Implementation of a Comprehensive Real-Time Flight Simulator for XV-15 Tilt-Rotor Aircraft

Original
Implementation of a Comprehensive Real-Time Flight Simulator for XV-15 Tilt-Rotor Aircraft / Primatesta, Stefano; Barra, Federico; Godio, Simone; Guglieri, Giorgio; Capone, Pierluigi. - ELETTRONICO. - (2023), pp. 1-14. (Intervento presentato al convegno AIAA SCITECH 2023 Forum tenutosi a National Harbor (USA) nel 23-27 gennaio) [10.2514/6.2023-0336].

Publisher:
AIAA

Published
DOI:10.2514/6.2023-0336

Implementation of a Comprehensive Real-Time Flight Simulator for XV-15 Tilt-Rotor Aircraft

Stefano Primatesta*, Federico Barra†, Simone Godio‡ and Giorgio Guglieri§
Politecnico di Torino, Torino, Italy

Pierluigi Capone¶
ZHAW Zurich University of Applied Sciences, Winterthur, Switzerland

This paper presents a tilt-rotor flight simulation platform implementing a real-time simulation of the Bell XV-15 aircraft for teaching and research purposes. The mathematical model of the tilt-rotor aircraft is implemented in MATLAB/Simulink© including simplified models of aircraft dynamics, actuators, sensors, and Flight Control Computer. The implemented tilt-rotor mathematical model is interfaced with flight control hardware, i.e. a flight stick and a rudder pedal, used by the pilot to set input commands. Instead, the graphics environment is provided by FlightGear, an open-source and cross-platform software widely used in research activities. Another contribution of the paper is the design and implementation of a Stability Control and Augmentation System to enhance aircraft stability and improve handling qualities. The developed simulator is tested with several simulations validating the developed mathematical model and the effectiveness of the Stability Control and Augmentation System. The result is a tilt-rotor flight simulation platform executable on a commercial laptop with real-time performance for research and teaching activities.

I. Nomenclature

EoM = Equations of Motion
FCC = Flight Control Computer
$GTRS$ = Generic Tilt Rotor Simulator
PFD = Primary Flight Display
PI = Proportional-Integral
SAS = Stability Augmentation System
$SCAS$ = Stability Control and Augmentation System
$SLDRT$ = Simulink© Desktop Real-Time
$TRRA$ = Tilt Rotor Research Aircraft

II. Introduction

Tilt-rotor aircraft are unique thanks to their hybrid configuration. A tilt-rotor is able to perform vertical take-off and landing in helicopter mode and, then, can rotate its nacelles converting to airplane mode with enhanced performance in terms of speed, endurance, and fuel consumption [1]. Thanks to their flexibility, tilt-rotor aircraft are used to perform several applications both in civil and military applications.

One of the most popular projects with tilt-rotor aircraft was developed by NASA and Bell Helicopter Textron in the 1970s with the Tilt Rotor Research Aircraft (TRRA) XV-15 [2]. The XV-15 is considered one of the most successful tilt-rotor experimental aircraft programs and his heritage was leveraged for the development of the Bell Boeing V-22 Osprey. Recently, tilt-rotor aircraft have gained further popularity for the development of a new generation of aircraft for

---

*Assistant Professor, Department of Mechanical and Aerospace Engineering, Corso Duca degli Abruzzi 24, Torino, Italy.
†Ph.D. Student, Department of Mechanical and Aerospace Engineering, Corso Duca degli Abruzzi 24, Torino, Italy.
‡Ph.D. Student, Department of Mechanical and Aerospace Engineering, Corso Duca degli Abruzzi 24, Torino, Italy.
§Full Professor, Department of Mechanical and Aerospace Engineering, Corso Duca degli Abruzzi 24, Torino, Italy.
¶Dozent, ZAV Centre for Aviation, Technikumstrasse 71, Winterthur, Switzerland.

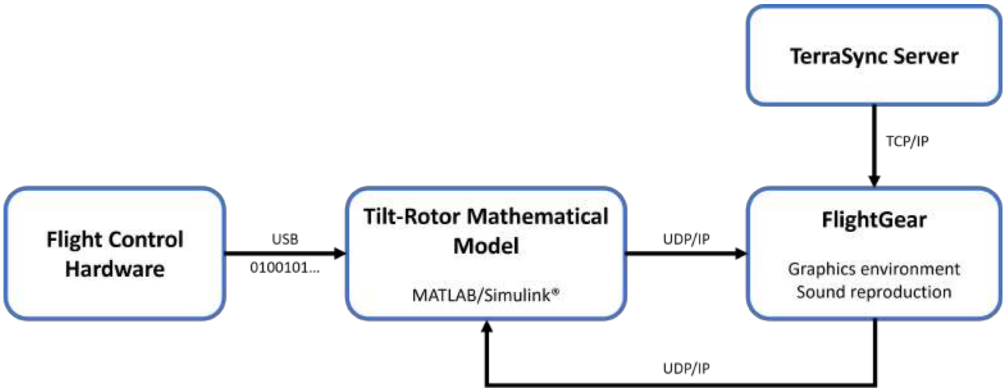


Fig. 1 The main architecture of the proposed tilt-rotor simulation platform

urban air mobility to provide several urban applications, such as air taxis and delivery to name a few [3]. Performing a vertical take-off and landing from vertiports in helicopter mode offers several advantages in urban environments. On the other hand, the conversion to airplane mode guarantees high performance during the cruise. For the same reasons, this unconventional configuration is also used for drones [4, 5].

However, the hybrid configuration of tilt-rotor aircraft is not without problems. The considerable wing download caused by the flow stream of the rotors in helicopter mode, the tilting mechanism which allows the conversion of the rotors during flight, the behavior of the wing in backward flight, and the coexistence of both conventional aircraft and rotorcraft flight controls are some of the main problems concerning tilt-rotor aircraft, compromising the in-flight stability and maneuverability.

In order to study such an innovative and unconventional platform, the development of a research and development simulator is required. In the literature, several tilt-rotor aircraft models have been proposed mainly related to the Bell Boeing V-22 Osprey and the Bell XV-15 [6–8]. A notable study has been proposed in [9], where the authors introduce the Generic Tilt Rotor Simulator (GTRS) widely used as a reference model at state of the art. Another important contribution has been presented in [10], where the GTRS model is enhanced by reporting the last issue on this model available in the public domain. Recently, several works based on the GTRS have been published for the development of new mathematical models [1, 11, 12].

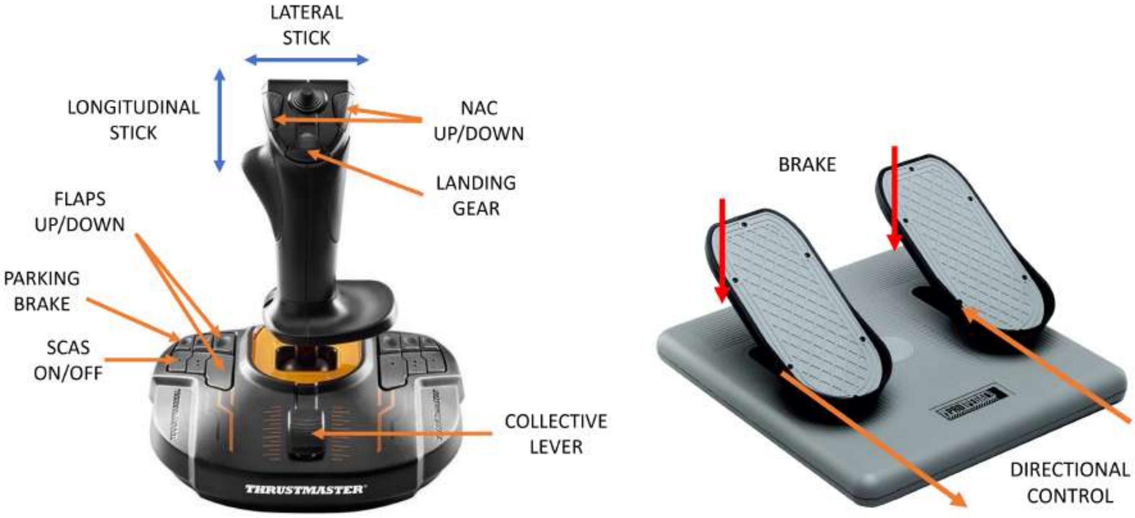
In this context, the research group has already carried out several works. In [13] the GTRS mathematical model has been revised by replacing the rotor dynamic model and improving the computational performance, an essential feature for real-time simulation. Instead, a novel gimballed rotor mathematical model has been introduced in [14], enhancing the GTRS model. Based on the studies presented in [13, 14], a promising tilt-rotor simulation platform has been proposed in [15], where a preliminary real-time simulator of the XV-15 aircraft has been developed for research and teaching purposes. For this reason, the proposed tilt-rotor simulation platform has been designed to be executed on a portable workstation, while ensuring real-time performance.

However, despite the satisfactory results, the developed tilt-rotor flight simulation platform presents several limitations both in the developed mathematical model and in the graphic elements, as discussed in [15]. One of these is identified in the preliminary implementation of the SAS (Stability Augmentation System) which is not working properly resulting in poor handling qualities.

In this paper, an enhanced tilt-rotor flight simulation platform is presented. In particular, the SAS has been revised, as well as a Stability Control and Augmentation System (SCAS) has been implemented. Several simulations have been performed comparing the previously implemented SAS with the new one and demonstrating the effectiveness of the SCAS.

The resulting tilt-rotor flight simulation platform maintains the same benefits presented in [15], being a flight simulator executable on a portable workstation to be used by students and researchers without requiring specific and expensive hardware. Furthermore, the enhanced version presented in this paper presents better handling qualities, so that, the pilot has less difficulty in controlling the simulated tilt-rotor aircraft.

The portable tilt-rotor flight simulation platform presented in this paper is a simplified and adapted version of an advanced tilt-rotor simulator of the XV-15 integrated with the Research and Didactic Flight Simulator (ReDSim) of the ZAV Center of Aviation, at ZHAW Zurich University of Applied Science in Winterthur, Switzerland, as



(a) (b)

Fig. 2 The flight control hardware used by the pilot to provide flight commands. In (a) the flight stick, while in (b) the rudder pedal

discussed in [16]. ReDSim is a multi-purpose real-time flight simulation platform equipped with a professional 170° out-of-the-window-view projection system and a three-channel control loading system to simulate the forces on the pilot controls.

The paper is organized as follows. Section III details the simulation architecture describing the flight control hardware, the tilt-rotor mathematical model and the graphics environment provided by FlightGear. Section IV introduces the design and implementation of SAS and SCAS. The developed tilt-rotor flight simulator is validated with some simulations reported in Section V. Finally, our conclusions are drawn in Section VI.

III. Simulation Architecture

The developed tilt-rotor flight simulator platform implements all the necessary functionalities to ensure a real-time simulation: the interaction with the flight control hardware used by the pilot, the mathematical model that describes the aircraft system and its dynamics, and the graphics environment. The main architecture of the tilt-rotor flight simulator platform is shown in Figure 1. In the following, each element is detailed.

A. Flight Control Hardware

The flight control hardware is used by the pilot to provide flight commands. Specifically, it consists of two commercial USB peripherals: a flight stick and a rudder pedal.

The flight stick is mainly used to control the lateral and longitudinal axis, the flap position, and the position of the nacelles. A slider located at the bottom is used as a collective lever to set the pitch of blades, while the throttle is automatically regulated by the governor that set the speed of the engine based on the position of the nacelles. Other essential commands are set on the flight stick to control the parking brake, and the landing gear, as well as to enable the SCAS or only use the SAS. Figure 2(a) shows the flight stick configuration.

The rudder pedal, shown in Figure 2(b), is used to control the motion on the yaw axis, by acting with a differential cyclic command for the two rotors (in helicopter mode) or with a rudder command (in airplane mode). Furthermore, the rudder pedal is also used to provide a brake command on the landing gear by pressing pedals.

B. Tilt-Rotor Mathematical Model

The mathematical model is the core of the tilt-rotor flight simulator. The model used in this work mainly refers to the model presented in [9, 10], in which the GTRS and its theory are presented. However, the GTRS model is partially

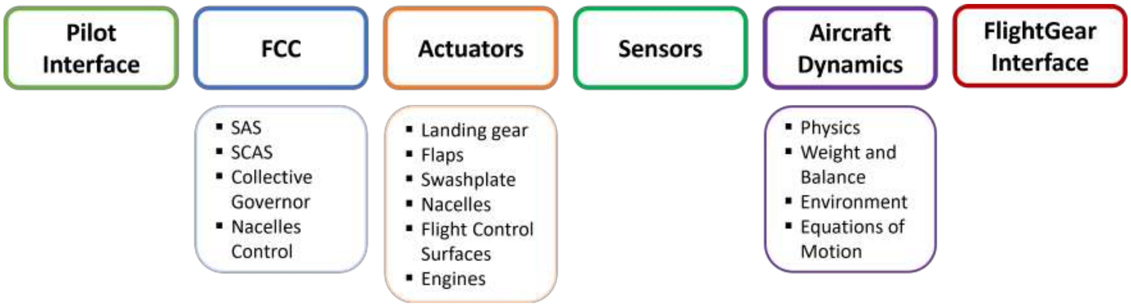


Fig. 3 The main elements included in the tilt-rotor mathematical model

revised and improved as discussed in our previous works in [13, 14].

The mathematical model includes several elements for a complete representation of the tilt-rotor aircraft. The main structure of the mathematical model is shown in Figure 3, in which the main elements of the tilt-rotor model are listed, such as the Flight Control Computer (FCC), actuators, sensors, and the aircraft dynamics, as well as the interfaces with the pilot control input and the graphics environment provided by FlightGear. In the following, each element of the structure of Figure 3 is described. Note that the validation of the mathematical model is not the main purpose of this paper, hence, for an in-depth study and analysis of the adopted model refer to [9, 10, 13, 14].

Pilot Interface

The Pilot Interface aims at processing the flight control commands provided by the pilot via the flight control hardware, i.e. the flight stick and the rudder pedal. Inputs are evaluated, conditioned and normalized to adapt the input command with the typical flight controls of the XV-15 aircraft.

Flight Control Computer

The Flight Control Computer (FCC) consists of several elements used to control the aircraft: a Stability Augmentation System (SAS), a Stability Control and Augmentation System (SCAS), a Collective Governor, and a Nacelle Control System.

The SAS provides control filtering and angular rate stabilization improving the handling qualities in both helicopter and airplane modes. The SCAS further improves the handling qualities by providing a tracking control on a desired pitch and roll angles, and yaw rate. While the SAS is always active, in our simulator the SCAS can be switched off by the pilot with a specific control command. The design and implementation of SAS and SCAS is one of the main contributions of this paper and is detailed in Section IV.

Based on the control command provided by the collective lever that controls the collective pitch, the Collective Governor adjusts the throttle to regulate the rotor speed depending on the deflections of nacelles.

The Nacelle Control System simply provides the command to nacelles actuators based on the input command imposed by the pilot. In fact, the pilot controls the position of the nacelles by using two buttons on the flight stick to set the up and down motion. The single pressure of these buttons enables the automatic deflection of nacelles in predefined positions at 90° (helicopter mode), 60°, and 0° (airplane mode).

Actuators

Typically, hydraulic and electric actuators are very complex systems introducing several non-linearities in the aircraft dynamics. However, all the actuators implemented in our tilt-rotor flight simulator are modeled with simple transfer functions to reduce the computational complexity of the whole system and, then, to ensure real-time performance.

Actuators for flight controls are modeled with first-order transfer functions to reproduce an appropriate delay, typical in hydraulic actuation systems. This delay is essential to simulate a realistic aircraft response both in the pilot control loop and the stabilization system.

Engines are also modeled with a first-order transfer function to introduce a delay between the pilot control input on collective and throttle, based on the flight mode, and the engine response.

The nacelle actuators are also modeled introducing a delay in their actuation and considering a constant actuation speed. This is essential for the development of a proper tilt-rotor flight simulator because the nacelle command is one of the main control to enable the longitudinal motion.

The swashplate is used to translate the pilot control input to cyclic or collective blade variations and is adopted with the helicopter mode. Also, this actuation system is modeled with a first-order transfer function.

Other actuation systems are the landing gear and flaps. Both systems are modeled with a first-order system introducing a proper delay and actuated with a constant velocity.

Sensors

Sensors are mainly used by the FCC to provide the measurements required to close the control loop, as well as to provide information about the aircraft status to the pilot via the Primary Flight Display (PFD). All the sensors are modeled with a first-order transfer function introducing an appropriate measurement delay in the sensor response. Among the most important physical quantities measured by sensors in our flight simulator, there are the aircraft attitude, angular rates, body accelerations, calibrated airspeed, and flight altitude.

Aircraft dynamics

The aircraft dynamics include force and momentum contributions that are integrated into the Equations of Motion (EoM). Specifically, the EoM compute the rigid body dynamics and, then, return the aircraft position, attitude, angular rates, linear velocities, and angular and linear accelerations.

Gravity is simply taken into account as a constant force proportional to the aircraft weight and its contribution is computed considering the Euler angles of the aircraft.

Engines produce a thrust provided by the tilt-rotor engines and due to the exhaust outflow.

Rotors generate a variable force and momentum affecting the aircraft's position and attitude in the helicopter flight mode. The mathematical model of rotors is defined based on the work presented in [13, 14] implementing a multi-blade formulation of the rotor's airloads, and a second-order flapping dynamics that models the gimbal attitude and each blade's elastic flapping motion. The model is also combined with a Pitt-Peters first-order dynamic formulation of the rotor inflow, and a model of the wake-induced velocity fields affecting each of the relevant aerodynamic surfaces of the aircraft.

Aerodynamics are modeled based on the contribution presented in [13, 14], in which the aerodynamics model originally proposed in [10] has been revised. Briefly, aerodynamics evaluates the aerodynamic forces introduced by the fuselage, horizontal and vertical stabilizers, and other elements of the tilt-rotor aircraft.

The landing gear is modeled considering the aerodynamic drag introduced during the motion and the interaction with the ground. Specifically, the model evaluates the friction produced with the ground, the resistance force produced by brakes, and the damping effect due to the relative compression of the landing gear stem.

Another essential factor is the computation of the center of gravity based on the position of nacelles and flight conditions. In fact, as formulated in [10], the center of gravity shifts along the longitudinal axis of the tilt-rotor during the conversion from helicopter mode to airplane mode due to the rotation of nacelles.

Lastly, also the environment has to be modeled providing an atmosphere model for air density, pressure, and temperature. Also, the effect of wind and gusts can be included in the flight simulation.

FlightGear Interface

The tilt-rotor mathematical model computes several aircraft parameters, such as the aircraft position and attitude, velocities, accelerations, and landing gear position, to name a few. This data is sent to the graphics environment provided by FlightGear which displays the aircraft in a realistic three-dimensional environment. More details about the graphics environment and the communication between MATLAB/Simulink© and FlightGear are reported in the following section.

C. FlightGear

As previously discussed, the graphics environment is provided by FlightGear [17], an open-source and cross-platform software for flight simulation developed for research, academic and industrial purposes. FlightGear provides a complete framework for flight simulation with a realistic three-dimensional view of the environment including terrain, buildings, airports, and cities in the world. Specifically, the realistic view is provided by TerraSync, a tool of FlightGear that continuously updates the flight scenario during the simulation, by showing a portion of planet Earth based on the position of the aircraft.

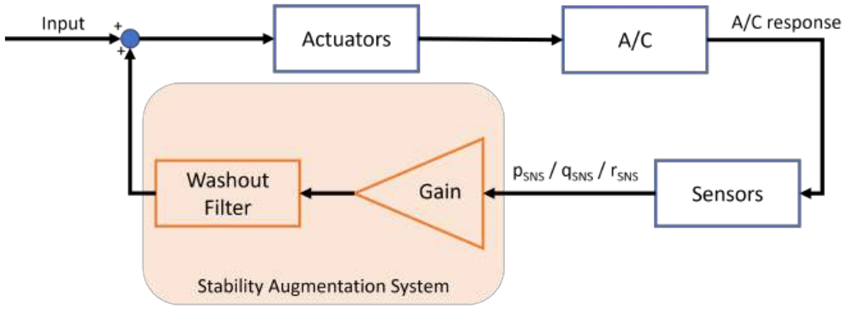


Fig. 4 The block diagram of the SAS implemented in the tilt-rotor flight simulator. The SAS is implemented both on pitch, roll, and yaw rates selecting different feedback gains

In our tilt-rotor flight simulator, FlightGear is exploited only for the visualization of the aircraft in a realistic environment, providing aircraft animation and sound reproduction. In fact, as previously described, the tilt-rotor mathematical model in MATLAB/Simulink© computes the aircraft status such as the aircraft position and attitude, rotational speed of rotors, position of the landing gear and of the flight control surfaces, to name a few. Hence, data is sent to the FlightGear environment that reproduces the aircraft animation and enables the visualization of flight data in the virtual cockpit.

The bi-directional communication between MATLAB/Simulink© and FlightGear is provided using a UDP (User Datagram Protocol) communication. UDP is a lightweight communication protocol suitable for time-sensitive applications.

For the visualization of the aircraft in FlightGear, an aircraft graphics model already present in the database of FlightGear is used, i.e. the Boeing V-22 Osprey, selected for the similarity with the Bell XV-15 aircraft. However, this choice is made only for the visualization of the aircraft. In fact, the V-22 graphics model has been updated by disabling the default simulation of flight dynamics provided by the open-source JSBsim and, then, replaced with the mathematical model developed with MATLAB/Simulink©.

Furthermore, the virtual cockpit has been updated by including a Primary Flight Display (PFD) to provide the aircraft flight data to the pilot. The PFD includes several instruments, such as the compass, the artificial horizon, the altimeter, the calibrated airspeed indicator, the vertical speed indicator, the RPM gauge, the nacelle position gauge, the flap position indicator, and the landing gear indicator.

Figure 9 displays some screen of FlightGear during a simulation showing both the external view of the aircraft and the internal view with the virtual cockpit.

IV. Stability Control and Augmentation System

One of the main contributions of the paper is the design and implementation of the SCAS. As previously discussed, a limitation of the tilt-rotor flight simulator platform presented in [15] was in the SAS which did not work properly, resulting in poor handling qualities. For this reason, a SAS is re-designed and, in addition, it is extended with a SCAS which definitely improves the handling qualities. In the following, both SAS and SCAS are briefly detailed. For an in-depth description of SAS and SCAS refer to [18].

A. Stability Augmentation System

The Stability Augmentation System (SAS) is used to improve the stability of the aircraft and, then, to provide desirable handling qualities. In fact, without the adoption of the SAS, the aircraft responds to control input in its natural short-period motion and the stability of the aircraft varies significantly over the flight envelope and, then, also the handling qualities change. In this paper, a common approach widely used at the state of the art is adopted by providing a pitch, roll, and yaw rate dumper [19]. As result, the SAS provides artificial damping without interfering with the pilot's control input.

The architecture of the adopted SAS is shown in Figure 4, where feedback gains and washout filters are included in the feedback of the closed-loop system. The washout filter is a high-pass filter that washes out steady-state inputs while preserving transient inputs. Instead, the feedback gain increases the damping ratio. Specifically, the diagram block of

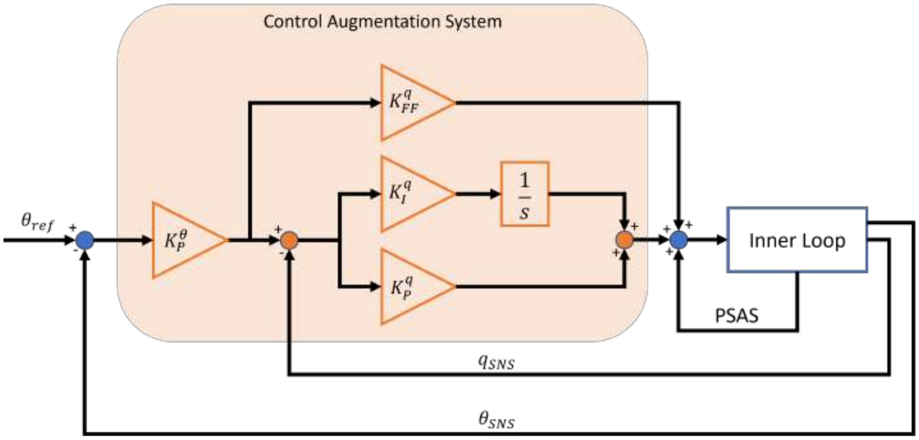


Fig. 5 The block diagram of the pitch hold controller implemented in the SCAS

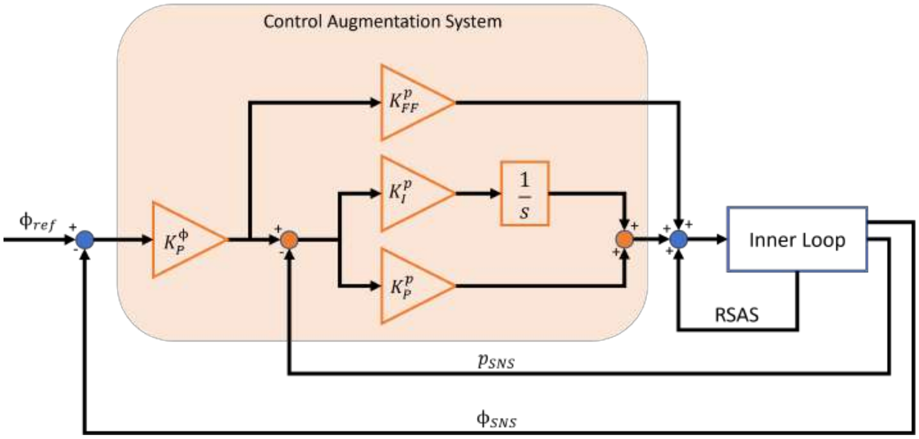


Fig. 6 The block diagram of the roll hold controller implemented in the SCAS

Figure 4 is used both on the pitch, roll, and yaw rates selecting different feedback gains properly selected in a tuning phase.

B. Stability Control Augmentation System

While SAS is used to improve aircraft stability improving handling qualities, a SCAS provides a sort of tracking control on attitude, altitude, and speed reducing the pilot workload.

Specifically, in our simulator, a longitudinal controller, a lateral controller, and a directional controller are implemented. All these controllers are designed as PI (Proportional-Integral) controllers [20]. The choice of this type of controller is mainly made for its simplicity and effectiveness, essential characteristics for their implementation in a flight simulator for teaching and research purposes.

The longitudinal controller provides control of the roll rate of the tilt-rotor aircraft. The block diagram of the controller is shown in Figure 5, in which the inner loop represents the SAS on the pitch rate. In particular, this controller consists of two cascade loops: a pitch rate hold controller implemented as a feedforward augmented PI controller to better reject disturbances; and, a pitch hold controller implemented as a proportional controller.

Similarly, the lateral motion controller is implemented as shown in Figure 6. The inner loop is the SAS on the roll rate and the cascade control loop contains the roll rate hold controller and the roll hold controller designed as a proportional controller and a feedforward augmented PI controller, respectively.

Regarding the directional control, two different controller strategies are adopted based on the aircraft configuration mode. In helicopter mode, a yaw rate hold controller is implemented by tracking a reference yaw rate set by pedals and, then, providing a differential cyclic command for the two rotors. The block diagram of this controller is shown in Figure

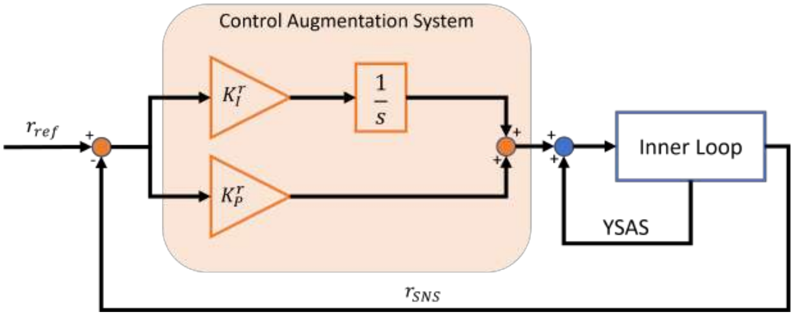


Fig. 7 The block diagram of the yaw rate hold controller implemented in the SCAS

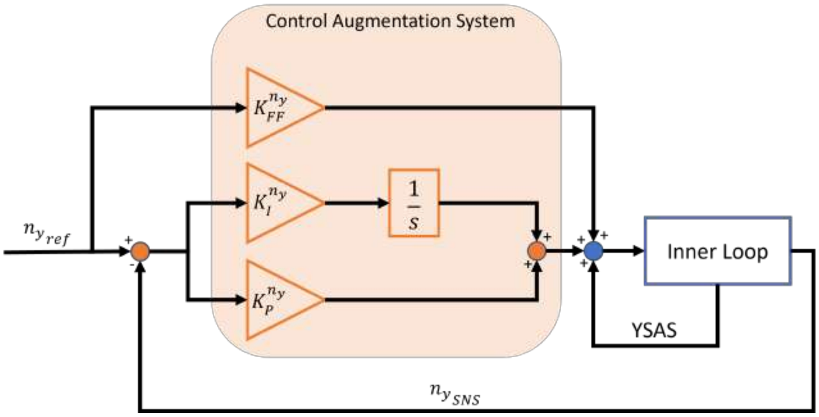


Fig. 8 The block diagram of the turn hold controller implemented in the SCAS

7, in which the inner loop is the SAS on the yaw rate. Specifically, this controller is designed as a PI controller.

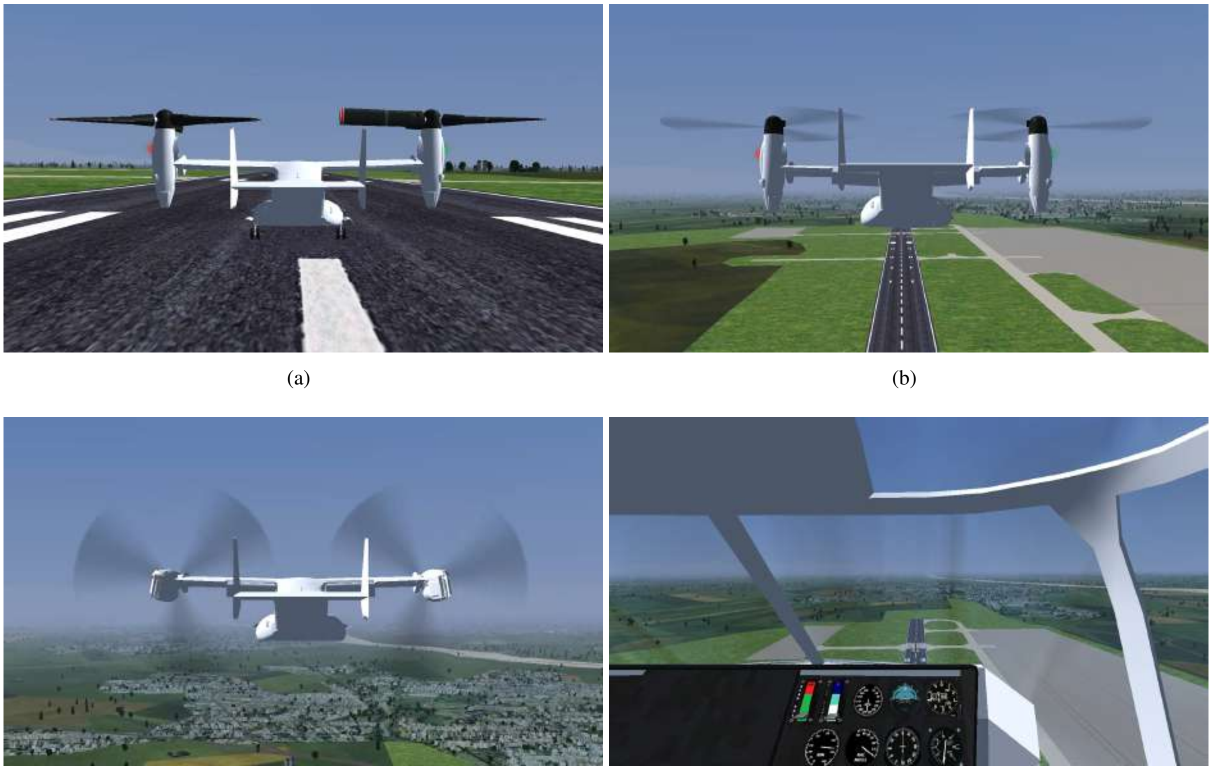
Differently, in airplane mode, a turn hold controller is implemented with the aim of tracking lateral accelerations. This turn hold controller is implemented as shown in the block diagram of Figure 8, designed as a feedforward augmented PI controller.

C. Gain scheduling

The previously described controllers, both for the SAS and SCAS, assume the use of a linear system. For this reason, it is required the linearization of aircraft model that is often performed considering a specific equilibrium point. The resulting controller may work well around the linearization point, but, performance degrades as one drifts away from the equilibrium point due to changing system dynamics. This occurs especially with tilt-rotors because the aircraft dynamics change a lot in the entire flight envelope and, in particular, with the position of nacelles.

In order to solve this problem, a gain scheduling approach is adopted, in which a set of linear controllers are designed for a corresponding set of plant linearizations associated with several operating points. Hence, a gain-scheduling controller is constructed by interpolating the set of linear controllers and then, the parameters of the resulting controller are selected based on the current operating point. In the present tilt-rotor simulator, the operating points are sampled considering the airspeed and, as a consequence, also the position of nacelles assuming the conversion corridor of the XV-15 [2].

As a general consideration, gain scheduling is a comprehensive method for the control of non-linear systems and, since it uses a linear control design, it is computationally efficient and suitable for our portable flight simulator.



(a) (b)

(c) (d)

Fig. 9 Screens captured from the FlightGear simulator. In (a), the aircraft on the ground is visualized from an external point of view. In (b), the tilt-rotor operates in helicopter mode with the position of nacelles at 90°. In (c), the tilt-rotor operates in airplane mode. In (d), the internal view of the aircraft with a Primary Flight Display consisting of several instruments

V. Results

A. Implementation

As previously detailed, the presented tilt-rotor flight simulator is implemented using MATLAB/Simulink$^{©}$ and FlightGear. While FlightGear is mainly used to visualize the aircraft animation in a realistic three-dimensional environment, the core of the simulator is implemented in MATLAB/Simulink$^{©}$ including not only the mathematical model, but also the management of real-time performance. Guarantee real-time performance is an essential feature for human-in-the-loop flight simulators to ensure a simulation with a realistic speed and low latency for a correct interpretation of the flight dynamics by the pilot.

The Simulink$^{©}$ Desktop Real Time (SLDRT) is used to provide a perfect synchronization of the processor clock and physical simulation, ensuring the real-time performance of the model. SLDRT provides real-time execution of Simulink$^{©}$ models providing specific blocks to connect input/output devices.

Thanks to this configuration, the tilt-rotor flight simulator is able to guarantee real-time performance using a commercial and portable laptop (Dell Precision 7550).

B. Controller Tuning

Before performing simulations, a tuning of controllers is required. The controller tuning requires a great effort because of the gain scheduling approach adopted. In fact, a specific controller tuning is performed for each operational region in order to achieve an adaptive control strategy.

Each constant parameter is tuned considering different operational regions evaluating the airspeed and the consequent position of nacelles according to the conversion corridor of the XV-15 aircraft. These values are linearly interpolated

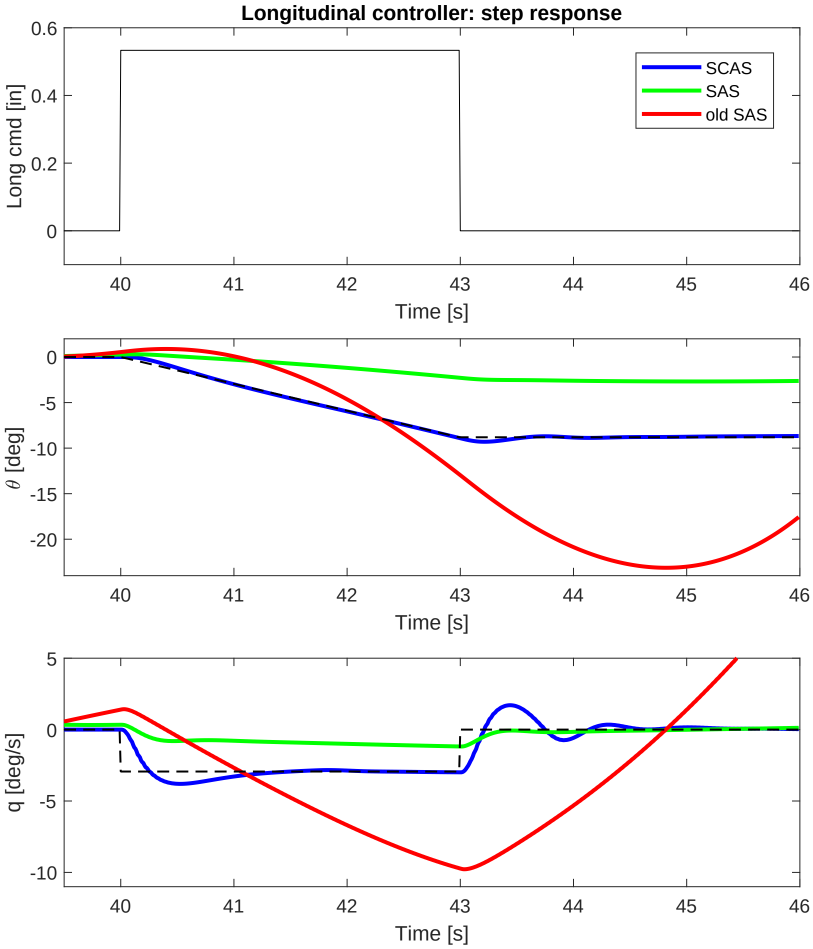


Fig. 10 Aircraft response after a longitudinal input step. In blue the response with the SCAS active. In green is the response with only the SAS. In red is the response of the SAS originally implemented in [15]

and included in a sort of lookup table used to select the proper parameters during the execution of the control system. The tuning of the controller is carried out by observing the aircraft response and obtaining adequate behavior. For a detailed description of tuning refer to [18],

C. Simulations

The tilt-rotor flight simulation platform is successfully implemented and tested by running several simulations. Figure 9 shows some screen captures of the simulation in the FlightGear environment. Figure 9(a) shows an external view of the aircraft in a realistic simulated environment periodically downloaded from TerraSync. As previously discussed, the visualized aircraft model is the Boeing V-22 Osprey, a model already present in the database of FlightGear. This model is only exploited to visualize the aircraft, but the mathematical model implemented in Simulink© considers the Bell XV-15 aircraft. Figures 9(b) and 9(c) show the tilt-rotor aircraft in helicopter and airplane modes, respectively. Instead, Figure 9(d) displays the internal view of the aircraft with the developed PFD consisting of the essential cockpit instruments.

Preliminary simulations are conducted to demonstrate the effectiveness of the implemented SCAS. Figure 10 shows the response of the simulated aircraft after a longitudinal input step command considering a null airspeed in helicopter mode, i.e. with the nacelles at 90°. Results of Figure 10 compare the step response of the SCAS, the SAS, and the previously implemented SAS. With the SCAS enabled, the longitudinal input step acts on the pitch angle with a constant pitch rate. The SCAS performs a tracking control of the reference with satisfactory behavior. Instead, the SAS responds to the step input by acting on the pitch rate maintaining system stability and with a damped response, thanks to the gain

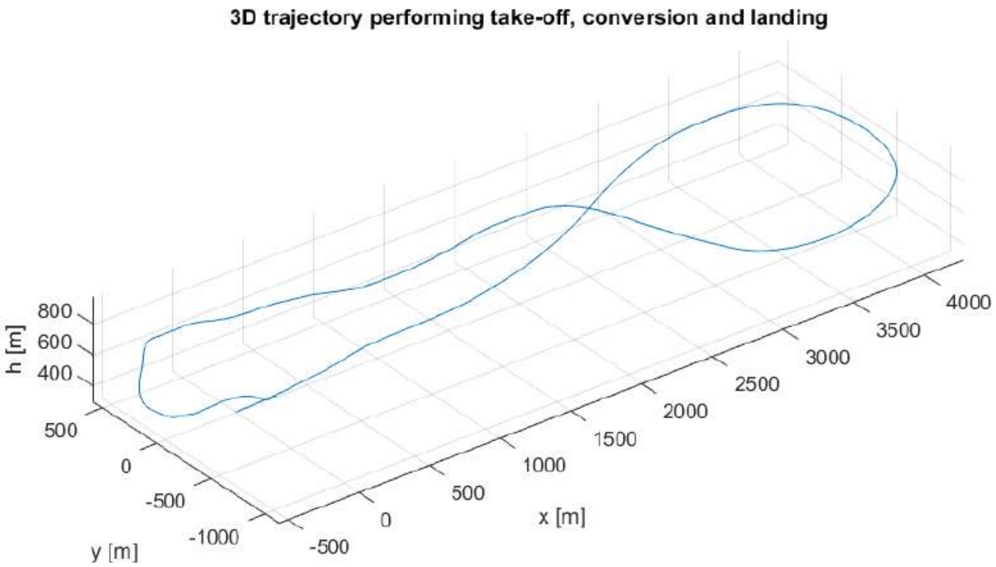


Fig. 11 The three-dimensional trajectory consisting of the vertical take-off, a conversion to airplane mode, and, then, a landing in helicopter mode

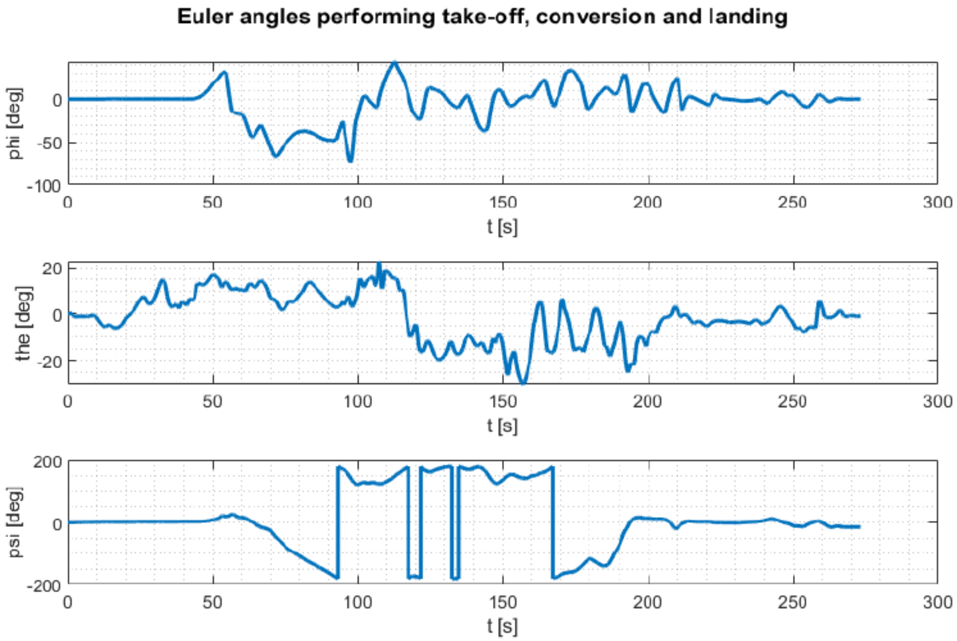


Fig. 12 The attitude angles during the simulation test of Figure 11

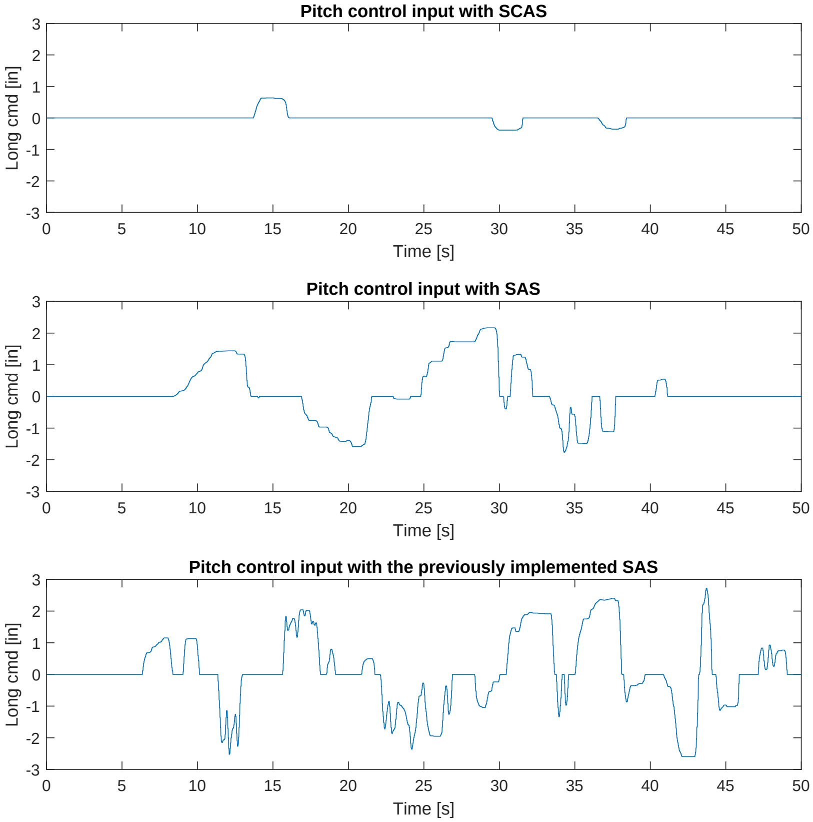


Fig. 13 The longitudinal control input provided by the pilot during a take-off in helicopter mode followed by a conversion to airplane mode. The longitudinal control input is analyzed considering a flight with the SCAS active, a flight with only the SAS, and with the SAS originally implemented in [15]

feedback and the washout filter properly tuned. On the other hand, the performance of the SAS implemented in [15] is completely different. The system response is not damped and, if no other correction commands are imposed, the system diverges toward instability.

Once SCAS and SAS have been validated, several simulations have been performed. Figure 11 shows a complete flight consisting of a vertical take-off followed by a conversion to airplane mode by progressively turning the nacelles forward. At the end of conversion, the aircraft performs a landing in helicopter mode close to the initial position. Thanks to the developed SCAS, the handling qualities of the tilt-rotor aircraft have definitely improved allowing to easily execute the trajectory of Figure 11. Figure 12 plots the attitude angles during the simulation coherent with Figure 11.

During the simulation test of Figure 11 the real-time performance is assessed by evaluating the overall latency introduced by the interaction with FlightGear and the latency introduced by the developed mathematical model. As result, as also proved and discussed in [15], the average latency is below 2.5 ms except for the first few seconds of simulation, in which a delay of up to 40 ms occurs mainly caused by the TerraSync Server that loads the simulated scenario at the beginning of the simulation and requiring specific computational resources. To summarize, the overall latency is definitely acceptable to guarantee real-time performance with an update frequency of 50/60 Hz even using a commercial portable workstation.

The improvement of handling qualities and its effect on the pilot workload can be preliminary assessed in the results of Figure 13, in which the longitudinal control input provided by the pilot using the flight stick is plotted performing a vertical take-off in helicopter mode and a conversion to the airplane mode. With the SCAS active, the pilot acts only a few control commands on the pith angle to provide the longitudinal motion and to make some corrections on the pitch during the conversion. Differently, with SAS only, the pilot acts several commands to maintain a desirable pitch during the flight. Anyway, the effect of the SAS is definitely appreciable by the pilot during the execution of the simulated flight. On the other hand, using the SAS originally implemented in [15], the pilot must impose a consistent level of commands to maintain a stable flight due to the poor handling qualities.

This last test demonstrates that an enhanced implementation of the SCAS increases consistently the handling qualities of the tilt-rotor aircraft providing an easy to fly aircraft. Moreover, the pilot reports a significant improvement in piloting, after comparison of the revised SAS with the one originally implemented in [15].

VI. Conclusions

In this paper, a tilt-rotor flight simulator for research and teaching purposes is presented. The tilt-rotor model is implemented in MATLAB/Simulink© and is interfaced with a pilot control hardware consisting of a flight stick and rudder pedal. The graphics environment is provided by FlightGear which enables the aircraft visualization in a realistic three-dimensional environment.

Compared with the tilt-rotor flight simulation platform previously presented in [15], the simulator implements a SCAS that enhances aircraft stability and definitely improves handling qualities. Moreover, even if the SCAS is turned off, the revised SAS has better performances compared with the SAS originally implemented in [15] by providing artificial damping without interfering with the pilot's control input.

As result, the developed flight simulation platform provides a real-time simulation of the XV-15 aircraft with adequate handling qualities. The main benefit is the possibility of executing the flight simulator on a portable workstation with commercial hardware and, then, without requiring specific and expensive hardware, an essential feature to facilitate the use of the simulator to students and researchers.

Future works will include improvements of graphics elements, such as the implementation of a graphic model that represents the Bell XV-15 aircraft, as well as the development of a realistic virtual cockpit that reproduces the one on board the XV-15.

Acknowledgments

This activity is part of a Doctoral Research Program (Ph.D. in Aerospace Engineering at Politecnico di Torino) and M.Sc. in Aerospace Engineering at Politecnico di Torino (Mr. Alessandro Briamonte) supported by Zurich University of Applied Sciences, Centre for Aviation.

References

[1] Padfield, G. D., *Helicopter flight dynamics: including a treatment of tiltrotor aircraft*, John Wiley & Sons, 2018.

[2] Maisel, M. D., *The history of the XV-15 tilt rotor research aircraft: from concept to flight*, National Aeronautics and Space Administration, Office of Policy and Plans, NASA, 2000.

[3] Silva, C., Johnson, W. R., Solis, E., Patterson, M. D., and Antcliff, K. R., "VTOL urban air mobility concept vehicles for technology development," *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 3847.

[4] Misra, A., Jayachandran, S., Kenche, S., Katoch, A., Suresh, A., Gundabattini, E., Selvaraj, S. K., and Legesse, A. A., "A Review on Vertical Take-Off and Landing (VTOL) Tilt-Rotor and Tilt Wing Unmanned Aerial Vehicles (UAVs)," *Journal of Engineering*, Vol. 2022, 2022.

[5] Derrouaoui, S. H., Bouzid, Y., Guiatni, M., and Dib, I., "A comprehensive review on reconfigurable drones: Classification, characteristics, design and control technologies," *Unmanned Systems*, Vol. 10, No. 01, 2022, pp. 3–29.

[6] Dreier, M. E., *Introduction to helicopter and tiltrotor flight simulation*, American Institute of Aeronautics and Astronautics, 2007.

[7] Dugan, D. C., Erhart, R. G., and Schroers, L. G., "The XV-15 tilt rotor research aircraft," Tech. rep., NASA, 1980.

[8] Churchill, G. B., and Dugan, D. C., "Simulation of the XV-15 tilt rotor research aircraft," Tech. rep., NASA, 1982.

[9] Harendra, P., Joglekar, M., Gaffey, T., and Marr, R., "V/STOL tilt rotor study. Volume 5: A mathematical model for real time flight simulation of the Bell model 301 tilt rotor research aircraft," Tech. rep., NASA, 1973.

[10] Ferguson, S. W., "A mathematical model for real time flight simulation of a generic tilt-rotor aircraft," Tech. rep., NASA, 1988.

[11] Kleinhesselink, K. M., *Stability and control modeling of tiltrotor aircraft*, University of Maryland, College Park, 2007.

[12] McVicar, J. S. G., "A generic tilt-rotor simulation model with parallel implementation," Ph.D. thesis, University of Glasgow, 1993.

[13] Barra, F., Capone, P., Monstein, R., Godio, S., and Guglieri, G., "Implementation of a comprehensive mathematical model for tilt-rotor real-time flight simulation," *45th European Rotorcraft Forum*, 2019.

[14] Abà, A., Barra, F., Capone, P., and Guglieri, G., "Mathematical modelling of gimballed tilt-rotors for real-time flight simulation," *Aerospace*, Vol. 7, No. 9, 2020, p. 124.

[15] Primatesta, S., Barra, F., Guglieri, G., and Capone, P., "Design and Integration of a Tilt-Rotor Flight Simulation Platform," *78th Annual Forum and Technology Display, Fort Worth, TX, USA, 10-12 May 2022*, Vertical Flight Society, 2022.

[16] Barra, F., "Mathematical Modelling of Tilt-Rotor Aircraft Configurations A comprehensive model for flight control system development and real-time piloted simulation," Ph.D. thesis, Politecnico di Torino, 2022.

[17] FlightGear developers & contributors, "FlighGear," `https://www.flightgear.org/`, 2022. Accessed: 2022-03-30.

[18] Briamonte, A., "Development of a Control Augmentation System for a Tilt-Rotor Aircraft," Master's thesis, Politecnico di Torino, 2022.

[19] Nelson, R. C., *Flight stability and automatic control*, Vol. 2, WCB McGraw Hill, 1998.

[20] Ellis, G., *Control system design guide: using your computer to understand and diagnose feedback controllers*, Butterworth-Heinemann, 2012.