UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Unmanned Aerial Vehicles Modelling and Control Design. A Multi-Objective Optimization Approach.

PhD Dissertation

Author: Jesús Velasco Carrau

Supervisor: Xavier Blasco
Supervisor: Sergio García-Nieto

September 2020

Instituto Universitario de Automática e Informática Industrial
Departamento de Ingeniería de Sistemas y Automática
Universitat Politènica de València

Agradecimientos Personales

Cada vez que abro y ojeo por encima este documento me asalta, lo primero, una fuerte sensación de satisfacción. Orgullo de ver el trabajo de muchos años y diferentes momentos de mi vida como investigador, recopilado y reflejado en estos párrafos. A continuación, siempre la misma pregunta ¿Cómo es posible que esto lo haya hecho yo? En seguida viene a mi mente un segundo nombre: Sergio. Tú has sido y eres mi mentor, compañero de pasiones y, por encima de todo, un buen amigo. Además de aquello que se le presupone a un buen director de tesis, has estirado incansablemente de mí cuando eso de ser doctor era ya un imposible que daba por perdido. Puedo afirmar sin temor a equivocarme que este documento, y lo que él representa, no existirían de no ser por ti. Así que gracias.

Xavi, director de directores, gracias por tu paciencia y comprensión, porque lo que iba a ser no fue. Has sido el oráculo al que se acaba acudiendo cuando surgen los problemas reales. Sudáfrica quedará siempre en mi memoria.

Gilberto, tu nombre debería aparecer debajo del de Xavi y Sergio, en portada como tercer director, y es una pena que no haya sido posible. Méritos no te faltaban. Me has aleccionado, inspirado, empujado y corregido, además de ser partícipe o colaborador en gran parte de los trabajos que aquí se presentan. Muchas gracias.

Terminando con el ámbito académico, quiero agradecer a todos los integrantes del CPOH que en uno u otro momento me habéis ayudado, a veces con gran esfuerzo y otras con un simple comentario, a sacar estos trabajos adelante. También a mis compañeros de sufrimiento: Ale, Gabi, Yadira, Vanessa, Diego y los demás; porque las cervesetas de los viernes han sido más esclarecedoras que días enteros de investigación científica.

Papá, mamá, voy a centrarme en lo referente a la tesis, porque los agradecimientos no se acabarían. Mientras Sergio estiraba, vosotros empujabais desde atrás ¡Así es imposible darse por vencido! No creo que haya habido una semana desde hace 4 años sin la pregunta de rigor... ¿Cómo va la tesis? No fuera a ser que me la quitase de la mente. Con esos y otros comentarios menos sutiles, pero sobre todo con vuestro apoyo incondicional ante cualquier adversidad, habéis conseguido que este trabajo vea la luz. No sé qué haría sin vosotros. El mérito es en gran medida vuestro.

Por supuesto al resto de esta familia gitana. Siempre atentos a todos los méritos de cada uno de sus integrantes. Seguidores incondicionales. Quiero nombrar especialmente a Tito que, en memoria del abuelo Vicente, recordaba incansable lo importante que es llegar a ser Doctor Ingeniero. Y, aunque todo el mundo sabe que los ingenieros de ahora ya no son como los de antes, y los Doctores Ingenieros mucho menos, me siento muy orgulloso de seguir sus pasos y convertirme en el segundo de esta familia. Me hubiera hecho muy feliz que estuviera aquí para verlo.

Para terminar, no un agradecimiento, sino una dedicatoria muy especial. Esta tesis se la dedico a mi hija Lucía. Eres hoy muy pequeña para leer y mucho menos entender estas palabras, pero algún día espero que inspiren tu camino. El esfuerzo, antes o después, se ve siempre recompensado. Elijas los objetivos que elijas en tu vida, esfuérzate al máximo para conseguirlos, y no dejes que nadie diga nunca que no puedes (ni siquiera tú misma). La actitud es infinitamente más importante que la aptitud. Si al esfuerzo le sumas una familia que te quiere y te apoya pase lo que pase (y tú la tienes) la combinación es infalible.

General Abstract

This thesis presents the results of the research work carried out on the modelling and design of controllers for micro-unmanned aerial vehicles by means of multi-objective optimization techniques. Two main fields of study are present throughout it. On one hand, the study of how to model and control small aerial platforms. And, on the other, the study on the use of heuristic multi-objective optimization techniques to apply in the process of models and controllers parameterization in micro-unmanned aerial vehicles. The main result is a series of tools that make it possible manage without wind tunnel experiments or high-cost air-data sensors, going directly to the use of experimental flight data in the parametric identification of dynamic models. In addition, a demonstration is given on how the use of multi-objective optimization tools in different phases of controller development helps to increase knowledge about the platform to be controlled and increases the reliability and robustness of the controllers developed, reducing the risk of hoping from the initial design phases to validation in real flight.

Resumen

Esta tesis presenta los resultados del trabajo de investigación llevado a cabo sobre el modelado y el diseño de controladores para micro-aeronaves no tripuladas mediante técnicas de optimización multi-objetivo. Dos principales campos de estudio están presentes a lo largo de ella. Por un lado, el estudio de cómo modelar y controlar plataformas aéreas de pequeña envergadura. Y, por otro, el estudio sobre el empleo de técnicas heurísticas de optimización multi-objetivo para aplicar en el proceso de parametrización de modelos y controladores en micro-aeronaves no tripuladas. Se obtienen como resultado principal una serie de herramientas que permiten prescindir de experimentos en túneles de viento o de sensórica de alto coste, pasando directamente a la utilización de datos de vuelo experimental en la identificación paramétrica de modelos dinámicos. Además, se demuestra como la utilización de herramientas de optimización multi-objetivo en diferentes fases del desarrollo de controladores ayuda a aumentar el conocimiento sobre la plataforma a controlar y aumenta la fiabilidad y robustez de los controladores desarrollados, disminuyendo el riesgo de pasar de las fases previas del diseño a la validación en vuelo real.

Resum

Aquesta tesi presenta els resultats de la feina de recerca dut a terme sobre el modelatge i el disseny de controladors per a micro-aeronaus no tripulades mitjançant tècniques d'optimització multi-objectiu. Dos principals camps d'estudi estan presents al llarg d'ella. D'una banda, l'estudi de com modelar i controlar plataformes aèries de petita envergadura. I, de l'altra, l'estudi sobre l'ús de tècniques heurístiques d'optimització multi-objectiu per aplicar en el procés de parametrització de models i controladors en micro-aeronaus no tripulades. S'obtenen com a resultat principal una sèrie d'eines que permeten prescindir d'experiments en túnels de vent o de sensòrica d'alt cost, passant directament a la utilització de dades de vol experimental a la identificació paramètrica de models dinàmics. A més, es demostra com la utilització d'eines d'optimització multi-objectiu en diferents fases de desenvolupament de controladors ajuda a augmentar el coneixement sobre la plataforma a controlar i augmenta la fiabilitat i robustesa dels controladors desenvolupats, disminuint el risc de passar de les fases prèvies de el disseny a la validació en vol real.

Contents

Chapter 1

Introduction and General Overview

1.1 Motivation and Goals

Normally, when a new aircraft is designed, computational fluid dynamics (CFD) software is used to study its aerodynamic behavior. Furthermore, before entering in production phase, scale models are constructed and wind tunnel experiments are conducted to confirm that specifications are complied. In this sense, although micro-unmanned aerial vehicles are very good choice in research applications (among others), they lack of the information given by both CFD software and wind tunnel experiments. The reasons for this are mainly two: they are frequently handmade and the price of wind tunnel experiments are prohibiting and much more expensive than the aircraft itself. In addition, when doing research, it is particularly difficult to implement new concepts and ideas in a real platform to test them. There are too demanding computational burdens, equipment safety risks, and others issues normally translated to economical limitations. In that sense, the use of realistic simulation environments becomes especially important for the successful study of those new concepts and ideas in the field of control systems.

When this research first started, it was supposed to be focused on the study of advanced control strategies applied to micro-unmanned aerial vehicles (MAVs). Techniques such as linear-quadratic regulation (LQR) or model predictive control (MPC) were intended to be applied. This type of controllers are highly based on the dynamic model that is used to design on the former and even to control on the latter.

Thus, a roadblock was found on the lack of a good dynamic model that could be used, first, to design such control techniques on MAVs and, second, to perform realistic simulations before real flight validation. With the believe that the same roadblock would be found by many others, the research focus was set on developing dynamic models in the absence of CFD or wind tunnel experiments information. The approach taken was to adopt first principles structures, where the physical interactions are well known, but using the power of multi-objective optimization to adjust them by means of real flights data.

When that task was partially fulfilled, the research was reoriented to the objective of taking profit of those models for control design and also for realistic simulation in platforms such us Hardware-In-the-Loop (HIL). This kind of simulation integrates a computational unit solving the equations of a dynamic model in real-time. This unit is integrated with the flight controller emulating a real situation in which the aircraft is flying and the autopilot is controlling it. The use of HIL platforms has become a standard practice in order to evaluate embedded controllers, with the goal of getting a more reliable measure of their performance [18, 37]. Such platforms are common in automotive [8] and aeronautic/aerospace sectors [15], where it is required to enhance the quality, safety and verification testing of their subsystems [28].

Presenting heuristic multi-objective optimization as the common denominator is motivated by the revelation of an overwhelming reality: a person seldom confronts a problem with a single objective, that has no restrictions and is convex at same time. Besides, even if the problem is multi-objective, the person does not normally have a deep understanding of how far each goal can be achieved and which is the most preferable trade-off among his/her objectives. Therefore, heuristic multi-objective optimization raises as a very powerful tool to address realistic problems, and opens a wide variety of opportunities to pose and solve engineering problems from a holistic perspective.

Therefore, two main fields of study are present on this thesis. On one hand, the study of how to model and control an aircraft system. And, on the other hand, the study of heuristic multi-objective optimization techniques to be applied on parameterizing models and controllers in aircraft systems. A brief introduction on those two fields of study is now given.

1.2 Unmanned Aerial Vehicles: brief introduction

Before entering in more details and with the aim of offering a brief background on the field of unmanned aerial vehicles, an explanation on their definition, classification, recent history and some of their already existing and potential applications is given.

1.2.1 Definition and classification

Unmanned aerial vehicle (UAV) or unmanned aeronautical system (UAS) means an aircraft capable of carrying out a mission without the need of a crew on board. It must be understood that this condition does not exclude the existence of a pilot, mission controller or other operators, who can carry out their work from the ground. The extension of the concept of vehicle to system reflects that the UAS requires not only the aircraft with instrumented precision but also a ground station that completes the instrumentation and embedded capabilities [3].

The above definition is perhaps too general, since it covers some systems that might fall into it but are not considered UAV. In this way, and to complete the definition, we can add that an unmanned aircraft is an unmanned vehicle, reusable, capable of maintaining a controlled and sustained level of flight, and propelled by an electrical motor or by a combustion engine.

It is clear from the latter that, for example, a missile, although it is often controlled and unmanned, is not considered UAV because it is not reusable. Another example could be found in hot air balloons which, although reusable and unmanned, they are not controlled.

It should be taken into account then that these definitions do not specify that the aircraft must be autonomous and, consequently, they could easily be controlled from the ground and still be treated as UAV or UAS. It is therefore possible to define the concept of autonomous aircraft (AA: Autonomous Aircraft) or Autonomous Air System (AAS: Autonomous Aerial System) as one capable of carrying out the mission autonomously without the need of human intervention [3].

Once the concept is defined, we can classify UAVs according to some of their most important characteristics. We could make a first classification according to the type of aircraft itself. Figure 1.1 shows this classification.

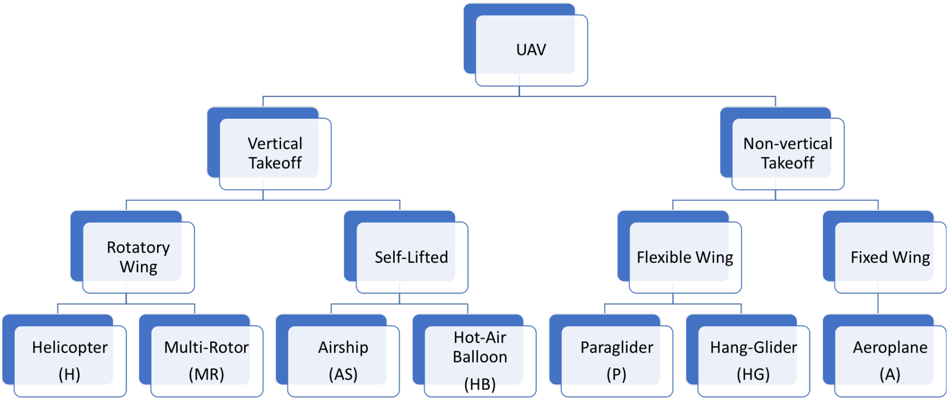
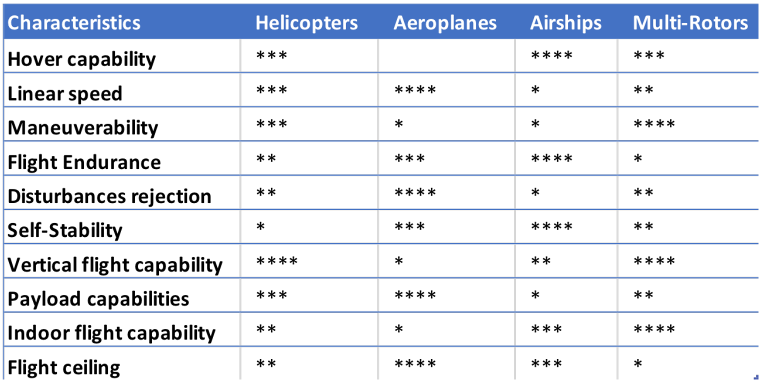


Figure 1.1. UAV classification according to the aircraft type [3]

From diagram on fig. 1.1 we can highlight the rotating wing vehicles, airships and airplanes as those that cover the largest number of unmanned aircraft, relegating the rest of them to more specific applications. According to this, table 1.1 shows the most important characteristics of the UAVs mentioned.

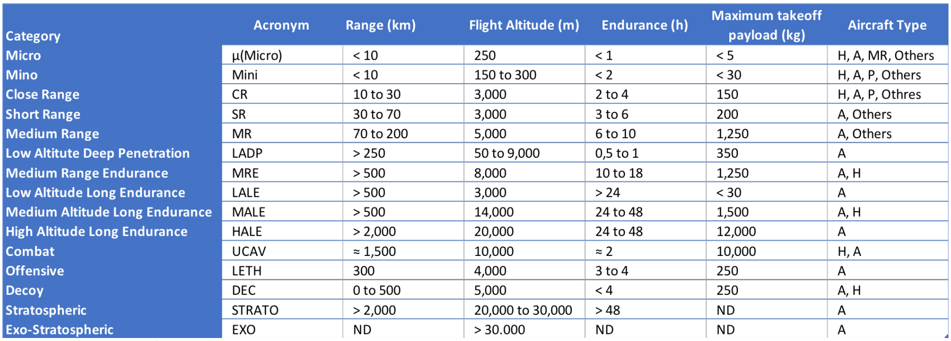


Characteristics	Helicopters	Aeroplanes	Airships	Multi-Rotors
Hover capability	***		****	***
Linear speed	***	****	*	**
Maneuverability	***	*	*	****
Flight Endurance	**	***	****	*
Disturbances rejection	**	****	*	**
Self-Stability	*	***	****	**
Vertical flight capability	****	*	**	****
Payload capabilities	***	****	*	**
Indoor flight capability	**	*	***	****
Flight ceiling	**	****	***	*

Table 1.1. UAV characteristics according to aircraft type [3]

Thus, depending on the application, the design engineer should put the focus on one or more of these characteristics, which will lead to the choice of one type of aircraft over another.

We can also classify UAVs according to the flight capabilities they have, such as range, flight altitude, endurance and maximum takeoff payload. Table 1.2 shows indeed this classification in 15 categories and gives insights of the most suitable type or types of aircraft for each category.



Category	Acronym	Range (km)	Flight Altitude (m)	Endurance (h)	Maximum takeoff payload (kg)	Aircraft Type
Micro	μ(Micro)	< 10	250	< 1	< 5	H, A, MR, Others
Mino	Mini	< 10	150 to 300	< 2	< 30	H, A, P, Others
Close Range	CR	10 to 30	3,000	2 to 4	150	H, A, P, Othres
Short Range	SR	30 to 70	3,000	3 to 6	200	A, Others
Medium Range	MR	70 to 200	5,000	6 to 10	1,250	A, Others
Low Altitute Deep Penetration	LADP	> 250	50 to 9,000	0,5 to 1	350	A
Medium Range Endurance	MRE	> 500	8,000	10 to 18	1,250	A, H
Low Altitude Long Endurance	LALE	> 500	3,000	> 24	< 30	A
Medium Altitude Long Endurance	MALE	> 500	14,000	24 to 48	1,500	A, H
High Altitude Long Endurance	HALE	> 2,000	20,000	24 to 48	12,000	A
Combat	UCAV	≈ 1,500	10,000	≈ 2	10,000	H, A
Offensive	LETH	300	4,000	3 to 4	250	A
Decoy	DEC	0 to 500	5,000	< 4	250	A, H
Stratospheric	STRATO	> 2,000	20,000 to 30,000	> 48	ND	A
Exo-Stratospheric	EXO	ND	> 30.000	ND	ND	A

Table 1.2. UAV classification according to range, flying altitude, endurance and maximum takeoff payload [3]

1.2.2 Background

The history of UAVs dates back to the mid-19th century: a primitive UAV formed by a balloon loaded with bombs was used on August 22, 1849 in an Austrian attack on the city of Venice. Subsequently, the cruise missiles arrived, controlled by a system of gyroscopes during World War I and radio-controlled aircraft used to train British anti-aircraft shooters during World War II [2].

In 1950, the US Army realised of the importance and need of an aerial platform that delivered truthful and timely information that allowed for correct and timely decision making. In the wars of Korea and Vietnam, the United States army found in the UAVs a way to divert enemy attacks from their bombers and manned fighters and the first reconnaissance UAVs were also developed. Although the United States used unmanned aerial platforms in the Vietnam War, it was Israel who confirmed, during its operations in Lebanon in 1982, the importance of having such systems, which increased international interest in UAVs [1].

In the Desert Storm operation in 1991, the US Navy used the Israeli Pioneer UAV system to provide intelligence at the tactical level. In Afghanistan, during the Enduring Peace operation, the UAV Predator system carried out armed reconnaissance missions and in 2003 in Iraq it attacked objectives of great value for the coalition. Predator also cooperated with the Special Forces in the search and information of SCUD missile locations. Figure 1.2 shows both images of Pioner and Predator UAVs.

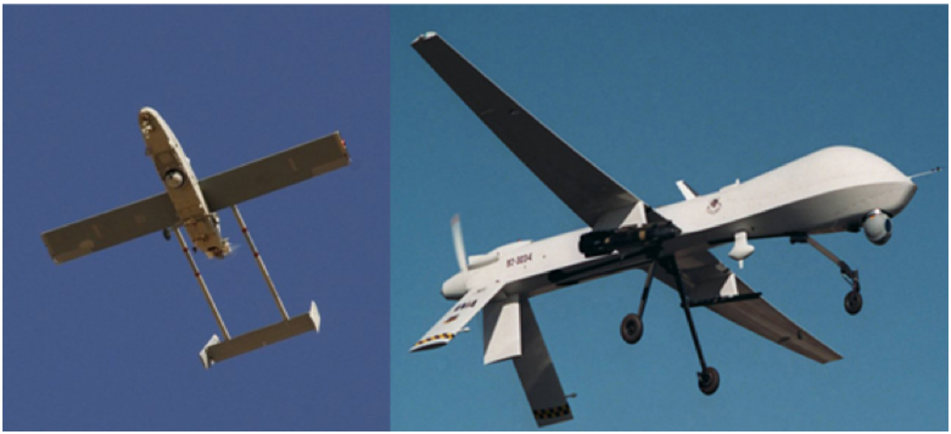
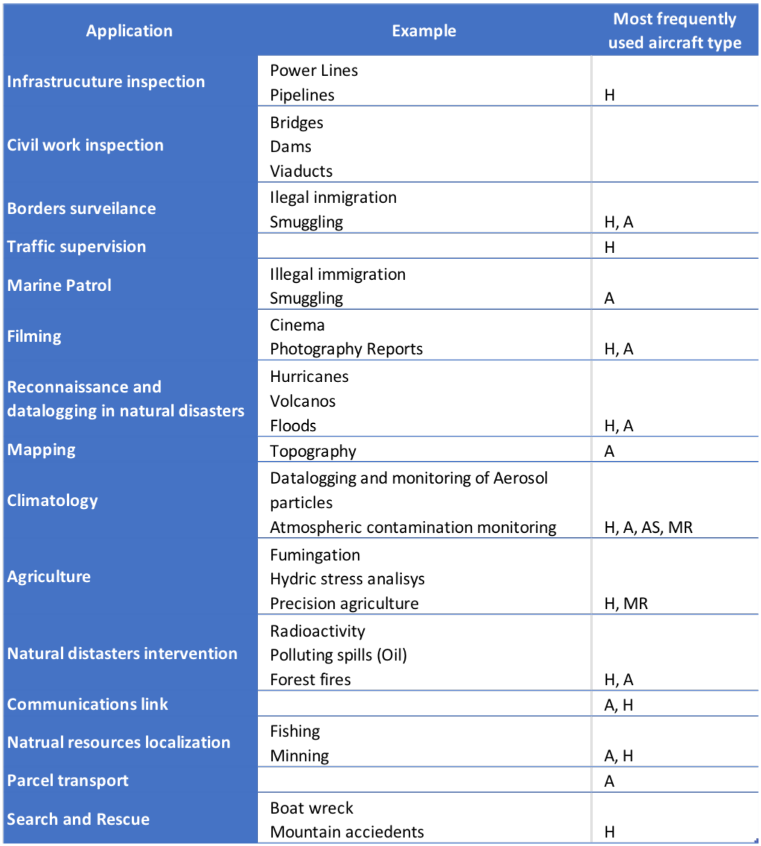


Figure 1.2. Pioneer and Predator UAVs[3]

1.2.3 Applications

The general usages given to the different type of UAVs similar to those of their manned counterparts. However, UAVs have an advantage over conventional airplanes in those applications in which the activity carried out involves an inherent risk for the pilot, is too monotonous or is generally unwanted by the pilots.

As it has been seen, the first UAVs were designed for military applications and there are a large number of applications oriented to this field. However, a growing number of UAVs designed for endless civil applications can be found recently. Table 1.3 shows many of those applications and the type or types of aircraft mostly used.



Application	Example	Most frequently used aircraft type
Infrastrucuture inspection	Power Lines Pipelines	H
Civil work inspection	Bridges Dams Viaducts	
Borders surveilance	Ilegal inmigration Smuggling	H, A
Traffic supervision		H
Marine Patrol	Illegal immigration Smuggling	A
Filming	Cinema Photography Reports	H, A
Reconnaissance and datalogging in natural disasters	Hurricanes Volcanos Floods	H, A
Mapping	Topography	A
Climatology	Datalogging and monitoring of Aerosol particles Atmospheric contamination monitoring	H, A, AS, MR
Agriculture	Fumingation Hydric stress analisys Precision agriculture	H, MR
Natural distasters intervention	Radioactivity Polluting spills (Oil) Forest fires	H, A
Communications link		A, H
Natrual resources localization	Fishing Minning	A, H
Parcel transport		A
Search and Rescue	Boat wreck Mountain acciedents	H

Table 1.3. UAV Civil applications for UAVs and type of aircraft most used for each application [3].

1.2.4 Research flight platforms

Attending to flying capabilities, this research is focused on the study of parametric identification and control design for MAVs. The reason behind that choice is multiple. First, this type of platforms are considerably cheaper than any other type, what makes them attractive not only for academic purposes but also for companies and end users. Despite MAVs reduced range, endurance and maximum takeoff payload, the miniaturisation and price reduction path of electronic components within the last decade still qualify them for most of the UAVs applications in Table 1.3. Hence, being the first choice in most end-user applications make them also the best choice for a meaningful research. Second, when dynamics identification and automatic control are the main core of the research, it is common that flying envelope limits are reached and even exceeded. In this situation accidents are likely to happen. Therefore, a reduced price that allows easy replacement and a small size helping to limit the consequences of an accident are very helpful features. Finally, to keep them cost effective, MAVs are rarely tested by manufacturers in wind-tunnel experiments. There is, so, not much information about their aerodynamic parameters and other constructive values. For this reason, there is a need for developing parametric identification strategies that help to extract data without performing expensive tests. This was also a need found to achieve the objectives related to control design for MAVs of this thesis.

Two types of aircraft have been used along the research as flight platform. First, a non-vertical takeoff and fixed wing aeroplane of around 5 kg mass, the Kadett 2400. It is presented on chapters 2, 3, 4 and 5 as the platform used in experimental flights. Its very lightweight frame, wing surface and the available volume to house control hardware made it suitable to be used in the research. On chapter 6 a tilt rotor MAV is presented with the nickname of V-Skye. It possesses the capability of behaving as a VTOL and as a fixed wing aircraft at the same time. This is what is called a hybrid UAV. Such a platform presents an interesting modelling and control challenge due to the special characteristics of its dynamic model and, in consequence, it is a good test-bench for the techniques developed along this thesis.

These two flight platforms and the equations of motion that describe their dynamics are presented next.

Kadett 2400

The Kadett 2400 is an aircraft manufactured by Graupner. The aircraft has a very lightweight frame and characteristics that make it suitable for the purposes of this research. These characteristics include a 2.4 m wing span, 0.9 m^2 of wing surface, 48.07 $\mathrm{N/m}^2$ wing loading, and 1.65×10^{-2} m^3 of available volume to house control hardware.

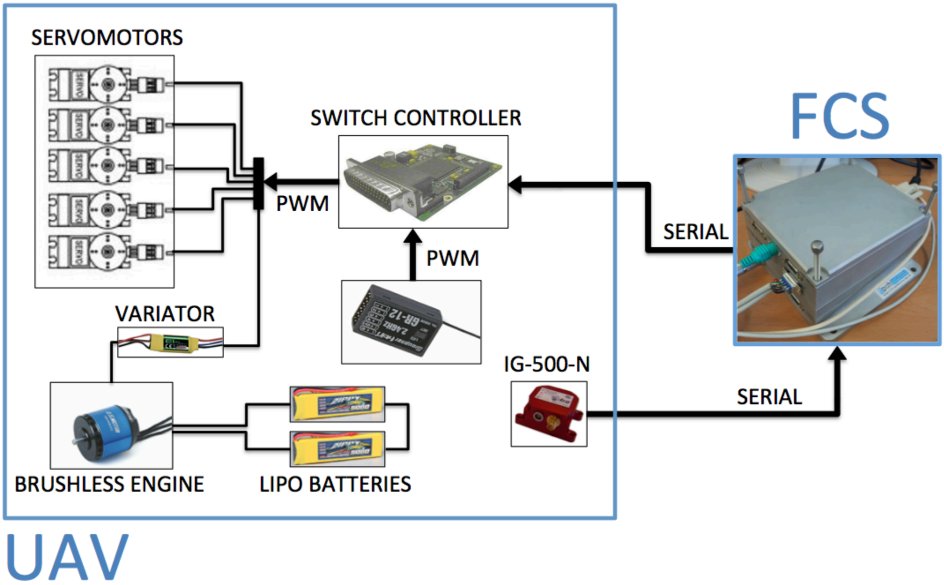


Figure 1.3. Interconnection between the UAV devices and the flight control system

Table 1.4. IG500N unit characteristics

Sensor	Characteristic	Value
Unit attitude		
	Static accuracy (Pitch)	±0.5 deg
	Static accuracy (Roll)	±0.5 deg
	Static accuracy (Heading)	±1.0 deg
	Dynamic accuracy	±1.0 deg rms
Accelerometers		
	Non-linearity	< 0.2 % of full scale
	Bias stability	±5 mg
Gyroscopes		
	Non-linearity	< 0.1 % of full scale
	Bias stability	±0.5 deg/s
Magnetometers		
	Non-linearity	< 0.2 % of full scale
	Bias stability	±0.5 mG
GPS Receiver		
	Horizontal accuracy	2.0 m
	Vertical accuracy	5.0 m

Figure 1.3 illustrates interconnection between the UAV devices and the flight control system. The aircraft houses all the devices necessary for manual, as well as automatic, control. During normal flight, the tail rudder, elevators, and ailerons serve as the control surfaces. Propulsion is provided by a brushless alternating current engine supplied by two lithium-ion polymer (LiPo) batteries through a frequency variator. The variator and the servomotors are controlled by pulse width modulated (PWM) command signals. The servo switch controller (SSC) switches between manual and autonomous flight modes. It also enables data acquisition and the application of control surface deflections and motor torque changes.

The flight control station (FCS), housed in a PC-104, hosts the control algorithms. The control loop is closed by a IG500N unit from SBG Systems, that integrates a wide range of sensors, including the accelerometers, gyroscopes, and magnetometers. A Kalman filter fuses the sensor information to estimate the position, orientation, linear and angular speed, and acceleration. Table 1.4 provides manufacturer's accuracy data for the IG500N unit. This same platform was presented in [35, 34, 33, 32] together with the results of the first flight tests.

Now the aircraft model equations are stated as a compendium of dynamics, aerodynamics and kinematics equations:

Translational dynamics equations

$$\begin{aligned} \dot{u} &= rv - qw + \frac{\overline{q}S}{m} C_X(\delta_{[e,a,r]}) - g\sin\theta + \frac{T}{m} \\ \dot{v} &= pw - ru + \frac{\overline{q}S}{m} C_Y(\delta_{[e,a,r]}) + g\cos\theta\sin\phi \\ \dot{w} &= qu - pv + \frac{\overline{q}S}{m} C_Z(\delta_{[e,a,r]}) + g\cos\theta\cos\phi \end{aligned} \tag{1.1}$$

Rotational dynamics equations

$$\begin{aligned} \dot{p} - \frac{I_{xz}}{I_x}\dot{r} &= \frac{\overline{q}Sb}{I_x} C_l(\delta_{[e,a,r]}) - \frac{I_z - I_y}{I_x} qr + \frac{I_{xz}}{I_x} qp \\ \dot{q} &= \frac{\overline{q}S\overline{c}}{I_y} C_m(\delta_{[e,a,r]}) - \frac{I_x - Iz}{I_y} pr - \frac{I_{xz}}{I_y}(p^2 - r^2) + I_p\Omega_p r \\ \dot{r} - \frac{I_{xz}}{I_z}\dot{p} &= \frac{\overline{q}Sb}{I_z} C_n(\delta_{[e,a,r]}) - \frac{I_y - I_x}{I_z} pq - \frac{I_{xz}}{I_z} qr - I_p\Omega_p q \end{aligned} \tag{1.2}$$

Translational kinematic equations

$$\begin{aligned}
u &= \dot{x}_e \cos\theta\cos\psi + \dot{y}_e \cos\theta\sin\psi - \dot{z}_e \sin\theta \\
v &= \dot{x}_e \left(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi\right) \\
&\quad + \dot{y}_e \left(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi\right) + \dot{z}_e \sin\phi\cos\theta \\
w &= \dot{x}_e \left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right) \\
&\quad + \dot{y}_e \left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right) + \dot{z}_e \cos\phi\cos\theta
\end{aligned} \tag{1.3}$$

Rotational kinematic equations (Euler angles)

$$\begin{aligned}
\dot{\phi} &= p + \tan\theta(q\sin\phi + r\cos\phi) \\
\dot{\theta} &= q\cos\phi - r\sin\phi \\
\dot{\psi} &= \frac{q\sin\phi + r\cos\phi}{\cos\theta}
\end{aligned} \tag{1.4}$$

In Eq. 1.1, Eq. 1.2, Eq. 1.3 and Eq. 1.4 g is the gravitational field intensity near the Earth's surface, and m is the total mass of the system. (x_e, y_e, z_e) are the components of the aircraft position on an Earth reference frame situated and attached to a point of the Earth surface. Given the body reference frame $X_b Y_b Z_b$ illustrated in Fig. 1.4, (u, v, w) are the components of the translational velocity, (p, q, r) the components of the angular velocity, (I_x, I_y, I_z) are the moments of inertia, and I_{xz} is a product of inertia. The products of inertia I_{xy} and I_{yz}, related to the longitudinal plane ($Y_b = 0$), are both null because of the aircraft's symmetry with respect to this plane. I_p is the rotating inertia of the tandem motor and propeller, Ω_p is its rotating speed, and T is the motor thrust. S, b and $\overline{c}$ are the the Kadett 2400 aerodynamic surfaces, wingspan, and wing chord respectively, and $\overline{q}$ is the dynamic pressure, which is a function of the air density and airspeed relative to the local wind. The aerodynamic coefficients (AC) C_X, C_Y, C_Z, C_l, C_m, and C_n, are functions of the system variables. In particular, the δ symbol in brackets represents its dependency on the deflections of the control surfaces (δ_e, δ_a and δ_r are the elevators, ailerons, and rudder deflections respectively). The aerodynamic coefficients will be presented in further detail in chapters 2 and 3. Finally, the aircraft orientation is represented by the Euler angles of roll ϕ, pitch θ, and yaw ψ.

Aircraft Aerodynamic Model

In Klein and Morelli [16], detailed information on the aerodynamic coefficients is provided. Firstly, if we assume a scenario in which the aircraft is in steady flight, and only performs short manoeuvres that begin from this state, we can truncate the Taylor series expansion and retain only the first or second-order terms. Furthermore, under the assumption of small perturbations, and based on the symmetry of the vehicle, it can be assumed that: 1) the symmetrical (longitudinal) variables u, w and q do not affect asymmetrical (lateral) force and torques; and similarly, 2) asymmetric (lateral) variables v, p and r do not affect the symmetrical (longitudinal) forces and torque. The aerodynamic coefficients are given by the longitudinal aerodynamic models,

$$\begin{aligned}
C_D = \quad & C_{D_0} + C_{D_{V_{air}}} \tfrac{1}{V_0} \Delta V_{air} + C_{D_\alpha} \Delta\alpha + C_{D_{\alpha^2}} \Delta\alpha^2 \\
& + C_{D_q} \tfrac{\bar{c}}{2V_0} q + C_{D_{\delta_e}} \Delta\delta_e \\
C_L = \quad & C_{L_0} + C_{L_{V_{air}}} \tfrac{1}{V_0} \Delta V_{air} + C_{L_\alpha} \Delta\alpha + C_{L_{\alpha^2}} \Delta\alpha^2 + C_{L_{\dot\alpha}} \tfrac{\bar{c}}{2V_0} \dot\alpha \\
& + C_{L_q} \tfrac{\bar{c}}{2V_0} q + C_{L_{\delta_e}} \Delta\delta_e \\
C_m = \quad & C_{m_0} + C_{m_{V_{air}}} \tfrac{1}{V_0} \Delta V_{air} + C_{m_\alpha} \Delta\alpha + C_{m_{\alpha^2}} \Delta\alpha^2 + C_{m_{\dot\alpha}} \tfrac{\bar{c}}{2V_0} \dot\alpha \\
& + C_{m_q} \tfrac{\bar{c}}{2V_0} q + C_{m_{\delta_e}} \Delta\delta_e
\end{aligned} \tag{1.5}$$

and the lateral aerodynamic models,

$$\begin{aligned}
C_Y = & C_{Y_0} + C_{Y_\beta} \Delta\beta + C_{Y_p} \frac{b}{2V_0} p + C_{Y_r} \frac{b}{2V_0} r + C_{Y_{\delta_a}} \Delta\delta_a + C_{Y_{\delta_r}} \Delta\delta_r \\
C_l = & C_{l_0} + C_{l_\beta} \Delta\beta + C_{l_p} \frac{b}{2V_0} p + C_{l_r} \frac{b}{2V_0} r + C_{l_{\delta_a}} \Delta\delta_a + C_{l_{\delta_r}} \Delta\delta_r \\
C_n = & C_{n_0} + C_{n_\beta} \Delta\beta + C_{n_p} \frac{b}{2V_0} p + C_{n_r} \frac{b}{2V_0} r + C_{n_{\delta_a}} \Delta\delta_a + C_{n_{\delta_r}} \Delta\delta_r
\end{aligned} \tag{1.6}$$

where α and β are the angle of attack and of sideslip, respectively, and V_{air} is the airspeed (see Fig. 1.4). In particular, V_0 is airspeed measured at the steady state of flight, before a manoeuvre begins. These variables are velocity dependent and calculated as:

$$\alpha = \arctan\left(\frac{w_{air}}{u_{air}}\right); \quad \text{and} \quad \beta = \arcsin\left(\frac{v_{air}}{V_{air}}\right); \tag{1.7}$$

where $V_{air} = |\mathbf{V_{air}}|$. As denoted in Fig. 1.4, u_{air}, v_{air} and w_{air} are the three components of the aircraft velocity with respect to air. Under zero-wind conditions $(u_{air}, v_{air}, w_{air}) = (u, v, w)$. Finally, C_L and C_D are the lift and drag coefficients and their relation to C_X and C_Z is:

$$C_L = -C_Z \cos\alpha + C_X \sin\alpha; \quad \text{and} \quad C_D = -C_X \cos\alpha - C_Z \sin\alpha; \tag{1.8}$$

Thus, the aerodynamic model identification is based on extracting the constants of the polynomials of Eq. (1.5) and (1.6) from the flight data and by means of the

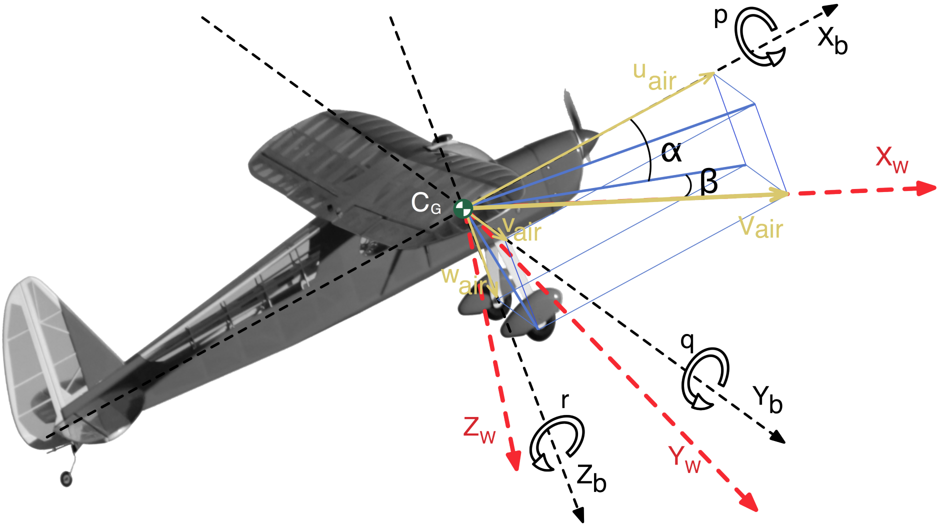


Figure 1.4. Aircraft body axes, aircraft wind axes, and wind relative velocity

dynamic model. Those constants are called non-dimensional stability and control derivatives.

V-Skye

This hybrid UAV concept can be categorized as Tail-Sitter with the exceptions that in this case the aircraft takes off and lands vertically on its nose (using an external ground-station) and that this platform changes the sense of the rotors in order to perform the transition phase between hovering and cruising. Figure 1.5 shows the maneuverability scheme of this unmanned aerial vehicle. In addition, the prototype built based on this philosophy has been *nicknamed* V-Skye.

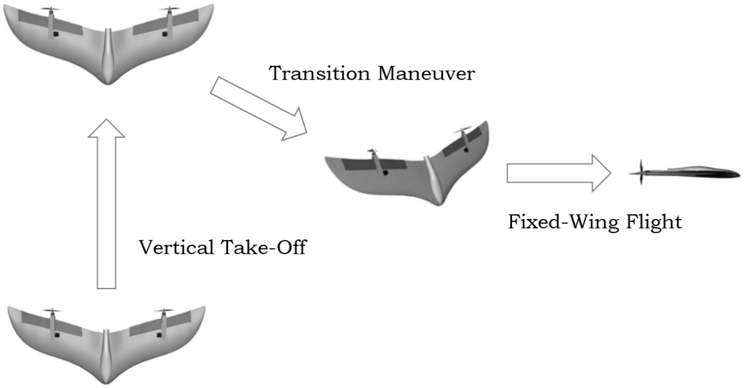


Figure 1.5. Scheme of the transition maneuver between flight modes

The V-Skye is designed with two tilting-rotors moved by servo-mechanisms. The result is a vehicle with two motors for which thrust $\vec{T}_R$ and $\vec{T}_L$ can be independently modified, not only in magnitude, but also in one direction. The system is thus provided with the amount of independent inputs needed for the hovering maneuver.

Figure 1.6 shows an outline drawing of the V-Skye. In order to simplify the dynamics, all actuation parts (motors, motor frames, servomotors and their transmission parts) are allocated as symmetrically as possible about the fixed coordinate axis $\{\hat{X}_b, \hat{Y}_b, \hat{Z}_b\}$ of the aircraft reference frame. In particular, all elements are placed on the $\hat{Y}_b\hat{Z}_b$ plane and symmetrical to the $\hat{X}_b\hat{Z}_b$ plane.

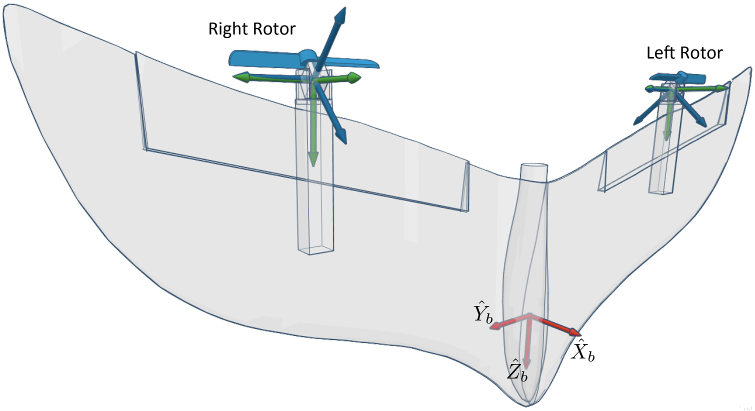


Figure 1.6. Local axis in the 3D graphical model of the V-Skye UAV

The aircraft model as a fixed wing aircraft is similar to the one presented for the Kadett 2400. The following sets of equations describe the dynamics of the V-Skye as a VTOL:

Translational dynamics equations

$$\dot{u} = rv - qw - g\sin\theta - \frac{k_T}{m}\left(\delta_R \sin\lambda_R + \delta_L \sin\lambda_L\right) \tag{1.9}$$

$$\dot{v} = pw - ru + g\cos\theta\sin\phi \tag{1.10}$$

$$\dot{w} = qu - pv + g\cos\theta\cos\phi - \frac{k_T}{m}\left(\delta_R \cos\lambda_R + \delta_L \cos\lambda_L\right) \tag{1.11}$$

where δ_R and δ_L are the right and left throttle rates respectively, k_T is the motors thrust constant and λ_R and λ_L are the right and left motors tilt angle respectively.

Rotational dynamics equations

$$\begin{aligned}\dot{p} - \frac{I_{xz}}{I_{xx}}\dot{r} &= +\frac{I_{xz}}{I_{xx}}pq + \frac{I_{yy} - I_{zz}}{I_{xx}}qr \\ &\quad +\frac{\delta_R}{I_{xx}}\left(k_\tau \sin\lambda_R - k_T y_{m_R} \cos\lambda_R\right) \\ &\quad -\frac{\delta_L}{I_{xx}}\left(k_\tau \sin\lambda_L + k_T y_{m_L} \cos\lambda_L\right)\end{aligned} \tag{1.12}$$

$$\begin{aligned}\dot{q} &= \frac{I_{xz}}{I_{yy}}\left(p^2 - r^2\right) + \frac{I_{zz} - I_{xx}}{I_{yy}}pr \\ &\quad -\frac{1}{I_{yy}}\delta_L k_T z_{m_L} \sin\lambda_L \\ &\quad -\frac{1}{I_{yy}}\delta_R k_T z_{m_R} \sin\lambda_R\end{aligned} \tag{1.13}$$

$$\begin{aligned}\dot{r} - \frac{I_{xz}}{I_{zz}}\dot{p} &= -\frac{I_{xz}}{I_{zz}}rq + \frac{I_{xx} - I_{yy}}{I_{zz}}pq \\ &\quad +\frac{\delta_R}{I_{zz}}\left(k_\tau \cos\lambda_R + k_T y_{m_R} \sin\lambda_R\right) \\ &\quad +\frac{\delta_L}{I_{zz}}\left(k_T y_{m_L} \sin\lambda_L - k_\tau \cos\lambda_L\right)\end{aligned} \tag{1.14}$$

where $\{x_{m_L}, y_{m_R}, z_{m_R}\}$ and $\{x_{m_L}, y_{m_L}, z_{m_L}\}$ are the right and left rotor positions withing the body axes and k_τ is the motors torque constant.

Translational kinematics equations

$$\begin{aligned} u &= \dot{x}_e \cos\theta\cos\psi + \dot{y}_e \cos\theta\sin\psi - \dot{z}_e \sin\theta \qquad (1.15) \\ v &= \dot{x}_e \left(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi\right) \\ &\quad + \dot{y}_e \left(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi\right) + \dot{z}_e \sin\phi\cos\theta \qquad (1.16) \\ w &= \dot{x}_e \left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right) \\ &\quad + \dot{y}_e \left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right) + \dot{z}_e \cos\phi\cos\theta \qquad (1.17) \end{aligned}$$

Rotational kinematic equations (Euler angles)

$$\begin{aligned} p &= \dot{\phi} - \dot{\psi}\cdot\sin\theta \qquad (1.18) \\ q &= \dot{\theta}\cdot\cos\phi + \dot{\psi}\cdot\cos\theta\cdot\sin\phi \qquad (1.19) \\ r &= \dot{\psi}\cdot\cos\theta\cdot\cos\phi - \dot{\theta}\cdot\sin\phi \qquad (1.20) \end{aligned}$$

1.3 Multi-objective optimization brief introduction

In engineering problems, dealing with situations that require posing and solving an optimization program is a common issue. As an example, this thesis is focused on UAV's model identification and controller tuning. Both types of problem are aligned in the sense that they both require seeking several parameters values based either on experimental data or on the data generated by a dynamic model. These two tasks suit perfectly the optimization paradigm, since they pursue finding the best parameters to fit the available data. Hence, they both should be tackled from an optimization perspective.

Additionally, a single-objective optimization might fall short in real problems, which generally require the optimization of multiple objectives that include, in addition, physical constraints, operational constraints and non-convexity. Due to this fact, addressing these problems from the standpoint of classical optimization could be insufficient.

A schematic of the design methodology through optimization is presented in fig. 1.7. The column on the left shows the process followed when trying to achieve a single objective. This would be the case of data fitting by least squares, for example. On the contrary, the right column shows a way to address multi-objective problems. As the main difference one can see that, in this case, a selection stage is reached. As it will be presented later, this is because when trying to achieve several objectives simultaneously, normally there is no optimal solution for all of them at the same time. Hence the designer must choose the one that, in his opinion, fits best with the final purpose of the design.

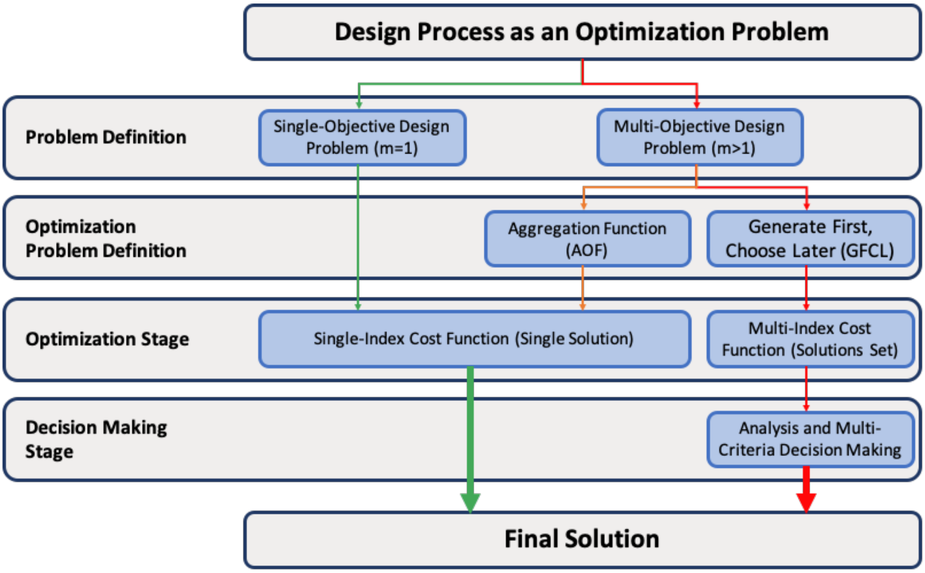

Design Process as an Optimization Problem
Problem Definition
Single-Objective Design Problem (m=1)
Multi-Objective Design Problem (m>1)
Optimization Problem Definition
Aggregation Function (AOF)
Generate First, Choose Later (GFCL)
Optimization Stage
Single-Index Cost Function (Single Solution)
Multi-Index Cost Function (Solutions Set)
Decision Making Stage
Analysis and Multi-Criteria Decision Making
Final Solution

Figure 1.7. Design methodology under the optimization perspective

As referred in [23], any Multi-Objective Optimization Problem (MOP) [1], can be stated as follows:

$$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x}) \quad = \quad [J_1(\boldsymbol{x}), \ldots, J_m(\boldsymbol{x})] \tag{1.21}$$

subject to

$$\boldsymbol{K}(\boldsymbol{x}) \quad \leq \quad 0 \tag{1.22}$$
$$\boldsymbol{L}(\boldsymbol{x}) \quad = \quad 0 \tag{1.23}$$
$$\underline{x_i} \leq x_i \quad \leq \quad \overline{x_i}, \quad \forall i = [1, \ldots, n] \tag{1.24}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ is defined as the decision vector; $\boldsymbol{J}(\boldsymbol{x})$ as the objective vector and $\boldsymbol{K}(\boldsymbol{x})$, $\boldsymbol{L}(\boldsymbol{x})$ as the inequality and equality constraint vectors respectively; $\underline{x_i}, \overline{x_i}$ are the lower and upper bounds in the decision (or search) space $\boldsymbol{X}$.

As mentioned before, in general, there are not unique solutions to MOPs, because the best solution for a single objective does not happen to be so good for the

[1] Any maximization problem can be converted to a minimization one. For each of the objectives that have to be maximized, the transformation: $\max J_i(\boldsymbol{x}) = -min(-J_i(\boldsymbol{x}))$ could be applied.

rest of the objectives at the same time. A set of solutions which represent the best performance combinations is more generally observed. That set is known as the Pareto set $\mathbf{\Theta}_P$. Each solution on the Pareto set generates a vector in the objective space and all those vectors define the Pareto front $\boldsymbol{J}_P$. The solutions in the Pareto set conform a group of Pareto optimal and non-dominated solutions. These concepts are all represented on Fig. 1.8.

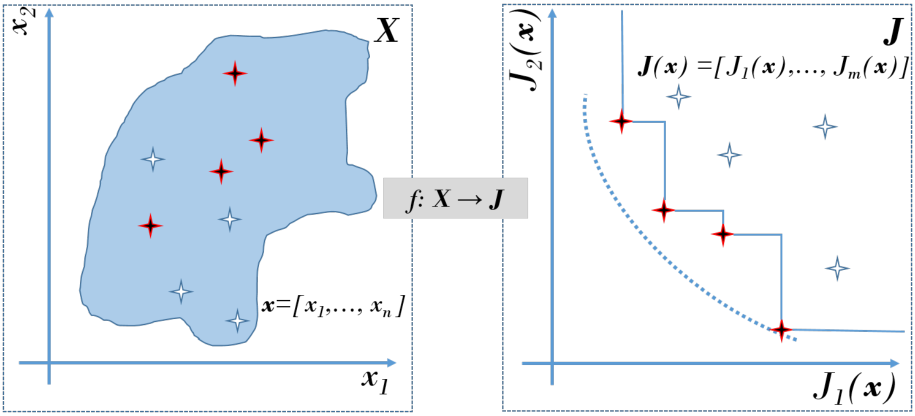

$\boldsymbol{x}_2$
$\boldsymbol{X}$
$\boldsymbol{x}$=[x_1,..., x_n]
$\boldsymbol{x}_1$
f: $\boldsymbol{X} \rightarrow \boldsymbol{J}$
$J_2(\boldsymbol{x})$
$\boldsymbol{J}$
$\boldsymbol{J}(\boldsymbol{x})$ =[$J_1(\boldsymbol{x})$,..., $J_m(\boldsymbol{x})$]
$J_1(\boldsymbol{x})$

Figure 1.8. Pareto optimality and dominance concepts. A Pareto front (dotted line in objective space $\boldsymbol{J}$) is approximated with a set solutions (depicted with stars) selected from the feasible decision space $\boldsymbol{X}$. Dark solutions are non-dominated solutions in the set and therefore, they are used to build a Pareto front approximation (solid line). Remainder solutions are dominated solutions.

Definition 1. *(Pareto optimality [23]): An objective vector $\boldsymbol{J}(\boldsymbol{x}^1)$ is Pareto optimal if there is not another objective vector $\boldsymbol{J}(\boldsymbol{x}^2)$ such that $J_i(\boldsymbol{x}^2) \leq J_i(\boldsymbol{x}^1)$ for all $i \in [1, 2, \dots, m]$ and $J_j(\boldsymbol{x}^2) < J_j(\boldsymbol{x}^1)$ for at least one j, $j \in [1, 2, \dots, m]$.*

Definition 2. *(Dominance [9]): An objective vector $\boldsymbol{J}(\boldsymbol{x}^1)$ is dominated by another objective vector $\boldsymbol{J}(\boldsymbol{x}^2)$ iff $J_i(\boldsymbol{x}^2) \leq J_i(\boldsymbol{x}^1)$ for all $i \in [1, 2, \dots, m]$ and $J_j(\boldsymbol{x}^2) < J_j(\boldsymbol{x}^1)$ for at least one j, $j \in [1, 2, \dots, m]$. This is denoted as $\boldsymbol{J(x_2)} \preceq \boldsymbol{J(x_1)}$.*

It is important to notice that the Pareto front is usually unknown, and the designer can only rely on a Pareto front approximation $\boldsymbol{J}_P^*$ and Pareto set approximation $\mathbf{\Theta}_P^*$.

In order to successfully embed the multi-objective optimization concept into a design process, three fundamental steps are required: the MOP statement (measure); the multi-objective optimization (MOO) process (search); and the multi-criteria decision making (MCDM) step (multicriteria analysis). This procedure will be named Multi-objective Optimization Design (MOOD) procedure (Fig. 1.9). The technique must be viewed as a holistic process in which equal importance is assigned to each stage so that the design process is successfully driven [7].

- **MOP statement**. It is probably the most challenging phase. In an optimization problem, solutions, objectives, restrictions and preferences must be expressed into mathematical terms that relate them, expressing the goal of the design problem correctly. This is challenging because how to translate qualitative objectives into quantitative terms is not always obvious. It is crucial that the mathematical expressions posed on the MOP represent adequately the actual objectives of the design, otherwise there will not be match between what is mathematically solved and the expected solution to the original design problem.

- **MOO process**. This stage is where the MOP stated in the previous stage is solved. Depending on its features (number of decision variables, number of objectives, convexity, complexity), one or other optimization algorithm should be chosen. Particularly on the research differential evolution algorithms have been used. Section 1.3.2 gives further details on how this algorithms work.

- **Multi-criteria decision making**. This stage is what differentiates GFCL MOOD from single objective design or AOF multi-objective design (see Fig. 1.7). Since there is not a single solution in MOPs a compromise solution should chosen. The designer has to carefully analyze the outcome of the MOO process and decide which solution its preferable for his/her goals.

Hence, objective and decision spaces and their constraints must be well defined in the MOP definition stage so that the correct problem is optimized in the optimization process. Finally, a deep analysis should be carried out (once an approximation to the Pareto front is available) to detect the most convenient solution in the decision making stage. This same topology is followed in the wind estimation process.

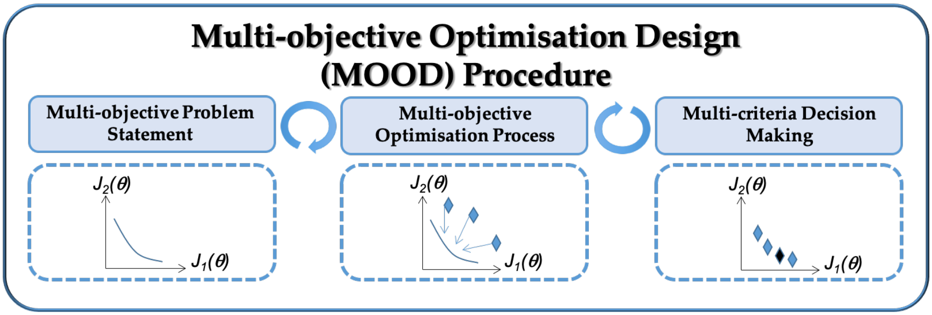

Multi-objective Optimisation Design (MOOD) Procedure
Multi-objective Problem Statement
Multi-objective Optimisation Process
Multi-criteria Decision Making
$J_2(\theta)$
$J_1(\theta)$
$J_2(\theta)$
$J_1(\theta)$
$J_2(\theta)$
$J_1(\theta)$

Figure 1.9. Multi-objective Optimization Design (MOOD) procedure. Stage 1, MOP statement: a Pareto front exists but it is unknown by the designer; Stage 2, MOO process: MOP solutions are pushed towards the real Pareto set, getting to the best Pareto front approximation possible; Stage 3, MCDM: solutions on the Pareto set approximation from previous stage are tested against some other selection criteria and the preferred solution is chosen.

Multi-objective techniques applied to model identification have achieved great results in many cases, as shown in ([27], [13] and [38]). Also, MOOD procedures have shown to be a valuable tool for controller tuning applications (see [25, 21], and references therein). Such techniques have been used in different controller structures [20]; for example PID controllers [36], fractional order controllers [29, 39] and state space controllers [11]. They enable the designer or decision maker (DM) to have a close embedment into the design process; since it is possible to take into account each design objective individually, they also enable comparing design alternatives, in order to select a controller fulfilling the expected trade-off among conflicting objectives. Such procedures have been used with success when (1) it is difficult to find a reasonable trade-off for parameter tuning problems fulfilling several requirements; and (2) it is worthwhile analyzing design objectives exchange among design alternatives.

1.3.1 Genetic Algorithms

When the optimization problem turns out to be non-convex, there exist solutions in the Pareto set that remain unreachable for a more classic optimization methods. Figure 1.10 shows a non-convex Pareto front and how the straight lines resulting from the different combination of objectives weights are unable to reach part of the Pareto front. Thus, optimizing objectives separately present a great advantage when non-convex problems must be solved.

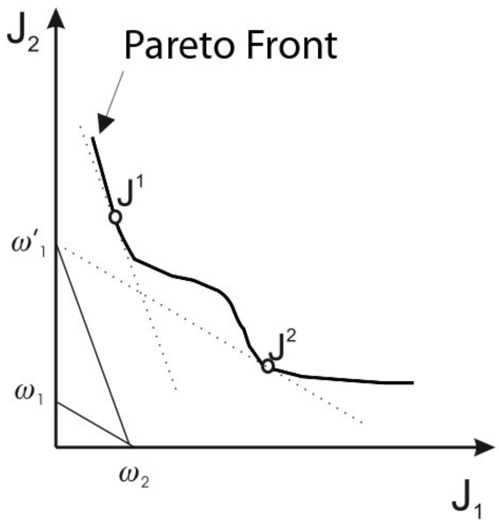


Figure 1.10. Pareto front approximation by weighted-sum technique. w_1 and w'_1 are two different weighting coefficients for index J_1 and w_2 is index J_2 weighting factor.

Genetic algorithms (GAs) are a subset of evolutionary algorithms (EAs) based on the rules of natural selection of species. In this type of routines, a set of possible solutions is called population, where each member represents a solution candidate for the function to be optimized. A population is nothing more than a set of points in the search space and each individual a point in that space by means of his chromosomes. Thus, the chances of survival of each individual are related to their "physical fitness", which in turn has a direct relationship with the value of the cost function for that point.

The evolution mechanism of individuals is achieved by genetic operators. The usual operators are:

- **Selection.** Its main goal consists on selecting the chromosomes that will integrate the next population by taking into account the cost function value for each individual[14].
- **Crossover.** The generation and integration of a new individual by combining the chromosomes of two individuals[14].
- **Mutation.** Random variation of some parts of the chromosome of an individual in the population to generate new individuals[14].

Figure 1.11 illustrates how the GAs work conceptually. An initial population (initial set of solutions) is stated and the cost functions are evaluated with it (population fitness). If the end condition is not satisfied, then a new population is generated from the previous one and by means of the genetic operators (evolution). Again, the cost functions are evaluated. This process is repeated until the end conditions is achieved. If the end condition is satisfied, the process ends and the last populations is the solution to the MOP.

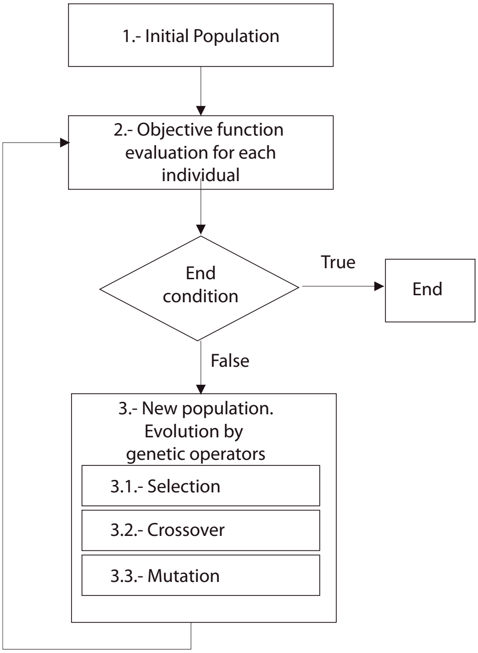

1.- Initial Population
2.- Objective function evaluation for each individual
End condition
True
End
False
3.- New population. Evolution by genetic operators
3.1.- Selection
3.2.- Crossover
3.3.- Mutation

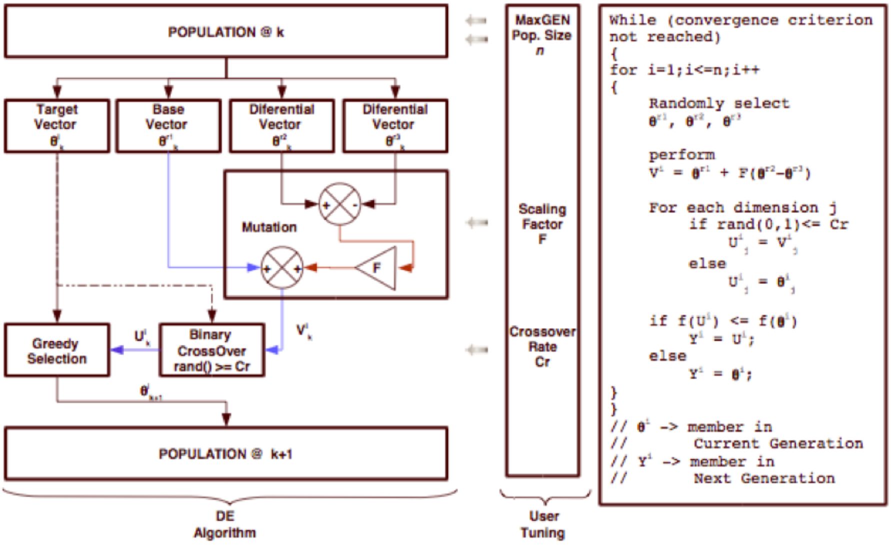
Figure 1.11. Genetic algorithms flowchart.

The different possible variations of GAs can be distinguished by the chromosome codification and the genetic operators used. One of this variations is the differential evolution algorithms that will be explained in the following section. Due to their nature, GAs are able to deal with non-convex and highly restrictive optimization problems [24]. GAs have demonstrated very good performance as global optimizers in many types of applications [22, 4, 6].

1.3.2 Differential Evolution Algorithms

Another type of evolutionary algorithm that has gained in popularity due to its simplicity and its powerful capabilities when converging into global solutions is the differential evolution (DE) algorithm [31]. It is an evolutionary floating-point algorithm that mainly contains 3 adjustment parameters, as it will be seen below.

There are many versions for the differential evolution algorithm. The standard version will be used to explain the algorithm itself. This version is known as the DE/rand/1/bin strategy [31]. The evolution process uses three operators: Mutation, Crossing and Selection (see Fig. 1.12). Thus, the individuals of an initial population (generation 1), cross each other and even mutate, to evolve in new generations thanks to the selection operation.



```
While (convergence criterion
not reached)
{
for i=1;i<=n;i++
{
    Randomly select
    $\theta^{r1}$, $\theta^{r2}$, $\theta^{r3}$

    perform
    $V^i = \theta^{r1} + F(\theta^{r2} - \theta^{r3})$

    For each dimension j
        if rand(0,1)<= Cr
            $U^i_j = V^i_j$
        else
            $U^i_j = \theta^i_j$

    if $f(U^i) <= f(\theta^i)$
        $Y^i = U^i$;
    else
        $Y^i = \theta^i$;
}
}
// $\theta^i$ -> member in
//          Current Generation
// $Y^i$ -> member in
//          Next Generation
```

Figure 1.12. Differential Evolution Algorithm

Where n is the number of decision variables, N_p is the size of the initial population, k is the generation (number of iteration) and θ^i_k a vector such that $\theta^i \in R^n$ and $i \in \{1, 2, 3, ..., N_p\}$ in generation k.

The DE algorithm has obtained great results in a wide variety of numerical optimization problems, many of them within the identification of systems. Within the world of academic problems, the DE algorithms have positioned themselves in the first places of the competitions organized by the Congress on Evolutionary Computing (CEC) of the IEEE. Specifically, in [38] an identification of parameters of a model in first principles for an Electro-Hydraulic Servo system [24] is presented.

sp-MODE algorithm

In chapters 2, 3 and 4, the spherical pruning Multi-objective Optimization Differential Evolutive Algorithm (sp-MODE) is the algorithm chosen to carry out the optimization process. The main characteristics for which it has been chosen are the ones shown below:

- It uses a Differential Evolution algorithm, which is stochastic and has real coding.
- It uses an initial population $P(0)$ to explore the search space, avoiding convergence in sub-optimal spaces.
- The best solutions are stored, to ensure that quality solutions will not be lost during the evolution process.
- The best solutions participate in the evolution process to increase the speed of convergence of the algorithm.
- Instead of using a concept of ϵ-dominance [17] or other related approach to achieve a uniformly distributed solution set, it uses a spherical pruning technique in the target space to reduce the cardinality of θ_P^*.
- The Spherical Selection helps to overcome the problems related to ϵ-dominance, where non-managed solutions could be lost in the evolutionary process [12].

In addition to all this, the algorithm has the great advantage of being available in a free *MATLAB* toolbox. Thus, the user has access to the code, being able to modify both a series of adjustable parameters that characterize the optimization process, and, obviously, the cost functions to be minimized. A set of guidelines for parameter adjustment in Differential Evolutionary algorithms are found in [31].

sp-MODEII algorithm

As an evolution to the sp-MODE, sp-MODEII[2] algorithm [25] is used in chapter 5. Its main characteristics are useful in order to deal with many-objective optimization statements, covering the basic properties of convergence, diversity and pertinency of the Pareto front. Such characteristics are:

- It uses Differential Evolution [30] algorithm as optimization engine, which has shown a good trade off between global search and convergence to the Pareto front [10].
- The objective space is partitioned in spherical sectors [26], in order to improve the spreading along the Pareto front.
- Preferences $\mathfrak{P}$ can be coded *a priori* in the form of a preference matrix $\mathfrak{m}$ by the designer. For each design objective $J_i(\boldsymbol{x})$, $i \in [1, \cdots, m]$ six values $(J_i^0, \cdots, J_i^5)$ must be stated (in the original units for each design objective)

[2] Scripts and tutorials available at www.mathworks.com/matlabcentral/fileexchange/authors/289050.

in order to define 5 preference ranges: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

- Preferability function $\mathfrak{f}(\cdot)$ is computed using the above commented matrix $\mathfrak{m}$. With such preference matrix, the algorithm computes a global physical programming index (GPP) [5] to evaluate the preferability of one solution over another solution, which is a modified form of the physical programming methods [19]. Such index is used to prune the approximated Pareto front, in order to get a compact and pertinent approximation, with the number of desired solutions imposed by the designer.

- This approach enables to state a difference between design objectives for decision making and for optimization. In the former case, they represent the design objectives where the DM is willing to perform a decision making and where the objective space is partitioned; the latter, are design objectives that are used in the optimization stage, used to calculate the GPP index, but are not used *a priori* in the design objective partitioning. This feature of the algorithm is further explained in chapter 5.

1.4 Thesis content and structure

This document includes the literal transcripts of five publications that expose results obtained as part of the PhD studies that culminate with the current dissertation. They are structured in two blocks as follows.

The first block, Modelling and Parameters Identification, covers the first two publications. They compile the efforts on developing first principles models for MAVs. In particular, chapter 3 provides insights on how to improve model parameterization when there is no real airspeed data to be used.

The second block, Control, Simulation and Test, integrates the other three publications in three different chapters. It is focused on using models to design and test control strategies before flight on unmanned aerial vehicles. Chapter 4 presents a comparative study between classic PID control techniques, and other model based structures where, the power of taking a multi-objective design approach will show advantages on reducing uncertainties. With a typical control structure on this type of systems, chapter 5 addresses the parameterization of controllers by means of multi-objective optimization. The approach on the research treats the problematic of hoping from simulation to real experimentation, with the intention of limiting the consequences of uncertainty when implementing a new control strategy. The main concept behind this approach is that, while passing through the Multi-Objective Optimization Design procedure, the new designs and the designer enriches of a deeper knowledge of the problem and, hence, of the objectives that should be posed to the optimization program. The final results is a set of controllers that can be safely implemented on the real system. Chapter 6 presents a new UAV concept. It is an innovative configuration that confers a fix-wing aircraft the capability of vertical take-off and landing. In this publication, modeling, control design and HIL simulation are integrated altogether to stabilize the platform for the take-off maneuver. The success of the design is shown by experimentation data gathered on a real flight test.

It must be remarked again that chapters 2 to 6 of this dissertation are literal transcripts of 5 research publications. For this reason redundant information and discrepancies on nomenclature will be found among the chapters. Those five publications are listed here:

- J. Velasco, S. García-Nieto. Unmanned Aerial Vehicles Model Identification using Multi-Objective Optimization Techniques. 19th World Congress of The International Federation of Automatic Control. Cape Town, South Africa. August 24-29, 2014. IFAC Proceedings Volumes, vol. 47, n.o 3, pp. 8837-8842, 2014.
  https://doi.org/10.3182/20140824-6-ZA-1003.02023

- J. Velasco-Carrau, S. García-Nieto, J. V. Salcedo, and R. H. Bishop. Multi-Objective Optimization for Wind Estimation and Aircraft Model Identification. Journal of Guidance, Control, and Dynamics, vol. 39, n^o 2, pp. 372-389, feb. 2016.
  https://doi.org/10.2514/1.G001294

- J. Velasco, S. García-Nieto, R. Simarro, J. Sanchis. Control Strategies for Unmanned Aerial Vehicles under Parametric Uncertainty and Disturbances: a Comparative Study. IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles June 10-12, 2015. Seville, Spain. IFAC-PapersOnLine, vol. 48, n.o 9, pp. 1-6, 2015.
  https://doi.org/10.1016/j.ifacol.2015.08.050

- J. Velasco-Carrau, G. Reynoso-Meza, S. García-Nieto, X. Blasco Ferragud. Enhancing controller's tuning reliability with multi-objective optimisation: from Model in the loop to Hardware in the loop. Engineering Applications of Artificial Intelligence, vol. 64, pp. 52-66, sep. 2017.
  https://doi.org/10.1016/j.engappai.2017.05.005

- Sergio García-Nieto, Jesús Velasco-Carrau, Federico Paredes-Valles, José Vicente Salcedo and Raúl Simarro. Motion Equations and Attitude Control in the Vertical Flight of a VTOL Bi-Rotor UAV. Electronics, vol. 8, n.o 2, p. 208, feb. 2019.
  https://doi.org/10.3390/electronics8020208

Bibliography

[1] A. Arévalo. Vehículos aéreos no tripulados, descripción y capacidades para la obtención de información.

[2] José Luis Asensio, Fernando Pérez, and Paola Morán. UAVs Beneficios y límites.

[3] A. Barrientos, J. del Cerro, P. Gutiérrez, R. San Martín, A. Martínez, and C. Rossi. Vehículos aéreos no tripulados para uso civil. Tecnología y aplicaciones. *Grupo de Robótica y Cibernética, Universidad Politécnica de Madrid.*

[4] Francesc Xavier Blasco. *Model based predictive control using heuristic optimization techniques. Application to non-linear and multivariables proceses (In Spanish).* PhD thesis, Universidad Politécnica de Valencia, Valencia, Spain, 1999.

[5] Xavier Blasco, Sergio García-Nieto, and Gilberto Reynoso-Meza. Autonomous trajectory control of a quadricopter vehicle. simulation and evaluation. *Revista Iberoamericana de Automática e Informática Industrial*, 9(2):194 – 199, 2012.

[6] Xavier Blasco, Miguel Martinez, Juan Senent, and Javier Sanchis. Generalized predictive control using genetic algorithms (GAGPC). An application to control of a non-linear process with model uncertainty. In Jaime G. Carbonell, Jörg Siekmann, G. Goos, J. Hartmanis, J. van Leeuwen, José Mira, Angel Pasqual del Pobil, and Moonis Ali, editors, *Methodology and Tools in Knowledge-Based Systems*, volume 1415, pages 428–437. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. Series Title: Lecture Notes in Computer Science.

[7] P.P. Bonissone, R. Subbu, and J. Lizzi. Multicriteria decision making (mcdm): a framework for research and applications. *Computational Intelligence Magazine, IEEE*, 4(3):48–61, August 2009.

[8] Chinchul Choi and Wootaik Lee. Analysis and compensation of time delay effects in hardware-in-the-loop simulation for automotive pmsm drive system. *Industrial Electronics, IEEE Transactions on*, 59(9):3403–3410, Sept 2012.

[9] Carlos A. Coello Coello and Gary B. Lamont. *Applications of Multi-Objective evolutionary algorithms.* World scientific publishing, advances in natural computation vol. 1 edition, 2004.

[10] S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, PP(99):1 –28, 2010.

[11] Kaveh Hassani and Won-Sook Lee. Multi-objective design of state feedback controllers using reinforced quantum-behaved particle swarm optimization. *Applied Soft Computing*, 41:66 – 76, 2016.

[12] Alfredo G. Hernández-Díaz, Luis V. Santana-Quintero, Carlos A. Coello Coello, and Julián Molina. Pareto-adaptive ϵ-dominance. *Evolutionary Computation*, 15(4):493–517, 2007.

[13] JM Herrero, X Blasco, M Martínez, C Ramos, and J Sanchis. Non-linear robust identification of a greenhouse model using multi-objective evolutionary algorithms. *Biosystems Engineering*, 98(3):335–346, 2007.

[14] Juan Manuel Herrero, Frances Xavier Blasco, Miguel Martínez, and Javier Sanchis. Identification of Continuous Processes Parameters Using Genetic Algorithms. In *Proceedings of the 10th Mediterranean Conference on Control and Automation*, volume 2, Lisbon, Portugal, July 2002.

[15] S.W. Jeon and Seul Jung. Hardware-in-the-loop simulation for the reaction control system using pwm-based limit cycle analysis. *Control Systems Technology, IEEE Transactions on*, 20(2):538–545, March 2012.

[16] Vladislav Klein and Eugene A. Morelli. *Aircraft system identification: theory and practice*, chapter 3, pages 27–75. AIAA Education Series. American Institute of Aeronautics and Astronautics, Inc., 2006.

[17] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation*, 10(3):263–282, 2002.

[18] Bin Lu, Xin Wu, H. Figueroa, and A. Monti. A low-cost real-time hardware-in-the-loop testing approach of power electronics controls. *Industrial Electronics, IEEE Transactions on*, 54(2):919–931, April 2007.

[19] Achille Messac. Physical programming-effective optimization for computational design. *AIAA journal*, 34(1):149–158, 1996.

[20] Gilberto Reynoso Meza, Xavier Blasco Ferragud, Javier Sanchis Saez, and Juan Manuel Herrero Durá. *Controller tuning with evolutionary multiobjective optimization: A holistic multiobjective optimization design procedure*, volume 85. Springer, 2016.

[21] Gilberto Reynoso Meza, Xavier Blasco Ferragud, Javier Sanchis Saez, and Juan Manuel Herrero Durá. Background on multiobjective optimization for controller tuning. In *Controller Tuning with Evolutionary Multiobjective Optimization*, pages 23–58. Springer, 2017.

[22] Zbigniew Michalewicz. *Genetic algorithms + data structures = evolution programs.* Springer, Berlin; London, 2011. OCLC: 751526653.

[23] Kaisa M. Miettinen. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, 1998.

[24] Gilberto Reynoso-Meza. *Design, coding and implementation of a multiobjective optimization algorithm based on Differential Evolution with spherical pruning: applications for system identification and controller tuning.* PhD thesis, Universidad Politécnica de Valencia, Valencia, Spain, 2009.

[25] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Sergio García-Nieto. Multiobjective evolutionary algortihms for multivariable PI controller tuning. *Appied Soft Computing*, 24:341 – 362, 2014.

[26] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Miguel Martínez. Multiobjective design of continuous controllers using differential evolution and spherical pruning. In Cecilia Di Chio et Al., editor, *Applications of Evolutionary Computation, Part I*, volume LNCS 6024, pages 532–541. Springer-Verlag, 2010.

[27] K. Rodriguez-Vazquez and P. J. Fleming. Multi-objective genetic programming for nonlinear system identification. *Electronics Letters*, 34(9):930–931, April 1998.

[28] Tariq Samad and George Stewart. Perspectives on innovation in control systems technology: compatibility with industry practices. *Control Systems Technology, IEEE Transactions on*, 21(2):284–288, 2013.

[29] Helem Sabina Sánchez, Fabrizio Padula, Antonio Visioli, and Ramon Vilanova. Tuning rules for robust fopid controllers based on multi-objective optimization with fopdt models. *ISA transactions*, 66:344–361, 2017.

[30] Rainer Storn and Kenneth Price. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341 – 359, 1997.

[31] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[32] J. Velasco and S. García-Nieto. Unmanned aerial vehicles model identification using multi-objective optimization techniques. Cape Town, South Africa, August 2014.

[33] Jesús Velasco. Identificación de modelos dinámicos y ajuste de controladores basado en algoritmos evolutivos multiobjetivo. Master's thesis, Universidad Politècnica de València, Spain, September 2013.

[34] Jesús Velasco, Sergio García-Nieto, Gilberto Reynoso-Meza, and Javier Sanchis. Implementación de un sistema hardware-in-the-loop para la simulación en tiempo real de pilotos automáticos para UAVs. In *Actas de las XXXIV Jornadas de Automática*, October 2013.

[35] Jesus Velasco, Sergio García-Nieto Rodríguez, Gilberto Reynoso Meza, and Javier Sanchis Saez. Desarrollo y evaluación de una estación de control de tierra para vehículos aéreos no tripulados. In *Actas de las XXXIII Jornadas de Automática*, 2012.

[36] Zhu Wang, Qixin Su, and Xionglin Luo. A novel htd-cs based pid controller tuning method for time delay continuous systems with multi-objective and multi-constraint optimization. *Chemical Engineering Research and Design*, 115:98 – 106, 2016.

[37] Andrew White, Guoming George Zhu, and Jongeun Choi. Hardware-in-the-loop simulation of robust gain-scheduling control of port-fuel-injection processes. *Control Systems Technology, IEEE Transactions on*, 19(6):1433–1443, 2011.

[38] Hassan Yousefi, Heikki Handroos, and Azita Soleymani. Application of differential evolution in system identification of a servo-hydraulic system with a flexible load. *Mechatronics*, 18(9):513–528, November 2008.

[39] Abbas-Ali Zamani, Saeed Tavakoli, and Sadegh Etedali. Fractional order pid control design for semi-active control of smart base-isolated structures: A multi-objective cuckoo search approach. *ISA Transactions*, 67:222 – 232, 2017.

Part I

Modelling and Parameters Identification

Chapter 2

Unmanned Aerial Vehicles Model Identification using Multi-Objective Optimization Techniques [1]

Abstract

The total amount of UAVs civil applications is getting bigger and bigger. The cost and the risks of the development phase of this systems has to be decreased in order to make them affordable. It is required to minimize the number of hours of real flight, making use of simulation tools and taking full advantage of the acquired data. Thus, obtaining a dynamic model that tightly adjusts to the real flight behaviour of the aircraft gains in importance, in the way that it will lead to precise simulation results and, therefore, to correctly designed control algorithms. A model identification technique based on experimental data and Multi-Objective optimization evolution algorithm, is presented here. This methodology makes profit of the possibility given by this type of algorithm of facing different objectives at the same time, to take full advantage of the experimental data and to get better adjusted models.

[1] J. Velasco, S. García-Nieto. Unmanned Aerial Vehicles Model Identification using Multi-Objective Optimization Techniques. 19th World Congress of The International Federation of Automatic Control. Cape Town, South Africa. August 24-29, 2014. IFAC Proceedings Volumes, vol. 47, n.o 3, pp. 8837-8842, 2014. https://doi.org/10.3182/20140824-6-ZA-1003.02023

2.1 Introduction

There exists an increasingly popular variety of applications that justify the development of Unmanned Aerial Vehicles (UAVs) in the civil aviation field. Tasks such as photography for coastline control and beaches erosion tracing, fire detection and control [3], infrastructures inspection, or measurement for agriculture [9] are just some of the possible applications. In this new aeronautics field, a sufficiently low cost, which suits companies requirements is the main objective.

There are several fronts that have to be attended for the achievement of this aim. First, it is necessary to reduce the cost of the aircraft itself. This brings therefore, a completely new generation of tiny airplanes, which size is the minimum necessary to house propulsion, sensorization and control equipments.

Second, the integrated systems (sensors, actuators and control units), have to be powerful enough to control the fast dynamics of these vehicles and carry out, at the same time, the mission for which they have been purchased. The cost of such devices is becoming lower and lower, thanks to the fast evolution experienced by the computer technology in the last few years.

Finally, because of the characteristics of the product, the cost of the development phase has become an important percentage of the final price. It is required in this point to minimize the number of hours of real flight, making use of simulation tools and squeezing, as much as possible, already acquired data. Besides, it is in this phase in which the hardware integrity is in bigger danger. Thus, obtaining a dynamic model that tightly adjusts to the real flight behaviour of the aircraft seems vital, in the way that it will lead to precise simulation results and, therefore, to correctly designed control algorithms.

In this article, a methodology based on Multi-Objective (MO) optimization is presented and applied to a real system. To present this methodology, flight data are used instead of wind tunnel experiments, to identify the non-dimensional derivatives of stability and control of an UAV. Nevertheless, the technique exposed is not limited by the data source, but the other way around, it is enhanced with the quality and diversity of the performed experiments. As an example of this, despite experiments in this article are lacking measures relative to air, once available, they could be added in a straightforward manner to the identification process, with the obvious improvement of the obtained results, but without any change on the methodology itself. Thus, the presented identification technique allows the designer to test flight data from different types of experiment and, thereby, to obtain models with acceptable performances in several kinds of situation.

The article is divided in 5 sections. Section 2.2 introduces the aircraft and the hardware used in the experiments, along with the dynamic and aerodynamic models willing to be identified. In section 2.3 the methodology used in the identification will be shown and explained. Flight tests are presented in section 2.4. Every experiment has been performed twice so that a second set of values could be obtained for validating results. Results from the identification and the validation processes are also given in section 2.4. Finally, some conclusions are commented in section 2.5.

2.2 UAV testbench

2.2.1 Platform and Hardware

As the main component of the flight platform, a Kadett 2400 aircraft, manufactured by Graupner, may be found. It houses all necessary devices for its control, not only manually, but also in autonomous mode. During normal flight, 3 control surfaces are provided: tail rudder, elevators and ailerons. As the unit of propulsion a brushless engine of altern current is integrated, which is fed by two LIPO batteries through a frequency variator. Alike the servomotors, the variator is controlled by sending PWM signals as commanding signals.

There exists a device bridge between the manual and the autonomous states. The SSC (Servo Switch Controller) is able to perform the commutation between different sources of entrance. However, its principle property is allowing a computer to measure and also to introduce deflections in control surfaces (δ) and changes in the motor thrust (T).

As already stated, control actions are sent from the Flight Control Station (FCS), constituted by a PC-104. Is this unit the one housing the control algorithms, performing, therefore, all necessary tasks at each phase of the flight. The loop is closed by the IG500N unit. This all-in-one device, joins the efforts of a wide range of sensors, such as accelerometers, gyroscopes and magnetometers. Its Kalman filter is capable of mixing the information coming from those sensors in order to offer precise measurements of position, orientation, linear and angular speed, and acceleration, in the 3 aircraft body-axes. This same platform was presented in ([8], [7] and [6]) together with the results of the first flight tests. The figure 2.1 shows the hardware elements described here.

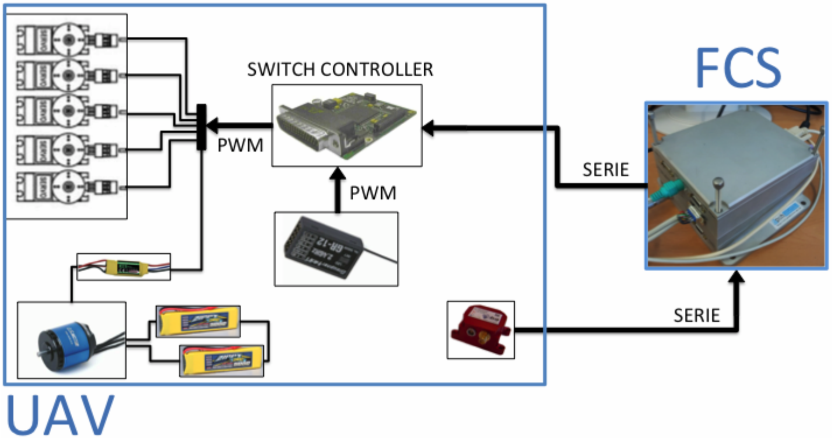


Figure 2.1. Interconnection between UAV devices and FCS

2.2.2 Aircraft Dynamic Model

Once the hardware platform has been introduced, a model that approximately explains the behaviour of the real process is going to be derived. In this case, the final model will be used for simulation and design of control algorithms for a UAV. This means obtaining the expressions that relate the input variables: deflection in the control surfaces and motor load; to a series of output signals: linear and angular velocities, accelerations and position in 3D space.

As expressed in [2], the seeking of these expressions normally begins from the linear and the angular momentum conservation principles, that can be expressed as:

$$\sum_{ext} \overrightarrow{F} = \frac{d}{dt}(m\overrightarrow{V}) \tag{2.1}$$

$$\sum_{ext} \overrightarrow{M} = \frac{d}{dt}(I\overrightarrow{\omega}) \tag{2.2}$$

Where $\sum_{ext} \overrightarrow{F}$ and $\sum_{ext} \overrightarrow{M}$ are the summ of external forces and moments respectively, m and I are the mass and the inertia tensor of the aircraft, and $\overrightarrow{V}$ and $\overrightarrow{\omega}$ are linear and angular velocity vectors. In particular 3 are the types of external forces that affect to the behaviour of the vehicle. They are: aerodynamic force (F_A), force applied by the motor (F_T) and the gravitational force (F_G). At the same time, only the aerodynamic force generates aerodynamic torque (M_A). Thereby, the equations 2.1 and 2.2 remain:

$$\overrightarrow{F_A} + \overrightarrow{F_T} + \overrightarrow{F_G} = m\dot{\overrightarrow{V}} + \overrightarrow{\omega} \times \overrightarrow{V} \tag{2.3}$$

$$\overrightarrow{M_A} = I\dot{\overrightarrow{\omega}} + \overrightarrow{\omega} \times I\overrightarrow{\omega} \tag{2.4}$$

The two previous equations are actually vectorial equations, so that, there is a total of 6 equations that correspond to the 6 degrees of freedom of a rigid body in the space. Deriving 2.3 and 2.4 the following expressions are obtained:

$$\begin{aligned} \overline{q}S\begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} + \begin{bmatrix} -g\sin\theta \\ g\sin\phi\cos\theta \\ g\cos\phi\cos\theta \end{bmatrix} + \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} \\ = m\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times m\begin{bmatrix} u \\ v \\ w \end{bmatrix} \end{aligned} \tag{2.5}$$

$$\begin{aligned} \overline{q}S\begin{bmatrix} bC_l \\ \overline{c}C_m \\ bC_n \end{bmatrix} = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{zx} & 0 & I_z \end{bmatrix}\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \\ + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{zx} & 0 & I_z \end{bmatrix}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \end{aligned} \tag{2.6}$$

Where u, v, and w are the components of the linear velocity of the aircraft in its body axes (x_b, y_b, z_b). In the same way, p, q, and r are the 3 components of the angular velocity. It is important to highlight the apparition in the equations 2.5 and 2.6 of the variables C_i, that represent the aerodynamic coefficients of each component of the resultant aerodynamic force (X, Y, and Z) and torque (L, N, and M). Such coefficients, are functions that relate those components to some of the system variables. The expressions that those coefficients adopt is of great interest in this work and therefore, they will be studied in further detail in section 2.2.3. Finally S, b, $\overline{c}$ are constructive constants of the airplane and $\overline{q}$ is the dynamic pressure of the air.

The aircraft orientation is usually denoted with the well known Euler angles of roll ϕ, pitch θ, and yaw ψ, wich express the rotation of a body from a global reference system to the body-axes. The kinematic equations that relate angular velocities of the aircraft to the Euler angles are:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\theta & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.7}$$

Finally, equations 2.8 to 2.16 are the result of reordering equations 2.5, 2.6, and 2.7, so that they can be directly used on the calculation of the simulation output values. Such values will be the same as those coming from the unit IG500N in real flights.

Force equations:

$$\dot{u} = rv - qw + \frac{\overline{q}S}{m}C_X(\delta) - g\sin\theta + \frac{T}{m} \tag{2.8}$$

$$\dot{v} = pw - ru + \frac{\overline{q}S}{m}C_Y(\delta) + g\cos\theta\sin\phi \tag{2.9}$$

$$\dot{w} = qu - pv + \frac{\overline{q}S}{m}C_Z(\delta) + g\cos\theta\cos\phi \tag{2.10}$$

Torque equations:

$$\dot{p} - \frac{I_{xz}}{I_x}\dot{r} = \frac{\overline{q}Sb}{I_x}C_l(\delta) - \frac{I_z - I_y}{I_x}qr + \frac{I_{xz}}{I_x}qp \tag{2.11}$$

$$\dot{q} = \frac{\overline{q}S\overline{c}}{I_y}C_m(\delta) - \frac{I_x - Iz}{I_y}pr - \frac{I_{xz}}{I_y}(p^2 - r^2) \tag{2.12}$$

$$\dot{r} - \frac{I_{xz}}{I_z}\dot{p} = \frac{\overline{q}Sb}{I_z}C_n(\delta) - \frac{I_y - I_x}{I_z}pq - \frac{I_{xz}}{I_z}qr \tag{2.13}$$

Kinematic equations:

$$\dot{\phi} = p + \tan\theta(q\sin\phi + r\cos\phi) \tag{2.14}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{2.15}$$

$$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta} \tag{2.16}$$

2.2.3 Aircraft Aerodynamic Model

It was said in 2.2.2 that aerodynamic forces and torques were related to some of the system variables through a series of functions that were called aerodynamic coefficients. In this section the structure used in the present article for those functions will be stated and, likewise, the parameters to be identified will be highlighted.

In [2] detailed information on how to proceed to obtain the dependencies of aerodynamic coefficients with other system variables is provided. First, if we assume a scenario in which the aircraft is generally in steady flight conditions, and it only performs short maneuvers that take it off from this state, we can truncate the Taylor series expansion to keep only the first or second order terms. Furthermore, under this assumption of small perturbations, and based on the symmetry of the vehicle, it can be assumed that 1) the symmetrical (longitudinal) variables u, w and q do not affect asymmetrical (lateral) force and torques Y, L and N; and similarly, 2) asymmetric (lateral) variables v, p and r do not affect the symmetrical (longitudinal) forces and torque X, Z and M.

Equations from (2.17) to (2.22) show the approximation of the aerodynamic equations that have been adopted for this article.

Longitudinal aerodynamic models:

$$\begin{aligned} C_D(t) =& \mathbf{C_{D_0}} + \mathbf{C_{D_V}}\frac{1}{V_0}\Delta V(t) + \mathbf{C_{D_\alpha}}\Delta\alpha(t) \\ &+ \mathbf{C_{D_{\alpha^2}}}\Delta\alpha(t)^2 + \mathbf{C_{D_q}}\frac{\bar{c}}{2V_0}q(t) + \mathbf{C_{D_{\delta_e}}}\Delta\delta_e(t) \end{aligned} \tag{2.17}$$

$$\begin{aligned} C_L(t) =& \mathbf{C_{L_0}} + \mathbf{C_{L_V}}\frac{1}{V_0}\Delta V(t) + \mathbf{C_{L_\alpha}}\Delta\alpha(t) \\ &+ \mathbf{C_{L_{\alpha^2}}}\Delta\alpha(t)^2 + \mathbf{C_{L_{\dot\alpha}}}\frac{\bar{c}}{2V_0}\dot\alpha(t) + \mathbf{C_{L_q}}\frac{\bar{c}}{2V_0}q(t) \\ &+ \mathbf{C_{L_{\delta_e}}}\Delta\delta_e(t) \end{aligned} \tag{2.18}$$

$$\begin{aligned} C_m(t) =& \mathbf{C_{m_0}} + \mathbf{C_{m_V}}\frac{1}{V_0}\Delta V(t) + \mathbf{C_{m_\alpha}}\Delta\alpha(t) \\ &+ \mathbf{C_{m_{\alpha^2}}}\Delta\alpha(t)^2 + \mathbf{C_{m_{\dot\alpha}}}\frac{\bar{c}}{2V_0}\dot\alpha(t) \\ &+ \mathbf{C_{m_q}}\frac{\bar{c}}{2V_0}q(t) + \mathbf{C_{m_{\delta_e}}}\Delta\delta_e(t) \end{aligned} \tag{2.19}$$

Lateral aerodynamic models:

$$\begin{aligned} C_Y(t) =& \mathbf{C_{Y_0}} + \mathbf{C_{Y_\beta}}\Delta\beta(t) + \mathbf{C_{Y_p}}\frac{b}{2V_0}p(t) \\ &+ \mathbf{C_{Y_r}}\frac{b}{2V_0}r(t) + \mathbf{C_{Y_{\delta_{al}}}}\Delta\delta_{al}(t) + \mathbf{C_{Y_{\delta_r}}}\Delta\delta_r(t) \end{aligned} \tag{2.20}$$

$$\begin{aligned} C_l(t) =& \mathbf{C_{l_0}} + \mathbf{C_{l_\beta}}\Delta\beta(t) + \mathbf{C_{l_p}}\frac{b}{2V_0}p(t) \\ &+ \mathbf{C_{l_r}}\frac{b}{2V_0}r(t) + \mathbf{C_{l_{\delta_{al}}}}\Delta\delta_{al}(t) + \mathbf{C_{l_{\delta_r}}}\Delta\delta_r(t) \end{aligned} \tag{2.21}$$

$$\begin{aligned} C_n(t) =& \mathbf{C_{n_0}} + \mathbf{C_{n_\beta}}\Delta\beta(t) + \mathbf{C_{n_p}}\frac{b}{2V_0}p(t) \\ &+ \mathbf{C_{n_r}}\frac{b}{2V_0}r(t) + \mathbf{C_{n_{\delta_{al}}}}\Delta\delta_{al}(t) + \mathbf{C_{n_{\delta_r}}}\Delta\delta_r(t) \end{aligned} \tag{2.22}$$

Where α and β are the angle of attack and of sideslip respectively and V is the airspeed. In particular, V_0 is the airspeed measured at the steady state of flight, before a maneuver begins. Theese variables are velocity dependent and they can be calculated as follows:

$$\alpha = \arctan\left(\frac{w}{u}\right) \qquad ; \qquad \beta = \arcsin\left(\frac{v}{V}\right) \tag{2.23}$$

$$V = |\overrightarrow{V}| = \sqrt{u^2 + v^2 + w^2} \tag{2.24}$$

Besides, C_L and C_D are the Lift and Drag coefficients and their relation with C_X and C_Z is:

$$C_L(t) = -\,C_Z(t)\cos(\alpha(t)) + C_X(t)\sin(\alpha(t)) \tag{2.25}$$

$$C_D(t) = -\,C_X(t)\cos(\alpha(t)) - C_Z(t)\sin(\alpha(t)) \tag{2.26}$$

Thus, the aerodynamic model identification is based on extracting the polynomial constants of the equations 2.17 to 2.22 (marked in bold) from the flight data and by means of the dynamic model. Those constants are called non-dimensional derivatives of stability and control.

2.3 Aerodynamic Model Identification

2.3.1 Previous calculations

It is easy to understand that there is no sensor capable of measuring aerodynamic coefficients directly. Thus, before performing the optimization it is necessary to calculate the actual value of the aerodynamic coefficients during the flight. The dynamic model equations will be used for that purpose.

Equations 2.27 to 2.32 describe the methodology used to obtain the values taken by the aerodynamic coefficients at the time instants in which measurements are available. These equations are easily deduced from the first principles model presented in 2.2.2 [2].

$$C_X(t) = \frac{1}{\overline{q}(t)S}(ma_x(t) - T(t)) \tag{2.27}$$

$$C_Y(t) = \frac{ma_y(t)}{\overline{q}(t)S} \tag{2.28}$$

$$C_Z(t) = \frac{ma_z(t)}{\overline{q}(t)S} \tag{2.29}$$

$$\begin{aligned} C_l(t) =& \frac{1}{\overline{q}(t)Sb}[I_x\dot{p}(t) - I_{xz}\left(p(t)q(t) + \dot{r}(t)\right) \\ &+ \left(I_z - I_y\right)q(t)r(t)] \end{aligned} \tag{2.30}$$

$$\begin{aligned} C_m(t) =& \frac{1}{\overline{q}(t)S\overline{c}}[I_y\dot{q}(t) + \left(I_x - I_z\right)p(t)r(t) \\ &+ I_{xz}\left(p(t)^2 - r(t)^2\right)] \end{aligned} \tag{2.31}$$

$$\begin{aligned} C_n(t) =& \frac{1}{\overline{q}(t)Sb}[I_z\dot{r}(t) - I_{xz}\left(\dot{p}(t) - q(t)r(t)\right) \\ &+ \left(I_y - I_x\right)p(t)q(t)] \end{aligned} \tag{2.32}$$

Refer to equations 2.25 and 2.26 for the calculation of $C_L(t)$ and $C_D(t)$ respectively.

2.3.2 Multi-Objective Optimization

In engineering problems, it is a common issue to deal with situations that require the optimization of multiple objectives that include, in addition, physical constraints, operational constraints and nonlinearities. Due to this fact, addressing these problems from the standpoint of classical optimization could be insufficient.

Any multi-objective optimization problem (MOP) can be stated as:

$$\min_{\theta \in \mathbb{R}} J(\theta) = [J_1(\theta), J_2(\theta), \dots, J_m(\theta)] \tag{2.33}$$

Where θ is the solution that minimizes the m cost functions J_i at the same time. Generally, it will not be possible to find a solution that satisfies all requirements at the same time, so the optimizer will have to provide the amount of solutions which are not improved by any other in all the objectives at the same time. That set of solutions is the Pareto set and their value in the objectives space is the Pareto front.

Multi-objective techniques applied to model identification have achieved great results in many cases, as shown in ([5], [1] and [10]).

In our case, elevators deflection and motor thrust variations, generate changes in longitudinal variables and, in the same way, ailerons and rudder deflections do likewise in lateral ones. Therefore, longitudinal and lateral coefficient models can be identified from different kind of experiments. As an example, if a C_D modeled is obtained by optimizing an elevators test, the model performance on a motor experiment data will be decreased, and vice versa. Therefore, an identification that takes both experiments in account at the same time, may be stated as a bi-objective optimization problem.

If the Mean Square Error (MSE) is used as performance index of the identification process, two cost functions can be defined for each aerodynamic coefficient. Equations 2.34 and 2.35 are the two cost functions to minimize for obtaining any of the longitudinal models.

$$J_1 = \frac{1}{N_{elevator}} \sum_{i=1}^{N_{elevator}} \left(C_j(t_i) - \hat{C}_j(t_i)\right)^2 \forall j \in \{D, L, m\} \tag{2.34}$$

$$J_2 = \frac{1}{N_{motor}} \sum_{i=1}^{N_{motor}} \left(C_j(t_i) - \hat{C}_j(t_i)\right)^2 \forall j \in \{D, L, m\} \tag{2.35}$$

where $\hat{C}_j(t_i)$ is the model approximation of the C_j value at the instant t_i and $N_{elevator}$ and N_{motor} are the number of samples of each kind of experiment. Similar cost functions can be defined for the three lateral models.

Then, if, $C_l(t)$ is to be modeled by using one ailerons experiment and one rudder experiment, the identification problem from this MO point of view should be stated as:

$$\begin{aligned} \min_{\theta \in \mathbb{R}^6} \quad & \left[\tfrac{1}{N_{ailerons}} \textstyle\sum_{i=1}^{N_{ailerons}} \left(C_l(t_i) - \hat{C}_l(t_i, \theta)\right)^2, \right. \\ & \left. \tfrac{1}{N_{rudder}} \textstyle\sum_{i=1}^{N_{rudder}} \left(C_l(t_i) - \hat{C}_l(t_i, \theta)\right)^2 \right] \\ : \theta = \quad & \left[C_{l_0}, C_{l_\beta}, C_{l_p}, C_{l_r}, C_{l_{\delta_{al}}}, C_{l_{\delta_r}} \right] \end{aligned} \tag{2.36}$$

To solve the MOP stated above, any Multi-Objective optimizer can be used. In this work, the sp-MODE[2] algorithm has been chosen [4].

[2] Available in http://www.mathworks.es/matlabcentral/ fileexchange/39215-multi-objective-differential-evolution-algorithm-with-spherical-pruning

2.4 Results

2.4.1 Flight Tests

In section 2.2.3 short maneuvers from a steady state flight are mentioned. In aeronautics, an airplane in steady flight is an aircraft which is maintaining constant heading and altitude, at a constant speed also and with leveled wings orientation (zero roll angle). At that point, the pilot does not need to make any correction on control surfaces or motor to maintain this steady flight.

In order to obtain data that can be employed in adjusting the aerodynamic parameters, the designed experiments simulate such short maneuvers. Thus, starting always at a steady state flight, each system input has been excited separately and, after that excitation, the aircraft has been left to evolve naturally, until the pilot deemed it appropriate and safe to recover the aircraft. Each experiment has been performed twice, in order to count with different data sets for the adjustment and the validation. It should be noted finally, that in the absence of any sensor capable of measuring airspeed, all maneuvers described below have been carried out against the wind. This restriction was imposed to the pilot in order to reduce variability between tests.

The flight plan provided to the pilot before beginning the experiments was:

1. Stable flight:
   (a) Adjust ailerons and rudder. Leveled wings.
   (b) Set the motor load around 50%.
   (c) Adjust elevators until the altitude gets constant without touching the control stick.
2. Elevators up and down trying to copy a positive plus negative step sequence.
3. Repeat step 1.
4. Ailerons side to side in the appropriate frequency to avoid extreme rotations. First in one direction and then in the opposite one.
5. Repeat step 1.
6. Tail rudder side to side. First in one direction and then in the oposite one.
7. Repeat step 1.

8. Positive and negative steps in motor load. Sequence: 50%-100%-50%-0%-50%

9. Repeat the whole process a second time.

Figures 2.2 and 2.3 show the evolution of the so called longitudinal and lateral variables during an elevators and ailerons excitation test respectively. As it can be seen, when a longitudinal input is excited, the rest of the longitudinal variables are excited too, which finally produces variations in the symmetrical aerodynamic coefficients. This same behaviour can be observed for the asymmetrical variables. All these variations can be collected and used to calculate the aerodynamic derivatives of stability and control.

2.4.2 Optimization Results

Figure 2.4 shows the lateral models Pareto front found by the algorithm after the programmed optimization. As supposed, the better an experiment is fitted by a model, the greater error it gets for a second test. That is why the person in charge of identifying the aircraft model cannot be satisfied by using just one test, but should face the model to different experiments data.

Besides, confronting experiments in a multi-objective optimization, instead of using all of them as one in a mono-objective minimization, gives the main following advantages:

- Using multi-objective optimization involves the selection of a solution among others, which gives the designer the power of defining the importance of each experiment basing that definition on his requirements.

- The resultant Pareto front shows how good the different models are for each experiment. Thanks to that, the designer may get an idea of how good the collected data is and thereby, decide which are the requirements that he should ask for to the final model.

- It is possible to add as many objectives as wanted in the identification process. This means that confronting tests of the same kind is also possible, what could be a good practice for variability reducing.

For example, if the C_l coefficient is taken into account, it can be observed that the ailerons test gets a much better approximation than the rudder one. This fact, which can be deduced from the values that the square error takes, is also logical, since the ailerons are precisely thought to introduce a moment in the X axis. In this case, the designer should probably prefer models which fit better this type of experiments over the ones that do a better job with rudder tests.

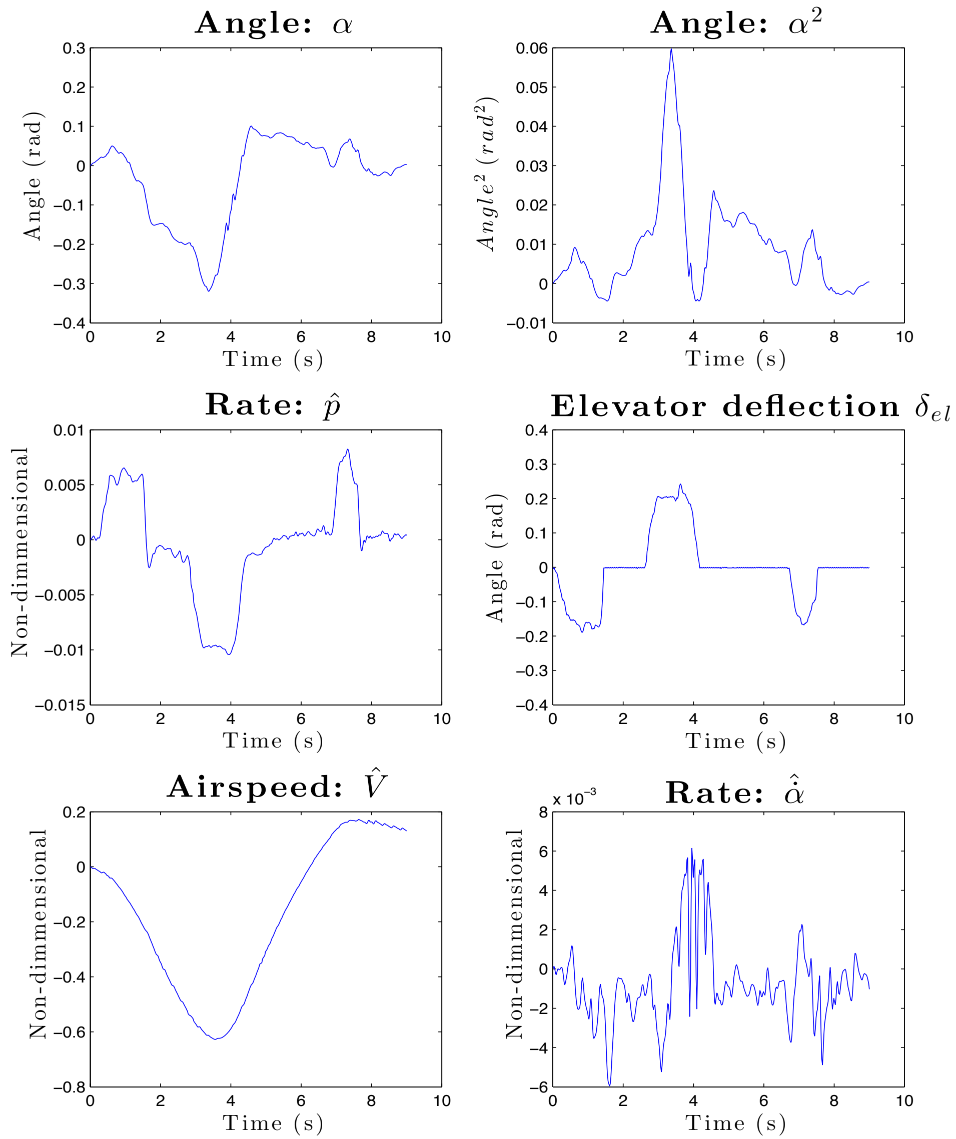


Figure 2.2. Flight Test: longitudinal variables evolution in an elevators test

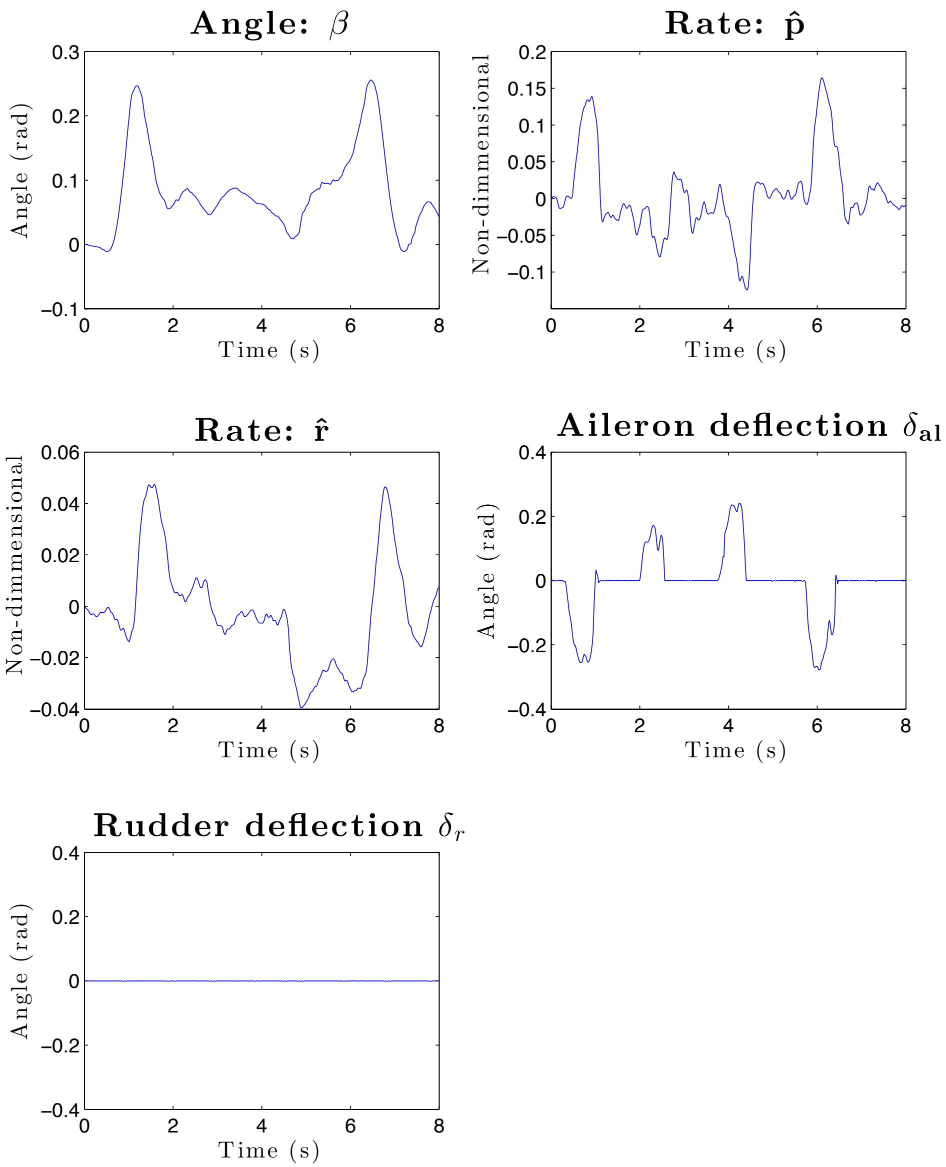


Figure 2.3. Flight Test: lateral variables evolution in an ailerons test

The green point on figure 2.4 is the elected model for each of the lateral aerodynamic coefficients. It represents a solution of compromise between a situation in which the ailerons deflection is modified and a situation in which that modification is suffered by the tail rudder. This fact can be checked in figure 2.5. That graph shows the approximation to C_l (calculated with (2.30)) given by the chosen model for validation data. Two more models, identified by using the classical least squares technique, are also included in figure 2.5. Those two models represent the best approximation, in terms of MSE, for the ailerons and the elevators experiments separately (see [6]). As can be deduced from the figure, the MO solution (green curve) represents a good intermediate approximation in both situations.

2.5 Conclusion

A methodology for the identification of UAVs aerodynamic models has been presented. Besides a demonstration of its application in a real system has been carried out with satisfying results. The technique presented gives the already stated advantages in the data analysis and the identification process, since it involves a phase of decision by the designer. This phase allows then, the study of several models and the election of the one that fits better with the designer needs. In addition, confronting experiments offers information about the difficulties of finding a model that fits different flight conditions at the same time, which improves the understanding of the system. This technique may also give information about the importance of a particular kind of experiment in the identification of the model. All these advantages lead to better models that may save time and money when designing autonomous aircraft control algorithms.

Bibliography

[1] JM Herrero, X Blasco, M Martínez, C Ramos, and J Sanchis. Non-linear robust identification of a greenhouse model using multi-objective evolutionary algorithms. *Biosystems Engineering*, 98(3):335–346, 2007.

[2] Vladislav Klein and Eugene A. Morelli. *Aircraft system identification: theory and practice.* American Institute of Aeronautics and Astronautics Reston, VA, USA, 2006.

[3] Wolfgang Krüll, Robert Tobera, Ingolf Willms, Helmut Essen, and Nora von Wahl. Early forest fire detection and verification using optical smoke, gas and microwave sensors. *Procedia Engineering*, 45:584–594, 2012.

[4] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Miguel Martínez. Design of continuous controllers using a multiobjective differential evolution

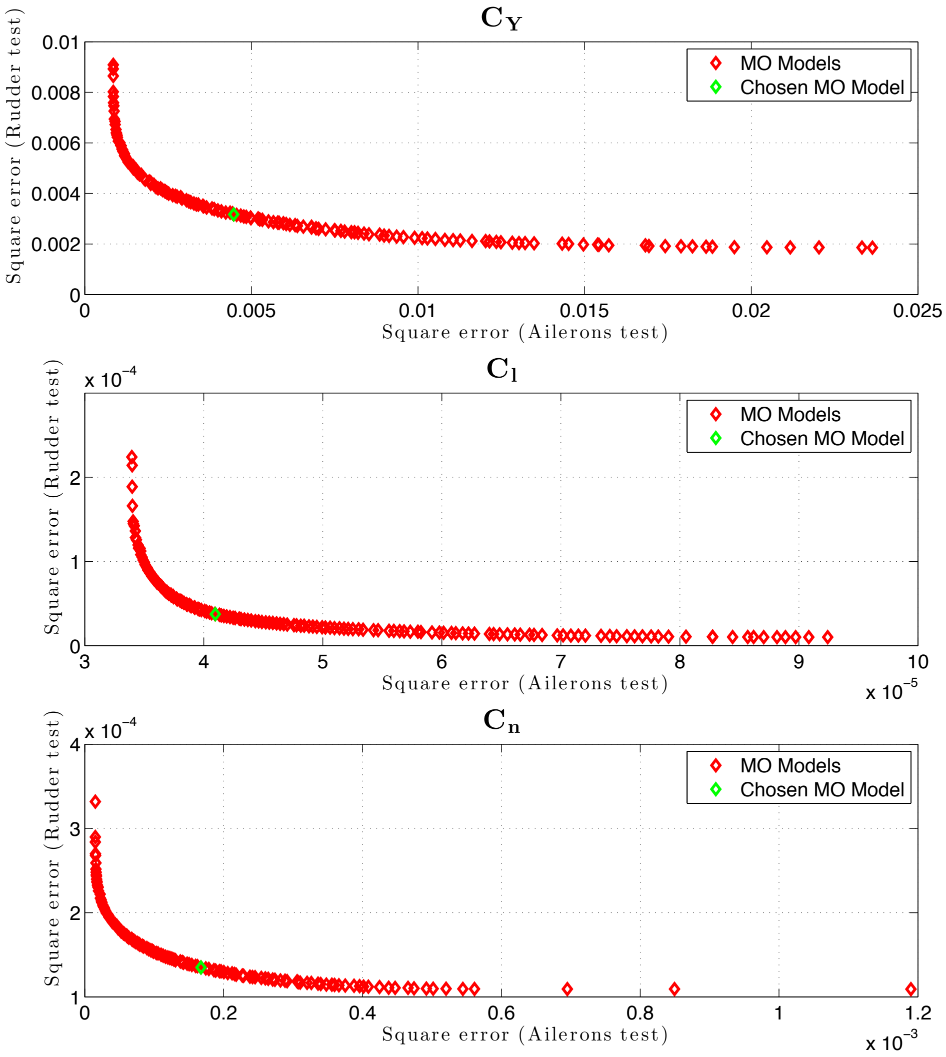
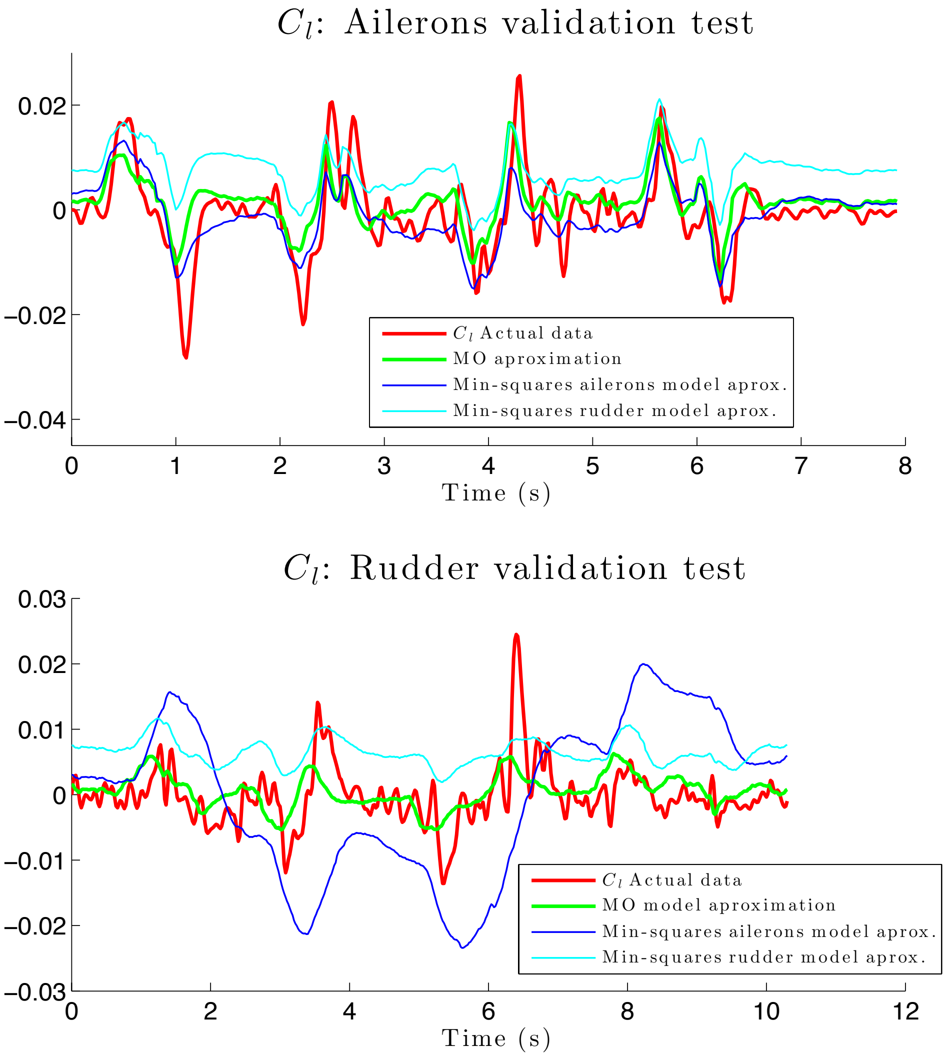


Figure 2.4. Objetives space: Pareto Front of lateral models



C_l: Ailerons validation test

C_l: Rudder validation test

Figure 2.5. C_l Validation results

algorithm with spherical pruning. In *Applications of Evolutionary Computation*, volume 6024 of *Lecture Notes in Computer Science*, pages 532–541. Springer Berlin Heidelberg, January 2010.

[5] K. Rodriguez-Vazquez and P. J. Fleming. Multi-objective genetic programming for nonlinear system identification. *Electronics Letters*, 34(9):930–931, April 1998.

[6] Jesús Velasco. Identificación de modelos dinámicos y ajuste de controladores basado en algoritmos evolutivos multiobjetivo. Master's thesis, Universidad Politècnica de València, Spain, September 2013.

[7] Jesús Velasco, Sergio García-Nieto, Gilberto Reynoso-Meza, and Javier Sanchis. Implementación de un sistema hardware-in-the-loop para la simulación en tiempo real de pilotos automáticos para UAVs. In *Actas de las XXXIV Jornadas de Automática*, October 2013.

[8] Jesus Velasco, Sergio García-Nieto Rodríguez, Gilberto Reynoso Meza, and Javier Sanchis Saez. Desarrollo y evaluación de una estación de control de tierra para vehículos aéreos no tripulados. In *Actas de las XXXIII Jornadas de Automática*, 2012.

[9] Haitao Xiang and Lei Tian. Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV). *Biosystems engineering*, 108(2):174–190, 2011.

[10] Hassan Yousefi, Heikki Handroos, and Azita Soleymani. Application of differential evolution in system identification of a servo-hydraulic system with a flexible load. *Mechatronics*, 18(9):513–528, November 2008.

Chapter 3

Multi-Objective Optimization for Wind Estimation and Aircraft Model Identification [1]

Abstract

In this paper, a novel method for aerodynamic model identification of a micro-air vehicle is proposed. The principal contribution is a technique of wind estimation that provides information about the existing wind during flight when no air-data sensors are available. The estimation technique employs multi-objective optimization algorithms that utilize identification errors to propose the wind-speed components that best fit the dynamic behavior observed. Once the wind speed is estimated, the flight experimentation data are corrected and utilized to perform an identification of the aircraft model parameters. A multi-objective optimization algorithm is also used, but with the objective of estimating the aerodynamic stability and control derivatives. Employing data from different flights offers the possibility of obtaining sets of models that form the Pareto fronts. Deciding which model best adjusts to the experiments performed (compromise model) will be the ultimate task of the control engineer.

[1] J. Velasco-Carrau, S. García-Nieto, J. V. Salcedo, and R. H. Bishop. Multi-Objective Optimization for Wind Estimation and Aircraft Model Identification. Journal of Guidance, Control, and Dynamics, vol. 39, n° 2, pp. 372-389, feb. 2016. https://doi.org/10.2514/1.G001294

Nomenclature

b	=	aircraft wingspan [m]
C_D	=	drag force coefficient
C_{D_i}	=	polynomial parameters of the drag coefficient model with $i = \{0, V, \alpha, \alpha^2, q, \delta_e\}$
C_l	=	torque coefficient in the X direction
C_{l_i}	=	polynomial parameters of the X aerodynamic moment coefficient model with $i = \{0, \beta, p, r, \delta_a, \delta_r\}$
C_L	=	lift force coefficient
C_{L_i}	=	polynomial parameters of the lift coefficient model with $i = \{0, V, \alpha, \alpha^2, \dot{\alpha}, q, \delta_e\}$
C_m	=	torque coefficient in the Y direction
C_{m_i}	=	polynomial parameters of the Y aerodynamic moment coefficient model with $i = \{0, V, \alpha, \alpha^2, \dot{\alpha}, q, \delta_e\}$
C_n	=	torque coefficient in the Z direction
C_{n_i}	=	polynomial parameters of the Z aerodynamic moment coefficient model with $i = \{0, \beta, p, r, \delta_a, \delta_r\}$
C_X	=	force coefficient in the X direction
C_Y	=	force coefficient in the Y direction
C_{Y_i}	=	polynomial parameters of the Y aerodynamic force coefficient model with $i = \{0, \beta, p, r, \delta_a, \delta_r\}$
C_Z	=	force coefficient in the Z direction
$\overline{c}$	=	aircraft wing chord [m]
$\mathbf{F}$	=	resulting force vector acting on aircraft body [N]
$\mathbf{F_A}$	=	aerodynamic force vector [N]
$\mathbf{F_G}$	=	gravity force vector [N]
$\mathbf{F_T}$	=	motor force vector [N]
$F_{\{x,y,z\}}$	=	$\{X, Y, Z\}$ components of the resultant force acting on the vehicle and expressed in the body system of reference $\{x_b, y_b, z_b\}$ [N]
g	=	gravitational field intensity [m/s^2]
I	=	aircraft tensor of inertia [kg/m^2]
I_p	=	propeller and rotor set inertia about their rotation axis [kg/m^2]
$I_{\{x,y,z\}}$	=	moments of inertia on $\{x_b, y_b, z_b\}$ axes [kg/m^2]
$I_{\{xy,xz,yz\}}$	=	products of inertia on x_b, y_b, z_b axes [kg/m^2]
J_i	=	Cost index for the i^{th} objective
m	=	aircraft total mass [kg]
$\mathbf{M}$	=	resulting moment vector acting on aircraft body [N·m]
$\mathbf{M_A}$	=	aerodynamic moment vector [N·m]
$\mathbf{M_T}$	=	motor moment vector [N·m]
$M_{\{x,y,z\}}$	=	$\{X, Y, Z\}$ components of the resultant moment acting on the vehicle and expressed in the body system of reference $\{x_b, y_b, z_b\}$ [N·m]
N	=	number of samples in a data set

p = aircraft angular X velocity component respect to ground and expressed in body axes [rad/s]

q = aircraft Y angular velocity component respect to ground and expressed in body axes [rad/s]

$\overline{q}$ = dynamic pressure [Pa]

r = aircraft Z angular velocity component respect to ground and expressed in body axes [rad/s]

S = aircraft aerodynamic surface [m^2]

t_i = i^{th} time instant. $i = 1, 2, ..., N$

T = propeller thrust acting in the direction of its rotation axis [N]

u = aircraft X velocity component respect to ground and expressed in body axes [m/s]

u_{air} = aircraft X velocity component respect to air and expressed in body axes [m/s]

$\mathbf{V}$ = aircraft velocity vector [m/s]

v = aircraft Y velocity component respect to ground and expressed in body axes [m/s]

v_{air} = aircraft Y velocity component respect to air and expressed in body axes [m/s]

V_{air} = aircraft airspeed [m/s]

V_0 = steady state airspeed [m/s]

w = aircraft Z velocity component respect to ground and expressed in body axes [m/s]

w_{air} = aircraft Z velocity component respect to air and expressed in body axes [m/s]

$\mathbf{W}$ = wind velocity vector [m/s]

X^* = unitary scaled value of a variable X

(x_b, y_b, z_b) = aircraft body axes

α = angle of attack [rad]

β = sideslip angle [rad]

δ_a = ailerons deflection [rad]

δ_e = elevators deflection [rad]

δ_r = rudder deflection [rad]

ζ = wind elevation when expressed in spherical coordinates [rad]

θ = aircraft pitch angle [rad]

μ = multi-objective optimization solution

σ = standard deviation of a data set

ϕ = aircraft roll angle [rad]

ξ = wind azimuth when expressed in spherical coordinates [rad]

ψ = aircraft yaw angle [rad]

Ω_p = propeller and rotor rotating speed [rad/s]

3.1 Introduction

There is an increasingly popular variety of applications that justify the development of unmanned aerial vehicles (UAVs) in the civil aviation field. Possible applications include photography for coastline control and beach erosion tracing, fire detection and control [23], infrastructure inspection, and measurements for agriculture [56]. In this new aeronautics field, high performance at the lowest cost is the main objective.

Several steps towards the achievement of this aim have already been taken. Firstly, it was necessary to reduce the cost and complexity of the aircraft itself. The result was a completely new generation of small airplanes whose size is the minimum necessary to house propulsion, sensorization, and control equipment. Secondly, the integrated systems (sensors, actuators, and control units) had to be powerful enough to control the fast dynamics of these vehicles when completing challenging missions. The cost of such devices is falling thanks to evolution in computer technology. The cost of the development phase has now become an important percentage of the total cost. In addition, hardware integrity is in greater danger during this phase. Therefore, a minimization of the total number of test hours is desired. Making use of simulation tools and utilizing acquired data as much as possible can lower development costs and risks. Thus, obtaining a dynamic model that tightly adjusts to the real flight behavior of the aircraft is essential for obtaining precise simulation results and correctly designing control algorithms. The process of going from observed data to a mathematical model is fundamental in science and engineering. In system theory, this process is known as system identification and the objective is to obtain dynamic models from observed input and output signals [35]. In particular, system identification methods have been used for flight-test evaluations [14, 13, 53, 17, 33, 16, 32], control analysis and design [11, 43] and advanced simulation [34, 4, 55].

Identifying the aerodynamic model of a low-cost micro-air vehicle (MAV) is a major challenge. Generally, wind tunnel tests are too expensive to be driven, and experimental flight data has to be used instead. In addition, this type of aircraft usually has a light body and flies slowly, meaning that the slightest breeze contributes significantly to overall airspeed. Hence, the information available from the inertial sensors is insufficient [54, 31, 30] for the identification of their aerodynamic model. In [41, 24, 5], different wind estimation techniques are presented. Those works make use of an extended Kalman filter to fuse inertial information with external sensors, such as pitot tubes or optical flow sensors. However, due to lack of space and resources air-data sensors may be unavailable or highly inaccurate in some occasions. Designing control strategies in these cases becomes a hard process, since there is no trustworthy model.

To improve a situation in which no air-data sensor is available, a two step identification methodology based on multi-objective optimization (MO) is presented in this paper. The methodology makes use of flight data instead of wind tunnel experiments to identify the non-dimensional stability and control derivatives of a micro-air vehicle. As the main contribution, our methodology starts with a wind estimation technique that complements the information collected by the inertial sensors. This technique takes information from the inertial unit, the global positioning system (GPS) sensor, and the control inputs to estimate the wind that best fits a given model structure. The quality of the identified models is consequently improved and no additional air-data sensor is used for that purpose. Model identification is performed in a second step. The identification technique also relies on the advantages offered by an MO perspective, enabling the designer to test flight data from different types of experiments. Thereby, models with acceptable performance in various realistic flight regimes are obtained.

The paper is organized as follows. Section 3.2 introduces the aircraft and the hardware used in the experiments along with the dynamic and aerodynamic models. In Section 3.3, the estimation procedure is presented and developed. Section 3.4 provides a step-by-step explanation of how to obtain the final parameters of the aerodynamic models once the data has been corrected. The results are divided in two sections. Section 3.5 presents simulation results that verify the validity of the wind estimation technique. Section 3.6 presents results for both the estimation and identification tasks. Section 3.7 presents the final conclusions.

3.2 UAV Testbench

3.2.1 Flight System

The main component of the UAV flight system is a Kadett 2400 aircraft manufactured by Graupner. The aircraft has a very lightweight frame and characteristics that make it suitable for the purposes of this research. These characteristics include a 2.4 m wing span, 0.9 m^2 of wing surface, 48.07 N/m^2 wing loading, and 1.65×10^{-2} m^3 of available volume to house control hardware.

Figure 3.1 illustrates interconnection between the UAV devices and the flight control system. The aircraft houses all the devices necessary for manual, as well as automatic, control. During normal flight, the tail rudder, elevators, and ailerons serve as the control surfaces. Propulsion is provided by a brushless alternating current engine supplied by two lithium-ion polymer (LiPo) batteries through a frequency variator. The variator and the servomotors are controlled by pulse width modulated (PWM) command signals. The servo switch controller (SSC) switches between manual and autonomous flight modes. It also enables data acquisition and the application of control surface deflections and motor torque changes.

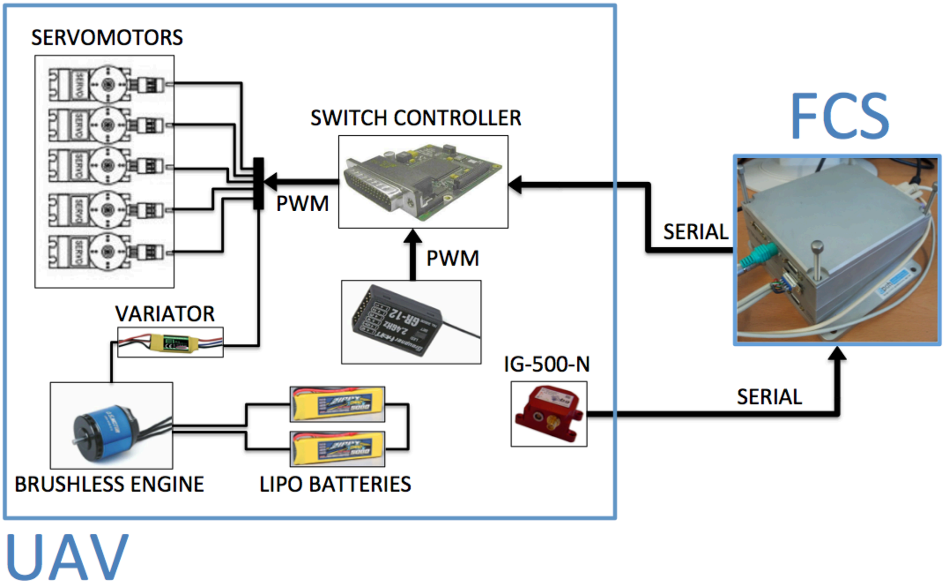


Figure 3.1. Interconnection between the UAV devices and the flight control system

The flight control station (FCS), housed in a PC-104, hosts the control algorithms. The control loop is closed by a IG500N unit from SBG Systems, that integrates a wide range of sensors, including the accelerometers, gyroscopes, and magnetometers. A Kalman filter fuses the sensor information to estimate the position, orientation, linear and angular speed, and acceleration. Table 3.1 provides manufacturer's accuracy data for the IG500N unit. This same platform was presented in [52, 51, 50, 49] together with the results of the first flight tests.

3.2.2 Aircraft Dynamic Model

As stated in [20], the aircraft dynamic model is given by the force equations,

$$\begin{aligned}\dot{u} &= rv - qw + \frac{\overline{q}S}{m}C_X(\delta_{[e,a,r]}) - g\sin\theta + \frac{T}{m}\\ \dot{v} &= pw - ru + \frac{\overline{q}S}{m}C_Y(\delta_{[e,a,r]}) + g\cos\theta\sin\phi\\ \dot{w} &= qu - pv + \frac{\overline{q}S}{m}C_Z(\delta_{[e,a,r]}) + g\cos\theta\cos\phi\end{aligned} \tag{3.1}$$

, torque equations,

Table 3.1. IG500N unit characteristics

Sensor	Characteristic	Value
Unit attitude		
	Static accuracy (Pitch)	±0.5 deg
	Static accuracy (Roll)	±0.5 deg
	Static accuracy (Heading)	±1.0 deg
	Dynamic accuracy	±1.0 deg rms
Accelerometers		
	Non-linearity	< 0.2 % of full scale
	Bias stability	±5 mg
Gyroscopes		
	Non-linearity	< 0.1 % of full scale
	Bias stability	±0.5 deg/s
Magnetometers		
	Non-linearity	< 0.2 % of full scale
	Bias stability	±0.5 mG
GPS Receiver		
	Horizontal accuracy	2.0 m
	Vertical accuracy	5.0 m

$$\begin{aligned}\dot{p} - \frac{I_{xz}}{I_x}\dot{r} &= \frac{\overline{q}Sb}{I_x}C_l(\delta_{[e,a,r]}) - \frac{I_z - I_y}{I_x}qr + \frac{I_{xz}}{I_x}qp \\ \dot{q} &= \frac{\overline{q}S\overline{c}}{I_y}C_m(\delta_{[e,a,r]}) - \frac{I_x - Iz}{I_y}pr - \frac{I_{xz}}{I_y}(p^2 - r^2) + I_p\Omega_p r \\ \dot{r} - \frac{I_{xz}}{I_z}\dot{p} &= \frac{\overline{q}Sb}{I_z}C_n(\delta_{[e,a,r]}) - \frac{I_y - I_x}{I_z}pq - \frac{I_{xz}}{I_z}qr - I_p\Omega_p q\end{aligned} \tag{3.2}$$

and kinematic equations,

$$\begin{aligned}\dot{\phi} &= p + \tan\theta(q\sin\phi + r\cos\phi) \\ \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\psi} &= \frac{q\sin\phi + r\cos\phi}{\cos\theta}\end{aligned} \tag{3.3}$$

In Eq. 3.1, Eq. 3.2 and Eq. 3.3 g is the gravitational field intensity near the Earth's surface, and m is the total mass of the system. Given the body reference frame $X_bY_bZ_b$ illustrated in Fig. 3.2, (u, v, w) are the components of the translational velocity, (p, q, r) the components of the angular velocity, (I_x, I_y, I_z) are the moments of inertia, and I_{xz} is a product of inertia. The products of inertia I_{xy}

and I_{yz}, related to the longitudinal plane ($Y_b = 0$), are both null because of the aircraft's symmetry with respect to this plane. I_p is the rotating inertia of the tandem motor and propeller, Ω_p is its rotating speed, and T is the motor thrust. S, b and $\bar{c}$ are the the Kadett 2400 aerodynamic surfaces, wingspan, and wing chord respectively, and $\bar{q}$ is the dynamic pressure, which is a function of the air density and airspeed relative to the local wind. The aerodynamic coefficients (AC) C_X, C_Y, C_Z, C_l, C_m, and C_n, are functions of the system variables. In particular, the δ symbol in brackets represents its dependency on the deflections of the control surfaces (δ_e, δ_a and δ_r are the elevators, ailerons, and rudder deflections respectively). The aerodynamic coefficients will be presented in further detail in Section 3.2.3. Finally, the aircraft orientation is represented by the Euler angles of roll ϕ, pitch θ, and yaw ψ.

3.2.3 Aircraft Aerodynamic Model

In Klein and Morelli [20], detailed information on the aerodynamic coefficients is provided. Firstly, if we assume a scenario in which the aircraft is in steady flight, and only performs short maneuvers that begin from this state, we can truncate the Taylor series expansion and retain only the first or second-order terms. Furthermore, under the assumption of small perturbations, and based on the symmetry of the vehicle, it can be assumed that: 1) the symmetrical (longitudinal) variables u, w and q do not affect asymmetrical (lateral) force and torques; and similarly, 2) asymmetric (lateral) variables v, p and r do not affect the symmetrical (longitudinal) forces and torque. The aerodynamic coefficients are given by the longitudinal aerodynamic models,

$$\begin{aligned}
C_D = \;& C_{D_0} + C_{D_{V_{air}}}\tfrac{1}{V_0}\Delta V_{air} + C_{D_\alpha}\Delta\alpha + C_{D_{\alpha^2}}\Delta\alpha^2 \\
& + C_{D_q}\tfrac{\bar{c}}{2V_0}q + C_{D_{\delta_e}}\Delta\delta_e \\
C_L = \;& C_{L_0} + C_{L_{V_{air}}}\tfrac{1}{V_0}\Delta V_{air} + C_{L_\alpha}\Delta\alpha + C_{L_{\alpha^2}}\Delta\alpha^2 + C_{L_{\dot\alpha}}\tfrac{\bar{c}}{2V_0}\dot\alpha \\
& + C_{L_q}\tfrac{\bar{c}}{2V_0}q + C_{L_{\delta_e}}\Delta\delta_e \\
C_m = \;& C_{m_0} + C_{m_{V_{air}}}\tfrac{1}{V_0}\Delta V_{air} + C_{m_\alpha}\Delta\alpha + C_{m_{\alpha^2}}\Delta\alpha^2 + C_{m_{\dot\alpha}}\tfrac{\bar{c}}{2V_0}\dot\alpha \\
& + C_{m_q}\tfrac{\bar{c}}{2V_0}q + C_{m_{\delta_e}}\Delta\delta_e
\end{aligned} \tag{3.4}$$

and the lateral aerodynamic models,

$$\begin{aligned}
C_Y =& C_{Y_0} + C_{Y_\beta}\Delta\beta + C_{Y_p}\frac{b}{2V_0}p + C_{Y_r}\frac{b}{2V_0}r + C_{Y_{\delta_a}}\Delta\delta_a + C_{Y_{\delta_r}}\Delta\delta_r \\
C_l =& C_{l_0} + C_{l_\beta}\Delta\beta + C_{l_p}\frac{b}{2V_0}p + C_{l_r}\frac{b}{2V_0}r + C_{l_{\delta_a}}\Delta\delta_a + C_{l_{\delta_r}}\Delta\delta_r \\
C_n =& C_{n_0} + C_{n_\beta}\Delta\beta + C_{n_p}\frac{b}{2V_0}p + C_{n_r}\frac{b}{2V_0}r + C_{n_{\delta_a}}\Delta\delta_a + C_{n_{\delta_r}}\Delta\delta_r
\end{aligned} \tag{3.5}$$

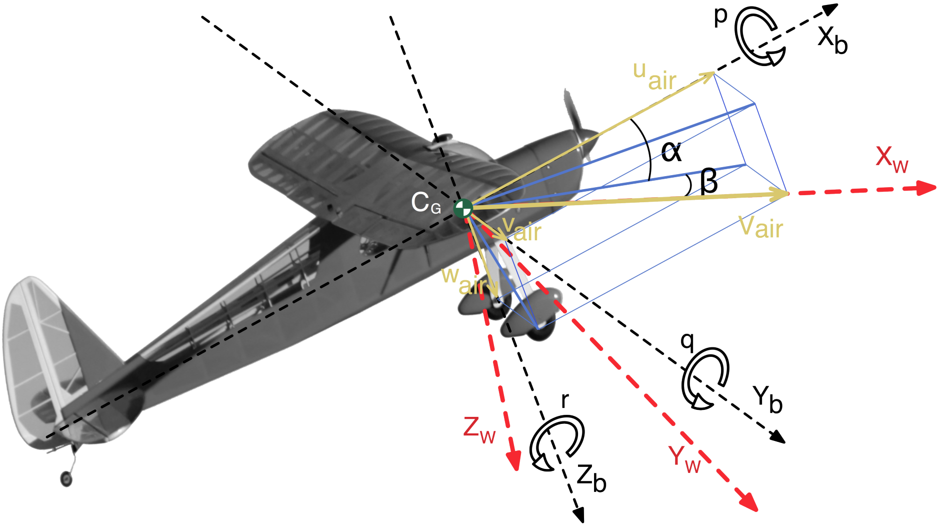


Figure 3.2. Aircraft body axes, aircraft wind axes, and wind relative velocity

where α and β are the angle of attack and of sideslip, respectively, and V_{air} is the airspeed (see Fig. 3.2). In particular, V_0 is airspeed measured at the steady state of flight, before a maneuver begins. These variables are velocity dependent and calculated as:

$$\alpha = \arctan\left(\frac{w_{air}}{u_{air}}\right); \quad \text{and} \quad \beta = \arcsin\left(\frac{v_{air}}{V_{air}}\right); \tag{3.6}$$

where $V_{air} = |\mathbf{V_{air}}|$. As denoted in Fig. 3.2, u_{air}, v_{air} and w_{air} are the three components of the aircraft velocity with respect to air. Under zero-wind conditions $(u_{air}, v_{air}, w_{air}) = (u, v, w)$. Finally, C_L and C_D are the lift and drag coefficients and their relation to C_X and C_Z is:

$$C_L = -C_Z \cos\alpha + C_X \sin\alpha; \quad \text{and} \quad C_D = -C_X \cos\alpha - C_Z \sin\alpha; \tag{3.7}$$

Thus, the aerodynamic model identification is based on extracting the constants of the polynomials of Eq. (3.4) and (3.5) from the flight data and by means of the dynamic model. Those constants are called non-dimensional stability and control derivatives.

3.3 Wind Estimation Technique

3.3.1 Methodology Outline

MAVs are generally unable to carry precise airspeed sensors that provide three-dimensional data needed for correct identification of the aerodynamic model. In the case of the platform presented in this paper, a GPS and an inertial-magnetic unit (IMU) supply a reasonably good estimate of the velocities relative to the Earth's surface, but do not provide information about the velocities relative to air. To improve the identified stability and control derivatives, a wind estimation methodology is incorporated into the procedure of parameter identification. The estimation methodology is based on the fact that, for small airplanes, which fly at relatively low airspeeds, the smallest breeze may be a large percentage of the total airspeed value, thus introducing a large error if not taken into account. Hence, assuming the model structure is well defined, an optimization problem can be posed in which a three-component solution (wind) is searched to minimize the error of a particular aerodynamic coefficient model.

It will be shown in the following sections that longitudinal or lateral experiments can be used to obtain all the coefficient models. This means that a total number of three models per experiment can be derived. Therefore, the aforementioned optimization becomes a particular multi-objective optimization problem in which a unique solution should be obtained if the actual wind is found. In practice, a cloud of solutions close to the real wind will be obtained by the optimizer.

3.3.2 Multi-objective Optimization

In engineering problems, it is a common issue to deal with situations that require the optimization of multiple objectives that include physical constraints, operational constraints, and non-linearities. Due to this fact, addressing these problems from the standpoint of classical optimization is insufficient [38]. The multi-objective optimization problem (MOP) can be stated as:

$$\min_{\mu \in \mathbb{R}} J(\mu) = [J_1(\mu), J_2(\mu), \ldots, J_m(\mu)] \tag{3.8}$$

where μ is the solution that minimizes the m cost functions J_i at the same time. Generally, it will not be possible to find a solution that satisfies all requirements at once, so the optimizer will have to provide the set of solutions that are not improved by any other set in all the objectives simultaneously. That set of solutions is known as the Pareto set Θ^*, and their values in the objective space create the Pareto front J_p^*.

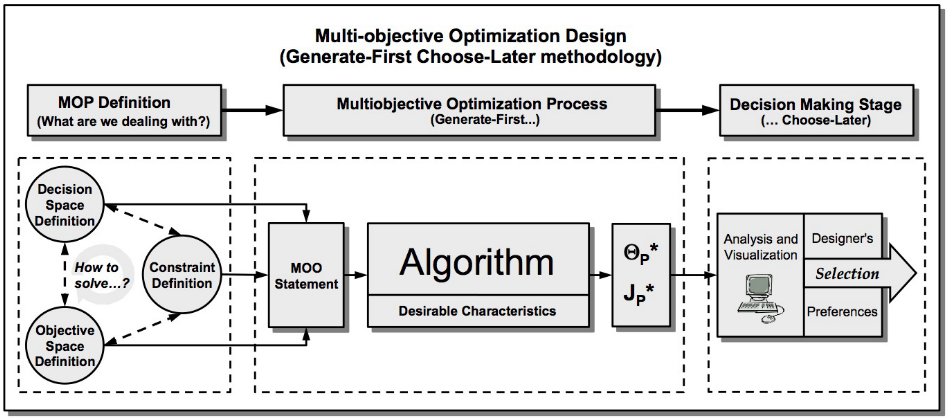

Multi-objective Optimization Design
(Generate-First Choose-Later methodology)
MOP Definition
(What are we dealing with?)
Multiobjective Optimization Process
(Generate-First...)
Decision Making Stage
(... Choose-Later)
Decision Space Definition
How to solve...?
Constraint Definition
Objective Space Definition
MOO Statement
Algorithm
Desirable Characteristics
Θ_P^*
J_P^*
Analysis and Visualization
Designer's
Selection
Preferences

Figure 3.3. Multi-objective Optimization Design (MOOD) [45]

Definition 3. *(Pareto Optimality [29]): An objective vector $J(\mu_1)$ is Pareto optimal if there is no other objective vector $J(\mu_2)$ such that $J_i(\mu_2) \leq J_i(\mu_1)$ for all $i \in [1, 2, \ldots, m]$ and $J_j(\mu_2) < J_j(\mu_1)$ for at least one $j, j \in [1, 2, \ldots, m]$.*

Definition 4. *(Strict Dominance [29]): An objective vector $J(\mu_1)$ is dominated by another objective vector $J(\mu_2)$ if $J_i(\mu_2) < J_i(\mu_1)$ for all $i \in [1, 2, \ldots, m]$.*

Definition 5. *(Dominance [29]): An objective vector $J(\mu_1)$ dominates another vector $J(\mu_2)$ if $J(\mu_1)$ is not worse than $J(\mu_2)$ in all objectives and is better in at least one objective; that is $J(\mu_1) \prec J(\mu_2)$.*

Figure 3.3 illustrates how MO is employed as a design methodology. Three stages comprise the procedure: MOP definition; a multi-objective optimization process; and decision making [1]. The technique must be viewed as a holistic process in which equal importance is assigned to each stage so that the design process is successfully driven [3]. Hence, objective and decision spaces and their constraints must be well defined in the MOP definition stage so that the correct problem is optimized in the optimization process. Finally, a deep analysis should be carried out (once an approximation to the Pareto front is available) to detect the most convenient solution in the decision making stage. This same topology is followed in the wind estimation process.

MOP Definition

An aircraft is a complex system with multiple control inputs that simultaneously excite multiple state variables. As already mentioned, the aerodynamic coefficients depend on the inputs and state variables. If an experiment is performed in which a longitudinal input is altered by collecting the longitudinal variable values then any coefficient of this type can be modeled. Thus, elevator deflection and motor thrust variations (which generate changes in the longitudinal variables) can be used to model any longitudinal coefficient and, in the same way, ailerons and rudder deflections can be used to model lateral coefficients. Experiments in which elevators or motors are moved from their setpoints will be denoted as longitudinal experiments, and similarly, experiments in which the tail rudder or ailerons are moved from their setpoints will be denoted as lateral experiments.

For the purpose of the wind estimation problem as proposed here, the methodology is reinforced by the fact that at least three models can be extracted from the same data set. Indeed, if the correct wind is estimated, estimation errors for all coefficient models will decrease simultaneously. From a different standpoint, if a solution in the wind components that minimizes the error of the three coefficient models at the same time is found, it is probable that this solution is the actual wind experienced during the flight experiment.

Short time experiments are performed and utilized in the wind estimation and identification process. Test duration is an important question because it directly affects the wind estimation process. Bidirectional input-step experiments were made with the minimum time required so that the assumption of constant wind remains reasonable. Three constant wind components are then used as an approximation of the wind along each experiment. Three objectives are defined, one per aerodynamic coefficient model. If the MSE is used as the performance index of the identification process, three cost functions can be defined for each experiment. The three longitudinal cost functions are:

$$
\begin{aligned}
J_1(\mathbf{W}) =& \frac{1}{N}\sum_{h=1}^{N}\Big[C_D(t_h,\mathbf{W},\cdots) - \hat{C}_D(t_h,\mathbf{W},\cdots)\Big]^2 \\
J_2(\mathbf{W}) =& \frac{1}{N}\sum_{h=1}^{N}\Big[C_L(t_h,\mathbf{W},\cdots) - \hat{C}_L(t_h,\mathbf{W},\cdots)\Big]^2 \\
J_3(\mathbf{W}) =& \frac{1}{N}\sum_{h=1}^{N}\Big[C_m(t_h,\mathbf{W},\cdots) - \hat{C}_m(t_h,\mathbf{W},\cdots)\Big]^2
\end{aligned}
\tag{3.9}
$$

Similarly, the lateral cost functions are:

$$\begin{aligned}J_1(\mathbf{W}) =& \frac{1}{N}\sum_{h=1}^{N}\Big[C_Y(t_h,\mathbf{W},\cdots) - \hat{C}_Y(t_h,\mathbf{W},\cdots)\Big]^2\\ J_2(\mathbf{W}) =& \frac{1}{N}\sum_{h=1}^{N}\Big[C_l(t_h,\mathbf{W},\cdots) - \hat{C}_l(t_h,\mathbf{W},\cdots)\Big]^2\\ J_3(\mathbf{W}) =& \frac{1}{N}\sum_{h=1}^{N}\Big[C_n(t_h,\mathbf{W},\cdots) - \hat{C}_n(t_h,\mathbf{W},\cdots)\Big]^2\end{aligned} \tag{3.10}$$

In Eq. 3.9 and Eq. 3.10 $\hat{C}_D$, $\hat{C}_L$, $\hat{C}_m$, $\hat{C}_Y$, $\hat{C}_l$, and $\hat{C}_n$ are the values that the identified coefficient models provide for C_D, C_L, C_m, C_Y, C_l, and C_n, respectively. These cost functions constitute the objective space while the three possible wind components define the decision space. In this paper, the wind speed is expressed in spherical coordinates with the vector magnitude $|\mathbf{W}|$ as the radius, and the two rotation angles, elevation, denoted by ζ, and azimuth, denoted by ξ; giving the triple $\mathbf{W} = (|\mathbf{W}|, \zeta, \xi)$. With the aim of unequivocally defining the decision space, the radius, the elevation, and the azimuth should be enclosed into three intervals consistent with the cost functions domain. The space definition of this interval in this paper is:

$$D_{(J)} = \{(|\mathbf{W}|, \zeta, \xi) : |\mathbf{W}| \in [0, +\infty[\,, \zeta \in [-\pi/2, \pi/2]\,, \xi \in [0, 2\pi[\} \tag{3.11}$$

Finally, constraints may be included in the objectives, as well as in the decision variables. In this work, constraints have been introduced only in the decision space in order to narrow the space of possible solutions. Such a narrowing has been performed based on knowledge about the maximum magnitude of the wind speed during the day of the flight experiments.

Multi-objective Optimization Process

Extensive literature exists about how multi-objective optimization problems can be solved. Some of the classical strategies to approximate the Pareto set include: normal constraint method [26, 44], normal boundary intersection (NBI) method [9], epsilon constraint techniques [29] and physical programming [27]. Multi-objective evolutionary algorithms (MOEA) have been used to approximate a Pareto set [58], due to their flexibility when evolving an entire population towards the Pareto front. A comprehensive review of the early stages of MOEAs is contained in [6]. There are several popular evolutionary and nature-inspired techniques used by MOEAs. The most popular techniques include genetic algorithms (GA) [47, 22], particle swarm optimization (PSO) [19, 7], and differential evolution (DE) [48, 28, 10]. Nevertheless, evolutionary techniques such as artificial bee colony (ABC) [18] or ant colony optimization (ACO) [12] algorithms are becoming popular. No evolutionary technique is better than the others, since all have drawbacks and advantages. These evolutionary/nature-inspired techniques require mechanisms to deal with evolutionary multi-objective optimization (EMO) since they were originally used for single-objective optimization. While the dominance criterion (definition 5) could be used to evolve the population towards a Pareto front, it could be insufficient to achieve a minimum degree of satisfaction in other desirable characteristics for a MOEA (diversity, for instance)[36].

The authors of this paper have taken part in the development of a MOEA called the spMODE algorithm [40, 39]. It is a heuristic algorithm that makes use of the convergence properties of evolution to approximate the Pareto front. It uses physical programming to incorporate the designer's preferences, size control of the approximated Pareto front, as well as spherical pruning to improve spreading. Hence it is a MOEA with mechanisms to improve and deal with diversity, pertinency, many-objective optimization instances, and constrained optimization instances. Although spMODE has been chosen to solve this MOP, any other multi-objective optimizer could be used for this purpose.

Since an evolution algorithm is used, multiple wind candidates are proposed in each generation by the optimizer and all are then evaluated. Figure 3.4 illustrates the routine followed by the optimizer. Starting from a given wind-speed, the airspeed denoted by $\mathbf{V_{air}}$ is calculated as:

$$\mathbf{V_{air}} = \mathbf{V_{GPS}} - \mathbf{W} \tag{3.12}$$

where $\mathbf{V_{GPS}}$ denotes the aircraft velocity relative to the Earth's surface. Note that $\overline{q}$ is dependent on $\mathbf{V_{air}}$. Once the aircraft velocity relative to air is available, the airspeed dependent variables on the right side of the aircraft aerodynamic model (in Eqs. (3.4) and (3.5)) can be obtained. Furthermore, as the aerodynamic coefficients cannot be measured directly, dynamic expressions must be used for the purpose of estimating their values. These relationships are given by [20]:

$$\begin{aligned}
C_X =& \frac{1}{\overline{q}S}(ma_x - T)\\
C_Y =& \frac{ma_y}{\overline{q}S}\\
C_Z =& \frac{ma_z}{\overline{q}S}\\
C_l =& \frac{1}{\overline{q}Sb}\left[I_x\dot{p} - I_{xz}\left(pq + \dot{r}\right) + \left(I_z - I_y\right)qr\right]\\
C_m =& \frac{1}{\overline{q}S\overline{c}}\left[I_y\dot{q} + \left(I_x - I_z\right)pr + I_{xz}\left(p^2 - r^2\right) - I_p\Omega_p r\right]\\
C_n =& \frac{1}{\overline{q}Sb}\left[I_z\dot{r} - I_{xz}\left(\dot{p} - qr\right) + \left(I_y - I_x\right)pq + I_p\Omega_p q\right]
\end{aligned} \tag{3.13}$$

Note that $\overline{q}$ is present in each relationship of Eq.(3.13). This means that the aerodynamic coefficients are directly dependent on airspeed and thus, on the wind during data recollection. Due to this fact, recalculation of the aerodynamic coefficients is carried out in each evaluation performed by the MOEA.

The next step is the cost calculation. After calculating the airspeed dependent variables, the aerodynamic coefficients and the regressors are scaled according to their standard deviations. The scaling expression is:

$$X^* = \frac{X - X_0}{\sqrt{N}\Delta\sigma(X)} \tag{3.14}$$

where X represents any of those airspeed dependent variables, X^* is its value after being scaled by applying Eq. (3.14), and N and $\sigma(X)$ are the number of samples and the standard deviation of X during the experiment, respectively. This strategy is often followed in regression analysis of a multivariate distribution to overcome the problem of variances of the residuals changing at different input variable values [8]. Since we are recalculating the estimated variables C_i at each iteration and they are directly proportional to V_{air}^{-2} (given that $\overline{q} = 1/2\rho V_{air}^2$), this scaling makes the estimation error independent of the velocity's magnitude. After scaling, the least-squares method is applied to obtain three longitudinal auxiliary models,

$$\begin{aligned}
C_D^* =& C_{D_{V^*}} V^* + C_{D_{\alpha^*}} \alpha^* + C_{D_{(\alpha^2)^*}} (\alpha^2)^* + C_{D_{q^*}} q^* + C_{D_{\delta_e^*}} \delta_e^* \\
C_L^* =& C_{L_{V^*}} V^* + C_{L_{\alpha^*}} \alpha^* + C_{L_{(\alpha^2)^*}} (\alpha^2)^* + C_{L_{\dot{\alpha}^*}} \dot{\alpha}^* + C_{L_{q^*}} q^* + C_{L_{\delta_e^*}} \delta_e^* \\
C_m^* =& C_{m_{V^*}} V^* + C_{m_{\alpha^*}} \alpha^* + C_{m_{(\alpha^2)^*}} (\alpha^2)^* + C_{m_{\dot{\alpha}^*}} \dot{\alpha}^* + C_{m_{q^*}} q^* + C_{m_{\delta_e^*}} \delta_e^*
\end{aligned} \tag{3.15}$$

or three lateral auxiliary models

$$\begin{aligned}
C_Y^* =& C_{Y_{\beta^*}} \beta^* + C_{Y_{p^*}} p^* + C_{Y_{r^*}} r^* + C_{Y_{\delta_a^*}} \delta_a^* + C_{Y_{\delta_r^*}} \delta_r^* \\
C_l^* =& C_{l_{\beta^*}} \beta^* + C_{l_{p^*}} p^* + C_{l_{r^*}} r^* + C_{l_{\delta_a^*}} \delta_a^* + C_{l_{\delta_r^*}} \delta_r^* \\
C_n^* =& C_{n_{\beta^*}} \beta^* + C_{n_{p^*}} p^* + C_{n_{r^*}} r^* + C_{n_{\delta_a^*}} \delta_a^* + C_{n_{\delta_r^*}} \delta_r^*
\end{aligned} \tag{3.16}$$

Note that these models are utilized to acquire a value of the fitting goodness and do not represent the actual behavior of the aerodynamic coefficients. Identifying the real models is accomplished after estimating the wind and correcting the experimental data.

Once these auxiliary models are available, the MSE is computed and with it, the value of three cost functions. Finally, if the currently evaluated wind is a non-dominated solution, it is added as part of the Pareto front approximation.

Otherwise, it is discarded as a solution, though used as valuable information in the evolution process (see Fig. 3.4).

Decision Making Stage

A unique wind must be chosen. Acknowledging that this technique is not being used for design but for estimating, a best solution does exist that is factually and independent of the designer's preferences. The question is, does the estimate accurately represent the wind? To address that key question, the following two-step validation process is proposed:

1. Firstly, a 3D representation of the whole set of wind vectors (set of solutions given by the optimizer) is obtained. If that set is concentrated around a given point, then this is likely a good estimation of the wind speed. On the contrary, if the set of solutions is scattered or concentrated at multiple points, then there is not a unique global minimum and, therefore, the estimation process was unsuccessful.

2. Secondly, if the first step led to the conclusion of a successful estimation, this fact may still be refuted or confirmed with a set of validation data. The way to proceed is:

    (a) Perform a least-squares identification with two sets of data. This will result in two models for each of the aerodynamic coefficients. Note that the estimated wind is not yet used.

    (b) Perform a cross validation for each of the models obtained in 2a. This is, taking a model identified with set 1, compute its MSE for set 2 and *vice versa*.

    (c) Correct each set of data with the estimated wind, recalculating airspeed and all airspeed dependent variables accordingly.

    (d) As in 2a, use the least-squares method to identify the aerodynamic coefficient models that best fit each set of treated data. Note that unlike step 2a, this time the estimated wind has been used to correct the data.

    (e) Perform a cross validation of the models identified in 2d, to obtain their fitting errors. If validation errors are now smaller than the ones obtained in step 2b, then the wind speeds were successfully estimated. Otherwise, the minimums found by the algorithm are not the wind speeds acting during the experiments.

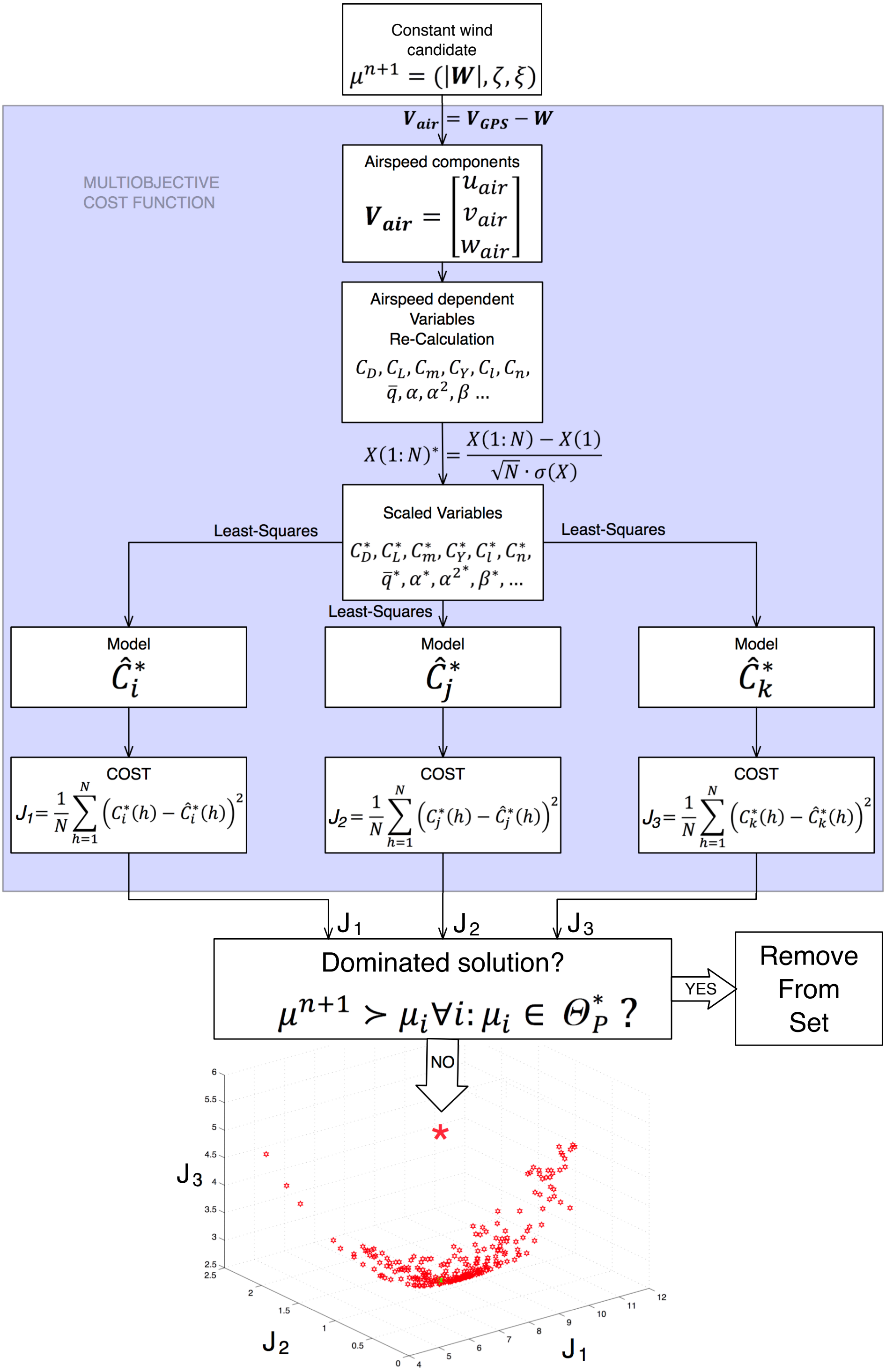

Constant wind candidate
$\mu^{n+1} = (|\boldsymbol{W}|, \zeta, \xi)$
$\boldsymbol{V_{air}} = \boldsymbol{V_{GPS}} - \boldsymbol{W}$
MULTIOBJECTIVE COST FUNCTION
Airspeed components
$\boldsymbol{V_{air}} = \begin{bmatrix} u_{air} \\ v_{air} \\ w_{air} \end{bmatrix}$
Airspeed dependent Variables Re-Calculation
$C_D, C_L, C_m, C_Y, C_l, C_n, \bar{q}, \alpha, \alpha^2, \beta$ ...
$X(1{:}N)^* = \frac{X(1{:}N) - X(1)}{\sqrt{N} \cdot \sigma(X)}$
Scaled Variables
$C_D^*, C_L^*, C_m^*, C_Y^*, C_l^*, C_n^*, \bar{q}^*, \alpha^*, \alpha^{2^*}, \beta^*, \ldots$
Least-Squares
Least-Squares
Least-Squares
Model $\hat{C}_i^*$
Model $\hat{C}_j^*$
Model $\hat{C}_k^*$
COST $J_1 = \frac{1}{N}\sum_{h=1}^{N}\left(C_i^*(h) - \hat{C}_i^*(h)\right)^2$
COST $J_2 = \frac{1}{N}\sum_{h=1}^{N}\left(C_j^*(h) - \hat{C}_j^*(h)\right)^2$
COST $J_3 = \frac{1}{N}\sum_{h=1}^{N}\left(C_k^*(h) - \hat{C}_k^*(h)\right)^2$
J_1
J_2
J_3
Dominated solution?
$\mu^{n+1} \succ \mu_i \forall i{:}\ \mu_i \in \Theta_P^*$?
YES
Remove From Set
NO
J_3
J_2
J_1
6
5.5
5
4.5
4
3.5
3
2.5
2.5
2
1.5
1
0.5
0
4
5
6
7
8
9
10
11
12

Figure 3.4. Wind estimation process

3.4 Aerodynamic Model Identification

A methodology based on least-squares is commonly used for modeling aerodynamic coefficients. When multiple inputs excite the variables of one model, considerations such as coordination, correlation, and relative effectiveness appear. In [21] the authors give detailed information about how to design experiments for aerodynamic model identification with multiple inputs involved. Optimally designed time-skewed doublet inputs seem to be a good option in these cases. However, conducting optimal experiments becomes impossible for a pilot controlling the aircraft from earth, as in the case of MAVs. A problem of experiment effectiveness appears because when time-skewed doublet inputs are used, the duration of each experiment determines its weight in the optimization process. For this reason, a multi-objective optimization is proposed here for the aerodynamic model identification of an MAV. Multi-objective techniques applied to model identification have achieved very good results in many cases, as shown in [42, 46, 57]. When the optimization problem turns out to be non-convex, there exist solutions in the Pareto set that remain unreachable for a weighted-sum method. Figure 3.5 shows a non-convex Pareto front and how the straight lines resulting from the different combination of objectives weights are unable to reach part of the Pareto front. Thus, optimizing objectives separately present a great advantage when non-convex problems must be solved. Several additional advantages derive from an MO perspective applied to this particular problem. Firstly, the weights of each type of experiment can be determined *a posteriori*. Secondly, flight conditions do not depend on the previous experiment. Thirdly, metrics other than mean squared error (MSE) can be used in the optimization. And lastly, the duration of the experiments is reduced. The latter favors our wind estimation process because the constant wind assumption weakens as the duration of the experiments increase.

Thus, the second part of this work is the estimation of the aerodynamic model that describes how the MAV reacts to changes in control inputs. As mentioned, an accurate MAV model cannot be obtained without taking wind disturbances into account. For that purpose a process of wind estimation was detailed in the previous section and now an aerodynamic model identification that makes use of the wind information is needed. Once information about the estimated wind acting during tests is available, variables affected by the relative airspeed may be corrected. The relative velocity vector is computed in first place, and then the angle of attack α and the side-slip angle β are estimated by means of Eq. (3.6).

As stated, longitudinal and lateral experiments, independently excite different sets of aerodynamic coefficients. Four different experiments can be performed to excite the longitudinal and lateral aerodynamic coefficients. Elevator deflections and motor thrust variations, generate changes in the longitudinal variables and, aileron and rudder deflections generate changes in the lateral variables. Therefore, longitudinal and lateral coefficient models can be identified from different type of

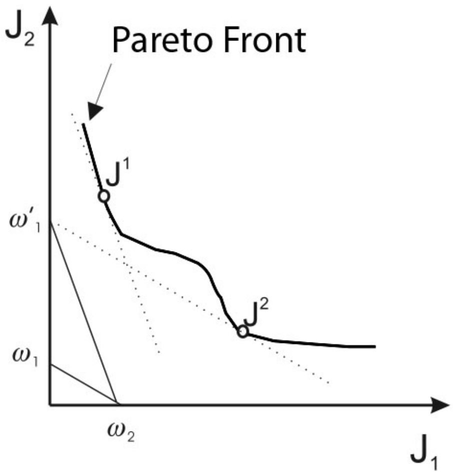


Figure 3.5. Pareto front approximation by weighted-sum technique

experiments. As an example, if a C_D model is obtained by optimizing an elevator deflection test, the model performance on motor experiment data will decrease, and *vice versa*. So, an identification process that takes both experiments into account simultaneously is a bi-objective optimization problem.

Figure 3.6 has been included to illustrate the bi-objective optimization concept. If the MSE is again used as the performance index of the identification process, two cost functions can be defined for each aerodynamic coefficient. The two cost functions used for obtaining any of the longitudinal models are:

$$J_1 = \frac{1}{N_{elevator}} \sum_{i=1}^{N_{elevator}} \left[C_j(t_i) - \hat{C}_j(t_i) \right]^2 \qquad \forall j \in \{D, L, m\} \tag{3.17}$$

$$J_2 = \frac{1}{N_{motor}} \sum_{i=1}^{N_{motor}} \left[C_j(t_i) - \hat{C}_j(t_i) \right]^2 \qquad \forall j \in \{D, L, m\} \tag{3.18}$$

where $\hat{C}_j(t_i)$ is the model approximation of the C_j value at the instant t_i and $N_{elevator}$ and N_{motor} are the number of samples of each type of experiment. Similar cost functions can be defined for the three lateral models.

Then, if C_l is to be modeled using aileron and rudder experiments, the identification problem from this MO point of view should be stated as

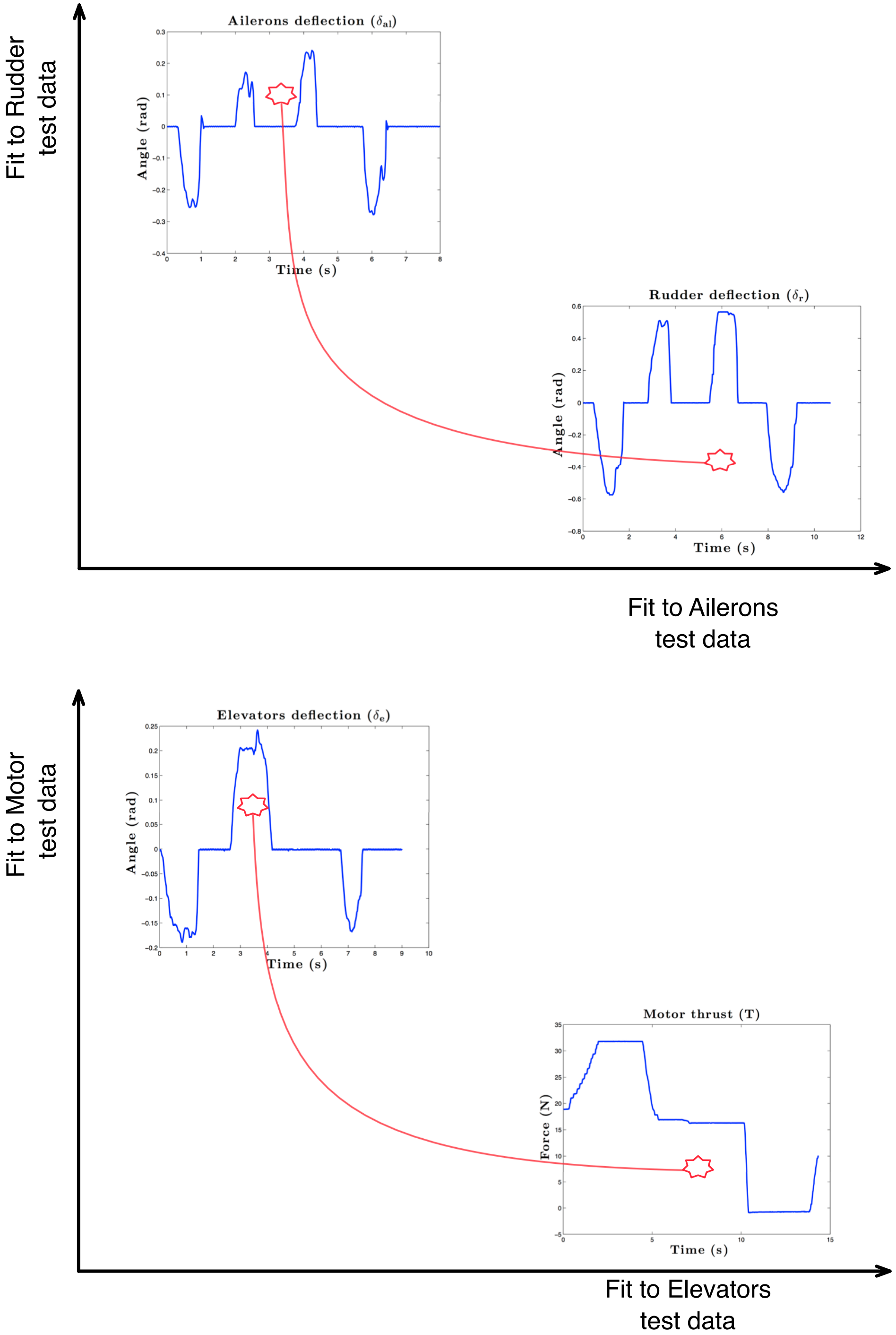

Fit to Rudder test data
Ailerons deflection (δ_{al})
Angle (rad)
Time (s)
Rudder deflection (δ_r)
Angle (rad)
Time (s)
Fit to Ailerons test data
Fit to Motor test data
Elevators deflection (δ_e)
Angle (rad)
Time (s)
Motor thrust (T)
Force (N)
Time (s)
Fit to Elevators test data

Figure 3.6. Bi-objective optimization identification concept

$$\min_{\mu \in \mathbb{R}^6} \left(\frac{1}{N_{ailerons}} \sum_{i=1}^{N_{ailerons}} \left[C_l(t_i) - \hat{C}_l(t_i, \mu) \right]^2, \frac{1}{N_{rudder}} \sum_{i=1}^{N_{rudder}} \left[C_l(t_i) - \hat{C}_l(t_i, \mu) \right]^2 \right) \tag{3.19}$$

where $\mu = \left[C_{l_0}, C_{l_\beta}, C_{l_p}, C_{l_r}, C_{l_{\delta_a}}, C_{l_{\delta_r}}\right]$. A total of six full optimization processes are required to obtain the complete set of solutions for the aerodynamic model. A decision making stage will complete the methodology. In that stage, exhaustive analysis of the aircraft behavior in the different tests must be made to determine the best approximation for each coefficient model.

3.5 Simulation Results

In [49] an initial approach on the identification of the Kadett 2400 aircraft model was performed. A MOOD strategy was also employed to achieve the aerodynamic stability and control derivatives. However, no wind estimation was made for compensating the sampled data. That work represents our starting point for this paper. Particularly, the models obtained in [49] are employed here to perform the simulations.

A simulation environment has been created as a validation tool in which the aircraft model can be subjected to different winds. Those winds are always known by the user, but the measured variables are GPS-like, in the sense that they refer to the Earth's surface and not to the air. In this way, if the estimated wind is similar to the one subjected to the model, it may be concluded that the technique successfully reached its objective. The spMODE algorithm is being employed. The decision space has been set as indicated in Eq. (3.11) but with the particularity of a maximum wind magnitude of 20 m/s. Different wind directions have been tested with similar results.

3.5.1 Constant Wind Simulations

In this first simulation, longitudinal and lateral experiments were conducted in which actuators were used independently to excite the system. A constant wind of 5 m/s with an elevation of $-20\,^\circ$ and a direction from North to South (*i.e.* $180\,^\circ$) was incorporated as the true wind. After the optimizer has completed the maximum number of generations it provides a set of solutions in approximation to the Pareto set (Θ^*). Table 3.2 shows some values extracted from the set of solutions. First column in Table 3.2 gives the mean average of the set of solutions obtained for each experiment. It is a three-component vector that represents the wind vector $\mathbf{W} = (|\mathbf{W}|, \zeta, \xi)$ where $|\mathbf{W}|$ is expressed in m/s and ζ and ξ are expressed in rad. Values in the second column express the standard deviations of the whole set. The third and fourth columns give the absolute and relative errors of each component of the wind estimate.

Table 3.2. Simulation results for constant wind

Test type	**W mean**	**W standard deviation**	**W absolute error**	**W relative error**
	$\begin{bmatrix} m/s \\ rad \\ rad \end{bmatrix}$	$\begin{bmatrix} m/s \\ rad \\ rad \end{bmatrix}$	$\begin{bmatrix} m/s \\ rad \\ rad \end{bmatrix}$	$\begin{bmatrix} \% \\ \% \\ \% \end{bmatrix}$
Elevators test	$\begin{bmatrix} 4.998988 \\ -0.353133 \\ 3.141773 \end{bmatrix}$	$\begin{bmatrix} 0.004948 \\ 0.000690 \\ 0.002964 \end{bmatrix}$	$\begin{bmatrix} 0.001012 \\ 0.004067 \\ 0.000181 \end{bmatrix}$	$\begin{bmatrix} 0.020242 \\ 1.165153 \\ 0.005754 \end{bmatrix}$
Motor test	$\begin{bmatrix} 4.947027 \\ -0.358709 \\ 3.140767 \end{bmatrix}$	$\begin{bmatrix} 0.031492 \\ 0.024524 \\ 0.012347 \end{bmatrix}$	$\begin{bmatrix} 0.052973 \\ 0.009643 \\ 0.000826 \end{bmatrix}$	$\begin{bmatrix} 1.059466 \\ 2.762445 \\ 0.026283 \end{bmatrix}$
Ailerons test	$\begin{bmatrix} 4.997899 \\ -0.349612 \\ 3.140767 \end{bmatrix}$	$\begin{bmatrix} 0.007045 \\ 0.002230 \\ 0.000613 \end{bmatrix}$	$\begin{bmatrix} 0.002101 \\ 0.000546 \\ 0.000826 \end{bmatrix}$	$\begin{bmatrix} 0.042013 \\ 0.156511 \\ 0.026277 \end{bmatrix}$
Rudder test	$\begin{bmatrix} 5.005375 \\ -0.351246 \\ 3.142247 \end{bmatrix}$	$\begin{bmatrix} 0.001417 \\ 0.000841 \\ 0.000307 \end{bmatrix}$	$\begin{bmatrix} 0.005375 \\ 0.002181 \\ 0.000654 \end{bmatrix}$	$\begin{bmatrix} 0.107498 \\ 0.624692 \\ 0.020826 \end{bmatrix}$

It can be seen that the estimation process converges to the actual wind with little error. It is interesting to see how, even for a constant wind and a known structure of the model, that estimated winds are not unique but a cloud of points very close to the real one. Two reasons may lead to this situation. First, the optimization problem has not been fully converged to the optimum value. Second, there are other (very similar) winds that explain discrepancies in the identification process as well as the real one. An observability issue can explain these discrepancies. This issue is fully dependent on the type of experiment used for the estimation. As an example, in an elevator test in which the aircraft does not change yaw orientation, the optimizer does not observe how the wind azimuth affects the identification errors. As a result, there will be a whole set of winds with different azimuth angles explaining, as well as the real one, those identification errors in the aerodynamic coefficients. As will be later seen, this situation leads to a poor estimation of the wind in more realistic simulations in the case of longitudinal models.

3.5.2 Variable Wind Simulations

A second set of simulations were conducted in which the wind was modeled as a sinusoidal signal of $\left[5 + sin(\frac{2\pi}{10}t)\right]$ m/s, with a band limited white noise added to the azimuth and elevation angles. The nominal elevation ζ was again $-20\,^{\circ}$ while two different azimuth angles ξ were simulated: $180\,^{\circ}$ and $270\,^{\circ}$. In this case, a cloud of solutions around the nominal wind was expected, since there was no constant wind during the experiments.

Table 3.3 shows results for the north wind direction and Table 3.4 shows results for the east wind. Both tables include the same type of values shown in Table 3.2 but with different results. First, although the estimation errors were higher, the procedure is capable of estimating the simulated wind when lateral experiments were carried out. It is interesting to see how the error in the rudder test is higher than that observed in the ailerons test. This can also be explained by the variance in observability depending on the type of experiment. When a deflection in the ailerons is applied, the aircraft orientation relative to the wind vector covers a sufficiently wide range of values. On the contrary, when deflections are applied to the tail rudder, it is mostly the heading angle that changes, leading to a less observable experiment in terms of wind. Following the same reasoning, the first two rows of Table 3.3 and Table 3.4 represent the set of solutions proposed by the algorithm when the longitudinal tests are used in the procedure. The longitudinal experiments do not lead to a good estimation of the wind for any of the tested wind directions. Not only were the errors in the estimation higher, but the standard deviation of each set of solutions was also wider. To support this statement Fig. 3.7 has been included. As the graph shows, the real wind is among the cloud of solutions obtained by the MOEA: however, the cloud is so spread out that a reliable wind estimation cannot be extracted. Again, this means that there is a whole set of winds with explaining, as well as the real one, those identification errors in the aerodynamic coefficients. By comparing Table 3.3 and Table 3.4 we can see that similar results were obtained for different wind azimuth angles. Hence, we can conclude that the longitudinal models suffer an observability issue in terms of wind estimation. This issue is also present in the rudder experiments but to a lesser degree. Therefore, the longitudinal experiments carried out here are inadequate for wind estimation purposes.

The implementation of experiments in which different actuators are excited simultaneously is proposed as a possible solution. Such a test could be used for wind estimation as well as for the adjustment of all the stability and control derivatives. As a proof of concept, this type of test has been performed in simulation.

Figure 3.8 illustrates how the elevators, ailerons, and rudder are used during the experiment. Although this approach for applying input steps is not common, it is necessary due to time requirements. Time-skewed doublets could also be used here, but the authors want to highlight that the assumption of constant wind weakens as the duration of the experiments increase. In both cases, if the wind estimation is successful, it will be possible to use this information to correct the aerodynamic variables from the data and then accomplish an identification of both lateral and longitudinal models. The last row in Table 3.3 and Table 3.4 shows the wind estimation obtained for the same wind conditions as the previous cases. As can be observed, the wind estimation is accurately achieved this time.

Finally, Fig. 3.9 and Fig. 3.10 show the model identification results after the wind correction. When performing a simulation, models were already being used for

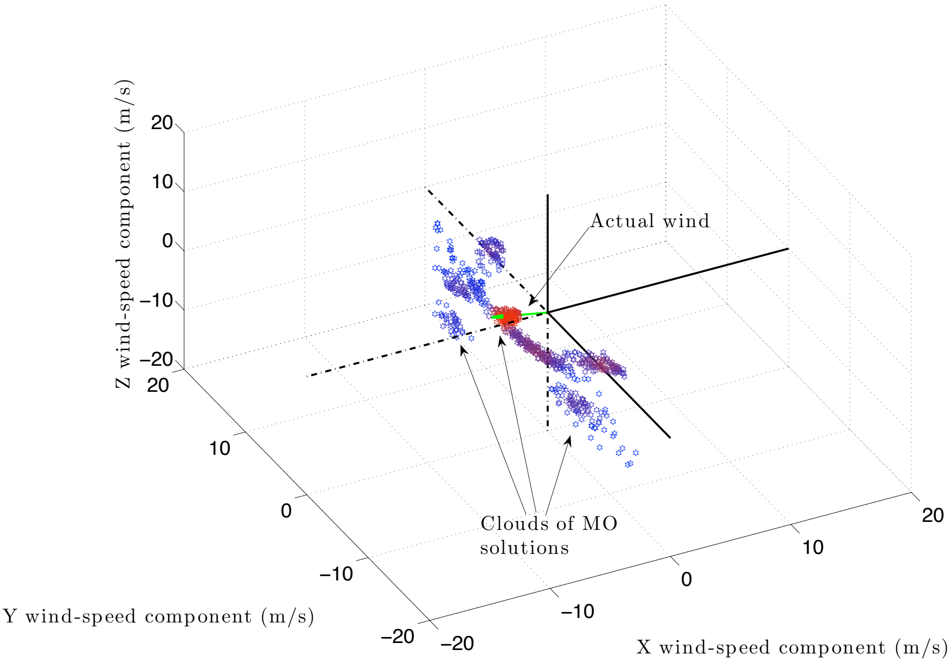


Figure 3.7. Wind estimation results from simulation of elevators experiment with fluctuating wind

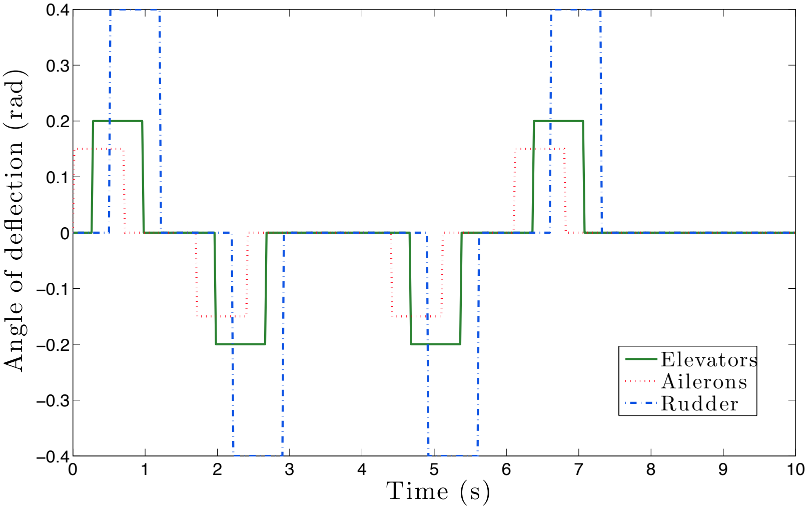


Figure 3.8. Simulated experiment of multiple actuators

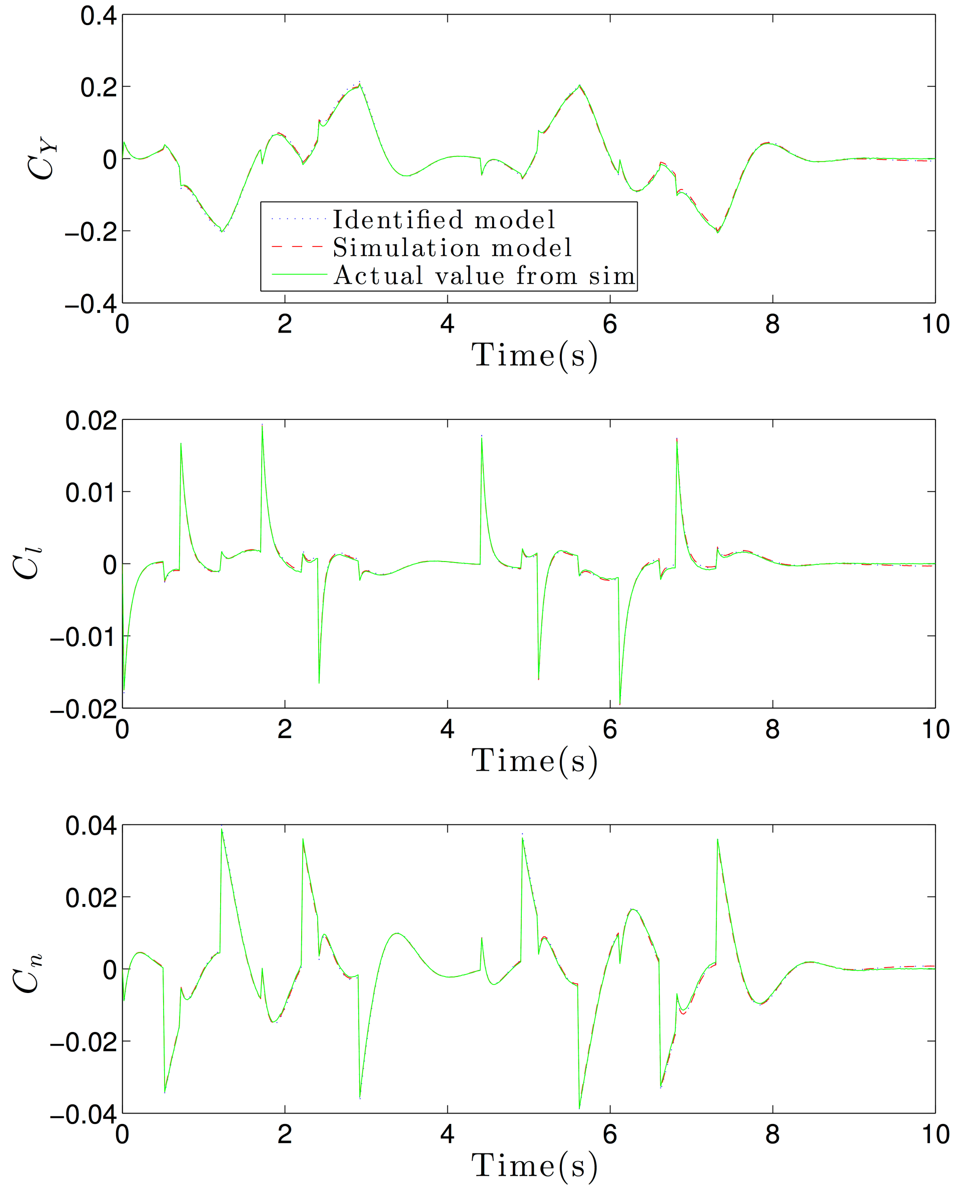


Figure 3.9. Estimated lateral aerodynamic coefficients with simulation of multiple actuators

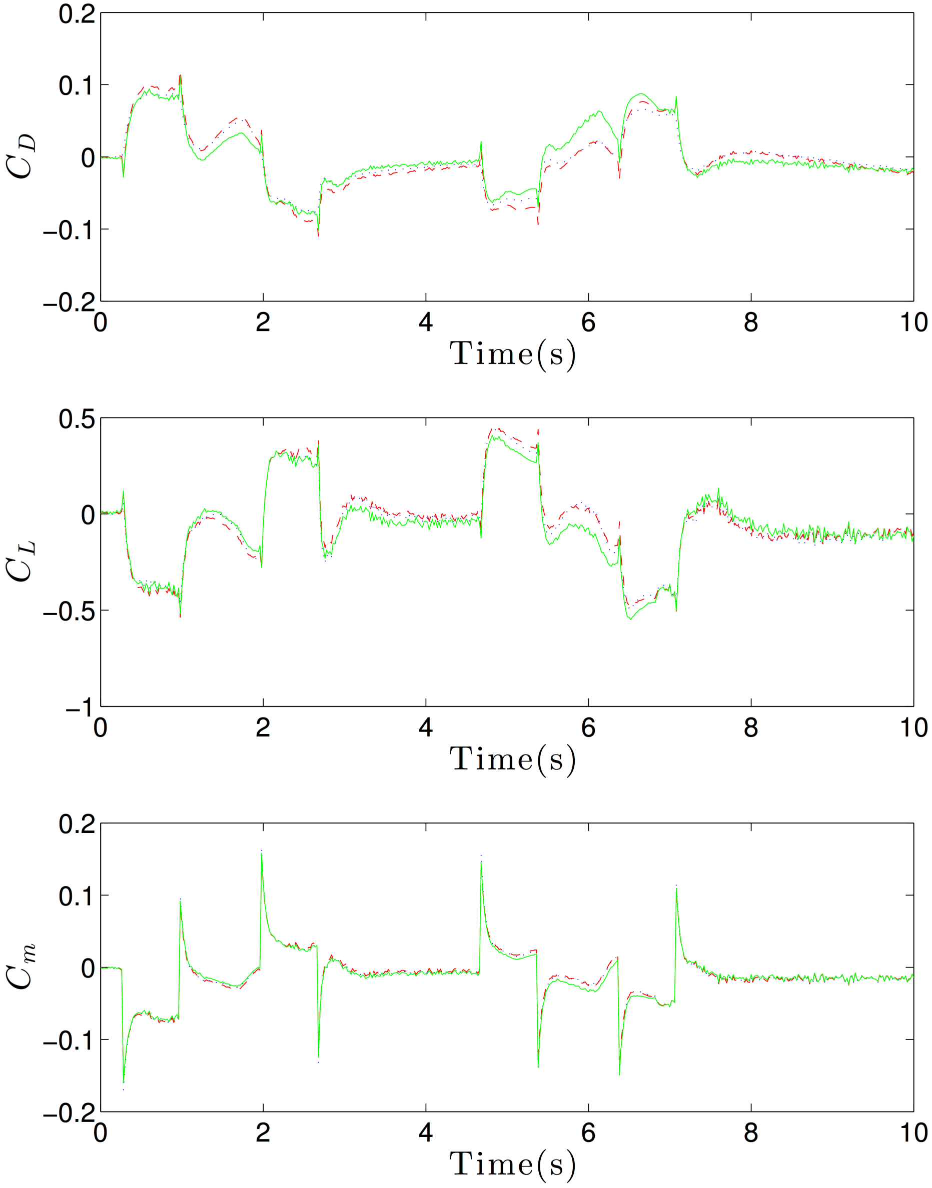


Figure 3.10. Estimated longitudinal aerodynamic coefficients with simulation of multiple actuators

Table 3.3. Simulation results for variable wind

Test type	**W mean**	**W standard deviation**	**W absolute error**	**W relative error**
	$\begin{bmatrix} \mathrm{m/s} \\ \mathrm{rad} \\ \mathrm{rad} \end{bmatrix}$	$\begin{bmatrix} \mathrm{m/s} \\ \mathrm{rad} \\ \mathrm{rad} \end{bmatrix}$	$\begin{bmatrix} \mathrm{m/s} \\ \mathrm{rad} \\ \mathrm{rad} \end{bmatrix}$	$\begin{bmatrix} \% \\ \% \\ \% \end{bmatrix}$
Elevators test	$\begin{bmatrix} 7.966485 \\ -0.06888 \\ 3.406746 \end{bmatrix}$	$\begin{bmatrix} 2.96925 \\ 0.300134 \\ 1.105017 \end{bmatrix}$	$\begin{bmatrix} 2.966485 \\ 0.280186 \\ 0.265153 \end{bmatrix}$	$\begin{bmatrix} 59.329696 \\ 80.267362 \\ 8.440095 \end{bmatrix}$
Motor test	$\begin{bmatrix} 17.247653 \\ -0.554826 \\ 2.807634 \end{bmatrix}$	$\begin{bmatrix} 3.462794 \\ 0.663819 \\ 2.150964 \end{bmatrix}$	$\begin{bmatrix} 12.247653 \\ 0.20576 \\ 0.333959 \end{bmatrix}$	$\begin{bmatrix} 244.953066 \\ 58.945868 \\ 10.630239 \end{bmatrix}$
Ailerons test	$\begin{bmatrix} 5.268963 \\ -0.347613 \\ 3.038388 \end{bmatrix}$	$\begin{bmatrix} 0.011684 \\ 0.002634 \\ 0.022588 \end{bmatrix}$	$\begin{bmatrix} 0.268963 \\ 0.001453 \\ 0.103204 \end{bmatrix}$	$\begin{bmatrix} 5.379259 \\ 0.416195 \\ 3.285098 \end{bmatrix}$
Rudder test	$\begin{bmatrix} 5.521515 \\ -0.296587 \\ 2.720627 \end{bmatrix}$	$\begin{bmatrix} 0.033722 \\ 0.008135 \\ 0.021629 \end{bmatrix}$	$\begin{bmatrix} 0.521515 \\ 0.052479 \\ 0.420966 \end{bmatrix}$	$\begin{bmatrix} 10.430297 \\ 15.034222 \\ 13.399754 \end{bmatrix}$
Multiple actuators test	$\begin{bmatrix} 5.106339 \\ -0.393798 \\ 2.973994 \end{bmatrix}$	$\begin{bmatrix} 0.025308 \\ 0.00435 \\ 0.010241 \end{bmatrix}$	$\begin{bmatrix} 0.106339 \\ 0.044732 \\ 0.167599 \end{bmatrix}$	$\begin{bmatrix} 2.126776 \\ 12.814694 \\ 5.334845 \end{bmatrix}$

each aerodynamic coefficient. Therefore, a distinction can be made between the actual value of the aerodynamic coefficient measured during simulation (continuous line) and the value obtained for each aerodynamic coefficient by taking the corrected regressors and using the simulation model to make a calculation (dashed line). The reader should note that these two values are only equal if the wind is correctly estimated and hence the regressors are perfectly corrected. The third variable depicted in Fig. 3.9 and Fig. 3.10 is the value of the coefficient that the newly identified model proposes with the corrected regressors (dotted line). It can be seen that all three are nearly superimposed, meaning a successful wind estimation and adjustment of the model parameters. Results show that both objectives can be accomplished simultaneously with this type of experiment.

3.6 Experimental Results

3.6.1 Flight Tests

An aircraft which is maintaining constant heading and altitude, at a constant speed and with level wings (zero roll angle), is considered to be in steady flight.

Table 3.4. Simulation results for variable wind

Test type	**W mean**	**W standard deviation**	**W absolute error**	**W relative error**
	m/s rad rad	m/s rad rad	m/s rad rad	% % %
Elevators test	4.344096 −0.521659 4.314185	1.408924 0.395895 1.187131	0.655904 0.172594 0.398204	13.118077 49.444434 8.450145
Motor test	11.157674 −0.07673 3.53739	4.040244 0.350458 2.706765	6.157674 0.272336 1.174999	123.153488 78.018598 24.934243
Ailerons test	5.129344 −0.446838 4.461761	0.030426 0.005952 0.002271	0.129344 0.097773 0.250628	2.586887 28.009761 5.318495
Rudder test	4.74749 −0.304223 3.917106	0.83811 0.075543 0.110649	0.25251 0.044843 0.795283	5.050198 12.846522 16.87643
Multiple actuators test	5.08521 −0.275279 4.438785	0.065749 0.007701 0.005862	0.08521 0.073787 0.273604	1.704192 21.138398 5.806064

In the absence of disturbances, the pilot does not need to make any corrections to maintain this steady state.

To obtain data that can be employed in adjusting the aerodynamic parameters, step-input experiments have been performed. Thus, starting always from a steady-state flight such as the one described in the previous paragraph, each system input was manipulated separately and, after manipulation, the aircraft was left to evolve naturally, until the pilot deemed it appropriate to recover the aircraft. Each experiment was performed twice to obtain different data sets for identification and validation. It should be noted that, in the absence of a sensor capable of measuring airspeed, all maneuvers described below were carried out against the wind. This restriction was imposed on the pilot because of two reasons. First, to reduce variability between the flight tests. And second, due to the better wind estimation results obtained during the simulation phase.

The flight plan provided to the pilot before beginning the experiments was:

1. Stable flight:

   (a) Adjust ailerons and rudder and level wings.

   (b) Set the motor load around 50%.

   (c) Adjust elevators until the altitude remained constant without touching the control stick.

2. Elevators up and down to create a positive plus negative step sequence.
3. Repeat step 1.
4. Ailerons side to side in the appropriate frequency to avoid extreme rotations. First in one direction and then in the opposite direction.
5. Repeat step 1.
6. Tail rudder side to side. First in one direction and then in the opposite direction.
7. Repeat step 1.
8. Positive and negative steps in motor load. Sequence: 50%-100%-50%-0%-50%
9. Repeat the entire flight plan a second time.

Figure 3.11 and Fig. 3.12 show the evolution of the longitudinal and lateral variables during the elevator and aileron excitation tests, respectively. As shown, when a longitudinal input is activated, the remaining longitudinal variables are also activated, which finally produces variations in the symmetrical aerodynamic coefficients. This same behavior can be observed for the asymmetrical variables. All these variations can be collected and used to estimate the aerodynamic stability and control derivatives. As a final remark, Fig. 3.11 and Fig. 3.12 show the values $\hat{p} = \frac{b}{2V_0}p$, $\hat{q} = \frac{\bar{c}}{2V_0}p$, $\hat{r} = \frac{b}{2V_0}r$, $\hat{V} = \frac{1}{V_0}\Delta V_{air}$, and $\hat{\dot{\alpha}} = \frac{\bar{c}}{2V_0}\dot{\alpha}$. These are the values that multiply the stability and control derivatives in the aerodynamic coefficient models and, therefore, the regressors used in the identification procedure.

3.6.2 Wind Estimation Results

Separate step-input flight tests were performed to estimate the aerodynamic coefficient models. No experiment was carried out in which multiple control surfaces were employed at the same time. Thus, only lateral experiments were used in the wind estimation. For this reason, the information obtained from those tests is used to correct the longitudinal experiments as well.

Table 3.5 shows the set of solutions that the multi-objective optimizer converges to during the ailerons and rudder tests, respectively. Although wind direction and elevation vary slightly among sets, similar winds are obtained for every experiment. A population density criterion has been used in the final selection of one of the winds among all the solutions set. A sphere of radius $R = \max\{d_{ij}\}/20$ m/s

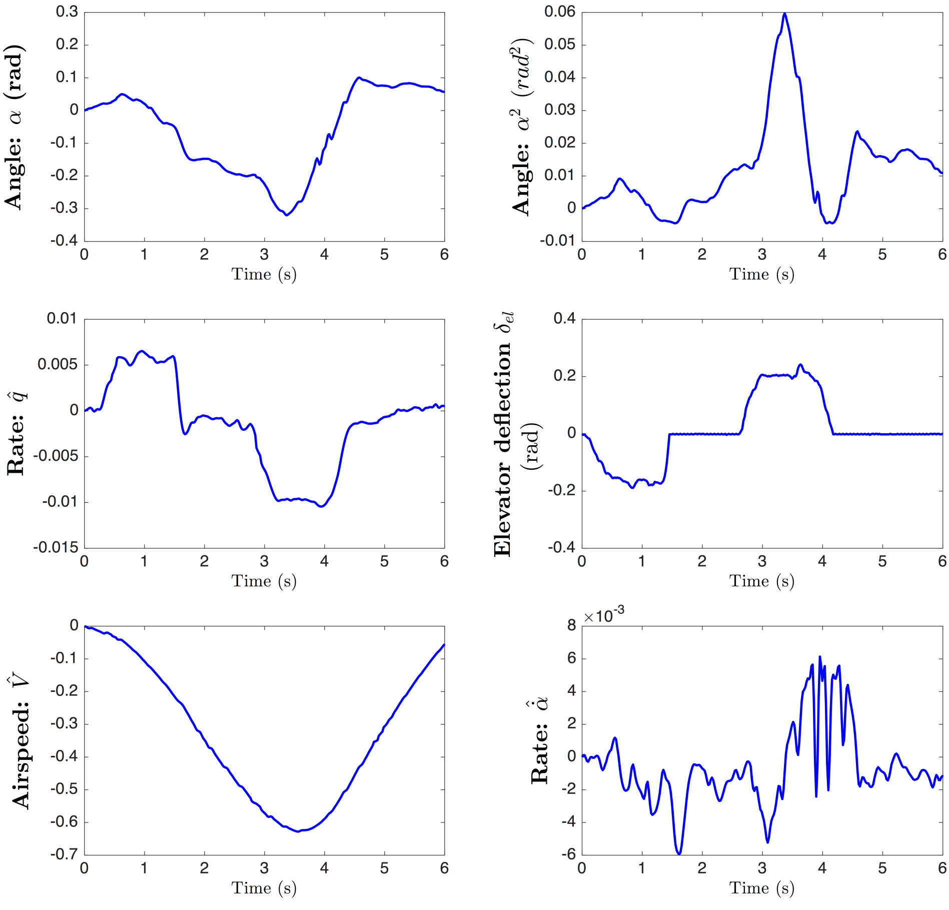


Figure 3.11. Flight Test. Longitudinal variables evolution in an elevators test

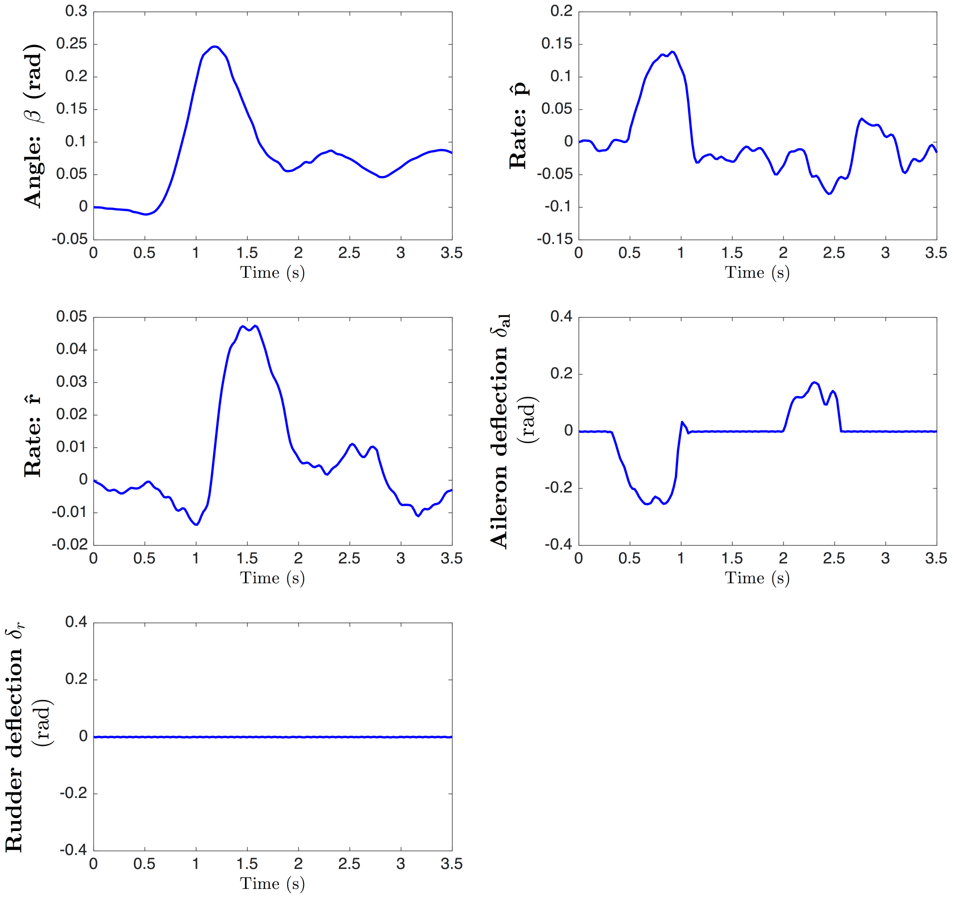


Figure 3.12. Flight Test. Lateral variables evolution in an ailerons test

around each solution has been placed for that purpose. The solution whose sphere contained the largest number of enclosed points was selected. That solution has been included in the third column of Table 3.5.

Table 3.5. Wind estimation experimental results

Test type	**W mean** $\begin{bmatrix} \text{m/s} \\ \text{rad} \\ \text{rad} \end{bmatrix}$	**W standard deviation** $\begin{bmatrix} \text{m/s} \\ \text{rad} \\ \text{rad} \end{bmatrix}$	**Chosen W** $\begin{bmatrix} \text{m/s} \\ \text{rad} \\ \text{rad} \end{bmatrix}$
Ailerons test 1	$\begin{bmatrix} 8.490372 \\ -0.216576 \\ 5.437922 \end{bmatrix}$	$\begin{bmatrix} 0.906376 \\ 0.211093 \\ 0.027067 \end{bmatrix}$	$\begin{bmatrix} 8.283345 \\ -0.056781 \\ 5.449877 \end{bmatrix}$
Rudder test 1	$\begin{bmatrix} 9.811319 \\ -1.041670 \\ 5.352327 \end{bmatrix}$	$\begin{bmatrix} 1.789285 \\ 0.182896 \\ 0.105825 \end{bmatrix}$	$\begin{bmatrix} 10.133260 \\ -1.390814 \\ 5.992735 \end{bmatrix}$
Ailerons test 2	$\begin{bmatrix} 8.133613 \\ -1.071944 \\ 5.567295 \end{bmatrix}$	$\begin{bmatrix} 0.443366 \\ 0.200113 \\ 0.033953 \end{bmatrix}$	$\begin{bmatrix} 8.133613 \\ -1.071944 \\ 5.567295 \end{bmatrix}$
Rudder test 2	$\begin{bmatrix} 10.299536 \\ -0.869243 \\ 5.465950 \end{bmatrix}$	$\begin{bmatrix} 1.801400 \\ 0.168777 \\ 0.122687 \end{bmatrix}$	$\begin{bmatrix} 11.454439 \\ -1.249881 \\ 5.894774 \end{bmatrix}$

After selection, the MSE was computed in a cross validation analysis. Fewer quadratic errors were found for such validations than before correcting data. As an example, Fig. 3.13 shows the identification and validation mean squared errors found for the coefficient C_n with two different sets of data. Four groups of two bars are shown in Fig. 3.13. Given that there are two sets of data, each bar in a group represents the model identified using one of those two sets. In particular, the striped bar always represents the error for the model identified with Set 1, and the dotted bar the error for the model obtained with Set 2. Hence, the two groups on the right show the quadratic error found in the identification and the cross validation before the data was corrected with wind information. The two groups on the left give quadratic errors for identification and cross validation, once wind information was incorporated. It can be observed that quadratic errors have been reduced significantly (at least three times lower).

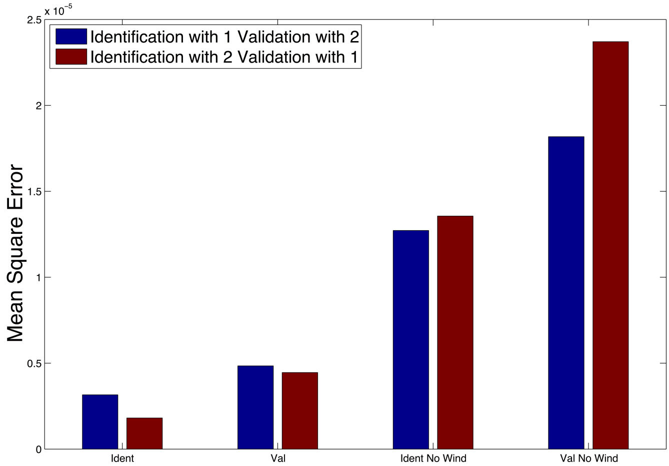


Figure 3.13. $\mathbf{C_n}$ model mean squared errors: cross validation before and after wind correction

3.6.3 Identification Results

Once information about the wind acting during the flight tests is available, any airspeed-dependent variable may be corrected. With all the experimental data corrected, the process of finding the stability and control derivatives for each aerodynamic model began. Figure 3.14 and Fig. 3.15 show the Pareto fronts constituted by the possible models found by the algorithm for each aerodynamic coefficient (hexagonal stars front). As can be seen, the better an experiment is fitted by a model, the more errors it obtains for a second test. This is why the person in charge of identifying the aircraft model cannot be satisfied after just one test, but should use the model with data from various experiments.

Moreover, testing experiments in a multi-objective optimization, instead of combining them in a mono-objective minimization, gives the main following advantages:

- Optimizing objectives separately results in solutions that could not be reachable if other optimization techniques are used (see Fig. 3.5).
- Using multi-objective optimization involves the selection of a solution among others, which gives the designer the power to define the importance of each experiment *a posteriori*, basing that definition on the requirements and the observed performance.
- The resultant Pareto front shows how good the different models are for each experiment. Thus, the designer may obtain an idea of how good the collected

data is and so decide which are the requirements that should be satisfied in the final model.

- Flight conditions do not depend on the previous experiment.
- Duration of the experiments is the minimum required. This improves reliability on the wind estimation procedure.
- Metrics other than MSE can be used in the optimization.
- It is possible to add as many objectives as desired in the identification process. This means that comparing tests of the same type (*e.g.* two aileron tests and two rudder tests) is also possible, and this may prove to be a good practice for reducing variability.

For example, if the C_l coefficient is considered, it can be observed that the aileron tests produce a much better approximation than the rudder tests (see Fig. 3.14). This fact, which can be deduced from the mean squared error values, is also logical, since the ailerons introduce a moment about the roll axis. In this case, the designer should probably prefer models that fit better this type of experiment over models that do a better job with rudder tests.

With the intention of comparing solutions before and after wind correction, a second Pareto front was added to Fig. 3.14 and Fig. 3.15(triangles front). This front is the result of evaluating the Pareto set of solutions found by the MO algorithm when non-corrected flight data is used in the identification. As an interesting observation, Fig. 3.14 and Fig.3.15 illustrate that each Pareto front obtained with non-corrected flight data (dots front) is dominated by the corresponding Pareto front obtained with corrected flight data (hexagonal stars front). This means that the MSE of every model becomes smaller after correcting the flight data with the estimated wind.

The MOP ends with a multi-criteria decision making (MCDM) stage. In this case, a solution among the Pareto set will give us the final model parameters for each aerodynamic coefficient. Decision making is commonly a difficult task when many objectives and decision variables are involved. It is widely accepted that visualization tools are valuable and provide a meaningful method to analyze the Pareto front and take decisions [25]. Possibly the most common choices for Pareto front visualization and analysis are: scatter diagrams, parallel coordinates [15], and level diagrams [2, 37]. In this work a level diagrams tool has been used to analyze Pareto fronts and Pareto sets and to decide a particular model for each coefficient. The final decision was made by taking into account the distance to the ideal solution, generated from the minimum values for each objective in the calculated Pareto front. This distance is a widely used metric in MCDM because it correctly represents the existing trade off among objectives. The squares on Fig. 3.14 are the selected models for each of the lateral aerodynamic coefficients.

Those models represent a compromise between a situation in which the ailerons deflection is modified and a situation in which that modification is suffered by the tail rudder. This fact can be checked in Fig. 3.16. The graph shows the approximation given by the chosen model with validation data. Two more models from the Pareto set (circles on Fig. 3.14) have also been included in Fig. 3.16 (edge model 1 and edge model 2). Those two models have been named edge models here, and represent the best approximation in terms of MSE for the ailerons and rudder experiments separately (see [50]). The chosen MO solution (dotted curve) represents a good intermediate approximation in both situations. Finally, Table 3.6 presents validation results of every identified aerodynamic coefficient. Longitudinal coefficients have been subjected to validation data obtained in longitudinal experiments and, equally, lateral coefficients have been evaluated with validation sets obtained in lateral experiments. MSE presents similar values in identification and validation and so the identification may be taken as a success.

Table 3.6. Validation MSE

Aerodynamic coefficient	**Elevators experiment**	**Motor experiment**	**Ailerons experiment**	**Rudder experiment**
$\mathbf{C_D}$	0.000507759	0.000663917	-	-
$\mathbf{C_L}$	0.0058858	0.00145734	-	-
$\mathbf{C_Y}$	-	-	0.00166596	0.00122309
$\mathbf{C_l}$	-	-	$1.65789e-05$	$3.40064e-06$
$\mathbf{C_m}$	0.000326971	0.000167846	-	-
$\mathbf{C_n}$	-	-	$3.21581e-05$	$2.70232e-05$

3.7 Conclusions

A two-step identification technique for aerodynamic models of micro-air vehicles (MAV) in the absence of air-data sensors is presented. In the first step, a multi-objective optimization procedure is proposed to estimate wind during the flight experiments. A simulation environment that includes a MAV model that can be subjected to constant and variable winds was used to confirm the estimate process. Conditions in different flight tests in which one or more system inputs were excited were simulated, and after acquiring any necessary data from the simulations, the wind estimation technique was applied. Several conclusions can be extracted from these simulations. Firstly, under ideal conditions, wind estimation is successfully achieved in any experiment. Secondly, it can be concluded that only lateral experiments offer enough information to enable wind estimation under realistic conditions. Aircraft orientation relative to the wind azimuth does not vary during longitudinal experiments. Therefore, wind observability is significantly reduced. The same reason is reflected in the improved performance detected during

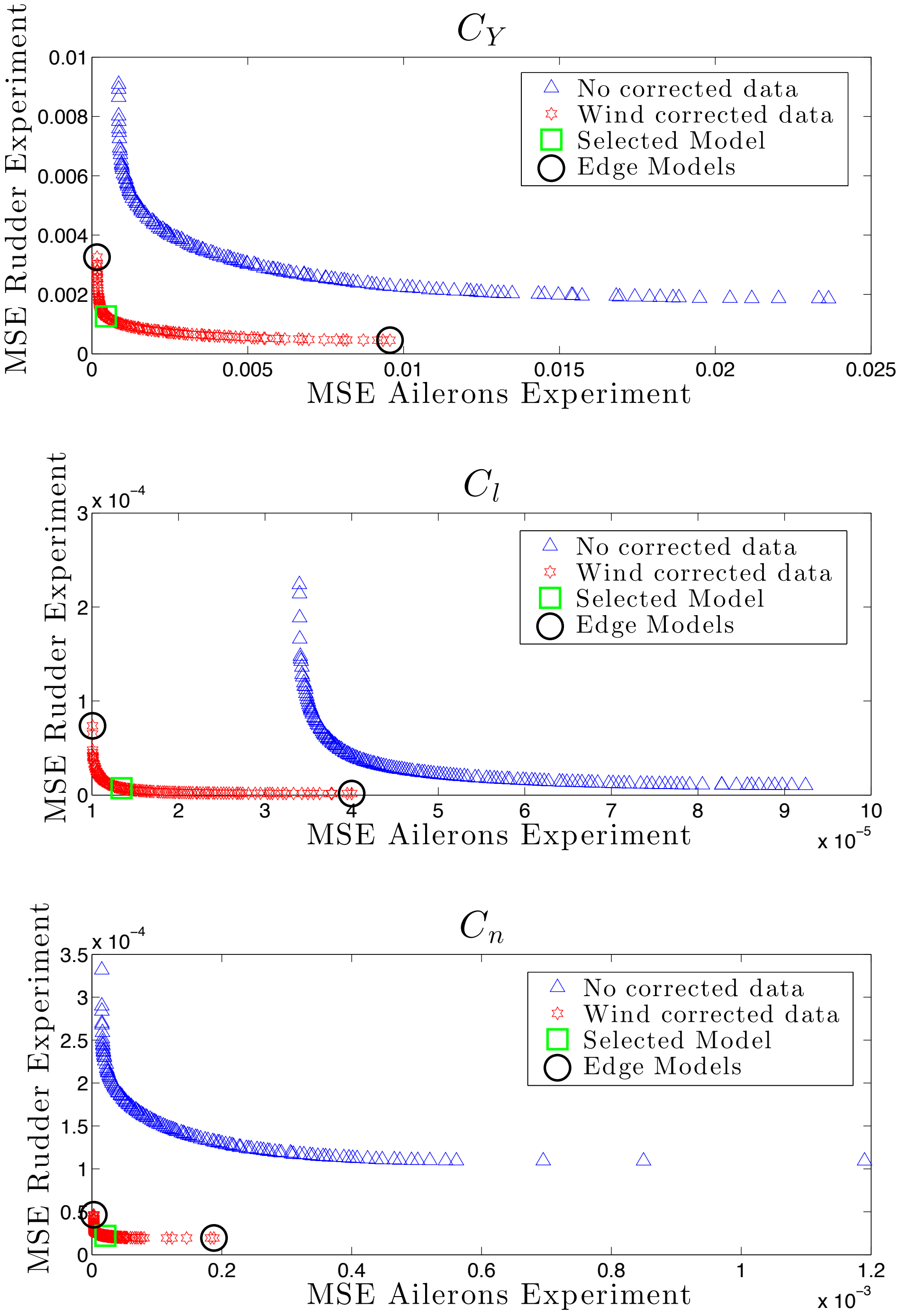


Figure 3.14. MO Model Identification: control and stability derivatives for Lateral Aerodynamic Models

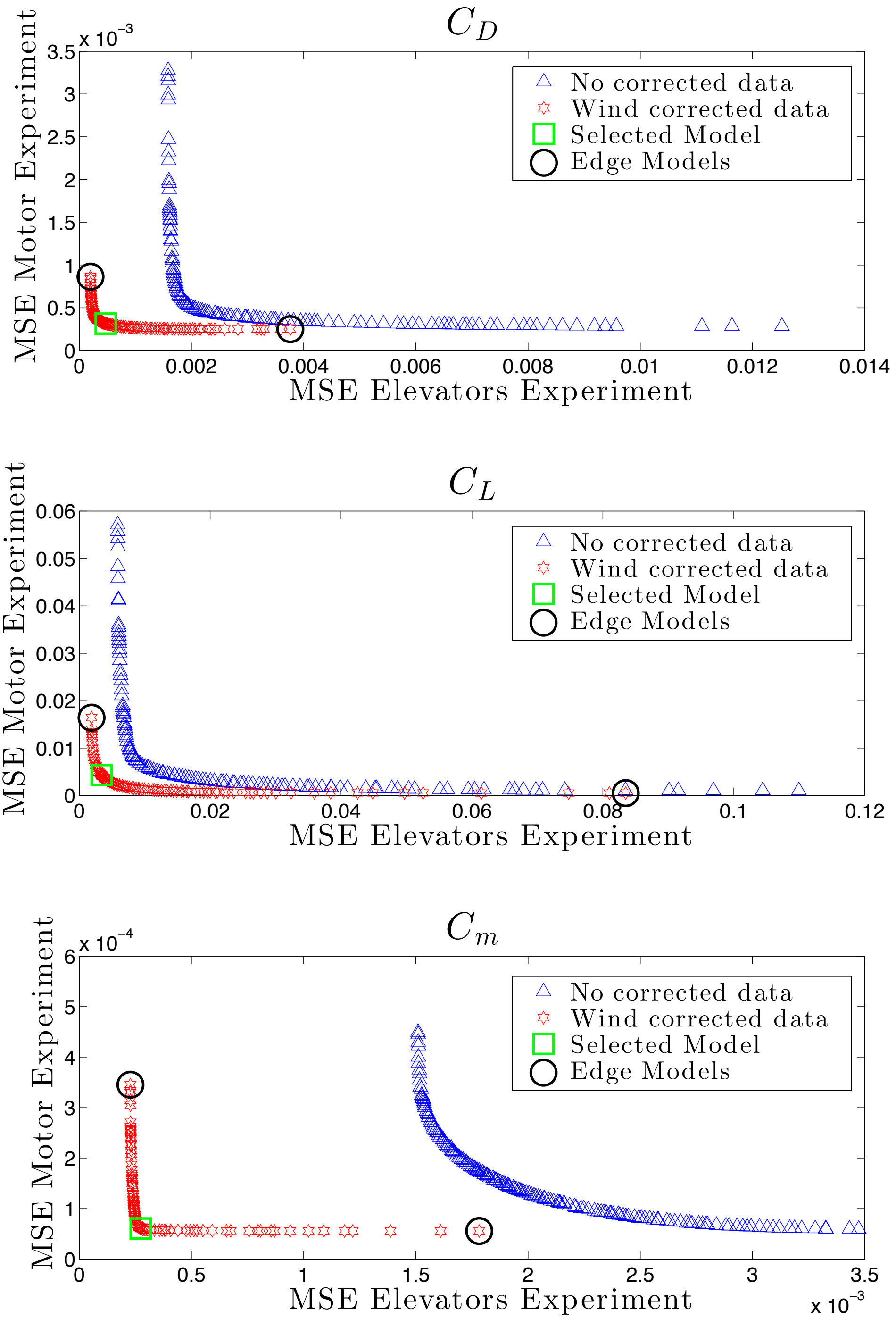


Figure 3.15. MO Model Identification: control and stability derivatives for Longitudinal Aerodynamic Models

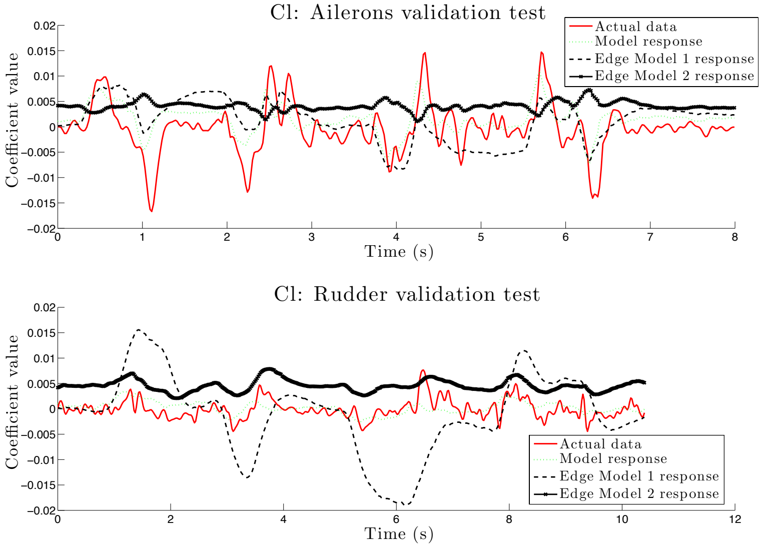


Figure 3.16. $\mathbf{C}_l$ coefficient validation results

the aileron experiments when compared with rudder experiments. Observability is improved when a wider range of orientation values are covered. For this reason, mixed experiments, *i.e.*, experiments that excite longitudinal and lateral variables simultaneously, achieved better results for wind estimation. As shown in section 3.5.2, this type of experiments can theoretically be used to estimate the wind and identify any of the aerodynamic coefficient models. But this fact has only been checked on simulation.

The same wind estimation procedure was applied to real flight data for lateral experiments. The obtained results infer that the wind was estimated and that the information obtained can be used to correct airspeed dependent measurements. As a final remark, the authors want to highlight that the estimation technique presented in this work is not intended to replace air-data sensors (whenever available). However, in those cases when no information at all can be used, a rough estimation of wind speed can significantly improve model quality. In addition, a similar multi-objective optimization approach might also be employed when partial airspeed information is available. That information could be incorporated in the optimization problem in order to improve airspeed measurements.

In the second step of the methodology, multi-objective optimization is again proposed to take advantage of the available flight data. The presented approach enables diverse experiments to be utilized, so that adjusting model parameters becomes, in reality, a multi-objective problem. This approach enabled us to obtain a compromise model that suited some flight situations without losing much performance in others. Furthermore, the visualization of the model fitness for several trials provides an idea of the quality of the obtained data and of the selected model structure. Although mean squared error has been used here, using a heuristic optimizer also enables the use of other performance indicators. For example, the mean absolute error is normally more meaningful to engineers because it has the same magnitude as the variable being modeled.

Bibliography

[1] Multi-criteria decision making. In *Evolutionary Algorithms for Solving Multi-Objective Problems*, Genetic and Evolutionary Computation Series, pages 515–545. Springer US, 2007.

[2] X. Blasco, J.M. Herrero, J. Sanchis, and M. Martínez. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Special Issue on Industrial Applications of Neural Networks 10th Engineering Applications of Neural Networks 2007*, 178(20):3908–3924, October 2008.

[3] P.P. Bonissone, R. Subbu, and J. Lizzi. Multicriteria decision making (mcdm): a framework for research and applications. *Computational Intelligence Magazine, IEEE*, 4(3):48–61, August 2009.

[4] Guowei Cai, Ben M. Chen, Tong H. Lee, and Miaobo Dong. Design and implementation of a hardware-in-the-loop simulation system for small-scale UAV helicopters. *Mechatronics*, 19(7):1057–1066, 2009.

[5] Am Cho, Jihoon Kim, Sanghyo Lee, and Changdon Kee. Wind estimation and airspeed calibration using a UAV with a single-antenna GPS receiver and pitot tube. *Aerospace and Electronic Systems, IEEE Transactions on*, 47(1):109–117, 2011.

[6] Carlos A Coello Coello. Evolutionary multi-objective optimization: a historical view of the field. *Computational Intelligence Magazine, IEEE*, 1(1):28–36, 2006.

[7] Carlos A Coello Coello. An introduction to multi-objective particle swarm optimizers. In *Soft Computing in Industrial Applications*, pages 3–12. Springer, 2011.

[8] R. Dennis Cook. *Residuals and influence in regression.* Monographs on statistics and applied probability. Chapman and Hall, New York, 1982.

[9] Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[10] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. Differential evolution: a survey of the state-of-the-art. *Evolutionary Computation, IEEE Transactions on*, 15(1):4–31, 2011.

[11] John Dean, Scott Morton, David McDaniel, James Clifton, and David Bodkin. Aircraft stability and control characteristics determined by system identification of CFD simulations. American Institute of Aeronautics and Astronautics, August 2008.

[12] Marco Dorigo and Thomas Stützle. Ant colony optimization: overview and recent advances. In *Handbook of metaheuristics*, pages 227–263. Springer, 2010.

[13] Peter G. Hamel and Ravindra V. Jategaonkar. Evolution of flight vehicle system identification. *Journal of Aircraft*, 33(1):9–28, January 1996.

[14] Kenneth W. Iliff. Parameter estimation for flight vehicles. *Journal of Guidance, Control, and Dynamics*, 12(5):609–622, September 1989.

[15] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[16] Ravindra V. Jategaonkar. *Flight vehicle system identification: a time domain methodology*. Number v. 216 in Progress in astronautics and aeronautics. American Institute of Aeronautics and Astronautics, Reston, Va, 2006.

[17] Ravindra V. Jategaonkar, Dietrich Fischenberg, and Wolfgang Gruenhagen. Aerodynamic modeling and system identification from flight data-recent applications at DLR. *Journal of Aircraft*, 41(4):681–691, July 2004.

[18] Dervis Karaboga, Beyza Gorkemli, Celal Ozturk, and Nurhan Karaboga. A comprehensive survey: artificial bee colony (abc) algorithm and applications. *Artificial Intelligence Review*, 42(1):21–57, 2014.

[19] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of the 1995 IEEE international conference on neural networks*, pages 1942–1948, Piscataway, NJ., 1995. IEEE Service Center.

[20] Vladislav Klein and Eugene A. Morelli. *Aircraft system identification: theory and practice*, chapter 3, pages 27–75. AIAA Education Series. American Institute of Aeronautics and Astronautics, Inc., 2006.

[21] Vladislav Klein and Eugene A. Morelli. *Aircraft system identification: theory and practice*, chapter 9, pages 299–323. AIAA Education Series. American Institute of Aeronautics and Astronautics, Inc., 2006.

[22] Abdullah Konak, David W Coit, and Alice E Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.

[23] Wolfgang Krüll, Robert Tobera, Ingolf Willms, Helmut Essen, and Nora von Wahl. Early forest fire detection and verification using optical smoke, gas and microwave sensors. *Procedia Engineering*, 45:584–594, 2012.

[24] Jack W. Langelaan, Nicholas Alley, and James Neidhoefer. Wind field estimation for small unmanned aerial vehicles. *Journal of Guidance, Control, and Dynamics*, 34(4):1016–1030, July 2011.

[25] Alexander V Lotov and Kaisa Miettinen. Visualizing the Pareto frontier. In J Branke, Kalyanmoy Deb, Kaisa Miettinen, and R Slowinski, editors, *Multiobjective optimization*, volume 5252 of *Lecture Notes in Computer Science*, pages 213–243. Springer, Berling Heidelberg, 2008.

[26] Achille Messac, Amir Ismail-Yahaya, and Christopher A Mattson. The normalized normal constraint method for generating the pareto frontier. *Structural and multidisciplinary optimization*, 25(2):86–98, 2003.

[27] Achille Messac and Christopher A Mattson. Generating well-distributed sets of pareto points for engineering design using physical programming. *Optimization and Engineering*, 3(4):431–450, 2002.

[28] Efrén Mezura-Montes, Margarita Reyes-Sierra, and Carlos A Coello Coello. Multi-objective optimization using differential evolution: a survey of the state-of-the-art. In *Advances in differential evolution*, pages 173–196. Springer, 2008.

[29] Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12 of *International series in operations research & management science*, chapter Concepts, pages 5–36. Kluwer Academic Publishers, Boston, 1999.

[30] A. Mohamed, R. Clothier, S. Watkins, R. Sabatini, and M. Abdulrahim. Fixed-wing MAV attitude stability in atmospheric turbulence, part 1: Suitability of conventional sensors. *Progress in Aerospace Sciences*, 70:69–82, October 2014.

[31] Abdulghani Mohamed, Kevin Massey, Simon Watkins, and Reece Clothier. The attitude control of fixed-wing MAVS in turbulent environments. *Progress in Aerospace Sciences*, 66:37–48, April 2014.

[32] Eugene A. Morelli. Flight test maneuvers for efficient aerodynamic modeling. *Journal of Aircraft*, 49(6):1857–1867, November 2012.

[33] Eugene A. Morelli and Vladislav Klein. Application of system identification to aircraft at NASA langley research center. *Journal of Aircraft*, 42(1):12–25, January 2005.

[34] Eric R. Mueller. Hardware-in-the-loop simulation design for evaluation of unmanned aerial vehicle control systems. In *Proceedings of AIAA Modeling and Simulation Technologies Conference*, volume 1, pages 530–543, 2007.

[35] Ales Procházka, Jan Uhlír, P. W. J. Rayner, N. G. Kingsbury, and John J. Benedetto, editors. *Signal Analysis and Prediction.* Applied and Numerical Harmonic Analysis. Birkhäuser Boston, Boston, MA, 1998.

[36] Gilberto Reynoso-Meza. *Controller Tuning by Means of Evolutionary Multiobjective Optimization: a Holistic Multiobjective Optimization Design Procedure.* PhD thesis, Universidad Politecnica de Valencia, Valencia, Spain, June 2014.

[37] Gilberto Reynoso-Meza, Xavier Blasco, Javier Sanchis, and Juan M Herrero. Comparison of design concepts in multi-criteria decision-making using level diagrams. *Information Sciences*, 221(221):124–141, February 2013. doi:10.1016/j.ins.2012.09.049.

[38] Gilberto Reynoso-Meza, Javier Sanchis, and Xavier Blasco. Design, coding and implementation of a multiobjective optimization algorithm based on differential evolution with spherical pruning: applications for system identification and controller tuning. Master's thesis, 2009.

[39] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Miguel Martínez. Design of continuous controllers using a multiobjective differential evolution algorithm with spherical pruning. In Cecilia Chio, Stefano Cagnoni, Carlos Cotta, Marc Ebner, Anikó Ekárt, AnnaI. Esparcia-Alcazar, Chi-Keong Goh, JuanJ. Merelo, Ferrante Neri, Mike Preuß, Julian Togelius, and GeorgiosN. Yannakakis, editors, *Applications of Evolutionary Computation*, volume 6024 of *Lecture Notes in Computer Science*, pages 532–541. Springer Berlin Heidelberg, January 2010.

[40] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Miguel Martínez. Multi-objective differential evolution algorithm with spherical pruning, 2012.

[41] Andres Rodriguez, Evan Andersen, Justin Bradley, and Clark Taylor. Wind estimation using an optical flow sensor on a miniature air vehicle. American Institute of Aeronautics and Astronautics, August 2007.

[42] K. Rodriguez-Vazquez and P. J. Fleming. Multi-objective genetic programming for nonlinear system identification. *Electronics Letters*, 34(9):930–931, April 1998.

[43] Detlef Rohlf, Stefan Schmidt, and Jonathan Irving. Stability and control analysis for an unmanned aircraft configuration using system-identification techniques. *Journal of Aircraft*, 49(6):1597–1609, November 2012.

[44] J Sanchis, M Martinez, X Blasco, and JV Salcedo. A new perspective on multiobjective optimization by enhanced normalized normal constraint method. *Structural and multidisciplinary optimization*, 36(5):537–546, 2008.

[45] Javier Sanchis, Miguel A. Martínez, Xavier Blasco, and Gilberto Reynoso-Meza. Modelling preferences in multi-objective engineering design. *Engineering Applications of Artificial Intelligence*, 23(8):1255–1264, December 2010.

[46] Shih-Lian Cheng and Chyi Hwang. Optimal approximation of linear systems by a differential evolution algorithm. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 31(6):698–707, November 2001.

[47] Mandavilli Srinivas and Lalit M Patnaik. Genetic algorithms: A survey. *Computer*, 27(6):17–26, 1994.

[48] Rainer Storn and Kenneth Price. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[49] J. Velasco and S. García-Nieto. Unmanned aerial vehicles model identification using multi-objective optimization techniques. Cape Town, South Africa, August 2014.

[50] Jesús Velasco. Identificación de modelos dinámicos y ajuste de controladores basado en algoritmos evolutivos multiobjetivo. Master's thesis, Universidad Politècnica de València, Spain, September 2013.

[51] Jesús Velasco, Sergio García-Nieto, Gilberto Reynoso-Meza, and Javier Sanchis. Implementación de un sistema hardware-in-the-loop para la simulación en tiempo real de pilotos automáticos para UAVs. In *Actas de las XXXIV Jornadas de Automática*, October 2013.

[52] Jesus Velasco, Sergio García-Nieto Rodríguez, Gilberto Reynoso Meza, and Javier Sanchis Saez. Desarrollo y evaluación de una estación de control de tierra para vehículos aéreos no tripulados. In *Actas de las XXXIII Jornadas de Automática*, 2012.

[53] K. Charles Wang and Kenneth W. Iliff. Retrospective and recent examples of aircraft parameter identification at NASA dryden flight research center. *Journal of Aircraft*, 41(4):752–764, July 2004.

[54] Simon Watkins, Juliette Milbank, Benjamin J. Loxton, and William H. Melbourne. Atmospheric winds and their implications for microair vehicles. *AIAA Journal*, 44(11):2591–2600, November 2006.

[55] Huang Wenbo and Zhang Qiang. The hardware-in-the-loop simulation on the control system of a small launch vehicle. *Procedia Engineering*, 29:1867–1871, 2012.

[56] Haitao Xiang and Lei Tian. Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV). *Biosystems engineering*, 108(2):174–190, 2011.

[57] Hassan Yousefi, Heikki Handroos, and Azita Soleymani. Application of differential evolution in system identification of a servo-hydraulic system with a flexible load. *Mechatronics*, 18(9):513–528, November 2008.

[58] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

Part II

Control, Simulation and Test

Chapter 4

Control Strategies for Unmanned Aerial Vehicles under Parametric Uncertainty and Disturbances: a Comparative Study [1]

Abstract

UAVs within the class of Mini Aerial Vehicles (MAVs) are autonomous aircrafts with low inertia that fly at relatively low speeds. In this sense, MAVs are exposed to high airspeed uncertainties since unexpected changes in wind velocity represent an important percentage of the total airspeed of the vehicle. Moreover, such changes directly modify the aerodynamic forces acting along the vehicle, which leads to important variations on their acceleration owed to their low inertia. Although the structure of the dynamic model of an aircraft is well known, important difficulties arise on the identification of an specific MAV due to its particular characteristics. Thus, modelling errors become an additional source of uncertainty when control algorithms are designed. In this situation, studying the ability that different control strategies present in performing trajectory tracking is of great interest on the development of applications for this type of UAVs. In this paper a comparative study of four control strategies is presented. All algorithms have

[1] J. Velasco, S. García-Nieto, R. Simarro, J. Sanchis. Control Strategies for Unmanned Aerial Vehicles under Parametric Uncertainty and Disturbances: a Comparative Study. IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles June 10-12, 2015. Seville, Spain. IFAC-PapersOnLine, vol. 48, n.o 9, pp. 1-6, 2015. https://doi.org/10.1016/j.ifacol.2015.08.050

been implemented in a MAV flight computer. Results from both, Hardware-In-the-Loop (HIL) simulations and real flight experiments, are presented as the main contribution of this work.

4.1 Introduction

MAVs are UAVs with masses between 1 and $15kg$ on take off and flying altitudes up to $3000m$ [7]. Those characteristics along with their relative low velocity compared to the wind speed make engineers confront to high uncertainties when system dynamics have to be identified. Although the structure of the dynamic model of an aircraft is well known, important difficulties arise on the particularisation of that structure to an specific MAV. These modelling errors become an important source of uncertainty when control algorithms are designed. Thus, the performance of a given control strategy might decrease significantly when implemented, compared to the observed behaviour by using the identified model.

In this paper a comparative study of four control strategies is presented: PID tuned by root locus, PID tuned by means of multiobjective optimization, a Linear Quadratic Regulator (LQR) and a Model Based Predictive controller (MPC). All algorithms have been implemented in a MAV flight computer. Comparison results between Hardware-In-the-Loop (HIL) simulations and real flight experiments are presented as the main contribution of this work.

This paper is divided in five sections as follows: in section 4.2 a complete description of the UAV is made, including a dynamic model of the system and its linearization; section 4.3 presents each of the control strategies implemented. Then, results from simulation and experimentation will be shown in section 4.4 and some final conclusions will be remarked in section 4.5.

4.2 UAV testbench

Three main points are develop along this section. Firstly, both simulation and flight experiments setups are introduced. Including on one side, the aircraft and its hardware components, and on the other, a HIL platform to previously test the designed control algorithms. HIL simulations, the designing process itself and the final controllers performance are mainly based and hence rely on a first principles model of the platform. Therefore that model structure is derived in the second point of this section. Finally, the last point of this section presents the linearisation of the equations presented in 4.2.2 which is used inside the MPC controller and in the LQR design.

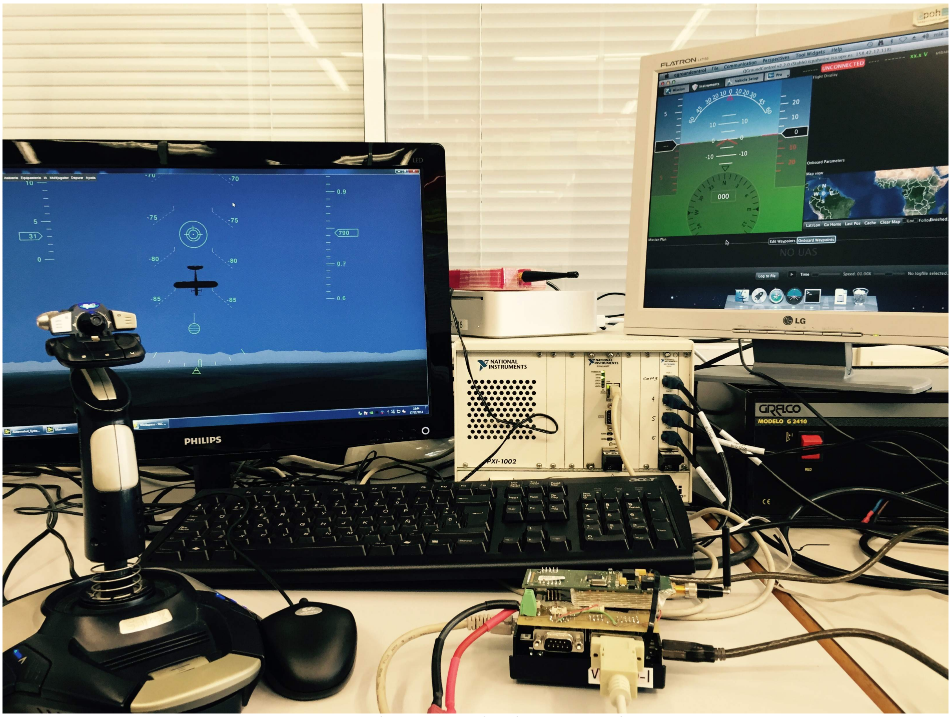


Figure 4.1. Hardware in the loop simulations setup

4.2.1 Experiments and simulations setups

As the main component of the flight platform, a Kadett 2400 aircraft, manufactured by Graupner, is found. It is a light weight air-frame with some features that make it suitable for the purposes of this research. Some of those characteristics are:

- $2.4m$ wing span.
- $0.9m^2$ wing area.
- $49g/dm^2$ weight/area ratio.
- $16.5l$ free volume.

It houses all necessary devices for its control, not only manually, but also in autonomous mode. During flight, three control surfaces are provided: tail rudder, elevators and ailerons. As the unit of propulsion a brushless engine of alternating current is integrated, which is fed by two LIPO batteries through a frequency variator. Alike the servomotors, the variator is controlled by sending Pulse Width Modulated (PWM) signals as commanding signals.

There exists a bridge device between the manual and the autonomous states. The Servo Switch Controller (SSC) is able to perform the commutation between different command sources. Moreover, it offers the possibility of measuring the applied deflections in control surfaces and changes in the motor torque.

Control actions are sent from the Flight Control Station (FCS), constituted by a Beagle Bone Black (BBB)[2] board. This unit houses the control algorithms and performs all necessary tasks at each phase of the flight. The loop is closed by the GPS-AHRS IG500N[3] unit. It is a all-in-one device, which joins the efforts of a wide range of sensors, such as accelerometers, gyroscopes and magnetometers. Its Kalman filter is capable of mixing the information coming from those sensors in order to offer precise measurements of position, orientation, linear and angular speeds and accelerations, in the three aircraft body-axes. In [14, 13, 12, 11] is presented this platform with more details together with the results of the first flight tests.

Regarding the HIL simulation setup (Fig. 4.1), a National Instruments PXI with a real time running model substitutes most of the hardware components, with the exception of the FCS. In this way, the control algorithms are implemented in the BBB and executed exactly as they will be during real flight experiments. This strategy of simulation increases the confidence on the designed controllers by assuring a higher level of safety in the hop from simulation to experimentation.

4.2.2 Aircraft Dynamic Model

Its particularization to our aircraft is the result of previous works published by the authors and has proved to accurately describe the vehicle dynamics.

The dynamic model will not only be used for simulations in the design stage of control algorithms, but also part of the control algorithm in the MPC strategy. Hence the expressions that relate the input variables, deflection in the control surfaces and motor load, to a series of output signals such as linear and angular velocities and acceleration, and position in a 3D space, will have a direct impact in some of the controllers developed.

Linear and angular momentum conservation principles conform the starting point to derive such model [5]:

[2] http://www.beagleboard.org/

[3] http://www.sbg-systems.com/products/ig500n-miniature-ins-gps

$$\sum_{ext} \overrightarrow{F} = \frac{d}{dt}(m\overrightarrow{V}) \tag{4.1}$$

$$\sum_{ext} \overrightarrow{M} = \frac{d}{dt}(I\overrightarrow{\omega}) \tag{4.2}$$

where (4.1) and (4.2) deal with the sum of external forces and moments respectively; m and I are the mass and the inertia tensor of the aircraft, and $\overrightarrow{V}$ and $\overrightarrow{\omega}$ are linear and angular velocity vectors. In particular three are the types of external forces that affect to the behaviour of the vehicle: aerodynamic force (F_A), force applied by the motor (F_T) and the gravitational force (F_G). At the same time, two different sources can be counted as torque generators: the air flow -generating aerodynamic torque (M_A) and the motor moment (M_T). Thereby, the equations (4.1) and (4.2) can be rewritten as:

$$\overrightarrow{F_A} + \overrightarrow{F_T} + \overrightarrow{F_G} = m\dot{\overrightarrow{V}} + \overrightarrow{\omega} \times m\overrightarrow{V} \tag{4.3}$$

$$\overrightarrow{M_A} + \overrightarrow{M_T} = I\dot{\overrightarrow{\omega}} + \overrightarrow{\omega} \times I\overrightarrow{\omega} \tag{4.4}$$

The two previous equations are actually vectorial equations, so that, there is a total of 6 equations that correspond to the 6 degrees of freedom of a rigid body in the space. Deriving (4.3) and (4.4) the following expressions are obtained:

$$\begin{aligned} \overline{q}S\begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} + \begin{bmatrix} -g\sin\theta \\ g\sin\phi\cos\theta \\ g\cos\phi\cos\theta \end{bmatrix} + \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} \\ = m\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times m\begin{bmatrix} u \\ v \\ w \end{bmatrix} \end{aligned} \tag{4.5}$$

$$\begin{aligned} \overline{q}S\begin{bmatrix} bC_l \\ \overline{c}C_m \\ bC_n \end{bmatrix} + \begin{bmatrix} 0 \\ I_p\Omega_p r \\ -I_p\Omega_p q \end{bmatrix} = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{zx} & 0 & I_z \end{bmatrix}\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \\ + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{zx} & 0 & I_z \end{bmatrix}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \end{aligned} \tag{4.6}$$

Where u, v, and w are the components of the aircraft linear velocity in its body axes (x_b, y_b, z_b), and p, q, and r are the three components of the angular velocity; I_p is the rotating inertia of the tandem motor and propeller; Ω_p its rotating speed and T the motor thrust. As it can be noticed, the products of inertia related to the longitudinal plane ($Y_b = 0$), are both null. This is owed to the aircraft's symmetry with respect to this plane. Finally S, b and $\overline{c}$ are constructive constants of the airplane whilst $\overline{q}$ is the dynamic pressure of the air.

It is important to highlight the apparition in equations (4.5) and (4.6) of the variables C_i, that represent the Aerodynamic Coefficients (AC) of each component of the resultant aerodynamic force and torque. Such coefficients are functions that relate those components to some of the system variables, e.g. the deflection of the control surfaces. In this way, their value directly influence the stability and controlability of the system, basic issues for the controllers design. The AC can be expressed as in ([11]):

Longitudinal aerodynamic models:

$$\begin{aligned} C_D =& \mathbf{C_{D_0}} + \mathbf{C_{D_{V_{air}}}} \frac{1}{V_0} \Delta V_{air} + \mathbf{C_{D_\alpha}} \Delta\alpha \\ & + \mathbf{C_{D_{\alpha^2}}} \Delta\alpha^2 + \mathbf{C_{D_q}} \frac{\overline{c}}{2V_0} q + \mathbf{C_{D_{\delta_e}}} \Delta\delta_e \end{aligned} \tag{4.7}$$

$$\begin{aligned} C_L =& \mathbf{C_{L_0}} + \mathbf{C_{L_{V_{air}}}} \frac{1}{V_0} \Delta V_{air} + \mathbf{C_{L_\alpha}} \Delta\alpha \\ & + \mathbf{C_{L_{\alpha^2}}} \Delta\alpha^2 + \mathbf{C_{L_{\dot\alpha}}} \frac{\overline{c}}{2V_0} \dot\alpha + \mathbf{C_{L_q}} \frac{\overline{c}}{2V_0} q \\ & + \mathbf{C_{L_{\delta_e}}} \Delta\delta_e \end{aligned} \tag{4.8}$$

$$\begin{aligned} C_m =& \mathbf{C_{m_0}} + \mathbf{C_{m_{V_{air}}}} \frac{1}{V_0} \Delta V_{air} + \mathbf{C_{m_\alpha}} \Delta\alpha \\ & + \mathbf{C_{m_{\alpha^2}}} \Delta\alpha^2 + \mathbf{C_{m_{\dot\alpha}}} \frac{\overline{c}}{2V_0} \dot\alpha + \mathbf{C_{m_q}} \frac{\overline{c}}{2V_0} q \\ & + \mathbf{C_{m_{\delta_e}}} \Delta\delta_e \end{aligned} \tag{4.9}$$

Lateral aerodynamic models:

$$\begin{aligned} C_Y =& \mathbf{C_{Y_0}} + \mathbf{C_{Y_\beta}} \Delta\beta + \mathbf{C_{Y_p}} \frac{b}{2V_0} p + \mathbf{C_{Y_r}} \frac{b}{2V_0} r \\ & + \mathbf{C_{Y_{\delta_{al}}}} \Delta\delta_{al} + \mathbf{C_{Y_{\delta_r}}} \Delta\delta_r \end{aligned} \tag{4.10}$$

$$\begin{aligned} C_l =& \mathbf{C_{l_0}} + \mathbf{C_{l_\beta}} \Delta\beta + \mathbf{C_{l_p}} \frac{b}{2V_0} p + \mathbf{C_{l_r}} \frac{b}{2V_0} r \\ & + \mathbf{C_{l_{\delta_{al}}}} \Delta\delta_{al} + \mathbf{C_{l_{\delta_r}}} \Delta\delta_r \end{aligned} \tag{4.11}$$

$$\begin{aligned} C_n =& \mathbf{C_{n_0}} + \mathbf{C_{n_\beta}} \Delta\beta + \mathbf{C_{n_p}} \frac{b}{2V_0} p + \mathbf{C_{n_r}} \frac{b}{2V_0} r \\ & + \mathbf{C_{n_{\delta_{al}}}} \Delta\delta_{al} + \mathbf{C_{n_{\delta_r}}} \Delta\delta_r \end{aligned} \tag{4.12}$$

Where α and β are the angle of attack and of side-slip respectively and V_{air} is the airspeed. In particular, V_0 is the airspeed measured at the steady state of flight, before a maneuver begins. These variables are velocity dependent and they can be calculated as follows:

$$\alpha = \arctan\left(\frac{w}{u}\right) \tag{4.13}$$

$$\beta = \arcsin\left(\frac{v}{V}\right) \tag{4.14}$$

$$V_{air} = |\overrightarrow{V}| = \sqrt{u^2 + v^2 + w^2} \tag{4.15}$$

C_L and C_D are the Lift and Drag coefficients and their relation with C_X and C_Z is:

$$C_L = -C_Z \cos(\alpha) + C_X \sin(\alpha) \tag{4.16}$$

$$C_D = -C_X \cos(\alpha) - C_Z \sin(\alpha) \tag{4.17}$$

The aircraft orientation is usually denoted with the Euler angles: roll ϕ, pitch θ, and yaw ψ; which express the rotation of a body from a global reference system to the body-axes. The kinematic equations that relate angular velocities of the aircraft to the Euler angles are:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\theta & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{4.18}$$

Finally, equations (4.19) to (4.27) are the result of reordering equations (4.5), (4.6), and (4.18), so they can be directly used on the calculation of output values (such values will be the same as those coming from the GPS-AHRS unit in real flights). The δ symbol in brackets represents the dependency of the AC on the deflections of the control surfaces.

Force equations:

$$\dot{u} = rv - qw + \frac{\overline{q}S}{m} C_X(\delta_{[e,al,r]}) - g\sin\theta + \frac{T}{m} \tag{4.19}$$

$$\dot{v} = pw - ru + \frac{\overline{q}S}{m} C_Y(\delta_{[e,al,r]}) + g\cos\theta\sin\phi \tag{4.20}$$

$$\dot{w} = qu - pv + \frac{\overline{q}S}{m} C_Z(\delta_{[e,al,r]}) + g\cos\theta\cos\phi \tag{4.21}$$

Torque equations:

$$\dot{p} - \frac{I_{xz}}{I_x}\dot{r} = \frac{\overline{q}Sb}{I_x}C_l(\delta_{[e,al,r]}) - \frac{I_z - I_y}{I_x}qr + \frac{I_{xz}}{I_x}qp \tag{4.22}$$

$$\begin{aligned}\dot{q} &= \frac{\overline{q}S\overline{c}}{I_y}C_m(\delta_{[e,al,r]}) - \frac{I_x - Iz}{I_y}pr \\ &- \frac{I_{xz}}{I_y}(p^2 - r^2) + I_p\Omega_p r\end{aligned} \tag{4.23}$$

$$\begin{aligned}\dot{r} - \frac{I_{xz}}{I_z}\dot{p} &= \frac{\overline{q}Sb}{I_z}C_n(\delta_{[e,al,r]}) - \frac{I_y - I_x}{I_z}pq - \frac{I_{xz}}{I_z}qr \\ &- I_p\Omega_p q\end{aligned} \tag{4.24}$$

Kinematic equations:

$$\dot{\phi} = p + \tan\theta(q\sin\phi + r\cos\phi) \tag{4.25}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{4.26}$$

$$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta} \tag{4.27}$$

4.2.3 Model Linearization

Both in the case of PID design as in the LQR design a linear system associated to the previous model is needed. Furthermore, the MPC calculates the optimum values of future inputs based on the predictions of the future states. This predictions are actually given by a model of the system. For practical reasons, it is a requirement that such model be linear, given that, in presence of a quadratic cost function and a set of affine constraints, the resulting optimization problem is convex and so can be rapidly solved. Therefore, the model derived in section 4.2.2 has to be linearised around an equilibrium point. The following conditions are imposed:

$$\theta_0 \neq 0 \qquad \phi_0 \neq 0 \qquad \psi_0 = 0 \tag{4.28}$$

$$u_0 = 18m/s \qquad v_0 = 0 \qquad w_0 = 0 \tag{4.29}$$

$$p_0 = 0 \qquad q_0 = 0 \qquad r_0 = 0 \tag{4.30}$$

$$\dot{p}_0 = 0 \qquad \dot{q}_0 = 0 \qquad \dot{r}_0 = 0 \tag{4.31}$$

Then solving the equations under these conditions, the solution obtained is:

$$\begin{array}{llll}\delta_{e_0} = 0.0041 & \delta_{al_0} = 0 & \delta_{r_0} = 0 & T_0 = 19.920762 \\ \alpha_0 = 0.019775 & \beta_0 = 0 & \theta_0 = 0 & \phi_0 = -0.0305\end{array}$$

Giving the following linear system:

$$\begin{bmatrix} \dot{V}_x \\ \dot{V}_y \\ \dot{V}_z \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = A \begin{bmatrix} V_x \\ V_y \\ V_z \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{bmatrix} + B \begin{bmatrix} \delta_e \\ \delta_{al} \\ \delta_r \\ \delta_T \end{bmatrix}; \begin{bmatrix} V_x \\ \phi \\ \theta \end{bmatrix} = C \begin{bmatrix} V_x \\ V_y \\ V_z \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{bmatrix} \tag{4.32}$$

where:

$$A = \begin{bmatrix} 9833 & 0 & 461 & 0 & 1784 & 0 & 0 & -4864 & 0 \\ 0 & 9687 & 0 & 0512 & 0 & 345 & 4828 & 0 & 0 \\ -151 & 0 & 8583 & 0 & -7575 & 0 & 0 & 37 & 0 \\ 0 & -314 & 0 & 5655 & 0 & 354 & -85 & 0 & 0 \\ 17 & 0 & -79 & 0 & 5255 & 0 & 0 & -4.6 & 0 \\ 0 & 244 & 0 & -995 & 0 & 8931 & 6 & 0 & 0 \\ 0 & -8.7 & 0 & 382 & 0 & 9.9 & 9999 & 0 & 0 \\ 0.5 & 0 & -2.2 & 0 & 382 & 0 & 0 & 1e4 & 0 \\ 0 & 6.0 & 0 & -28 & 0 & 473 & 0 & 0 & 1e4 \end{bmatrix} e^{-4} \tag{4.33}$$

$$B = \begin{bmatrix} 2832 & 994 & 0 & 0 \\ 0 & 0 & 3165 & 81 \\ -22 & 3361 & 0 & 0 \\ 0 & 0 & -3.53e4 & -2893 \\ -2.7 & -1.54e4 & 0 & 0 \\ 0 & 0 & -4378 & -9167 \\ 0 & 0 & 0 & -963 \\ & -77 & & \\ 0 & -428 & 0 & 0 \\ 0 & 0 & -132 & -235 \end{bmatrix} e^{-4} \tag{4.34}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.35}$$

4.3 Compared Control Strategies

4.3.1 PID Control

The first control scheme[4] proposed for the study is the well-known PID ISA algorithm (Anti-Windup included). In particular, a cascade PID scheme has been

[4] Numerical values for the controllers described in this section can be download at: http://goo.gl/U5xlku

proposed [1]. The inner loop manages the pitch and roll variables, therefore 2 PIDs have been adjusted in this stage. On the other hand, the outer loop is dedicated to navigation and is composed by 3 PIDs (altitude, heading and velocity). Moreover, K_R, T_I and T_D for this set of 5 PIDs have been adjusted using two different approaches. The first one, a classical technique based on closed loop poles assignment, the well-known Root Locus design procedure [15] (denoted as PID 1 in the comparative results).

In a second approach a multi-objective optimization design (MOOD) method has been used. MOOD have shown to be a valuable tool for controller tuning applications [9]. For a successful implementation of the MOOD procedure, three fundamental steps are required: the multi-objective problem (MOP) definition, the multi-objective optimization process and the multi-criteria decision making step (denoted as PID 2 in the comparative results).

For the MOP definition, 7 design objectives (in time domain) have been included: settling time of altitude (J_1) and yaw (J_2), integral of the absolute value of motor usage (J_3), and total variation of ailerons (J_4), elevators (J_5), roll (J_6) and pitch (J_7). For the optimization process, the sp-MODE-II[5] algorithm has been used; its selection is due to its capabilities to improve convergence, diversity and pertinence of solutions in the approximated Pareto front [8]. Finally, the MCDM step has been performed using Level Diagrams [3] visualization technique[6].

4.3.2 Linear Quadratic Regulator

The second control scheme proposed for the comparative study is a LQR with infinite horizon [6]. This control technique minimizes the index (4.36) subject to $\dot{x}(t) = Ax(t) + Bu(t)$ and the feedback control law $u(t) = -Kx(t)$.

$$J = \int_0^\infty \left(x^T Q x + u^T R u\right) dt \tag{4.36}$$

The main LQR advantage is that the optimal input signal $u(t)$ is obtained from full state feedback. The feedback matrix K is obtained by solving the associated Ricatti equation. LQR controllers are widely used due to the implicit advantages in this control structure as the guaranteed stability when a system has all of its states available for feedback, stability margins under accuracy model representations and simply adjustment using the weighting matrix Q and R.

[5] Implementation available at: http://www.mathworks.com/matlabcentral/fileexchange/47035

[6] Implementation available at: http://www.mathworks.com/matlabcentral/fileexchange/24042

4.3.3 Model Predictive Control

Model Predictive Control (MPC) has been used in process industries such as chemical plants and oil refineries since the 1980s [2]. Maybe, some interesting features of MPC as its multivariable nature, constraints management, or explicit predictors, can be pointed out as the responsible for its success. The basic idea of MPC is the use of a process model to predict the behaviour of the system over a specified (finite) prediction horizon [10]. Assuming that the system under control is described by a state transition equation

$$x(k+1) = Ax(k) + Bu(k) \tag{4.37}$$

Given the current state $x(k)$, a predictive controller calculates the optimal sequence of future control outputs $u(k), \cdots, u(k+C-1)$, over a control horizon, such that a cost function is minimized -giving a non *static* control law. Most of the objective functions used are modifications of a quadratic function. The cost index selected for our approach is:

$$J = \sum_{j=1}^{H} x(k+j)^T Q x(k+j) + \sum_{i=0}^{C-1} u(k+i)^T R u(k+i) \tag{4.38}$$

Index (4.38) is minimized, subject to the process model

$$x(k+j+1) = Ax(k+j) + Bu(k+j), j = 1, \cdots, H \tag{4.39}$$

and a set of control constraints,

$$min(U) \leq u(k+i) \leq max(U), i = 0, 1, \cdots, C-1 \tag{4.40}$$

The MPC control strategy is the most complex from the implementation point of view compared with PIDs and LQR.

4.4 Results of the Comparative Study

The study presented compares the dynamical behaviour for the four control techniques described in section 4.3 under a simple flight mission. In particular, the flight mission has been defined as four 3D-waypoints area close to Valencia (Spain), table 4.1 and figure 4.2 described the mission plan in detail:

The comparative study has been applied in two main and complementary scenarios. On one hand, simulations based on a Hardware-In-Loop platform has been performed. This technique is used for the non-destructive development and testing of control algorithms. In a HIL simulation the physical part of a machine or

	Latitude^o	Longitude^o	Altitude (m)
WP 0	39.49612634133568	-0.6241178512573242	300.0
WP 1	39.49827887271285	-0.6208562850952148	350.0
WP 2	39.49923921136225	-0.6230878829956054	350.0
WP 3	39.49751721538407	-0.6264352798461914	300.0

Table 4.1. Waypoints information.

system is replaced by a simulation setup. In this case, all the complex sensors and actuators present in the real UAV are virtualized and included in the simulation procedure [4].

Data in figure 4.3(a) show the control performances reached by the four control systems tested under HIL simulations. PID1 and PID2 algorithms present better performances, related with minimum values shown in figure 4.3(b). The trial and error procedure to tune the Q and R matrices in LQR and the constraints related to control effort introduced in MPC lead to worse performance for these two control algorithms when they are compared with PIDs.

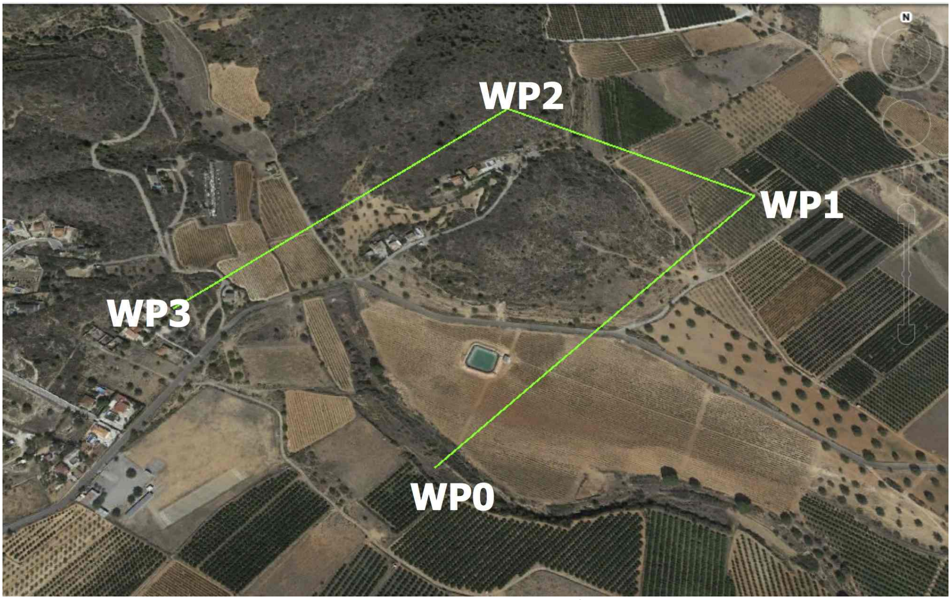
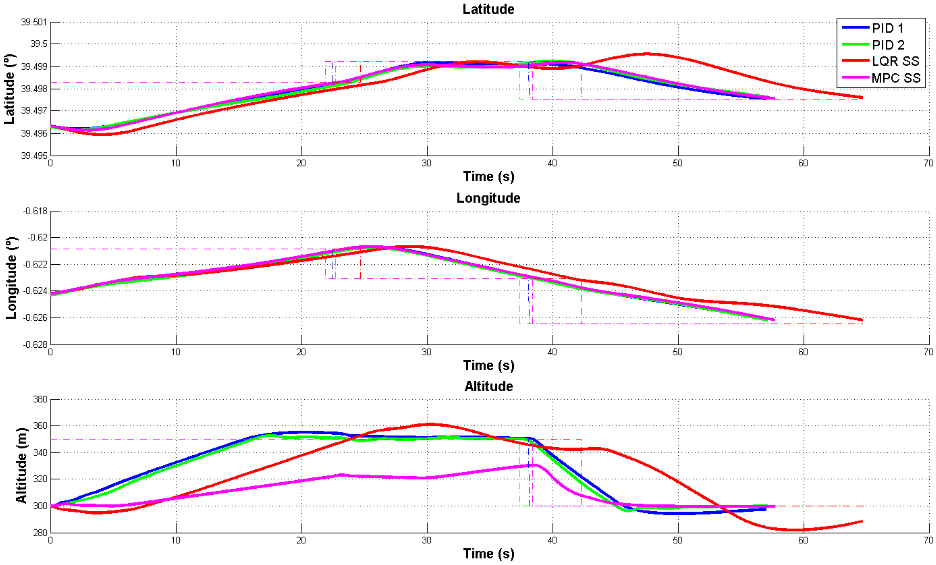


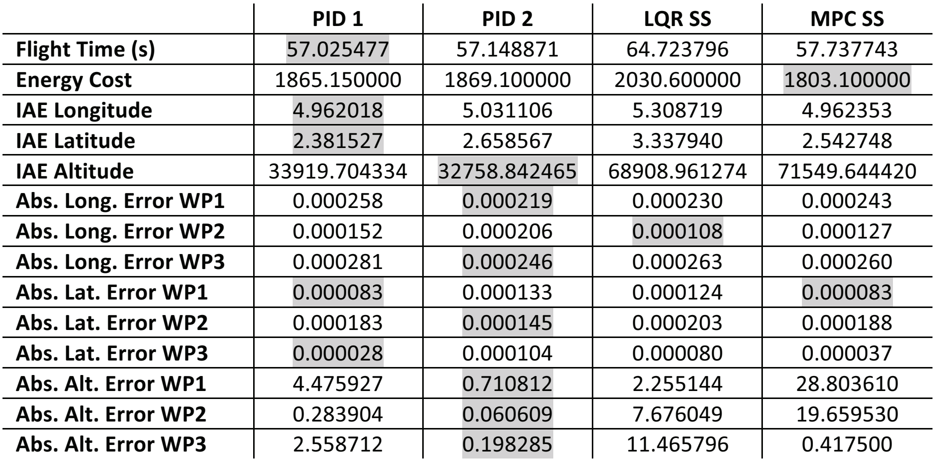
Figure 4.2. Mission Plan.

On the other hand, a comparative real flight allows to study the dynamic behaviour under uncertainties and disturbances involved in any real context. This second scenario will corroborate, or not, the predicted results obtained under HIL simulations, to determine the best alternative from the control scheme point of view in the studied UAV platform. Results from flight tests are depicted on figures 4.4(a) and 4.4(b). Again PID controllers have a better performance in terms of robustness to uncertainty and process disturbances. In particular, the PID controller adjusted by optimization techniques (PID 2) obtains prominent results in both HIL simulation and experimental flight. Whereas HIL simulations have been performed assuming no wind and flight test were performed under windy scenarios, advanced control techniques as LQR or MPC -both techniques relying in the linear dynamic model, are less robust than PID controllers. That means that wind information might be managed by the model to reach better results with these control methodologies.

However, the results achieved by MPC related with the energy consumption in the HIL simulations must be highlighted (Energy Cost row in figure 4.3(b)). The possibility of including constraints in control actions within the optimization process is one of the major advantages of MPC, and could explain the observed behaviour (Notice how, that kindness, is not present in the real flight test). A possible explanation for such phenomenon resides in the existence of parametric uncertainties in the model employed by the MPC. Those uncertainties are directly transmitted to the controller in the form of prediction errors and, therefore, the advantage reached on simulation disappears on a more realistic environment.



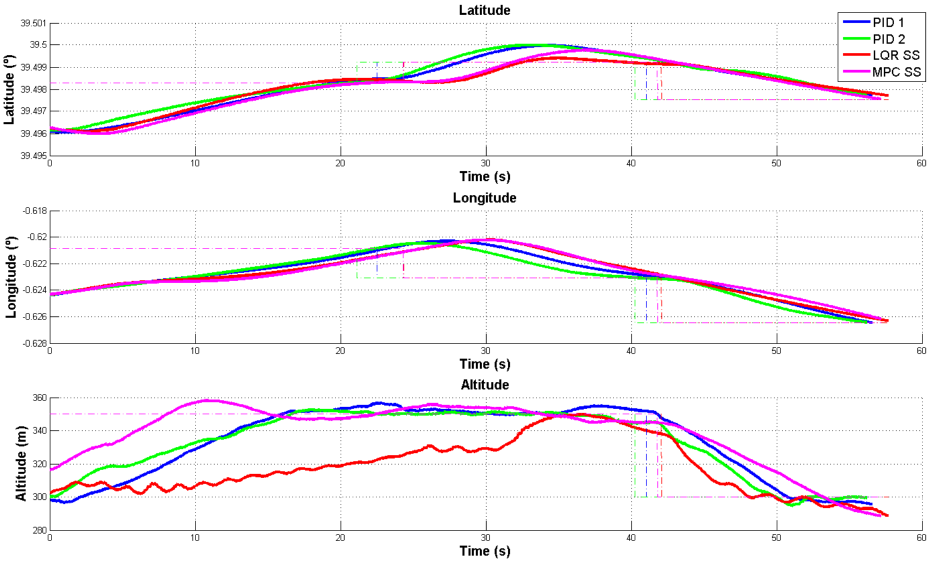
(a) Comparative Dynamic responses



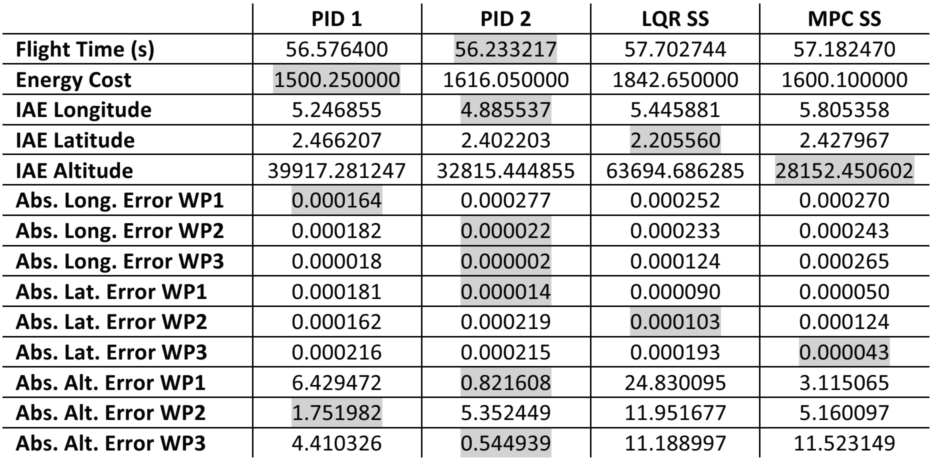
	PID 1	PID 2	LQR SS	MPC SS
Flight Time (s)	57.025477	57.148871	64.723796	57.737743
Energy Cost	1865.150000	1869.100000	2030.600000	1803.100000
IAE Longitude	4.962018	5.031106	5.308719	4.962353
IAE Latitude	2.381527	2.658567	3.337940	2.542748
IAE Altitude	33919.704334	32758.842465	68908.961274	71549.644420
Abs. Long. Error WP1	0.000258	0.000219	0.000230	0.000243
Abs. Long. Error WP2	0.000152	0.000206	0.000108	0.000127
Abs. Long. Error WP3	0.000281	0.000246	0.000263	0.000260
Abs. Lat. Error WP1	0.000083	0.000133	0.000124	0.000083
Abs. Lat. Error WP2	0.000183	0.000145	0.000203	0.000188
Abs. Lat. Error WP3	0.000028	0.000104	0.000080	0.000037
Abs. Alt. Error WP1	4.475927	0.710812	2.255144	28.803610
Abs. Alt. Error WP2	0.283904	0.060609	7.676049	19.659530
Abs. Alt. Error WP3	2.558712	0.198285	11.465796	0.417500

(b) Comparative performance indexes

Figure 4.3. HIL simulation data



(a) Comparative Dynamic responses



	PID 1	PID 2	LQR SS	MPC SS
Flight Time (s)	56.576400	56.233217	57.702744	57.182470
Energy Cost	1500.250000	1616.050000	1842.650000	1600.100000
IAE Longitude	5.246855	4.885537	5.445881	5.805358
IAE Latitude	2.466207	2.402203	2.205560	2.427967
IAE Altitude	39917.281247	32815.444855	63694.686285	28152.450602
Abs. Long. Error WP1	0.000164	0.000277	0.000252	0.000270
Abs. Long. Error WP2	0.000182	0.000022	0.000233	0.000243
Abs. Long. Error WP3	0.000018	0.000002	0.000124	0.000265
Abs. Lat. Error WP1	0.000181	0.000014	0.000090	0.000050
Abs. Lat. Error WP2	0.000162	0.000219	0.000103	0.000124
Abs. Lat. Error WP3	0.000216	0.000215	0.000193	0.000043
Abs. Alt. Error WP1	6.429472	0.821608	24.830095	3.115065
Abs. Alt. Error WP2	1.751982	5.352449	11.951677	5.160097
Abs. Alt. Error WP3	4.410326	0.544939	11.188997	11.523149

(b) Comparative performance indexes

Figure 4.4. Real flight data

4.5 Conclusions

Results presented in the previous section show that the most robust strategies for the proposed scenario, are PID controllers, and particularly the PID adjusted by means of mutli-objective optimization. This is mainly due to the existence of parametric uncertainties residing on the aerodynamic coefficients. Since more complex techniques, such as LQR and MPC, have a much higher reliance on the information provided by the model (used in the design phase or, in the case of the MPC, even in the control phase), their behaviour decrease significantly on real situations.

However, despite the results obtained from the point of view of dynamic response, the MPC strategy presents a very interesting advantage, the proposal of input signals much less aggressive and smoother than the rest of controllers, as it is shown in figure 4.5. This fact by itself can be considered as a sufficient motivation for choosing such techniques, since they ensure a much longer life cycle to the set aircraft actuators, which is an issue of vital importance in this type of systems. In addition, an iterative process of improving the values of the derivatives of stability and control could significantly improve the performance of the LQR and MPC controllers.

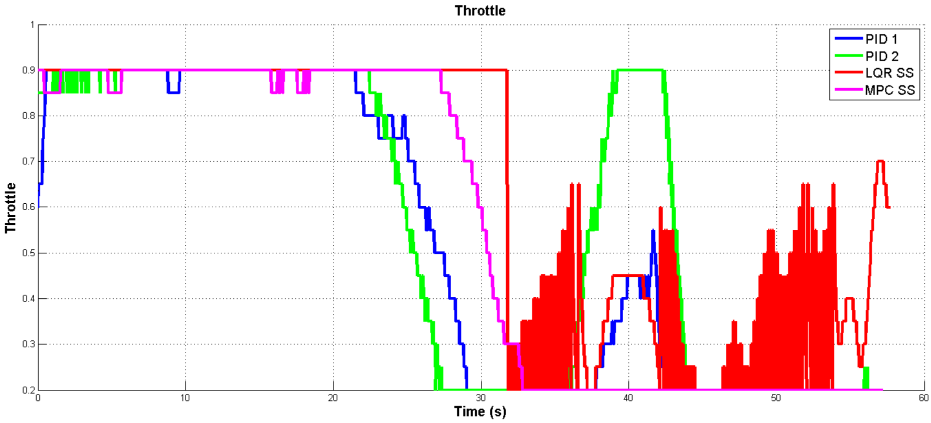


Figure 4.5. Flight Data: comparative throttle behavior. Max throttle=1

Bibliography

[1] Kiam Heong Ang, Gregory Chong, and Yun Li. PID control system analysis, design, and technology. *Control Systems Technology, IEEE Transactions on*, 13(4):559–576, 2005.

[2] T Badgwell and S Qin. Review of nonlinear model predictive control applications. *Nonlinear Predictive Control: Theory and Practice*, Jan 2001.

[3] X. Blasco, J.M. Herrero, J. Sanchis, and M. Martínez. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences*, 178(20):3908 – 3924, 2008.

[4] Guowei Cai, Ben M Chen, Tong H Lee, and Miaobo Dong. Design and implementation of a hardware-in-the-loop simulation system for small-scale uav helicopters. *Mechatronics*, 19(7):1057–1066, 2009.

[5] Vladislav Klein and Eugene A. Morelli. *Aircraft system identification: theory and practice*. American Institute of Aeronautics and Astronautics Reston, VA, USA, 2006.

[6] Leonard Lublin and Michael Athans. Linear quadratic regulator control. *The control handbook*, pages 635–650, 1996.

[7] Jeffrey M. Maddalon, Kelly J. Hayhurst, Daniel M. Koppen, Jason M. Upchurch, and A. Terry Morris. Perspectives on unmanned aircraft. classification for civil airworthiness standards, 2013.

[8] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Sergio García-Nieto. Multiobjective evolutionary algortihms for multivariable PI controller tuning. *Appied Soft Computing*, 24:341 – 362, 2014.

[9] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Miguel Martínez. Controller tuning using evolutionary multi-objective optimisation: current trends and applications. *Control Engineering Practice*, 1:58 – 73, 2014.

[10] J.M Sousa, R Babuska, P Bruijn, and H.B Verbruggen. Comparison of conventional and fuzzy predictive control. *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, 3:1782–1787 vol.3, 1996.

[11] J. Velasco and S. García-Nieto. Unmanned aerial vehicles model identification using multi-objective optimization techniques. Cape Town, South Africa, August 2014.

[12] Jesús Velasco. Identificación de modelos dinámicos y ajuste de controladores basado en algoritmos evolutivos multiobjetivo. Master's thesis, Universidad Politècnica de València, Spain, September 2013.

[13] Jesús Velasco, Sergio García-Nieto, Gilberto Reynoso-Meza, and Javier Sanchis. Implementación de un sistema hardware-in-the-loop para la simulación en tiempo real de pilotos automáticos para UAVs. In *Actas de las XXXIV Jornadas de Automática*, October 2013.

[14] Jesus Velasco, Sergio García-Nieto Rodríguez, Gilberto Reynoso Meza, and Javier Sanchis Saez. Desarrollo y evaluación de una estación de control de tierra para vehículos aéreos no tripulados. In *Actas de las XXXIII Jornadas de Automática*, 2012.

[15] Qing-Guo Wang, Tong-Heng Lee, Ho-Wang Fung, Qiang Bi, and Yu Zhang. Pid tuning for improved performance. *Control Systems Technology, IEEE Transactions on*, 7(4):457–465, 1999.

Chapter 5

Enhancing controller's tuning reliability with multi-objective optimisation: from *Model in the loop* to *Hardware in the loop* [1]

Abstract

In general, the starting point for the complex task of designing a robust and efficient control system is the use of nominal models that allow to establish a first set of parameters for the selected control scheme. Once the initial stage of design is achieved, control engineers face the difficult task of Fine-Tuning for a more realistic environment, where the environment conditions are as similar as possible to the real system. For this reason, in the last decades the use of Hardware-in-The-Loop (HiL) systems has been introduced. This simulation technique guarantees realistic simulation environments to test the designs but without danger of damaging the equipment. Also, in this iterative process of Fine-Tuning, it is usual to use different (generally conflicting/opposed) criteria that take into account the sensitivities that always appear in every project, such as economic, security, robustness, perfor-

[1] J. Velasco-Carrau, G. Reynoso-Meza, S. García-Nieto, X. Blasco Ferragud. Enhancing controller's tuning reliability with multi-objective optimisation: from Model in the loop to Hardware in the loop. Engineering Applications of Artificial Intelligence, vol. 64, pp. 52-66, sep. 2017. https://doi.org/10.1016/j.engappai.2017.05.005

mance, for example. In this framework, the use of multi-objective techniques are especially useful since they allow to study the different design alternatives based on the multiple existing criteria. Unfortunately, the combination of multi-objective techniques and verification schemes based on Hardware-In-The-Loop presents a high incompatibility. Since obtaining the optimal set of solutions requires a high computational cost that is greatly increased when using Hardware- In-the-Loop. For this reason, it is often necessary to use less realistic but more computationally efficient verification schemes such as Model in the Loop (MiL), Software in the Loop (SiL) and Processor in the Loop (PiL). In this paper, a combined methodology is presented, where multi-objective optimisation and multi-criteria decision making steps are sequentially performed to achieve a final control solution. The authors claim that while going towards the optimisation sequence over MiL → SiL → PiL → HiL platforms, the complexity of the problem is unveiled to the designer, allowing to state meaningful design objectives. In addition, safety in the step between simulation and reality is significantly increased.

5.1 Introduction

A controller tuning task typically starts with a certain nominal model of the process under consideration. With such a nominal model, and with a previously selected controller structure, the tuning process will seek a suitable controller, fulfilling several requirements and performance specifications (hereafter design objectives) imposed by the designer. Such design objectives range from time to frequency domain exigencies, requirements and/or constraints.

In spite of the usefulness of using a nominal model for controller tuning purposes, for some applications further performance evaluation is required. Therefore, with the aim of enhancing controller's performance evaluation, different platforms could be used; for example, using a *hardware in the loop* (HiL) platform has become an standard practice in order to evaluate embedded controllers, with the goal of getting a more reliable measure of their performance [21, 48]. Such platforms are common in automotive [7] and aeronautic/aerospace sectors [19], where it is required to enhance the quality, safety and verification testing of their subsystems [37].

On the other hand, it is not unusual to state a controller tuning task as an optimisation problem. The designer's task is to define one or more performance objectives to fulfil; afterwards, adjusting the tunable controller's parameters using an optimisation algorithm in order to meet such design objectives. Nevertheless, designs found with a *pure-performance* optimisation approach are often prone to be highly sensitive to the parameters used in the nominal model [27, 2, 13]; therefore, they might be useless in a practical sense. According to this, assessing robustness and reliability constraints (or objectives) has become the standard in such optimisation instances. The former lead to robust design optimisation (RDO), where the aim is to optimise the performance of the controller in the nominal model and simultaneously minimize its sensitivity; the latter leads to reliability-based design optimisation (RBDO), commonly based on stochastic analysis and its aim is to provide a measure of risk of failure [12]. Different approaches for RBDO have been used, as montecarlo sampling, simulation techniques or first/second order reliability methods [44]

Therefore, the designer is, in general, dealing with a multi-objective problem (MOP), where performance measures are in conflict with the reliability or robustness indexes. Multi-objective optimisation (MOO) has shown to be a valuable tool for controller tuning [35] when multiple and conflictive design objectives appear. It handles the simultaneous optimisation of several conflicting objectives, in order to provide what is known as the Pareto set [25], where all solutions are Pareto optimal *i.e.* they have different trade-off between conflicting objectives.

The aim of this paper is to provide a systematic approach, using (successively) different platforms in order to evaluate the controller's performance with multi-objective optimisation techniques. Reliability methods have been merged before with multi-objective optimisation [8] or HiL platforms within the MOO process [41, 49] or within the MCDM stage [14]; nevertheless new methodologies to integrate such approach when the computational burden in the HiL is considerable, might be useful for control engineers. This is because, although tuning controllers directly in a HiL set-up by means of MOO would be a perfect match, it is usually too time-demanding in practice. This time cost leads to other difficulties that make optimising *from scratch* in the HiL platform prohibitive.

Other less realistic (and less complex) platforms such as *Model in the Loop* (MiL), *Software in the Loop* (SiL) and *Processor in the Loop* (PiL) can be previously used in the multi-objective optimisation procedure. Thereby, in this paper a methodology is presented, where multi-objective optimisation and multi-criteria decision making steps are sequentially performed over those platforms, going from the least to the most complex, in order to achieve a final control solution. First, more meaningful objectives can be posed as the designer gets more knowledge about the interaction between the system and the control structure. Also preferences on the objectives are more "maturely" included. Second, objectives and decision variables bounds can be better delimited.

The remainder of this paper is as follows: in Section 5.2 brief backgrounds on controller's performance and MOO are given; in Section 5.3 the methodological proposal of this work is presented and it is evaluated in an aircraft platform in Section 5.4. The purpose will be to accomplish a certain flight mission via the supervision of several way-points autonomously, which is reported in Section 5.5. Finally, some concluding remarks and further directions of this work are commented.

5.2 Background

In this section a brief background on controller's performance evaluation in engineering design and MOO techniques will be given, in order to state a common framework for the methodological proposal in this work.

5.2.1 Controllers' evaluation in engineering design

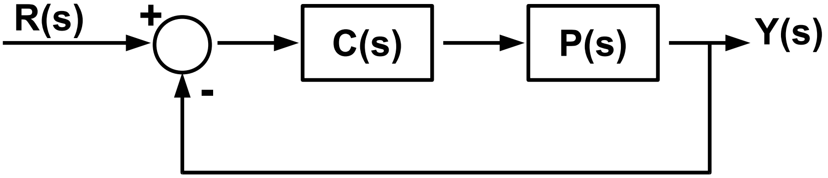


Figure 5.1. Basic control loop.

According to [2], any controller tuning procedure should consider design objectives related with:

- Load disturbance response
- Measurement noise response
- Setpoint response
- Robustness to model uncertainties

In agreement with the problem at hand, fulfilling one or some of them will be more (or less) preferable by the designer. According to the basic control loop of Fig. 5.1, some common choices in controller tuning [35] for design objectives are:

- Maximum value of sensitivity function

$$J_{M_s}(\boldsymbol{x}) = \left\|(I + P(s)C(s))^{-1}\right\|_{\infty} \tag{5.1}$$

- Integral of the absolute error value

$$J_{IAE}(\boldsymbol{x}) = \int\limits_{t=t_0}^{T_f} |r(t) - y(t)|\, dt \tag{5.2}$$

- Total variation of control action

$$J_{TV}(\boldsymbol{x}) = \int\limits_{t=t_0}^{T_f} \left|\frac{du}{dt}\right| \tag{5.3}$$

where $r(t)$, $y(t)$, $u(t)$ are the reference, measured variable and control action in time t. Equations (5.2) and (5.3) are commonly used for setpoint response and load disturbance, while for example Equation (5.1) has been used to guarantee a desired level of robustness. Time performance design objectives are usually preferred in industrial applications over frequency domain, as industrial requirements are usually expressed in such terms [26].

Different platforms are available to evaluate the performance of a controller. Regarding proximity to the real set-up, the authors are using the following division:

- *Model in the loop* (MiL): a classical approach, where a nominal model is used to calculate and evaluate the performance of a controller.
- *Software in the loop* (SiL): the approach where the controller is evaluated *as it will be embedded*; that is, using the coding/script as it will be implemented in the embedded control device.
- *Processor in the loop* (PiL): the approach where the controller is executed in the processor/device where it will be embedded. Note that this is normally a real-time simulation.
- *Hardware in the loop* (HiL): the platform where the interactions (including physical communications) among processor, sensors and actuators are placed inside the real-time simulation loop.

The goal of using one platform over another, is on the one hand, getting a more meaningful and deeper understanding of the controller's performance to be implemented; on the other hand, getting a certain grade of reliability on its performance measure. Such measure can be expressed as risk of failure [40] or with probabilistic indices [1] Hereafter, this set of platforms will be denoted as **X**iL platforms.

In any case, the conflict between robustness and performance arises [13], and therefore, MOO techniques might be an appealing tool to address the controller tuning problem.

5.2.2 Multi-objective optimisation design review

As referred in [25], a MOP [2], can be stated as follows:

$$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x}) = [J_1(\boldsymbol{x}), \ldots, J_m(\boldsymbol{x})] \tag{5.4}$$

subject to

[2] Any maximisation problem can be converted to a minimization one. For each of the objectives that have to be maximised, the transformation: $\max J_i(\boldsymbol{x}) = -min(-J_i(\boldsymbol{x}))$ could be applied.

$$\boldsymbol{K}(\boldsymbol{x}) \leq 0 \tag{5.5}$$

$$\boldsymbol{L}(\boldsymbol{x}) = 0 \tag{5.6}$$

$$\underline{x_i} \leq x_i \leq \overline{x_i}, \ \forall i = [1, \ldots, n] \tag{5.7}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ is defined as the decision vector; $\boldsymbol{J}(\boldsymbol{x})$ as the objective vector and $\boldsymbol{K}(\boldsymbol{x})$, $\boldsymbol{L}(\boldsymbol{x})$ as the inequality and equality constraint vectors respectively; $\underline{x_i}, \overline{x_i}$ are the lower and upper bounds in the decision (or search) space $\boldsymbol{X}$.

It has been pointed out that there is not a single solution in MOPs, because there is not (in general) a better solution in all the objectives. Therefore, a set of solutions, the Pareto set $\boldsymbol{\Theta}_P$, is defined. Each solution in the Pareto set defines an objective vector in the Pareto front $\boldsymbol{J}_P$. All solutions in the Pareto front conform a set of Pareto optimal and non-dominated solutions (Fig. 5.2):

Definition 6. *(Pareto optimality [25]): An objective vector $\boldsymbol{J}(\boldsymbol{x}^1)$ is Pareto optimal if there is not another objective vector $\boldsymbol{J}(\boldsymbol{x}^2)$ such that $J_i(\boldsymbol{x}^2) \leq J_i(\boldsymbol{x}^1)$ for all $i \in [1, 2, \ldots, m]$ and $J_j(\boldsymbol{x}^2) < J_j(\boldsymbol{x}^1)$ for at least one j, $j \in [1, 2, \ldots, m]$.*

Definition 7. *(Dominance [9]): An objective vector $\boldsymbol{J}(\boldsymbol{x}^1)$ is dominated by another objective vector $\boldsymbol{J}(\boldsymbol{x}^2)$ iff $J_i(\boldsymbol{x}^2) \leq J_i(\boldsymbol{x}^1)$ for all $i \in [1, 2, \ldots, m]$ and $J_j(\boldsymbol{x}^2) < J_j(\boldsymbol{x}^1)$ for at least one j, $j \in [1, 2, \ldots, m]$. This is denoted as $\boldsymbol{J(x_2)} \preceq \boldsymbol{J(x_1)}$.*

It is important to notice that the Pareto front is usually unknown, and the designer can only rely on a Pareto front approximation $\boldsymbol{J}_P^*$ and Pareto set approximation $\boldsymbol{\Theta}_P^*$. In order to successfully embed the multi-objective optimisation concept into a design process, three fundamental steps are required: the MOP statement (measure); the MOO process (search); and the multi-criteria decision making (MCDM) step (multicriteria analysis). This procedure will be named multi-objective optimisation design (MOOD) procedure (Fig. 5.3).

MOOD procedures have shown to be a valuable tool for controller tuning applications (see [33, 24], and references therein). Such techniques have been used in different controller structures [23]; for example PID controllers [47], fractional order controllers [38, 51] and state space controllers [16]. They enable the designer or decision maker (DM) to have a close embedment into the design process; since it is possible to take into account each design objective individually, they also enable comparing design alternatives, in order to select a controller fulfilling the expected trade-off among conflicting objectives. Such procedures have been used with success when (1) it is difficult to find a reasonable trade-off for a controller tuning fulfilling several requirements; and (2) it is worthwhile analysing design objectives exchange among design alternatives.

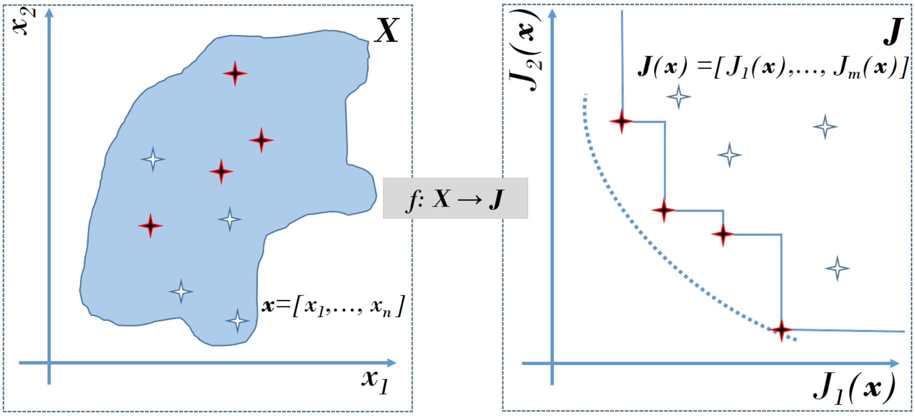
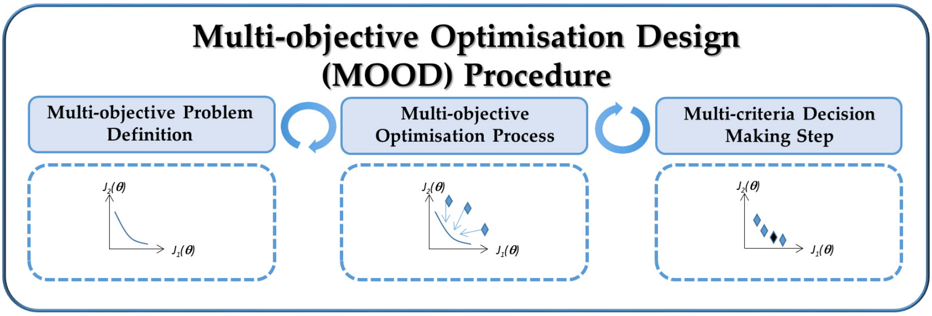


Figure 5.2. Pareto optimality and dominance concepts. A Pareto front (dotted line in objective space $\boldsymbol{J}$) is approximated with a set solutions (depicted with stars) selected from the feasible decision space $\boldsymbol{X}$. Dark solutions are non-dominated solutions in the set and therefore, they are used to build a Pareto front approximation (solid line). Remainder solutions are dominated solutions.



Multi-objective Optimisation Design (MOOD) Procedure

Multi-objective Problem Definition

Multi-objective Optimisation Process

Multi-criteria Decision Making Step

Figure 5.3. Multi-objective Optimisation Design (MOOD) procedure.

In spite of its success in controller tuning applications, few works focus on incorporating the **X**iL platforms (besides the HiL) in their procedures ([41, 49] in the MOO process or [14] in the MCDM stage). Next, an integrative framework between **X**iL platforms and the MOOD procedure for controller tuning will be presented. Here it is assumed that:

- Using directly the HiL is not always possible in the MOO process, due to computational burden.
- While going towards the sequence MiL → SiL → PiL → HiL the efforts required to evaluate a controller are gradually increased. This is because (1) the more meaningful the objectives are the more complex they become (2) setup and analysis also increase in complexity and time.
- Not only robustness but also reliability of the controller's performance will be sought.

5.3 XiL platforms within a MOOD framework for controller tuning applications

In this section, an integrative framework to enhance controller performance evaluation in MOOD procedures is given. Such framework considers using different **X**iL platforms throughout the optimisation and decision making process. Firstly, the MOP statement, the MOO process and the MCDM stage will be stated for controller tuning; afterwards, a full integration with the **X**iL platforms will be proposed.

5.3.1 The MOOD process

MOP statement

According to Section 5.2, the general MOP that must be considered in controller tuning applications is:

$$\begin{aligned}\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x}) &= [\boldsymbol{J}_{SR}(\boldsymbol{x}), \boldsymbol{J}_{LR}(\boldsymbol{x}), \boldsymbol{J}_{NR}(\boldsymbol{x}), \\ &\quad \boldsymbol{J}_{RDO}(\boldsymbol{x}), \boldsymbol{J}_{RBDO}(\boldsymbol{x}, \phi)] \qquad (5.8)\end{aligned}$$

Where $\boldsymbol{x}$ is now the vector of tunable parameters of the controller structure selected; $\boldsymbol{J}_{SR}(\boldsymbol{x})$, $\boldsymbol{J}_{LR}(\boldsymbol{x})$, $\boldsymbol{J}_{NR}(\boldsymbol{x})$ are the set of design objectives related with setpoint response, load disturbance response and measurement noise response respectively (performance objectives); $\boldsymbol{J}_{RBO}(\boldsymbol{x})$, $\boldsymbol{J}_{RBDO}(\boldsymbol{x})$ are the set of design objectives related with robust design optimisation and reliability-based design optimisation respectively (robustness/reliability objectives). RBDO design objectives will be related with the stochastic evaluation of the performance objectives for different scenarios. That is, for example, given a set of scenarios Φ and a performance objective for a given scenario $J(\boldsymbol{x}, \cdot)$, plausible design objectives are (for example) the worst case performance (Equation (5.9)) or the variance on the performance (Equation (5.10))

$$\begin{aligned} J_{worst} &= \max\left(\boldsymbol{\varsigma}\right) \\ & \quad \varsigma = J(\boldsymbol{x}, \phi), \forall \phi \in \Phi \end{aligned} \tag{5.9}$$

$$\begin{aligned} J_{var} &= \sigma\left(\boldsymbol{\varsigma}\right) \\ & \quad \varsigma = J(\boldsymbol{x}, \phi), \forall \phi \in \Phi \end{aligned} \tag{5.10}$$

While it is common to state the design objectives or constraints in a RDO sense, stating them to get reliability (RBDO) and actively using them in the optimisation process is less common. However, they are quite useful, since they provide a deeper (more reliable) insight on controller performance, its risk of failures and its expected behaviour [1, 50, 26, 40].

As a system might comprise several sub-processes and their interactions (that is, a multi-variable process), several design objectives for each of the sub-processes may appear. The designer might need to measure and optimise them all. In [31] it was intended to provide a general framework for controller tuning, dealing with such many-objective optimisation statement (usually $m > 3$). Nevertheless, in spite of its usefulness, the approximated Pareto front could contain a considerable amount of solutions, which could in turn overwhelm the DM in the MCDM stage, even for a 2x2 multi-variable process. Therefore a different approach is needed.

A feasible approach in controller tuning can be found when incorporating preferences in the optimisation process[3]. This is possible since (1), the designer has an idea of the objectives which he/she needs to meet and/or (2), a reference controller exists (manually tuned or using tuning rules)[4]. Lets assume that the designer can incorporate such preferences via the following function $\mathfrak{f}$:

[3] Correlation analysis might be useful in order to reduce the dimensionality of the problem, nevertheless they don't embed information about the preferences that might be useful for the designer.

[4] If no reference controller is available, a preliminary analysis on a Pareto front approximation (calculated without preferences) might provide the required information in order to state the preference matrix.

$$\mathfrak{f} : \boldsymbol{x} \subseteq X \rightarrow (\mathfrak{P} \vee \mathfrak{P}^C) \tag{5.11}$$

That is, $\mathfrak{f}(x)$ is a function which determines if a given solution is aligned with the designer's preferences ($\mathfrak{P}$) or not ($\mathfrak{P}^C$), regarding the desired performance of the control loop (time and/or frequency domain). Such function and preferences [5] can be incorporated numerically into the set of constraints previously stated in eq. (5.4). Hence, the MOP statement of eq. (5.8) will be subject to:

$$\begin{aligned} \boldsymbol{K}(\boldsymbol{x}) &\leq 0 && (5.12)\\ \boldsymbol{L}(\boldsymbol{x}) &= 0 && (5.13)\\ \mathfrak{f}(\boldsymbol{x}) &\in \mathfrak{P} && (5.14)\\ \underline{x_i} \leq x_i &\leq \overline{x_i}, \ \forall i = [1, \ldots, n] && (5.15) \end{aligned}$$

Given the above, an algorithm to deal with this many-objectives optimisation instance, as well as with preferences and constraints, is required.

MOO process

Several algorithms exist, and the selection of one over another should be pondered by the characteristics of the problem (for instance multimodality, many-objectives, expensive optimisation) and the expected outcome (convergence, diversity and pertinency).

The approach presented by [36] using the sp-MODEII[6] algorithm [33] will be followed in this paper. Its main characteristics are useful in order to deal with many-objective optimisation statements, covering the basic properties of convergence, diversity and pertinency of the Pareto front. Such characteristics are:

- It uses Differential Evolution [42] algorithm as optimisation engine, which has shown a good trade off between global search and convergence to the Pareto front [10].
- The objective space is partitioned in spherical sectors [34], in order to improve the spreading along the Pareto front.
- Preferences $\mathfrak{P}$ are coded *a priori* in the form of a preference matrix $\mathfrak{m}$ by the designer (Table 5.1). For each design objective $J_i(\boldsymbol{x})$, $i \in [1, \cdots, m]$ six

[5] Hereafter, *fraktur* style will be used to denote such input from the designer.

[6] Scripts and tutorials available at www.mathworks.com/matlabcentral/fileexchange/authors/289050.

Table 5.1. Typical preference matrix $\mathfrak{m}$ for GPP index. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

					Preference matrix $\mathfrak{m}$						
	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_1(\boldsymbol{x})$ [-]	J_1^0		J_1^1		J_1^2		J_1^3		J_1^4		J_1^5
$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\vdots$
$J_m(\boldsymbol{x})$ [-]	J_m^0		J_m^1		J_m^2		J_m^3		J_m^4		J_m^5

values ($J_i^0, \cdots, J_i^5$) are stated (in the original units for each design objective) in order to define 5 preference ranges: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

- Preferability function $\mathfrak{f}(\cdot)$ is computed using the above commented matrix $\mathfrak{m}$. With such preference matrix, the algorithm computes a global physical programming index (GPP) [3] to evaluate the preferability of one solution over another solution, which is a modified form of the physical programming methods [22]. Such index is used to prune the approximated Pareto front, in order to get a compact and pertinent approximation, with the number of desired solutions imposed by the designer.

- This approach enables to state a difference between design objectives for decision making and for optimisation. In the former case, they represent the design objectives where the DM is willing to perform a decision making and where the objective space is partitioned; the latter, are design objectives that are used in the optimisation stage, used to calculate the GPP index, but are not used *a priori* in the design objective partitioning. For example, in Fig. 5.4 such difference is depicted for a 3-objective problem where two Pareto-optimal solutions lie. Assume that square solution has a better GPP index than the circle solution. In the 2D projection (design objective for decision making), seems that the circle solution should be selected, since it dominates the square solution and both belong to the same spherical sector; nevertheless when considering the overall objective space in the 3D representation the selection of one over the another will rely on the $\mathfrak{f}(\boldsymbol{x})$ preferability function and thus, the square solution is selected. Such technique is helpful for many-objectives optimisation instances in controller tuning purposes [36]. Basically, we are spreading solutions in a 2D space, where the designer might feel more comfortable to analyse the Pareto front, but taking into account the m-dimensional space with the GPP index, and therefore, all design objectives are considered.

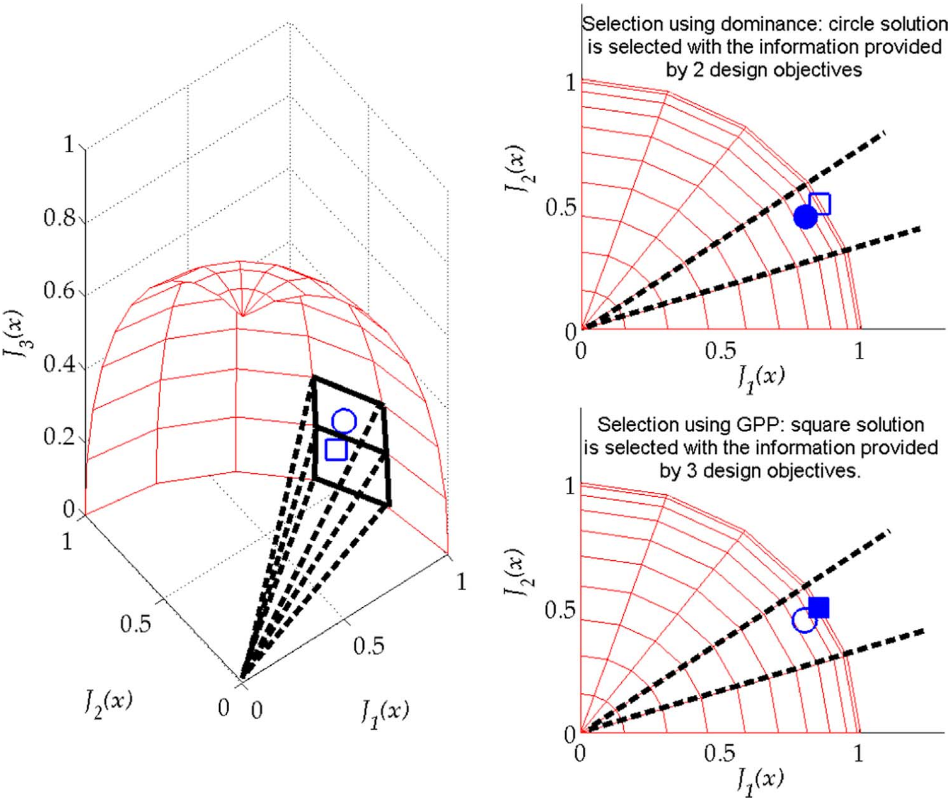


Figure 5.4. Difference between design objectives for optimisation (3-objective) and for decision making (2-objective) using dominance and preferences. Square-solution has a better GPP index for 3-objective than circle-solution.

MCDM step

The main goal in the MCDM step, is to (1) select one preferable solution $\mathfrak{x} \in \mathbf{\Theta}_P^*$ or (2) select a subset $\mathfrak{X} \subseteq \mathbf{\Theta}_P^*$ of preferable and feasible solutions for further evaluation. In any case, the DM needs to perform an accurate analysis of the $\mathbf{\Theta}_P^*$ approximation in a multidimensional and multicriteria environment. It is widely accepted that visualization tools are valuable and provide to the DM meaningful methods to analyse the Pareto front and take decisions [5, 43].

For two-dimensional problems (and sometimes for three-dimensional problems) it is usually straightforward to make an accurate graphical analysis of the Pareto front, but the difficulty increases with the dimension of the problem. Common alternatives to tackle an analysis in higher dimensions are Scatter Diagrams, Parallel Coordinates [17, 18] and Level Diagrams (LD) [4, 30]. Recently, hybrid tools merging Parallel Coordinates, Dendograms, and Cluster Maps have been proposed [6].

Given that design objectives for decision making and for optimisation will be stated, a reduced space for MCDM will be sought (up to 3 objectives); nevertheless, as auxiliary visualization, a LD tool will be used [7], due to its capabilities to propagate interpretability from $\boldsymbol{J}_P^*$ to $\mathbf{\Theta}_P^*$ and due to its robustness, scalability and simplicity properties [43]. They have been used before for design applications [28, 29] and controller tuning [15, 20]. Additionally, in order to evaluate 2-5 design alternatives, radial plots[8] are used due to their simplicity to depict few design options with several design objectives.

5.3.2 Integration to enhance controller's performance evaluation

We will state a minimal example of an integrated MOOD using the MiL and SiL platforms. Again, it is important to remember that, enhancing controller's performance evaluation is always possible with a HiL platform. What is not always possible, however, is to use such platform actively in the MOO process (globally or locally) at least *from scratch.*

The general purpose, before getting into the SiL platform, is to get a suitable set of solutions, as well as to improve the DM's knowledge on the problem's trade-off by using the information extracted from the MiL platform. That is, the designer will define a MOP at the MiL platform:

[7] Available at http://www.mathworks.com/matlabcentral/fileexchange/24042

[8] Also known as star, rose, spider diagrams.

$$
\begin{array}{rcll}
 & \displaystyle\min_{\boldsymbol{x}} \boldsymbol{J(x)}\,|_{MiL} & & \\
s.t. & & & \\
\boldsymbol{K}(\boldsymbol{x})\,|_{MiL} & \leq & 0 & \\
\boldsymbol{L}(\boldsymbol{x})\,|_{MiL} & = & 0 & \\
\mathfrak{f}_{MiL}(\boldsymbol{x}) & \in & \mathfrak{P}\,|_{MiL} & \\
\underline{x_i} \leq x_i & \leq & \overline{x_i}, & \forall i = [1, \ldots, n]
\end{array} \tag{5.16}
$$

And starting from such MOP statement, a MOO process will output the Pareto Set and Front approximations $\boldsymbol{\Theta}_P^*\,|_{MiL}$, $\boldsymbol{J}_P^*\,|_{MiL}$ respectively, with a given $\mathfrak{m}_{MiL}$.

A MCDM analysis on $\boldsymbol{\Theta}_P^*\,|_{MiL}$, $\boldsymbol{J}_P^*\,|_{MiL}$ might include the SiL platform. Design objectives could be evaluated in the SiL environment, or new indexes might be calculated (different form the $\boldsymbol{J(x)}\,|_{MiL}$ statement, but important to consider). After such procedure at the MCDM stage, the designer will select a set $\mathfrak{X} \subseteq \boldsymbol{\Theta}_P^*\,|_{MiL}$ of preferable solutions; lets denote this subset as $\mathfrak{X}_{MiL}$.

Nevertheless, if such additional indexes are important, the designer could also consider an active seeking in the SiL platform. Lets suppose this is the case, then a new MOP can be stated:

$$
\begin{array}{rcll}
 & \displaystyle\min_{\boldsymbol{x}} \boldsymbol{J(x)}\,|_{SiL} & & \\
s.t. & & & \\
\boldsymbol{K}(\boldsymbol{x})\,|_{SiL} & \leq & 0 & \\
\boldsymbol{L}(\boldsymbol{x})\,|_{SiL} & = & 0 & \\
\mathfrak{f}_{SiL}(\boldsymbol{x}, \mathfrak{X}_{MiL}) & \in & \mathfrak{P}\,|_{SiL} & \\
\underline{x_i} \leq x_i & \leq & \overline{x_i}, & \forall i = [1, \ldots, n]
\end{array} \tag{5.17}
$$

Again, after a MOO process based on the MOP from eq. (5.17), the designer will obtain the Pareto Set and Front approximations $\boldsymbol{\Theta}_P^*\,|_{SiL}$, $\boldsymbol{J}_P^*\,|_{SiL}$ respectively, with a given $\mathfrak{m}_{SiL}$. It is important to notice that the $\mathfrak{f}_{SiL}(\boldsymbol{x}, \mathfrak{X}_{MiL})$ function includes, besides preferences of the DM, the information gained in the previous MCDM step (here represented by the subset $\mathfrak{X}_{MiL}$). This is aligned with the philosophy of *innovization* [11], where information from the MOO stage is retrieved in order to gain a deeper knowledge on trade-off of the current MOP and its design objectives (that is, innovation trough optimisation).

The advantage of following this process relies on two facts. First, from the computational sense, evaluating performance of a controller in SiL platforms could be more expensive than doing it in MiL platforms; this is due to the fact that SiL platforms would include (for example) sampling rate effects (hence slowing down the simulations). Therefore, performing the MOO process *from scratch* in a SiL platform is sometimes impractical. More practical could be however, using previous information from the $\boldsymbol{\Theta}_P^* \,|_{MiL}$, $\boldsymbol{J}_P^* \,|_{MiL}$ in order to refine the preference matrix $\mathfrak{m}_{MiL}$ and the objective space bounds $\overline{x_i}$ and $\underline{x_i}$ $\forall i = [1, \ldots, n]$; furthermore, solutions $\mathfrak{X}_{MiL}$ might be used as initial population (initial candidate solution in the optimisation algorithm) to accelerate the convergence in the SiL platform. Second, from the problem knowledge sense, a progressive approximation to reality may help to gradually reduce the engineer's uncertainty about the problem. Note that an approach to reality usually means an increase in complexity. In this way, the more information the designer gets from previous iterations, the more accurate is the request on the controller performance.

Following this idea, the MOPs of Equations (5.18) and (5.19) could be defined. Such elements are summarised in Fig. 5.5.

$$\min_{\boldsymbol{x}} \boldsymbol{J(x)} \,|_{PiL} \tag{5.18}$$

$$
\begin{array}{lrcll}
s.t. & & & & \\
& \boldsymbol{K}(\boldsymbol{x}) \,|_{PiL} & \leq & 0 & \\
& \boldsymbol{L}(\boldsymbol{x}) \,|_{PiL} & = & 0 & \\
& \mathfrak{f}_{PiL}(\boldsymbol{x}, \mathfrak{X}_{MiL}, \mathfrak{X}_{SiL}) & \in & \mathfrak{P} \,|_{PiL} & \\
& \underline{x_i} \;\leq\; x_i & \leq & \overline{x_i}, & \forall i = [1, \ldots, n]
\end{array}
$$

$$\min_{\boldsymbol{x}} \boldsymbol{J(x)} \,|_{HiL} \tag{5.19}$$

$$
\begin{array}{lrcll}
s.t. & & & & \\
& \boldsymbol{K}(\boldsymbol{x}) \,|_{HiL} & \leq & 0 & \\
& \boldsymbol{L}(\boldsymbol{x}) \,|_{HiL} & = & 0 & \\
& \mathfrak{f}_{HiL}\,(\boldsymbol{x}, \mathfrak{X}_{MiL}, \mathfrak{X}_{SiL}, \mathfrak{X}_{PiL}) & \in & \mathfrak{P} \,|_{HiL} & \\
& \underline{x_i} \;\leq\; x_i & \leq & \overline{x_i}, & \forall i = [1, \ldots, n]
\end{array}
$$

It is important to remark that sometimes using all the **X**iL platforms is not possible nor even practical. As an example, the designer might use only two of them: a SiL (comprising the MiL) and the HiL (comprising the PiL). This election will highly depend on the facilities and infrastructures available for such tests, as well as on the complexity of the problem. A case of study for which this methodology could be suitable will be presented next.

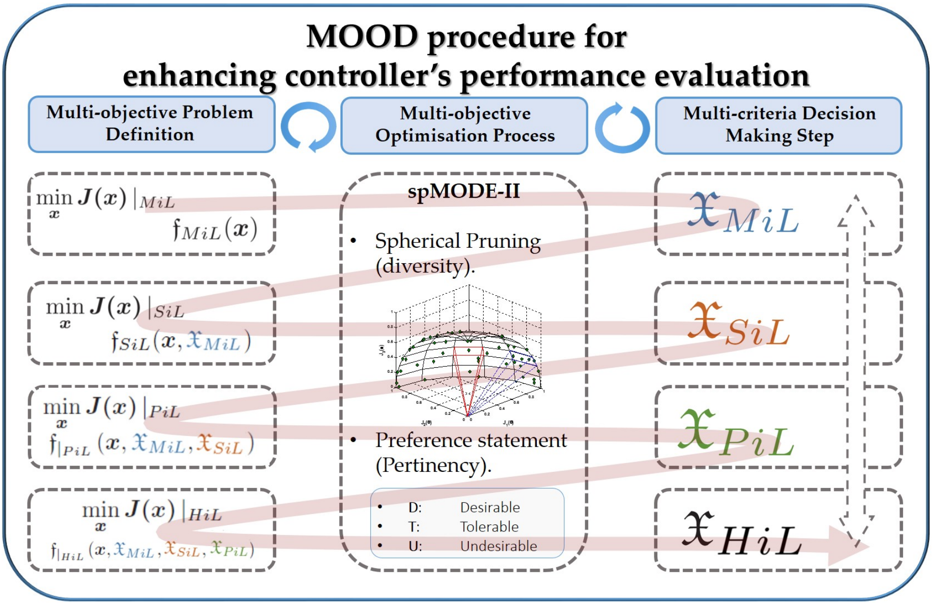

MOOD procedure for enhancing controller's performance evaluation
Multi-objective Problem Definition
Multi-objective Optimisation Process
Multi-criteria Decision Making Step
$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x})|_{MiL}$
$\mathfrak{f}_{MiL}(\boldsymbol{x})$
$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x})|_{SiL}$
$\mathfrak{f}_{SiL}(\boldsymbol{x}, \mathfrak{X}_{MiL})$
$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x})|_{PiL}$
$\mathfrak{f}_{|PiL}(\boldsymbol{x}, \mathfrak{X}_{MiL}, \mathfrak{X}_{SiL})$
$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x})|_{HiL}$
$\mathfrak{f}_{|HiL}(\boldsymbol{x}, \mathfrak{X}_{MiL}, \mathfrak{X}_{SiL}, \mathfrak{X}_{PiL})$
spMODE-II
Spherical Pruning (diversity).
Preference statement (Pertinency).
D: Desirable
T: Tolerable
U: Undesirable
$\mathfrak{X}_{MiL}$
$\mathfrak{X}_{SiL}$
$\mathfrak{X}_{PiL}$
$\mathfrak{X}_{HiL}$

Figure 5.5. The Multi-objective Optimisation Design Procedure (MOOD) for controller tuning using **X**iL platforms.

5.4 Methodology implementation

As commented before, **X**iL platforms are very useful for automotive and aeronautic applications. Therefore, in order to validate the usability of this methodology using MOOD procedures, the attitude and navigation control of an aircraft intended to perform way-points supervision tasks will be tuned. Hence, MOPs increase in complexity not only because of the change of platform but also because of the addition of new objectives.

5.4.1 System description

The aircraft for test and validation is presented in Fig. 5.6. The main component of the UAV flight system is a Kadett 2400 aircraft manufactured by Graupner. The aircraft has a very lightweight frame and characteristics that make it suitable for the purposes of this research. These characteristics include a 2.4 m wing span, 0.9 m^2 of wing surface, 48.07 N/m^2 wing loading, and 1.65×10^{-2} m^3 of available volume to house control hardware.

During normal flight, the tail rudder, elevators, and ailerons serve as the control surfaces. Propulsion is provided by a brushless alternating current engine supplied by two lithium-ion polymer (LiPo) batteries through a frequency variator. The variator and the servomotors are controlled by pulse width modulated (PWM) command signals.

The flight control station (FCS), housed in an ARM-based microcontroller, hosts the control algorithms. The control loop is closed by a IG500N unit from SBG Systems, that integrates a wide range of sensors, including the accelerometers, gyroscopes, and magnetometers. A Kalman filter fuses the sensor information to estimate position, orientation, linear and angular speed, and acceleration. This same platform was presented with more detail in [45, 46] together with the results of the first flight tests.

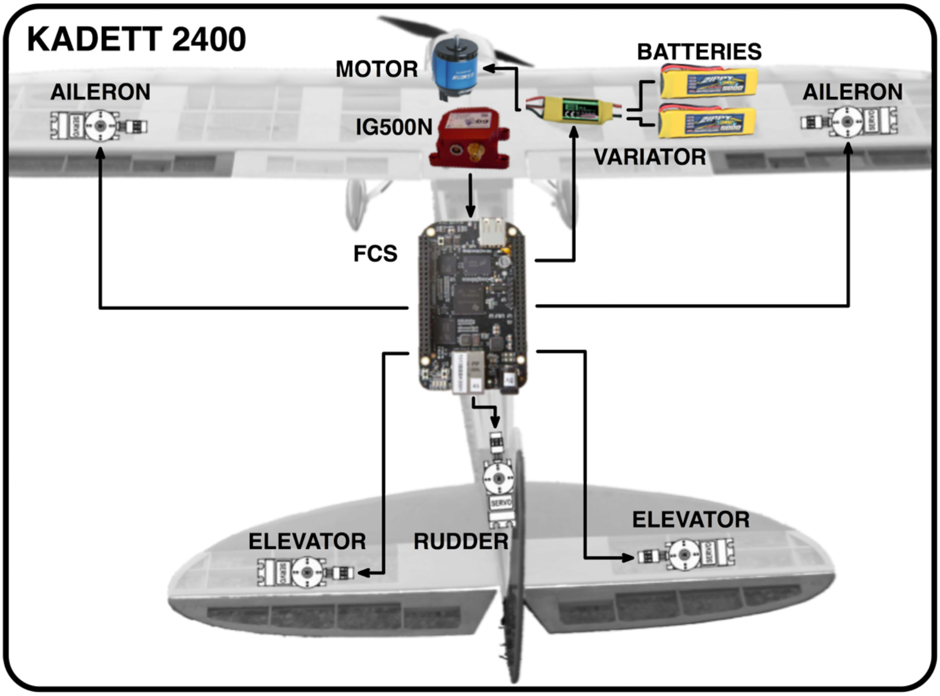


Figure 5.6. Hardware elements on board experimental platform, the aircraft Kadett 2400

The purpose of the aircraft (for this paper), is to perform a supervision of several way-points. As general approach and without loss of generality, a proportional-integral (PI) controller structure is selected to drive each control variable to its set-point.

$$C(s) = k_p \left(1 + \frac{1}{T_i s}\right) \tag{5.20}$$

where k_p is the proportional gain and T_i the integral time.

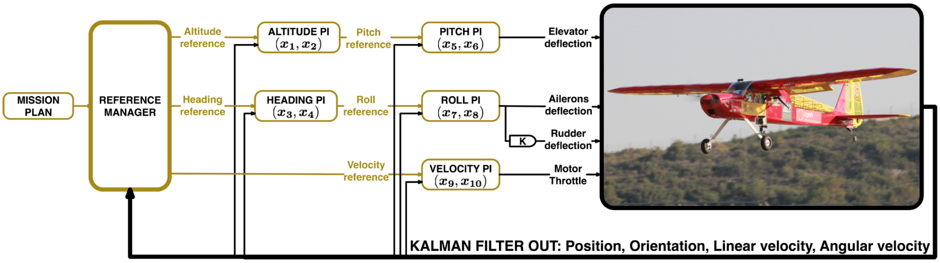


Figure 5.7. Kadett's control loop structure. Control and guidance over a user-defined mission.

The complete control structure is a set of five PI regulators as the one shown in Fig. 5.7. Three references (altitude, heading and velocity) are served to three of the five regulators. Those references are calculated by a reference manager based on the mission plan (way-points through which the aircraft must pass) and the aircraft current position. Thrust is directly applied to the motor as result of the velocity's PI, whereas pitch and roll references are respectively obtained from altitude and heading regulators. Finally, pitch and roll PIs generate deflections to be applied on elevators and ailerons. A total of ten variables $[x_1, \ldots, x_{10}]$, five pairs of the form (k_p,T_i), must then be adjusted so that a set of user-defined objectives becomes Pareto optimal.

5.4.2 The MOOD-XiL definition

Now, following the methodology, three **X**iL platforms are presented for this example:

MiL: A non-linear model has been identified according to [45, 46][9]; roughly speaking, multi-objective optimisation techniques have been used to adjust several constants of a first principle model of our aircraft.

SiL: The same non-linear model is used, but here the controller's scripts are added as they will be coded in the FCS; this includes different sampling rates (50 ms for outer loops and 20 ms for inner loops), together with bumpless transfer and anti-windup mechanisms.

[9] This model has been implemented in Simulink© Matlab© version R2013a, with ode3(Bogacki-Shampine) solver with a fixed-step size of 1ms.

HiL: A National Instruments PXI with a real time running model substitutes most of the hardware components; those components are "virtualized" so that the FCS can be added to the loop without any software modification, *i.e.* as it is programmed in a real flight (see Fig. 5.8).

Given these three platforms, a particularized version of the methodology depicted in Fig. 5.5 can be posed for the problem at hand. As we can see, only MiL and SiL platforms are used actively in the MOO process whereas all three (MiL, SiL, and HiL) are employed in the MCDM stage. This is because the current HiL platform's infrastructure does not allow for complete integration with the MOO algorithm. The MIL platform will be used with a simple control test in order to identify suitable controllers in the optimisation phase; afterwards, with such results a new optimisation process will be carried out in the SiL platform, where a flight mission will be stated and used in order to evaluate the performance of a given controller. Finally, after this optimisation phase using the simulation model, a final decision making process will be carried out in the HiL platform.

Additionally, a reference controller $\boldsymbol{x}_{ref}$, adjusted via pole placement, is available from previous works. A reference controller is useful for two main reasons: (*i*) it provides a rough first idea of what preferences might be reasonable to ask for; (*ii*) it can be taken as a starting point for the MOO process in the first round.

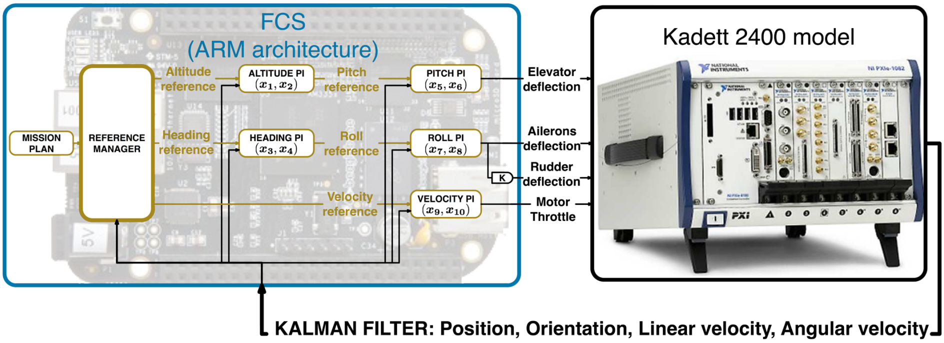


Figure 5.8. HiL platform for Kadett aircraft.

MOOD at the MiL platform

This MOOD statement will be named hereafter, *TimePiL* (time performance in the loop) and it will be stated as a RDO instance. For this step the MOP definition is:

$$\min_{\boldsymbol{x}} \boldsymbol{J}(\boldsymbol{x})\,|_{MiL} = [J_{M_1}(\boldsymbol{x}), \ldots, J_{M_7}(\boldsymbol{x})] \tag{5.21}$$

$$
\begin{array}{rcl}
 & s.t. & \\
\mathfrak{f}_{MiL}(\boldsymbol{x}) & \in & \mathfrak{P}|_{MiL} \\
\underline{x_i} \leq x_i & \leq & \overline{x_i}, i = [1, \ldots, n]
\end{array}
$$

with $\boldsymbol{x} = [k_{p_1}, T_{i_1}, \cdots, k_{p_5}, T_{i_5}]$ and $|k_p| \in [0, 5]$ and $T_i \in]0, 50] \cup \{+\infty\}$; a setpoint response is evaluated for a simultaneous change in heading and altitude. The design objectives are:

- $J_{M_1}(\boldsymbol{x})$: Settling time for yaw at ±2%.
- $J_{M_2}(\boldsymbol{x})$: Settling time for altitude at ±2%.
- $J_{M_3}(\boldsymbol{x})$: Throttle's total variation of control action (Eq. (5.3)).
- $J_{M_4}(\boldsymbol{x})$: Aileron's total variation of control action (Eq. (5.3)).
- $J_{M_5}(\boldsymbol{x})$: Elevator's total variation of control action (Eq. (5.3)).
- $J_{M_6}(\boldsymbol{x})$: Roll's total variation of control action (Eq. (5.3)).
- $J_{M_7}(\boldsymbol{x})$: Pitch's total variation of control action (Eq. (5.3)).

Design objectives $J_{M_1}(\boldsymbol{x})$ and $J_{M_2}(\boldsymbol{x})$ are stated for performance ($J_{SR}(\boldsymbol{x})$) while $J_{M_3}(\boldsymbol{x})$ to $J_{M_7}(\boldsymbol{x})$ for robustness ($J_{RBO}(\boldsymbol{x})$), since total variation is a valid measure for robustness [39].

The preference matrix $\mathfrak{m}_{MiL}$ is depicted in Table 5.2. The idea behind preferences $J_{M_3}(\boldsymbol{x})$ to $J_{M_7}(\boldsymbol{x})$ is to provide some meaning to the values obtained from Eq. (5.3) by posing them relative to the reference controller $\boldsymbol{x}_{ref}$. This idea comes from the fact that the total variation of a control action by itself does not provide the same level of interpretability as, for example, time related indexes $J_{M_1}(\boldsymbol{x})$ and $J_{M_2}(\boldsymbol{x})$, for which preferences are easy to state. This has been previously exposed in [32]. For example, for J_{M_3} in Table 2 it has been defined as a tolerable value up to a 10% of additional control effort of the reference controller $\boldsymbol{x}_{ref}$. The desirable value ranges from a reduction of 10% and 20% of such control action, and the highly desirable value for a reduction up to 30%.

Now $J_{M_1}(\boldsymbol{x})$ to $J_{M_7}(\boldsymbol{x})$ are used in the MOO process. Pareto set $\boldsymbol{\Theta}_P^*|_{MiL}$ and front $\boldsymbol{J}_P^*|_{MiL}$ are obtained for the seven design objectives, however only $J_{M_1}(\boldsymbol{x})$ and $J_{M_2}(\boldsymbol{x})$ are analysed in the MCDM stage; this means that, while all the design objectives are considered in the evolution process, and used to calculate the GPP index in the pruning mechanism of the spMODE-II algorithm, only the first two (the most interpretable) are used to partition the objective space. Optimisations were carried out in a desktop computer, with Intel^{R} Cor^{TM} i7-4790

Table 5.2. Preference matrix $\mathfrak{m}_{MiL}$ for the TimePiL ($\boldsymbol{J(x)}|_{MiL}$) statement. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU). For readability purposes, $J_{M_i}(\boldsymbol{x_{ref}})$ has been substituted by $\widehat{J_{M_i}}$.

Preference Matrix											
	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_{M_1}(\boldsymbol{x})$ [s]	10		15		20		25		50		100
$J_{M_2}(\boldsymbol{x})$ [s]	10		20		30		40		80		160
$J_{M_3}(\boldsymbol{x})$ [-]	$0.7 \cdot \widehat{J_{M_3}}$		$0.8 \cdot \widehat{J_{M_3}}$		$0.9 \cdot \widehat{J_{M_3}}$		$1.1 \cdot \widehat{J_{M_3}}$		$1.2 \cdot \widehat{J_{M_3}}$		$1.4 \cdot \widehat{J_{M_3}}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\vdots$
$J_{M_7}(\boldsymbol{x})$ [-]	$0.7 \cdot \widehat{J_{M_7}}$		$0.8 \cdot \widehat{J_{M_7}}$		$0.9 \cdot \widehat{J_{M_7}}$		$1.1 \cdot \widehat{J_{M_7}}$		$1.2 \cdot \widehat{J_{M_7}}$		$1.4 \cdot \widehat{J_{M_7}}$

processor, 3.60GHz and RAM memory 32GB; a total of 5000 function evaluations were computed in a time lapse of 6h27m.

The resulting Pareto set and front approximations $\boldsymbol{\Theta}_P^*|_{MiL}$ and $\boldsymbol{J}_P^*|_{MiL}$ are illustrated in Fig. 5.9. From such Figure, and with the help of the SiL platform, a subset $\mathfrak{X}_{MiL}$ (depicted as ◯) is selected for the next step within the MOOD-**X**iL procedure.

MOOD definition at the SiL platform

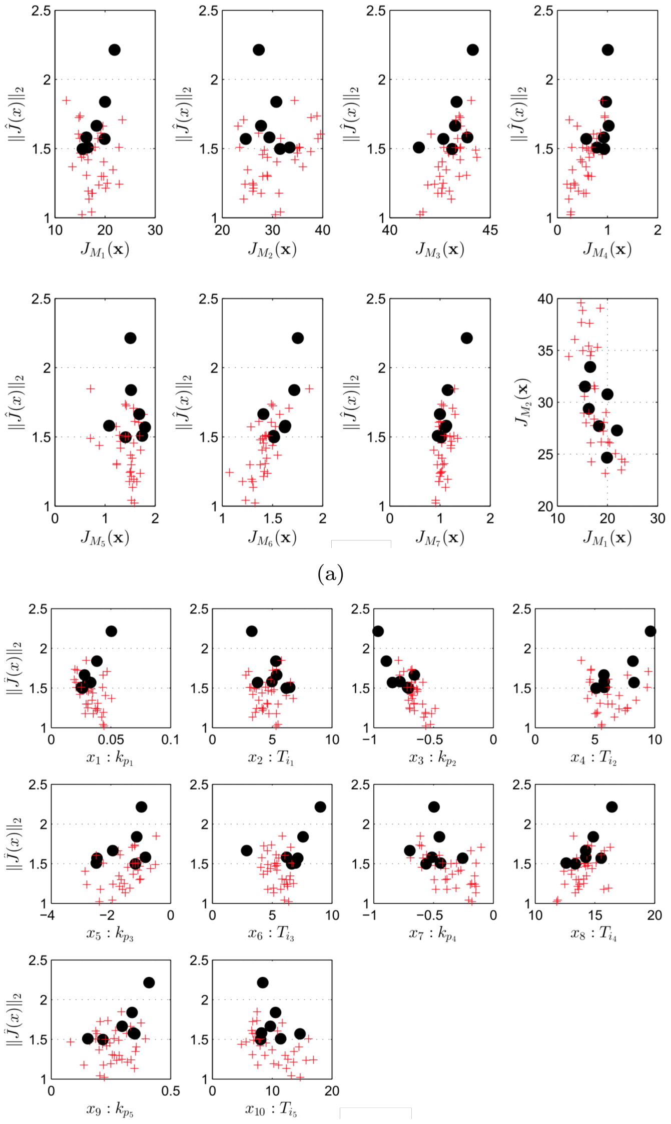
The performance of the adjusted control structure with a given mission ϕ will be evaluated in this round. For that reason, this MOOD statement will be named hereafter, *MissionPiL* (mission performance in the loop). Such mission comprises the supervision of five different way points in a bounded air space. MissionPiL statement is intended to be a RBDO instance, where reliability on controller's performance to fulfil different flight missions is evaluated. For this purpose, a set Φ of 15 flight missions are defined. Each mission has five randomly distributed and feasible (according to the characteristics of the aircraft) way-points.

Therefore, the MOP definition for this step stays as follows:

$$\min_{\boldsymbol{x}} \boldsymbol{J(x)}|_{SiL} \quad = \quad [J_{S_1}(\boldsymbol{x}), \ldots, J_{S_9}(\boldsymbol{x})] \tag{5.22}$$

$$\begin{array}{rcl} & s.t. & \\ \mathfrak{f}_{SiL}(\boldsymbol{x}, \mathfrak{X}_{MiL}) & \in & \mathfrak{P}|_{SiL} \\ \underline{x_i} \leq x_i & \leq & \overline{x_i}, i = [1, \ldots, n] \end{array}$$

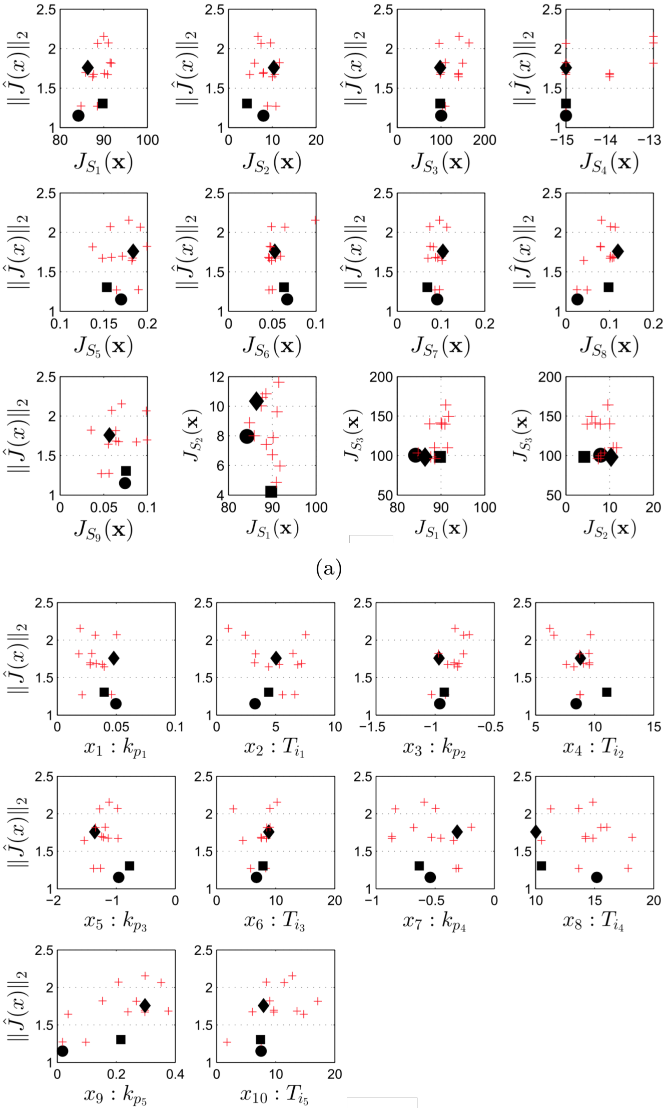
with $\boldsymbol{x} = [k_{p_1}, T_{i_1}, \cdots, k_{p_5}, T_{i_5}]$ and $k_p \in [0, 5]$ and $T_i \in]0, 50] \cup \{+\infty\}$; The design objectives stated are:



(a)

(b)

Figure 5.9. Pareto Front (a) and Pareto Set (b) for TimePiL statement at MiL platform. Solutions marked with ○ are the subset $\mathfrak{X}_{MiL}$ from $\mathbf{\Theta}_P^*|_{MiL}$ selected for further analysis.



(a)

(b)

Figure 5.10. Pareto Front (a) and Pareto Set (b) for MissionPiL statement at SiL platform. Solutions marked with ○,□,◇ are the subset $\mathfrak{X}_{SiL}$ from $\mathbf{\Theta}_P^*|_{SiL}$ selected for further analysis.

- $J_{S_1}(\boldsymbol{x})$: Median of time required to perform a flight mission ς[s] $\forall \phi \in \Phi$:

$$\begin{aligned} J_{S_1}(\boldsymbol{x}) &= median(\boldsymbol{\varsigma}) \\ \varsigma &= MissionTime(\boldsymbol{x}, \phi), \forall \phi \in \Phi \end{aligned} \tag{5.23}$$

- $J_{S_2}(\boldsymbol{x})$: Median absolute deviation (MAD) of time required [s] to perform a flight mission $\forall \phi \in \Phi$

$$\begin{aligned} J_{S_2}(\boldsymbol{x}) &= median(|\boldsymbol{\varsigma} - J_{S_1}(\boldsymbol{x})|) \\ \varsigma &= MissionTime(\boldsymbol{x}, \phi), \forall \phi \in \Phi \end{aligned} \tag{5.24}$$

- $J_{S_3}(\boldsymbol{x})$: Maximum time required [s] to perform a flight mission $\forall \phi \in \Phi$ (Eq. (5.9)).
- $J_{S_4}(\boldsymbol{x})$: (Negative) number of successful fight missions.
- $J_{S_5}(\boldsymbol{x})$: Median of roll's total variation of control action per flight time duration (Eq. (5.25)).
- $J_{S_6}(\boldsymbol{x})$: Median of pitch's total variation of control action per flight time duration (Eq. (5.25)).
- $J_{S_7}(\boldsymbol{x})$: Median of elevator's total variation of control action per flight time duration (Eq. (5.25)).
- $J_{S_8}(\boldsymbol{x})$: Median of throttle's total variation of control action per flight time duration (Eq. (5.25)).
- $J_{S_9}(\boldsymbol{x})$: Median of aileron's total variation of control action per flight time duration (Eq. (5.25)).

$$\begin{aligned} J_{TV_2}(\boldsymbol{x}) &= median\,(\boldsymbol{\upsilon}) \\ \upsilon &= \left[\int\limits_{t=t_0}^{T_f} \left|\frac{du}{dt}\right|\right] \cdot \left[\frac{1}{\varsigma}\right], \forall \phi \in \Phi \end{aligned} \tag{5.25}$$

The preference matrix $\mathfrak{m}_{SiL}$ is depicted in Table 5.3. In this case, preferences values are in accordance with the values observed when testing $\mathfrak{X}_{MiL}$ solutions in

Table 5.3. Preference matrix $\mathfrak{m}_{SiL}$ for the MissionPiL statement. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

Preference Matrix $\mathfrak{m}_{SiL}$

	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_{S_1}(\boldsymbol{x})$ [s]	60		85		90		95		120		150
$J_{S_2}(\boldsymbol{x})$ [s]	5		7		10		15		30		50
$J_{S_3}(\boldsymbol{x})$ [s]	60		95		120		175		240		300
$J_{S_4}(\boldsymbol{x})$ [-]	-15		-15		-15		-13		-10		-5
$J_{S_5}(\boldsymbol{x})$ [-]	0.10		0.15		0.18		0.20		0.25		0.30
$J_{S_6}(\boldsymbol{x})$ [-]	0.10		0.40		0.50		0.67		0.10		0.15
$J_{S_7}(\boldsymbol{x})$ [-]	0.05		0.08		0.10		0.12		0.15		0.20
$J_{S_8}(\boldsymbol{x})$ [-]	0.02		0.05		0.08		0.12		0.15		0.20
$J_{S_9}(\boldsymbol{x})$ [-]	0.02		0.05		0.08		0.10		0.15		0.20

the SiL platform. Indeed, when controllers optimised for **M**iL platform are confronted to the more realistic **S**iL platform, they reveal additional information and trade-off among design objectives. When passing through this "experience", the DM reaches better understanding on the capabilities of the overall control structure for the problem at hand, and hence, he/she is able to glimpse the limits, in terms of performance, that the control algorithm can be led to. Therefore, mixing the observed trade-off in the preference matrix $\mathfrak{m}_{SiL}$, aligns with the search of a controller able to satisfy every preference simultaneously. In the same way, initialization bounds are designed given the results of the $\boldsymbol{\Theta}_P^* |_{MiL}$. Optimisations were carried out in a desktop computer, with Intel^{R} Cor^{TM} i7-4790 processor, 3.60GHz and RAM memory 32GB; a total of 528 function evaluations were computed in a time lapse of 33h03m.

Again the Pareto set and front approximations $\boldsymbol{\Theta}_P^* |_{SiL}$ and $\boldsymbol{J}_P^* |_{SiL}$ result from the MOO process. They can be visualized in Fig. 5.10. For example, circle solution has a better performance on J_{S_1} when compared with the square solution; nevertheless, this is at expenses of worsening J_{S_2}. That is, a trade-off between median performance and dispersion when evaluated with the flight missions set. Similar analysis can be performed with the remainder plots in level diagrams. After the MCDM analysis, a subset $\mathfrak{X}_{SiL}$ of three suitable controllers (in their trade-off sense) is selected.

Final MCDM stage at the HiL platform

In order to help with the final decision making, the subset $\mathfrak{X}_{SiL}$ will be analysed in the HiL platform. Although an active MOO is not possible at the moment (and could even be impractical from a computational point of view), manually implementing those pre-selected controllers in the HiL might definitely shed some light on which controller is the best final choice. Five random missions of eight way-points each have been generated for this purpose. The three controllers $\mathfrak{x}_1, \mathfrak{x}_2, \mathfrak{x}_3$ from $\mathfrak{X}_{SiL}$, together with the reference controller, have been implemented in the aircraft FCS. To ensure the most similar initial conditions, each flight path comes preceded by three extra way-points that constitute the initialisation track. Those way-points are only used for homogenisation purposes and hence they are not considered in the performance analysis. Since five sets of data are available for each controller, cost functions $J_{S_i} \forall i = \{1, ..., 9\}$ can be now obtained.

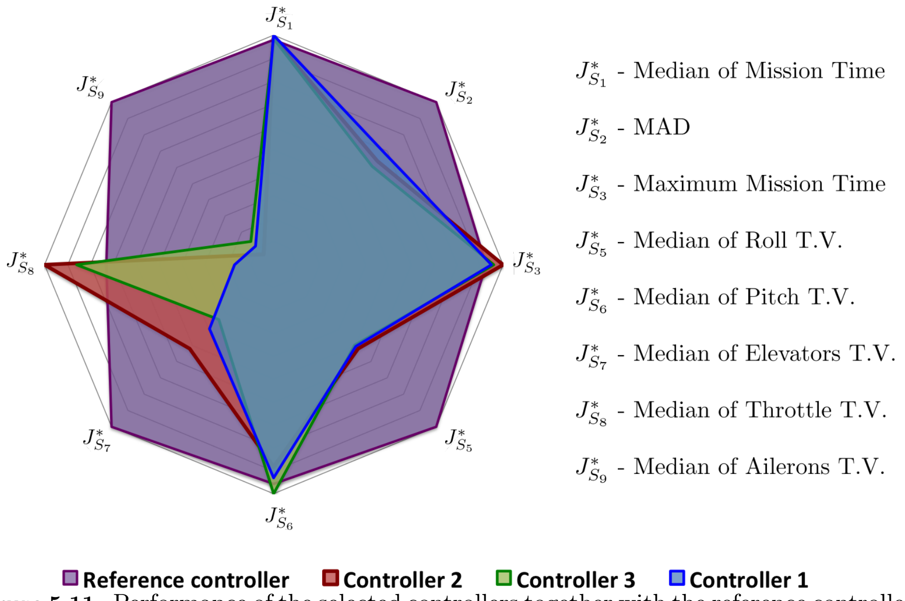


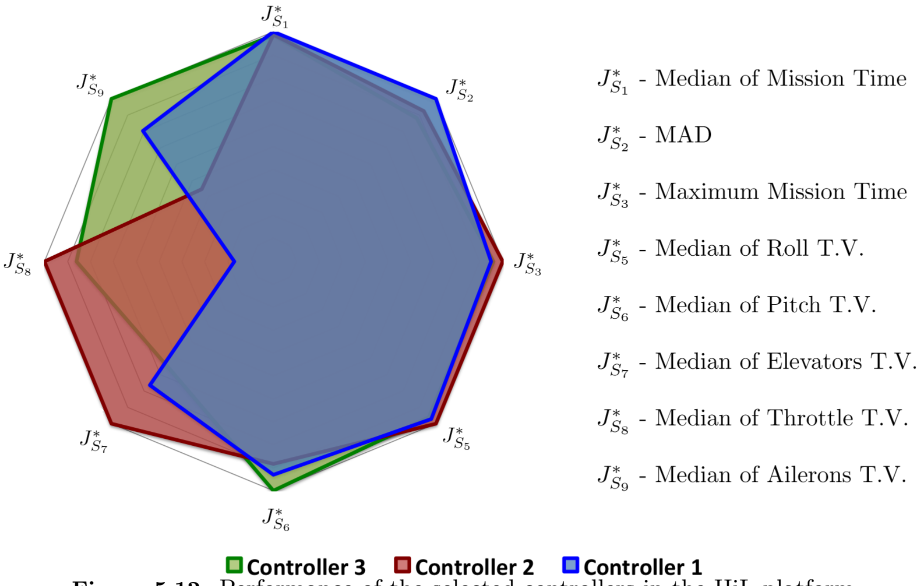
Figure 5.11. Performance of the selected controllers together with the reference controller in the HiL platform

Figure 5.11 is a radial representation of the values of J_{S_i} functions with a particularity. This radial representation gives an idea about the trade-off of a particular solution when compared with others. A solution covering all the radial space is a solution which is worsening all design objectives. The smaller the area, the closer to the ideal solution (center of the representation). All indices J_{S_i} in Fig.5.11 have been scaled over their worst value. This is, if a specific controller shows a value of 1 in one cost index, it means that such controller has the worst performance observed for that particular objective (see Eq. (5.26)). Additionally, objective 4 has been removed from this analysis because all controllers were able to complete the five flight missions successfully. When observing a radial representation like the one in Fig. 5.11, the closeness to the nadir solution and utopian solution are directly related to the vertices position of the polygon given by one specific solution[10]. Hence, those controllers whose vertices are closer to the center are to be preferred, since they fulfil each individual objective with a better performance. We can then conclude from Fig. 5.11 that $\mathfrak{X}_{SiL}$ controllers outperform the reference one. They do not only show smaller costs for most of the objectives, but also have a very similar value in those for which the reference regulator is slightly better.

$$J^*_{S_i}(\boldsymbol{x}) = J_{S_i}(\boldsymbol{x})/J^{max}_{S_i} \tag{5.26}$$

where

$$\begin{aligned} J^{max}_{S_i} = \quad & \max\{J_{S_i}(\mathfrak{x}_1), J_{S_i}(\mathfrak{x}_2), J_{S_i}(\mathfrak{x}_3), \\ & J_{S_i}(\boldsymbol{x_{ref.}})\} \end{aligned}$$



$J^*_{S_1}$ - Median of Mission Time

$J^*_{S_2}$ - MAD

$J^*_{S_3}$ - Maximum Mission Time

$J^*_{S_5}$ - Median of Roll T.V.

$J^*_{S_6}$ - Median of Pitch T.V.

$J^*_{S_7}$ - Median of Elevators T.V.

$J^*_{S_8}$ - Median of Throttle T.V.

$J^*_{S_9}$ - Median of Ailerons T.V.

Figure 5.12. Performance of the selected controllers in the HiL platform

[10] In an approximation set, nadir solution is the objective vector with the worst values in the set; on the opposite, the utopian solution is the objective vector with the better values contained in the set.

Table 5.4. Parameter of the selected controllers

	Altitude PI		Heading PI		Pitch PI		Roll PI		Velocity PI	
Controller	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1	0.0497	3.2483	−0.9604	8.4343	−0.9591	6.7427	−0.5409	15.1889	0.0179	7.5268
2	0.0398	4.4099	−0.9232	11.0291	−0.7739	7.8579	−0.6339	10.5024	0.2155	7.4387

Now, if the reference controller is removed from the analysis a new perspective is achieved. Thereby, the three controllers $\mathfrak{x}_1$, $\mathfrak{x}_2$ and $\mathfrak{x}_3$, that result from our MOO methodology are illustrated again in Fig. 5.12. The first fact we can observe is that they are very similar in performance when the first six objectives are taken into account. Differences appear though, when low-level actuators are analysed. As we see, $\mathfrak{x}_1$ has its vertices closer to the center than the other two, being much better when throttle total variation is considered, and with an intermediate performance for elevators and ailerons. Note that smooth behaviours in the engine are directly related to lower energy consumptions. Hence, $\mathfrak{x}_1$ is to be preferred as the final controller. $\mathfrak{x}_2$ behaves very well with ailerons and turns out to be the worst in managing throttle and elevators. On the opposite, $\mathfrak{x}_3$ is smoother with elevators and throttle, but more aggressive when using the ailerons. Controllers 1 and 2 have been tested in real flight. Since HiL platform includes the aircraft FCS, controllers were tested exactly as they were coded for the simulations. Results from both flights and some conclusions are now exposed in Section 5.5.

5.5 Results and validation on Flight test

Controllers selected for further validation appear in Table 5.4. A mission of six way-points (plus three from the initialisation track) has been programmed and performed in real flight. A way-point is defined in the 3-D space by its latitude and longitude coordinates and its altitude above sea level. Around each of them, a tolerance cylinder (defined by an altitude error and a plain circle) is placed. Therefore, a way-point is considered to be passed when the aircraft is targeting that way-point and enters its tolerance cylinder. Besides, the flight path has been randomly generated to lie within the volume enclosed inside a 500 m diameter and 150 m height cylinder and does not share any of the way-points previously used in simulations.

Figs. 5.13 and 5.14 and Tables 5.5 and 5.6 have been included in order to present flight test's results. On one side, Figs. 5.13 and 5.14 show the resulting trajectories (cyan line) along the mission path (red line), obtained by controllers 1 and 2 respectively. Reference tracking of altitude and velocity are also represented. The tolerance cylinders are depicted by green circles around the way-points (graphs on the top) and two green lines underneath and above the altitude references (graphs in the middle). To support discussion, Table 5.5 includes the value of the mean IAE (MIAE) got by each controller for every controlled variable. MIAE gives a sense of the tracking error in average, and is easy to interpret since it has the magnitude of the variable for which it is calculated. The expression to calculate MIAE is

$$J_{MIAE}(\boldsymbol{x}) = \frac{1}{T_m} \int\limits_{t=0}^{T_m} \left| r_{\boldsymbol{x}}(t) - y_{\boldsymbol{x}}(t) \right| dt \qquad (5.27)$$

where T_m is the mission time, $r_{\boldsymbol{x}}(t)$ is the reference signal and $y_{\boldsymbol{x}}(t)$ is the controlled variable value during a mission in which controller $\boldsymbol{x}$ is active. Finally, Table 5.6 shows the values of the subset of SiL design objectives that can be calculated for a single real flight. They are:

- $J_{R_1}(\boldsymbol{x})$: time required to perform the flight mission (in representation of $[J_{S_1}(\boldsymbol{x}), J_{S_2}(\boldsymbol{x}), J_{S_3}(\boldsymbol{x})]$).
- $J_{R_2}(\boldsymbol{x})$: roll total variation of control action per flight time duration (in representation of $J_{S_5}(\boldsymbol{x})$).
- $J_{R_3}(\boldsymbol{x})$: pitch total variation of control action per flight time duration (in representation of $J_{S_6}(\boldsymbol{x})$).
- $J_{R_4}(\boldsymbol{x})$: elevators total variation of control action per flight time duration (in representation of $J_{S_7}(\boldsymbol{x})$).
- $J_{R_5}(\boldsymbol{x})$: throttle total variation of control action per flight time duration (in representation of $J_{S_8}(\boldsymbol{x})$).
- $J_{R_6}(\boldsymbol{x})$: ailerons total variation of control action per flight time duration (in representation of $J_{S_9}(\boldsymbol{x})$).

Now, going back to Figs. 5.13 and 5.14, we can see that both controllers are able to successfully complete the mission, driving the aircraft through every way-point of the flight path. Every time the UAV enters the tolerance cylinder of its targeted way-point, a new reference is imposed by the reference manager. This process keeps going until the last way-point is reached and the mission is finished. Although the reference manager also imposes different velocities for each path section, accomplishing them is not a requirement of the mission. Several points can be highlighted from Figs. 5.13 and 5.14. First, $\mathfrak{x}_1$ makes the aircraft draw smother trajectories with more opened turns; this has an influence in the trajectory length and consequently, in the mission time (for which $\mathfrak{x}_2$ is slightly better). Second, every change in the aircraft orientation strongly disturbs altitude tracking and even more velocity tracking; this is logic, since we are dealing with a coupled non-linear system. Third, while $\mathfrak{x}_1$ is better in sticking to the altitude reference, it is significantly worse than $\mathfrak{x}_2$ when trying to follow velocity references. Finally, the reader should note that velocity tracking is not within the design objectives, and hence, no cost index explicitly accounts for velocity tracking performance; even

Table 5.5. Mean IAE indices for the five controlled variables achieved by controller 1 and 2 in a real flight experiment

	Mean IAE of the controlled variables				
Objective	$J(\mathfrak{x}_1)$	$J(\mathfrak{x}_2)$	J^{max}	$J^*(\mathfrak{x}_1)$	$J^*(\mathfrak{x}_2)$
Mean IAE Roll	0.06	0.08	0.08	**0.73**	1.00
Mean IAE Pitch	0.03	0.04	0.04	**0.91**	1.00
Mean IAE Heading	0.71	0.68	0.71	1.00	**0.96**
Mean IAE Altitude	4.17	4.98	4.98	**0.84**	1.00
Mean IAE Velocity	2.94	1.51	2.94	1.00	**0.51**

Table 5.6. Objective function values achieved by controller 1 and 2 in a real flight experiment

	Cost indices for real flight test				
Objective	$J(\mathfrak{x}_1)$	$J(\mathfrak{x}_2)$	J^{max}	$J^*(\mathfrak{x}_2)$	$J^*(\mathfrak{x}_2)$
J_{R_1} - Mission Time	110.13	104.84	110.13	1.00	**0.95**
J_{R_2} - Roll Total Variation	0.17	0.21	0.21	**0.80**	1.00
J_{R_3} - Pitch Total Variation	0.06	0.08	0.08	**0.74**	1.00
J_{R_4} - Elevators Total Variation	0.15	0.33	0.33	**0.44**	1.00
J_{R_5} - Throttle Total Variation	0.04	0.24	0.24	**0.15**	1.00
J_{R_6} - Ailerons Total Variation	0.16	0.11	0.16	1.00	**0.69**

so, both controllers are able to drive the system towards the velocity references; this is interesting, and might be explained under the assumption that it could be a correlation between velocity tracking and the success in a flight mission.

If Fig. 5.7 is looked, three control flows are observed. On the top, the reference manager proposes altitude references to be achieved by the altitude PI, which in turn proposes pitch references to be accomplished by the pitch PI acting over elevators deflections. If we now return to Tables 5.5 and 5.6, it can be observed that $\mathfrak{x}_1$ is softer than $\mathfrak{x}_2$ both managing elevators and proposing pitch references (66% in the case of J_{R_4} and 26% in the case of J_{R_3}); and even so, $\mathfrak{x}_1$ is capable of outperforming $\mathfrak{x}_2$ by obtaining 9% and 16% smaller MIAEs for pitch and altitude respectively. This last data confirms the better behaviour in altitude tracking already observed in Figs. 5.13 and 5.14. The second control flow shown in Fig. 5.7 is the lateral control, where the reference manager imposes heading references to the heading PI, which in turn proposes roll references tracked by the roll PI acting over ailerons deflections. In this case $\mathfrak{x}_2$ is 31% smoother with the ailerons ($J^*_{R_6}(\mathfrak{x}_2) = 0.69$) and slightly better in heading tracking (4% smaller heading

MIAE), while $\mathfrak{x}_1$ achieves a 26% smaller roll total variation J_{R_3}, and 27% better roll MIAE; it means that the heading PI in $\mathfrak{x}_1$ is softer proposing control actions and, at the same time, $\mathfrak{x}_1$'s roll PI is better in following them, at the cost of a higher usage of the ailerons. This is again aligned with what was observed in Figs. 5.13 and 5.14, where controller 2 exhibits closer turns than controller 1. Finally, Fig. 5.13 showed that $\mathfrak{x}_1$ is slower in converging velocity to its set-point; this fact is also supported by the values on Table 5.5 where velocity's MIAE of controller 2 is 49% smaller than that of controller 1. However, Table 5.6 evidences that the throttle total variation of $\mathfrak{x}_1$ is 85% smaller ($J^*_{R_5}(\mathfrak{x}_1) = 0.15$) than the throttle total variation obtained by $\mathfrak{x}_2$.

Fig. 5.15 has been included to derive final conclusions. Two radial graphs are present in that figure. On the top, (Fig. 5.15 (a)) the values of the six cost indices $\{J_{R_1}, ..., J_{R_6}\}$ are represented both for $\mathfrak{x}_1$ and $\mathfrak{x}_2$. The values obtained from the HIL platform for controllers 1 and 2 have been again represented in Fig. 5.15 (b). Two remarks must be mentioned. First, note that only six over the nine indices are represented; this is because only the six SiL design objectives that can be directly compared to the real flight indices J_{R_i}, have been included in the graph. Second, those indices have been re-scaled, taking now into account only the values obtained by $\mathfrak{x}_1$ and $\mathfrak{x}_2$. Thus, Fig. 5.15 (b) can be seen as a version of Fig. 5.12 with a smaller amount of objectives and without $\mathfrak{x}_3$. Something remarkable from Fig. 5.15 is the resemblance among graphs (a) and (b). Although obviously not equal, the shapes that one controller gets for the HiL platform and for the real flight are quite similar. This is a significant fact because clearly shows the importance of having an accurate model of the system and reliable **X**iL platforms for the RBDO statement in the MOOD procedure. And there is where the strength of our methodology resides. With a good dynamic model, a designer passing through every step gains real knowledge on the design problem and is able to understand what should be optimised and how to do it. In addition, a realistic HiL platform (only achievable with a good dynamic model) assures safety in the hop from simulation to real experimentation. All this is finally translated in a controller that is well designed and behaves in reality as expected from simulations. As a final comment, Fig. 5.15 (a) proves that $\mathfrak{x}_1$ should be preferred over $\mathfrak{x}_2$, as its vertices are closer to the center. This was already concluded is Section 5.4, and evidences that a final MCDM stage with several simulations in the HiL platform is indeed a good practice.

5.6 Conclusions and future work

A systematic approach to enhance controller performance evaluation and design has been presented throughout this paper. Multi-objective optimisation is used in conjunction with different simulation platforms in order to provide the integrative framework on which the methodology is based. Thereby, MiL, SiL, PiL and HiL platforms (or a subset of them) can be employed in successive multi-objective optimisations in order to gain better understanding of the problem as the process moves forward. As we saw, the information obtained in previous stages is used in two directions when a new optimisation is to be posed: (*i*) more meaningful objectives can be stated and (*ii*) preferences and solution constraints can be better delimited. On one side, (*i*) generally leads to more complex MOPs, what obviously increases the computational burden. On the other side, (*ii*) reduces the search space helping, therefore, to decrease that complexity during the optimisation stage. Both (*i*) and (*ii*) go in the direction of getting more adequate solutions (in the sense of what the designer prefers) for the problem at hand.

As a demonstrator, a UAV system with a predefined control structure has been presented. That structure is formed by a total of five PI controllers that perform attitude control and navigation tasks. The ten PI parameters had to be tuned so that the resulting controller was able to drive the aircraft in the supervision of several way-points. Section 5.4 showed how the methodology can be adapted to this specific problem, in accordance to the available simulation platforms. As the process moved forward, reliability objectives, where several missions were actively accomplished inside the MOO, were posed. At the same time, a better understanding of the problem allowed the designer to refine preferences' matrices, as well as optimisation limits. A final MCDM stage comprising several missions inside the HiL platform has been performed. In that last phase, a controller was chosen from the rest.

To validate the final choice, a real flight mission has been carried out. The chosen controller and a second one within the final Pareto set have been tested exactly as they were coded for HiL simulations. In Section 5.5 experimental results confirmed that the selected controller was as good accomplishing flight missions as the MCDM stage suggested. This evidenced that a realistic model of the system is of great importance to obtain a controller that behaves as the designer expects. A second conclusion is that, even when the HiL platform cannot be actively used in the MOO process, including it in the final MCDM stage is a good practice.

Several new flight paths can be taken as future works. First, adapting the HiL simulation platform so that it can be actively used in the MOO process could be of great interest (although practicality must be had in mind). Second, the authors would like to study the applicability of the technique on the design of multi-variable controllers. Indeed, tuning weighting matrices for linear quadratic regulators or adjusting design parameters in model based predictive controllers can be something tricky when the system comprises many states and inputs. Third, applying this technique to systems with modelled parametric uncertainties can be a good practice to achieve a certain level of robustness.

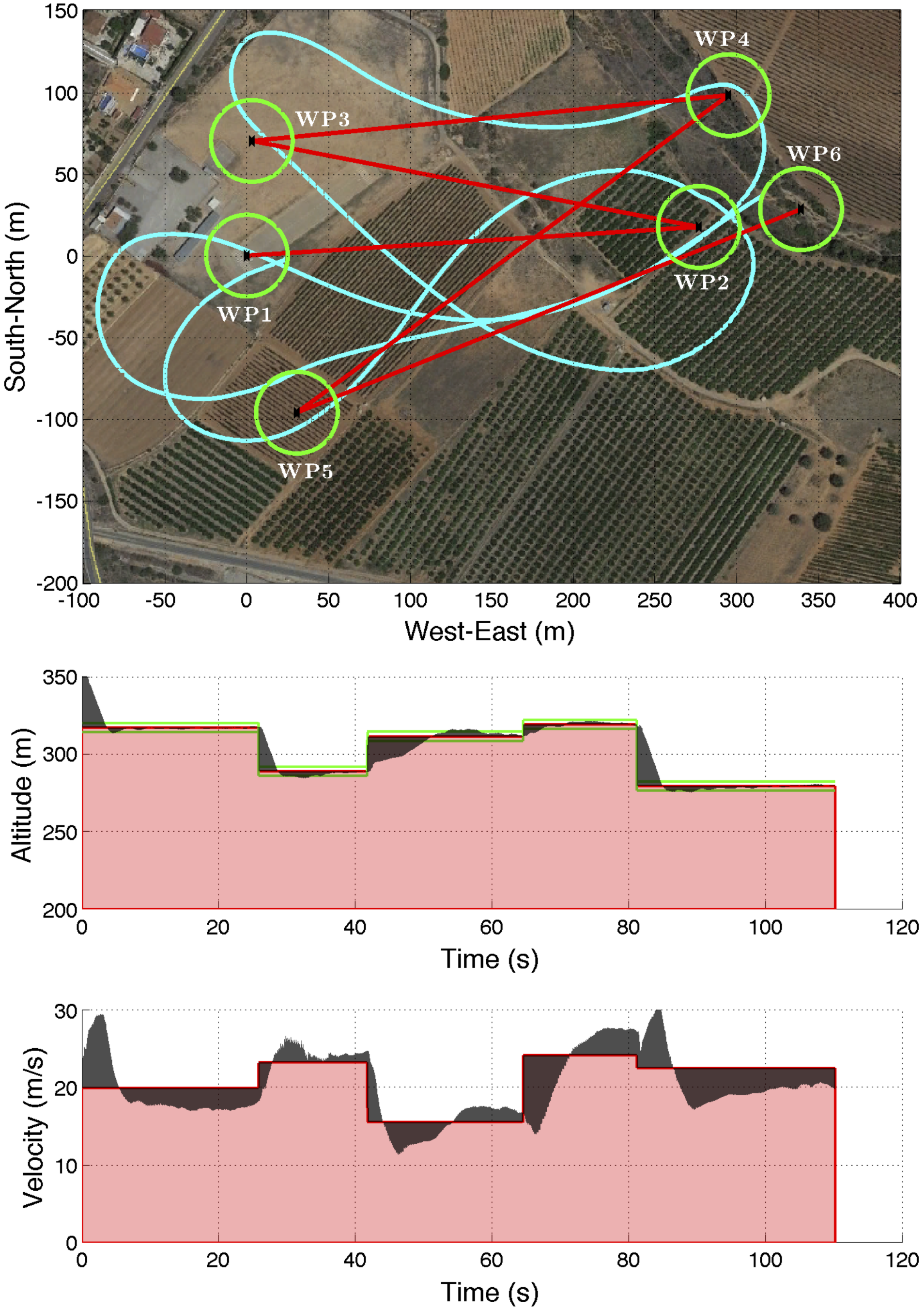


Figure 5.13. Performance of the Controller 1 in the Kadett in a real flight mission.

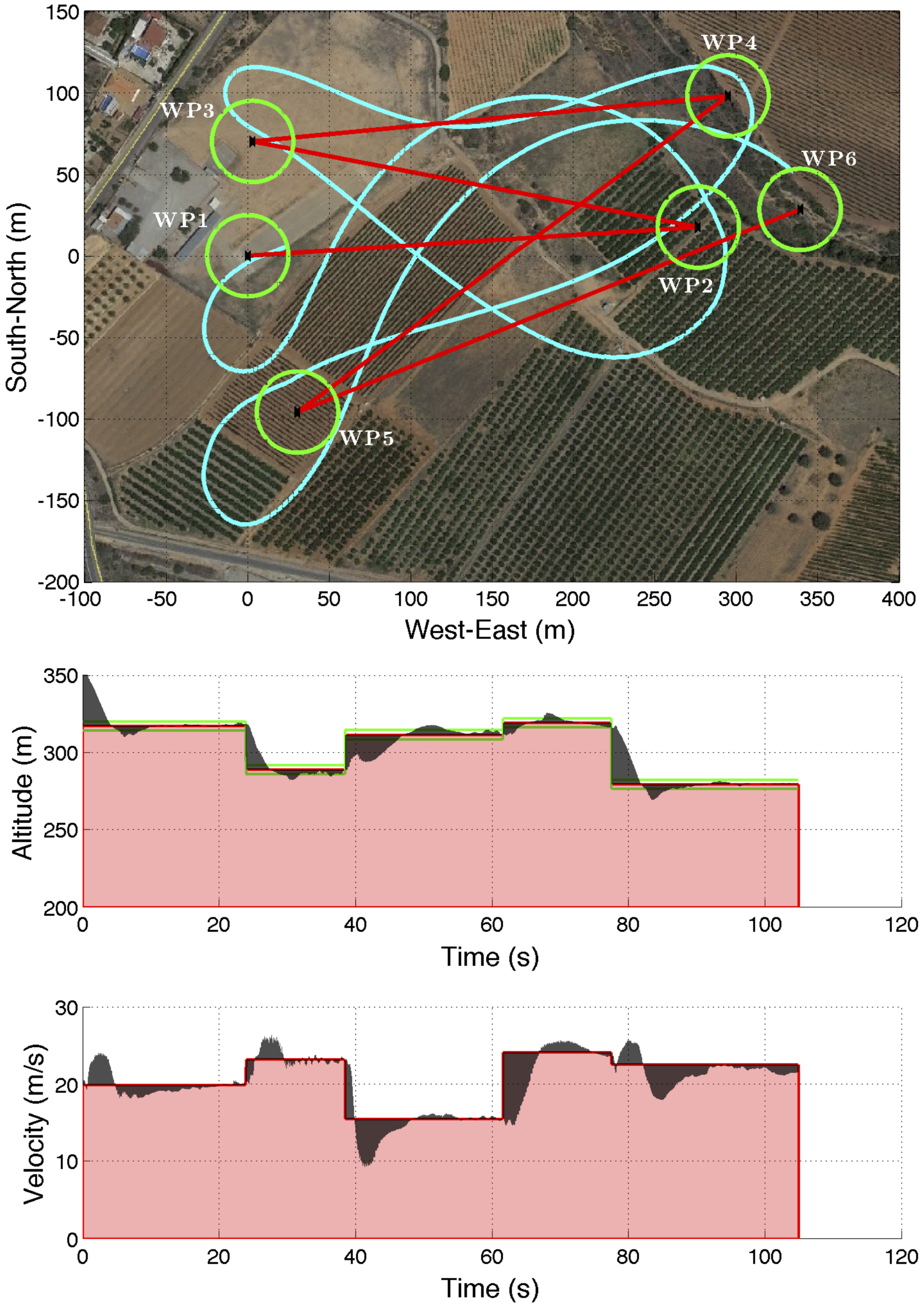
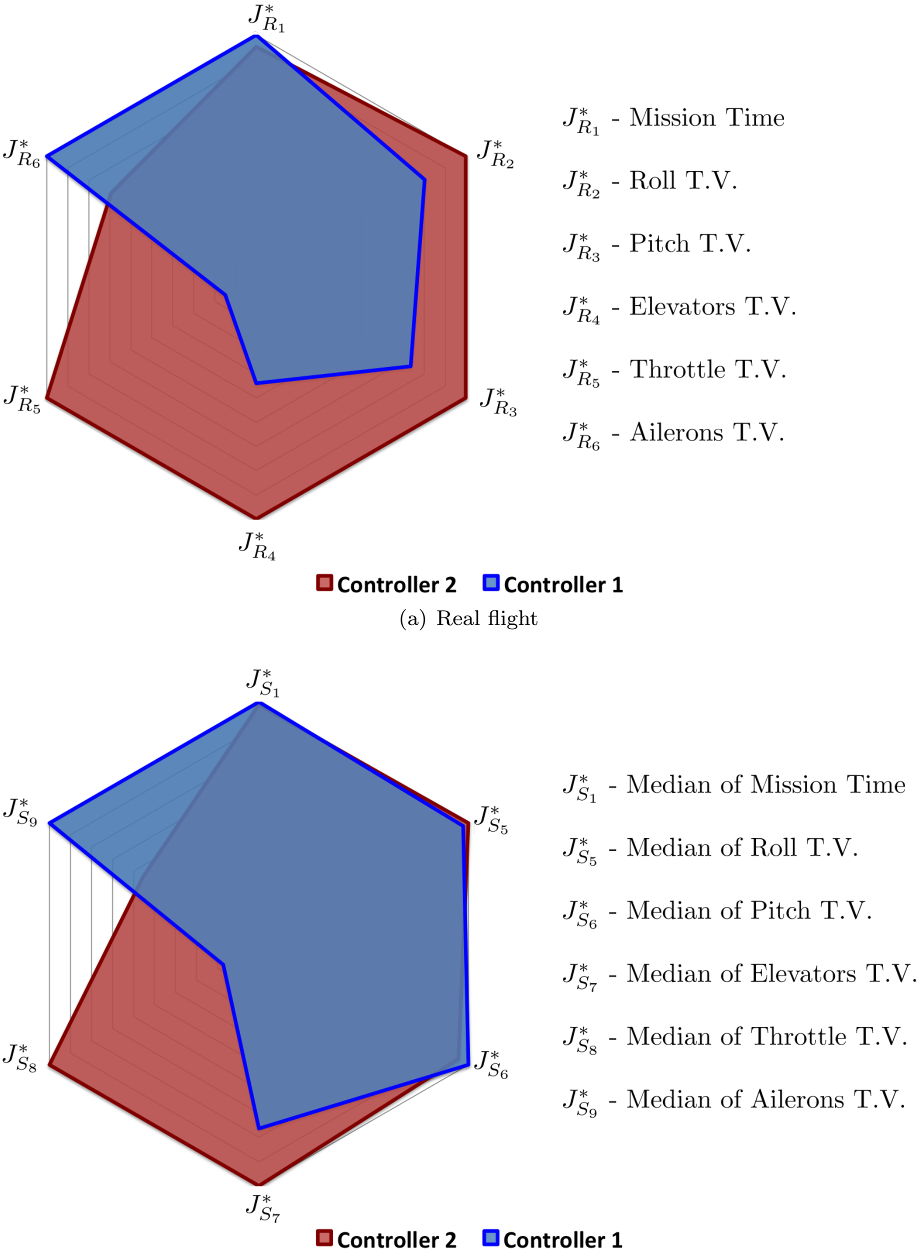


Figure 5.14. Performance of the Controller 2 in the Kadett in a real flight mission.



(a) Real flight

(b) HIL flights

Figure 5.15. Scaled Pareto front of the selected controllers during Real flight

Acknowledgement

The authors would like to acknowledge the Spanish Ministry of Economy and Competitiveness for providing funding through the project DPI2015-71443-R and the grant BES-2012-056210. This work has been partially supported by the National Council of Scientific and Technological Development of Brazil (CNPq) through the BJT/304804/2014-2 and PQ-2/304066/2016-8 grants.

References

Bibliography

[1] Alireza Alfi, Amirreza Shokrzadeh, and Mehrnoosh Asadi. Reliability analysis of h-infinity control for a container ship in way-point tracking. *Applied Ocean Research*, 52:309–316, 2015.

[2] Karl Johan Åström and Tore Hägglund. The future of pid control. *Control engineering practice*, 9(11):1163–1175, 2001.

[3] Xavier Blasco, Sergio García-Nieto, and Gilberto Reynoso-Meza. Autonomous trajectory control of a quadricopter vehicle. simulation and evaluation. *Revista Iberoamericana de Automática e Informática Industrial*, 9(2):194 – 199, 2012.

[4] Xavier Blasco, Juan Manuel Herrero, Javier Sanchis, and Miguel Martínez. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences*, 178(20):3908 – 3924, 2008.

[5] P.P. Bonissone, R. Subbu, and J. Lizzi. Multicriteria decision making (MCDM): a framework for research and applications. *IEEE Computational Intelligence Magazine*, 4(3):48 –61, 2009.

[6] R. Cela and M.H. Bollaín. New cluster mapping tools for the graphical assessment of non-dominated solutions in multi-objective optimization. *Chemometrics and Intelligent Laboratory Systems*, 114(0):72 – 86, 2012.

[7] Chinchul Choi and Wootaik Lee. Analysis and compensation of time delay effects in hardware-in-the-loop simulation for automotive pmsm drive system. *Industrial Electronics, IEEE Transactions on*, 59(9):3403–3410, Sept 2012.

[8] Rajan Filomeno Coelho. Probabilistic dominance in multiobjective reliability-based optimization: theory and implementation. *IEEE Transactions on Evolutionary Computation*, 19(2):214–224, 2015.

[9] Carlos A. Coello Coello and Gary B. Lamont. *Applications of Multi-Objective evolutionary algorithms.* World scientific publishing, advances in natural computation vol. 1 edition, 2004.

[10] S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, PP(99):1 –28, 2010.

[11] Kalyanmoy Deb, Sunith Bandaru, David Greiner, António Gaspar-Cunha, and Cem Celal Tutum. An integrated approach to automated innovization for discovering useful design principles: Case studies from engineering. *Applied Soft Computing*, 15(0):42 – 56, 2014.

[12] Dan M Frangopol and Kurt Maute. Life-cycle reliability-based optimization of civil and aerospace structures. *Computers & structures*, 81(7):397–410, 2003.

[13] Olof Garpinger, Tore Hägglund, and Karl Johan Åström. Performance and robustness trade-offs in pid control. *Journal of Process Control*, 24(5):568–577, 2014.

[14] Dan Gladwin, Paul Stewart, Jill Stewart, Rui Chen, and Edward Winward. Improved decision support for engine-in-the-loop experimental design optimization. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 224(2):201–218, 2010.

[15] A. Hajiloo, N. Nariman-zadeh, and Ali Moeini. Pareto optimal robust design of fractional-order pid controllers for systems with probabilistic uncertainties. *Mechatronics*, 22(6):788 – 801, 2012. Special Issue on Intelligent Mechatronics (LSMS2010 & ICSEE2010).

[16] Kaveh Hassani and Won-Sook Lee. Multi-objective design of state feedback controllers using reinforced quantum-behaved particle swarm optimization. *Applied Soft Computing*, 41:66 – 76, 2016.

[17] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1:69–91, 1985.

[18] Alfred Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications.* Springer, 2009.

[19] S.W. Jeon and Seul Jung. Hardware-in-the-loop simulation for the reaction control system using pwm-based limit cycle analysis. *Control Systems Technology, IEEE Transactions on*, 20(2):538–545, March 2012.

[20] T Koetje, M Braae, and M Tsoeu. Multi-objective performance evaluation of controllers for a thermal process. In *Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering*, pages 131–145. Springer, 2013.

[21] Bin Lu, Xin Wu, H. Figueroa, and A. Monti. A low-cost real-time hardware-in-the-loop testing approach of power electronics controls. *Industrial Electronics, IEEE Transactions on*, 54(2):919–931, April 2007.

[22] Achille Messac. Physical programming-effective optimization for computational design. *AIAA journal*, 34(1):149–158, 1996.

[23] Gilberto Reynoso Meza, Xavier Blasco Ferragud, Javier Sanchis Saez, and Juan Manuel Herrero Durá. *Controller tuning with evolutionary multiobjective optimization: A holistic multiobjective optimization design procedure*, volume 85. Springer, 2016.

[24] Gilberto Reynoso Meza, Xavier Blasco Ferragud, Javier Sanchis Saez, and Juan Manuel Herrero Durá. Background on multiobjective optimization for controller tuning. In *Controller Tuning with Evolutionary Multiobjective Optimization*, pages 23–58. Springer, 2017.

[25] Kaisa M. Miettinen. *Nonlinear multiobjective optimization.* Kluwer Academic Publishers, 1998.

[26] Stig Moberg, Jonas Öhr, and Svante Gunnarsson. A benchmark problem for robust feedback control of a flexible manipulator. *Control Systems Technology, IEEE Transactions on*, 17(6):1398–1405, 2009.

[27] H. Panagopoulos and K.J. Åström. Pid control design and h-infinity loop shaping. *International Journal of Robust and Nonlinear Control*, 10(15):1249–1261, 2000.

[28] A.T.D. Perera, R.A. Attalage, K.K.C.K. Perera, and V.P.C. Dassanayake. A hybrid tool to combine multi-objective optimization and multi-criterion decision making in designing standalone hybrid energy systems. *Applied Energy*, 107:412 – 425, 2013.

[29] S. Pourzeynali, S. Salimi, and H. Eimani Kalesar. Robust multi-objective optimization design of tmd control device to reduce tall building responses against earthquake excitations using genetic algorithms. *Scientia Iranica*, 20(2):207 – 221, 2013.

[30] Gilberto Reynoso-Meza, Xavier Blasco, Javier Sanchis, and Juan M. Herrero. Comparison of design concepts in multi-criteria decision-making using level diagrams. *Information Sciences*, 221:124 – 141, 2013.

[31] Gilberto Reynoso-Meza, Sergio García-Nieto, Javier Sanchis, and Xavier Blasco. Controller tuning using multiobjective optimization algorithms: a global tuning framework. *IEEE Transactions on Control Systems Technology*, 21(2):445–458, 2013.

[32] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Roberto Z Freire. Evolutionary multi-objective optimisation with preferences for multivariable pi controller tuning. *Expert Systems with Applications*, 51:120–133, 2016.

[33] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Sergio García-Nieto. Multiobjective evolutionary algortihms for multivariable PI controller tuning. *Applied Soft Computing*, 24:341 – 362, 2014.

[34] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Miguel Martínez. Multiobjective design of continuous controllers using differential evolution and spherical pruning. In Cecilia Di Chio et Al., editor, *Applications of Evolutionary Computation, Part I*, volume LNCS 6024, pages 532–541. Springer-Verlag, 2010.

[35] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Miguel Martínez. Controller tuning using evolutionary multi-objective optimisation: current trends and applications. *Control Engineering Practice*, 1:58 – 73, 2014.

[36] Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco, and Miguel Martínez. Preference driven multi-objective optimization design procedure for industrial controller tuning. *Information Sciences*, 339:108–131, 2016.

[37] Tariq Samad and George Stewart. Perspectives on innovation in control systems technology: compatibility with industry practices. *Control Systems Technology, IEEE Transactions on*, 21(2):284–288, 2013.

[38] Helem Sabina Sánchez, Fabrizio Padula, Antonio Visioli, and Ramon Vilanova. Tuning rules for robust fopid controllers based on multi-objective optimization with fopdt models. *ISA transactions*, 66:344–361, 2017.

[39] Helem Sabina Sanchez and Ramon Vilanova. Multiobjective tuning of pi controller using the nnc method: Simplified problem definition and guidelines for decision making. In *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on*, pages 1–8. IEEE, 2013.

[40] Robert F Stengel and Christopher I Marrison. Robustness of solutions to a benchmark control problem. *Journal of guidance, control, and dynamics*, 15(5):1060–1067, 1992.

[41] Paul Stewart, DA Stone, and PJ Fleming. Design of robust fuzzy-logic control systems by multi-objective evolutionary methods with hardware in the loop. *Engineering Applications of Artificial Intelligence*, 17(3):275–284, 2004.

[42] Rainer Storn and Kenneth Price. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341 – 359, 1997.

[43] Tea Tušar and Bogdan Filipič. Visualization of pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation*, 19, 2014.

[44] Marcos A. Valdebenito and Gerhart I. Schuëller. A survey on approaches for reliability-based optimization. *Structural and Multidisciplinary Optimization*, 42(5):645–663, Nov 2010.

[45] Jesús Velasco and Sergio García Nieto. Unmanned aerial vehicles model identification using multi-objective optimization techniques. *IFAC Proceedings Volumes*, 47(3):8837–8842, 2014.

[46] Jesús Velasco-Carrau, Sergio García-Nieto, Jose Vicente Salcedo, and Robert H. Bishop. Multi-objective optimization for wind estimation and aircraft model identification. *Journal of Guidance, Control, and Dynamics*, 39(2):372–389, 2015.

[47] Zhu Wang, Qixin Su, and Xionglin Luo. A novel htd-cs based pid controller tuning method for time delay continuous systems with multi-objective and multi-constraint optimization. *Chemical Engineering Research and Design*, 115:98 – 106, 2016.

[48] Andrew White, Guoming George Zhu, and Jongeun Choi. Hardware-in-the-loop simulation of robust gain-scheduling control of port-fuel-injection processes. *Control Systems Technology, IEEE Transactions on*, 19(6):1433–1443, 2011.

[49] Piotr Woźniak. Preferences in multi-objective evolutionary optimisation of electric motor speed control with hardware in the loop. *Applied Soft Computing*, 11(1):49–55, 2011.

[50] Daniil Yurchenko and Panagiotis Alevras. Stability, control and reliability of a ship crane payload motion. *Probabilistic Engineering Mechanics*, 38:173–179, 2014.

[51] Abbas-Ali Zamani, Saeed Tavakoli, and Sadegh Etedali. Fractional order pid control design for semi-active control of smart base-isolated structures: A multi-objective cuckoo search approach. *ISA Transactions*, 67:222 – 232, 2017.

Chapter 6

Motion Equations and Attitude Control in the Vertical Flight of a VTOL Bi-Rotor UAV [1]

Abstract

This paper gathers the design and implementation of the control system that allows an unmanned Flying-wing to perform a Vertical Take-Off and Landing (VTOL) maneuver using two tilting rotors (Bi-Rotor). Unmanned Aerial Vehicles (UAVs) operating in this configuration are also categorized as Hybrid UAVs due to their ability of having a dual flight envelope: hovering like a multi-rotor and cruising like a traditional fixed-wing, providing the opportunity of facing complex missions in which these two different dynamics are required. This work exhibits the Bi-Rotor nonlinear dynamics, the attitude tracking controller design and also, the results obtained through Hardware-In-the-Loop (HIL) simulation and experimental studies that ensure the controller's efficiency in hovering operation.

[1] Sergio García-Nieto, Jesús Velasco-Carrau, Federico Paredes-Valles, José Vicente Salcedo and Raúl Simarro. Motion Equations and Attitude Control in the Vertical Flight of a VTOL Bi-Rotor UAV. Electronics, vol. 8, n.o 2, p. 208, feb. 2019. https://doi.org/10.3390/electronics8020208

6.1 Introduction

In recent years, the continuous development in engineering-related fields, as automatic systems, flight control and the aerospace industry as a whole, has contributed to the rapid growth of the area of Unmanned Aerial Vehicles (UAV), becoming to represent an appealing research topic in both military and civil applications. In terms of civil application it is important to mention those related with agricultural services, marine operations, natural disaster support, etc. Within the military field, UAVs are mostly used in missions in which, due to the high risk, the presence of a human-pilot is not justified.

In order to increase the number and complexity, and so the performance efficiency, of these applications, UAVs characterized by a dual flight envelop are currently needed. This unmanned vehicles, inheriting the advantages of both traditional fixed-wing aircraft and rotorcraft, have the ability to execute a VTOL maneuver and to inspect aggressively a certain area, as well as to perform a high-speed aerial surveillance over a wide region. For the aforementioned reasons, this vehicles are known as Hybrid UAVs.

According to [27], hybrid UAVs can be categorized into two main types: Convertiplanes and Tail-Sitters. Firt of all, Convertiplanes category regroup those aerial vehicles that take off, cruise, hover and land with the aircraft reference line remaining horizontal. Respect to this class, there exist several vehicles implementing the idea such as FireFLY6 [36] and TURAC [1]; and also projects researching in this direction [12, 23]. Second, a Tail-Sitter is an aircraft that takes off and lands vertically on its tail and the whole aircraft tilts forward using differential thrust or control surfaces to achieve horizontal flight. This category, as it is considered as a complex challenge from the point of view of control systems engineering, has become an interesting research concept as shown by vehicles like Quadshot [30] or prototype [28].

This paper presents recent work concerning the first stage in the development of a hybrid UAV that can be categorized as Tail-Sitter with the exceptions that in this case the aircraft takes off and lands vertically on its nose (using an external ground-station) and that this platform changes the sense of the rotors in order to perform the transition phase between hovering and cruising. Figure 6.1 shows the maneuverability scheme of the proposed unmanned aerial vehicle. In addition, the prototype built based on this philosophy has been *nicknamed* V-Skye.

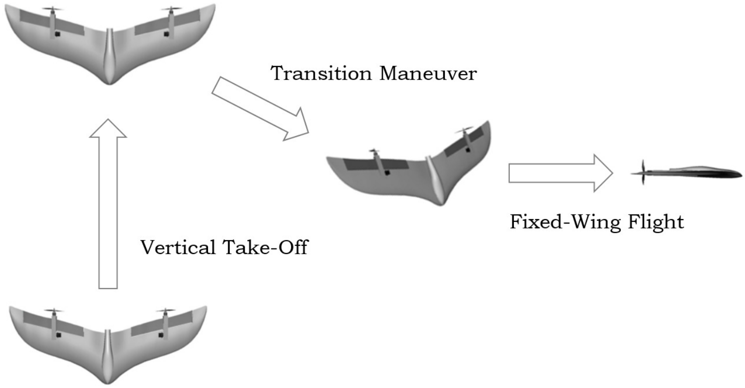


Figure 6.1. Scheme of the transition maneuver between flight modes

In this article, the design of the control system is not only based on simulations, but also on an experimental procedure in which the controllers have to adequately stabilize the UAV allowing it to hover filtering external disturbances. In order to control the attitude, the vehicle is provided with two tilting rotors that allow alterations of its pitch angle and yaw rate and also, modifications in the motor throttles in order to handle roll and vertical speed variables. This is the first step of the development of the entire autonomous system, that will provide this UAV with the hybrid characteristics required by autonomous aviation market, as presented in [33].

Different types of controllers can be designed for UAVs. The simplest ones are linear PID based on linearized models of UAVs. In the literature it is possible to find several approaches which solve the problem of controlling non-linear UAVs: non-linear PID based solutions [18, 10, 22], non-linear robust approaches [39, 24, 38], back-stepping algorithms [14, 26, 15], sliding mode control [39, 15], $H\infty$ control [26] or non-linear observer based [17, 3].

As commented, the objective of this article is to adequately stabilize the designed hybrid UAV in an experimental procedure. This is performed by using 4 linear PIDs tuned by a genetic algorithm. The genetic algorithm searches for the PID parameters that minimize a performance index such as the integral squared error or the settling time.

Using 4 linear PIDs can be considered as a first approach to the design of the control system, and also the easiest way to implement a control system from an experimental point of view. In further researches authors will try to apply more complex techniques such as non-linear PID [18, 10, 10] and non-linear robust approaches [39, 24, 38].

The rest of this article is structured as follows. Section 6.2 covers the description of the airframe that has been used during this project, while Section 6.3 is focused on the explanation of the mathematical model that describes this aerial vehicle. In Section 6.4, the design of the attitude control system is presented. Section 6.5 gathers information related to the HIL simulation platform and finally, Section 6.6 presents the simulation and real-test results that ensure the controller's performance.

6.2 Airframe Description

A rigid body moving inside a three-dimensional (3D) space has a total of six-degrees-of-freedom (6DoF). In this way, a mechanical system formed by a single rigid body needs at least six independently manipulated interactions with the system (inputs) to drive it to an arbitrary orientation and position.

In a hovering maneuver, a flying vehicle is maintained motionless over a reference point at a constant altitude (constant reference position) and on a constant heading angle ψ. Hence, only four of the six degrees of freedom are forced to a reference value (controlled) when hovering. The other two, i.e. pitch θ and roll ϕ angles are dependent variables that evolve along time according to the system equations of motion.

The V-Skye is designed with two tilting-rotors moved by servo-mechanisms. The result is a vehicle with two motors for which thrust $\vec{T}_R$ and $\vec{T}_L$ can be independently modified, not only in magnitude, but also in one direction. The system is thus provided with the amount of independent inputs needed for the hovering maneuver.

Figure 6.2 shows an outline drawing of the V-Skye. In order to simplify the dynamics, all actuation parts (motors, motor frames, servomotors and their transmission parts) are allocated as symmetrically as possible about the fixed coordinate axis $\{\hat{X}_b, \hat{Y}_b, \hat{Z}_b\}$ of the aircraft reference frame. In particular, all elements are placed on the $\hat{Y}_b\hat{Z}_b$ plane and symmetrical to the $\hat{X}_b\hat{Z}_b$ plane.

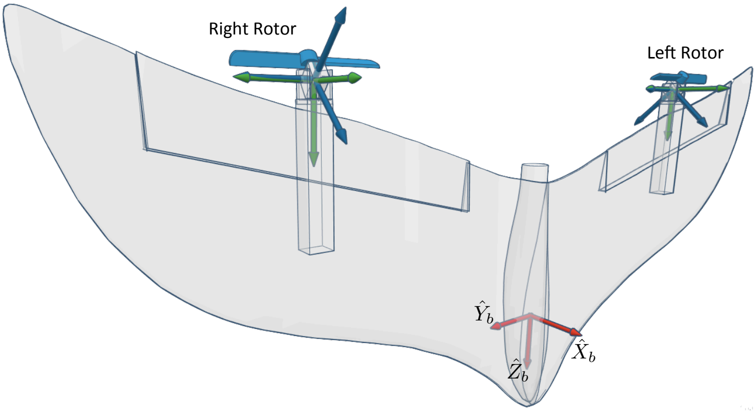


Figure 6.2. Local axis in the 3D graphical model of the V-Skye UAV

For simplicity on the explanations, authors have divided the aircraft into three well-differentiated frames.

6.2.1 Main body frame

As depicted in Fig. 6.2, the main body has the constructive shape of a Flying-Wing aircraft, such as the ones used in [37, 25]. It is a rigid body housing all the electronics as well as the two servomotors that allow rotation of the motor frames (see section 6.2.2 and 6.2.3).

The reference system $\{\hat{X}_b, \hat{Y}_b, \hat{Z}_b\}$ has its origin at the aircraft centre of gravity and it is fixed to the main body frame. For its vertical flight phase, the $\hat{X}_b$ direction points front, towards what would naturally be the upper part of the fuselage. The $\hat{Z}_b$ direction points down, towards the nose of the flying-wing. Finally, $\hat{Y}_b$ axis is perpendicular to the other two and points towards the right-side wing.

The earth coordinate axis $\{\hat{X}_e, \hat{Y}_e, \hat{Z}_e\}$ is a North-East-Down (NED) inertial frame of reference, also positioned at the aircraft centre of gravity but fixed to the earth surface. Euler angles roll ϕ, pitch θ and yaw ψ define the main body orientation respect to the earth axis. Figure 6.5 shows those three independent rotations.

6.2.2 Right motor frame

It is formed by the right motor, its right-handed propeller and a structure specially designed to hold it and stand any reaction force derived from the flight. It can be seen as a second rigid body attached to the aircraft by a rotatory joint, as Fig. 6.3 describes in detail.

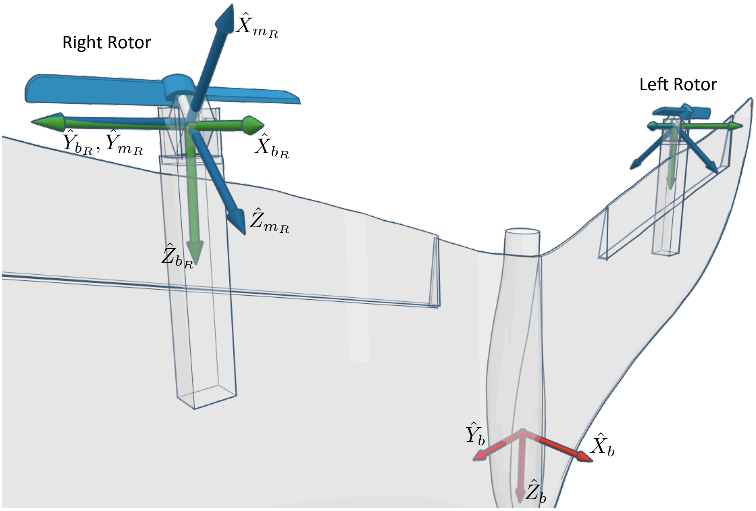


Figure 6.3. Right rotor coordinate reference system.

Two coordinate systems are defined to describe the motion of this frame with respect to the main body. $\{\hat{X}_{b_R}, \hat{Y}_{b_R}, \hat{Z}_{b_R}\}$ is fixed to the the main body frame and parallel to $\{\hat{X}_b, \hat{Y}_b, \hat{Z}_b\}$; its centre O_{b_R} is placed where the rotatory joint intersects the motor shaft axis. On the other hand, $\{\hat{X}_{m_R}, \hat{Y}_{m_R}, \hat{Z}_{m_R}\}$ have its origin O_{m_R} at O_{b_R}; the $\hat{Y}_{m_R}$ axis coincides with $\hat{Y}_{b_R}$ $(\hat{Y}_{b_R} \parallel \hat{Y}_{m_R})$ and $\hat{Z}_b$ axis coincides with the motor shaft axis.

Because the right motor frame is attached to the main body by a rotatory joint, it has one single DOF: the angle λ_R rotated about the axis $\hat{Y}_{b_R} \parallel \hat{Y}_{m_R}$. When $\lambda_R = 0$, both $\{\hat{X}_{m_R}, \hat{Y}_{m_R}, \hat{Z}_{m_R}\}$ and $\{\hat{X}_{b_R}, \hat{Y}_{b_R}, \hat{Z}_{b_R}\}$ have the exact same position and orientation. The direction of the right motor's thrust is changed by actuating on the λ_R value, since the thrust has always the $\hat{Z}_{m_R}$ direction. For this reason, a servomotor is used to manipulate λ_R.

6.2.3 Left motor frame

The left motor frame includes namely the left motor with a left-handed propeller and the structure to hold it. The right and left propellers are designed to be right and left handed respectively. This makes the motors to rotate in opposite senses, helping to compensate motors torques. Figure 6.4 illustrates the configuration and coordinate reference systems.

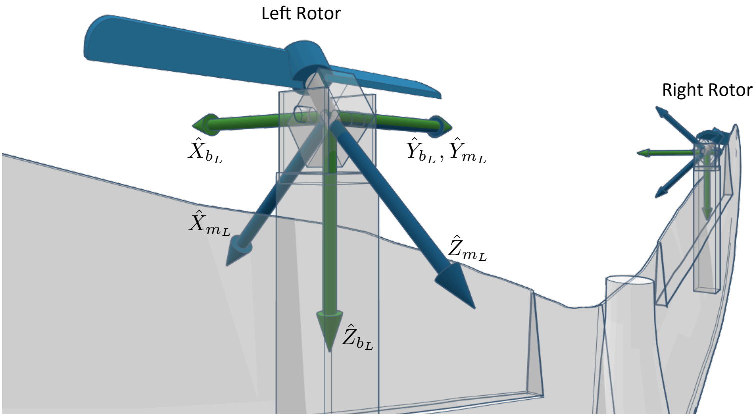


Figure 6.4. Left rotor coordinate reference system.

Similarly to the right motor frame, two coordinate systems are defined for the left motor frame: $\{\hat{X}_{b_L}, \hat{Y}_{b_L}, \hat{Z}_{b_L}\}$ and $\{\hat{X}_{m_L}, \hat{Y}_{m_L}, \hat{Z}_{m_L}\}$ with λ_L being the angle rotated by $\{\hat{X}_{m_L}, \hat{Y}_{m_L}, \hat{Z}_{m_L}\}$ with respect to $\{\hat{X}_{b_L}, \hat{Y}_{b_L}, \hat{Z}_{b_L}\}$ about the $\hat{Y}_{b_L} \parallel \hat{Y}_{m_L}$ axis.

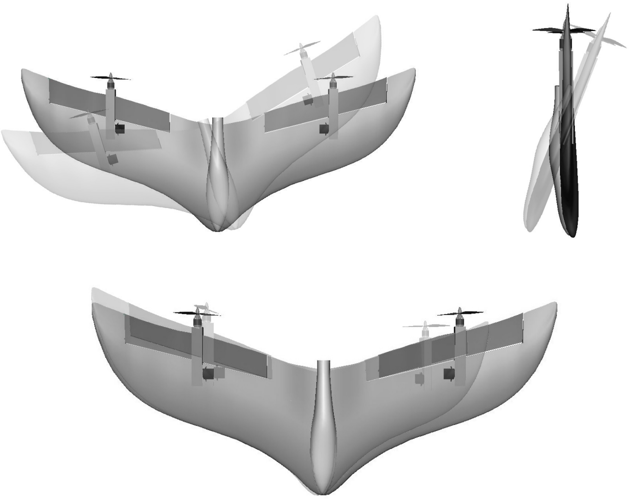


Figure 6.5. Roll, pitch and yaw motions

6.3 Mathematical Model

The equations that conform the 6-DOF non-linear dynamical model are derived in this paper assuming the following hypothesis:

1. The whole aircraft is assumed to be rigid body; it means that the distance between any two points in the airframe remains constant. This is a fundamental condition because it allows to understand the movement of the vehicle as a translation and a rotation around the center of gravity independently.

2. Derived from the previous item, the changes in λ_R and λ_L angles do not affect the mass distribution along the aircraft body.

3. The rotational movement of the Earth is negligible with respect to the accelerations on the vehicle. i.e. the Earth frame is an inertial frame of reference.

4. The atmosphere is assumed to be calm (no wind or turbulence)

5. The plane $\{Y_b = 0\}$ is a plane of symmetry. Hence, the inertia products about the Y_b axis $I_{y_b x_b} = I_{y_b z_b} = 0$

6.3.1 Translational equations

Let $\vec{F}$ be the resultant force vector of all the external forces acting on the system, m the total mass of the aircraft and $\vec{V}$ the aircraft linear velocity with respect to the earth frame. Newton's second law can be written as:

$$\begin{aligned}\vec{F} &= m \cdot \dot{\vec{V}} && (6.1)\\ \{F_{x_b}, F_{y_b}, F_{z_b}\} &= m \cdot \frac{{}^{E}d}{dt}\left(\{u, v, w\}\right) && (6.2)\end{aligned}$$

where F_{x_b}, F_{y_b}, F_{z_b}, u, v and w are the three components of the resultant force and the system velocity respectively, both magnitudes expressed in body axis. Velocity's time derivative with respect to the earth frame might be now rewritten as the summation of its time derivative with respect to the body frame and the cross product of angular and linear velocities as follows

$$\begin{aligned}\dot{\vec{V}} &= \frac{{}^{B}d}{dt}\left(\{u, v, w\}\right) +^{E} \vec{\omega}^{B} \times \vec{V} && (6.3)\\ \dot{\vec{V}} &= \{\dot{u}, \dot{v}, \dot{w}\} + \{p, q, r\} \times \{u, v, w\} && (6.4)\\ \dot{\vec{V}} &= \{\dot{u} + q \cdot w - r \cdot v, \dot{v} + r \cdot u - p \cdot w, \dot{w} + p \cdot v - q \cdot u\} && (6.5)\end{aligned}$$

where ${}^{E}\vec{\omega}^{B}$ is the angular velocity of the body frame with respect to the earth frame, and p, q and r its components expressed in body axis. Therefore, vector equation (6.1) is separated into three independent equations as shown next

$$\begin{aligned}m \cdot (\dot{u} + q \cdot w - r \cdot v) &= F_{x_b} && (6.6)\\ m \cdot (\dot{v} + r \cdot u - p \cdot w) &= F_{y_b} && (6.7)\\ m \cdot (\dot{w} + p \cdot v - q \cdot u) &= F_{z_b} && (6.8)\end{aligned}$$

The external forces considered in this work are the rotors thrust and the aircraft weight. Since the work is focused on the design of a control scheme for a hovering maneuver, the dynamics model do not consider aerodynamic effects on the aircraft body. In a VTOL procedure, the lift is totally generated through the thrust produced by the rotors. On the other hand, drag produced by the flying-wing airframe is taken as an external disturbance for the attitude tracking controller.

Denoting by T_R and T_L the thrust magnitudes of right and left rotors respectively, the definition of the thrust forces in body axis is characterised by the angles λ_R and λ_L of the right and left motor frames with respect to the body frame.

$$
\begin{aligned}
\vec{T_R} &= -T_R \cdot \{\sin\lambda_R, 0, \cos\lambda_R\} \qquad (6.9)\\
\vec{T_L} &= -T_L \cdot \{\sin\lambda_L, 0, \cos\lambda_L\} \qquad (6.10)
\end{aligned}
$$

In [4] a complete study of the performance of several types of propeller at different airflow conditions is presented. Along the paper, John B. Brandt and Michael S. Selig., explain how to model thrust and torque at low Reynolds numbers, gathering data of these values to calculate aerodynamic coefficients for a large number of commercial propellers. Now taking the equations presented in [4] as a reference, and letting T and τ denote thrust and torque magnitudes of a propeller and C_T and C_τ its force and torque coefficients, then

$$
\begin{aligned}
T &= C_T \rho n^2 D^4 \qquad (6.11)\\
\tau &= C_\tau \rho n^2 D^5 \qquad (6.12)
\end{aligned}
$$

Hence, the relation between torque and thrust on the propeller can be written as

$$T = \frac{C_T}{D \cdot C_\tau}\tau \qquad (6.13)$$

If we denote by δ_R and δ_L the motors throttle, and given that the aircraft will be mounting two brushless DC motors managed by electronic speed controllers (ESC), the two motor torques are related to their throttle by the expressions (6.14) and (6.15).

$$
\begin{aligned}
\tau_R &= k_\tau \delta_R \qquad (6.14)\\
\tau_L &= k_\tau \delta_L \qquad (6.15)
\end{aligned}
$$

and hence

$$
\begin{aligned}
T_R &= \frac{C_T}{D \cdot C_Q} k_\tau \delta_R = k_T \delta_R \qquad (6.16)\\
T_L &= \frac{C_T}{D \cdot C_Q} k_\tau \delta_L = k_T \delta_L \qquad (6.17)
\end{aligned}
$$

k_τ is the torque's throttle coefficient and is assumed to be constant for a given motor and ESC combination whereas k_T is the thrust's throttle coefficient that depends on the airspeed and is constant for the static case. This leads to the following definition of thrust force:

$$\begin{aligned}
F_{Tx} &= -k_T\delta_R \cdot \sin\lambda_R - k_T\delta_L \cdot \sin\lambda_L && (6.18)\\
F_{Ty} &= 0 && (6.19)\\
F_{Tz} &= -k_T\delta_R \cdot \cos\lambda_R - k_T\delta_L \cdot \cos\lambda_L && (6.20)
\end{aligned}$$

The aforementioned force variables $\{F_{x_b}, F_{y_b}, F_{z_b}\}$ are now substituted by their corresponding terms of thrust and weight forces in the body axis.

$$\begin{aligned}
m\left(\dot{u} + q \cdot w - r \cdot v\right) &= -k_T\delta_R \cdot \sin\lambda_R - k_T\delta_L \cdot \sin\lambda_L \\
&\quad -m \cdot g \cdot \sin\theta && (6.21)\\
m\left(\dot{v} + r \cdot u - p \cdot w\right) &= m \cdot g \cdot \cos\theta \sin\phi && (6.22)\\
m\left(\dot{w} + p \cdot v - q \cdot u\right) &= -k_T\delta_R \cdot \cos\lambda_R - k_T\delta_L \cdot \cos\lambda_L \\
&\quad +m \cdot g \cdot \cos\theta\cos\phi && (6.23)
\end{aligned}$$

6.3.2 Rotational equations

By definition of angular momentum about the gravity centre $\vec{H}^{c.g.}$ and considering a rigid-body configuration

$$\vec{H}^{c.g.} = \begin{bmatrix} I_{x_bx_b} & -I_{x_by_b} & -I_{x_bz_b} \\ -I_{y_bx_b} & I_{y_by_b} & -I_{y_bz_b} \\ -I_{z_bx_b} & -I_{z_by_b} & I_{z_bz_b} \end{bmatrix}^{c.g.} \cdot^{E} \vec{\omega}^{B} \qquad (6.24)$$

where $I_{ii} \forall i \in \{x_b, y_b, z_b\}$ are the moments of inertia of the aircraft body about its gravity centre in body axis and $I_{ij} \forall ij \in \{x_b, y_b, z_b\}$with $j \neq i$ are the products of inertia.

Now, the total moment of forces about the aircraft's gravity centre is equal to its angular momentum's time derivative with respect to the earth frame. Additionally, we can again split the time derivative of the angular momentum with respect to the earth frame into its time derivative with respect to the body frame and the cross product of the angular velocity and the angular momentum.

$$\begin{aligned}
\vec{Q}^{c.g.} &= \dot{\vec{H}}^{c.g.} && (6.25)\\
\vec{Q}^{c.g.} &= \frac{^{B}d}{dt}\left(\vec{H}^{c.g.}\right) +^{E} \vec{\omega}^{B} \times \vec{H}^{c.g.} && (6.26)
\end{aligned}$$

Equation (6.26) gives the following three independent equations

$$L = I_{xx}\dot{p} - I_{xz}\dot{r} - I_{xz}pq + \left(I_{zz} - I_{yy}\right) qr \tag{6.27}$$

$$M = I_{yy}\dot{q} - I_{xz}\left(p^2 - r^2\right) + \left(I_{xx} - I_{zz}\right) pr \tag{6.28}$$

$$N = I_{zz}\dot{r} - I_{xz}\dot{p} + I_{xz}rq + \left(I_{yy} - I_{xx}\right) pq \tag{6.29}$$

being L, M and N the three body axis components of the total external moments applied on the aircraft centre of gravity.

The right rotor thrust applied at right motor frame generates an external moment about the aircraft centre of gravity. Denoting $\vec{r}_{m_R} = \{x_{m_R}, y_{m_R}, z_{m_R}\}$ as the position vector of the right motor frame, the total moment of the right rotor thrust about the centre of gravity is given by equations (6.30) and (6.31). Note that $x_{m_R} = 0$ by design.

$$\vec{Q}_{m_R} = \vec{r}_{m_R} \times \vec{T_R} \tag{6.30}$$

$$\vec{Q}_{m_R} = k_T \delta_R \cdot \{-y_{m_R} \cos\lambda_R, -z_{m_R} \sin\lambda_R, y_{m_R} \sin\lambda_R\} \tag{6.31}$$

Additionally, the right motor torque applied to the propeller is translated to the right motor frame as a reaction torque (same direction but opposite sense). That torque is then translated to the body frame according to λ_R angle. From eq. (6.14)

$$\vec{\tau}_R = \tau_R \cdot \{\sin\lambda_R, 0, \cos\lambda_R\} \tag{6.32}$$

$$\vec{\tau}_R = k_\tau \delta_R \cdot \{\sin\lambda_R, 0, \cos\lambda_R\} \tag{6.33}$$

In accordance, thrust moment and motor torque for the left rotor are given by expressions (6.34) and (6.35)

$$\vec{Q}_{m_L} = k_T \delta_L \cdot \{-y_{m_L} \cos\lambda_L, -z_{m_L} \sin\lambda_L, y_{m_L} \sin\lambda_L\} \tag{6.34}$$

$$\vec{\tau}_L = -k_\tau \delta_L \cdot \{\sin\lambda_L, 0, \cos\lambda_L\} \tag{6.35}$$

Now, substituting the total external moments and torques into equations (6.27) - (6.29)

$$\begin{aligned}
I_{xx}\dot{p} - I_{xz}\dot{r} & \\
-I_{xz}pq + \left(I_{zz} - I_{yy}\right) qr &= \delta_R \left(k_\tau \sin\lambda_R - k_T y_{m_R} \cos\lambda_R\right) \\
&\quad -\delta_L \left(k_\tau \sin\lambda_L + k_T y_{m_L} \cos\lambda_L\right) \qquad (6.36) \\
I_{yy}\dot{q} - I_{xz}\left(p^2 - r^2\right) & \\
+\left(I_{xx} - I_{zz}\right) pr &= -\delta_L k_T z_{m_L} \sin\lambda_L - \delta_R k_T z_{m_R} \sin\lambda_R \qquad (6.37) \\
I_{zz}\dot{r} - I_{xz}\dot{p} & \\
+I_{xz}rq + \left(I_{yy} - I_{xx}\right) pq &= \delta_R \left(k_\tau \cos\lambda_R + k_T \sin\lambda_R y_{m_R}\right) \\
&\quad +\delta_L \left(k_T y_{m_L} \sin\lambda_L - k_\tau \cos\lambda_L\right) \qquad (6.38)
\end{aligned}$$

6.3.3 Collection of non-linear equations

In addition to the six dynamics equations derived above, six kinematic equations can be stated to express the transformation from the body to the earth system of reference. The aircraft behaviour is therefore described by a total of twelve equations of motion.

As a summary, the aircraft model is divided in the following sets.

Translational dynamics equations

$$\begin{aligned}
\dot{u} &= rv - qw - g\sin\theta - \frac{k_T}{m}\left(\delta_R \sin\lambda_R + \delta_L \sin\lambda_L\right) \qquad (6.39) \\
\dot{v} &= pw - ru + g\cos\theta\sin\phi \qquad (6.40) \\
\dot{w} &= qu - pv + g\cos\theta\cos\phi - \frac{k_T}{m}\left(\delta_R \cos\lambda_R + \delta_L \cos\lambda_L\right) \qquad (6.41)
\end{aligned}$$

Rotational dynamics equations

$$\begin{aligned}
\dot{p} - \frac{I_{xz}}{I_{xx}}\dot{r} &= +\frac{I_{xz}}{I_{xx}}pq + \frac{I_{yy} - I_{zz}}{I_{xx}}qr \\
&\quad +\frac{\delta_R}{I_{xx}}\left(k_\tau \sin\lambda_R - k_T y_{m_R}\cos\lambda_R\right) \\
&\quad -\frac{\delta_L}{I_{xx}}\left(k_\tau \sin\lambda_L + k_T y_{m_L}\cos\lambda_L\right) \qquad (6.42) \\
\dot{q} &= \frac{I_{xz}}{I_{yy}}\left(p^2 - r^2\right) + \frac{I_{zz} - I_{xx}}{I_{yy}}pr \\
&\quad -\frac{1}{I_{yy}}\delta_L k_T z_{m_L}\sin\lambda_L \\
&\quad -\frac{1}{I_{yy}}\delta_R k_T z_{m_R}\sin\lambda_R \qquad (6.43) \\
\dot{r} - \frac{I_{xz}}{I_{zz}}\dot{p} &= -\frac{I_{xz}}{I_{zz}}rq + \frac{I_{xx} - I_{yy}}{I_{zz}}pq \\
&\quad +\frac{\delta_R}{I_{zz}}\left(k_\tau \cos\lambda_R + k_T \sin\lambda_R y_{m_R}\right) \\
&\quad +\frac{\delta_L}{I_{zz}}\left(k_T y_{m_L}\sin\lambda_L - k_\tau \cos\lambda_L\right) \qquad (6.44)
\end{aligned}$$

Kinematic translational equations

$$\begin{aligned}
u &= \dot{x}_e\cos\theta\cos\psi + \dot{y}_e\cos\theta\sin\psi - \dot{z}_e\sin\theta \qquad (6.45) \\
v &= \dot{x}_e\left(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi\right) \\
&\quad +\dot{y}_e\left(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi\right) + \dot{z}_e\sin\phi\cos\theta \qquad (6.46) \\
w &= \dot{x}_e\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right) \\
&\quad +\dot{y}_e\left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right) + \dot{z}_e\cos\phi\cos\theta \qquad (6.47)
\end{aligned}$$

Kinematic rotational equations (Euler angles)

$$p = \dot{\phi} - \dot{\psi} \cdot \sin\theta \tag{6.48}$$

$$q = \dot{\theta} \cdot \cos\phi + \dot{\psi} \cdot \cos\theta \cdot \sin\phi \tag{6.49}$$

$$r = \dot{\psi} \cdot \cos\theta \cdot \cos\phi - \dot{\theta} \cdot \sin\phi \tag{6.50}$$

6.4 Control System

The control problem associated with the presented UAV stabilization is challenging for several reasons. The complexity of flight dynamics resides on the system nonlinearity, unstable nature and high degree of coupling. Furthermore, in this case, the system is under-actuated because only four control inputs can be used during take off, landing and hover maneuvers, while the whole system is six-degree-of-freedom (6DoF). Therefore, robust and reliable feedback control strategies are needed to regulate the attitude of the UAV within an operational range.

Due to their simplicity, ease of implementation and robust performance, a decentralised and linear control scheme based on four proportional-integral-derivative controllers (PID) has been chosen to design the attitude tracking controller. Figure 6.6 shows the block diagram of the proposed feedback control scheme. It is composed by a controller for each angle of orientation and one for the vertical velocity.

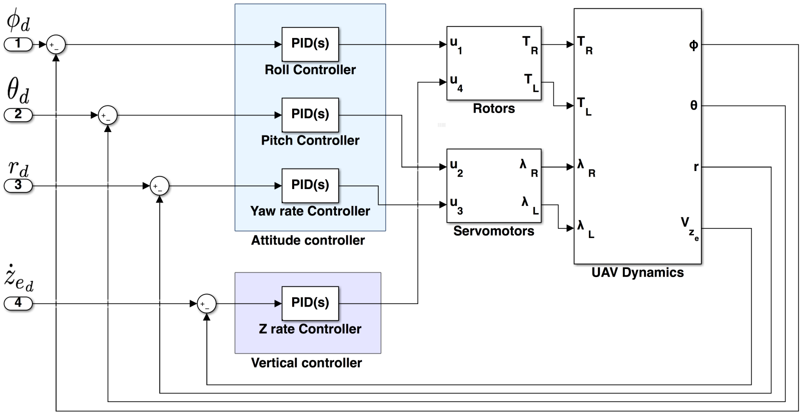


Figure 6.6. Attitude and vertical speed controller scheme of the UAV V-Skye

Attending to the nonlinear equations of motion presented in Section 6.3, a modification on one of the four system inputs δ_R, δ_L, λ_R or λ_L, excites more than one state variable at the same time. This means that the system dynamics are highly coupled. In an attempt to reduce the effect of coupled actuators, a set of four inputs u_1, u_2, u_3 and u_4 are defined as a combination of the real input variables. Equations (6.51) to (6.54) show the new system input variables and their relationship with motors throttle and thrust angles

$$\begin{aligned}
u_1 &= \frac{1}{2}\left(\delta_L - \delta_R\right) && (6.51)\\
u_2 &= -(\lambda_R + \lambda_L) && (6.52)\\
u_3 &= \lambda_R - \lambda_L && (6.53)\\
u_4 &= \frac{1}{2}\left(\delta_R + \delta_L\right) && (6.54)\\
& && (6.55)
\end{aligned}$$

With this new definition of the system inputs, a change in u_1 has high effect on the $\dot{p}$ variable, u_2 on $\dot{q}$, u_3 on $\dot{r}$ and u_4 on $\dot{w}$. Now the different PID control schemes are defined accordingly.

6.4.1 Roll and pitch controllers

Controllers for roll and pitch angles are implemented similarly: both use standard PID controllers with feedback of estimated angles (roll or pitch) from a complementary filter, resulting in the following control law:

$$\begin{aligned}
u_1 &= K_{p,1}((\phi_d - \phi) + \frac{1}{T_{i,1}}\int(\phi_d - \phi)dt + T_{d,1}(\dot{\phi_d} - \dot{\phi})) && (6.56)\\
u_2 &= K_{p,2}((\theta_d - \theta) + \frac{1}{T_{i,2}}\int(\theta_d - \theta)dt + T_{d,2}(\dot{\theta_d} - \dot{\theta})) && (6.57)
\end{aligned}$$

Where $K_{p,i}$, $K_{i,i}$ and $K_{d,i}$ are the proportional, integral and derivative gains, respectively. ϕ_d and θ_d is associated with the reference angles or desired roll and pitch, while u_i is the controller action; on one hand, u_1 corresponds to the differential thrust between two rotors, while u_2 is associated with the tilt angle of both rotors.

6.4.2 Angular velocity r controller

Despite of the fact that the heading of the UAV is generally a less critical degree of freedom, external disturbances can generate undesired rotational movements around the Z axis during flying. In order to overcome this drift, a yaw rate PI controller has been implemented using the next expression:

$$u_3 = K_{p,3}((r_d - r) + \frac{1}{T_{i,3}} \int (r_d - r) dt) \tag{6.58}$$

In this case, the reference (r_d) is compared with the feedback yaw angular velocity (r) measured by a yaw rate gyroscope. The resulting difference is sent to the controller in order to generate opposite tilt angles by using a differential servo deflection (u_3).

6.4.3 Vertical velocity controller

Due to the high coupling, changes on angles λ_R and λ_L, from the roll and r controllers, induce variations on the vertical thrust, which leads to variations on vertical velocity. Therefore, a vertical velocity PID controller is implemented as shown in expression (6.59).

$$u_4 = K_{p,4}((\dot{z}_{e_d} - \dot{z}_e) + \frac{1}{T_{i,4}} \int (\dot{z}_{e_d} - \dot{z}_e) dt + K_{d,4}(\ddot{z}_{e_d} - \ddot{z}_e)) \tag{6.59}$$

Where, $\dot{z}_{e_d}$ is the reference vertical velocity, $\dot{z}_e$ is the measured vertical velocity estimated from measures of the inertial magnetic unit (IMU) sensors. u_4 corresponds to the increase in the same amount of thrust in both rotors.

6.4.4 Local Stability

The global close loop stability has been considered from two aspects. Firstly, the control structures presented in this Section have been adjusted to ensure that the poles of the closed loop system have all of them negative real terms [21]. For this purpose, the particular transfer functions presented in Section 6.5 (equation (6.60)) have been used. Obviously, this only guarantees stability close to the equilibrium point defined for the linearization and, therefore, it is a local stability constraint.

Secondly, in order to study the margin of the local stability, HIL simulations have been performed using the designed controllers against the full non-linear model of the UAV. In this way, the validity range of the controllers designed from the linear transfer function is tested by realistic simulations [2, 5]. Obviously, this is not an overall guarantee of stability, but it is possible to define a range of operation with a high degree of confidence for the control design.

6.5 Test Prototype

The initial test platform presented in the work is a Flying-wing from Multiplex, model XENO UNI [19]. This RC plane has been modified to add two tilt rotors, following the main concept of VTOL UAV described previously. Figures 6.7 and 6.8 show the prototype assembled for real flight tests.

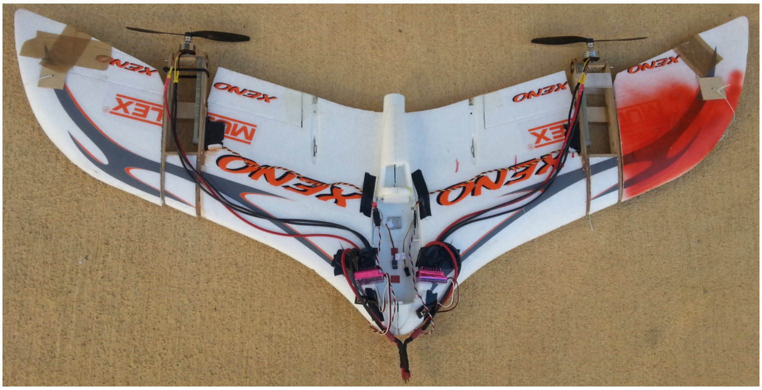


Figure 6.7. Xeno UNI form Multiplex with two customised tilt rotors.

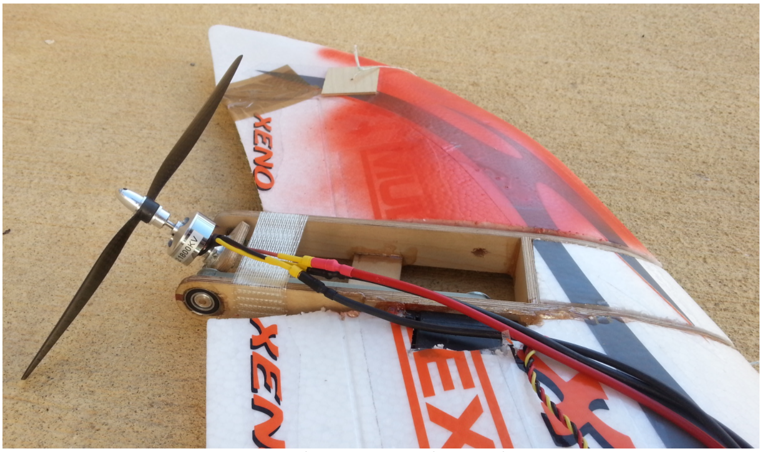


Figure 6.8. Tilt rotor mechanical structure.

The set of aircraft parameters used for control design and simulation have been derived form the test platform described in the previous paragraph, and their particular values are as follows:

- Wingspan: 1.26 m
- Fuselage length: 0.526 m
- Empty weight: 0.1904 kg
- Operating weight: 0.7484 kg
- k_T=15.7 and k_τ=0.34
- Brushless motors: T-MOTOR Antigravity MT2814 770 KV
- Propellers: T-MOTOR 12x4 CF
- Electric Speed Controllers: HW-09-V2-OEM
- Tilt Rotor Servos: HITEC HS-5475HB
- Flight Controller: CC3D OpenPilot Revolution
- Battery: 2200mAh 4S 80/160C

Apart from these geometrical characteristics, the engines provide a maximum thrust of $15,7$ N and the tilt-mechanism admit a deflection up to 0.5235 rad. The combination of engines, propellers, Electronic Speed Controllers (ESC) and servos employed in this airframe provide enough power and maneuverability to successfully accomplish the hovering mission even with a larger payload, such as an action camera.

In order to complete the tilt-rotor model parameters, the corresponding moments of inertia are: $I_{xx} = 0.0015\ kg\ m^2$, $I_{yy} = 0.0160\ kg\ m^2$ y $I_{zz} = 0.0176\ kg\ m^2$, and also a product of inertia of $I_{xz} = -1.4182 \cdot 10^{-5}\ m^4$.

6.5.1 Prototype model linearization

In order to design the linear control approach described in section 6.4, first of all, the attitude dynamic model described by expressions of section 6.3 is decoupled into four linearized single-input single-output (SISO) models around the operational range of a hovering maneuver, which implies that $p \simeq q \simeq r \simeq 0$ and $u \simeq v \simeq w \simeq 0$.

The four linear models are obtained by replacing the nonlinear equations of motion with their Taylor series approximation truncated to the first order with respect to the controlled variables and inputs. This linearization approach is well know and, frequently, it is defined as *Small Perturbation Theory* [31, 35, 34, 29]. Once the set of equations have a linear structure and after replacing the model parameters by their prototype value, the following transfer functions are obtained:

$$G_\phi(s) = \tfrac{\phi(s)}{u_1(s)} = \frac{554.78}{s^2(s+19.05)} \tag{6.60}$$

$$G_\theta(s) = \tfrac{\theta(s)}{u_2(s)} = \frac{-13.95(s-138.5)(s+138.5)}{s^2(s+153.2)(s+21.75)} \tag{6.61}$$

$$G_r(s) = \tfrac{r(s)}{u_3(s)} = \frac{1.1674(s-56160)}{s(s+153.2)(s+21.75)} \tag{6.62}$$

$$G_{\dot{z}_e}(s) = \tfrac{\dot{z}_e(s)}{u_4(s)} = \frac{-117.09}{s(s+19.05)} \tag{6.63}$$

6.5.2 Tuning PID loops for prototype control

After calculating the transfer functions, the controllers parameters can be obtained either using classical techniques, such as the Root-Locus method, or more modern techniques based on optimisation with genetic algorithms.

The use of Root-Locus method for PID tuning proves to be quite easy and useful compared to others techniques, since it indicates the manner in which the open-loop poles and zeros should be modified so that the response meets system performance specifications [21]. However, a disadvantage of using this technique is that it is necessary to employ a linear model, which is only an approximation of the complex dynamics of the UAV.

On the other hand, PID tuning and optimisation using genetic algorithms is a modern technique that provides an adaptive searching mechanism inspired on Darwin's principle of reproduction and survival of the fittest. The individuals (solutions) in a population are represented by chromosomes that are associated to a fitness value (problem evaluation). The chromosomes are subjected to an evolutionary process which takes several cycles. Basic operations are selection, reproduction, crossover and mutation [11]. One of the main advantage of using genetic algorithms is that it is a global search technique of optimal and sub-optimal solutions of a problem and hence it can directly interact with the non-linear dynamics model.

Parameters of PID controllers obtained using both techniques are presented in Tables 6.1 and 6.2.

PID Controller	K_p	T_i	T_d
ϕ	0.25886	∞	0.952
θ	0.11936	∞	0.95
r	0.415	0.83	0
$\dot{z}_e$	0.85	∞	1.076

Table 6.1. Pid parameters using the Root-Locus method

PID Controller	K_p	T_i	T_d
ϕ	0.2979	4.2997	0.2945
θ	0.2080	9.8801	0.368
r	0.2	4	0
$\dot{z}_e$	0.55	6.5	0.5

Table 6.2. Pid parameters using a genetic algorithm

In order to compare the performance of the attitude and vertical speed control system tuned using both classical and modern strategies, a numerical simulation with the nonlinear model of the UAV is made using both set of parameters and the accuracy requirements for the system are formulated in terms of the settling time (t_s) and the integral squared error index (ISE), which is related to the time response of the system.

$$ISE = \int_0^t e^2 dt \tag{6.64}$$

The presented simulations consisted in transition with predefined dynamics from one steady state flight to another. Numerical results evaluating the proposed indexes are shown in Tables 6.3 and 6.4.

PID Controller	t_s	ISE
ϕ	4.603	96075
θ	3.618	128310
r	3.986	95095
$\dot{z}_e$	3.986	4922

Table 6.3. Controller performance using the Root-Locus method

PID Controller	t_s	ISE	$Enhancement_{t_s}(\%)$	$Enhancement_{ISE}(\%)$
ϕ	2.931	95385	36.32	0.72
θ	1.890	127080	47.76	0.96
r	2.835	95140	28.88	-0.05
$\dot{z}_e$	2.168	4419	45.61	10.22

Table 6.4. Controller performance using a genetic algorithm

Taking into consideration the proposed control effort indexes, PID tuning obtained by a genetic algorithm is the most comprehensive choice. For this reason, this set of parameters will be adopted during the simulations and flight tests (Section 6.6).

6.5.3 HIL Simulation Platform

The HIL simulation platform is shown in Fig. 6.9. The main computation unit is a PXI laboratory computer from National Instruments. This equipment includes several boards to interface a great amount of external devices. As an example digital and analogue input/output boards, Ethernet and Serial ports or four USB ports. This hardware comes with a real-time operative system that can be configured to run Real-Time simulations [5, 13, 9].

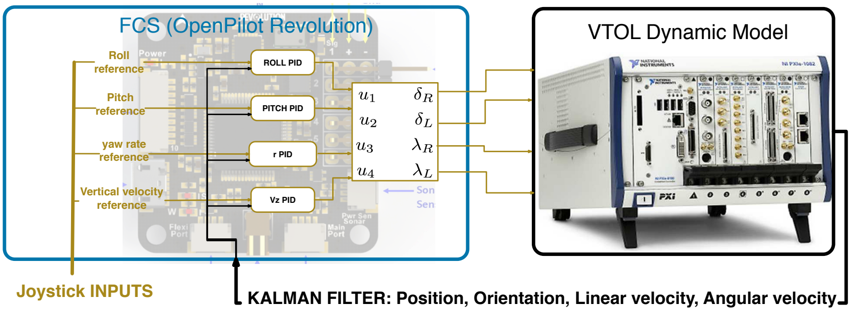


Figure 6.9. Hardware In the Loop platform diagram

The set of 12 equations of motion derived in section 6.3 have been particularised for the prototype model and implemented in Matlab/Simulink. Simulink allows to compile the model to be run on the PXI real-time target [20, 6].

A CC3D OpenPilot Revolution Board [7] has been elected as the flight control unit (FCS). It is a digital board with a micro-controller that comes with the necessary onboard sensors and autopilot software already installed. The control algorithms have been modified and implemented on the FCS. Then the onboard sensors have been bypassed so that the values coming from the model are used to close the loop. Finally, the board has been connected to the PXI to send actuators values and receive state variables values. A joystick has also been connected to the board to send the reference value to the controllers.

6.6 Results

This section covers the analysis of the results obtained in simulation (using HIL simulation platform) and in real flight tests. The main goal is to understand the dynamical behaviour of the system while it is hovering at certain attitude configurations.

6.6.1 Simulation results

The first simulation consists in coupled changes of roll and pitch that allow the reader to understand how the control actions (throttle and tilt-angle) have to be modified in order to follow the reference in attitude and vertical velocity.

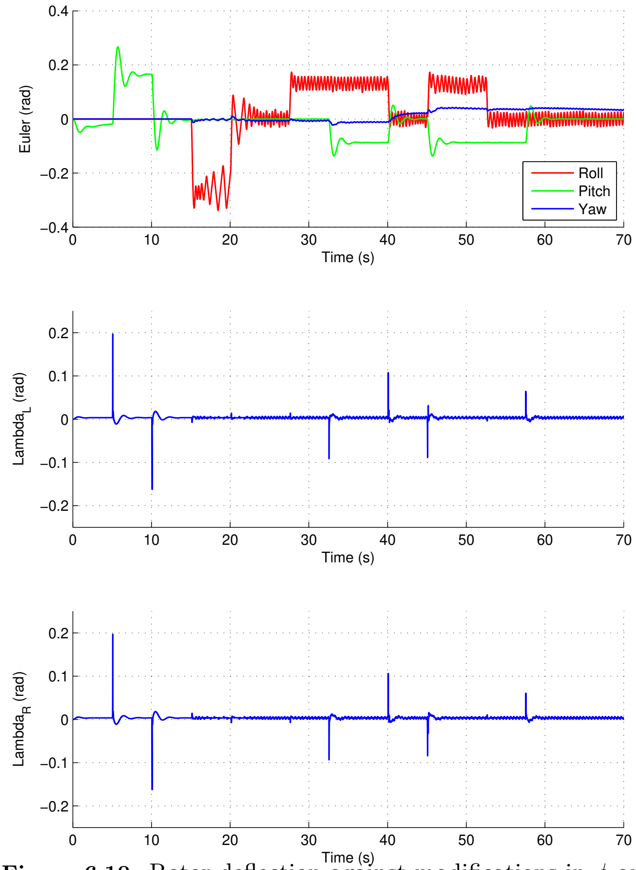


Figure 6.10. Rotor deflection against modifications in ϕ and θ

Two conclusions can be drawn based on the previous Fig. 6.10. First of all, as it was already mentioned, the system deflects the rotors in the same direction in order to achieve certain pitch angles. As it can be observed, until $t = 15$ seconds there have only been modifications in the pitch angle, and therefore, $\lambda_L = \lambda_R$.

However, the interesting aspect of this result is what happens with the tilt-angle when a roll maneuver is being carried out. As mentioned, this rotation needs a different thrust generated by each of the rotors until the air-frame reaches the desired configuration. Therefore, during this procedure the total torque is no longer null, meaning that there exists a tendency to modify the yaw of the system by increasing its angular velocity r. The result of the coupling of these dynamical movements is that the attitude tracking controller produces opposite tilt-angles in order to maintain $r = 0$.

This effect is better explained in Fig. 6.11.

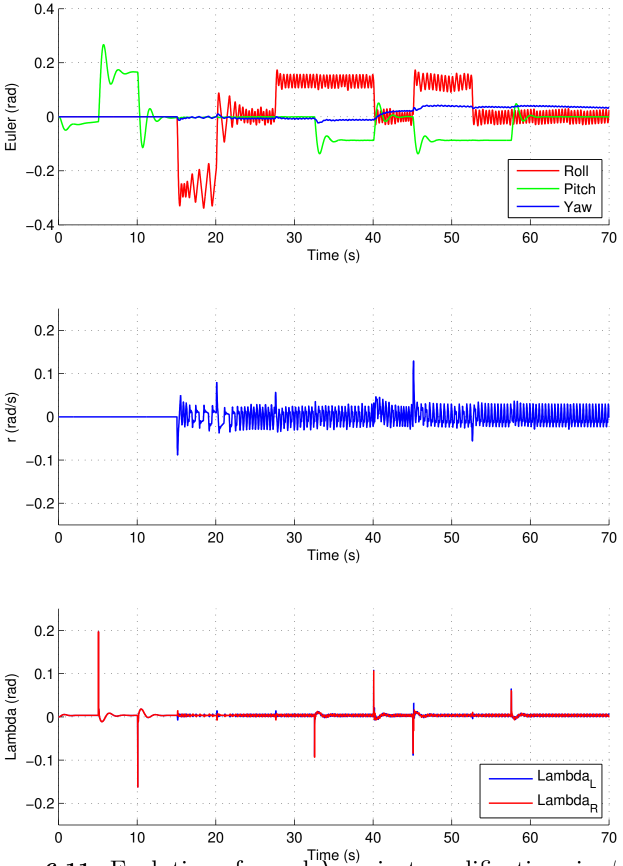


Figure 6.11. Evolution of r and λ against modifications in ϕ and θ

It is also important to remark that the system tends to modify its heading when the hovering takes place at configurations characterised by non-null pitch and roll angles. The reason of this azimuth variation is that the controller is based on the manipulation of the angular velocity r instead of the yaw angle ψ, and therefore, the system acts in order not to change the heading that it has at a specific time t but if this modification occurs, the new heading would be the new reference and the system would not try to recover the initial orientation.

Once the time-response of the tilt-angles has been analysed at different attitude configurations, the same study with the thrust generated by each of the rotors has been performed, as shown in Fig. 6.12.

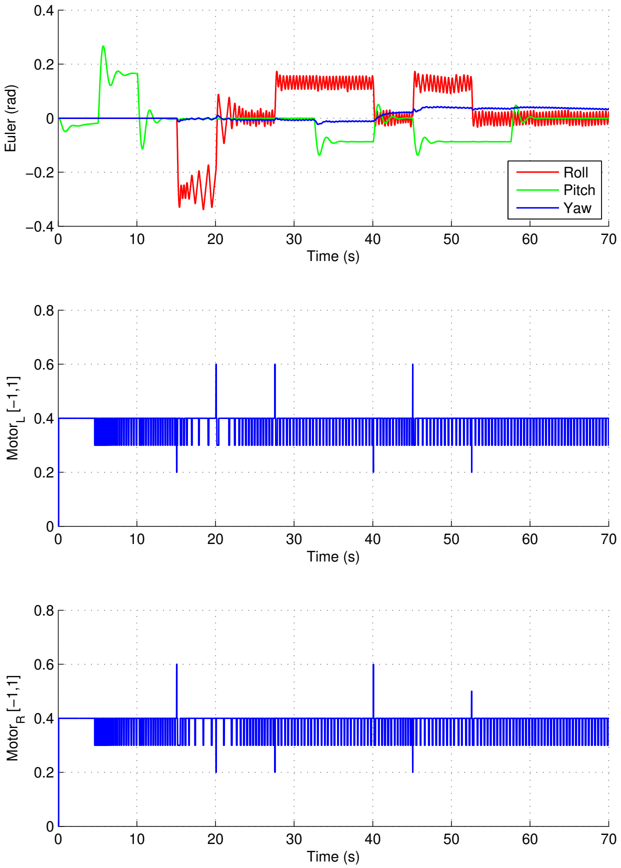


Figure 6.12. Thrust signal against modifications in ϕ and θ

In this second study, it is important to remark two conclusions about the behaviour of the thrust when the system follows the attitude configurations shown in Fig. 6.10 and Fig. 6.11. First of all, as it was established in Section 6.2, a change in roll implies the controller to modify the same absolute value of the rotational velocity in each of the rotor but with different sign. As a result, the thrust is not longer equal until the reference is reached.

The other conclusion refers to the thrust modification associated with changes in pitch from the equilibrium configuration. Although, this effect can not be graphically observed due to the resolution of the thrust signal, when a certain pitch angle is established as a reference, the engines have to create more thrust in order to compensate the fact that the entire system is tilted, and therefore, to maintain the equilibrium of external forces. This means that the higher the pitch angle is, the higher the increase in thrust has to be.

The last study linked to this changes in attitude is focused on the analysis of the time-response of the vertical velocity $\dot{z}_e$ shown in Fig. 6.13. Due to the control scheme design, when the aircraft takes non-equilibrium attitude configurations, it has a tendency to descend, *i.e.*, to increase the positive value of its velocity in the Z_e axis of the earth reference frame. The vertical velocity controller tries to maintain a given reference value allowing to reduce variations in this velocity, but, because the altitude is not directly controlled, the system keeps changing its vertical position.

Figure 6.13 illustrates that each pitch or roll angle apart from the equilibrium ($\phi \neq 0$ or $\theta \neq 0$) implies an increase in $\dot{z}_e$, which is compensated by an increase in the thrust generated by the rotors. Again, since the controlled variable is the vertical velocity, each time a perturbation occurs there is a loss of altitude that is never recovered, but the vertical speed controller manages to stop falling and reaches again the $\dot{z}_e = 0$ set-point.

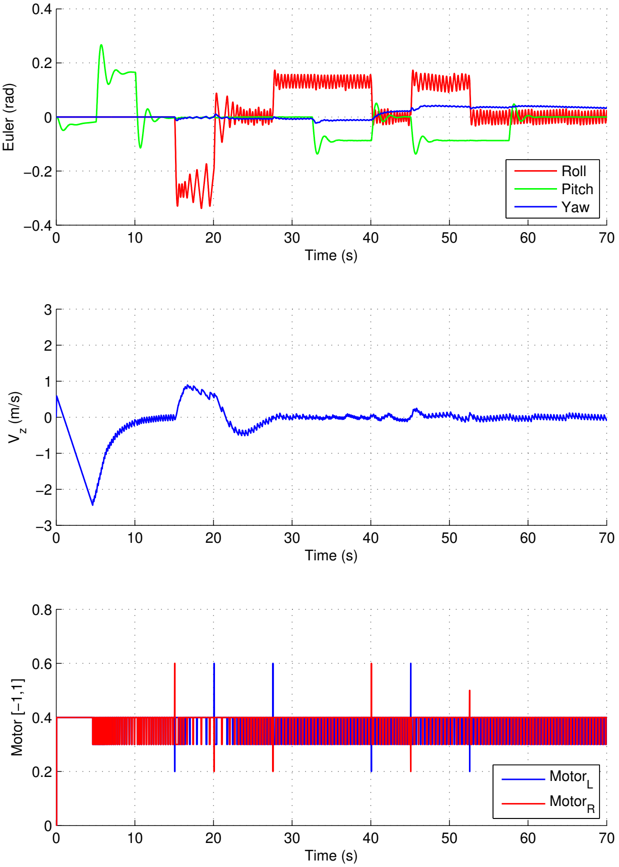


Figure 6.13. Evolution of V_z and thrust signal against modifications in ϕ and θ

6.6.2 Real flight results

In order to verify the performance of attitude control system designed for this UAV, the proposed PID control laws have been implemented in the on-board hardware using the OpenPilot Revolution Board [7], which contains a full 10 DOF IMU with gyroscopes, accelerometers, magnetometers and barometric pressure sensors.

In the experimental test flight, the goal was the stabilisation of the UAV, compensating any external disturbance during vertical flight. Due to the lack of GPS, velocity and position can only be calculated by integration of accelerometers combined with the gyroscopes measures and with the barometric pressure sensor (sensor fusion [16, 32]). As result, the measures of velocity and position are not reliable enough to close the loop. For this reason, in the experimental test presented, the vertical velocity controller has been annulled and the pilot directly acts on the u_4 system input. That means, direct modification of motors nominal thrust which implies a manual (piloted) control of the speed and vertical position.

The experimental results for the platform control are presented in Fig. 6.14 as a time plot of all angles of the UAV and the controller actions.

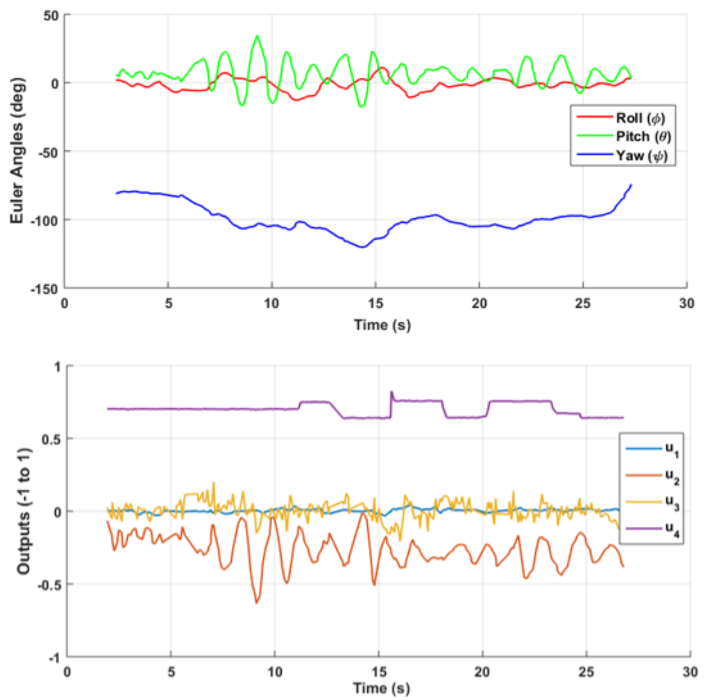


Figure 6.14. Time series of Euler angles and controller outputs

As can be seen, the closed loop response of the UAV is stable because of the fact that Euler angles vary within a limited range, confirming the effectiveness of the proposed approach and theoretical results. Since vertical speed and altitude are manually piloted and the absence of GPS, it is impossible to define a 3D trajectory through waypoints. To cover this drawback, the real test flight demonstration video can be visualized on Youtube [8][2]. Thanks to this video, readers can have an approximate idea of the 3D trajectory of the UAV during the test flight.

[2] V-Skye Prototype: Test Flight - https://youtu.be/zS9oWur-Pss

6.7 Conclusions and Future Work

The design and implementation of the attitude tracking controller for a Bi-Rotor VTOL UAV have been presented in this article. The simulations performed in a HIL environment showed the main aspects of the dynamical behaviour of this system, while the real test flight results presented verify the stability and viability for this UAV platform. Therefore, the UAV designed has the ability to hover despite its complex dynamics and allows future designs based on cascade control schemes focused on controlling global position and velocities. In addition, future works will explore the robust control design that allows the automatic transition between the two different flying modes: VTOL and cruise flight.

Bibliography

[1] Yucel Orkut Aktas, Ugur Ozdemir, Yasin Dereli, Ahmed Farabi Tarhan, Aykut Cetin, Aslihan Vuruskan, Burak Yuksek, Hande Cengiz, Serkan Basdemir, Mesut Ucar, et al. A low cost prototyping approach for design analysis and flight testing of the turac vtol uav. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 1029–1039. IEEE, 2014.

[2] Saffet Ayasun, Robert Fischl, Sean Vallieu, Jack Braun, and Dilek Cadırlı. Modeling and stability analysis of a simulation–stimulation interface for hardware-in-the-loop applications. *Simulation Modelling Practice and Theory*, 15(6):734–746, 2007.

[3] Hanyu Ban, Zhuoran Qi, Bo Li, and Wenquan Gong. Nonlinear disturbance observer based dynamic surface control for trajectory tracking of a quadrotor uav. In *2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI)*, pages 1–6. IEEE, 2018.

[4] John Brandt and Michael Selig. Propeller performance data at low reynolds numbers. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 1255, 2011.

[5] Jesús Velasco Carrau, Gilberto Reynoso-Meza, Sergio García-Nieto, and Xavier Blasco. Enhancing controller's tuning reliability with multi-objective optimisation: From model in the loop to hardware in the loop. *Engineering Applications of Artificial Intelligence*, 64:52–66, 2017.

[6] Hao Chang, Dafang Wang, Hui Wei, Qi Zhang, and GuangLi Dong. Design of tracked model vehicle measurement and control system based on veristand and simulink. In *MATEC Web of Conferences*, volume 175, page 03047. EDP Sciences, 2018.

[7] LibrePilot Community. *Revolution Board Setup — LibrePilot/OpenPilot Wiki 0.1.4 documentation*, 2019 (accessed January 1, 2019). http://opwiki.readthedocs.io/en/latest/user_manual/revo/revo.html.

[8] CPOH. *V-Skye Prototype: Test Flight*, 2019 (accessed January 1, 2019). https://youtu.be/zS9oWur-Pss.

[9] Denes Fodor and Krisztian Enisz. Vehicle dynamics based abs ecu verification on real-time hardware-in-the-loop simulator. In *Power Electronics and Motion Control Conference and Exposition (PEMC), 2014 16th International*, pages 1247–1251. IEEE, 2014.

[10] Salvador Gonzalez-Vazquez and Javier Moreno-Valenzuela. A new nonlinear pi/pid controller for quadrotor posture regulation. In *Electronics, Robotics and Automotive Mechanics Conference (CERMA), 2010*, pages 642–647. IEEE, 2010.

[11] Ian Griffin and Jennifer Bruton. On-line pid controller tuning using genetic algorithms. *Dublin City University*, 2003.

[12] Farid Kendoul, Isabelle Fantoni, and Rogelio Lozano. Modeling and control of a small autonomous aircraft having two tilting rotors. *IEEE Transactions on Robotics*, 22(6):1297–1302, 2006.

[13] Yangkai Liao, Xinchun Shi, Chao Fu, and Jianhui Meng. Hardware in-the-loop simulation system based on ni-pxi for operation and control of microgrid. In *Industrial Electronics and Applications (ICIEA), 2014 IEEE 9th Conference on*, pages 1366–1370. IEEE, 2014.

[14] ZH Ma, QQ Zhang, and LP Chen. Attitude control of quadrotor aircraft via adaptive back-stepping control. *CAAI Trans. Intell. Syst*, 10(3):1–7, 2015.

[15] Tarek Madani and Abdelaziz Benallegue. Sliding mode observer and backstepping control for a quadrotor unmanned aerial vehicles. In *American Control Conference, 2007. ACC'07*, pages 5887–5892. IEEE, 2007.

[16] Panos Marantos, Yannis Koveos, and Kostas J Kyriakopoulos. Uav state estimation using adaptive complementary filters. *IEEE Transactions on Control Systems Technology*, 24(4):1214–1226, 2016.

[17] M Rida Mokhtari, Amal Choukchou Braham, and Brahim Cherki. Extended state observer based control for coaxial-rotor uav. *ISA transactions*, 61:1–14, 2016.

[18] Javier Moreno-Valenzuela, Ricardo Pérez-Alcocer, Manuel Guerrero-Medina, and Alejandro Dzul. Nonlinear pid-type controller for quadrotor trajectory tracking. *IEEE/ASME Transactions on Mechatronics*, 23(5):2436–2447, 2018.

[19] Multiplex. *XENO UNI*, 2019 (accessed January 1, 2019). https://www.multiplex-rc.de/produkte/214241-bk-xeno-uni.

[20] Raul-Octavian Nemes, Mircea Ruba, and Claudia Martis. Integration of real-time electric power steering system matlab/simulink model into national instruments veristand environment. In *2018 IEEE 18th International Power Electronics and Motion Control Conference (PEMC)*, pages 700–703. IEEE, 2018.

[21] Katsuhiko Ogata and Yanjuan Yang. *Modern control engineering*, volume 4. Prentice hall India, 2002.

[22] Juan Paul Ortiz, Luis Ismael Minchala, and Manuel Jeova Reinoso. Nonlinear robust h-infinity pid controller for the multivariable system quadrotor. *IEEE Latin America Transactions*, 14(3):1176–1183, 2016.

[23] Christos Papachristos, Kostas Alexis, and Anthony Tzes. Design and experimental attitude control of an unmanned tilt-rotor aerial vehicle. In *Advanced Robotics (ICAR), 2011 15th International Conference on*, pages 465–470. IEEE, 2011.

[24] Ricardo Pérez-Alcocer, Javier Moreno-Valenzuela, and Roger Miranda-Colorado. A robust approach for trajectory tracking control of a quadrotor with experimental validation. *ISA transactions*, 65:262–274, 2016.

[25] Morgan Quigley, Blake Barber, Steve Griffiths, and Michael A Goodrich. Towards real-world searching with fixed-wing mini-uavs. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3028–3033. IEEE, 2005.

[26] Guilherme V Raffo, Manuel G Ortega, and Francisco R Rubio. Backstepping/nonlinear h∞ control for path tracking of a quadrotor unmanned aerial vehicle. In *American Control Conference, 2008*, pages 3356–3361. IEEE, 2008.

[27] Adnan S Saeed, Ahmad Bani Younes, Shafiqul Islam, Jorge Dias, Lakmal Seneviratne, and Guowei Cai. A review on the platform design, dynamic modeling and control of hybrid uavs. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 806–815. IEEE, 2015.

[28] Anand Sanchez, Juan Escareno, Octavio Garcia, Rogelio Lozano, et al. Autonomous hovering of a noncyclic tiltrotor uav: Modeling, control and implementation. In *Proc. of the 17th IFAC Wold Congress*, pages 803–808. Citeseer, 2008.

[29] Louis V Schmidt. *Introduction to aircraft flight dynamics.* American Institute of Aeronautics and Astronautics, 1998.

[30] Pranay Sinha, Piotr Esden-Tempski, Christopher A Forrette, Jeffrey K Gibboney, and Gregory M Horn. Versatile, modular, extensible vtol aerial platform with autonomous flight mode transitions. In *Aerospace Conference, 2012 IEEE*, pages 1–17. IEEE, 2012.

[31] Robert F Stengel. *Flight dynamics.* Princeton University Press, 2015.

[32] Matias Tailanian, Santiago Paternain, Rodrigo Rosa, and Rafael Canetti. Design and implementation of sensor data fusion for an autonomous quadrotor. In *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014 IEEE International*, pages 1431–1436. IEEE, 2014.

[33] Kimon P Valavanis. *Advances in unmanned aerial vehicles: state of the art and the road to autonomy*, volume 33. Springer Science & Business Media, 2008.

[34] J Velasco and Sergio García-Nieto. Unmanned aerial vehicles model identification using multi-objective optimization techniques. *IFAC Proceedings Volumes*, 47(3):8837–8842, 2014.

[35] Jesús Velasco-Carrau, S García-Nieto, JV Salcedo, and Robert H Bishop. Multi-objective optimization for wind estimation and aircraft model identification. *Journal of Guidance, Control, and Dynamics*, 39(2):372–389, 2015.

[36] Aslihan Vuruskan, Burak Yuksek, Ugur Ozdemir, Adil Yukselen, and Gokhan Inalhan. Dynamic modeling of a fixed-wing vtol uav. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 483–491. IEEE, 2014.

[37] Rui Wang, Zhou Zhou, and Yanhang Shen. Flying-wing uav landing control and simulation based on mixed h 2/h. In *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, pages 1523–1528. IEEE, 2007.

[38] Dunzhu Xia, Limei Cheng, and Yanhong Yao. A robust inner and outer loop control method for trajectory tracking of a quadrotor. *Sensors*, 17(9):2147, 2017.

[39] Bo Zhao, Bin Xian, Yao Zhang, and Xu Zhang. Nonlinear robust sliding mode control of a quadrotor unmanned aerial vehicle based on immersion and invariance method. *International Journal of Robust and Nonlinear Control*, 25(18):3714–3731, 2015.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

This thesis is dedicated to the study of modelling and control of unmanned aircraft systems, always posed as a multi-objective optimization problem. Although all of them are framed in this theme, each chapter itself is the publication of a research work that begins and ends in that chapter. Therefore, the conclusions drawn have been reflected here separately, grouping them according to the two main parts of this thesis.

7.1.1 Conclusions on part I: Modelling and Parameters Identification

In chapter 2, a methodology for the identification of UAVs aerodynamic models is presented. Besides a demonstration of its application in a real system is carried out with satisfying results. The technique presented gives the already stated advantages in the data analysis and the identification process, since it involves a phase of decision making by the designer. This phase allows then, the study of several models and the election of the one that fits better with the designer needs. In addition, confronting experiments offers information about the difficulties of finding a model that fits different flight conditions at the same time, which improves the understanding of the system. This technique may also give information about the importance of a particular kind of experiment in the identification of the model. All these advantages lead to better models that may save time and money when designing autonomous aircraft control algorithms and are particularly interesting for low cost MAVs for which CFD or wind tunnel experiments information is normally not present.

In chapter 3 a two-step identification technique for MAV in the absence of air-data sensors is presented. In the first step, a multi-objective optimization procedure is proposed to estimate wind during the flight experiments. A simulation environment that includes a MAV model that can be subjected to constant and variable winds is used to confirm the estimate process. Conditions in different flight tests in which one or more system inputs are excited are simulated, and after acquiring necessary data from simulations, the wind estimation technique is applied. Several conclusions can be extracted from these simulations. First, under ideal conditions, wind estimation is successfully achieved in any experiment. Second, it can be concluded that only lateral experiments offer enough information to enable wind estimation under realistic conditions. Aircraft orientation relative to the wind azimuth does not vary during longitudinal experiments. Therefore, wind observability is significantly reduced. The same reason is reflected in the improved performance detected during the aileron experiments when compared with rudder experiments. Observability is improved when a wider range of orientation values are covered. For this reason, mixed experiments, *i.e.*, experiments that excite longitudinal and lateral variables simultaneously, achieved better results for wind estimation. As shown in section 3.5.2, this type of experiments can theoretically be used to estimate the wind and identify any of the aerodynamic coefficient models. But this fact has only been checked on simulation.

The same wind estimation procedure is applied to real flight data for lateral experiments. The obtained results infer that the wind is estimated and that the information obtained can be used to correct airspeed dependent measurements. As a final remark, the estimation technique presented in chapter 3 is not intended to replace air-data sensors (whenever available). However, MAVs price and size make it sometimes impossible to include quality data sensors. Therefore, in those cases when no information at all can be used, a rough estimation of wind speed can significantly improve models quality. In addition, a similar multi-objective optimization approach might also be employed when partial airspeed information is available. That information could be incorporated in the optimization problem in order to improve airspeed measurements.

In the second step of the methodology presented in publication of chapter 3, multi-objective optimization is again proposed to take advantage of the available flight data. The presented approach enables diverse experiments to be utilized, so that adjusting model parameters becomes, in reality, a multi-objective problem. This approach enables obtaining a compromise model that suits some flight situations without losing much performance in others. Furthermore, the visualization of the model fitness for several trials provides an idea of the quality of the obtained data and of the selected model structure. Although mean squared error has been used here, using a heuristic optimizer also enables the use of other performance indicators. For example, the mean absolute error is normally more meaningful to engineers because it has the same magnitude as the variable being modeled. This second step, is in accordance to the methodology used in chapter 2.

7.1.2 Conclusions on part II: Control, Simulation and Test

Chapter 4 presents a comparative study between classic PID control techniques, and other model based structures where, the power of taking a multi-objective design approach shows advantages on reducing uncertainties. Indeed, results presented in chapter 4 show that the most robust strategies for the proposed scenario are PID controllers and, particularly, PIDs adjusted by means of mutli-objective optimization stand out over the others. This is mainly due to the existence of parametric uncertainties residing on the aerodynamic coefficients. More complex techniques, such as LQR and MPC, have much higher reliance on the information provided by the model, since it is used in the design phase for LQR and even in the control phase for MPC. Therefore, the behavior of these strategies deteriorates significantly on real situations when the reference model is not reliable.

However, despite the results obtained from the point of view of dynamic response, the MPC strategy presents a very interesting advantage, the proposal of input signals much less aggressive and smoother than the rest of controllers. This fact by itself can be considered as a sufficient motivation for choosing such techniques, since they ensure a much longer life cycle to the set aircraft actuators, which is an issue of vital importance in this type of systems. In addition, an iterative process of improving the values of the derivatives of stability and control could significantly improve the performance of the LQR and MPC controllers.

Going deeper on the tuning strategy follow in chapter 4, a systematic approach to enhance controller performance evaluation and design is presented throughout chapter 5. Multi-objective optimization is used in conjunction with different simulation platforms in order to provide the integrative framework on which the methodology is based. Thereby, MiL, SiL, PiL and HiL platforms (or a subset of them) can be employed in successive multi-objective optimizations in order to gain better understanding of the problem as the process moves forward. As we saw, the information obtained in previous stages is used in two directions when a new optimization is to be posed: (*i*) more meaningful objectives can be stated and (*ii*) preferences and solution constraints can be better delimited. On one side, (*i*) generally leads to more complex MOPs, what obviously increases the computational burden. On the other side, (*ii*) reduces the search space helping, therefore, to decrease that complexity during the optimization stage. Both (*i*) and (*ii*) go in the direction of getting more adequate solutions (in the sense of what the designer prefers) for the problem at hand.

As a demonstrator, a UAV system with a predefined control structure has been presented. That structure is formed by a total of five PI controllers that perform attitude control and navigation tasks. The ten PI parameters had to be tuned so that the resulting controller was able to drive the aircraft in the supervision of several way-points. Section 5.4 showed how the methodology can be adapted to this specific problem, in accordance to the available simulation platforms. As the process moved forward, reliability objectives, where several missions were actively accomplished inside the MOO, were posed. At the same time, a better understanding of the problem allowed the designer to refine preferences' matrices, as well as optimization limits. A final MCDM stage comprising several missions inside the HiL platform has been performed. In that last phase, a controller was chosen from the rest.

To validate the final choice, a real flight mission has been carried out. The chosen controller and a second one within the final Pareto set have been tested exactly as they were coded for HiL simulations. In Section 5.5 experimental results confirmed that the selected controller was as good accomplishing flight missions as the MCDM stage suggested. This evidenced that a realistic model of the system is of great importance to obtain a controller that behaves as the designer expects. A second conclusion is that, even when the HiL platform cannot be actively used in the MOO process, including it in the final MCDM stage is a good practice.

Several new flight paths can be taken as future works. First, adapting the HiL simulation platform so that it can be actively used in the MOO process could be of great interest (although practicality must be had in mind). Second, the authors would like to study the applicability of the technique on the design of multi-variable controllers. Indeed, tuning weighting matrices for linear quadratic regulators or adjusting design parameters in model based predictive controllers can be something tricky when the system comprises many states and inputs. Third, applying this technique to systems with modeled parametric uncertainties can be a good practice to achieve a certain level of robustness.

Finally, the design and implementation of the attitude tracking controller for a Bi-Rotor VTOL UAV have been presented in chapter 6. The simulations performed in a HIL environment showed the main aspects of the dynamical behavior of this system, while the real test flight results presented verify the stability and viability for this UAV platform. Therefore, the UAV designed has the ability to hover despite its complex dynamics and allows future designs based on cascade control schemes focused on controlling global position and velocities. In addition, future works will explore the robust control design that allows the automatic transition between the two different flying modes: VTOL and cruise flight.

7.2 Research Impact

Table 7.1 gives different indices that sum up the overall contribution of this research. Journals' Impact factor and CiteScore values correspond either to the year when the papers where published or the most recent values known.

Paper Title	Journal/Congress	Year	Citations[1]	CiteScore[2]	JCR-Q[3]
Unmanned Aerial Vehicles Model Identification using Multi-Objective Optimization Techniques	19th World Congress of The International Federation of Automatic Control.	2014	6	N/A	N/A
Multi-Objective Optimization for Wind Estimation and Aircraft Model Identification.	Journal of Guidance, Control, and Dynamics	2016	21	2.77	1.856 - Q1
Control Strategies for Unmanned Aerial Vehicles under Parametric Uncertainty and Disturbances: a Comparative Study.	IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles	2015	4	N/A	N/A
Enhancing controller's tuning reliability with multi-objective optimisation: from Model in the loop to Hardware in the loop.	Engineering Applications of Artificial Intelligence.	2017	14	3.75	2.819 - Q1
Motion Equations and Attitude Control in the Vertical Flight of a VTOL Bi-Rotor UAV.	Electronics	2019	4	1.9	2.412 - Q2

Table 7.1. Research citations and impact measurements

From indices and citations presented on table 7.1 it can be concluded that the works exposed on this thesis have received a successful acceptance/impact on the community of the field of research.

7.3 Future Works

Every research work opens new research paths to explore. In the case of this thesis, the platform presented in chapter 6, represents a very interesting dynamic system on which some of the techniques developed throughout this work could be applied or even extended.

[1] Data source: Google Scholar

[2] Data source: Scopus

[3] Data source: Web of Science

A very interesting field of study is the inclusion of some key aircraft constructive parameters in the MOOD process from chapter 5 together with the controller parameters. The main idea is to optimize control robustness and performance objectives by tuning, not only the controller parameters themselves, but also those constructive parameters that affect such performance at the same time. This idea seems interesting and a very powerful way to tackle the pursued goals even before the aircraft has been designed and built. The complexity of the VTOL aircraft presented in chapter 6 seems a perfect platform to apply and test this concept. As an example, the position of the two rotors (distances to the gravity center and from each other) affects directly to the maneuverability of the system, and therefore, to the control performance and robustness. Including those parameters on the MOOD process can help to improve the desired objectives indexes.

Making use of the concept presented in chapter 6 again, the dynamic model based on the aircraft constructive parameters might not reflect its real dynamic behavior in a reliable manner. Besides, the work performed on chapter 6 is only focused on the vertical flight mode. For these reasons, designing and executing experimental flights to adjust the dynamic model for every flight mode would be one of the following steps to be taken. To do so, multi-objective optimization techniques presented in chapters 2 and 3 have proven to be useful strategies.

On chapter 4, MPC controller stands out because of how this controllers handles actuators. In that work, PID controllers tuned by means of MOOD is taken as a better strategy because of its better performance in the cost indexes. On one side, it seems logical, since the PID controller has been optimized for those objectives. And, on the other side, model dependence is an intrinsic feature of MPC controller, what makes it somehow fragile to model uncertainties. In this sense, working on improving the dynamic model incorporated to the controller could be a way to increase its overall performance, making MPC a good candidate to substitute PID strategy. On the other hand, weighting matrices can also be tuned from a MOOD perspective. The problem here is that the space of solution widens, since the number of parameters to tune increases significantly. However, working on this possibility is also an interesting way to make MPC a realistic alternative to PID controllers.

Acknowledgments

The author and supervisors would like to thank the following grants for funding part of the research presented along this thesis:

- RTI2018-096904-B-I00 from *Ministerio de Ciencia, Innovación y Universidades* of the Spanish government.
- BES-2012-056210 from the past *Ministerio de ciencia e innovación* of the Spanish government.
- TIN-2011-28082 from the past *Ministerio de ciencia e innovación* of the Spanish government.
- ENE-25900 from the past *Ministerio de ciencia e innovación* of the Spanish government.
- PROMETEO/2012/028 from the *Generalitat Valenciana.*