Politecnico di Torino

SUPAIR

Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria Aerospaziale LM-20

Sessione di laurea Marzo 2026

Sviluppo e validazione di un modello dinamico per la taratura del controllore di volo mediante simulazione Hardware-In-The-Loop

Relatore:

Prof. Lerro Angelo

Laureando:

Pattacini Filippo

Tutor Aziendale:

Boukas Filippo

Anno Accademico 2025/2026

Indice

Elenco delle figure

Elenco delle tabelle

Abstract

Il presente lavoro di tesi ha l'obiettivo di sviluppare un modello dinamico di un UAV (Unmanned Aerial Vehicle), implementato in ambiente MATLAB/Simulink, finalizzato alla taratura dei parametri del controllore mediante l'impiego di una metodologia di simulazione Hardware-In-The-Loop (HIL). Nello specifico, sono stati utilizzati un velivolo ad ala fissa a propulsione elettrica, il *FunCub NG* prodotto da Multiplex®, e il flight controller CubePilot, basato su autopilota PX4. La gestione delle simulazioni e delle prove HIL è stata effettuata mediante il software QGroundControl, utilizzato come interfaccia di configurazione e monitoraggio del sistema.

Il modello di simulazione sviluppato è un modello lineare basato sulle equazioni del moto rigido a sei gradi di libertà e pone particolare attenzione alla ricostruzione della geometria, delle caratteristiche aerodinamiche e del sistema propulsivo del velivolo reale. La modellazione geometrica e aerodinamica del velivolo è stata realizzata mediante il software OpenVSP, mentre i coefficienti propulsivi sono stati integrati nel modello numerico al fine di riprodurre in modo coerente il comportamento dinamico osservato in volo. Tale modello è stato successivamente validato mediante confronto diretto delle principali variabili di stato e dei modi dinamici ottenuti in simulazione con quelli misurati durante le prove di volo sperimentali condotte sul velivolo reale.

La taratura dei parametri del controllore PID implementato sull'autopilota è stata effettuata mediante il metodo di Ziegler–Nichols all'interno di un ambiente di simulazione HIL. L'impiego di questa metodologia consente di ridurre i rischi associati alla taratura in volo, migliorando la sicurezza e l'affidabilità del processo di sviluppo. In tale contesto, la simulazione HIL si configura come uno strumento efficace per il testing integrato delle componenti software e hardware dell'autopilota e come una valida alternativa alle tradizionali procedure sperimentali in volo, favorendo l'integrazione di tecniche di simulazione avanzata nel workflow di progettazione dei sistemi di controllo per UAV.

Capitolo 1

Simulazione Hardware-in-the-Loop (HIL)

L'evoluzione delle tecnologie per la progettazione e il controllo di sistemi autonomi, come i droni, ha aperto nuove prospettive in numerosi settori quali logistica, monitoraggio ambientale, ispezioni industriali e sicurezza. Tuttavia, l'affidabilità e la sicurezza di questi sistemi rappresentano un requisito fondamentale per il loro impiego in contesti critici. Di conseguenza, lo sviluppo del software di controllo di un UAV costituisce una fase cruciale del progetto, poiché deve soddisfare stringenti requisiti di safety che impongono un'estesa attività di testing e validazione prima della sua implementazione sul sistema reale. Parallelamente, l'industria richiede tempi di progettazione, sviluppo e produzione sempre più ridotti, senza poter tollerare eventuali failure che potrebbero comportare significative perdite economiche. Per queste ragioni, nel settore aeronautico si è rapidamente affermato l'utilizzo della **simulazione Hardware-in-the-Loop (HIL)**.

L'HIL è una tecnica che consente di integrare componenti fisici reali all'interno di un ambiente simulato, creando un banco di prova nel quale è possibile testare algoritmi di controllo e moduli diagnostici in condizioni molto simili a quelle operative.
L'idea di base consiste nell'includere parte dell'hardware reale nel ciclo di simulazione durante lo sviluppo del sistema. Invece di validare l'algoritmo di controllo esclusivamente su un modello matematico del sistema, è possibile introdurre componenti fisici reali direttamente nel loop di simulazione. Un esempio tipico riguarda gli attuatori, notoriamente complessi da modellare: qualora disponibili, il loro impiego nella simulazione consente di aumentarne significativamente la validità.[1]

Prima di introdurre più nel dettaglio il concetto di Hardware-in-the-Loop, è utile definire alcune metodologie di simulazione affini, comunemente utilizzate nelle fasi di sviluppo precedenti a un test HIL [2]:

- **Software-in-the-Loop (SIL)**: la simulazione SIL differisce dalla HIL poiché consente di eseguire il codice di controllo senza la necessità di disporre dell'hardware fisico. Il codice compilato viene eseguito su hardware standard (tipicamente un PC), permettendo una prima validazione anche in assenza di prototipi reali. Ciò rende la simulazione più rapida, snella e facilmente ripetibile.
- **Processor-in-the-Loop (PIL)**: metodologia in cui il codice del controllore viene eseguito direttamente sul microprocessore del sistema target, mentre il modello del plant resta simulato sul PC. A differenza della HIL, non viene coinvolto l'intero hardware reale ma esclusivamente il processore, consentendo di valutare prestazioni computazionali e correttezza numerica del codice direttamente sul dispositivo finale.

La figura 1.1 fornisce una panoramica delle parti fondamentali di una simulazione HIL. Gli output di un cosiddetto simulatore HIL vengono utilizzati come input per il SUT (System Under Test). Gli output del SUT vengono utilizzati come input per il simulatore HIL. [3]

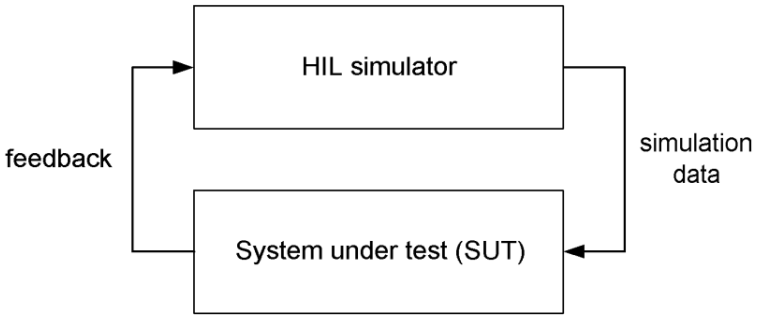


Figura 1.1: Schema di una simulazione HIL

Nello specifico la simulazione Hardware-in-the-Loop (HIL) sviluppata in questo progetto consente di integrare l'autopilota reale (PX4) all'interno di un ambiente di simulazione del velivolo, eseguendo un test completo del sistema di controllo in condizioni operative realistiche. In questo schema, il modello dinamico dell'UAV viene eseguito in real-time attraverso Simulink, mentre il flight controller fisico elabora in tempo reale i dati sensoriali simulati e genera i comandi degli attuatori, che vengono reinseriti nel modello del velivolo. Questo permette di validare l'intera catena sensori–controllore–attuatori rispettando i vincoli temporali reali del sistema e verificando la robustezza dell'algoritmo di controllo prima dell'impiego su un UAV reale. [4]

1.1 Vantaggi e Svantaggi della Simulazione HIL

Negli ultimi anni il concetto di simulazione Hardware-in-the-Loop (HIL) ha conosciuto una rapida diffusione, diventando un elemento imprescindibile in numerose fasi del ciclo di vita dei sistemi complessi, quali la progettazione, lo sviluppo, l'implementazione e il test. Le sue applicazioni si estendono ai settori aerospaziale, automobilistico, marittimo e della difesa, oltre che alla robotica e ai sistemi per la gestione dell'energia. L'impiego di una configurazione HIL nelle diverse fasi di progettazione consente di aumentare in modo significativo l'efficienza e l'affidabilità del sistema, permettendo di identificare e correggere tempestivamente numerosi errori, sia a livello software sia a livello hardware.[5]
La crescente adozione della metodologia HIL conferma i numerosi vantaggi associati al suo impiego, tra cui:

- Riduzione dei costi complessivi di sviluppo. Uno scenario è più facile da configurare in un ambiente virtuale che in un ambiente reale. Quindi, ad esempio, sono necessarie meno prove su strada o esperimenti su banco prova, e anche meno prototipi di veicoli.
- Diminuzione del tempo che intercorre tra fase progettuale e produzione, accelerando il processo di simulazione e validazione.
- Possibilità di sviluppare e validare algoritmi e sistemi di controllo in condizioni altamente realistiche, prossime a quelle operative, ma all'interno di un ambiente di laboratorio controllato, che consente anche la sperimentazione di scenari ambientali critici.
- Capacità di analizzare condizioni complesse relative ad attuatori e sensori e di studiare le interazioni reciproche tra software e hardware.
- Individuazione delle principali fonti di errore e valutazione della tolleranza ai guasti del sistema, introducendo malfunzionamenti deliberati nei vari sottosistemi.
- Possibilità di ripetere gli scenari di test in modo sistematico e di automatizzare campagne di verifica anche molto lunghe.
- Possibilità di condurre test non distruttivi, migliorando la sicurezza del processo di sviluppo e riducendo sensibilmente il rischio di danni economici.

Tutti i benefici sopra elencati possono essere ottenuti con un costo relativamente contenuto nelle principali fasi di progettazione e con esigenze di manutenzione generalmente trascurabili.

Per quanto riguarda gli svantaggi, i principali limiti dei sistemi HIL possono essere sintetizzati come segue:

- Assenza di soluzioni completamente standardizzate, con conseguente complessità e lentezza nelle fasi di integrazione dei diversi componenti.
- Impossibilità di ottenere con precisione informazioni sul modulo responsabile dell'errore, poiché la piattaforma HIL opera tipicamente come un *black box tester*.
- Tempi di preparazione elevati e procedure complesse per la definizione degli scenari di test relativi a condizioni anomale o di guasto, con configurazioni e prestazioni del simulatore che possono limitare l'estensione e la realizzabilità delle condizioni di simulazione.

1.2 Architettura

In questa sezione viene approfondita l'architettura del sistema Hardware-in-the-Loop (HIL) implementato, seguendo quanto descritto nella documentazione MathWorks alla sezione `"PX4 Hardware-in-the-Loop System Architecture"` [6].

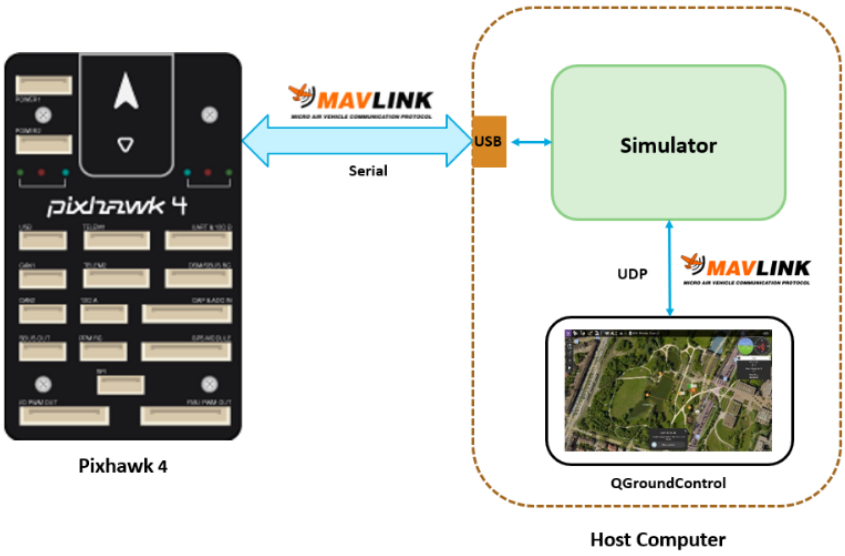


Figura 1.2: Configurazione fisica dei componenti in una simulazione HIL

In figura 1.2 sono illustrate le principali comunicazioni tra i tre componenti fondamentali dell'architettura HIL.
Nella configurazione adottata, il flight controller PX4 è connesso al computer host esclusivamente tramite USB. Sul computer risultano in esecuzione sia il modello MATLAB/Simulink, responsabile della simulazione dell'evoluzione dinamica dell'UAV e della generazione dei dati sensoriali inviati a PX4, sia QGroundControl (QGC), utilizzato come interfaccia di monitoraggio per visualizzare telemetria, assetto, posizione e per registrare i log di volo, oltre a consentire l'eventuale invio di comandi manuali tramite joystick.
Durante la simulazione HIL, sia QGroundControl sia il modello Simulink necessitano di comunicare con l'hardware PX4 tramite porta seriale. Tuttavia, tale porta può essere occupata da una sola applicazione alla volta. Per ovviare a questo vincolo, il simulatore stabilisce direttamente una connessione seriale MAVLink con l'autopilota PX4 e, in parallelo, apre un endpoint UDP accessibile da QGroundControl mediante i blocchi `"MAVLink Bridge Source"` e `"MAVLink Bridge Sink"`. Questi blocchi fungono da vero e proprio bridge, inoltrando i pacchetti MAVLink tra PX4 e QGroundControl.

Ne consegue che QGroundControl può comunicare con l'autopilota solo quando il simulatore è attivo; a tal fine, nelle impostazioni di QGC è stata abilitata esclusivamente la connessione via UDP. La configurazione complessiva del workflow di comunicazione è riportata in figura 1.3.

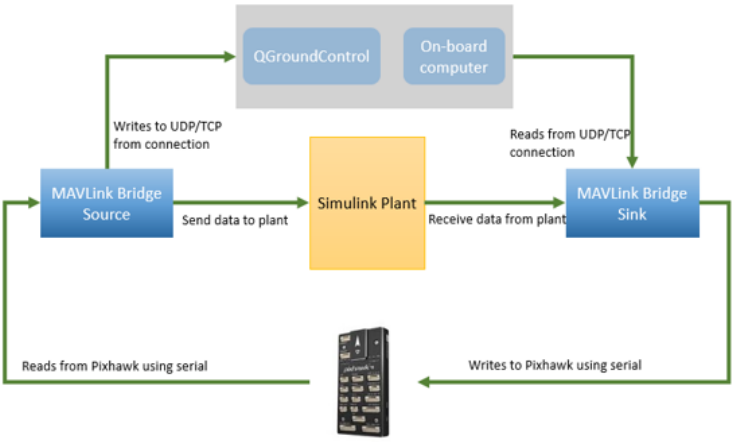


Figura 1.3: Interfaccia di alto livello tra più componenti utilizzata nel workflow HIL

1.2.1 Comunicazione MAVLink

All'interno della simulazione HIL, tutte le comunicazioni tra i diversi componenti e tra i vari sottosistemi avvengono tramite messaggi **MAVLink**. MAVLink (Micro Air Vehicle Link) è un protocollo di comunicazione ampiamente utilizzato in ambito UAV che standardizza lo scambio bidirezionale di messaggi tra PX4 e Simulink, nonché tra PX4 e QGroundControl (tramite bridge). Ogni messaggio MAVLink presenta quindi una struttura uniforme, riportata in figura 1.4.
Di particolare interesse risultano l'ID del messaggio, contenuto nel sesto byte e responsabile dell'identificazione e della corretta decodifica del pacchetto, e il payload, che dai byte successivi contiene le informazioni effettive da trasmettere.

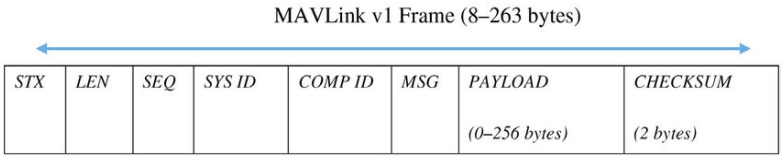


Figura 1.4: Struttura di un messaggio MAVLink

Decodifica lato Simulink (Plant)

La comunicazione tra SIMULINK e PX4 è completamente bidirezionale: il simulatore riceve in ingresso il messaggio MAVLink `"HIL_ACTUATOR_CONTROLS"`, contenente i comandi diretti agli attuatori, mentre fornisce in uscita all'hardware PX4 i messaggi `"HIL_SENSOR"`, `"HIL_GPS"` e `"HEARTBEAT"`, contenenti i dati sensoriali simulati.
Tale scambio è reso possibile dagli appositi blocchi per la comunicazione MAVLink inclusi nei toolbox `"UAV Toolbox"` e `"UAV Toolbox Support Package for PX4 Autopilots"` [7, 8].

I messaggi MAVLink provenienti dal PX4 vengono letti dal blocco `"MAVLink Bridge Source"` tramite la porta seriale *"/dev/ttyACM0"*, il quale fornisce in uscita il vettore di byte grezzi e lo scalare *"DataLength"*, indicante il numero di byte validi.
Tali valori vengono passati al blocco `"Input Signal Conditioning"`, incaricato della gestione del buffer: poiché il vettore ha dimensione fissa, solo i primi *DataLength* elementi risultano significativi, in assenza di dati viene restituito zero.
Il segnale viene poi fornito al blocco `"MAVLink Deserializer"`, che converte il buffer di byte in un bus non virtuale SIMULINK contenente l'intero pacchetto MAVLink, comprensivo di ID messaggio, ID sistema, ID componente, sequenza e le informazioni del payload corrispondenti al messaggio MAVLink `"HIL_ACTUATOR_CONTROLS"`, ovvero il tipo di messaggio MAVLink selezionato nel blocco.
Infine, il blocco `"Bus Selector"` estrae la variabile `"controls"`, contenente i valori adimensionalizzati delle deflessioni delle superfici mobili e della manetta.

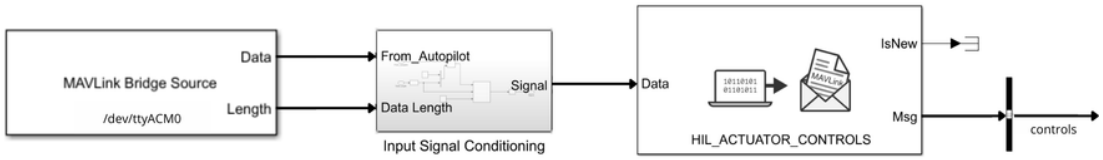


Figura 1.5: Flusso di comunicazione PX4 → Simulink

Analogamente, i dati sensoriali generati dal modello dinamico del velivolo `"UAV_Dynamics"` vengono impacchettati in bus SIMULINK mediante il blocco `"MAVLink Blank Message"`. Questi bus attraversano il blocco `"Rate Transition"`, che gestisce il trasferimento dei dati ad un determinato rate, e vengono infine passati al blocco `"MAVLink Serializer"`, il quale li converte nei messaggi MAVLink `"HIL_SENSOR"` e `"HIL_GPS"`.
I pacchetti così generati vengono inviati al blocco `"MAVLink Bridge Sink"`, che li unisce ai dati provenienti dalle connessioni UDP (QGROUNDCONTROL) e invia il flusso risultante all'hardware PX4 tramite la porta seriale.

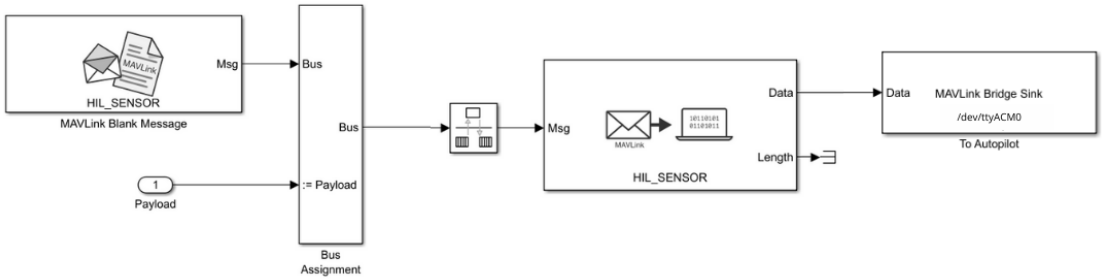


Figura 1.6: Flusso di comunicazione Simulink → PX4

Bridge tramite UDP

Il blocco `"MAVLink Bridge Source"`, oltre alla configurazione della porta seriale da cui riceve i pacchetti MAVLink, consente l'apertura di una connessione UDP tramite il parametro *Destination Ports*, impostato a ('UDP', '127.0.0.1', '14550'). Tale connessione permette l'inoltro dei messaggi verso QGroundControl, che non comunica direttamente con PX4.

In modo del tutto analogo, il blocco `"MAVLink Bridge Sink"` presenta il parametro *Source Ports*, anch'esso configurato come ('UDP', '127.0.0.1', '14550'), attraverso il quale riceve i dati provenienti da QGroundControl. Questi vengono poi combinati con quelli generati dal modello Simulink e inviati al controller PX4 tramite la porta seriale *"/dev/ttyACM0"*.

1.2.2 Sample Time

È stata riscontrata una particolare importanza nella corretta scelta del *sample time* da impostare nella simulazione MATLAB/Simulink. Tale parametro determina infatti la frequenza di aggiornamento con cui i segnali vengono elaborati e scambiati all'interno del modello Simulink.
Durante la simulazione HIL è stato osservato che il sistema risultava stabile soltanto riducendo il *sample time* di Simulink da 0.01 s a 0.002 s. Sebbene, in linea teorica, un passo di integrazione più ampio comporti un minore carico computazionale, nella pratica l'interazione in tempo reale con il firmware PX4 ha evidenziato che tale configurazione introduceva ritardi e irregolarità temporali nei pacchetti di dati sensoriali inviati al controllore.
Il firmware PX4, infatti, esegue i propri cicli di stima e controllo a frequenze tipicamente comprese tra 200 e 400 Hz e si aspetta un flusso di dati coerente con tali tempi di aggiornamento. Quando la frequenza di esecuzione di Simulink risultava inferiore, il controllore riceveva informazioni meno aggiornate e soggette a *jitter*

temporale, con conseguenti instabilità numeriche e discrepanze di sincronizzazione tra simulazione e firmware.
Riducendo il *sample time* a 0.002 s (pari a una frequenza di 500 Hz), la latenza media e le variazioni di *timing* sono state significativamente ridotte. Ciò ha consentito a SIMULINK di fornire dati maggiormente sincronizzati con i loop interni di PX4, ripristinando la stabilità complessiva della simulazione HIL e garantendo una maggiore coerenza dinamica tra modello e controllore reale.

Capitolo 2

Strumenti utilizzati

In questo capitolo vengono introdotti i software e gli strumenti utilizzati durante il progetto, al fine di presentarne l'utilità e il funzionamento generale nei diversi ambienti di lavoro considerati.

2.1 PX4

L'autopilota può essere considerato il "cervello" del drone. Esso consiste in un software di controllo di volo che utilizza un sistema operativo real-time ed è implementato su un flight controller, ovvero la parte hardware.
PX4 è esattamente il software alla base dell'autopilota, è un software open-source che utilizza NuttX come RTOS ed è particolarmente apprezzato per le sue caratteristiche. Tra queste:

- Supporta diverse tipologie di veicoli (fixed-wing, multicopter, VTOL, ...) utilizzando lo stesso codice sorgente.
- Ha una vasta scelta di componenti hardware con i quali può essere integrato, come flight controller e sensori.
- Supporta potenti *Flight Modes*
- Flessibilità e sicurezza.

PX4 si compone di due parti fondamentali: il **flight stack** che consiste nell'estimatore e controllore di volo vero e proprio, and il **middleware** che fornisce l'infrastruttura di comunicazione tra il flight stack e l'hardware esterno.[9]

2.1.1 Flight Stack

Il flight stack è l'insieme degli algoritmi di guida, navigazione e controllo per l'UAV. In figura 2.1 è rappresentato il diagramma che mostra il principio di funzionamento del flight stack.

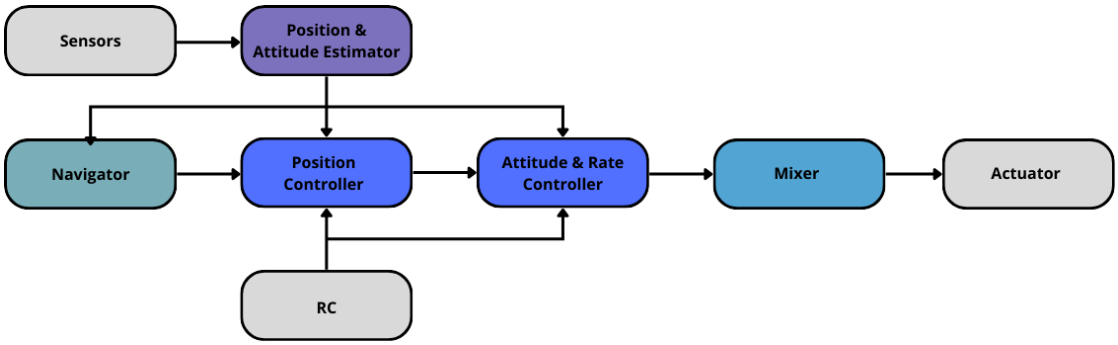


Figura 2.1: Flight Stack

All'interno di esso entrano in gioco diversi moduli:

- Estimator: sfrutta i dati provenienti dai sensori, combinandoli, e stima lo stato attuale dell'UAV.
- Navigator: si occupa della gestione delle missioni e delle modalità di volo automatiche. In base allo stato stimato genera i *setpoint* della traiettoria.
- Controller: In funzione del *setpoint* o dei comandi manuali, il controllore si occupa di generare i comandi che permettono di correggere l'attuale stato del velivolo e raggiungere lo stato desiderato.
- Mixer: legge i comandi di forza (per esempio "spinta verso destra") e li traduce in comandi per i singoli attuatori.

Il sistema di controllo del volo fornisce funzioni essenziali di stabilizzazione e sicurezza, oltre che assistenza al pilota per il volo manuale e la gestione dei comandi per il volo automatico.

2.1.2 Middleware

Il middleware di PX4 rappresenta lo strato software intermedio che consente al flight stack di operare comunicando con i sensori, gli attuatori e le interfacce di comunicazione esterne. Esso è ciò che permette al sistema di essere altamente estensibile, scalabile e portabile su numerose piattaforme hardware.
Il middleware svolge tre funzioni principali:

- Gestione della comunicazione interna tra i moduli del flight stack tramite oggetto `uORB`. Un messaggio uORB (Micro Object Request Broker) è una struttura dati definita all'interno di un file *.msg* che rappresenta un *topic* specifico utilizzato per lo scambio di informazioni tra moduli. Ogni file .msg contiene il nome dei campi e i tipi di dato, oltre alla struttura *timestamp* essenziale come riferimento temporale per il logging.
- HAL (Hardware Abstraction Layer) che consente al flight stack di essere indipendente dalla piattaforma su cui viene eseguito, interagendo solo con API astratte.
- Integrazione di protocolli che permettono a PX4 di interagire con altri sistemi.

2.1.3 Flight Controller

Il Flight Controller è l'hardware sul quale viene implementato ed eseguito il firmware di PX4. PX4 può essere eseguito su vari hardware supportati, il FC viene quindi scelto in base a vincoli fisici e alle attività che si vogliono performare. Nel nostro caso di studio è stato utilizzato il CubePilot `Cube Orange +`.

Figura 2.2: Cube Orange +

Il flight controller Cube Orange+ è un autopilota flessibile, progettato per essere utilizzato con una scheda carrier ADS-B al fine di ridurre il cablaggio, migliorare

l'affidabilità e facilitare l'assemblaggio. Esso è equipaggiato con un processore dual-core (`ARM Cortex M7+M4`) in modo da garantire stabilità e alte prestazione e da un failsafe co-processor, al fine di incrementare la sicurezza. [10]. All'interno dispone di vari sensori connessi via SPI, tra i quali tre accelerometri, tre giroscopi , un magnetometro e due sensori di pressione barometrica
In figura 2.3 sono rappresentate le porte presenti sulla scheda carrier. Oltre a queste sono presenti anche una porta microUSB, che verrà usata per il collegamento seriale dell'autopilota al computer host, e l'alloggiamento per la scheda microSD necessaria a registrare i log di volo.

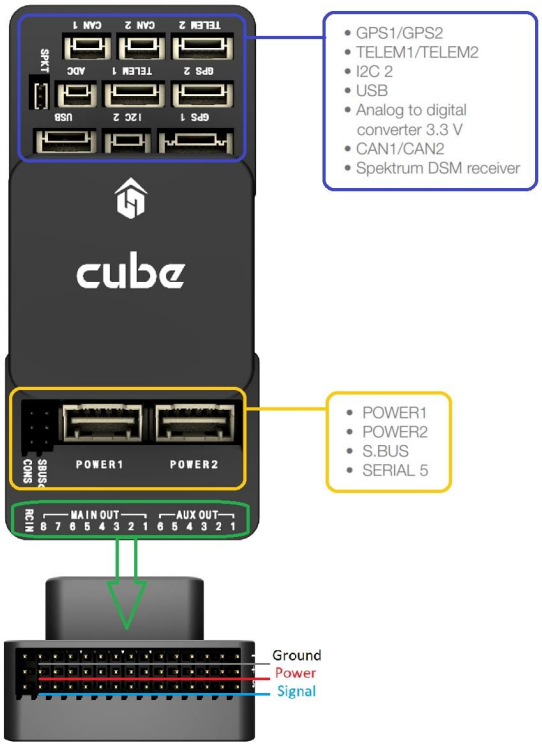


Figura 2.3: Porte del Cube Orange +

2.1.4 Flight Modes

I diversi *flight modes* presenti in PX4 forniscono supporto per semplificare il pilotaggio del velivolo in modalità manuale. Ognuna di queste modalità rappresenta un diverso livello di controllo.

- **Manual**: Il controllo del velivolo è completamente manuale ed è gestito dagli input forniti tramite radiocomando o joystick, che controllano direttamente la deflessione delle superfici mobili.
- **Acro**: Modalità manuale nella quale gli stick dei comandi controllano le velocità angolari attorno ai rispettivi assi. Quando le leve dei comandi sono in posizione neutra, il velivolo smette di ruotare ma mantiene l'orientamento raggiunto.
- **Stabilized**: Modalità che fornisce un livello base di assistenza al pilota, il quale controlla direttamente gli angoli di rollio e beccheggio. Il controllore mantiene i valori impostati finché non vengono impartiti nuovi comandi tramite gli stick.
- **Altitude**: Modalità manuale assistita nella quale il comando di beccheggio controlla la velocità di salita o discesa e, se lasciato in posizione neutra, consente di mantenere la quota. Il pilota controlla invece direttamente l'angolo di rollio, senza alcun intervento del controllore sul moto orizzontale.
- **Position**: Modalità nella quale il velivolo è in grado di mantenere automaticamente la traiettoria e la posizione orizzontale, compensando disturbi esterni. Il comando di rollio impartito dal pilota genera una virata coordinata, mentre il comando di beccheggio controlla la velocità di salita o discesa. In assenza di input, il velivolo si riporta automaticamente in assetto livellato e mantiene una rotta rettilinea.

2.2 QGroundControl

Le *Ground Control Stations (GCS)* sono sistemi che permettono agli operatori di UAV di monitorare e controllare un drone e i suoi payloads. La GCS utilizzata nel nostro caso è QGroundControl (QGC) il quale offre un interfaccia completa per il controllo e il monitoraggio di veicoli alimentati da PX4.[11]
Tra le varie caratteristiche di QGroundControl troviamo:

- Set up completo per UAV che utilizzano PX4. La configurazione include l'upload del Firmware, la configurazione dei parametri per la missione, la definizione dell'airframe del veicolo, la configurazione e assegnazione degli attuatori e il tuning del controllore.
- Pianificazione delle missioni per il volo autonomo dell'UAV. La definizione del piano di volo si trova alla sezione `"Fly Plan"`, nella quale possono essere scelti vari *waypoints*, oltre al punto di decollo e di atterraggio.
- Visualizzazione della mappa di volo che mostra la posizione del veicolo, i waypoint e gli strumenti di volo.
- Visualizzazione 3D della traiettoria e possibilità di attivare lo streaming video.
- Analisi dei dati di volo tramite `"MAVLink Inspector`, che consente di visualizzare i messaggi MAVLink provenienti da PX4. Permette inoltre di scaricare i file `.ulog` all'interno dei quali sono presenti i log di volo.
- Connessione del joystick per il volo manuale.

In figura 2.4 è riportata la schermata principale di QGC, utilizzata per il monitoraggio del velivolo. In particolare evidenziamo:

- Flight Telemetry: dove vengono mostrate altitudine (rispetto alla posizione di partenza), velocità orizzontale e verticale, tempo totale di volo e distanza tra il veicolo e la stazione a terra.
- Compass/Attitude: widget che fornisce informazioni sull'heading e sull'orizzonte virtuale.
- Map: mostra la posizione del veicolo collegato e l'attuale missione.
- Toolbar: barra di strumenti che permetti di impostare la missione, impostare il *flight mode* e definire i vari parametri del velivolo, oltre a permettere la visualizzazione del *flight status*, dei messaggi del velivolo e dello status dei vari componenti.

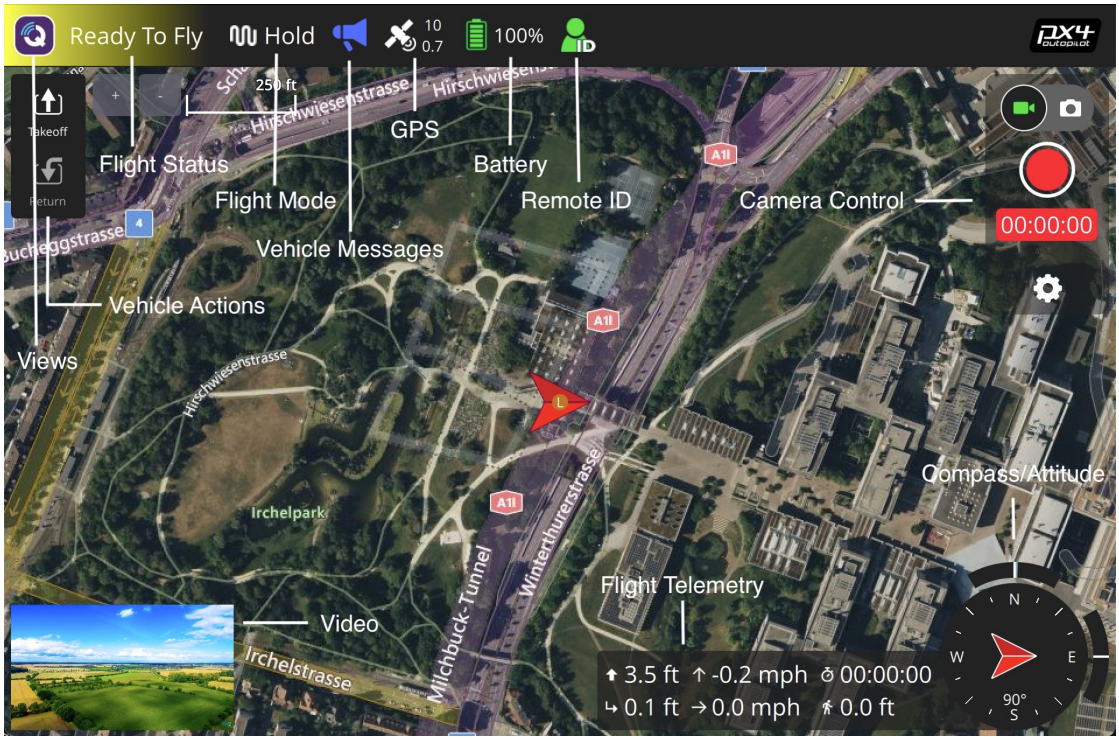


Figura 2.4: QGroundControl Fly View

2.3 Modello Simulink

In figura 2.5 è schematizzato l'intero modello SIMULINK utilizzato per la simulazione HIL.

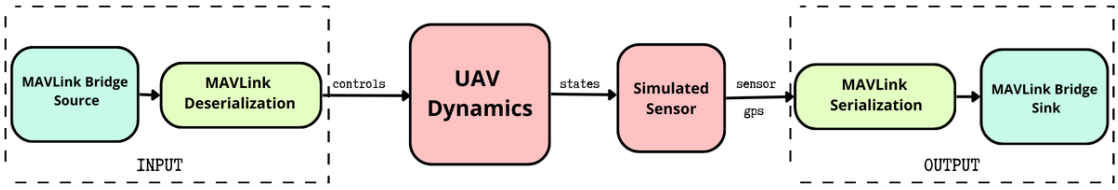


Figura 2.5: Rappresentazione modello Simulink

I blocchi riguardanti la gestione degli input e degli output, e quindi la comunicazione con PX4, sono stati trattati all'interno della sezione 1.2, nella quale è stata descritta l'implementazione e la gestione della comunicazione `MAVLink`. In questa sezione verrano invece trattati i due macro-blocchi centrali del modello, ovvero `"UAV Dynamics"`, che risolve le equazione della dinamica per calcolare lo stato del velivolo ad ogni iterazione, e `"Simulated Sensor"` che genera i dati dei sensori simulati.

2.3.1 UAV Dynamics

Il sottosistema `"UAV Dynamics"` è riportato in figura 2.6. Esso calcola, a ogni passo di integrazione, l'evoluzione delle variabili di stato del velivolo a partire dai comandi di controllo generati da PX4. Il blocco implementa il modello dinamico completo dell'UAV, includendo la dinamica traslazionale e rotazionale, e fornisce in uscita posizione, velocità, accelerazione, assetto e velocità angolari del velivolo.

Ad ogni iterazione temporale il modello dinamico calcola le forze e i momenti agenti sul velivolo in funzione dello stato dell'UAV al passo di integrazione precedente e dei comandi forniti dall'autopilota *PX4*. In figura 2.7 è riportato lo schema del macro-blocco `"Force and Moment calculation"`.
Il segnale `controls` proveniente da PX4 è un vettore contenente i valori adimensionalizzati delle deflessioni delle superfici mobili e della manetta. Il blocco `"Actuator Model"` presente in figura 2.7 è quindi necessario per convertire il valore adimensionalizzato della deflessione delle superfici (tra -1 e 1) in un effettivo angolo espresso in radianti. Per ogni superficie quindi il valore adimensionale viene moltiplicato per l'angolo di deflessione massimo della rispettiva superficie.
Successivamente i 4 segnali vengono fatti passare attraverso il blocco `"Linear Second-Order Actuator"` il quale modella la dinamica di un attuatore reale come

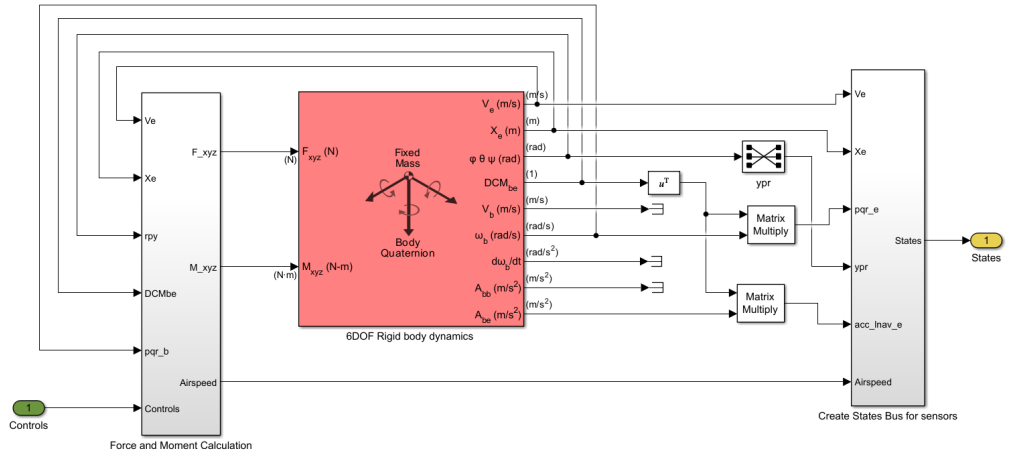


Figura 2.6: UAV Dynamics

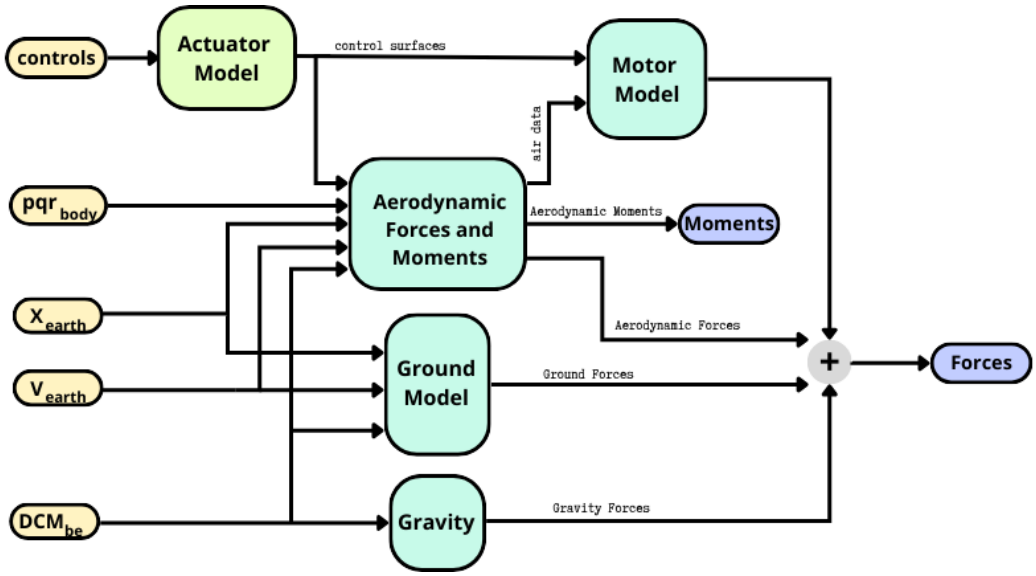


Figura 2.7: Force and Moment calculation

un sistema lineare del secondo ordine, riproducendo il comportamento inerziale e smorzato dell'attuatore, ovvero il ritardo, la velocità di risposta e le eventuali oscillazioni tra il comando in ingresso e la posizione effettivamente applicata.

Dopodichè si passa al calcolo delle forze, che possono essere suddivise in quattro contributi principali; ciascuno di essi viene inizialmente definito nel sistema di riferimento più opportuno e successivamente riportato nel sistema di riferimento *Body* tramite le matrici di rotazione descritte in Appendice A.

- **Forza di gravità**: La forza peso agisce costantemente lungo l'asse z del sistema di riferimento inerziale. Per essere correttamente utilizzata all'interno delle equazioni del moto del velivolo, essa viene trasformata nel sistema di riferimento solidale al corpo mediante la matrice di rotazione DCM_{be}, funzione degli angoli di assetto.

- **Spinta del motore**: La spinta propulsiva è calcolata tramite un modello del gruppo propulsivo sviluppato ad hoc e descritto in dettaglio nella Sezione 3.2. In questa fase la spinta generata dal motore viene assunta sempre allineata con l'asse longitudinale X_{body} del velivolo.

- **Forze di contatto al suolo**: Il modello di contatto al suolo è finalizzato a rappresentare le interazioni tra il velivolo e il terreno durante le fasi di decollo, atterraggio e rullaggio. Il velivolo viene approssimato come una massa puntiforme rigida, mentre il contatto verticale con il suolo è modellato tramite un sistema molla–smorzatore. La reazione normale è espressa come:

$$F_n = k \cdot (-z_{earth}) + c \cdot (-v_{z_{earth}}) \tag{2.1}$$

  dove k rappresenta la rigidezza equivalente del suolo e c il coefficiente di smorzamento. La posizione e la velocità verticali vengono espresse con segno invertito in modo tale che una penetrazione del suolo produca una forza normale positiva verso l'alto. La reazione normale viene attivata esclusivamente in condizioni di contatto, ovvero per $-z_{earth} > 0$. Una volta calcolata la forza normale, viene determinata la forza di attrito tramite un modello di tipo *Soft Coulomb friction*, nel quale il coefficiente di attrito è una funzione continua della velocità di scorrimento, modellata mediante una funzione tangente iperbolica [12]. Le componenti della forza di attrito lungo gli assi orizzontali del sistema *Earth* risultano quindi:

$$\begin{aligned} F_x &= \mu(v_{x_{earth}})\, F_n \\ F_y &= \mu(v_{y_{earth}})\, F_n \end{aligned} \tag{2.2}$$

  Le forze di attrito così ottenute vengono successivamente invertite di segno, in quanto agiscono in direzione opposta alla velocità di scorrimento relativa.

Analogamente, anche la forza normale viene orientata coerentemente con il sistema di riferimento solidale alla Terra.

Infine, il vettore complessivo delle forze di contatto al suolo viene trasformato nel sistema di riferimento *Body* mediante la matrice di rotazione:

$$F_{ground} = DCM_{be} \begin{pmatrix} -F_x \\ -F_y \\ -F_n \end{pmatrix} \tag{2.3}$$

- **Forze e momenti aerodinamici**: Il calcolo dei contributi aerodinamici richiede preliminarmente la determinazione delle proprietà dell'aria. In funzione della coordinata verticale z_{earth} viene quindi calcolata la densità dell'aria mediante il modello atmosferico standard ISA. Il vettore velocità del velivolo, inizialmente espresso nel sistema di riferimento *Earth*, viene successivamente trasformato nel sistema *Body*, consentendo il calcolo dell'angolo di incidenza α, dell'angolo di sideslip β e del modulo della velocità in assi vento.

A partire da tali grandezze, dalle deflessioni delle superfici di controllo e dalle velocità angolari del velivolo, vengono calcolati i contributi delle forze e dei momenti aerodinamici nei piani longitudinale e latero-direzionale. Il modello aerodinamico adottato si basa sulla linearizzazione delle equazioni della dinamica attorno alla condizione di equilibrio e sull'impiego dei coefficienti aerodinamici adimensionali identificati in Sezione 3.4. Le forze e i momenti aerodinamici vengono quindi determinati nel sistema di riferimento degli assi vento mediante le equazioni 2.4 per la dinamica longitudinale e le equazioni 2.5 per la dinamica latero-direzionale:

$$\begin{aligned} L &= \frac{1}{2}\rho V^2 S C_L \\ D &= \frac{1}{2}\rho V^2 S C_D \\ \mathcal{M} &= \frac{1}{2}\rho V^2 S c C_m \end{aligned} \tag{2.4}$$

$$\begin{aligned} Y &= \frac{1}{2}\rho V^2 S C_Y \\ \mathcal{L} &= \frac{1}{2}\rho V^2 S b C_l \\ \mathcal{N} &= \frac{1}{2}\rho V^2 S b C_n \end{aligned} \tag{2.5}$$

In Appendice B sono approfondite le conseguenze della linearizzazione del modello dinamico e i contributi che compongono i singoli coefficienti aerodinamici di forza e di momento.
Le forze e i momenti così ottenuti vengono infine trasformati nel sistema di riferimento *Body* mediante l'applicazione della matrice di rotazione $RW2B$.

Dopo aver calcolato forze e momenti agenti sull'UAV nel s.d.r *body*, questi vengono passati al blocco `"6DOF Rigid body dynamics"` il quale integra le equazioni di moto rigido del corpo calcolando posizione, velocità, accelerazione, assetto e velocità angolari allo step temporale successivo [13]. Il blocco presuppone che le forze applicate agiscano nel centro di gravità del corpo e che la massa e l'inerzia siano costanti. In appendice B vengono riportate le equazioni di eulero e la loro integrazione.
Per effettuare l'integrazione è necessario fornire al blocco le condizioni iniziali del velivolo, oltre che i dati di massa e inerzie.
I dati in uscita verrano poi utilizzati per il calcolo delle forze e dei momenti al passo temporale successivo, e per formare il bus nominato `states` il quale verrà poi passato al macro-blocco `"Simulated sensors"` come si pùo osservare dalla figura 2.5.

2.3.2 Simulated Sensors

Il sottosistema `Simulated Sensors`, mostrato in Figura 2.8, ha il compito di generare i segnali dei sensori simulati a partire dallo stato del velivolo calcolato dal modello dinamico. Le grandezze fisiche ideali vengono opportunamente perturbate mediante l'aggiunta di rumore e bias, così da riprodurre in maniera realistica il comportamento dei sensori reali installati a bordo dell'UAV.
Il sottosistema è composto dai seguenti blocchi principali:

- **IMU Simulation**: il blocco legge i valori di accelerazione, velocità angolare e assetto del velivolo (espresso in quaternioni) e li fornisce al blocco `IMU`, che genera misure *sensor-like* di un'unità di misura inerziale. Il modello include effetti di rumore, bias e saturazione, consentendo di simulare il comportamento realistico di accelerometri e giroscopi.

- **GNSS/GPS Simulation**: il blocco genera misurazioni di posizione e velocità affette da rumore a partire dalle coordinate e dalle velocità espresse nel sistema di riferimento locale. Le coordinate vengono successivamente convertite nel sistema geodetico latitudine–longitudine–altitudine (LLA) mediante l'utilizzo del modello terrestre WGS84.

- **Baro Simulation**: a partire dal valore di quota fornito dal modello di volo, viene aggiunto rumore per simulare l'uscita di un sensore barometrico. Il

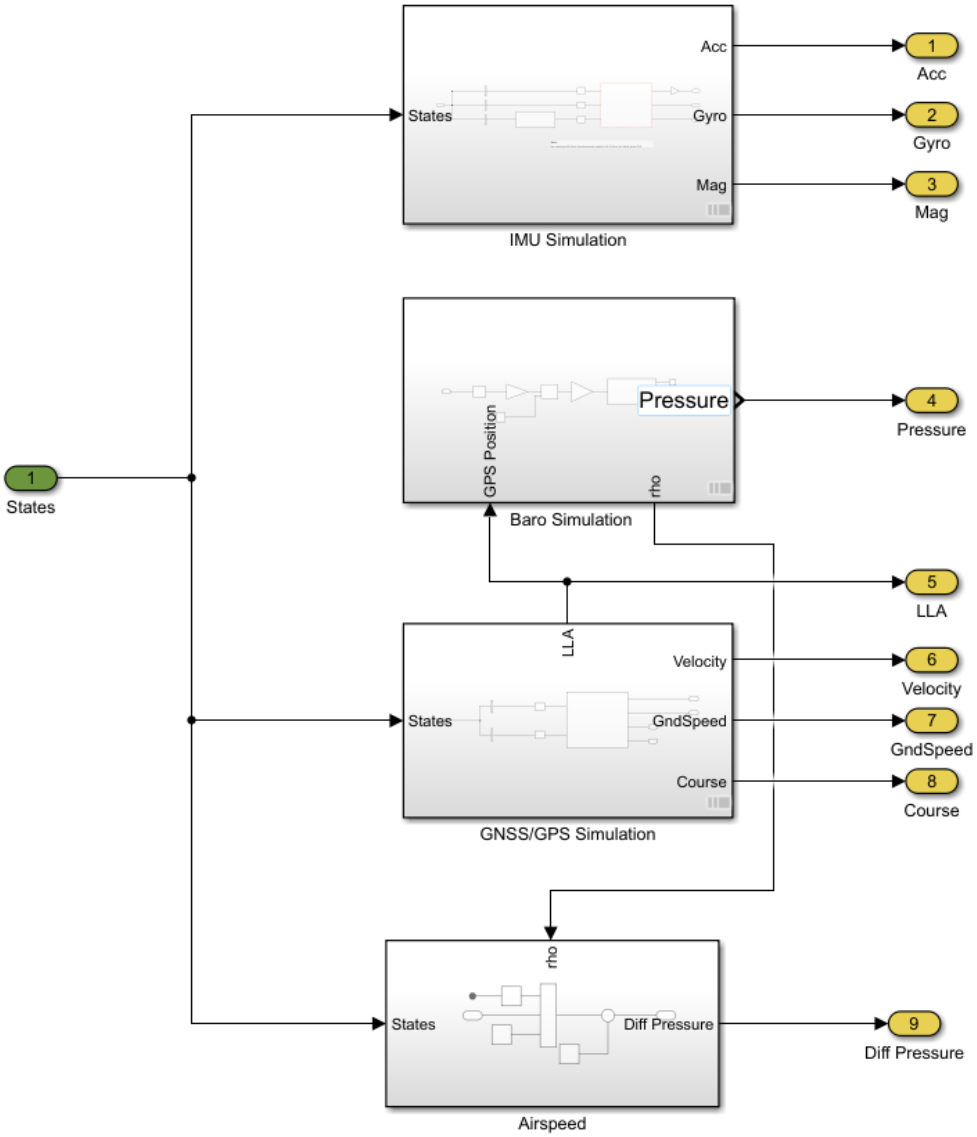


Figura 2.8: Sottosistema `Simulated Sensors` in Simulink

segnale così ottenuto viene successivamente fornito al blocco `Lapse Rate Model`, che calcola i corrispondenti valori di pressione e densità atmosferica in accordo con un modello standard dell'atmosfera.

- **Airspeed**: il blocco calcola la pressione dinamica a partire dalla velocità del velivolo, ricavata dal bus `states`, e dalla pressione statica ottenuta dal sottosistema *Baro Simulation*. A tale grandezza viene infine sommato un opportuno rumore per simulare il segnale misurato da un sensore di velocità dell'aria.

Capitolo 3

Modellazione dell'UAV

In questo capitolo viene presentata una descrizione del velivolo ***FunCub***, utilizzato durante le prove di volo, e delle tecniche di modellazione impiegate per l'estrazione dei dati necessari all'implementazione del modello numerico in MATLAB/Simulink.

3.1 Descrizione

Il ***FunCub NG (Next Generation)*** è un aeromodello a propulsione elettrica prodotto da Multiplex ®, noto per le sue eccellenti qualità di volo e un'elevata manovrabilità del modello. [14]

Una delle principali caratteristiche di questo modello, comune a molti prodotti Multiplex ®, è l'impiego del materiale `ELAPOR`, un polipropilene espanso che conferisce alla struttura una notevole leggerezza combinata a un'elevata elasticità meccanica.
Il FunCub presenta un'ala rettangolare con apertura alare pari a 1410 mm e corda media di 216 mm. Il controllo aerodinamico è garantito da un totale di sei servoattuatori, che azionano le quattro superfici mobili: quattro servo `HS-55+` dedicati agli alettoni e due servo `HS-65HB` per il comando dell'equilibratore e del timone.
La fusoliera, realizzata con una struttura cava interna, consente l'alloggiamento del power set dedicato (descritto in sezione 3.2), della batteria di alimentazione, del `CubeOrange+` e della relativa componentistica elettronica e cablaggi.

Figura 3.1: Illustrazione del FunCub NG

3.2 Modello Propulsivo

Il sistema propulsivo installato offre ampie riserve di potenza [15]. Esso è infatti costituito da:

- Motore: `ROXXY BL Outrunner C-35-42-930kv FUNCUB NG`
- ESC: `ROXXY BL Control 740 S-BEC`
- Elica: `APC Propeller 13" x 4"`
- Batteria: `ManiaX Batteria Lipo 4S 14,8V 4200mA`

Questi componenti caratterizzano un sistema propulsivo versatile, in grado di fornire ottime prestazioni mantenendo al contempo una notevole leggerezza.
I dettagli tecnici dei singoli elementi sono riassunti in tabella 3.1.

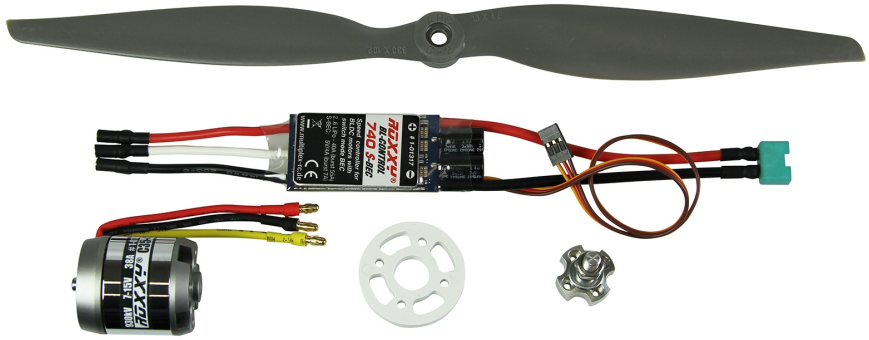


Figura 3.2: Power Kit FunCub NG

K_V	930 rpm/V
Potenza massima P_{max}	680 W
Corrente a vuoto I_0	2 A
Corrente massima I_{max}	55 A

(a) Motore

Capacità nominale Q_{nom}	4200 mAh
Tensione a vuoto V_{oc}	14.8 V
Resistenza interna R_m	0.013 Ω
R_{pol}	0.03 Ω
C_{pol}	80 F

(b) Batteria

Tabella 3.1: Dati tecnici del sistema propulsivo

Conoscendo le specifiche di tali componenti è stato sviluppato un modello matematico dedicato, mostrato in figura 3.3, con l'obiettivo di ricavare le prestazioni del sistema propulsivo a ogni passo temporale, tenendo conto di diversi fattori.
In particolare, viene calcolata la spinta prodotta dal sistema elica–motore al variare della potenza richiesta dal comando di manetta. Tale spinta dipende inoltre dalla velocità di volo dell'UAV, dallo stato di carica della batteria e dal calo di tensione dovuto ai transitori.
La spinta generata viene considerata allineata all'asse X_{body} del velivolo, mentre la coppia indotta dal sistema propulsivo sul corpo del velivolo è stata trascurata.

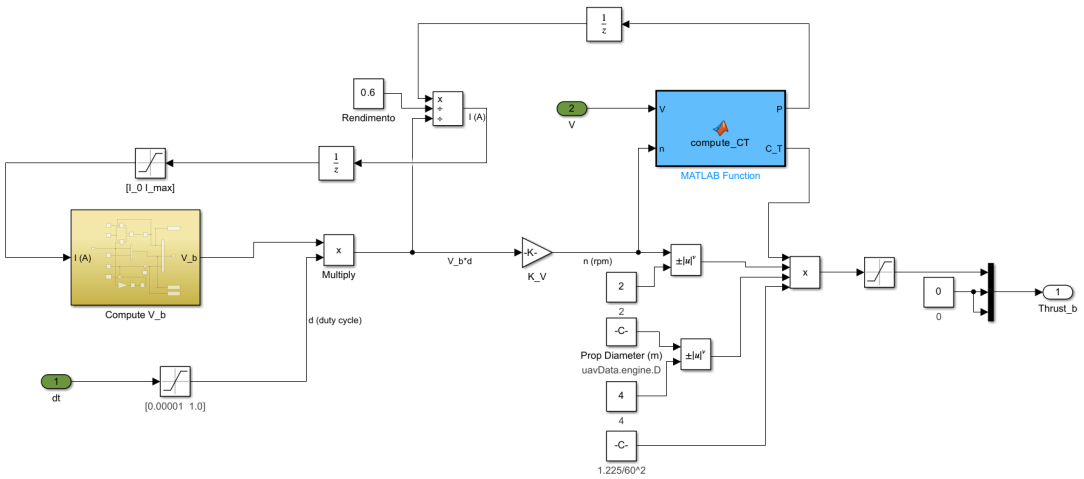


Figura 3.3: Modello propulsivo sviluppato in Simulink

Il modello sviluppato consente di determinare la spinta T generata dal motore attraverso la seguente formula empirica:

$$T = C_T \cdot \rho \cdot n^2 \cdot D^4 \tag{3.1}$$

dove:

- C_T: coefficiente adimensionale di spinta, funzione delle caratteristiche dell'elica e delle condizioni di volo;
- ρ: densità dell'aria, assunta costante e pari a $\rho = 1.225\ kg/m^3$;
- n: velocità di rotazione dell'elica, espressa in rps;
- D: diametro dell'elica, pari a $D = 13\,\text{in} = 0.3302\,\text{m}$.

Essendo ρ e D costanti, l'obiettivo è calcolare, a ogni istante temporale, il numero di giri del motore (uguale a quello dell'elica) e il corrispondente coefficiente di spinta C_T.

3.2.1 Calcolo del numero di giri

Il numero di giri è ottenuto dalla relazione:

$$n\,(\text{rpm}) = d \cdot V_b \cdot K_V \tag{3.2}$$

dove d rappresenta il comando di manetta (compreso tra 0 e 1) ottenuto dai valori di PWM forniti dall'autopilota, mentre V_b è la tensione ai morsetti della batteria nell'istante considerato. Il prodotto $d \cdot V_b$ rappresenta quindi la tensione effettivamente richiesta dal motore alla batteria tramite l'ESC.
Il parametro K_V è una costante caratteristica del motore che consente di determinare il numero di giri in funzione del comando di throttle.
Il valore di V_b viene aggiornato a ogni iterazione (figura 3.4). La tensione ai morsetti della batteria non coincide infatti con la tensione a vuoto V_{oc}, ma risulta ridotta per diverse cause:

- **Caduta di tensione istantanea per resistenza interna**: quando la batteria alimenta un carico, la corrente I che la attraversa genera una caduta di tensione dovuta alla resistenza interna.

$$V_{ist} = I \cdot R_{battery} \tag{3.3}$$

- **Caduta di tensione per polarizzazione**: durante i transitori, la tensione ai morsetti presenta un comportamento dinamico modellabile tramite il circuito equivalente di Thévenin di primo ordine.

$$\dot{V}_{pol} = -\frac{1}{C_{pol} R_{pol}} \cdot V_{pol} + \frac{I(t)}{C_{pol}} \tag{3.4}$$

- **Scarica della batteria**: con il tempo, la tensione a vuoto V_{oc} diminuisce in funzione dello stato di carica (SOC), compreso tra 0 e 1, descritto dalla relazione:

$$\dot{SOC} = -\frac{I(t)}{Q_{nom}} \tag{3.5}$$

La tensione effettiva ai morsetti risulta quindi[1]:

$$V_b = V_{oc}(SOC) - V_{ist} - V_{pol} \tag{3.6}$$

[1]Durante le simulazioni HITL si è osservato che l'integrazione dei termini $\dot{V}_{pol}$ e $\dot{SOC}$ rallentava il calcolo, compromettendo la convergenza dei risultati. Per questo motivo, tali contributi sono stati esclusi, considerando solo il calo di tensione V_{ist}.

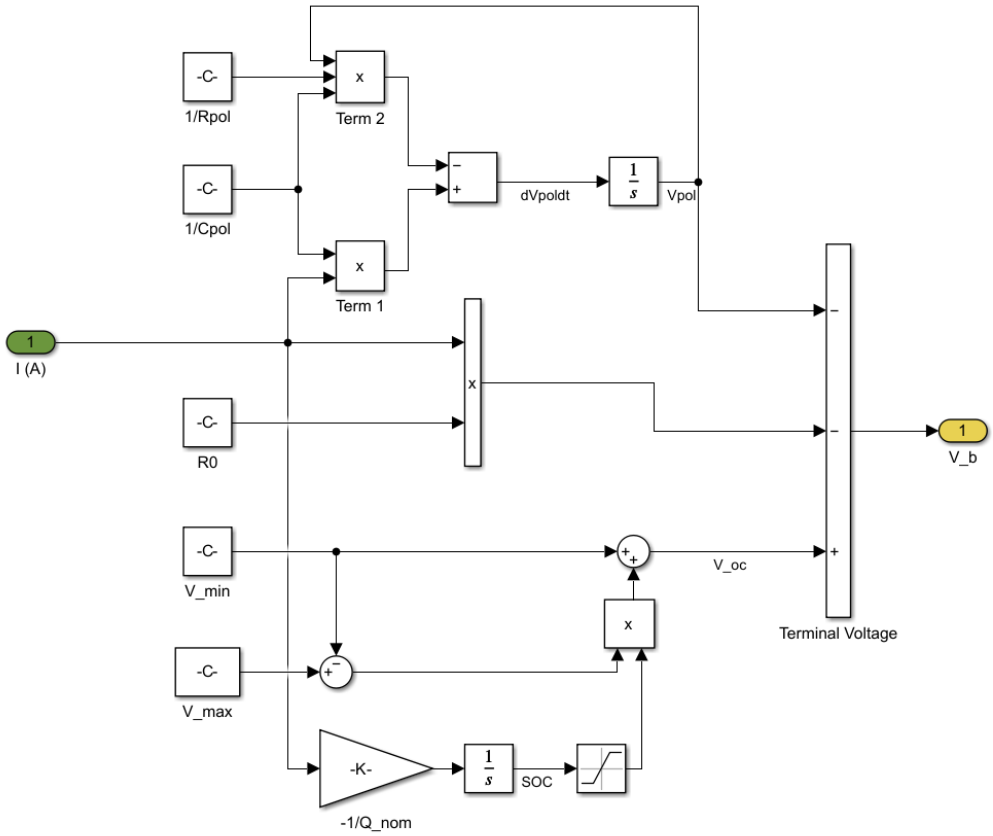


Figura 3.4: Modello Simulink per il calcolo di V_b

Per il calcolo dei contributi è necessario ricavare la corrente I invertendo la relazione:

$$P_{motor} = V_b \cdot I \tag{3.7}$$

dove P_{motor} rappresenta la potenza richiesta dal motore, pari alla potenza assorbita dall'elica divisa per il rendimento del sistema:

$$P_{motor} = \frac{P_{elica}}{\eta} \tag{3.8}$$

La potenza richiesta dall'elica è calcolata nella sezione successiva.

Una volta determinato V_b, il numero di giri n si ottiene direttamente dall equazione 3.2 e viene poi convertito in *rps*:

$$n\,(\text{rps}) = \frac{n\,(\text{rpm})}{60} \tag{3.9}$$

3.2.2 Calcolo del coefficiente di spinta C_T

Per la stima del coefficiente di spinta C_T si sono utilizzati i *Performance Data* messi a disposizione dal produttore dell'elica, APC Propellers. [16]
L'elica impiegata sul `FunCub NG` è una `13x4E`. Per questo modello, l'azienda fornisce un file di dati, `"PER3_13X4E.dat"`, che riporta le principali caratteristiche di performance al variare del numero di giri n (in *rpm*) e della velocità di avanzamento V. Tra questi dati sono di particolare interesse il coefficiente di spinta C_T (utilizzato nell'equazione 3.1) e la potenza assorbita P (usata nell'equazione 3.7).
Poiché i dati sono forniti in forma discretizzata, è stata sviluppata una funzione MATLAB per interpolare i valori tabellati, consentendo così di ottenere C_T e P per ogni combinazione di n e V.

Tale funzione, riportata integralmente nel listato 3.1, opera secondo i seguenti passaggi principali:

1. Il file `"PER3_13X4E.dat"` viene preprocessato e da esso vengono generate le tabelle `C_T_tab` e `P_tab`, inserite nella struttura `uavData`. Ciascuna tabella (`19×29`) riporta i valori tabellati di C_T e P per i valori tabellati di numero di giri (righe) e velocità di volo (colonne).

2. Il numero di giri calcolato tramite l'equazione 3.2 viene confrontato con i valori tabulati nel file, individuando le due righe corrispondenti ai numeri di giri tabulati immediatamente inferiori e superiori.

3. Per ciascuna delle due righe di `C_T_tab` si costruisce una funzione interpolante rispetto alla velocità V, così da ottenere C_T al variare di V per quel numero di giri. Conoscendo quindi la velocità V, si otterranno due valori di C_T ciascuno riferito ad un diverso numero di giri. La stessa procedura viene applicata a *P_tab*.

4. Infine, in funzione del numero di giri effettivo n, si esegue un'interpolazione lineare tra i due valori di C_T e di P ottenuti, restituendo così i corrispondenti valori per i parametri considerati.

```
% Calcolo C_T e P per ogni valore di n e V
function [C_T, P] = compute_CT(n, V, uavData)

C_T = 0;    P=0;    lim_min=0;  lim_max=0;

if n<uavData.engine.n_tab(1)
    C_T=0.000; % Imposto spinta nulla se n<1000rpm
else
    for i=2:length(uavData.engine.n_tab)
        if n<uavData.engine.n_tab(i)
            lim_max=i;   lim_min=i-1;
            break
        end
    end
    % Creo 2 funzioni che interpolano i valori di velocità
    f_min_CT = @(x) interp1(uavData.engine.V_tab(lim_min,:),
    uavData.engine.C_T_tab(lim_min,:), x, 'spline');
    f_max_CT = @(x) interp1(uavData.engine.V_tab(lim_max,:),
    uavData.engine.C_T_tab(lim_max,:), x, 'spline');

    f_min_PWR = @(x) interp1(uavData.engine.V_tab(lim_min,:)
   , uavData.engine.P_tab(lim_min,:), x, 'spline');
    f_max_PWR = @(x) interp1(uavData.engine.V_tab(lim_max,:)
   , uavData.engine.P_tab(lim_max,:), x, 'spline');

    if V<uavData.engine.V_tab(lim_min,end)
        C_T= (f_max_CT(V) - f_min_CT(V))/1000 * (n - uavData
   .engine.n_tab(lim_min))  + f_min_CT(V);
        P= (f_max_PWR(V) - f_min_PWR(V))/1000 * (n - uavData
   .engine.n_tab(lim_min))  + f_min_PWR(V);
    elseif V<uavData.engine.V_tab(lim_max,end)
        C_T= ( (f_max(V) - f_min(uavData.engine.V_tab(
   lim_min,end)))/1000 * (n - uavData.engine.n_tab(lim_min))
     + f_min(uavData.engine.V_tab(lim_min,end)) );
        P= (f_max_PWR(uavData.engine.V_tab(lim_max,end))-
   f_min_PWR(uavData.engine.V_tab(lim_min,end)))/1000 * (n -
    uavData.engine.n_tab(lim_min))  + f_min_PWR(uavData.
   engine.V_tab(lim_min,end));
    else
        C_T = 0.0004 * (n/1000)^1.5;
    end
end
```

Listing 3.1: MATLAB Function per il calcolo di C_T

3.3 Modello Geometrico e Inerziale

Le principali caratteristiche del velivolo sono state ricavate tramite il prototipo reale del drone in congiunzione con il manuale fornito da MULTIPLEX ® [14] nel quale è presente la rappresentazione in scala del `FunCub NG`.

3.3.1 Geometria

Le principali lunghezze sono riportate nella vista in pianta e vista laterale del velivolo in figura 3.5 e figura 3.6. [2]

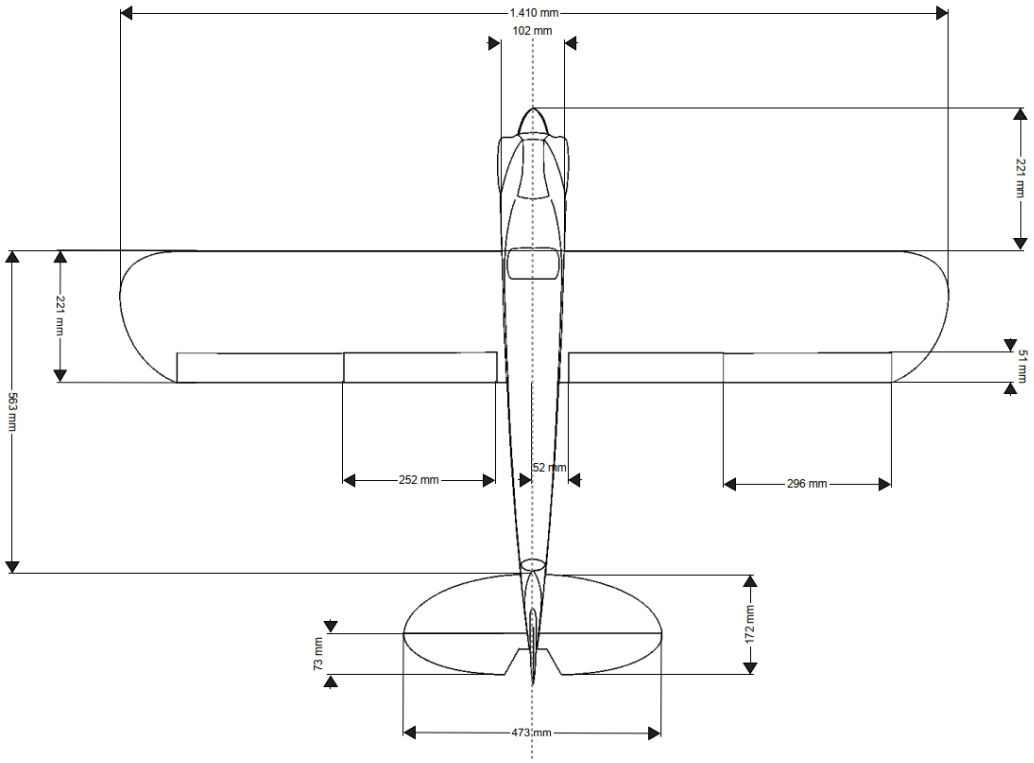


Figura 3.5: Vista dall'alto del FunCub NG

Utilizzando ulteriori misurazioni acquisite direttamente sul velivolo (non riportate nella figura precedente), è stato possibile ricostruire un modello tridimensionale accurato all'interno di OPENVSP, un software open-source sviluppato dalla NASA, che permette di sviluppare modelli geometrici parametrizzati di velivoli [17]. (figura 3.7).

[2]Le immagini non sono in scala, ma solo a scopo rappresentativo

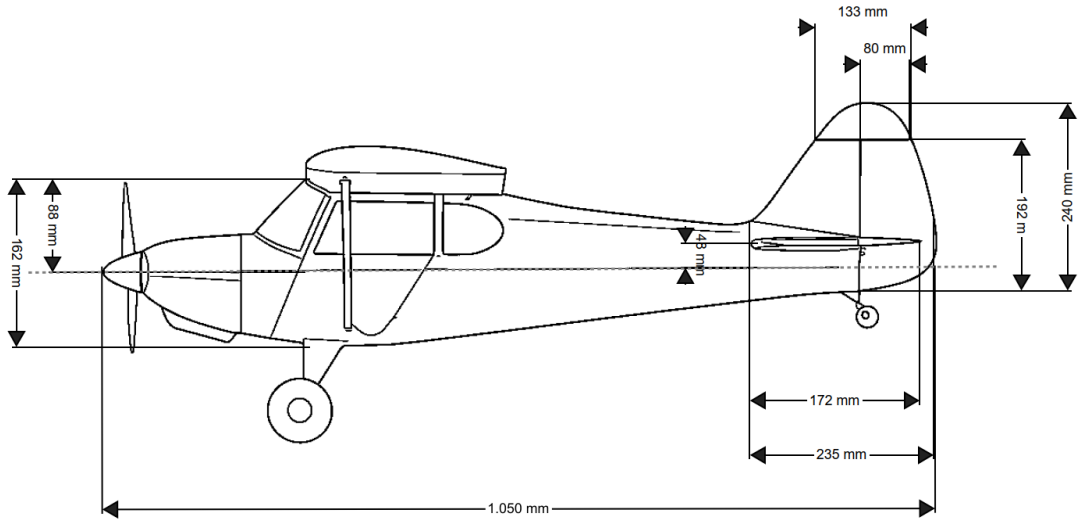


Figura 3.6: Vista laterale del FunCub NG

La geometria è stata rappresentata mediante quattro superfici aerodinamiche principali, alle quali è stata aggiunta una componente dedicata alla modellazione dell'elica. Gli altri elementi strutturali non aerodinamici (motore, batteria, avionica, carrello, servi, ecc.) sono stati invece modellati come masse concentrate.
Di seguito si riportano nel dettaglio le superfici che costituiscono il modello.

- **Ala**: L'ala è stata modellata come una superficie rettangolare con profilo aerodinamico `Eppler E205`. Ciascuna semiala è stata suddivisa in cinque sezioni per riprodurre con maggiore fedeltà la leggera rastremazione verso le estremità, migliorando così la qualità della mesh generata da VSPAero. È stato imposto un angolo di diedro pari a 3°. Gli alettoni, modellati come superfici di controllo dedicate, sono posizionati alle estremità alari e presentano corda costante lungo tutta la loro estensione. La presenza dei flap, sebbene presente sul velivolo reale, non è stata considerata poiché non rilevante per le manovre analizzate.

- **Stabilizzatore Orizzontale (HTP)**: Lo stabilizzatore orizzontale è stato rappresentato come una superficie di forma ellittica equipaggiata con un profilo simmetrico `NACA 0010`. Come per l'ala, anche l'HTP è stato suddiviso in più sezioni con l'obiettivo di approssimare con maggiore precisione la geometria reale e ottenere una distribuzione più regolare dei pannelli aerodinamici. Il piano orizzontale è posizionato circa 40 mm al di sotto del piano alare, in accordo con la configurazione costruttiva del FunCub. L'elevatore si estende lungo l'intero bordo di fuga ed è modellato come superficie mobile con hinge line coincidente con il margine d'uscita della parte fissa.

- **Stabilizzatore Verticale (VTP)**: Lo stabilizzatore verticale è stato modellato come una superficie con profilo `NACA 0010`, coerente con quello impiegato sul piano orizzontale. Esso si sviluppa sia al di sopra sia al di sotto del piano HTP, riproducendo la caratteristica configurazione cruciforme del velivolo. Il timone si estende lungo tutto il bordo d'uscita del VTP ed è suddiviso in due regioni: la porzione superiore, a corda piena, e la porzione inferiore, che presenta una corda pari alla metà di quella della parte fissa. Tale suddivisione permette di rappresentare correttamente la morfologia e la cinematica reale della superficie.

- **Fusoliera**: La fusoliera è stata suddivisa in dieci sezioni, definite tramite curve di sezione rettangolari e arrotondate per seguire l'evoluzione reale della geometria. La sua massima espansione è collocata in corrispondenza del bordo d'attacco dell'ala, mentre la sezione si riduce progressivamente verso la coda fino alla giunzione con HTP e VTP. Questa discretizzazione permette di ottenere un controllo più accurato della mesh, limitando variazioni brusche tra le superfici contigue.

- **Elica**: L'elica è stata modellata come una superficie semplificata composta da due pale, utilizzando le dimensioni riportate nella sezione precedente. Essa è stata inclusa principalmente per coerenza geometrica e come riferimento spaziale, mentre la sua influenza aerodinamica non viene calcolata tramite OpenVSP ma attraverso un modello propulsivo dedicato.

All'interno della modellazione su OpenVSP viene utilizzato un sistema di riferimento solidale al velivolo differente rispetto a quello descritto in Appendice A. Il sistema di coordinate ha infatti origine nel piano di simmetria nel bordo di attacco dell'ala, con l'asse $X_{OpenVSP}$ che punta verso la coda dell'UAV e l'asse $Z_{OpenVSP}$ che punta verso l'alto, mentre l'asse $Y_{OpenVSP}$ completa la terna levogira. Il velivolo modellato e il sistema di riferimento adottato sono rappresentati in Figura 3.7.

3.3.2 Baricentro e Matrice d'inerzia

Le misurazioni dei pesi dei singoli componenti sono state effettuate in laboratorio utilizzando una bilancia di precisione.
Poiché non è stato possibile smontare completamente il velivolo, alcune pesature sono state eseguite su gruppi di componenti; i pesi dei singoli elementi sono stati poi ricavati sottraendo i valori standard riportati nelle specifiche.

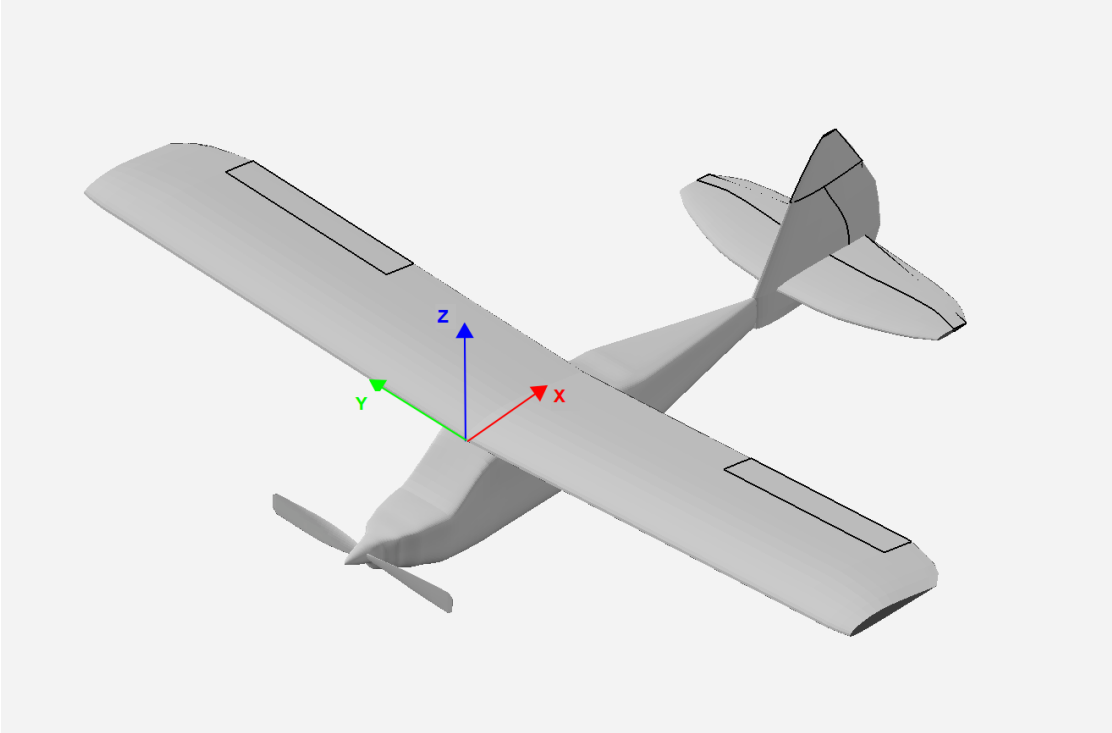
Sono state effettuate le seguenti misure:

- Peso del velivolo privo di ala (e relativi servo), elica e batteria: 842.1 g, comprendente quindi fusoliera, impennaggi (con servo), CubePilot, ESC, ricevente, motore, ruote e set di cavi.
- Ala comprensiva dei servoattuatori: 373.2 g
- Ruota anteriore: 18.5 g
- Elica: 31.2 g

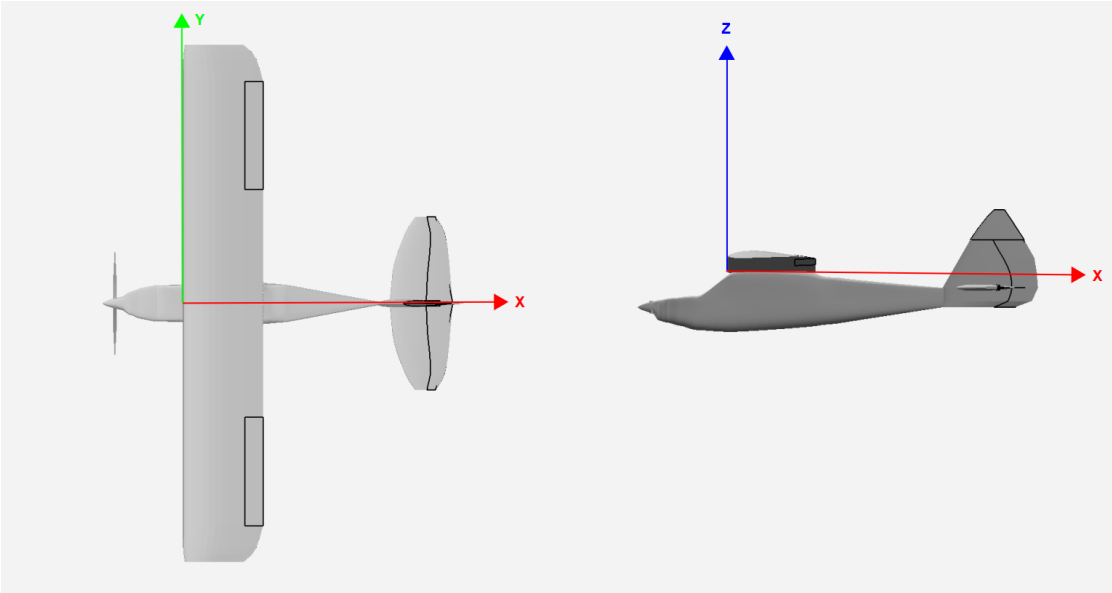
Sottraendo da queste misure i pesi dei componenti standardizzati riportati nel manuale (tabella 3.2), è stato possibile ricavare i pesi dei principali insiemi strutturali: l'ala, pari a 341.2 g, e il gruppo fusoliera+impennaggi, pari a 463.7 g.

Analisi delle proprietà di massa con OpenVSP

Le proprietà di massa del `FunCub` sono state ottenute utilizzando l'analisi `"Mass Properties"` disponibile nella sezione `"Analysis"` di OpenVSP. Il calcolo sfrutta il modello geometrico descritto in sezione 3.3.1, all'interno del quale ogni componente include una sotto-sezione dedicata all'assegnazione delle proprietà di massa.



(a) Left ISO



(b) Top and Left Side

Figura 3.7: FunCub NG su OpenVSP

Per ricavare una matrice d'inerzia il più accurata possibile è necessario modellare le superfici aerodinamiche come masse distribuite. A tale scopo OpenVSP richiede l'inserimento di una densità (superficiale o volumetrica) per ciascun componente. Poiché tali densità e le geometrie delle superfici non sono note con sufficiente precisione, si è optato per un valore di densità del materiale che, moltiplicato per il volume della geometria modellata, restituisse esattamente il peso reale del componente.
Segue una descrizione dettagliata del processo di modellazione per ciascuna superficie aerodinamica:

1. **Ala**: Considerando il peso misurato del rivestimento alare privo di servi (341.2 g) e assumendo una densità volumetrica uniforme, si ottiene una densità pari a 77.9 kg/m^3 per riprodurre correttamente la massa reale della superficie alare.

2. **Impennaggi (HTP e VTP)**: Poiché gli impennaggi sono realizzati con lo stesso materiale dell'ala, è stata utilizzata la stessa densità volumetrica ricavata al punto precedente. In questo modo si ottengono automaticamente le masse dei singoli componenti, pari a 49 g per l'HTP e 38 g per il VTP, valori non disponibili da manuale.

3. **Fusoliera**: Dalla massa misurata complessiva della fusoliera comprensiva degli impennaggi, sottraendo le masse di HTP e VTP, si ottiene una massa della sola fusoliera pari a 376.7 g. Poiché la fusoliera è cava e serve principalmente da struttura contenitiva, si è deciso di modellarla come massa distribuita lungo la superficie esterna. L'assegnazione di una densità superficiale pari a 1.52 kg/m^2 consente di riprodurre correttamente la massa reale.

4. **Elica**: Applicando lo stesso procedimento iterativo utilizzato per ala e impennaggi, si è assegnata all'elica una densità volumetrica pari a 1300 kg/m^3, ottenendo una massa di 31.2 g.

Alle masse distribuite così determinate sono state aggiunte le masse concentrate dei servoattuatori, del motore, dell'ESC, dei sistemi di bordo, della batteria e delle ruote. Inoltre è stato inserito per le prove di volo un tubo di pitot sulla semiala sinistra, il momento che tale sensore genera sull'ala del velivolo è stato quindi opportunamente bilanciato mediante un piccolo peso posto sulla semiala destra. Tali elementi sono stati inseriti come masse puntuali con posizione e valore riportati in tabella 3.2.
Una volta inserite tutte le masse distribuite e concentrate, è stata eseguita l'analisi delle proprietà di massa. Il calcolo è stato effettuato utilizzando uno slicing di 200 elementi lungo la direzione `X`. I risultati ottenuti sono riportati in tabella 3.3.

Elemento	**Massa (g)**	**X (mm)**	**Y (mm)**	**Z (mm)**
Cube Orange+ [18]	73	0	0	−80
ESC [19]	64	−50	0	−80
Motore [20]	132	−176	0	−88
Batteria [21]	414.7	30	0	−70
Ricevente [22]	10	−100	0	−100
GNSS	33	0	0	−150
Set di cavi [23]	40	−50	0	−80
Ruota	18.5	0	±120	−230
Servo HS-55+ (ala) [24]	8 cad.	100	±160 / ± 450	0
Servo HS-65HB (impennaggi) [25]	11.2 cad.	568	±10	−60
Pitot	30	70	−330	−0
Peso di bilancio	15	70	660	0

Tabella 3.2: Masse concentrate

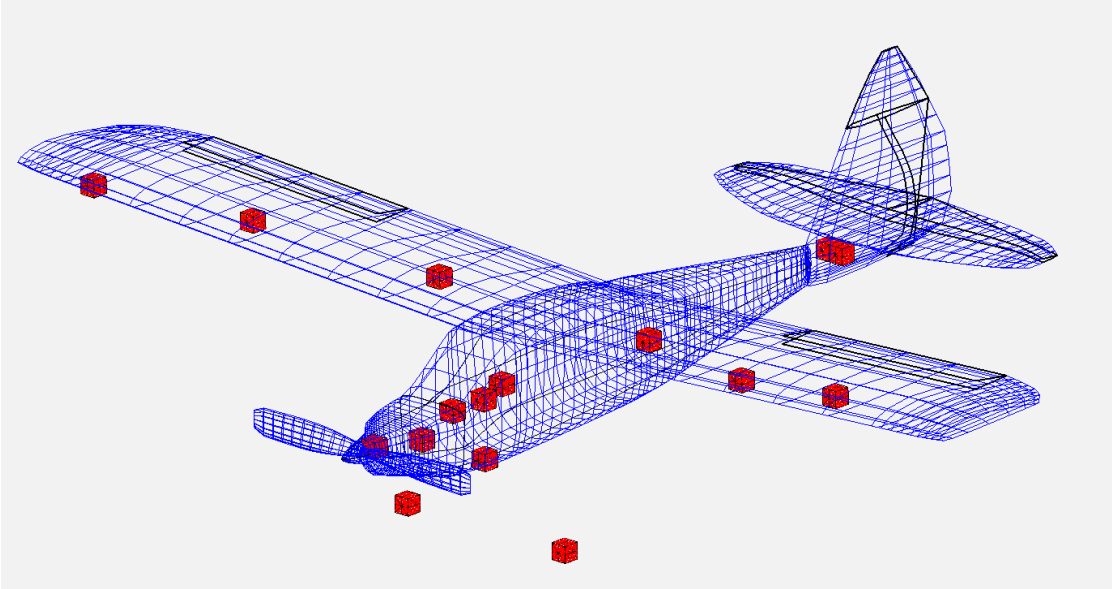


Figura 3.8: Rappresentazione delle masse concentrate su OpenVSP

Massa totale	1.739 kg
X_{CG}	82 mm
Y_{CG}	0 mm
Z_{CG}	-53 mm
I_{xx}	0.074 $kg \cdot m^2$
I_{yy}	0.069 $kg \cdot m^2$
I_{zz}	0.133 $kg \cdot m^2$

Tabella 3.3: Proprietà di massa ottenute da OpenVSP

Confronto con i dati del manuale

Si vuole verificare la coerenza tra le proprietà di massa ricavate sperimentalmente e quelle indicate nel manuale di costruzione del FunCub. La massa totale del velivolo in configurazione di volo è pari a 1.7062 kg; tuttavia, nel presente modello è stata utilizzata una batteria `LiPo 4S` sensibilmente più pesante rispetto alla batteria `ROXXY EVO LiPo 3-2600M 40C` indicata nel manuale, che ha una massa di 219 g. Inoltre, il manuale non considera la presenza della `CubeOrange+`, del GNSS, del tubo di Pitot e del relativo peso di bilancio.
Tenendo conto di tali differenze, si può stimare la massa che il velivolo avrebbe se equipaggiato secondo la configurazione del manuale:

$$\begin{aligned}\text{Massa ipotetica} &= \text{Massa misurata} - \text{Cube} - \text{GNSS} - \text{Pitot e bilancio} - \text{Batteria attuale} \\ &\quad + \text{Batteria manuale} \\ &= 1739\,g - 414.7\,g - 73\,g - 33\,g - 45\,g + 219\,g = 1392.3\,g\end{aligned} \tag{3.10}$$

La massa così stimata risulta in buon accordo con il valore riportato nel manuale, pari a 1380 g. Anche il baricentro ottenuto dall'analisi coincide con quello indicato dal costruttore (82 mm).

3.4 Modello Aerodinamico

Per l'identificazione dei coefficienti aerodinamici dell'UAV è stato utilizzato VSPAERO, il solver aerodinamico integrato in OpenVSP, che utilizza un modello aerodinamico a potenziale basato su una combinazione di Panel Method per corpi tozzi, e Vortix Lattice Method per superfici sottili.

Il software sfrutta il modello geometrico descritto in sezione 3.3.1 per eseguire analisi di portanza, resistenza e stabilità. Per ragioni di praticità e robustezza numerica si è scelto di includere esclusivamente le superfici aerodinamiche principali, escludendo il carrello, i servoattuatori e l'effetto dell'elica, il cui contributo avrebbe complicato la mesh senza apportare benefici significativi ai fini dell'identificazione dei coefficienti aerodinamici a corpo fisso.

Un'ulteriore semplificazione riguarda il timone dell'impennaggio verticale: nella configurazione reale esso è composto da due superfici separate (come descritto in sezione 3.3.1), tuttavia OpenVSP non consente di rappresentare tale complessità mediante un'unica superficie mobile, e VSPAERO non permette la gestione simultanea di due superfici di timone distinte. Si è quindi optato per una modellazione equivalente mediante una singola superficie a corda costante, avente area complessiva pari all'area totale del timone reale.

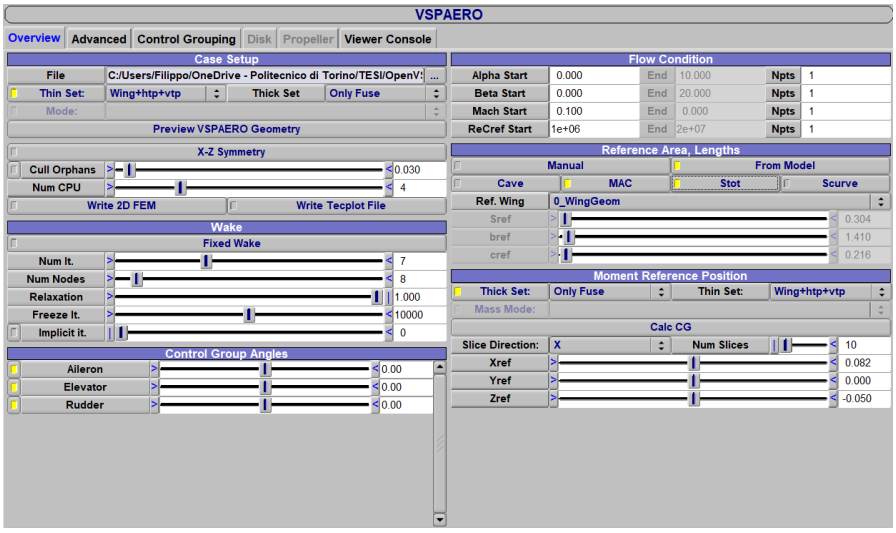


Figura 3.9: Schermata principale di VSPAERO

Per le analisi è stato adottato un modello *"thin"* per ala e impennaggi, che li idealizza come superfici portanti prive di spessore, in grado di catturare efficacemente la distribuzione della portanza e i principali effetti di stabilità. La fusoliera è invece stata modellata con un approccio *"thick"*, che considera il suo volume e lo spessore

effettivo, permettendo di includere gli effetti di interferenza e della pressione associati a un corpo tozzo.
Le simulazioni sono state eseguite in modalità *free wake*, nella quale la scia è lasciata evolvere liberamente e viene aggiornata dal solver attraverso sette iterazioni. Questo approccio consente di rappresentare in modo più realistico l'induzione vorticosa e le interazioni tra le superfici.
I valori di riferimento delle superfici e le condizioni di flusso adottate sono riportati in Tabella 3.4.

c_{ref}	$0.216\ m$
b_{ref}	$1.410\ m$
S_{ref}	$0.304\ m^2$
X_{CG}	$0.082\ m$
Y_{CG}	$0\ mm$
Z_{CG}	$-0.053\ m$
V_{ref}	$15\ m/s$
ρ_{ref}	$1.225\ kg/m^3$
Mach	0.1
Re	10^6

Tabella 3.4: Valori di riferimento per analisi aerodinamica

3.4.1 Analisi Aerodinamica Longitudinale

Un'analisi aerodinamica preliminare è stata eseguita per valutare il comportamento globale dell'UAV al variare dell'angolo d'incidenza α. L'obiettivo di questa fase iniziale è verificare che il modello sviluppato sia in grado di riprodurre correttamente le principali caratteristiche statiche del velivolo reale.
L'analisi è stata quindi effettuata variando l'angolo di incidenza nel range tra $-5°$ e $10°$ con uno step di $1°$, valutando la variazione dei principali coefficienti aerodinamici della dinamica longitudinale. Durante questa fase tutte le superfici mobili vengono mantenute in posizione neutra a $0°$ di deflessione.

In figura 3.10 è rappresentata la variazione della distribuzione di pressione sul velivolo al variare dell'incidenza, in particolare è possibile notare che sia sull'ala che sull'impennaggio verticale vengono raggiunti valori di c_p più negativi all'aumentare dell'incidenza, con conseguente aumento della portanza generata.
In figura 3.11 è mostrata la relazione tra i coefficienti aerodinamici di portanza e di beccheggio e l'angolo di incidenza α. In figura 3.12 sono invece mostrati i

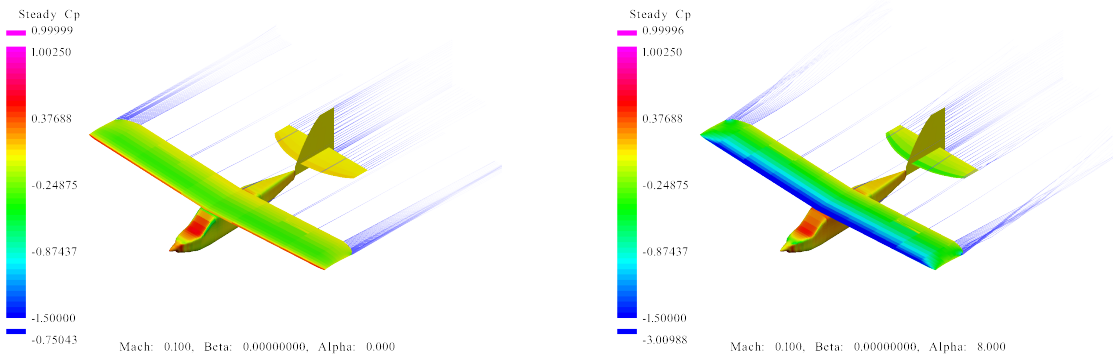


Figura 3.10: Distribuzione pressione al variare di α

contributi al coefficiente di resistenza totale del velivolo, che aumenta all'aumentare dell'incidenza, rendendo quindi il volo inefficiente ad alte incidenze.

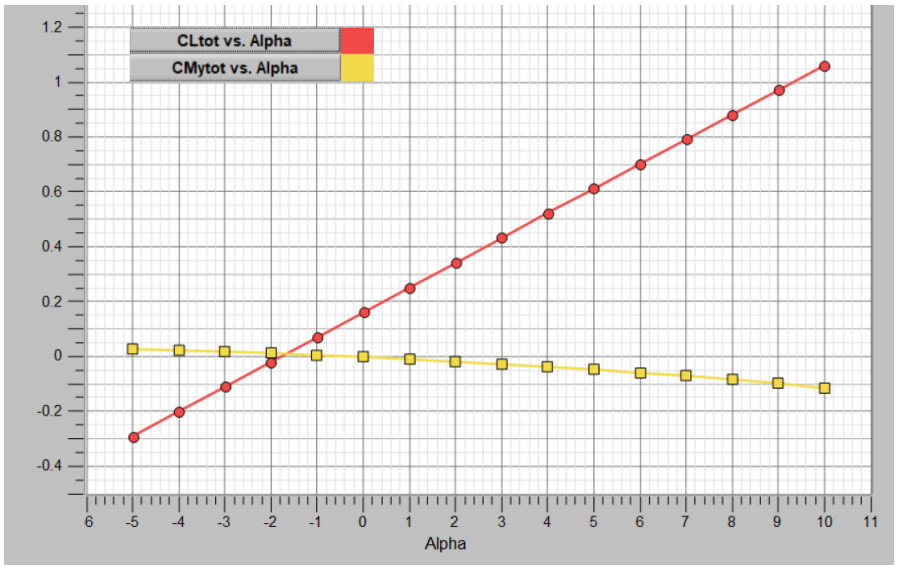


Figura 3.11: Curve $C_L - \alpha$ e $C_m - \alpha$

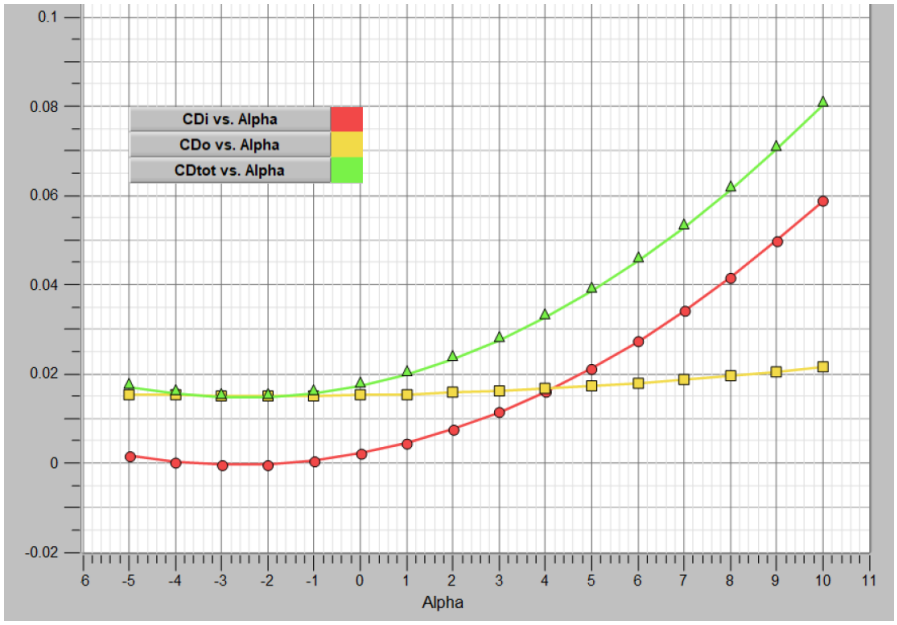


Figura 3.12: Curva $C_D - \alpha$

3.4.2 Analisi Aerodinamica Latero-Direzionale

Un analisi aerodinamica simile alla precedente è stata condotta per valutare le caratteristiche latero-direzionale del velivolo. Questa analisi è stata effettuata variando l'angolo di sideslip β nel range tra 0° e 10° con uno step di 2°, mantenendo le superfici di controllo in posizione neutra.

In figura 3.13 è rappresentata la variazione della distribuzione di pressione sul velivolo al variare dell'angolo di sideslip, in particolare è possibile notare che all'aumentare di β si forma una scia irregolare fino a separazione e elimina la simmetria del problema.

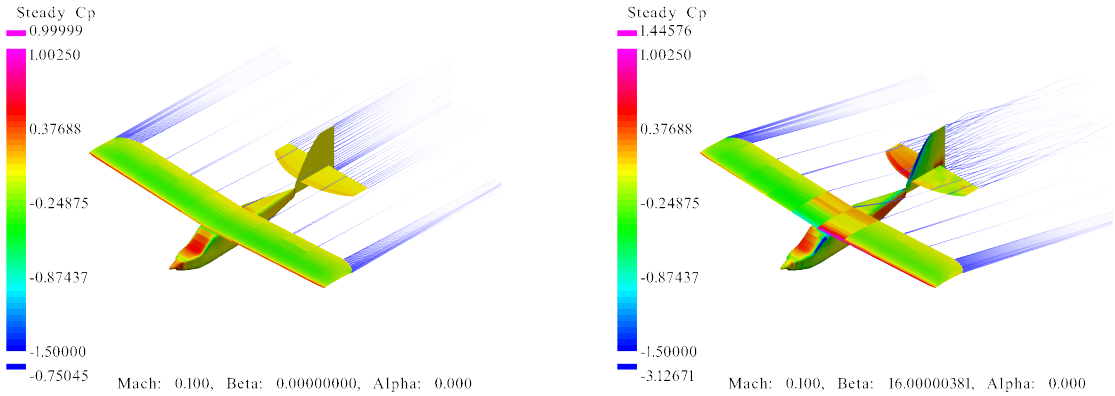


Figura 3.13: Distribuzione pressione al variare di β

In figura 3.14 è mostrata la relazione tra i principali coefficienti aerodinamici della dinamica latero-direzionale e l'angolo di sideslip β. In figura 3.15 è invece mostrato il rapporto tra portanza e resistenza al variare di β, mostrando come un volo simmetrico è notevolmente conveniente in termini di efficienza.

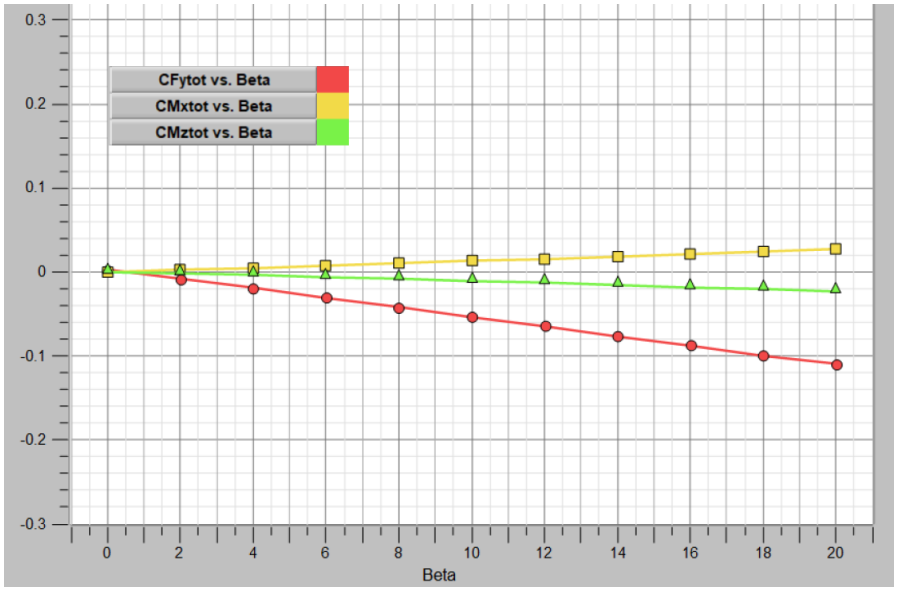


Figura 3.14: Curve $C_Y - \beta$, $C_l - \beta$ e $C_n - \beta$

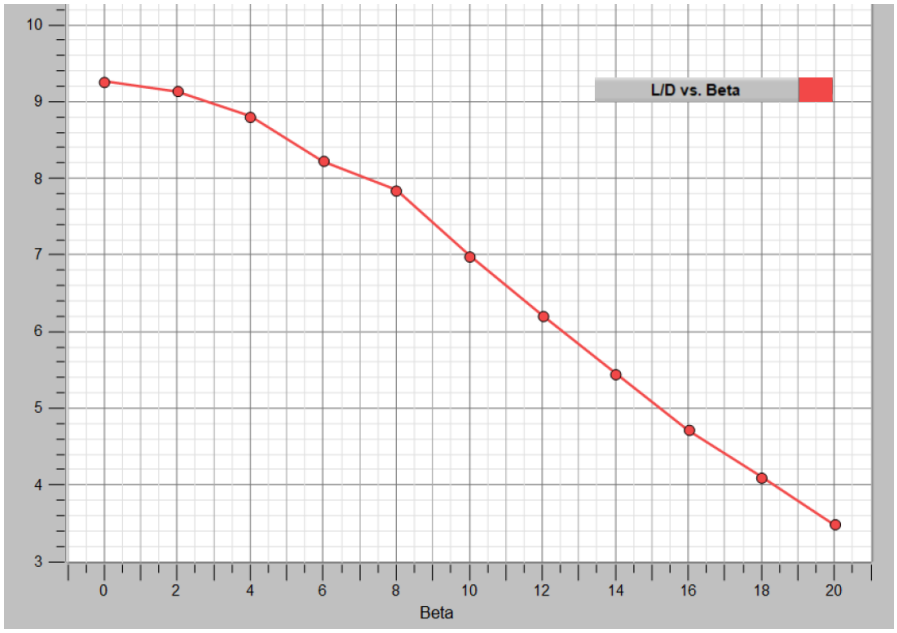


Figura 3.15: Curva $L/D - \beta$

3.4.3 Analisi di Stabilità

L'analisi di stabilità rappresenta lo strumento attraverso il quale è possibile ricavare i principali coefficienti aerodinamici e le relative derivate del modello. Per il loro calcolo, VSPAERO introduce piccole perturbazioni attorno alla condizione di equilibrio impostata (nel nostro caso $\alpha = 0°$, $\beta = 0°$, $V_{inf} = 15\ m/s$) e valuta la variazione indotta sui coefficienti aerodinamici globali del velivolo. Le derivate vengono quindi ottenute tramite una differenziazione numerica, applicando differenze finite tra la soluzione nominale e le soluzioni perturbate.
Al termine dell'analisi, il software produce un file `".stab"` contenente tutti i coefficienti e le derivate di interesse, riportati in tabella 3.5.

Coef	Base	α	β	p	q	r	δ_a	δ_e	δ_r
C_L	0.1615	5.2300	0.0421	0.0124	7.8985	0.0120	0.0550	0.5562	0.0060
C_Y	0.0003	0.0572	-0.2298	-0.1003	0.1421	0.2820	-0.02060	0.0059	-0.2253
C_l	0.0002	-0.0971	-0.1662	-0.5532	-0.2261	0.0216	-0.2688	-0.0090	-0.0134
C_m	-0.0005	-0.2480	0.1923	0.0702	-9.8399	0.0753	0.0196	-1.2962	0.0159
C_n	-0.0001	-0.0560	0.0008	-0.0367	-0.1368	-0.1126	-0.0017	-0.0056	0.0889

Tabella 3.5: Coefficienti aerodinamici calcolati con VSPAERO.

Il coefficiente di base (talvolta indicato con pedice "0") rappresenta il valore del coefficiente aerodinamico in condizioni non perturbate. Le restanti grandezze corrispondono invece alle derivate rispetto alla variabile indicata in colonna e sono tutte grandezze adimensionali.
Si osserva inoltre che il sistema di riferimento utilizzato da OpenVSP differisce da quello impiegato nel modello MATLAB/Simulink: per questo motivo la derivata rispetto alla deflessione del timone di direzione δ_r deve essere invertita di segno rispetto al valore presente nel file `".stab"`.
L'analisi di stabilità tramite VSPAERO fornisce dunque un insieme completo di derivate, alcune delle quali non verranno utilizzate nel modello implementato in MATLAB/Simulink; allo stesso tempo, essa non permette di ricavare altri parametri necessari, che verranno invece determinati tramite stime empiriche.

Coefficiente di resistenza

Il coefficiente di resistenza del velivolo C_D viene modellato tramite la polare aerodinamica rappresentata in figura 3.16 e descritta dall'equazione:

$$C_D = C_{D_0} + A_1 \cdot C_L + A_{polar} \cdot C_L^2 \tag{3.11}$$

Dall'analisi di stabilità è stato ottenuto il coefficiente di resistenza a portanza nulla

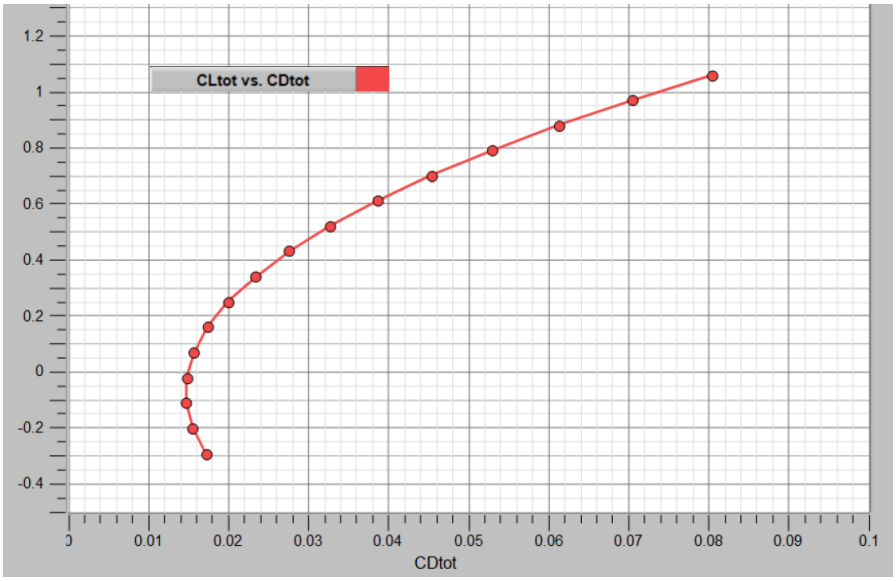


Figura 3.16: Polare Aerodinamica

C_{D_0}, mentre i termini A_1 ed A_{polar} devono essere modellati separatamente. Per il termine quadratico si utilizza la nota relazione empirica basata sull'allungamento alare $\lambda = b^2/S$ e sul fattore di Oswald, assunto pari a $e = 0.9$:

$$A_{polar} = \frac{1}{\lambda \pi e} = 0.0540 \tag{3.12}$$

Il termine lineare, di minore influenza, viene assunto pari a: $A_1 = 0.002$.

Derivate di $\dot{\alpha}$

Le derivate rispetto ad $\dot{\alpha}$ rappresentano la variazione dei coefficienti aerodinamici dovuti all'effetto dell'isteresi, ovvero il ritardo nell'instaurarsi della circuitazione (e quindi della variazione della portanza) nel caso in cui α vari rapidamente ($\dot{\alpha}$ elevato).
Questo fenomeno interessa principalmente il piano di coda, in quanto maggiormente distante dal baricentro. Le derivate più significative sono $C_{L_{\dot{\alpha}}}$ e $C_{m_{\dot{\alpha}}}$, stimabili attraverso le seguenti relazioni empiriche [26]:

$$C_{L_{\dot{\alpha}}} = 2 \cdot \frac{S_t}{S} \cdot C_{L_\alpha} \frac{d\epsilon}{d\alpha} \frac{l_t}{c} = 0.4064 \tag{3.13}$$

$$C_{m_{\dot{\alpha}}} = -C_{L_{\dot{\alpha}}} \cdot \frac{l_t}{c} = -1.2718 \tag{3.14}$$

3.5 Riepilogo parametri

Vengono riportati in tabella 3.6 tutti i parametri utilizzati nel modello MATLAB/Simulink durante le simulazioni, molti dei quali sono stati ricavati nelle sezioni precedenti.
Alcuni coefficienti aerodinamici sono stati modificati rispetto ai valori ricavati da VSPAERO in modo da ottenere una migliore modellazione della dinamica del velivolo. Tali variazioni sono state effettuate in seguito al confronto tra i dati reali misurati dalle prove di volo e i dati ottenuti dalle simulazioni, effettuato nel Capitolo 4 per la validazione del modello.

Tabella 3.6: Parametri utilizzati nel modello MATLAB/Simulink

Parametro	**Simbolo**	**Valore**	**Unità**
Geometria e massa			
Span	b	1.41	m
Chord	c	0.216	m
Superficie alare	S	0.30363	m^2
Massa	m	1.739	kg
Momenti d'inerzia	I_{xx}	0.074	$\mathrm{kg\,m}^2$
	I_{yy}	0.069	$\mathrm{kg\,m}^2$
	I_{zz}	0.133	$\mathrm{kg\,m}^2$
Aerodinamica			
Coefficienti di portanza	C_{L0}	0.1615	–
	$C_{L\alpha}$	5.2300	rad^{-1}
	$C_{L\dot{\alpha}}$	0.4064	rad^{-1}
	C_{Lq}	7.8985	rad^{-1}
	$C_{L\delta_e}$	0.7787	rad^{-1}
Coefficienti di resistenza	C_{D0}	0.0134	–
	A_1	0.0020	–
	A_{polar}	0.0540	–
Coefficienti di forza laterale	$C_{Y\beta}$	-0.2298	rad^{-1}
	$C_{Y\delta_r}$	0.2253	rad^{-1}
	C_{Yp}	-0.1003	rad^{-1}
	C_{Yr}	0.2820	rad^{-1}
Coefficienti di rollio	$C_{l\beta}$	-0.0499	rad^{-1}
	C_{lp}	-0.6638	rad^{-1}
	C_{lr}	0.0216	rad^{-1}
	$C_{l\delta_a}$	-0.2150	rad^{-1}
	$C_{l\delta_r}$	0.0134	rad^{-1}
Coefficienti di beccheggio	C_{m0}	-0.0005	–

Parametro	**Simbolo**	**Valore**	**Unità**
	$C_{m\alpha}$	-0.3968	rad^{-1}
	$C_{m\dot{\alpha}}$	-1.2718	rad^{-1}
	C_{mq}	-13.284	rad^{-1}
	$C_{m\delta_e}$	-1.0370	rad^{-1}
Coefficienti di imbardata	$C_{n\beta}$	0.008	rad^{-1}
	C_{np}	-0.0367	rad^{-1}
	C_{nr}	-0.1464	rad^{-1}
	$C_{n\delta_a}$	-0.0017	rad^{-1}
	$C_{n\delta_r}$	-0.0445	rad^{-1}
Propulsione			
N°giri per Volt	K_V	930	rpm/V
Tensione batteria	V_{batt}	14.8	V
Resistenza polarizzazione	R_{pol}	0.03	Ω
Capacità polarizzazione	C_{pol}	80	F
Capacità nominale	Q_{nom}	4200	mAh
Resistenza interna	R_m	0.013	Ω
Corrente a vuoto	I_0	2	A
Corrente massima	$I_{\max}$	55	A
Diametro elica	D	0.3302	m
Rendimento trasmissione	η	0.6	–
Deflessioni massime superfici			
Aileron	$\delta_{a,\max}$	20 [3]	°
Elevator	$\delta_{e,\max}$	30	°
Rudder	$\delta_{r,\max}$	35	°

[3]La deflessione reale dell'alettone è asimmetrica, ovvero di 20° verso l'alto e 15° verso il basso, per ridurre l'effetto imbardata. Nel modello viene considerata una deflessione simmetrica di 20°

Capitolo 4

Validazione del modello dinamico

Nei capitoli precedenti è stata descritta l'architettura fisica necessaria alla simulazione Hardware-in-the-Loop (HIL), insieme ai principali componenti hardware e software che ne consentono l'esecuzione.
Questo capitolo rappresenta invece il collegamento operativo tra la modellazione descritta in precedenza e le attività di simulazione e validazione del sistema di controllo. Vengono infatti illustrati i passaggi fondamentali che sono stati seguiti per rendere possibile l'implementazione e l'esecuzione sia delle prove di volo reali sia della simulazione.

Il primo passo fondamentale riguarda il set-up e l'installazione sull'hardware del firmware PX4 fornito dal `UAV Toolbox Support Package for PX4 Autopilots`, il quale consente l'integrazione del flight controller in configurazione Hardware-in-the-Loop con Simulink. Il firmware personalizzato sviluppato da MathWorks mantiene il controllore nativo PX4, descritto in sezione 2.1.1, ma introduce la compatibilità con i blocchi Simulink di UAV Toolbox e fornisce supporto completo alla comunicazione MAVLink, in particolare per i messaggi `HIL_SENSOR`, `HIL_GPS` e `HIL_ACTUATOR_CONTROLS`.
Per eseguire il set-up è necessario collegare il flight controller al PC tramite connessione USB e seguire la procedura di *Hardware Setup* fornita dal `UAV Toolbox Support Package for PX4 Autopilots`.
Successivamente, mantenendo attiva la connessione con il flight controller, viene avviato il software QGroundControl, all'interno del quale viene selezionato il tipo di velivolo e l'airframe su cui verrà configurato il controllore. Questo passaggio consente di inizializzare una serie di parametri che caratterizzano la configurazione del velivolo selezionato e facilita le successive operazioni di configurazione di attuatori e motori. Nel caso in esame è stato selezionato l'airframe *"Generic Standard Plane"*, in quanto il sistema oggetto di studio è un UAV ad ala fissa dotato di quattro superfici di controllo e un motore.

Avendo impostato correttamente l'airframe, è possibile procedere con un'assegnazione degli attuatori di tipo airframe-based, nella quale QGroundControl si occupa di

associare automaticamente le output functions generate dal mixer ai rispettivi canali fisici PWM dell'hardware. In questo modo, i comandi relativi alle deflessioni delle superfici mobili (alettoni sinistro e destro, elevatore e timone), calcolati dal mixer dell'autopilota, vengono instradati direttamente verso i canali di uscita PWM utilizzati per il collegamento fisico con gli attuatori dell'UAV nel caso di volo reale. L'assegnazione automatica è illustrata in figura 4.1
Nel caso di simulazione Hardware-in-the-Loop non vengono generati segnali PWM reali, in quanto non sono presenti attuatori fisici, tuttavia la stessa assegnazione delle uscite viene comunque mantenuta a livello logico. Tale scelta garantisce la piena coerenza tra la simulazione HIL e il volo reale, mantenendo invariata la catena di controllo e i mixer nativi dell'autopilota PX4.

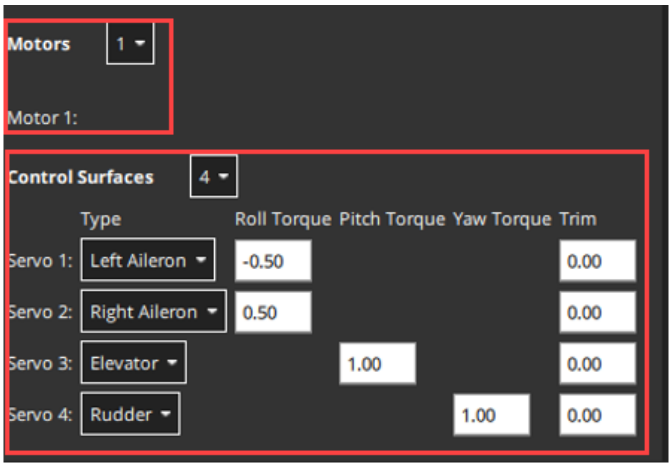


Figura 4.1: Finestra "Actuator Setup"

La finestra *Actuator Outputs* viene inoltre utilizzata per la configurazione dei parametri legati alle caratteristiche fisiche ed elettriche degli attuatori, quali i limiti minimo e massimo del segnale PWM, il valore di trim e l'eventuale inversione del verso di rotazione. La regolazione di tali parametri consente di adattare il comportamento degli attuatori alle specifiche dei servocomandi.

4.1 Prove di volo

Per la preparazione delle prove di volo sono state seguite le procedure riportate nella guida pratica di assemblaggio e configurazione dell'UAV fornite da PX4 [27].

Successivamente alle operazioni di caricamento del firmware e di configurazione degli attuatori, è stato effettuato l'assemblaggio del velivolo FunCub e dell'avionica di bordo necessaria alle prove sperimentali. In questa fase assume particolare importanza il corretto posizionamento del flight controller `Cube Orange+` all'interno dell'alloggiamento dedicato,

nonché il montaggio e la calibrazione dei sensori e del ricevitore radio. Infine, vengono descritte le procedure adottate per la gestione dei log di volo e le prove sperimentali effettuate.

FC Orientation and Calibration

Il flight controller è collegato agli attuatori e all'ESC tramite le uscite PWM della scheda, inoltre presenta connessioni fisiche con il sensore di velocità (pitot), il modulo GNSS e il ricevitore RC. I sensori inerziali (accelerometri e giroscopi), il magnetometro e i barometri sono invece integrati internamente nel flight controller. Per tale motivo, il dispositivo deve essere rigidamente fissato all'interno della fusoliera, evitando qualsiasi movimento relativo che comprometterebbe la qualità delle misure inerziali e, di conseguenza, l'accuratezza della stima di assetto e stato del velivolo.
Per motivi di spazio e al fine di garantire una corretta connessione con tutti i sottosistemi di bordo, il flight controller è stato installato con l'asse longitudinale allineato alla direzione di prua del velivolo, ma con una rotazione di 90° attorno all'asse di rollio. Tale configurazione è stata correttamente impostata all'interno di QGroundControl, consentendo una corretta interpretazione delle misure dei sensori e una calibrazione coerente del magnetometro.

Figura 4.2: Orientazione del Flight controller

Dopo aver fissato il flight controller all'interno dell'alloggiamento, è opportuno procedere alla calibrazione dei sensori integrati all'interno di esso. La calibrazione di tali sensori è guidata da QGroundControl all'interno della sezione *Vehicle Setup > Sensors Setup*. Per la calibrazione del magnetometro è necessario procedere ad una serie di rotazioni del velivolo attorno ad assi specifici, bisogna quindi scegliere una posizione con sufficiente

spazio per il movimento e lontano da campi magnetici. Analogamente, la calibrazione degli accelerometri viene eseguita posizionando il velivolo secondo una serie di orientamenti indicati dal software. Una volta terminati tutti gli step, viene completata la calibrazione. Per quanto riguarda i giroscopi è invece sufficiente posizionare il velivolo su una superficie piana, e lasciarlo fermo finchè la calibrazione eseguita da QGroundControl non è terminata.

Avionica di bordo

Oltre al CubePilot, all'interno dell'alloggiamento è presente quasi tutta l'avionica di bordo:

- La batteria è posizionata nella cavità sotto il bordo d'attacco dell'ala. Essa è collegata direttamente all'ESC, il quale svolge il ruolo di interfaccia tra il flight controller e il motore elettrico, convertendo il segnale di comando PWM proveniente dal FC in potenza elettrica modulata per il motore.
- Il ricevitore RC è posizionato affianco al FC e collegato ad esso. Dopo l'associazione con il trasmettitore, è stata eseguita la calibrazione dei comandi tramite procedura che consente di mappare correttamente le escursioni delle leve del radiocomando sui comandi principali (rollio, beccheggio, imbardata e throttle), garantendo una corretta interpretazione degli input manuali da parte del sistema di controllo.
- Il modulo GNSS è connesso al flight controller tramite la porta dedicata. Esso è installato esternamente sul ventre della fusoliera, in una zona sufficientemente distante da sorgenti di disturbo elettromagnetico, ed è fissato mediante adesivo.
- Il sensore di velocità (tubo di pitot) è montato sul ventre della semiala sinistra dell'UAV, sufficientemente distante dalla fusoliera e dall'alettone. Anch'esso viene collegato al FC e calibrato attraverso QGroundControl.

Gestione dei log di volo

Per impostazioni predefinite il logging dei dati di volo inizia in maniera automatica quando il velivolo viene armato, e termina quando viene disarmato. Per ogni volo viene creato un nuovo file *.ulg* all'interno della microSD presente nel Cube Orange+, tuttavia di default vengono salvati molteplici messaggi uORB, ad una specifica frequenza di log (10 Hz).
Ai fini del presente lavoro, risulta invece necessario registrare esclusivamente un sottoinsieme selezionato di variabili, ma con frequenze di campionamento significativamente più elevate (fino a 200 Hz). Questa scelta è motivata dalla necessità di analizzare con elevata risoluzione temporale l'evoluzione dei comandi durante le prove di volo e di riutilizzare tali segnali come ingressi nella simulazione Hardware-in-the-Loop. In questo modo è possibile effettuare un confronto diretto e accurato tra i risultati sperimentali e quelli ottenuti in ambiente simulato. Per una corretta ricostruzione dei segnali e per garantire la coerenza con il tempo di campionamento adottato nella simulazione, è quindi fondamentale disporre di dati acquisiti a frequenza sufficientemente elevata. Inoltre, poiché nelle fasi successive del lavoro vengono confrontate anche le variabili di stato del

velivolo (posizione, velocità, assetto, ecc.) tra le due configurazioni di studio, risulta essenziale una descrizione temporale il più possibile accurata dell'evoluzione del moto. Per soddisfare tali requisiti, è stato creato il file `logger_topics.txt` all'interno della cartella *etc/logging* della scheda microSD. In questo file sono elencati i messaggi da registrare, ciascuno associato a un valore numerico che rappresenta l'intervallo di tempo Δt, espresso in millisecondi, tra due successive acquisizioni del dato. Il file utilizzato è riportato in Figura 4.3.

```
actuator_controls_status_0 10
actuator_controls_0 5
actuator_motors 5
actuator_outputs 5
actuator_servos 5
airspeed_validated 5
estimator_states 5
home_position 5
manual_control_setpoint 5
sensor_accel 5
sensor_baro 5
sensor_gps 5
sensor_gyro 5
sensor_mag 5
vehicle_acceleration 5
vehicle_angular_velocity 5
vehicle_attitude 5
vehicle_air_data 5
vehicle_global_position 5
vehicle_imu 5
vehicle_local_position 5
wind 5
```

Figura 4.3: File `logger_topics.txt`

Descrizione del volo

L'esecuzione delle prove di volo è stata finalizzata esclusivamente alla validazione del modello dinamico descritto in precedenza. In simulazione sono stati infatti applicati gli stessi comandi impartiti durante il volo reale, così da consentire un confronto diretto dell'evoluzione delle variabili di stato del velivolo e verificare che la dinamica simulata riproduca in modo fedele il comportamento reale.
Per tale motivo, precedentemente al volo è stato selezionato un insieme di manovre progettate per consentire lo studio sia della dinamica longitudinale sia di quella latero-direzionale del velivolo, garantendo al contempo un'adeguata eccitazione dei modi dinamici fondamentali.

Tuttavia, la scelta delle manovre è stata vincolata dalla necessità di non discostarsi eccessivamente dalla condizione di equilibrio attorno alla quale è stata eseguita la linearizzazione del modello dinamico. Un'eccessiva deviazione da tale punto di lavoro avrebbe infatti reso non più valida l'approssimazione lineare adottata, compromettendo l'affidabilità del confronto tra simulazione e volo reale.
Per questa ragione sono state eseguite manovre semplici e controllate, caratterizzate da piccole deflessioni delle superfici di controllo e da variazioni graduali dei comandi, evitando manovre brusche o fortemente non lineari rispetto allo stato di equilibrio.
L'esecuzione del volo è stata affidata a un pilota certificato ed è stata condotta presso il campo di volo *"Volare su Tetti"* di Torino. Successivamente alla verifica, da parte del pilota, della corretta mappatura dei comandi di volo sul radiocomando, è stato effettuato il decollo del velivolo, a seguito del quale sono state eseguite le seguenti manovre:

- **Rampa di salita**: Partendo da una condizione di equilibrio in volo rettilineo e livellato, è stata eseguita una manovra di salita mediante una rampa caratterizzata da un rateo di salita pressoché costante, pari a circa 7.5 m/s, mantenuta per una durata di circa 10 secondi.
- **Virata a quota costante**: È stata eseguita una manovra di virata con un angolo di rollio di circa $-20°$, mantenuta fino al completamento di un giro completo, senza variazioni significative di quota.
- **Accelerazione a manetta massima**: È stata eseguita una breve fase di accelerazione in volo a quota pressoché costante, incrementando il comando di throttle fino al raggiungimento del valore massimo.
- **Step di equilibratore**: È stato applicato un breve comando di tipo *step-and-return* sull'equilibratore, della durata di circa mezzo secondo, con l'obiettivo di eccitare in modo controllato e facilmente osservabile i principali modi della dinamica longitudinale del velivolo.
- **Step di timone**: È stato applicato un breve comando di tipo *step-and-return* sul timone, mantenuto per circa un secondo, con lo scopo di eccitare e rendere osservabili i principali modi della dinamica latero-direzionale.

É necessario considerare che il pilotaggio manuale del velivolo introduce inevitabilmente una variabilità nei comandi e nelle condizioni operative, rendendo difficile la perfetta riproducibilità delle manovre elencate. Inoltre, la presenza di disturbi esterni, quali raffiche di vento o lievi variazioni delle condizioni atmosferiche, può influenzare la risposta dinamica del velivolo, in particolare nei modi a bassa frequenza.
Tuttavia nel complesso, le prove di volo hanno fornito un insieme di dati coerente e sufficientemente ricco per la validazione del modello dinamico e per l'analisi dei principali modi di moto del velivolo, costituendo una base solida per i confronti presentati nei capitoli successivi.

Figura 4.4: FunCub NG in fase di preparazione al volo

4.2 Simulazione

Questa prima fase di simulazione ha l'obiettivo di replicare, all'interno dell'ambiente numerico, le condizioni operative e i comandi di volo impartiti durante le prove reali, limitatamente alle finestre temporali descritte in sezione 4.1.

A tal fine, si è scelto di escludere l'autopilota dal processo di simulazione, concentrandosi esclusivamente sulla dinamica del velivolo. La simulazione è stata quindi condotta interamente all'interno dell'ambiente `"UAV Dynamics"`, descritto in sezione 2.3.1, bypassando i blocchi dedicati alla gestione della comunicazione con l'autopilota PX4.
In questa configurazione è stata pertanto eseguita una simulazione puramente numerica della dinamica del velivolo, finalizzata alla valutazione del modello e al confronto diretto con i dati di volo reale. La simulazione hardware-in-the-loop vera e propria, comprensiva dell'interazione con il flight controller, verrà invece trattata in una fase successiva del lavoro per la taratura del controllore di volo..
Al fine di ottenere un confronto significativo tra simulazione e volo reale all'interno delle finestre temporali selezionate, risulta necessario inizializzare lo stato del velivolo simulato in modo coerente con la configurazione assunta dal velivolo reale in un istante ben definito precedente all'esecuzione della manovra. In questo modo è possibile applicare, a partire da tale istante, gli stessi comandi inviati alle superfici di controllo durante il volo reale. L'approccio adottato consente quindi di definire un ambiente simulato le cui condizioni iniziali coincidono esattamente con quelle del volo reale all'istante t_1, e la cui evoluzione temporale è governata esclusivamente dalle equazioni dinamiche del modello, alimentate in ingresso da una *timeseries* contenente i comandi di volo effettivamente impartiti nel corso dell'intervallo temporale $[t_1, t_2]$.
La durata dell'intervallo temporale simulato è stata limitata a un massimo di circa dieci secondi. Tale scelta è motivata dal fatto che il modello dinamico impiegato rappresenta un'approssimazione che genera inevitabilmente discrepanze rispetto al comportamento reale, che vengono amplificate ad ogni passo d'integrazione. Superata una certa durata della simulazione, le variabili di stato simulate possono quindi divergere in maniera significativa da quelle reali, per questo motivo l'analisi è stata limitata a finestre temporali sufficientemente brevi da garantire la validità del confronto tra i due domini.

Elaborazione del file ULog

Il macro-blocco `"UAV Dynamics"` è stato quindi modificato in modo tale da sostituire il segnale `controls` proveniente da PX4 con una *timeseries* creata nel workspace all'interno della quale è presente l'evoluzione nell'intervallo $[t_1, t_2]$ dei valori adimensionalizzati delle deflessioni delle superfici mobili e della manetta. La restante parte del blocco `"Actuator Model"` è stata mantenuta inalterata, come mostrato in figura 4.5.
La timeseries è stata generata a partire dal file *.ulg* contenente il log di volo della prova sperimentale. All'interno di tale file sono registrati tutti i messaggi uORB, elencati in Figura 4.3. Tra questi, sono stati selezionati i messaggi *actuator_servos*, che riportano le deflessioni adimensionalizzate delle superfici di controllo, e *actuator_motors*, contenente il valore della manetta.

L'elaborazione dei suddetti messaggi e la successiva costruzione della timeseries `u_ts` sono state effettuate mediante lo script MATLAB riportato nel listing 4.1.

I messaggi vengono importati in MATLAB sotto forma di *timetable*, in cui ciascuna riga è associata ad un istante temporale, mentre le colonne riportano i valori delle variabili del segnale. Il primo passo consiste nell'estrazione dell'intervallo temporale di interesse, che in generale è definito come $[t_1, t_1 + 10s]$. Per fare questo bisogna identificare l'indice corrspondente all'istante t_1, considerando che la variabile temporale nella *timetable* è di tipo duration e non è inizializzata a zero, mentre il nostro riferimento temporale t_1 ci indica il tempo in cui inizia la manovra assumendo come $t = 0$ l'istante di avvio del log dati di PX4. Viene indicato con j_1 l'indice corrispondente all'istante t_1.
Considerando infine che i segnali sono stati campionati durante il volo ad una frequenza di 200 Hz, per ottenere una finestra temporale di 10 secondi è sufficiente considerare 2000 step temporali. La creazione della *timeseries* è stata quindi effettuata limitando i messaggi *actuator_servos* e *actuator_motors* all'intervallo di indici $[j_1, j_1 + 2000]$.

Le colonne dei messaggi che identificano i quattro comandi di volo vengono poi uniti all'interno del vettore U. In questo passaggio, i segnali relativi all'equilibratore e al timone vengono invertiti di segno, poichè la convenzione adottatata da PX4 differisce rispetto a quella utilizzata nel modello dinamico. In questo modo quindi verranno considerate con angolo positivo una deflessione dell'equilibratore verso il basso e una deflessione del timone verso la semiala sinistra.
Infine viene creata una *timetable* a partire dal vettore dei comandi U, utilizzando una variabile temporale reinizializzata in modo tale che $t_1 = 0$. Questa *timetable*, viene poi convertita in una *timeseries* tramite l'apposito comando MATLAB.

La *timeseries* ottenuta a partire dai dati di volo è costituita da campioni acquisiti a una frequenza pari a 200 Hz. Tuttavia, le simulazioni numeriche sono state eseguite utilizzando un passo di integrazione pari a 0.001 s, corrispondente a una frequenza di simulazione di 1000 Hz.
Tale discrepanza tra la frequenza di campionamento dei dati di ingresso e il passo temporale della simulazione è gestita mediante il blocco Simulink `"From Workspace"`, il quale consente di fornire in ingresso una *timeseries* caratterizzata da un rate di campionamento differente rispetto a quello della simulazione. In particolare, il blocco esegue una ricostruzione del segnale di ingresso mantenendo costante ciascun valore della *timeseries* fino alla disponibilità del campione successivo.
Di conseguenza, il segnale in uscita dal blocco risulta definito alla frequenza di 1000 Hz, mentre i valori originariamente campionati a 200 Hz vengono mantenuti invariati per più step temporali consecutivi, realizzando di fatto un'operazione di tipo *zero-order hold*.

Un'analoga procedura di elaborazione è stata applicata anche al messaggio uORB *wind*, che contiene le informazioni relative al vento misurato durante le prove sperimentali. In particolare, tale messaggio fornisce una stima dell'intensità del vento relativo nel sistema di riferimento NED (North–East–Down), ottenuta a partire dalle misure del tubo di Pitot installato a bordo del velivolo.

A partire da questo messaggio è stata costruita una *timeseries* contenente le componenti del vento nelle tre direzioni del sistema di riferimento NED. Le componenti orizzontali, ovvero quelle lungo gli assi North ed East, sono direttamente ricavate dalle misure del sensore di velocità, mentre la componente verticale del vento è stata assunta nulla.
La *timeseries* così ottenuta viene infine fornita come ingresso al blocco `"Force and Moment Calculation"`, all'interno del quale le informazioni sul vento vengono utilizzate per il calcolo delle velocità relative aria–velivolo e, conseguentemente, per la determinazione delle forze e dei momenti aerodinamici agenti sul velivolo.

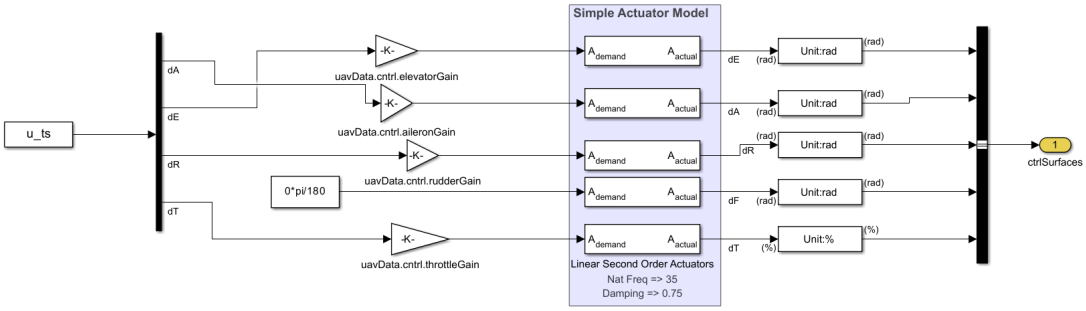


Figura 4.5: Blocco `"Actuator Model"`

```
%% LETTURA COMANDI VOLO REALE

uLogOBJ=ulogreader("prova_volo.ulg");
t1= 116.5;

% Leggi comandi superfici mobili
dat=readTopicMsgs(uLogOBJ,'TopicNames',{'actuator_servos'});
actuator_servos=dat.TopicMessages{1,1};

% Trovo indice corrispondente al primo istante superiore al
   tempo t1
for i=1:size(actuator_servos, 1)
    tempo= seconds(actuator_servos.timestamp(i)-
   actuator_servos.timestamp(1));
    if tempo > t1
        j1=i;
        break
    end
end
actuator_servos=actuator_servos(j1:j1+2000, :);

% Leggi comandi motori
dat=readTopicMsgs(uLogOBJ,'TopicNames',{'actuator_motors'});
actuator_motors=dat.TopicMessages{1,1};
actuator_motors=actuator_motors(j1:j1+2000, :);

% Creo vettore dei comandi
U = [actuator_servos.control(:,1), -actuator_servos.control
   (:,3), -actuator_servos.control(:,4), actuator_motors.
   control(:,1)];
U = fillmissing(U,'constant',0);

% Creo TimeTable reinizializzando il tempo
controls_TT = timetable(actuator_servos.timestamp(:)-
   actuator_servos.timestamp(1), U);

% Creo Timeseries
u_ts = timeseries(controls_TT.U, seconds(controls_TT.Time));
```

Listing 4.1: Elaborazione dei comandi di volo reali a partire da un file ULog

4.3 Confronto variabili di stato

Il confronto tra le variabili di stato è stato condotto su finestre temporali della durata di dieci secondi, ciascuna rappresentativa di una specifica fase del volo.
Al fine di garantire la significatività dell'analisi, il confronto è stato limitato alle sole grandezze direttamente misurabili dai sensori installati a bordo del velivolo. Sono state pertanto escluse variabili quali l'angolo d'attacco α e l'angolo di deriva β, che, pur rivestendo un ruolo fondamentale nella generazione delle forze e dei momenti aerodinamici, non sono direttamente disponibili nei log di volo. Tali grandezze possono infatti essere unicamente stimate in modo indiretto a partire dalle misure del tubo di Pitot e dei sensori inerziali (IMU), introducendo inevitabilmente un ulteriore livello di incertezza nel confronto tra simulazione e dati sperimentali.
La rappresentazione dell'evoluzione temporale delle grandezze misurate sperimentalmente è stata resa possibile mediante l'elaborazione dei messaggi uORB effettuata in MATLAB. Le variabili di stato sono state estratte dai principali topic elencati in figura 4.3. In particolare:

- **Velocità angolari:** ricavate direttamente dal messaggio uORB `vehicle_angular_velocity`. Esse rappresentano le misure delle velocità angolari nel sistema di riferimento *body*, corrette dal bias (*bias-corrected*), fornite dai tre giroscopi integrati nel flight controller.
- **Angoli di assetto:** ottenuti convertendo in angoli di Eulero i quaternioni loggati nel messaggio `vehicle_attitude`. Tali grandezze non costituiscono una misura diretta dei sensori di bordo, bensì il risultato del processo di integrazione delle velocità angolari e di fusione sensoriale implementato dall'algoritmo EKF2 (*Extended Kalman Filter*).
- **Accelerazioni lineari:** le accelerazioni nel sistema di riferimento *body* sono ricavate dal messaggio `vehicle_acceleration`, che fornisce le misure corrette degli accelerometri. Tali valori sono stati successivamente rielaborati sottraendo il contributo dell'accelerazione gravitazionale.
- **Velocità:** estratta dal messaggio `airspeed_validated`, che fornisce il modulo della velocità vera del velivolo stimato mediante la combinazione delle misure provenienti da più sensori e dall'algoritmo EKF.
- **Posizione:** il messaggio `vehicle_local_position` fornisce la posizione del velivolo nel sistema di riferimento NED, stimata a partire dalle misure GPS e dalla fusione con gli altri sensori attraverso il filtro di Kalman.

Per quanto riguarda i dati simulati, ai fini del confronto sono state considerate esclusivamente le variabili di stato fornite in uscita dal blocco `"6DOF Rigid Body Dynamics"`.

4.3.1 Salita

Dall'analisi comparativa riportata nei grafici seguenti si osserva come la simulazione riesca a riprodurre correttamente l'andamento globale della manovra di salita nella prima parte dell'intervallo temporale considerato, introducendo tuttavia un errore sistematico che determina una progressiva divergenza tra le due curve a partire da circa 4 s dall'inizio della simulazione.
Tale comportamento è chiaramente evidenziato in figura 4.6, dove è mostrato il confronto tra la velocità angolare di beccheggio simulata e quella misurata dai giroscopi di bordo. Il modello è in grado di catturare la successione dei picchi e di riprodurre qualitativamente la dinamica oscillatoria principale; tuttavia, tende a sovrastimare i picchi positivi di q e a mantenere valori medi più elevati rispetto al dato sperimentale. Questo scostamento si riflette direttamente nell'andamento dell'angolo di pitch riportato in figura 4.7, dove, oltre i 4 s dall'inizio della simulazione, l'angolo θ risulta progressivamente sovrastimato, con un offset crescente rispetto ai dati reali. Tale deriva conduce verosimilmente il modello verso una condizione prossima allo stallo, nella quale il velivolo simulato manifesta una rapida riduzione dell'incidenza.
Il comportamento descritto suggerisce una possibile sovrastima dei coefficienti aerodinamici longitudinali responsabili del momento di beccheggio, quali C_{L_α} o C_{m_α}, oppure una sottostima del termine di smorzamento dinamico C_{m_q}. Inoltre, la divergenza progressiva tra simulazione e prova di volo evidenzia un disallineamento tra la condizione di equilibrio del modello e quella effettivamente assunta dal velivolo reale, indicando che il punto di trim simulato potrebbe non coincidere esattamente con quello sperimentale.
Una possibile spiegazione risiede nell'offset presente durante le prove sperimentali tra l'effettivo asse X_{body}, allineato con la direzione di spinta, e l'asse X_{body} definito a seguito della calibrazione dei sensori e dell'orientamento del flight controller all'interno dell'alloggiamento del FunCub. Un'ulteriore conferma di tale ipotesi è fornita dalla figura 4.8, in cui è riportata l'accelerazione lineare lungo Z_{body}: pur seguendo correttamente la dinamica del velivolo reale, il segnale simulato presenta un offset costante sin dai primi istanti, con una sistematica sovrastima dell'accelerazione verticale.

Nonostante le discrepanze precedentemente evidenziate, nel complesso il modello risulta comunque in grado di descrivere in modo soddisfacente la dinamica longitudinale del velivolo. In figura 4.9 è riportato il confronto tra la traiettoria simulata e quella ottenuta dai dati di volo: si osserva come, almeno nei primi istanti della manovra, la dinamica simulata segua con buona approssimazione il percorso realmente compiuto durante le prove sperimentali.
Tale risultato assume particolare rilievo considerando che il rateo di salita applicato si colloca ai limiti della dinamica lineare del modello, condizione che può amplificare eventuali errori di approssimazione.

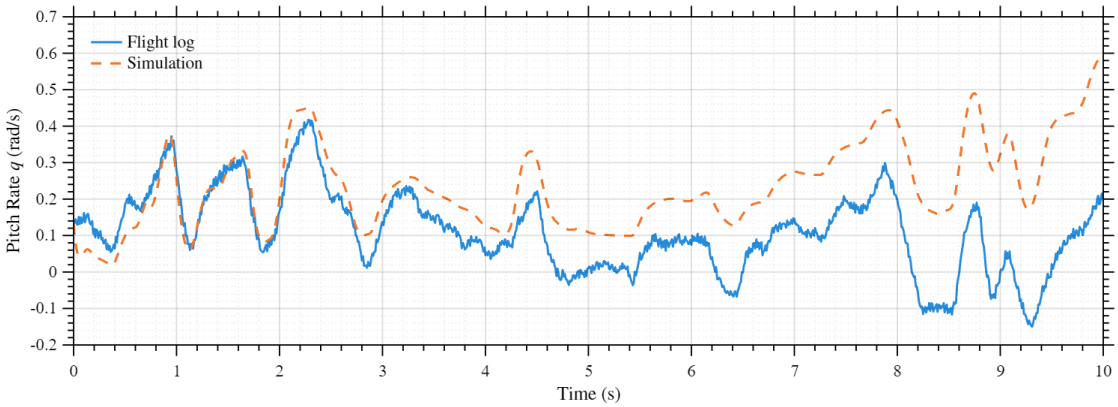


Figura 4.6: Confronto velocità angolare di beccheggio in salita

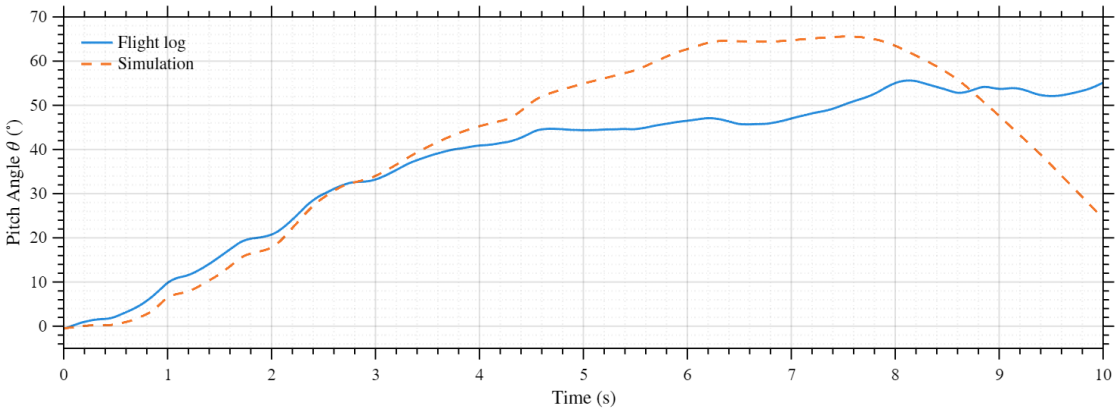


Figura 4.7: Confronto dell'angolo di beccheggio θ in salita

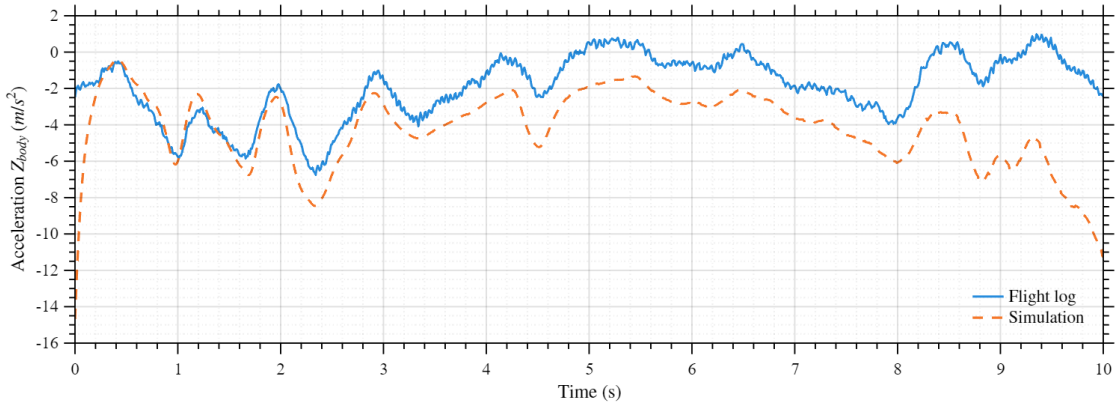


Figura 4.8: Confronto dell'accelerazione in direzione Z_{body} in salita

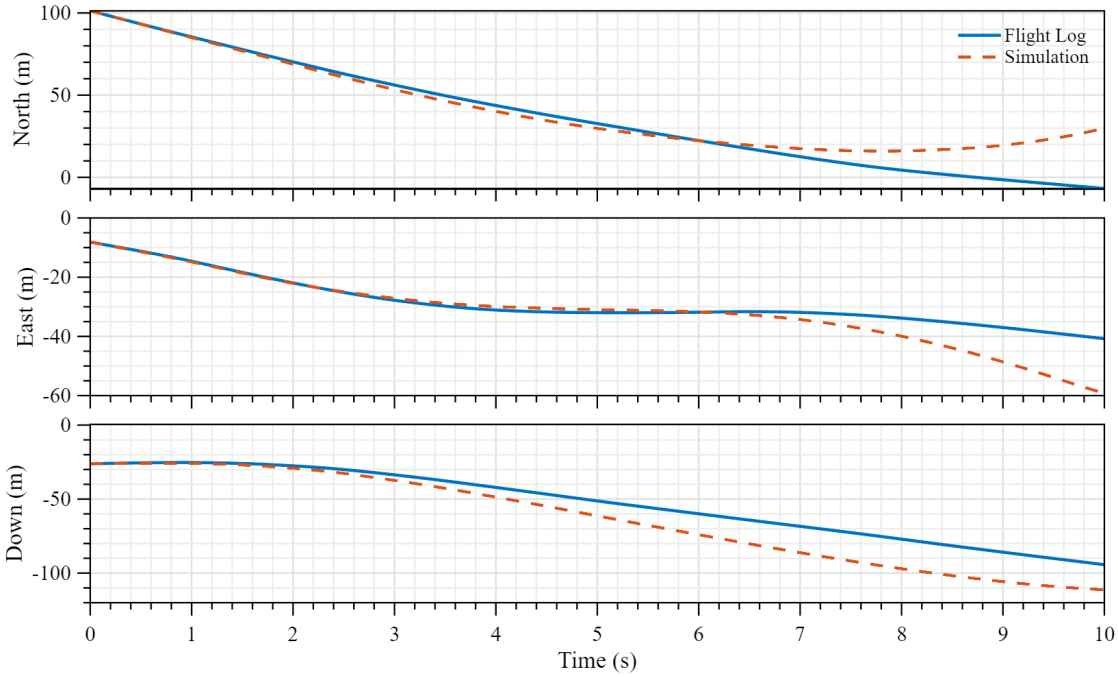


Figura 4.9: Confronto della traiettoria di volo in salita in assi NED

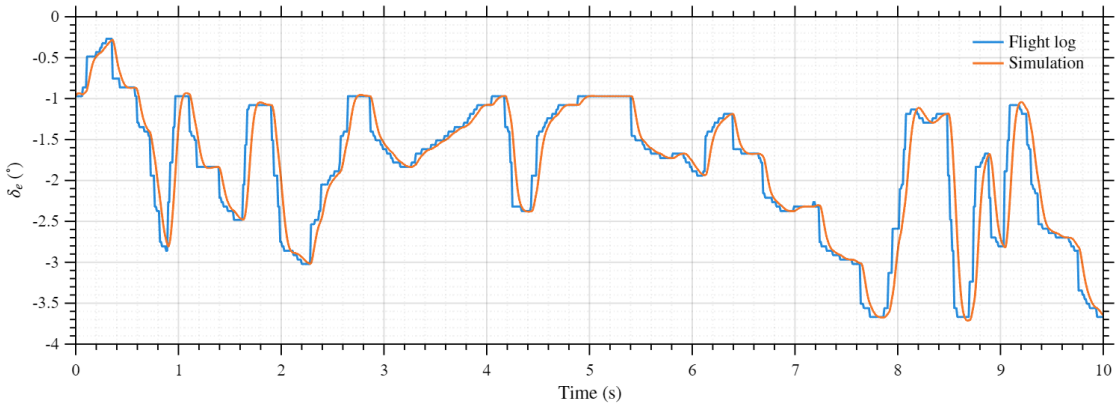


Figura 4.10: Comando di elevatore δ_e in salita

In figura 4.10 è riportata l'evoluzione del comando di elevatore nella finestra temporale analizzata. Il tracciato in blu, relativo alla prova sperimentale, rappresenta il segnale PWM generato dal radiocomando e successivamente convertito in deflessione angolare della superficie; esso risulta pertanto costituito da valori discretizzati nel tempo. Il segnale relativo alla simulazione, invece, è ottenuto a valle del modello dinamico dell'attuatore, il quale tiene conto del tempo necessario alla superficie di controllo per raggiungere la posizione richiesta, nonché delle eventuali oscillazioni associate alla dinamica del servo. È quindi opportuno considerare che la modellazione dell'attuatore introduce un ulteriore livello di incertezza nel sistema simulato, potendo contribuire alla propagazione degli errori e influenzare il confronto complessivo tra dinamica simulata e dati sperimentali.

4.3.2 Virata

Il confronto tra volo reale e simulazione durante la manovra di virata evidenzia come il modello sia in grado di riprodurre correttamente l'andamento qualitativo delle principali grandezze dinamiche, pur presentando alcune discrepanze riconducibili alla modellazione della dinamica latero–direzionale.
Per quanto riguarda la velocità angolare di rollio p, rappresentata in figura 4.11, la simulazione riproduce correttamente la struttura temporale della manovra e i cambi di segno associati alle variazioni di comando. Tuttavia, l'ampiezza della risposta risulta sistematicamente inferiore rispetto ai dati di volo, con una sottostima dei picchi positivi e una dinamica complessivamente più smorzata, che non riesce a riprodurre le variazioni di alta frequenza. Questo comportamento può essere imputabile ad una sovrastima della derivata di smorzamento C_{ℓ_p} o ad una sottostima della derivata di comando $C_{\ell_{\delta_a}}$, ma può essere anche dovuto ad effetti non lineari dei quali il modello non tiene conto. È importante sottolineare che, nel processo di taratura del modello, si è reso necessario ridurre significativamente le derivate aerodinamiche C_{ℓ_β} e C_{n_β}, in quanto i valori inizialmente adottati portavano a gravi errori nella riproduzione della dinamica longitudinale. La grandezza β risulta infatti una delle variabili meno correttamente riprodotte dal modello, con ampiezze e dinamica temporale significativamente diverse rispetto ai dati sperimentali. Tale scelta, introduce inevitabilmente un livello di incertezza nella dinamica laterale, specialmente durante manovre di virata coordinata, dove il contributo di β ai momenti di rollio e imbardata risulta determinante.
Poichè l'angolo di rollio ϕ rappresenta l'integrale della velocità di rollio, la sottostima dei momenti aerodinamici laterali si riflette direttamente sull'angolo ϕ, rappresentato in figura 4.12, che presenta un'inclinazione massima maggiore rispetto al caso reale. Parte dello scostamento osservato può tuttavia essere attribuita anche a possibili differenze tra il sistema di riferimento reale e quello simulato. Un non perfetto allineamento del CubePilot rispetto all'asse Z_{body} effettivo del velivolo può infatti introdurre un offset nelle condizioni di equilibrio. Un comportamento analogo si riscontra nell'angolo di imbardata ψ, riportato in figura 4.13, il quale presenta un errore sistematico pressoché costante per l'intera durata della simulazione, compatibile con un disallineamento iniziale tra le terne di riferimento.

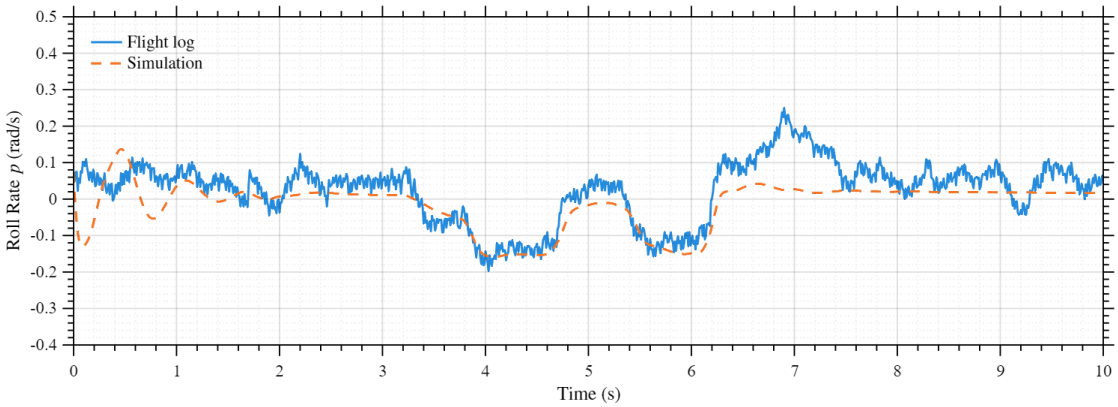


Figura 4.11: Confronto velocità angolare di rollio p in virata

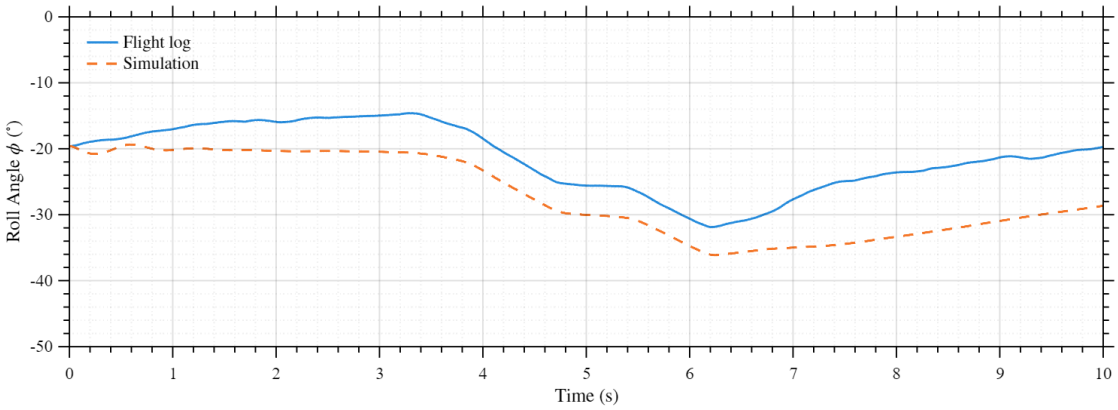


Figura 4.12: Confronto dell'angolo di rollio ϕ in virata

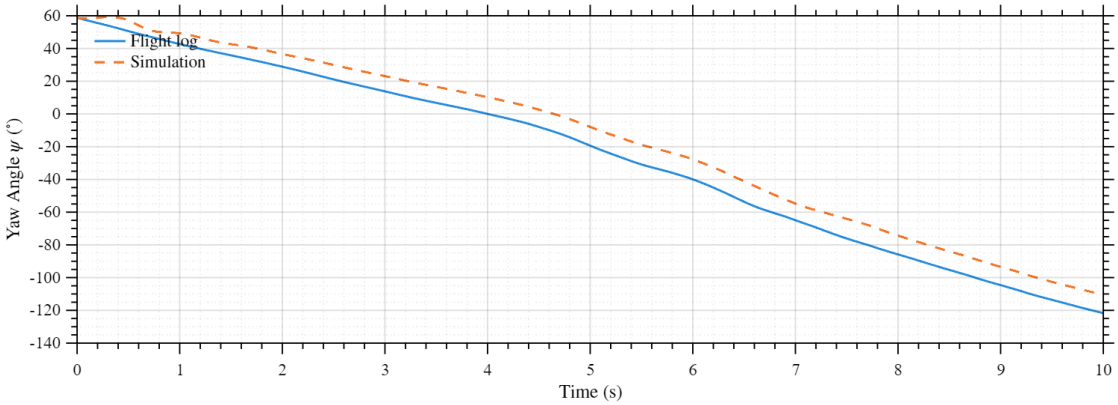


Figura 4.13: Confronto dell'angolo di imbardata ψ in virata

Nonostante le differenze precedentemente evidenziate nella dinamica latero–direzionale, il modello dinamico riesce a riprodurre in maniera soddisfacente la velocità di avanzamento del velivolo, riportata in figura 4.14. In particolare, l'andamento temporale della velocità risulta coerente con i dati sperimentali e ne viene correttamente catturata l'evoluzione pressoché costante durante la manovra di virata.
Anche per quanto riguarda la velocità angolare di imbardata r, in figura 4.15 si osserva una buona sovrapposizione tra simulazione e volo reale. La grandezza si mantiene approssimativamente costante per l'intera durata della manovra, in accordo con il comportamento atteso per una virata stabilizzata.
Analogamente, per quanto riguarda l'accelerazione lungo la direzione Y_{body}, in figura 4.16 si osserva una buona sovrapposizione tra le curve simulata e sperimentale. Pur in presenza di lievi differenze locali, il modello è in grado di descrivere in maniera adeguata la distribuzione delle forze laterali durante la virata, indicando che l'errore riscontrato nelle grandezze angolari non compromette in modo significativo la rappresentazione delle accelerazioni globali.
Infine, in figura 4.17 la traiettoria in assi NED risulta globalmente ben riprodotta nell'intervallo temporale analizzato, con scostamenti limitati nelle coordinate North, East e Down. Tale risultato conferma che, per finestre temporali dell'ordine di alcuni secondi, il modello è in grado di descrivere in modo soddisfacente l'evoluzione globale del moto del velivolo, nonostante le semplificazioni introdotte nella modellazione aerodinamica.
In figura 4.18 è infine riportata la deflessione dell'alettone applicata nell'intervallo temporale considerato, a scopo di confronto tra l'ingresso registrato durante la prova di volo e la deflessione modellata in Simulink.

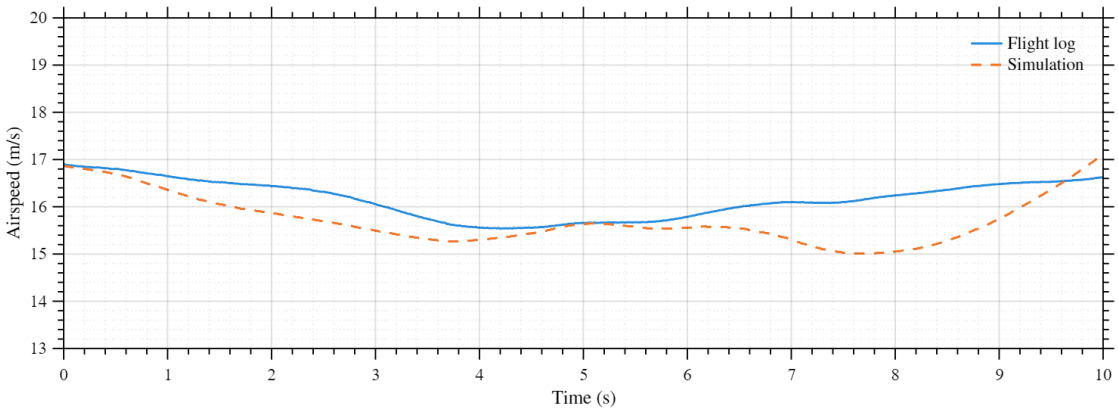


Figura 4.14: Confronto della velocità di volo in virata

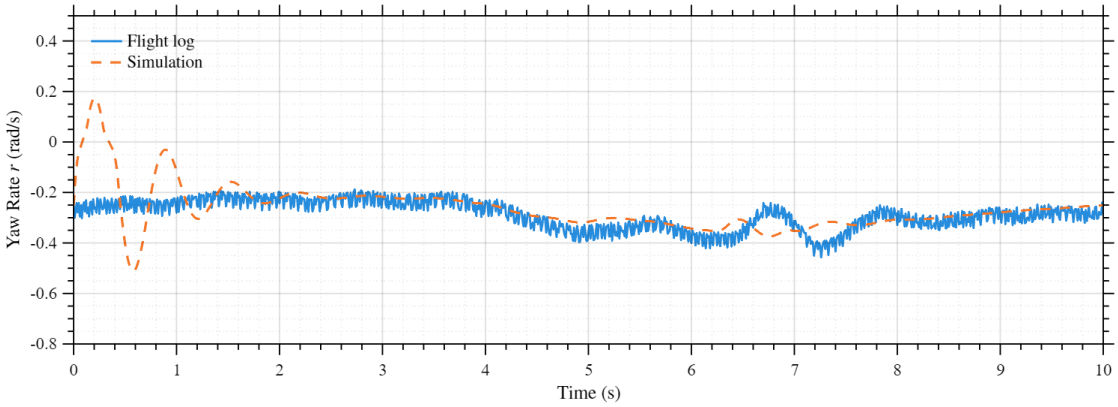


Figura 4.15: Confronto velocità angolare di imbardata r in virata

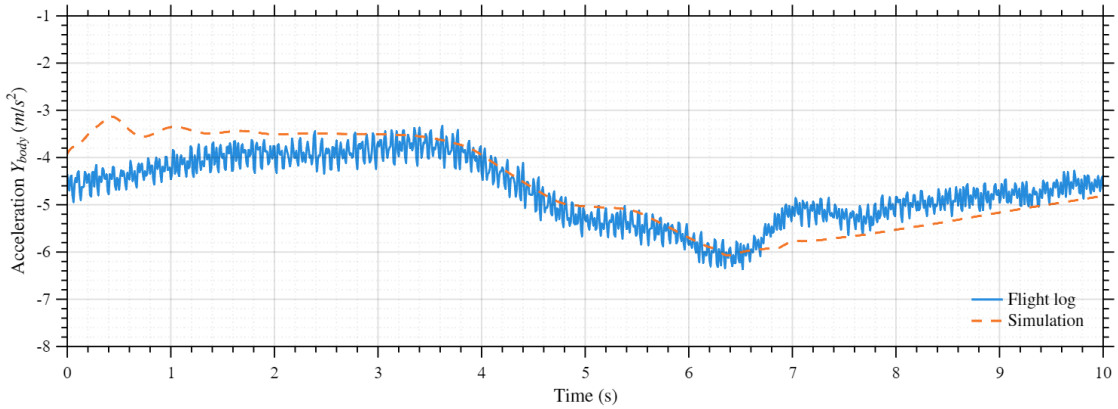


Figura 4.16: Confronto dell'accelerazione in direzione Y_{body} durante la virata

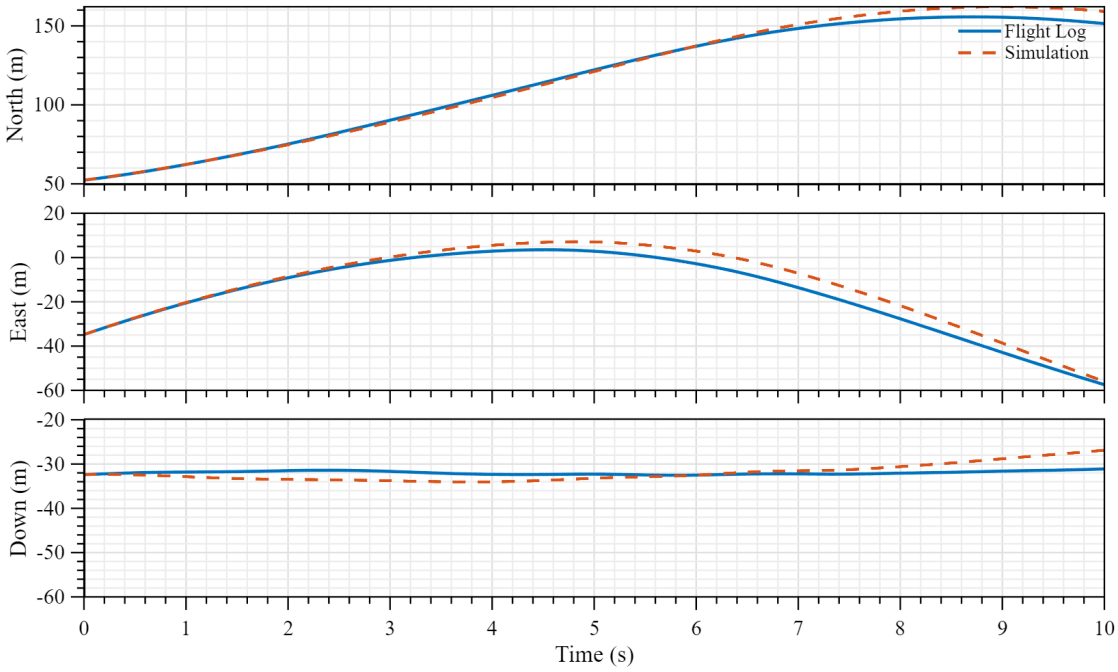


Figura 4.17: Confronto della traiettoria di volo durante la virata in assi NED

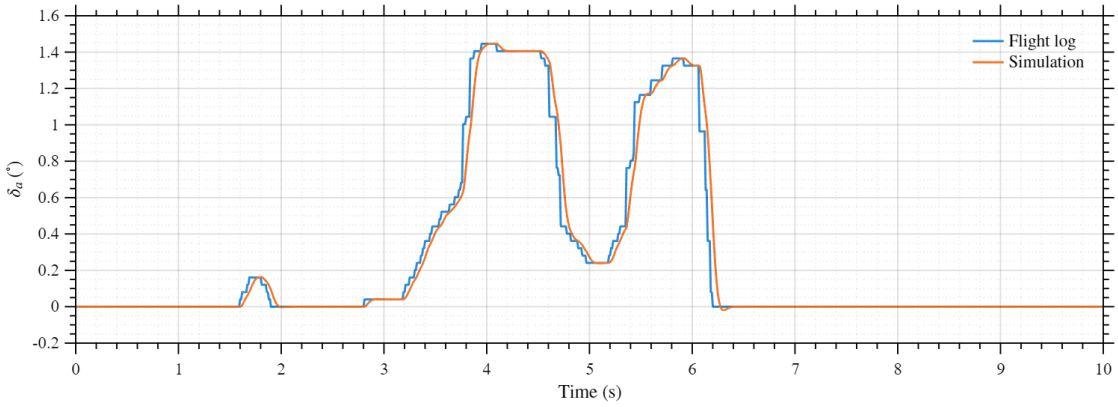


Figura 4.18: Comando di alettone δ_a in virata

4.3.3 Accelerazione

L'analisi comparativa è stata effettuata anche nel caso di volo rettilineo con rapida accelerazione del velivolo, ottenuta portando il comando di throttle al valore massimo. La finestra temporale esaminata può essere suddivisa in tre fasi: una fase iniziale in condizioni quasi stazionarie, una fase di incremento del comando di throttle e una fase finale di riduzione della potenza.

In figura 4.19 è rappresentata l'evoluzione della velocità di volo, grandezza particolarmente significativa per valutare la qualità della riproduzione della manovra. Nella fase iniziale (0–4.5 s) la velocità risulta ben riprodotta dal modello: le due curve si sovrappongono in modo soddisfacente, mantenendosi pressoché costanti intorno al valore di trim. Ciò conferma la correttezza dell'inizializzazione dello stato e la coerenza della condizione di equilibrio aerodinamico imposta in simulazione.

A partire da circa 5 s si osserva l'incremento del comando di throttle, rappresentato in figura 4.21. A tale variazione corrisponde un aumento della velocità di volo, registrato sia nel caso reale sia nella simulazione, con un andamento qualitativamente coerente tra i due casi. Tuttavia, la simulazione evidenzia una crescita leggermente più rapida e un picco di velocità superiore rispetto al dato sperimentale. Tale comportamento suggerisce una possibile sovrastima della spinta generata dal modello propulsivo, sviluppato su basi teoriche, oppure una sottostima della resistenza aerodinamica, in particolare del contributo parassita rappresentato da C_{D0}. Nella fase successiva, in corrispondenza della riduzione del throttle, la simulazione mostra una decelerazione più marcata rispetto al volo reale. Anche tale discrepanza può essere attribuita a una modellazione non pienamente accurata del sistema propulsivo, che non tiene conto di effetti dissipativi e non linearità presenti nel sistema reale.

Il confronto dell'accelerazione lungo l'asse X_{body} mette ulteriormente in evidenza tali differenze. Durante la fase di incremento di potenza, il picco di accelerazione simulato risulta maggiore rispetto a quello misurato in volo, indicando una forza risultante longitudinale più elevata nel modello. Inoltre, nella risposta simulata sono presenti due picchi, rispettivamente intorno a 3 s e a 7.5 s, che non trovano corrispondenza nel dato sperimentale e non sono riconducibili alla manovra reale; essi sono presumibilmente imputabili a imperfezioni del modello dinamico o a transitori numerici.

In figura 4.22 sono riportati i valori del coefficiente di spinta C_T e del numero di giri dell'elica n calcolati dal modello propulsivo. Sebbene non siano disponibili dati sperimentali diretti per tali grandezze, essi risultano utili per interpretare il comportamento complessivo della manovra. Si osserva come il numero di giri segua coerentemente l'andamento del comando di manetta, poiché la potenza fornita al motore si traduce direttamente in un aumento del regime dell'elica, essendo presente una trasmissione diretta tra motore ed elica. Il coefficiente di spinta, invece, risulta proporzionale al numero di giri ma inversamente dipendente dalla velocità di volo del velivolo. Quando, nella fase finale della simulazione, la velocità diminuisce in modo più marcato rispetto al caso reale, il valore di C_T aumenta significativamente. Tale incremento risulta pertanto sovrastimato a causa della decrescita di velocità non correttamente riprodotta dal modello.

Per quanto riguarda la dinamica di beccheggio, in figura 4.23 il confronto della velocità angolare q mostra un andamento qualitativamente coerente tra simulazione e volo reale.

L'incremento di q durante la fase di accelerazione è ben riprodotto, sebbene il picco simulato risulti leggermente superiore e lo smorzamento nella fase finale presenti alcune differenze.
Nel complesso, il modello dinamico è in grado di riprodurre correttamente l'andamento globale della manovra di accelerazione, cogliendone le principali variazioni di velocità e di assetto e mantenendo in modo sufficientemente accurato la traiettoria sviluppata in volo, come riportato in figura 4.24. Le discrepanze maggiori si concentrano nella fase di decelerazione successiva alla riduzione del throttle e sono attribuibili principalmente alla modellazione del sistema propulsivo e della resistenza aerodinamica. Per finestre temporali dell'ordine di alcuni secondi, il modello può pertanto considerarsi adeguato alla descrizione della dinamica longitudinale durante manovre di variazione di potenza.

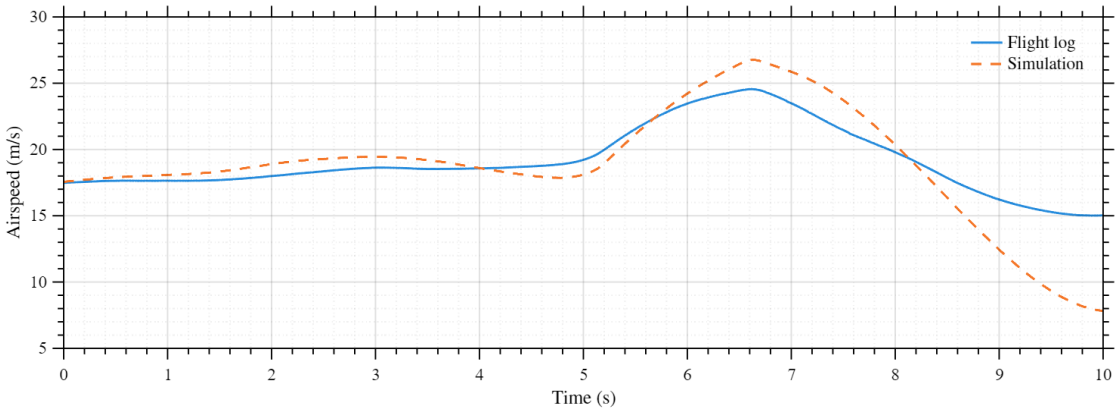


Figura 4.19: Confronto della velocità di volo durante l'accelerazione

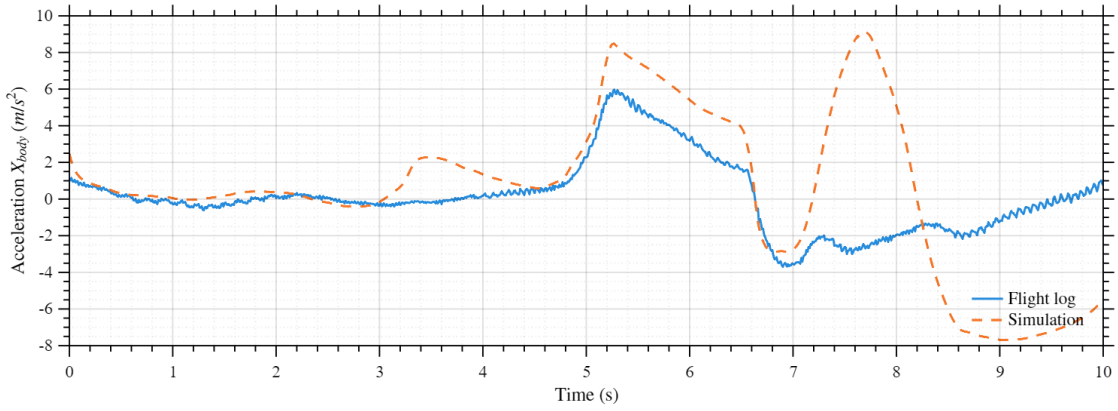


Figura 4.20: Confronto dell'accelerazione in direzione X_{body} durante l'accelerazione

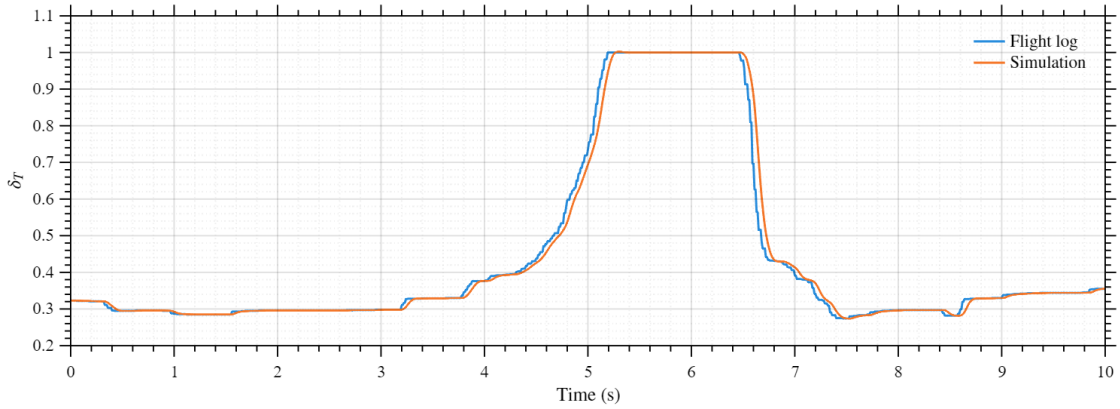


Figura 4.21: Comando di throttle δ_T durante l'accelerazione

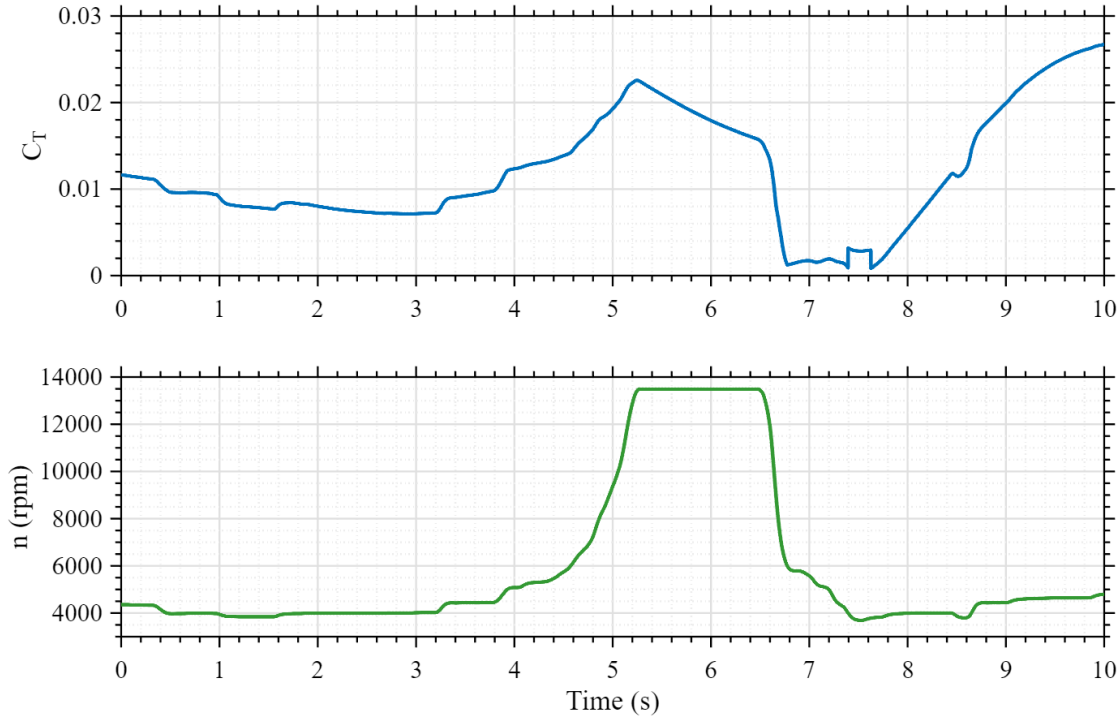


Figura 4.22: Variazione del coefficiente C_T e del numero di giri n modellati su Simulink

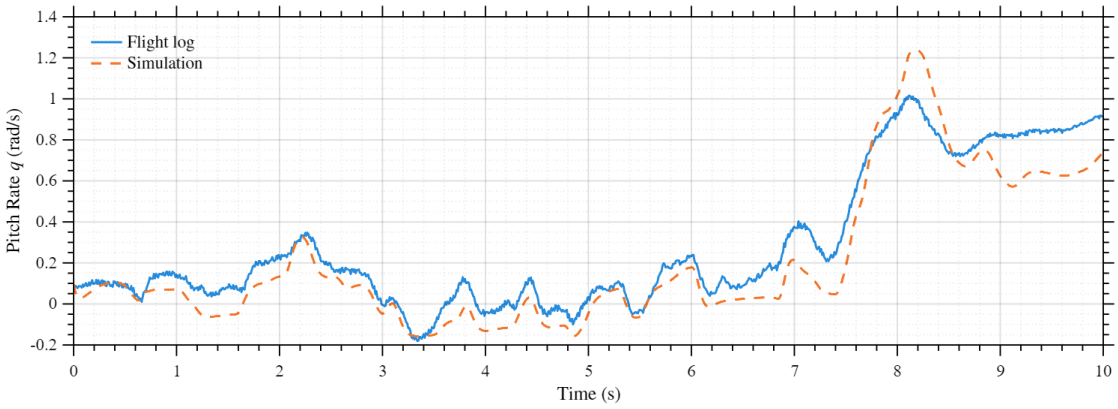


Figura 4.23: Confronto velocità angolare di beccheggio q durante l'accelerazione

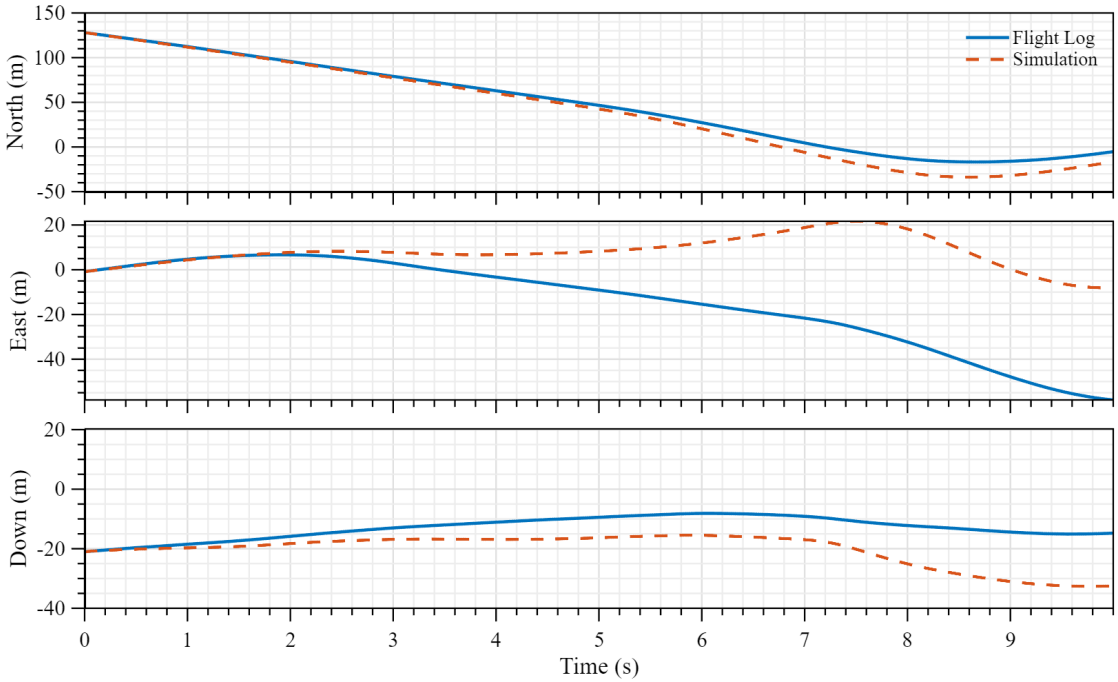


Figura 4.24: Confronto della traiettoria di volo durante la fase di accelerazione in assi NED

4.4 Confronto modi dinamici

Oltre al confronto dell'evoluzione delle variabili di stato, è stato effettuato anche un confronto relativo ai modi dinamici eccitati in seguito a comandi di tipo step applicati alle superfici mobili. Al fine di ottenere una rappresentazione quanto più possibile aderente alla risposta libera del velivolo, sono state selezionate finestre temporali di durata limitata (pochi secondi), all'interno delle quali è presente il comando di una sola superficie di controllo, mentre le restanti sono mantenute in posizione neutra.
In particolare, l'analisi è stata condotta per la valutazione del modo di short period in seguito a uno step di elevator e del modo di dutch roll in seguito a uno step di rudder. Non è stato invece possibile estrarre gli altri modi tipici della dinamica del velivolo. Il fugoide e lo spiral mode presentano infatti dinamiche troppo lente per essere adeguatamente osservate nell'ambito di un volo condotto in closed loop (con intervento del pilota), mentre il roll mode, caratterizzato da una dinamica molto rapida e prevalentemente non oscillatoria, risulta difficilmente identificabile con la metodologia adottata.
Per l'estrazione delle principali caratteristiche dei modi dinamici è stata sviluppata un'apposita funzione, riportata nel listato 4.2. A partire dall'evoluzione temporale delle variabili di stato maggiormente coinvolte nel modo analizzato, la funzione consente di stimare i parametri caratteristici del moto, quali la frequenza propria e il coefficiente di smorzamento.
La funzione è implementata in modo da eseguire un'interpolazione tramite il metodo dei minimi quadrati, adattando ai dati sperimentali una funzione modello rappresentativa di un moto oscillatorio smorzato. Tale procedura consente di ottenere una legge analitica che approssima l'andamento della risposta reale, permettendo così di identificare in maniera quantitativa i parametri dinamici del modo considerato. È importante sottolineare che l'interpolazione non viene effettuata direttamente sulla variabile di interesse, bensì sulla sua variazione rispetto a una media mobile. Tale scelta è motivata dal fatto che, ai fini dell'identificazione del modo dinamico, risulta rilevante l'evoluzione della componente oscillatoria della risposta, piuttosto che il valore assoluto assunto dalla variabile. L'interpolazione viene inoltre eseguita su una finestra temporale limitata, la cui estensione è pari a circa due o tre periodi del modo analizzato, così da isolare la dinamica dominante e ridurre l'influenza di altre componenti dinamiche.
L'equazione modello utilizzata per il fit dei modi dinamici è:

$$x(t) = A \cdot e^{-\sigma t} \sin(\omega_d \cdot t + \phi) + x_0 \tag{4.1}$$

Tramite l'interpolazione ai minimi quadrati vengono quindi definiti i cinque parametri liberi della funzione, ovvero l'ampiezza iniziale, σ, la pulsazione smorzata ω_d, la fase iniziale ϕ e il valore di equilibrio x_0. A partire da questi valori è possibile definire le caratteristiche principali del modo ed effettuare il confrontro tra dinamica reale e quella simulata.

$$f = \frac{\omega_d}{2\pi} \tag{4.2}$$

$$\omega_n = \sqrt{\sigma^2 + \omega_d^2} \tag{4.3}$$

$$\zeta = -\frac{\sigma}{\omega_n} \tag{4.4}$$

```
function [f, zita, phi]=find_modi(t, x, Fs)
global t1
global t2
global w0

% Variazione della variabile rispetto alla media mobile (
    finestra 1 s)
x_detr = x - movmean(x, round(1*Fs));

% Isolo finestra temporale e normalizzo tempo
idx = (t>=t1 & t<=t2);
tt = t(idx);
xx = x_detr(idx);
tau = tt - tt(1);

% Stime iniziali modello
A0 = max(xx) - min(xx); sigma0 = -0.5;
phi0 = 0; C0 = mean(xx);
p0 = [A0, sigma0, w0, phi0, C0];

% Funzione modello interpolante (moto oscillatorio smorzato)
model = @(p,t) p(1).*exp(p(2).*t).*sin(p(3).*t + p(4)) + p
    (5);

% Interpolazione con metodo minimi quadrati
opts = optimoptions('lsqcurvefit','Display','off');
lb = [0,   -10,  0,    -2*pi, -Inf];     % vincolo minimo
    dei parametri del polinomio p
ub = [Inf,  0,  2*pi*20, 2*pi, Inf];  % vincolo massimo dei
    parametri del polinomio p
p = lsqcurvefit(@(pp,ttt) model(pp,ttt), p0, tau, xx, lb, ub
    , opts);

% Definizione parametri
A = p(1);  sigma = p(2);  w = p(3);  phi = p(4);  C = p(5);
f   = w/(2*pi);
wn    = sqrt(sigma^2 + w^2);
zita = -sigma/wn;
end
```

Listing 4.2: Funzione per il fit dei modi dinamici tramite modello di oscillatore armonico smorzato di secondo ordine

4.4.1 Short Period

Per identificare il modo dinamico di *short period* sono state analizzate le risposte delle variabili q e α al comando di elevatore rappresentato in figura 4.25 , che risultano le più significative nella descrizione della dinamica longitudinale veloce del velivolo. In presenza di un disturbo longitudinale, il modo di short period si manifesta infatti come un'oscillazione smorzata in beccheggio, nella quale α e l'angolo di assetto θ oscillano approssimativamente in fase, mentre la velocità angolare di beccheggio q risulta in anticipo di fase rispetto a tali grandezze.

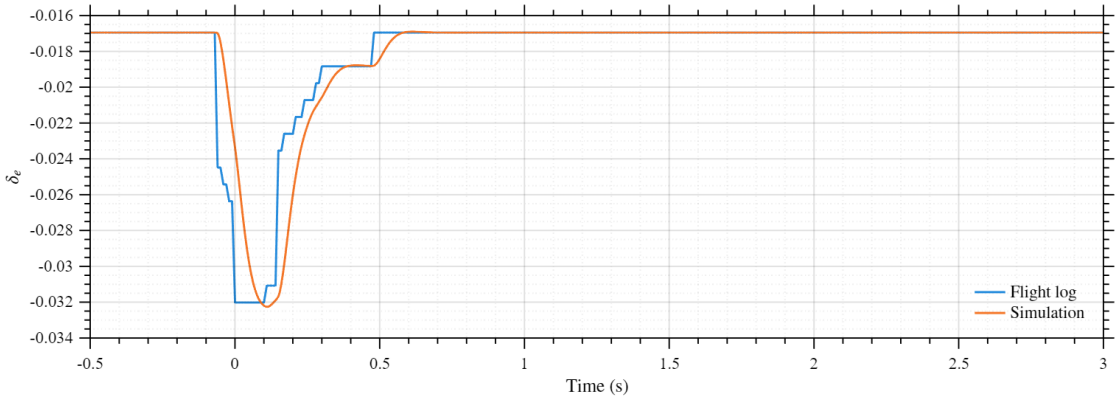


Figura 4.25: Comando di tipo step di elevator

In Tabella 4.1 e in Figura 4.26 sono riportate le caratteristiche della risposta dinamica e l'andamento temporale della variabile q, sia nel caso del volo reale sia in quello simulato. Un'analoga analisi è stata condotta per la variabile α, i cui risultati sono presentati in Tabella 4.2 e in Figura 4.27.
In entrambi i casi la risposta è caratterizzata da un'oscillazione a frequenza relativamente elevata, con valori coerenti con quelli tipici del modo di short period, e da un rapido decadimento dell'ampiezza, con estinzione significativa entro circa 1–1.5 s. È inoltre possibile osservare che la risposta della velocità angolare di beccheggio q risulta in anticipo rispetto a quella dell'angolo d'attacco α: quest'ultimo attraversa infatti il valore nullo in prossimità dell'istante in cui q raggiunge il proprio valore massimo, comportamento coerente con la relazione di fase tipica di un moto oscillatorio smorzato.
Il confronto tra dati di volo e dati simulati evidenzia una buona coerenza nella stima della frequenza naturale: il primo picco e la fase iniziale dell'oscillazione risultano infatti correttamente riprodotti dal modello. Per quanto riguarda lo smorzamento, si osserva invece una lieve discrepanza: nel caso simulato la dinamica appare più regolare e il decadimento più rapido, indicando uno smorzamento leggermente maggiore rispetto a quello misurato in volo. Nei dati sperimentali, dopo il primo secondo, permangono piccole oscillazioni residue e componenti ad alta frequenza, attribuibili al rumore di misura.
Una possibile causa della differenza nello smorzamento può essere ricondotta a una non perfetta modellazione del contributo del piano di coda nella dinamica longitudinale. Quest'ultimo rappresenta infatti il principale responsabile dello smorzamento del modo

di short period, poiché tende a riallinearsi al flusso incidente generando un momento stabilizzante proporzionale alla velocità angolare di beccheggio.
È infine opportuno sottolineare che il fit della risposta mediante un'equazione modello del secondo ordine introduce inevitabilmente un errore nella stima di frequenza e smorzamento; tali parametri devono quindi essere interpretati principalmente come indicatori per un confronto qualitativo tra volo reale e simulazione. Nel complesso, il modello dinamico implementato risulta comunque in grado di riprodurre in modo soddisfacente le caratteristiche principali del modo di short period, catturandone correttamente la frequenza e fornendo una stima dello smorzamento coerente con l'andamento iniziale della risposta.

Analisi	**Frequenza f (Hz)**	**Smorzamento ζ**
Volo sperimentale	1.3957	0.2633
Simulazione	1.6013	0.3831

Tabella 4.1: Confronto della risposta di q nella dinamica di short period

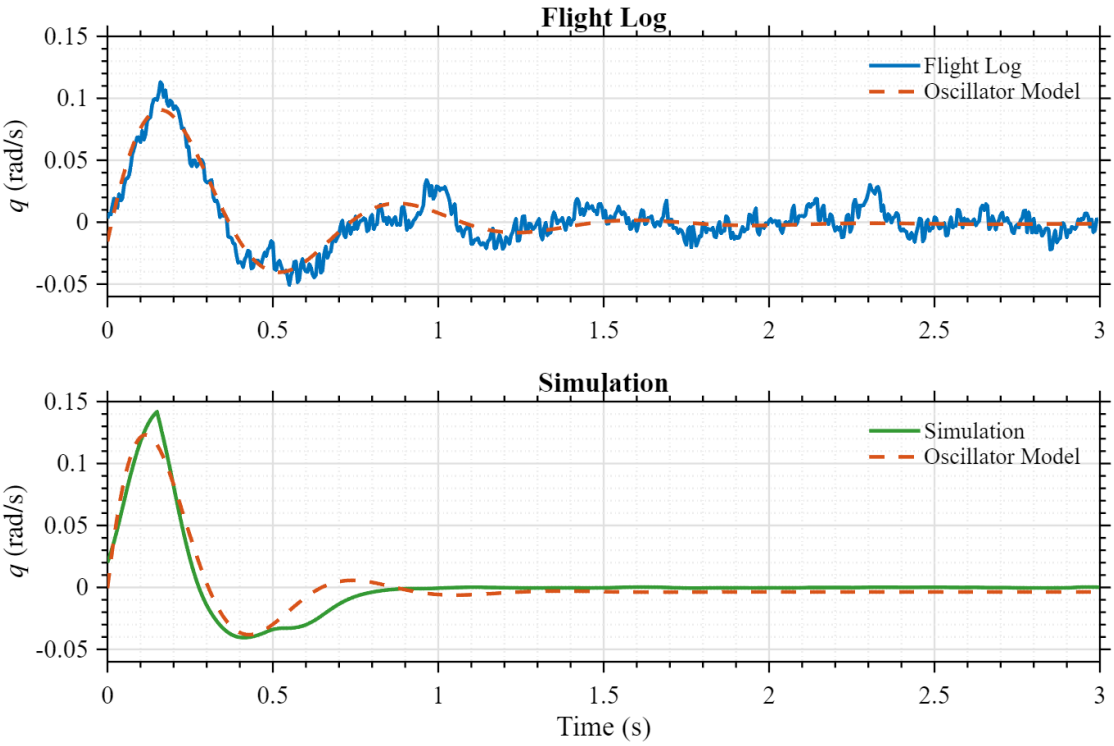


Figura 4.26: Confronto della risposta di q al comando di tipo step di elevator

Allo scopo di evidenziare in modo più chiaro le differenze tra la dinamica simulata e quella misurata in volo reale, è stata condotta un'analisi in dominio tempo–frequenza

Analisi	**Frequenza f (Hz)**	**Smorzamento ζ**
Volo sperimentale	1.2406	0.3213
Simulazione	1.3413	0.3580

Tabella 4.2: Confronto della risposta di α nella dinamica di short period

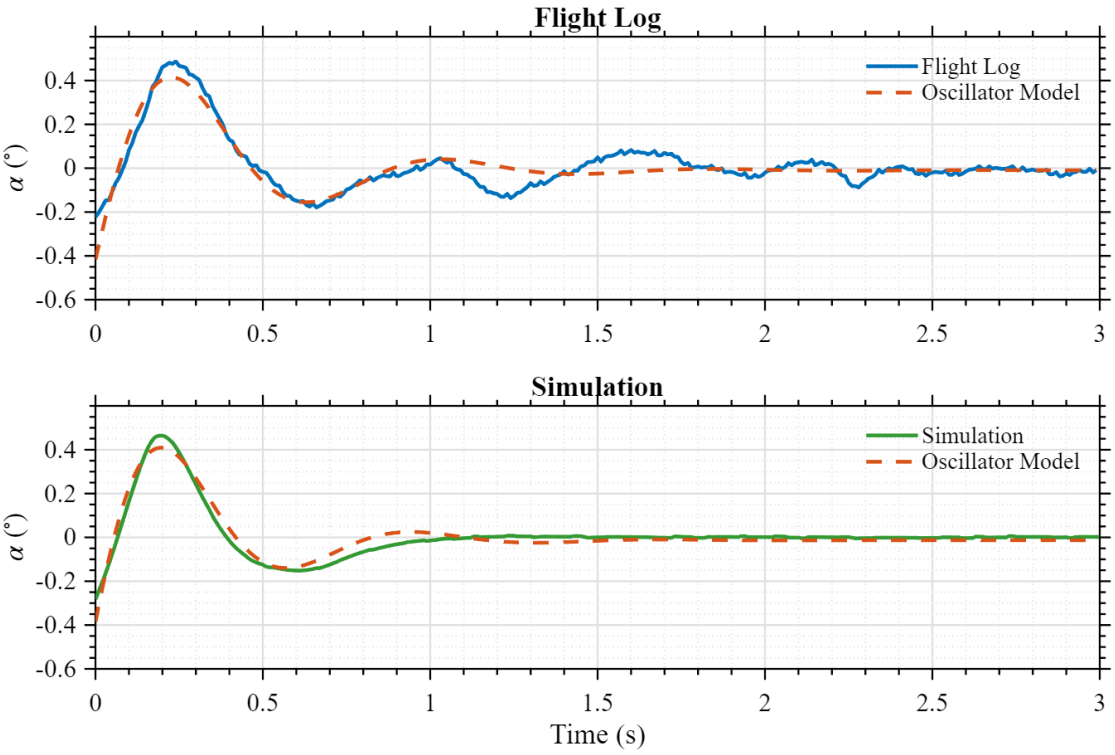


Figura 4.27: Confronto della risposta di α al comando di tipo step di elevator

mediante trasformata di Fourier a finestra mobile, valutando la *power spectral density* (PSD) in scala logaritmica. Tale rappresentazione consente di individuare, all'interno della finestra temporale considerata, i principali contributi energetici associati alle diverse frequenze che caratterizzano la risposta longitudinale del velivolo.
In Figura 4.28 è riportato il risultato dell'analisi per il volo sperimentale. Per ciascun istante temporale è stata calcolata la distribuzione spettrale di potenza su un intervallo di frequenze; una colorazione tendente al rosso indica un maggiore contenuto energetico a quella specifica frequenza, mentre una colorazione blu corrisponde a un contributo ridotto. La medesima analisi è stata eseguita sulla risposta simulata del modello dinamico, il cui risultato è mostrato in Figura 4.29.
Dal confronto tra i due grafici si osserva come la simulazione riesca a riprodurre correttamente, negli istanti iniziali, la frequenza dominante associata al modo di *short period*, evidenziando una distribuzione energetica concentrata nella medesima banda di frequenza presente nel volo reale. Tuttavia, la risposta simulata risulta più "pulita" e maggiormente concentrata attorno alla frequenza principale, mentre nel caso sperimentale si nota una distribuzione energetica più ampia, indice della sovrapposizione di più contributi dinamici e di una risposta complessivamente più complessa.
Nel grafico relativo al volo reale si distingue inoltre una banda energetica persistente attorno alla frequenza di circa 15 Hz. Tale contributo non è presente nella simulazione e può essere attribuito al rumore dei sensori, a vibrazioni strutturali del velivolo o a fenomeni aeroelastici non modellati. L'assenza di tale componente nella risposta simulata conferma che il modello dinamico, pur riproducendo correttamente la dinamica modale principale, non include le sorgenti di eccitazione ad alta frequenza tipiche del sistema reale.
Nel complesso, l'analisi spettrale conferma quanto già osservato nel dominio del tempo: il modello è in grado di catturare le caratteristiche fondamentali del modo dominante, ma tende a sottostimare i contributi ad alta frequenza presenti nel volo reale, legati a fenomeni secondari e non lineari non inclusi nella modellazione.

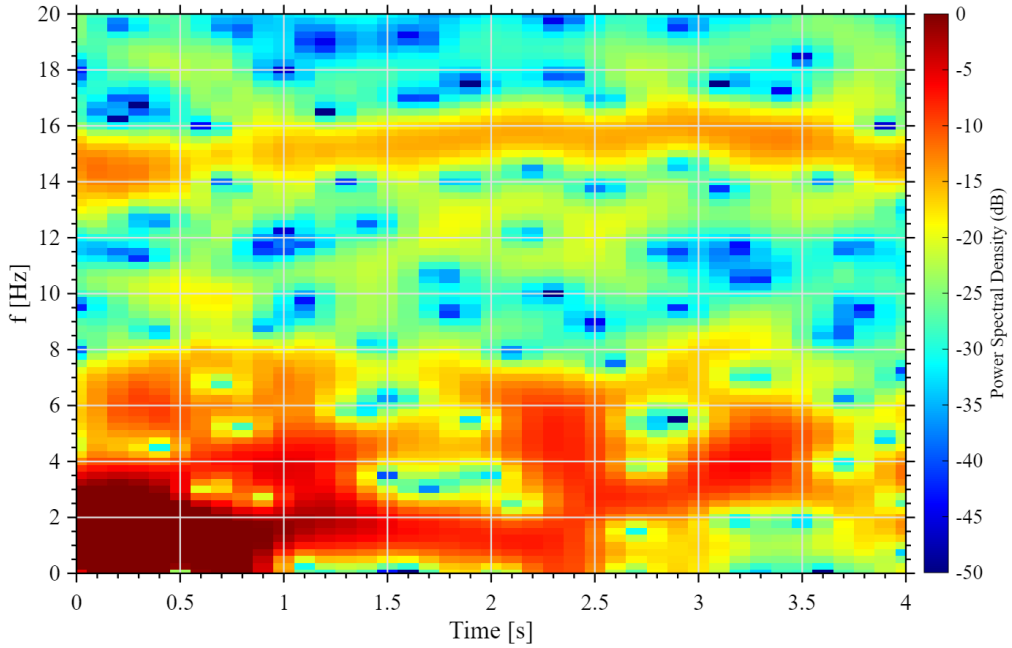


Figura 4.28: PSD della risposta di q misurata sperimentalmente

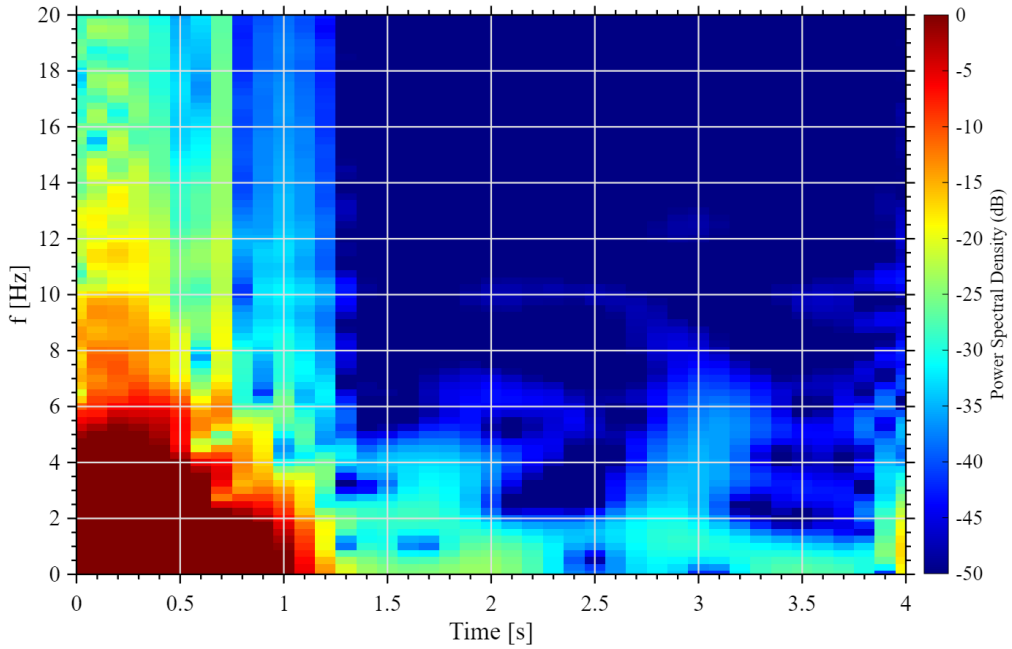


Figura 4.29: PSD della risposta di q misurata in simulazione

4.4.2 Dutch Roll

Per identificare il modo dinamico di dutch roll sono state analizzate le risposte delle variabili r e β al comando di tipo step imposto al timone, rappresentato in figura 4.30. Il modo di dutch roll è l'equivalente al modo di short period per la dinamica longitudinale, in presenza di un disturbo si forma infatti un oscillazione smorzata in imbardata accoppiata al rollio.

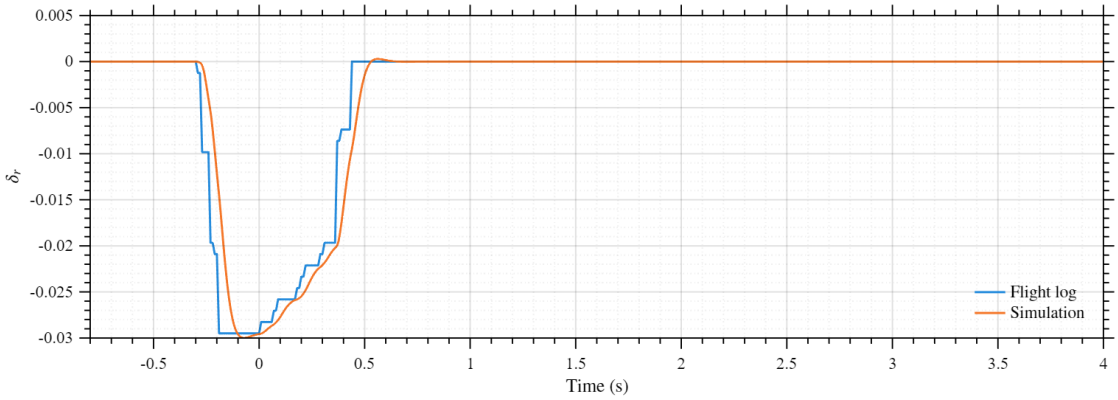


Figura 4.30: Comando di tipo step imposto al timone

In Tabella 4.3 e in Figura 4.31 sono riportate le caratteristiche della risposta dinamica e l'andamento temporale della variabile r, sia per il volo reale sia per la simulazione. Un'analoga analisi è stata condotta per la variabile β, i cui risultati sono presentati in Tabella 4.4 e in Figura 4.32.

In entrambi i casi la risposta è caratterizzata da una frequenza di circa 1 Hz, valore tipico del modo di dutch roll, e da uno smorzamento significativo, sebbene leggermente inferiore rispetto a quello che contraddistingue il modo di short period. Per quanto riguarda la dinamica latero-direzionale, si osserva tuttavia una certa discrepanza tra i modi estratti dai dati di volo e quelli ottenuti dal modello dinamico.

Il modello riesce a riprodurre in modo adeguato l'ampiezza iniziale della perturbazione, ma presenta successivamente una dinamica più smorzata e leggermente più lenta rispetto a quella reale. Tali differenze sono verosimilmente riconducibili a una non perfetta modellazione dei coefficienti aerodinamici latero-direzionali, criticità già emersa nel confronto tra le variabili di stato discusso nella sezione precedente. In particolare, si riscontra una tendenza del modello a sovrastimare lo smorzamento associato al modo, con conseguente attenuazione più rapida dell'oscillazione.

È inoltre opportuno ricordare che la variabile β non è direttamente misurata durante le prove di volo, ma ricavata tramite elaborazione dei dati disponibili; ciò introduce un'ulteriore fonte di incertezza nella valutazione del fit tra modello e realtà.

Nel complesso, il modello dinamico implementato risulta comunque in grado di riprodurre in maniera soddisfacente le caratteristiche principali del modo di dutch roll,

cogliendone correttamente l'andamento qualitativo della risposta a una perturbazione latero-direzionale, pur con alcune limitazioni nella stima dello smorzamento.

Analisi	**Frequenza f (Hz)**	**Smorzamento ζ**
Volo sperimentale	0.9751	0.1627
Simulazione	0.6842	0.3180

Tabella 4.3: Confronto della risposta di r nella dinamica di dutch roll

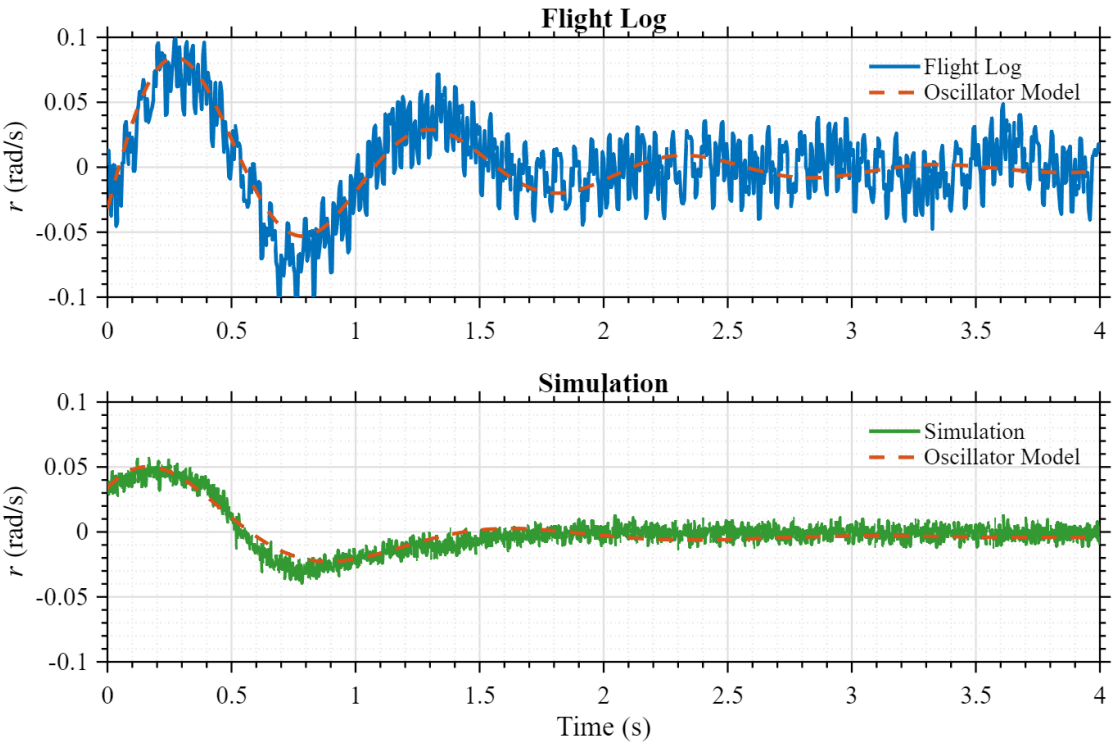


Figura 4.31: Confronto della risposta di r al comando di tipo step del timone

Analisi	**Frequenza f (Hz)**	**Smorzamento ζ**
Volo sperimentale	0.8083	0.1672
Simulazione	0.3583	0.6100

Tabella 4.4: Confronto della risposta di β nella dinamica di dutch roll

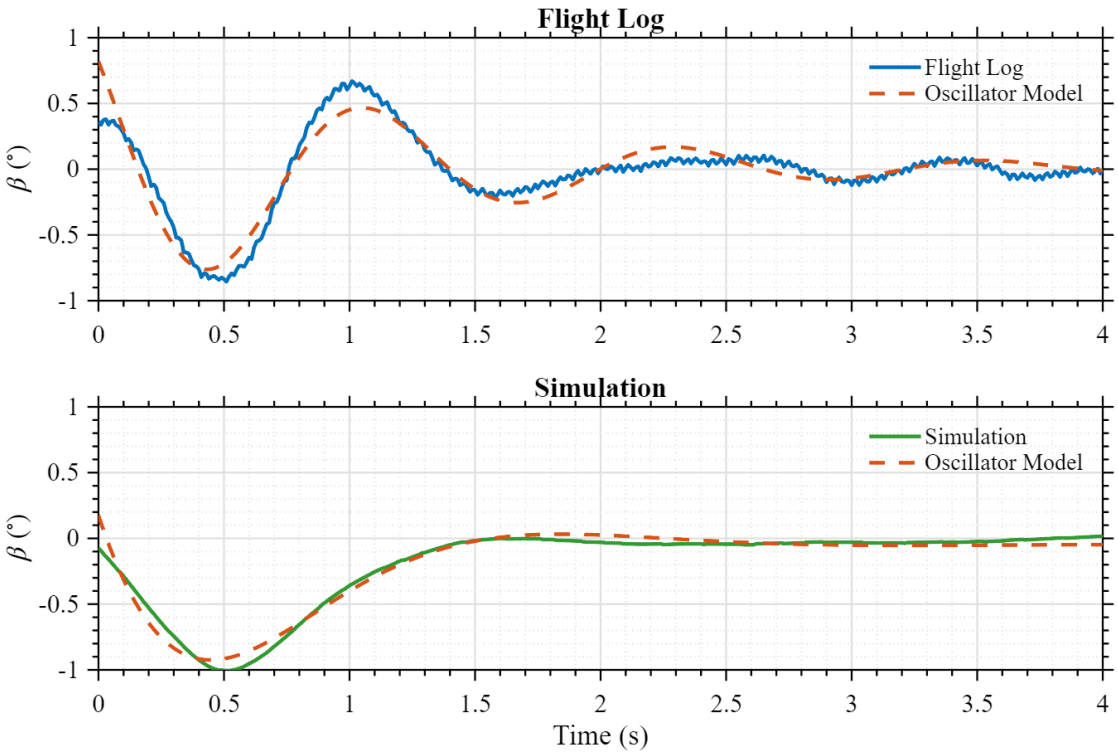


Figura 4.32: Confronto della risposta di β al comando di tipo step del timone

Analogamente a quanto svolto per il modo di short period, è stata effettuata un'analisi nel dominio tempo–frequenza mediante la valutazione della power spectral density (PSD) in scala logaritmica. Tale rappresentazione consente di evidenziare i principali contributi energetici associati alle diverse componenti in frequenza che caratterizzano la risposta latero-direzionale del velivolo.
Dal confronto tra i grafici riportati in Figura 4.33 e 4.34 si osserva come la simulazione sia in grado di riprodurre correttamente la frequenza dominante associata al modo dinamico, confermando quanto emerso dall'analisi nel dominio del tempo. Inoltre, a differenza di quanto riscontrato per la dinamica di short period, il modello riesce a cogliere anche componenti a frequenza più elevata, non immediatamente distinguibili dall'osservazione diretta dell'andamento temporale della variabile.
Permane tuttavia, nei dati di volo reale, una distribuzione di energia concentrata attorno ai 15 Hz, verosimilmente riconducibile alla banda caratteristica del rumore dei sensori o della catena di acquisizione. Tale contributo non è presente nella simulazione, in quanto il modello dinamico utilizzato in questa fase non include una rappresentazione del rumore di misura.

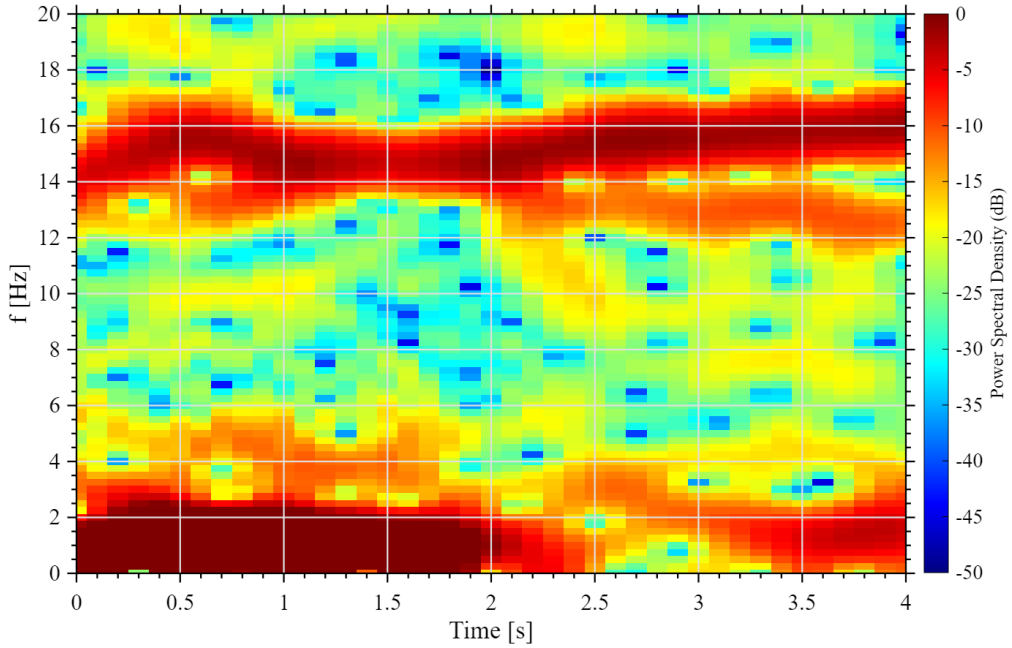


Figura 4.33: PSD della risposta di r misurata sperimentalmente

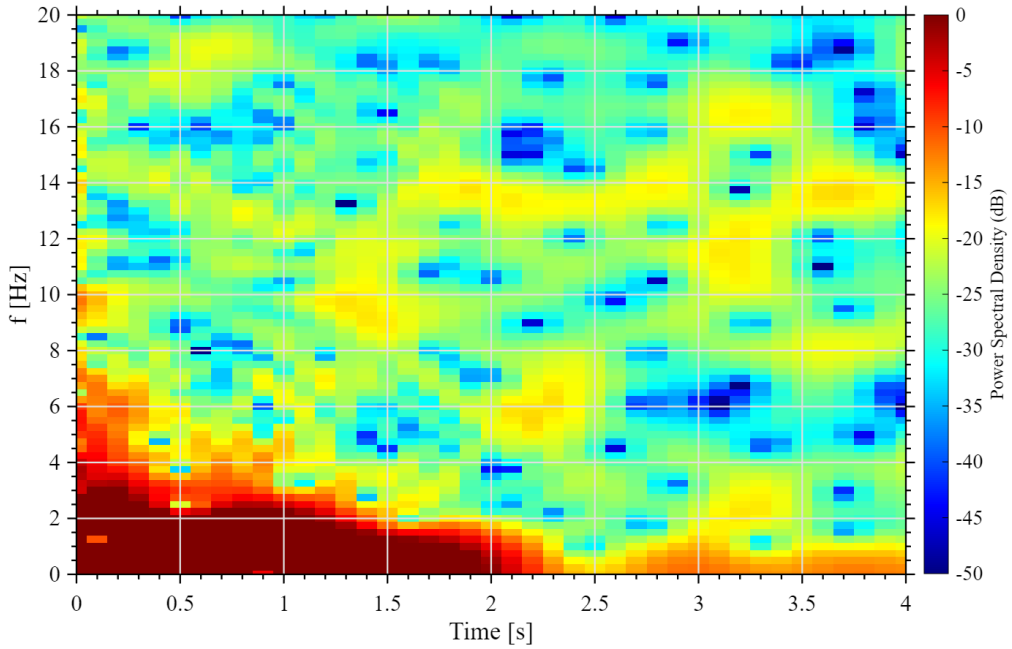


Figura 4.34: PSD della risposta di r misurata in simulazione

Capitolo 5

Taratura del controllore di volo

Il sistema di controllo PX4 si basa su una struttura a controllori PID annidati (rate, attitude, velocity e position), incaricati di calcolare le azioni di controllo necessarie per guidare il velivolo dallo stato stimato corrente al riferimento desiderato, imposto dal pilota o dall'autopilota. La corretta taratura di tali controllori riveste un ruolo fondamentale per garantire prestazioni elevate e un comportamento dinamico stabile del veicolo. In particolare, una calibrazione non adeguata del controllore di rate si riflette in una ridotta stabilità del volo in tutte le modalità operative e in una maggiore sensibilità ai disturbi, con tempi di recupero più lunghi a seguito di perturbazioni esterne.
In questo capitolo verrà quindi implementato un processo di taratura del controllore di rate applicato al velivolo simulato mediante il modello dinamico, attraverso simulazioni Hardware-In-The-Loop (HIL). I risultati ottenuti saranno infine confrontati con quelli derivanti dalla taratura effettuata in volo sul velivolo reale.

5.1 Controllo PID

Il controllo proporzionale-integrale-derivativo, in breve controllo PID, è un sistema di controllo in retroazione ampiamente impiegato nei sistemi di controllo automatico nell'industria aerospaziale. Un controllore PID si basa su un semplice funzionamento: il controllore di volo acquisisce in ingresso le variabili di stato stimate del velivolo (ad esempio angoli di assetto, velocità angolari o quota) e le confronta con i corrispondenti valori di riferimento imposti dal pilota o dai loop di guida. La differenza tra riferimento e stato misurato, definita segnale di errore, viene quindi elaborata dall'algoritmo di controllo per generare i comandi alle superfici mobili o al sistema propulsivo. Tali comandi rappresentano le variabili manipolate del sistema e determinano l'evoluzione dinamica del velivolo verso la condizione desiderata. [28]
Le variabili in uscita sono determinate attraverso diversi algoritmi:

- **Azione Proporzionale (P)**: L'azione proporzionale produce un contributo direttamente proporzionale al segnale di errore:

$$u_P(t) = K_P\, e(t)$$

dove K_P è il guadagno proporzionale. Un aumento di K_P rende il sistema più reattivo, riducendo generalmente l'errore a regime, ma può comportare un incremento dell'overshoot e una possibile riduzione dei margini di stabilità.

- **Azione Integrativa (I)**: L'azione integrale tiene conto della storia passata dell'errore ed è definita come:

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

dove K_I è il guadagno integrale. Essa introduce una memoria del sistema e consente di eliminare l'errore a regime (offset). In presenza di un errore persistente, l'integrale cresce nel tempo aumentando il segnale di controllo fino a compensare completamente l'errore. Tuttavia, un eccessivo contributo integrale può generare fenomeni di overshoot e rallentare la risposta dinamica.

- **Azione Derivativa (D)**: L'azione derivativa è proporzionale alla derivata temporale dell'errore:

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

dove K_D è il guadagno derivativo. Questo termine agisce in modo predittivo, opponendosi alle rapide variazioni dell'errore e contribuendo ad aumentare lo smorzamento del sistema. Sebbene raramente utilizzata singolarmente, l'azione derivativa migliora la stabilità e riduce l'overshoot se opportunamente tarata.

Queste tre azioni vengono infine sommate algebricamente per formare il segnale di controllo:

$$u_{PID}(t) = u_P(t) + u_I(t) + u_D(t) = K_P \cdot e(t) + K_I \cdot \int_0^t e(\tau)\, d\tau + K_D \cdot \frac{de(t)}{dt} \tag{5.1}$$

Esso può anche essere espresso come funzione di trasferimento applicando la trasformazione di Laplace:

$$G(s) = \frac{U(s)}{E(s)} = K_P \frac{1 + T_i s + T_i T_d s^2}{T_i s} \tag{5.2}$$

$$T_i = \frac{K_P}{K_I} \tag{5.3}$$

$$T_d = \frac{K_D}{K_P} \tag{5.4}$$

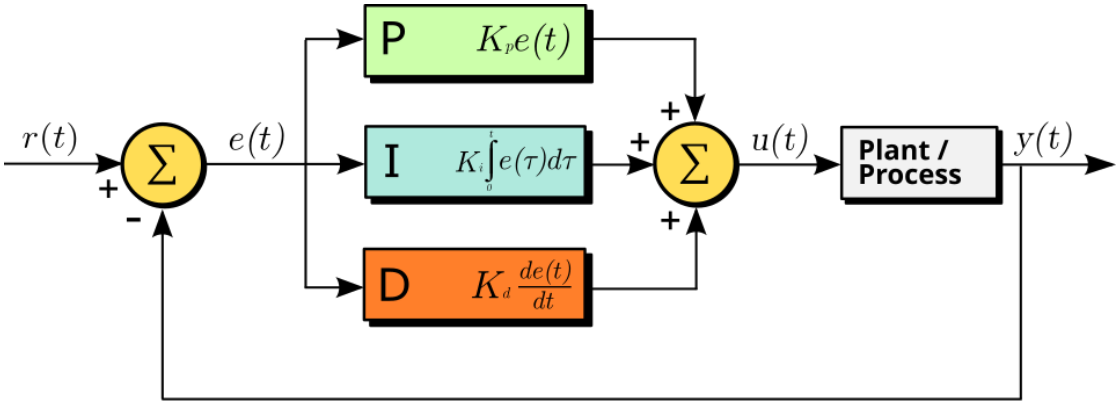


Figura 5.1: Diagramma del controllore PID

Nel contesto del controllo automatico di UAV, il controllore PID è frequentemente integrato con un termine di *feed-forward*. A differenza delle azioni in retroazione, che operano sul segnale di errore, il contributo di feed-forward agisce direttamente sul riferimento $r(t)$, senza passare dal calcolo dell'errore $e(t) = r(t) - y(t)$.
L'introduzione di tale termine consente di migliorare la rapidità di inseguimento del riferimento e di ridurre l'errore transitorio, anticipando la risposta del sistema sulla base del comando desiderato. In questo modo si alleggerisce il contributo delle azioni proporzionale e integrale, migliorando le prestazioni dinamiche complessive.
Il segnale di controllo finale risulta pertanto:

$$u(t) = K_{FF}\, r(t) + u_{PID}(t) \tag{5.5}$$

Ognuna delle quattro azioni di controllo (P, I, D e feed-forward) contribuisce in modo differente alla definizione della risposta dinamica del sistema. L'obiettivo complessivo è garantire che le variabili controllate seguano il riferimento imposto durante una manovra o mantengano una condizione di equilibrio stabile in presenza di disturbi.
La qualità dell'inseguimento può essere valutata analizzando alcune caratteristiche fondamentali della risposta al gradino di una determinata variabile:

- **Overshoot**: massimo superamento percentuale del valore di riferimento durante il transitorio. Indica la tendenza del sistema a sovraelongare prima di stabilizzarsi.
- **Rise Time**: tempo necessario affinché la risposta raggiunga un valore in prossimità del riferimento (tipicamente 90%). Misura la rapidità iniziale del sistema.
- **Steady-State Error**: errore residuo tra riferimento e uscita a regime permanente. Valuta l'accuratezza statica del sistema.
- **Settling Time**: tempo necessario affinché la risposta rimanga stabilmente entro una determinata banda di tolleranza attorno al valore finale.

Mediante uno studio comparativo è possibile valutare come la variazione dei parametri del controllore influenzi tali caratteristiche. La Tabella 5.1 riassume qualitativamente

l'effetto di un incremento dei singoli guadagni sulle principali proprietà della risposta dinamica [29].
Risulta pertanto evidente che l'ottenimento di una risposta conforme alle specifiche richiede un opportuno processo di taratura, i cui parametri risultano strettamente dipendenti dalle caratteristiche dinamiche del sistema in esame.

Parametro	**Overshoot**	**Rise Time**	**Steady-State Error**	**Settling Time**
K_P	$\uparrow$	$\downarrow$	$\downarrow$	$\sim$
K_I	$\uparrow$	$\downarrow$	$\downarrow$	$\uparrow$
K_D	$\downarrow$	$\sim$	$\sim$	$\downarrow$
K_{FF}	$\sim$	$\downarrow$	$\sim$	$\downarrow$

Tabella 5.1: Effetto qualitativo di un incremento dei parametri del controllore sulle caratteristiche della risposta

5.2 Metodo Ziegler-Nichols

La definizione dei parametri del controllore PX4 adottata nel presente studio è basata sulla procedura di taratura proposta negli anni '40 da John G. Ziegler e Nathaniel B. Nichols [30].
La taratura di Ziegler–Nichols rappresenta uno dei metodi empirici più noti e diffusi per la determinazione dei parametri di un controllore PID. L'obiettivo del lavoro originale era fornire una procedura semplice, ripetibile e facilmente applicabile in ambito industriale, evitando la complessità legata all'analisi matematica completa delle equazioni differenziali che descrivono il comportamento del sistema in retroazione.
Nel loro studio, gli autori propongono due distinti approcci di taratura. Il primo si basa sull'analisi del comportamento del sistema in *closed-loop* in condizioni di oscillazione limite ed è la metodologia adottata nel presente progetto. Il secondo approccio è invece fondato sulla caratterizzazione dinamica del processo mediante l'analisi della cosiddetta *curva di reazione*, che verrà brevemente richiamata per completezza. In entrambi i casi, le procedure conducono a regole pratiche per la determinazione dei parametri associati alle tre azioni di controllo fondamentali: proporzionale, integrale e derivativa.

5.2.1 Regola di Ziegler-Nichols in closed-loop

Ziegler e Nichols individuano tre principali effetti di controllo presenti nei regolatori industriali, ciascuno dei quali è direttamente riconducibile a un parametro del controllore PID. In particolare, la risposta proporzionale è associata al termine K_P, la risposta di *reset automatico* rappresenta l'azione integrativa, corrispondente al parametro $T_i = K_P/K_i$, mentre la risposta di *pre-act* è equivalente all'azione derivativa ed è quindi associata al termine $T_d = K_D/K_P$.
La regolazione dell'azione proporzionale, che nel lavoro originale viene quantificata tramite il concetto di *sensibilità*, è fondamentale per il raggiungimento di un comportamento stabile del sistema controllato. Tuttavia, un'azione proporzionale eccessivamente elevata conduce inevitabilmente a instabilità, con una risposta caratterizzata da oscillazioni persistenti. Esiste infatti un valore critico del guadagno proporzionale, indicato con K_{cr}, facilmente determinabile sperimentalmente, al di sopra del quale qualsiasi perturbazione genera oscillazioni a ampiezza crescente. Al di sotto di tale valore, invece, un'oscillazione di qualunque ampiezza tende progressivamente a smorzarsi, conducendo a un comportamento stabile.
Impostando il guadagno proporzionale pari al valore critico, $K_P = K_{cr}$, il sistema manifesta una risposta oscillatoria a ampiezza costante, nota come *ciclo limite*. Il periodo di tali oscillazioni è definito *periodo ultimo* P_u. Nel caso di un controllore puramente proporzionale, è stato osservato che un'impostazione pari a $K_P = 0.5K_{cr}$ consente di ottenere un buon compromesso tra stabilità e rapidità di risposta, con un rapporto di ampiezza (definito come il rapporto tra l'ampiezza di un'oscillazione e quella della precedente) di circa il 25 %.
L'azione integrativa, o *reset automatico*, viene introdotta per eliminare l'errore a regime (*offset*), inevitabile nel controllo puramente proporzionale. La grandezza caratteristica associata a tale azione è il *reset rate* T_i, espressa in min^{-1}, definita come il numero di

volte al minuto in cui l'azione integrale replica l'effetto iniziale dell'azione proporzionale. È stato osservato che il periodo ultimo P_u costituisce un buon indice per la determinazione della velocità di reset ottimale. Tuttavia, sebbene l'azione integrativa consenta di annullare l'offset, essa introduce anche una riduzione del margine di stabilità del sistema. Un valore di compromesso comunemente adottato per il tempo integrale risulta quindi pari a $T_i = 1.2/P_u$.

L'azione di *pre-act* corrisponde all'attuale azione derivativa e genera un contributo di controllo proporzionale alla velocità di variazione della variabile misurata. Essa viene impiegata esclusivamente in combinazione con le altre azioni di controllo e non garantisce sempre un miglioramento della risposta del sistema. Tuttavia, se correttamente tarata, può condurre a un incremento della stabilità e a una riduzione del *settling time*. La grandezza rappresentativa è il *pre-act time* T_d, espresso in minuti, che rappresenta il tempo equivalente con cui l'uscita del controllore anticipa la variazione dell'errore. È stato osservato che, per un'ampia gamma di applicazioni, il valore ottimale del tempo di pre-azione dipende direttamente dal periodo di oscillazione ultimo. In particolare, una scelta efficace consiste nell'impostare $T_d = P_u/8$

La presenza dell'azione derivativa comporta una lieve modifica dei valori di riferimento dei parametri K_P e T_i. In Tabella 5.2 sono riassunte le regole di taratura proposte da Ziegler e Nichols per i diversi tipi di controllore [31].

PID Type	K_P	T_i	T_d
P	0.5 K_{cr}	∞	0
PI	0.45 K_{cr}	$1.2/P_u$	0
PID	0.6 K_{cr}	$2/P_u$	$P_u/8$

Tabella 5.2: Metodo Ziegler-Nichols in closed-loop

5.2.2 Regola di Ziegler-Nichols in open-loop

Il secondo approccio rappresenta invece una procedura predittiva basata sulla curva di reazione del processo.
Esso viene applicato scollegando il controllore, e valutando quindi la risposta open-loop ad un comando step sulla variabile di controllo. La risposta definisce due parametri fondamentali:

- **Unit Reaction Rate (R)**: pendenza massima della curva normalizzata rispetto a uno step unitario. La pendenza massima, e quindi la massima velocità di reazione, si verifica nel punto di flesso della curva.
- **Lag (L)**: tempo di ritardo equivalente, ottenuto prolungando la tangente nel punto di massima pendenza fino all'asse temporale.

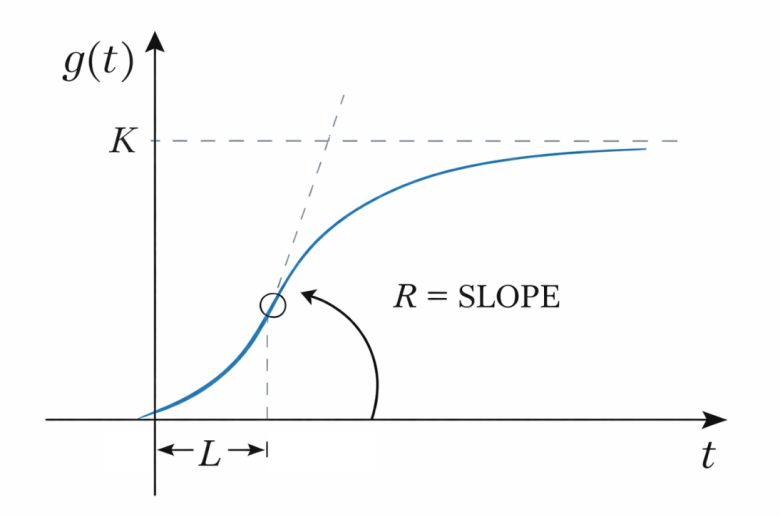


Figura 5.2: Curva di reazione per il metodo di Ziegler-Nichols

Il prodotto RL rappresenta un parametro chiave nella progettazione del controllore. L'impostazione ottimale del parametro proporzionale del controllore è infatti inversamente proporzionale al prodotto RL, mentre i parametri integrativi e derivativi T_i e T_d possono direttamente essere determinati in funzione del tempo di ritardo L, come riassunto in tabella 5.3.

PID Type	K_P	T_i	T_d
P	$1/RL$	∞	0
PI	$0.9/RL$	$0.3/L$	0
PID	$1.2/RL$	$0.5/L$	$0.5L$

Tabella 5.3: Metodo Ziegler-Nichols in open-loop

5.3 Taratura tramite simulazione HIL

L'effettiva taratura del controllore PX4 è stata condotta mediante simulazione *Hardware-in-the-Loop* (HIL), facendo uso degli strumenti descritti nel Capitolo 2 e del modello dinamico sviluppato nel Capitolo 3.
Questo approccio consente di testare le qualità di volo del velivolo, mantenendo però l'ambiente di prova in condizioni completamente simulate e quindi controllabili. In tal modo è possibile analizzare il comportamento del sistema al variare dei parametri del controllore, osservandone gli effetti sulle principali variabili dinamiche senza esporre il velivolo a potenziali condizioni critiche.
Un processo di taratura di questo tipo presenta numerosi vantaggi. In primo luogo, riduce significativamente i rischi associati alla regolazione diretta in volo, specialmente nelle fasi iniziali in cui i parametri del controllore possono risultare lontani dalla configurazione ottimale e generare instabilità o oscillazioni indesiderate. In secondo luogo, permette di ottenere una prima stima coerente dei guadagni dei regolatori, limitando il numero di prove sperimentali necessarie e ottimizzando tempi e costi di sviluppo.
Infine, la metodologia HIL rende possibile avviare il processo di taratura anche in assenza del velivolo completo o prima della sua disponibilità operativa, anticipando la fase di integrazione e validazione del sistema di controllo. I parametri così ottenuti costituiscono una base di partenza solida per la successiva rifinitura sperimentale, riducendo l'entità delle correzioni necessarie e aumentando il livello complessivo di sicurezza del processo di sviluppo.

L'architettura del processo di simulazione e la gestione del flusso dei dati sono state descritte in modo dettagliato nel Capitolo 1. Di seguito si riporta un elenco sintetico degli step operativi eseguiti in questo lavoro per l'esecuzione della taratura del controllore PID in ambiente HIL.

- **Joystick Setup**: La modalità di volo manuale può essere gestita tramite un joystick collegato al computer host. In questa configurazione, QGroundControl acquisisce gli input del joystick e li codifica in messaggi MAVLink, successivamente inviati al modello Simulink tramite connessione UDP. Per abilitare il supporto joystick in PX4 è necessario impostare il parametro `COM_RC_IN_MODE` su *"1-Joystick"* e collegare il dispositivo via USB. Segue la fase di calibrazione attraverso la sezione QGroundControl > Vehicle Setup, che consente di associare ciascun comando del velivolo (throttle, roll, pitch, yaw) alle rispettive leve del joystick. È inoltre possibile configurare specifici pulsanti per l'attivazione rapida di *flight modes* o funzioni dedicate.
- **Avvio della simulazione**: Dopo aver verificato la corretta connessione fisica tra flight controller e computer host, è necessario eseguire lo script MATLAB di inizializzazione, nel quale vengono caricati nel *workspace* i parametri e i dati di simulazione riepilogati in Sezione 3.5. Successivamente, nei blocchi `"MAVLink Bridge Source"` e `"MAVLink Bridge Sink"` del modello Simulink, deve essere impostata la porta seriale corretta (*"/dev/ttyACM0"*) per consentire la comunicazione con QGroundControl. Una volta verificata la connessione, la simulazione può

essere avviata assicurandosi che sia configurata in modalità *paced time*, ovvero con esecuzione in tempo reale, requisito fondamentale per la corretta interazione con l'hardware.

- **Decollo in modalità manuale**: All'avvio della simulazione dinamica viene attivata anche la comunicazione tra hardware e QGroundControl, che fornisce l'interfaccia grafica per la gestione del volo simulato. È quindi possibile armare il velivolo e procedere al decollo in modalità manuale. Durante questa fase si utilizzano la *Flight Telemetry* e l'indicatore di assetto per monitorare l'atteggiamento del velivolo. Raggiunta una quota di sicurezza, si stabilizza il volo orizzontale mantenendo un livello di throttle pari a circa il 60%, così da operare in prossimità della condizione di trim e predisporre il sistema alla successiva fase di taratura.
- **Procedura di Taratura**: La regolazione dei guadagni dei controllori di *rate* segue la metodologia di Ziegler–Nichols descritta in Sezione 5.2. la procedura di tuning può essere eseguita in modo agevole all'interno di QGroundControl accedendo al menu Vehicle Setup > PID Tuning. All'interno di questa sezione è necessario selezionare il *Rate Controller* tab per procedere alla taratura dell'anello interno di velocità angolare. Per l'esecuzione è necessario impostare il velivolo in modalità *Acro*, in modo da attivare il controllore e far sì che lo stick dei comandi controlli direttamente le velocità angolari. La regolazione dei parametri avviene separatamente per ciascun asse, pertanto è necessario selezionare di volta in volta l'asse di interesse e modificarne i relativi guadagni. Tale approccio consente di analizzare e ottimizzare in maniera indipendente la risposta dinamica lungo ciascun asse, semplificando il processo di identificazione dei parametri più appropriati. Gli step operativi della procedura eseguita sono riassunti nelle Tabelle 5.4 e 5.5.

Step	**Impostazione Iniziale**	**Esecuzione**
$N°$ 1	Impostare i parametri `FW_RR_P`, `FW_RR_I` e `FW_RR_D` a zero	Posizionare la manetta al trim (circa 60%) e mantenere la quota
$N°$ 2	Impostare il parametro di feed-forward `FW_RR_FF`= 0.4	Eseguire colpetti di rollio destra-sinistra e verificare la risposta ottenuta per il *roll rate*. Raddoppiare il valore fino ad ottenere una buona risposta, poi ridurre il valore finale del 20%
$N°$ 3	Impostare il parametro `FW_RR_P`= 0.006	Eseguire colpetti di rollio destra-sinistra e verificare la risposta ottenuta per il *roll rate*. Raddoppiare il valore fino a che non si genera il ciclo limite nella risposta, poi ridurre il valore finale del 50%
$N°$ 4	Impostare il parametro `FW_RR_I`= 0.001	Eseguire colpetti di rollio destra-sinistra e verificare la risposta ottenuta per il *roll rate*. Raddoppiare il valore fino ad ottenere una buona riduzione dello steady-state error, mantenere il valore finale ottenuto
$N°$ 5	Impostare il parametro `FW_RR_D`= 0.005	Eseguire colpetti di rollio destra-sinistra e verificare la risposta ottenuta per il *roll rate*. Verificare che la risposta sia effettivamente migliore di quella ottenuta in assenza del parametro derivativo, raddoppiare il valore fino a che si ottine un miglioramento della risposta, poi ridurre il valore finale del 50%

Tabella 5.4: Procedura per la taratura dei parametri di rollio

Step	**Impostazione Iniziale**	**Esecuzione**
$N°$ 1	Impostare i parametri `FW_PR_P`, `FW_PR_I` e `FW_PR_D` a zero	Posizionare la manetta al trim (circa 60%) e mantenere la quota
$N°$ 2	Impostare il parametro di feed-forward `FW_PR_FF`= 0.01	Effettuare colpetti di cabrata/picchiata e verificare la risposta ottenuta per il *pitch rate*. Raddoppiare il valore fino ad ottenere una buona risposta, poi ridurre il valore finale del 20%
$N°$ 3	Impostare il parametro `FW_PR_P`= 0.06	Effettuare colpetti di cabrata/picchiata e verificare la risposta ottenuta per il *pitch rate*. Raddoppiare il valore fino a che non si genera il ciclo limite nella risposta, poi ridurre il valore finale del 50%
$N°$ 4	Impostare il parametro `FW_PR_I`= 0.01	Effettuare colpetti di cabrata/picchiata e verificare la risposta ottenuta per il *pitch rate*. Raddoppiare il valore fino ad ottenere una buona riduzione dello steady-state error, mantenere il valore finale ottenuto
$N°$ 5	Impostare il parametro `FW_PR_D`= 0.005	Effettuare colpetti di cabrata/picchiata e verificare la risposta ottenuta per il *pitch rate*. Verificare che la risposta sia effettivamente migliore di quella ottenuta in assenza del parametro derivativo, raddoppiare il valore fino a che si ottine un miglioramento della risposta, poi ridurre il valore finale del 50%

Tabella 5.5: Procedura per la taratura dei parametri di beccheggio

5.3.1 Anello di controllo di rollio

Successivamente all'impostazione dei parametri iniziali indicati nel primo step della Tabella 5.4, è stato configurato il parametro di feedforward impostando `FW_RR_FF` = 0.4. Come indicato nel secondo step della procedura, è stato quindi valutato l'inseguimento della velocità angolare di rollio effettiva rispetto a piccoli comandi di rollio impartiti tramite joystick.
L'analisi della risposta ha evidenziato che un valore pari a $FF = 0.4$ determina un overshoot eccessivo in seguito all'applicazione del comando. Per tale motivo, il parametro è stato progressivamente ridotto, dapprima a 0.3 e successivamente a 0.25. Quest'ultimo valore ha consentito di ottenere un buon allineamento tra la risposta del modello simulato e il setpoint imposto. La risposta corrispondente a tale configurazione è riportata in Figura 5.3.
In accordo con la procedura di taratura adottata, il valore così individuato è stato ulteriormente ridotto del 20%, portando all'impostazione finale di `FW_RR_FF` = 0.2.

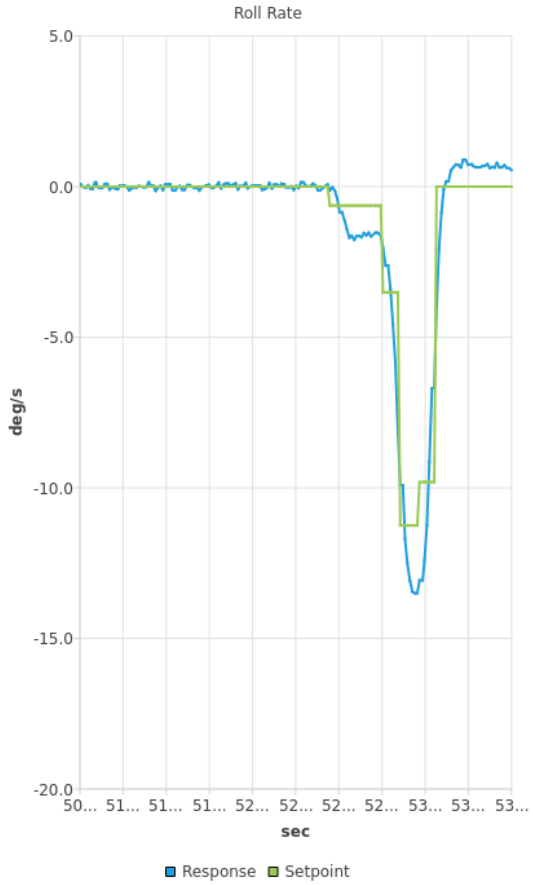


Figura 5.3: Risposta di Roll Rate con FF=0.25

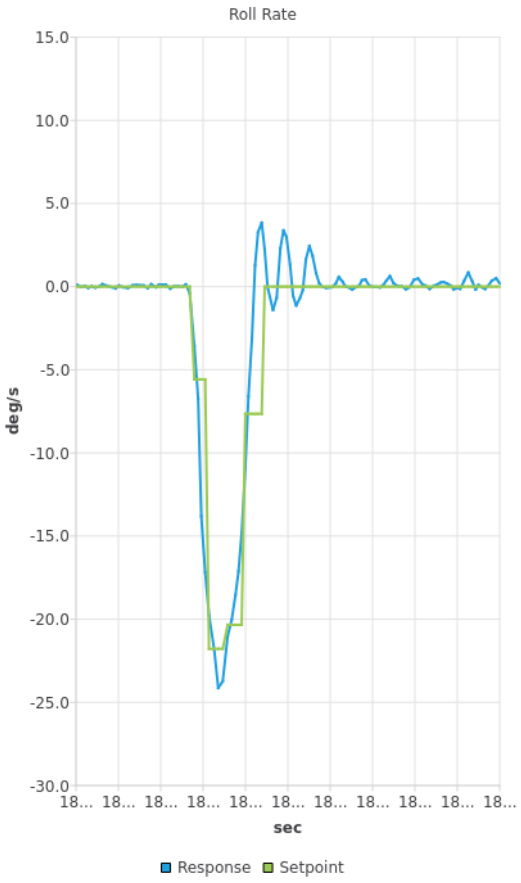


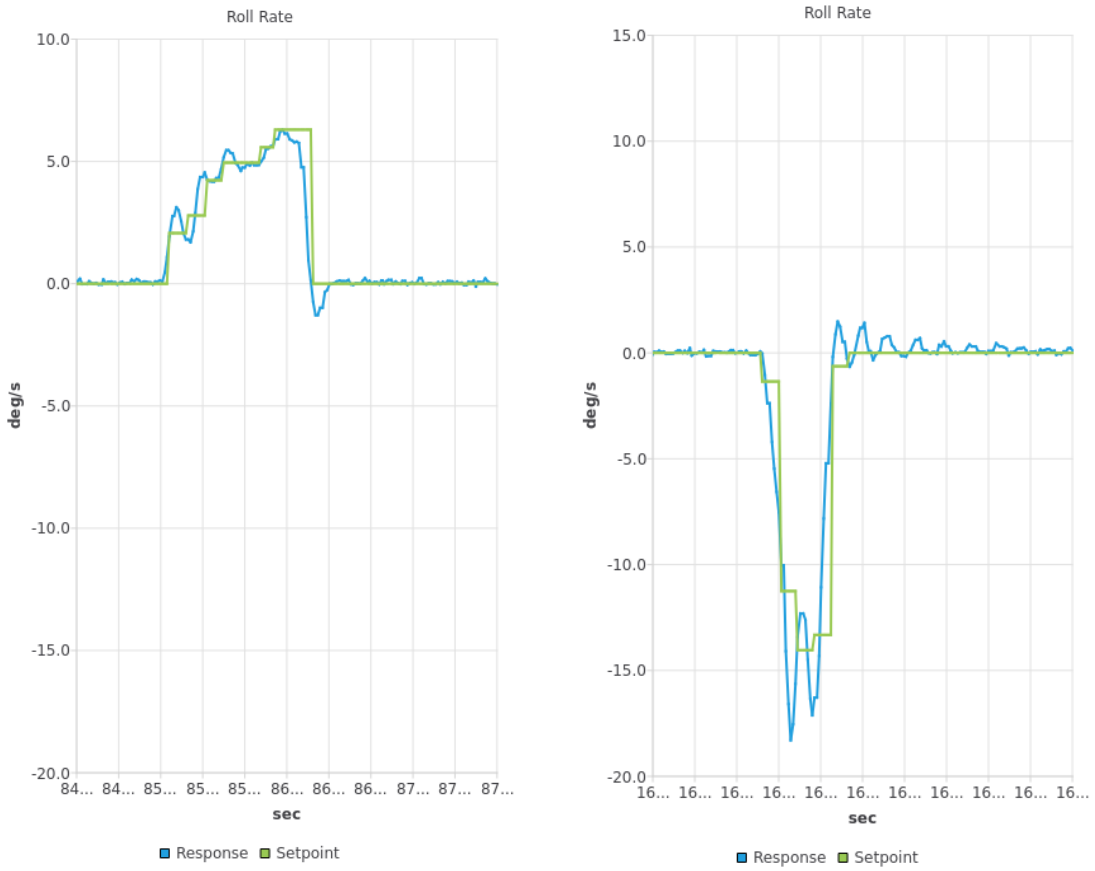
Figura 5.4: Risposta di Roll Rate con ciclo limite

Seguendo lo step successivo della procedura, si è proceduto alla determinazione del guadagno proporzionale in grado di generare un comportamento in ciclo limite nella risposta del sistema.
Per valori inizialmente contenuti del guadagno proporzionale si osserva una progressiva e rapida riduzione dell'errore statico, come mostrato nelle immagini riportate in Figura 5.5. L'incremento graduale del parametro `FW_RR_P` porta a una risposta sempre più reattiva, fino al raggiungimento della condizione di oscillazione sostenuta.
In particolare, è stato necessario impostare `FW_RR_P` $= 0.5$ per osservare la comparsa del ciclo limite nella risposta della velocità angolare di rollio, come illustrato in Figura 5.4. In tale configurazione si evidenzia chiaramente che, a seguito del ritorno del setpoint al valore nullo, la velocità angolare effettiva continua a oscillare attorno allo zero con ampiezza pressoché costante, senza manifestare smorzamento nel tempo. Tale comportamento è indicativo di una condizione di stabilità marginale ed è attribuibile a un'azione proporzionale eccessiva.
In accordo con la procedura adottata e con le regole di taratura di Ziegler–Nichols, il valore individuato come guadagno critico viene quindi ridotto del 50%, ottenendo il valore finale `FW_RR_P` $= 0.25$.
Nel penultimo step della procedura viene determinato il valore del parametro integrativo. Si osserva che, per valori ridotti del guadagno integrale, la riduzione dell'errore a regime risulta insufficiente e permane un evidente steady-state error nella risposta del roll rate. Diventa pertanto necessario incrementare progressivamente il parametro fino a `FW_RR_I` $= 0.02$, che consente di ottenere un errore a regime trascurabile, come evidenziato in Figura 5.6. Il valore individuato risulta inoltre coerente con le indicazioni fornite dal metodo di Ziegler–Nichols. Considerando infatti che il periodo di oscillazione associato al ciclo limite è pari a circa $P_u \approx 0.1\,\mathrm{s}$, l'applicazione delle relazioni previste dalla metodologia conduce al seguente valore teorico del parametro integrale:

$$T_i = \frac{1.2}{P_u} = 12 \tag{5.6}$$

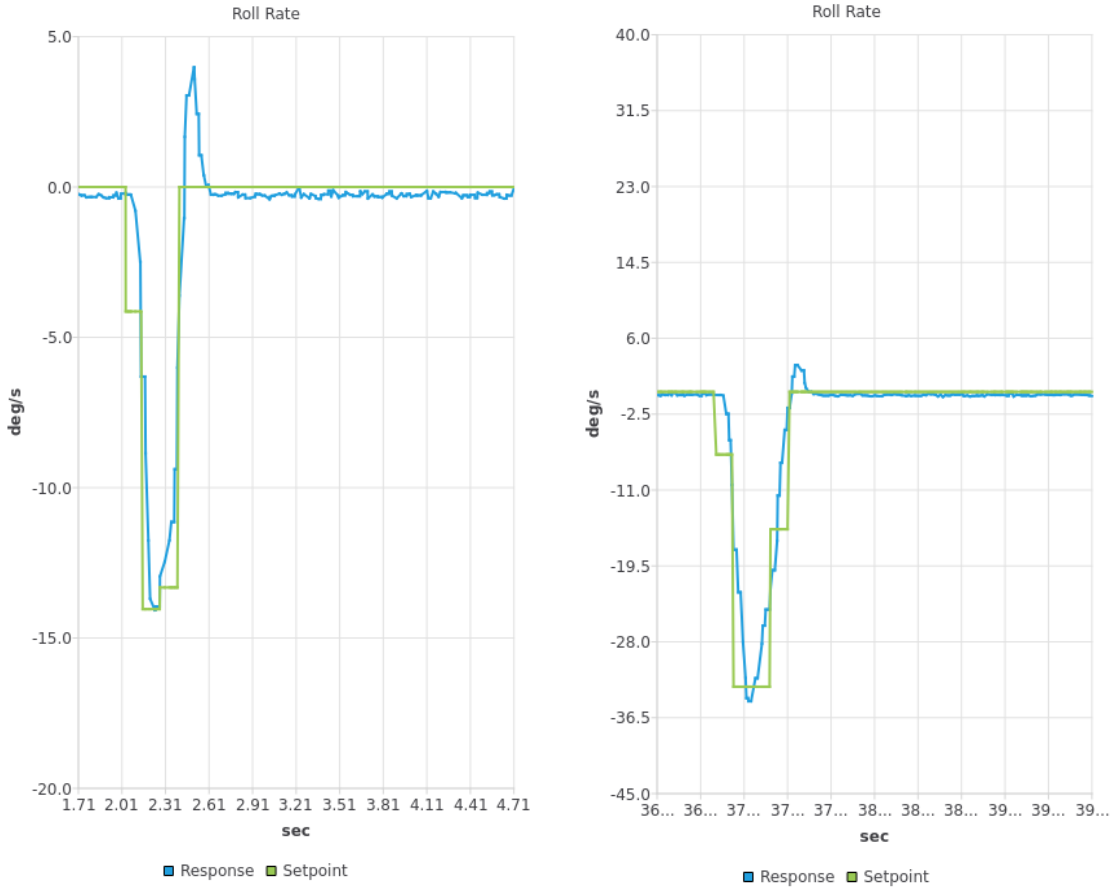
$$K_I = \frac{K_P}{T_i} = \frac{0.25}{12} \approx 0.02 \tag{5.7}$$

Infine, è stata valutata la possibilità di migliorare ulteriormente la risposta del sistema introducendo il termine derivativo. L'inserimento di un contributo derivativo con `FW_RR_D` $= 0.005$ consente effettivamente di ridurre il settling time, rendendo la risposta più pronta nel rientrare attorno al valore di riferimento dopo una variazione del setpoint. Tuttavia, incrementando ulteriormente il valore del parametro derivativo, si osserva un deterioramento significativo della stabilità del sistema, come mostrato in Figura 5.7.



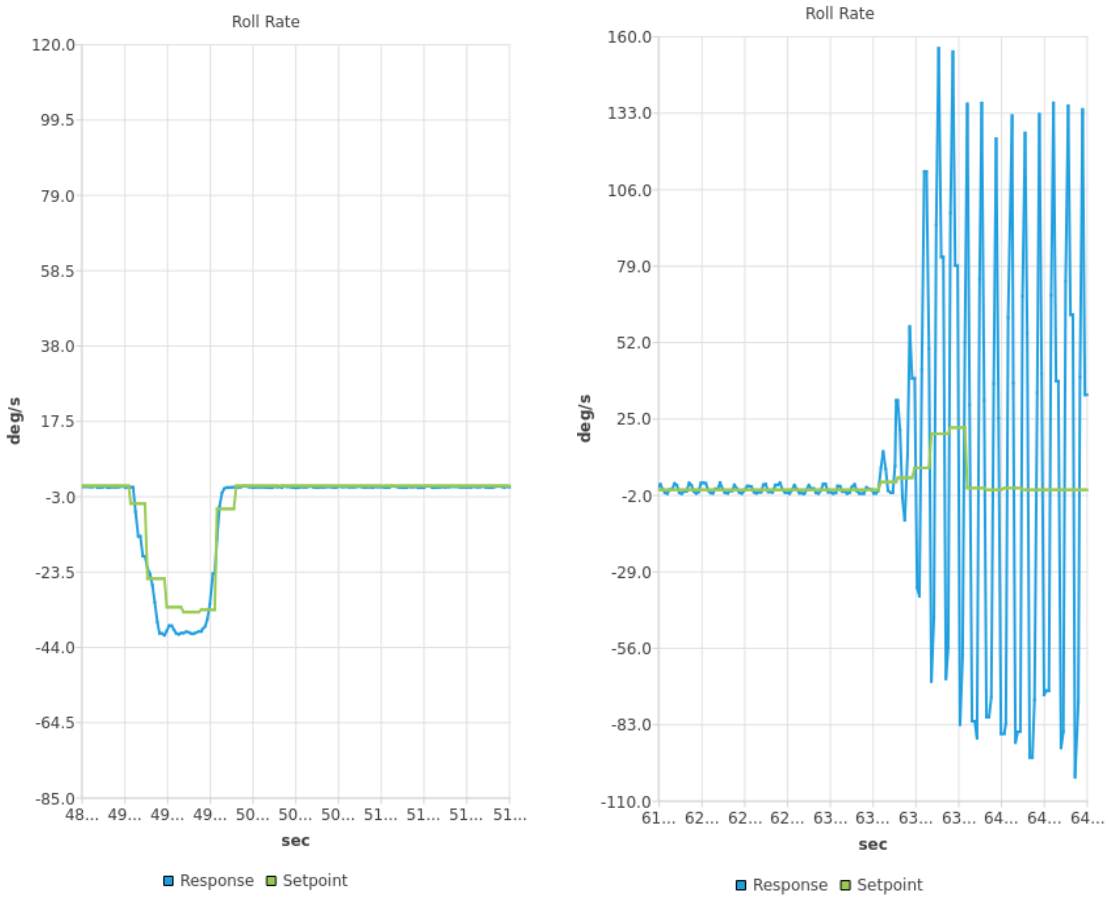
P=0.15 P=0.3

Figura 5.5: Risposta di Roll Rate al variare di K_P



I=0.01 I=0.02

Figura 5.6: Risposta di Roll Rate al variare di K_I

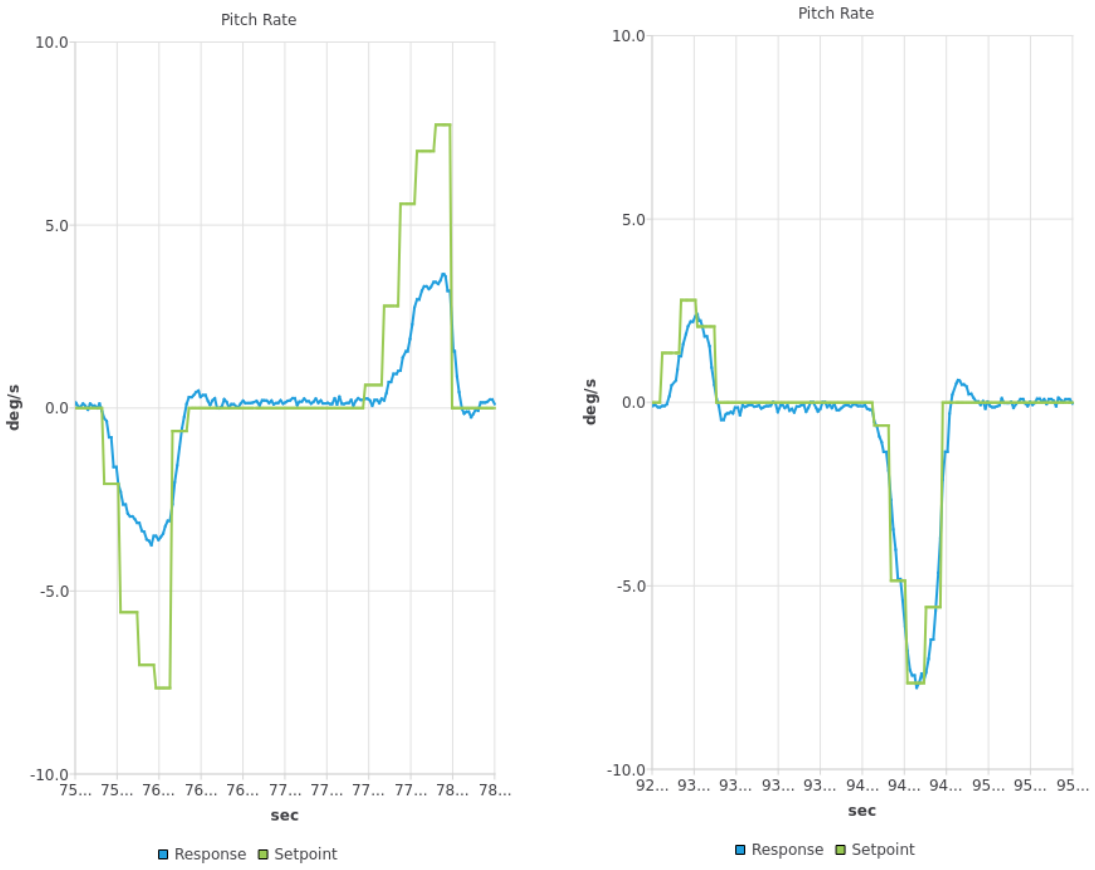


D=0.005 D=0.01

Figura 5.7: Risposta di Roll Rate al variare di K_D

5.3.2 Anello di controllo di beccheggio

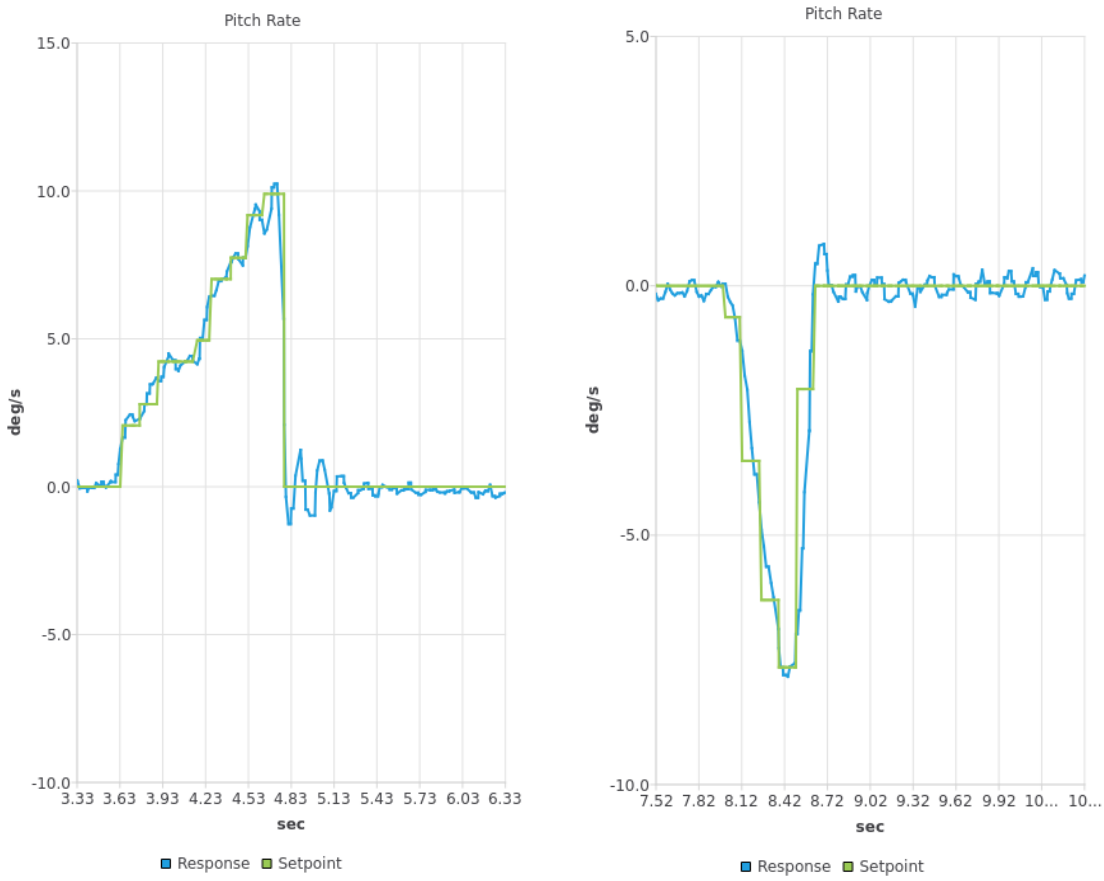
Analogamente, seguendo la procedura riportata in Tabella 5.5, vengono determinati i parametri per il controllo del rate di beccheggio.
In particolare, in Figura 5.8 si osserva come un valore del termine di feedforward pari a `FW_PR_FF`= 0.1 non permetta di seguire adeguatamente il setpoint imposto dal comando di joystick. Per questo motivo, il parametro è stato aumentato fino a ottenere un buon fit per un valore di $FF = 0.25$; il valore finale è quindi ridotto del 20%, ottenendo `FW_PR_FF`= 0.2.



FF=0.1 FF=0.25

Figura 5.8: Risposta di Pitch Rate al variare del parametro di feedforward

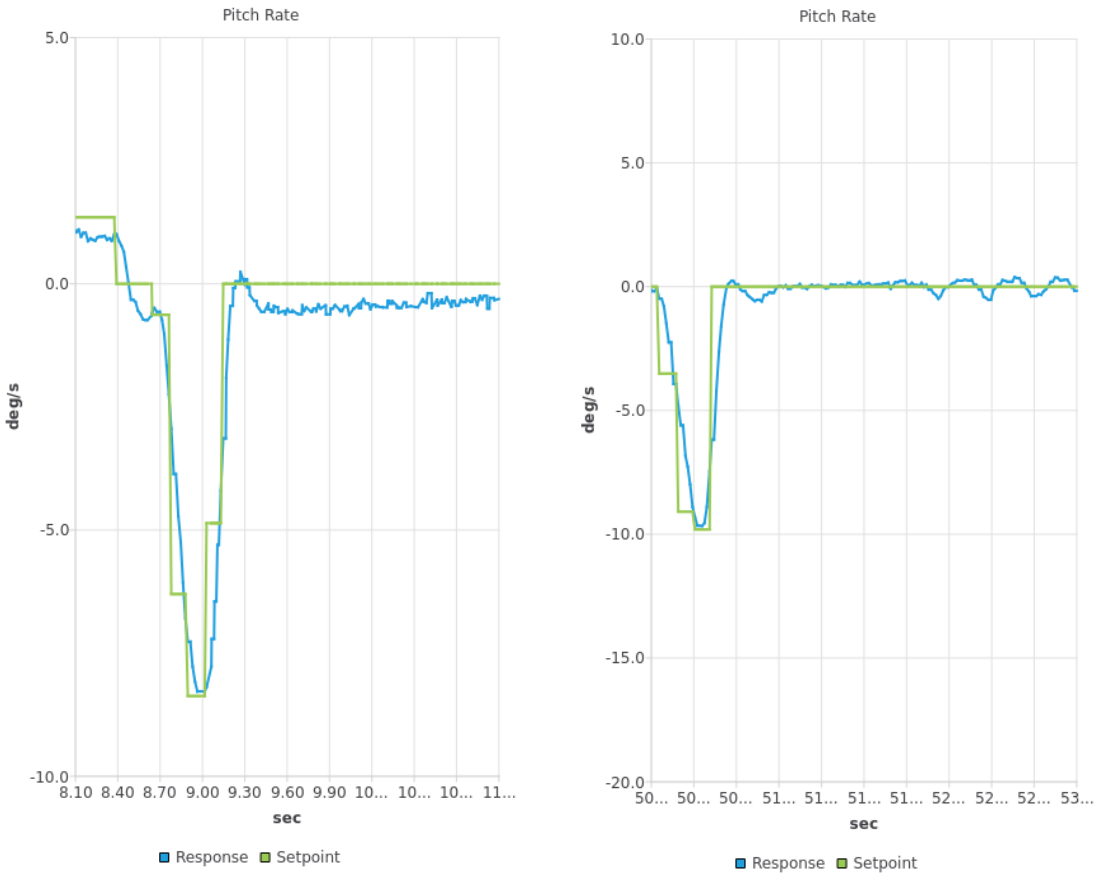
Per quanto riguarda la ricerca del ciclo limite, esso viene individuato per un valore del termine proporzionale pari a $K_{cr} = 0.48$; il parametro finale è stato quindi impostato a `FW_PR_P`= 0.24. Per valori inferiori, la risposta del sistema in seguito a un comando risulta opportunamente smorzata, riducendosi a un valore quasi nullo in pochi istanti, come mostrato in Figura 5.9.



P=0.2 P=0.48

Figura 5.9: Risposta di Pitch Rate al variare di K_P

Il termine integrativo per il modello dinamico simulato deve essere impostato a valori elevati in modo da garantire un'opportuna riduzione dell'errore statico a regime. Il termine derivativo è stato invece aggiunto al fine di migliorare la risposta complessiva del sistema, ma non risulta strettamente necessario. Questi ultimi due valori sono stati impostati rispettivamente a `FW_PR_I`= 0.16 e `FW_PR_D`= 0.01, e la loro rappresentazione viene mostrata nelle Figure 5.10 e 5.11.



I=0.02 I=0.16

Figura 5.10: Risposta di Pitch Rate al variare di K_I

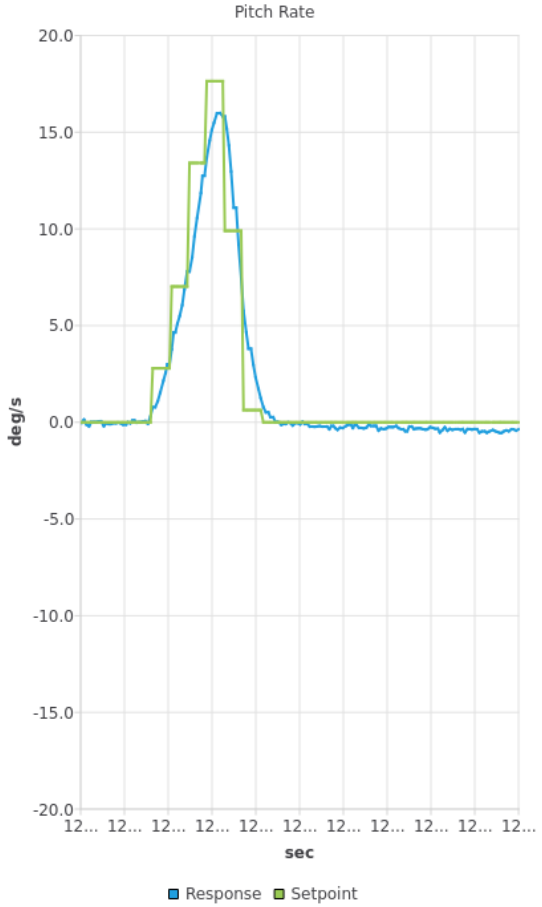


Figura 5.11: Risposta di Pitch Rate con $K_D = 0.01$

5.4 Confronto risultati

I risultati ottenuti mediante la taratura manuale eseguita tramite simulazione HIL sul modello dinamico sono stati confrontati sia con quelli derivanti dalla procedura di *autotune* applicata al modello, sia con i risultati della taratura in volo condotta sul velivolo reale.

L'autotuning dei parametri del controllore implementato nel firmware PX4 è una procedura automatica finalizzata alla determinazione dei guadagni dei controllori PID. Tale procedura è basata sull'applicazione di manovre di eccitazione controllata sugli assi di rollio, beccheggio e imbardata, eseguite uno alla volta in condizioni di volo stazionario. Durante ciascuna manovra automatizzata, il sistema impone variazioni limitate del riferimento di velocità angolare e analizza la risposta dinamica del velivolo in termini di ampiezza, smorzamento e stabilità, valutando la presenza di oscillazioni o ritardi di fase. A partire da tali osservazioni, i guadagni del controllore vengono aggiornati in modo incrementale secondo criteri empirici volti a migliorare la rapidità della risposta e a ridurre l'errore a regime, mantenendo al contempo un adeguato margine di stabilità. L'avanzamento del processo è mostrato nella barra accanto al pulsante di autotune e, se non viene interroto da errori, termina dopo circa 40 secondi.

La taratura del velivolo reale è stata invece condotta durante prove di volo sperimentali, seguendo le medesime procedure operative descritte nelle Tabelle 5.4 e 5.5. In tale contesto, le manovre sono state eseguite in modalità *Acro* da un pilota certificato, il quale ha avuto il compito di garantire il mantenimento di condizioni di volo stabili prima dell'esecuzione di ciascun step, evitando al contempo manovre eccessivamente brusche o situazioni di instabilità che potessero compromettere la sicurezza del velivolo. La selezione finale dei parametri di controllo è stata effettuata non solo sulla base dell'analisi della risposta del sistema all'inseguimento delle velocità angolari, ma anche tenendo conto del feedback qualitativo fornito dal pilota durante le prove di volo.
Il confronto dei parametri ottenuti è riassunto nelle Tabelle 5.6 e 5.7.

Tuning Type	**FW_RR_FF**	**FW_RR_P**	**FW_RR_I**	**FW_RR_D**
Modello (manuale)	0.2	0.25	0.02	0.005
Modello (Autotune)	0.15	0.16	0.02	0
Velivolo reale	0.25	0.455	0.035	0

Tabella 5.6: Confronto dei parametri del controllore PID di rollio

Tuning Type	**FW_PR_FF**	**FW_PR_P**	**FW_PR_I**	**FW_PR_D**
Modello (manuale)	0.2	0.24	0.16	0.01
Modello (Autotune)	0.25	0.165	0.19	0.01
Velivolo reale	0.15	0.305	0.03	0

Tabella 5.7: Confronto dei parametri del controllore PID di beccheggio

Nel complesso si osserva un buon allineamento tra le diverse procedure di taratura, pur in presenza di lievi scostamenti nei valori dei parametri.
Le differenze tra il tuning manuale e l'autotune del modello dinamico in simulazione non sono attribuibili a variazioni della dinamica del sistema, la quale rimane invariata, bensì al differente criterio adottato per la selezione dei parametri. Il metodo di Ziegler–Nichols tende infatti a privilegiare la rapidità della risposta e la riduzione dell'errore statico, conducendo generalmente a valori dell'azione proporzionale più elevati e a un contributo integrativo più incisivo. Tale impostazione può tuttavia comportare un maggiore overshoot e una risposta più aggressiva. L'autotune di PX4, al contrario, si basa su una procedura automatizzata che utilizza eccitazioni controllate del sistema per stimarne le caratteristiche dinamiche e determinare parametri in grado di garantire adeguati margini di stabilità. L'obiettivo principale è quindi la robustezza del controllo più che la massima rapidità di risposta; per questo motivo è frequente ottenere valori di K_P e K_I inferiori rispetto a quelli derivanti dalla procedura manuale, con una risposta complessivamente più conservativa ma maggiormente stabile.

Le differenze tra la taratura del modello dinamico e quella del velivolo reale in volo sono invece riconducibili alle inevitabili discrepanze tra la fisica del sistema reale e le approssimazioni introdotte nel modello per riprodurlo.
Il termine di feedforward, strettamente legato all'efficacia delle superfici mobili, ha il compito di fornire un contributo diretto al comando, riducendo il lavoro richiesto alle azioni proporzionale e integrativa e migliorando l'immediatezza della risposta. I valori ottenuti risultano infatti piuttosto simili tra modello e velivolo reale, a conferma di una buona stima dell'efficacia aerodinamica delle superfici di controllo.
Differenze più marcate si osservano invece nel guadagno proporzionale, che nel caso del volo reale assume valori più elevati. Ciò può essere attribuito alla presenza di fenomeni non lineari, ritardi nella catena di attuazione e comunicazione tra i componenti, nonché accoppiamenti longitudinali e latero-direzionali che rendono il velivolo reale meno reattivo rispetto al modello simulato. Per garantire la stessa prontezza ai comandi risulta quindi necessario incrementare l'azione proporzionale. A ciò si aggiunge la possibile non perfetta stima di alcuni parametri del modello, quali le inerzie o i coefficienti di smorzamento aerodinamico, che possono alterare la dinamica simulata rispetto a quella effettiva.
Una differenza particolarmente evidente riguarda infine il guadagno integrativo K_I nel controllo di beccheggio, significativamente più elevato nella simulazione rispetto al volo reale. Tale comportamento può essere ricondotto a un diverso assetto di trim aerodinamico, già emerso nella fase di validazione del modello, che in simulazione genera uno

steady-state error più marcato e richiede quindi un contributo integrativo maggiore per essere annullato.

Nel complesso, tali differenze confermano la buona coerenza qualitativa tra modello e sistema reale, evidenziando al contempo come la taratura in volo rimanga uno strumento indispensabile per compensare effetti non modellati e garantire prestazioni ottimali nelle reali condizioni operative.

Capitolo 6

Conclusioni

La finalità principale della presente tesi consiste nel valutare l'efficacia della taratura del controllore di un UAV effettuata mediante simulazione di un modello dinamico capace di replicare, con adeguato livello di fedeltà, la fisica del sistema reale. L'obiettivo è quindi verificare la validità di tale metodologia e comprenderne il potenziale impiego sistematico all'interno del processo di progettazione, sviluppo e testing di nuovi velivoli a pilotaggio remoto.
I risultati ottenuti possono essere considerati promettenti. Dal confronto tra i parametri del controllore determinati in ambiente HIL e quelli successivamente ricavati dalla taratura in volo del velivolo reale emerge infatti una buona corrispondenza tra i valori individuati. Inoltre, la risposta del controllore ai disturbi risulta sufficientemente reattiva e stabile, confermando che la simulazione è in grado di fornire una stima coerente dei guadagni necessari per garantire prestazioni adeguate.
È tuttavia necessario sottolineare che il processo di simulazione adottato rappresenta un'approssimazione non trascurabile del sistema reale. Le discrepanze introdotte sono riconducibili sia alla natura lineare del modello dinamico, valido in prossimità del punto di trim, sia agli strumenti utilizzati per la determinazione dei parametri aerodinamici e propulsivi, oltre che alla variabilità dei fenomeni nel flusso di comunicazione tra i componenti del sistema.
Per tali ragioni, l'approccio sviluppato non intende sostituire la taratura in volo, che rimane una fase imprescindibile per l'ottimizzazione finale delle prestazioni, ma configurarsi come uno step preliminare ad essa. La simulazione HIL consente infatti di individuare valori iniziali per i parametri del controllore, riducendo in modo significativo i rischi associati a prove di volo condotte con un sistema non adeguatamente tarato e contribuendo a rendere il processo di sviluppo più sicuro, strutturato ed efficiente.

Parallelamente, il lavoro di tesi ha approfondito lo sviluppo di un modello dinamico specifico per il velivolo FunCub NG, con l'obiettivo di ottenere una rappresentazione del comportamento dinamico quanto più possibile coerente con quello del sistema reale, preservando al contempo la semplicità e la flessibilità tipiche di un modello lineare. In relazione a tale obiettivo, e alla scelta di integrare il modello all'interno di un ambiente

di simulazione Hardware-in-the-Loop per valutarne le prestazioni, è possibile trarre le seguenti conclusioni:

- La dinamica longitudinale e quella latero–direzionale risultano complessivamente ben riprodotte, con una corretta identificazione delle frequenze principali associate ai modi dinamici fondamentali. Il modello lineare si dimostra quindi adeguato, pur non consentendo la rappresentazione di fenomeni fortemente non lineari o di accoppiamenti complessi tra le dinamiche.
- Le incertezze nella determinazione dei coefficienti aerodinamici e nella modellazione del sistema propulsivo introducono errori sistematici su alcune variabili di stato. Tali errori possono generare condizioni di trim differenti rispetto al sistema reale e, a causa del processo di integrazione numerica, tendono ad amplificarsi nel tempo, fino a compromettere la validità della simulazione su finestre temporali estese.
- La simulazione HIL si è rivelata uno strumento particolarmente efficace per l'individuazione delle principali fonti di errore del modello e per la verifica del comportamento del controllore in un ambiente sicuro e riproducibile. Sebbene le fasi iniziali di configurazione richiedano procedure articolate e tempi di preparazione non trascurabili, i benefici in termini di sicurezza, tracciabilità dei test e possibilità di iterazione rapida risultano significativi.

6.1 Sviluppi Futuri

Nonostante i risultati ottenuti possano ritenersi complessivamente soddisfacenti, il lavoro svolto apre diverse possibili linee di sviluppo e approfondimento:

1. **Raffinamento del modello**: Il modello attualmente sviluppato può essere ulteriormente migliorato alla luce delle discrepanze osservate durante la fase di validazione. Un primo intervento potrebbe riguardare una stima più accurata dei parametri aerodinamici, ottenuta mediante strumenti di calcolo a maggiore fedeltà oppure tramite prove sperimentali dedicate e tecniche di system identification sui dati di volo. Il sistema propulsivo rappresenta l'aspetto più critico della modellazione, essendo stato sviluppato principalmente su base teorica. Una caratterizzazione sperimentale su banco prova del motore elettrico e dell'elica consentirebbe di identificare con maggiore precisione i parametri elettrici e meccanici, includendo effetti attualmente trascurati quali l'inerzia dell'elica, i transitori elettrici e le perdite non lineari. Un ulteriore miglioramento potrebbe consistere nell'integrazione di modelli di disturbo ambientale e di vento, rendendo la simulazione più realistica e permettendo di valutare la robustezza del controllore in condizioni operative non ideali.
2. **Estensione a modello non lineare**: L'adozione di un modello completamente non lineare, sebbene più complessa dal punto di vista implementativo e computazionale, consentirebbe di rappresentare in modo più accurato la dinamica del velivolo in un ampio inviluppo di volo. Un tale modello permetterebbe di analizzare manovre lontane dalle condizioni di trim, includere in maniera più realistica la dinamica degli attuatori e descrivere fenomeni aerodinamici non lineari, migliorando la validità delle simulazioni su intervalli temporali estesi.

3. **Estensione a livelli superiori di controllo**: Il lavoro si è focalizzato principalmente sulla taratura del controllore di rate. In prospettiva futura, l'ambiente Hardware-in-the-Loop sviluppato potrebbe essere esteso alla taratura e validazione dei controllori di assetto, velocità e posizione, consentendo una verifica completa dell'architettura di controllo del velivolo. Ciò renderebbe il framework HIL uno strumento ancora più integrato all'interno del processo di progettazione e validazione dell'intero sistema di guida e controllo.

Appendice A

Sistemi di Riferimento

La definizione dei vari sistemi di riferimento utilizzati è fondamentale per modellare correttamente il velivolo e descriverne la dinamica.
Per questo motivo è necessario introdurre diversi sistemi di coordinate, ciascuno dei quali svolge una funzione specifica. I sistemi di coordinate adottati sono gli stessi utilizzati nel pacchetto `UAV Toolbox` di MATLAB/Simulink [32].

A.1 Sistema Body

Il sistema di riferimento *Body* è un sistema di coordinate solidale al velivolo [33]. La terna degli assi corpo, avente origine nel baricentro dell'UAV, è composta da:

- X_{body}: Asse longitudinale contenuto nel piano di simmetria e orientato positivamente verso la prua.
- Z_{body}: Asse contenuto nel piano di simmetria e orientato positivamente verso il basso.
- Y_{body}: Asse che completa la terna levogira, orientato positivamente verso destra.

Sebbene tale sistema abbia origine nel baricentro dell'UAV, ai soli fini della definizione della geometria del velivolo mostrata in sezione 3.3 è stato impiegato un sistema di riferimento solidale al velivolo con origine nel bordo d'attacco dell'ala.

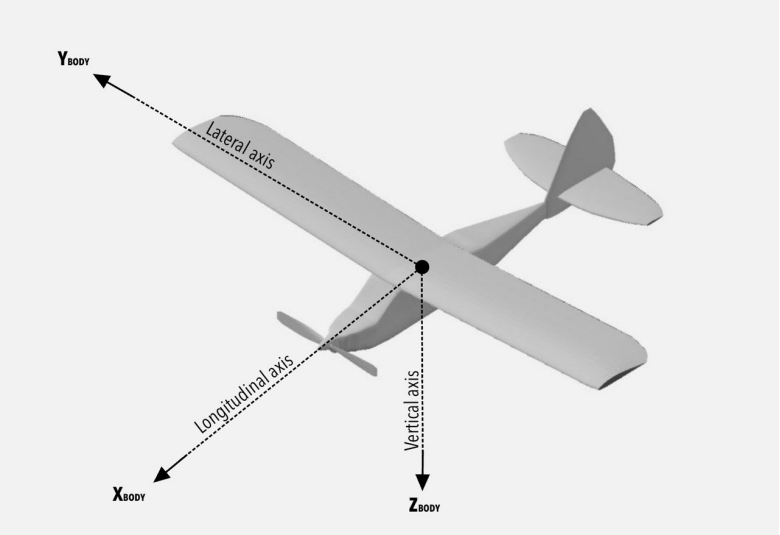


Figura A.1: Sistema di Riferimento Body

A.2 Sistema NED inerziale

Come sistema di riferimento inerziale è stato scelto il sistema North-East-Down (NED) [34], un sistema fisso solidale alla Terra la cui origine è definita sulla superficie terrestre e coincide con le coordinate di homing dell'UAV.
L'origine del sistema O^{NED} è specificata tramite coordinate geodetiche (latitudine, longitudine e altitudine), mentre gli assi sono definiti come:

- X_{NED}: Diretto verso Nord, lungo il meridiano passante per la longitudine di O^{NED}.
- Y_{NED}: Diretto verso Est, lungo il parallelo corrispondente alla latitudine di O^{NED}.
- Z_{NED}: Diretto verso il basso, così da rispettare la regola della mano destra.

È inoltre utile definire un sistema NED traslato nel baricentro del velivolo. Tale sistema non è inerziale, poiché trasla insieme all'UAV; tuttavia, a differenza del sistema *Body*, mantiene gli assi paralleli a quelli del sistema NED inerziale [35]. Questo riferimento intermedio permette di caratterizzare gli angoli di assetto, definiti dall'offset di rotazione tra tale sistema e quello body.
L'orientamento relativo tra i due sistemi può essere descritto tramite gli angoli di Eulero di imbardata (ψ), beccheggio (θ) e rollio (ϕ) che, seguendo la sequenza di rotazione Z–Y–X (3–2–1), consentono di definire la matrice di trasformazione dalle coordinate NED a quelle body. Immaginando di avere l'UAV posizionato tale che la terna di assi body sia parallela alla terna inerziale, possiamo applicare le seguenti rotazioni (figura A.3) [36]:

1. Rotazione del sistema x_f, y_f, z_f di un angolo ψ attorno all'asse z_f, si ottiene il sistema x_1, y_1, z_1.
2. Rotazione del sistema x_1, y_1, z_1 di un angolo θ attorno all'asse y_1, si ottiene il sistema x_2, y_2, z_2.

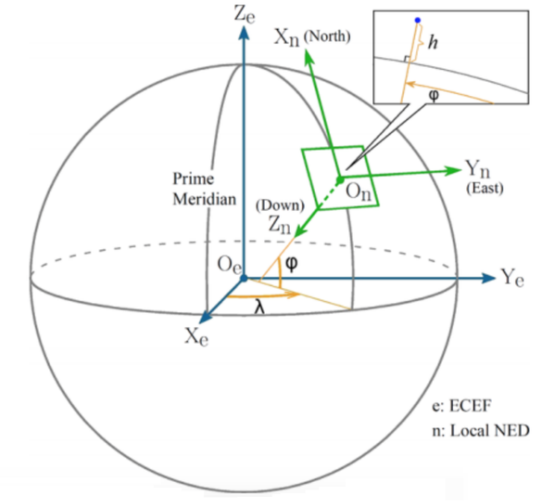


Figura A.2: Sistema inerziale NED

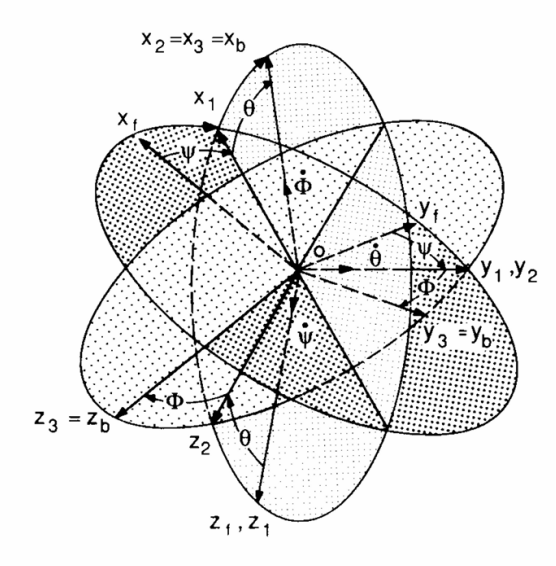


Figura A.3: Relazione tra sistema di riferimento body e inerziale

3. Rotazione del sistema x_2, y_2, z_2 di un angolo ϕ attorno all'asse x_2, si ottiene il sistema $x_3 = x_b$, $y_3 = y_b$, $z_3 = z_b$.

Ciascuna rotazione è descritta da una matrice; la loro composizione fornisce la matrice di rotazione complessiva, indicata in MATLAB/Simulink come DCM_{be}, che consente di convertire un vettore espresso in coordinate NED in un vettore espresso in coordinate body:

$$DCM_{be} = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\psi s_\theta s_\phi + c_\phi c_\psi & c_\theta s_\phi \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\theta c_\phi \end{bmatrix} \tag{A.1}$$

dove c_x indica il coseno dell'angolo x e s_x il seno dell'angolo x.
La trasformazione inversa può essere ottenuta invertendo la sequenza delle rotazioni o invertendo la matrice DCM_{be}. Poiché tale matrice è ortogonale, si ha che $[DCM_{be}]^{-1} = [DCM_{be}]^T$, ovvero la matrice che converte un vettore dalle coordinate body a quelle NED. Le operazioni utili sono quindi:

$$\begin{aligned} \begin{pmatrix} x_{body} \\ y_{body} \\ z_{body} \end{pmatrix} &= DCM_{be} \begin{pmatrix} x_{NED} \\ y_{NED} \\ z_{NED} \end{pmatrix} \\ \begin{pmatrix} x_{NED} \\ y_{NED} \\ z_{NED} \end{pmatrix} &= [DCM_{be}]^T \begin{pmatrix} x_{body} \\ y_{body} \\ z_{body} \end{pmatrix} \end{aligned} \tag{A.2}$$

A.3 Sistema assi vento

Per il calcolo delle forze aerodinamiche agenti sul velivolo è necessario definire la terna degli assi vento.
Questa terna, avente origine nel baricentro dell'UAV, è solidale al vento relativo e presenta i seguenti assi:

- X_{wind}: Coincidente con la direzione e verso della velocità relativa.
- Z_{wind}: Perpendicolare a X_{wind} e giacente nel piano di simmetria dell'aeromobile.
- Y_{wind}: Perpendicolare a X_{wind} e Z_{wind}, con verso tale da formare una terna levogira.

La rotazione dal sistema vento al sistema body può essere definita tramite due rotazioni elementari basate sull'angolo di incidenza α e sull'angolo di sideslip β. La composizione di tali rotazioni permette di ottenere la matrice di trasformazione completa:

$$RW2B = \begin{bmatrix} \cos\beta\cos\alpha & -\sin\beta\cos\alpha & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \cos\beta\sin\alpha & -\sin\beta\sin\alpha & \cos\alpha \end{bmatrix} \tag{A.3}$$

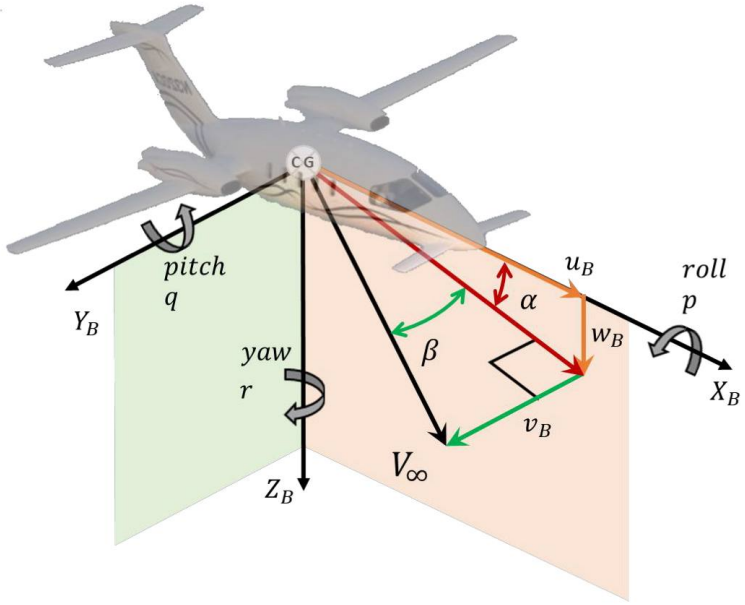


Figura A.4: Sistema Vento

Appendice B

Equazioni di Eulero e Dinamica del Volo

B.1 Equazioni di Eulero

Le equazioni di Eulero descrivono la dinamica di un corpo rigido soggetto all'azione di forze e momenti esterni e costituiscono la base per la modellazione del moto di un velivolo nello spazio tridimensionale.
La prima equazione cardinale della dinamica descrive il moto traslatorio del centro di massa del velivolo, il quale può essere assimilato a un punto materiale di massa pari alla massa totale dell'UAV, soggetto alla risultante delle forze esterne agenti. Tale relazione, espressa nel sistema di riferimento solidale al corpo (*body frame*), assume la forma:

$$\mathbf{F}_b = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m\left(\dot{\mathbf{V}}_b + \boldsymbol{\omega} \times \mathbf{V}_b\right) \tag{B.1}$$

dove $\mathbf{V}_b$ rappresenta il vettore velocità del centro di massa espresso in assi *body*, $\dot{\mathbf{V}}_b$ la corrispondente accelerazione, $\boldsymbol{\omega}$ il vettore delle velocità angolari del velivolo e m la massa totale.
La *seconda equazione cardinale della dinamica* descrive invece il moto rotatorio del velivolo attorno al centro di massa ed è espressa come:

$$\mathbf{M}_b = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \mathbf{I}\,\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I}\,\boldsymbol{\omega}) \tag{B.2}$$

dove $\mathbf{M}_b$ è il vettore dei momenti esterni applicati, $\dot{\boldsymbol{\omega}}$ il vettore delle accelerazioni angolari e $\mathbf{I}$ la matrice d'inerzia del velivolo, assunta costante nel sistema di riferimento *body*.
L'integrazione nel tempo delle equazioni cardinali consente di ottenere l'evoluzione delle velocità traslazionali e angolari del velivolo nel sistema di riferimento solidale al corpo.

Mediante l'utilizzo della matrice di rotazione DCM_{be}, tali grandezze possono essere successivamente riportate nel sistema di riferimento inerziale.
Per quanto riguarda la descrizione dell'assetto del velivolo, la variazione temporale degli angoli di Eulero può essere espressa in funzione delle velocità angolari p, q e r, secondo la seguente relazione cinematica:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\,\tan\theta & \cos\phi\,\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{B.3}$$

L'integrazione di tale equazione permette di ricavare l'evoluzione temporale degli angoli di Eulero e, conseguentemente, di costruire la matrice di rotazione DCM_{be} necessaria alla trasformazione tra i sistemi di riferimento.

B.2 Modello Aerodinamico

Le azioni aerodinamiche agenti su un velivolo dipendono dalla pressione dinamica di volo e dalle dimensioni caratteristiche dell'aeromobile. In particolare, le forze aerodinamiche risultano proporzionali a un'area di riferimento, mentre i momenti aerodinamici sono proporzionali al prodotto tra un'area e una lunghezza caratteristica.
Seguendo la convenzione comunemente adottata in letteratura, tali dipendenze vengono espresse mediante opportuni coefficienti di proporzionalità adimensionali, detti *coefficienti aerodinamici*, che dipendono esclusivamente dalla forma del velivolo e dalle condizioni di volo. Il modulo F_A della forza aerodinamica risultante può quindi essere scritto come:

$$F_A = C_F\,\bar{q}_\infty\,S \tag{B.4}$$

dove C_F è il coefficiente aerodinamico di forza, $\bar{q}_\infty$ è la pressione dinamica indisturbata e S è l'area di riferimento, assunta coincidente con l'area alare in pianta [37].
L'introduzione dei coefficienti adimensionali consente di descrivere la dipendenza delle azioni aerodinamiche dai parametri di volo e di controllo. In generale, tali coefficienti sono funzione degli angoli aerodinamici di assetto α e β, delle componenti della velocità angolare del velivolo p, q e r, nonché delle deflessioni delle superfici di controllo δ_a, δ_e e δ_r. Fa eccezione il coefficiente di resistenza C_D, che nel modello adottato viene ricavato tramite la polare aerodinamica del velivolo.

Ipotesi e assunzioni del modello

Il modello aerodinamico implementato si basa sulle seguenti ipotesi principali:

- **Ipotesi di quasi stazionarietà**: le forze e i momenti aerodinamici dipendono istantaneamente dallo stato del velivolo, trascurando effetti non stazionari di ordine superiore.
- **Disaccoppiamento longitudinale e latero–direzionale**: le caratteristiche aerodinamiche longitudinali dipendono esclusivamente da variabili simmetriche, mentre quelle latero–direzionali dipendono solo da variabili asimmetriche.

- **Piccole perturbazioni**: il modello è valido in un intorno limitato di una condizione di equilibrio stabilizzata.
- **Comportamento lineare**: le dipendenze dei coefficienti aerodinamici dai parametri di volo e di controllo vengono approssimate tramite relazioni lineari.

Alla luce di tali assunzioni, le dipendenze funzionali dei coefficienti aerodinamici possono essere espresse come:

$$\begin{aligned} C_L &= C_L(\alpha,\ \dot{\alpha},\ q,\ \delta_e) \\ C_D &= C_D(C_L) \\ C_m &= C_m(\alpha,\ \dot{\alpha},\ q,\ \delta_e) \\ C_Y &= C_Y(\beta,\ p,\ r,\ \delta_r) \\ C_\ell &= C_\ell(\beta,\ p,\ r,\ \delta_a,\ \delta_r) \\ C_n &= C_n(\beta,\ p,\ r,\ \delta_a,\ \delta_r) \end{aligned} \tag{B.5}$$

Linearizzazione del modello

Il modello adottato è di tipo lineare ed è ottenuto mediante la linearizzazione dei coefficienti aerodinamici attorno a una condizione di equilibrio stabilizzata, sfruttando la teoria delle piccole perturbazioni. In tale contesto, ciascun coefficiente viene espresso come somma del valore in equilibrio e dei contributi dovuti alle variazioni dei parametri di stato e di controllo.
Questa approssimazione risulta valida esclusivamente finché il sistema opera in un intorno sufficientemente ridotto della condizione di equilibrio considerata.
I coefficienti aerodinamici assumono pertanto la seguente forma:

$$\begin{aligned} C_L &= C_{L_0} + C_{L_\alpha}\,\alpha + C_{L_{\dot{\alpha}}}\,\dot{\alpha} + C_{L_q}\,q + C_{L_{\delta_e}}\,\delta_e \\ C_D &= C_{D_0} + A_1\,C_L + A_{\text{polar}}\,C_L^2 \\ C_m &= C_{m_0} + C_{m_\alpha}\,\alpha + C_{m_{\dot{\alpha}}}\,\dot{\alpha} + C_{m_q}\,q + C_{m_{\delta_e}}\,\delta_e \\ C_Y &= C_{Y_\beta}\,\beta + C_{Y_{\delta_r}}\,\delta_r + C_{Y_p}\,p + C_{Y_r}\,r \\ C_\ell &= C_{\ell_\beta}\,\beta + C_{\ell_{\delta_r}}\,\delta_r + C_{\ell_{\delta_a}}\,\delta_a + C_{\ell_p}\,p + C_{\ell_r}\,r \\ C_n &= C_{n_\beta}\,\beta + C_{n_{\delta_r}}\,\delta_r + C_{n_{\delta_a}}\,\delta_a + C_{n_p}\,p + C_{n_r}\,r \end{aligned} \tag{B.6}$$

I valori delle derivate aerodinamiche sono stati ricavati per il velivolo in esame e sono riportati nella Sezione 3.4.

Le derivate aerodinamiche rispetto alle velocità angolari fornite dagli strumenti di analisi sono espresse in forma adimensionale, mediante l'impiego di una velocità di riferimento V_{ref} e di una lunghezza caratteristica (c_{ref} per il moto longitudinale e b_{ref} per il moto latero–direzionale).

Nel modello dinamico implementato, tuttavia, si rende necessario utilizzare le corrispondenti derivate dimensionali, espresse in s/rad, così da poterle moltiplicare direttamente per le velocità angolari istantanee durante l'integrazione numerica.
A titolo di esempio, la procedura di dimensionalizzazione assume la forma:

$$\begin{aligned} C_{L_q}\ (\text{s/rad}) &= \hat{C}_{L_q}\,\frac{c_{\text{ref}}}{V_{\text{ref}}} \\ C_{\ell_p}\ (\text{s/rad}) &= \hat{C}_{\ell_p}\,\frac{b_{\text{ref}}}{2\,V_{\text{ref}}} \\ C_{m_q}\ (\text{s/rad}) &= \hat{C}_{m_q}\,\frac{c_{\text{ref}}}{2\,V_{\text{ref}}} \\ C_{n_r}\ (\text{s/rad}) &= \hat{C}_{n_r}\,\frac{b_{\text{ref}}}{2\,V_{\text{ref}}} \end{aligned} \tag{B.7}$$

Bibliografia

[1] IEEE member M. Bacic. «On hardware-in-the-loop simulation». In: *44th IEEE Conference on Decision and Control, and the European Control Conference 2005* (2005) (cit. a p. 1).

[2] Yousefpour Samereh Sarhadi Pouria. «State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software». In: *International Journal of Dynamics and Control* 3 (2015), pp. 470–479. ISSN: 2195-2698. URL: https://doi.org/10.1007/s40435-014-0108-3 (cit. a p. 1).

[3] Martin Schlager, Wilfried Elmenreich e Ingomar Wenzel. «Interface Design for Hardware-in-the-Loop Simulation». In: 2 (2006), pp. 1554–1559. DOI: 10.1109/ISIE.2006.295703 (cit. a p. 2).

[4] Eric R. Mueller. «Hardware-in-the-loop Simulation Design for Evaluation of UnmannedAerial Vehicle Control Systems». In: *AIAA Modeling and Simulation Technologies Conference and Exhibit 20* (2007) (cit. a p. 2).

[5] Franc Mihalič, Mitja Truntič e Alenka Hren. «Hardware-in-the-Loop Simulations: A Historical Overview of Engineering Challenges». In: *Electronics* 11.15 (2022). ISSN: 2079-9292. URL: https://www.mdpi.com/2079-9292/11/15/2462 (cit. a p. 3).

[6] Inc. MathWorks. *Hardware-in-the-loop Simulation (HITL) with PX4*. 2025. URL: https://it.mathworks.com/help/uav/px4-hitl.html?s_tid=CRUX_lftnav (cit. a p. 5).

[7] Inc. MathWorks. *Develop Algorithms and Deploy on PX4 Autopilot*. 2025. URL: https://it.mathworks.com/help/uav/develop-algorithm-deploy-px4.html (cit. a p. 7).

[8] Inc. MathWorks. *MAVLink Support*. 2025. URL: https://it.mathworks.com/help/uav/mavlink-support.html?s_tid=CRUX_lftnav (cit. a p. 7).

[9] PX4 Autopilot. *PX4 Autopilot User Guide*. 2025. URL: https://docs.px4.io/main/en/ (cit. a p. 10).

[10] CubePilot. *The Cube Orange+ Standard Set*. 2025. URL: https://www.cubepilot.com/#/cube/features (cit. a p. 13).

[11] QGroundControl. *QGroundControl Guide*. 2025. URL: https://docs.qgroundcontrol.com/Stable_V5.0/en/qgc-user-guide/ (cit. a p. 15).

[12] E. Pennestrì, V. Rossi, P. Salvini e P. P. Valentini. «Review and comparison of dry friction force models». In: *Nonlinear Dynamics* 83 (2016), pp. 1785–1801. URL: https://doi.org/10.1007/s11071-015-2485-3 (cit. a p. 19).

[13] Inc. MathWorks. *6DOF (Quaternion)*. 2025. URL: https://it.mathworks.com/help/aeroblks/6dofquaternion.html (cit. a p. 21).

[14] MUTIPLEX Modellsport GmbH & Co. *Kit FUNCUB NG green*. 2025. URL: https://shop.multiplex-rc.de/en/kit-funcub-ng-green-made-by-mpx-p8213/ (cit. alle pp. 24, 33).

[15] MUTIPLEX Modellsport GmbH & Co. *Power set FUNCUB NG*. 2025. URL: https://shop.multiplex-rc.de/en/power-set-funcub-ng-p4169/ (cit. a p. 26).

[16] APC Propellers. *Performance Data*. 2025. URL: https://www.apcprop.com/technical-information/performance-data/ (cit. a p. 30).

[17] NASA. *OpenVSP Ground School*. 2025. URL: https://vspu.larc.nasa.gov/ (cit. a p. 33).

[18] Unmanned Systems Technology. *The Cube Autopilot, GNSS Module, HD Video Transmission System and Accessories for UAVs*. 2025. URL: https://www.unmannedsystemstechnology.com/company/cubepilot/cube-orange/ (cit. a p. 39).

[19] MUTIPLEX Modellsport GmbH & Co. *Speed controller ROXXY BL Control 740 S-BEC*. 2025. URL: https://shop.multiplex-rc.de/en/speed-controller-roxxy-bl-control-740-s-bec-p4166/ (cit. a p. 39).

[20] Flash RC. *Motore brushless Roxxy C35-42-06*. 2025. URL: https://www.flashrc.com/it/corridore/36489-motore-brushless-roxxy-c35-42-06-132g-930kv-680w-max-4041033089651.html (cit. a p. 39).

[21] Jonathan Modellismo. *ManiaX Batteria Lipo 4S 14,8V 4200mAh 35C - XT60*. 2025. URL: https://shop.jonathan.it/it/maniax-batteria-lipo-4s-14-8v-4200mah-35c-xt60 (cit. a p. 39).

[22] Flash RC. *RX7 DR Light M-LINK Multiplex Receiver*. 2025. URL: https://www.flashrc.com/en/m-link-light-receivers-without-telemetry/4394-2-4-ghz-rx7-dr-light-m-link-multiplex-receiver-4041033038109.html?l=it (cit. a p. 39).

[23] MUTIPLEX Modellsport GmbH & Co. *Set of leads FUNCUB NG complete*. 2025. URL: https://shop.multiplex-rc.de/en/set-of-leads-funcub-ng-complete-p4194/ (cit. a p. 39).

[24] Flash RC. *Microservo Hitec HS-55+*. 2025. URL: https://www.flashrc.com/it/servi-rc-precisione-e-affidabilita-per-il-modellismo/29974-microservo-hitec-hs-55-8g-1-3kg-cm-0-14s-60.html (cit. a p. 39).

[25] HITEC RCD USA. *HS-65HB 11.2g Nylon Gear Analog Micro Servo*. 2025. URL: https://hitecrcd.com/hs-65hb-mighty-karbonite-feather-servo/ (cit. a p. 39).

[26] David K. Schmidt. *Modern Flight Dynamics*. McGraw-Hill Book Company, 2011 (cit. a p. 49).

[27] PX4 Autopilot. *Configuration Steps*. 2025. URL: https://docs.px4.io/main/en/config/ (cit. a p. 53).

[28] Karl J. Åström e Tore Hägglund. *PID Controllers: Theory, Design, and Tuning*. 2ª ed. International Society of Automation (ISA), 1995. ISBN: 978-1-55617-516-9 (cit. a p. 87).

[29] Vishakha Vijay Patel. «Ziegler–Nichols Tuning Method - Understanding the PID Controller». In: (2020) (cit. a p. 90).

[30] N. B. Nichols J.G. Ziegler. «Optimum Settings for Automatic Controllers». In: *American Society of Mechanical Engineers (ASME)* (1942) (cit. a p. 91).

[31] Brian R. Copeland. «The Design of PID Controllers using Ziegler Nichols Tuning». In: (2008) (cit. a p. 92).

[32] Inc. MathWorks. *Coordinate Systems in UAV Toolbox*. 2025. URL: https://it.mathworks.com/help/uav/ug/coordinate-systems-in-uav-toolbox.html (cit. a p. 114).

[33] Lloyd Duff Reid. Bernard Etkin. *Dynamics of Flight*. John Wiley & Sons, 1995 (cit. a p. 114).

[34] Wikipedia. *Local tangent plane coordinates*. 2025. URL: https://en.wikipedia.org/wiki/Local_tangent_plane_coordinates (cit. a p. 115).

[35] Tong Heng Lee Guowei Cai Ben M. Chen. *Unmanned Rotorcraft Systems*. Springer, 2011 (cit. a p. 115).

[36] Robert C. Nelson. *Flight Stability and Automatic Control*. McGraw-Hill Book Company, 1989 (cit. a p. 115).

[37] D.P. Coiro A. De Marco. *Quaderno 13 - Modellazione aerodinamica e propulsiva*. Università degli Studi di Napoli Federico II, 2017 (cit. a p. 120).