



**POLITECNICO**  
**MILANO 1863**

SCUOLA DI INGEGNERIA INDUSTRIALE  
E DELL'INFORMAZIONE

# Aeroswitch project: fidelization of the flight simulator of a Distributed Electric Propulsion airplane

TESI DI LAUREA MAGISTRALE IN  
AERONAUTICAL ENGINEERING - INGEGNERIA AERONAUTICA

Author: **Stefano Agnati**

Student ID: 944473

Advisor: Prof. Lorenzo Trainelli

Co-advisors: Stefano Cacciola, Carlo Emanuele Dionigi Riboldi

Academic Year: 2021-22



# Abstract

The interest for electric aviation is nowadays largely diffused. The use of electric engine on aircraft has opened the doors to novel aircraft architecture, such as the Distributed Electric Propulsion (DEP). To demonstrate qualities and limits of this technology, an existing radio-commanded aircraft has been modified with the use of DEP: like this the *SwitchMaster* was borned. In parallel, a flight simulator of *SwitchMaster*, running on X-Plane through a Simulink interface, has been developed. In the present work it is taken up the baton of these existing studies, in order to improve the flight simulator. In particular, it has been updated the Simulink interface in order to send to X-Plane some custom inputs. This permitted to validate the fidelity of the simulator, not tested herein, replicating experimental flight test on it, and comparing flight data. Once the directional dynamics has been identified as the critical one, an update of the geometries of flight model has been performed through an iterative procedure, based on the minimization of the prediction error. The outcome was an aircraft characterized by a lateral dynamics closer to the real one, even if the fidelity is not fully achieved. The work will also show the advantages and limitations of the proposed procedure.

**Keywords:** Distributed Electric Propulsion, Flight Simulator, Lateral dynamics, Flight testing, UDP communication, Aircraft Modelling.





# Sommario

L'interesse per l'aviazione elettrica è ampiamente diffuso al giorno d'oggi. L'uso del motore elettrico sugli aeromobili ha aperto le porte a configurazioni innovative, come la propulsione elettrica distribuita. Per dimostrare i limiti e pregi di tale tecnologia, è stato modificato un aeromodello esistente tramite l'installazione di tale propulsione: è nato così lo *SwitchMaster*. Parallelamente, è stato sviluppato il suo simulatore di volo in ambiente X-Plane, funzionante attraverso un'interfaccia Simulink. Questa tesi raccoglie il testimone dei lavori sopraccitati per migliorare il simulatore di volo. In particolare, è stata aggiornata l'interfaccia Simulink al fine di mandare a X-Plane comandi di volo personalizzati. Questo ha permesso di validare lo stato di fidelizzazione del modello su X-plane, cosa non fatta in precedenza, replicando alcune prove in volo eseguite sul velivolo fisico. Da queste verifiche è emerso che vi era una criticità nello stato di fidelizzazione della dinamica laterale: dunque è stata sviluppata una procedura iterativa, basata sulla minimizzazione di un funzionale, per aggiornare le geometrie del modello di volo. Il risultato della procedura ha migliorato le prestazioni del modello, anche se la fidelizzazione non è del tutto completa. L'elaborato analizza anche i vantaggi e i limiti della procedura seguita.

**Parole chiave:** Propulsione elettrica distribuita, Simulatore di volo, dinamica laterale, Sperimentazione in volo, comunicazione UDP, Modellazione di aeroplani.



# Contents

<b>Abstract</b>	<b>i</b>
<b>Sommario</b>	<b>iii</b>
<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>ix</b>
<b>List of symbols</b>	<b>xi</b>
<b>Acronyms</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Electrification of aircraft . . . . .	1
1.1.1 Advantages of electric propulsion . . . . .	1
1.1.2 The distributed electric propulsion . . . . .	2
1.2 AeroSwitch concept . . . . .	3
1.2.1 The <i>Trybrid</i> . . . . .	3
1.2.2 The <i>SwitchMaster</i> . . . . .	4
1.3 <i>SwitchMaster</i> flight simulator . . . . .	5
1.3.1 Choice of X-Plane: blade element theory and UDP protocol . . . . .	5
1.3.2 Simulink interface . . . . .	6
1.3.3 Reliability of simulator . . . . .	7
1.4 Goal of the present work . . . . .	8
<b>2 Using X-Plane for simulating experimental tests</b>	<b>9</b>
2.1 Inputting flight data signals . . . . .	9
2.1.1 X-Plane output data Simulink update . . . . .	9

2.1.2	Time interpolation . . . . .	11
2.2	Validation . . . . .	13
<b>3</b>	<b>Update of X-Plane aircraft model</b>	<b>17</b>
3.1	Model update . . . . .	17
3.1.1	Iterative repetition of flight test . . . . .	17
3.1.2	Functional evaluation . . . . .	18
3.2	Test of the method . . . . .	20
<b>4</b>	<b><i>SwitchMaster</i> model identification updating</b>	<b>23</b>
4.1	Flight model fidelity . . . . .	23
4.2	Updating process of directional dynamics . . . . .	26
4.2.1	Mathematical model modification . . . . .	26
4.2.2	Parameter selection . . . . .	27
4.3	Analysis and results . . . . .	29
4.3.1	Case 1: vertical tail longitudinal position and scale factor . . . . .	29
4.3.2	Case 2: vertical tail volume, scale factor, and rudder percentage . . . . .	32
4.3.3	Case 3: best vertical tail volume, scale factor, and rudder percentage . . . . .	34
4.3.4	Comparative analysis . . . . .	37
4.3.5	Final configuration . . . . .	44
<b>5</b>	<b>Conclusions</b>	<b>47</b>
5.1	Future developments . . . . .	48
	<b>Bibliography</b>	<b>49</b>
<b>A</b>	<b>Appendix</b>	<b>51</b>
A.1	MatLab routines . . . . .	51
A.1.1	Interpolator block . . . . .	51
A.1.2	Iterative routine . . . . .	51
A.2	Simulations flight data . . . . .	55
	<b>Ringraziamenti</b>	<b>59</b>

## List of Figures

1.1	NASA X-57 Maxwell. . . . .	2
1.2	The <i>Trybrid</i> . . . . .	4
1.3	Aerodynamic properties of <i>SwitchMaster</i> , from [6]. For these test point, free stream velocity of wind tunnel was set on 15 m/s. . . . .	6
1.4	Simulink block structure: receiver, elaborating function, and sender. . . . .	7
2.1	Receiver block. . . . .	11
2.2	Simulink Interpolation block 1. . . . .	12
2.3	Robin DR401. . . . .	13
2.4	Comparison between target flight data and repeated flight data. . . . .	14
2.5	Comparison between target flight data and repeated flight data. . . . .	15
2.6	Comparison between target flight data and repeated flight data with the insertion of the advance. . . . .	16
3.1	Geometrical parameter modified for flight tests. . . . .	20
3.2	Functional plot . . . . .	21
4.1	<i>SwitchMaster</i> real and simulated aircraft. . . . .	24
4.2	Geometrical parameters of <i>SwitchMaster</i> . . . . .	28
4.3	Result analysis and aircraft configuration for different values of $Z_{ARM}$ and $p$ . In both the functionals, Minimum is highlighted by the green point. . . . .	30
4.4	Flight histories for analysis on $Z_{ARM}$ and $p$ . . . . .	31
4.5	Aircraft configuration for the analysis conducted on $C_{VT}$ , $p$ and $c$ . . . . .	33
4.6	Flight histories for analysis on $C_{VT}$ , $p$ and $c$ . . . . .	33
4.7	Result analysis and aircraft configuration for different values of $p$ and $c$ with a constant value of $C_{VT}$ . In both the functionals, Minimum is highlighted by the green point. . . . .	35
4.8	Flight histories for analysis on $p$ and $c$ with a constant value of $C_{VT}$ . . . . .	36
4.9	Comparative plot for Case 1, Case 2, and Case 3. . . . .	38
4.10	Error on $r$ , $\psi$ and $\beta$ in time domain. . . . .	40
4.11	Error on $r$ , $\psi$ and $\beta$ in frequency domain. . . . .	41

4.12	Error on $r$ , $\psi$ and $\beta$ in time domain during free response. . . . .	42
4.13	Error on $r$ , $\psi$ and $\beta$ in time domain during free response. . . . .	43
4.14	<i>SwitchMaster</i> in its updated configuration flying. . . . .	44
A.1	Flight data of validation process with Robin 401. . . . .	55
A.2	Dutch roll test experimental data compared with baseline. . . . .	56
A.3	Roll test experimental data compared with baseline. . . . .	57

## List of Tables

1.1	<i>Trybrid</i> main specifications, from [6]. . . . .	3
1.2	<i>SwitchMaster</i> specifications, from [6]. . . . .	4
2.1	Surfaces deflection and throttle level. . . . .	9
2.2	Dataset in output from X-Plane, divided accordingly with class of data. . .	10
2.3	<i>Trybrid</i> main specifications, from [6]. . . . .	13
3.1	Mathematical implementation validation test results . . . . .	20
4.1	<i>SwitchMaster</i> baseline geometrical parameter value. . . . .	28
4.2	$Z_{ARM}$ and $p$ and $c$ range and number of steps for current analysis. . . . .	29
4.3	Results for current analysis. . . . .	30
4.4	$C_{VT}$ , $p$ and $c$ range and number of steps for current analysis. . . . .	32
4.5	Results for current analysis. . . . .	32
4.6	$p$ and $c$ range and number of steps for current analysis. . . . .	34
4.7	Results for current analysis. . . . .	35
4.8	Dimensions recap. . . . .	37
4.9	Error in time. . . . .	39
4.10	Error in frequency. . . . .	41
4.11	Score table. . . . .	45





## List of symbols

$C_D$	Drag coefficient	$\theta$	Parameter vector
$C_L$	Lift coefficient	$\delta_A$	Aileron deflection
$C_{ROOT}$	Vertical tail root chord	$\delta_E$	Elevator deflection
$C_{RUD}$	Rudder chord	$\delta_R$	Rudder deflection
$C_{TIP}$	Vertical tail root chord	$\delta_T$	Throttle level
$C_{VT}$	Tail volume ratio	$\eta$	Percentage decrease of error
$H$	Vertical tail span	$\phi$	Roll angle
$S_{VT}$	Vertical tail surface	$\psi$	Yaw angle
$S_W$	Wing surface	$\theta$	Pitch angle
$V_t$	Vertical stabilizer span	$b_W$	Wing span
$V_{MC}$	Vertical tail mean chord	$c$	Rudder mean chord over vertical tail mean chord ratio
$Z_{ARM}$	Longitudinal position of vertical tail	$p$	Roll rate or area scale factor parameter
$Z_{CG}$	Longitudinal position of center of gravity	$q$	Pitch rate
$\alpha$	Angle of attack	$r$	Yaw rate
$\beta$	Angle of sideslip		



# Acronyms

AIAA American Institute of Astronautics

DEP Distributed Electric Propulsion

GA General Aviation

IAS Indicated Airspeed

UDP User Datagram Protocol



# 1 | Introduction

This Master thesis work is the natural development of Aeroswitch project [6], born in the context of academic course *Aircraft Design* at Politecnico di Milano and presented at the 2020 American Institute of Astronautics (AIAA) competition for graduate students. In this chapter it will be briefly summed up, taking in account main aspects and main open points, that will be the starting point for different developments.

## 1.1. Electrification of aircraft

Nowadays, lots of pressure comes from public opinion and customers to improve the effort in designing greener transportation vehicles. However, while electrification of ground applications is well underway, on aircraft is still in the early stages of development. In fact, its realization, especially for wide body implementations, is more problematic because of many critical aspects, like the fact that higher energy densities and powers are required. Lots of studies have been conducted in the last decades to improve aircraft electrification and some concepts are starting to come together. For example, the idea of partially substituting mechanical and hydraulic connections with electrical ones, and then only locally transform energy, or *power by wire* concept, is of great interest. In any case, the propulsive subsystem requires more attention since, in a conventional aircraft, emits greenhouse gases such as carbon dioxide, nitrous oxides, sulfates, and soot and is therefore extremely harmful for the environment. Another important issue, on this behalf, is noise pollution, which could be reduced through silent electric motor [2].

### 1.1.1. Advantages of electric propulsion

Hydrocarbon fuels are the most used prime movers for aerospace applications because of their superior energy densities. However, the following advantages can be achieved by implementing an electric motor:

- Higher efficiency and power-to-weight ratio;
- Less or no pollution in the case of hybrid or full electric journeys;

- Silent propulsion for flight over populated areas or night departures and landings;
- Lower operation costs, in terms of maintenance and fuel (note that, on average, 51% of total operative costs are due to fuel expenses);
- Safety increase with redundant energy sources.

Since there are still great challenges to overcome in order to design 100% electric aircraft, hybrid electric propulsion systems could be the short term winning solution. In fact, they represents a trade-of to take advantage of both technologies, a combustion engine and an electric motor [4]. Its important to underline that the electric motor clears the way to new architectures for improving efficiency such as DEP, described in following section.

### 1.1.2. The distributed electric propulsion

DEP concepts for aircraft is one of nowadays most discussed research topics in aeronautics, since it improves some significant performance parameters. The key aspect of DEP is the novel approach of using electrically-driven propulsors which are not mechanically, but electrically connected to energy sources. As a result, propulsors can be placed, sized and operated with greater flexibility to leverage the synergistic benefits of aero-propulsive coupling. In particular, the set of propellers creates a blowing effect on the wing, which increase the relative velocity of free stream on airfoils: as a result, equal lift is produced with smaller surface, then weight. At the moment, there are some experimental implementations of fixed wing of short and vertical take-off aircraft. NASA's "Maxwell X-57" is for sure one of the most promising. Its first flight is planned for the nearby future. It is a hybrid vehicle which uses twelve smaller distributed electric propellers, and two bigger ones on the tip of the wing as shown in fig. 1.1.



Figure 1.1: NASA X-57 Maxwell.

## 1.2. AeroSwitch concept

As outcome of *Aircraft Design* course, held by professor Lorenzo Trainelli and professor Carlo E.D. Riboldi, AeroSwitch team projected the *Trybrid*. This aircraft exploits the advantages of DEP to simulate the behaviour of a single or multi-engine, propeller driven, trainer. The concept of *Trybrid* is a clever attempt to link the green features of an electric aircraft with the novel aero-propulsive features of DEP.

### 1.2.1. The *Trybrid*

One of the key aspects of DEP is the possibility to control yaw using propulsor instead of rudder. This feature has been exploited to develop a General Aviation (GA) trainer aircraft that can simulate the non-symmetrical behaviour of a single engine aircraft, or symmetric behaviour of multi-engine aircraft. This aircraft, the *Trybrid*, is a high wing, tricycle gear, six electric engines aircraft; its main specifications are reported in table 1.1. The clever solution of propulsive configuration faces the problem of high costs of training for new pilots: first of all, the operational costs of an electric aircraft are greatly lower than a traditional aircraft; secondly, the possibility to train new pilots with just a machine is an advantage for flight schools, which can operate just one aircraft instead of two or more, saving money consequently: this is an opportunity make licences more affordable, then training efficiently more new pilots. A part from the scope, technically the flight mode can be chosen through a switch on ground: once airborne, the outer propulsor work at different power with respect to the inner one, in order to create a yawing moment that simulate the single engine, propeller driven, behaviour. In fig. 1.2 is reported a rendering of the aircraft.

Table 1.1: *Trybrid* main specifications, from [6].

	Unit	Value
Wing span	m	10.50
Wing surface	$m^2$	15.76
Weight	kg	1038.7
Stall speed	kt	45
Never exceed speed	kt	80
Installed power	hp	117.2



Figure 1.2: The *Trybrid*.

### 1.2.2. The *SwitchMaster*

The first development of Aeroswitch work is *SwitchMaster*. This is an RC model, realized in context of Master Thesis work, intended to demonstrate the concept of *Trybrid*. The scaled demonstrator has been realized modifying an existing RC model equipped with a single standard propulsor, through the installation of DEP. It is a high wing aircraft, with conventional landing gear, equipped with six electric engines. In particular, three propellers per semi-wing were mounted on leading edge. Main specification are reported in table 1.2. The purpose was to demonstrate following points:

- Multi-engine flight mode, in which the aircraft shall demonstrate normal flight behavior;
- Multi-engine One Engine Inoperative (OEI) simulation, employed for multi-engine pilot training;
- Single-engine normal flight mode, in which the aircraft shall demonstrate a behavior similar to that of a conventional propeller-driven single-engine aircraft, namely generating a yawing moment when adding power;
- Single-engine One Electric Motor Inoperative (OEMI) emergency simulation, in order to assure safety is always guaranteed also for a single-engine rated pilot.

Table 1.2: *SwitchMaster* specifications, from [6].

	Unit	Value
Wing span	m	2.13
Length	m	1.65
Wing surface	$m^2$	0.52
Weight	kg	4.5
Maximum thrust	N	93



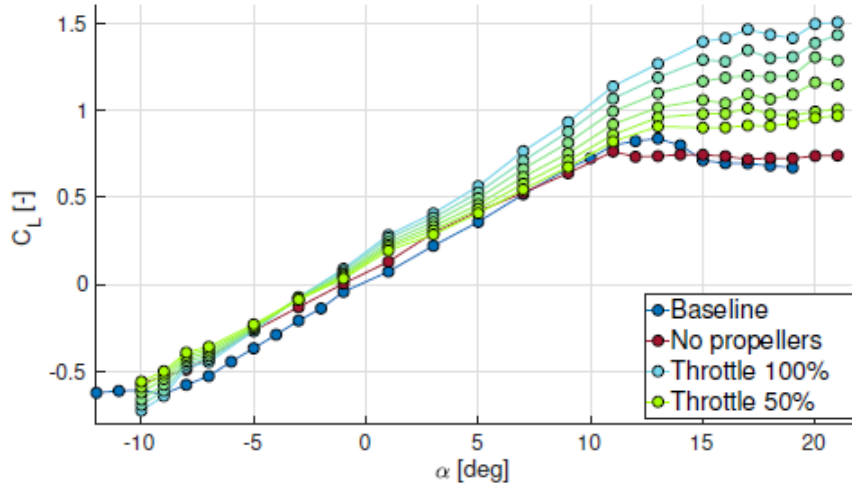
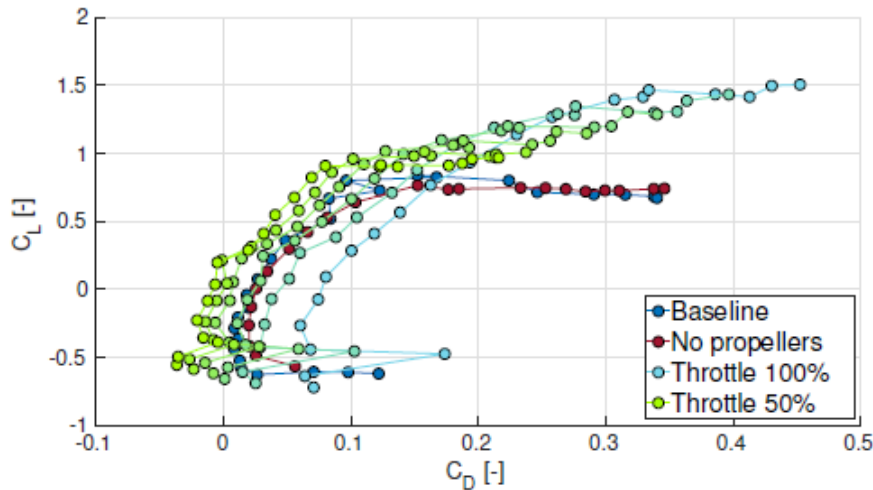
Through a reproduction of wing, it has been possible to measure in a wind tunnel the difference in terms of lift coefficient  $C_L$  between DEP aircraft and standard one. Main results are plotted in fig. 1.3. As it is possible to observe, a great improvement of  $C_L$  is obtained through the effect of blowing, which strongly depend on throttle level. On the other hand, the increasing of  $C_L$  correspond to a increasing of  $C_D$ . DEP effects have been investigated in one of its first flying practical realizations. The accomplished flight test activity with *SwitchMaster* confirmed the predicted increase in lift coefficient due to wing blowing. Conversely, at low throttle settings, the propeller windmilling highly disturbed the airflow on the wing, negatively affecting flight conduction. As a consequence, the landing phase should be performed with high RPM and a reduced flare, in order to face the different handling qualities of the aircraft. Despite handling difficulties at low speed, the single engine and multi-engine behaviour was successfully demonstrated: in particular, single-engine propeller effects could be isolated with the conceived flight test techniques, consisting in the imposition of maximum throttle at constant speed or altitude. Moreover, the simulation exhibited the generation of sideslip, turn rate and bank angle at high-power regimes, in perfect harmony with a real GA aircraft.

### 1.3. *SwitchMaster* flight simulator

After the *Trybrid*, in parallel with *SwitchMaster* physical aircraft building and testing, a flight simulator of it was developed using X-Plane<sup>©</sup> and Simulink<sup>®</sup>. Main purpose and result are summed up in the following, integral text is reported in [1].

#### 1.3.1. Choice of X-Plane: blade element theory and UDP protocol

The flight model of *SwitchMaster* runs on X-Plane. The powerful of this flight simulator is a user interface, Plane Maker, in which it is possible to model new aircraft, giving as input the geometries and the propulsive features. Through the blade element theory, all the forces and moments are evaluated at each integration time step on a single element of the model. Later, all forces are summed up to obtain the resultant force: through an integration it is possible to obtain velocity, then position. This method permit to describe the dynamic behaviour of aircraft with the just knowledge of geometry, bypassing the stability derivatives system. Regarding the latter, it is used by other simulators to model the dynamic behaviour. Through this specific feature of X-Plane a model of *SwitchMaster* was developed. Another advantage of X-Plane is the possibility to communicate with external software via User Datagram Protocol (UDP) protocol. Its peculiarity is that no

(a)  $C_L$  vs  $\alpha$  curve.

(b) Drag polar curve.

Figure 1.3: Aerodynamic properties of *SwitchMaster*, from [6]. For these test point, free stream velocity of wind tunnel was set on 15 m/s.

reception control of the message sent is performed: this downgrade the safety, but ensure an high rate of words sent.

### 1.3.2. Simulink interface

Simulink is a MatLab<sup>TM</sup> application in which it is possible to visualize different functions through blocks, and flow of data through arrows. Simulink is able to receive and send a UDP word, with a proper block. In the simulator architecture, Simulink had the function of receiving the input of a controller, connected via USB, and sending instruction to surfaces and each engine of *SwitchMaster*, running on X-Plane. Moreover, an elaborating

function was programmed in order to simulate the single or multi-engine behaviour. A working scheme is reported in fig. 1.4. In this manner it was possible to govern the aircraft model with the same type of joystick of real aircraft, choosing the flight mode, but on a flight simulator.

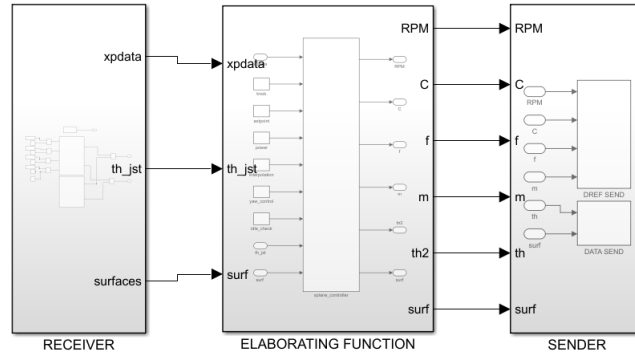


Figure 1.4: Simulink block structure: receiver, elaborating function, and sender.

### 1.3.3. Reliability of simulator

The main problem to be solved was reliability of simulator, in the sense that a same input given to simulator or to real aircraft must provide the same output. This must be a fundamental requirement of a good simulator. Test action has been performed repeating some real flight test maneuvers with USB joystick: the result obtained was not precise, because it is difficult to repeat exactly the same input given to real aircraft. In fact, in future development of [1] following statements are reported:

- *Automatic simulator control input in substitution of RC emulator: possibility to evaluate airplane responses to specific inputs; possibility to replicate exactly command inputs of real airplane flight testing maneuvers to evaluate differences in aircraft behavior.*
- *Complete stability derivative identification of the real Switch Master flight model and of the simulator one: better identification of simulator model and error sources.*
- *Closed loop simulator trim: better airplane controllability and more accurate conducting of RC emulator based flight testing maneuvers.*
- *Designed yaw control testing on the real airplane.*

## 1.4. Goal of the present work

Some of the conclusions reported in section 1.3.3 are the goal for this thesis. In particular, this work will be focus on:

- Coupling X-Plane with Simulink, in order to send to the simulator an input stored during experimental flight test;
- Development of a procedure to replicate experimental flight test on the simulator;
- Updating of X-Plane aircraft model from experimental data.

## 2 | Using X-Plane for simulating experimental tests

In this chapter the problem of communication between X-plane simulator and Simulink will be faced. In particular, it will be described how to send an experimental dataset, stored during a real flight test, to the simulator, in order to reproduce the flight test itself. In the end, the Simulink interface will be validated.

### 2.1. Inputting flight data signals

First action to perform is the one of modifying the existing Simulink interface, presented in section 1.4, in order to command the model aircraft with a input given to the real one. The input is a set of four vector, stored by telemetry during real flight test on *SwitchMaster*, which correspond to the three surfaces deflection and throttle, given at each time step, with a sample time of 0.1 s. Values of surface vectors space from -1 to 1 or 0 to 1, according with table 2.1.

Table 2.1: Surfaces deflection and throttle level.

	Symbol	-1	0	1
Elevator	$\delta_E$	full down		full up
Rudder	$\delta_R$	full left		full right
Aileron	$\delta_A$	roll right		roll left
Throttle	$\delta_T$		zero power	full power

#### 2.1.1. X-Plane output data Simulink update

One of the most peculiar advantage of X-Plane is the possibility to send in output a heavy set of flight data. One can selectively choose between more then a hundred type of data, such as physical quantity related to flight (angular positions, angular velocities, aerodynamic angles, altitude, ...) or state of the system (control surfaces or flap position,

transponder status, autopilot status...). The chosen dataset is delivered outside of X-Plane through UDP connection. In this update, the dataset chosen is the one in table 2.2. Note that, as stated in section 1.4, it is possible also to give in input some state data: in this case, of course, the input are the control surfaces position and throttle (which are also in output, for a sanity check), in order to conduct the aircraft.

Table 2.2: Dataset in output from X-Plane, divided accordingly with class of data.

Velocities	Positions	System	Aerodynamics	Other
Pitch rate	Pitch angle	Elevator position	Attack angle	Time
Roll rate	Roll angle	Aileron position	Sideslip angle	
Yaw rate	Yaw angle	Rudder position		
Indicated Airspeed (IAS)	Altitude	Throttle position		

Simulink is used to receive, elaborate, and send back, all the dataset. The UDP communication is the channel through which communication is performed, at a speed of 50 packets every second. Simulink reads the messages arriving at its UDP port every time step of its simulation, which generally runs at a time rate proper of Simulink itself. Thus, a new function has been developed, reported in fig. 2.2. The main Simulink structure, depicted in fig. 1.4, has been update in its receiver block (fig. 2.2), in order to manage all the data in table 2.2

The role of each block in fig. 2.1 can be summed up as it follows:

- **X-Plane receiver:** it is the UDP port, from which, at each Simulink time step, all messages from X-Plane are read. Those are given in form of a matrix with 8 column, as is the UDP world, and as row as are the required output. This matrix is sent both to output port `xpdata` and to *Decoder* function.
- **Decoder:** a MatLab function that divide each component of the input matrix, and save each information in a vector on MatLab work space.
- **Interpolation subsystem:** this block is responsible for the correct input sending, interpolating time. This block will be deeply faced in section 2.1.2.
- **Control function:** this function take as input, step by step, the value of deflection of each surface and save them in a  $3 \times 1$  vector. Throttle value is just reported to output as it is.
- **Joystick:** a block that take as input the signal from USB connected joystick. Through a switch, signal from prescribed vectors or signal from joystick are commutable.

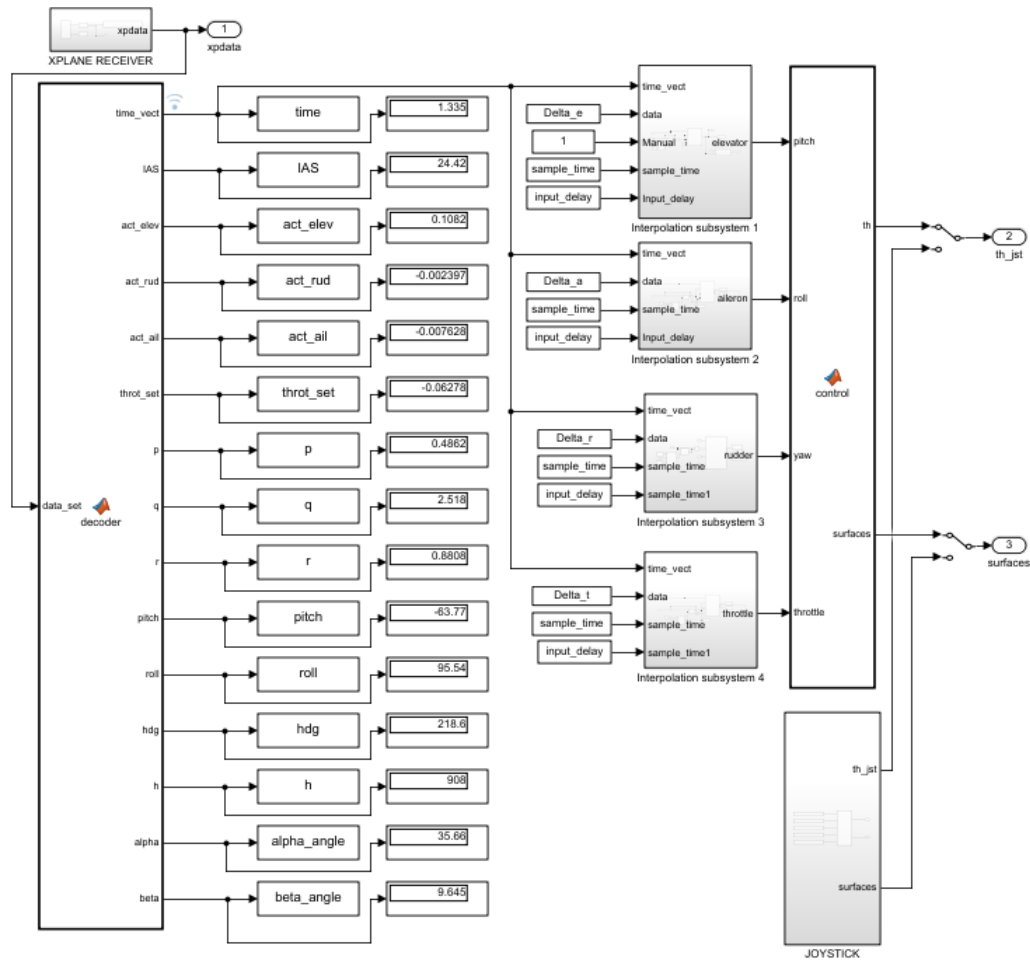


Figure 2.1: Receiver block.

- **Output 1, 2, 3:** output goes directly in input to the control function in fig. 1.4. No further modifications has been performed.

### 2.1.2. Time interpolation

The key aspect of the Simulink update is the possibility to send a custom dataset in input to X-Plane, in order to reproduce an experimental flight test. This action is possible if the input vectors are given to X-Plane with the correct sample time. This is the function of the four interpolation subsystem in fig. 2.1. In fig. 2.2 is depicted the Interpolation subsystem 1. It is distinguished from the others one for the presence of some accessory functions that will be briefly explained in the following.

Working scheme of fig. 2.2 is described in the following:

- **Input 1:** this is the effective simulation time, directly from X-Plane: if X-Plane

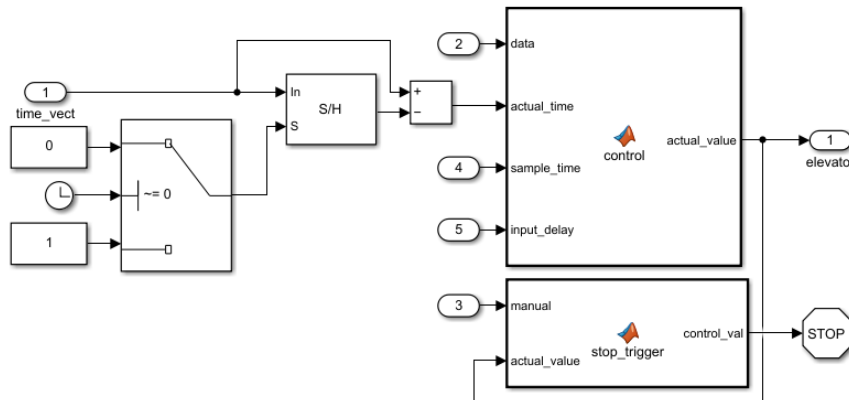


Figure 2.2: Simulink Interpolation block 1.

is frozen, this time is stopped. Effective time is brought to control function at `actual_time` port, so that when Simulink is started, `actual_time` starts from zero. This operation is performed saving the X-Plane time value when Simulink simulation starts, through a sample and hold block; then, the value is subtracted to X-Plane time, with a proper block, at each Simulink step.

- **Input 2:** this is a constant vector where the value of input (surface or throttle), for each time step of 0.1 s, is stored. This vector is a  $2 \times N$  matrix (where N is the number of samples).
- **Input 4** This input is the sample time, that can be modified according with necessities.
- **Input 5** This is a custom delay time: its function is the one to anticipate or delay the interpolation of samples of a fixed time. Its insertion will be clarified in section 2.2. It is crucial to anticipate the interpolation with a proper time, in order to obtain a correct simulation.
- **Control:** the block interpolate the history of deflections or throttle, given as constant vector at port `data`, with X-Plane time, given as single value at port `actual_time`. Value obtained, in output at `actual_value` port is the deflection of surface, with correct time step. In algorithm A.1 it is reported the script of this function.
- **Input 3 and Stop trigger** This block arrest the Simulink simulation at the end of the dataset if input 3 is 1. Else, at the end of dataset, all deflections and throttle are set to 0, but Simulink simulation continues. It is important, for reasons introduced in the following, that input 3 is set to 1, but for some test and debugging purposes,



one could exploit this possibility. It is not necessary to replicate this function in Subsystem 2, 3, or 4.

## 2.2. Validation

In order to perform the functionality test, the X-Plane aircraft Robin DR401 has been taken in account. Wing is a distinctive feature of Robin 401, with a double dihedral. In table 2.3 main specification of test aircraft are reported. It is a single piston engine aircraft, low wing and four seats. It has been chosen for its simple architecture and easy piloting features.

Table 2.3: *Trybrid* main specifications, from [6].

	Unit	Value
Wing span	m	8.72
Wing surface	$m^2$	14.20
Weight	kg	1100
Stall speed	kt	51
Never exceed speed	kt	191
Installed power	hp	180



Figure 2.3: Robin DR401.

The DR401 has been piloted by the use of a joystick on X-Plane, performing a dutch roll excitation. Related input ( $\delta_A, \delta_E, \delta_R, \delta_T$ ), and output (aerodynamics angle, angular position and velocity, TAS) flight data has been saved according with fig. 2.2. Thereafter,

the test has been repeated giving as commands the stored input automatically through Simulink. If the implementation of the system is correct, target flight data (the one obtained piloting with joystick) and repeated flight data (the one obtained piloting through Simulink) should be overlappable. In fig. 2.4 is reported the result of the test: in this case it has been reported just  $\delta_R$  input and its direct effect  $r$ . In both plot there is a delay between the target dataset and its repetition through Simulink. This delay is independent by sample time of the signal, since for different sample time of the input, it is possible to overlap the output with same delay.

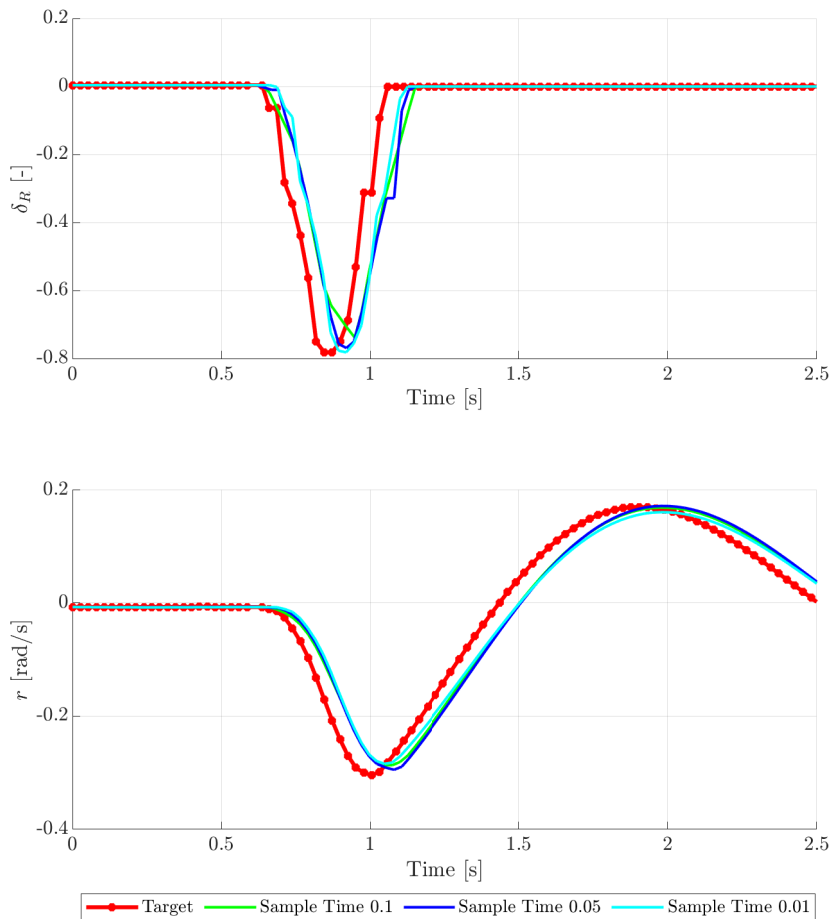


Figure 2.4: Comparison between target flight data and repeated flight data.

To fix the issue, flight data in input through Simulink should be interpolated with an adequate advance. The test just performed has been repeated many times inserting different advances in the Interpolation subsystem, until a satisfactory value of it permit to overlap the target input and the repeated input. Result has been depicted in fig. 2.5. Best match input has been obtained inserting in the Interpolation subsystem in fig. 2.2

an advance of 0.0632 s.

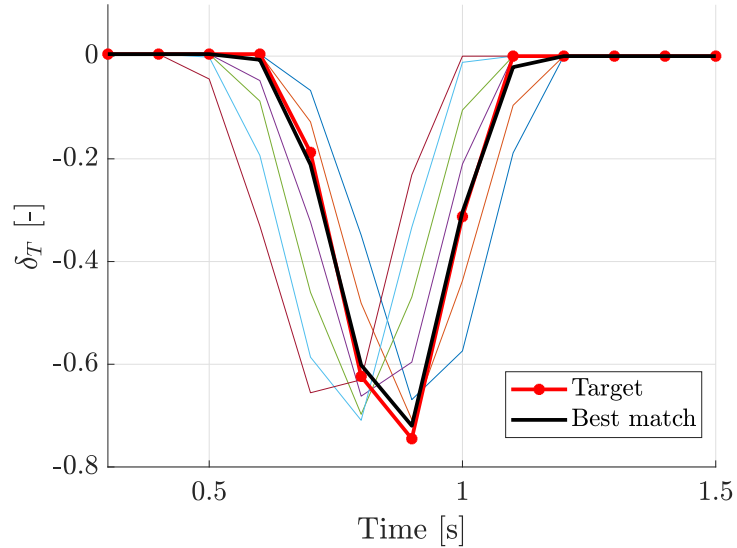


Figure 2.5: Comparison between target flight data and repeated flight data.

Test depicted in fig. 2.4 has been repeated with the update: result are plotted in fig. 2.6. Even in this case different test for different sample time has been conducted, to demonstrate the fact that the replication of an input through Simulink is independent from its sampling time. In this case the two set of data are overlappable with a sufficient precision: in conclusion, the code implementation is now correct. Consider that overlapping, in any case, suffer from a small error.

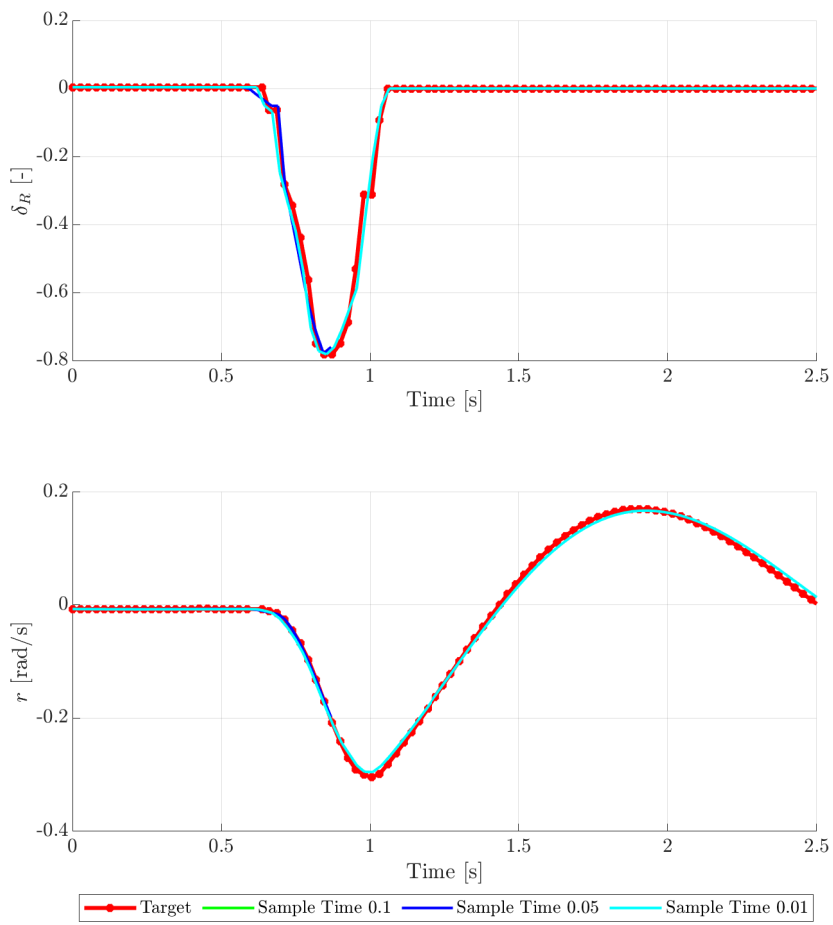


Figure 2.6: Comparison between target flight data and repeated flight data with the insertion of the advance.

# 3 | Update of X-Plane aircraft model

In this chapter it is described the method to find an aircraft model that behaves as the real one: in particular, the scope is to find a model aircraft that responds as the real aircraft to the same inputs. First of all, it is necessary to replicate a flight test manoeuvre on the simulator. Thereafter, experimental flight data and real flight data should be overplotted, in order to compare the quality of fidelization: in a good fidelization, the simulated aircraft behaves as the real one. If a mismatch is present, it is necessary to update the simulated aircraft with an iterative method.

## 3.1. Model update

The update of the model is performed in two steps: in the first it is performed the iterative repetition of flight test for different values of a specific geometrical parameter, in the second a functional is calculated to find which configuration correspond to the best match of data.

### 3.1.1. Iterative repetition of flight test

This method consist in a series of iteration in which the experimental flight test is repeated on X-Plane: at each iteration, a certain geometrical parameter of model aircraft is changed (though a little increment). Then, flight data are recorded: obviously, they depend on the variation of the geometrical parameter. In conclusion, different operations that involve different programs should be performed at each iteration. MatLab has been exploited to coordinate all the steps. In each iteration, driven by a for cycle (integral script is reported in algorithm A.2), following steps are performed:

- Update the aircraft configuration file with new geometrical parameter. This action is performed modifying the aircraft configuration file, written in text format, in X-Plane installation folder. The aircraft configuration file is a large text file, in which

X-Plane stores the values of all geometric parameters of the aircraft and the mesh that is generated consequently. In this step of the procedure, through a MatLab function, the file is modified at the correct line with the value of the geometrical parameter, valid for the current iteration;

- Update the aircraft model in running simulation on X-Plane. This action is performed pressing a pre-assigned keyboard letter on X-Plane (that execute the instruction) through a MatLab command, that is able to digit keyboard letters on another running program;
- Move aircraft to initial position for flight test. This action is performed pressing a pre-assigned keyboard letter on X-Plane (that execute the instruction) through a MatLab command that is able to digit keyboard letters on another running program;
- Execute Simulink simulation, in order to perform the manoeuvre on X-Plane;
- At the end of Simulink simulation, flight data are reported on MatLab workspace. Those data should be stored in suitable structures.

### 3.1.2. Functional evaluation

After having stored all the data for each iteration, the functional can be evaluated a posteriori: as a result, the configuration that generate flight data closer to the target solution is figured out. As in [3], to use the *output error method* it is necessary to define a function to be optimized, that depend on the error between predicted output (simulated aircraft) and real output (real aircraft). The function to be optimized is reported in eq. (3.1).

$$J = \frac{1}{KN} \sum_{i=1}^N \nu(i)^T R^{-1} \nu(i) \quad (3.1)$$

where:

$$\nu(i) = \begin{bmatrix} \omega_{1,\text{sim}}(i; \boldsymbol{\theta}_j) - \omega_{1,\text{meas}}(i) \\ \dots \\ \omega_{K,\text{sim}}(i; \boldsymbol{\theta}_j) - \omega_{K,\text{meas}}(i) \end{bmatrix} \quad (3.2)$$

The weight matrix  $R$  is defined as:

$$R = \begin{pmatrix} R_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & R_K \end{pmatrix} \quad (3.3)$$

where:

$$R_1 = \max(\omega_{1,\text{sim}}(i; \boldsymbol{\theta}_{\text{bsl}}) - \omega_{1,\text{meas}}(i)) \quad (3.4)$$

$$R_K = \max(\omega_{K,\text{sim}}(i; \boldsymbol{\theta}_{\text{bsl}}) - \omega_{K,\text{meas}}(i)) \quad (3.5)$$

and  $N$  is the number of time instants.

In eq. (3.2),  $\omega$  is a generic flight data: it could be an aerodynamics angle such  $\alpha$  or  $\beta$ , or the angular position or velocity referred to an axis. Index *sim* stands for *simulated output*, and *meas* stands for *measured output*. The first one can be evaluated running a flight test simulation, the second is the data set of real flight test.  $\boldsymbol{\theta}$  is the vector of some geometrical parameters, that at each iteration are modified: it is crucial to underline that the flight model is parametrized on its geometry, and the physical quantity depend on these parameter. Once again, the parameter can be as many as one prefer. Matrix  $R$ , as defined in eq. (3.3), is a weight matrix that re-scales the value of the elements of  $\nu$  between 0 and 1. The  $\boldsymbol{\theta}_{\text{meas}}$  is the aircraft baseline configuration, the one in which no geometrical modifications has been performed.

The eq. (3.1) is evaluated for each geometrical parameter increment at every iteration. If a single parameter has been taken in account, it is possible to build a 2D Cartesian plot that report the value of  $J$  depending on the geometrical parameter value. The minimum of  $J$  correspond to the value of geometrical parameter that ensure a behaviour of model aircraft similar to the behaviour of real aircraft to an equal input. If two parameters are taken in account,  $J$  function will be a surface still representable on a 3D plot. The process works even for three or more parameters, but  $J$  function is no more representable.

### 3.2. Test of the method

To test the mathematical implementation of the iterative problem, another dutch roll flight test on X-Plane DR401, taken in account in section 2.2, has been performed: aircraft has been stabilized at a desired trim condition, and a rudder impulse has been given to aircraft. Flight data recorded constitute the target dataset, which is the one used to evaluate the error in the identification process. The flight data chosen, the one that replace  $\omega$  in eq. (3.2) are  $\beta$ ,  $r$ ,  $q$ ,  $\psi$ , and  $\phi$ . Thus, the vertical stabilizer span  $V_t$ , depicted in fig. 3.1, has been increased arbitrarily from 4.32 ft (original value) to 9.49 ft: starting from the same trim condition, the flight test has been repeated with same rudder impulse. Dataset now recorded constitute the baseline, from which the identification process begins.



Figure 3.1: Geometrical parameter modified for flight tests.

If the iterative identification process is correctly set up, the functional  $J$  should return its minimum in correspondence of the target  $V_t$ , equal to 4.32 ft. The validation test has been conducted first time without inserting correction on data delay described in section 2.2, and second time inserting the correction. Result are reported in table 3.1 and in plot format in fig. 3.2. In case of advance insertion the relative error of identified value of  $V_t$  is under 5%, which is acceptable. Consider that the advance inserted in the code is a correction that make the input very similar to the target one, but not exact: for this reason is almost impossible to obtain a null relative error. For the sake of completeness, in fig. A.1 is reported the whole dataset of the flight test.

Table 3.1: Mathematical implementation validation test results

	With advance	Without advance
Target $V_t$ [ft]	4.32	4.32
Identified $V_t$ [ft]	4.54	5.05
Relative error [%]	4.96	14.46



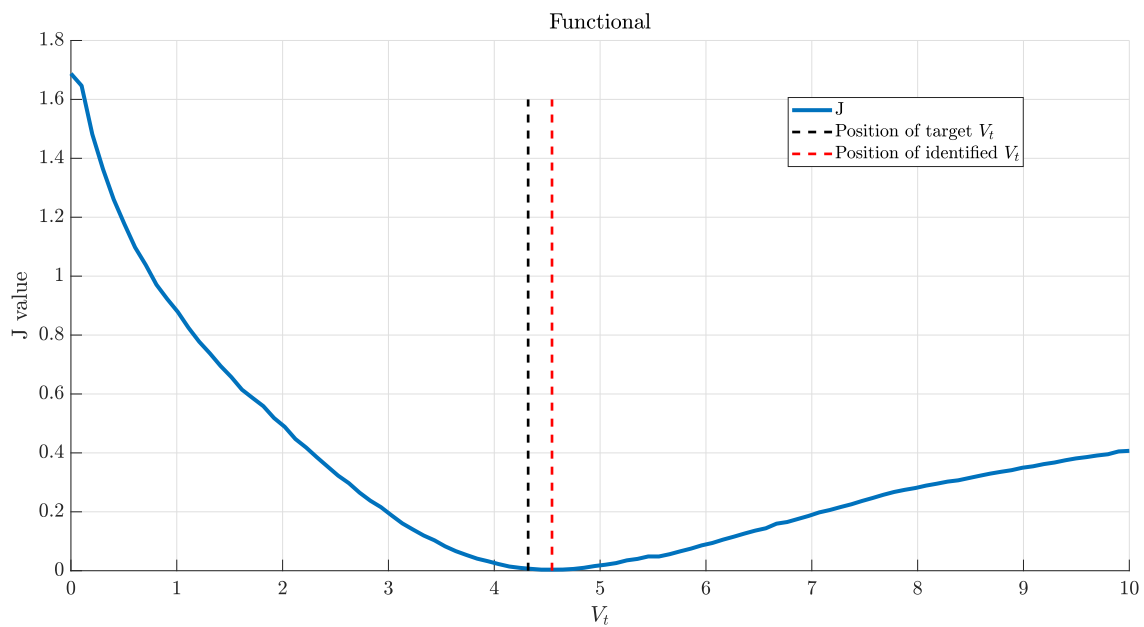
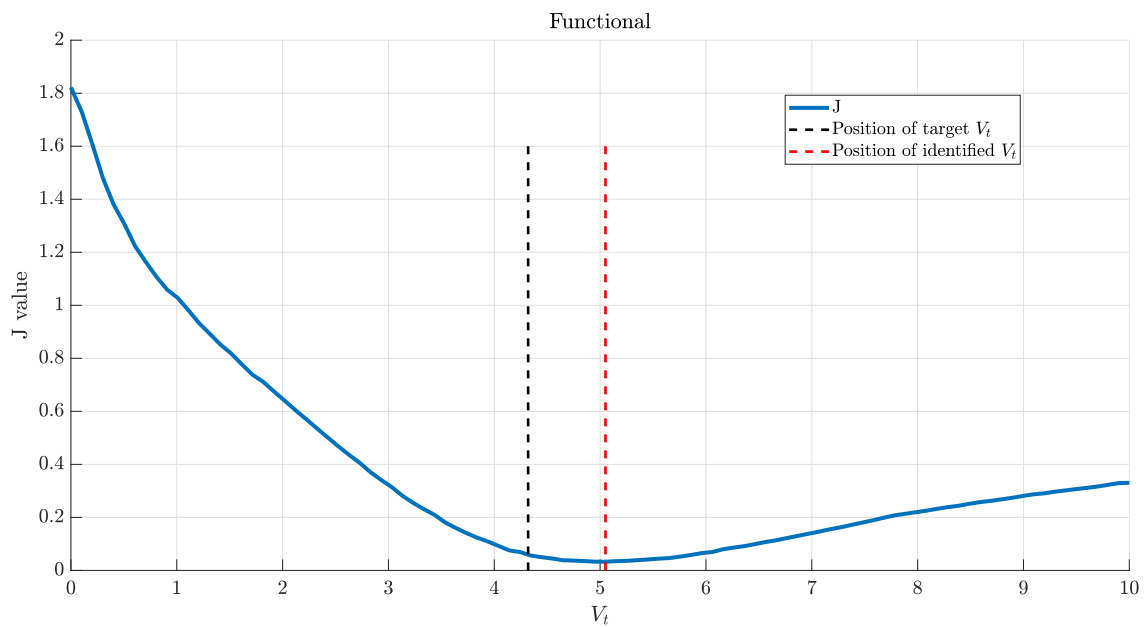
(a) *With advance*(b) *Without advance.*

Figure 3.2: Functional plot



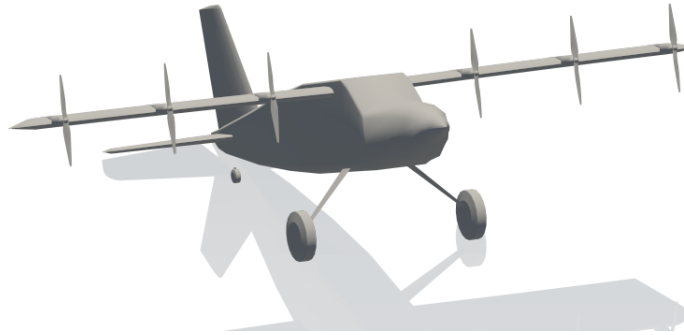
# 4 | *SwitchMaster* model identification updating

In this chapter the fidelization of *SwitchMaster* will be taken in account, through the strategies discussed herein. The baseline aircraft, compared with the real one, are depicted in fig. 4.1. It has been programmed through Plane Maker, an application of X-Plane that allows the user to build a custom aircraft designing all the components. For more details, consult [1].

## 4.1. Flight model fidelity

During the flight test campaign, many test points has been performed: unfortunately, the difficulty in piloting, together with the gust susceptibility (due to low weight aircraft), have greatly compromised the data. Most of them are not usable for computations and analysis. At the current time, *SwitchMaster* identification and fidelization procedure will be performed firstly by the lateral directional point of view, since no longitudinal manoeuvre has been achieved sufficiently. In a future development also this aspect should be taken in account, but several test point should be repeated again. A pseudo-roll excitation and a pseudo-dutch roll excitation has been taken in account. These real flight tests has been repeated on X-Plane: simulator flight data has been overplotted with real flight data, in order to evaluate the initial state of art of fidelization. In the following, some consideration are brought to the attention. To ensure the repeatability of results, a step-by-step flight test procedure on X-Plane and Simulink interface were fixed, as it follow:

- Through X-Plane map, with paused simulation, aircraft is positioned at trim altitude, heading and velocity, of real flight test. It is also possible to set the initial pitch value but neither roll angle nor angular velocities. This is an important restriction that will be soon discussed: initial roll angle and initial angular velocities, in all the test, start from zero instead from initial value of real data set.

(a) *Aircraft baseline model*(b) *Aircraft real model*Figure 4.1: *SwitchMaster* real and simulated aircraft.

- Simulink simulation starts. This action do not activate the simulation on X-Plane, since the time of Simulink is the time of X-Plane. If the latter is put in pause, also the first is stopped.
- X-Plane simulation is unpaused. Flight test begin.
- At the end of flight test, simulation is put again in pause, Simulink simulation is stopped, and data can be analyzed in MatLab work-space.

Thought this procedure, following results were obtained.

**Pseudo roll test** In fig. A.3 has been conducted a maneuver that is aimed to excite roll dynamics. The input is a rudder doublet, quite precise from the point of view of frequency and amplitude. Despite this, an important contribution of elevator, which excite longitudinal dynamic is present. It is possible to observe a good correspondence

between  $\phi$  and  $p$  and no correspondence in  $r$  and  $\beta$ : in particular, amplitude and frequency is different. In this case, the issue of initial roll angle and rates do not affect the test, since the real test is begun with null roll angle.  $\psi$  initial baseline value is different from target initial value, but notice that timeline starts from 1s, to highlight the relevant part of the test: at 0 time instant, out of scale, the aircraft has been correctly oriented. Even if it is not test subject, the elevator excitation permit to observe the behaviour of longitudinal data: correspondence is no perfect but acceptable.

**Pseudo dutch roll test** In fig. A.2 there is an attempt to excite dutch roll through a rudder doublet. The input given is not precise: the doublet has an overshoot. In this case,  $\phi$  and  $p$  do not start from correct value, and for this reason are not taken in account.  $\psi$  period in baseline case is not in phase with target one, indeed it anticipates the real value of about half a second; its amplitude is not adequate too. Same considerations are suitable for and  $\beta$ .

**Conclusions** What emerge from flight test performed herein is principally a mismatch in the directional behaviour of *SwitchMaster*. The latter is determined by vertical tail geometry: an identification process of its dimensions and position will be set up in the following, in order to find the geometry that ensure a good matching between identified data and real, or target, flight data. The manoeuvre taken in account for model update will be the pseudo dutch roll, in fig. A.2. For what concern lateral behaviour, flight data shows an acceptable behaviour: it should be necessary other flight test to seal this result.

## 4.2. Updating process of directional dynamics

### 4.2.1. Mathematical model modification

To take in account the fact that in X-Plane is not possible to set  $\phi$  to a desired value, hence is not possible to give the aircraft the correct initial  $\phi$ , its contribution, together with  $r$ , is removed from dataset. The error vector and weight matrix used for identification process are the one in (eq. (4.1)) and (section 4.2.1). The flight data selected to evaluate the error are  $\beta$ ,  $r$ , and  $\psi$ .

$$\nu(i) = \begin{bmatrix} r_{\text{sim}}(i; \boldsymbol{\theta}_j) - r_{\text{meas}}(i) \\ \beta_{\text{sim}}(i; \boldsymbol{\theta}_j) - \beta_{\text{meas}}(i) \\ \psi_{\text{sim}}(i; \boldsymbol{\theta}_j) - \psi_{\text{meas}}(i) \end{bmatrix} \quad (4.1)$$

$$R = \begin{pmatrix} R_1 & 0 & 0 \\ 0 & R_2 & 0 \\ 0 & 0 & R_3 \end{pmatrix} \quad (4.2)$$

$$R_1 = \max(r_{\text{sim}}(i; \boldsymbol{\theta}_{\text{bsl}}) - r_{\text{meas}}(i)) \quad (4.3)$$

$$R_2 = \max(\beta_{\text{meas}}(i; \boldsymbol{\theta}_{\text{bsl}}) - \beta_{\text{meas}}(i)) \quad (4.4)$$

$$R_3 = \max(\psi_{\text{meas}}(i; \boldsymbol{\theta}_{\text{bsl}}) - \psi_{\text{meas}}(i)) \quad (4.5)$$

Even eq. (3.1) has been slightly modified, in order to take in account the fact that the minimum of the functional should lead to an aircraft configuration which is not heavily distant from the baseline configuration. The reason is that baseline aircraft has its geometrical dimensions (wing surface, control surfaces, horizontal and vertical empennages) which are the same of real aircraft. Obtaining a geometry which is greatly different to the real one could lead to some unexpected results. Hence, in eq. (4.6), a contribution from the parameter values has been added: this contribution is greater as the parameters are different from their baseline value. This approach is called *Bayesian*, and it is outlined in [5].

$$J(\boldsymbol{\theta})_{BS} = \frac{1}{KN} \sum_{i=1}^N \nu(i)^T R^{-1} \nu(i) + \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{bsl}})^T \Sigma_P^{-1} (\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{bsl}}) \quad (4.6)$$

where:

- $\boldsymbol{\theta}$  is the vector of parameter value at the actual iteration;
- $\boldsymbol{\theta}_{\text{bsl}}$  is the vector of parameter value in baseline configuration;
- $\Sigma_P$  is a diagonal weight matrix. In first approximation it is an identity matrix, whose dimension is equal to the number of parameters chosen for a specific analysis. In this manner, equal weight is given to the whole parameter set.

In any case, it is possible to handle the data even with eq. (3.1), updating the definition of  $R$  and  $\nu$ : as it is clear, the main difference is that no variation of the parameter is taken in account: in this sense, the solution could lead to an aircraft geometry that is not compatible with standard design criteria.

#### 4.2.2. Parameter selection

At this point of the work it is necessary to choose which aircraft geometrical parameters should be identified. There are no fixed rules, but a incorrect parameter choice could lead to a not identifiable problem. What influence primarily the lateral motion is the dimension and position of vertical tail, and the the dimension of rudder. In fig. 4.2 main parameters that affect the lateral behaviour are indicated on side view of *SwitchMaster*. In table 4.1 are reported the value of each parameter in the baseline configuration. Note that nomenclature  $Z_{ARM}$  has been given in consistence with X-Plane reference system, that place the Z-axis longitudinally: in body reference system it should be  $X_{ARM}$ . The more parameters are chosen to perform the identification, the more will be the computational time:  $C_{ROOT}$ ,  $C_{TIP}$ , and  $H$  can be used to evaluate the vertical tail surface. In this sense, in order to reduce the number of parameter, it has been chosen to identify an area scale factor  $p$ , defined in eq. (4.7) and eq. (4.8) and eq. (4.9), that bound the three parameters in a single one.

$$C_{ROOT}(i) = p(i)C_{ROOT\text{bsl}} \quad (4.7)$$

$$C_{TIP}(i) = p(i)C_{TIP\text{bsl}} \quad (4.8)$$

$$H(i) = p(i)H_{\text{bsl}} \quad (4.9)$$

Where  $i$  is the iteration index.

Table 4.1: *SwitchMaster* baseline geometrical parameter value.

	$Z_{\text{ARMbsl}}$	$C_{\text{ROOTbsl}}$	$C_{\text{TIPbsl}}$	$H_{\text{bsl}}$
<b>Value [ft]</b>	3.97	1.15	1.15	0.72

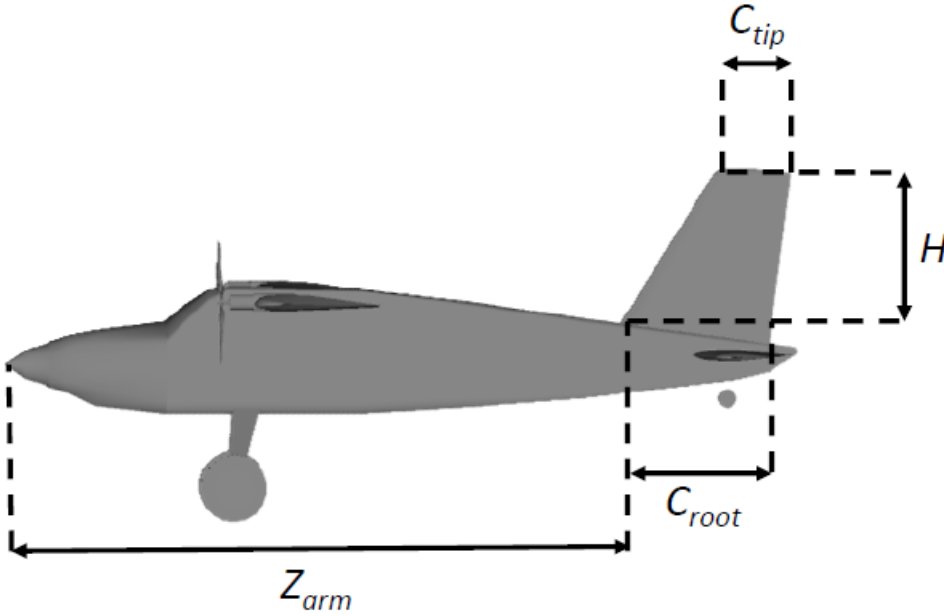


Figure 4.2: Geometrical parameters of *SwitchMaster*.

Regarding the rudder, its span is equal to  $H$ , and chordwise, as in eq. (4.10), the dimension is given as a ratio with vertical tail mean chord (eq. (4.11)). The baseline value is 0.5.

$$c = \frac{C_{\text{RUD}}}{V_{\text{MC}}} \quad (4.10)$$

where

$$V_{\text{MC}} = \frac{C_{\text{ROOT}} + C_{\text{TIP}}}{2} \quad (4.11)$$

Another crucial parameter is  $C_{\text{VT}}$ : it is defined as in eq. (4.12), and it combines the longitudinal vertical tail position with its dimension, the wing surface and span. Aircraft having the same volume coefficients tend to have similar static stability characteristics.



Generally this coefficient spaces from 0.02 for a sailplane to 0.09 for a Jet transport. The *SwitchMaster* has a  $C_{VT}$  in baseline configuration equal to 0.05.

$$C_{VT} = \frac{(Z_{ARM} - Z_{CG})S_{VT}}{b_W S_W} \quad (4.12)$$

### 4.3. Analysis and results

In the following, main results are reported varying different parameter sets. These analysis has been performed with the main function reported in algorithm A.2. The latter is responsible for coordinating all the programs (MatLab, Simulink, X-Plane) that, during a single iteration, dialogue one each other. In each analysis, one or more parameter vary within an arbitrary domain through a *for* cycle. Through the MatLab function *linspace* it is possible to set the number of values that a certain parameter must assume between the lower bound and upper bound of the domain. Higher the steps assigned to a parameter, higher will be the computational time: consider that a single iteration last 16s.

#### 4.3.1. Case 1: vertical tail longitudinal position and scale factor

First analysis has been conducted testing the aircraft behaviour for different values of  $Z_{ARM}$  and  $p$ . Domain data and number of steps has been reported in table 4.2. The computational time for this analysis has been more than 7 hours.

Table 4.2:  $Z_{ARM}$  and  $p$  and  $c$  range and number of steps for current analysis.

Parameter	Lower bound	Upper bound	Steps
$Z_{ARM}$	2.9 ft	3.9 ft	40
$p$	0.5	2	40

Results are depicted in fig. 4.3, and numerical results are reported in table 4.3. The functional has been evaluated according to both eq. (3.1) and eq. (3.1) namely in non-Bayesian sense and Bayesian sense. It is possible to observe that in non-Bayesian case, the absolute minimum lead to an unacceptable geometry, but the contour plot, in the lower part, identify a domain in which the value of the functional is augmented of no more of 10% of minimum itself: it is possible to find a locus of acceptable points that could lead to an aircraft configuration which is not too far from baseline one. In this case it is helpful the Bayesian functional: adding a contribution from the parameters, which increase as the geometry is different from baseline configuration, it is found an absolute minimum that correspond an acceptable configuration.

Table 4.3: Results for current analysis.

Parameter	Non-Bayesian funct.	Bayesian funct.
$Z_{ARM}$	2.95 ft	3.69 ft
$p$	0.808	0.692
$J$	0.3654	0.4792

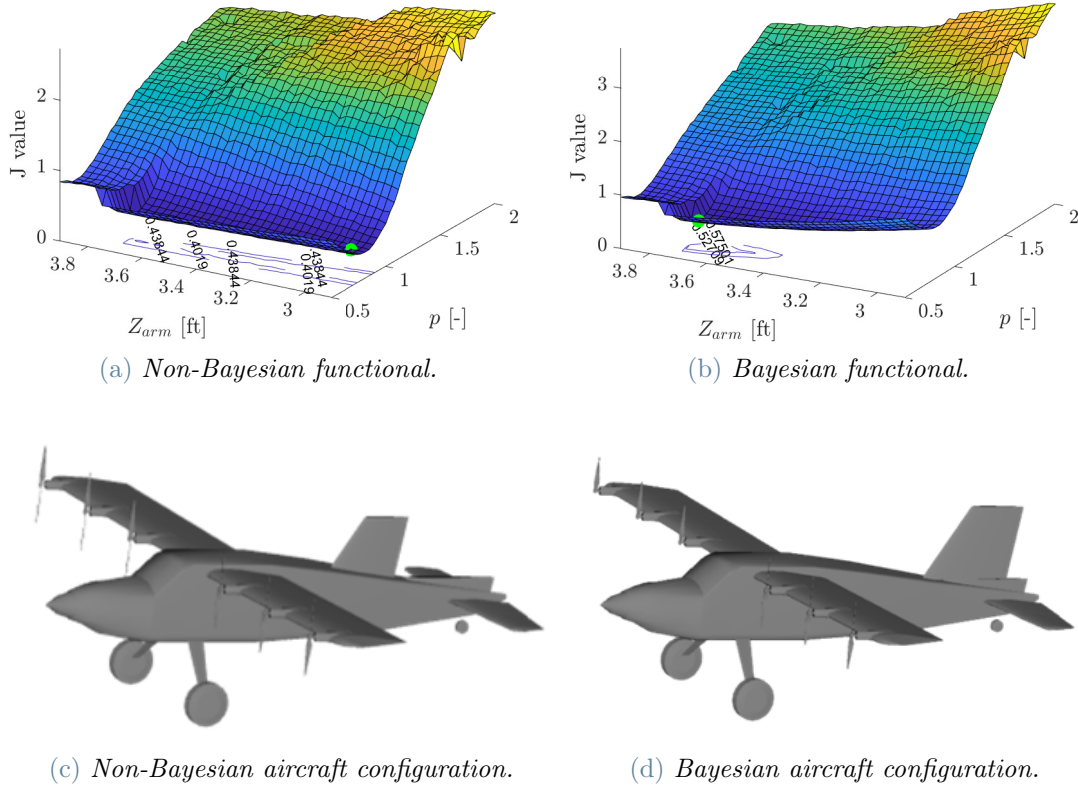


Figure 4.3: Result analysis and aircraft configuration for different values of  $Z_{ARM}$  and  $p$ . In both the functionals, Minimum is highlighted by the green point.

Performance of the aircraft, in terms of correspondence with target of angular position, angular velocity, and sideslip angle, are reported in fig. 4.4. Flight histories reported are the one associated with the aircraft configuration that has returned the lower value of the functional, in Bayesian and non-Bayesian sense. First consideration is that the Bayesian and non-Bayesian aircraft behaviour are similar: for most of the time the two flight histories, for all the data, are superimposed. Secondly, there is an improvement by the point of view of the frequency with respect to the baseline, but the amplitude is not replicated as well. Most important issue is the fact that data are not damped as the target:

as the command is interrupted (namely free response), the target reduce drastically the oscillation: with respect to baseline configuration, this behaviour is slightly corrected, but it is far from a satisfactory result.

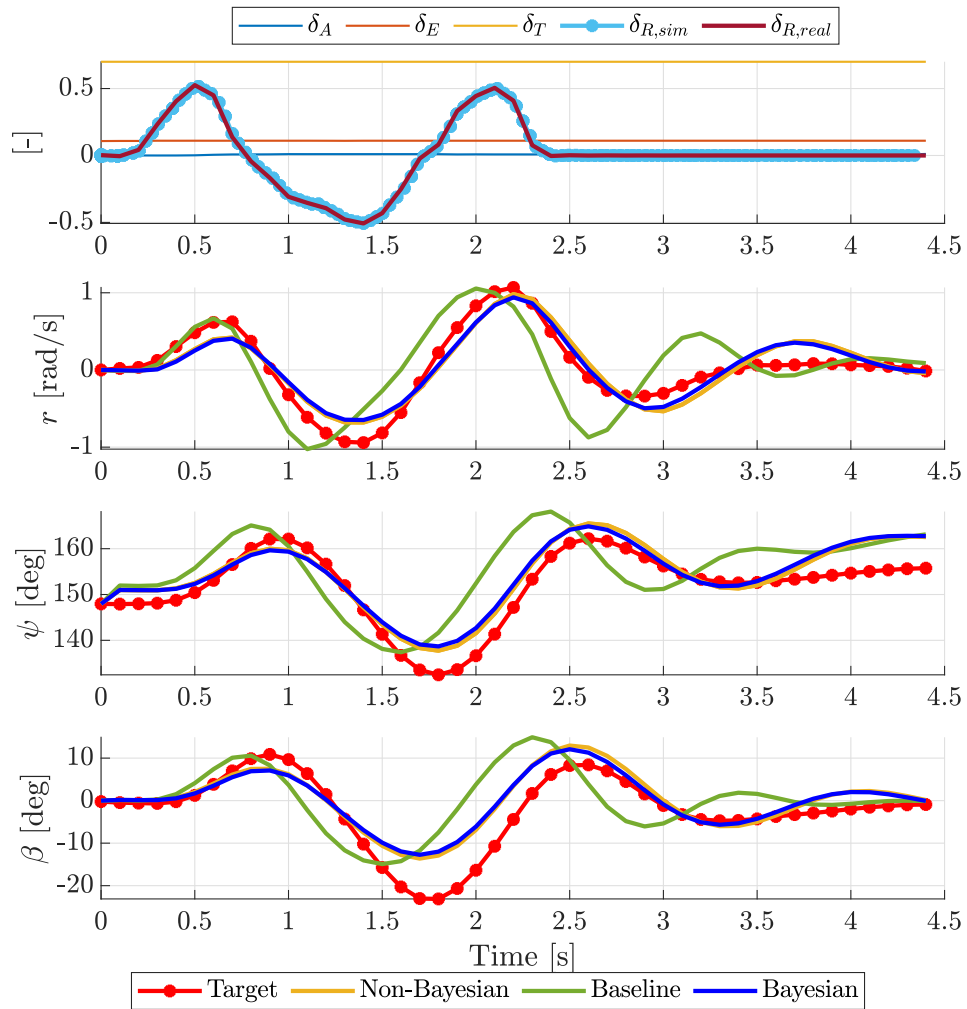


Figure 4.4: Flight histories for analysis on  $Z_{ARM}$  and  $p$ .

### 4.3.2. Case 2: vertical tail volume, scale factor, and rudder percentage

Second analysis conducted has been on  $C_{VT}$ ,  $p$  and  $c$ . Adding a parameter correspond to adding a degree of freedom to the problem, with the aim to damp data in free response. Domain data and number of steps has been reported in table 4.4. The computational time for this analysis has been more than 8 hours.

Table 4.4:  $C_{VT}$ ,  $p$  and  $c$  range and number of steps for current analysis.

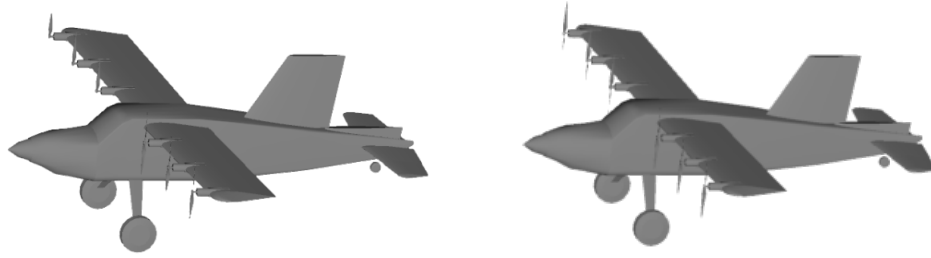
Parameter	Lower bound	Upper bound	Steps
$C_{VT}$	0.005	0.07	12
$p$	0.8	1.2	12
$c$	0.1	0.5	12

Since three parameter are involved, in this case in is not possible to plot the functional. Numerical results of the analysis are reported in table 4.5. The functional has been evaluated again according to both eq. (3.1) and eq. (3.1). In this case it is possible to notice that the geometrical parameters value are similar in both cases, in fact differences in aircraft configurations in fig. 4.5 are negligible: in particular, the  $C_{VT}$  obtained is the same for Bayesian result and for not-Bayesian one. As in [7], this value is distinctive of a general aviation aircraft such as the Ryan Navion: it is assumed to be a correct value. The value of  $c$  obtained is at the boundary of the given variation range: unfortunately, X-Plane admits value of  $c$  exclusively between 0.1 and 0.5. As one can expect, the value of the longitudinal position of vertical tail  $Z_{ARM}$  is higher, as it is closer to baseline value.

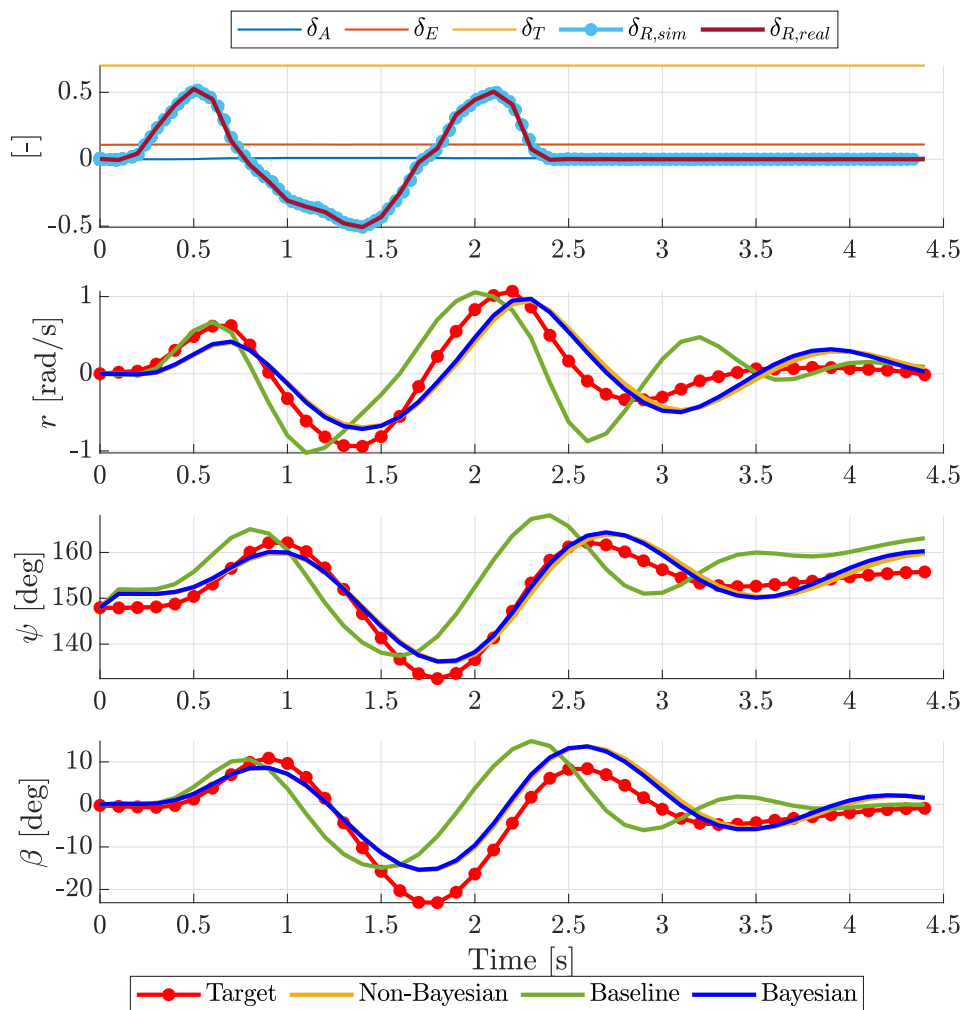
Table 4.5: Results for current analysis.

Parameter	Non-Bayesian funct.	Bayesian funct.
$C_{VT}$	0.0405	0.0405
$p$	1.2	1.13
$c$	0.5	0.5
$J$	0.2776	0.2877
$Z_{ARM}$	2.82 ft	2.98 ft

Considerations done on the quality of results in fig. 4.4 are feasible for fig. 4.6 too: it is possible to observe an improvement in terms of frequency in angular velocity and position. A major improvement is obtained in sideslip angle, with respect to previous result. In any case, the result is far from perfection: the frequency is not the same of target, and once again the damping is not achieved. An additional consideration is on aircraft configuration

(a) *Non-Bayesian aircraft configuration.*(b) *Bayesian aircraft configuration.*Figure 4.5: Aircraft configuration for the analysis conducted on  $C_{VT}$ ,  $p$  and  $c$ .

on fig. 4.5. In both cases the vertical tail is placed in an unconventional position that could implies stability issues.

Figure 4.6: Flight histories for analysis on  $C_{VT}$ ,  $p$  and  $c$ .

### 4.3.3. Case 3: best vertical tail volume, scale factor, and rudder percentage

Since the analysis on  $C_{VT}$ ,  $p$  and  $c$  led to an unconventional aircraft configuration, with also some issue on frequency and damping, in this analysis the value of  $C_{VT}$  of 0.0405, previously obtained, has been fixed: then, a variation on  $p$  and  $c$  has been performed: the aim is the one of finding a conventional configuration, drawing back the vertical tail, but conserving the  $C_{VT}$ . This operation has been performed changing the  $p$  value, which implies a variation of  $Z_{ARM}$ . The relationship between  $p$  and  $Z_{ARM}$  is obtained solving eq. (4.12) for  $Z_{ARM}$  itself, as in eq. (4.13). Range of the parameters is reported in table 4.6: since the aim is finding a suitable aircraft configuration, the value of  $p$  boundaries has been fixed in order to explore the  $Z_{ARM}$  range between 3.0 ft and 5.0 ft, according to eq. (4.13).

$$Z_{ARM} = \frac{C_{VT}S_W b_W}{S_{VT}} + Z_{CG} \quad (4.13)$$

Table 4.6:  $p$  and  $c$  range and number of steps for current analysis.

Parameter	Lower bound	Upper bound	Steps
$c$	0.1	0.5	25
$p$	0.6	0.8	25

In this case, the functional evaluated is just the Bayesian one, in order to find an aircraft geometries not too dissimilar from the baseline one. The numerical results obtained are reported in table 4.7, and relative aircraft configuration and functional plot in fig. 4.7. The vertical tail is set in a position that is assumed to be acceptable. The functional has been plotted in term of  $Z_{ARM}$ , applying on  $p$  the eq. (4.13). The functional shows an irregular step for values of  $Z_{ARM}$  between about 3.7 ft and 4.4 ft: the reason could be that, if the vertical tail works in the trail of horizontal tail, it is less effective. Further investigation should be necessary to confirm this consideration. This irregularity is also present in fig. 4.3, and the explanation could be the same.

Taking in account fig. 4.8, as for fig. 4.4 and for fig. 4.6, persist the issues on frequency and damping: same considerations on Case 1 and Case 2 are suitable for the present one.

Table 4.7: Results for current analysis.

Parameter	Bayesian funct.
$c$	0.5
$p$	0.76
$J$	0.5361
$Z_{ARM}$	3.68 ft

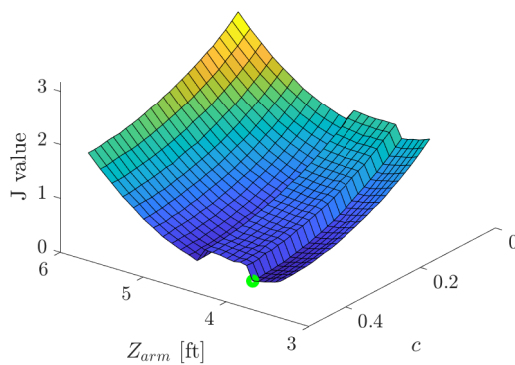
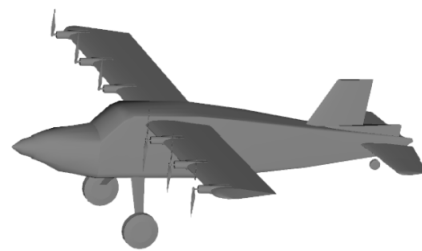
(a) *Bayesian functional.*(b) *Bayesian aircraft configuration.*

Figure 4.7: Result analysis and aircraft configuration for different values of  $p$  and  $c$  with a constant value of  $C_{VT}$ . In both the functionals, Minimum is highlighted by the green point.

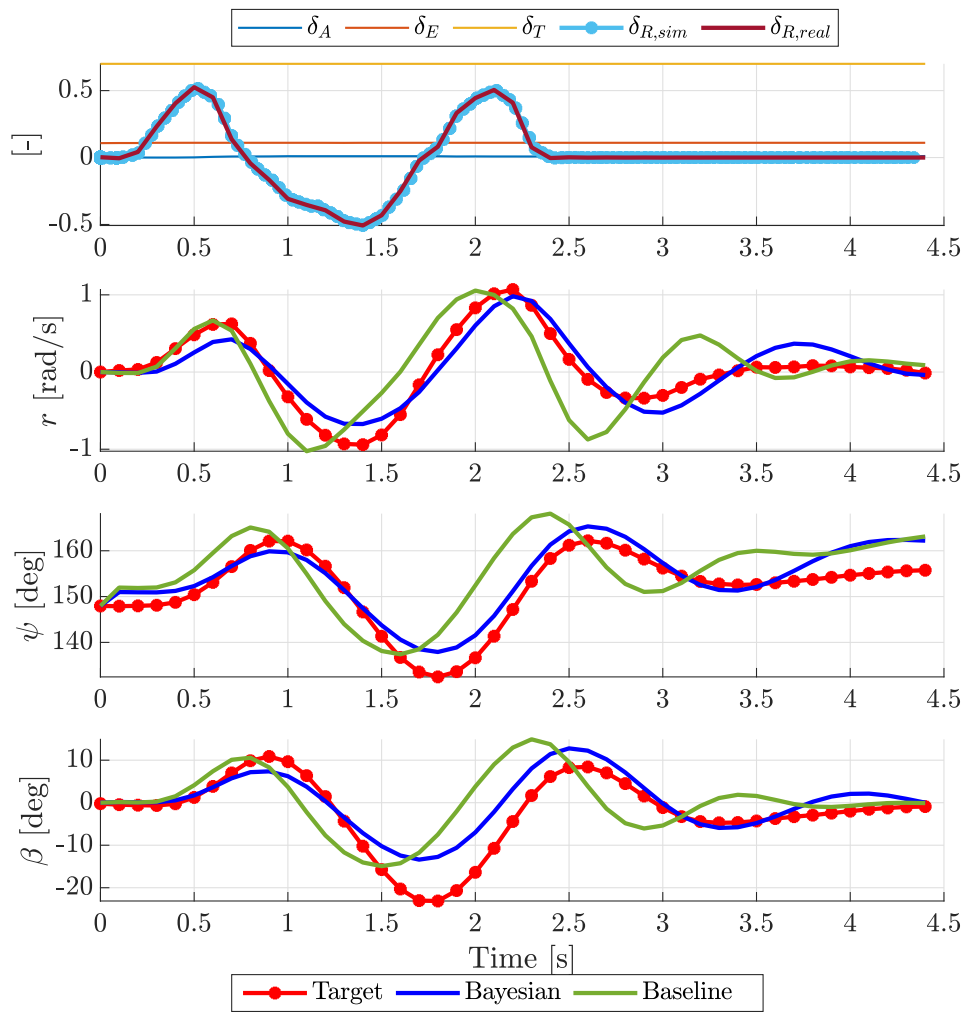


Figure 4.8: Flight histories for analysis on  $p$  and  $c$  with a constant value of  $C_{VT}$ .



#### 4.3.4. Comparative analysis

In fig. 4.9 the Bayesian results of Case 1, Case 2, and Case 3 are overplotted, in order to well appreciate the reciprocal differences. In table 4.8, dimensions of parameters involved in the identification process are listed for each case. One can observe that the outcome of  $Z_{ARM}$  in Case 1 and Case 3 is very similar, as the flight histories could suggest in fig. 4.9, since are similar too. Consider now the functional of each case. In general, its meaning consist in an overall quality of the solution. Hence, the lower value is the one of Case 2 ( $J = 0.2877$ ), which also correspond to a configuration that is the most dissimilar from the baseline. Case 1 has  $J = 0.4792$  which is greater of the 39.96% with respect to the best  $J$  value obtained; Case 3 has  $J = 0.5361$  greater of the 46.33%. All the solution improve, but not completely, the fidelity of *SwitchMaster* X-Plane flight model, compared with real flight model. Indeed, an important issue of damping and amplitude is present in all flight stories. A simple comparison of functionals is not sufficient to define which configuration gives the major improvement, because many aspects are not taken in account. This is the reason why in following section is presented an analysis on the error of each measured quantity in time and frequency domain.

Table 4.8: Dimensions recap.

	$Z_{ARM}$ ft	$C_{ROOT}$ ft	$C_{TIP}$ ft	$H$ ft	$c$ [-]
Case 1	3.69	0.796	0.796	0.547	0.5
Case 2	2.98	1.299	1.299	0.893	0.5
Case 3	3.68	0.874	0.874	0.600	0.5

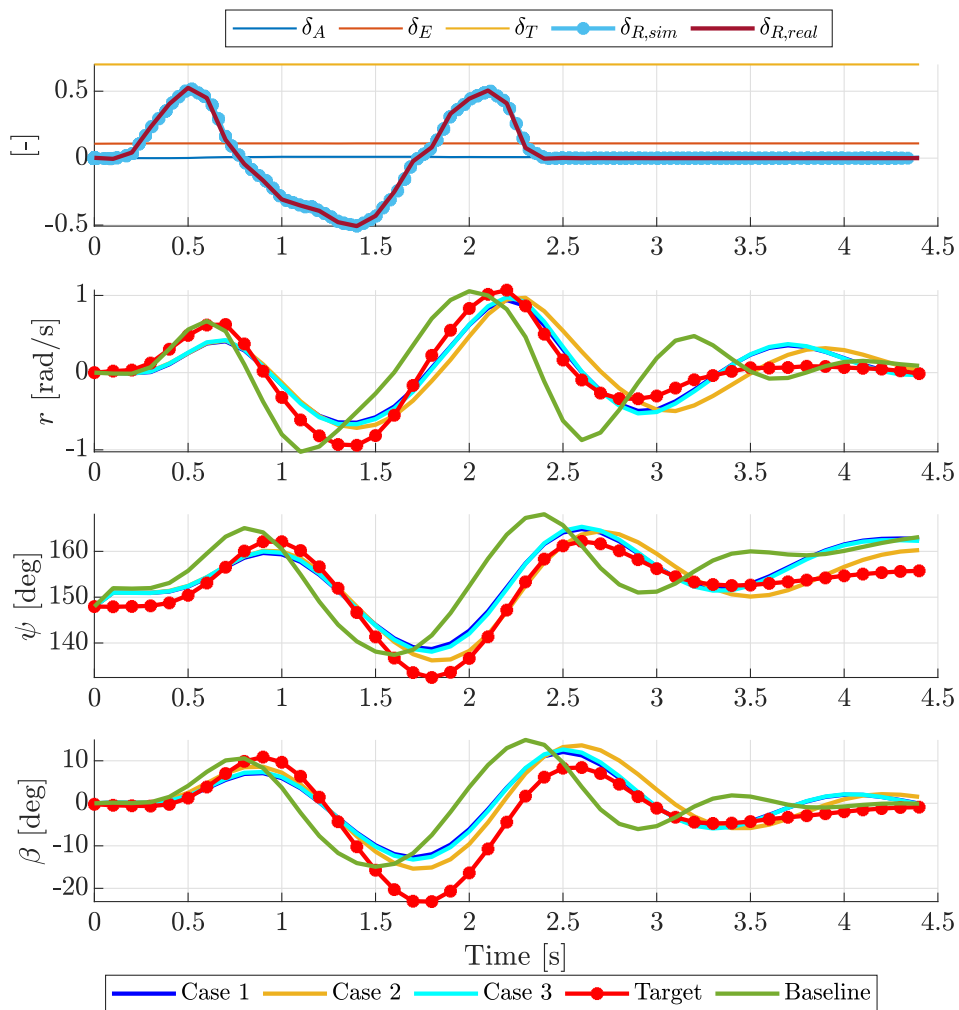


Figure 4.9: Comparative plot for Case 1, Case 2, and Case 3.

## Error analysis

Some further considerations regarding the quality of the solutions obtained can be done considering the error of the baseline flight history and best Bayesian results with respect to target. Defining the errors as in eq. (4.14) (where  $\omega$  stands for the physical quantity taken in account), the plot obtained is the one in fig. 4.10. In table 4.9 are listed the mean error and the maximum error for each case: a significant improvement, quantified in the quality factor  $\eta$  is achieved. The latter is defined as in eq. (4.15).

$$\begin{aligned}\Delta\omega(i) &= \omega(i) - \omega_{\text{target}}(i) \\ \Delta\omega_{\text{MEAN}} &= \sum_{i=1}^N \frac{\Delta\omega(i)}{N}\end{aligned}\tag{4.14}$$

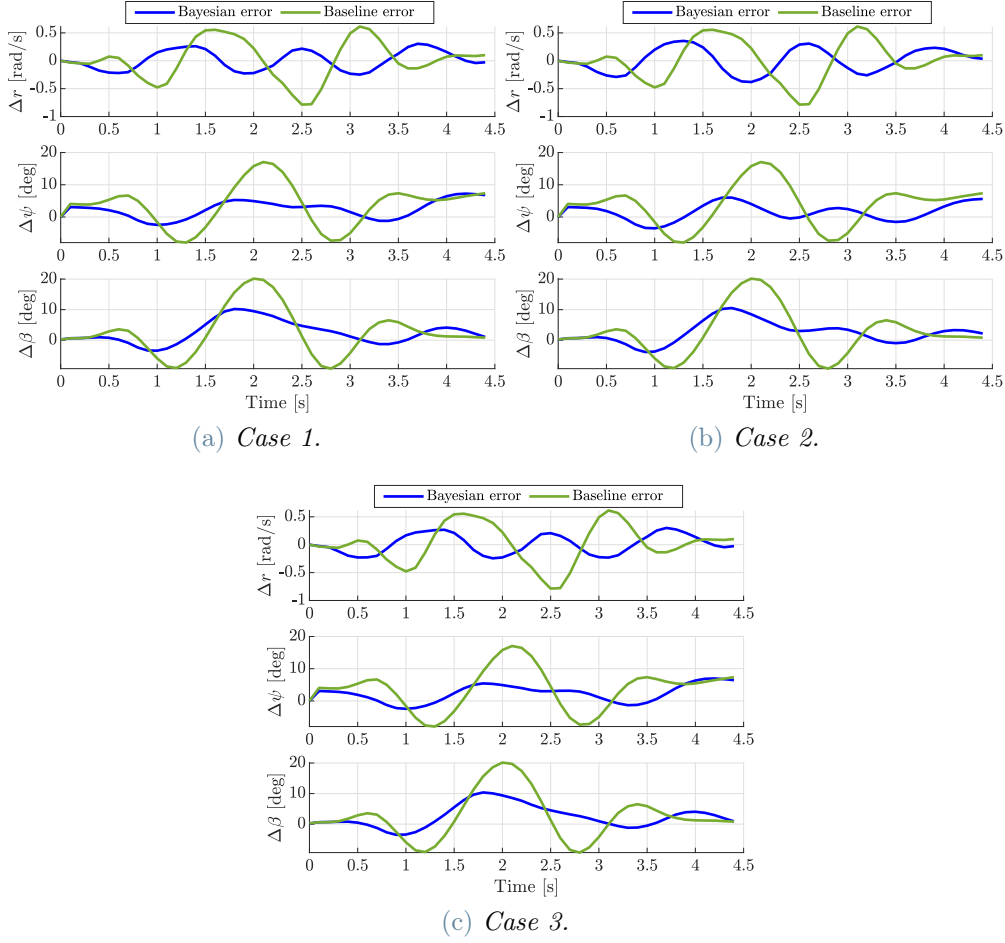
$$\Delta\omega_{\text{MAX}} = \max(\Delta\omega(i))$$

$$\begin{aligned}\eta_{\omega, \text{MEAN}} &= \left(1 - \frac{\Delta\omega_{\text{MEAN}}}{\Delta\omega_{\text{MEAN, target}}}\right) 100 \\ \eta_{\omega, \text{MAX}} &= \left(1 - \frac{\Delta\omega_{\text{MAX}}}{\Delta\omega_{\text{MAX, target}}}\right) 100\end{aligned}\tag{4.15}$$

Table 4.9: Error in time.

	Unit	Baseline	Case 1	Case 2	Case 3
$\Delta r_{\text{MEAN}}$	rad/s	0.0118	0.0113	0.0074	0.0103
$\Delta r_{\text{MAX}}$	rad/s	0.6144	0.3045	0.3562	0.3007
$\eta_{r, \text{MEAN}}$	%		4.17	37.18	13.38
$\eta_{r, \text{MAX}}$	%		50.43	42.02	51.06
$\Delta\psi_{\text{MEAN}}$	deg	3.67	2.38	1.51	2.32
$\Delta\psi_{\text{MAX}}$	deg	17.05	7.2	6.0	6.97
$\eta_{\psi, \text{MEAN}}$	%		35.11	58.86	36.94
$\eta_{\psi, \text{MAX}}$	%		57.48	64.75	59.14
$\Delta\beta_{\text{MEAN}}$	deg	2.48	2.48	2.42	2.44
$\Delta\beta_{\text{MAX}}$	deg	20.16	10.23	10.52	10.37
$\eta_{\beta, \text{MEAN}}$	%		0.91	3.13	2.35
$\eta_{\beta, \text{MAX}}$	%		49.39	47.79	48.56

In fig. 4.11 is depicted the frequency content of the error in semi-logarithmic scale. The plot has been obtained performing the Fourier transformation of error itself. In this case the attention should be focus on the peak amplitude. For what concerns  $r$ , the frequency of the signal is slightly improved in Case 1 and Case 3, while in Case 2 the improvement

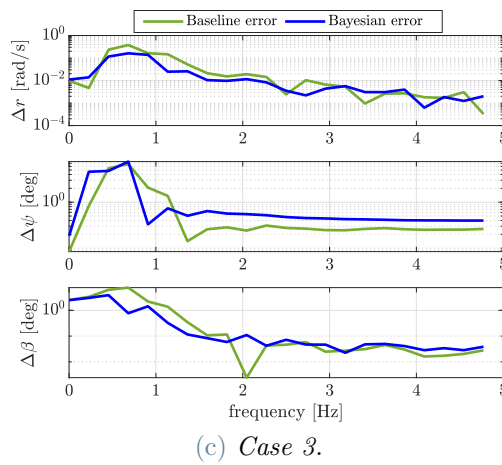
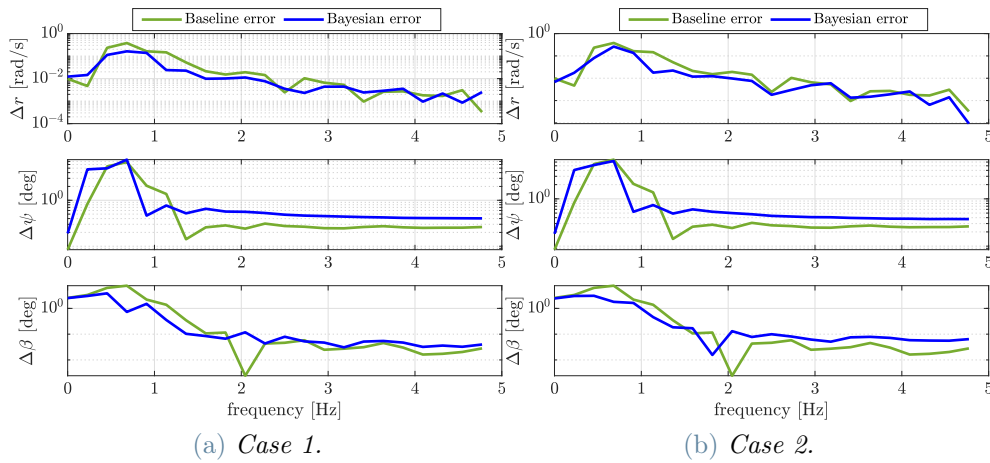
Figure 4.10: Error on  $r$ ,  $\psi$  and  $\beta$  in time domain.

is poor. In every cases the peak of  $\psi$  is overlapped both in baseline case and in Bayesian case: the frequency is not greatly improved. Regarding  $\beta$ , the frequency of the signals is enhanced in all the cases. Consider also that for higher frequencies, which could be noise or disturbance, in all the cases an all the signals the frequency error is equal or greater for the Bayesian solution. This is not a issues because, as explained, the main interest is on the peak frequency. In table 4.10 are the amplitude of the peak for each quantity and in each case. The quality factor  $\eta$ , in this case, is define as in eq. (4.16).

$$\eta_{\omega, freq} = \left(1 - \frac{|\Delta\omega|_{\text{PEAK}}}{|\Delta\omega|_{\text{PEAK, target}}}\right) 100 \quad (4.16)$$

Table 4.10: Error in frequency.

	Unit	Baseline	Case 1	Case 2	Case 3
$ \Delta r _{\text{PEAK}}$		0.380	0.163	0.262	0.162
$\eta_{r,\text{freq}}$	%		57.19	31.10	57.49
$ \Delta \psi _{\text{PEAK}}$		6.750	7.659	6.312	7.639
$\eta_{\psi,\text{freq}}$	%		-13.47	6.486	-13.17
$ \Delta \beta _{\text{PEAK}}$		7.604	3.893	3.08	3.881
$\eta_{\beta,\text{freq}}$	%		49.51	59.45	48.94

Figure 4.11: Error on  $r$ ,  $\psi$  and  $\beta$  in frequency domain.

## Free response analysis

The free response is the part of flight history where no input is given to surfaces: having as reference fig. 4.9, it is the one between 2.5 s and 4 s. In order to evaluate the fidelity of the flight model, it's fundamental to take in account this part of response, because it depend only on the nature of the system and its initial conditions. In fig. 4.13 is depicted the frequency content of free response: considering the peak, the plots suggest that, in every cases and for every data, the Bayesian solution has perfectly achieved the target frequency. This is a wrong conclusion, since data in time domain are sampled every 0.1 s, while, according to Fourier transformation, data are sampled every 0.5 Hz. Latter sample frequency is not fine enough to catch the true frequency of the peak: however, this analysis suggested that the identified solutions oscillate with a frequency closer to the target one.

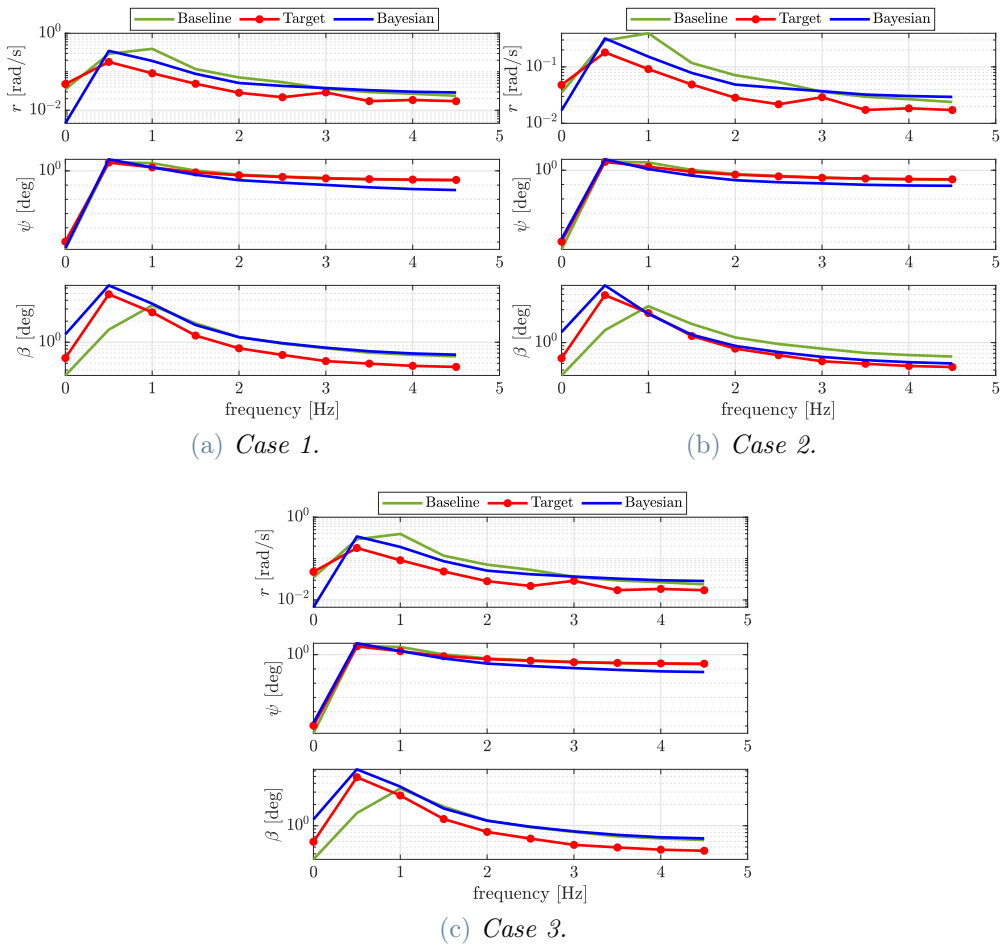


Figure 4.12: Error on  $r$ ,  $\psi$  and  $\beta$  in time domain during free response.

In fig. 4.13 it depicted the frequency vs. amplitude plot of the error between baseline solution and Bayesian solution, similarly to fig. 4.11. What emerge commonly by all the three cases, is that error in frequency of  $r$  and  $\beta$  is improved, while in  $\psi$  is unchanged. However, frequency of  $\psi$  is well described by baseline case too.

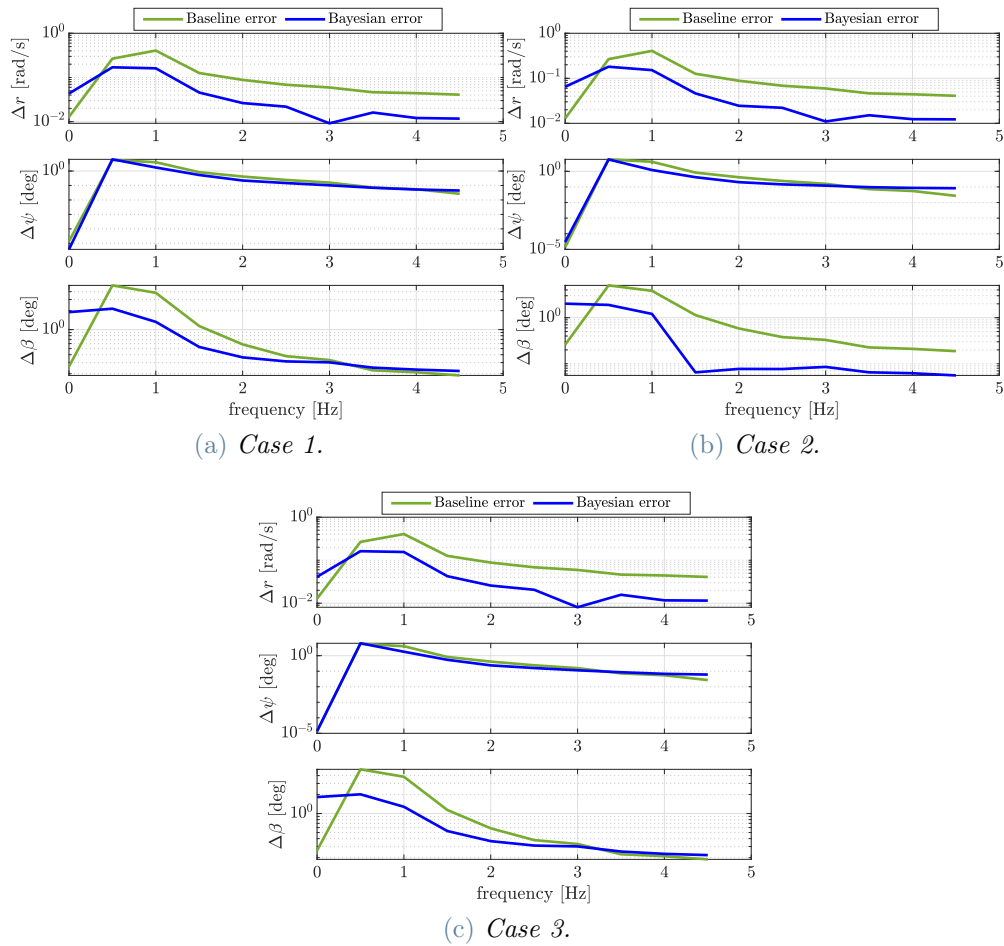


Figure 4.13: Error on  $r$ ,  $\psi$  and  $\beta$  in time domain during free response.

### 4.3.5. Final configuration

Having deeply analyzed the outcome of three cases, it is necessary to define which configuration more improve the lateral dynamics. The adopted procedure consists in assigning a figure of merit, through a score, to quality factor  $\eta$ , for each case and for each physical quantity. Then, summing up the total score it is possible to figure out the best solution. Considering a single quantity, two points are assigned to the case which is distinguished by the higher  $\eta$ , no points to the lower case, a single point for the middle case. If a quantity has been worsened in the identification procedure, no point are assigned. The system has been similarly adopted also to give a weight in the selection to the value of functional, obtained previously. Finally, Case 1 and case 3 are assigned seven extra points because of the acceptable geometry in final configuration. In table 4.11 the result of the procedure.

Final configuration selected is the one obtained in Case 3, in fig. 4.14, which has a score higher of point with respect to Case 2. The latter is the one which more improve flight data, but do not has a geometry similar to baseline case. Note that this solution improve the aircraft behaviour just on the point of view of the lateral dynamics, and the result is still far from a complete fidelization to real aircraft.



Figure 4.14: *SwitchMaster* in its updated configuration flying.



Table 4.11: Score table.

	Case 1	Case 2	Case 3
$\eta_{r,MEAN}$ Score	4.17	37.18 2	13.38 1
$\eta_{r,MAX}$ Score	50.43 2	42.02	51.06 1
$\eta_{\psi,MEAN}$ Score	35.11	58.86 2	36.94 1
$\eta_{\psi,MAX}$ Score	57.48	64.75 2	59.14 1
$\eta_{\beta,MEAN}$ Score	0.91	3.13 2	2.35 1
$\eta_{\beta,MAX}$ Score	57.48	64.75 2	59.14 1
$\eta_{r,freq}$ Score	57.19 1	31.10	57.49 2
$\eta_{\psi,freq}$ Score	-13.47	6.486 2	-13.17
$\eta_{\beta,freq}$ Score	57.19 1	31.10	57.49 2
<i>J</i> Value	1	2	
Geometry	5		5
<b>Total</b>	10	14	15



## 5 | Conclusions

The main goal of the present work was to improve the Simulink interface developed in a previous MSc thesis and to enhance the fidelity of *SwitchMaster* flight model, running on X-Plane. The present work has been articulated in two main steps:

- **Simulink update.** The Simulink update has been performed adding the possibility to send to X-Plane an input stored in a vector, instead of piloting the simulator through a joystick. This improvement has been performed modifying the existing platform, and executing a suitable test to confirm the result.
- **Model update.** The crucial aspect solved in the previous phase permitted to evaluate objectively the fidelity of *SwitchMaster* flight model. The critical aspect was the directional behaviour. Through an iterative method, the geometry of the aircraft has been gradually changed, in order to find a configuration that implies a flight dynamics more similar to the one of real aircraft.

The first step has been successfully achieved. Through the Simulink interface, it is possible to send a command input to the X-Plane model in order to replicate an experimental manoeuvre, previously executed with the a real aircraft. The operation has been performed to an adequate degree of precision, even if a small uncertainty is still present, due to a delay in communication between Simulink and X-Plane. This does not seem to be easily adjusted, as discussed in section 2.2. An open point is finding the reason of the existence of delay.

The second step has been only partially achieved. The iterative procedure, build through the use of a functional, with the aim of improve the fidelity of *SwitchMaster* flight model, correctly worked: unfortunately the outcome was not close enough to the target, namely the real aircraft behaviour. In particular, the flight model has been improved with respect the baseline aircraft, but some issues persist on the values of damping and frequency of the lateral dynamics. The following open points should be faced, in view of the full completion of the activity here discussed:

- Verify if the X-Plane dynamics model is suitable for an aircraft with peculiar propulsive configuration, such as DEP.

- Find a procedure to customize the initial roll angle in X-Plane: solving this, it would be possible to execute a more precise iterative procedure.
- Find a procedure to customize the initial angular velocities: solving this, it would be possible to execute a more precise iterative procedure.
- Repeat experimental flight testing to obtain more target flight histories, in order to cross check the results obtained. For example, once a satisfactory configuration has been found (through an iterative method), basing on a peculiar manoeuvre, one can run the simulation of another manoeuvre that excites the same dynamics. If the aircraft geometry found in first case is the correct one, there should be a good matching of flight data even in the second case.
- Verify the state of fidelity of longitudinal dynamics. This aspect should be investigated performing other experimental flights, in which longitudinal behaviour of the aircraft is tested in manoeuvres such as phugoid excitation or short period excitation.

## 5.1. Future developments

Summarizing the whole work done in this thesis, it is possible to identify some aspects suitable for future developments or research opportunities. The following list resumes the considered possibilities:

- Perform standard identification procedure for the estimation of stability and control derivative, and compare the obtained results in terms of prediction error;
- Once the *SwitchMaster* flight model will be corrected, whatever method will be adopted, it will be possible to design and tune a control law on the simulator to manage the peculiarities on piloting given by DEP: in fact, as discussed in section 1.4, at low power the handling qualities of a DEP aircraft are different from a standard architecture aircraft;
- Use the same control law, validated on simulated X-Plane flight model, to control the physical aircraft and fix the handling quality issues.

## Bibliography

- [1] Alberti L. Aeroswitch project: Development of the flight simulator of a distributed electric propulsion demonstrator. Master's thesis, Politecnico di Milano, 2020.
- [2] Bills A. et al. Performance metrics required of next-generation batteries to electrify commercial aircraft. *ACS Energy Letters*, 5(2):663–668, 2020.
- [3] Cacciola S. Identificazione parametrica dell'aerodinamica del velivolo aleniaaermacchi m346: Sperimentazione, analisi, validazione. Master's thesis, Politecnico di Milano, 2008.
- [4] Friedrich C. and Robertson P.A. Hybrid-electric propulsion for aircraft. *Journal of Aircraft*, 52(1):176–189, 2015.
- [5] Klein V and Morelli E.A. *Aircraft System Identification: Theory and Practice*. AIAA Education Series, 2007.
- [6] Pasquali D. and Tombolini M. Aeroswitch project: Proof of concept of a distributed electric propulsion aircraft enforcing flight symmetry control. Master's thesis, Politecnico di Milano, 2020.
- [7] Raymer D.P. *Aircraft Design: A conceptual approach*. American Institute of Aeronautics and Astronautics, 1992.



# A | Appendix

## A.1. MatLab routines

### A.1.1. Interpolator block

In algorithm A.1 is reported the content of the interpolator block. Note that data input comes in a form of a matrix with two rows and as many columns as are the time steps. On first line it is reported the number of sample, on second line the correspondent value. First value of first line is 1: as a consequence, it must be subtracted before interpolation, taking in account the delay.

---

**Algorithm A.1** Iteration execution main script through the core function.

---

```

1: function actual_value = control(data, actual_time, sample_time, input_delay)
2: sample_time_interp=data(1,:)*sample_time-input_delay;
3: sample_values=data(2,:);
4: actual_value = interp1(sample_time_interp - sample_time, ...
5: sample_values, actual_time, 'linear', 0);
6: end

```

---

### A.1.2. Iterative routine

Following routines has been developed to handle the iteration of optimization. algorithm A.2 works synergically with algorithm A.4: in particular, the first one gives in input a set of geometrical parameters, and the second one runs a iteration with a set of parameter given in input by the first one. Once the simulation has run, output data are stored in suitable structure. A posteriori of the evaluation of all iterations, the functional is calculated by appendix A.1.2.

---

**Algorithm A.2** Iteration execution main script through the core function.

---

```

1: t = 0;
2: index_vector = [];
3:
4: for p = linspace(0.5,1.5,25) % Geometrical parameter variation update.
5: for Z_arm = linspace(1.0,4.2,25) % Geometrical parameter variation update.
6:
7: t = t + 1
8: H = p * 0.79;
9: Ctip = 1.15 * p;
10: Croot = 1.15 * p;
11:
12: % Execution of a Simulink and X-Plane iteration.
13: [J_compl,J_reduced,p_interp,r_interp,beta_interp,roll_interp,hdg_interp] = ...
14: sim_routine7(address_simulink, real_time, real_angvel_pqr, real_angpos_rpy, ...
15: beta_real, H, label_H, id_H, Ctip, label_Ctip, id_Ctip, Croot,
16: label_Croot, id_Croot, Zarm,label_Zarm,id_Zarm);
17:
18: % Flight data are stored in suitable structures, as it follows.
19: index_vector(t,:)= [p Z_arm J_reduced];
20: p_matrix(:,t)=p_interp;
21: beta_matrix(:,t)=beta_interp;
22: hdg_matrix(:,t)=hdg_interp;
23:
24: end
25: end
26:
27: save('sim_data') % All data are finally saved in a .mat file.

```

---



---

**Algorithm A.3** A posteriori evaluation of functional.
 

---

```

1: load (sim_data) %Data of the whole iteration are loaded
2:
3: % Calculation of R matrix, as defined.
4: v_p_base = p_interp_base' - real_angvel_pqr(:,1);
5: v_beta_base = beta_interp_base' + beta_real;
6: v_hdg_base = (hdg_interp_base' - hdg_interp_base(1)') - (real_angpos_rpy(:,3)
   - real_angpos_rpy(1,3));
7: R=diag([abs(max(v_p_base)) abs(max(v_beta_base)) abs(max(v_hdg_base))]);
8:
9: for k = 1 : size(p_matrix,2) % Cycle for error evaluation.
10: v_p = p_matrix(:,k)-real_angvel_pqr(:,1); % Error on yaw rate.
11: v_beta = beta_matrix(:,k)+beta_real; % Error on sideslip angle.
12: v_hdg = hdg_matrix(:,k)-real_angpos_rpy(:,3); % Error on yaw angle.
13:
14: for j=1:length(real_time) %Cycle for functional evaluation
15: V_operation= [v_p(j) v_beta(j) v_hdg(j)];
16: J_operation(j)=V_operation * R^-1 *V_operation';
17: J = sum(J_operation(2:end)) / (length(real_time) * length(V_operation));
18: end
19: J = double(J);
20: end
21:
22: % Update of index vector with functional vector, at each configuration.
23: index_vector(:,4)=J';
24: [min_J,min_index]=min(index_vector(:,4));
25: [max_J,max_index]=max(index_vector(:,4));

```

---

---

**Algorithm A.4** Core function.
 

---

```

1: function [p_interp, beta_interp, hdg_interp] ...
2: = sim_routine7 (addres, real_time, real_angvel_pqr, real_angpos_rpy, ...
3: beta_real, new1, row_id1, row_number1)
4:
5: acf_update_general3(new1,row_id1,row_number1); %This command update the
   aircraft configuration text file
6:
7: h = actxserver('WScript.Shell');
8: h.AppActivate('X-System'); %This command put X-Plane window in evidence.
9:
10: h.SendKeys('n'); %This command reset the aircraft physical model
11:
12: h.AppActivate('X-System');
13: h.SendKeys('m'); %This command reset the aircraft position to trim point
14:
15: sim(addres,inf) %This command execute Simulink simulation
16:
17: %At the end of Simulink iteration, flight data are stored by Simulink itself in
   workspace. Following functions re-sample the output data with same sample time
   of real data.
18: sim_time=time.signals.values;
19: p_interp = resampler(sim_time,p,real_time);
20: beta_interp = resampler(sim_time,beta_angle,real_time);
21: hdg_interp = resampler(sim_time,hdg,real_time);
22: end

```

---

## A.2. Simulations flight data

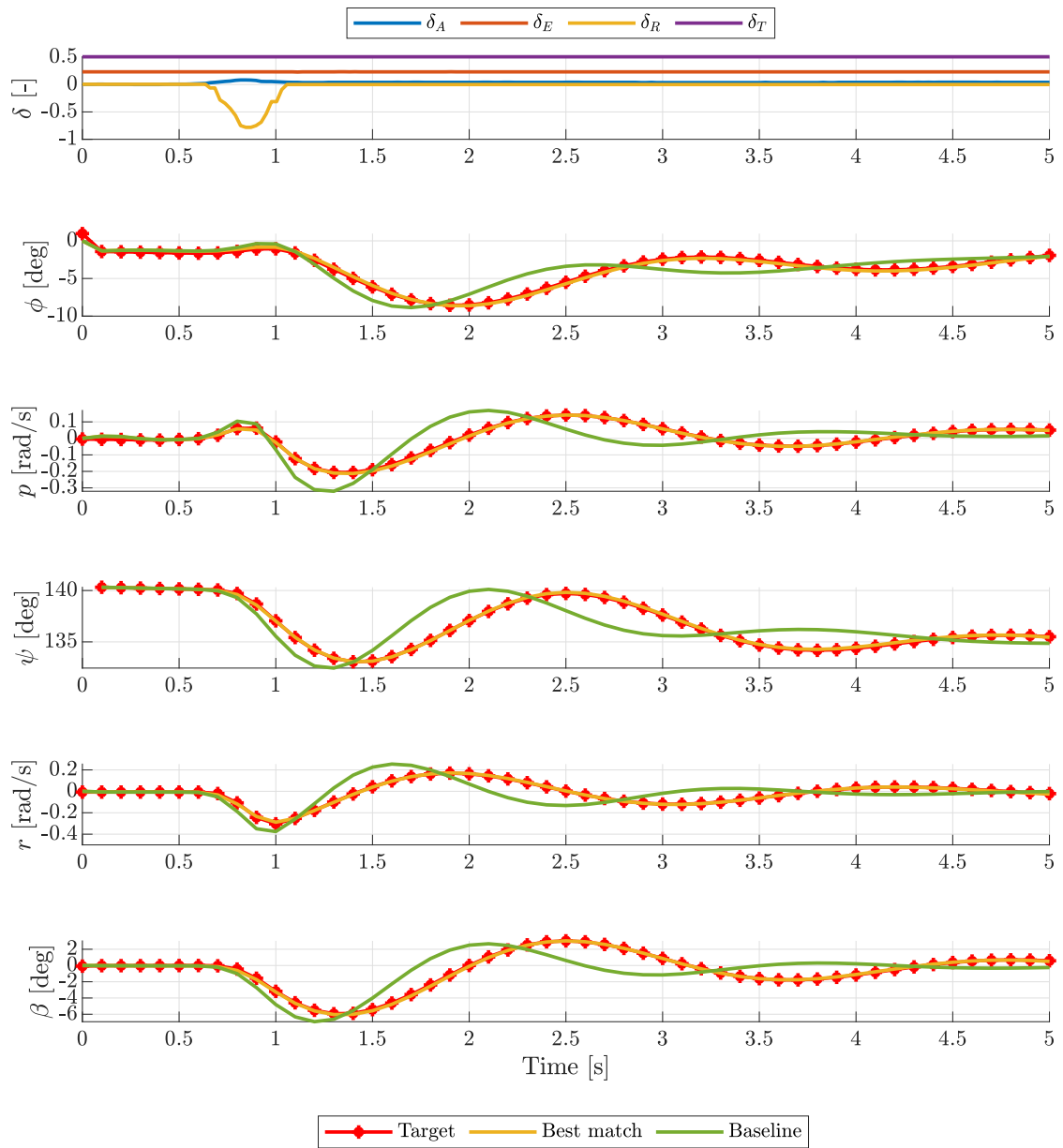


Figure A.1: Flight data of validation process with Robin 401.

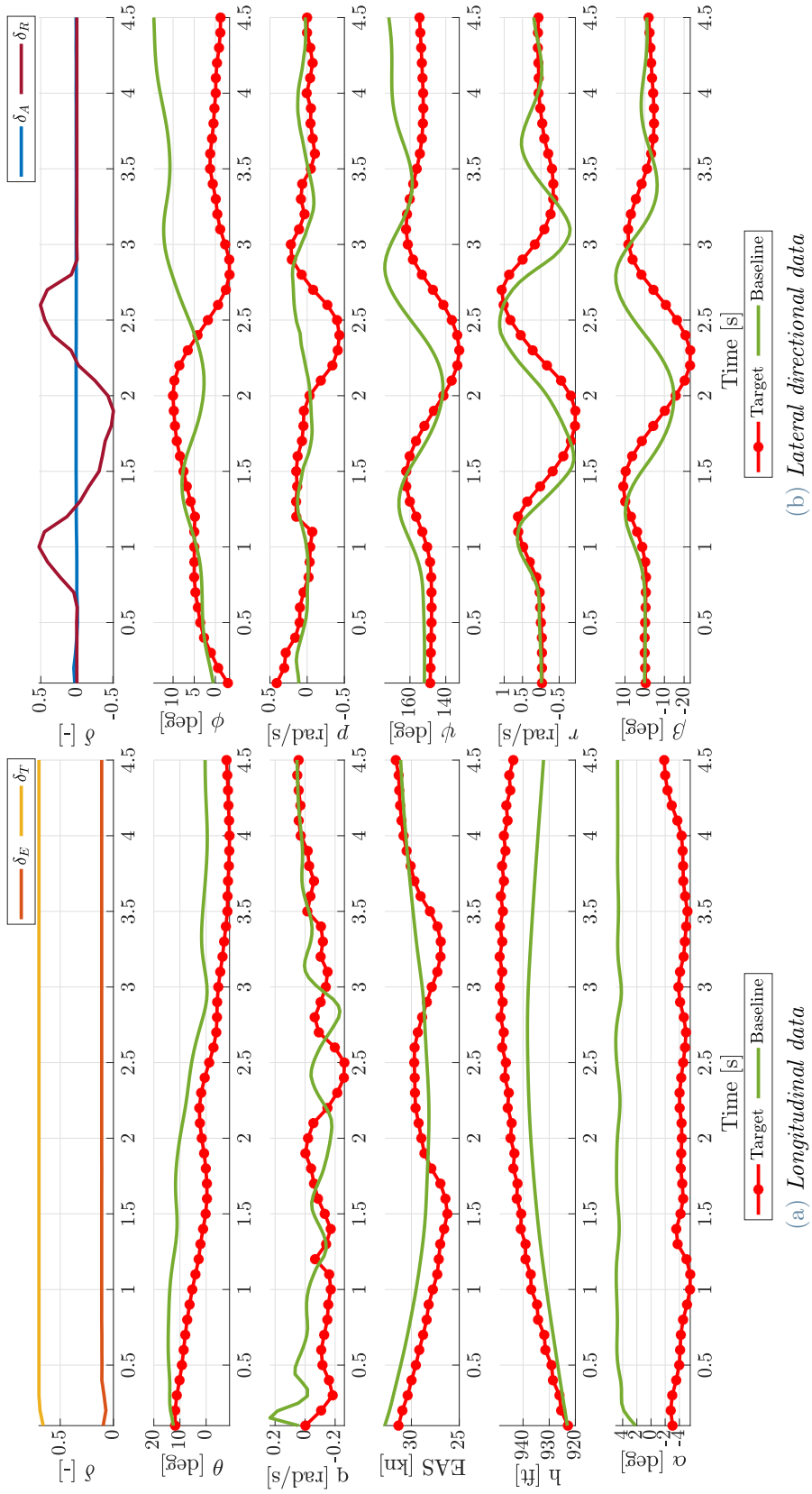


Figure A.2: Dutch roll test experimental data compared with baseline.

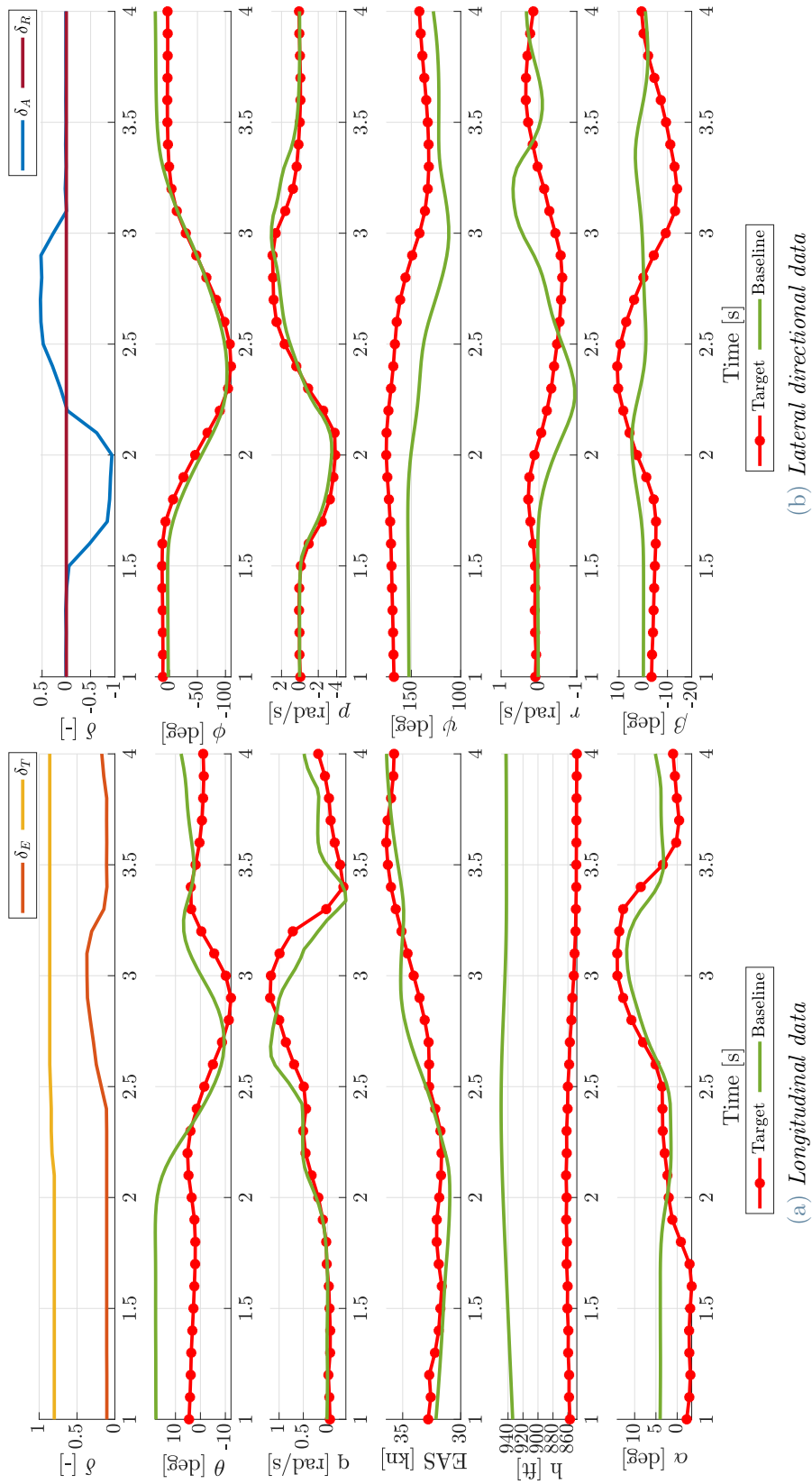


Figure A.3: Roll test experimental data compared with baseline.



## Ringraziamenti

Con questo lavoro si conclude il ciclo di studi in Ingegneria aeronautica. Il primo dei ringraziamenti va ai miei relatori, il prof. L. Trainelli, il prof. S. Cacciola, ed il prof. C.E.D. Riboldi, che con passione e determinazione mi hanno condotto negli ultimi anni di università.

E' stato imprescindibile il sostegno della mia famiglia, che è stata la culla della mia passione per il volo (a partire dai pomeriggi passati al campo volo di Bresso), e ha reso possibili gli studi.

La fatica di questo percorso è stata condivisa con gli amici Riccardo A., Riccardo B., Davide, Vincenzo e Matteo, assieme ai quali abbiamo portato a termine il fardello di tanti esami (e tante partite a carte).

Un ringraziamento particolare a Livia, che è sempre stata con me.

