derivate lineari.

Coefficienti longitudinali:

$$C_L = \underbrace{C_{L,\text{wing}}(\alpha)}_{\text{portanza ala (NN)}} + \underbrace{C_{L,\delta_e}\,\delta_e}_{\text{elevone}} \tag{4.18}$$

$$C_D = \underbrace{C_{D,\text{wing}}(\alpha)}_{\text{resistenza ala (NN)}} + \underbrace{C_{D,0,\text{nw}}}_{\text{parassita non-alare}} \tag{4.19}$$

$$C_m = \underbrace{C_{m,\text{wing}}(\alpha)}_{\text{momento ala (NN)}} + \underbrace{C_{m,q}\,\hat{q}}_{\text{smorzamento}} + \underbrace{C_{m,\delta_e}\,\delta_e}_{\text{elevone}} + \underbrace{C_L\,\frac{\Delta x_{\text{CG}}}{c}}_{\text{offset CG}} \tag{4.20}$$

La resistenza dell'ala è predetta direttamente dalla rete neurale $C_{D,\text{wing}}(\alpha)$, che incorpora sia la resistenza indotta sia quella di profilo. Per isolare il contributo parassita dei componenti non-alari (fusoliera, bracci, pod) si calcola:

$$C_{D,0,\text{nw}} = C_{D,0,\text{tot}} - C_{D,\text{wing}}(0) \tag{4.21}$$

dove $C_{D,0,\text{tot}} = 0.15$ è la resistenza parassita totale del drone stimata da VSPAero e $C_{D,\text{wing}}(0)$ è la resistenza dell'ala a $\alpha = 0$ restituita dalla rete neurale. In questo modo il contributo alare non viene contato due volte.

L'ultimo termine di C_m trasla il momento di beccheggio per tenere conto della differenza tra centro aerodinamico e baricentro: un baricentro a valle del centro aerodinamico ($\Delta x_{\text{CG}} > 0$) riduce la stabilità longitudinale statica.

Coefficienti latero-direzionali:

$$C_Y = \underbrace{C_{Y,\beta}\,\beta}_{\text{forza laterale}} \tag{4.22}$$

$$C_l = \underbrace{C_{l,\beta}\,\beta}_{\text{diedro}} + \underbrace{C_{l,p}\,\hat{p}}_{\text{smorzamento rollio}} + \underbrace{C_{l,r}\,\hat{r}}_{\text{accopp. yaw}\to\text{roll}} + \underbrace{C_{l,\delta_a}\,\delta_a}_{\text{elevone}} \tag{4.23}$$

$$C_n = \underbrace{C_{n,\beta}\,\beta}_{\text{stabilità direz.}} + \underbrace{C_{n,p}\,\hat{p}}_{\text{accopp. roll}\to\text{yaw}} + \underbrace{C_{n,r}\,\hat{r}}_{\text{smorzamento imbardata}} + \underbrace{C_{n,\delta_a}\,\delta_a}_{\text{accopp. elevone}} \tag{4.24}$$

I coefficienti latero-direzionali sono rimasti invariati rispetto alla formulazione lineare, poiché le reti neurali sono state addestrate solo sui coefficienti longitudinali dell'ala.

I valori numerici delle derivate di stabilità sono raccolti nella Tabella 4.1, accompagnati dal significato fisico di ciascuna. I valori sono stati estratti dall'analisi VSPAero sul modello geometrico del velivolo [17, 26], secondo le convenzioni di segno standard della meccanica del volo [12, 23, 47].

Tabella 4.1: Derivate di stabilità aerodinamica e relativo significato fisico.

Derivata	**Valore**	**Significato fisico**
Longitudinali		
C_{L,δ_e}	1.5	Incremento di portanza per deflessione elevone simmetrica
C_{m,δ_e}	-0.8	Momento di beccheggio per deflessione elevone simmetrica
$C_{m,q}$	-3.72	Smorzamento in beccheggio (stabilizzante in quanto < 0)
$C_{D,0,\text{tot}}$	0.15	Resistenza parassita totale
Latero-direzionali		
$C_{Y,\beta}$	-2.13	Forza laterale per angolo di derapata
$C_{l,\beta}$	-1.08	Effetto diedro: rollio indotto dalla derapata (stabilizzante in quanto < 0)
$C_{l,p}$	-1.40	Smorzamento in rollio (stabilizzante in quanto < 0)
$C_{l,r}$	-1.28	Accoppiamento yaw $\rightarrow$ roll
C_{l,δ_a}	-0.33	Efficacia di rollio dell'elevone asimmetrico
$C_{n,\beta}$	0.26	Stabilità direzionale statica (stabilizzante in quanto > 0)
$C_{n,p}$	-0.08	Accoppiamento roll $\rightarrow$ yaw
$C_{n,r}$	-0.13	Smorzamento in imbardata (stabilizzante in quanto < 0)
C_{n,δ_a}	0	Imbardata da deflessione elevone asimmetrica (trascurata)

Convenzioni di segno. Le convenzioni di segno adottate in tabella seguono lo standard aerospaziale per il body frame e meritano una breve esplicitazione, dato che diverse derivate hanno segno non banale. La convenzione per gli elevoni è la stessa del Capitolo 5: $\delta_e > 0$ corrisponde a deflessione simmetrica verso il basso, e $\delta_a > 0$ a deflessione antisimmetrica con elevone destro abbassato e sinistro alzato. Il segno negativo di C_{m,δ_e} riflette il fatto che deflettendo simmetricamente verso il basso si sposta il centro di pressione verso il bordo d'uscita, generando un momento a cabrare (pitch negativo nel body frame con pitch positivo verso il muso in alto). Il segno negativo di C_{l,δ_a} riflette analogamente che, con elevone destro abbassato, il velivolo rolla verso sinistra. I segni negativi di $C_{Y,\beta}$, $C_{l,\beta}$ e i positivi di $C_{n,\beta}$ corrispondono al comportamento di stabilità statica direzionale e laterale standard: il velivolo tende a opporsi alla derapata e a riallinearsi spontaneamente al vento relativo.

Passo 4, Trasformazione dal riferimento aerodinamico al riferimento del corpo. I coefficienti C_L e C_D sono definiti nel sistema wind (portanza perpendicolare alla velocità, resistenza parallela). Per ottenere le forze lungo gli assi del corpo si proietta tramite l'angolo di attacco:

$$C_{X,B} = -C_D \cos\alpha + C_L \sin\alpha \tag{4.25}$$

$$C_{Z,B} = -C_D \sin\alpha - C_L \cos\alpha \tag{4.26}$$

La resistenza ha sempre componente negativa lungo x_B (frena il moto in avanti), mentre la portanza genera una componente positiva lungo x_B quando $\alpha > 0$ (contributo propulsivo della portanza) e una componente negativa lungo z_B (sostentamento). Il coefficiente di forza laterale C_Y è già espresso lungo y_B.

Passo 5, Forze e momenti dimensionali. Le forze e i momenti aerodinamici nel riferimento del corpo sono:

$$\boldsymbol{F}_{\text{aero},B} = \bar{q}\, S \begin{bmatrix} C_{X,B} \\ C_Y \\ C_{Z,B} \end{bmatrix}, \qquad \boldsymbol{\tau}_{\text{aero},B} = \bar{q}\, S \begin{bmatrix} b\, C_l \\ c\, C_m \\ b\, C_n \end{bmatrix} \tag{4.27}$$

dove $S = 0.1022\,\text{m}^2$ è la superficie alare di riferimento. I momenti di rollio e imbardata utilizzano l'apertura alare b come lunghezza di riferimento, mentre il momento di beccheggio utilizza la corda c, secondo la convenzione standard in aerodinamica.

4.5.4 Contatto con il Terreno (`fcn_ground`)

Il modello di contatto simula l'interazione del veicolo con il suolo durante le fasi di decollo e atterraggio. La coordinata p_D del riferimento NED è positiva verso il

basso; il piano del terreno corrisponde a $p_D = 0$, per cui $p_D > 0$ indica penetrazione nel suolo.

Condizione di attivazione. Il contatto è attivato se e solo se $p_D > 0$. Per $p_D \leq 0$ la forza di contatto è identicamente nulla.

Forza normale. La componente verticale della forza di contatto nel riferimento NED è modellata come una molla-smorzatore unidirezionale:

$$F_{\text{norm},D} = \min\Big(0,\ -k_G\, p_D - b_G\, \max(0,\, v_D)\Big) \tag{4.28}$$

dove $k_G = 50000\,\text{N}\,\text{m}^{-1}$ è la rigidezza, $b_G = 2000\,\text{N}\,\text{s}\,\text{m}^{-1}$ è il coefficiente di smorzamento e v_D è la componente verticale della velocità nel riferimento NED. L'operatore $\min(\cdot, 0)$ garantisce che la forza sia diretta esclusivamente verso l'alto (il terreno può spingere ma non tirare). L'operatore $\max(0, v_D)$ limita lo smorzamento alla sola fase di compressione ($v_D > 0$, ovvero il veicolo si muove verso il basso), evitando che lo smorzamento freni il rimbalzo.

La velocità nel riferimento NED si ottiene trasformando la velocità del corpo:

$$\boldsymbol{v}_{\text{NED}} = \mathbf{R}(\mathbf{q})\, \boldsymbol{v}_B, \qquad v_D = \boldsymbol{e}_3^T\, \boldsymbol{v}_{\text{NED}} \tag{4.29}$$

Attrito. Il modello include una forza di attrito di Coulomb nel piano orizzontale NED, proporzionale alla forza normale:

$$F_{\text{frict},N} = -\mu\, |F_{\text{norm},D}|\ \text{sign}(v_N), \qquad F_{\text{frict},E} = -\mu\, |F_{\text{norm},D}|\ \text{sign}(v_E) \tag{4.30}$$

dove $\mu = 0.5$ è il coefficiente di attrito e v_N, v_E sono le componenti orizzontali della velocità NED. Per evitare oscillazioni numeriche a velocità molto basse, l'attrito è disattivato per $|v_N| < 0.01\,\text{m}\,\text{s}^{-1}$ (analogamente per v_E).

Trasformazione nel riferimento del corpo. La forza di contatto complessiva nel riferimento NED è $\boldsymbol{F}_{\text{ground,NED}} = [F_{\text{frict},N},\, F_{\text{frict},E},\, F_{\text{norm},D}]^T$, e viene trasformata nel riferimento del corpo prima della somma con gli altri contributi:

$$\boldsymbol{F}_{\text{ground},B} = \mathbf{R}^T(\mathbf{q})\, \boldsymbol{F}_{\text{ground,NED}} \tag{4.31}$$

Il modello assume contatto puntuale nel baricentro e non genera momenti di contatto.

4.6 Sottosistema Equations of Motion

Il sottosistema delle equazioni del moto (Figura 4.5) integra nel tempo le equazioni differenziali che governano l'evoluzione dello stato. È organizzato in due blocchi: *Dynamics* (dinamica traslazionale e rotazionale) e *Kinematics* (cinematica dell'assetto e della posizione).

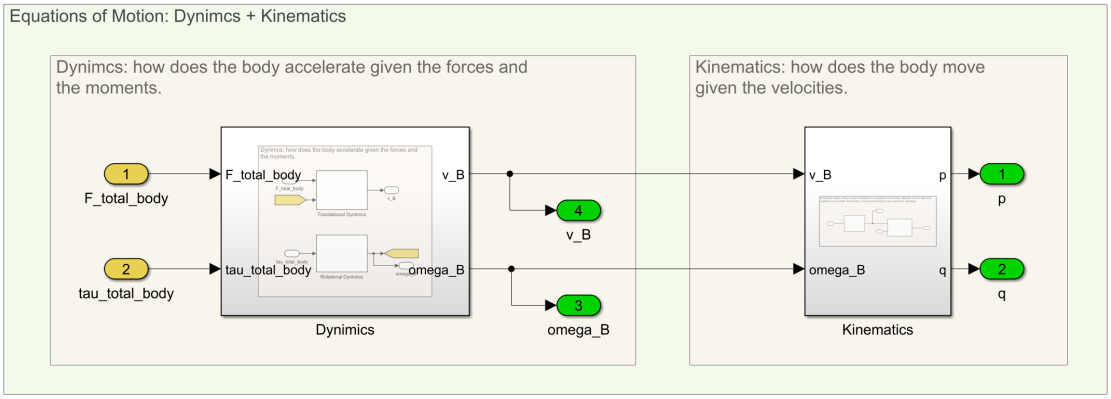


Figura 4.5: Sottosistema Equations of Motion: blocchi Dynamics e Kinematics con integratori.

L'integrazione numerica è effettuata mediante solutore Runge-Kutta di quarto ordine a passo fisso (`ode4`).

4.6.1 Dinamica Traslazionale (`fcn_trans_dynamics`)

La seconda legge di Newton, formulata nel riferimento non inerziale del corpo, è:

$$m\,\dot{\boldsymbol{v}}_B = \boldsymbol{F}_{\text{tot},B} - \boldsymbol{\omega}_B \times (m\,\boldsymbol{v}_B) \tag{4.32}$$

Il termine $\boldsymbol{\omega}_B \times (m\,\boldsymbol{v}_B)$ è la forza apparente dovuta alla non-inerzialità del riferimento rotante. Dividendo per m e riordinando:

$$\dot{\boldsymbol{v}}_B = \frac{\boldsymbol{F}_{\text{tot},B}}{m} - \boldsymbol{\omega}_B \times \boldsymbol{v}_B \tag{4.33}$$

Il prodotto vettoriale si sviluppa come:

$$\boldsymbol{\omega}_B \times \boldsymbol{v}_B = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} qw - rv \\ ru - pw \\ pv - qu \end{bmatrix} \tag{4.34}$$

Sostituendo in (4.33) si ottengono le tre equazioni scalari:

$$\dot{u} = \frac{F_{\text{tot},x}}{m} - (qw - rv) = \frac{F_{\text{tot},x}}{m} + rv - qw \tag{4.35}$$

$$\dot{v} = \frac{F_{\text{tot},y}}{m} - (ru - pw) = \frac{F_{\text{tot},y}}{m} + pw - ru \tag{4.36}$$

$$\dot{w} = \frac{F_{\text{tot},z}}{m} - (pv - qu) = \frac{F_{\text{tot},z}}{m} + qu - pv \tag{4.37}$$

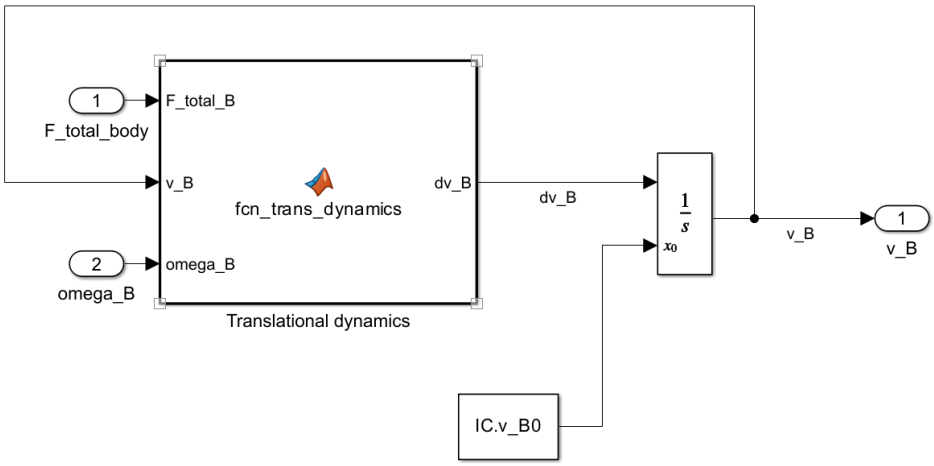


Figura 4.6: Blocco Simulink della dinamica traslazionale.

4.6.2 Dinamica Rotazionale (`fcn_rot_dynamics`)

Le equazioni di Eulero per il moto rotatorio attorno al baricentro, nel riferimento del corpo, sono:

$$\mathbf{I}\,\dot{\boldsymbol{\omega}}_B = \boldsymbol{\tau}_{\text{tot},B} - \boldsymbol{\omega}_B \times (\mathbf{I}\,\boldsymbol{\omega}_B) \tag{4.38}$$

dove $\mathbf{I} = \text{diag}(I_{xx},\, I_{yy},\, I_{zz})$ è la matrice di inerzia diagonale. Il termine $\boldsymbol{\omega}_B \times (\mathbf{I}\,\boldsymbol{\omega}_B)$ rappresenta l'effetto giroscopico, ovvero la coppia apparente che nasce dall'interazione tra la rotazione del riferimento e la distribuzione di momento angolare.

Definendo il momento angolare nel corpo $\boldsymbol{L}_B = \mathbf{I}\,\boldsymbol{\omega}_B$, il prodotto vettoriale si scrive:

$$\boldsymbol{\omega}_B \times \boldsymbol{L}_B = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}\,p \\ I_{yy}\,q \\ I_{zz}\,r \end{bmatrix} = \begin{bmatrix} q\,(I_{zz}\,r) - r\,(I_{yy}\,q) \\ r\,(I_{xx}\,p) - p\,(I_{zz}\,r) \\ p\,(I_{yy}\,q) - q\,(I_{xx}\,p) \end{bmatrix} \tag{4.39}$$

Risolvendo per l'accelerazione angolare:

$$\dot{\boldsymbol{\omega}}_B = \mathbf{I}^{-1}\left[\boldsymbol{\tau}_{\text{tot},B} - \boldsymbol{\omega}_B \times (\mathbf{I}\,\boldsymbol{\omega}_B)\right] \tag{4.40}$$

Le tre equazioni scalari, sfruttando la diagonalità di $\mathbf{I}$, sono:

$$I_{xx}\,\dot{p} = \tau_{\mathrm{tot},x} - (I_{zz} - I_{yy})\,qr \tag{4.41}$$
$$I_{yy}\,\dot{q} = \tau_{\mathrm{tot},y} - (I_{xx} - I_{zz})\,pr \tag{4.42}$$
$$I_{zz}\,\dot{r} = \tau_{\mathrm{tot},z} - (I_{yy} - I_{xx})\,pq \tag{4.43}$$

La matrice inversa $\mathbf{I}^{-1} = \mathrm{diag}(1/I_{xx},\, 1/I_{yy},\, 1/I_{zz})$ è pre-calcolata e memorizzata nella struttura dei parametri per efficienza computazionale. Nell'implementazione MATLAB il calcolo di $\boldsymbol{\omega}_B \times \boldsymbol{L}_B$ è effettuato direttamente in forma componente per evitare la costruzione esplicita della matrice antisimmetrica.

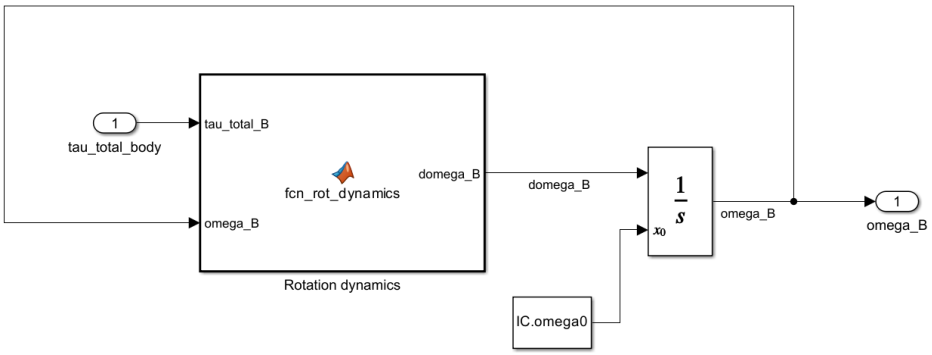


Figura 4.7: Blocco Simulink della dinamica rotazionale.

4.6.3 Cinematica dell'Assetto (`fcn_quat_kinematics`)

La derivata temporale del quaternione in funzione della velocità angolare nel corpo è data dall'equazione differenziale:

$$\dot{\mathbf{q}} = \frac{1}{2}\,\boldsymbol{\Omega}\,\mathbf{q} \tag{4.44}$$

dove $\boldsymbol{\Omega}$ è la matrice antisimmetrica 4×4 costruita dalle componenti di $\boldsymbol{\omega}_B = [p,\, q,\, r]^T$:

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \tag{4.45}$$

Sviluppando il prodotto $\frac{1}{2}\boldsymbol{\Omega}\,\mathbf{q}$ si ottengono le quattro equazioni scalari:

$$\dot{q}_0 = \tfrac{1}{2}(-q_1\,p - q_2\,q - q_3\,r) \tag{4.46}$$
$$\dot{q}_1 = \tfrac{1}{2}(+q_0\,p - q_3\,q + q_2\,r) \tag{4.47}$$
$$\dot{q}_2 = \tfrac{1}{2}(+q_3\,p + q_0\,q - q_1\,r) \tag{4.48}$$
$$\dot{q}_3 = \tfrac{1}{2}(-q_2\,p + q_1\,q + q_0\,r) \tag{4.49}$$

Questa formulazione è priva di singolarità per qualsiasi assetto del veicolo, a differenza della cinematica basata su angoli di Eulero, che presenta una singolarità a $\theta = \pm 90°$, corrispondente esattamente all'assetto di hover del tailsitter.

4.6.4 Normalizzazione dei Quaternioni (`fcn_quat_normalize`)

L'integrazione numerica introduce inevitabilmente una deriva nella norma del quaternione dovuta agli errori di arrotondamento. Se non corretta, questa deriva si accumula nel tempo e compromette la proprietà di ortonormalità della matrice di rotazione, con effetti che si propagano rapidamente attraverso l'intero modello (possibile instabilità della simulazione).

A ogni passo di integrazione, il quaternione è normalizzato:

$$\mathbf{q}_{\text{norm}} = \frac{\mathbf{q}}{\|\mathbf{q}\|} = \frac{\mathbf{q}}{\sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}} \tag{4.50}$$

In Simulink, il blocco di normalizzazione è posizionato sul segnale di uscita dell'integratore, prima del feedback verso tutti i blocchi che dipendono dall'assetto.

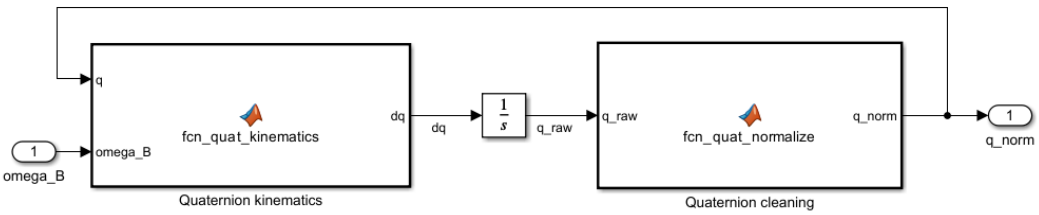


Figura 4.8: Blocco Simulink della cinematica dell'assetto: integrazione dei quaternioni con normalizzazione.

4.6.5 Cinematica della Posizione (`fcn_pos_kinematics`)

La velocità nel riferimento inerziale NED è ottenuta trasformando la velocità del corpo mediante la matrice di rotazione:

$$\dot{\boldsymbol{p}} = \mathbf{R}(\mathbf{q})\,\boldsymbol{v}_B \tag{4.51}$$

Espandendo il prodotto matrice-vettore con gli elementi di $\mathbf{R}$ espressi in funzione dei quaternioni:

$$\begin{bmatrix} \dot{p}_N \\ \dot{p}_E \\ \dot{p}_D \end{bmatrix} = \begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2)\, u + 2(q_1 q_2 - q_0 q_3)\, v + 2(q_1 q_3 + q_0 q_2)\, w \\ 2(q_1 q_2 + q_0 q_3)\, u + (q_0^2 - q_1^2 + q_2^2 - q_3^2)\, v + 2(q_2 q_3 - q_0 q_1)\, w \\ 2(q_1 q_3 - q_0 q_2)\, u + 2(q_2 q_3 + q_0 q_1)\, v + (q_0^2 - q_1^2 - q_2^2 + q_3^2)\, w \end{bmatrix} \tag{4.52}$$

Nell'implementazione MATLAB, la matrice $\mathbf{R}$ è costruita esplicitamente dai quaternioni correnti ad ogni passo di integrazione. Si noti che la forma utilizzata nel codice ($q_0^2 + q_1^2 - q_2^2 - q_3^2$ per il termine diagonale) è algebricamente equivalente a quella semplificata $1 - 2(q_2^2 + q_3^2)$ in virtù del vincolo $\|\mathbf{q}\| = 1$; entrambe le forme sono corrette, ma la prima è leggermente più robusta numericamente quando la norma del quaternione ha una piccola deriva prima della normalizzazione.

L'integrazione di (4.51) fornisce la posizione $\boldsymbol{p} = [p_N,\, p_E,\, p_D]^T$ nel riferimento NED.

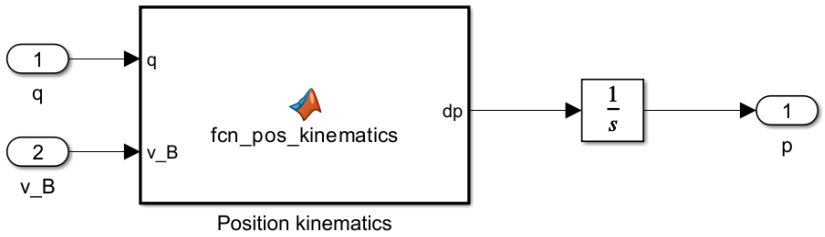


Figura 4.9: Blocco Simulink della cinematica della posizione: trasformazione body-to-inertial e integrazione.

Capitolo 5

Allocazione dei Controlli

5.1 Introduzione

Il sottosistema di allocazione dei controlli traduce i comandi prodotti dal controllore di assetto e dal generatore del comando di assetto, ovvero il vettore di coppia desiderata $\boldsymbol{\tau}_{\text{des}}$ e la spinta totale T_{des}, negli ingressi fisici degli attuatori: le velocità angolari al quadrato dei quattro motori $(\Omega_1^2, \ldots, \Omega_4^2)$ e le deflessioni dei due elevoni $(\delta_{eL}, \delta_{eR})$ [6, 47]. Questa traduzione non è immediata, poiché il tailsitter dispone di due sottosistemi attuativi, motori ed elevoni, in grado di generare entrambi momento di rollio e beccheggio, ma con autorità di controllo differenti a seconda della velocità aerodinamica.

In hover gli elevoni sono inefficaci, perché la pressione dinamica $\bar{q} = \frac{1}{2}\rho V^2$ tende a zero e con essa la coppia aerodinamica generata da una deflessione di superficie. Il controllo di rollio e beccheggio è affidato interamente al differenziale di spinta dei motori. In crociera, viceversa, $\bar{q}$ è significativa e gli elevoni diventano il principale attuatore di momento; i motori sono allora liberi di concentrarsi sulla generazione della spinta richiesta a bilanciare la resistenza aerodinamica. Tra questi due regimi esiste un'intera fase di transizione in cui i due sottosistemi devono cooperare in modo coordinato.

5.1.1 Formulazione del Problema di Allocazione

Da un punto di vista formale, il problema è il seguente: in uscita dal controllore di assetto si dispone di un comando virtuale di quattro grandezze, $\boldsymbol{u}_{\text{virt}} = (T_{\text{des}}, \tau_{x,\text{des}}, \tau_{y,\text{des}}, \tau_{z,\text{des}})^T$ (una di spinta e tre di coppia), e occorre determinare le sei grandezze fisiche $(\Omega_1^2, \Omega_2^2, \Omega_3^2, \Omega_4^2, \delta_e, \delta_a)$ che le producono. Il velivolo è dunque *sovra-attuato* (*overactuated*): dispone di più attuatori dei gradi di libertà di comando da controllare. Il problema di allocazione che ne risulta è formalmente

sottodeterminato, più incognite che equazioni, e ammette in linea di principio infinite soluzioni. Occorre dunque un criterio aggiuntivo per fissarne una.

Per il velivolo qui considerato la ridondanza è circoscritta a due gradi di libertà: ciascuno dei momenti di rollio e di beccheggio può essere generato sia dai motori (tramite differenziale di spinta) sia dagli elevoni (tramite deflessione antisimmetrica e simmetrica), mentre la spinta totale e il momento di imbardata sono producibili dai soli motori. Ciò suggerisce di scomporre l'allocazione in due fasi successive: dapprima la *ripartizione* dei momenti di rollio e beccheggio fra i due sottosistemi attuativi; quindi l'inversione standard fra i quattro comandi residui $(T,\, \tau_x^{(\mathrm{mot})},\, \tau_y^{(\mathrm{mot})},\, \tau_z)$ e le quattro velocità motore, in parallelo alla conversione della quota elevone $(\tau_x^{(\mathrm{elev})},\, \tau_y^{(\mathrm{elev})})$ nelle deflessioni virtuali. Le due inversioni a valle del blending sono ben determinate.

Criterio di scelta della ripartizione. Resta da specificare come ripartire la coppia richiesta fra i due sottosistemi. L'approccio tradizionale nella letteratura sull'allocazione di controllo per veicoli ibridi è il *blending sigmoidale*, in cui una funzione $\sigma(V) \in [0,1]$ pesa con continuità il contributo dell'uno e dell'altro sottosistema in funzione della velocità aerodinamica. Questa formulazione, descritta in dettaglio più avanti, presenta tuttavia problema strutturale legata alla saturazione fisica degli elevoni, che la rende inadatta a un'applicazione in cui sia richiesta garanzia di conservazione della coppia comandata.

L'architettura adottata in questa tesi rinuncia al blending sigmoidale e introduce in suo luogo una *ripartizione adattiva basata sui tetti fisici reali degli attuatori*, ricalcolati istante per istante in funzione della pressione dinamica corrente. Il principio guida è il seguente: gli elevoni sono sfruttati al massimo della loro capacità (entro un margine di sicurezza), perché ad alta velocità sono efficienti e perché il loro impiego non distrae i motori dalla generazione della spinta primaria; i motori coprono l'eventuale deficit.

La strategia garantisce due proprietà fondamentali, dimostrate nelle sezioni seguenti:

1. la somma dei due contributi è esattamente $\boldsymbol{\tau}_{\mathrm{des}}$, indipendentemente da V;

2. gli elevoni non vengono mai comandati oltre la loro saturazione fisica, eliminando l'ambiguità tra "comando" e "risposta" tipica delle architetture sigmoidali pure.

La prima è una proprietà algebrica della legge di ripartizione (formalizzata in Sezione 5.3.4); la seconda è una proprietà di sicurezza meccanica, importante perché saturazioni di elevone prive di gestione esplicita propagherebbero al controllore di assetto errori di coppia non realizzata.

5.2 Architettura del Sottosistema Simulink

Il sottosistema *Control Allocation* in Simulink riflette fedelmente la struttura logica dell'allocazione. La Figura 5.1 mostra l'interfaccia esterna del blocco, con i due ingressi, il vettore di coppia desiderata $\boldsymbol{\tau}_{\text{des}}$ e la spinta totale T_{des}, e le due uscite, il vettore delle velocità angolari al quadrato $\boldsymbol{\Omega}^2$ e la deflessione degli elevoni δ_e.

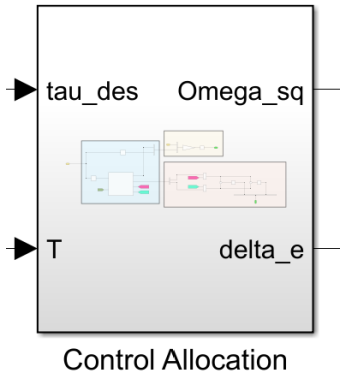


Figura 5.1: Interfaccia esterna del blocco Control Allocation nel modello Simulink.

All'interno del sottosistema (Figura 5.2), la struttura è articolata in tre parti:

1. **Blending and Allocation:** riceve $\boldsymbol{\tau}_{\text{des}}$ e la velocità del vento relativo V_{air} e ripartisce i momenti di rollio e beccheggio tra motori ed elevoni secondo la legge adattiva descritta nella Sezione 5.3. Il momento di imbardata τ_z è estratto e inviato direttamente ai motori senza blending.

2. **Motors:** riceve la spinta totale T_{des} e la componente motore dei momenti $\boldsymbol{\tau}_{\text{mot}}$, esegue l'inversione della matrice di allocazione $\mathbf{M}_{\text{mix}}^{-1}$ e satura le velocità angolari al quadrato.

3. **Elevons:** riceve la componente elevone dei momenti $\boldsymbol{\tau}_{\text{elev}}$, la moltiplica per i guadagni K_{pitch} e K_{roll} (calcolati istantaneamente da `fcn_blending` in funzione di V_{air}) per ottenere le deflessioni virtuali simmetrica e antisimmetrica, le combina nelle deflessioni fisiche dei due elevoni e satura ai limiti meccanici.

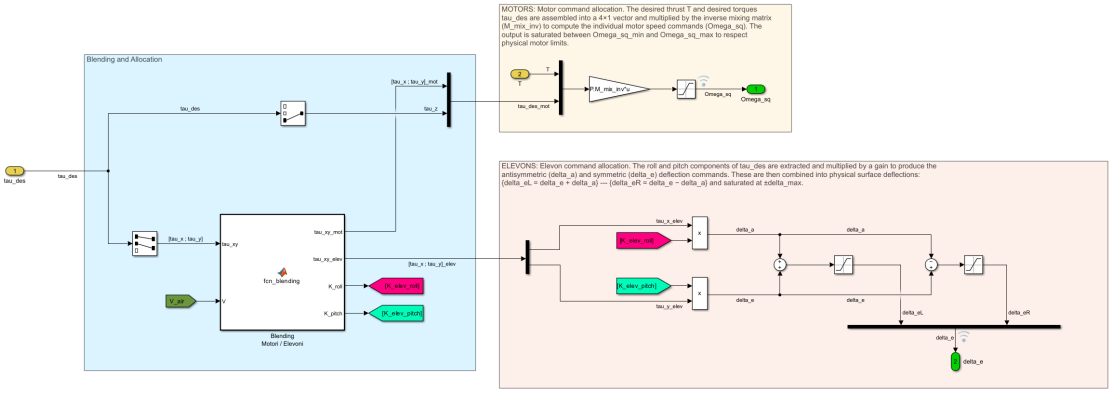


Figura 5.2: Sottosistema Simulink dell'allocazione dei controlli: blocco di blending (a sinistra, in azzurro), allocazione motore (in alto a destra, in giallo) e allocazione elevone (in basso a destra, in rosso).

5.3 Blending Adattivo: `fcn_blending`

La ripartizione dell'autorità di controllo tra i due sottosistemi attuativi è governata dalla funzione MATLAB `fcn_blending`, il cui dettaglio implementativo è mostrato in Figura 5.3. La funzione riceve il vettore $\boldsymbol{\tau}_{xy} = [\tau_x,\, \tau_y]^T$ (componenti rollio e beccheggio del momento desiderato) e la velocità V_{air}, e produce la ripartizione $(\boldsymbol{\tau}_{xy,\text{mot}},\, \boldsymbol{\tau}_{xy,\text{elev}})$ assieme ai guadagni di conversione momento–deflessione $(K_{\text{pitch}},\, K_{\text{roll}})$ validi all'istante corrente.

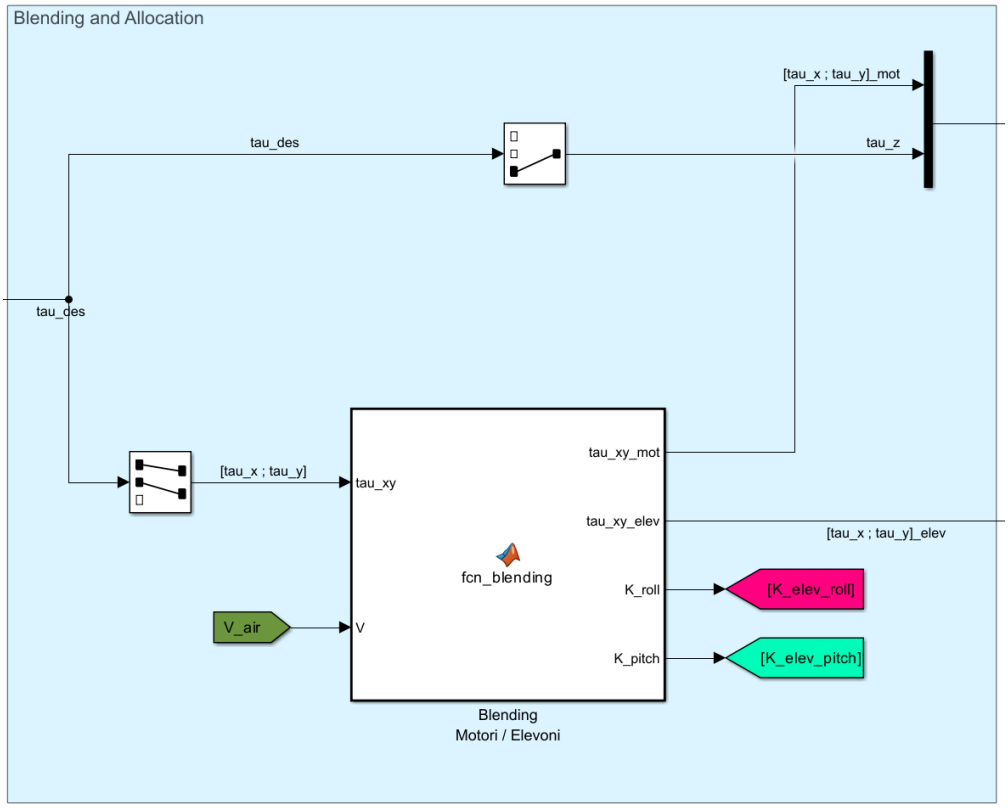


Figura 5.3: Blocco di blending: calcolo della pressione dinamica reale, dei tetti fisici degli elevoni e ripartizione dei momenti tra motori ed elevoni.

5.3.1 Pressione Dinamica e Coefficienti di Efficacia

Il primo passo è il calcolo della pressione dinamica reale, che governa l'efficacia delle superfici aerodinamiche. Si introduce una saturazione inferiore per garantire che la quantità sia ben definita anche per valori transienti negativi della velocità

(anche se con lo sviluppo al quadrato non si dovrebbero avere problemi):

$$V_{\text{safe}} = \max(V_{\text{air}},\, 0), \qquad \bar{q} = \frac{1}{2}\,\rho\, V_{\text{safe}}^2 \tag{5.1}$$

Da $\bar{q}$ si calcolano i *coefficienti di efficacia* delle deflessioni di elevone, ossia i guadagni dimensionali che convertono la deflessione [rad] nel momento aerodinamico generato [N m]. La derivazione segue dalla teoria classica del controllo aerodinamico. Consideriamo per primo il momento di beccheggio. Il coefficiente adimensionale di pitch è linearizzato attorno alla condizione $\delta_e = 0$:

$$C_m(\alpha,\, \delta_e) \approx C_{m,0}(\alpha) + C_{m,\delta_e}\,\delta_e \tag{5.2}$$

e il momento dimensionale corrispondente è:

$$M_y = \bar{q}\, S\, c\, C_m(\alpha,\, \delta_e) \tag{5.3}$$

La componente di momento dipendente dalla deflessione è dunque:

$$\tau_y^{(\text{elev})} = \bar{q}\, S\, c\, C_{m,\delta_e}\,\delta_e \equiv G_{\text{pitch}}\,\delta_e \tag{5.4}$$

con la definizione del coefficiente di efficacia di pitch:

$$\boxed{G_{\text{pitch}} = \bar{q}\, S\, c\, C_{m,\delta_e}} \tag{5.5}$$

Analogamente, per il momento di rollio prodotto dalla deflessione antisimmetrica, la lunghezza di riferimento è l'apertura alare b (e non la corda c, poiché il momento di rollio è dimensionalmente legato al braccio trasversale):

$$\boxed{G_{\text{roll}} = \bar{q}\, S\, b\, C_{l,\delta_a}} \tag{5.6}$$

Segno e convenzione. La convenzione di segno adottata per le deflessioni è la seguente: $\delta_e > 0$ corrisponde a deflessione simmetrica con bordo d'uscita verso il basso di entrambe le superfici, mentre $\delta_a > 0$ corrisponde a deflessione antisimmetrica con elevone destro abbassato e sinistro alzato. Da questa scelta discendono i segni dei coefficienti di efficacia, entrambi *negativi*:

$$C_{m,\delta_e} = -0.8\ \text{rad}^{-1}, \qquad C_{l,\delta_a} = -0.33\ \text{rad}^{-1}.$$

Fisicamente, una deflessione simmetrica verso il basso incrementa la portanza dell'ala, che si trova a poppa del baricentro: l'incremento di portanza locale, agendo su tale braccio, produce un momento di pitch picchiante ($\tau_y^{(\text{elev})} < 0$ per $\delta_e > 0$). Analogamente, una deflessione antisimmetrica con elevone destro abbassato genera un eccesso di portanza sull'ala destra che fa rollare il velivolo verso sinistra rispetto al body frame ($\tau_x^{(\text{elev})} < 0$ per $\delta_a > 0$).

I segni negativi si propagano per costruzione ai coefficienti di efficacia dimensionali G_{pitch} e G_{roll} e ai guadagni di conversione K_{pitch} e K_{roll} definiti più avanti, garantendo che la legge $\delta = K \cdot \tau$ produca automaticamente la deflessione del segno appropriato per generare la coppia richiesta.

Dipendenza quadratica da V. Si osservi la dipendenza dei coefficienti di efficacia dalla velocità: $G_{\text{pitch}}, G_{\text{roll}} \propto V^2$. Per il velivolo nominale ($\rho = 1.112\ \text{kg m}^{-3}$, $S = 0.1022\ \text{m}^2$, $b = 0.73\ \text{m}$, $c = 0.14\ \text{m}$, $C_{m,\delta_e} = -0.8$, $C_{l,\delta_a} = -0.33$):

$$G_{\text{pitch}}(V) = -6.36 \times 10^{-3}\, V^2\ \text{N m rad}^{-1} \tag{5.7}$$

$$G_{\text{roll}}(V) = -1.369 \times 10^{-2}\, V^2\ \text{N m rad}^{-1} \tag{5.8}$$

A $V = 10\ \text{m s}^{-1}$: $|G_{\text{pitch}}| = 0.636$, $|G_{\text{roll}}| = 1.369\ \text{N m rad}^{-1}$.
A $V = V_{\text{cruise}} = 61\ \text{m s}^{-1}$: $|G_{\text{pitch}}| = 23.7$, $|G_{\text{roll}}| = 50.9\ \text{N m rad}^{-1}$.
Il rapporto tra i valori a crociera e a 10 m/s è $V_{\text{cruise}}^2/100 = 37$, ossia gli elevoni sono 37 volte più efficaci a crociera che a $10\ \text{m s}^{-1}$, come conseguenza diretta della dipendenza quadratica.

Questa forte dipendenza è precisamente la ragione per cui il blending adattivo è preferibile a un set di guadagni costanti: usare un guadagno costante K_{pitch} richiederebbe di scegliere il valore corrispondente a una specifica V, e a tutte le altre velocità il comando δ_e produrrebbe un momento di entità non corrispondente alla richiesta. Con $K_{\text{pitch}} = 1/G_{\text{pitch}}(V_{\text{air}})$ ricalcolato istantaneamente, la conversione momento→deflessione è esatta a ogni velocità.

5.3.2 Cutoff a Bassa Velocità

A velocità molto basse $\bar{q}$ è trascurabile e gli elevoni non possono produrre momento utile. Comandarli sarebbe inutile e potrebbe introdurre rumore nel sistema, oltre a indurre inutili sollecitazioni meccaniche sul servo. Si introduce dunque una soglia di cutoff:

$$\bar{q}_{\text{min}} = \frac{1}{2}\,\rho\, V_{\text{min,elev}}^2, \qquad V_{\text{min,elev}} = 4\ \text{m/s} \tag{5.9}$$

Per $\bar{q} < \bar{q}_{\text{min}}$ la funzione `fcn_blending` restituisce immediatamente:

$$\boldsymbol{\tau}_{xy,\text{mot}} = \boldsymbol{\tau}_{xy}, \qquad \boldsymbol{\tau}_{xy,\text{elev}} = \mathbf{0}, \qquad K_{\text{pitch}} = K_{\text{roll}} = 0 \tag{5.10}$$

ossia tutto il carico è assegnato ai motori e gli elevoni non sono comandati.

Inoltre, il valore $V_{\text{min,elev}} = 4\ \text{m s}^{-1}$ è scelto al di sotto della velocità di hover-ready ($V_{\text{hover_ready}} = 8\ \text{m s}^{-1}$) per garantire che il blending sia attivo durante tutta la fase di transizione, disattivandosi unicamente a velocità inferiori a quelle di volo nominale (hover stazionario, fase di takeoff verticale, fase di landing). In tutti questi regimi, gli elevoni sono effettivamente inutili e il cutoff è benigno.

Continuità della transizione di cutoff. A $V = V_{\text{min,elev}}$, la transizione tra cutoff e blending attivo avviene in modo discontinuo: per $V < 4\ \text{m s}^{-1}$, $\tau_{xy,\text{elev}} = 0$ identicamente; per $V > 4\ \text{m s}^{-1}$, $\tau_{xy,\text{elev}}$ può essere non nullo. La discontinuità si verifica solo se la richiesta di coppia τ_{xy} è non nulla a $V = 4\ \text{m s}^{-1}$. In pratica, a

queste basse velocità il velivolo è in regime quasi-stazionario di hover, con $\tau_{xy} \approx 0$, e la discontinuità è invisibile. Si è scelto deliberatamente di non smussare il cutoff con una funzione di transizione continua (es. peso lineare in $V \in [3, 5]$) perché ciò avrebbe introdotto un secondo regime di blending, complicando inutilmente l'analisi. Il cutoff rigido a $V_{\min,\text{elev}}$ è preferibile per chiarezza di interpretazione e, nelle condizioni operative considerate, non produce effetti percepibili sul comportamento dinamico.

5.3.3 Tetto Fisico degli Elevoni e Allocazione

Per $\bar{q} \geq \bar{q}_{\min}$ si calcola il tetto fisico di momento erogabile dagli elevoni. La derivazione segue dalla relazione lineare (5.4): il momento massimo in modulo che l'elevatore può produrre si ottiene al limite di deflessione $\delta_{\max} = 30°$, da cui

$$\tau_{x,\text{cap}}^{(\text{elev})} = |G_{\text{roll}}|\, \delta_{\max}, \qquad \tau_{y,\text{cap}}^{(\text{elev})} = |G_{\text{pitch}}|\, \delta_{\max}. \tag{5.11}$$

In pratica si introduce un fattore di sfruttamento $\alpha_{\text{share}} = 0.95$, che riduce di un 5% il tetto operativo rispetto a quello teorico. Il margine, applicato separatamente sulle componenti di pitch e di roll, lascia al singolo elevone un piccolo *buffer* per accomodare la sovrapposizione delle due deflessioni virtuali δ_e e δ_a nella ricombinazione fisica $\delta_{eL} = \delta_e + \delta_a$, $\delta_{eR} = \delta_e - \delta_a$ prima di incontrare la saturazione meccanica finale (Sezione 5.5):

$$\tau_{x,\text{elev,cap}} = \alpha_{\text{share}}\, |G_{\text{roll}}|\, \delta_{\max}, \qquad \tau_{y,\text{elev,cap}} = \alpha_{\text{share}}\, |G_{\text{pitch}}|\, \delta_{\max}. \tag{5.12}$$

L'*allocazione adattiva.* La legge è:

$$\tau_{x,\text{elev}} = \text{sign}(\tau_x)\ \min(|\tau_x|,\ \tau_{x,\text{elev,cap}}) \tag{5.13}$$

$$\tau_{y,\text{elev}} = \text{sign}(\tau_y)\ \min(|\tau_y|,\ \tau_{y,\text{elev,cap}}) \tag{5.14}$$

$$\boldsymbol{\tau}_{xy,\text{elev}} = [\tau_{x,\text{elev}},\ \tau_{y,\text{elev}}]^T \tag{5.15}$$

$$\boldsymbol{\tau}_{xy,\text{mot}} = \boldsymbol{\tau}_{xy} - \boldsymbol{\tau}_{xy,\text{elev}} \tag{5.16}$$

Il significato della legge è il seguente: ciascun asse di rollio o beccheggio è allocato preferenzialmente agli elevoni, fino al loro tetto fisico; il deficit residuo è assegnato ai motori. Il segno della richiesta è preservato.

5.3.4 Conservazione del Momento Totale

L'allocazione (5.13)–(5.16) garantisce per costruzione la conservazione del momento totale comandato:

$$\boxed{\boldsymbol{\tau}_{xy,\text{mot}} + \boldsymbol{\tau}_{xy,\text{elev}} \equiv \boldsymbol{\tau}_{xy} \quad \forall\, V_{\text{air}},\, \forall\, \boldsymbol{\tau}_{xy}} \tag{5.17}$$

Dimostrazione. La proprietà segue immediatamente dalle definizioni. Per la componente di rollio:

$$\tau_{x,\mathrm{mot}} + \tau_{x,\mathrm{elev}} = (\tau_x - \tau_{x,\mathrm{elev}}) + \tau_{x,\mathrm{elev}} = \tau_x \tag{5.18}$$

e analogamente per il pitch. La proprietà è dunque algebrica, non dipende dal valore di V_{air}, $\bar{q}$, $\boldsymbol{\tau}_{xy}$, né dalla saturazione $\min(\cdot, \tau_{\mathrm{cap}})$ applicata a $\tau_{x,\mathrm{elev}}$ in (5.13).

Significato fisico. La conservazione (5.17) significa che il *comando di coppia totale* alla cella di mixing è esattamente uguale alla richiesta proveniente dal controllore di assetto. Naturalmente, ciò non garantisce che il momento *effettivamente prodotto* dal velivolo coincida con la richiesta, a causa delle saturazioni a valle. Tuttavia, il vantaggio della conservazione (5.17) è di eliminare un'altra fonte di errore: l'incoerenza tra il *progetto del controllore di assetto* (che produce $\boldsymbol{\tau}_{\mathrm{des}}$ supponendo che esso venga prodotto integralmente) e l'*allocazione effettiva*.

Confronto con il blending sigmoidale puro. Il vantaggio principale del blending adattivo rispetto all'approccio sigmoidale puro, $(1-\sigma)\,\boldsymbol{\tau} \to \mathrm{mot}$, $\sigma\,\boldsymbol{\tau} \to \mathrm{elev}$, è la garanzia di conservazione del momento espressa da (5.17). Nel blending sigmoidale l'allocazione *ideale* è:

$$\tau^{(\mathrm{sig})}_{\mathrm{x,mot}} = (1 - \sigma(V))\,\tau_x, \qquad \tau^{(\mathrm{sig})}_{\mathrm{x,elev}} = \sigma(V)\,\tau_x \tag{5.19}$$

e algebricamente si ha $\tau^{(\mathrm{sig})}_{\mathrm{x,mot}} + \tau^{(\mathrm{sig})}_{\mathrm{x,elev}} \equiv \tau_x$. Tale identità vale però soltanto a livello di comando, sotto l'ipotesi implicita che ciascun sottosistema sia in grado di produrre realmente la frazione ad esso assegnata. Si tratta di un'assunzione di progetto, non di una proprietà verificata in tempo reale: il peso $\sigma(V)$ è una funzione predeterminata della sola velocità, scelta off-line, e non riflette in alcun modo la capacità istantanea degli elevoni nella specifica condizione di volo. Niente garantisce dunque che la somma delle coppie effettivamente prodotte dai due sottosistemi coincida con τ_x: la conservazione del comando totale è un'aspettativa, non un vincolo strutturale.

In particolare, poiché gli elevoni sono soggetti al limite fisico $|\delta_a| \le \delta_{\max}$, in alcuni regimi (tipicamente a bassa velocità con G_{roll} piccolo, oppure con richieste di coppia elevata) accade che $|\sigma\,\tau_x| > |G_{\mathrm{roll}}|\,\delta_{\max}$ e gli elevoni saturano alla deflessione massima, erogando $|G_{\mathrm{roll}}|\,\delta_{\max}$ anziché $\sigma\,\tau_x$. Il momento totale effettivamente prodotto diventa allora:

$$\tau^{(\mathrm{sig})}_{\mathrm{x,prodotto}} = (1-\sigma)\,\tau_x + \mathrm{sign}(\tau_x)\,|G_{\mathrm{roll}}|\,\delta_{\max} \neq \tau_x, \tag{5.20}$$

e il deficit $\tau_x - \tau^{(\mathrm{sig})}_{\mathrm{x,prodotto}}$ rimane una coppia non realizzata, che il controllore di assetto a monte percepisce come una perturbazione esterna non modellata.

Nel blending adattivo la quota di coppia assegnata agli elevoni è limitata per costruzione alla loro capacità istantanea per asse e il deficit rispetto alla richiesta è trasferito ai motori in (5.16). Il momento totale *comandato* alla cella di mixing coincide dunque sempre con $\boldsymbol{\tau}_{\text{des}}$.

Come anticipato, resta la possibilità che, in regimi di richiesta particolarmente intensa, la coppia trasferita ai motori superi anche la loro capacità fisica (saturazione di Ω^2), producendo un residuo di momento non effettivamente erogato dal veicolo. Il meccanismo di ripartizione è però *auto-complementare* nei due regimi dominanti: a bassa velocità gli elevoni hanno tetto modesto, ma i motori dispongono dell'intera autorità di controllo e coprono il deficit con ampio margine; ad alta velocità il tetto degli elevoni cresce con $\bar{q}$ e assorbe la maggior parte della richiesta, lasciando ai motori solo eventuali residui marginali e liberandoli per la generazione di spinta primaria. Nei regimi di transizione, in cui entrambi i sottosistemi sono attivi, quando uno dei due è prossimo alla propria saturazione l'altro è tipicamente lontano dalla propria, e la cooperazione fra i due garantisce che la coppia totale richiesta sia effettivamente prodotta nella stragrande maggioranza degli scenari operativi.

Limite del modello. La proprietà di conservazione (5.17) si basa sull'ipotesi che gli elevoni e i motori producano effettivamente la coppia comandata. Nella realtà ci sono ritardi di attuazione e effetti non lineari che fanno scostare la coppia prodotta dalla coppia comandata.

5.3.5 Guadagni di Conversione Momento–Deflessione

Una volta calcolata $\boldsymbol{\tau}_{xy,\text{elev}}$, occorre convertirla in deflessioni (δ_e, δ_a) da applicare agli elevoni. Dalla definizione (5.5)–(5.6), le relazioni inverse sono:

$$\delta_e = \frac{\tau_{y,\text{elev}}}{G_{\text{pitch}}} = K_{\text{pitch}}\,\tau_{y,\text{elev}}, \qquad \delta_a = \frac{\tau_{x,\text{elev}}}{G_{\text{roll}}} = K_{\text{roll}}\,\tau_{x,\text{elev}} \tag{5.21}$$

con i guadagni:

$$K_{\text{pitch}} = \frac{1}{G_{\text{pitch}}} = \frac{1}{\bar{q}\,S\,c\,C_{m,\delta_e}}, \qquad K_{\text{roll}} = \frac{1}{G_{\text{roll}}} = \frac{1}{\bar{q}\,S\,b\,C_{l,\delta_a}} \tag{5.22}$$

Si noti che K_{pitch} e K_{roll} sono entrambi *negativi* (poiché C_{m,δ_e} e C_{l,δ_a} lo sono) e *variabili in* V: a velocità più alte $\bar{q}$ è grande e i guadagni sono piccoli (basta una piccola deflessione per generare una coppia significativa); a velocità più basse $\bar{q}$ è piccola e i guadagni crescono in modulo. Il calcolo è effettuato istantaneamente a ogni passo di simulazione, garantendo che la deflessione comandata produca esattamente la coppia richiesta in qualunque condizione operativa.

I segnali K_{pitch} e K_{roll} sono trasmessi dal blocco `fcn_blending` al sotto-sistema di allocazione elevoni tramite due segnali `GoTo/From` (Figura 5.2), nominati rispettivamente `K_elev_pitch` e `K_elev_roll`.

5.4 Allocazione ai Motori

Il sottosistema motore riceve la spinta totale T_{des} e i tre comandi di momento $(\tau_{x,\text{mot}},\, \tau_{y,\text{mot}},\, \tau_z)$. Si noti che il momento di imbardata τ_z è allocato *interamente* ai motori, senza blending: gli elevoni del tailsitter, essendo superfici alari posizionate sul piano di simmetria, non producono momento di imbardata diretto, e il velivolo non dispone di un timone aerodinamico. L'autorità di yaw è quindi affidata in toto alle coppie generate dalla spinta differenziale delle eliche.

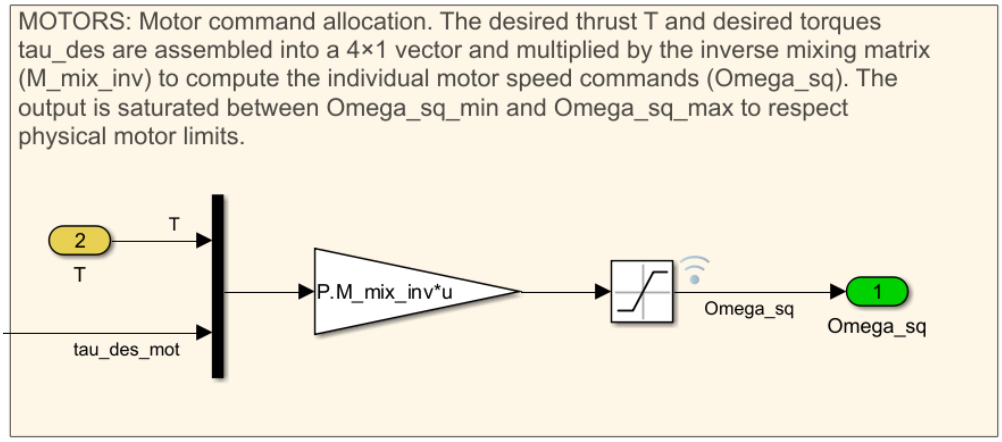


Figura 5.4: Blocco di allocazione ai motori: inversione della matrice di allocazione e saturazione delle velocità angolari al quadrato.

5.4.1 Inversione della Matrice di Allocazione

La relazione tra le velocità angolari al quadrato dei quattro motori e i quattro comandi (spinta totale e tre momenti) è data dalla matrice di mixing $\mathbf{M}_{\text{mix}}$ (Capitolo 4):

$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \mathbf{M}_{\text{mix}}\, \boldsymbol{\Omega}^2 \tag{5.23}$$

Per ottenere le velocità angolari dai comandi desiderati, si inverte la relazione:

$$\boldsymbol{\Omega}^2_{\text{des}} = \mathbf{M}^{-1}_{\text{mix}} \begin{bmatrix} T_{\text{des}} \\ \tau_{x,\text{mot}} \\ \tau_{y,\text{mot}} \\ \tau_z \end{bmatrix} \tag{5.24}$$

L'inversione è ben posta: $\mathbf{M}_{\text{mix}}$ è una matrice 4×4 a rango pieno, proprietà garantita dalla simmetria della configurazione X con quattro motori a sensi di rotazione alternati.

5.4.2 Saturazione delle Velocità Angolari

Le velocità angolari al quadrato ottenute dall'inversione devono essere vincolate ai limiti fisici dei motori:

$$\Omega_i^{2,\text{sat}} = \text{sat}(\Omega^2_{i,\text{des}};\ 0,\ \Omega^2_{\text{max}}), \quad i = 1, \ldots, 4 \tag{5.25}$$

con $\Omega_{\text{max}} = 679\ \text{rad}\,\text{s}^{-1}$ (corrispondente al regime massimo del motore T-Motor MN5008 KV340 a 22.2 V, ossia 6482 RPM). La saturazione è componente per componente: ciascun motore è limitato indipendentemente dagli altri, comportamento coerente con il fatto che ciascun ESC controlla un singolo motore.

L'output di $\boldsymbol{\Omega}^2$ alimenta direttamente il blocco Plant.

5.5 Allocazione agli Elevoni

Il sottosistema elevone riceve $\boldsymbol{\tau}_{xy,\text{elev}}$ dalla `fcn_blending` e i guadagni (K_{pitch}, K_{roll}) tramite `GoTo/From`. Lo schema implementativo è in Figura 5.5.

5.5.1 Calcolo delle Deflessioni Virtuali

Le deflessioni *virtuali*, simmetrica e antisimmetrica, sono calcolate componente per componente:

$$\delta_e = K_{\text{pitch}}\, \tau_{y,\text{elev}} \tag{5.26}$$

$$\delta_a = K_{\text{roll}}\, \tau_{x,\text{elev}} \tag{5.27}$$

Sono dette virtuali perché non corrispondono a deflessioni fisiche di una superficie reale, ma a quote ideali della deflessione complessiva: δ_e è la componente comune alle due superfici (deflessione simmetrica, produce pitch puro), δ_a è la componente differenziale (deflessione antisimmetrica, produce roll puro).

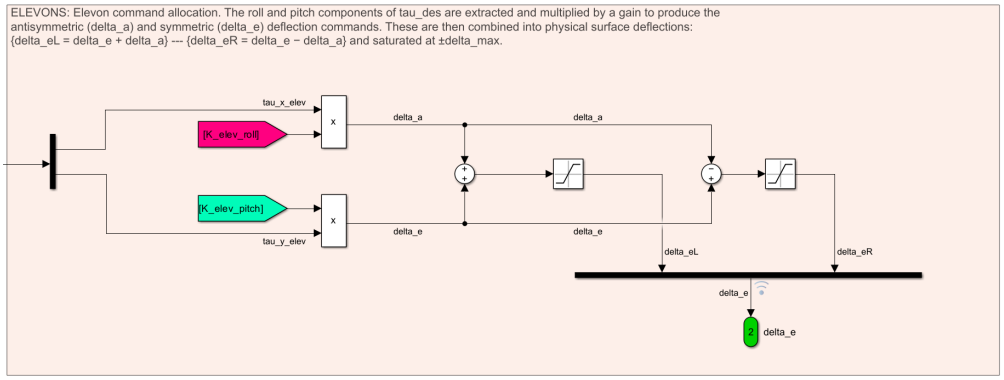


Figura 5.5: Blocco di allocazione elevoni: calcolo delle deflessioni virtuali (δ_e, δ_a), combinazione nelle deflessioni fisiche (δ_{eL}, δ_{eR}) e saturazione ai limiti meccanici.

5.5.2 Combinazione in Deflessioni Fisiche

Le deflessioni fisiche dei due elevoni si ricavano combinando le quote virtuali secondo le convenzioni standard:

$$\delta_{eL} = \delta_e + \delta_a \tag{5.28}$$

$$\delta_{eR} = \delta_e - \delta_a \tag{5.29}$$

ossia, l'elevone sinistro porta la somma delle due quote, quello destro la differenza. Una deflessione simmetrica pura ($\delta_a = 0$) muove le due superfici in modo identico e produce solo pitch; una deflessione antisimmetrica pura ($\delta_e = 0$) muove le due superfici in opposizione e produce solo roll.

5.5.3 Saturazione ai Limiti Meccanici

Le deflessioni fisiche sono infine saturate ai limiti meccanici del servo:

$$\delta_{eL}^{\text{sat}} = \text{sat}(\delta_{eL};\ -\delta_{\max},\ \delta_{\max}), \quad \delta_{eR}^{\text{sat}} = \text{sat}(\delta_{eR};\ -\delta_{\max},\ \delta_{\max}) \tag{5.30}$$

con $\delta_{\max} = 30°$. Si noti che, grazie al margine $\alpha_{\text{share}} = 0.95$ introdotto nel blending (5.12), le deflessioni virtuali $|\delta_e|$ e $|\delta_a|$ sono per costruzione limitate a $\alpha_{\text{share}}\,\delta_{\max}$. Tuttavia la loro somma o differenza può eccedere $\delta_{\max}$ in casi limite, ed è per questo che la saturazione finale è essenziale per la sicurezza meccanica.

L'output $(\delta_{eL},\ \delta_{eR})$ alimenta il blocco Plant, chiudendo l'anello di controllo.

5.6 Riepilogo del Flusso dei Segnali

L'allocazione complessiva può essere sintetizzata come segue:

$$(\boldsymbol{\tau}_{\text{des}},\ T_{\text{des}},\ V_{\text{air}}) \xrightarrow{\texttt{fcn_blending}} (\boldsymbol{\tau}_{xy,\text{mot}},\ \boldsymbol{\tau}_{xy,\text{elev}},\ K_{\text{pitch}},\ K_{\text{roll}}) \tag{5.31}$$

$$(T_{\text{des}},\ \boldsymbol{\tau}_{xy,\text{mot}},\ \tau_z) \xrightarrow{\mathbf{M}_{\text{mix}}^{-1}} \boldsymbol{\Omega}_{\text{des}}^2 \xrightarrow{\text{sat}} \boldsymbol{\Omega}^2 \tag{5.32}$$

$$(\boldsymbol{\tau}_{xy,\text{elev}},\ K_{\text{pitch}},\ K_{\text{roll}}) \xrightarrow{(5.26),(5.27)} (\delta_e,\ \delta_a) \xrightarrow{(5.28),(5.29)} (\delta_{eL},\ \delta_{eR}) \xrightarrow{\text{sat}} (\delta_{eL}^{\text{sat}},\ \delta_{eR}^{\text{sat}}) \tag{5.33}$$

Capitolo 6

Controllo di Assetto

6.1 Introduzione

Il presente capitolo descrive l'architettura di controllo dell'assetto del tailsitter, responsabile della stabilizzazione dell'orientamento del veicolo e dell'inseguimento dei riferimenti di assetto generati dal livello di controllo superiore. Il controllore deve operare in modo continuo su tutto l'inviluppo di volo, dall'hover stazionario alla crociera ad alta velocità, senza ricorso a commutazione modale e, a differenza di precedenti versioni dell'architettura, senza programmazione dei guadagni in funzione della velocità. La copertura uniforme dell'inviluppo è ottenuta grazie a un termine di feed-forward inerziale che compensa anticipatamente la dinamica nominale del plant, lasciando al PI il solo compito di reiezione dei disturbi.

L'architettura adottata è di tipo a cascata: un anello esterno di controllo dell'assetto genera i riferimenti di velocità angolare, che vengono inseguiti da un anello interno di controllo della velocità angolare (rate loop) [7, 14, 15, 22, 25]. Il vantaggio fondamentale di questa struttura è che lo smorzamento del sistema, ottenibile con un termine derivativo sull'errore di assetto, è realizzato dall'anello interno utilizzando direttamente la misura giroscopica della velocità angolare, che è intrinsecamente la derivata temporale dell'assetto. Si evita così il calcolo numerico della derivata, operazione notoriamente sensibile al rumore di misura, ottenendo al contempo un comportamento complessivo equivalente a un controllore PD sull'assetto [2].

La trattazione segue l'ordine del flusso dei segnali: si presenta dapprima la struttura Simulink del sottosistema, poi si descrivono in dettaglio l'errore di assetto in quaternioni, l'anello esterno, l'anello interno, la programmazione dei guadagni e l'analisi della separazione delle bande passanti.

6.2 Architettura del Sottosistema Simulink

Il sottosistema *Attitude Control* in Simulink riflette fedelmente l'architettura a cascata. La Figura 6.1 mostra l'interfaccia esterna del blocco, con i tre ingressi, il quaternione desiderato $\mathbf{q}_{\text{des}}$ (dal blocco di generazione dei comandi), il quaternione corrente $\mathbf{q}$ e la velocità angolare $\boldsymbol{\omega}_B$ (dal plant), e l'uscita, il vettore di coppia desiderata $\boldsymbol{\tau}_{\text{des}}$.

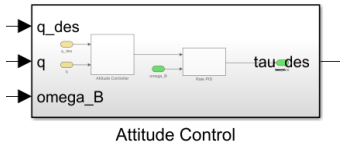


Figura 6.1: Interfaccia del blocco Attitude Control: ingressi di stato e uscita di coppia desiderata.

All'interno del sottosistema (Figura 6.2), la struttura è articolata in due blocchi funzionali disposti in serie:

1. **Attitude Controller:** riceve $\mathbf{q}_{\text{des}}$ e $\mathbf{q}$ e produce il riferimento di velocità angolare $\boldsymbol{\omega}_{\text{des}}$.
2. **Rate PID:** riceve $\boldsymbol{\omega}_{\text{des}}$ dall'anello esterno e $\boldsymbol{\omega}_B$ dal plant, e produce la coppia desiderata $\boldsymbol{\tau}_{\text{des}}$.

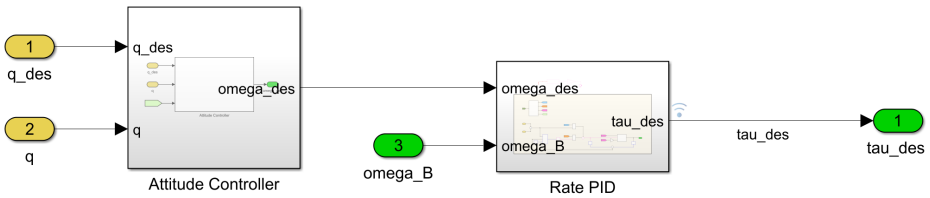


Figura 6.2: Struttura interna del sottosistema Attitude Control: il blocco Attitude Controller genera $\boldsymbol{\omega}_{\text{des}}$, il blocco Rate PID produce la coppia $\boldsymbol{\tau}_{\text{des}}$.

Le sezioni seguenti descrivono in dettaglio la formulazione matematica di ciascun blocco, nell'ordine in cui i segnali vengono elaborati.

6.3 Errore di Assetto in Quaternioni

L'assetto corrente del veicolo è rappresentato dal quaternione unitario $\mathbf{q}$ e l'assetto desiderato dal quaternione $\mathbf{q}_{\text{des}}$, entrambi appartenenti alla sfera unitaria $S^3 \subset \mathbb{R}^4$.

L'errore di assetto è definito come la rotazione che porta dall'orientamento corrente a quello desiderato, espressa nel riferimento del corpo:

$$\mathbf{q}_{\text{err}} = \mathbf{q}^* \otimes \mathbf{q}_{\text{des}} \tag{6.1}$$

dove $\mathbf{q}^* = [q_0,\, -q_1,\, -q_2,\, -q_3]^T$ è il coniugato del quaternione corrente e $\otimes$ denota il prodotto di Hamilton.

Il prodotto di Hamilton tra due quaternioni generici $\mathbf{p} = [p_0,\, p_1,\, p_2,\, p_3]^T$ e $\mathbf{r} = [r_0,\, r_1,\, r_2,\, r_3]^T$ è definito come:

$$\mathbf{p} \otimes \mathbf{r} = \begin{bmatrix} p_0 r_0 - p_1 r_1 - p_2 r_2 - p_3 r_3 \\ p_0 r_1 + p_1 r_0 + p_2 r_3 - p_3 r_2 \\ p_0 r_2 - p_1 r_3 + p_2 r_0 + p_3 r_1 \\ p_0 r_3 + p_1 r_2 - p_2 r_1 + p_3 r_0 \end{bmatrix} \tag{6.2}$$

Sostituendo $\mathbf{p} = \mathbf{q}^*$ e $\mathbf{r} = \mathbf{q}_{\text{des}}$, le componenti dell'errore sono:

$$\mathbf{q}_{\text{err}} = \begin{bmatrix} q_0\, q_{\text{des},0} + q_1\, q_{\text{des},1} + q_2\, q_{\text{des},2} + q_3\, q_{\text{des},3} \\ q_0\, q_{\text{des},1} - q_1\, q_{\text{des},0} - q_2\, q_{\text{des},3} + q_3\, q_{\text{des},2} \\ q_0\, q_{\text{des},2} + q_1\, q_{\text{des},3} - q_2\, q_{\text{des},0} - q_3\, q_{\text{des},1} \\ q_0\, q_{\text{des},3} - q_1\, q_{\text{des},2} + q_2\, q_{\text{des},1} - q_3\, q_{\text{des},0} \end{bmatrix} \tag{6.3}$$

Quando l'assetto corrente coincide con quello desiderato ($\mathbf{q} = \mathbf{q}_{\text{des}}$), si ha $\mathbf{q}_{\text{err}} = [1,\, 0,\, 0,\, 0]^T$ (quaternione identità), e la parte vettoriale dell'errore è nulla.

6.3.1 Selezione del Percorso Più Breve

Poiché i quaternioni $\mathbf{q}$ e $-\mathbf{q}$ rappresentano la medesima rotazione nello spazio tridimensionale, l'errore $\mathbf{q}_{\text{err}}$ ammette un'ambiguità di segno. Per garantire che il controllore generi sempre la rotazione correttiva lungo il percorso geodetico più breve sulla sfera S^3, si introduce un fattore di segno:

$$s = \text{sign}(q_{\text{err},0}) \tag{6.4}$$

dove $\text{sign}(\cdot)$ restituisce $+1$ se l'argomento è non negativo, -1 altrimenti. Moltiplicando la parte vettoriale dell'errore per s, si assicura che la rotazione correttiva sia sempre inferiore a π radianti.

6.4 Anello Esterno: Controllore di Assetto

Il controllore di assetto genera il vettore di velocità angolare desiderata $\boldsymbol{\omega}_{\text{des}}$ mediante una legge proporzionale applicata alla parte vettoriale dell'errore di quaternione:

$$\boldsymbol{\omega}_{\text{des}} = K_{p,\text{att}} \circ \boldsymbol{e}_{\text{att}} \tag{6.5}$$

dove $\circ$ denota il prodotto di Hadamard, $K_{p,\text{att}} = [K_{p,1},\, K_{p,2},\, K_{p,3}]^T$ è il vettore dei guadagni proporzionali e il vettore di errore di assetto è:

$$\boldsymbol{e}_{\text{att}} = 2\, s \begin{bmatrix} q_{\text{err},1} \\ q_{\text{err},2} \\ q_{\text{err},3} \end{bmatrix} \tag{6.6}$$

Il fattore 2 emerge dalla linearizzazione della cinematica dei quaternioni attorno all'equilibrio. Per piccoli angoli di errore $\boldsymbol{\delta\theta} = [\delta\phi,\, \delta\theta,\, \delta\psi]^T$, il quaternione di errore assume la forma approssimata $\mathbf{q}_{\text{err}} \approx [1,\, \delta\phi/2,\, \delta\theta/2,\, \delta\psi/2]^T$, per cui $2\,\mathbf{q}_{\text{err,vec}} \approx \boldsymbol{\delta\theta}$. Il prodotto di Hadamard (anziché una moltiplicazione scalare) consente di assegnare guadagni differenti ai tre assi, accomodando le diverse inerzie e autorità di controllo lungo rollio, beccheggio e imbardata.

6.4.1 Equivalenza con un Controllore PD sull'Assetto

Il controllore di assetto non include un termine derivativo esplicito. Per comprendere perché non sia necessario, e perché non sia necessario nemmeno un termine integrale sull'assetto, è utile analizzare il comportamento complessivo della cascata su un singolo asse.

Il problema del controllore PD diretto. Il minimo necessario per controllare un angolo di assetto θ_k è un'azione proporzionale (che "tira" verso il riferimento) e un'azione derivativa (che "frena" per evitare oscillazioni). Un controllore PD classico applicato direttamente all'angolo produrrebbe:

$$\tau_k = K_p\, e_{\theta,k} + K_d\, \dot{e}_{\theta,k} \tag{6.7}$$

dove $e_{\theta,k} = \theta_{\text{des},k} - \theta_k$. Il termine $\dot{e}_{\theta,k}$ è la derivata temporale dell'errore di assetto, pari alla differenza di velocità angolare $\omega_{\text{des},k} - \omega_k$. In un'implementazione diretta, questa derivata andrebbe calcolata numericamente dal segnale di assetto, amplificando il rumore di misura alle alte frequenze. L'architettura a cascata risolve questo problema in modo strutturale, producendo un comportamento equivalente al PD senza mai derivare il segnale di assetto.

Dimostrazione dell'equivalenza. Si consideri il solo contributo proporzionale dell'anello interno (il termine integrale sarà discusso separatamente). L'anello esterno genera il comando di velocità angolare proporzionale all'errore di assetto:

$$\omega_{\text{des},k} = K_{p,\text{att},k}\, e_{\theta,k} \tag{6.8}$$

L'anello interno produce la coppia proporzionale all'errore di velocità angolare:

$$\tau_k = K_{p,\text{rate},k}\, (\omega_{\text{des},k} - \omega_k) \tag{6.9}$$

Sostituendo (6.8) in (6.9) e riordinando:

$$\tau_k = \underbrace{K_{p,\mathrm{rate},k}\, K_{p,\mathrm{att},k}}_{K_p^{\mathrm{eq}}}\, e_{\theta,k} \;-\; \underbrace{K_{p,\mathrm{rate},k}}_{K_d^{\mathrm{eq}}}\, \omega_k \tag{6.10}$$

Confrontando con (6.7), si riconosce un controllore PD equivalente sull'assetto con $K_p^{\mathrm{eq}} = K_{p,\mathrm{rate}} \cdot K_{p,\mathrm{att}}$ e $K_d^{\mathrm{eq}} = K_{p,\mathrm{rate}}$. Il primo termine (prodotto dei due guadagni proporzionali) genera l'azione proporzionale sull'angolo; il secondo termine (guadagno proporzionale del rate loop) genera l'azione di smorzamento agendo sulla velocità angolare ω_k *misurata direttamente* dal giroscopio, senza alcuna derivazione numerica.

Ruolo del termine integrale del rate loop. L'equazione (6.10) considera solo il contributo proporzionale del rate loop. In realtà, il controllore di velocità angolare include anche un termine integrale $K_{i,\mathrm{rate}} \int \omega_{\mathrm{err}}\, dt$ (Sezione 6.5.2), il cui ruolo è eliminare l'errore di velocità angolare a regime in presenza di coppie di disturbo costanti (vento, asimmetrie costruttive, offset del baricentro). Un controllore PD puro sull'assetto non avrebbe questa capacità: sotto una coppia di disturbo costante, l'angolo raggiungerebbe un valore di equilibrio spostato rispetto al riferimento.

Il termine integrativo sull'assetto non è necessario. Un integrale sull'assetto (che produrrebbe un controllore PID complessivo) non è necessario per due ragioni. In primo luogo, i disturbi costanti nel dominio dell'assetto sono coppie, non angoli, e vengono già compensati dall'integrale del rate loop. In secondo luogo, aggiungere un integrale sull'assetto renderebbe il sistema del terzo ordine (PID su un doppio integratore $\theta \to \omega \to \tau$), più difficile da stabilizzare.

Sintesi. La cascata P (assetto) + PI (rate) produce simultaneamente:

1. un controllore PD equivalente sull'assetto, con smorzamento basato sulla misura giroscopica diretta di ω_k anziché sulla derivata numerica di θ_k;
2. l'annullamento delle coppie di disturbo costanti, grazie al termine integrale del rate loop.

Questo doppio risultato è ottenuto con una struttura minimale e rappresenta il motivo principale per cui l'architettura a cascata è lo standard nel controllo di volo.

6.5 Anello Interno: Controllore di Velocità Angolare

Il controllore di velocità angolare (rate loop) è il livello più interno della cascata e opera con la banda passante più elevata. La Figura 6.3 mostra il dettaglio

implementativo del blocco in Simulink. Si riconoscono tre elementi: il ramo di *feed-forward* (in alto), il ramo *proporzionale* sull'errore $\boldsymbol{\omega}_{\text{err}} = \boldsymbol{\omega}_{\text{des}} - \boldsymbol{\omega}_B$ e il ramo *integrale* con anti-windup condizionale. I tre contributi sono sommati e inviati al blocco di saturazione, con limiti $\pm\boldsymbol{\tau}_{\max}$. I guadagni $K_{p,\text{rate}}$, $K_{i,\text{rate}}$ e i limiti $\boldsymbol{\tau}_{\max}$ sono parametri costanti, non programmati in funzione della velocità.

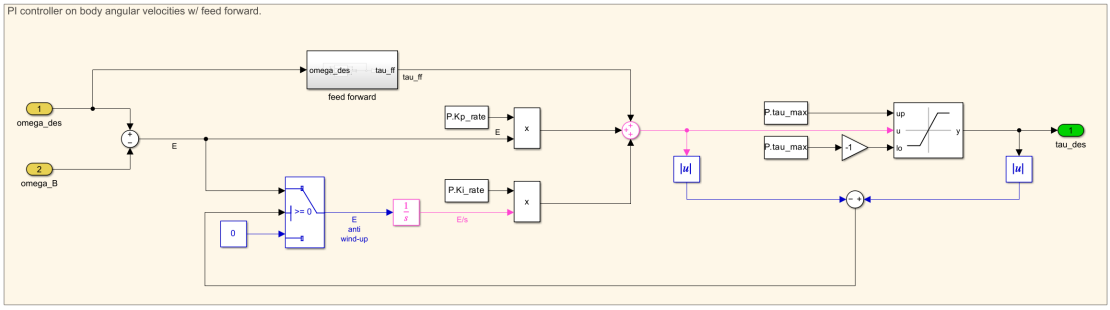


Figura 6.3: Dettaglio del blocco Rate PI con feed-forward: ramo di feed-forward (in alto), ramo proporzionale, ramo integrale con anti-windup condizionale e saturazione (in basso).

Le sottosezioni seguenti formalizzano matematicamente ciascun elemento del blocco.

6.5.1 Errore di Velocità Angolare

L'errore è definito come la differenza tra riferimento e misura:

$$\boldsymbol{\omega}_{\text{err}} = \boldsymbol{\omega}_{\text{des}} - \boldsymbol{\omega}_B \tag{6.11}$$

6.5.2 Legge di Controllo PI con Feed-Forward

Il comando di coppia è la somma di tre contributi: un termine proporzionale, un termine integrale e un termine di feed-forward inerziale:

$$\boldsymbol{\tau} = K_{p,\text{rate}} \circ \boldsymbol{\omega}_{\text{err}} + K_{i,\text{rate}} \circ \boldsymbol{I} + \boldsymbol{\tau}_{\text{ff}} \tag{6.12}$$

dove $K_{p,\text{rate}}$ e $K_{i,\text{rate}}$ sono i vettori di guadagni proporzionali e integrali, $\boldsymbol{I}$ è l'integrale dell'errore:

$$\boldsymbol{I}(t) = \int_0^t \boldsymbol{\omega}_{\text{err}}(\tau)\, d\tau \tag{6.13}$$

e $\boldsymbol{\tau}_{\text{ff}}$ è il contributo di feed-forward, descritto in dettaglio nella Sezione 6.5.3.

Il termine integrale elimina l'errore di velocità angolare a regime in presenza di coppie di disturbo costanti, come discusso nella Sezione 6.4.1. Non è incluso

un termine derivativo poiché, come dimostrato nella medesima sezione, l'azione derivativa sull'assetto è già fornita implicitamente dal termine proporzionale del rate loop.

6.5.3 Feed-Forward Inerziale

L'introduzione del feed-forward sul rate loop ha l'obiettivo di alleggerire il carico del PI fornendo *anticipatamente* la coppia richiesta dalla dinamica nominale del plant. La motivazione fisica è la seguente. Il modello rotazionale del veicolo, linearizzato per piccole velocità angolari (cioè trascurando i termini di accoppiamento giroscopico $\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}$), si scrive:

$$\mathbf{I}\,\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{aero}}(\boldsymbol{\omega},\, V_{\text{air}}) + \boldsymbol{\tau}_{\text{dist}} \tag{6.14}$$

dove $\boldsymbol{\tau}$ è la coppia di controllo prodotta dagli attuatori, $\boldsymbol{\tau}_{\text{aero}}$ è la coppia aerodinamica di smorzamento (tipicamente proporzionale a $-\boldsymbol{\omega}$ tramite $C_{l,p}$, $C_{m,q}$, $C_{n,r}$ moltiplicati per la pressione dinamica), e $\boldsymbol{\tau}_{\text{dist}}$ aggrega tutti i disturbi non modellati (asimmetrie di spinta, errori di costruzione, vento). Linearizzando ulteriormente attorno al punto operativo si ottiene la dinamica nominale del primo ordine:

$$\mathbf{I}\,\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} \tag{6.15}$$

Per inseguire un riferimento $\boldsymbol{\omega}_{\text{des}}(t)$ in modo perfetto in questa dinamica idealizzata, occorrerebbe applicare la coppia $\boldsymbol{\tau} = \mathbf{I}\,\dot{\boldsymbol{\omega}}_{\text{des}}$. Il feed-forward calcola un'approssimazione filtrata di questa quantità.

Schema implementativo. Lo schema, illustrato in Figura 6.4, è costituito da:

1. un filtro passa-basso del primo ordine, applicato componente per componente al riferimento $\boldsymbol{\omega}_{\text{des}}$ (tre filtri identici, uno per ciascun asse di rotazione):

$$\tilde{\omega}_{\text{des},k}(s) = H(s)\,\omega_{\text{des},k}(s), \qquad H(s) = \frac{1}{\tau_{\text{rate,cl}}\,s + 1} \tag{6.16}$$

   con costante di tempo $\tau_{\text{rate,cl}} = 0.05$ s;

2. un nodo sommatore che calcola la differenza $\boldsymbol{\omega}_{\text{des}} - \tilde{\boldsymbol{\omega}}_{\text{des}}$;

3. un guadagno matriciale $\mathbf{I}/\tau_{\text{rate,cl}}$, ovvero un guadagno scalare $I_{kk}/\tau_{\text{rate,cl}}$ applicato componente per componente.

L'uscita complessiva del feed-forward è dunque:

$$\boldsymbol{\tau}_{\text{ff}} = \frac{\mathbf{I}}{\tau_{\text{rate,cl}}} \left(\boldsymbol{\omega}_{\text{des}} - \tilde{\boldsymbol{\omega}}_{\text{des}}\right) \tag{6.17}$$

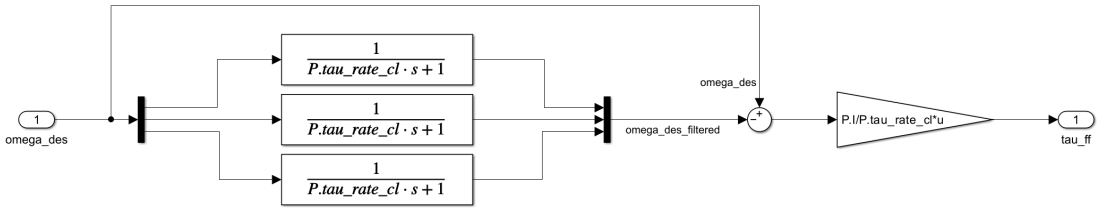


Figura 6.4: Sottoblocco del feed-forward inerziale: filtro a primo ordine su ciascun asse, differenza con il segnale originale e moltiplicazione per $\mathbf{I}/\tau_{\mathrm{rate,cl}}$.

Equivalenza a una derivata filtrata. Si dimostra ora che $\boldsymbol{\tau}_{\text{ff}}$ è una stima filtrata di $\mathbf{I}\,\dot{\boldsymbol{\omega}}_{\text{des}}$. Considerando la singola componente in trasformata di Laplace:

$$\begin{aligned}\omega_{\text{des}}(s) - \tilde{\omega}_{\text{des}}(s) &= \omega_{\text{des}}(s)\left(1 - H(s)\right) \\ &= \omega_{\text{des}}(s)\left(1 - \frac{1}{\tau_{\text{rate,cl}}\,s + 1}\right) \\ &= \omega_{\text{des}}(s)\,\frac{\tau_{\text{rate,cl}}\,s}{\tau_{\text{rate,cl}}\,s + 1}\end{aligned} \tag{6.18}$$

da cui:

$$\tau_{\text{ff},k}(s) = \frac{I_{kk}}{\tau_{\text{rate,cl}}} \cdot \omega_{\text{des},k}(s) \cdot \frac{\tau_{\text{rate,cl}}\,s}{\tau_{\text{rate,cl}}\,s + 1} = I_{kk}\,\frac{s}{\tau_{\text{rate,cl}}\,s + 1}\,\omega_{\text{des},k}(s) \tag{6.19}$$

L'operatore $G_{\text{deriv}}(s) = s/(\tau_{\text{rate,cl}}\,s + 1)$ è un *derivatore filtrato*, anche noto in letteratura come *washout filter*. Le sue proprietà nel dominio della frequenza sono:

- A basse frequenze, $\omega \ll 1/\tau_{\text{rate,cl}}$, il modulo $|G_{\text{deriv}}(j\omega)| \approx \omega$, cioè un'approssimazione asintotica della derivata pura. La fase tende a $+\pi/2$ (anticipo). In questa regione, $\tau_{\text{ff},k} \approx I_{kk}\,\dot{\omega}_{\text{des},k}$, esattamente la coppia richiesta dalla dinamica nominale.
- Alla pulsazione $\omega = 1/\tau_{\text{rate,cl}} = 20\ \text{rad}\,\text{s}^{-1}$ il filtro presenta il caratteristico ginocchio del primo ordine.
- A frequenze elevate, $\omega \gg 1/\tau_{\text{rate,cl}}$, il modulo $|G_{\text{deriv}}(j\omega)| \to 1/\tau_{\text{rate,cl}}$, ovvero il guadagno satura a $1/\tau_{\text{rate,cl}}$. La fase tende a zero (no anticipo). Questa è la differenza essenziale rispetto alla derivata pura s, il cui modulo crescerebbe indefinitamente alle alte frequenze, amplificando il rumore di misura.

Reiezione del disturbo. Il feed-forward agisce solo sul tracking del riferimento, non sulla reiezione dei disturbi. Per disturbi costanti $\boldsymbol{\tau}_{\text{dist}}$ il termine integrale del PI (Sezione 6.5.2) fornisce errore stazionario nullo, indipendentemente dal feed-forward. Per disturbi tempo-varianti, il guadagno proporzionale governa la reiezione, e i valori scelti producono un'attenuazione di 20–30 dB nella banda di interesse ($\omega < 5\ \text{rad}\,\text{s}^{-1}$).

6.5.4 Anti-Windup Condizionale

Quando il comando di coppia raggiunge i limiti di saturazione degli attuatori, l'integrale dell'errore continuerebbe ad accumularsi nonostante l'impossibilità fisica di produrre la coppia richiesta, fenomeno noto come *integral windup* [3, 10]. Il

windup si manifesta come un ritardo nella risposta del controllore al cessare della saturazione, poiché l'integrale accumulato deve prima essere "scaricato".

Per prevenire questo comportamento è implementata una strategia di anti-windup condizionale che ricalca la formulazione adottata nel materiale didattico del corso di Modellazione, Simulazione e Sperimentazione dei Sistemi Aerospaziali del Politecnico di Torino [10]: l'aggiornamento dell'integrale è congelato quando il comando di coppia è saturo:

$$\boldsymbol{I}(t+\Delta t) = \begin{cases} \boldsymbol{I}(t) + \boldsymbol{\omega}_{\text{err}}\,\Delta t & \text{se } |\tau_k| < \tau_{\max,k} \quad \forall\, k \\ \boldsymbol{I}(t) & \text{se } |\tau_k| \geq \tau_{\max,k} \quad \text{per l'asse } k \end{cases} \tag{6.20}$$

dove la condizione è verificata asse per asse ($k = x,\, y,\, z$), consentendo il congelamento selettivo dell'integrale su ciascun asse indipendentemente dagli altri. In Figura 6.3, questa logica è realizzata confrontando il valore assoluto della coppia prima e dopo il blocco di saturazione: se i due valori coincidono, la saturazione non è attiva e l'errore viene integrato normalmente; se differiscono, la saturazione sta limitando l'uscita e lo switch porta a zero l'ingresso dell'integratore. Poiché i segnali sono vettoriali, l'operazione è eseguita elemento per elemento, consentendo il congelamento selettivo dell'integrale su ciascun asse.

6.5.5 Saturazione

Il comando di coppia è saturato componente per componente ai valori massimi degli attuatori:

$$\tau_{\text{sat},k} = \text{sat}(\tau_k;\ -\tau_{\max,k},\ \tau_{\max,k}) = \max\Big(-\tau_{\max,k},\ \min(\tau_k,\ \tau_{\max,k})\Big) \tag{6.21}$$

Il vettore $\boldsymbol{\tau}_{\text{sat}} = [\tau_{\text{sat},x},\, \tau_{\text{sat},y},\, \tau_{\text{sat},z}]^T$ costituisce l'uscita del sottosistema di controllo d'assetto e alimenta il blocco di allocazione dei controlli (Capitolo 5).

6.6 Valori dei Guadagni

L'architettura di controllo adotta un set di guadagni *unificato*, valido sull'intero inviluppo di volo, senza programmazione in funzione della velocità aerodinamica.

I valori, riportati in Tabella 6.1, sono stati determinati tramite una procedura iterativa di taratura per tentativi (*trial and error*), condotta direttamente in simulazione ad anello chiuso. I parametri sono stati aggiustati progressivamente fino a ottenere una risposta transitoria e una robustezza ritenute soddisfacenti sull'intero inviluppo di volo, inclusa la manovra critica di transizione. In particolare, anche i limiti di saturazione $\boldsymbol{\tau}_{\max}$ sono stati fissati per questa via: valori eccessivamente elevati o eccessivamente conservativi alterano la dinamica effettiva del veicolo,

in alcuni casi al punto da rendere non controllabili determinate manovre, come appunto la transizione.

Tabella 6.1: Guadagni del controllore di assetto e del rate loop. Valori unificati su tutto l'inviluppo di volo.

Parametro	**Rollio** (x)	**Beccheggio** (y)	**Imbardata** (z)
$K_{p,\mathrm{att}}$ [$\mathrm{rad\,s^{-1}}$]	5.0	3.5	3.0
$K_{p,\mathrm{rate}}$ [Nm/(rad/s)]	10	12	12
$K_{i,\mathrm{rate}}$ [$\mathrm{N\,m\,rad^{-1}}$]	5	6	6
$\tau_{\max}$ [N m]	11.2	11.2	5.0
$\tau_{\mathrm{rate,cl}}$ [s]	0.05	0.05	0.05

I valori così ottenuti sono stati successivamente sottoposti a una verifica *a posteriori*. La Sezione 6.7 mostra che essi soddisfano la condizione di separazione delle bande passanti richiesta dall'architettura a cascata, condizione che legittima la progettazione indipendente dei due anelli e conferma l'adeguatezza del set di guadagni adottato.

6.7 Separazione delle Bande Passanti

L'architettura a cascata richiede una netta separazione tra le bande passanti dei due anelli affinché ciascuno possa essere progettato in modo indipendente. Questa sezione ricava le bande passanti dei due anelli mediante analisi nel dominio di Laplace e ne quantifica il rapporto.

La *banda passante* di un anello di controllo è la frequenza massima (espressa in $\mathrm{rad\,s^{-1}}$) alla quale l'anello è in grado di inseguire fedelmente un riferimento sinusoidale. Oltre questa frequenza, il segnale di riferimento varia troppo rapidamente rispetto alla capacità di reazione dell'anello, e l'inseguimento degrada.

6.7.1 Banda Passante del Rate Loop

Si consideri un singolo asse k del rate loop. Il controllore è un PI con feed-forward, ma ai fini del calcolo della banda passante in modalità senza FF, il termine integrale K_i/s può essere trascurato a frequenze elevate, come segue. La banda passante è per definizione una proprietà alle alte frequenze (è la frequenza alla quale il sistema inizia a non inseguire più il riferimento), e a frequenze elevate il contributo proporzionale domina su quello integrale: per $\omega \gg K_i/K_p$, il modulo del termine integrale $|K_i/(j\omega)|$ diventa trascurabile rispetto a K_p. A titolo di verifica numerica, sull'asse di rollio si ha $K_i/K_p = 5/10 = 0.5\,\mathrm{rad\,s^{-1}}$, mentre la banda passante

risultante (*senza* feed-forward) sarà dell'ordine di $30\,\mathrm{rad\,s^{-1}}$, quasi due ordini di grandezza superiore, a conferma che l'approssimazione è lecita.

Quindi, considerando il solo termine proporzionale, il controllore produce la coppia $\tau_k = K_{p,\mathrm{rate},k}\,(\omega_{\mathrm{des},k} - \omega_k)$ e il plant è la dinamica rotazionale $I_{kk}\,\dot{\omega}_k = \tau_k$. Passando nel dominio di Laplace:

$$I_{kk}\,s\,\Omega_k(s) = K_{p,\mathrm{rate},k}\left(\Omega_{\mathrm{des},k}(s) - \Omega_k(s)\right) \tag{6.22}$$

Raccogliendo $\Omega_k(s)$:

$$\Omega_k(s)\left(I_{kk}\,s + K_{p,\mathrm{rate},k}\right) = K_{p,\mathrm{rate},k}\,\Omega_{\mathrm{des},k}(s) \tag{6.23}$$

La funzione di trasferimento ad anello chiuso è dunque:

$$\frac{\Omega_k(s)}{\Omega_{\mathrm{des},k}(s)} = \frac{K_{p,\mathrm{rate},k}}{I_{kk}\,s + K_{p,\mathrm{rate},k}} = \frac{1}{\dfrac{I_{kk}}{K_{p,\mathrm{rate},k}}\,s + 1} = \frac{1}{\tau_{\mathrm{rate},k}\,s + 1} \tag{6.24}$$

Questo è un sistema del primo ordine con costante di tempo $\tau_{\mathrm{rate},k} = I_{kk}/K_{p,\mathrm{rate},k}$. La banda passante di un sistema del primo ordine è l'inverso della costante di tempo:

$$\omega_{n,\mathrm{rate},k} = \frac{1}{\tau_{\mathrm{rate},k}} = \frac{K_{p,\mathrm{rate},k}}{I_{kk}} \tag{6.25}$$

Questa relazione mostra che, a parità di guadagno, un asse con inerzia maggiore ha una banda passante inferiore: il controllore deve dare un contributo maggiore per ottenere la stessa rapidità di risposta.

6.7.2 Banda Passante dell'Attitude Loop

Per l'anello esterno si assume che il rate loop sia sufficientemente veloce da poter essere approssimato come un guadagno unitario, ovvero $\omega_k \approx \omega_{\mathrm{des},k}$. La cinematica dell'assetto, linearizzata attorno all'equilibrio, si riduce a $\dot{\theta}_k = \omega_k$, cioè un integratore puro. Con il controllore proporzionale $\omega_{\mathrm{des},k} = K_{p,\mathrm{att},k}\,(\theta_{\mathrm{des},k} - \theta_k)$, nel dominio di Laplace:

$$s\,\Theta_k(s) = K_{p,\mathrm{att},k}\left(\Theta_{\mathrm{des},k}(s) - \Theta_k(s)\right) \tag{6.26}$$

La funzione di trasferimento ad anello chiuso è:

$$\frac{\Theta_k(s)}{\Theta_{\mathrm{des},k}(s)} = \frac{K_{p,\mathrm{att},k}}{s + K_{p,\mathrm{att},k}} = \frac{1}{\dfrac{1}{K_{p,\mathrm{att},k}}\,s + 1} \tag{6.27}$$

Nuovamente un sistema del primo ordine, con costante di tempo $\tau_{\mathrm{att},k} = 1/K_{p,\mathrm{att},k}$ e dunque banda passante:

$$\omega_{n,\mathrm{att},k} = K_{p,\mathrm{att},k} \tag{6.28}$$

6.7.3 Calcolo Numerico e Rapporto di Separazione

Utilizzando i momenti d'inerzia del veicolo ($I_{xx} = 0.326\,\mathrm{kg\,m^2}$, $I_{yy} = 0.500\,\mathrm{kg\,m^2}$, $I_{zz} = 0.500\,\mathrm{kg\,m^2}$) e i guadagni unificati dalla Tabella 6.1, le bande passanti risultano:

Rate loop senza feed-forward. Considerando il solo termine proporzionale del PI:

$$\omega_{n,\mathrm{rate},x}^{(P)} = \frac{K_{p,\mathrm{rate},x}}{I_{xx}} = \frac{10}{0.326} = 30.7\,\mathrm{rad\,s^{-1}} \tag{6.29}$$

$$\omega_{n,\mathrm{rate},y}^{(P)} = \frac{K_{p,\mathrm{rate},y}}{I_{yy}} = \frac{12}{0.500} = 24.0\,\mathrm{rad\,s^{-1}} \tag{6.30}$$

$$\omega_{n,\mathrm{rate},z}^{(P)} = \frac{K_{p,\mathrm{rate},z}}{I_{zz}} = \frac{12}{0.500} = 24.0\,\mathrm{rad\,s^{-1}} \tag{6.31}$$

Rate loop con feed-forward. L'introduzione del feed-forward modifica la dinamica nominale: la coppia totale comandata si compone di τ_{ff} (che insegue $I\dot{\omega}_{\mathrm{des}}$) più il contributo PI di correzione. Trascurando il PI per l'analisi della banda passante (è un correttore rispetto a una dinamica nominale già compensata), e nel limite di feed-forward esatto, la banda passante target del rate loop è imposta dal filtro stesso:

$$\omega_{n,\mathrm{rate}}^{(\mathrm{FF})} = \frac{1}{\tau_{\mathrm{rate,cl}}} = 20\,\mathrm{rad\,s^{-1}} \quad \text{(comune ai 3 assi)} \tag{6.32}$$

Il valore $\omega_{n,\mathrm{rate}}^{(\mathrm{FF})}$ è il riferimento di progetto: il rate loop è in grado di inseguire variazioni di ω_{des} con dinamica del primo ordine a costante di tempo $\tau_{\mathrm{rate,cl}} = 0.05$ s.

Attitude loop.

$$\omega_{n,\mathrm{att},x} = 5.0, \qquad \omega_{n,\mathrm{att},y} = 3.5, \qquad \omega_{n,\mathrm{att},z} = 3.0\,\mathrm{rad\,s^{-1}} \tag{6.33}$$

Rapporto di separazione.

$$r_k = \frac{\omega_{n,\mathrm{rate}}^{(\mathrm{FF})}}{\omega_{n,\mathrm{att},k}} \tag{6.34}$$

I valori numerici sono riassunti nella Tabella 6.2.

In letteratura, un rapporto di separazione $r_k \geq 3$–5 è considerato sufficiente a garantire che l'anello interno insegua fedelmente i riferimenti dell'anello esterno senza introdurre ritardi significativi. Dal punto di vista dell'anello esterno, la risposta dell'anello interno appare quasi istantanea, validando l'ipotesi di progetto indipendente dei due anelli. I rapporti ottenuti, $r_k \in [4.0,\ 6.7]$, risultano al di sopra della soglia minima e validano la separazione delle bande passanti.

Tabella 6.2: Bande passanti e rapporto di separazione (set di guadagni unificati con feed-forward).

Grandezza	**Rollio** (x)	**Beccheggio** (y)	**Imbardata** (z)
$\omega_{n,\mathrm{rate}}^{(\mathrm{FF})}$ $[\mathrm{rad\,s^{-1}}]$	20.0	20.0	20.0
$\omega_{n,\mathrm{att}}$ $[\mathrm{rad\,s^{-1}}]$	5.0	3.5	3.0
Rapporto r_k	4.0	5.7	6.7

Capitolo 7

Generazione del Comando di Assetto

7.1 Introduzione

Il sottosistema *Attitude Command*, oggetto del presente capitolo, è il livello del sistema di controllo che traduce la richiesta cinematica proveniente dal controllore di posizione (Capitolo 8) in un comando di assetto destinato al controllore d'assetto a cascata (Capitolo 6). I suoi ingressi sono il vettore di forza desiderata $\boldsymbol{F}_{\mathrm{tot},I}$ nel frame inerziale, la velocità inerziale corrente e quella desiderata, il quaternione corrente $\mathbf{q}$ e l'angolo di rotta desiderato ψ_{des}; i suoi output sono il quaternione di assetto desiderato $\mathbf{q}_{\mathrm{des}}$ e il modulo della spinta desiderata T_{des}.

La formulazione adottata è interamente unificata e priva di commutazione modale: non vi è una macchina di stato che selezioni un controllore fra più alternative. Una singola funzione MATLAB (`fcn_coordinated_flight`) gestisce in modo continuo l'intero inviluppo, dall'hover stazionario alla crociera ad alta velocità, generando il riferimento di assetto come *interpolazione sferica* (SLERP) tra due soluzioni candidate: una di hover, costruita geometricamente, e una di crociera, ottenuta come soluzione di trim del volo coordinato. Il peso fra le due è governato da una sigmoide sulla velocità di riferimento, e l'assenza di commutazione si traduce in un riferimento di assetto continuo su tutto l'inviluppo di volo.

L'architettura è ispirata al lavoro di Ritz e D'Andrea sulle configurazioni *flying-wing* [37]; il presente capitolo ne specializza il formalismo al body frame adottato in questa tesi, esplicitando in modo rigoroso le convenzioni di asse e la struttura della cascata di rotazioni che porta dall'asse di velocità inerziale all'assetto di crociera trimmato. Per il formalismo generale di rappresentazione dell'assetto e l'uso dei quaternioni nelle leggi di controllo si rimanda inoltre ai classici testi di dinamica e controllo di assetto [21, 45, 50].

7.2 Architettura del Sottosistema Simulink

La Figura 7.1 mostra l'interfaccia esterna del sottosistema *Attitude Command* come appare al livello superiore del modello.

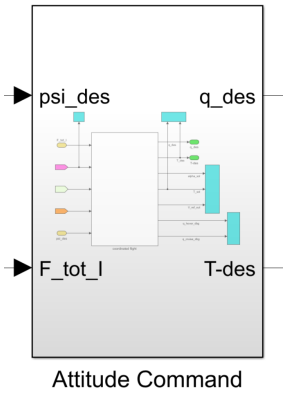


Figura 7.1: Interfaccia esterna del blocco *Attitude Command*.

Aprendo il sottosistema (Figura 7.2), si osserva che il cuore funzionale è costituito da un'unica MATLAB Function, `fcn_coordinated_flight`, che riceve cinque segnali di ingresso ($\boldsymbol{F}_{\mathrm{tot},I}$, $\boldsymbol{v}_I$, $\boldsymbol{v}_{\mathrm{des},I}$, $\mathbf{q}$, ψ_{des}) e produce sette segnali di uscita: i due output operativi ($\mathbf{q}_{\mathrm{des}}$, T_{des}) e cinque *debug ports* (α_{sol}, T_{sol}, $V_{\mathrm{ref,out}}$, $\mathbf{q}_{\mathrm{hover,dbg}}$, $\mathbf{q}_{\mathrm{cruise,dbg}}$) utili per il *logging* e per la diagnostica del solver di trim.

Il vantaggio di concentrare l'intero algoritmo in una singola funzione MATLAB, anziché distribuirlo su una rete di blocchi Simulink elementari, è dato dalla formulazione algoritmica: la costruzione in cascata di rotazioni, il solver di Newton–Raphson, lo SLERP e la sigmoide di blending si esprimono in modo più naturale in codice sequenziale che in schema a blocchi.

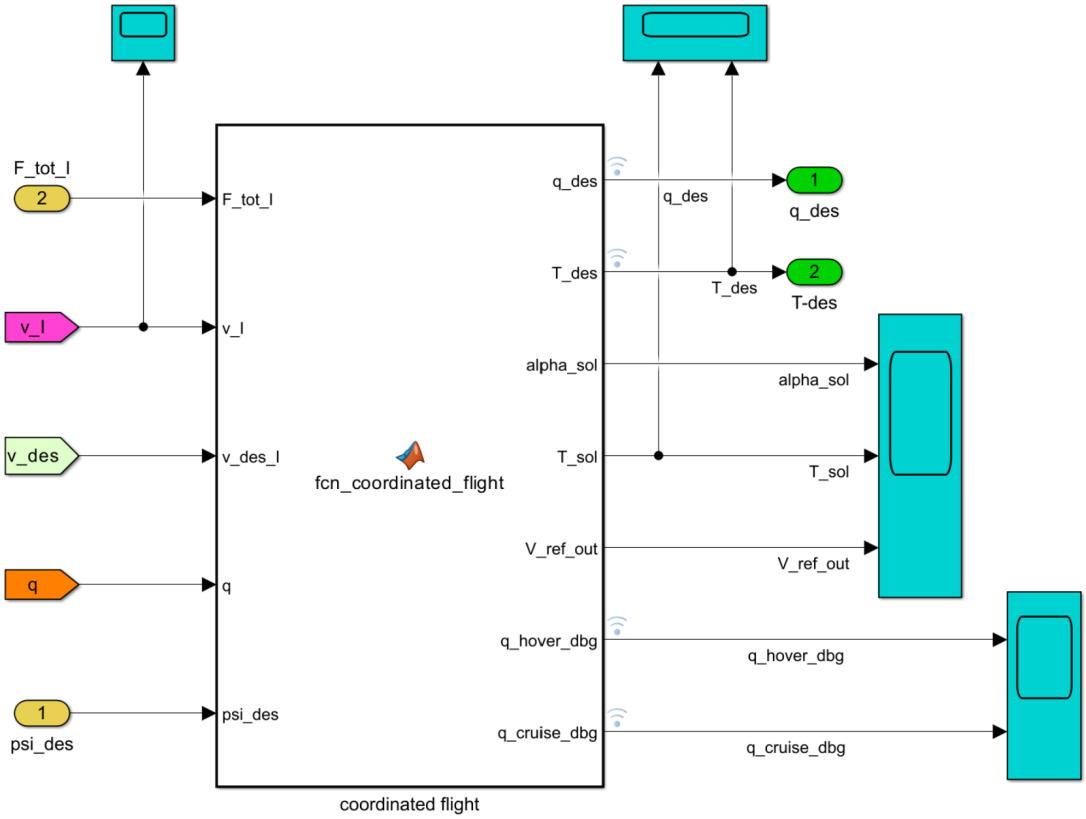


Figura 7.2: Struttura interna del sottosistema *Attitude Command*: una singola MATLAB Function `fcn_coordinated_flight` aggrega l'intera logica di generazione del comando di assetto e ne espone gli output principali e di debug.

7.3 Convenzioni e Mappatura degli Assi

L'algoritmo di volo coordinato richiede una notazione esplicita per la sequenza di rotazioni che porta dalla terna inerziale alla terna del corpo. Si adottano le convenzioni seguenti, già introdotte nel Capitolo 4 e riassunte qui per chiarezza:

- Frame inerziale $\mathcal{F}_I = (e_N, e_E, e_D)$ in convenzione NED.
- Frame del corpo $\mathcal{F}_B = (\hat{b}_x, \hat{b}_y, \hat{b}_z)$, con $\hat{b}_x$ direzione di spinta primaria (*forward*), $\hat{b}_y$ lungo l'ala destra, $\hat{b}_z$ verso il basso del corpo.
- Matrice di rotazione $\mathbf{R} = \mathbf{R}(\mathbf{q})$ che mappa *vettori espressi nel frame del corpo* nelle loro componenti nel frame inerziale: $\boldsymbol{v}_I = \mathbf{R}\,\boldsymbol{v}_B$. Equivalentemente, le *colonne* di $\mathbf{R}$ sono le componenti di $\hat{b}_x$, $\hat{b}_y$, $\hat{b}_z$ nel frame inerziale.

7.4 Calcolo della Velocità di Riferimento

Il primo passaggio dell'algoritmo è la determinazione della velocità scalare di riferimento V_{ref}, che governa l'intero blending fra hover e crociera. Detta $\boldsymbol{v}_{\text{des},I}$ la velocità desiderata generata dal pianificatore di missione, si calcola

$$V_{\text{ref}} = \|\boldsymbol{v}_{\text{des},I}\|, \quad V_{\text{real}} = \|\boldsymbol{v}_I\|. \tag{7.1}$$

Per garantire la robustezza del blending in condizioni in cui il pianificatore richieda esplicitamente la fermata ($V_{\text{ref}} \to 0$) ma il velivolo si trovi ancora in volo veloce per inerzia, si applica la regola di sostituzione

$$V_{\text{ref}} \leftarrow V_{\text{real}} \quad \text{se} \quad V_{\text{ref}} < 1\ \text{m}\,\text{s}^{-1} \ \wedge\ V_{\text{real}} > 5\ \text{m}\,\text{s}^{-1}. \tag{7.2}$$

La giustificazione fisica è la seguente: una richiesta di hover ($V_{\text{des}} \approx 0$) imposta dal pianificatore non può essere soddisfatta istantaneamente se il velivolo si trova a $V_{\text{real}} \gg V_{\text{hover,ready}}$; in tale frangente è preferibile far inseguire all'attitude command la velocità reale.

7.5 Costruzione dell'Assetto di Hover

L'assetto di hover si costruisce direttamente in forma geometrica, sfruttando il fatto che, in regime stazionario verticale, l'asse di spinta $\hat{b}_x$ del velivolo deve coincidere con la direzione della forza richiesta $\boldsymbol{F}_{\text{tot},I}$, mentre l'asse $\hat{b}_y$ deve giacere nel piano orizzontale ed essere coerente con la rotta desiderata ψ_{des}.

7.5.1 Determinazione dell'Asse di Spinta

Si pone

$$\hat{b}_x^{(h)} = \begin{cases} \boldsymbol{F}_{\mathrm{tot},I} \,/\, \|\boldsymbol{F}_{\mathrm{tot},I}\| & \text{se } \|\boldsymbol{F}_{\mathrm{tot},I}\| > F_{\mathrm{th,min}}, \\ -e_D = (0,0,-1)^T & \text{altrimenti.} \end{cases} \tag{7.3}$$

La seconda alternativa, attivata quando la forza richiesta è prossima a zero, corrisponde all'hover puro a peso bilanciato: l'asse di spinta è diretto verso l'alto (in NED, $-e_D$). La spinta nominale assume in questo caso il valore $T_h = mg$; altrimenti $T_h = \|\boldsymbol{F}_{\mathrm{tot},I}\|$.

7.5.2 Determinazione dell'Asse Laterale via Yaw Desiderato

L'asse laterale $\hat{b}_y^{(h)}$ è scelto compatibilmente con la rotta ψ_{des} richiesta dal pianificatore. Si parte da un candidato nel piano orizzontale

$$\hat{b}_y^{(h),0} = \big(\cos(\psi_{\mathrm{des}} + \pi/2),\ \sin(\psi_{\mathrm{des}} + \pi/2),\ 0\big)^T, \tag{7.4}$$

che è ruotato di $\pi/2$ rispetto al vettore di rotta. Il candidato è proiettato ortogonalmente al piano normale a $\hat{b}_x^{(h)}$:

$$\hat{b}_y^{(h)} = \frac{\hat{b}_y^{(h),0} - (\hat{b}_y^{(h),0} \cdot \hat{b}_x^{(h)})\, \hat{b}_x^{(h)}}{\|\hat{b}_y^{(h),0} - (\hat{b}_y^{(h),0} \cdot \hat{b}_x^{(h)})\, \hat{b}_x^{(h)}\|}. \tag{7.5}$$

Se la norma del numeratore risulta inferiore a una soglia di 10^{-6} (caso di $\hat{b}_x^{(h)}$ allineato con il candidato laterale), si ripiega su $\hat{b}_y^{(h)} = (0,1,0)^T$ come scelta di default. Il terzo asse si ottiene per chiusura della terna destrorsa:

$$\hat{b}_z^{(h)} = \hat{b}_x^{(h)} \times \hat{b}_y^{(h)}. \tag{7.6}$$

Si compone infine la matrice di rotazione di hover $\mathbf{R}_{\mathrm{hover}} = [\hat{b}_x^{(h)} \mid \hat{b}_y^{(h)} \mid \hat{b}_z^{(h)}]$ e la si converte in quaternione $\mathbf{q}_{\mathrm{hover}}$ tramite l'algoritmo di Shepperd, descritto nella Sezione 7.9.

7.6 Costruzione dell'Assetto di Crociera: Cascata di Rotazioni e Trim

In regime di crociera, l'assetto desiderato non è univocamente determinato dalla sola direzione di $\boldsymbol{F}_{\mathrm{tot},I}$, perché il velivolo deve *anche* produrre portanza aerodinamica per soddisfare la richiesta verticale. La determinazione dell'assetto di crociera richiede

dunque la soluzione di un problema di trim non lineare in cui spinta T e angolo di attacco α sono incognite, mentre i restanti gradi di libertà dell'orientamento sono fissati geometricamente dalla direzione di volo e dalla direzione della forza richiesta.

Si segue il formalismo di Ritz e D'Andrea, articolato in tre rotazioni elementari $\mathbf{R}_a$, $\mathbf{R}_b$, $\mathbf{R}_c$ composte in cascata.

7.6.1 Rotazione $\mathbf{R}_a$: Allineamento alla Direzione di Volo

Sia $\hat{v} = \boldsymbol{v}_{\text{ref}}/V_{\text{ref}}$ il versore della velocità di riferimento, definito quando $V_{\text{ref}} > 1.5$ m/s e posto a e_x altrimenti. La rotazione $\mathbf{R}_a$ porta l'asse e_x inerziale sulla direzione $\hat{v}$. L'angolo è

$$\sigma_a = \arccos(\hat{v} \cdot e_x) = \arccos(\hat{v}_1), \tag{7.7}$$

e l'asse di rotazione si calcola come prodotto vettoriale $e_x \times \hat{v}$, le cui componenti sono $(0, -\hat{v}_3, \hat{v}_2)^T$. Detto $\hat{n}_a = (0, -\hat{v}_3, \hat{v}_2)^T / \|(0, -\hat{v}_3, \hat{v}_2)\|$ il versore di rotazione, $\mathbf{R}_a$ è la matrice di rotazione asse–angolo

$$\mathbf{R}_a = \mathbf{R}(\hat{n}_a, \sigma_a) = \mathbf{I} - \sin\sigma_a \, [\hat{n}_a]_\times + (1 - \cos\sigma_a) \, [\hat{n}_a]_\times^2, \tag{7.8}$$

dove $[\hat{n}_a]_\times$ è la matrice di prodotto vettoriale di $\hat{n}_a$. Nel caso degenere in cui $\|(0, -\hat{v}_3, \hat{v}_2)\|$ è inferiore a 10^{-4}, ovvero quando $\hat{v}$ è allineato con $\pm e_x$, si applica la correzione: $\mathbf{R}_a = \mathbf{I}$ se $\hat{v}_1 > 0$, altrimenti $\mathbf{R}_a = \mathbf{R}(e_y, \pi)$.

La forza inerziale viene quindi espressa nel frame ruotato da $\mathbf{R}_a$:

$$\boldsymbol{F}_a = \mathbf{R}_a \, \boldsymbol{F}_{\text{tot},I}. \tag{7.9}$$

7.6.2 Rotazione $\mathbf{R}_b$: Allineamento del Vettore di Spinta

Nel frame intermedio determinato da $\mathbf{R}_a$, l'asse longitudinale del veicolo è ora coerente con la direzione di volo, ma la componente verticale di $\boldsymbol{F}_a$ può ancora avere una componente $F_{a,y}$ in direzione laterale. La rotazione $\mathbf{R}_b$ attorno al nuovo asse x (σ_b indica l'angolo di rollio nel frame intermedio) viene scelta in modo da annullare $F_{a,y}$, portando la spinta a giacere interamente nel piano x–z del nuovo frame:

$$\sigma_b = \operatorname{atan2}(F_{a,y}, \, -F_{a,z}). \tag{7.10}$$

La forza si trasforma in

$$\boldsymbol{F}_{ab} = \mathbf{R}_b \, \boldsymbol{F}_a, \qquad \mathbf{R}_b = \mathbf{R}(e_x, \sigma_b), \tag{7.11}$$

e ha ora componenti $\boldsymbol{F}_{ab} = (F_{ab,x}, \, 0, \, F_{ab,z})^T$. Questa condizione è ciò che rende il successivo problema di trim un sistema 2×2 e non 3×2.

Se la norma planare $\sqrt{F_{a,y}^2 + F_{a,z}^2}$ è inferiore alla soglia $F_{\text{th,min}} = 0.5$ N, la rotazione σ_b è indefinita e si ripiega su $\sigma_b = \psi_{\text{des}}$, come scelta di default che preserva la rotta desiderata.

7.6.3 Rotazione $\mathbf{R}_c$: Angolo di Attacco di Trim

Resta da determinare l'angolo di attacco α tale che, nel frame $\mathbf{R}_b\,\mathbf{R}_a$, l'equilibrio fra spinta, peso e azioni aerodinamiche sia soddisfatto. Le due equazioni di equilibrio lungo gli assi x e z del frame intermedio si scrivono

$$T - D\cos\alpha + L\sin\alpha = F_{ab,x}, \tag{7.12}$$

$$-D\sin\alpha - L\cos\alpha = F_{ab,z}, \tag{7.13}$$

dove $L = \bar{q}\,S\,C_L(\alpha)$ e $D = \bar{q}\,S\,C_D(\alpha)$ sono portanza e drag dimensionali calcolati alla pressione dinamica $\bar{q} = \frac{1}{2}\rho V_{\text{dyn}}^2$, con $V_{\text{dyn}} = \max(V_{\text{ref}}, 5\,\text{m}\,\text{s}^{-1})$ per regolarizzare il regime a bassa velocità.

Il sistema (7.12)–(7.13) è non lineare nelle incognite (T, α) a causa della dipendenza non lineare di C_L e C_D da α, mediata dalle reti neurali addestrate sul profilo NACA 0012. La sua soluzione è ottenuta iterativamente tramite il metodo di Newton–Raphson descritto nella Sezione 7.7.

A soluzione raggiunta, la rotazione $\mathbf{R}_c$ si applica come pitch attorno al nuovo asse y:

$$\mathbf{R}_c = \mathbf{R}(e_y, \alpha). \tag{7.14}$$

7.6.4 Composizione della Cascata e Estrazione del Quaternione

La matrice di rotazione desiderata in regime di crociera è

$$\mathbf{R}_{\text{des}} = \mathbf{R}_c\,\mathbf{R}_b\,\mathbf{R}_a. \tag{7.15}$$

Tale matrice è applicata in senso *passivo*: trasforma le componenti di un vettore dal frame inerziale a quello del corpo, ovvero $\boldsymbol{v}_B = \mathbf{R}_{\text{des}}\,\boldsymbol{v}_I$. La matrice che mappa i vettori del corpo nel frame inerziale è dunque la sua trasposta: $\mathbf{R}_{IB} = \mathbf{R}_{\text{des}}^T$.

Le colonne di $\mathbf{R}_{IB}$ rappresentano i versori del *body* $\hat{b}_x^{(b)}$, $\hat{b}_y^{(b)}$, $\hat{b}_z^{(b)}$ espressi nel frame inerziale. Per garantire l'ortonormalità rigorosa di tale terna in presenza di piccoli errori numerici dovuti all'accumulo delle operazioni di matrice, si applica una procedura di ri-ortonormalizzazione di Gram–Schmidt sulle prime due colonne:

$$\hat{b}_x^{(c)} \leftarrow \hat{b}_x^{(c)} / \|\hat{b}_x^{(c)}\|, \tag{7.16}$$

$$\hat{b}_y^{(c)} \leftarrow \hat{b}_y^{(c)} - (\hat{b}_y^{(c)} \cdot \hat{b}_x^{(c)})\,\hat{b}_x^{(c)}, \tag{7.17}$$

$$\hat{b}_y^{(c)} \leftarrow \hat{b}_y^{(c)} / \|\hat{b}_y^{(c)}\|, \tag{7.18}$$

$$\hat{b}_z^{(c)} = \hat{b}_x^{(c)} \times \hat{b}_y^{(c)}. \tag{7.19}$$

La matrice di rotazione ri-ortonormalizzata $\mathbf{R}_{\text{cruise}} = [\hat{b}_x^{(c)} \mid \hat{b}_y^{(c)} \mid \hat{b}_z^{(c)}]$ è convertita in quaternione $\mathbf{q}_{\text{cruise}}$ tramite Shepperd. La spinta di crociera è $T_{\text{cruise}} = T$ ottenuta dal trim.

7.7 Solver di Trim: Newton–Raphson Smorzato

Il sistema (7.12)–(7.13) è risolto a ogni passo del controllore mediante iterazione di Newton–Raphson con smorzamento adattivo e *warm-start* dal valore della soluzione al passo precedente, memorizzata in variabili persistenti α_{prev}, T_{prev} [33, 36].

7.7.1 Residui e Matrice Jacobiana

Definiti i residui

$$r_1(T, \alpha) = T - D(\alpha)\cos\alpha + L(\alpha)\sin\alpha - F_{ab,x}, \tag{7.20}$$

$$r_2(T, \alpha) = -D(\alpha)\sin\alpha - L(\alpha)\cos\alpha - F_{ab,z}, \tag{7.21}$$

la matrice Jacobiana ha la forma

$$\mathbf{J}(T, \alpha) = \begin{bmatrix} \frac{\partial r_1}{\partial T} & \frac{\partial r_1}{\partial \alpha} \\ \frac{\partial r_2}{\partial T} & \frac{\partial r_2}{\partial \alpha} \end{bmatrix} = \begin{bmatrix} 1 & J_{12} \\ 0 & J_{22} \end{bmatrix}, \tag{7.22}$$

con

$$J_{12} = D\sin\alpha + L\cos\alpha + \bar{q}S\,(-\dot{C}_D\cos\alpha + \dot{C}_L\sin\alpha), \tag{7.23}$$

$$J_{22} = -D\cos\alpha + L\sin\alpha + \bar{q}S\,(-\dot{C}_D\sin\alpha - \dot{C}_L\cos\alpha), \tag{7.24}$$

dove $\dot{C}_L = \partial C_L/\partial\alpha$ e $\dot{C}_D = \partial C_D/\partial\alpha$ sono le derivate analitiche delle reti neurali aerodinamiche (eq. 2.11). La triangolarità superiore di $\mathbf{J}$ è una conseguenza diretta della linearità del residuo r_1 in T (a parità di α, r_1 varia di ΔT esattamente come ΔT) e dell'indipendenza di r_2 da T.

7.7.2 Schema Iterativo

A ogni iterazione $k = 1, \ldots, 20$:

1. Si valutano $C_L(\alpha_k)$, $C_D(\alpha_k)$, $\dot{C}_L(\alpha_k)$, $\dot{C}_D(\alpha_k)$ tramite le reti neurali e si calcolano $L = \bar{q}SC_L$, $D = \bar{q}SC_D$.

2. Si calcolano i residui r_1, r_2 e gli elementi della Jacobiana.

3. Si calcola il determinante $\det\mathbf{J} = J_{22}$ (per la triangolarità) e si esce dal ciclo se $|J_{22}| < 10^{-6}$ (Jacobiana singolare).

4. Si calcola il passo di Newton:

$$\Delta T = \frac{J_{22}\,r_1 - J_{12}\,r_2}{\det\mathbf{J}}, \quad \Delta\alpha = \frac{-J_{21}\,r_1 + r_2}{\det\mathbf{J}} = \frac{r_2}{J_{22}}. \tag{7.25}$$

5. Si applica lo *smorzamento*:

$$\lambda_k = 0.4 + 0.3\,\min(k/5, 1) \;\in\; [0.4,\, 0.7], \tag{7.26}$$

ovvero λ parte da 0.4 e cresce linearmente a 0.7 dopo cinque iterazioni. La scelta di un iniziale smorzamento più marcato protegge dall'*overshoot* quando il *warm-start* è lontano dalla soluzione.

6. Si aggiornano le incognite e si saturano nei limiti operativi:

$$T \in [-T_{\max},\; +4T_{\max}], \qquad \alpha \in [-\pi/2,\; +\pi/2]. \tag{7.27}$$

L'intervallo di α è esteso fino a $\pm\pi/2$ per consentire al solver di operare correttamente nei regimi di transizione, in cui α percorre l'intero quadrante. La rete neurale aerodinamica resta valida anche in tale intervallo perché è stata addestrata su dati NACA 0012 fino a $\pm 130°$.

7. Si verifica il criterio di arresto: $|\Delta T| < 0.01$ N e $|\Delta\alpha| < 10^{-4}$ rad. Se soddisfatto, si esce.

7.7.3 Filtro Passa-Basso sull'Uscita

Per attenuare le oscillazioni numeriche residue che potrebbero propagarsi al controllore d'assetto in seguito a variazioni rapide del riferimento di forza, l'uscita del solver è filtrata da un primo ordine discreto a costante di tempo $\tau_{\text{filter}} = 0.05\,\text{s}$:

$$\alpha_{\text{out}} = (1-\mu)\,\alpha_{\text{prev}} + \mu\,\alpha, \qquad T_{\text{out}} = (1-\mu)\,T_{\text{prev}} + \mu\,T, \tag{7.28}$$

con $\mu = dt_{\text{ctrl}}/(\tau_{\text{filter}} + dt_{\text{ctrl}})$. Il valore filtrato è anche quello che viene memorizzato nelle variabili persistenti per il *warm-start* al passo successivo, chiudendo il loop.

7.8 Blending Hover/Cruise e Output Finale

L'output finale del sottosistema si ottiene combinando i due quaternioni candidati $\mathbf{q}_{\text{hover}}$ e $\mathbf{q}_{\text{cruise}}$ tramite *interpolazione sferica* (SLERP) pesata da una sigmoide della velocità di riferimento.

7.8.1 Sigmoide di Pesatura

Il peso del termine di crociera è

$$\sigma(V_{\text{ref}}) = \frac{1}{1 + \exp\!\big(-k_\sigma\,(V_{\text{ref}} - V_{\text{trans}})\big)} \;\in\; (0,1), \tag{7.29}$$

con $k_\sigma = 0.3$ e $V_{\text{trans}} = 32\,\text{m}\,\text{s}^{-1}$. A $V_{\text{ref}} = 0$ si ottiene $\sigma \approx 0.00007$ (assetto quasi-integralmente di hover); a $V_{\text{ref}} = 32\,\text{m}\,\text{s}^{-1}$ si ha $\sigma = 0.5$ (transizione bilanciata); a $V_{\text{ref}} = 61\,\text{m}\,\text{s}^{-1}$ si ottiene $\sigma \approx 0.999$ (assetto integralmente di crociera). La sigmoide è $\mathcal{C}^\infty$, e la sua derivata $k_\sigma\,\sigma(1-\sigma)$ è massima a $V_{\text{ref}} = V_{\text{trans}}$, dove vale $k_\sigma/4 = 0.075$ s/m. Tale derivata massima quantifica la rapidità della transizione e influenza direttamente il valore di ω_{des} richiesto al controllore d'assetto.

7.8.2 Interpolazione SLERP

Applicata al blending hover–crociera, la *spherical linear interpolation* assume la forma [41]

$$\text{slerp}(\mathbf{q}_{\text{hover}}, \mathbf{q}_{\text{cruise}}, \sigma) = \frac{\sin\Big((1-\sigma)\theta\Big)}{\sin\theta}\,\mathbf{q}_{\text{hover}} + \frac{\sin(\sigma\,\theta)}{\sin\theta}\,\mathbf{q}_{\text{cruise}}, \tag{7.30}$$

con $\theta = \arccos(\mathbf{q}_{\text{hover}} \cdot \mathbf{q}_{\text{cruise}})$ angolo geodetico tra i due quaternioni sulla sfera unitaria S^3. Per evitare la lunga percorrenza geodetica si normalizza il segno di $\mathbf{q}_{\text{cruise}}$ in modo che $\mathbf{q}_{\text{hover}} \cdot \mathbf{q}_{\text{cruise}} \geq 0$ prima dell'interpolazione. Per $\theta < 10^{-6}$ si ripiega su interpolazione lineare con ri-normalizzazione, per evitare la divisione per $\sin\theta$ prossimo a zero. Infine, per garantire continuità del segnale di assetto al controllore d'assetto (che è sensibile al segno del quaternione), si allinea anche il segno di $\mathbf{q}_{\text{out}}$ a quello del quaternione *corrente* $\mathbf{q}$.

7.8.3 Output di Spinta

Il modulo di spinta finale è una combinazione lineare delle due spinte candidate, con lo stesso peso del SLERP:

$$T_{\text{des}} = (1-\sigma)\,T_h + \sigma\,T_{\text{cruise}}. \tag{7.31}$$

Si noti che, a differenza del quaternione, il blending della spinta è una combinazione lineare nello spazio reale, perché lo spazio dei valori di spinta è $\mathbb{R}$ e non una varietà non lineare.

7.9 Conversione Matrice di Rotazione → Quaternione (Algoritmo di Shepperd)

La conversione di una matrice di rotazione $\mathbf{R} \in SO(3)$ in quaternione unitario è implementata tramite l'algoritmo di Shepperd [40], scelto per la sua robustezza numerica rispetto alla formula basata sull'estrazione dei tre elementi della diagonale.

L'idea dell'algoritmo è di scegliere fra quattro espressioni candidate per la componente del quaternione il cui modulo è massimo (e dunque numericamente più

affidabile da calcolare in radice quadrata), sulla base del segno e del valore di $\mathrm{tr}(\mathbf{R})$ e dei tre elementi diagonali. Detto $s_0 = 2\sqrt{1+\mathrm{tr}(\mathbf{R})}$, il caso $\mathrm{tr}(\mathbf{R}) > 0$ produce

$$q_0 = s_0/4, \tag{7.32}$$
$$q_1 = (R_{32} - R_{23})/s_0, \tag{7.33}$$
$$q_2 = (R_{13} - R_{31})/s_0, \tag{7.34}$$
$$q_3 = (R_{21} - R_{12})/s_0. \tag{7.35}$$

Gli altri tre rami (selezionati quando il massimo elemento diagonale è R_{11}, R_{22}, R_{33} rispettivamente) sono strutturalmente analoghi. Le matrici fornite in ingresso all'algoritmo ($\mathbf{R}_{\text{hover}}$ e $\mathbf{R}_{\text{cruise}}$) sono DCM body-to-inertial, le cui colonne sono i versori del corpo espressi nel frame inerziale; il quaternione restituito è dunque nella stessa convenzione di $\mathbf{R}(\mathbf{q})$ adottata in tutta la tesi. Una volta calcolato il quaternione, lo si normalizza e si allinea il suo segno a quello del quaternione corrente $\mathbf{q}$, per evitare discontinuità di segno fra passi consecutivi del simulatore.

7.10 Sintesi del Flusso e Riepilogo delle Variabili Persistenti

Per chiarezza, si riporta in forma sequenziale la sequenza di operazioni compiute dalla funzione `fcn_coordinated_flight` a ogni invocazione (Tabella 7.1).

Tabella 7.1: Sequenza di operazioni del generatore del comando di assetto.

Passo	**Operazione**
1	Calcolo di V_{ref} con regola di fall-back V_{real}.
2	Costruzione di $\mathbf{R}_{\text{hover}}$ da $\boldsymbol{F}_{\text{tot},I}$ e ψ_{des}, estrazione di $\mathbf{q}_{\text{hover}}$ via Shepperd.
3	Costruzione della cascata di crociera $\mathbf{R}_a$, $\mathbf{R}_b$.
4	Newton–Raphson smorzato a 20 iterazioni con *warm-start* su $(\alpha_{\text{prev}}, T_{\text{prev}})$, smorzamento adattivo $\lambda_k \in [0.4, 0.7]$ e bound $\alpha \in [-\pi/2, \pi/2]$, $T \in [-T_{\max}, 4T_{\max}]$.
5	Filtro passa-basso del primo ordine su (α, T) con $\tau_{\text{filter}} = 0.05$ s.
6	Aggiornamento di α_{prev}, T_{prev} per il passo successivo.
7	Costruzione di $\mathbf{R}_c = \mathbf{R}(e_y, \alpha)$, composizione cascata $\mathbf{R}_{\text{des}} = \mathbf{R}_c\,\mathbf{R}_b\,\mathbf{R}_a$, riortonormalizzazione di $\mathbf{R}_{\text{cruise}} = \mathbf{R}_{\text{des}}^T$, estrazione di $\mathbf{q}_{\text{cruise}}$ via Shepperd.
8	Calcolo di $\sigma(V_{\text{ref}})$, SLERP fra $\mathbf{q}_{\text{hover}}$ e $\mathbf{q}_{\text{cruise}}$, blending lineare di T_h e T_{cruise}, allineamento di segno con $\mathbf{q}$.
9	Output $\mathbf{q}_{\text{des}}$, T_{des} verso i blocchi *Attitude Control* e *Control Allocation*.

Le variabili persistenti del solver sono soltanto due, α_{prev} e T_{prev}, inizializzate alla prima invocazione rispettivamente a 4.5° e 5 N. Tutte le altre variabili sono ricalcolate a ogni passo.

Capitolo 8

Controllo di Posizione

8.1 Introduzione

Il controllore di posizione costituisce il livello più esterno della cascata del sistema di controllo del veicolo. Esso riceve dal sottosistema di *Guidance* (Capitolo 9) la terna di riferimenti di posizione $\boldsymbol{p}_{\text{des}}$, velocità $\boldsymbol{v}_{\text{des}}$ e accelerazione in feed-forward $\boldsymbol{a}_{\text{ff}}$, tutti espressi nel frame inerziale NED, e produce in uscita il vettore di forza inerziale totale $\boldsymbol{F}_{\text{tot},I}$ che alimenta il blocco di generazione del comando di assetto (Capitolo 7).

L'architettura adottata è quella di un controllore proporzionale - integrale - derivativo (PID) sull'errore di posizione nel frame inerziale, integrato con un termine di feed-forward sull'accelerazione di riferimento [20, 25, 30, 37, 47]. L'azione integrale agisce sull'errore di posizione e ha la funzione di annullare l'errore a regime in presenza di disturbi costanti.

8.2 Architettura del Sottosistema Simulink

La Figura 8.1 mostra l'interfaccia esterna del sottosistema *Position Control* al livello superiore del modello Simulink.

Entrando nel sottosistema (Figura 8.2), si distingue una struttura a quattro rami sommati a formare l'accelerazione desiderata, una saturazione componente per componente, una compensazione esplicita dell'accelerazione gravitazionale e una moltiplicazione finale per la massa, che produce il vettore di forza in uscita.

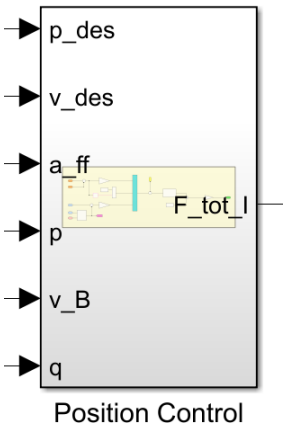


Figura 8.1: Interfaccia esterna del blocco *Position Control*.

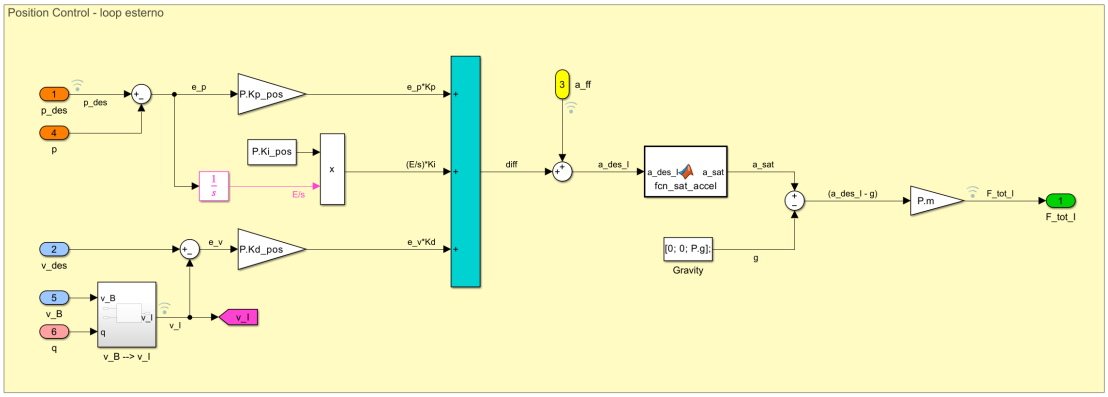


Figura 8.2: Struttura interna del blocco *Position Control*: anello PID nel frame inerziale con feed-forward, saturazione di accelerazione, compensazione di gravità e forza aero-propulsiva richiesta.

8.3 Formulazione Matematica

8.3.1 Errori e Conversione di Frame

Si denotino con $\boldsymbol{p} \in \mathbb{R}^3$ la posizione corrente del veicolo nel frame inerziale NED e con $\boldsymbol{v}_B \in \mathbb{R}^3$ la sua velocità espressa nel frame body. La velocità inerziale impiegata dall'anello di controllo si ottiene ruotando $\boldsymbol{v}_B$ tramite la matrice di assetto $\mathbf{R}(\boldsymbol{q})$ associata al quaternione corrente,

$$\boldsymbol{v}_I = \mathbf{R}(\boldsymbol{q})\, \boldsymbol{v}_B, \tag{8.1}$$

operazione svolta dal sottosistema `v_B --> v_I`. Gli errori di posizione e di velocità, entrambi nel frame inerziale, sono allora

$$\boldsymbol{e}_p = \boldsymbol{p}_{\text{des}} - \boldsymbol{p}, \qquad \boldsymbol{e}_v = \boldsymbol{v}_{\text{des}} - \boldsymbol{v}_I. \tag{8.2}$$

Per costruzione vale $\dot{\boldsymbol{e}}_p = \boldsymbol{e}_v$, identità usata nell'analisi che segue.

8.3.2 Anello PID nel Frame Inerziale

L'accelerazione desiderata nel frame inerziale è la somma di un termine proporzionale all'errore di posizione, di un termine integrale sullo stesso errore, di un termine derivativo e di un termine di feed-forward sull'accelerazione di traiettoria:

$$\boldsymbol{a}_{\text{des},I} = \boldsymbol{k}_{p,\text{pos}} \odot \boldsymbol{e}_p + \boldsymbol{k}_{i,\text{pos}} \odot \int_0^t \boldsymbol{e}_p \, \mathrm{d}\tau + \boldsymbol{k}_{d,\text{pos}} \odot \boldsymbol{e}_v + \boldsymbol{a}_{\text{ff}}, \tag{8.3}$$

dove $\odot$ denota il prodotto componente per componente: i guadagni sono *vettori* e i blocchi Gain del modello sono configurati in modalità element-wise. Il secondo termine è l'azione integrale, che accumula l'errore di posizione; il suo effetto a regime è discusso nella Sezione 8.4. Il terzo termine costituisce l'azione derivativa del controllore: essendo $\boldsymbol{e}_v = \dot{\boldsymbol{e}}_p$ per riferimenti regolari, $\boldsymbol{k}_{d,\text{pos}} \odot \boldsymbol{e}_v$ equivale a $\boldsymbol{k}_{d,\text{pos}} \odot \dot{\boldsymbol{e}}_p$. La sua realizzazione mediante retroazione diretta della velocità stimata, anziché per derivazione numerica di $\boldsymbol{e}_p$, evita l'amplificazione del rumore.

I guadagni sono definiti come segue:

$$\begin{aligned} \boldsymbol{k}_{p,\text{pos}} &= [\, K_{p,h},\ K_{p,h},\ K_{p,v} \,]^\top, \\ \boldsymbol{k}_{i,\text{pos}} &= [\, K_{i,h},\ K_{i,h},\ K_{i,v} \,]^\top, \\ \boldsymbol{k}_{d,\text{pos}} &= [\, K_{d,h},\ K_{d,h},\ K_{d,v} \,]^\top, \end{aligned} \tag{8.4}$$

con $K_{p,h} = 1.2$, $K_{d,h} = 1.3$, $K_{i,h} = 0.08$ sugli assi orizzontali e $K_{p,v} = 3.5$, $K_{d,v} = 4.5$, $K_{i,v} = 0.3$ su quello verticale.

In assenza di feed-forward e per riferimento costante, la (8.3) fornisce, asse per asse, una dinamica dell'errore che, derivata una volta per eliminare il termine integrale, assume la forma del terzo ordine

$$\dddot{\boldsymbol{e}}_p + \boldsymbol{k}_{d,\mathrm{pos}} \odot \ddot{\boldsymbol{e}}_p + \boldsymbol{k}_{p,\mathrm{pos}} \odot \dot{\boldsymbol{e}}_p + \boldsymbol{k}_{i,\mathrm{pos}} \odot \boldsymbol{e}_p = \boldsymbol{0}, \tag{8.5}$$

con equazione caratteristica $s^3 + K_d\, s^2 + K_p\, s + K_i = 0$ su ciascun asse.

8.3.3 Termine di Feed-Forward

Assumendo che l'anello interno realizzi fedelmente l'accelerazione comandata ($\boldsymbol{a} \approx \boldsymbol{a}_{\mathrm{des},I}$) e ricordando $\ddot{\boldsymbol{e}}_p = \dot{\boldsymbol{v}}_{\mathrm{des}} - \boldsymbol{a}$, la sostituzione della (8.3) fornisce la dinamica forzata dell'errore

$$\ddot{\boldsymbol{e}}_p + \boldsymbol{k}_{d,\mathrm{pos}} \odot \dot{\boldsymbol{e}}_p + \boldsymbol{k}_{p,\mathrm{pos}} \odot \boldsymbol{e}_p + \boldsymbol{k}_{i,\mathrm{pos}} \odot \int_0^t \boldsymbol{e}_p \,\mathrm{d}\tau = \dot{\boldsymbol{v}}_{\mathrm{des}} - \boldsymbol{a}_{\mathrm{ff}}. \tag{8.6}$$

Il membro di destra è il residuo che il feed-forward non trasferisce in modo anticipativo. Se il pianificatore fornisce $\boldsymbol{a}_{\mathrm{ff}} = \dot{\boldsymbol{v}}_{\mathrm{des}}$ (accelerazione nominale, derivata seconda dei polinomi settici di traiettoria), il residuo si annulla e l'errore converge a zero su qualunque segmento. Anche qualora il residuo si mantenga su un valore costante non nullo, l'azione integrale ne annulla l'effetto a regime.

8.3.4 Saturazione di Accelerazione

Prima della conversione in forza, $\boldsymbol{a}_{\mathrm{des},I}$ è limitata componente per componente dalla MATLAB Function `fcn_sat_accel`, con limiti $\pm 15\,\mathrm{m\,s^{-2}}$ sugli assi orizzontali e $\pm 13\,\mathrm{m\,s^{-2}}$ su quello verticale.

8.3.5 Compensazione di Gravità e Generazione della Forza

L'ultimo stadio converte l'accelerazione saturata $\boldsymbol{a}_{\mathrm{sat}}$ nel vettore di forza richiesto. Il bilancio di Newton nel frame inerziale,

$$m\,\boldsymbol{a} = \boldsymbol{F}_{\mathrm{tot},I} + m\,\boldsymbol{g}, \qquad \boldsymbol{g} = [\,0,\ 0,\ g\,]^{\top}, \tag{8.7}$$

con $\boldsymbol{g}$ accelerazione gravitazionale (componente positiva lungo D), risolto rispetto alla forza imponendo $\boldsymbol{a} = \boldsymbol{a}_{\mathrm{sat}}$ fornisce

$$\boxed{\boldsymbol{F}_{\mathrm{tot},I} = m\,(\boldsymbol{a}_{\mathrm{sat}} - \boldsymbol{g})} \tag{8.8}$$

operazione realizzata nel modello dal nodo che sottrae il vettore `[0; 0; P.g]` (blocco *Gravity*) e dal prodotto per `P.m`. La compensazione del peso è dunque esplicita e locale: il termine $-m\boldsymbol{g}$ rappresenta la forza di solo sostentamento. Una verifica

immediata si ha in hover: con $\boldsymbol{a}_{\text{sat}} = \mathbf{0}$ risulta $\boldsymbol{F}_{\text{tot},I} = -m\boldsymbol{g}$, forza di modulo mg diretta verso l'alto che bilancia il peso, laddove la formulazione $m\,\boldsymbol{a}_{\text{sat}}$ darebbe forza nulla. Il vettore $\boldsymbol{F}_{\text{tot},I}$, già comprensivo della compensazione gravitazionale, è passato al generatore del comando di assetto (Capitolo 7), che ne ricerca la realizzazione in assetto e spinta.

8.4 Proprietà di Stabilità e Convergenza

Essendo i guadagni puramente diagonali (prodotto element-wise), l'anello PID è disaccoppiato fra gli assi inerziali: l'equazione caratteristica $s^3+K_d\,s^2+K_p\,s+K_i = 0$ della (8.5) si specializza in un polinomio distinto per gli assi orizzontali e per quello verticale, e la stabilità si analizza separatamente su ciascuno. Sostituendo i guadagni si ottengono le due cubiche

$$\text{assi orizzontali:} \quad s^3 + 1.3\,s^2 + 1.2\,s + 0.08 = 0, \tag{8.9}$$

$$\text{asse verticale:} \quad s^3 + 4.5\,s^2 + 3.5\,s + 0.3 = 0. \tag{8.10}$$

Le radici della prima sono una coppia di poli complessi coniugati $s_{1,2} \approx -0.61 \pm j\,0.86$ s^{-1} (sottosmorzati, $\zeta_h \approx 0.58$, sovraelongazione $\approx 11\%$) e un polo reale $s_3 \approx -0.072$ s^{-1}; le radici della seconda sono tre poli reali, $s_1 \approx -0.87$ s^{-1}, $s_2 \approx -3.53$ s^{-1} e $s_3 \approx -0.098$ s^{-1}. Avendo calcolato esplicitamente le radici, la stabilità si verifica per ispezione diretta: tutte hanno parte reale negativa, dunque la dinamica dell'errore è asintoticamente stabile su ogni asse e, per disaccoppiamento, lo è l'intero anello esterno.

I tre poli di ciascun asse agiscono su due scale temporali distinte. La coppia complessa, sugli assi orizzontali, e i poli a -0.87 e -3.53 s^{-1}, su quello verticale, governano il transitorio di inseguimento, con costanti di tempo dell'ordine del secondo. Il terzo polo, reale e lento ($s_3 \approx -0.072$ s^{-1} sugli assi orizzontali, -0.098 s^{-1} su quello verticale), è il modo aggiunto dall'azione integrale: la sua costante di tempo, dell'ordine di da 10 s a 14 s, fissa la scala temporale, volutamente lenta, su cui vengono annullati gli errori a regime prodotti da disturbi costanti.

8.5 Sintesi

Il controllore di posizione è un anello esterno PID nel frame inerziale, arricchito di feed-forward e privo di override modali. L'azione integrale sull'errore di posizione garantisce errore di posizione nullo a regime anche in presenza di disturbi costanti. La sua uscita, $\boldsymbol{F}_{\text{tot},I} = m\,(\boldsymbol{a}_{\text{sat}} - \boldsymbol{g})$ è la forza aero-propulsiva netta necessaria a imporre l'accelerazione desiderata in presenza del peso, ed è passata al generatore del comando di assetto senza ulteriore compensazione gravitazionale.

Capitolo 9

Guida e Navigazione

9.1 Introduzione

Il sistema di guida e navigazione costituisce il livello più esterno dell'architettura di controllo e ha il compito di trasformare una *specifica di missione* di alto livello, espressa come una sequenza di *leg* con waypoint di rotta, quote di crociera e siti operativi, in un riferimento di traiettoria continuo $\big(\boldsymbol{p}_{\mathrm{des}}(t),\, \boldsymbol{v}_{\mathrm{des}}(t),\, \boldsymbol{a}_{\mathrm{ff}}(t),\, \psi_{\mathrm{des}}(t)\big)$ destinato al controllore di posizione (Capitolo 8). A differenza dei livelli interni, il *guidance* non manipola direttamente lo stato dinamico del veicolo: genera segnali di riferimento in funzione del tempo trascorso, della posizione corrente e dello stato logico della missione.

Il livello deve conciliare almeno le seguenti esigenze. La prima è che si vuole rappresentare la missione da un numero modesto di scalari (waypoint, quote, velocità nominali, durate di hover), affinché la riconfigurazione fra campagne di simulazione resti rapida. La seconda è che il riferimento prodotto deve possedere una regolarità sufficiente a non costringere il controllore di posizione a compensare discontinuità di accelerazione, che si tradurrebbero in salti del vettore di forza desiderata e, a cascata, in transitori bruschi sul comando di assetto.

La risposta a queste esigenze è una macchina a stati finita le cui transizioni sono governate sia da eventi temporali (esaurimento della durata pianificata di una primitiva) sia da eventi spaziali (raggiungimento del segmento residuo di decelerazione). Ogni primitiva genera localmente un riferimento polinomiale, e i raccordi fra primitive contigue sono costruiti in modo da preservare la continuità delle derivate fino all'ordine richiesto, secondo quanto formalizzato nella Sezione 9.5.

9.2 Architettura del Sottosistema Simulink

La Figura 9.1 mostra l'interfaccia esterna del sottosistema *Guidance* al livello superiore del modello.

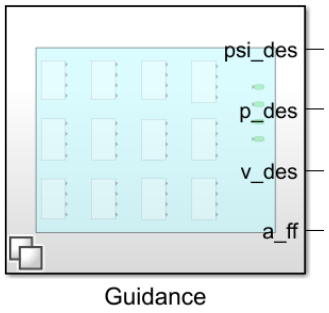


Figura 9.1: Interfaccia esterna del sottosistema *Guidance*: i riferimenti di traiettoria ($\boldsymbol{p}_{\text{des}}$, $\boldsymbol{v}_{\text{des}}$, $\boldsymbol{a}_{\text{ff}}$, ψ_{des}) sono prodotti in funzione del tempo, dello stato logico della missione e della specifica di missione `mission_spec`. L'identificatore di fase corrente (`flight_mode`) è esposto come output e impiegato altrove nel modello per il logging e la visualizzazione per fase.

Internamente, il sottosistema è realizzato come una singola MATLAB Function (`nav_mode8_delivery_mission`) che, a ogni passo di simulazione, riceve in ingresso il tempo corrente, la posizione e la velocità del veicolo e la struttura di missione, e produce in uscita i riferimenti sopra elencati. Lo stato logico della macchina è mantenuto fra invocazioni in due variabili persistenti: la struttura di missione M, copiata una sola volta all'inizializzazione, e la struttura di stato S, che raccoglie la primitiva attiva, il suo istante di ingresso, gli indici di *leg* e di sotto-segmento e i parametri pianificati all'ingresso della primitiva corrente.

9.2.1 Configurazione della Missione e Comandi Elementari

Poiché l'intera logica di guida è incapsulata in un'unica MATLAB Function, il comportamento del sottosistema è determinato esclusivamente dalla `mission_spec` passata in ingresso, costruita a monte da uno script di configurazione che definisce quote operative, posizioni di base e dei siti operativi, waypoint di rotta, quote di crociera e durate di hover. Non è previsto alcun meccanismo di commutazione strutturale (*Variant Subsystem*) per selezionare modalità di volo distinte: la selezione di varianti è servita unicamente come ausilio in fase di sviluppo, per isolare e collaudare le singole primitive, e non fa parte dell'architettura definitiva.

Una conseguenza diretta è che i comandi elementari si ottengono come casi particolari della medesima macchina a stati, agendo sul solo script di configurazione.

Una missione di puro hover, per esempio, si ottiene specificando una *leg* i cui waypoint coincidono con la posizione di partenza e con quota finale pari alla quota di crociera: la macchina esegue il decollo, raggiunge lo stazionamento e vi permane, senza che sia necessario intervenire sul modello Simulink o predisporre un sottosistema dedicato. La riconfigurazione della missione è quindi interamente dichiarativa e confinata al file di waypoint.

All'interno del file di configurazione le virate non sono fra i dati da specificare. Per ciascuna *leg* l'utente fornisce soltanto i waypoint da sorvolare e il punto di fine *leg*, e lo script ricava ampiezza e verso di ogni virata dalla geometria della polilinea risultante. A ogni waypoint intermedio la rotta entrante e quella uscente sono le direzioni dei due segmenti adiacenti nel piano orizzontale, $\psi = \mathrm{atan2}(\Delta E,\, \Delta N)$; la virata in quel waypoint ha ampiezza $|\Delta\psi|$ e verso $\mathrm{sgn}(\Delta\psi)$, dove $\Delta\psi$ è la differenza fra rotta uscente ed entrante riportata nell'intervallo $(-\pi,\, \pi]$. La specifica delle virate contenuta nella `mission_spec` è quindi generata in modo automatico e geometricamente coerente, senza che l'utente debba inserire né angoli né versi.

9.3 Specifica della Missione e Variabili di Stato

9.3.1 Struttura della `mission_spec`

La specifica della missione è un array di *leg*, ciascuna rappresentata da una struttura che contiene:

- la posizione di partenza $\boldsymbol{p}_{\mathrm{start}}$ e di arrivo $\boldsymbol{p}_{\mathrm{end}}$ in coordinate NED;
- la quota di crociera iniziale h_{init} e la quota finale h_{final}, quest'ultima impiegata nelle *leg* che prevedono una variazione di quota;
- un array di waypoint intermedi;
- un array di virate, ciascuna definita dall'ampiezza del cambio di rotta $\Delta\psi$ e dal verso ($+1$ destra, -1 sinistra), calcolate dalla geometria dei waypoint come descritto nella Sezione 9.2.

La dimensione degli array di waypoint e virate è costante (rispettivamente 2×3 e 2×2). La numerosità effettiva di waypoint e virate è dichiarata dai rispettivi contatori.

9.3.2 Variabili di Stato della Macchina

La struttura di stato S mantiene fra invocazioni consecutive:

- `primitive` $\in \{1, \ldots, 12\}$, l'identificatore della primitiva corrente (Tabella 9.1);

- `t_phase_start`, l'istante di ingresso nella primitiva corrente;
- `leg_idx` e `segment_idx`, gli indici della *leg* e del sotto-segmento di rotta correnti;
- i campi di pianificazione, valorizzati *una volta sola* all'ingresso della primitiva (durata pianificata, distanza di decelerazione, profilo di quota, geometria della virata) e i riferimenti del passo precedente, necessari alla continuità del *setpoint*.

Vi è separazione fra l'identificatore di fase e i campi di pianificazione: la pianificazione di una primitiva avviene una sola volta, al momento dell'ingresso, e i parametri così fissati sono poi impiegati per la generazione del riferimento a ogni passo del simulatore. Lo schema evita la ri-pianificazione continua.

9.4 Le Primitive di Volo

La macchina a stati gestisce undici primitive di volo implementate, numerate da 1 a 12 con lo slot 7 non assegnato e riassunte in Tabella 9.1.

Ogni *leg* consuma le primitive in ordine quasi-monotono, con l'eccezione di `cruise` e `turn`, che si alternano più volte all'interno della stessa *leg* quando la geometria prevede più di una virata. La numerazione interna 1–12 non coincide con i valori dell'output `flight_mode`, che assume i valori nell'insieme {1,2,3,4,5,6,8,9}. La differenza non è arbitraria: `flight_mode` non è una copia dell'indice di primitiva, ma un'etichetta del regime fisico di volo. Le quattro primitive di stazionamento (hover iniziale, hover pre-landing, hover operativo a terra e congelamento di fine missione) condividono lo stesso regime fisico e sono perciò mappate tutte sul valore 2, mentre la primitiva di atterraggio (ID 10) è mappata sul valore 9. È questa rappresentazione, e non l'indice logico, a essere consumata dai blocchi di logging e dalle routine di visualizzazione degli assetti per fase.

Tabella 9.1: Primitive di volo della macchina di guidance.

ID	Primitiva	Descrizione sintetica
1	`takeoff`	Salita verticale dalla base alla quota di crociera iniziale, profilo septico di posizione.
2	`hover`	Stazionamento alla quota di crociera per il tempo nominale di hover iniziale.
3	`fw_transition`	Accelerazione dal regime di hover al regime di crociera lungo la direzione della prima *sub-leg*, profilo septico di velocità.
4	`cruise`	Volo a velocità di crociera lungo una *sub-leg*, con eventuale rampa di quota septica.
5	`turn`	Virata a raggio costante e velocità angolare costante.
6	`decel`	Decelerazione da V_{cruise} a $V_{\text{hover-ready}}$ sull'ultima *sub-leg*, con eventuale rampa di quota verso l'approach.
7	*(non assegnato)*	Slot non utilizzato nell'architettura corrente.
8	`approach`	Decelerazione finale da $V_{\text{hover-ready}}$ a 0 sul segmento di buffer prima del sito operativo.
9	`hover_pre_land`	Stazionamento alla quota di approach per il tempo di pre-landing pianificato.
10	`landing`	Discesa verticale fino al suolo presso il sito operativo, profilo septico di posizione.
11	`hover_delivery`	Stazionamento a terra per il tempo dedicato all'operazione di recupero o rilascio del dispositivo.
12	`end`	Fine missione: il riferimento si congela sull'ultima posizione di arrivo.

9.5 Profili Septici: Smoothness Garantita

I profili polinomiali di grado elevato sono un costrutto classico della pianificazione di traiettorie per sistemi meccatronici e aerospaziali, nei quali si richiede continuità non solo della posizione ma anche delle derivate di ordine più elevato [10].

9.5.1 La Funzione *Septic Norm*

Tutti i profili temporali della macchina sono costruiti a partire da una funzione di base, denominata *septic norm*, definita sull'intervallo unitario $\xi \in [0,1]$ [29, 30]:

$$s(\xi) = 35\,\xi^4 - 84\,\xi^5 + 70\,\xi^6 - 20\,\xi^7. \tag{9.1}$$

Il polinomio gode delle proprietà di raccordo

$$s(0) = 0, \qquad s(1) = 1, \qquad s^{(k)}(0) = s^{(k)}(1) = 0 \quad \text{per } k = 1, 2, 3, \tag{9.2}$$

ovvero si annulla, insieme alle prime tre derivate, a entrambi gli estremi. Raccordando in serie tratti che impiegano $s(\xi)$ per descrivere l'evoluzione di una grandezza scalare (velocità, quota, posizione angolare), il profilo risultante è di classe $\mathcal{C}^3$: funzione, derivata prima, seconda e terza sono continue sui raccordi. La continuità della derivata seconda implica accelerazione continua, condizione *sine qua non* per evitare salti del vettore di forza desiderata.

Accanto a $s(\xi)$, la funzione di base fornisce anche il suo integrale normalizzato

$$S(\xi) = \int_0^{\xi} s(\eta)\,\mathrm{d}\eta = 7\,\xi^5 - 14\,\xi^6 + 10\,\xi^7 - 2.5\,\xi^8, \tag{9.3}$$

impiegato per ricavare lo spazio percorso quando è la velocità, e non la posizione, a essere descritta da $s(\xi)$.

9.5.2 Caratterizzazione del Massimo della Derivata

Per una transizione di una grandezza y da y_a a y_b in tempo T, parametrizzata come $y(t) = y_a + (y_b - y_a)\,s(t/T)$, la derivata istantanea è

$$\dot{y}(t) = \frac{y_b - y_a}{T}\, s'(t/T), \tag{9.4}$$

e il suo massimo assoluto si raggiunge in $\xi = 1/2$, dove

$$s'\!\left(\tfrac{1}{2}\right) = \frac{35}{16} = 2.1875. \tag{9.5}$$

Pertanto $\max_t |\dot{y}(t)| = 2.1875\,|y_b - y_a|/T$. Quando $s(\xi)$ descrive una posizione, è invece il massimo della derivata seconda a fissare il picco di accelerazione,

$$\max_{\xi\in[0,1]} |s''(\xi)| = \frac{16.8}{\sqrt{5}} \approx 7.513, \tag{9.6}$$

da cui $\max_t |\ddot{y}(t)| = (16.8/\sqrt{5})\,|y_b - y_a|/T^2$ per una rampa septica di posizione.

Queste relazioni sono impiegate in modo sistematico per dimensionare le durate delle primitive: data una transizione $V_a \to V_b$ con accelerazione massima ammissibile $a_{\max}$, la durata minima compatibile è

$$T_{\text{geom}} = 2.1875\,\frac{|V_b - V_a|}{a_{\max}}, \tag{9.7}$$

e, per $V_a > V_b$, lo spazio percorso vale

$$d_{\text{geom}} = T_{\text{geom}}\,\frac{V_a + V_b}{2} = 2.1875\,\frac{V_a^2 - V_b^2}{2\,a_{\max}}. \tag{9.8}$$

La (9.8) ricorre esplicitamente nel dimensionamento della primitiva di decelerazione.

9.6 Descrizione Dettagliata delle Primitive

9.6.1 Decollo (`takeoff`)

Il decollo è una rampa septica di *posizione* verticale da z_{start} alla quota di crociera iniziale, secondo

$$z_{\text{des}}(t) = z_{\text{start}} + \Delta z\, s(\xi), \qquad \xi = \frac{t - t_0}{T_{\text{to}}}, \tag{9.9}$$

con $\Delta z = -h_{\text{init}} - z_{\text{start}}$. La velocità verticale è la derivata analitica $\dot{z}_{\text{des}} = (\Delta z/T_{\text{to}})\,s'(\xi)$: assume forma di campana, si annulla all'inizio e alla fine della salita e raggiunge il valore di picco $2.1875\,|\Delta z|/T_{\text{to}}$ a metà manovra. La durata della manovra è dimensionata come il maggiore fra due vincoli fisici:

$$T_{\text{to}} = \max\left(2.1875\,\frac{|\Delta z|}{V_{\text{climb,takeoff}}},\ \sqrt{\frac{16.8}{\sqrt{5}}\,\frac{|\Delta z|}{a_{\max,z}}}\right). \tag{9.10}$$

Il primo termine fissa la velocità verticale di picco a $V_{\text{climb,takeoff}}$ e cresce linearmente con $|\Delta z|$; il secondo, ricavato dalla (9.6), fissa l'accelerazione verticale di picco al limite $a_{\max,z}$ e cresce come $\sqrt{|\Delta z|}$. Per salite di grande entità domina il vincolo di velocità, mentre per salite di entità ridotta domina quello di accelerazione, che evita rampe troppo brevi e quindi accelerazioni elevate. La rotta ψ è mantenuta costante e pari a quella della prima *sub-leg*, così da minimizzare la rotazione di imbardata richiesta nella successiva transizione di volo in avanti.

9.6.2 Transizione in Avanti (`fw_transition`)

La transizione dal regime di hover al regime di crociera è una rampa septica della *velocità* da 0 a V_{cruise} lungo il versore $\hat{\boldsymbol{u}}_{\text{fw}}$ della prima *sub-leg*. Velocità e spazio percorso valgono

$$v(\xi) = V_{\text{cruise}}\, s(\xi), \qquad s_\varphi(\xi) = V_{\text{cruise}}\, T_{\text{fw}}\, S(\xi), \tag{9.11}$$

con $S(\xi)$ dato dalla (9.3); la quota è mantenuta costante a h_{init}. La durata si ottiene da

$$T_{\text{fw}} = \min\Big(T_{\text{dyn}},\ T_{\text{geom}}\Big), \qquad T_{\text{dyn}} = 2.1875\,\frac{V_{\text{cruise}}}{a_{\text{nav}}}, \tag{9.12}$$

dove T_{dyn} è dimensionata in modo che l'accelerazione di picco eguagli a_{nav}; trattandosi di una rampa di velocità, questo termine vincola già di per sé l'accelerazione e non richiede un pavimento aggiuntivo. Il limite geometrico $T_{\text{geom}} = 2\,D_{\text{fw}}/V_{\text{cruise}}$ garantisce che la transizione si completi entro la distanza D_{fw} disponibile fino al primo waypoint. Se il limite geometrico è attivo, la transizione adotta un'accelerazione di picco superiore ad a_{nav} pur di non superare il waypoint: una saturazione spaziale a $s_\varphi = D_{\text{fw}}$ impedisce comunque l'*overshoot*.

9.6.3 Crociera (`cruise`)

Volo rettilineo a velocità nominale V_{cruise} costante lungo una *sub-leg*, con spazio percorso $s = V_{\text{cruise}}\,(t - t_0)$. Se la *leg* prevede una variazione di quota, la crociera esegue contestualmente una rampa septica di quota distribuita sull'intera *sub-leg*; dette $\Delta h = h_{\text{final}} - h_{\text{init}}$ e D la lunghezza della *sub-leg*, la velocità verticale di riferimento è

$$v_{D,\text{des}}(t) = \frac{\Delta h}{T_{\text{total}}}\, s'\!\left(\frac{s}{D}\right), \qquad T_{\text{total}} = \frac{D}{V_{\text{cruise}}}. \tag{9.13}$$

Poiché $s'(0) = s'(1) = 0$, la velocità verticale si annulla agli estremi della *sub-leg*: la rampa di quota è quindi confinata al segmento e non perturba né l'ingresso né l'uscita delle virate adiacenti. La rampa è omessa quando la variazione di quota richiesta è inferiore alla soglia di tolleranza $dz_{\min} = 10^{-3}$ m.

9.6.4 Virata (`turn`)

La virata è un arco di circonferenza di raggio fisso R_{turn} percorso a velocità di avanzamento V_{cruise} costante e a quota costante. La rotta avanza a velocità angolare *costante*

$$\omega = \operatorname{sgn}(\Delta\psi)\,\frac{V_{\text{cruise}}}{R_{\text{turn}}}, \qquad \psi(t) = \psi_{\text{in}} + \omega\,(t - t_0), \tag{9.14}$$

sicché l'arco di ampiezza $|\Delta\psi|$ è descritto nel tempo $T_{\text{turn}} = |\Delta\psi|\, R_{\text{turn}}/V_{\text{cruise}}$. La posizione, la velocità e l'accelerazione di riferimento sono i valori cinematici esatti del moto circolare uniforme attorno al centro $\boldsymbol{c}$ della virata:

$$\boldsymbol{p}_{\text{des}} = \boldsymbol{c} + R_{\text{turn}} \left[\cos\psi,\ \sin\psi,\ 0\right]^{\top}, \tag{9.15}$$

$$\boldsymbol{v}_{\text{des}} = R_{\text{turn}}\, \omega \left[-\sin\psi,\ \cos\psi,\ 0\right]^{\top}, \tag{9.16}$$

$$\boldsymbol{a}_{\text{ff}} = -R_{\text{turn}}\, \omega^2 \left[\cos\psi,\ \sin\psi,\ 0\right]^{\top}. \tag{9.17}$$

Il centro $\boldsymbol{c}$ è collocato in modo che la circonferenza risulti tangente alla traiettoria di riferimento nel punto di ingresso, sul lato indicato dal verso della virata. Questa costruzione garantisce la continuità della velocità ai raccordi: in ingresso e in uscita $\|\boldsymbol{v}_{\text{des}}\| = R_{\text{turn}}\,|\omega| = V_{\text{cruise}}$ e il versore tangente coincide con quello della crociera adiacente. La rotta di riferimento è ricavata dalla direzione del vettore velocità, $\psi_{\text{des}} = \operatorname{atan2}(v_{\text{des},y}, v_{\text{des},x})$.

9.6.5 Decelerazione (`decel`)

La decelerazione è una rampa septica della velocità da V_{cruise} a $V_{\text{hover-ready}}$, distribuita su un tratto s_{total} pari alla distanza residua fino al sito operativo diminuita del buffer di approach:

$$s_{\text{total}} = \|\boldsymbol{p}_{\text{end}} - \boldsymbol{p}_{\text{enter,decel}}\| - d_{\text{approach,buffer}}. \tag{9.18}$$

La distanza minima necessaria alla rampa discende dalla (9.8), applicata a $V_a = V_{\text{cruise}}$, $V_b = V_{\text{hover-ready}}$, $a_{\max} = a_{\text{max,decel}}$. La distanza pianificata all'ingresso della *leg* è

$$d_{\text{decel,leg}} = 2.1875\, \frac{V_{\text{cruise}}^2 - V_{\text{hover-ready}}^2}{2\, a_{\text{max,decel}}}\, \alpha_{\text{margin}} + d_{\text{approach,buffer}}, \tag{9.19}$$

maggiorata del fattore $\alpha_{\text{margin}} = 1.3$ per robustezza alle variazioni della velocità reale rispetto a quella nominale; essa definisce la distanza residua alla quale si attiva la transizione da `cruise` a `decel`. La durata effettiva è il maggiore fra il vincolo geometrico e il vincolo di accelerazione, $T_{\text{decel}} = \max(T_{\text{geom}}, T_{a,\max})$, con $T_{\text{geom}} = s_{\text{total}}/V_{\text{avg}}$ e $T_{a,\max} = 2.1875\,|V_{\text{cruise}} - V_{\text{hover-ready}}|/a_{\text{max,decel}}$.

A differenza di quanto avviene in approach, la decelerazione non si svolge necessariamente a quota costante: quando la quota di crociera della *leg* differisce dalla quota di approach, la primitiva include una rampa septica di quota dal valore corrente a z_{approach}, parametrizzata sullo stesso ξ del profilo longitudinale.

9.6.6 Approach (`approach`)

Sul segmento finale, di lunghezza pari al buffer $d_{\text{approach,buffer}}$ prima del sito operativo, il velivolo decelera ulteriormente da $V_{\text{hover-ready}}$ a $V = 0$ secondo la rampa

septica $V(\xi) = V_{\text{hover-ready}}\big(1 - s(\xi)\big)$, a quota costante. La durata è $T_{\text{app}} = 2\,d_{\text{total}}/V_{\text{hover-ready}}$, con d_{total} la distanza residua al sito; la decelerazione di picco vale di conseguenza $2.1875\,V_{\text{hover-ready}}/T_{\text{app}}$. La transizione di assetto da volo a hover è gestita in modo automatico dal generatore del comando di assetto, che tramite il blending SLERP ruota gradualmente il velivolo all'aumentare del peso relativo della soluzione di hover (Capitolo 7).

9.6.7 Hover Pre-Landing, Landing, Hover Delivery

Le tre primitive che concludono la *leg* hanno struttura semplice. `hover_pre_land` è uno stazionamento alla quota di approach per la durata nominale $T_{\text{hover,pre}}$. `landing` è una discesa verticale costruita, in piena analogia con il decollo, come rampa septica di posizione dalla quota di approach al suolo. La durata

$$T_{\text{land}} = \max\left(2.1875\,\frac{h}{V_{\text{descent,landing}}},\ \sqrt{\frac{16.8}{\sqrt{5}}\,\frac{h}{a_{\max,z}}}\right) \tag{9.20}$$

segue lo stesso criterio del decollo: è il maggiore fra il vincolo di velocità di picco, che fissa la velocità verticale di picco a $V_{\text{descent,landing}}$, e il vincolo di accelerazione di picco, che ne limita l'accelerazione al valore $a_{\max,z}$. Poiché il profilo termina esattamente al suolo all'istante T_{land}, il contatto è un evento deterministico e non richiede una rilevazione separata. `hover_delivery` è infine uno stazionamento a terra per T_{delivery} secondi, durante il quale avviene l'operazione di missione: il recupero del dispositivo medico presso il primo sito operativo o il suo rilascio presso il secondo.

9.7 Phase Planner e Logica di Transizione

La logica della macchina è il *phase planner*, una struttura `switch` sui dieci stati che, alla conclusione della primitiva corrente, determina quella successiva. Le condizioni di transizione sono temporali (esaurimento della durata pianificata), spaziali (raggiungimento di un waypoint o di una distanza residua), cinematiche (discesa della velocità sotto una soglia) o miste; sono riassunte in Tabella 9.2.

Due aspetti della tabella meritano di essere sottolineati. Il primo è l'anticipo spaziale della transizione `cruise` $\rightarrow$ `turn`: la virata è ingaggiata 100 m prima del waypoint nominale, margine che consente di posizionare il centro della circonferenza e di raccordare la rotta senza arretramenti del *setpoint*. Il secondo è la ridondanza delle condizioni di uscita delle primitive dinamiche: la transizione `decel` $\rightarrow$ `approach` avviene non appena la velocità misurata scende sotto $V_{\text{hover-ready}}$, ma

Tabella 9.2: Transizioni nominali del phase planner.

Da	**Condizione di passaggio**	**A**
`takeoff` (1)	$t - t_0 \geq T_{\text{to}}$	`hover` (2)
`hover` (2)	$t - t_0 \geq T_{\text{hover,initial}}$	`fw_transition` (3)
`fw_transition` (3)	$t - t_0 \geq T_{\text{fw}}$ oppure $s_{\varphi} \geq D_{\text{fw}} - \text{tol}$	`cruise` (4)
`cruise` (4), *sub-leg* intermedia	$s_{\text{des}} \geq D_{\text{to,next}} - 100\,\text{m}$	`turn` (5)
`cruise` (4), *sub-leg* terminale	$s_{\text{des}} \geq D_{\text{to,next}} - d_{\text{decel,leg}}$	`decel` (6)
`turn` (5)	$\psi_{\text{progr}} \geq \lvert\Delta\psi\rvert$ oppure $t - t_0 \geq 1.5\,T_{\text{turn}}$	`cruise` (4)
`decel` (6)	$V_{\text{air}} \leq V_{\text{hover-ready}}$ oppure $t - t_0 \geq T_{\text{decel}}$	`approach` (8)
`approach` (8)	$\lVert\boldsymbol{p}_{xy} - \boldsymbol{p}_{\text{target},xy}\rVert < r_{\text{arrived}}$	`hover_pre_land` (9)
`hover_pre_land` (9)	$t - t_0 \geq T_{\text{hover,pre}}$	`landing` (10)
`landing` (10)	$t - t_{\text{land,start}} \geq T_{\text{land}}$	`hover_delivery` (11)
`hover_delivery` (11)	$t - t_0 \geq T_{\text{delivery}}$	*leg* successiva oppure `end` (12)

un limite temporale su T_{decel} garantisce comunque il passaggio anche in presenza di errori di tracking; analogamente, l'uscita dalla virata su $1.5\,T_{\text{turn}}$ funge da salvaguardia rispetto al criterio primario sull'angolo cumulato.

All'ingresso di ogni nuova primitiva viene invocata una routine `enter_*` che pianifica i parametri locali e li deposita nei campi di pianificazione di S. In particolare `enter_cruise_segment` valuta i waypoint residui per stabilire se la *sub-leg* sfocia in una virata o nella decelerazione, scartando i waypoint che coincidono con il *setpoint* corrente; `enter_decel` calcola $d_{\text{decel,leg}}$ secondo la (9.19) e prepara la rampa septica; `enter_turn` calcola raggio, ampiezza, durata T_{turn}, verso e centro della virata.

9.8 Generazione dell'Output di Riferimento

A ogni passo del simulatore, il blocco produce le grandezze seguenti:

- $\boldsymbol{p}_{\text{des}}(t) \in \mathbb{R}^3$, la posizione di riferimento nel frame NED;
- $\boldsymbol{v}_{\text{des}}(t) \in \mathbb{R}^3$, la velocità di riferimento nel frame NED, derivata temporale analitica di $\boldsymbol{p}_{\text{des}}$;
- $\boldsymbol{a}_{\text{ff}}(t) \in \mathbb{R}^3$, l'accelerazione di feed-forward passata al controllore di posizione come anticipato nella Sezione 8.3.2;
- $\psi_{\text{des}}(t) \in [-\pi, \pi)$, la rotta di riferimento, impiegata dal generatore del comando di assetto per la costruzione della terna di hover;
- `flight_mode`, l'identificatore del regime di volo corrente, usato per il logging e la visualizzazione.

Le derivate della posizione di riferimento sono calcolate *analiticamente* dai profili polinomiali, e non per differenza finita, così da escludere rumore numerico dal feed-forward. Il termine $\boldsymbol{a}_{\text{ff}}$ è la derivata seconda analitica del profilo nelle primitive di decollo, transizione, crociera, approach e atterraggio, ed è l'accelerazione centripeta esatta nella virata.

La regolarità complessiva del riferimento è quindi una conseguenza diretta delle proprietà di $s(\xi)$ e dello schema di pianificazione, che fissa i raccordi fra primitive sui medesimi valori di posizione, velocità e accelerazione (questi ultimi nulli per le primitive di hover). Su tutti i raccordi fra primitive septiche, e fra una primitiva septica e uno stazionamento, il riferimento è di classe $\mathcal{C}^3$. La sola eccezione sono i raccordi della virata: l'arco a velocità angolare costante preserva la continuità di posizione e velocità, ma introduce in ingresso e in uscita il gradino limitato di accelerazione centripeta caratterizzato nella Sezione 9.6.4. Il riferimento prodotto è dunque globalmente di classe $\mathcal{C}^1$ e $\mathcal{C}^3$ a tratti, una proprietà sufficiente a garantire un comando di forza, e a cascata di assetto, privo di transitori bruschi.

Capitolo 10

Risultati e Analisi delle Simulazioni

10.1 Introduzione

Questo capitolo è dedicato alla discussione dei risultati delle simulazioni di validazione del sistema complessivo descritto nei capitoli precedenti. La trattazione ha un duplice obiettivo. Il primo è verificare che l'architettura di controllo unificata, sintetizzata e analizzata in modo modulare nei Capitoli 5–9, sia in grado di gestire l'intero inviluppo di volo del tailsitter, dall'hover stazionario alla crociera ad alta velocità, su una missione operativa completa e priva di commutazione modale. Il secondo è caratterizzare quantitativamente le prestazioni di inseguimento della traiettoria, la qualità delle transizioni e l'impegno degli attuatori, individuando i regimi di volo in cui il sistema lavora con maggiore margine e quelli in cui si avvicina ai propri limiti fisici.

La validazione è condotta mediante simulazione non lineare a sei gradi di libertà del modello completo *plant* più sistema di controllo, integrato con il solutore Runge–Kutta a passo fisso del quarto ordine (`ode4`) al passo $dt = 1\,\mathrm{ms}$. Tutti i parametri del veicolo e dei controllori coincidono con la configurazione di riferimento riepilogata nell'Appendice A; nessun guadagno è stato modificato fra le diverse fasi della missione. Lo scenario di prova è la missione operativa completa di recupero e rilascio di un dispositivo medico, introdotta nel Capitolo 3 e dettagliata nella Sezione 10.2.

L'esposizione procede dal generale al particolare. La Sezione 10.2 richiama la geometria della missione simulata. La Sezione 10.3 analizza l'inseguimento della traiettoria nel suo complesso, mentre la Sezione 10.4 ne quantifica l'errore. Le Sezioni 10.5 e 10.6 approfondiscono le due manovre più critiche, rispettivamente le transizioni hover–crociera e le virate coordinate. La Sezione 10.7 esamina l'impegno

degli attuatori e la Sezione 10.8 la fase di atterraggio verticale. La Sezione 10.9 raccoglie infine una sintesi quantitativa dei risultati.

10.2 Scenario di Missione Simulato

La missione di riferimento è la *three-leg* di recupero e rilascio di un dispositivo medico descritta nel Capitolo 3. Il velivolo decolla verticalmente dalla base, raggiunge il primo sito operativo (Sito A) dove avviene il recupero del dispositivo, prosegue verso il secondo sito operativo (Sito B) dove il dispositivo viene rilasciato durante la seconda discesa, e infine rientra alla base di partenza. Ciascuna *leg* è composta dalla medesima sequenza di primitive di volo, decollo verticale, hover, transizione in avanti, crociera con eventuali virate, decelerazione, *approach*, hover di pre-atterraggio, atterraggio verticale e hover operativo a terra, generate dalla macchina a stati di guidance del Capitolo 9 senza alcuna commutazione strutturale fra modalità di volo.

La missione completa simulata ha una durata complessiva di circa 2050 s e comprende tre transizioni in avanti, tre decelerazioni con conseguente richiamo all'assetto di hover, cinque virate coordinate e tre atterraggi verticali. Le quote di crociera delle tre *leg* sono $h_1 = 100\,\mathrm{m}$, $h_2 = 200\,\mathrm{m}$ e $h_3 = 150\,\mathrm{m}$; la velocità di crociera nominale è $V_{\mathrm{cruise}} = 61\,\mathrm{m\,s^{-1}}$. Questo scenario è volutamente articolato: l'alternanza di salite, discese, virate di ampiezza diversa e ripetuti cicli hover–crociera–hover sollecita l'intero inviluppo operativo e costituisce una prova esauriente dell'architettura di controllo unificata.

10.3 Inseguimento della Traiettoria

La Figura 10.1 mostra la traiettoria tridimensionale percorsa dal baricentro del velivolo, sovrapposta al riferimento generato dal sistema di guidance e colorata in funzione del modulo della velocità relativa V_{air}. La traiettoria misurata e quella di riferimento risultano praticamente indistinguibili alla scala dell'intera missione: i tratti di crociera, in cui la mappa cromatica satura sul valore massimo, sono percorsi rettilinei a quota costante, mentre i raccordi a bassa velocità in prossimità dei siti operativi corrispondono ai segmenti di transizione e di stazionamento.

La proiezione sul piano orizzontale Nord–Est, riportata nella Figura 10.2, evidenzia la topologia della missione: i tre segmenti di andata e ritorno fra la base e i due siti operativi, raccordati dalle cinque virate. Anche in pianta la traiettoria misurata segue fedelmente il riferimento; i punti di partenza e di arrivo coincidono, a conferma del corretto rientro alla base al termine della terza *leg*.

Il profilo di quota in funzione della distanza orizzontale percorsa è riportato nella Figura 10.3. Si riconoscono le tre quote di crociera delle *leg* e le discese verticali

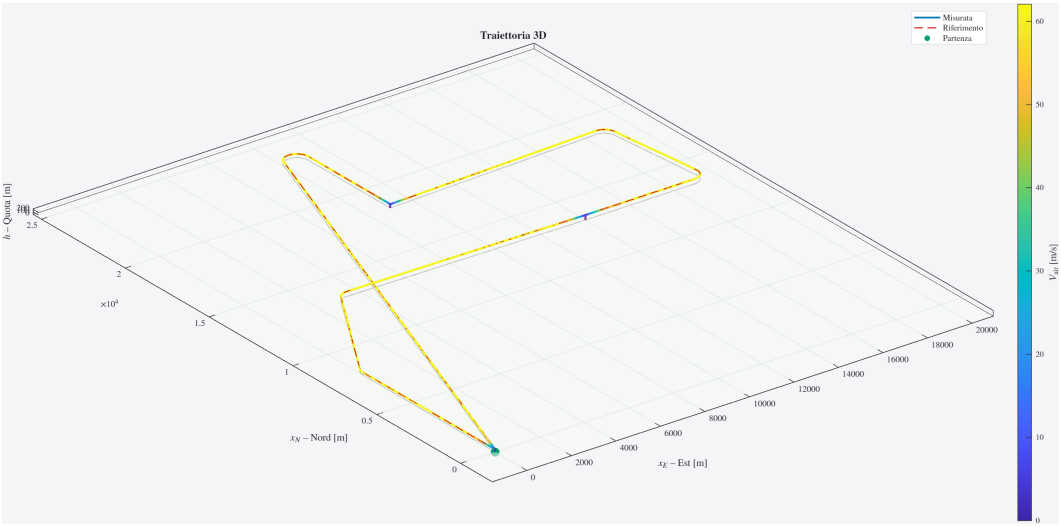


Figura 10.1: Traiettoria tridimensionale dell'intera missione: confronto fra la traiettoria misurata e quella di riferimento. La scala cromatica rappresenta il modulo della velocità relativa V_{air}.

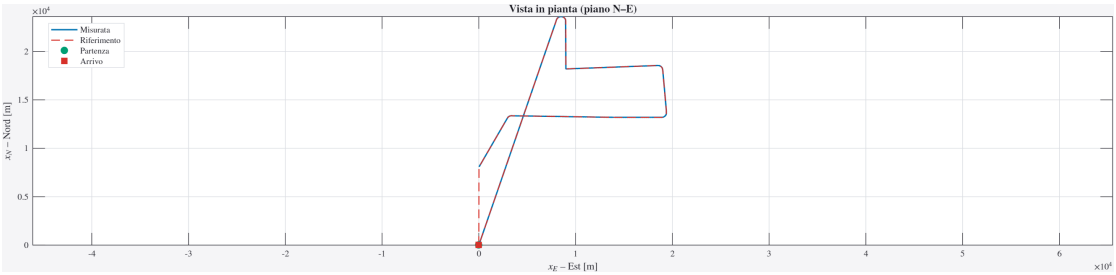


Figura 10.2: Vista in pianta (piano Nord–Est) della traiettoria di missione. La coincidenza dei marcatori di partenza e di arrivo conferma il rientro alla base.

fino al suolo in corrispondenza dei due siti operativi e della base. Le rampe di quota interne alle *leg*, generate dai profili septici di posizione descritti nella Sezione 9.5, raccordano in modo regolare i diversi livelli di volo senza sovraelongazioni apprezzabili.

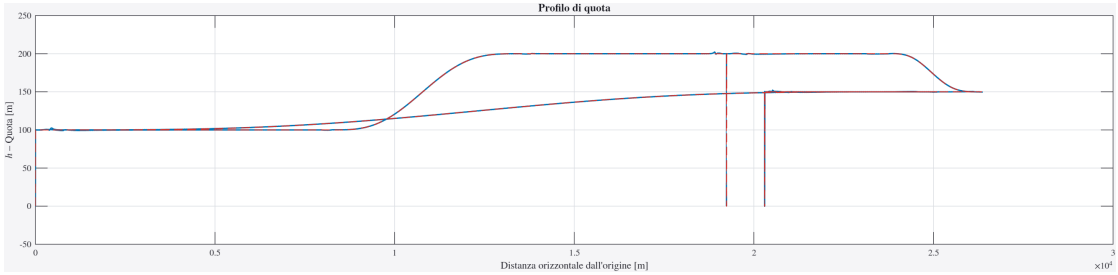
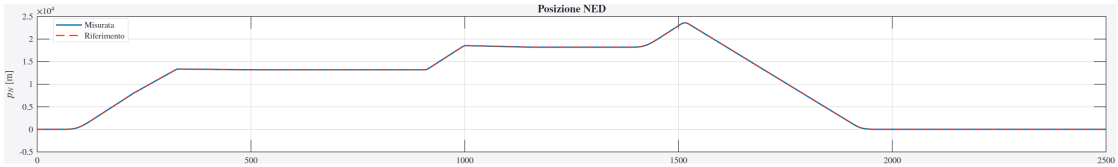


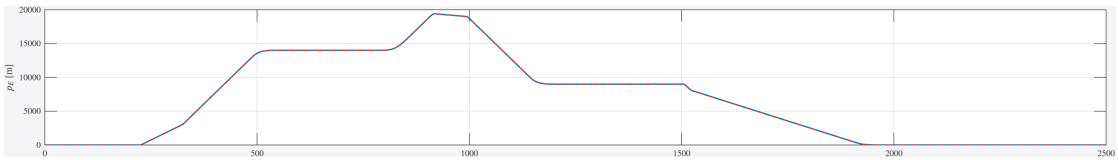
Figura 10.3: Profilo di quota in funzione della distanza orizzontale percorsa. Le discese verticali fino a quota nulla corrispondono agli atterraggi presso i siti operativi e la base.

L'evoluzione temporale delle tre componenti di posizione nel frame inerziale NED è presentata nella Figura 10.4. Per ciascuna componente la curva misurata si sovrappone a quella di riferimento per l'intera durata della missione; gli scostamenti, non distinguibili a questa scala, sono quantificati nella Sezione 10.4. La componente verticale h (Figura 10.4c) rende particolarmente evidente la struttura della missione: i plateau corrispondono alle fasi di crociera alle tre quote operative e ai relativi stazionamenti, mentre le rampe di salita e discesa verticale corrispondono ai decolli e agli atterraggi.

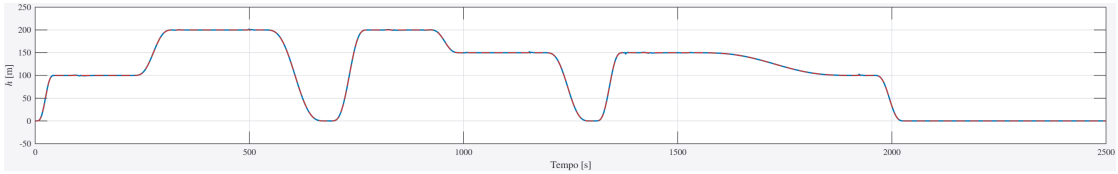
La Figura 10.5 riporta le tre componenti della velocità inerziale. Nelle fasi di crociera le componenti orizzontali v_N e v_E si attestano sui valori di plateau imposti dal pianificatore, con modulo prossimo a V_{cruise}, e il segno cambia in modo coerente con la direzione delle *leg*; la componente verticale v_D resta prossima a zero salvo durante le rampe di quota e le manovre verticali. L'assenza di oscillazioni residue sui tratti a velocità costante conferma che l'anello di posizione insegue i riferimenti di velocità senza eccitare modi mal smorzati.



(a) Componente Nord p_N.



(b) Componente Est p_E.



(c) Quota h.

Figura 10.4: Evoluzione temporale delle componenti di posizione: confronto fra valore misurato e riferimento.

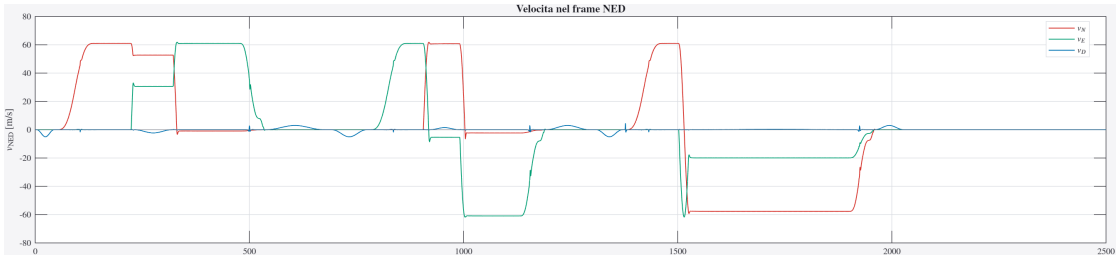


Figura 10.5: Componenti della velocità inerziale nel frame NED durante l'intera missione.

10.4 Analisi dell'Errore di Inseguimento

La Figura 10.6 riporta le tre componenti dell'errore di posizione $\boldsymbol{e}_p = \boldsymbol{p} - \boldsymbol{p}_{\text{des}}$ in funzione del tempo. L'errore si mantiene prossimo allo zero per la maggior parte della missione, in particolare durante i tratti di crociera rettilinea e di hover, dove le tre componenti restano contenute entro pochi centimetri. Gli scostamenti significativi si concentrano in corrispondenza delle manovre dinamiche, ossia le transizioni e le virate, dove il riferimento varia più rapidamente e l'errore raggiunge picchi dell'ordine di alcuni metri. Tale comportamento è atteso e fisiologico: in queste fasi il velivolo è soggetto a variazioni rapide del comando di forza e di assetto, e l'errore transitorio riflette la banda passante finita degli anelli di controllo.

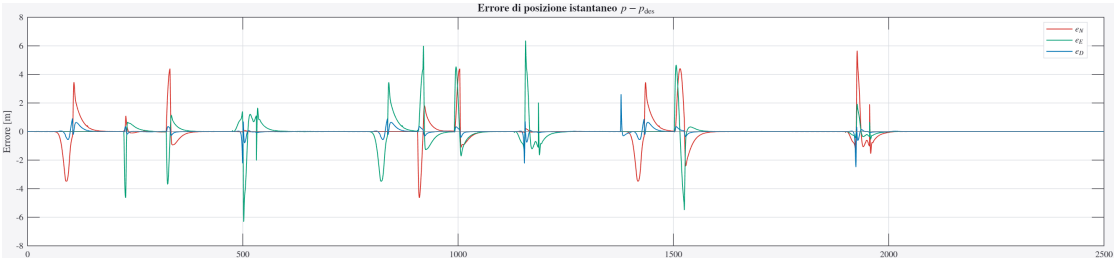


Figura 10.6: Errore di posizione istantaneo $\boldsymbol{p} - \boldsymbol{p}_{\text{des}}$, scomposto nelle tre componenti NED.

La norma dell'errore di posizione, mostrata nella Figura 10.7, sintetizza l'informazione precedente in una singola grandezza scalare. Si osserva un errore quasi nullo durante le fasi stazionarie e i tratti di crociera, e una sequenza di picchi isolati, di ampiezza compresa indicativamente da 3 m a 6 m, in corrispondenza delle manovre. È significativo che, dopo ciascun picco, la norma dell'errore decada rapidamente verso lo zero: l'azione integrale del controllore di posizione (Sezione 8.4) riassorbe lo scostamento residuo e il sistema ritrova l'inseguimento nominale prima dell'inizio della manovra successiva. Non si osserva alcun accumulo progressivo dell'errore lungo la missione, indice di stabilità asintotica dell'anello esterno su tutto l'inviluppo.

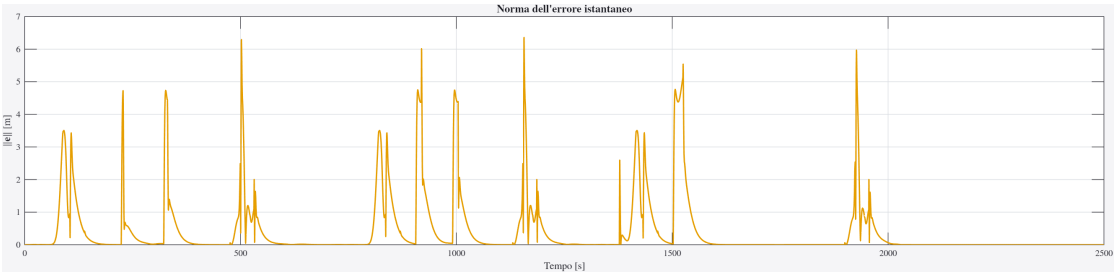


Figura 10.7: Norma dell'errore di posizione istantaneo $\|\boldsymbol{e}_p\|$ durante l'intera missione.

10.5 Le Transizioni Hover–Crociera

Le transizioni fra assetto verticale e assetto di crociera costituiscono la manovra più caratteristica del tailsitter e la prova più severa per l'architettura di controllo unificata. La Figura 10.8 mostra l'angolo di beccheggio θ per l'intera missione: la grandezza alterna ripetutamente fra il valore prossimo a 90°, tipico dell'assetto di hover con asse x_B verticale, e il valore di trim livellato di crociera, prossimo a 4°. Le escursioni complete corrispondono alle tre transizioni in avanti e alle tre decelerazioni con richiamo all'hover. In tutte le escursioni la curva misurata segue il riferimento generato dal blending SLERP del comando di assetto (Capitolo 7) senza discontinuità, a conferma del carattere globale e privo di commutazione modale del controllore.

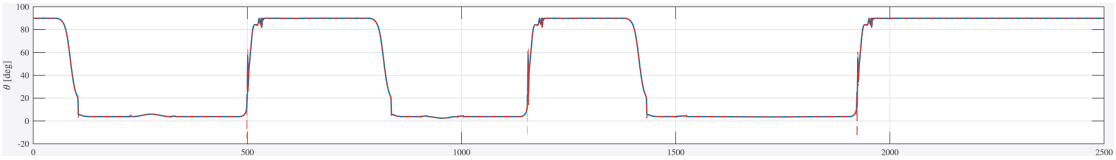


Figura 10.8: Angolo di beccheggio θ durante l'intera missione: confronto fra valore misurato e riferimento. Le escursioni fra 90° e 4° corrispondono alle transizioni hover–crociera.

L'andamento del modulo della velocità relativa nel corso della missione è riportato nella Figura 10.9. La velocità alterna fra i plateau di crociera, prossimi a V_{cruise}, e i valori prossimi a zero delle fasi di hover e di stazionamento a terra. La linea tratteggiata indica la velocità di transizione V_{trans} che governa il punto di flesso della sigmoide di blending: è attorno a tale soglia che il peso relativo delle soluzioni di hover e di crociera si bilancia, e che il velivolo compie la maggior parte della rotazione di beccheggio.

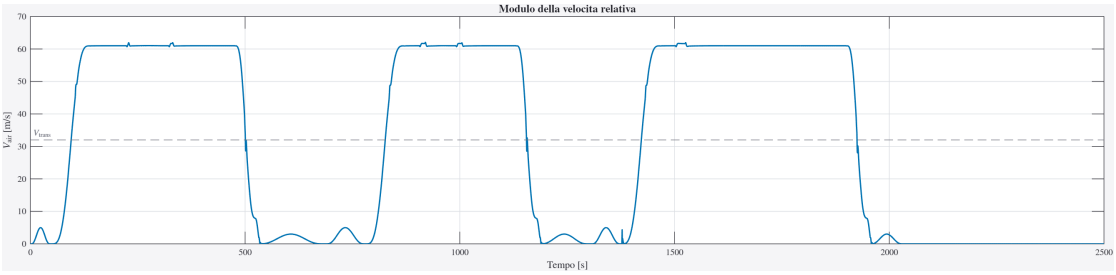


Figura 10.9: Modulo della velocità relativa V_{air} durante l'intera missione. La linea tratteggiata indica la velocità di transizione V_{trans}.

La Figura 10.10 isola, su una finestra temporale ristretta, una transizione in avanti seguita a distanza dalla successiva decelerazione, mentre la Figura 10.11 riporta il corrispondente andamento della velocità relativa. Il confronto fra le due figure mostra la natura coordinata della manovra: la rotazione di beccheggio e la variazione di velocità procedono congiuntamente, e l'attraversamento della soglia V_{trans} coincide con la fase di massima velocità di rotazione dell'assetto.

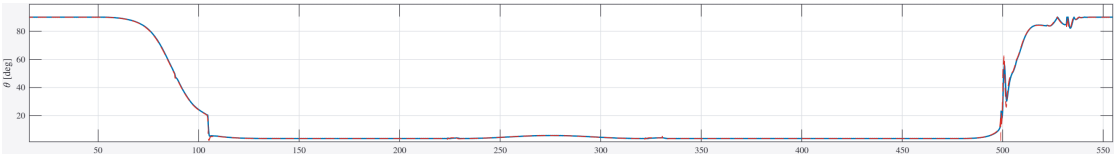


Figura 10.10: Dettaglio dell'angolo di beccheggio θ su una finestra comprendente una transizione in avanti e la successiva decelerazione.

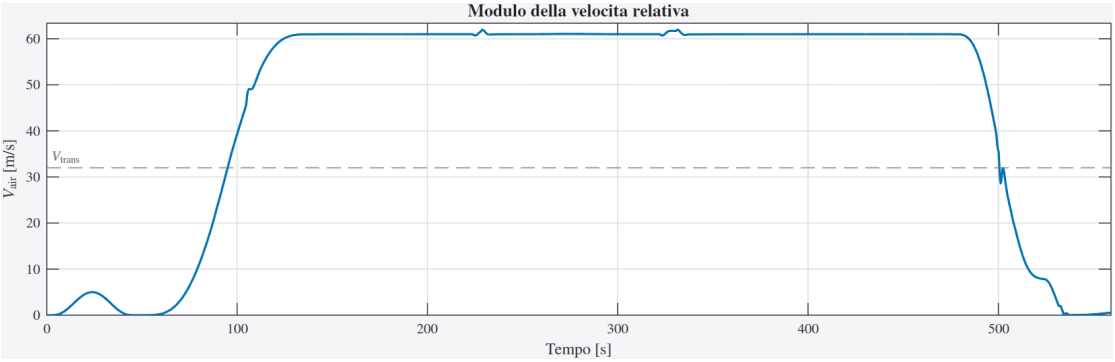


Figura 10.11: Modulo della velocità relativa V_{air} durante una transizione in avanti e la successiva decelerazione.

Le Figure 10.12 e 10.13 riportano in dettaglio l'angolo di beccheggio durante, rispettivamente, una transizione in avanti e una decelerazione con richiamo all'hover. Nella transizione in avanti la rotazione da 90° all'assetto di crociera è monotona e regolare, e la curva misurata segue strettamente il riferimento; si nota un breve tratto a pendenza ridotta in prossimità dei 20°, dove il contributo aerodinamico della portanza diventa dominante. Nella decelerazione, viceversa, il richiamo verso l'assetto verticale presenta un transitorio più marcato: in corrispondenza dell'inizio della manovra la velocità decresce rapidamente, l'autorità aerodinamica si riduce e l'assetto misurato compie alcune oscillazioni di ampiezza contenuta attorno al riferimento, smorzate entro pochi secondi. Questo comportamento è coerente con la maggiore criticità della back-transition, in cui il veicolo passa da un regime *wing-borne* a un regime *thrust-borne* con autorità di controllo aerodinamica decrescente.

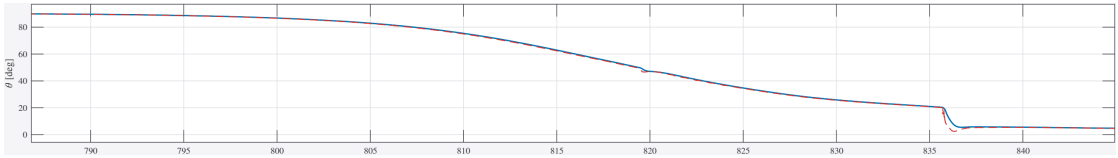


Figura 10.12: Dettaglio dell'angolo di beccheggio θ durante una transizione in avanti.

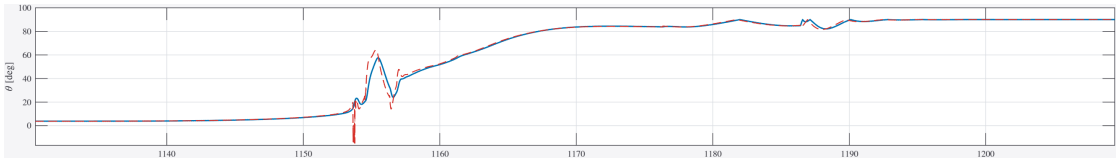


Figura 10.13: Dettaglio dell'angolo di beccheggio θ durante una decelerazione con richiamo all'assetto di hover.

Una rappresentazione tridimensionale dell'assetto del velivolo durante le transizioni è fornita dalle Figure 10.14–10.17, in cui il modello solido del veicolo è campionato a istanti successivi lungo la traiettoria. La Figura 10.14 mostra una transizione in avanti: si osserva la progressiva rotazione del corpo dall'assetto verticale a quello di crociera lungo un tratto rettilineo a quota costante. La Figura 10.15 ne fornisce la vista laterale, che evidenzia con maggiore chiarezza la rotazione di beccheggio attorno all'asse y_B e il mantenimento della quota durante la manovra. Le Figure 10.16 e 10.17 illustrano due decelerazioni: la rotazione inversa dall'assetto di crociera a quello verticale si sviluppa su un tratto in cui il velivolo, perdendo velocità, riduce gradualmente la distanza percorsa per unità di tempo, come testimoniato dall'addensamento dei modelli solidi verso la fine della manovra.

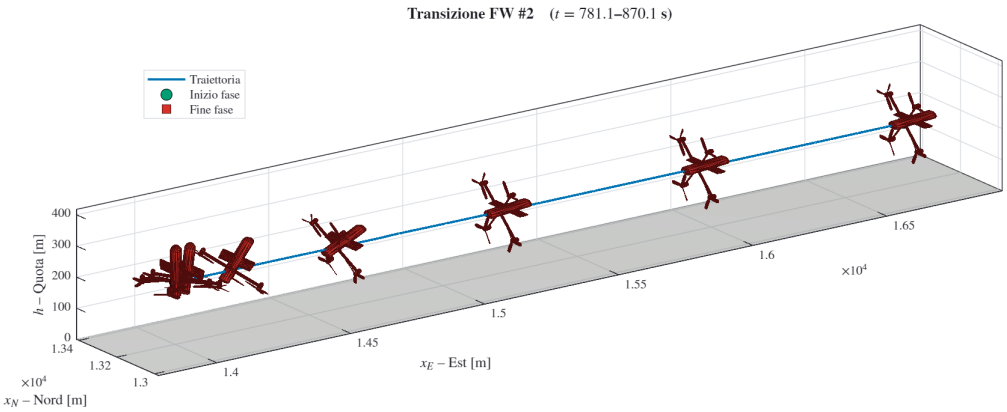


Figura 10.14: Rappresentazione tridimensionale dell'assetto del velivolo durante una transizione in avanti.

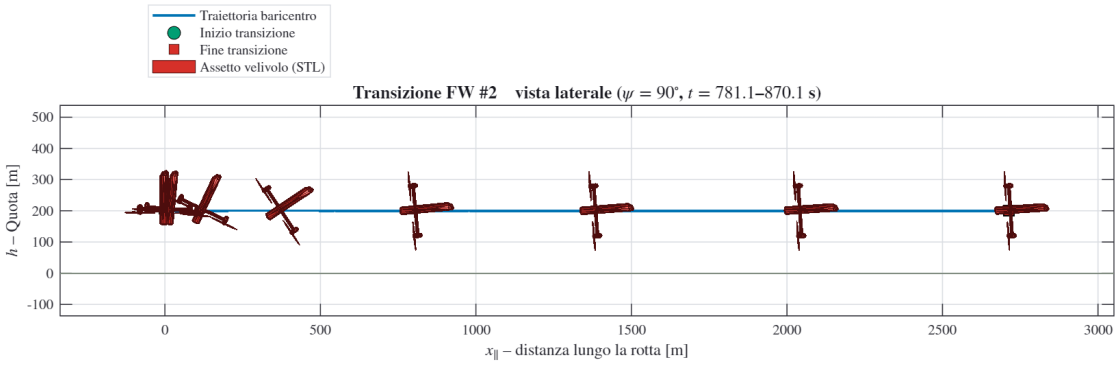


Figura 10.15: Vista laterale della transizione in avanti: la rotazione di beccheggio avviene a quota sostanzialmente costante.

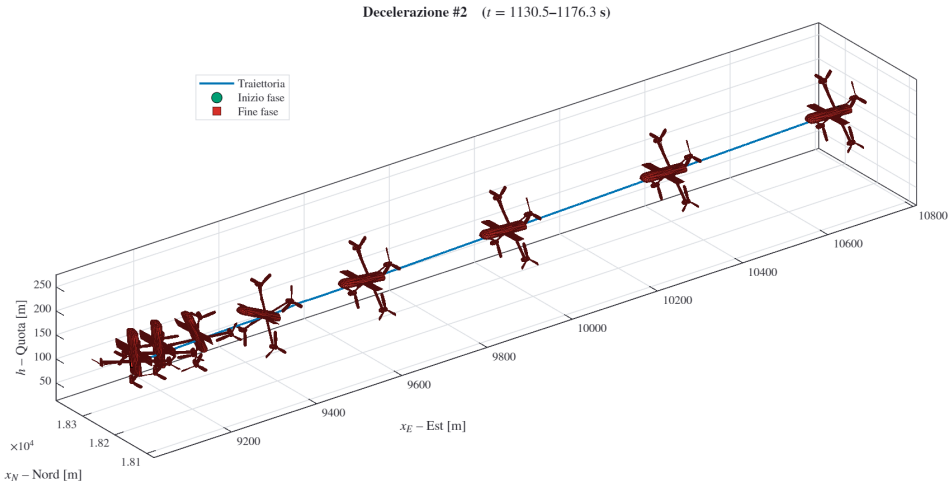


Figura 10.16: Rappresentazione tridimensionale dell'assetto del velivolo durante una decelerazione con richiamo all'assetto di hover.

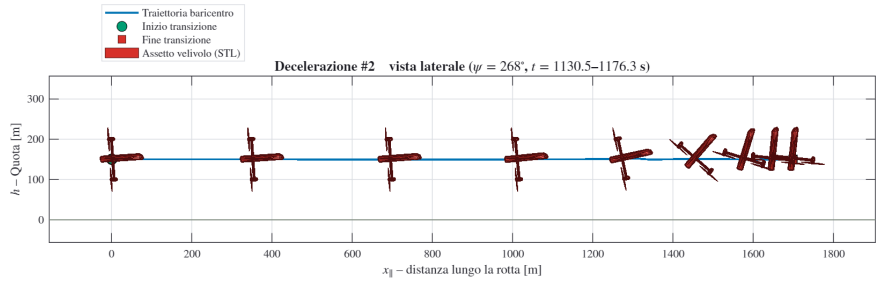
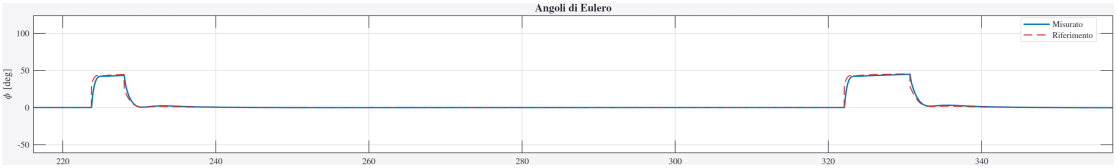


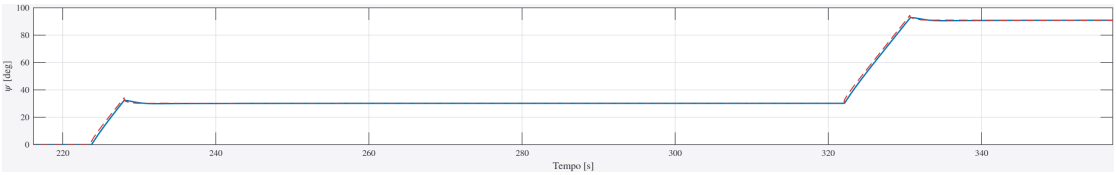
Figura 10.17: Vista laterale della transizione all'indietro: la rotazione di beccheggio avviene a quota sostanzialmente costante.

10.6 Le Virate Coordinate

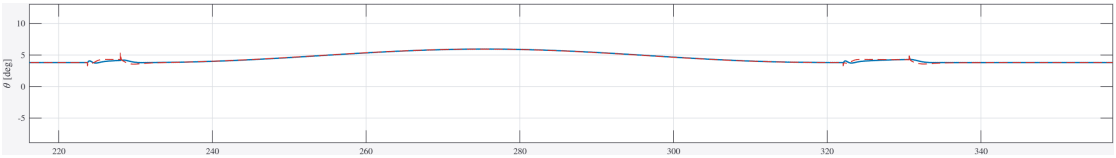
Le virate rappresentano la seconda manovra critica della missione. La Figura 10.18 riporta i tre angoli di Eulero su una finestra temporale comprendente due virate consecutive della prima *leg*. L'angolo di imbardata ψ (Figura 10.18b) avanza per rampe in corrispondenza delle due virate, raccordando le rotte delle *sub-leg* adiacenti; l'angolo di rollio ϕ (Figura 10.18a) assume un valore non nullo, dell'ordine di 45°, solo durante l'esecuzione della virata, e ritorna a zero sui tratti rettilinei, come atteso per una virata coordinata a quota costante; l'angolo di beccheggio θ (Figura 10.18c) resta prossimo al valore di trim di crociera, con escursioni contenute. In tutte e tre le componenti la curva misurata segue fedelmente il riferimento.



(a) Angolo di rollio ϕ.



(b) Angolo di imbardata ψ.



(c) Angolo di beccheggio θ.

Figura 10.18: Angoli di Eulero durante due virate coordinate consecutive: confronto fra valore misurato e riferimento.

Il dettaglio in pianta di una virata, riportato nella Figura 10.19, mostra che la traiettoria misurata segue l'arco di riferimento con uno scostamento contenuto, concentrato nella fase centrale della virata. Tale scostamento è la manifestazione geometrica del gradino di accelerazione centripeta che la primitiva di virata introduce in ingresso e in uscita dall'arco, discusso nella Sezione 9.6.4: il riferimento di virata garantisce la continuità di posizione e velocità ma non quella di accelerazione, e l'errore transitorio che ne deriva è coerente con quanto previsto in fase di progetto.

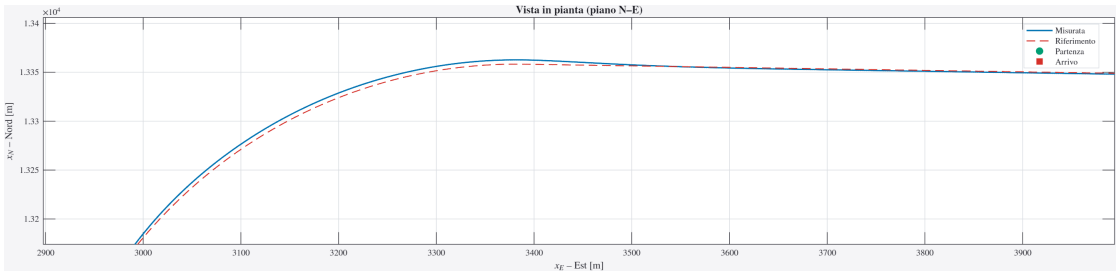


Figura 10.19: Dettaglio in pianta di una virata coordinata: confronto fra traiettoria misurata e arco di riferimento.

Le Figure 10.20 e 10.21 quantificano l'errore di posizione in corrispondenza di due virate successive. Le componenti dell'errore raggiungono valori dell'ordine di alcuni metri.

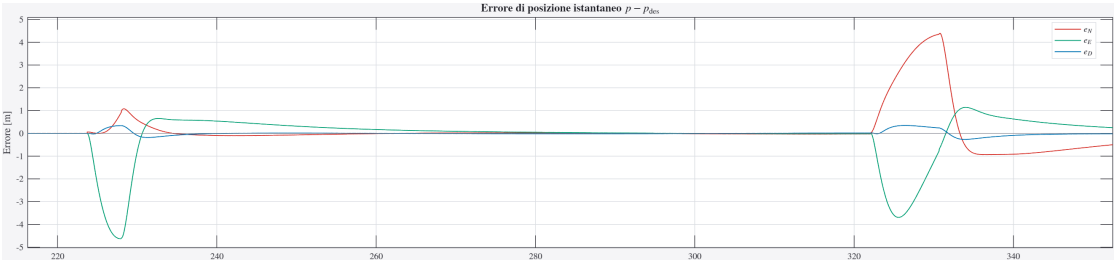


Figura 10.20: Errore di posizione istantaneo durante due virate coordinate consecutive, scomposto nelle tre componenti NED.

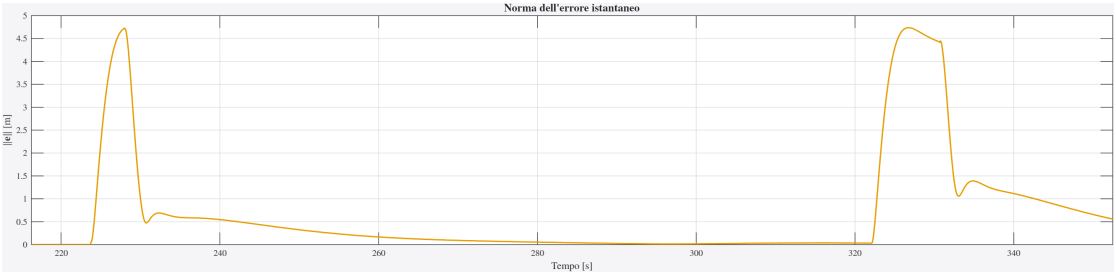


Figura 10.21: Norma dell'errore di posizione durante due virate coordinate consecutive.

La Figura 10.22 fornisce una rappresentazione tridimensionale dell'assetto durante una virata: il velivolo, in regime di crociera, percorre l'arco mantenendo un angolo di rollio non nullo e la quota costante, in accordo con l'andamento degli angoli di Eulero della Figura 10.18.

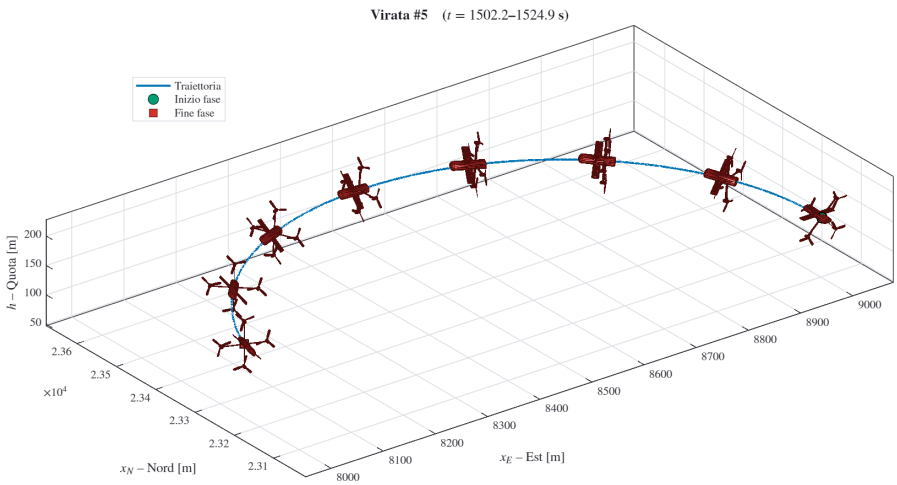


Figura 10.22: Rappresentazione tridimensionale dell'assetto del velivolo durante una virata coordinata.

10.7 Impegno degli Attuatori

L'analisi dell'impegno degli attuatori consente di verificare che i comandi prodotti dal sistema di controllo siano fisicamente realizzabili e che sussista un margine adeguato rispetto ai limiti di saturazione. La Figura 10.23 riporta la spinta totale comandata T_{des} durante l'intera missione. Nelle fasi di hover la spinta si attesta sul valore necessario a bilanciare il peso, prossimo a mg e indicato dalla linea tratteggiata; nelle fasi di crociera essa si riduce sensibilmente, poiché la portanza alare si fa carico di gran parte del sostentamento e ai motori resta il solo compito di vincere la resistenza aerodinamica. I picchi isolati, di breve durata, si verificano all'avvio delle transizioni, quando il comando di forza varia rapidamente; il loro valore resta comunque entro l'autorità complessiva del sistema propulsivo.

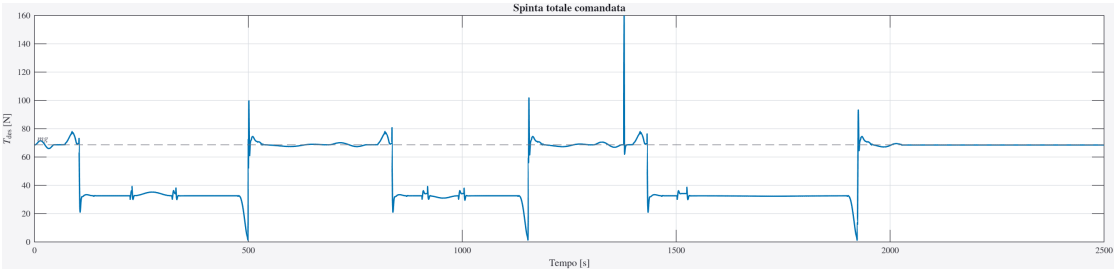


Figura 10.23: Spinta totale comandata T_{des} durante l'intera missione. La linea tratteggiata indica il livello di equilibrio mg.

La Figura 10.24 mostra la velocità di rotazione dei quattro motori. I quattro regimi sono sostanzialmente sovrapposti per gran parte della missione, con piccole differenze in corrispondenza delle manovre, dove il differenziale di spinta genera i momenti di controllo. Il regime di hover è più elevato di quello di crociera, in coerenza con l'andamento della spinta totale; i transitori in corrispondenza delle transizioni restano contenuti entro il regime massimo ammissibile del motore, confermando che il sistema propulsivo non raggiunge la saturazione in condizioni operative nominali.

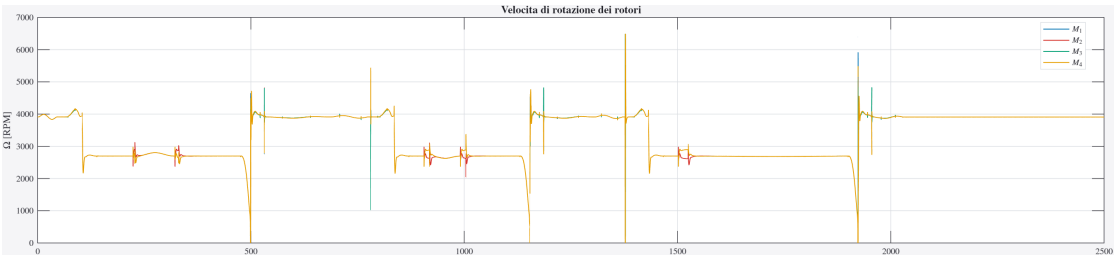


Figura 10.24: Velocità di rotazione dei quattro motori durante l'intera missione.

La deflessione dei due elevoni è riportata nella Figura 10.25. Durante le fasi di hover e di volo a bassa velocità la deflessione è prossima a zero, in conseguenza del *cutoff* a bassa velocità della funzione di blending (Sezione 5.3.2), che disattiva gli elevoni quando la pressione dinamica è insufficiente a generare momenti utili. In crociera le due superfici assestano sul valore di trim, mentre in corrispondenza delle transizioni e delle virate si osservano escursioni più ampie, che in alcuni transitori raggiungono i limiti meccanici di saturazione $\pm\delta_{\max}$. Il raggiungimento della saturazione è circoscritto a brevi intervalli ed è gestito strutturalmente dall'architettura di allocazione adattiva, che, per costruzione, trasferisce ai motori l'eventuale deficit di coppia non realizzabile dagli elevoni (Sezione 5.3.4), preservando la coerenza fra coppia comandata e coppia allocata.

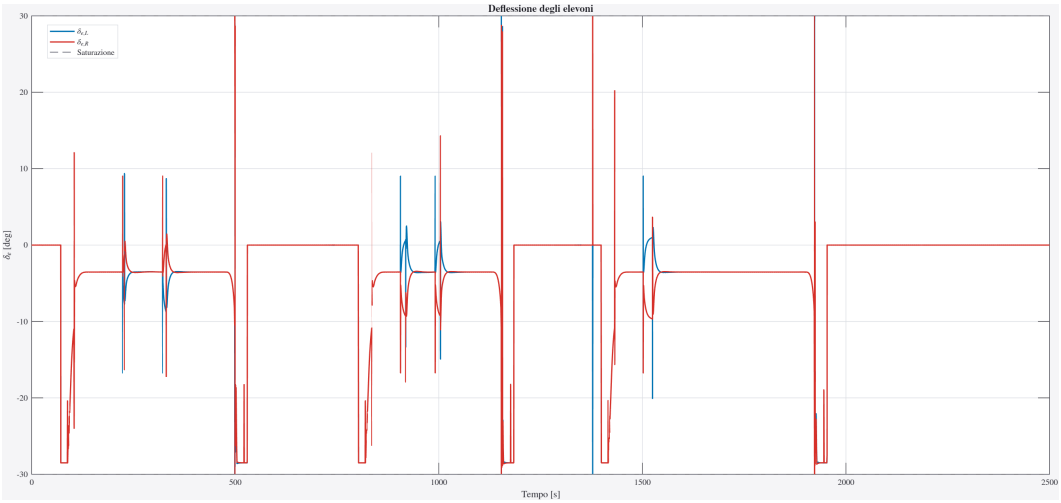


Figura 10.25: Deflessione dei due elevoni $\delta_{e,L}$ e $\delta_{e,R}$ durante l'intera missione. Le linee tratteggiate indicano i limiti di saturazione meccanica.

10.8 Atterraggio Verticale

La fase di atterraggio conclude ciascuna *leg* della missione. La Figura 10.26 fornisce una rappresentazione tridimensionale dell'assetto del velivolo durante una discesa verticale: il veicolo, riportato in assetto di hover dalla precedente decelerazione, discende lungo una traiettoria sostanzialmente verticale fino al contatto con il suolo, mantenendo l'asse x_B allineato alla verticale locale. La discesa segue il profilo septico di posizione generato dalla primitiva di `landing` (Sezione 9.6), che garantisce un contatto con il suolo a velocità verticale prossima a zero ed evita sollecitazioni impulsive sulla struttura.

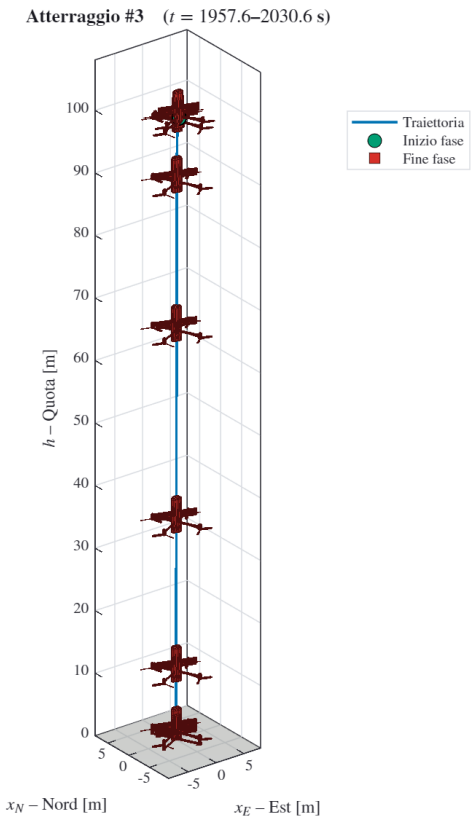


Figura 10.26: Rappresentazione tridimensionale dell'assetto del velivolo durante un atterraggio verticale.

10.9 Sintesi dei Risultati

Le simulazioni presentate in questo capitolo dimostrano che l'architettura di controllo unificata sviluppata nei capitoli precedenti è in grado di gestire l'intera missione operativa di recupero e rilascio del dispositivo medico, dall'hover stazionario alla crociera ad alta velocità, senza alcuna commutazione modale e con un unico set di guadagni valido su tutto l'inviluppo di volo.

L'inseguimento della traiettoria è risultato accurato: durante le fasi di crociera e di stazionamento l'errore di posizione si mantiene prossimo allo zero, mentre nelle manovre dinamiche, transizioni e virate, raggiunge picchi transitori dell'ordine di alcuni metri, prontamente riassorbiti dall'azione del controllore. Non si è osservato alcun accumulo progressivo dell'errore lungo la missione, indice della stabilità asintotica del sistema complessivo. Le transizioni hover–crociera, in entrambi i versi, si sono sviluppate in modo continuo e regolare; la decelerazione si è confermata la manovra più critica, con un transitorio di assetto più marcato ma comunque smorzato e stabile. Le virate coordinate sono state eseguite mantenendo la quota e con un errore di traiettoria coerente con il gradino di accelerazione centripeta intrinseco alla primitiva di virata. L'analisi degli attuatori ha infine confermato la realizzabilità fisica dei comandi: il sistema propulsivo opera entro il proprio inviluppo e gli elevoni raggiungono la saturazione soltanto in transitori brevi e isolati, gestiti strutturalmente dall'allocazione adattiva.

Nel complesso, i risultati validano numericamente l'approccio proposto e ne confermano l'adeguatezza per la classe di missioni considerata. Le considerazioni conclusive e le possibili direzioni di sviluppo sono raccolte nel Capitolo 11.

Capitolo 11

Conclusioni

11.1 Sintesi del Lavoro Svolto

Il presente lavoro di tesi ha affrontato la modellazione, la guida e il controllo di un velivolo a decollo e atterraggio verticale di tipo tailsitter, con l'obiettivo di sviluppare e validare un'architettura di controllo capace di gestire l'intero inviluppo di volo, dall'hover stazionario alla crociera ad alta velocità, senza ricorrere a una commutazione esplicita fra modalità di volo distinte.

A partire dalla definizione della configurazione del velivolo (Capitolo 2), il lavoro ha sviluppato un modello dinamico non lineare a sei gradi di libertà (Capitolo 4), formulato su una parametrizzazione dell'assetto mediante quaternione unitario al fine di evitare le singolarità proprie degli angoli di Eulero in prossimità dell'assetto verticale. Su questo modello è stata costruita un'architettura di controllo a cascata, composta da un'allocazione adattiva dei controlli basata sulla massima autorità erogabile dagli elevoni (Capitolo 5), da un controllore di assetto con feed-forward inerziale sull'anello di velocità angolare (Capitolo 6), da un generatore del comando di assetto fondato sulla legge di volo coordinato di Ritz e D'Andrea (Capitolo 7) e da un controllore di posizione di tipo PID operante nel frame inerziale (Capitolo 8). L'intero sistema è coordinato da un modulo di guida e navigazione (Capitolo 9) che genera missioni complete a partire da primitive di volo a continuità di classe $\mathcal{C}^3$, raccordate da una macchina a stati priva di discontinuità strutturali.

L'architettura così sviluppata è stata implementata in ambiente MATLAB/-Simulink e validata mediante simulazione non lineare a sei gradi di libertà su una missione operativa completa di recupero e rilascio di un dispositivo medico, articolata in tre *leg* e comprendente transizioni, virate coordinate e atterraggi verticali, secondo quanto discusso nel Capitolo 10.

11.2 Principali Risultati

I risultati delle simulazioni di validazione confermano che l'architettura proposta soddisfa l'obiettivo posto in apertura. Gli elementi più significativi possono essere riassunti come segue.

In primo luogo, il sistema di controllo gestisce l'intera missione con un unico set di guadagni, valido senza modifiche dall'hover alla crociera. Le transizioni fra assetto verticale e assetto livellato, in entrambi i versi, si sviluppano in modo continuo e regolare, senza discontinuità nei comandi e senza commutazione modale, a conferma del carattere globale dell'approccio adottato.

In secondo luogo, l'inseguimento della traiettoria è risultato accurato. Durante le fasi di crociera rettilinea e di stazionamento l'errore di posizione si mantiene prossimo allo zero; nelle manovre dinamiche esso raggiunge picchi transitori dell'ordine di alcuni metri. L'assenza di accumulo progressivo dell'errore lungo l'intera missione costituisce un'evidenza numerica della stabilità asintotica del sistema complessivo.

In terzo luogo, l'analisi dell'impegno degli attuatori ha dimostrato la realizzabilità fisica dei comandi prodotti. Il sistema propulsivo opera entro il proprio inviluppo per tutta la durata della missione, mentre gli elevoni raggiungono i limiti di saturazione soltanto in transitori brevi e isolati, gestiti in modo strutturale dall'allocazione adattiva mediante il trasferimento ai motori del deficit di coppia non realizzabile dalle superfici aerodinamiche.

Nel complesso, il lavoro fornisce una validazione numerica completa di un'architettura di controllo unificata per velivoli tailsitter e ne conferma l'adeguatezza per la classe di missioni considerata.

11.3 Limiti del Lavoro

I risultati ottenuti vanno interpretati alla luce delle ipotesi adottate, che ne delimitano la portata e individuano al contempo le direzioni di approfondimento.

La validazione è stata condotta in condizioni atmosferiche nominali, in assenza di vento, raffiche e turbolenza. La robustezza del sistema rispetto a disturbi atmosferici realistici, particolarmente critici nelle fasi a bassa velocità e durante le transizioni, non è quindi stata caratterizzata in modo sistematico.

Le simulazioni sono state eseguite assumendo la disponibilità ideale dello stato del velivolo. L'effetto del rumore di misura, dei ritardi di elaborazione e degli errori residui di stima non è stato incluso nelle prove di validazione complessiva, e la loro influenza sulle prestazioni degli anelli di controllo resta da quantificare.

L'analisi è inoltre riferita a una singola configurazione del velivolo e a un unico insieme di parametri nominali. La sensibilità delle prestazioni rispetto alle

incertezze sui parametri inerziali, aerodinamici e propulsivi non è stata oggetto di una campagna dedicata.

Infine, la validazione è interamente numerica. Il comportamento del sistema in una implementazione su calcolatore di bordo, con le conseguenti limitazioni di risorse di calcolo e di rappresentazione numerica, non è stato verificato.

11.4 Sviluppi Futuri

Le limitazioni individuate suggeriscono in modo naturale alcune direzioni di sviluppo.

Un primo filone riguarda la valutazione della robustezza. L'introduzione di modelli di vento e turbolenza nelle simulazioni, unitamente a campagne di tipo Monte Carlo sulle incertezze parametriche, permetterebbe di caratterizzare in modo statistico i margini di prestazione e di stabilità del sistema su tutto l'inviluppo.

Un secondo filone riguarda il percorso verso l'implementazione. La generazione automatica di codice a partire dal modello Simulink e la conseguente verifica mediante prove *software-in-the-loop* e *hardware-in-the-loop* costituiscono il passo preliminare necessario a una eventuale sperimentazione in volo, che rappresenta la validazione conclusiva dell'architettura proposta.

Appendice A

Parametri del Modello e dei Controllori

A.1 Introduzione

Questa appendice raccoglie in forma tutti i parametri numerici del modello e dei controllori del tailsitter VTOL. I valori riportati corrispondono alla configurazione di riferimento utilizzata nelle simulazioni del Capitolo 10.

I parametri sono raggruppati in cinque categorie: parametri fisici del velivolo (massa, inerzia, geometria), propulsione, aerodinamica, controllo (PID di posizione, attitude, rate, allocazione, blending) e navigazione. Tutti i valori sono coerenti con il file di inizializzazione `plant_params.m` e `controller_gains.m`.

A.2 Parametri Fisici del Velivolo

Tabella A.1: Parametri fisici del velivolo tailsitter.

Parametro	**Simbolo**	**Valore**	**Unità**
Massa totale (MTOW)	m	7.00	kg
Momento d'inerzia, asse x (roll)	I_{xx}	0.326	$\mathrm{kg\,m^2}$
Momento d'inerzia, asse y (pitch)	I_{yy}	0.500	$\mathrm{kg\,m^2}$
Momento d'inerzia, asse z (yaw)	I_{zz}	0.500	$\mathrm{kg\,m^2}$
Superficie alare di riferimento	S	0.1022	$\mathrm{m^2}$
Apertura alare totale	b	0.73	m
Corda media aerodinamica	c	0.14	m
Distanza motore–baricentro	d	0.325	m
Posizione baricentro (x dal muso)	x_{CG}	0.2333	m
Offset baricentro vs. centro aerod.	Δx_{CG}	-0.0217	m
Diametro elica	D_{prop}	0.432	m
Accelerazione di gravità	g	9.81	$\mathrm{m\,s^{-2}}$
Densità aria di riferimento	ρ	1.112	$\mathrm{kg\,m^{-3}}$

A.3 Propulsione

Tabella A.2: Parametri della propulsione (T-Motor MN5008 KV340 + elica 17×5.8" tripala).

Parametro	**Simbolo**	**Valore**	**Unità**
Costante di velocità motore	K_V	340	$\mathrm{rpm\,V^{-1}}$
Tensione batteria (6S LiPo)	V_{batt}	22.2	V
Numero di giri massimo	$\Omega_{\mathrm{max,RPM}}$	6482	rpm
Velocità angolare massima	Ω_{max}	679	$\mathrm{rad\,s^{-1}}$
Spinta massima per motore (tripala)	T_{max}	47.1	N
Coefficiente di spinta	k_T	$1.022 \cdot 10^{-4}$	$\mathrm{N/(rad/s)^2}$
Coefficiente di coppia/spinta	κ	0.0198	m
Coefficiente di coppia	k_Q	$2.021 \cdot 10^{-6}$	$\mathrm{Nm/(rad/s)^2}$

La matrice di mixing $\mathbf{M}_{\mathrm{mix}}$, che lega le velocità angolari al quadrato dei quattro motori alla spinta totale e ai tre momenti, è:

$$\mathbf{M}_{\mathrm{mix}} = \begin{bmatrix} +k_T & +k_T & +k_T & +k_T \\ -k_Q & +k_Q & -k_Q & +k_Q \\ +dk_T & -dk_T & -dk_T & +dk_T \\ -dk_T & -dk_T & +dk_T & +dk_T \end{bmatrix} \tag{A.1}$$

A.4 Aerodinamica e Superfici di Controllo

Tabella A.3: Coefficienti aerodinamici e parametri delle superfici di controllo.

Parametro	Simbolo	Valore	Unità
Drag parassita totale (VSP)	$C_{D_0,\mathrm{VSP}}$	0.15	–
Efficacia elevatore (lift)	C_{L,δ_e}	1.5	rad^{-1}
Efficacia elevatore (pitch)	C_{m,δ_e}	–0.8	rad^{-1}
Efficacia alettone (roll)	C_{l,δ_a}	–0.33	rad^{-1}
Yaw avverso da alettone	C_{n,δ_a}	0	rad^{-1}
Sideforce da angolo di sideslip	$C_{Y,\beta}$	–2.13	rad^{-1}
Roll da sideslip	$C_{l,\beta}$	–1.08	rad^{-1}
Yaw stiffness	$C_{n,\beta}$	0.26	rad^{-1}
Roll damping	$C_{l,p}$	–1.40	rad^{-1}
Pitch damping	$C_{m,q}$	–3.72	rad^{-1}
Yaw damping	$C_{n,r}$	–0.13	rad^{-1}
Cross-coupling roll-yaw rate	$C_{l,r}$	–1.28	rad^{-1}
Cross-coupling yaw-roll rate	$C_{n,p}$	–0.08	rad^{-1}
Deflessione massima superfici	$\delta_{\max}$	30	$^\circ$
Velocità minima attivazione elevoni	$V_{\min,\mathrm{elev}}$	4	$\mathrm{m\,s}^{-1}$

I coefficienti non lineari $C_L(\alpha)$, $C_D(\alpha)$, $C_m(\alpha)$ sono modellati da reti neurali a singolo strato nascosto, allenate su dati sperimentali coprenti $\alpha \in [-130^\circ, +130^\circ]$.

A.5 Modello di Contatto col Suolo

Tabella A.4: Parametri del modello di contatto col suolo.

Parametro	Simbolo	Valore	Unità
Rigidezza terreno (verticale)	k_G	50000	$\mathrm{N\,m}^{-1}$
Smorzamento terreno (verticale)	b_G	2000	$\mathrm{N\,s\,m}^{-1}$
Coefficiente di attrito orizzontale	μ	0.5	–

A.6 Controllo di Posizione

Tabella A.5: Guadagni del controllore PID di posizione e parametri di saturazione.

Parametro	**Unità**	**Nord** (N)	**Est** (E)	**Down** (D)
$K_{p,\mathrm{pos}}$	s^{-2}	1.2	1.2	3.5
$K_{d,\mathrm{pos}}$	s^{-1}	1.3	1.3	4.5
$K_{i,\mathrm{pos}}$	s^{-3}	0.08	0.08	0.3
$a_{\mathrm{des},I,\max}$ (norma)	$\mathrm{m\,s}^{-2}$		15	
$a^{+}_{\max,xy}$ (accel)	$\mathrm{m\,s}^{-2}$		15	
$a^{-}_{\max,xy}$ (decel)	$\mathrm{m\,s}^{-2}$		15	
$a_{\max,z,\mathrm{up}}$	$\mathrm{m\,s}^{-2}$		13	
$a_{\max,z,\mathrm{dn}}$	$\mathrm{m\,s}^{-2}$		13	

A.7 Controllo di Assetto e Rate Loop

Tabella A.6: Guadagni del controllore di assetto, rate loop e feed-forward.

Parametro	**Unità**	**Roll** (x)	**Pitch** (y)	**Yaw** (z)
$K_{p,\mathrm{att}}$	$\mathrm{rad\,s}^{-1}$	5.0	3.5	3.0
$K_{p,\mathrm{rate}}$	Nm/(rad/s)	10	12	12
$K_{i,\mathrm{rate}}$	$\mathrm{N\,m\,rad}^{-1}$	5	6	6
$\tau_{\max}$	N m	11.2	11.2	5.0
$\tau_{\mathrm{rate,cl}}$	s	0.05	0.05	0.05
$\dot{b}_{x,\max}$ (rate $\boldsymbol{b}_x$)	$\mathrm{rad\,s}^{-1}$		$\pi/2$	
$b_{x,\max,\mathrm{tilt}}$ (tilt limite)	rad		$\pi/2$	

A.8 Allocazione e Blending

Tabella A.7: Parametri dell'allocazione adattiva.

Parametro	**Simbolo**	**Valore**	**Unità**
Margine sfruttamento elevoni	α_{share}	0.95	–
Cutoff velocità elevoni	$V_{\text{min,elev}}$	4	$\mathrm{m\,s^{-1}}$

A.9 Navigazione

Tabella A.8: Parametri della macchina a stati di navigazione.

Parametro	**Simbolo**	**Valore**	**Unità**
Velocità di crociera	V_{cruise}	61	$\mathrm{m\,s^{-1}}$
Velocità di hover-ready	$V_{\text{hover_ready}}$	8	$\mathrm{m\,s^{-1}}$
Raggio di virata	R_{turn}	500	m
Tolleranza arrivo waypoint	r_{arrived}	2	m

A.10 Parametri di Simulazione

Tabella A.9: Parametri di integrazione e simulazione.

Parametro	**Simbolo**	**Valore**	**Unità**
Integratore	–	ode4 (RK)	–
Passo di integrazione	Δt_{sim}	10^{-3}	s
Frequenza di campionamento controllo	f_{ctrl}	200	Hz

A.11 Note sui Parametri

I parametri presentati in questa appendice rappresentano la configurazione di riferimento utilizzata per le simulazioni di tutti i capitoli precedenti. Tutti i

valori dimensionali sono nel Sistema Internazionale (kg, m, s, rad, N) salvo dove esplicitamente indicato (gradi, RPM).

I parametri sono stati calibrati sulla base di analisi VSPAero della geometria CAD del velivolo, datasheet dei componenti commerciali (T-Motor MN5008, P17×5.8 CF) e procedure di tuning manuale sui controllori.

Bibliografia

[1] I. H. Abbott e A. E. von Doenhoff, *Theory of Wing Sections, Including a Summary of Airfoil Data*. New York: Dover Publications, 1959.

[2] K. J. Åström e T. Hägglund, *Advanced PID Control*. Research Triangle Park, NC: ISA — Instrumentation, Systems, e Automation Society, 2006.

[3] K. J. Åström e R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ: Princeton University Press, 2008.

[4] R. Austin, *Unmanned Aircraft Systems: UAVS Design, Development and Deployment*. John Wiley & Sons, 2010.

[5] R. Bapst, R. Ritz, L. Meier e M. Pollefeys, «Design and Implementation of an Unmanned Tail-Sitter,» in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015, pp. 1885–1890.

[6] R. W. Beard e T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton, NJ: Princeton University Press, 2012.

[7] D. Brescianini, M. Hehn e R. D'Andrea, «Nonlinear Quadrocopter Attitude Control,» ETH Zurich, Institute for Dynamic Systems e Control, Zurich, Switzerland, rapp. tecn., 2013.

[8] M. V. Cook, *Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control*, 2nd. Butterworth-Heinemann, 2007.

[9] C. C. Critzos, H. H. Heyson e R. W. Boswinkle, «Aerodynamic Characteristics of NACA 0012 Airfoil Section at Angles of Attack from 0 to 180,» National Advisory Committee for Aeronautics, rapp. tecn. NACA TN 3361, 1955.

[10] M. D. L. Dalla Vedova, «Modellazione, Simulazione e Sperimentazione dei Sistemi Aerospaziali,» Corso di Laurea Magistrale in Ingegneria Aerospaziale, Politecnico di Torino, A.A. 2025/26, 2026.

[11] J. Diebel, «Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors,» Stanford University, Stanford, CA, rapp. tecn., 2006.

[12] B. Etkin e L. D. Reid, *Dynamics of Flight: Stability and Control*, 3rd. New York: John Wiley & Sons, 1996.

[13] A. Frank, J. P. McGrew, M. Valenti, D. Levine e J. P. How, «Hover, Transition, and Level Flight Control Design for a Single-Propeller Indoor Airplane,» in *AIAA Guidance, Navigation and Control Conference*, 2007.

[14] G. F. Franklin, J. D. Powell e A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 7th. Boston: Pearson, 2015.

[15] E. Fresk e G. Nikolakopoulos, «Full Quaternion Based Attitude Control for a Quadrotor,» in *European Control Conference (ECC)*, Zurich, Switzerland, 2013, pp. 3864–3869.

[16] M. T. Hagan, H. B. Demuth, M. H. Beale e O. De Jesús, *Neural Network Design*, 2nd. Martin Hagan, Stillwater, OK, 2014, Available at: `https://hagan.okstate.edu/nnd.html` (Accessed: 28 March 2025).

[17] A. S. Hahn, «Vehicle Sketch Pad: A Parametric Geometry Modeler for Conceptual Aircraft Design,» in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, Orlando, FL, 2010. DOI: `10.2514/6.2010-657`.

[18] M. Hochstenbach, C. Notteboom, B. Theys e J. De Schutter, «Design and Control of an Unmanned Aerial Vehicle for Autonomous Parcel Delivery with Transition from Vertical Take-off to Forward Flight,» *International Journal of Micro Air Vehicles*, vol. 7, n. 4, pp. 395–405, 2015.

[19] K. Hornik, M. Stinchcombe e H. White, «Multilayer Feedforward Networks are Universal Approximators,» *Neural Networks*, vol. 2, n. 5, pp. 359–366, 1989.

[20] M.-D. Hua, T. Hamel, P. Morin e C. Samson, «Introduction to Feedback Control of Underactuated VTOL Vehicles: A Review of Basic Control Design Ideas and Principles,» *IEEE Control Systems Magazine*, vol. 33, n. 1, pp. 61–75, 2013.

[21] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton, NJ: Princeton University Press, 1999.

[22] T. Lee, M. Leok e N. H. McClamroch, «Geometric Tracking Control of a Quadrotor UAV on SE(3),» in *IEEE Conference on Decision and Control (CDC)*, Atlanta, GA, 2010, pp. 5420–5425.

[23] A. Lerro, «Meccanica del Volo — Dispense del corso,» Materiale didattico del Corso di Laurea Magistrale in Ingegneria Aerospaziale, Politecnico di Torino, anni accademici 2024/25, 2024.

[24] X. Lyu, H. Gu, J. Zhou, Z. Li, S. Shen e F. Zhang, «A Hierarchical Control Approach for a Quadrotor Tail-Sitter VTOL UAV and Experimental Verification,» in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5135–5141.

[25] R. Mahony, V. Kumar e P. Corke, «Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor,» *IEEE Robotics & Automation Magazine*, vol. 19, n. 3, pp. 20–32, 2012.

[26] D. D. Marshall, J. R. Gloudemans e R. A. McDonald, *VSPAERO: An Aerodynamic Analysis Tool for OpenVSP*, Modulo di analisi aerodinamica a pannelli integrato in OpenVSP. Available at: `https://openvsp.org/wiki/doku.php?id=vspaerotutorial` (Accessed: 25 March 2025), NASA Langley Research Center, 2024.

[27] *MATLAB R2024a Documentation*, Available at: `https://www.mathworks.com/help/matlab/` (Accessed: 10 April 2025), The MathWorks Inc., Natick, MA, 2024.

[28] R. A. McDonald e J. R. Gloudemans, «Open Vehicle Sketch Pad: An Open Source Parametric Geometry and Analysis Tool for Conceptual Aircraft Design,» in *AIAA SciTech Forum*, San Diego, CA, 2022. DOI: `10.2514/6.2022-0004`.

[29] D. Mellinger, «Trajectory Generation and Control for Quadrotors,» tesi di dott., University of Pennsylvania, 2012.

[30] D. Mellinger e V. Kumar, «Minimum Snap Trajectory Generation and Control for Quadrotors,» in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011, pp. 2520–2525.

[31] *MN5008 KV340 — Brushless Motor Datasheet*, Available at: `https://store.tmotor.com/product/mn5008-kv340-motor.html` (Accessed: 5 May 2025), T-Motor (Cube Mars Technology Co. Ltd.), Nanchang, China, 2023.

[32] R. C. Nelson, *Flight Stability and Automatic Control*, 2nd. Boston: McGraw-Hill, 1998.

[33] J. Nocedal e S. J. Wright, *Numerical Optimization*, 2nd. New York: Springer, 2006.

[34] *OpenVSP: Open Vehicle Sketch Pad — User's Manual, version 3.39*, Available at: `https://openvsp.org/` (Accessed: 25 March 2025), NASA Langley Research Center, 2024.

[35] W. F. Phillips, *Mechanics of Flight*, 2nd. Hoboken, NJ: John Wiley & Sons, 2010.

[36] W. H. Press, S. A. Teukolsky, W. T. Vetterling e B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd. Cambridge, UK: Cambridge University Press, 2007.

[37] R. Ritz e R. D'Andrea, «A Global Controller for Flying Wing Tailsitter Vehicles,» in *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 2731–2738. DOI: `10.1109/ICRA.2017.7989318`.

[38] J. Roskam, *Airplane Flight Dynamics and Automatic Flight Controls — Part I*. Lawrence, KS: DARcorporation, 2001.

[39] R. E. Sheldahl e P. C. Klimas, «Aerodynamic Characteristics of Seven Symmetrical Airfoil Sections Through 180-Degree Angle of Attack for Use in Aerodynamic Analysis of Vertical Axis Wind Turbines,» Sandia National Laboratories, Albuquerque, NM, rapp. tecn. SAND80-2114, 1981.

[40] S. W. Shepperd, «Quaternion from Rotation Matrix,» *Journal of Guidance and Control*, vol. 1, n. 3, pp. 223–224, 1978.

[41] K. Shoemake, «Animating Rotation with Quaternion Curves,» in *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '85)*, 1985, pp. 245–254.

[42] *Simulink Coder User's Guide — R2024a*, Available at: `https://www.mathworks.com/help/rtw/` (Accessed: 10 April 2025), The MathWorks Inc., Natick, MA, 2024.

[43] *Simulink User's Guide — R2024a*, Available at: `https://www.mathworks.com/help/simulink/` (Accessed: 10 April 2025), The MathWorks Inc., Natick, MA, 2024.

[44] E. J. J. Smeur, M. Bronz e G. C. H. E. de Croon, «Incremental Control and Guidance of Hybrid Aircraft Applied to a Tailsitter Unmanned Air Vehicle,» *Journal of Guidance, Control, and Dynamics*, vol. 43, n. 2, pp. 274–287, 2020. DOI: 10.2514/1.G004520.

[45] J. Solà, «Quaternion Kinematics for the Error-State Kalman Filter,» *arXiv preprint arXiv:1711.02508*, 2017.

[46] R. F. Stengel, *Flight Dynamics*. Princeton, NJ: Princeton University Press, 2004.

[47] B. L. Stevens, F. L. Lewis e E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd. Hoboken, NJ: John Wiley & Sons, 2015.

[48] R. H. Stone, «The T-Wing Tail-Sitter Research UAV,» in *2002 Biennial International Powered Lift Conference and Exhibit*, AIAA Paper 2002-5970, Williamsburg, VA, USA, 2002. DOI: 10.2514/6.2002-5970.

[49] S. Verling, B. Weibel, M. Boosfeld, K. Alexis, M. Burri e R. Siegwart, «Full Attitude Control of a VTOL Tailsitter UAV,» in *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016, pp. 3006–3012.

[50] B. Wie, *Space Vehicle Dynamics and Control*. Reston, VA: AIAA Education Series, 1998.

[51] H. Zhang, B. Song, H. Wang e J. Xuan, «A Method for Evaluating the Wind Disturbance Rejection Capability of a Hybrid UAV in the Quadrotor Mode,» *International Journal of Micro Air Vehicles*, vol. 11, pp. 1–19, 2019. DOI: 10.1177/1756829319869647.