Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria Aerospaziale

Tesi di Laurea Magistrale

SVILUPPO DI UN MODELLO GRAFICO PER SIMULATORE DI VOLO DEL VELIVOLO BELL XV-15 E INTEGRAZIONE IN FLIGHTGEAR

Relatore:
Prof. Giorgio Guglieri
Correlatore:
Dr. Stefano Primatesta

Candidato:
Alessandro Coppa

Anno Accademico 2022/2023

A mia zia Paola

Sommario

Lo scopo del presente elaborato di tesi è la realizzazione e l'implementazione in FlightGear di un modello grafico del convertiplano Bell XV-15. Tale modello grafico è integrato in un progetto più ampio che include la realizzazione di un simulatore di volo del velivolo Bell XV-15 per scopi didattici e di ricerca. Il progetto nasce da una collaborazione fra il Dipartimento di Ingegneria Meccanica e Aerospaziale del Politecnico di Torino e la ZHAW University di Winterthur, Svizzera. Il modello matematico dell'aereo è stato sviluppato in ambiente MATLAB/Simulink e integrato in FlightGear, un simulatore di volo open-source, che consente di visualizzare graficamente il modello 3D del proprio velivolo e di vedere su di esso l'effetto dei comandi dati in input dal pilota tramite un joystick e una pedaliera, in modo da avere una risposta intuitiva e immediatamente visibile all'utente.
Mentre il modello matematico era ed è tuttora quello dell'XV-15, il modello tridimensionale originariamente visualizzato in FlightGear era quello del velivolo Bell Boeing V-22 Osprey, un velivolo con una configurazione simile all'XV-15 e già presente nel database di FlightGear, utilizzato quindi come "placeholder". Tuttavia, questa semplificazione grafica è una limitazione del simulatore di volo sviluppato, in quanto la visualizzazione grafica non rappresentava in modo fedele il velivolo descritto dal modello matematico.
Il principale contributo dell'autore è stato, quindi, la realizzazione del modello grafico tridimensionale dell'XV-15 utilizzando il software Blender, e la successiva implementazione delle animazioni delle superfici mobili, carrello, motori e rotori. Anche il cockpit è stato migliorato, con l'aggiunta di alcuni strumenti utili a visualizzare lo stato del velivolo durante la simulazione. In questo modo il simulatore di volo si presenta ora in modo coerente con ciò che si vuole simulare, e la resa grafica migliorata permette di apprezzare maggiormente il lavoro svolto su questo progetto.

Indice

Capitolo 1

Introduzione

Si introduce, per iniziare, il concetto di "convertiplano" (o "tilt-rotor" in inglese, i termini verranno utilizzati in maniera interscambiabile), per presentare gli argomenti base di questo lavoro, a cui farà seguito una panoramica storica e tecnica sul velivolo chiave della tesi, il Bell XV-15.

1.1 Il convertiplano

La caratteristica che definisce la categoria di velivoli tilt-rotor è la possibilità di combinare le capacità di volo a punto fisso di un elicottero con quelle di volo traslato di un'ala fissa: i convertiplani sono dunque, per definizione, velivoli ibridi, progettati per fondere assieme due mondi profondamente differenti come ala rotante e ala fissa e ottenere un prodotto capace di soddisfare esigenze molto diverse fra loro.
Questa "ibridazione" è raggiunta principalmente mediante la rotazione delle nacelles che ospitano i motori: queste possono essere posizionate a 90° per il decollo, l'hovering e l'atterraggio, e successivamente ruotate per la fase di transizione fra volo a punto fisso e volo traslato, fino a raggiungere una posizione parallela col terreno per volare come una vera e propria ala fissa.

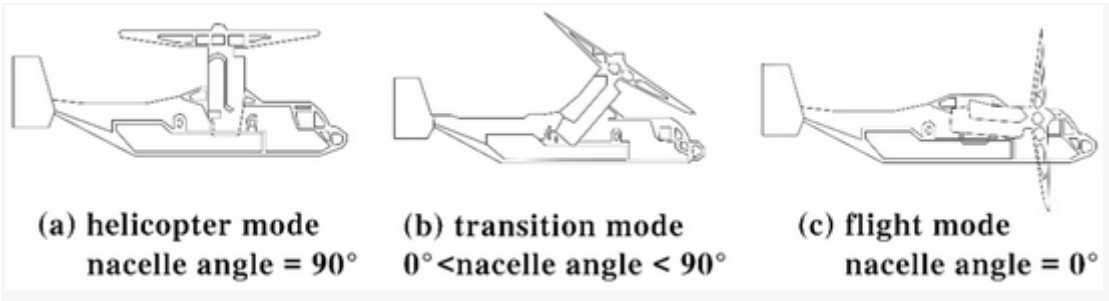

Figura 1.1: Rotazione dei motori (immagine tratta da [10])

L'impianto propulsivo dei tilt-rotor è quasi sempre costituito da due o più motori turboelica, che forniscono la potenza necessaria per tutte le fasi del volo e per i sistemi di bordo. Una prima differenza importante fra un convertiplano e un velivolo turboelica ad ala fissa tradizionale è la dimensione dei rotori, che sono molto più grossi nel tilt-rotor dovendo supportare anche l'hovering, mentre un aereo convenzionale deve solo effettuare

volo traslato.
Proprio le dimensioni dei rotori però impediscono al convertiplano di eseguire un atterraggio rullato convenzionale, limitandone le operazioni di decollo e atterraggio alla modalità elicottero. Un altro problema derivante dall'impianto propulsivo è la complessità meccanica: si deve infatti realizzare un sistema che permetta la rotazione dei motori e il funzionamento degli stessi ai diversi angoli di rotazione. Il sistema deve essere inoltre in grado, in caso di avaria ad uno dei due propulsori, di far ruotare entrambi i rotori con un solo motore funzionante. Questi fattori portano in definitiva ad una notevole complessità meccanica, con conseguenti aumenti di peso, costi e manutenzione.
Grazie alla sua natura ibrida fra ala fissa e ala rotante, la caratteristica certamente più appetibile del convertiplano è la possibilità di combinare l'atterraggio e decollo verticale di un elicottero, che non richiede intere piste ma solamente una piattaforma di atterraggio (helipad), con la più elevata velocità di un aereo turboelica: basti pensare che un elicottero raramente eccede i 300 km/h, mentre per un tilt-rotor la velocità massima si alza oltre i 500 km/h. Questo permette di mantenere la versatilità di impiego di un velivolo ad ala rotante eliminando i suoi due difetti principali, ovvero velocità massima e portata limitate.
Nell'ambito civile, al momento nessun convertiplano è in servizio attivo, ciononostante l'AgustaWestland AW609 (Figura 1.2) si trova nelle fasi finali di testing e dovrebbe entrare in servizio nei prossimi anni: esso si colloca nella fetta di mercato relativa al trasporto executive e sulle piattaforme offshore, ma è anche previsto che possa compiere missioni di ricerca e soccorso e di evacuazione medica.

Figura 1.2: AgustaWestland AW609 (immagine tratta da ilgiornaledellebuonenotizie.it)

Per quanto riguarda il campo militare, ci sono molti progetti per convertiplani futuri, come il Bell V-280 Valor, ma è presente anche un modello in servizio attivo dal 2007: si tratta del Bell Boeing V-22 Osprey (Figura 1.3), che vola sotto le insegne delle varie branche delle forze armate statunitensi (United States Air Force, United States Navy e United States Marine Corps). Questo velivolo ha partecipato con successo a numerosi teatri bellici, tra i quali Iraq, Libia e Afghanistan, dimostrando le sue eccezionali qualità come velivolo da trasporto ed evacuazione medica.

Figura 1.3: Bell Boeing V-22 Osprey (immagine tratta da raytheonintelligenceandspace.com)

1.2 Bell XV-15

A questo punto è doveroso dedicare una sezione al velivolo di cui tratterà questo lavoro, ovvero il Bell XV-15 (Figura 1.4), sia dal punto di vista storico che ingegneristico.

Figura 1.4: Bell XV-15 (immagine tratta da nasa.gov/centers/dryden)

1.2.1 Origine e sviluppo

Il progetto dell'XV-15 affonda le radici nell'Agosto 1950, quando due forze armate americane, United States Air Force e United States Army, formularono le richieste per il "Convertiplane Program": lo scopo era la creazione di un nuovo tipo di velivolo combinante le capacità di un elicottero, che da poco aveva fatto il suo ingresso operativo nel mondo dell'aviazione, con la velocità e la portata di una tradizionale ala fissa. Fra i vari progetti proposti, quello che venne ritenuto di migliori prospettive future fu il Bell XV-3,

che incorporava il concetto di "tilt-rotor" esattamente come inteso al giorno d'oggi: i rotori venivano posizionati verso l'alto per l'hovering e potevano essere ruotati per ottenere un volo traslato come quello di un velivolo tradizionale.
L'esperienza con l'XV-3 rese subito chiaro come realizzare un convertiplano non sarebbe stato un compito facile: durante i test sorsero numerosi problemi legati all'instabilità aeroelastica, alla resistenza a fatica di alcuni componenti (specialmente della trasmissione, che doveva trasferire la potenza dal motore situato in fusoliera fino alla punta delle ali dove erano presenti i rotori), alla scarsa potenza dell'impianto propulsivo e alle pessima qualità del volo in generale. Tuttavia, l'XV-3 dimostrò con successo una capacità fondamentale, la fattibilità della conversione tra elicottero e aereo (e viceversa) in volo: grazie a questo, la Bell ritenne che il concetto di tilt-rotor dovesse essere esplorato ulteriormente.
Dopo una serie di progetti intermedi, nel 1973 la NASA e lo United States Army fornirono alla Bell i fondi per un progetto di dimostratore tecnologico: nel 1977, dopo quattro anni di studi e progettazione, vennero finalmente costruiti i due prototipi dell'XV-15. Il primo volo avvenne il 3 Maggio 1977, con ai comandi il pilota Ron Erhart e il copilota Dorman Cannon. Da quel punto in poi i due velivoli vennero fatti volare e testati in maniera estensiva; uno dei due partecipò anche al Paris Air Show del 1981, la prima esibizione pubblica dell'XV-15.
Il 20 Agosto 1990 uno dei due prototipi si schiantò al suolo in fase di atterraggio: entrambi i piloti riportarono fortunatamente solo lesioni minori, tuttavia il velivolo era troppo danneggiato per poter essere riparato, dunque venne recuperato il cockpit e trasformato in un simulatore di volo.
Il velivolo rimanente continuò ad essere usato dalla NASA per condurre test di volo, fino al suo ritiro nel 2003, fornendo un'importante base ingegneristica per lo sviluppo del V-22 Osprey, l'unico convertiplano attualmente in servizio al mondo, e dell'AW609, ancora in sviluppo.

1.2.2 Impianto propulsivo

L'impianto propulsivo dell'XV-15 è costituito da due motori a turbina Lycoming T53-LTC1K-4K, ciascuno erogante una potenza massima al decollo pari a 1550 shp (limitata a 1250 shp per l'uso continuo). I motori sono posizionati sulla punta delle ali e ognuno muove il rispettivo rotore: questo permette, rispetto all'XV-3 che ha il motore in fusoliera, di ridurre la lunghezza dell'albero di trasmissione. I due motori sono comunque collegati fra loro di modo che un singolo motore possa muovere entrambi i rotori in caso di guasto.
Ciascun rotore è tri-pala con un diametro di 7.62 m, mentre l'apertura alare è di 10.67 m. Entrambi i motori possono ovviamente ruotare di 90° per passare dalla modalità elicottero alla modalità aereo: la rotazione completa impiega dai 10 ai 15 secondi. I motori possono essere ruotati di ulteriori 5° all'indietro (per un totale di 95°) per la manovra di autorotazione.

1.2.3 Comandi di volo

L'XV-15 si pilota con classici comandi di ciclico, collettivo e pedaliera, sia in modalità elicottero che in modalità aereo.
In modalità elicottero, il velivolo si controlla esattamente come un'ala rotante: agendo sulla leva del collettivo aumenta il passo di tutte le pale contemporaneamente, mentre agendo sulla leva del ciclico varia il passo locale della pala permettendo di inclinare il

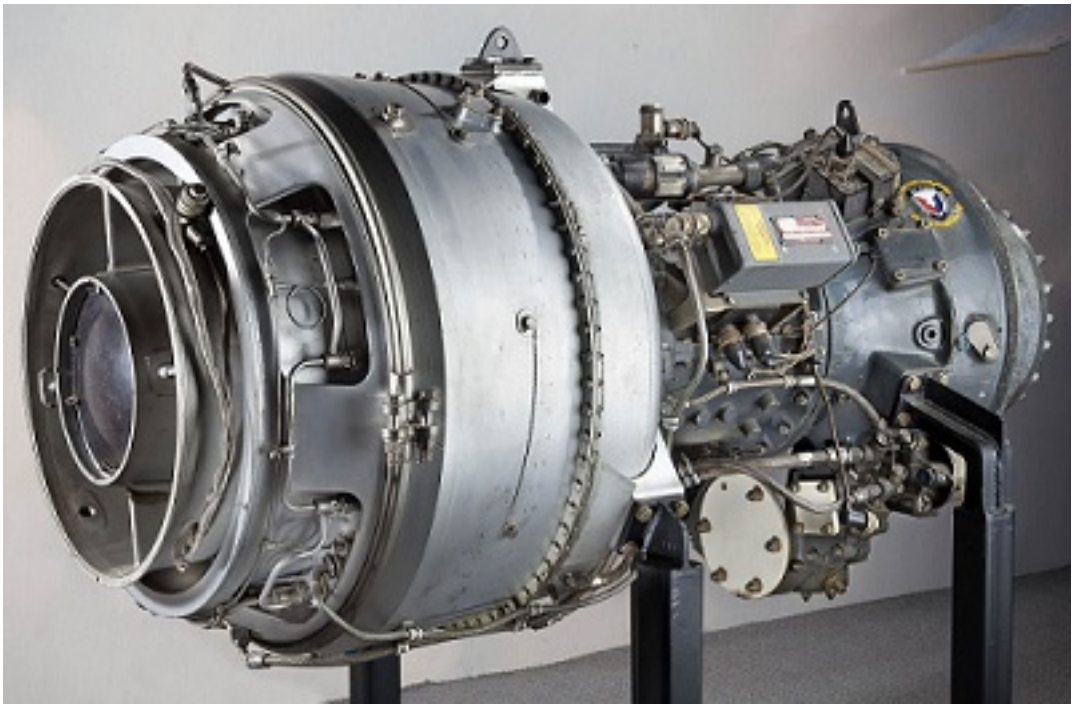

Figura 1.5: Lycoming T53-LTC1K-4K (immagine tratta da airandspace.si.edu)

disco rotore e direzionare dunque la portanza, specialmente per il beccheggio; il rollio è ottenuto per collettivo differenziale mentre l'imbardata è ottenuta per ciclico differenziale. In modalità aeroplano la generazione della portanza non è più affidata ai rotori, bensì all'ala, così come i momenti di controllo passano alle superfici mobili. Questo processo di "trasferimento" avviene automaticamente durante la rotazione dei motori da 90° a 0°. A questo punto la leva del collettivo agisce come la manetta di un normale velivolo ad ala fissa.
L'XV-15 possiede infine uno Stability and Control Augmentation System (SCAS) per i tre assi, che migliora le qualità di volo del velivolo.

Capitolo 2

Simulazione di volo dell'XV-15: stato dell'arte

In questo capitolo si esporrà lo stato dell'arte della simulazione del volo dei convertiplani e, soprattutto, dell'XV-15. Quello su cui l'autore ha lavorato non è infatti l'unico simulatore di volo dell'XV-15, ma ne esistono anche altri, sempre realizzati per scopi didattici e di ricerca. Questo particolare simulatore ha però il vantaggio di essere portatile e computazionalmente "semplice": la simulazione può essere effettuata anche su hardware non particolarmente potenti e performanti, come un comune laptop, e può essere dunque utilizzato da un ampio numero di ricercatori e studenti, senza andare a inficiare sulla bontà e sull'accuratezza della simulazione stessa.

2.1 Simulazione del volo: in generale

Con l'avanzamento della tecnologia e lo sviluppo di computer con potenze di calcolo sempre maggiori, la simulazione del volo è diventata di fondamentale importanza: essa non è infatti solamente utilizzata per scopi di ricerca, ma in una moltitudine di campi in continua espansione. Si pensi come esempio all'ambito ludico: sono ormai numerosi i simulatori di volo disponibili commercialmente, che propongono un'alternativa più "realistica" ai classici videogiochi arcade.
Tralasciando questa particolare applicazione, è innegabile la centralità dei simulatori di volo nel moderno mondo aerospaziale: l'addestramento dei piloti civili e militari comprende numerose ore nei simulatori di volo a terra, con un enorme abbassamento dei costi rispetto ad un volo vero e proprio e ovviamente senza i rischi intrisechi collegati ad esso. Nonostante i simulatori non possano riprodurre al 100% l'esperienza che un volo reale offre, essi rimangono un contributo fondamentale di cui nessuna compagnia aerea si priva per l'addestramento del proprio personale.

Figura 2.1: Cockpit del simulatore di volo dell'Airbus A320 in dotazione a Lufthansa (immagine tratta da lufthansa-aviation-training.com)

Come già scritto precedentemente, infine, la simulazione del volo è estremamente importante anche nella didattica e nella ricerca. La realizzazione di un simulatore di volo di un velivolo permette di esplorarne a fondo le caratteristiche e quindi di poter avere un'iniziale validazione di nuovi concetti o design senza ricorrere a costosa sperimentazione "fisica"; con un simulatore sufficientemente avanzato si possono inoltre riprodurre condizioni di volo che sarebbe pericoloso testare realmente, come ad esempio lo stallo oppure la perdita di un motore. Tutto ciò potenzia la ricerca e lo sviluppo di velivoli sempre più avanzati, performanti e sicuri.

2.1.1 Introduzione a FlightGear

FlightGear è un simulatore di volo open-source che permette di riprodurre un ambiente virtuale in cui poter inserire velivoli già presenti nel suo database oppure velivoli creati dall'utente. Esso si basa sulla libreria JSBSim, scritta in C++, che fornisce il modello di volo dei velivoli; questi ultimi vengono poi definiti in dei file XML, che servono a configurare le proprietà aerodinamiche, dei comandi di volo e di massa. Per quanto riguarda i modelli grafici, l'estensione più comunenemente utilizzata è ".ac". La sua natura open-source lo rende un software potente e allo stesso tempo fruibile in maniera semplice, in quanto chiunque può contribuire non solo al suo sviluppo ma anche aggiungere i propri progetti di simulazione personali, sia a scopo ricreativo che di ricerca. Al momento nel database di FlighGear sono presenti oltre 500 velivoli, modellati con diversi gradi di fedeltà dalla community degli utenti.
Un'altra feature importante riguarda i dati morfologici e di elevazione del terreno, ottenuti tramite i dati della Shuttle Radar Topography Mission (SRTM); sono presenti inoltre numerosi aeroporti e piste di tutto il mondo, oltre a strade, ferrovie, laghi e così via. Questo garantisce un grado maggiore di fedeltà della simulazione, potendo volare virtualmente nel mondo reale e non in scenari creati ad-hoc.
Maggiori dettagli riguardo a questo software verranno forniti in Sezione 3.2.

Figura 2.2: Modello del Concorde presente in FlightGear (immagine tratta da [13])

2.2 Simulazione di volo dell'XV-15

2.2.1 Simulazioni di volo eseguite dalla NASA

Il programma XV-15, come già riportato nell'introduzione, iniziò nel 1973, con il primo volo nel Maggio del 1977; già dagli inizi del progetto vennero però svolte simulazioni di volo presso l'Ames Research Center di Mountain View, California. In particolare, prima del 1977, vennero effettuate quattro simulazioni complete con il Flight Simulator for Advanced Aircraft (FSAA) e una simulazione ristretta al volo a punto fisso con il 6-DOF Simulator. Dopo che l'XV-15 ebbe effettuato il primo volo, vennero svolte ulteriori tre simulazioni con l'FSAA e una con il Vertical Motion Simulator (VMS). Il numero dei test potrebbe sembrare basso, ma bisogna ricordare che in quegli anni per effettuare una sola simulazione erano necessarie settimane o addirittura mesi. Si evince dunque quanto fosse centrale la simulazione del volo per validare il progetto e risolvere problemi prima che si verificassero in un test di volo, con conseguenze ben più gravi.
Il modello matematico era particolarmente importante e avanzato, essendo un modello non lineare del velivolo comprendente aerodinamica del velivolo e dei rotori, comandi di volo, sistemi di attuazione e carrello. I simulatori di volo impiegati furono dunque tre: l'FSAA, il 6-DOF e il VMS. Nonostante gli scopi diversi, essi utilizzavano la stessa architettura generale, ovvero il cockpit per il pilota con la relativa strumentazione e comandi, schermi in cui veniva riprodotto l'ambiente virtuale di simulazione (ad esempio un aeroporto o una portaerei), sistemi di attuazione e, ultimo ma non per importanza, il computer per la simulazione (XDS Sigma 8); per passare da un tipo all'altro bastava semplicemente modificare la strumentazione nel cockpit secondo necessità e inserire nel computer il modello matematico desiderato fra i tre elencati prima. Ognuno di questi modelli era realizzato per uno scopo differente:

- l'FSAA era il modello più generale, comprendente tutto l'inviluppo di volo del velivolo
- il 6-DOF serviva per la simulazione e valutazione del volo a punto fisso
- il VMS era usato per migliorare la manovrabilità del velivolo e la risposta ai comandi

In conclusione, la simulazione del volo svolta dalla NASA è stata di fondamentale importanza per il progetto, non solo perchè ha permesso di risolvere numerosi problemi relativi al velivolo ma anche perchè ha fornito una considerevole base di dati sia per gli ingegneri impiegati nel progetto dell'XV-15 che per i ricercatori impegnati nello studio delle caratteristiche di volo di questa categoria di velivoli.

2.2.2 Generic Tilt Rotor Simulation (GTRS)

Un lavoro fondamentale per quanto riguarda la simulazione del volo dei convertiplani è stato svolto da J. S. G. McVicar nel 1993, con lo sviluppo del Generic Tilt Rotor Simulation (GTRS). Inoltrarsi in esso esula dallo scopo di questa tesi, tuttavia è doveroso quantomeno riassumere di cosa si tratta per l'importanza che esso ha avuto e ha tuttora, anche nel simulatore su cui l'autore del presente documento ha lavorato.
Il GTRS si basa sul modello matematico di simulazione GTILT (Generic TILT-rotor), ovvero un modello sviluppato sempre da McVicar che simula la pala individuale del rotore in movimento, potendo calcolare su di essa forze e momenti; oltre a questo è anche in grado di calcolare forze e momenti sull'ala, fusoliera e superfici di controllo. Un'altra parte importante è l'algoritmo predittivo del trim: essendo le equazioni che compongono questo modello non lineari e periodiche, anche lo stato del velivolo trimmato varierà periodicamente, al contrario di un modello che utilizza equazioni stazionarie o quasi-stazionarie in cui non si riscontrerebbe questa periodicità.
Lo scopo di questo modello è principalmente essere usato come strumento di progetto per convertiplani. Il GTRS non è dunque limitato solo alla ricerca ma anche alla progettazione: è uno strumento estremamente potente che permette di simulare diverse configurazioni in base ai dati che gli vengono forniti, ad esempio quelli dell'XV-15, usati proprio da McVicar dello sviluppo del modello GTILT. Il limite di questa simulazione è quindi legato alla disponibilità o meno di dati attendibili per il velivolo da simulare.

2.3 Simulazione di volo del V-22 Osprey

Come esempio di simulazione del volo di convertiplani è doveroso citare brevemente quanto fatto per il Bell Boeing V-22 Osprey, l'unico velivolo di questo genere in servizio attivo al mondo.
Come per l'XV-15, che può considerarsi il precursore dell'Osprey per quanto riguarda la tecnologia impiegata, i progettisti del V-22 hanno fatto ampio uso di simulazioni anche nella fase di design preliminare. Di fatto, i simulatori utilizzati sono gli stessi riportati nella Sezione 2.2.1, con le dovute modifiche relative ad esempio alla disposizione degli strumenti nel cockpit; solo per questa fase sono state fatte più di 300 ore di simulazione di volo.
Descrivere nuovamente i simulatori di volo sarebbe ridondante, ma è interessante invece riportare quanto fatto per il simulatore di volo specifico per le operazioni navali, denominato Manned Flight Simulator (MFS), del V-22 presente a Patuxent River (Maryland):

la prima forza armata in cui l'Osprey è entrato in servizio è stata infatti il corpo dei Marines americano, che impiegano tuttora numerose navi per compiere le loro missioni, principalmente operazioni anfibie (che vedono le unità navali usate in collaborazione con unità terrestri e aeree). L'ambiente marino e il dover operare sulle navi pongono una serie di problemi aggiuntivi rispetto alle operazioni terrestri, che devono quindi essere riportati e modellati nel simulatore di volo per ottenere la maggiore fedeltà possibile e per certificare lo stesso come uno strumento utile al design del velivolo e all'addestramento dei piloti.
Per fare alcuni esempi, è stato necessario modellare matematicamente la scia causata dal moto del vascello, la turbolenza prodotta dalle strutture presenti sulla nave (come la cosiddetta "isola" dove sono situati fra le altre cose il ponte di comando e la torre di controllo) e come il vento interagisce con il campo di moto dell'aria sopra il ponte di volo. La fase di "hovering" è inoltre particolarmente critica, in quanto non è un vero e proprio volo a punto fisso: il velivolo è sì fermo rispetto alla nave, ma questa naviga a oltre 20-30 nodi, rendendo di fatto la manovra di appontaggio per un elicottero o convertiplano eseguita in volo traslato, seppur a velocità non molto elevate. Si può concludere dunque che un simulatore deve adeguarsi alle richieste dell'utente, poichè ambienti differenti richiedono approcci differenti e pongono ostacoli diversi ad una simulazione fedele.

Figura 2.3: Un V-22 dei Marines atterra sulla nave USNS Mercy (immagine tratta da navytimes.com)

Capitolo 3

Simulatore di volo

Questo capitolo sarà dedicato alla descrizione generale della struttura e del funzionamento del simulatore di volo su cui l'autore ha lavorato. Inoltrarsi nei dettagli non è lo scopo di questa tesi poichè questa parte non è stata realizzata dall'autore ma si basa sui lavori precedenti a questo, tuttavia verrà fornita una spiegazione che permetta di comprendere come le varie parti di questo progetto interagiscono fra di loro per formare un prodotto funzionante.

3.1 MATLAB/Simulink

In questa sezione si presenterà il modello MATLAB/Simulink dell'XV-15 realizzato da ZHAW e dal Politecnico di Torino: essendo stato sviluppato principalmente in Simulink, a meno che non scritto esplicitamente altro ci si riferirà sempre a questo software, che di fatto è un'estensione di MATLAB.

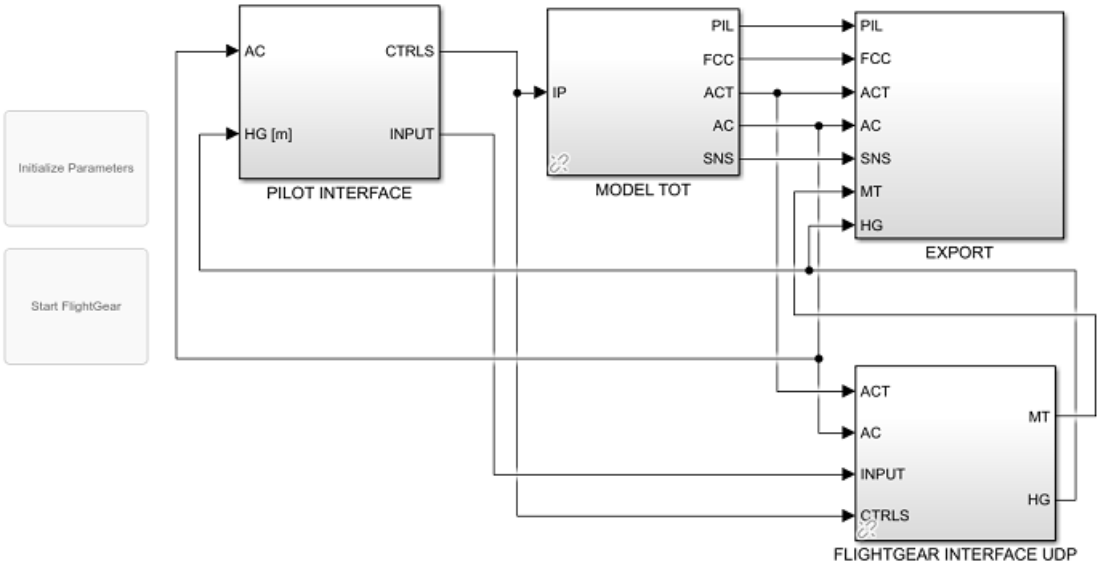

Figura 3.1: Architettura generale del modello Simulink

In Figura 3.1 è riportata l'architettura a livello più alto del modello Simulink. Si nota come ci siano quattro blocchi principali: PILOT INTERFACE, MODEL TOT, EXPORT e FLIGHTGEAR INTERFACE UDP.

3.1.1 PILOT INTERFACE

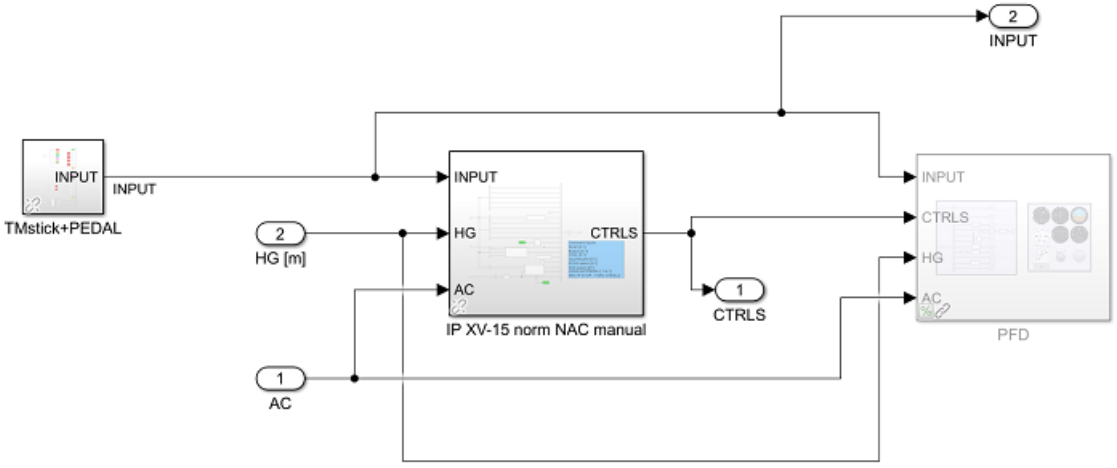

Figura 3.2: Blocco PILOT INTERFACE

Il blocco PILOT INTERFACE riceve gli input digitali (pressione di un bottone) e analogici (comandi di ciclico, collettivo e pedali) e li fornisce al modello matematico.

TMStick+PEDAL

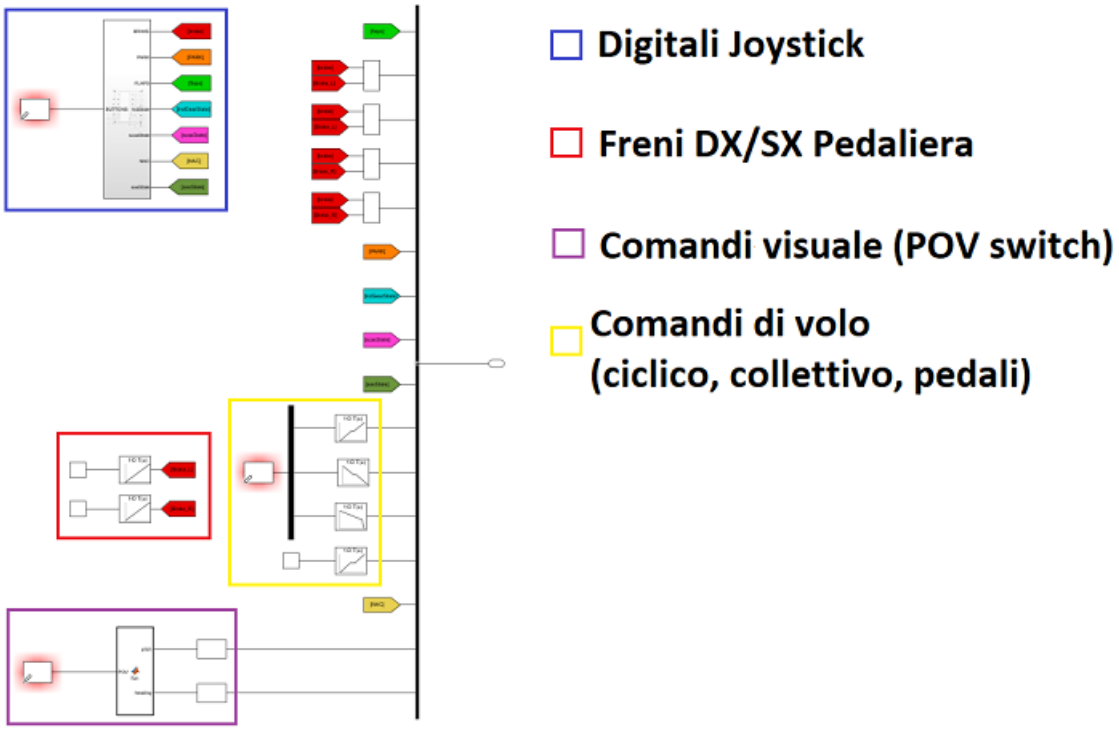

Figura 3.3: Sottoblocco TMStick+PEDAL

Questo sottoblocco (Figura 3.3) serve per ricevere gli input del pilota e per raggrupparli in modo da poterli trasferire dove servono. In particolare c'è una parte che si occupa di ricevere i segnali digitali del joystick (flap su/giù, parking brake e così via), una parte che gestisce i freni (assegnati alla pedaliera), una che gestisce la movimentazione della visuale interna ed esterna del velivolo (POV switch) e infine una che gestisce i comandi di volo, provenienti sia dal joystick (ciclico e collettivo) che dalla pedaliera (imbardata). Ogni tipo di segnale va necessariamente maneggiato in maniera differente, poichè non si può considerare un analogico come un digitale e trattarlo come tale, tuttavia, dopo aver subito alcune modifiche, tutti i segnali di input vengono convogliati nello stesso bus, in modo da facilitare l'elaborazione degli stessi nelle altre parti del modello. Se nei blocchi successivi non servono tutti i segnali ma solo alcuni, basta solamente inserire un selettore per far passare quelli richiesti.
Nella parte di questo sottoblocco che riceve i comandi digitali del joystick è presente anche la logica di comando, che verrà analizzata in seguito.

IP XV-15 norm NAC manual

In questo sottoblocco (evidenziato in Figura 3.4) gli input vengono normalizzati allo scopo di semplificare l'implementazione e permettere al modello di funzionare, con solo qualche modifica minore da effettuare, anche nel caso vengano cambiate le periferiche.
In particolare, gli input di ciclico e di pedale, come scritto nel riquadro blu della Figura 3.4, sono normalizzati in modo da avere un segnale con un'escursione che va da -1 a 1. La leva del collettivo produce invece un segnale che va da 0 (leva completamente abbassata) a 1 (leva completamente alzata).
Altri segnali possono assumere solo valori binari, ovvero il parking brake, il carrello e gli interruttori di SAS e SCAS; questo perchè possono trovarsi solamente in una di due situazioni, ad esempio il carrello può essere solamente abbassato o retratto.
Infine, i segnali relativi alla posizione di flap e gondole motrici sono particolari perchè i primi assumono quattro valori discreti fra 0 e 1 mentre i secondi ne assumono tre: questo perchè questi comandi si riferiscono a componenti che hanno diverse posizioni in cui possono trovarsi, ad esempio le gondole motrici possono essere posizionate a 90° (verticali), inclinate di 60° (per la transizione tra volo a punto fisso e volo traslato e viceversa) o a 0° (orizzontali).

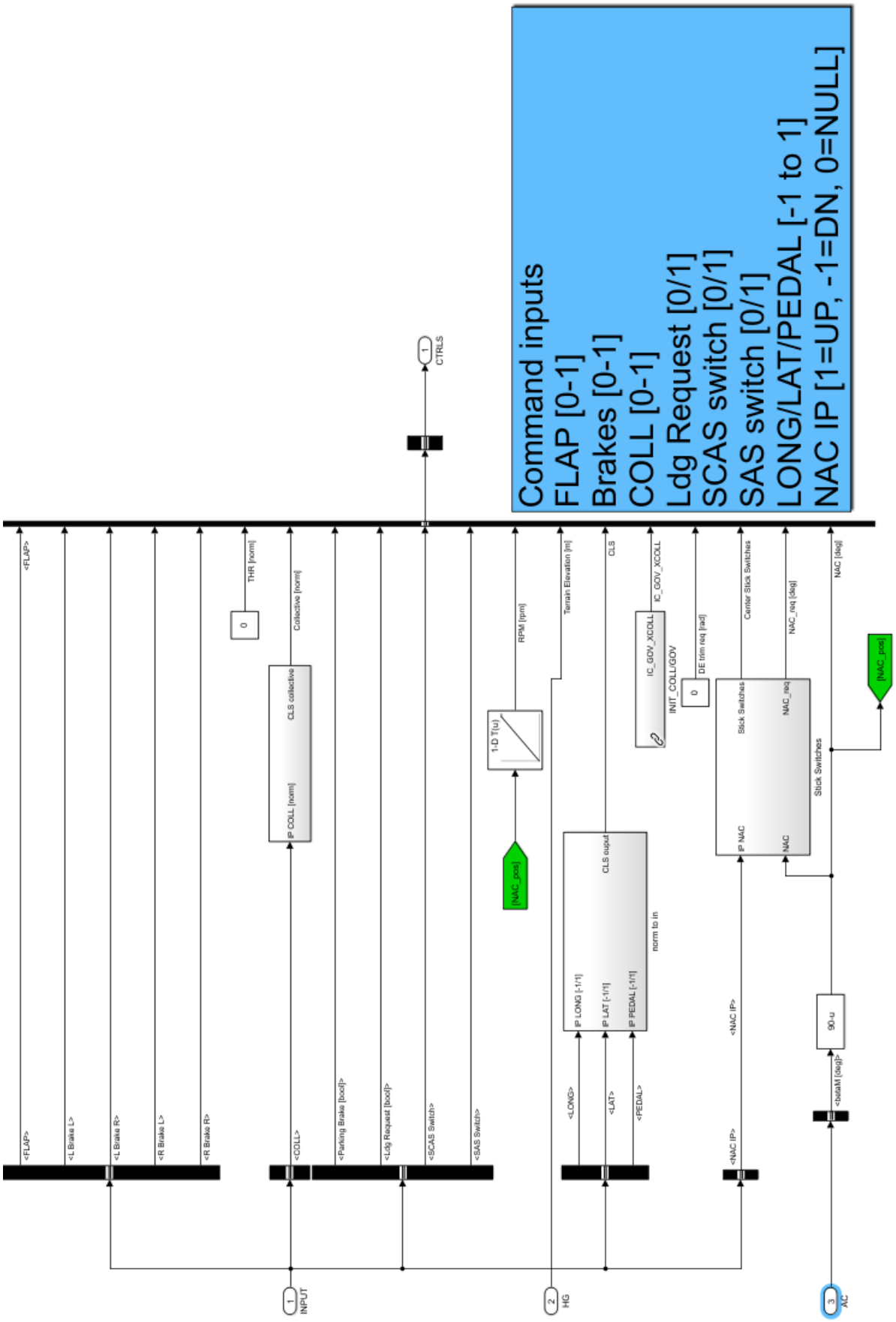

Figura 3.4: Sottoblocco IP XV-15 norm NAC manual

3.1.2 MODEL TOT

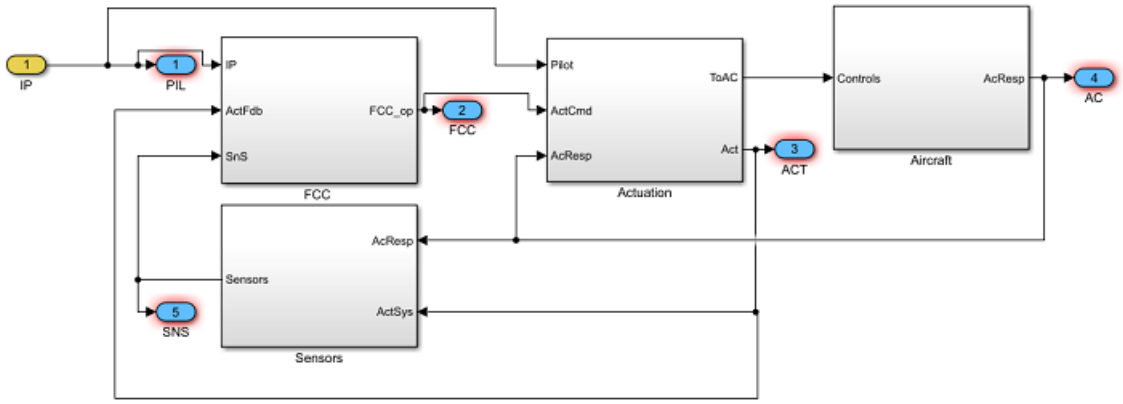

Figura 3.5: Blocco MODEL TOT

Nel blocco MODEL TOT si trova il modello matematico del velivolo: in esso è contenuta la modellizzazione relativa al Flight Control Computer (FCC), agli attuatori, alla dinamica del velivolo e quella dei sensori; può essere considerato come il "cervello" della simulazione, in quanto riceve gli input dal blocco PILOT INTERFACE e li elabora per ottenere lo stato del velivolo, il quale viene poi inviato sia ai blocchi EXPORT e FLIGHTGEAR INTERFACE UDP che nuovamente al blocco PILOT INTERFACE, in modo da avere un anello chiuso in cui, partendo da un certo stato "iniziale", il simulatore riceve gli input del pilota e li elabora per ricavare l'effetto che hanno sullo stato del velivolo e lo aggiorna di conseguenza, mandandolo come feedback a PILOT INTERFACE e assumendo il ruolo di nuovo stato iniziale, per poi ricominciare il ciclo. Questo processo avviene continuamente per assicurare una simulazione in tempo reale dell'XV-15.

FCC (Flight Control Computer)

Il sottoblocco FCC (Figura 3.6) include diversi elementi per il controllo del velivolo; di fatto, lo scopo dell'FCC è quello di assicurare che il velivolo compia la manovra desiderata (controllando ad esempio la deflessione delle superfici di controllo) riducendo al minimo le oscillazioni. Questi elementi sono lo Stability Augmentation System (SAS), che filtra i comandi del pilota per ridurre le oscillazioni, lo Stability Control and Augmentation System (SCAS), che fornisce un ulteriore miglioramento della manovrabilità del velivolo tramite un'azione di tracking delle velocità angolari sui tre assi, il sistema di controllo delle gondole motore (NAC) e il Collective Governor, che varia la potenza del motore in base all'angolo delle gondole motore e alla posizione della leva del collettivo per mantenere costante la velocità di rotazione dei rotori. Il blocco MIXER ha il compito di eseguire il mixing dei comandi del pilota in base all'angolo delle gondole motore, questo perchè per eseguire una certa manovra sono necessarie deflessioni di superfici mobili e valori di passo ciclico e collettivo dei rotori diversi a seconda della posizione in cui si trovano i motori.

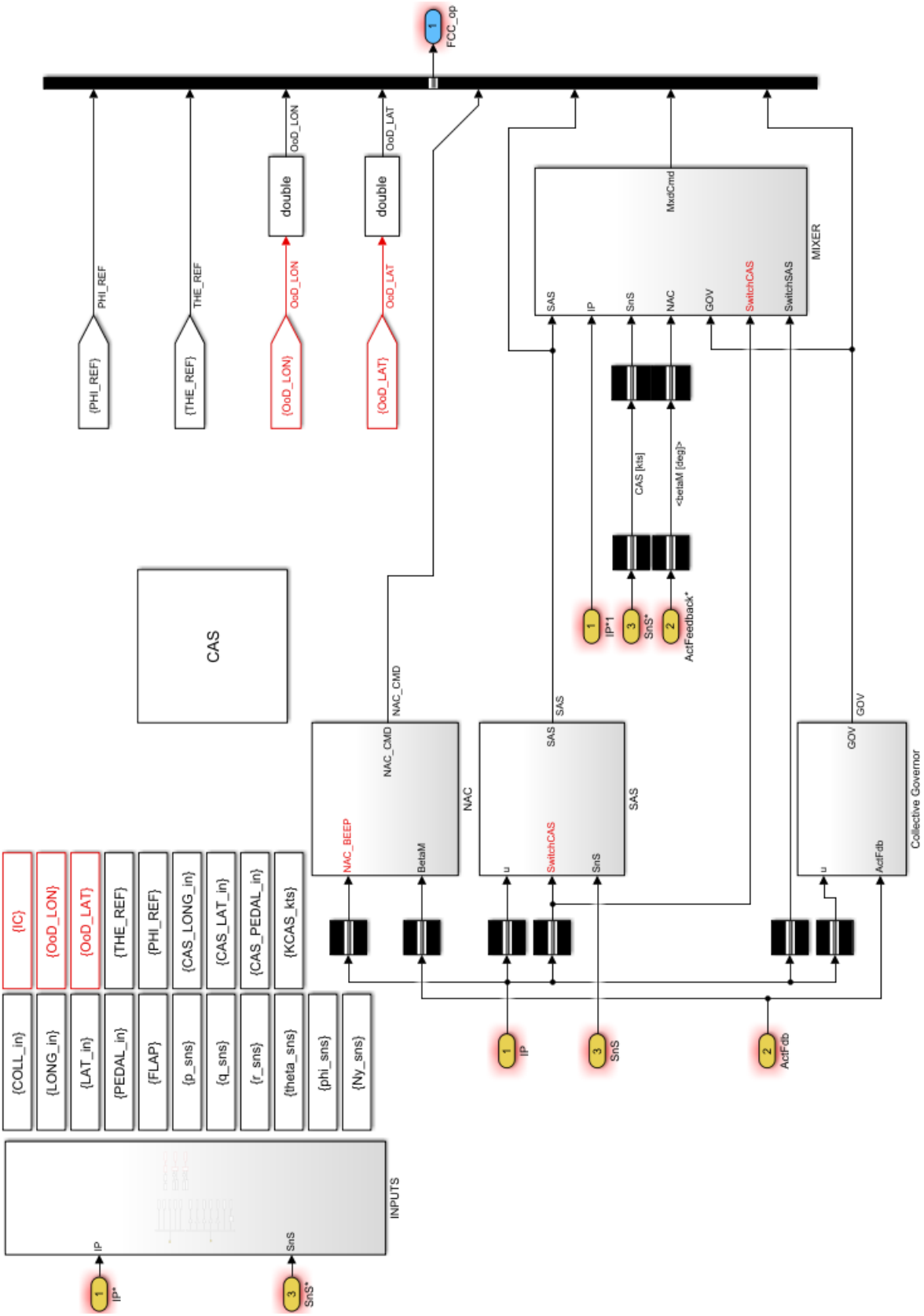

Figura 3.6: Sottoblocco FCC

Actuation

Questo sottoblocco si occupa di definire la dinamica dei sistemi di attuazione del velivolo: essi non agiscono instantaneamente ma hanno un certo ritardo, che è stato modellizzato tramite funzioni di trasferimento del primo ordine.
Le superfici mobili, il carrello, il piatto oscillante dei rotori e il sistema di rotazione dei motori sono stati tutti modellizzati con funzioni del primo ordine: questo non tiene conto delle non-linearità intrinsiche di questi sistemi, tuttavia è stato scelto questo metodo per ridurre il costo computazionale della simulazione mantenendo comunque un alto grado di fedeltà.

Aircraft

In questo sottoblocco è contenuta la dinamica dell'XV-15 e la risposta a forza e momenti prodotti dall'esterno. Per prima cosa è necessario considerare le condizioni ambientali in cui il velivolo si trova perchè forze e momenti sopracitati variano in base a pressione, temperatura e densità dell'aria.
Una forza che agisce sempre sul velivolo è la gravità, quindi la forza peso è calcolata in base alla massa del velivolo. Un altro fattore importante è il calcolo della posizione del baricentro perchè durante la conversione esso si sposta longitudinalmente.
Fondamentale è il contributo di forze e momenti aerodinamici, calcolati per le superfici portanti, di controllo ma anche per il carrello, che quando è abbassato produce una resistenza aerodinamica non trascurabile; sempre riguardo al carrello, è modellata anche la resistenza delle ruote quando il velivolo è a terra, per simulare l'attrito col terreno.
Il contributo dei rotori è stato inserito mediante una formulazione multipala dei carichi aerodinamici e equazioni della dinamica di flappeggio del secondo ordine, oltre ad un modello per l'influsso del rotore e della scia prodotta da esso che interagisce e modifica il campo di moto del flusso attorno al velivolo.
Infine, è inclusa anche la spinta prodotta dai motori, sia per il flusso accelerato dalle pale che per quello in uscita dagli ugelli di scarico.
Integrando tutti questi contributi nelle equazioni del moto si ottiene la risposta del velivolo ai comandi del pilota in termini di velocità e accelerazioni, angolari e lineari.

Sensors

Questo sottoblocco contiene i modelli dinamici dei sensori che effettuano le misurazioni fornite al pilota (ad esempio la velocità del velivolo rispetto all'aria viene misurata e riportata nell'anemometro nel cockpit) e inviate come feedback all'FCC per verificare la corretta esecuzione dei comandi relativi alla manovra desiderata. Anche i sensori, come i sistemi di attuazione, hanno una propria dinamica che inserisce un certo ritardo nella misurazione: tale ritardo è stato modellizzato con funzioni di trasferimento del primo ordine.

3.1.3 EXPORT

Figura 3.7: Blocco EXPORT

Il blocco EXPORT ha lo scopo di salvare i dati della simulazione nell'ambiente MATLAB: in particolare, come si vede nella Figura 3.7, tutti i dati relativi all'XV-15 vengono esportati in una struct chiamata "simout" la quale può essere poi indagata per vedere i valori delle varie grandezze allo scopo di controllare la bontà dei dati ottenuti oppure realizzare grafici per visualizzare l'andamento delle variabili.
La simulazione può essere effettuata in due modi: si possono dare i comandi in tempo reale tramite joystick e pedaliera oppure preconfigurare i valori dei comandi ai vari istanti di tempo per, ad esempio, far seguire al velivolo una determinata traiettoria. La seconda strategia è particolarmente utile se si vogliono controllare successivamente i valori che le variabili hanno assunto durante la simulazione poichè gli input sono precisi sia come valore che come tempo a cui vengono dati, cosa che necessariamente diventa più complicata se il compito viene affidato al pilota il quale non può umanamente replicare la precisione e la puntualità di un computer.

3.1.4 FLIGHTGEAR INTERFACE UDP

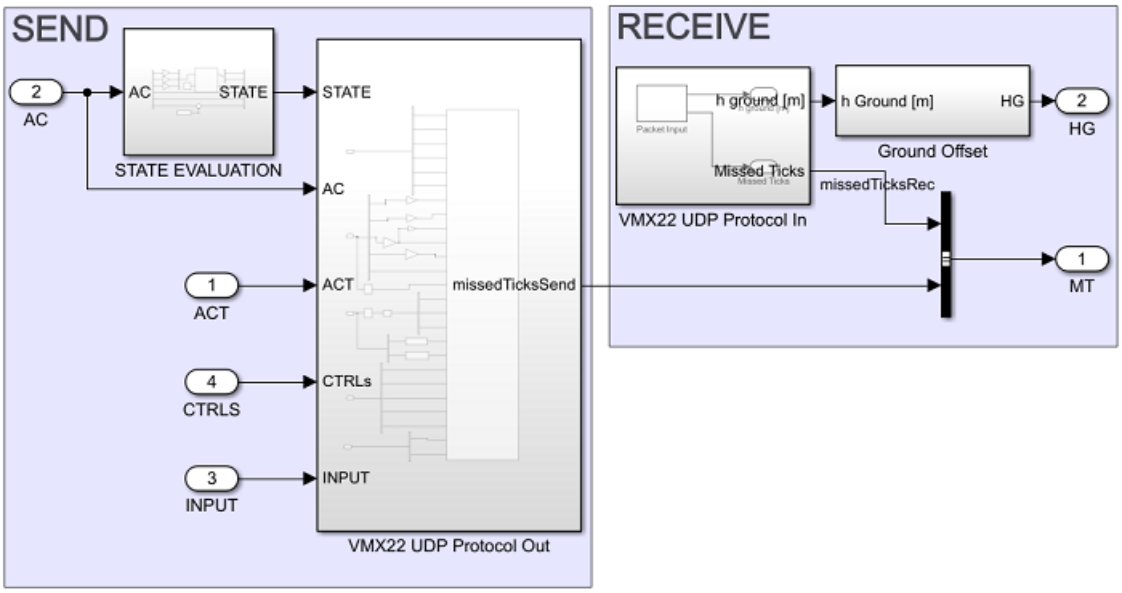

Figura 3.8: Blocco FLIGHTGEAR INTERFACE UDP

Il blocco FLIGHTGEAR INTERFACE UDP ha il compito, come suggerisce il nome, di interfacciarsi con FlightGear: in sostanza è la parte del modello che comunica con FlightGear inviandogli i valori delle variabili in modo che questo possa interpretarle e mostrare il velivolo nel proprio ambiente grafico tridimensionale. In base a ciò che Simulink invia a FlightGear quest'ultimo farà eseguire certe azioni ai modelli 3D caricati tramite i file XML, ad esempio finchè la variabile che controlla l'azionamento del carrello vale 1 questo rimane abbassato, ma quando l'utente preme il pulsante per la retrazione la variabile assume il valore 0. Questo ha due conseguenze principali: per quanto riguarda Simulink, il modello matematico del velivolo considera il carrello retratto (con modifiche ai valori, ad esempio, delle forze aerodinamiche agenti su di esso), e per quanto riguarda FlightGear viene avviata l'animazione della retrazione. Se lo stesso pulsante viene premuto di nuovo si avrà l'animazione dell'abbassamento del carrello, essenzialmente quella della retrazione ma al contrario. Maggiori dettagli riguardo le animazioni verranno forniti nel Capitolo 4, in quanto esse sono state uno degli argomenti principali su cui l'autore ha lavorato.
Si vede come il blocco sia composto da una parte SEND, che si occupa di inviare i segnali a FlightGear, e da una parte RECEIVE, che elabora invece i segnali ricevuti da FlightGear.

SEND

Il primo componente è STATE EVALUATION (Figura 3.9), in cui viene calcolata la posizione del velivolo nello spazio tridimensionale (latitudine, longitudine e quota sul livello del mare) per poi essere inviata a FlightGear. Il blocco riquadrato in blu è particolarmente importante perchè trasferisce la posizione da un sistema di riferimento "Flat-Earth" in uno che considera la curvatura del globo terrestre.

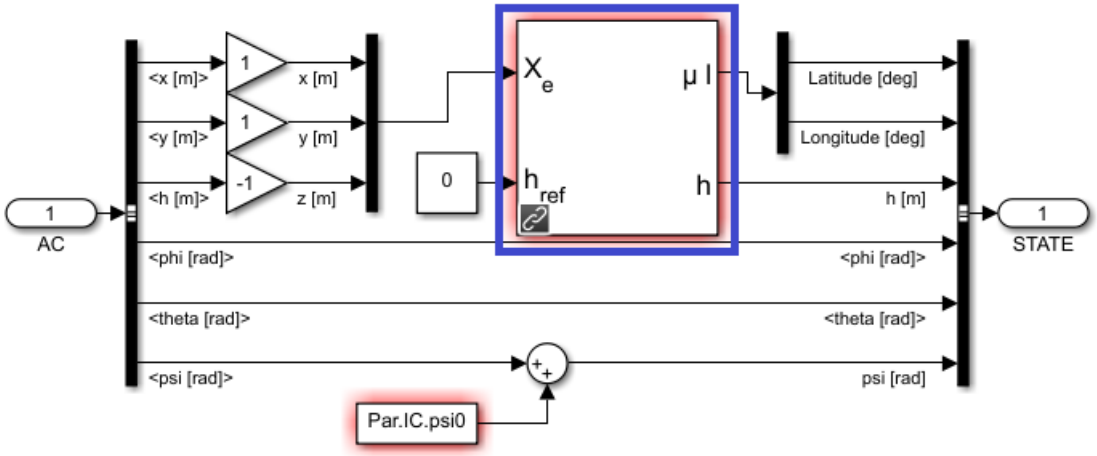

Figura 3.9: Sottoblocco STATE EVALUATION

Nel sottoblocco UDP PROTOCOL OUT (Figura 3.10) i dati arrivano da varie parti del modello e vengono passati al Packet Output, il quale li codifica secondo un determinato protocollo. In particolare, il protocollo utilizzato è UDP (Universal Datagram Protocol), che ha il vantaggio di essere facile da implementare e soprattutto veloce nella trasmissione dei dati, fattore di primo piano in una simulazione real-time come questa.
Nella definizione del Packet Output, tre elementi sono di fondamentale importanza:

- il pacchetto deve avere tante porte di input quante sono le variabili in ingresso (in questo caso 25)
- la dimensione e il tipo di dati dell'output del pacchetto devono essere coerenti con l'input (in questo caso la dimensione è 25x8, ovvero 25 byte, e il tipo di variabile è "double")
- l'ordine con cui sono collegati al pacchetto gli input deve essere lo stesso di quello definito nel file XML del protocollo lato FlightGear

Se una qualsiasi di queste condizioni non venisse rispettata la simulazione non partirebbe, poichè FlightGear riceverebbe dati errati che non saprebbe come elaborare e restituirebbe dunque un messaggio di errore.

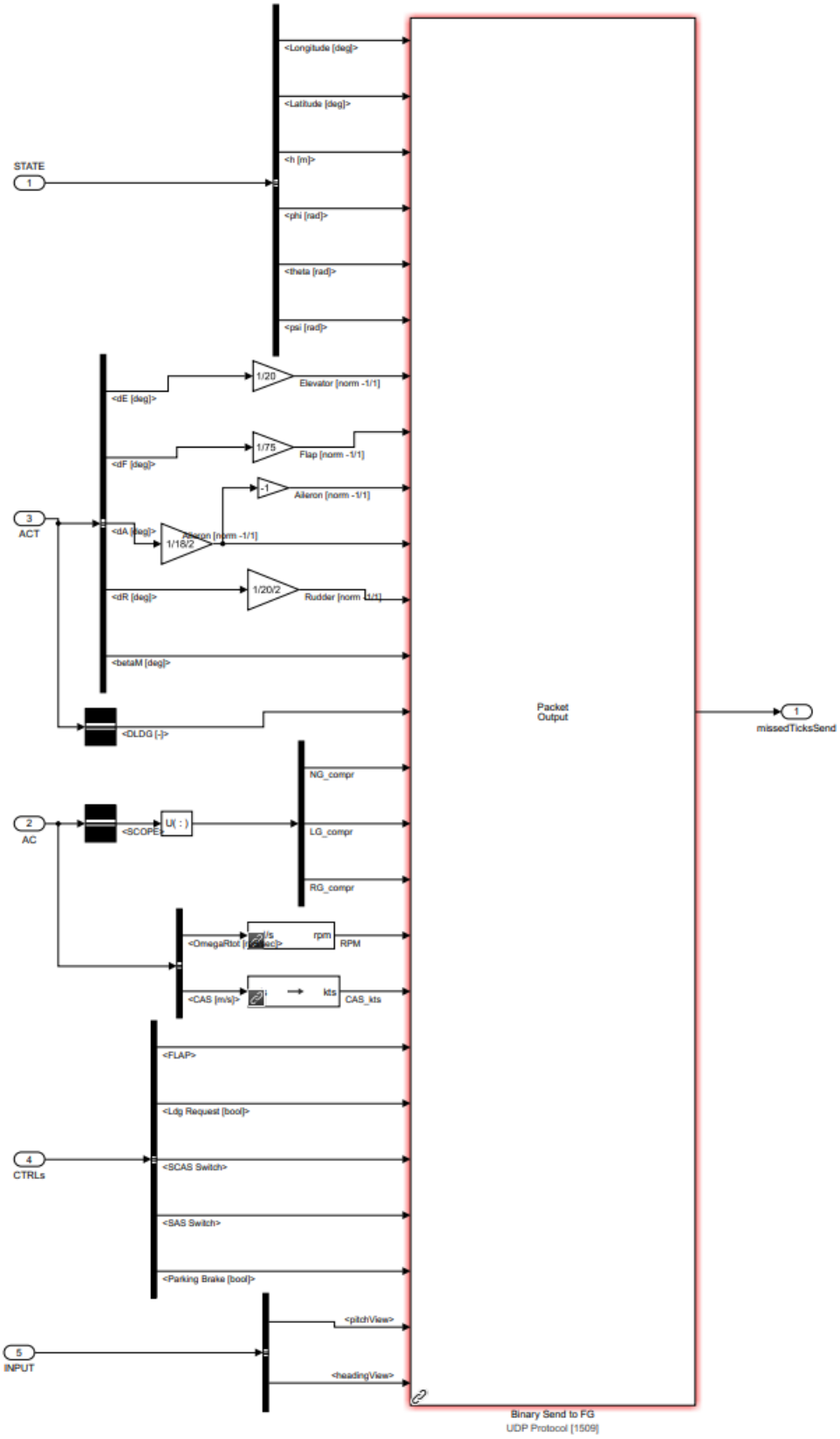

Figura 3.10: Sottoblocco UDP PROTOCOL OUT

RECEIVE

Il secondo componente è RECEIVE, che si occupa di ricevere i dati provenienti da FlightGear: di fatto, l'unico che viene trasferito è l'elevazione del terreno sottostante, che serve al modello Simulink per calcolare la posizione dell'XV-15 nell'ambiente tridimensionale, in termini di latitudine, longitudine e quota.

3.2 FlightGear

Figura 3.11: Schermata iniziale di FlightGear

FlightGear è un software open-source che permette di riprodurre un ambiente virtuale tridimensionale in modo da visualizzare graficamente la simulazione di volo del proprio velivolo. Non richiedendo computer di fascia elevata (il computer su cui gira il simulatore è un laptop Dell Precision 7550) ed essendo molto personalizzabile (è possibile creare i propri modelli di velivolo e inserirli in FlightGear) è stato scelto per la parte grafica. La versione di FlightGear utilizzata è la 2020.3.13.
Il ruolo che ha FlightGear nell'architettura generale del simulatore è quello di "motore grafico": i calcoli vengono effettuati lato Simulink, mentre FlightGear si occupa di caricare i modelli 3D all'interno del proprio ambiente e di farli rispondere ai comandi del pilota (per la disposizione dei comandi si faccia riferimento alla Figura 3.12), oltre a riprodurre elementi sonori come il rumore dei motori o della retrazione del carrello.
Un'altra caratteristica interessante di FlightGear è la possibilità di riprodurre lo scenario reale dove si sta effettuando la simulazione tramite un add-on chiamato TerraSync: questo necessita di una connessione a internet per scaricare automaticamente la zona geografica dove è situata la simulazione in base alla posizione geografica del velivolo. Se ad esempio si fa decollare il proprio velivolo dall'aeroporto di Torino Caselle, FlightGear scaricherà automaticamente i dati fotogrammetrici e morfologici della zona circostante (in particolare l'elevazione del terreno) e continuerà ad aggiornarli man mano che ci si sposta durante il volo.

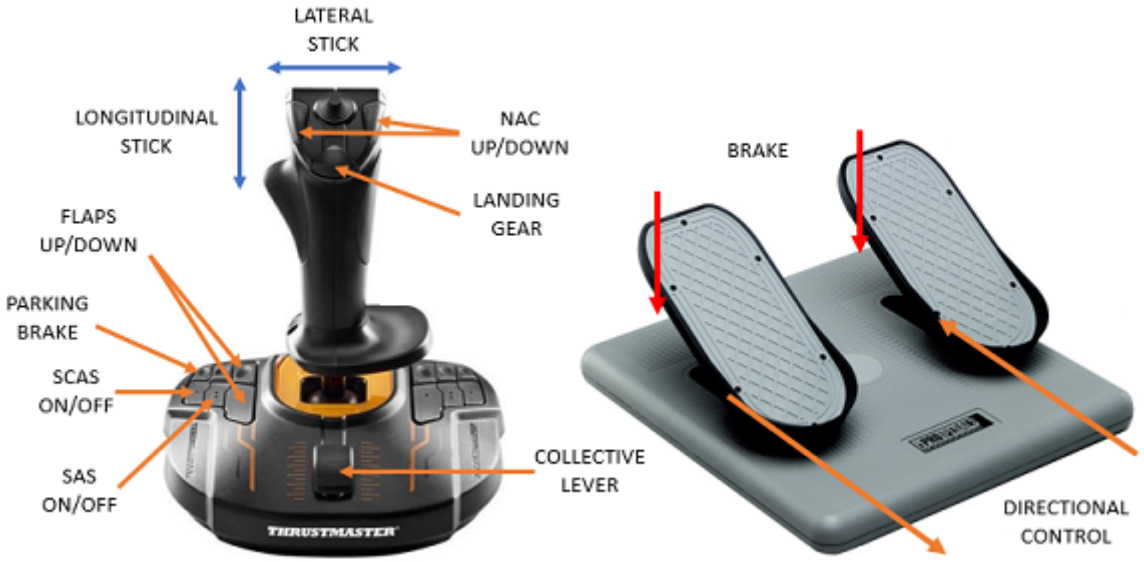

Figura 3.12: Periferiche utilizzate per la simulazione: a sinistra un joystick Thrustmaster T.16000M, a destra una pedaliera CH Products Pro Pedals

Oltre a TerraSync è presente anche un altro add-on chiamato TerraMaster che funziona in maniera simile ma non richiede una connessione continua a internet: l'utente seleziona precedentemente la zona geografica che desidera e TerraMaster la scaricherà in anticipo e la salverà sul computer. Il vantaggio è che, una volta scaricato, lo scenario rimane a disposizione dell'utente anche quando è offline, tuttavia tende a essere piuttosto oneroso per quanto riguarda l'occupazione della memoria del computer: per questo motivo si preferisce far affidamento a TerraSync.

Per creare un modello 3D in FlightGear e farlo riconoscere e funzionare da esso sono necessari fondamentalmente tre tipologie di file:

- i file .ac, che è il formato principale in cui sono realizzati i modelli grafici 3D del velivolo e delle sue parti
- i file .xml, che controllano ciò che FlightGear deve caricare, le animazioni dei modelli e l'interfaccia con Simulink
- i file .nas, scritti nel linguaggio di programmazione Nasal, per la realizzazione degli script all'interno di FlightGear

Altri tipi utilizzati sono i file .wav che contengono gli effetti sonori e i file di immagini in varie estensioni (.jpg, .png e così via) se si vogliono realizzare ad esempio delle texture. In definitiva FlightGear è uno strumento decisamente potente e allo stesso tempo semplice per creare e gestire la parte grafica di un simulatore di volo: maggiori dettagli verranno forniti nel Capitolo 4, dove verrà mostrato come l'autore ha contribuito a questo simulatore, principalmente proprio per quanto riguarda la modellazione 3D dell'XV-15 e la sua integrazione in FlightGear.

Capitolo 4

Sviluppo e integrazione del modello grafico in FlightGear

Lo scopo di questo capitolo è mostrare il contributo dell'autore al progetto. Come già scritto, il lavoro si è basato principalmente sulla realizzazione del modello grafico dell'XV-15 e sulla sua implementazione in FlightGear, oltre ad alcune aggiunte e migliorie al modello Simulink. Una precisazione prima di iniziare: gli oggetti dei modelli 3D realizzati hanno spesso il nome in francese. Questo perchè per la scrittura dei file XML sono stati presi come base di partenza quelli già presenti del V-22 e poi modificati appositamente, tuttavia i nomi sono appunto in francese ed è stato preferito adeguarsi a questa nomenclatura piuttosto che analizzare ogni singolo file XML e cambiare i nomi degli oggetti, un processo che sarebbe stato eccessivamente laborioso e che non avrebbe portato nessun beneficio. Al contrario, rinominare alcuni oggetti avrebbe richiesto una revisione attenta del progetto portando a dei bug ed errori, con conseguenti perdite di tempo per una questione solo formale.

4.1 Sviluppo del modello 3D dell'XV-15

La prima parte del lavoro dell'autore è stata la realizzazione del modello grafico 3D dell'XV-15. Come già scritto, il simulatore usava inizialmente come placeholder il modello grafico del V-22 Osprey, mentre il modello matematico è sempre stato quello dell'XV-15.

Figura 4.1: Modello del V-22 in FlightGear

Non avendo trovato modelli grafici dell'XV-15 disponibili online, si è deciso di crearne uno partendo da zero: il software utilizzato è Blender (versione 2.8), uno strumento di modellazione 3D estremamente potente che permette di creare modelli con elevata resa grafica grazie ai suoi tool avanzati di shading, texturing e così via.

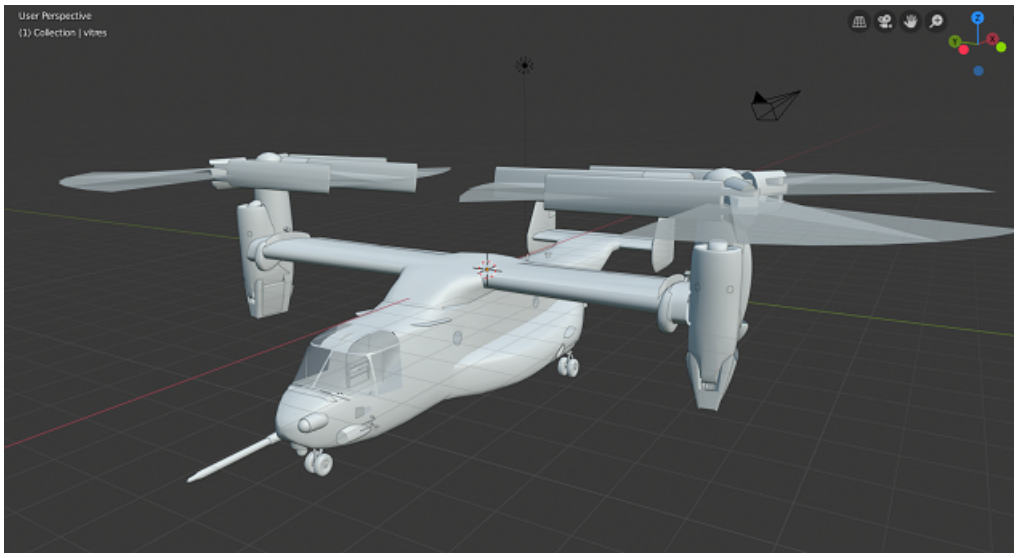

Figura 4.2: Modello del V-22 in Blender

Si precisa che questa sezione non si propone di essere una guida completa su come realizzare un modello 3D in Blender, ma si concentrerà maggiormente sulle fasi del lavoro e sugli strumenti utilizzati per ottenere determinati risultati. Un'altra premessa da fare è che FlightGear accetta file con estensione .ac per i modelli 3D, mentre Blender li salva in .blend: a tal proposito i modelli sono stati creati in formato .blend per comodità e, una volta completati, sono stati convertiti in .ac grazie a un tool apposito.
Per creare un modello su Blender si apre un nuovo progetto e si comincia a lavorare a partire da una forma semplice, in questo caso un cubo (Figura 4.3).

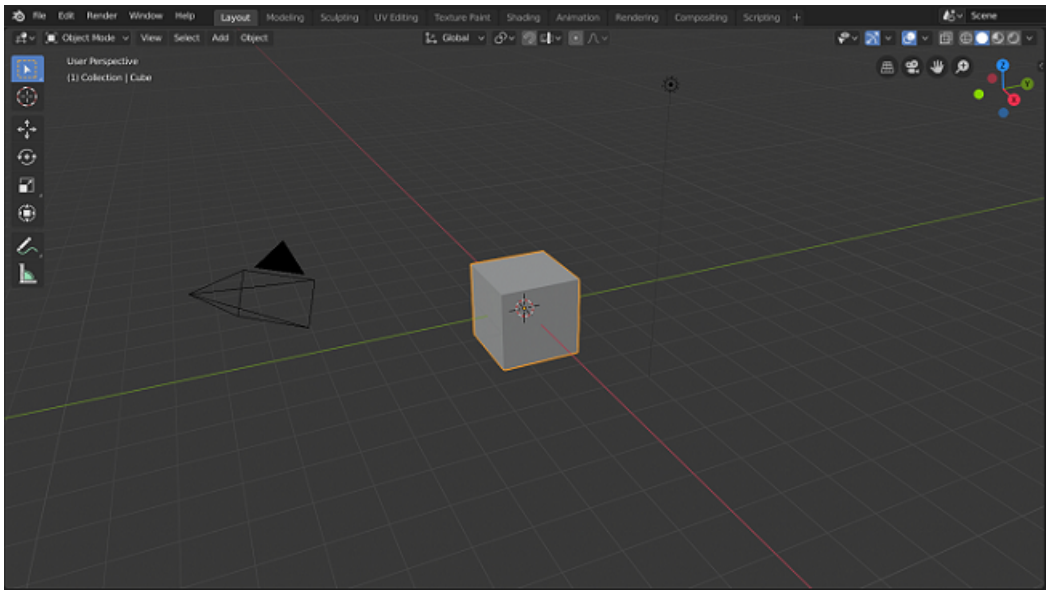

Figura 4.3: Nuovo progetto su Blender

Blender ha due modalità principali di interazione con gli oggetti, chiamate Object Mode ed Edit Mode: la prima è utile per organizzare le posizioni dei vari oggetti, la seconda permette di modificare la forma del proprio oggetto per ottenere ciò che si desidera. Gli strumenti base a disposizione in Edit Mode sono:

- traslazione, per spostare rigidamente un oggetto nello spazio tridimensionale
- rotazione, per ruotare rigidamente un oggetto attorno ad un certo asse
- ridimensionamento, per cambiare la dimensione di un oggetto mantenendone le proporzioni
- loop cut, che permette di dividere un oggetto in più sezioni in modo da lavorare su ciascuna in maniera indipendente dalle altre; si creano in sostanza dei nuovi vertici (ovvero punti) sull'oggetto che possono essere poi manipolati attraverso diverse funzioni
- estrusione, dove si selezionano i vertici che compongono la faccia di un poligono e la si allunga o accorcia lungo la normale a quella superficie
- bevel, che permette di smussare gli spigoli di un oggetto in modo da ottenere una forma più liscia
- shear, per deformare un oggetto lungo una particolare direzione

Utilizzando questi strumenti è possibile ottenere la forma desiderata. Una tecnica molto utile per ottenere un modello con le giuste proporzioni è quella di caricare nell'ambiente di Blender un'immagine di riferimento, in questo caso un trittico di viste dell'XV-15, e "ricamare" su di essa la forma desiderata; mettendo la vista corrispondente per ogni piano dello spazio 3D (quindi tre immagini orientate in modo coerente di modo che in ognuno dei tre piani ortogonali ci sia la relativa vista) si può ottenere, usando gli strumenti sopracitati, un modello che sia molto simile a quello desiderato e soprattutto con le proporzioni giuste. Se le proporzioni sono state fatte correttamente, basta semplicemente usare lo strumento di misura di Blender su una dimensione base, come ad esempio la lunghezza del velivolo, confrontarla con la dimensione reale e scalare l'intero modello per farlo combaciare con essa. Il risultato è mostrato in figura 4.4.

Figura 4.4: Modello dell'XV-15 in Blender

Quando si lavora su un modello è buona norma realizzarlo come "somma" di vari oggetti creati singolarmente: in pratica si crea un oggetto (ad esempio il motore, la fusoliera, il timone di coda e così via), gli si dà la forma desiderata e lo si sposta nella posizione di competenza. In questo modo non solo è più semplice lavorare sul modello, ma diventa anche un requisito se successivamente si vogliono realizzare delle animazioni come il movimento delle superfici mobili. L'elenco degli oggetti che compongono il modello dell'XV-15 è riportato in Figura 4.5.

Figura 4.5: Oggetti che compongono il modello dell'XV-15

Blender mette a disposizione anche una serie di "modifier" che consentono di ottenere particolari effetti: un modifier utilizzato in quello modello è "mirror", che semplifica il lavoro per le parti simmetriche. Ad esempio, nell'XV-15 i motori sono uguali e posizionati in maniera simmetrica rispetto al piano longitudinale del velivolo: tenendo conto di questa proprietà, si può creare e modellare il motore di destra, spostarlo nella posizione desiderata e applicare il modifier "mirror" per creare direttamente il motore di sinistra senza dover partire da zero, con forma e posizione simmetrica a quello di destra. Per l'XV-15 questo si può applicare su molti oggetti (motori, rotori, flap, alettoni e timoni di coda), agevolando di molto il lavoro.

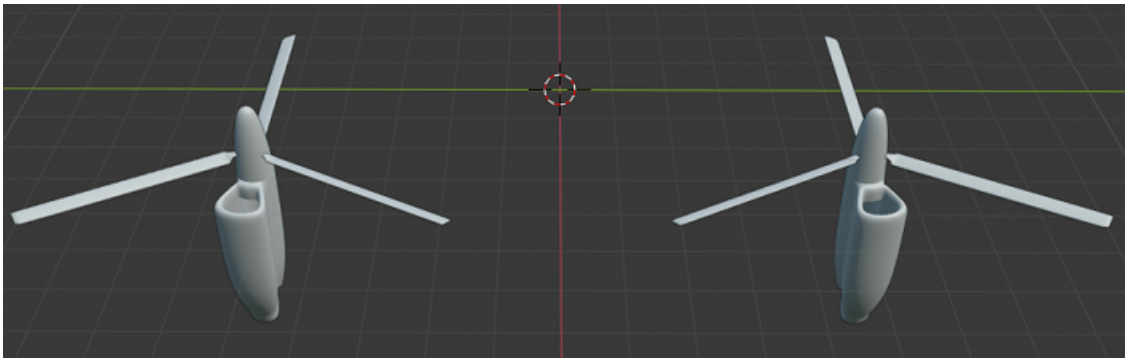

Figura 4.6: Motori e pale realizzati in maniera simmetrica con il modifier "mirror"

Un fattore a cui è necessario prestare attenzione è l'orientamento delle normali nel modello: se hanno il verso sbagliato, il modello potrebbe essere visualizzato in FlightGear in maniera errata. Per evitare problemi di questo tipo, è stato applicato il comando "Recalculate Normals", e in più a tutti gli oggetti è stato dato un certo spessore (seppur minimo) di modo che FlightGear sia in grado di renderizzarli correttamente.

4.1.1 Modello 3D dell'XV-15: materiali

Dopo aver creato la forma del modello dell'XV-15, si è proseguito con la definizione dei materiali di cui sono fatti i vari oggetti. La maggior parte dei componenti sono stati lasciati con il materiale di default (rinominato "Hull"), altri sono stati invece modificati, come ad esempio le ruote a cui è stato dato un colore nero (Figura 4.7) e le pale del rotore.

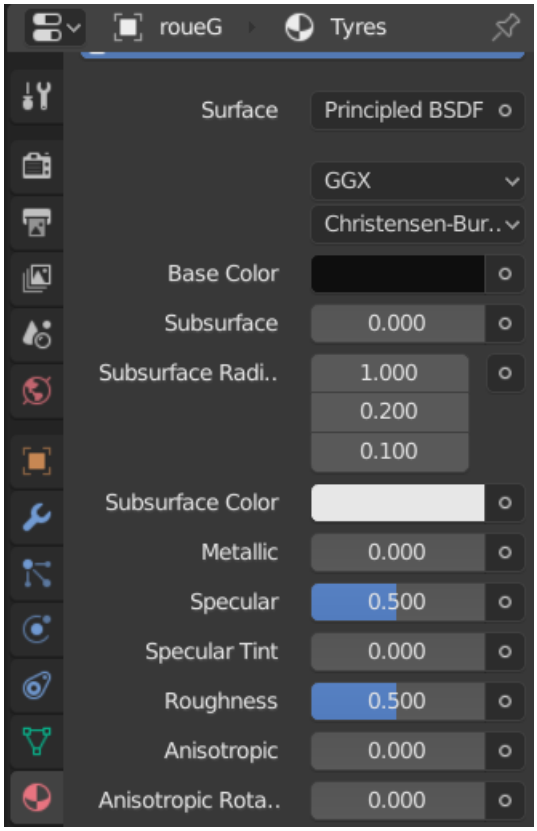

Figura 4.7: Alcune proprietà del materiale "Tyres" per le ruote

Particolare attenzione è stata rivolta al materiale di cui è fatto il vetro del cockpit (denominato "Glass"), in quanto è trasparente e non opaco come tutti gli altri oggetti: per controllare la trasparenza di un materiale si agisce sul parametro "Alpha", che varia da 0 (completamente trasparente) a 1 (completamente opaco). In questo caso, per il vetro è stato impostato un valore di Alpha pari a 0.5.

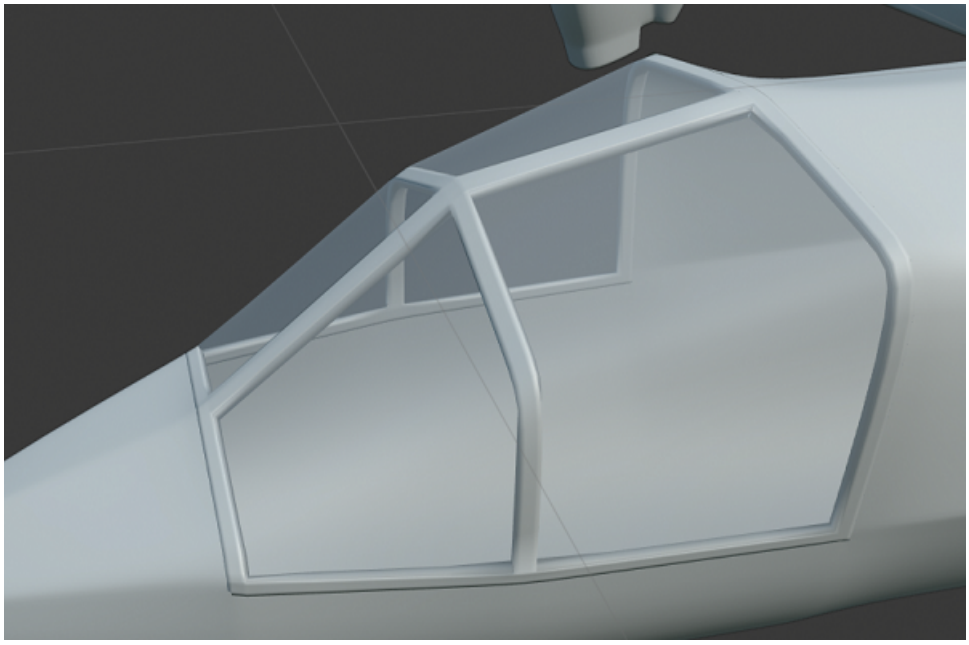

Figura 4.8: Dettaglio della trasparenza del vetro del cockpit

L'ultima considerazione da fare per quando riguarda i materiale è il tipo di shader utilizzato: Blender ne mette a disposizione numerosi tipi, tuttavia si è optato per "Principled BSDF", che è quello standard, poichè il tool sopracitato per convertire i file da .blend a .ac supporta solo quel particolare shader.

4.1.2 Modello 3D dell'XV-15: texture

Definiti forma e materiali, il modello grafico è completo e funzionale, pronto per essere convertito in formato .ac e esportato in FlightGear. Un'altra modifica, puramente estetica, che si può fare è aggiungere delle texture: per personalizzare questo modello è stato creato un oggetto chiamato "logo" e posizionato sul lato sinistro della fusoliera. Fatto ciò, si è creato e assegnato ad esso un materiale chiamato anch'esso "logo", e successivamente nel parametro "Base Color" non è stato scelto un colore bensì è stata selezionata l'opzione "Image Texture". A questo punto nel materiale è stata caricata l'immagine del logo del Politecnico di Torino come texture per l'oggetto. Il risultato è visibile in Figura 4.9.
Blender fornisce numerosi strumenti per realizzare texture: l'autore ha utilizzato questo metodo poichè è il più semplice e immediato, tuttavia è bene precisare che affinchè la texture venga visualizzata correttamente nell'ambiente di Blender e, soprattutto, nella simulazione in FlightGear è necessario che l'immagine utilizzata come texture si trovi sempre nella stessa cartella in cui si trova anche il file del modello 3D (sia per il formato .ac che per il .blend).

Figura 4.9: Dettaglio del logo sulla fusoliera

4.2 Esportazione del modello 3D in FlightGear

Quando il modello 3D è pronto in Blender, il passo successivo è l'esportazione in FlightGear. Come prima cosa è necessario convertire il file del modello da .blend a .ac: questo processo è stato fatto mediante un tool apposito chiamato Blender-AC3D, che consente di convertire file .blend in .ac e viceversa.
Fatto ciò, il file .ac dell'XV-15 è stato inserito nella cartella apposita del velivolo in FlightGear. Visto che, come già scritto in precedenza, per quanto riguarda i file XML si è partiti dalla base già presente per il V-22, la maggior parte dei file all'interno della cartella è chiamata come tale velivolo piuttosto che come l'XV-15. Non deve sorprendere ad esempio che il file .ac del modello 3D dell'XV-15 sia stato momentaneamente chiamato "v22.ac": correggere tutta la nomenclatura sarebbe stato una pura formalità che avrebbe creato errori e perdite di tempo evitabili.
A questo punto, è stata fatta una prima modifica al file XML "v22.xml" per consentire a FlightGear di riconoscere il modello 3D dell'XV-15: tale file non verrà riportato per intero all'interno di questa tesi in quanto occuperebbe troppo spazio, mentre verranno invece evidenziate le parti importanti per capirne il funzionamento. Il file è organizzato come segue:

```
<?xml version="1.0" encoding="UTF-8"?>
<PropertyList>

  <path>v22.ac</path>
    <animation>
    <object-name>fuselage</object-name>
    <object-name>logo</object-name>
    <object-name>profondeur</object-name>
    ...
    </animation>
</PropertyList>
```

La riga 1 è la cosiddetta "declaration", con cui devono iniziare i file XML. In seguito sono presenti una serie di tag (ovvero le proprietà racchiuse fra "<" e ">"), che devono essere "aperte" e "chiuse" affinchè siano riconosciute: per aprire una tag si fa come nella riga 2 <PropertyList> (start tag) e per chiuderla si fa come nella riga 11 </PropertyList> (end tag). All'interno delle tag vengono definiti gli elementi, come alla riga 4 dove la tag è "path" e l'elemento è "v22.ac": in questo modo FlightGear interpreta che il file da andare a cercare per caricare il modello 3D è "v22.ac" (non è necessario specificare il percorso perchè si trova nella stessa cartella del file XML).
Dopo aver selezionato il file corretto, vanno specificati anche tutti gli oggetti che compongono il modello grafico (fusoliera, flap, ruote e così via, nel codice precedente ne sono riportati solo alcuni a titolo di esempio alle righe 6-8): a questo punto FlightGear ha tutto ciò che gli serve per caricare il modello nel suo ambiente.

Figura 4.10: Modello 3D dell'XV-15 in FlightGear

4.3 Realizzazione delle animazioni

Una volta che il modello è stato esportato in FlightGear, si può passare alle animazioni, che sono di diverso tipo: possono essere rotazioni sul proprio asse, come per le pale del rotore, rotazioni attorno ad un certo asse definito, come per le gondole motore, traslazioni e così via. Per le animazioni dell'XV-15, il file in cui sono definite è lo stesso documento "v22.xml" che renderizza il velivolo il FlightGear.
Prima di iniziare a descrivere come sono state fatte le animazioni, è bene precisare che questo tipo di lavoro mal si presta ad un resoconto come la presente tesi, poichè difficilmente si può cogliere la bontà dell'animazione tramite una serie di immagini: per questo motivo, e anche per non appesantire eccessivamente il documento, le immagini riportate saranno quelle strettamente essenziali per capire in cosa consiste l'animazione, ad esempio per i flap non si raffigureranno tutti i vari gradi di estensione ma solamente le posizioni "completamente retratti" e "completamente estratti", con la consapevolezza che durante la loro movimentazione essi assumono tutti i valori intermedi e possono anche essere estratti parzialmente.

Ogni animazione è stata trattata in modo differente dalle altre poichè non si può fare lo stesso per le superfici mobili e per il carrello: di seguito si riporta ciò che è stato fatto per le varie componenti. Per quanto riguarda le coordinate in FlightGear, l'asse x è longitudinale orientato dal muso verso la coda dell'aereo, l'asse y va da sinistra verso destra e l'asse z dal basso verso l'alto.

4.3.1 Superfici mobili

Le superfici mobili su cui si è lavorato sono l'equilibratore, i timoni di coda, gli alettoni e i flap: questi componenti sono tutti caratterizzati da una movimentazione consistente in una rotazione attorno al proprio asse di cerniera, quindi i codici per le loro animazioni sono simili.
Di fatto, la parte fondamentale per questo tipo di animazione è l'impostazione dell'asse di rotazione: il metodo usato è definire nel file l'orientamento dell'asse e le coordinate spaziali di un punto appartenente ad esso, in modo da collocare univocamente l'asse nello spazio tridimensionale.

Equilibratore

La parte di codice che realizza l'animazione dell'equilibratore è il seguente:

```
1   <animation>
2     <type>rotate</type>
3     <object-name>profondeur</object-name>
4     <property>surface-positions/elevator-pos-norm</property>
5     <factor>15</factor>
6     <center>
7       <x-m>  3.750 </x-m>
8       <y-m>  0.000 </y-m>
9       <z-m> -0.600 </z-m>
10    </center>
11    <axis>
12      <x> 0 </x>
13      <y> 1 </y>
14      <z> 0 </z>
15    </axis>
16  </animation>
```

Con il tag "type" alla riga 2 si specifica il tipo di animazione, in questo caso una rotazione. Con il tag "object-name" si specifica quale o quali oggetti sono interessati dall'animazione: volendo è anche possibile creare un gruppo di oggetti per evitare di doverli sempre specificare singolarmente nelle parti di codice successive:

```
1   <animation>
2     <name>Group</name>
3         <object-name>Comp1</object-name>
4         <object-name>Comp2</object-name>
5         <object-name>Comp3</object-name>
6   </animation>
```

Così facendo si crea un macro-oggetto chiamato "Group" costituito da "Comp1", "Comp2" e "Comp3", e per selezionare tutti i componenti nelle parte successive del codice basta chiamare il macro-oggetto nella tag "object-name".
Tornando al codice dell'animazione, la tag "property" specifica la variabile, ricevuta dal modello matematico su Simulink, che FlightGear controlla per eseguire l'animazione. In questo caso Simulink invia il valore della posizione dell'equilibratore, FlightGear lo salva come "elevator-pos-norm" e lo controlla per effettuare l'animazione.
La tag "factor" alla riga 5 specifica il valore (in gradi) dell'escursione della superficie mobile. Le righe 6-10 definiscono le coordinate (in metri) del punto appartenente all'asse nel sistema di riferimento del velivolo. Le righe 11-15 servono a definire direzione e verso dell'asse di rotazione (direzione tramite il valore assoluto, verso tramite il segno): in questo caso la rotazione avviene attorno a un asse parallelo all'asse y del velivolo, infatti i valori di x e z sono 0 mentre y è 1.
L'animazione per l'equilibratore è ora completamente definita.

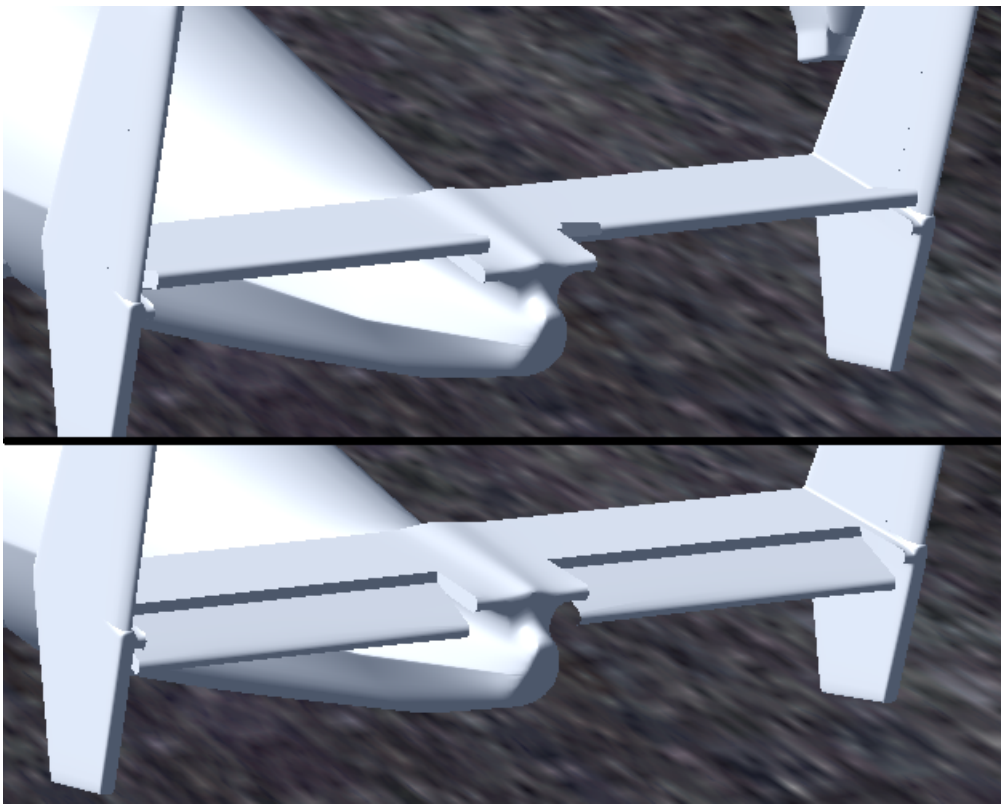

Figura 4.11: Animazione dell'equilibratore dell'XV-15

Timoni di coda

L'XV-15 ha due timoni di coda che si muovono in maniera accoppiata, ovvero entrambi ruotano a sinistra oppure entrambi ruotano a destra. Il codice per questa animazione è molto simile a quello per l'equilibratore: di seguito viene riportato il codice per il solo timone di destra (quello per il timone di sinistra è omologo con la coordinata y del centro invertita di segno):

```
1   <animation>
2     <type>rotate</type>
3     <object-name>directionD</object-name>
4     <property>surface-positions/rudder-pos-norm</property>
5     <factor>30</factor>
6     <center>
7       <x-m>  4.250 </x-m>
8       <y-m>  1.700 </y-m>
9       <z-m>  0.000 </z-m>
10    </center>
11    <axis>
12      <x> 0.30 </x>
13      <y> 0 </y>
14      <z> 1 </z>
15    </axis>
16  </animation>
```

I timoni di coda dell'XV-15 non ruotano attorno a un asse perfettamente verticale: per tale motivo alle righe 12-14 si può vedere come l'asse di rotazione abbia una componente in z ma anche una componente minore in x, in modo da realizzare una rotazione coerente con la geometria del velivolo.

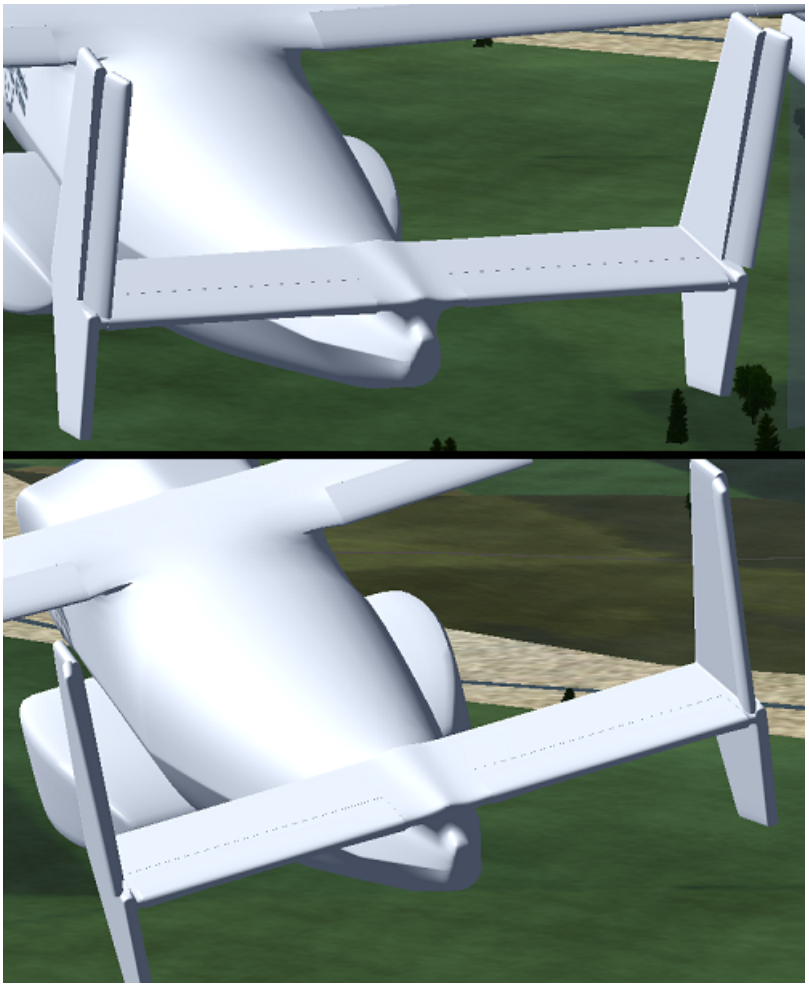

Figura 4.12: Animazione dei timoni dell'XV-15

Alettoni

Gli alettoni rispondono al comando in maniera opposta, ovvero se uno ruota verso il basso l'altro ruota verso l'alto. L'XV-15 inoltre ha un'ala a freccia leggermente negativa: per questo motivo gli assi di rotazione degli alettoni e, come si vedrà in seguito, dei flap avranno una componente in y e anche una in x. Viene riportato di seguito il codice per l'alettone di destra (quello di sinistra è identico tranne per il segno della coordinata y del centro e per il segno del valore di x dell'asse di rotazione):

```
1   <animation>
2     <type>rotate</type>
3     <object-name>aileronD</object-name>
4     <property>surface-positions/right-aileron-pos-norm</property>
5     <factor> 47.0 </factor>
6         <center>
7       <x-m> -2.850 </x-m>
8       <y-m>  2.700 </y-m>
9       <z-m> -0.900 </z-m>
10    </center>
11    <axis>
12      <x> -0.11 </x>
13      <y> 1 </y>
14      <z> 0 </z>
15    </axis>
16  </animation>
```

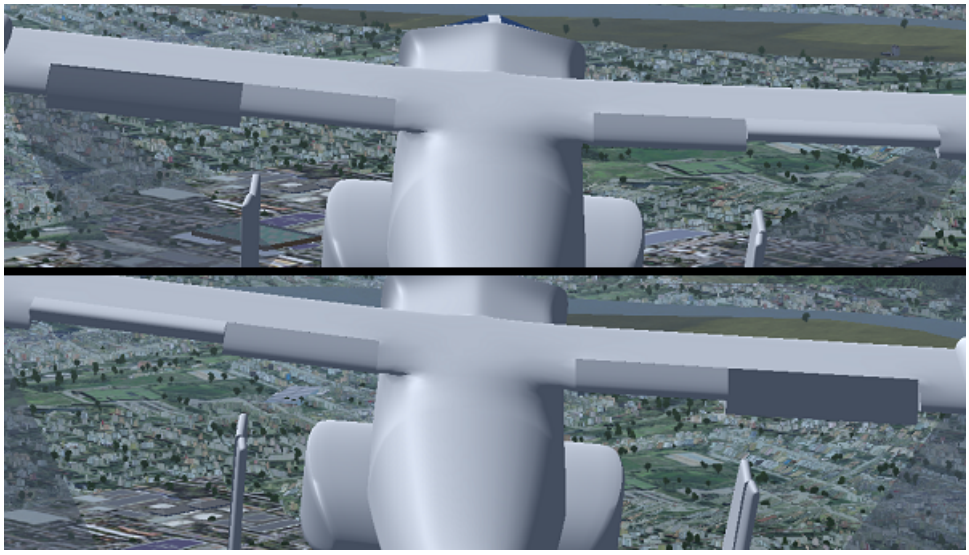

Figura 4.13: Animazione degli alettoni dell'XV-15

Flap

Il flap si estendono a delle inclinazioni prefissate in maniera simmetrica, con un'escursione totale di 75°. Come nei casi precedenti, viene riportato il codice del solo flap di destra, in quanto quello di sinistra è identico salvo per i segni della coordinata y del centro e x dell'asse di rotazione:

```
<animation>
  <type>rotate</type>
  <object-name>voletD</object-name>
  <property>surface-positions/flap-pos-norm</property>
  <factor>75.0</factor>
  <center>
    <x-m> -2.750 </x-m>
    <y-m>  1.700 </y-m>
    <z-m> -0.950 </z-m>
  </center>
  <axis>
    <x> -0.11 </x>
    <y> 1 </y>
    <z> 0 </z>
  </axis>
</animation>
```

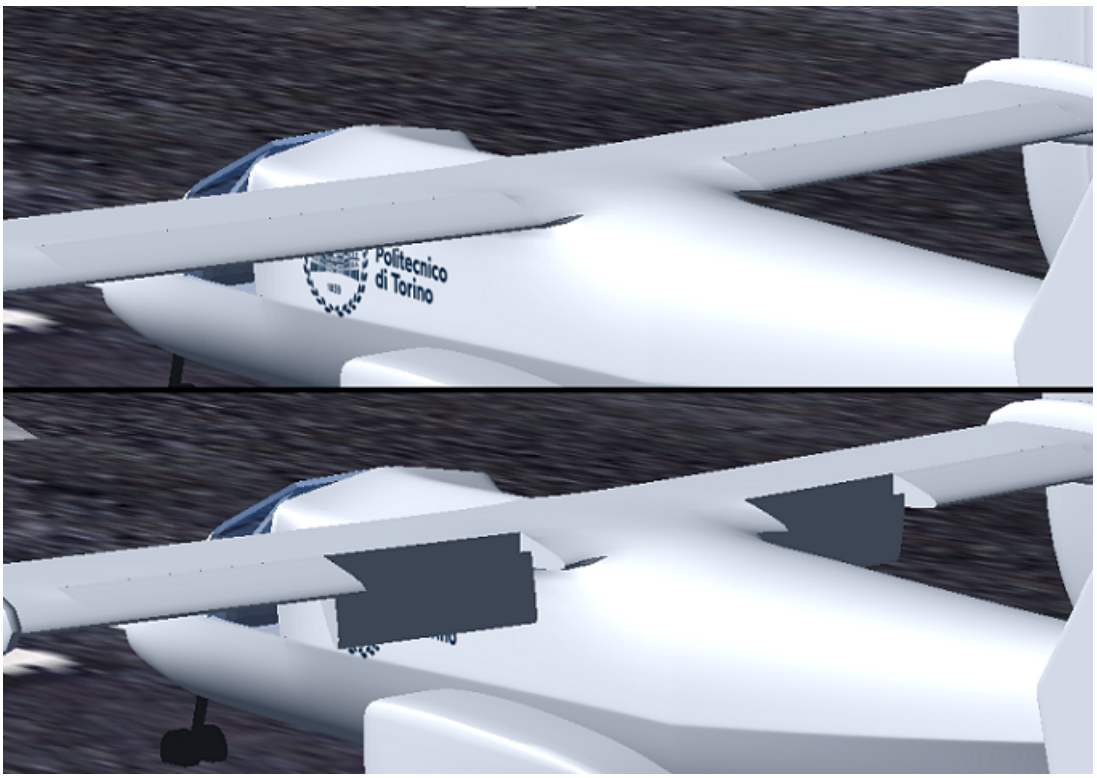

Figura 4.14: Animazione dei flap dell'XV-15, estensioni 0° e 75°

4.3.2 Carrello

Il carrello dell'XV-15 è composto dalla ruota anteriore e da quelle posteriori destra e sinistra. Ognuna di queste parti ha una sua animazione di rotazione, tutte molto simili tranne per le coordinate del centro e dell'asse di rotazione, inoltre quelle per le due ruote posteriori sono praticamente uguali tranne per i segni di alcuni valori.

Ruota anteriore

```
<animation>
  <type>rotate</type>
  <object-name>roueA</object-name>
  <property>gear/gear[0]/position-norm</property>
  <interpolation>
    <entry><ind> 0 </ind><dep> -97 </dep></entry>
    <entry><ind> 1 </ind><dep>   0 </dep></entry>
  </interpolation>
  <center>
    <x-m> -7.044 </x-m>
    <y-m>  0.000 </y-m>
    <z-m> -2.043 </z-m>
  </center>
  <axis>
    <x>  0 </x>
    <y>  1 </y>
    <z>  0 </z>
  </axis>
</animation>
```

Ruota posteriore destra

```
<animation>
  <type>rotate</type>
  <object-name>roueD</object-name>
  <property>gear/gear[0]/position-norm</property>
  <interpolation>
    <entry><ind> 0 </ind><dep> -52 </dep></entry>
    <entry><ind> 1 </ind><dep>  0</dep></entry>
  </interpolation>
  <center>
    <x-m> -3.400 </x-m>
    <y-m>  0.000 </y-m>
    <z-m> -1.250 </z-m>
  </center>
  <axis>
    <x> 0.2 </x>
    <y> 1 </y>
    <z> 0 </z>
```

```
18      </axis>
19    </animation>
```

Ruota posteriore sinistra

```
1     <animation>
2       <type>rotate</type>
3       <object-name>roueG</object-name>
4       <property>gear/gear[0]/position-norm</property>
5       <interpolation>
6       <entry><ind> 0 </ind><dep> -52 </dep></entry>
7       <entry><ind> 1 </ind><dep>   0 </dep></entry>
8       </interpolation>
9       <center>
10        <x-m> -3.400 </x-m>
11        <y-m>  0.000 </y-m>
12        <z-m> -1.250 </z-m>
13      </center>
14      <axis>
15        <x> -0.2 </x>
16        <y> 1 </y>
17        <z> 0 </z>
18      </axis>
19    </animation>
```

Figura 4.15: Animazione del carrello dell'XV-15

Alle righe 5-8 di tutti e tre i codici per le ruote compaiono delle nuove tag: queste servono a definire l'ampiezza del movimento che l'oggetto deve compiere durante l'animazione, e sono calibrate in modo che ogni ruota si retragga completamente all'interno della fusoliera.

4.3.3 Motori

Per quanto riguarda le animazioni dei motori, si è lavorato alla rotazione delle gondole motore e alla rotazione delle pale dei rotori attorno al mozzo. Un altro aspetto di cui si è tenuto conto è la transizione fra rotori fermi e rotori in movimento, che ha richiesto un tipo di animazione particolare.

Rotori

I rotori sono controllati da un'animazione di tipo spin, poichè essi ruotano attorno al mozzo, tuttavia se si applica questa animazione direttamente alle pale si ottiene un risultato pessimo. A tal proposito si è proceduto creando due oggetti: "blades" che rappresenta le pale "reali" e "propdisc" che rappresenta le pale in rotazione (Figura 4.16). Quest'ultimo è stato realizzato deformando la pala in modo da "allargarla" per dare l'effetto del rotore in movimento.

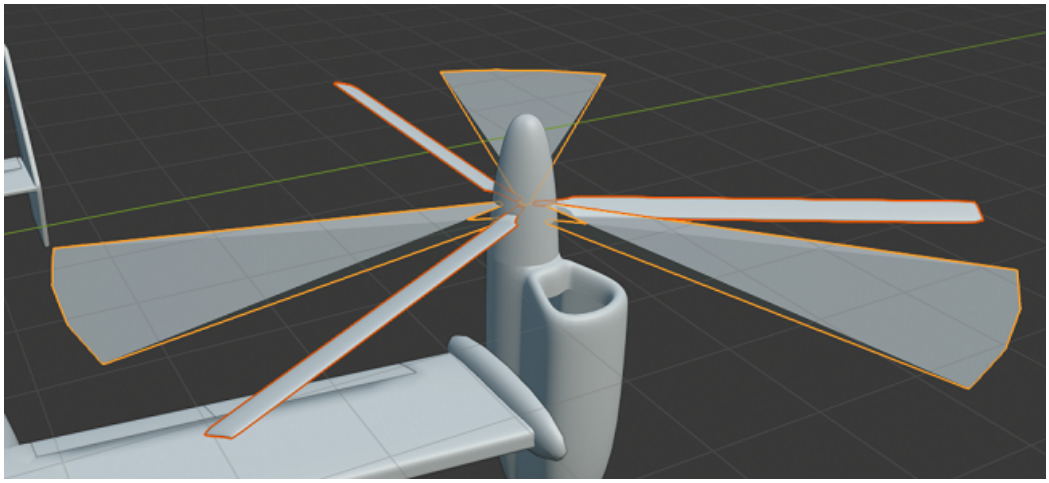

Figura 4.16: Dettaglio delle pale del motore sinistro in Blender

In FlightGear vengono visualizzate le pale "reali" (oggetto "blades") se i rotori ruotano lentamente e le pale in rotazione (oggetto "propdisc") se i rotori ruotano velocemente: quest'effetto si ottiene con un'animazione di tipo "select", di seguito riportata per il rotore destro.

```
1   <animation>
2     <type>select</type>
3     <object-name>bladesD</object-name>
4     <condition>
5                 <or>
6                 <less-than>
7                         <property>/engines/engine[0]/rpm</property>
8                         <value>150</value>
9                 </less-than>
```

```
10                <equals>
11                        <property>rotors/main/bladesvisible</property>
12                        <value>1</value>
13                </equals>
14                </or>
15          </condition>
16    </animation>
17
18    <animation>
19      <type>select</type>
20      <object-name>propdiscD</object-name>
21      <condition>
22        <greater-than>
23          <property>/engines/engine[0]/rpm</property>
24          <value>150</value>
25        </greater-than>
26      </condition>
27    </animation>
```

In questa parte di codice si vede il nuovo tipo di animazione "select" (righe 2 e 19) e le tag che servono a definire le condizioni affinchè l'animazione select funzioni. La proprietà che viene controllata sono i giri al minuto del motore (righe 7 e 23): se questo valore è minore di 150 rpm FlightGear carica l'oggetto "bladesD", mentre quando tale valore viene superato entra in gioco l'animazione select, che non fa visualizzare più "bladesD" e carica invece l'oggetto "propdiscD". Così facendo si ottiene un effetto visivo migliore per le pale in rotazione.
Per quanto riguarda l'animazione della rotazione attorno al mozzo, essa è ottenuta con un'animazione "spin", riportata di seguito per il motore destro.

```
1     <animation>
2       <type>spin</type>
3       <object-name>HeliceDroite</object-name>
4           <condition>
5         <greater-than>
6           <property>/engines/engine[0]/rpm</property>
7           <value>150</value>
8         </greater-than>
9       </condition>
10      <property>/engines/engine[0]/rpm</property>
11      <offset-deg>-90</offset-deg>
12      <factor> 1 </factor>
13      <center>
14        <x-m> -3.200 </x-m>
15        <y-m>  5.200 </y-m>
16        <z-m>  2.252 </z-m>
17      </center>
18      <axis>
19        <x> 0 </x>
20        <y> 0 </y>
```

```
21        <z> 1 </z>
22      </axis>
23    </animation>
```

Il nome dell'oggetto è "HeliceDroite" (riga 3), che è un macro-oggetto comprendente sia "bladesD" che "propdiscD" in modo che l'animazione di spin sia applicata indipendentemente da quale dei due oggetti è selezionato dall'animazione select. La variabile che definisce la velocità dello spin è il numero di giri dei motori (riga 10).

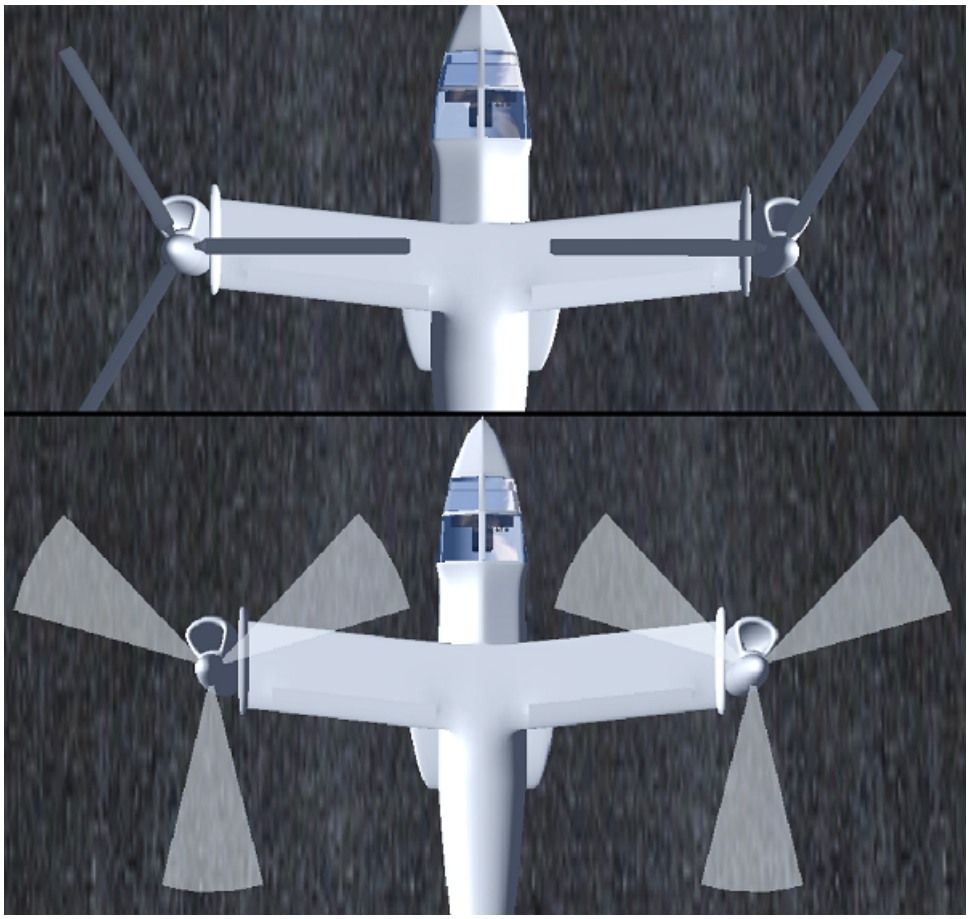

Figura 4.17: Pale ferme e in rotazione

Gondole motore

I motori ruotano attorno all'asse y e hanno tre posizioni prefissate: 90° (verticali), 60° e 0° (orizzontali). Per l'animazione, è stata realizzata una semplice rotazione.

```
1   <animation>
2     <type>rotate</type>
3     <object-name>LesMoteurs</object-name>
4     <property>surface-positions/tilt</property>
5     <factor>1</factor>
6     <center>
7       <x-m> -3.250 </x-m>
8       <y-m>  0.000 </y-m>
```

```
9         <z-m> -1.000 </z-m>
10      </center>
11      <axis>
12        <x>  0 </x>
13        <y> -1 </y>
14        <z>  0 </z>
15      </axis>
16    </animation>
```

Alla riga 3 "LesMoteurs" è un macro-oggetto che raggruppa le gondole motore e i rotori, perchè quando i motori ruotano allora anche i rotori devono ruotare solidalmente con essi.

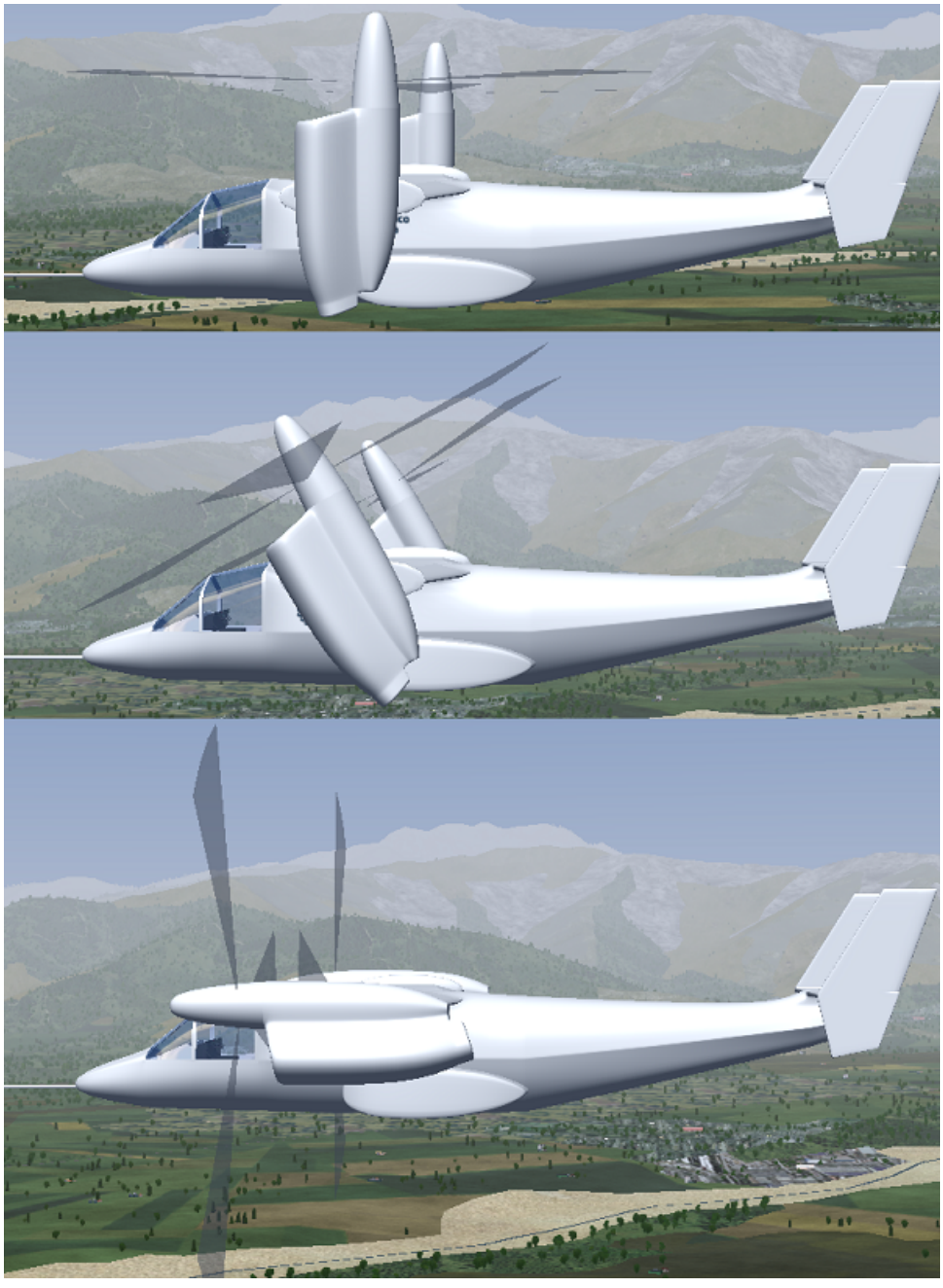

Figura 4.18: Rotazione dei motori

4.4 Modifiche al cockpit e alla strumentazione di bordo

Avendo modificato il modello 3D esterno, è stato necessario anche modificare il cockpit affinchè fosse adeguato all'XV-15: si precisa che non è stata realizzata una copia esatta del cockpit di tale velivolo, ma sono comunque presenti tutti gli strumenti e indicatori per pilotarlo in maniera soddisfacente.
In Figura 4.19 è riportato il cockpit nello stato iniziale del simulatore, con ancora il modello del V-22.

Figura 4.19: Cockpit del modello del V-22

4.4.1 Modello 3D del cockpit

La console e il pannello per la strumentazione sono stati presi dal modello del V-22 e ridimensionati per adattarsi all'XV-15.

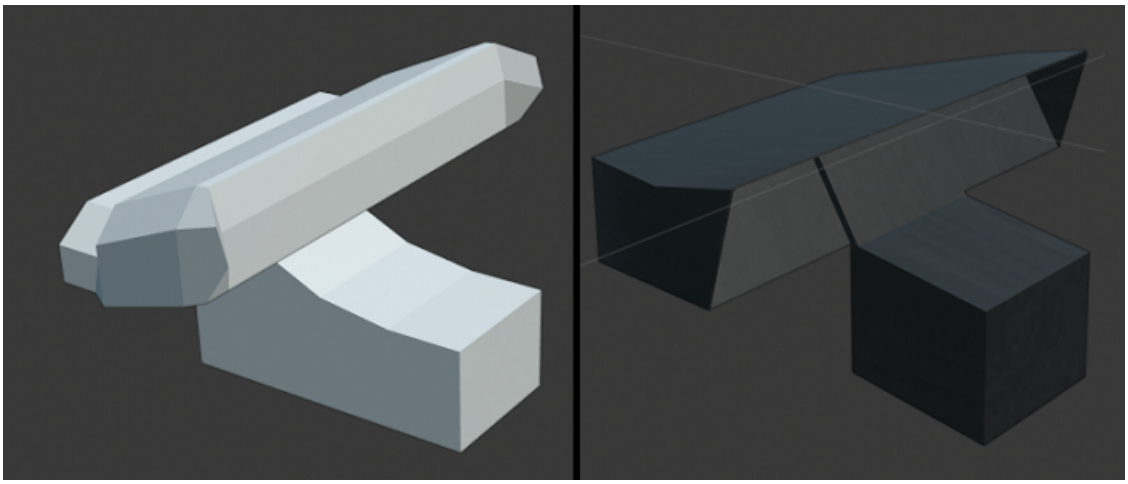

Figura 4.20: Console (sinistra) e pannello degli strumenti (destra) dell'XV-15 in Blender

Di seguito si riporta il codice per caricare la console e il pannello in FlightGear.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <PropertyList>
4
5    <path>interior.ac</path>
6
7    <animation>
8      <object-name>panel_center</object-name>
9      <object-name>panel_frame</object-name>
10    </animation>
11
12     <model>
13     <path>Aircraft/VMX22-Osprey/Models/Interior/Panel/panel.xml</path>
14     <offsets>
15       <x-m> -6.4 </x-m>
16       <y-m>  0.000 </y-m>
17       <z-m> -1.25 </z-m>
18       <roll-deg>    0.0 </roll-deg>
19       <pitch-deg>   -10.0 </pitch-deg>
20       <heading-deg> 0.0 </heading-deg>
21     </offsets>
22   </model>
23
24 </PropertyList>
```

Le righe 5-10 servono a caricare il modello della console, mentre alla riga 13 è specificato il percorso per il file XML relativo al pannello. Quest'ultimo file ha due scopi principali: caricare il modello 3D del pannello in FlightGear e caricare gli strumenti di bordo (ovviamente tutti funzionanti) e posizionarli correttamente sul pannello. Gli strumenti di bordo già presenti nel modello del V-22 sono stati riadattati al modello dell'XV-15, e in più sono stati realizzati dei modelli di indicatori luminosi dello stato di alcuni dei sistemi di bordo (SAS, SCAS e parking brake): maggiori dettagli verranno forniti nella Sezione 4.4.3.
La parte di codice per caricare il pannello è la seguente.

```
1  <path>panel.ac</path>
2
3    <animation>
4      <object-name>panel</object-name>
5    </animation>
```

Per caricare gli strumenti di bordo è necessario, per ognuno di essi, specificare il percorso dove si trova il relativo file XML (che ne controlla modello, animazioni e funzionamento) e definire la posizione dello strumento all'interno del cockpit. A titolo di esempio, visto che il procedimento è analogo per tutti gli strumenti, si riporta la parte di codice per l'altimetro.

```
1  <model>
2    <path>Aircraft/VMX22-Osprey/Models/Interior/Panel/Instruments/altime⌋
       ↪  ter/altimeter.xml</path>
3    <offsets>
4      <x-m> 0.26 </x-m>
5      <y-m> 0.49 </y-m>
6      <z-m> -0.425 </z-m>
7      <pitch-deg>-30.69</pitch-deg>
8    </offsets>
9  </model>
```

In questo caso alla riga 7 si vede la tag "pitch-deg" associata ad un valore di -30.69°, in modo da avere il modello dell'altimetro inclinato in avanti di tale valore. Nella Figura 4.21 è riportato il cockpit dell'XV-15 dopo che sono state effettuate le modifiche necessarie.

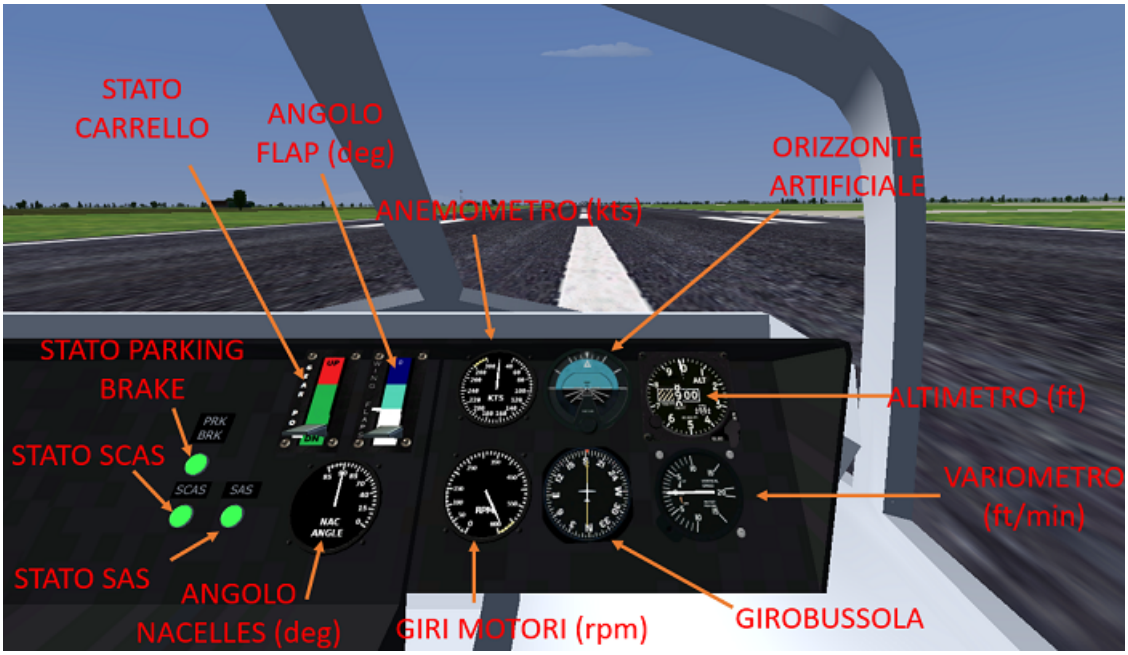

Figura 4.21: Cockpit dell'XV-15

4.4.2 Pulsante SAS ON/OFF

Inizialmente, il modello Simulink dell'XV-15 prevedeva un pulsante sul joystick che attivasse/disattivasse l'SCAS, in modo da poter effettuare dei test con tale sistema spento o acceso e valutarne l'efficacia sulla qualità del volo. Tuttavia, non sarebbe stato possibile effettuare gli stessi test sul SAS poichè non era presente un pulsante dedicato che svolgesse la stessa funzione del precedente: è stato dunque modificato il modello Simulink per aggiungerla.

Logica di controllo

Riprendendo l'architettura generale del modello (si veda Figura 3.1), si è partiti da PILOT INTERFACE, poi TMstick+PEDAL (Figura 3.3) e infine si è entrati nel blocco

"buttons to controls" dove sono definiti gli input digitali provenienti dal joystick. A questo punto si è inserito un nuovo componente "sasState" definito esattamente come gli altri e comprensivo di una funzione di trasferimento per simulare il ritardo del sistema nell'attuazione del comando.
Tale blocco si presenta come in Figura 4.22.

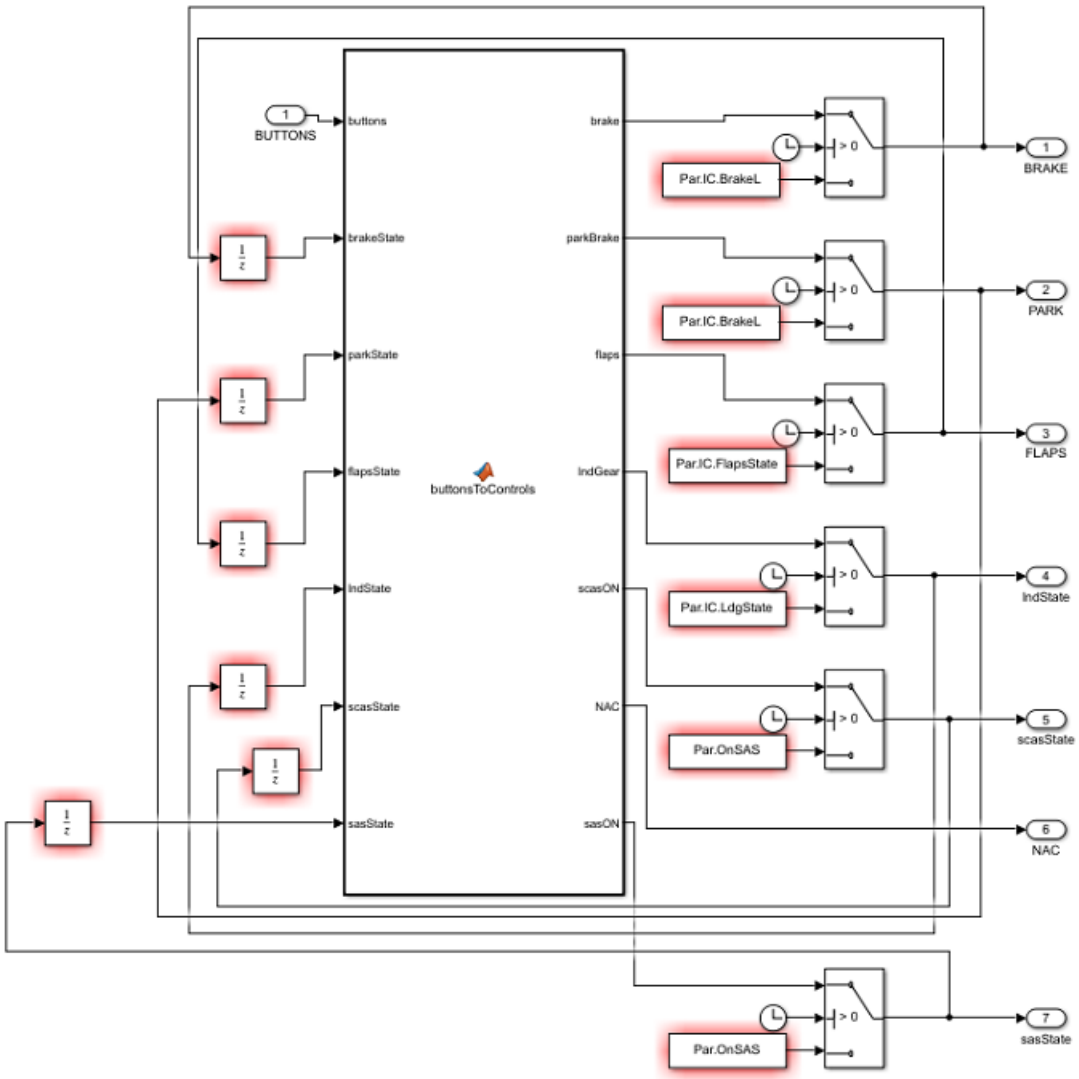

Figura 4.22: Blocco buttons to controls

Successivamente si è proceduto alla definizione, all'interno della funzione MATLAB "buttonsToControls" (di cui viene riportata solo la parte inerente ai bottoni di SAS e SCAS, ma in cui sono presente tutti gli input digitali del joystick come parking brake, flap e così via), della logica di controllo.

```
1 function [brake, parkBrake, flaps, ldgGear, scasON, NAC, sasON] =
  ↪ buttonsToControls(buttons, brakeState, parkState, flapsState,
  ↪ ldgState, scasState, sasState)
2
3
4 persistent toggleSCAS
5 if isempty(toggleSCAS)
```

```
6       toggleSCAS = 0;
7   end
8
9   persistent toggleSAS
10  if isempty(toggleSAS)
11      toggleSAS = 0;
12  end
13
14
15  scasON = scasState;
16  if (buttons(10) == 1 && toggleSCAS == 0)
17      if scasState == 1
18          scasON = 0;
19      elseif scasState == 0
20          scasON = 1;
21      end
22      toggleSCAS = 1;
23  end
24
25  sasON = sasState;
26  if (buttons(9) == 1 && toggleSAS == 0)
27      scasON = 0;
28      if sasState == 1
29          sasON = 0;
30      elseif sasState == 0
31          sasON = 1;
32      end
33      toggleSAS = 1;
34  end
35
36
37  if buttons(10) == 0
38     toggleSCAS = 0;
39  end
40
41  if buttons(9) == 0
42     toggleSAS = 0;
43  end
44
45
46  end
```

La funzione è una classica function MATLAB: gli input sono "buttons", "brakeState", "parkState" e gli altri nella parentesi tonda della riga 1, mentre gli output sono quelli compresi nella parentesi quadra. Alle righe 4-7 e 9-12 vengono definite delle variabili "toggleSCAS" e "toggleSAS" di tipo "persistent": queste servono successivamente nel codice per evitare che l'input del comando sia bloccato fino a quando il pulsante non viene rilasciato.
Alle righe 15-23 è presente la logica dei pulsanti per l'SCAS: viene definita una variabile

"scasON" e posta uguale a "scasState", la quale è un input della funzione che vale 0 se l'SCAS è disattivato oppure 1 se è attivato. Alla riga 16 si trova la prima condizione, ovvero il codice verifica se il bottone relativo all'SCAS (in questo caso il bottone numero 10 della periferica) è stato premuto: se così è, si passa alla seconda verifica, ovvero la condizione in cui si trova l'SCAS tramite la variabile "scasState". In pratica, visto che il pulsante è stato premuto, il sistema passa da acceso a spento o viceversa, quindi se scasState è pari a 1 allora scasON assume il valore 0 mentre se è pari a 0 scasON assume il valore 1. Alla riga 22 viene anche posto "toggleSCAS" pari a 1 di modo che il pulsante non possa venire premuto più volte all'interno della stessa iterazione (si guardi la condizione alla riga 16). Infine, alle righe 37-39, se il pulsante è stato rilasciato (quindi la condizione a riga 37 è soddisfatta) la variabile toggleSCAS viene posta pari a 0 di modo che sia possibile premere nuovamente il pulsante nell'iterazione successiva.
Lo stesso processo avviene per il SAS, con due differenze: innanzitutto il pulsante assegnato ad esso è il 9 e non il 10, e in seconda battuta alla riga 27 si può vedere come se il pulsante del SAS viene premuto allora l'SCAS viene automaticamente disattivato; questo perchè non ha senso che ci sia l'SCAS attivo e il SAS non attivo. In questo modo, premere il pulsante per attivare/disattivare l'SCAS non ha alcun effetto sul SAS, mentre premere il pulsante per attivare/disattivare il SAS disattiva automaticamente anche l'SCAS (indipendentemente dallo stato del SAS, alla pressione del pulsante la variabile scasON viene sempre posta a 0); allo stesso modo, premere il pulsante dell'SCAS non ha alcun effetto se il SAS è disattivato.
In sostanza, le uniche configurazioni possibili sono:

- SAS acceso e SCAS acceso
- SAS acceso e SCAS spento
- SAS spento e SCAS spento

In questo modo, oltre allo stato dell'SCAS che era già presente, si ottiene in output anche lo stato del SAS affinchè possa essere esportato dove necessario.

Modifiche in FCC

Il passo successivo è la modifica del sottoblocco FCC in MODEL TOT (si veda Figura 3.6). Nel blocco MIXER è stato aggiunto l'input "SwitchSAS", che porta l'informazione sullo stato del SAS. In Figura 4.23 è riportata la parte di questo blocco inerente al contributo del SAS: nella zona riquadrata in blu si ha in input l'effetto del SAS sui tre assi (RSAS, PSAS e YSAS). Dopodichè per ognuno è presente un blocco di controllo: se SwitchSAS vale 1 allora il SAS è attivo e viene fatto passare il suo contributo, viceversa se SwitchSAS vale 0 allora il SAS è spento e viene fatto passare un valore costante e pari a 0. Questo valore si somma poi ai comandi del pilota nella zona riquadrata in rosso. In questo modo si è realizzato un sistema che permette di attivare o disattivare il SAS, in aggiunta a quello per l'SCAS che, come già scritto precedentemente, era già presente nel modello.

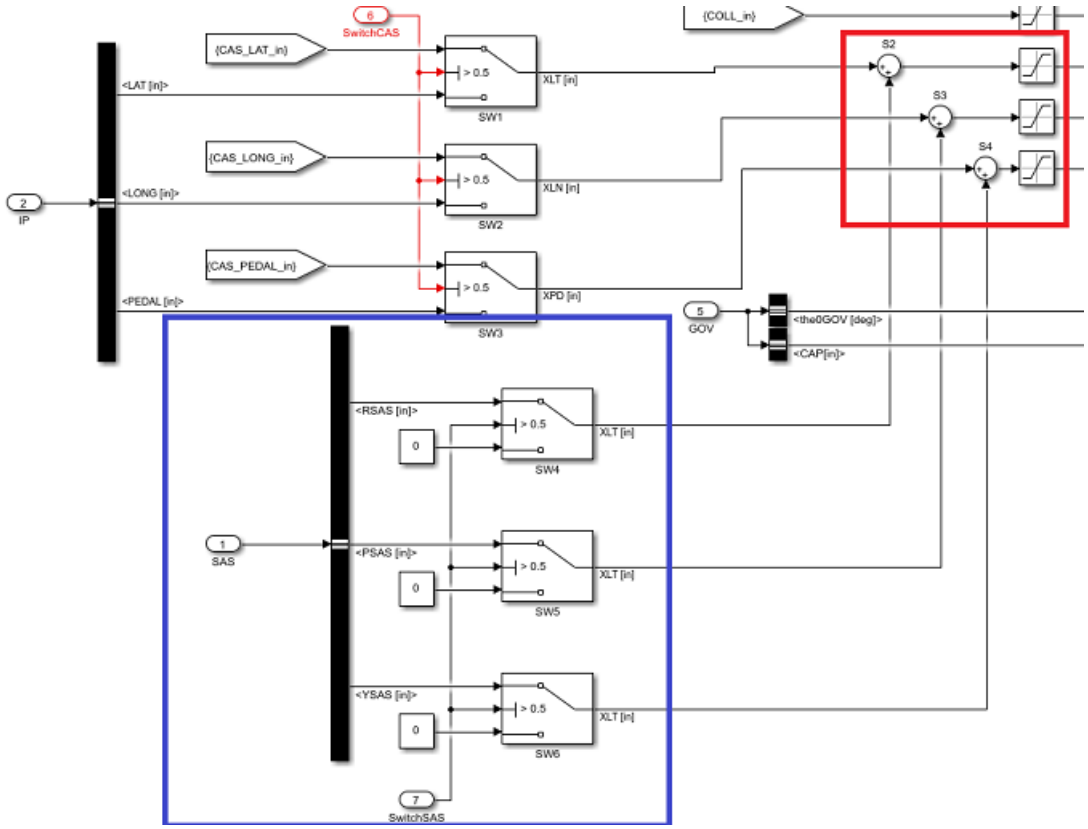

Figura 4.23: Contributo nel SAS nel blocco MIXER

Interfaccia con FlightGear

L'ultimo passaggio è l'esportazione della nuova variabile in FlightGear: per fare ciò si è lavorato sul sottoblocco VMX22 UDP Protocol Out (Figura 3.10) all'interno del blocco FLIGHTGEAR INTERFACE UDP, aggiungendo al Packet Output l'input "SAS Switch" proveniente da "CTRLs". Il Packet Output è stato ovviamente modificato adeguatamente aumentando di uno la dimensione del pacchetto in output e assegnando al nuovo output una ulteriore variabile di tipo "double", in accordo con le altre.
A questo punto si è passati al lato FlightGear, in particolare al file XML "SimulinkProtocolUDP", presente nella cartella "Protocol" del software, che interfaccia Simulink e FlightGear, permettendo al secondo di ricevere ed elaborare le variabili spedite dal primo: questo documento era già presente poichè al suo interno si trova tutto il necessario affinchè il modello grafico del velivolo si comporti come dettato dal modello matematico in termini di animazioni di superfici mobili, strumenti di bordo e così via. Di seguito si riporta una parte del file, in particolare quella inerente al controllo del SCAS, SAS e parking brake: questi sono i tre sistemi di cui è stato realizzato l'indicatore luminoso nel cockpit, come si vedrà nella sezione successiva.

```
1  <?xml version="1.0"?>
2  <PropertyList>
3  <generic>
4
5     <input>
6
7                  <binary_mode>true</binary_mode>
```

```
8
9               <!-- SCAS Request -->
10                  <chunk>
11                          <name>SCAS Request [norm 0/1]</name>
12                          <node>controls/scas</node>
13                          <type>double</type>
14                  </chunk>
15
16                     <!-- SAS Request -->
17                  <chunk>
18                          <name>SAS Request [norm 0/1]</name>
19                          <node>controls/sas</node>
20                          <type>double</type>
21                  </chunk>
22
23                  <!-- Parking Brake Request -->
24                  <chunk>
25                          <name>Parking Brake Request [norm 0/1]</name>
26                          <node>controls/prkbrk</node>
27                          <type>double</type>
28                  </chunk>
29
30    </input>
31    </generic>
32  </PropertyList>
```

Si prende come esempio il SAS, visto che gli altri funzionano in maniera analoga: le righe 17-21 sono incluse nella tag "chunk" che identifica una singola variabile in input. I vari chunk devono essere riportati nel codice nello stesso ordine con cui sono definiti in Figura 3.10 affinchè il tutto funzioni correttamente; alla riga 18 è definito il nome (in più è specificato che può assumere solo valori 0 e 1), alla riga 19 è definito il nodo (fondamentale per il funzionamento delle animazioni degli indicatori luminosi) e infine alla riga 20 è definito il tipo di variabile, che deve coincidere con quello indicato nel Packet Output di Simulink. In questo modo il file XML è stato configurato affinchè FlightGear possa ricevere correttamente ciò che Simulink gli invia.

4.4.3 Realizzazione degli indicatori luminosi nel cockpit

Un altro lavoro svolto dall'autore è stato l'aggiunta nel cockpit di indicatori luminosi per lo stato dei sistemi SAS, SCAS e parking brake. Inizialmente essi non erano presenti e l'unico feedback che il pilota aveva sul loro stato era la risposta più o meno gradevole del velivolo ai comandi, di certo non una soluzione valida: si è quindi deciso di inserire degli indicatori nel cockpit in modo che lo stato dei sistemi sia immediatamente visibile al pilota.

Creazione del modello 3D degli indicatori

Gli indicatori sono stati ideati come dei LED che sono illuminati di verde quando il relativo sistema è attivo e di rosso quando non è attivo, una soluzione semplice che

tuttavia migliora di molto la percezione che il pilota ha del proprio velivolo. Il primo passo è stato la creazione del modello del LED in Blender, concepito come due oggetti perfettamente sovrapposti ma di materiale diverso: uno ha il colore verde e l'altro ha il colore rosso. In questo modo, tramite un'animazione di tipo select che si vedrà in seguito, è possibile far comparire in FlightGear solo il LED verde o solo il LED rosso a seconda dello stato del relativo sistema, il tutto caricando un solo modello nel simulatore e snellendo dunque il processo.

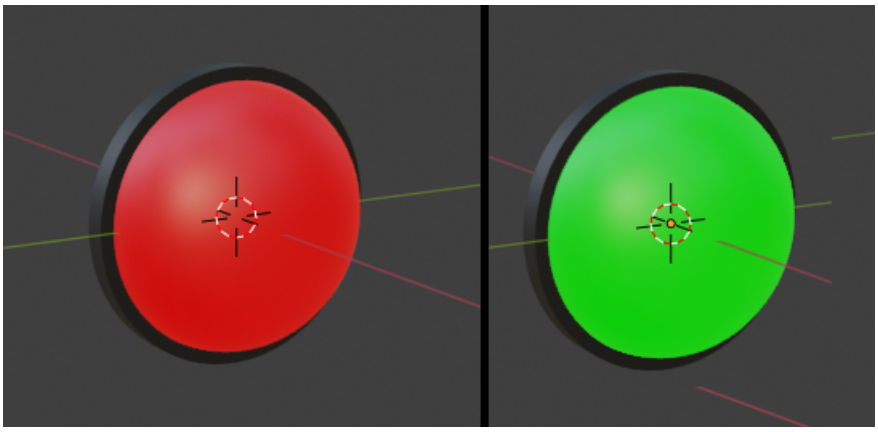

Figura 4.24: Modello del LED in Blender, a sinistra visualizzato spento e a destra acceso

Oltre ai LED sono stati realizzati anche dei semplici modelli di etichette per indicare a cosa si riferisce tale luce. Queste etichette sono dei rettangoli su cui è applicata una texture con la relativa scritta: a titolo di esempio in Figura 4.25 è riportato il modello di quella relativa al SAS.

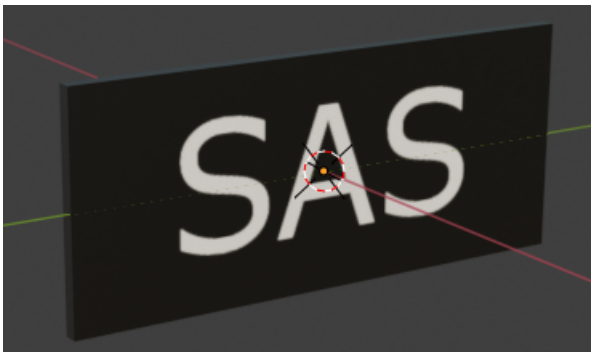

Figura 4.25: Etichetta del LED relativo al SAS

Integrazione e animazione in FlightGear

Una volta che i modelli sono stati creati in Blender, essi sono stati convertiti in formato .ac e inseriti nella cartella di FlightGear del velivolo, in particolare nella sottocartella "Instruments" dove ci sono tutti gli strumenti di volo: ogni oggetto del cockpit è inserito a sua volta nella propria sottocartella, in modo da rendere più agevoli le modifiche. Ogni strumento ha il relativo file XML che ne specifica le proprietà e le animazioni, e i LED non fanno eccezione.
Per integrare i LED e le relative etichette in FlightGear, il primo passo è stato inserire il relativo codice nel già citato file "panel.xml", dove si specifica la posizione dello strumento

nello spazio e il percorso del file XML legato a esso: come esempio è riportato il codice del documento "panel" per il LED del SAS e la relativa etichetta (si è proceduto in maniera analoga per SCAS e parking brake).

```
1  <model>
2      <path>Aircraft/VMX22-Osprey/Models/Interior/Panel/Instruments/sas/sa⌋
       ↪  s.xml</path>
3      <offsets>
4        <x-m> 0.21 </x-m>
5        <y-m> 0.05 </y-m>
6        <z-m> -0.55 </z-m>
7        <pitch-deg>-30.69</pitch-deg>
8      </offsets>
9    </model>
10
11   <model>
12     <path>Aircraft/VMX22-Osprey/Models/Interior/Panel/Instruments/sas/sa⌋
       ↪  s_label.xml</path>
13     <offsets>
14       <x-m> 0.21 </x-m>
15       <y-m> 0.059 </y-m>
16       <z-m> -0.525 </z-m>
17       <pitch-deg>-30.69</pitch-deg>
18     </offsets>
19   </model>
```

Dopo aver modificato l'XML generale che comprende tutti gli strumenti e avere specificato il percorso dove si trovano gli XML di LED e etichette (come fatto alle righe 2 e 12), questi XML vanno appunto creati e scritti affinchè gli oggetti vengano caricati da FlightGear con le relative animazioni. Per quanto riguarda le etichette l'XML è molto semplice, in quanto la loro posizione è già stata specificata in "panel.xml" e non hanno animazioni, di conseguenza il codice è il seguente.

```
1  <?xml version="1.0" ?>
2
3    <PropertyList>
4      <path>sas_label.ac</path>
5    </PropertyList>
```

Il precedente è per l'etichetta del SAS ma è analogo a quello delle altre. Di fatto, l'unica funzione di questo file è specificare alla riga 4 il percorso dove si trova il modello 3D da caricare (essendo nella stessa cartella dell'XML basta scrivere il nome del file .ac).
Per i LED (come al solito si riporta solo il codice fatto per il SAS) il procedimento è stato più complicato, in quanto si è anche dovuta inserire l'animazione che fa passare il colore da rosso a verde a seconda dello stato del sistema.

```
1  <?xml version="1.0" ?>
2
3    <PropertyList>
```

```
4
5       <path>sas.ac</path>
6
7           <animation>
8       <name>LEDRed</name>
9       <object-name>LEDR</object-name>
10      <object-name>baseR</object-name>
11      <object-name>SphereR</object-name>
12          </animation>
13
14          <animation>
15      <name>LEDGreen</name>
16      <object-name>LEDG</object-name>
17      <object-name>baseG</object-name>
18      <object-name>SphereG</object-name>
19          </animation>
20
21
22          <animation>
23      <type>select</type>
24      <object-name>LEDGreen</object-name>
25      <condition>
26        <greater-than>
27          <property>controls/sas</property>
28          <value>0.5</value>
29        </greater-than>
30      </condition>
31          </animation>
32
33          <animation>
34      <type>select</type>
35      <object-name>LEDRed</object-name>
36      <condition>
37        <lesser-than>
38          <property>controls/sas</property>
39          <value>0.5</value>
40        </lesser-than>
41      </condition>
42          </animation>
43
44  </PropertyList>
```

Alla riga 4 è specificato, come al solito, il modello 3D da caricare in FlightGear, in questo caso il LED per il SAS. Alle righe 4-9 e 11-16 sono stati creati due macro-oggetti, rispettivamente "LEDRed" e "LEDGreen", che sono i raggruppamenti degli oggetti che, nel modello 3D, compongono il LED rosso e il LED verde. In seguito è stata usata un'animazione di tipo select per passare "da un colore all'altro" a seconda dello stato del SAS: le righe 19-39 specificano che se la proprietà "controls/sas" (definita precedemente nel file di protocollo "SimulinkProtocolUDP.xml") è maggiore di 0.5, ovvero 1 poichè può

assumere solo quel valore oppure 0, FlightGear carica e mostra solamente il LED verde perchè il SAS è attivo, mentre se è minore di 0.5, ovvero 0, FlightGear carica e mostra solamente il LED rosso perchè il SAS è spento. In pratica, al posto di far cambiare effettivamente colore al modello del LED si carica un gruppo di oggetti oppure l'altro: il risultato è esattamente identico.
Lo stesso procedimento è stato fatto per i LED di SCAS e parking brake: le uniche differenze sono il modello da caricare (riga 3) e la proprietà da controllare per decidere quale colore scegliere (righe 24 e 35).
L'aggiunta di questi LED è una modifica apparentemente banale ma di grande importanza per il pilota poichè gli permette di capire con uno sguardo se un certo sistema è attivo o meno.

Figura 4.26: A sinistra: tutti i sistemi sono attivi. Al centro: tutti i sistemi non sono attivi. A destra: il SAS è attivo mentre SCAS e parking brake non sono attivi

4.5 Test di volo del modello dell'XV-15

In questa sezione si riporta un test di volo condotto nel simulatore per validare il funzionamento del modello 3D dell'XV-15, sia dal punto di vista della geometria degli oggetti che delle loro animazioni, in accoppiata col modello matematico. Per questo volo sono state definite le seguenti fasi:

- decollo verticale
- transizione da volo a punto fisso a volo traslato
- volo traslato
- virata a sinistra e riavvicinamento alla pista
- transizione da volo traslato a volo a punto fisso
- atterraggio

Si è iniziato con il velivolo fermo a terra, con motori accesi e pronto a partire.

Figura 4.27: XV-15 fermo a terra

Si è proceduto con l'aumento del passo collettivo delle pale per staccarsi da terra verticalmente.

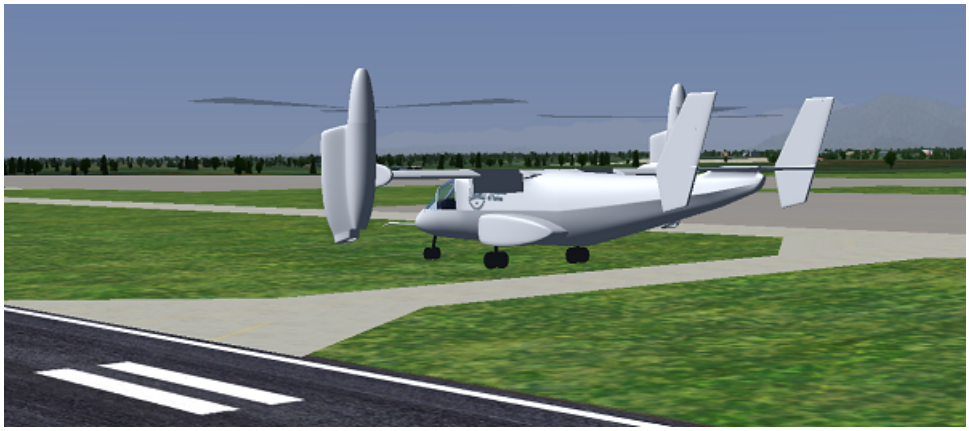

Figura 4.28: Decollo verticale

Ruotando le gondole motore e retraendo il carrello si inizia la transizione verso il volo traslato.

Figura 4.29: Fase di transizione

Nella Figura 4.30 è riportata la vista dal cockpit durante la transizione.

Figura 4.30: Fase di transizione vista dal cockpit

Completata la transizione si è passati al volo traslato.

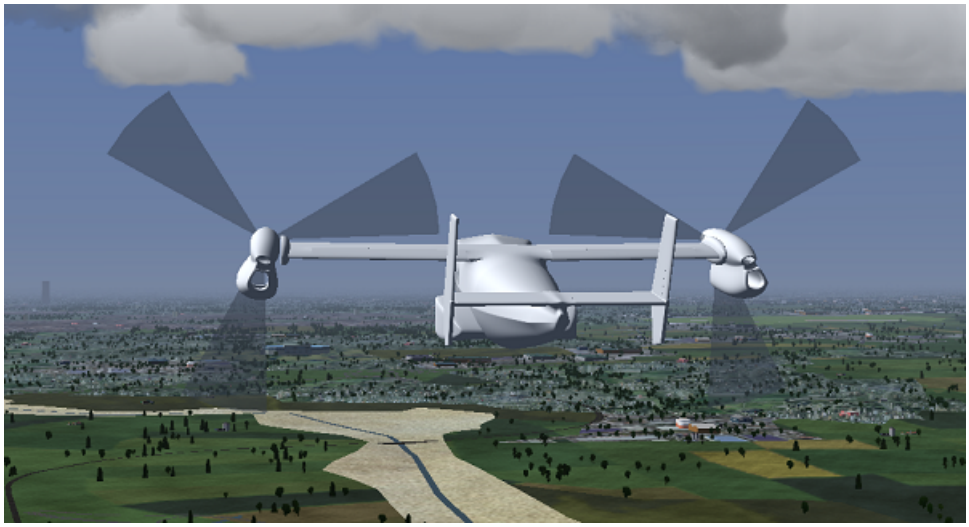

Figura 4.31: Volo traslato

Dopo aver livellato il velivolo, si è iniziata una virata verso sinistra con lo scopo di invertire la rotta, in preparazione all'avvicinamento alla pista e alla rotazione dei motori.

Figura 4.32: Virata verso sinistra

Nella parte in alto a sinistra della Figura 4.33 si può notare la pista di decollo vista dal cockpit dell'XV-15 durante la virata.

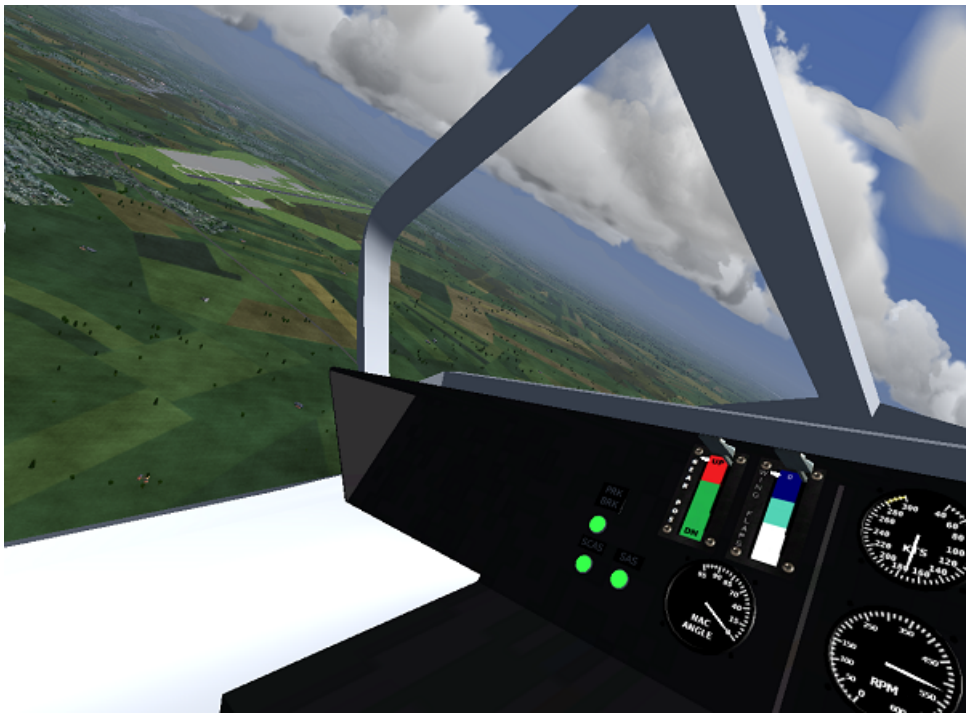

Figura 4.33: Vista dal cockpit durante la virata

Dopo la virata è stata mantenuta una direzione parallela alla pista ma in verso opposto a quello di decollo, per poi effettuare un'altra virata a sinistra, riallineare il velivolo alla pista e preparare il velivolo alla transizione per l'avvicinamento.

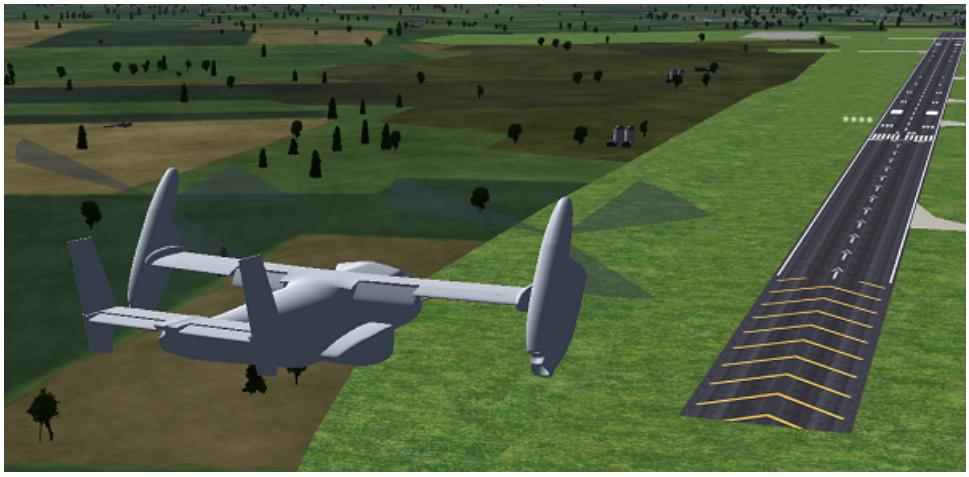

Figura 4.34: Transizione per l'atterraggio

In seguito i motori sono stati ruotati verticalmente e il carrello estratto per prepararsi al touchdown.

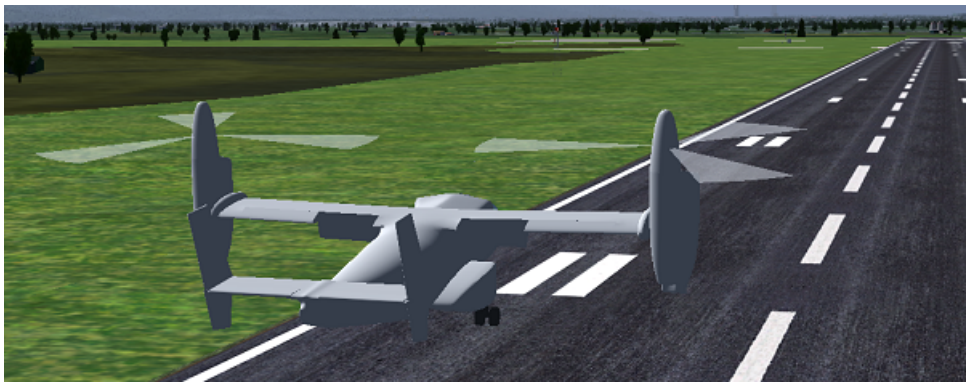

Figura 4.35: Avvicinamento al terreno

Il velivolo è ora atterrato sulla pista da cui è decollato e ha completato il volo di prova.

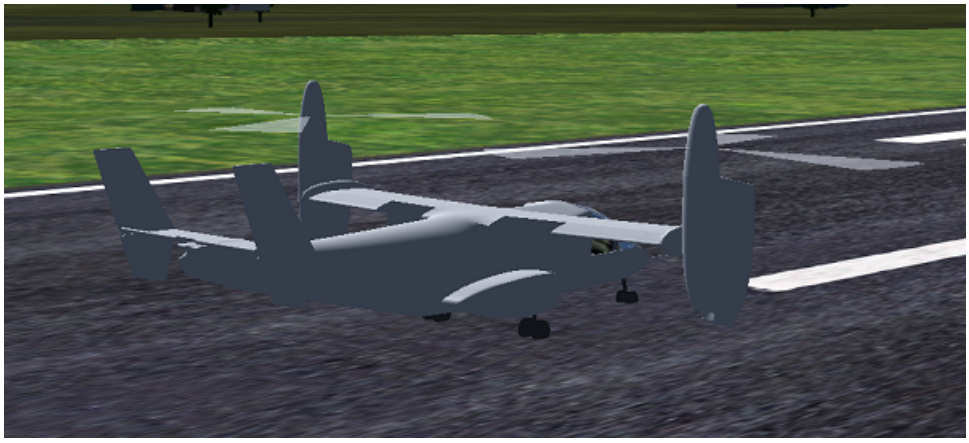

Figura 4.36: Atterraggio effettuato

Con questo volo di prova è stata confermata la validità del modello 3D e delle animazioni dell'XV-15. Il modello grafico ha eseguito in maniera precisa tutto quello che gli è stato richiesto e il risultato è stato soddisfacente. Nella Figura 4.37 è riportata la traiettoria che il velivolo ha seguito durante il volo di prova.

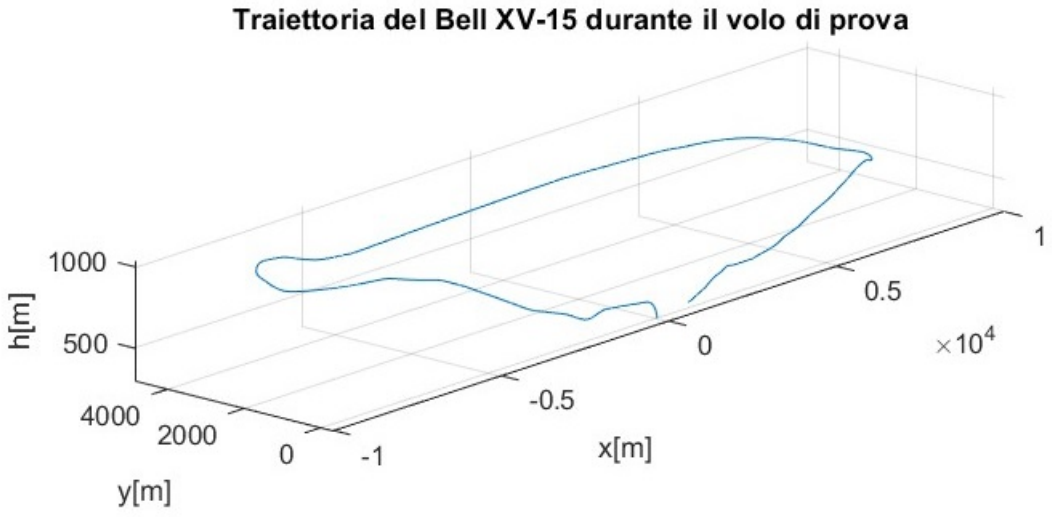

Figura 4.37: Traiettoria dell'XV-15 nel volo di prova

Capitolo 5

Conclusioni e sviluppi futuri

In questa tesi è stato sviluppato un modello grafico 3D per un simulatore di volo dell'XV-15, con scopi di didattica e di ricerca, ed è stato integrato in FlightGear. Tale simulatore è ora più fedele al velivolo reale, avendo sostituito il modello grafico precedente con uno che combacia con il modello matematico già presente.
Con il nuovo modello 3D dell'XV-15 è stato anche possibile realizzare le animazioni di superfici mobili, carrello e motori che riproducono accuratamente il loro comportamento reale affinchè la simulazione sia più immersiva. La nuova strumentazione nel cockpit permette al pilota di avere un feedback immediato sullo stato di sistemi fondamentali come SAS, SCAS e parking brake.
I test di volo, riportati nel capitolo precedente, confermano quanto affermato: il modello 3D dell'XV-15 viene caricato e visualizzato correttamente in FlightGear, con tutte le animazioni funzionanti. La user experience è notevolmente migliorata grazie al lavoro svolto.
Il simulatore non è però perfetto o privo di limiti e per il futuro sarebbe utile:

- migliorare il modello grafico in generale, soprattutto per l'interno del velivolo
- migliorare le texture sia esterne che interne
- inserire una periferica aggiuntiva che funga da leva di collettivo
- risolvere alcuni bug, in particolare quelli riguardanti TerraSync che causano la compenetrazione del modello 3D col terreno e il crash della simulazione
- aumentare il campo visivo (FOV) della simulazione, in quanto i riferimenti esterni (alberi, edifici e così via) sono molto usati dal pilota durante la fase di hover per controllare il velivolo: questo può essere fatto aumentando il numero di schermi su cui viene eseguita la simulazione oppure tramite l'implementazione e l'utilizzo di un visore per realtà virtuale (VR), che tuttavia richiede che l'intero simulatore venga trasferito su un altro sistema operativo (Linux), compito assolutamente non semplice

Bibliografia

[1] G. D. Padfield, *"Helicopter Flight Dynamics: including a Treatment of Tiltrotor Aircraft"*, John Wiley & Sons, 2018.

[2] M. D. Maisel, D. J. Giulianetti, D. C. Dugan, *"The History of the XV-15 Tilt Rotor Research Aircraft: from Concept to Flight"*, The NASA History Series, 2000

[3] S. Primatesta, F. Barra, P. Capone, G. Guglieri, *"Design and Integration of a Tilt-Rotor Flight Simulation Platform"*, VFS Annual Forum and Technology Display, 2022

[4] S. Primatesta, F. Barra, S. Godio, G. Guglieri, P. Capone, *"Implementation of a Comprehensive Real-Time Flight Simulator for XV-15 Tilt-Rotor Aircraft"*, Politecnico di Torino, 2022

[5] F. Veronese, *"Integration of a Tilt-Rotor Flight Simulation Platform"*, Politecnico di Torino, 2021

[6] A. Briamonte, *"Development of a Control Augmentation System for a Tilt-Rotor Aircraft"*, Politecnico di Torino, 2022

[7] G. B. Churchill, D. C. Dugan, *"Simulation of the XV-15 Tilt Rotor Research Aircraft"*, NASA, 1982

[8] J. S. G. McVicar, *"A Generic Tilt-Rotor Simulation Model with Parallel Implementation"*, University of Glasgow, 1993

[9] N. N. Batra, L. W. Dooley, T. A. Sheehan, K. W. Goldstein, *"Use of Simulation during Preliminary Design of the V-22 Osprey"*, Bell Helicopter Textron Inc. & Boeing Vertol Company, 1986

[10] H. Sheng, C. Zhang, Y. Xiang, *"Mathematical Modeling and Stability Analysis of Tiltrotor Aircraft"*, Nanjing University of Aeronautics and Astronautics, 2022

[11] W. D. Reddy, *"V-22 Simulator Evaluation for Shipboard Operations"*, Naval Air Warfare Center Aircraft Division, 1994

[12] Lufthansa Aviation Training, *Training Manual for Airbus A320-200 Full-Flight Simulator*

[13] Sviluppatori e contributori di FlightGear, https://www.flightgear.org/