Universitatea Politehnica din Bucuresti
Spl. Independentei 313
060042 Bucuresti
Romania

www.upb.ro

Facultatea de Inginerie Aerospatiala
str. Gh. Polizu 1-5
tel. +40 21 402 3812

www.aero.pub.ro
inginerie.aerospatiala@upb.ro

Ingineria Sistemelor Aeronautice "Nicolae Tipei" Department

BEng Final Project

Aeroespace Engineering

University POLITEHNICA of Bucharest

Development and comparison of modern and conventional automatic landing system architectures for a flying wing UAV

Research done by:

Belmonte Hernández, Carlos c.belmonte971a@gmail.com

Directed by:

Stoica, Adrian - Mihail adrian.stoica@upb.ro

Session: July 2019

Anti - Plagiarism Declaration

I the undersigned ***Carlos Belmonte Hernández****, student of the University Politehnica of Bucharest, Faculty of Aerospace Engineering, declare herewith and certify that this final project is the result of my own, original, individual work. All the external sources of information used were quoted and included in the References. All the figures, diagrams, and tables taken from external sources include a reference to the source.*

Date: Signature:

Contents

List of Figures

List of Tables

Acronyms

BEA British European Airways.

BLEU Blind Landing Experimental Unit.

CAT Category.

DCM Direction Cosine Matrix.

FAA Federal Aviation Agency.

ICAO International Civil Aviation Organization.

ILS Instrumental Landing System.

ISR Intelligence, Surveillance and Reconnaissance.

NLC National League of Cities.

PID Proportional-Integral-Derivative.

RAF Royal Air Force.

SAS Stability Augmentation System.

UAV Unmanned Aerial Vehicule.

REPORT

Abstract

The aim of this Final Thesis is to accurately make use of two different techniques in control theory in order to design and develop the automatic landing system of a generic flying wing UAV. This landing system will focus on the study of the tracking error response of the distance between the actual position of the drone and the desired landing trajectory, which will be determined by specifying a certain glide path angle in the control design.

The first technique to be implemented will be based on the PID controller theory: tuning the values corresponding to the proportional, integrator and derivative that give the name to this method, we will be able to reach the system stability and performance requirements. We will adapt the glide-path-coupled control system in McLean (Bibliography cite number [21]) to the needs of our UAV, determining the gain values in the loops in order to stabilize the system.

Then, we will study the behavior of the controller when applying different types of perturbations simulating the wind effect: firstly with constant values for the wind velocity components in the longitudinal axes, and secondly with random values, making use of the Dryden turbulence model.

For this approach we will consider a desired glide path angle of 2.5°, the standard slope for aircrafts in the landing phase.

The second technique will be the H infinity or gamma attenuation problem, which belongs to the family of the so called *robust controllers*. This method builds a closed loop system shaped by a plant model (determined by the aircraft dynamics, actuator, weightings, and so on) and the controller. We will have to define the inputs and outputs of the system, with the perturbations, commanded and measured variables, actuator terms, and tracking error and performance specifications. The aim of the process is to calculate and obtain a controller that optimizes these specifications.

For this approach we will consider a desired glide path angle of 12°, a possible realistic landing slope for small aircrafts like flying wing UAVs.

After obtaining the controller, we will perform a robustness analysis of the same, in order to study its tracking performance along the landing trajectory and to identify its level of independence of the uncertain parameters.

Abstract

Scopul acestei teze este de a prezenta două tehnici diferite în teoria controlului pentru a proiecta și dezvolta sistemul de aterizare automată a unui UAV cu configuratie aripa-zburatoare. Acest sistem de aterizare se va concentra asupra studierii răspunsului la eroarea de urmărire a distanței dintre poziția actuală a UAV-ului și traiectoria de aterizare dorită, care va fi determinată prin specificarea unui anumit unghi al pantei de aterizare.

Prima metoda care va fi implementată este conventionala si va fi bazată pe teoria controlerului PID: ajustarea valorilor corespunzătoare coeficientilor proportional, integrator si derivativ care dau numele acestei metode, se pot satisface cerințele de stabilitate și performanță ale sistemului. S-a adaptat arhitectura de control prezentata in McLean (Bibliografia citează numărul [21]) la nevoile UAV, determinând valorile amplificarilor în bucle pentru a stabiliza sistemul.

Apoi, s-a studiat comportamentul controlerului atunci când aplicăm diferite tipuri de perturbații care simulează efectul vântului: în primul rând cu valori constante pentru componentele vitezei vântului în axele longitudinale și, în al doilea rând, cu valori aleatorii, folosind modelul de turbulența Dryden.

Pentru această abordare s-a luat în considerare un unghi dorit al pantei de aterizare de 2.5°, panta standard pentru aeronave în faza de aterizare.

A doua tehnică va fi problema H infinit sau gamma atenuare, care aparține familiei așa-numitelor metode de control robust. Această metodă construiește un sistem de buclă închisă, modelat în funcție de asa-numitul sistem generalizat (determinat de dinamica aeronavei, de dispozitivul de acționare, de ponderi etc.) și de controler. Se definesc intrările și ieșirile sistemului, cu perturbatiile, variabilele comandate și măsurate, termenii de acționare și erorile de urmărire și specificațiile de performanță. Scopul proiectarii este de a determina un controler care minimizeaza norma H-infinit a sistemului rezultant obtinut prin cuplarea controllerului la sistemul generalizat.

Pentru această abordare vom lua în considerare un unghi dorit al pantei de aterizare de 12°, o posibilă pantă reală de aterizare pentru o mică aeronavă, cum ar fi UAV-urile cu configuratie aripa zburatoare.

După obținerea controlerului, s-a efectuat o analiză de robustețe a acestuia, pentru a studia performantele de urmarire a traiectoriei de aterizare sa și pentru a identifica nivelul său de independență a parametrilor incerți.

Chapter 1

Project approach

1.1 Introduction

In this project we will be dealing with two of the most present topics in the aviation environment. In first place, the automatic landing system in aircrafts. The development of a precise and accurate system has been a major concern in the last years and one of the most crucial aspects concerning safety and operation: such system that allows the aircraft to autonomously land supposes a great advantage, since the aircraft wouldn't be so affected by the weather conditions in such a critical phase of the flight. Increasing the number of operations, reducing the delays and route changes, or increasing safety are only some of the most direct consequences of developing such systems.

Several categories were designed, according to the level of autonomy of the landing system; the current trends aim to achieve the highest level for commercial flights.

In second place, UAVs. The development and production of these pilotless aircrafts has increased exponentially in the last decades: from military to industrial and commercial uses, currently UAVs are utilized in every imaginable field, given their versatility: agriculture, marketing, real state agencies, cinema, photography, topographic studies, search and rescue, as toys, are some examples. They are being developed and studied, increasing their capabilities, and are so widely spread that international and national aviation organizations are classifying them and legislating their use.

Since the autonomous flight nature of UAVs, it goes without saying that the implementation of control systems is a crucial aspect in their design: this work aims to get together those two important topics, through the development of an automatic landing sytem for a flying wing UAV.

The current state of the art is quite advanced: the landing system is one of the most crucial aspects in the development of a UAV, and therefore we can find multiple systems already designed in the industry. Besides, many of the automatic landing systems patents for the first drones, which were hold by the military sections, were published, what was benefited by the new commercial UAVs.

We have as many types of landing systems as drones we can find, since the landing procedure will change depending on the aircraft function, type and sensors used: flying wing drones will differ from multi-rotor, if they have infrared or optical sensors, if they need to land in movement targets or in immobile runways, and so on.

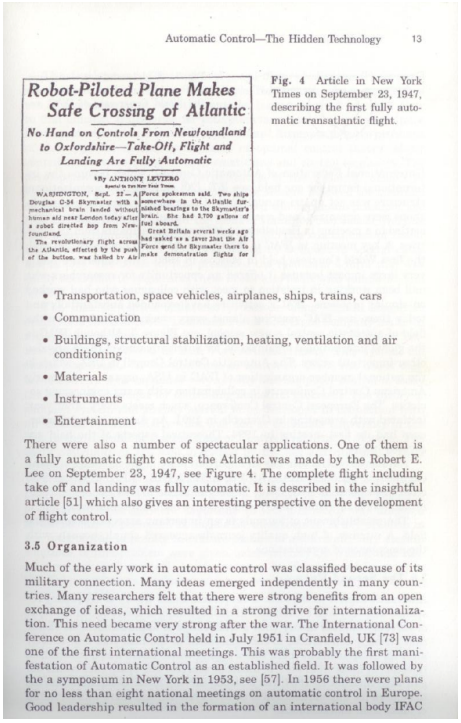
1.2 Development of the autolanding concept

Now, it is believed that it should be given a general overview of the automatic landing system importance, development, and current situation in aviation, in order to situate the context and actuality of such systems.

The advantages of automatically guiding an aircraft during its flight, with no help of the pilot, was introduced almost one hundred years ago. The application of automatic pilots to the landing phase has been developing for more than 70 years; indeed, the first mention to this kind of navigation technique appeared the first time on 1932, when The Naylor Company (San Antonio, Texas) published one book that set the grounds for what in the following years would became one of the main concerns in the air navigation field.

With the title of Blind Flight in theory and practice, major William C. Ocker and first lieutenant Carl J. Crane developed the main ideas about how a flight with no use of exterior visual references (called blind flight) could be carried out. In this book, both authors presented the navigation aids and instruments that would be necessary for a blind flight and the training that the pilot would have to follow. In this way, the writers wanted to join both the theoretical and practical aspects of an autoflight, pretending it to be useful for pilots in adverse weather conditions.

(a) Stout , Holloman, and Crane, with the Fokker in 1937. Taken from [1].



Automatic Control—The Hidden Technology 13

Robot-Piloted Plane Makes Safe Crossing of Atlantic

No Hand on Controls From Newfoundland to Oxfordshire—Take-Off, Flight and Landing Are Fully Automatic

By ANTHONY LEVERO

WASHINGTON, Sept. 27 — A Douglas C-54 Skymaster with a mechanical brain landed without human aid near London today after a robot directed hop from Newfoundland.

The revolutionary flight across the Atlantic, effected by the push of the button, was hailed by Air Force spokesman said. Two ships somewhere in the Atlantic furnished bearings to the Skymaster's brain. She had 3,700 gallons of fuel aboard.

Great Britain several weeks ago had asked as a favor that the Air Force send the Skymaster there to make demonstration flights for

Fig. 4 Article in New York Times on September 23, 1947, describing the first fully automatic transatlantic flight.

- Transportation, space vehicles, airplanes, ships, trains, cars
- Communication
- Buildings, structural stabilization, heating, ventilation and air conditioning
- Materials
- Instruments
- Entertainment

There were also a number of spectacular applications. One of them is a fully automatic flight across the Atlantic was made by the Robert E. Lee on September 23, 1947, see Figure 4. The complete flight including take off and landing was fully automatic. It is described in the insightful article [51] which also gives an interesting perspective on the development of flight control.

3.5 Organization

Much of the early work in automatic control was classified because of its military connection. Many ideas emerged independently in many countries. Many researchers felt that there were strong benefits from an open exchange of ideas, which resulted in a strong drive for internationalization. This need became very strong after the war. The International Conference on Automatic Control held in July 1951 in Cranfield, UK [73] was one of the first international meetings. This was probably the first manifestation of Automatic Control as an established field. It was followed by the a symposium in New York in 1953, see [57]. In 1956 there were plans for no less than eight national meetings on automatic control in Europe. Good leadership resulted in the formation of an international body IFAC

(b) Article in the *New York Times* on September 23, 1947, describing the first fully automatic transatlantic flight. Taken from [2].

Figure 1.1: Beginnings of the automatic control systems development in aviation.

The implementation of this theory would be materialized in the landing phase a few years later. Crane, together with an Army aviator called G. V. Holloman and a civil called Raymond Stout, carried out the first known successful demonstration of an auto-landing, the 23rd August of 1937, at Wright Field in Dayton, Ohio. They used a Fokker C-14B that carried air navigation instruments developed by Crane (who held more than 100 patents) [1].

Nevertheless, the U. S. government never decided to continue with the development of a fully automatic landing system, and the project didn't have any further continuation. It wasn't until 1945 that another military corporation got interested in the opportunities that the auto-land offered. After Second World War, Great Britain took conscience of their vulnerability to air attacks. With the development of nuclear weapons and their threaten, they decided that their military operations couldn't be conditioned to the fre-

quent very–low–visibility–weather conditions in the islands, so common in the Northwest part of Europe. Anticyclones, radiation fogs, and the big amount of carbon and smoke particles in the air coming from the coal burning heating and power generation systems leaded to long-persistence fogs that made it really difficult to successfully complete flight operations.

Due to this, the Royal Aircraft Establishment, encouraged by the advances in radio navigation aids and automatic pilots, formed the Blind Landing Experimental Unit (BLEU) which was in charge of the development of a reliable automatic landing system. By 1949, around five hundred landings were done at Woodbridge Field, in Suffolk, England, with special equipment that allowed the pilot to only take care of functions during landing such as the operation of throttles, flaps or brakes [3]. During the next decade the auto-landing system was improved and finally implemented on RAF aircrafts, with the help of British European Airways (BEA).

BEA was a state–owned airline corporation, established by the British government, whose routes were focused on airports in Europe and in the UK. Problems related with disorganizations in scheduled flights, and long indefinite periods of air traffic suspension due to bad weather, involved the airline in financial loss. After a series of continuous accidents in the approach and landing phases due to poor visibility, the company decided to take part in any way of assuring that their pilots could land in a safe way in such weather conditions. Since BEA was heavily involved in the improvement of the auto-land system together with the RAF personnel, after the complete development of the auto-land and posterior implementation on the RAF aircrafts it was added to the BEA fleet in 60's, formed by Hawcker Siddeley Tridents. The first commercial flight with passengers aboard in complete a fully automatic landing was the Trident 1 G-APR on 10 June 1965, in a flight from Paris to Heathrow.

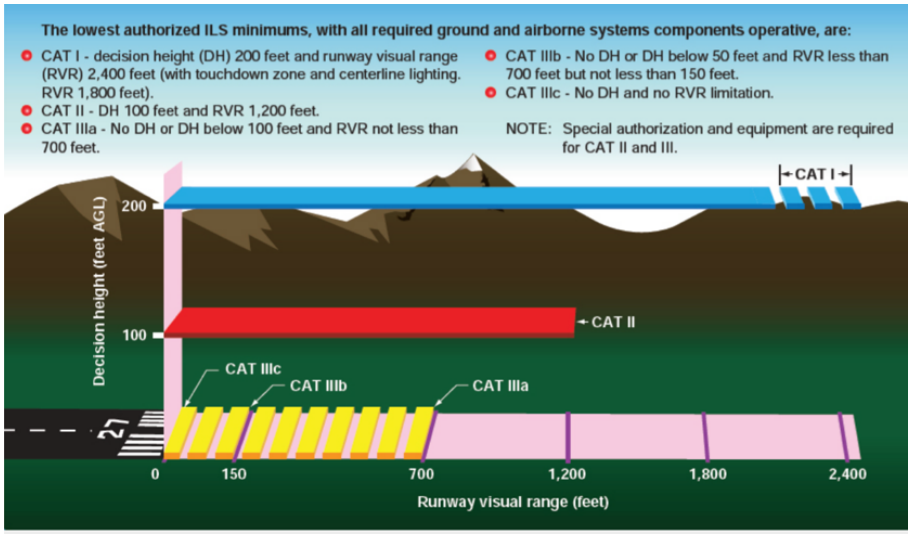


Figure 1.2: Classification of the lowest authorized ILS minimumt for the precission approaches of CAT I, CAT II, and CAT III. Taken from [6].

From now on, the auto-land would spread through all Europe, specially in those countries more affected by adverse weather. Airlines in France, Switzerland or Finland incorporated this system to their aircrafts. Indeed, the first aircraft to be certified to CAT III ICAO standards was the Sud Aviation Caravelle, on 28th December 1968, an aircraft that was part of France Airlines, Swiss Air, and Finnair. It was followed by the BEA's Hawker Siddeley Trident on May 1972.

Nowadays, the auto-land system has been completely incorporated to the air navigation field. Actual commercial automatic landing systems use the ILS system to adjust the lateral and vertical guidance, in order to keep the aircraft's track aligned with the centreline of the runway [4].

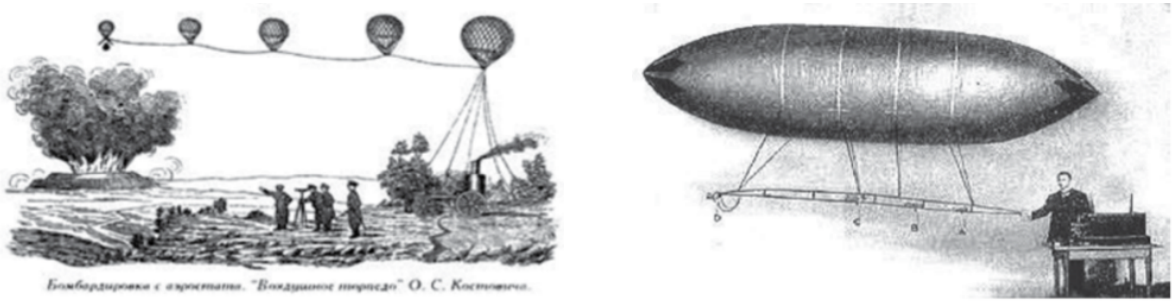
1.3 Considerations about UAVs

In this section we will discuss the history and current situation of UAVs, given the nature of the aircraft to be studied in this project. It will be summarized the information found concerning evolution, design, uses and future perspectives so that the reader can have a general idea of such type of pilotless systems.

An Unmanned Aerial Vehicle (UAV), commonly known as drone, is defined by ICAO as "a pilotless aircraft which is flown without a pilot-in-command on-board and is either remotely and fully controlled from another place (ground, another aircraft, space) or programmed and fully autonomous" [5].

UAVs technology was developed initially for military uses and then introduced in the civil world. They were initially designed to be unmanned weapons to be implemented in the most dangerous, difficult or dirty operations, and it wasn't until that their technology was fully operational and proven effective in the military field that they were introduced for commercial and civil purposes.

For understanding UAVs in our days, we must go back to the 19th century. One of the first recorders dates back on July 1849, during the Austrian war against Venice. A lieutenant named Franz von Uchatius had the idea of attacking Venice with pilotless balloons loaded with explosives. The city surrounded two days after the second attempt, although the operations wasn't really effective since the Austrian Forces wasn't able to control the balloons. Many of them exploded in the air, fell down to the sea, and even stroke back the Austrian army.



(a) Drawing of the Austrian unmanned balloons against Venice.

(b) Drawing of the presentation of the Tesla concept for his wireless controlled balloon.

Figure 1.3: First developed pilotless aircraft concept designs. Taken from [7].

During the U. S. Civil War of 1861, both forces (the Confederated and the United) used unmanned balloons for reconnaissance missions; in the Spanish – American war of 1989, the U. S. forces incorporated a camera to a kite, taking one of the first aerial reconnaissance photographs; a few years later, in 1900, Nikola Tesla introduced the concept of a wireless controlled balloon.

It was during the First World War that the modern concept of UAV was built. The brightest minds of the day were called to develop some kind of system combining wood and fabric with the ability to carry on explosives in a 40 miles range (around 64 km). However, those first aircrafts had two main operational problems: stabilization during flight, and launching and recovering of the airplanes. During the Interwar Period the advances in radio-communications and aircraft engineering allowed to advance significantly to an operational UAV system, mainly as radio–controlled portable platforms weapons. In fact, in the Second World War both the Allies and the Axis forces used them like weapon delivery platforms, radio–controlled gliding/flying bombs, and for anti gun practice [8].

Figure 1.4: On 15 September 1924, radio controlled Curtiss F – 5L flown remotely during all phases of flight, being the first aircraft in history of flight in achieving this. Taken from [8].

After the Cold War started, the U. S. Army leaded the development of UAVs. They began to be used as Intelligence, Surveillance and Reconnaissance (ISR) systems. The main concern of the U. S. was to suppress the spread of Communism, and one of the measurements they took was the creation of a intelligent database to support strategic planning [8]; although during the Korean War in 1950 UAVs were used against selected targets, continuing in a certain way with their former use in the Second World War, the rate of success of less than the 50% made the U. S. Government stop the program. During the Cuban Missile Crisis, it was generated the necessity of developing some kind of aircraft to image the island to confirm if the missiles had been removed. UAVs were thought to be a good option; nevertheless, the ones that were built were not tested yet, so and the U. S. military forces had to make use of a warcraft.

It wasn't until August 1964, in the Tonkin Gulf incident, that UAVs were accepted for service, when Teledyne Ryan AQM – 34 were launched for reconnaissance operations in China. Besides, during the Vietnam War, the American Forces designed one of the most sophisticated programmes of drone surveillance in the history of flight: between 1964 and 1975 a total of 3435 reconnaissance UAV missions were launched. It was the first war with an extensive UAV use.

Figure 1.5: 6 Pioneer Short Range (SR) UAV. Taken from [10]

Nevertheless, in the 70's and 80's Israel took the technological torch from the U. S. and they started to use and to research on unmanned vehicles against Syria and Palestine, creating the Pioneer UAV Corporation [9]. In 1973, they used their UAVs against Egyptian missile sites and armored vehicles in the Yum Kippur War, and in 1982 against Syrian missile emplacements [8].

However, on these days UAVs had to compete with new high–speed missiles systems, long–range bombers, and cruise missiles, and their development ceased during a decade. It wasn't until the First Persian Gulf War (1990 – 1991) that they started to be taken into consideration again, due to the high effectiveness of the Pioneer, the Israeli UAV, what made them a useful weapon in many wars, such as the Second Persian Gulf War, Bosnia, Haiti, Somalia, Afghanistan or Pakistan [8], spreading their use all over the world.

As the technology on UAVs started to improve in the military sector, it was thought that they could be used for other applications rather than military operations. Around 2006 the U. S. Government started to use them for disaster relief, border surveillance and wildfire fighting, and American private companies used them to spread pesticides on their farms [11]. Since there, the private use of UAVs has increased exponentially. In 2011, ICAO wrote the circular 328 to regulate the civil use of UAVs, and according to a Wall Street Journal report in 2012, many U. S. universities and law enforcement companies started to be given approval by the Federal Aviation Agency (FAA), the U. S. civil aviation regulatory organization, to fly their own drones for private use since 2010.

UAVs are a really useful tool whose uses and applications are almost unimaginable. In 2016 the National League of Cities (NLC) wrote a report giving some possible city uses such as rural ambulances, inspections or environmental protection and monitoring [12], but there are a huge number of other applications in many fields such as advertising and marketing, videography and photography in homemade videos, weddings, tourism, movies, action and sport, disaster response of search and rescue, flooding, wildfire, damage assessment or deliver emergency equipment, education, insurance, meteorology, engineering, mapping, maritime, or construction [13].

(a) H201S Spy Hawk: Flying wing commercial drone with camera and video. Taken from [14].

(b) Phantom 4 Professional: Quadcopter commercial drone with camera and video. Taken from [15].

Figure 1.6: Some examples of current commercial UAVs.

The FAA estimates that around 11 million commercial drones will be sold by 2020 only in the US. Only the drones sold in 2020 are expected to be its 40%, and the great majority of them providing aerial photography and data collection services [12]. According to Statista, "the market for commercial drones is expected to reach 13 billion U. S. dollars by 2025. Currently the commercial UAV production for mini – UAVs is valued at 58.4 million U. S. dollars. The global research and development budget for drones is estimated to reach around 4 billion U. S. dollars by 2020, and Amazon, the world's largest online retailer, is projected to have approximately 450,000 UAVs in its delivery fleet by 2020" [16].

The future perspectives for UAVs use are really optimistic, as each day more people gets aware of the possibilities that they offer. According to Airdronecraze, there are 7 drone generations, based on the technology used on them. Currently we are in the 6th generation, characterized for the following aspects:

- Commercial Suitability.
- Safety and Regulatory Standards Based Design.
- Platform and Payload Adaptability.
- Automated Safety Modes.
- Intelligent Piloting Models and Full Autonomy.
- Airspace Aware.

However, this technology is always being developed, and some of the higher–end professional grade UAVs are entering into the next UAV generation. Fully Compliant Safety, Platform and Payload Interchangeability, Auto Action (takeoff, land, mission execution) and Full Airspace Awareness are some of the new features that they will have [17].

1.3.1 Flying Wing UAVs

The aircraft to be studied in this project belongs to the flying wing UAVs' group. By definition, a flying wing aircraft is a fixed – wing aircraft with no tail assembly and no more lifting surfaces than its wings, where the aerodynamic control and stabilization in yaw, pitch and roll is incorporated.

Some examples of current flying wing UAVs are:

(a) Parrot Disco FPV. Taken from [18].

(b) RMRC E - Flite Opterra 2M. Taken from [19].

Figure 1.7: Some examples of current commercial flying wing UAVs.

The main advantages of using this type of drones are the reduction in weight and drag, what results in longer flight autonomy and better aerodynamic performance, when compared with the multi-copters UAVs.

Nevertheless, flying wing UAVs are quite unstable and difficult to control, given their lack of conventional stabilizing and control surfaces.

1.4 Aim of the dissertation

In this section we will explain which is the aim of this dissertation, the objectives of the project and what it is expected to achieve with it.

The goal that we want to reach is the accurate development of the automatic landing system for a generic flying wing UAV, studying and controlling the position of the aircraft during the landing phase with respect to the desired glide path trajectory.

We will approach this control problem making use of two different techniques, coming from a conventional and a modern background.

The conventional approach will be based on the PID controller, dating from the XIXth - XXth centuries, and considered to be among the first developed methods in control theory. Our objective will be the design of the pitch attitude control system by adjusting and giving values to the corresponding gains in the PID and feedback loops. We will simulate the effect of different atmospheric wind perturbations, so we get closer to the real flight conditions during the landing phase.

The modern approach will consist of the H infinity method (H_{inf}), or gamma attenuation problem. Developed in the 70's and 80's, this technique belongs to the robust controllers' family, and is considered to be one of the most widely used control methods nowadays. We will define the plant model for the approach to calculate the controller that we need for optimizing the performance specifications. Once obtained this, we will study its level of robustness.

In order to consider the greatest number of possible scenarios, we will specify two different glide slopes: 2.5° for the PID, which is the standard glide slope angle followed by the majority of aircrafts in the landing phase, and 12° for the H_{inf} method, a realistic possible glide slope angle trajectory for small size aircrafts such as the flying wing UAV considered in this project.

1.5 Structure of the dissertation

The structure of this project consists of one single part which will be called **Report**. This part will be divided in four big chapters, which are going to be divided in turn in those sections and subsections that was thought to be necessary during the development of the project along the year.

In this way, the divisions of the Report can be presented as follows:

- **Project approach** It includes the introductory information and presentation of the project. It is divided in:
  - Introduction. Explanation about the actuality of the project and justification of the same.
  - Development of the autolanding concept. History and development of the automatic landing system in aviation.
  - Considerations about the UAVs. History, development, and future perspectives of these pilotless aircrafts.
  - Aim of the dissertation.
  - Structure of the dissertation.
- **Theoretical background** Here we include certain theoretical concepts necessaries for a good understanding of the project. It is divided in:
  - Aircraft dynamics.
  - Non-linear equations of motion.
  - Linearization of the equations of motion.
  - Obtaining of the state equations.
- **Automatic Landing System**. In this chapter we will present the different control methods to be used in the developing of the autolanding system. It is divided in:
  - Dynamics model of the generic flying UAV. Definition of the longitudinal dynamics to be used in the development of the landing system.

  - Conventional architecture: PID. Explanation of the PID controller, design and implementation. Introduction of the wind perturbations.
  - Modern configuration: H infinity method. Explanation, design and implementation, robustness analysis.

- **Final conclusions**. We will present the conclusions reached after studying the different results obtained, and give some new ideas about improvements and possible future stages for the project. Therefore, it is divided in:
  - Conclusions, studying separately the conventional and modern approach, and them comparing both.
  - Future development.

Besides, three appendixes with the Matlab scripts created in the development of the thesis will be added, together with the bibliography used.

Chapter 2

Theoretical background

2.1 Aircraft dynamics

2.1.1 Previous considerations

Before developing the linear models of the aircraft dynamics (for copyright reasons we won't be able to use the nonlinear model, which would be more accurate and precise), we have to define how we are going to locate the aircraft in space, what we are going to measure in terms of dynamics, and how we are going to measure it, establishing the reference origin.

First of all, we are going to define two different set of axes: the Earth Axes, fixed to a point in the Earth surface, which we will use for determining the position of the aircraft in the air, and the Body Fixed Axes, fixed to the aircraft, that we are going to use to measure how disturbances modify the dynamics. Finally, a brief description will be added including the force and moment variables that will be considering for developing the aircraft dynamic model.

We will follow the references [20] and [21] in the development of the concepts in these theoretical background. The nomenclature to be used in the definition of the parameters will be the one developed by Hopkins (1970) as far as possible, since it is the mainly one used in [20].

Earth axes

Fist of all, we will use the convention for defining earth axes, determined by a set of orthogonal vectors $(o_0x_0y_0z_0)$ where o_0 is the origin of the set, o_0x_0 determines the direction pointing to the North, o_0y_0 determines the direction pointing to the East, and o_0z_0 determines the direction pointing down along the gravity vector.

The earth axes that we will use for flight is called datum–path earth axes, and is determined by $(o_Ex_Ey_Ez_E)$, parallel to $(o_0x_0y_0z_0)$ but with the difference that o_Ex_E will points towards the direction of flight of the aircraft, and that the origin of this set o_E is placed in the atmosphere so it is coincident with the origin of the aircraft body fixed axes. This set of axes will provide the inertial reference frame for short term aircraft motion.

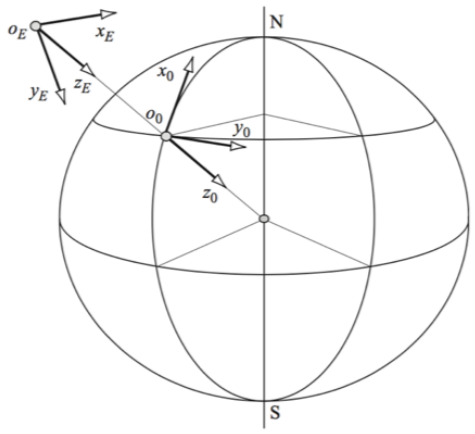


Figure 2.1: Convention for the Earth Axes. Taken from [20]

Aircraft Body Fixed axes

Generalized body axes The most general way of expressing those is through the set of axes ($ox_by_bz_b$), where ox_by_b defines the symmetry plane of the aircraft, ox_b towards the direction of flight, oy_b points towards the right wing of the plane, and oz_b downwards vertically. The origin o is usually arranged so it coincides with the center of gravity of the aircraft.

Aerodynamic, wind, or stability axes We call in this way to the set which ox axis is parallel to the total velocity vector V_0.

In the following picture we can appreciate the convention for both the generalized body axes and the wind axes:

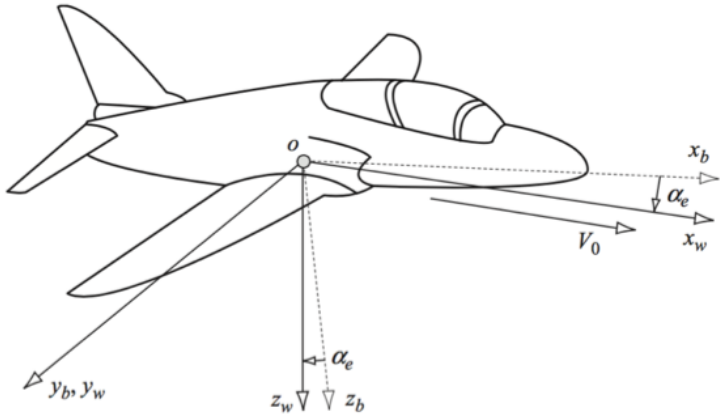


Figure 2.2: Moving axes system. Taken from [20]

Perturbation variables

When we describe the motion of an aircraft, we use forces, moments, linear and angular velocities and attitude angles resolved into components with respect to the chosen aircraft fixed axis system. Considering initially the flight to be rectilinear (although not necessarily level), steady and not accelerated, the sum of all forces and moments will be zero: this condition is called trimmed equilibrium. When the plane is disturbed

from equilibrium, the consequential motion is expressed in terms of perturbation variables:

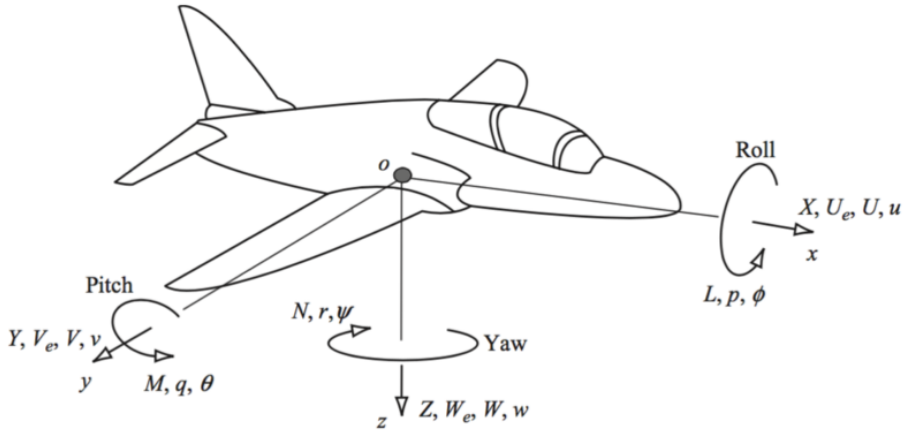


Figure 2.3: Motion variables notation. Taken from [20]

Summarizing those variables in one table:

Table 2.1: Summary of motion variables. Taken from [20]

Variable	Trimmed			Perturbed		
Aircraft axis	ox	oy	oz	ox	oy	oz
Force	0	0	0	X	Y	Z
Moment	0	0	0	L	M	N
Linear Velocity	U_e	V_e	W_e	U	V	W
Angular velocity	0	0	0	p	q	r
Attitude	0	θ_e	0	ϕ	θ	ψ

The sign convention for those variables is determined by a right handed axis system: a positive rotation around the x axis will be right wing down from the pilot point of view, a positive rotation around y axis will be nose up, and a positive rotation around z axis will be nose to the right from the pilot point of view, too. These rotations around the x, y and z axis are called attitude of the aircraft, and are designed by the name of roll (ϕ), pitch (θ), and yaw (ψ), respectively. They expressed the angular orientation of the aircraft body fixed axis with respect to the earth axes (the equivalent to the Euler angles).

2.2 Nonlinear equations of motion

We will assume that the aircraft is rigid, so that the distance between any point on the aircraft won't change in flight. We will use the body axis system previously considered, and it is assumed that the inertial frame of reference (the Earth) doesn't accelerate (keeps fixed in space).

Now, we will study the translation and rotation motion of our rigid aircraft; knowing the Second Law of Newton, we express:

$$F = \frac{d}{dt}(mV_T) \tag{2.1}$$

$$M = \frac{d}{dt}(H) \tag{2.2}$$

where F is the sum of all external applied forces, M is the sum of all applied torques, and H is the angular momentum.

As we are studying the effect of the perturbation on the aircraft dynamics, we will split the applied force and torque in both an equilibrium and perturbation component:

$$F = F_0 + \Delta F = \frac{d}{dt} m(V_T) \tag{2.3}$$

$$M = M_0 + \Delta M = \frac{d}{dt} m(H) \tag{2.4}$$

where F_0 and M_0 are the equilibrium components, and ΔF and ΔM are the component of the perturbation. Now, since we are using as inertial reference system the Earth axis system, we can re-express the former equations with respect this system as follows:

$$\Delta F = \frac{d}{dt} m(V_T)_E \tag{2.5}$$

$$\Delta M = \frac{d}{dt} (H)_E \tag{2.6}$$

Since equilibrium flight, by definition, has to be unaccelerated and straight along the path, the linear velocity vector relative to the fixed space is invariant (meaning that we will have 0 acceleration, so F_0 is 0), and the angular velocity is 0 (so M_0 is 0, too). Now, we will study both the translational and rotational motion of the aircraft independently.

2.2.1 Translational motion

If we develop the rate of change of relative to the Earth axis system, we have the following expression:

$$\begin{aligned} \frac{d}{dt}(V_T)_E = \frac{d}{dt}(V_T + \omega \times r_t) = \\ = \frac{d}{dt} V_T + \frac{d}{dt}(\omega \times r_t) = \\ = \frac{d}{dt} V_T + \frac{d}{dt} w \times r_T + \omega \times \frac{d}{dt} r_T = \\ = \frac{d}{dt} V_T + \omega \times V_T \end{aligned} \tag{2.7}$$

where ω is the angular velocity of the aircraft with respect to the body fixed system. Now, expanding the terms in expression (2.7):

$$V_T = Ui + Vj + Wk \tag{2.8}$$

$$\omega = Pi + Qj + Rk \tag{2.9}$$

$$\frac{d}{dt}V_T = \dot{U}i + \dot{V}j + \dot{W}k \tag{2.10}$$

$$\omega \times V_T = \begin{vmatrix} i & j & k \\ P & Q & R \\ U & V & W \end{vmatrix} = (QW - RV)i + (RRU - PW)j + (PV - UQ)k \tag{2.11}$$

Getting back to the Newton's equation, we can express in a similar way the force perturbation:

$$\Delta F = \Delta F_x i + \Delta F_y j + \Delta F_z k \tag{2.12}$$

And the equation of the force:

$$\Delta F = m\left\{(\dot{U} + QW - RV)i + (\dot{V} + RU - PW)j + (\dot{W} + PV - UQ)k\right\} \tag{2.13}$$

Equaling each corresponding term:

$$\Delta F_x = m\left(\dot{U} + QW - RY\right) \tag{2.14}$$

$$\Delta F_y = m\left(\dot{V} + RU - PW\right) \tag{2.15}$$

$$\Delta F_z = m\left(\dot{W} + PV - UQ\right) \tag{2.16}$$

However, it is proposed to change the notation of the perturbed force components, making it coincide with notation in table 2.1

$$\begin{aligned} \Delta F_x &\equiv \Delta X \\ \Delta F_y &\equiv \Delta Y \\ \Delta F_z &\equiv \Delta Z \end{aligned} \tag{2.17}$$

And so, we finally have our nonlinear equations for the inertial terms in the translational motion of our aircraft as follows:

$$\Delta X = m\left(\dot{U} + QW - RY\right) \tag{2.18}$$

$$\Delta Y = m\left(\dot{V} + RU - PW\right) \tag{2.19}$$

$$\Delta Z = m\left(\dot{W} + PV - UQ\right) \tag{2.20}$$

2.2.2 Rotational motion

Focusing now on the rotational motion of the rigid aircraft, we have to develop now the expression for the angular momentum, which is defined as:

$$H = I\omega \tag{2.21}$$

where I is the inertia matrix, defined as:

$$I = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{pmatrix} \tag{2.22}$$

I_{ii} denotes a moment of inertia and I_{ij} (for i $\neq$ j) denotes a product of inertia. However, because of aircrafts usually are symmetrical respect to the plane XZ, $I_{yx} = I_{zy} = I_{xy} = I_{yz} = 0$. Therefore, we can express the vector H as:

$$H = I\omega = \begin{pmatrix} PI_{xx} - QI_{xy} - RI_{xz} \\ -PI_{yx} + QI_{yy} - RI_{yz} \\ -PI_{zx} - QI_{zy} + RI_{zz} \end{pmatrix} = \begin{pmatrix} PI_{xx} - RI_{xz} \\ QI_{yy} \\ -PI_{zx} + RI_{zz} \end{pmatrix} = \begin{pmatrix} h_x \\ h_y \\ h_z \end{pmatrix} \tag{2.23}$$

Taken the equation (2.6), we can express the moment of the forces acting on our aircraft as follows:

$$\Delta M = \frac{d}{dt}H + \omega \times H \tag{2.24}$$

And, making use of the transformation expressions from body axes to the Earth axis system, we can re-express the above equation as:

$$\Delta M = I\left(\frac{d}{dt}\omega + \omega \times \omega\right) \tag{2.25}$$

Since the cross product of two parallel vectors is 0:

$$\omega \times \omega = 0 \tag{2.26}$$

The value for the rate of change with respect to time of the angular velocity is known to be, according to the notation that is being used:

$$\frac{d}{dt}\omega = \dot{P}i + \dot{Q}j + \dot{R}k \tag{2.27}$$

And, finally, the cross product of ω and H is:

$$\omega \times H = \begin{pmatrix} i & j & k \\ P & Q & R \\ h_x & h_y & h_z \end{pmatrix} \tag{2.28}$$

where h_x, h_y, and h_z are the components of the vector H, as given in the developed expression above.

Therefore, we can express the moment of the forces acting on the aircraft as:

$$\begin{aligned} \Delta M = I\left(\frac{d}{dt}\omega + \omega \times \omega\right) + \omega \times H = I\frac{d}{dt}\omega + \omega \times H = \\ = \left(I_{xx}\dot{P} + h_y R - h_z Q\right) i + \left(I_{yy}\dot{Q} + h_x R - h_z P\right) j + \left(I_{zz}\dot{R} + h_y P - h_x Q\right) k \end{aligned} \tag{2.29}$$

Equaling terms, in order to make the equality above clearer:

$$\Delta M_x = \left(I_{xx}\dot{P} + h_y R - h_z Q\right) \tag{2.30}$$

$$\Delta M_y = \left(I_{yy}\dot{Q} + h_x R - h_z P\right) \tag{2.31}$$

$$\Delta M_z = \left(I_{zz}\dot{R} + h_y P - h_x Q\right) \tag{2.32}$$

And changing to the American usage, as we did with the translation motion equation:

$$\Delta M_x = \Delta L \quad \Delta M_y = \Delta M \quad \Delta M_z = \Delta N \tag{2.33}$$

where L, M and N are the moments about the rolling, pitching and yawing axes, respectively.

Finally, we can express the equations for the inertial terms in the rotational motion of an aircraft as:

$$\Delta L = \left(I_{xx}\dot{P} + h_y R - h_z Q\right) \tag{2.34}$$

$$\Delta M = \left(I_{yy}\dot{Q} + h_x R - h_z P\right) \tag{2.35}$$

$$\Delta N = \left(I_{zz}\dot{R} + h_y P - h_x Q\right) \tag{2.36}$$

Which can be reduced and simplified as follows:

$$\begin{aligned} \Delta L &= \left(I_{xx}\dot{P} + h_y R - h_z Q\right) = \\ &= \left(I_{xx}\dot{P} - I_{zx}\dot{R}\right) - RQI_{yy} - Q\left(PI_{zx} - RI_{zz}\right) = \\ &= I_{xx}\dot{P} - I_{zx}\dot{P} - RQI_{yy} - QPI_{zx} + QRI_{zz} = \\ &= I_{xx}\dot{P} - I_{zx}\left(PQ + \dot{R}\right) + QR\left(I_{zz} - I_{yy}\right) \end{aligned} \tag{2.37}$$

The y axis term:

$$\begin{aligned} \Delta M &= \left(I_{yy}\dot{Q} + Rh_x - Ph_z\right) = \\ &= \left(-I_{yx}\dot{P} + I_{yy}\dot{R}\right) + \left(Rh_x - Ph_z\right) = \\ &= I_{yy}\dot{Q} + R\left(PI_{xx} - RI_{xz}\right) - P\left(PI_{zx} - RI_{zz}\right) = \\ &= I_{yy}\dot{Q} + PR\left(I_{xx} - I_{zz}\right) - I_{xz}\left(R^2 - P^2\right)\right) \end{aligned} \tag{2.38}$$

And the z axis term:

$$\begin{aligned} \Delta N &= \left(I_{zz}\dot{R} + Ph_y - Qh_x\right) = \\ &= \left(-I_{zx}\dot{P} + I_{zz}\dot{R}\right) + \left(Ph_y - Qh_x\right) = \\ &= I_{zz}\dot{R} - I_{zx}\dot{P} + PQI_{yy} - PQI_{xx} + QRI_{xz} = \\ &= I_{zz}\dot{R} + I_{xz}\left(QR - \dot{P}\right) + PQ\left(I_{yy} - I_{xx}\right) \end{aligned} \tag{2.39}$$

Finally, grouping expressions (2.37), (2.38) and (2.39):

$$\Delta L = I_{xx}\dot{P} - I_{zx}\left(PQ + \dot{R}\right) + QR\left(I_{zz} - I_{yy}\right) \tag{2.40}$$

$$\Delta M = I_{yy}\dot{Q} + PR\left(I_{xx} - Izz\right) - I_{xz}\left(R^2 - P^2\right) \tag{2.41}$$

$$\Delta N = I_{zz}\dot{R} + I_{xz}\left(QR - \dot{P}\right) + PQ\left(I_{yy} - I_{xx}\right) \tag{2.42}$$

2.2.3 Gravitational terms

In section 2.2.1 we only took into account the inertial terms acting on the aircraft. However, to increase the accuracy of our dynamic model we have to introduce the effect of gravity over the translation motion of the aircraft.

This is given by the expressions:

$$\delta X = mg \sin \Theta \tag{2.43}$$

$$\delta Y = -mg \cos \Theta \sin \Phi \tag{2.44}$$

$$\delta Z = -mg \cos \Theta \cos \Phi \tag{2.45}$$

where δX, δY, δZ are the components of the x, y and z axes of the gravitational force, g is the value of the gravity at the height which the aircraft is flying, and Φ and Θ are the Euler angles for the x and y body fixed axes, respectively.

2.2.4 Final expression for the nonlinear equations of the aircraft dynamics

Once we have obtained the gravitational and inertial terms for our equations of motion, we can define our aircraft dynamics model to be, for six degrees of freedom:

$$\begin{aligned} X &= \Delta X + \delta X \\ Y &= \Delta Y + \delta Y \\ Z &= \Delta Z + \delta Z \\ L &= \Delta L \\ M &= \Delta M \\ N &= \Delta N \end{aligned} \tag{2.46}$$

Expanding (2.46):

$$\begin{aligned} X \equiv ma_x &= m\left(\dot{U} + QW - RV + g \sin \Theta\right) \\ Y \equiv ma_y &= m\left(\dot{V} + RU - PW - g \cos \Theta \sin \Phi\right) \\ Z \equiv ma_z &= m\left(\dot{W} + PV - QU - g \cos \Theta \cos \Phi\right) \\ L &= I_{xx}\dot{P} - I_{zx}\left(PQ + \dot{R}\right) + QR\left(I_{zz} - I_{yy}\right) \\ M &= I_{yy}\dot{Q} + PR\left(I_{xx} - I_{zz}\right) - I_{xz}\left(R^2 - P^2\right)) \\ N &= I_{zz}\dot{R} + I_{xz}\left(QR - \dot{P}\right) + PQ\left(I_{yy} - I_{xx}\right) \end{aligned} \tag{2.47}$$

And we may use also the expression that relates the angular velocity components that represent the rotation of the body – fixed axes with the Earth axes. Those are given as a function of the Euler angles as follows:

$$\begin{aligned} P &= \dot{\Phi} - \dot{\Psi} \sin \Theta \\ Q &= \dot{\Theta} \cos \Phi + \dot{\Psi} \cos \Theta \sin \Phi \\ R &= -\dot{\Theta} \sin \Phi + \dot{\Psi} \cos \Theta \cos \Phi \end{aligned} \tag{2.48}$$

2.3 Linearization of the equations of motion

We have to keep in mind that our aim with all these equations is to establish an accurate model for the aircraft dynamics that can be used to control our aircraft. Therefore, we are interested in know how we can model the perturbations suffered by our aircraft from a reference point of equilibrium, i.e., how we can control the changes in attitude and altitude of the aircraft to make it fly in the desired conditions imposed by its controller.

Now that we have defined the dynamic model of our aircraft, we just need to decompose the variables obtained in two terms, the equilibrium and the perturbation term, and to re-express equatio (2.47) to represent the aircraft dynamics of the perturbations.

We will decompose each variable as follows:

$$\begin{aligned} U &\equiv U_0 + u \\ R &\equiv R_0 + r \\ M &\equiv M_0 + m \\ &etc \end{aligned} \tag{2.49}$$

where the subindex 0 indicates the trim or equilibrium term, and the lower case letter denotes the perturbation term, assumed to be much smaller.

Now we will determine the expressions for the perturbation terms. The procedure that is going to be followed is simply to expand each term in equation (2.47) and obtain the equivalent expression of the perturbation variable. It is going to be exemplified with force X:

$$X = m\left(\dot{U} + QW - RV + g\sin\Theta\right) \tag{2.50}$$

Expanding each variable:

$$\begin{aligned} X = X_0 + x = \\ = m\left(\dot{U_0} + \dot{u} + (Q_0 + q)(W_0 + w) - (R_0 + r)(V_0 + v) + g\sin(\Theta_0 + \theta)\right) \end{aligned} \tag{2.51}$$

We will simplify equation (2.51) considering that the products of the perturbation terms are negligible, that the cosine can be approximated to 1 and the sine to the values of the angle itself, and that in trim conditions there can be no acceleration:

$$\begin{aligned} X = X_0 + x = \\ = m\left(\dot{u} + Q_0W_0 + qW_0 + wQ_0 - R_0V_0 - rV_0 - R_0v + g\sin\Theta_0\right) \end{aligned} \tag{2.52}$$

In equation (2.52) the equilibrium component of force X can be easily identified to be:

$$X_0 = m\left(Q_0 W_0 - R_0 V_0 + g\sin\Theta_0\right) \tag{2.53}$$

Therefore, it can be re-expressed as:

$$X = X_0 + m\left(\dot{u} + qW_0 + wQ_0 - rV_0 - R_0 v + g\cos\Theta_0\theta\right) \tag{2.54}$$

And from here to the perturbation term of the force:

$$X - X_0 = x = dX = m\left(\dot{u} + qW_0 + wQ_0 - rV_0 - R_0 v + g\cos\Theta_0\theta\right) \tag{2.55}$$

If we apply these steps to the rest of the force and moment terms, we will obtain the expressions represented hereunder.

First of all, regarding the translation motion:

$$x = dX = m\left(\dot{u} + qW_0 + wQ_0 - rV_0 - R_0 v + g\cos\Theta_0\theta\right) \tag{2.56}$$

$$y = dY = m\left(\dot{v} + U_0 r + R_0 u - W_0 p - P_0 w - \left(g\cos\Theta_0\cos\Phi_0\right)\phi + \left(g\cos\Theta_0\sin\Phi_0\right)\theta\right) \tag{2.57}$$

$$z = dZ = m\left(\dot{w} + V_0 p + P_0 v - U_0 q - Q_0 u - \left(g\cos\Theta_0\sin\Phi_0\right)\phi + \left(g\sin\Theta_0\cos\Phi_0\right)\theta\right) \tag{2.58}$$

Secondly, regarding rotational motion:

$$l = dL = I_{xx}\dot{p} - I_{zx}\dot{r} - I_{zx}\left(P_0 q + Q_0 p\right) + \left(I_{zz} - I_{yy}\right)\left(P_0 r + R_0 p\right) \tag{2.59}$$

$$m = dM = I_{yy}\dot{q} - I_{xz}\left(2R_0 r - 2P_0 p\right) + \left(I_{xx} - I_{zz}\right)\left(P_0 r + R_0 p\right) \tag{2.60}$$

$$n = dN = I_{zz}\dot{r} - i_{xz}\dot{p} + \left(I_{yy} - I_{xx}\right)\left(P_0 q + Q_0 p\right) + I_{xz}\left(Q_0 r + R_0 q\right) \tag{2.61}$$

In this way, we have obtained the linear dynamic equations for six degrees of freedom, which can be used for modeling and controlling a given aircraft.

However, we may also take into account expression (2.48), which shows the relationship between the angular velocities with the Euler angles. Applying the same procedure, we reach the following expressions:

$$p = \dot{\psi} - \dot{\Gamma}\sin\Theta_0 - \theta\left(\dot{\Gamma}\cos\Theta_0\right) \tag{2.62}$$

$$q = \dot{\Theta}\cos\Phi_0 - \theta\left(\dot{\Psi}\sin\Phi_0\sin\Phi_0\right) + \dot{\Psi}\sin\Psi_0\cos\Theta + \phi\left(\dot{\Psi}_0\cos\Phi_0\cos\Theta_0 - \dot{\Theta}_0\sin\Phi_0\right) \tag{2.63}$$

$$r = \dot{\Psi}\cos\Theta_0\cos\Phi_0 - \phi\left(\dot{\Psi}_0\sin\Phi_0\cos\Theta_0 + \dot{\Psi}_0\cos\Phi_0\right) - \dot{\Theta}_0\sin\Phi_0 - \theta\left(\dot{\Psi}_0\cos\Phi_0\right) \tag{2.64}$$

And we may take into account the following conditions, that will help us simplify these long equations:

- Straight flight implies $\dot{\Psi}_0 = \dot{\Theta}_0 = 0$.
- Symmetric flight implies $\Psi_0 = V_0 = 0$.
- Flying with wing levels (zero bank angle) implies $\Phi_0 = 0$.

2.4 Simplifications of the linear equations of motion

We will consider a straight, symmetric, and wings level flight in order to simplify the equations and make it easier for us to calculate and operate with the linear equations of motion obtained in section 2.3. Besides, these conditions will be really important for us, since they determine the trim flight conditions of the aircraft, the equilibrium position during flight that we want to maintain.

Therefore, during trim, we have that:

- $\Phi_0 = 0$.
- $\Psi_0 = V_0 = 0$.
- $\dot{\Phi}_0 = \dot{\Theta}_0 = 0$.
- $P_0 = R_0 = Q_0 = 0$.

Also, we can rewrite and group equations (2.56), (2.57), (2.58), (2.59), (2.60), (2.61), (2.62), (2.63) and (2.64) as the following expressions:

$$\begin{aligned} x = dX &= m\left(\dot{u} + W_0 q + g\cos\Theta_0\theta\right) \\ y = dy &= m\left(\dot{v} + U_0 q - W_0 p - g\cos\Theta_0\Phi\right) \\ z = dz &= m\left(\dot{w} - U_0 q + g\sin\Theta_0\theta\right) \end{aligned} \tag{2.65}$$

Then:

$$\begin{aligned} l = dL &= I_{xx}\dot{p} - I_{zx}\dot{r} \\ m = dM &= I_{yy}\dot{q} \\ n = dN &= I_{zz}\dot{r} - I_{xz}\dot{p} \end{aligned} \tag{2.66}$$

And, finally:

$$\begin{aligned} p &= \dot{\phi} - \dot{\Psi}\sin\Theta_0 \\ q &= \dot{\Theta} \\ r &= \dot{\Psi}\cos\Theta_0 \end{aligned} \tag{2.67}$$

From this simplification, we can observe that there are some equations with similar structures; indeed, we can separate expressions (2.65) and (2.66) in two different groups, according to these similarities:

$$\begin{aligned} x = dX &= m\left(\dot{u} + W_0 q + g\cos\Theta_0\theta\right) \\ z = dz &= m\left(\dot{w} - U_0 q + g\sin\Theta_0\theta\right) \\ m = dM &= I_{yy}\dot{q} \end{aligned} \tag{2.68}$$

And:

$$\begin{aligned} y = dy &= m\left(\dot{v} + U_0 q - W_0 p - g\cos\Theta_0\Phi\right) \\ l = dL &= I_{xx}\dot{p} - I_{zx}\dot{r} \\ n = dN &= I_{zz}\dot{r} - I_{xz}\dot{p} \end{aligned} \tag{2.69}$$

Independent variables in equations (2.68) are u, w, q, and θ, confined to the plane $X_B Z_B$, so they are said to represent the longitudinal motion of the aircraft, while independent variables in equations (2.69) are v, p, ϕ and r, which represents the sideslip, rolling and yawing movement, so they are said to represent the lateral motion of the aircraft.

We have to keep in mind that this separation is done only due to the trim conditions assumed, which had the aim of simplify the equations of motion obtained.

2.5 Obtaining of the state equations

2.5.1 Expansion of the linear equations

Longitudinal motion

To do this expansion, we will use a Taylor series about the left-hand side of the equations of motion. For example, taken the perturbed force in the z axis:

$$z = \frac{\partial Z}{\partial u}u + \frac{\partial Z}{\partial \dot{u}}\dot{u} + \frac{\partial Z}{\partial w}w + \frac{\partial Z}{\partial \dot{w}}\dot{w} + \frac{\partial Z}{\partial q}q + \frac{\partial Z}{\partial \dot{q}}\dot{q} + \frac{\partial Z}{\partial \delta_E}\delta_E + \frac{\partial Z}{\partial \dot{\delta_E}}\dot{\delta_E} + ... \tag{2.70}$$

The expansion is done with all the independent variables that we find in the expression for the longitudinal motion, taken into account their derivatives too. We omitted the Euler angles because their contribution is generally insignificant. The final three points indicate that we can add all those actuators that we consider that may affect to the perturbed force z. In this case, we will take only the elevator, but we could add too the symmetrical spoilers, flaps, or changes in thrust.

So, we can re-express the expression (2.68) as:

$$\begin{aligned}
\frac{\partial X}{\partial u}u + \frac{\partial X}{\partial \dot{u}}\dot{u} + \frac{\partial X}{\partial w}w + \frac{\partial X}{\partial \dot{w}}\dot{w} + \frac{\partial X}{\partial q}q + \frac{\partial X}{\partial \dot{q}}\dot{q} + \frac{\partial X}{\partial \delta_E}\delta_E + \frac{\partial X}{\partial \dot{\delta}_E}\dot{\delta_E} = \\
= m\left(\dot{u} + W_0 q + g\cos\Theta_0\theta\right) \\
\frac{\partial Z}{\partial u}u + \frac{\partial Z}{\partial \dot{u}}\dot{u} + \frac{\partial Z}{\partial w}w + \frac{\partial Z}{\partial \dot{w}}\dot{w} + \frac{\partial Z}{\partial q}q + \frac{\partial Z}{\partial \dot{q}}\dot{q} + \frac{\partial Z}{\partial \delta_E}\delta_E + \frac{\partial Z}{\partial \dot{\delta}_E}\dot{\delta_E} = \\
= m\left(\dot{w} - U_0 q + g\sin\Theta_0\theta\right) \\
\frac{\partial M}{\partial u}u + \frac{\partial M}{\partial \dot{u}}\dot{u} + \frac{\partial M}{\partial w}w + \frac{\partial M}{\partial \dot{w}}\dot{w} + \frac{\partial M}{\partial q}q + \frac{\partial M}{\partial \dot{q}}\dot{q} + \frac{\partial M}{\partial \delta_E}\delta_E + \frac{\partial M}{\partial \dot{\delta}_E}\dot{\delta}_E = \\
I_{yy}\dot{q}
\end{aligned} \tag{2.71}$$

For simplification of the above expression, we will make use of the following notation:

$$\frac{1}{m}\frac{\partial X}{\partial j} = X_j \qquad \frac{1}{m}\frac{\partial Z}{\partial \dot{j}} = Z_{\dot{j}} \qquad \frac{1}{I_{yy}}\frac{\partial M}{\partial j} = M_j \tag{2.72}$$

where $X_u, Z_{\dot{w}}, M_q$ are called stability derivatives.

So, applying the stability derivatives notation to the expression (2.71):

$$\begin{aligned}
X_u u + X_{\dot{u}}\dot{u} + X_w w + X_{\dot{w}}\dot{w} + X_q q + X_{\dot{q}}\dot{q} + X_{\delta_E}\delta_E + X_{\dot{\delta}_E}\dot{\delta_E} = \\
= m\left(\dot{u} + W_0 q + g\cos\Theta_0\theta\right) \\
Z_u u + Z_{\dot{u}}\dot{u} + Z_w w + Z_{\dot{w}}\dot{w} + Z_q q + Z_{\dot{q}}\dot{q} + Z_{\delta_E}\delta_E + Z_{\dot{\delta}_E}\dot{\delta_E} = \\
= m\left(\dot{w} - U_0 q + g\sin\Theta_0\theta\right) \\
M_u u + M_{\dot{u}}\dot{u} + M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\dot{q}}\dot{q} + M_{\delta_E}\delta_E + M_{\dot{\delta}_E}\dot{\delta_E} = \\
I_{yy}\dot{q}
\end{aligned} \tag{2.73}$$

And the Euler angles' equation is usually added, that introduces the relationship among the angular velocity term in expression (2.73) and the Euler angles $q = \dot{\Theta}$.

Lateral motion

If we apply the same procedure, we obtain the following expressions, keeping in mind that this time the independent variables are *v, r* and *p*, and that the actuators that will contribute the most will be the rudder and the aileron, expressed as δ_T and δ_A , respectively.

$$\begin{aligned}
\frac{\partial Y}{\partial v}v + \frac{\partial Y}{\partial \dot{v}}\dot{v} + \frac{\partial Y}{\partial r}r + \frac{\partial Y}{\partial \dot{r}}\dot{r} + \frac{\partial Y}{\partial p}p + \frac{\partial Y}{\partial \dot{p}}\dot{p} + \frac{\partial Y}{\partial \delta_R}\delta_R + \frac{\partial Y}{\partial \dot{\delta}_R}\dot{\delta}_R + \frac{\partial Y}{\partial \delta_A}\delta_A + \frac{\partial Y}{\partial \dot{\delta}_A}\dot{\delta}_A = \\
= m\left(\dot{v} + U_0 q - W_0 p - g\cos\Theta_0\Phi\right) \\
\frac{\partial L}{\partial v}v + \frac{\partial L}{\partial \dot{v}}\dot{v} + \frac{\partial L}{\partial r}r + \frac{\partial L}{\partial \dot{r}}\dot{r} + \frac{\partial L}{\partial p}p + \frac{\partial L}{\partial \dot{p}}\dot{p} + \frac{\partial L}{\partial \delta_R}\delta_R + \frac{\partial L}{\partial \dot{\delta}_R}\dot{\delta}_R + \frac{\partial L}{\partial \delta_A}\delta_A + \frac{\partial L}{\partial \dot{\delta}_A}\dot{\delta}_A = \\
= I_{xx}\dot{p} - I_{zx}\dot{r} \\
\frac{\partial N}{\partial v}v + \frac{\partial N}{\partial \dot{v}}\dot{v} + \frac{\partial N}{\partial r}r + \frac{\partial N}{\partial \dot{r}}\dot{r} + \frac{\partial N}{\partial p}p + \frac{\partial N}{\partial \dot{p}}\dot{p} + \frac{\partial N}{\partial \delta_R}\delta_R + \frac{\partial N}{\partial \dot{\delta}_R}\dot{\delta}_R + \frac{\partial N}{\partial \delta_A}\delta_A + \frac{\partial N}{\partial \dot{\delta}_A}\dot{\delta}_A = \\
= I_z z\dot{r} - I_{xz}\dot{p}
\end{aligned} \tag{2.74}$$

And obtaining the stability derivatives such that:

$$\frac{1}{m}\frac{\partial Y}{\partial j} = Y_j \qquad \frac{1}{I_{xx}}\frac{\partial L}{\partial \dot{j}} = L_j \qquad \frac{1}{I_{zz}}\frac{\partial N}{\partial j} = N_j \tag{2.75}$$

We can obtain:

$$\begin{aligned}
Y_v v + Y_{\dot{v}}\dot{v} + Y_r r + Y_{\dot{r}}\dot{r} + Y_p p + Y_{\dot{p}}\dot{p} + Y_{\delta_R}\delta_R + Y_{\dot{\delta}_R}\dot{\delta_R} + Y_{\delta_A}\delta_A + Y_{\dot{\delta}_A}\dot{\delta}_A = \\
= m\left(\dot{v} + U_0 q - W_0 p - g\cos\Theta_0\Phi\right) \\
L_v v + L_{\dot{v}}\dot{v} + L_r r + L_{\dot{r}}\dot{r} + L_p p + L_{\dot{p}}\dot{p} + L_{\delta_R}\delta_R + L_{\dot{\delta}_R}\dot{\delta_R} + L_{\delta_A}\delta_A + L_{\dot{\delta}_A}\dot{\delta}_A = \\
= I_{xx}\dot{p} - I_{zx}\dot{r} \\
N_v v + N_{\dot{v}}\dot{v} + N_r r + N_{\dot{r}}\dot{r} + N_p p + N_{\dot{p}}\dot{p} + N_{\delta_R}\delta_R + N_{\dot{\delta}_R}\dot{\delta_R} + N_{\delta_A}\delta_A + N_{\dot{\delta}_A}\dot{\delta}_A = \\
= I_z z\dot{r} - I_{xz}\dot{p}
\end{aligned} \tag{2.76}$$

And here again it is introduced too the Euler angles' equation for the angular velocity terms in (2.76): $p = \dot{\phi} - \dot{\Psi}\sin\Theta_0$ and $r = \cos\Theta_0$.

2.5.2 Equations of the longitudinal motion

In expression (2.73), we may simplify the equations without loss of generality by considering that $X_{\dot{u}}, X_{\dot{w}}, X_q, X_{\dot{q}}$, $X_{\dot{\delta}_E}, Z_{\dot{u}}, Z_{\dot{w}}, Z_q, Z_{\dot{q}}, Z_{\dot{\delta}_E}, M_{\dot{u}}, M_{\dot{q}}, M_{\dot{\delta}_E}$ are often insignificant, and so they may be ignored.

Therefore, we have the final expression for the longitudinal motion of the aircraft for straight, symmetric and wings level flight is:

$$\begin{aligned} \dot{u} &= X_u u + X_w w - W_0 q - g\cos\Theta_0\theta + X_{\delta_E}\delta_E \\ \dot{w} &= Z_u u + Z_w w + U_0 q - g\sin\Theta_0\theta + Z_{\delta_E}\delta_E \\ \dot{q} &= M_u u + M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\delta_E}\delta_E \\ \dot{\Theta} &= q \end{aligned} \tag{2.77}$$

In order to eliminate the dotted term in the expression, since they are not admitted in the state equation, we substitute for $\dot{w}$ and we obtain:

$$\dot{q} = M_u u + M_w w + M_{\dot{w}} Z_u u + Z_w w + U_0 q - g\sin\Theta_0\theta + Z_{\delta_E}\delta_E + M_q q + M_{\delta_E}\delta_E \tag{2.78}$$

Grouping terms, we have:

$$\dot{q} = \left(M_u + M_{\dot{w}} Z_u\right) u + \left(M_w + M_{\dot{w}} Z_w\right) w + \left(M_q + M_{\dot{w}} U_0\right) q - g\sin\Theta_0\theta + \left(Z_{\delta_E} + M_{\dot{w}} M_{\delta_E}\right)\delta_E \tag{2.79}$$

Calling:

$$\begin{aligned} \tilde{M_u} &= M_u + M_{\dot{w}} Z_u \\ \tilde{M_w} &= M_w + M_{\dot{w}} Z_w \\ \tilde{M_q} &= M_q + M_{\dot{w}} U_0 \\ \tilde{M_{\delta_E}} &= Z_{\delta_E} + M_{\dot{w}} M_{\delta_E} \end{aligned} \tag{2.80}$$

And substituting in (2.79), we will have this expression in an appropriate form for building the state equation in future steps:

$$\dot{q} = \tilde{M_u} u + \tilde{M_w} w + \tilde{M_q} q - g\sin\Theta_0\theta + \tilde{M_{\delta_E}}\delta_E \tag{2.81}$$

2.5.3 Equations of lateral motion

In this case, the terms to be neglected are $Y_{\dot{v}}, Y_p, Y_{\dot{w}}, Y_r, Y_{\dot{r}}, Y_{\dot{\delta}_A}, Y_{\delta_A}, N_{\dot{\delta}_R}, L_{\dot{v}}, L_{\dot{r}}, l_{\dot{p}}, L_{\dot{\delta}_A}, L_{d\dot{e}lta_R}, N_{\dot{v}}, N_{\dot{r}}, N_{\dot{p}}$, $N_{\dot{\delta}_A}, N_{\dot{\delta}_R}$ and so we can obtain the equations for the lateral motion of the aircraft for straight, symmetric and wings level flight:

$$\begin{aligned}
\dot{v} &= Y_v v - U_0 r + W_0 p + g\cos\Theta_0\Phi + Y_{\delta_R}\delta_R \\
\dot{p} &= L_v v + L_r r + L_p p + \frac{I_{zx}}{I_{xx}}\dot{r} + L_{\delta_R}\delta_R + L_{\delta_A}\delta_A \\
\dot{r} &= N_v v + N_r r + N_p p + \frac{I_{xz}}{I_{zz}}\dot{p} + N_{\delta_R}\delta_R + N_{\delta_A}\delta_A \\
\dot{\phi} &= p + \frac{\sin\Theta_0}{\cos\Theta_0} r = p + r\tan\Theta_0 \\
\dot{\Psi} &= \frac{r}{\cos\Theta_0}
\end{aligned} \tag{2.82}$$

In order to eliminate the dotted terms in the angular velocity terms expressions, we will design:

$$\begin{aligned}
L_v^{'} &= L_v + I_B N_v \\
L_p^{'} &= L_p + I_B N_p \\
L_r^{'} &= L_r + I_B N_r \\
L_{\delta_R}^{'} &= L_{\delta_R} + I_B N_{\delta_R} \\
L_{\delta_A}^{'} &= L_{\delta_A} + I_B N_{\delta_A} \\
N_v^{'} &= N_v + I_A L_v \\
N_p^{'} &= N_p + I_A L_p \\
N_r^{'} &= N_r + I_A L_r \\
N_{\delta_R}^{'} &= N_{\delta_R} + I_A L_{\delta_R} \\
N_{\delta_A}^{'} &= N_{\delta_A} + I_A L_{\delta_A}
\end{aligned} \tag{2.83}$$

where

$$I_B = \frac{I_{zx}}{I_{xx}} \qquad I_A = \frac{I_{xz}}{I_{zz}} \tag{2.84}$$

And in this way expression (2.82) can be re-expressed as:

$$\begin{aligned}
\dot{v} &= Y_v v - U_0 r + W_0 p + g\cos\Theta_0\Phi + Y_{\delta_R}\delta_R \\
\dot{p} &= L_v^{'} v + L_r^{'} r + L_p^{'} p + L_{\delta_R}^{'}\delta R + L_{\delta_A}^{'}\delta_A \\
\dot{r} &= N_v^{'} v + N_r^{'} r + N_p^{'} p + N_{\delta_R}^{'}\delta R + N_{\delta_A}^{'}\delta_A \\
\dot{\phi} &= p + r\tan\Theta_0 \\
\dot{\Psi} &= \frac{r}{\cos\Theta_0}
\end{aligned} \tag{2.85}$$

2.5.4 Stability axis system

If the axis system is oriented such that W_0 is 0, we would orientate the X_B axis (from the body–fixed axes system) pointing into the relative wind, so the velocity vector would be aligned such that U_0 would be V_T. This orientation is called stability axis system, where the relationship:

$$\Theta_0 = \gamma_0 + \alpha_0 \tag{2.86}$$

can be found.

All perturbations are still measured in a body–fixed frame of reference. However, the alignment of the stability axis system with the body–fixed axis changes depending on the alteration of the trim conditions. When the aircraft is disturbed, the stability axes rotate with the airframe, and so, the X_S axis may or may not be parallel to the relative wind while the aircraft is being disturbed, as represented in the following figure:

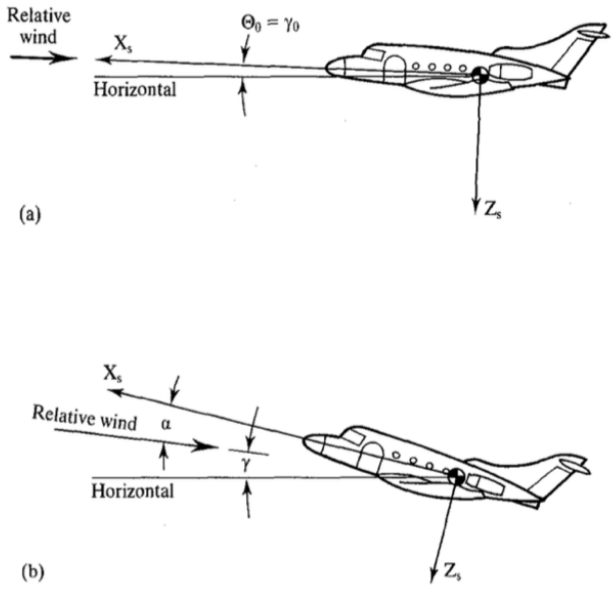


Figure 2.4: Direction of stability axes with respect to the relative wind in a) Steady flight and b) Perturbed flight. Taken from [21]

2.5.5 State and output equations for the longitudinal motion

We know that the state equation is expressed to be:

$$\dot{x} = Ax + Bu \tag{2.87}$$

where x is the state vector, u the control vector, A the (n × n) state matrix, and B the (n × m) input matrix.

Changing expression (2.77) to the stability axis system, we have that:

$$\begin{aligned} \dot{u} &= X_u u + X_w w - W_0 q - g\cos\gamma_0\theta + X_{\delta_E}\delta_E \\ \dot{w} &= Z_u u + Z_w w + U_0 q - g\sin\gamma_0\theta + Z_{\delta_E}\delta_E \\ \dot{q} &= \tilde{M}_u u + \tilde{M}_w w + \tilde{M}_q q - g\sin\Theta_0\theta + \tilde{M}_{\delta_E}\delta_E \\ \dot{\theta} &= q \end{aligned} \tag{2.88}$$

And we can define vector x, $\dot{x}$, and u to be:

$$x = \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} \qquad \dot{x} = \begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} \qquad u = \delta_E \tag{2.89}$$

And therefore the matrices A and B can be easily obtained from expression (2.88):

$$A = \begin{pmatrix} X_u & X_w & 0 & -g\cos\gamma_0 \\ Z_u & Z_w & U_0 & -g\sin\gamma_0 \\ \tilde{M}_u & \tilde{M}_w & \tilde{M}_q & -gM_{\dot{w}}\sin\gamma_0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad B = \begin{bmatrix} 0 \\ Z_{\delta_E} \\ \tilde{M}_{\delta_E} \\ 0 \end{bmatrix} \tag{2.90}$$

Finally, we can express the state equation for the longitudinal motion of an aircraft in steady, symmetric and wings level flight as:

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{pmatrix} X_u & X_w & 0 & -g\cos\gamma_0 \\ Z_u & Z_w & U_0 & -g\sin\gamma_0 \\ \tilde{M}_u & \tilde{M}_w & \tilde{M}_q & -gM_{\dot{w}}\sin\gamma_0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ Z_{\delta_E} \\ \tilde{M}_{\delta_E} \\ 0 \end{bmatrix} \delta_E \tag{2.91}$$

Regarding the output equation, it has to be of the form:

$$y = Cx + Du \tag{2.92}$$

where y is the output vector, C is the (r × n) output matrix and D is the (r × m) direct matrix. Since we will apply the aircraft dynamic equations to a control problem, we are interested in that the output vector is the same than the state vector, i. e.:

$$y = Cx \tag{2.93}$$

Due to this, this expression is quite easy to determine:

$$y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} \tag{2.94}$$

Although the matrix C will change depending on the output variables of the problem in which we are interested.

2.5.6 State and output equations for the lateral motion

First of all, we express equation (2.85) with respect the stability axis system:

$$\begin{aligned} \dot{v} &= Y_v v - U_0 r + W_0 p + g\cos\gamma_0 \Phi + Y_{\delta_R}\delta_R \\ \dot{p} &= L_v^{'} v + L_r^{'} r + L_p^{'} p + L_{\delta_R}^{'}\delta_R + L_{\delta_A}^{'}\delta_A \\ \dot{r} &= N_v^{'} v + N_r^{'} r + N_p^{'} p + N_{\delta_R}^{'}\delta_R + N_{\delta_A}^{'}\delta_A \\ \dot{\phi} &= p + r\tan\gamma_0 \\ \dot{\Psi} &= \frac{r}{\cos\gamma_0} \end{aligned} \tag{2.95}$$

This time the vectors are:

$$x = \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} \qquad \dot{x} = \begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} \qquad u = \begin{bmatrix} \delta_R \\ \delta_A \end{bmatrix} \tag{2.96}$$

And the matrices A and B:

$$A = \begin{pmatrix} Y_v & 0 & -U_0 & g\cos\gamma_0 & 0 \\ L_v^{'} & L_p^{'} & L_r^{'} & 0 & 0 \\ N_v^{'} & N_p^{'} & N_r^{'} & 0 & 0 \\ L_v^{'} & L_p & L_r^{'} & 0 & 0 \\ 0 & 1 & \tan\gamma_0 & 0 & 0 \\ 0 & 0 & \sec\gamma_0 & 0 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} Y_{\delta_R} & Y_{\delta_A} \\ L_{\delta_R}^{'} & L_{\delta_A}^{'} \\ N_{\delta_R}^{'} & N_{\delta_A}^{'} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{2.97}$$

And so, the state equation for the lateral motion of an aircraft, for symmetric, steady and wings level flight, is:

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{pmatrix} Y_v & 0 & -U_0 & g\cos\gamma_0 & 0 \\ L_v^{'} & L_p^{'} & L_r^{'} & 0 & 0 \\ N_v^{'} & N_p^{'} & N_r^{'} & 0 & 0 \\ L_v^{'} & L_p^{'} & L_r^{'} & 0 & 0 \\ 0 & 1 & \tan\gamma_0 & 0 & 0 \\ 0 & 0 & \sec\gamma_0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v \\ p \\ r \\ \phi \\ \psi \end{pmatrix} + \begin{pmatrix} Y_{\delta_R} & Y_{\delta_A} \\ L_{\delta_R}^{'} & L_{\delta_A}^{'} \\ N_{\delta_R}^{'} & N_{\delta_A}^{'} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_R \\ \delta_A \end{pmatrix} \tag{2.98}$$

As for the output equation, following the same reasoning than for the longitudinal motion, we will have:

$$y = Cx = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} \tag{2.99}$$

2.5.7 Final considerations

Now that we have properly determined the dynamics model of the aircraft, we will have to determine whether the state vector is appropriate to accurately monitoring the aircraft operation that we want to control.

As we are interested in being able to measure and control not only the horizontal displacement but the vertical motion too, since we will design the automatic landing system for the aircraft, we will modify the vector to control (2.89) to include the angle of attack. This can be easily obtained:

$$\tan\alpha = \frac{w}{U_0} \tag{2.100}$$

which can be simplified, for small angles, to:

$$\alpha = \frac{w}{U_0} \tag{2.101}$$

And so, from the heave velocity we obtain the angle of attack state expression:

$$\dot{\alpha} = \frac{\dot{w}}{U_0} = \frac{Z_u}{U_0}u + Z_w\frac{w}{U_0} + q - \frac{g\sin\gamma_0}{U_0}\theta + \frac{Z_{\delta_E}}{U_0}\delta_E \tag{2.102}$$

Changing the notation:

$$\dot{\alpha} = Z_u^* u + Z_\alpha \alpha + q - \frac{g\sin\gamma_0}{U_0}\theta + Z_{\delta_E}^*\delta_E \tag{2.103}$$

Therefore, we can re-express the state equation of the longitudinal motion of the aircraft for the landing phase obtained in equation (2.91) as:

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{pmatrix} X_u & U_0 X_\alpha & 0 & -g\cos\gamma_0 \\ Z_u^* & Z_\alpha & 1 & \frac{g\sin\gamma_0}{U_0} \\ \tilde{M}_u & U_0\tilde{M}_\alpha & \tilde{M}_q & -gM_{\dot{w}}\sin\gamma_0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ Z_{\delta_E}^* \\ \tilde{M}_{\delta_E} \\ 0 \end{bmatrix} \delta_E \tag{2.104}$$

We will suppose that the alignment with the runway was already done in prior phases of the flight.

Chapter 3

Automatic Landing System

In this chapter, it will be developed the automatic landing system of the UAV. Once we have the dynamics of the aircraft to be modeled, we will have to establish which approach we will consider to control the flight of the drone during the landing phase. We will consider two different techniques: one based on the conventional control theory architectures and another based on the newer advances.

3.1 Dynamics model of the generic flying wing UAV

For this work we will make use of the following longitudinal dynamics model, corresponding to a generic flying wing UAV in nominal conditions of $U_0 = 15m/s$ and $h_0 = 10m$. This was taken from [22]:

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{pmatrix} -1.3374e-02 & 1.3589e+00 & -9.0703e+00 & 0 \\ 1.2592e-01 & -1.1151e+01 & -1.4134e+00 & -5.0437e+01 \\ 0 & 0 & 0 & 1 \\ -7.2028e+00 & 1.3765e+03 & 2.6574e-01 & -3.5594e+01 \end{pmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} -6.4712e-03 \\ 3.6575e-02 \\ 0 \\ -7.7492e+00 \end{bmatrix} \delta_E \tag{3.1}$$

3.2 Conventional architecture: PID

3.2.1 Description of the method

Our first configuration will be based on the PID controller, a control loop feedback mechanism which consists of the application of proportional, integral, and derivative gains to achieve the desired performance and to meet the design requirements, whose theory was developed along the first decades of the XX^{th} century.

We will follow the glide - path - coupled control system found in McLean. He makes use of a pitch attitude control system and a pitch rate SAS inner loop to control the changes in the angle of attack. Those changes may arise when using the elevator to drive the aircraft onto the desired glide path.

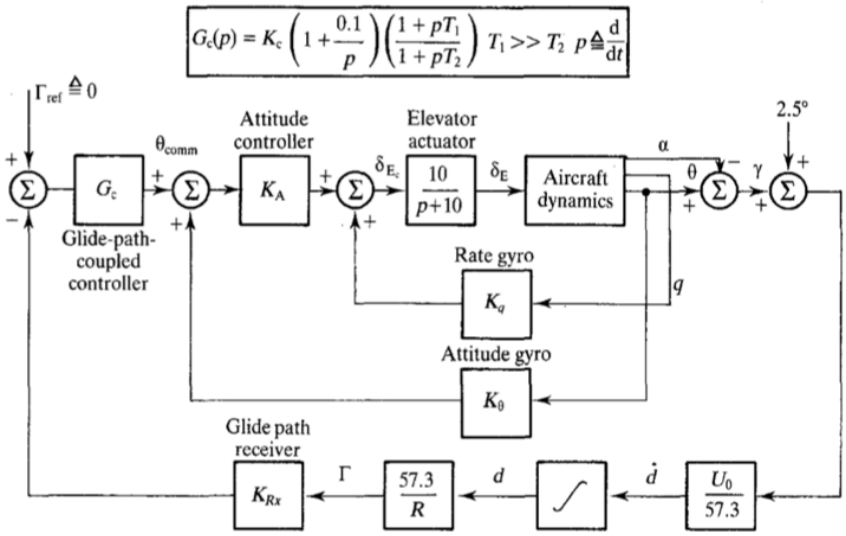


Figure 3.1: McLean's control loop for the landing phase of the aircraft. Taken from [21]

For our case, the objective of the control system depicted in figure 3.1 will be to make the drone descend following a continuous slope of a determined angle. The geometric representation of the trajectory, for a glide path of 2.5°, can be seen in the following images 3.2, 3.3 and 3.4:

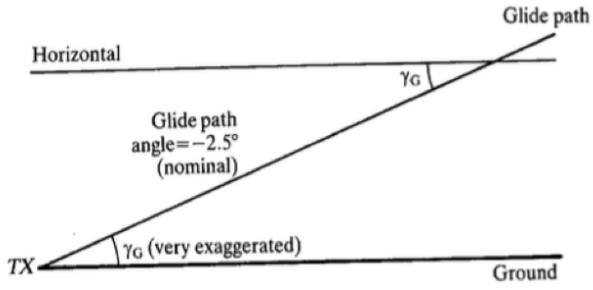


Figure 3.2: Glide path geometry. Taken from [21]

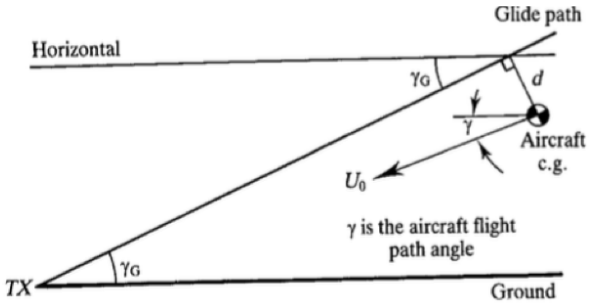


Figure 3.3: Aircarft below glide path geometry. Taken from [21]

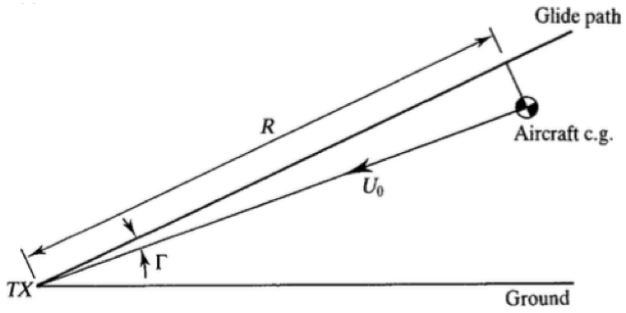


Figure 3.4: Slant range definition. Taken from [21]

Although for this first control architecture we will keep a glide path angle of 2.5° as specified in McLean, we will introduce a few changes to the model. First of all, in order to make the system non-dependent of an ILS for landing, which may reduce the application possibilities of the drone, we will suppress the glide path receiver term included in McLean. Since we already know the drone position at each instant and the glide path angle that we want it to follow, with those three elements it would be possible to calculate the distance from the aircraft to the landing point, and therefore make it possible to perform an accurate automatic guidance and control during the landing phase with no need of an ILS.

Then, we realize that the variable R, the distance from the aircraft to the runway threshold, depends upon time. Therefore, in order to solve this problem, we will add a gain multiplying the 57.3/R term of value R. In this way, no matter which value R takes along time, because we compensate with the next term we would be simulating in the model a constant value of 1 m. However, in reality R would be changing according to the real distance between the aircraft and the landing point. In this way it gets reflected in the model, but we don't have to calculate its value for each time period.

Finally, we will suppress the T1 and T2 terms that appears in the glide path coupled controller, since for this case they were considered to have any remarkable effect in improving the response characteristics of the signal.

The design process that we will have to carry out will consist of finding the unknown values for the gains in order to stabilize the system. Besides, we will study how the control system behaves in two different scenarios: for the drone flying in clear sky atmospheric conditions and for the drone flying in windy atmospheric conditions.

The Matlab scripts developed can be found on Appendix A A.1, A.2. and A.3.

3.2.2 Clear sky flight configuration

First of all, we will build our block diagram in Simulink.

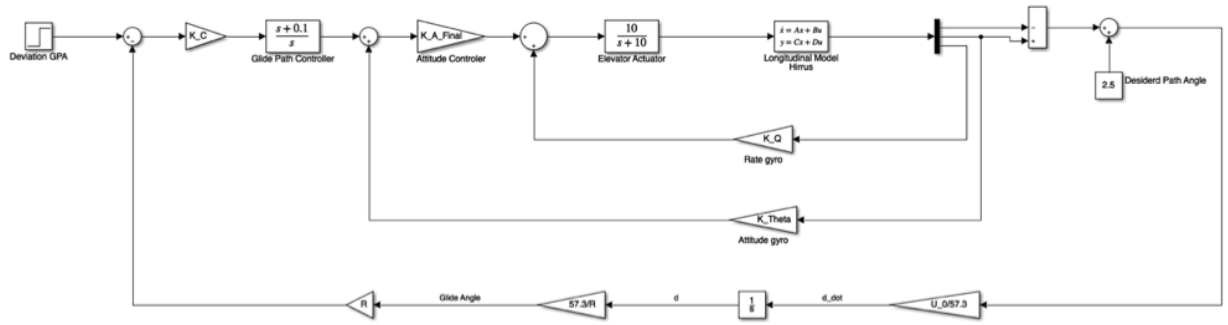


Figure 3.5: Simulink block diagram of the traditional autolanding in clear sky flight conditions. Own source.

Now, we will obtain the values for K_Q , K_Theta , K_A and K_C . In order to do so, we will solve the different loops in the diagram; however, the values obtained will be used as starting points to know how to stabilize the response: in some cases we will need to iterate until reaching the gain values that best define the response that we are looking for.

We start obtaining the value for K_Q solving the inner loop in the diagram.

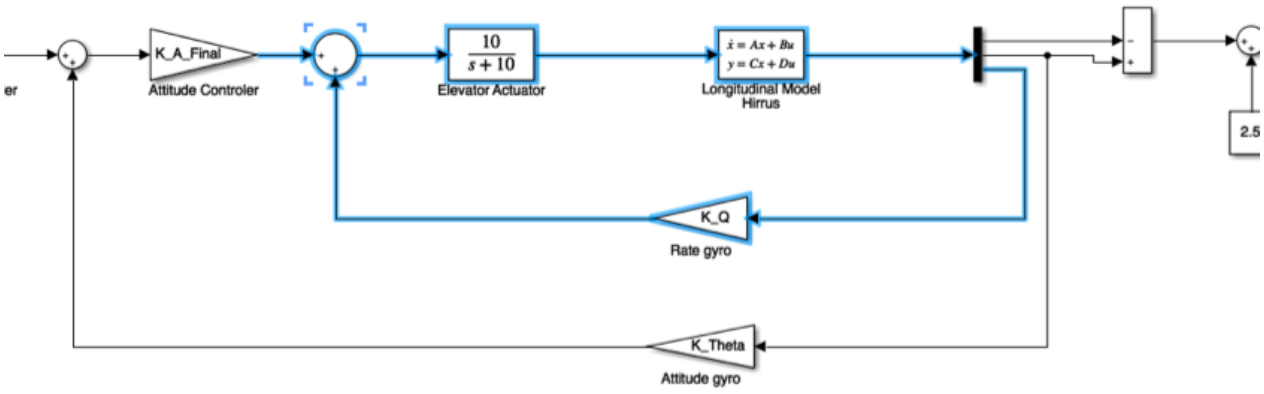


Figure 3.6: Q state loop. Own source.

In order to do so, we will make use of the *sisotool* function in Matlab: with this function we are able to see the response of the system to a unit step. We have to keep in mind that the response to be showed with *sisotool* won't be the same as in our diagram, since the input signal will be different. For K_Q $= -1$ we obtain the following response:

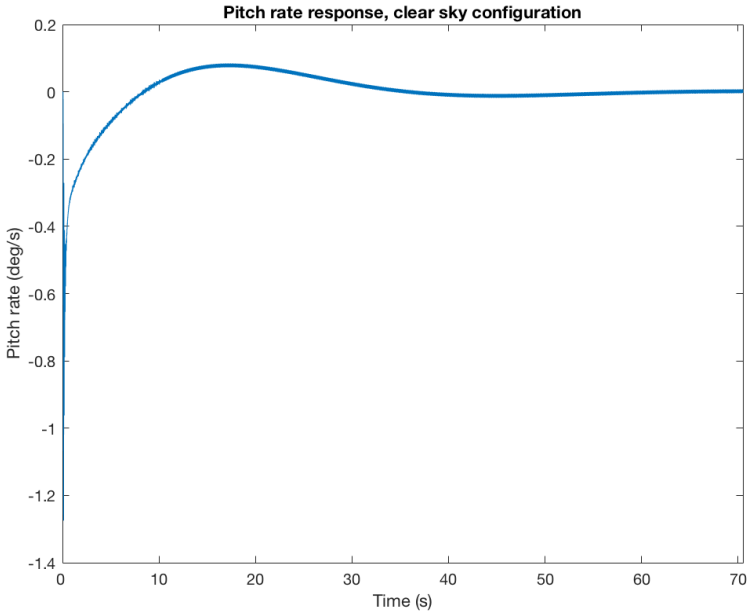


Figure 3.7: Pitch rate response, clear sky flight. Own source.

As for the second loop, corresponding to the pitch angle state:

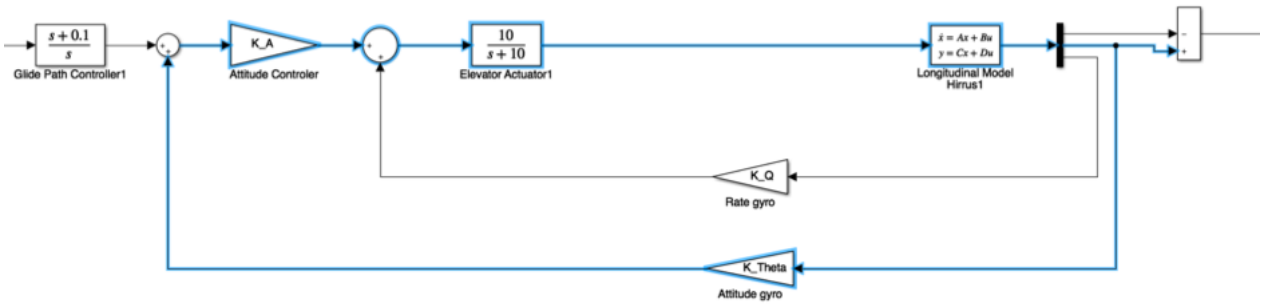


Figure 3.8: Pitch angle state loop. Own source.

In this case, we will need to define the values of K_Theta and K_A. Therefore, we will make use of the Routh-Hurwitz stability criteria in order to solve this problem. In order to do so, we will find an equivalent and simplified transfer function of 4^{th} order with the most significant poles and zeros of the pitch angle closed loop transfer function: in this way, it will be easier for us to study the sign of the Routh-Hurwitz matrix's major minors.

We will assign an initial value to K_Theta in order to find the numerical value of K_A; for K_Theta = 1, a possible solution is K_A = − 3, and with these parameters we obtain a pitch angle response as follows:

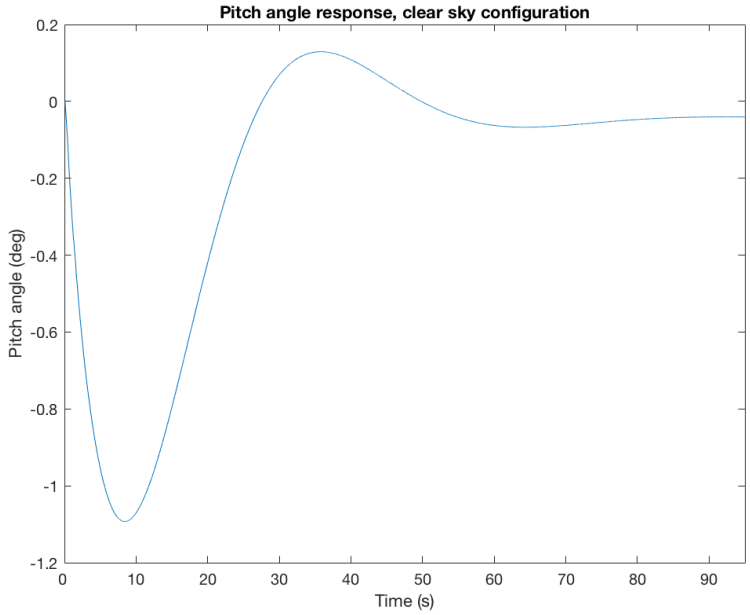


Figure 3.9: Pitch angle response, clear sky flight. Own source.

The last constant to solve will be K_C, relative to the glide path controller. Since we have already defined all the other unknowns, the strategy to design the value for K_C will be based on trying with several numbers until reaching the desiring response characteristics.

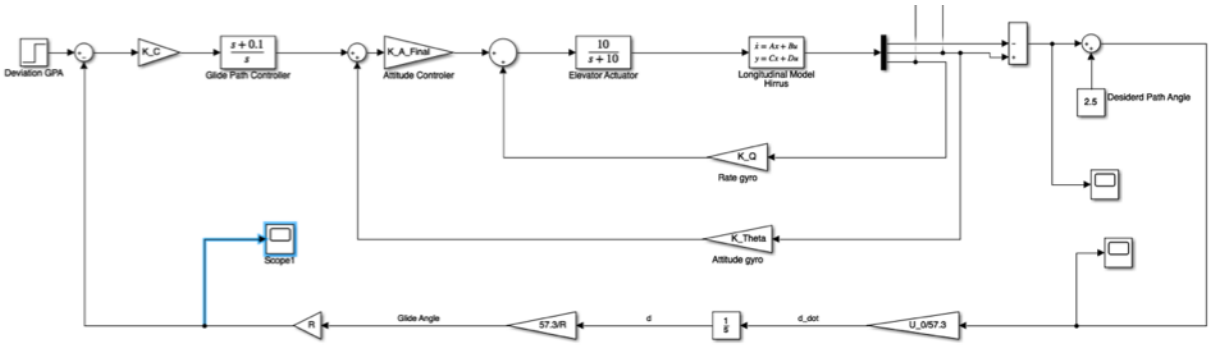


Figure 3.10: Scope situation in the diagram in order to obtain the output signal for the distance between the desired glide slope and the actual glide slope. Own source.

If we introduce K_C = 4, we obtain the following output signal for the distance between the desired glide slope and the actual glide slope:

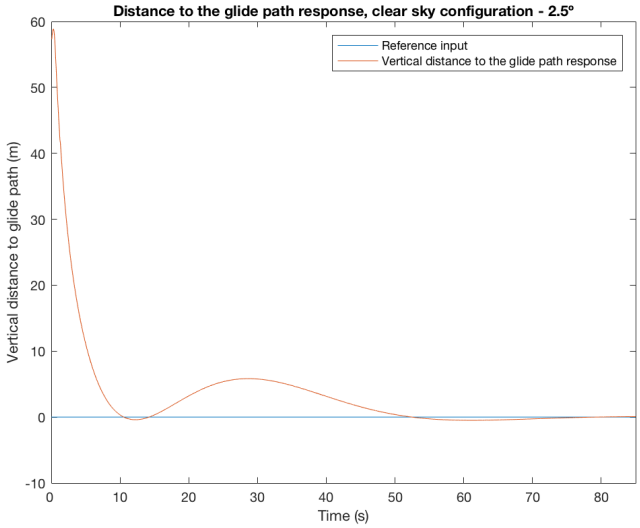


Figure 3.11: Distance to the glide path response, clear sky flight. Own source.

The values designed for the controller are summarized in the following table:

Table 3.1: Controller gains for the conventional configuration of the automatic landing system. Own source.

Element	Gain
K_ Q	−1
K_ Theta	1
K_ A	−3
K_ C	4

Now, if we change the glide angle to 7, 12 and 15°, in order to see how the drone behavior changes when modifying the design conditions:

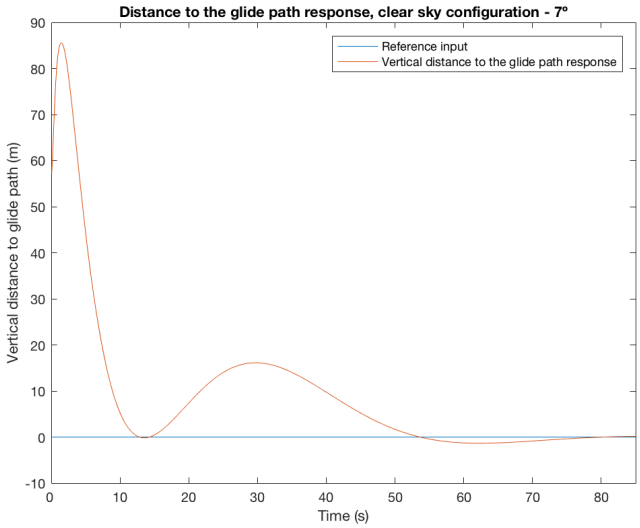


Figure 3.12: Distance to the glide path response, clear sky flight, 7°. Own source.

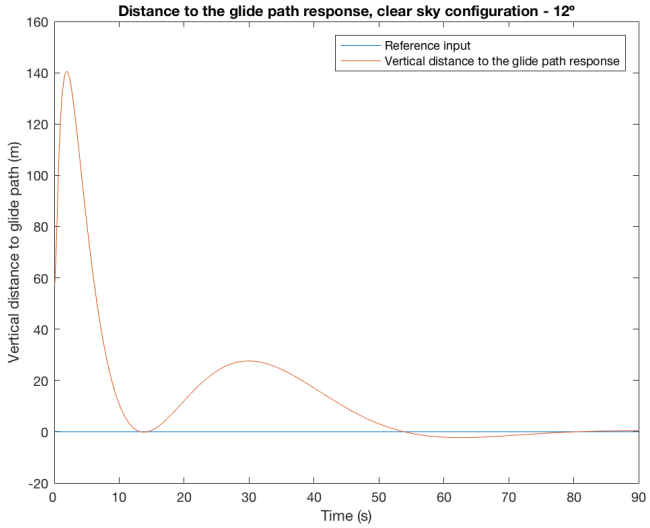


Figure 3.13: Distance to the glide path response, clear sky flight, 12°. Own source.

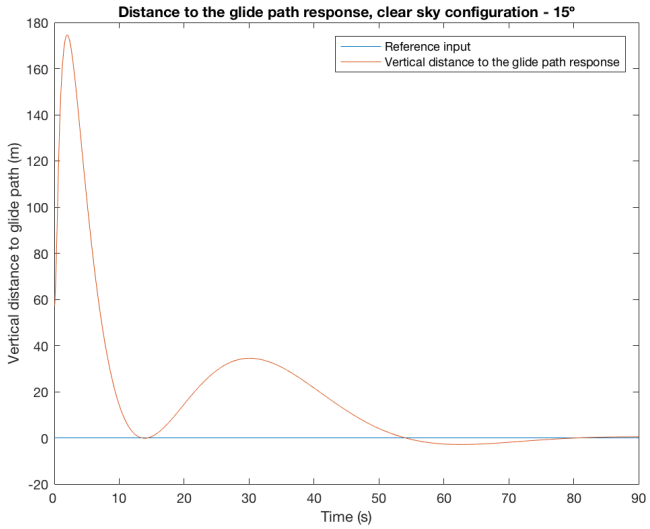


Figure 3.14: Distance to the glide path response, clear sky flight, 15°. Own source.

3.2.3 Windy flight conditions

In this scenario, we will include the effect of atmospheric wind on the aircraft. We will consider two different scenarios:

- **Constant wind perturbation.** We will introduce a constant wind velocity in the model.
- **Random wind perturbation.** This will be simulated with the Dryden wind turbulence model.

These perturbation terms would be represented in our state - space equations as:

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{pmatrix} Y_v & 0 & -U_0 & g\cos\gamma_0 & 0 \\ L_v^{'} & L_p^{'} & L_r^{'} & 0 & 0 \\ N_v^{'} & N_p^{'} & N_r^{'} & 0 & 0 \\ L_v^{'} & L_p^{'} & L_r^{'} & 0 & 0 \\ 0 & 1 & \tan\gamma_0 & 0 & 0 \\ 0 & 0 & \sec\gamma_0 & 0 & 0 \end{pmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} + \begin{pmatrix} Y_{\delta_R} & Y_{\delta_A} \\ L_{\delta_R}^{'} & L_{\delta_A}^{'} \\ N_{\delta_R}^{'} & N_{\delta_A}^{'} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{bmatrix} \delta_R \\ \delta_A \end{bmatrix} + \begin{pmatrix} X_u & x_\alpha \\ Z_u^* & Z_\alpha^* \\ M_u & M_\alpha \\ 0 & 0 \end{pmatrix} \begin{bmatrix} u_g \\ \alpha_g \end{bmatrix} \tag{3.2}$$

where u_g is the x component of the wind turbulence vector, and α_g is obtained making use of the approximation:

$$\alpha_g = \tan\frac{w_g + w}{U_0} \approx \frac{w_g + w}{U_0} \tag{3.3}$$

where w_g is the y component of the wind turbulence vector, w the vertical speed of the aircraft, and U_0 the module of the aircraft speed in t = 0.

Constant wind effect

We need to introduce two constant inputs in the dynamics model, which will correspond to the velocity component of the x and z axes:

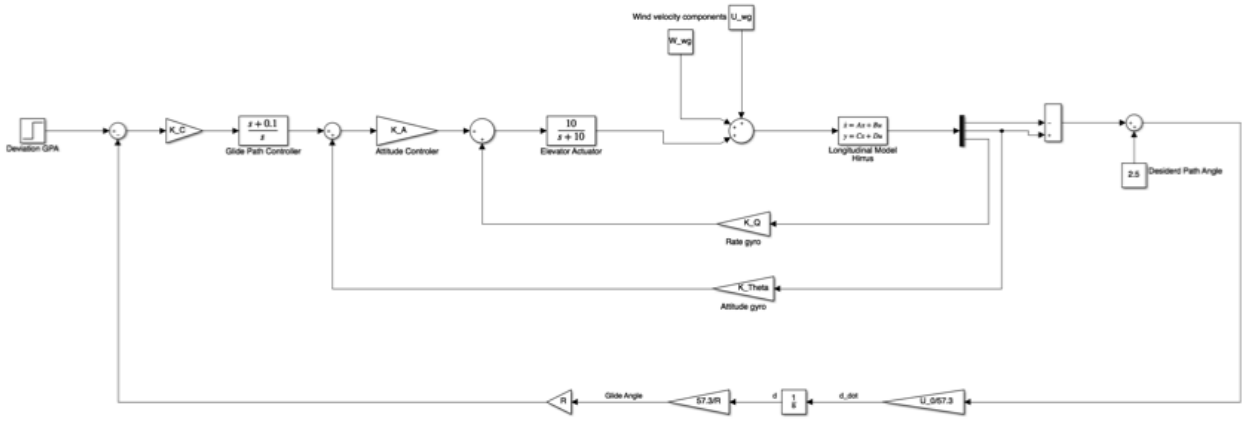


Figure 3.15: Simulink block diagram of the traditional autolanding in constant wind flight conditions. Own source.

We obtain the following responses for the pitch rate and pitch angle:

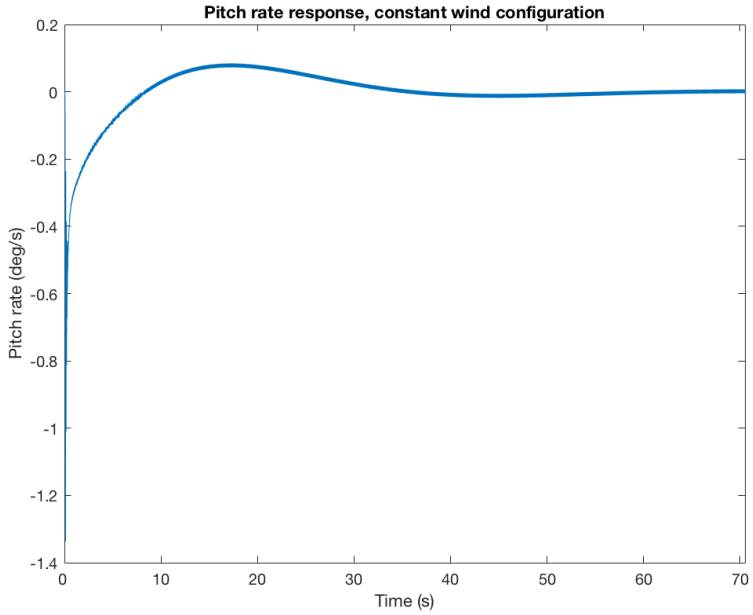


Figure 3.16: Pitch rate track response in constant wind flight conditions. Own source

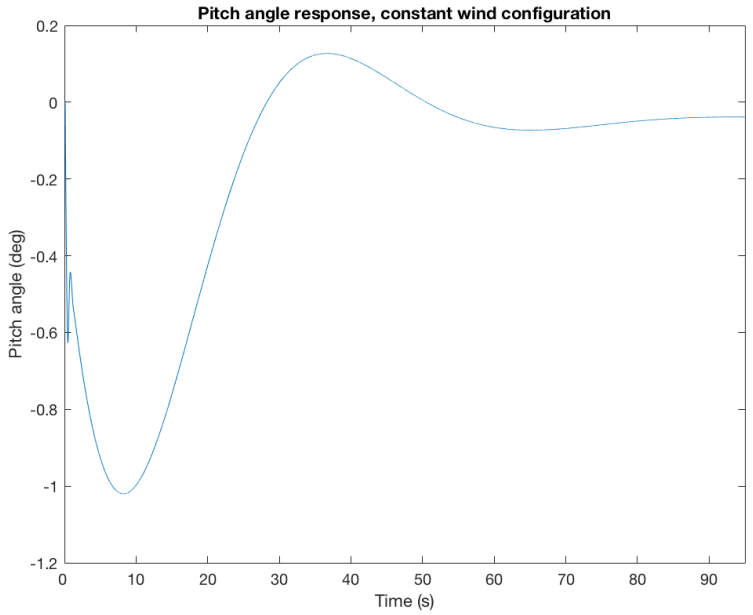


Figure 3.17: Pitch angle track response in constant wind flight conditions. Own source.

In figure 3.16 we can appreciate the effect of the perturbations , where multiple micro-oscillations appear in the response, of bigger amplitude than the oscillations that appeared in the clear sky flight conditions (figure 3.8).

Finally, as for the distance to the glide path:

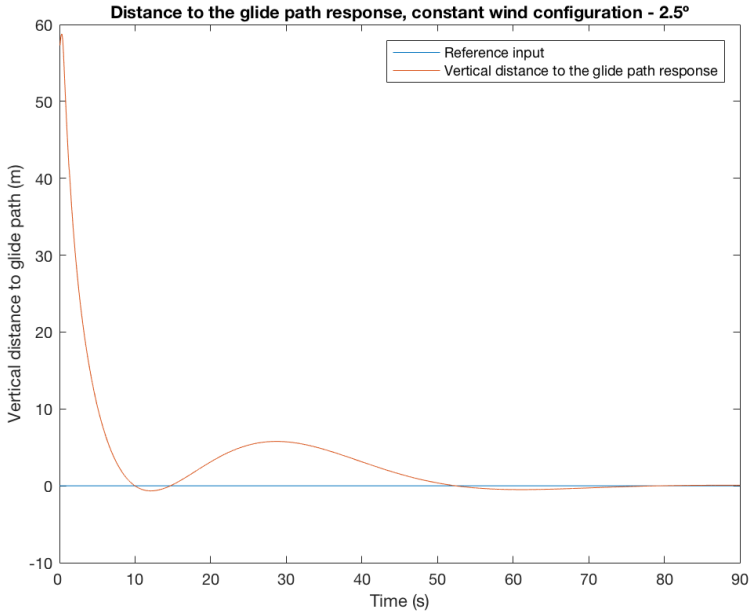


Figure 3.18: Distance to the glide path response in constant wind flight conditions. Own source.

Once more, if we include the responses obtained for the different designed glide angles, we obtain:

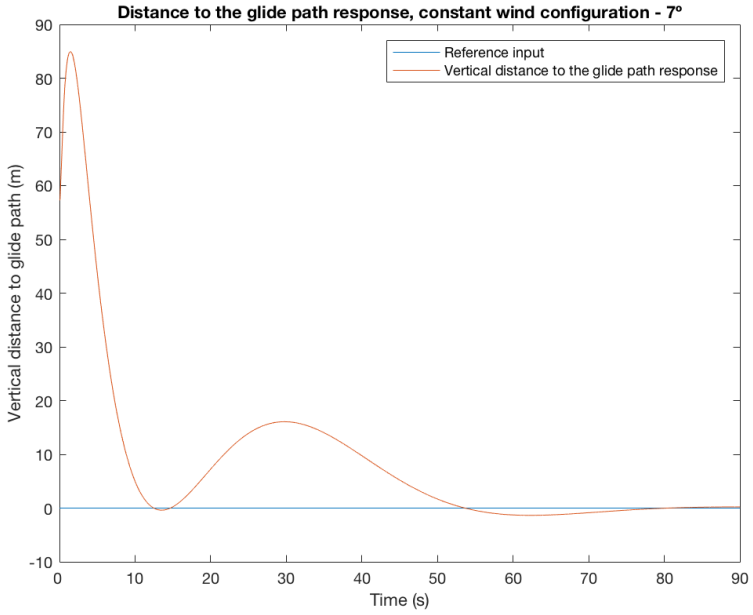


Figure 3.19: Distance to the glide path response, constant wind flight, 7°. Own source.

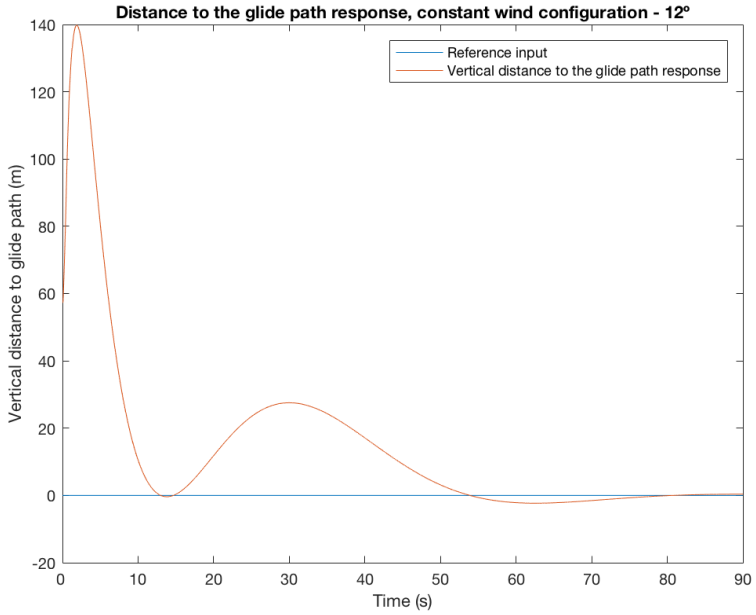


Figure 3.20: Distance to the glide path response, constant wind flight, 12°. Own source.

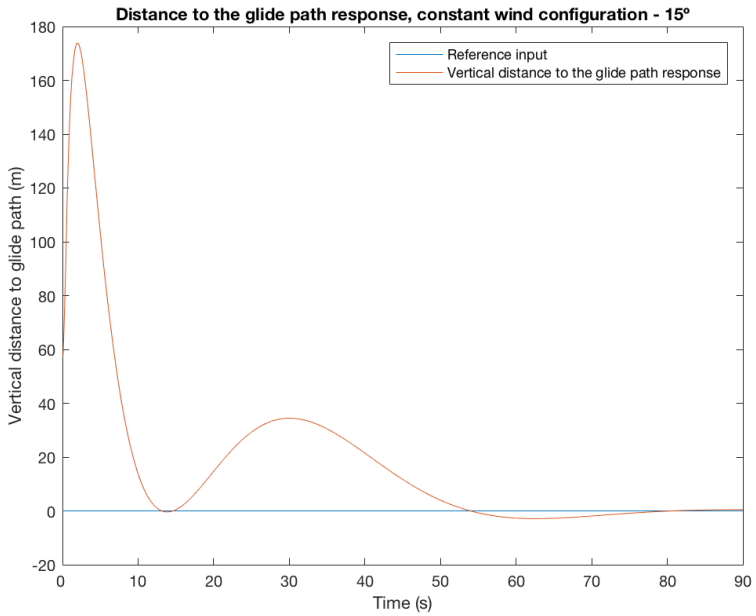


Figure 3.21: Distance to the glide path response, constant wind flight, 15°. Own source.

Random wind effect

This time, we will need to implement in our model the discrete Dryden (+q +r) block, found in the Aerospace Blockset (Simulink Library). The q and r terms make reference to the vertical and lateral turbulence angular rates of the perturbations, respectively.

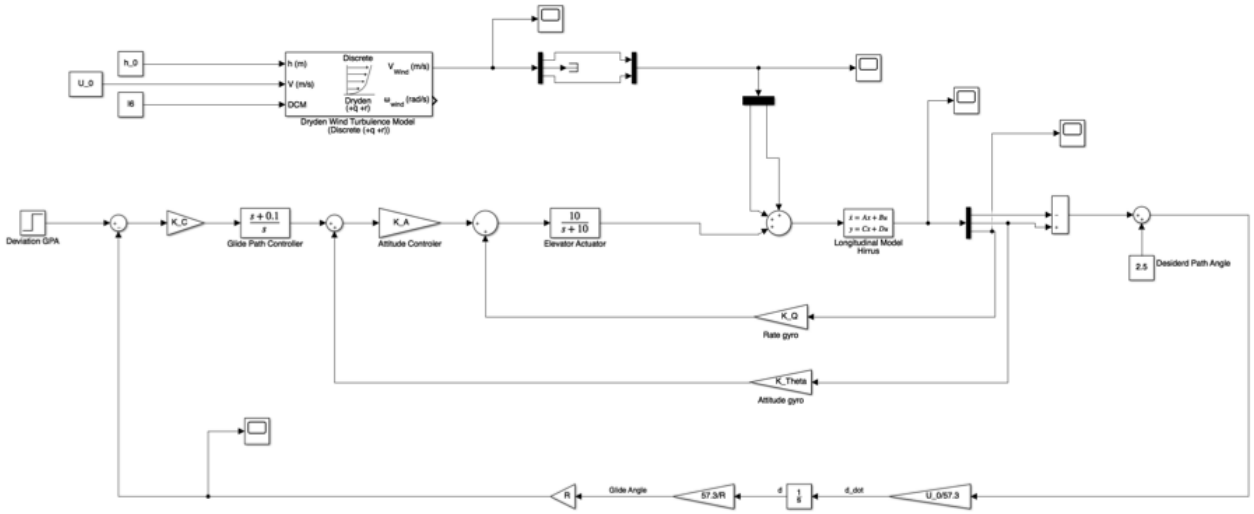


Figure 3.22: Simulink block diagram of the traditional autolanding in random wind flight conditions. Own source.

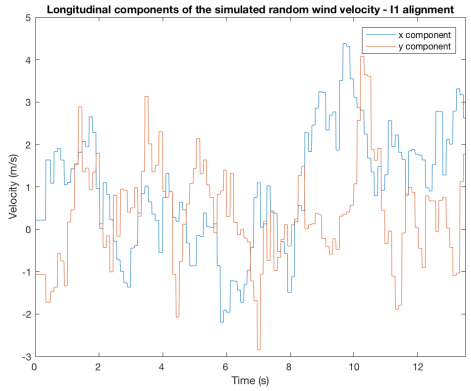
We introduce the state velocity and height as input, as well as the Direction Cosine Matrix (DCM), which gives the orientation of the wind gust with respect the aircraft. As output we have the wind turbulence lineal and angular velocities, although we will only make use of the lineal one. Besides, as we can appreciate, in spite the fact that the Dryden model gives us the 3D components of the wind velocity, we will consider only the x and z coordinates: we are working with the longitudinal dynamics of the drone, in 2D.

We will keep the same gain values calculated in the clear sky scenario, and change only the DCM. In this way we will be able to study the behavior of the control loop under different wind orientation scenarios.

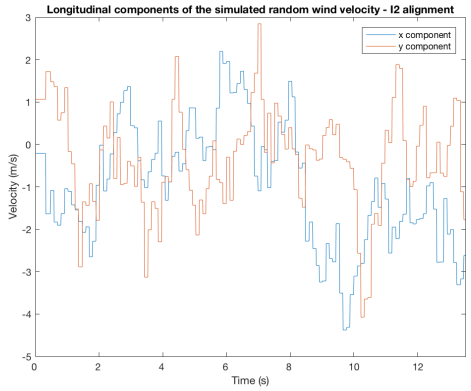
Table 3.2: Wind gust orientations to be applied.

Name	Description	DCM
I1	Aircraft and wind gust x, y and z axis aligned, same sense	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
I2	Aircraft and wind gust x, y and z axis aligned, opposite sense	$\begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$
I3	x axis aligned, random orientation in the y and z axis	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.1 & 0.8 \\ 0 & -0.7 & -0.2 \end{pmatrix}$
I4	y axis aligned, random orientation in the x and z axis	$\begin{pmatrix} 0.5 & 0 & 0.8 \\ 0 & 1 & 0 \\ -0.4 & 0 & -0.2 \end{pmatrix}$
I5	z axis aligned, random orientation in the x and y axis	$\begin{pmatrix} 0.5 & 0.1 & 0 \\ -0.4 & -0.7 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
I6	x, y and z axis unaligned, random orientation	$\begin{pmatrix} -0.1 & 0.4 & -0.3 \\ 0.5 & -0.8 & -0.7 \\ 0.2 & -0.6 & 0.9 \end{pmatrix}$

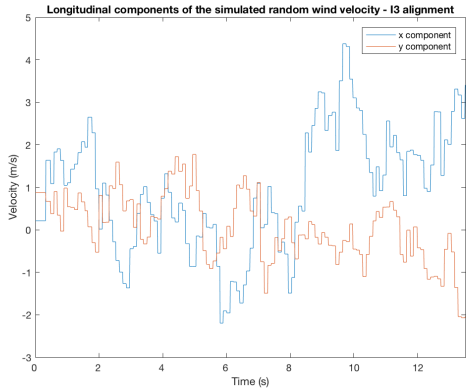
The following images depict the x and z components of the wind velocity outputs for each one of the cases determined in table 3.2:



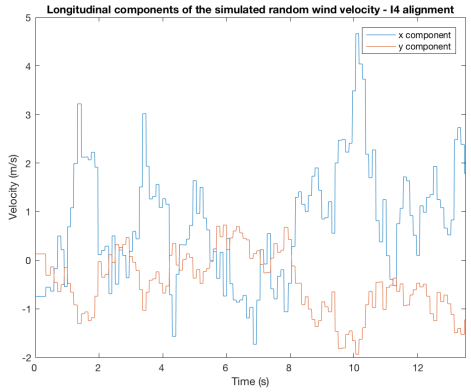
(a) Wind velocity outputs for I1.



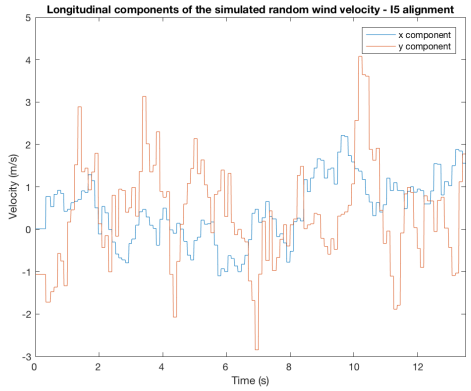
(b) Wind velocity outputs for I2.



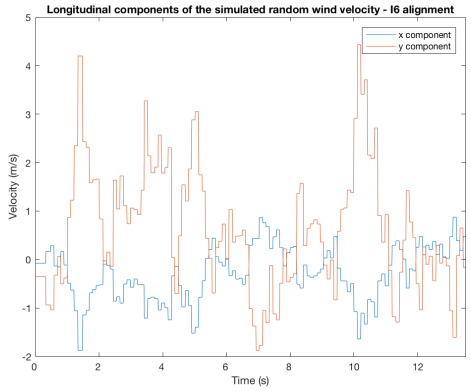
(c) Wind velocity outputs for I3.



(d) Wind velocity outputs for I4.



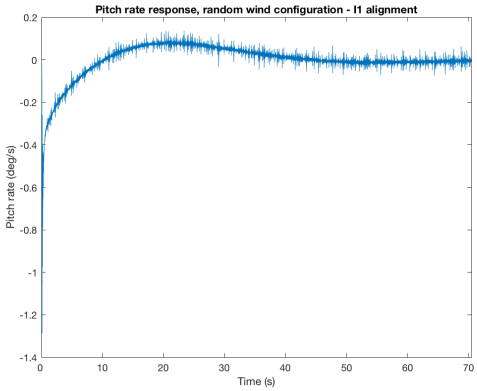
(e) Wind velocity outputs for I5.



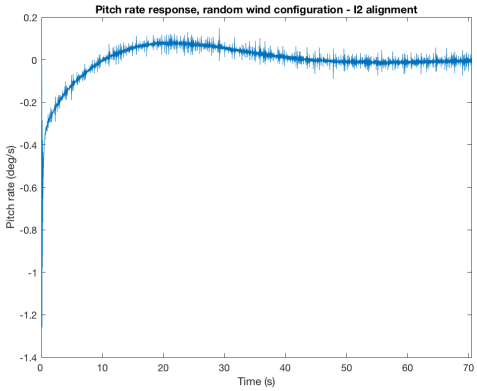
(f) Wind velocity outputs for I6.

Figure 3.23: X and z wind gust velocity component outputs for each orientation in random wind flight conditions. Own source.

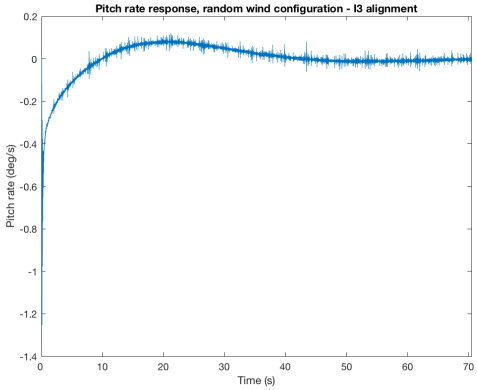
And we obtain the following responses:



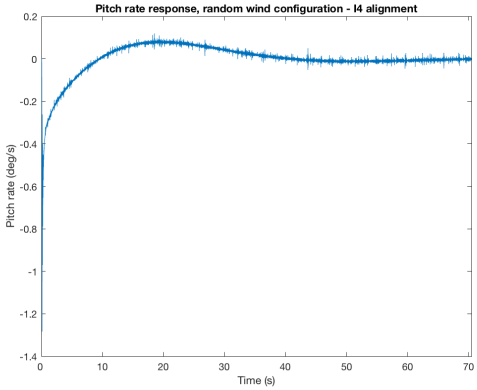
(a) Pitch rate track response for I1.



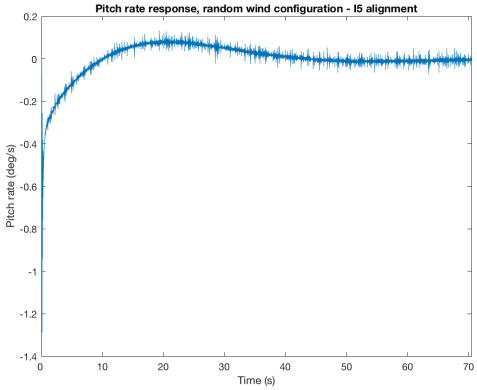
(b) Pitch rate track response for I2.



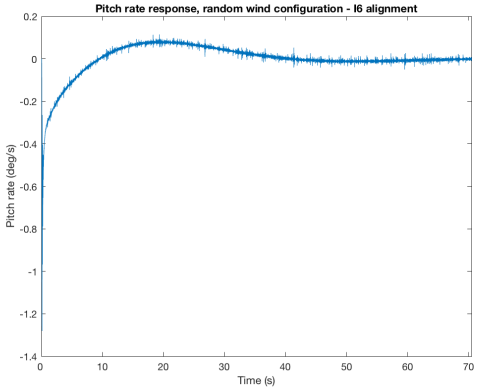
(c) Pitch rate track response for I3.



(d) Pitch rate track response for I4.



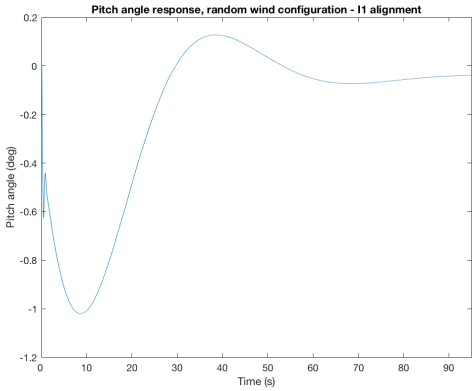
(e) Pitch rate track response for I5.



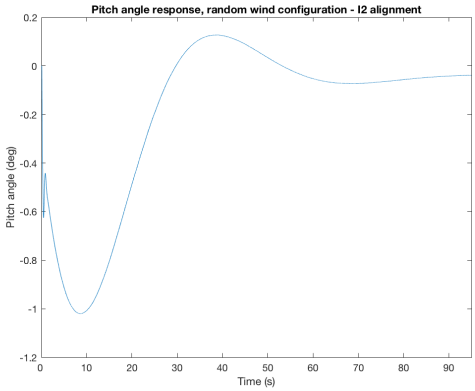
(f) Pitch rate track response for I6.

Figure 3.24: Pitch rate track response in random wind flight conditions. Own source.

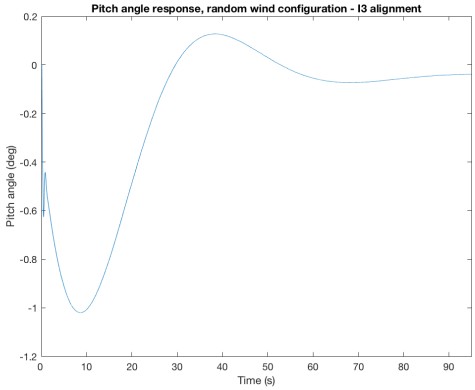
Where it can be clearly seen the effect of the perturbation in the signal. As for the pitch angle response, we obtain:



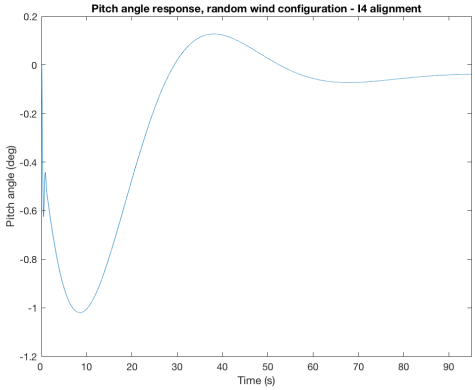
(a) Pitch angle track response for I1.



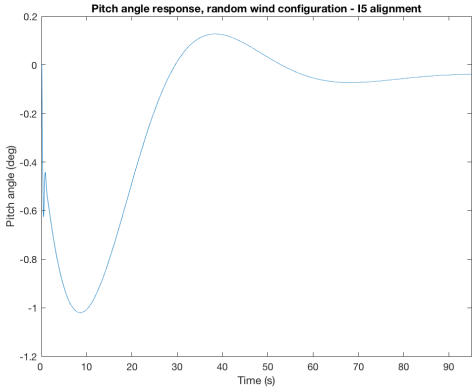
(b) Pitch angle track response for I2.



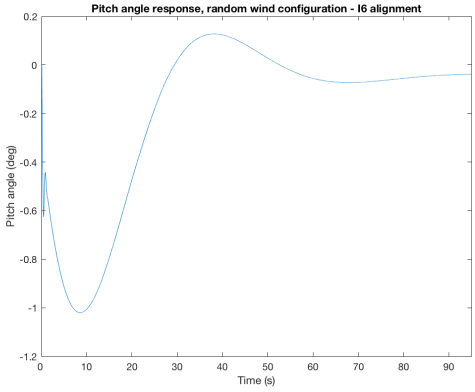
(c) Pitch angle track response for I3.



(d) Pitch angle track response for I4.



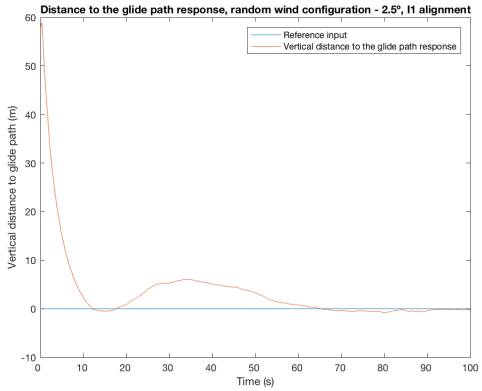
(e) Pitch angle track response for I5.



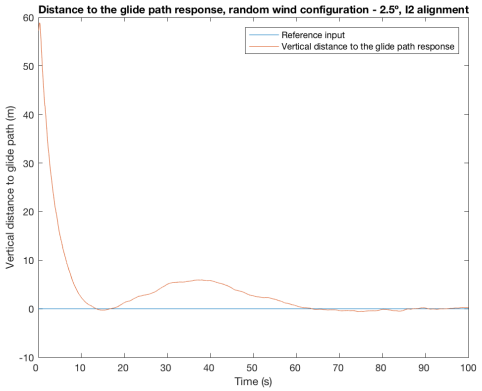
(f) Pitch angle track response for I6.

Figure 3.25: Pitch angle track response in random wind flight conditions. Own source.

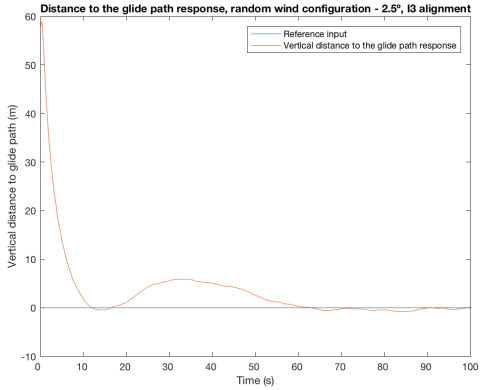
And finally, as for the distance response between the actual and desired glide slope trajectory, we have:



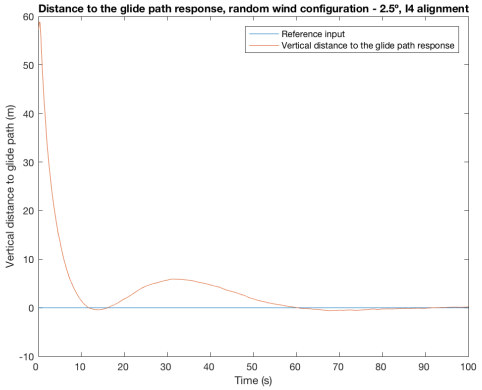
(a) Distance to the glide path response for I1.



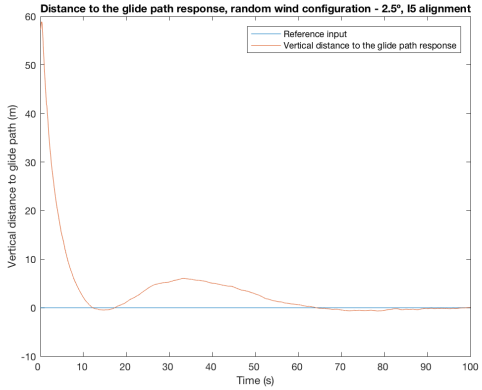
(b) Distance to the glide path response for I2.



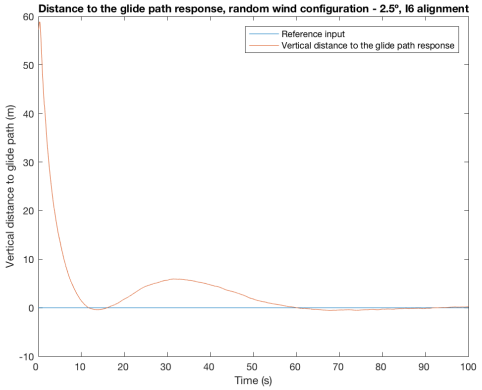
(c) Distance to the glide path response for I3.



(d) Distance to the glide path response for I4.



(e) Distance to the glide path response for I5.



(f) Distance to the glide path response for I6.

Figure 3.26: Distance to the glide path response plots for a glide angle of 2.5° in random flight conditions, I1 - I2 - I3 - I4 - I5 and I6 alignments. Own source.

To conclude, if we keep one alignment constant, we can study how the response changes when modifying the glide angle (as already done with the previous flight configurations). If we keep the I6 alignment, we obtain:

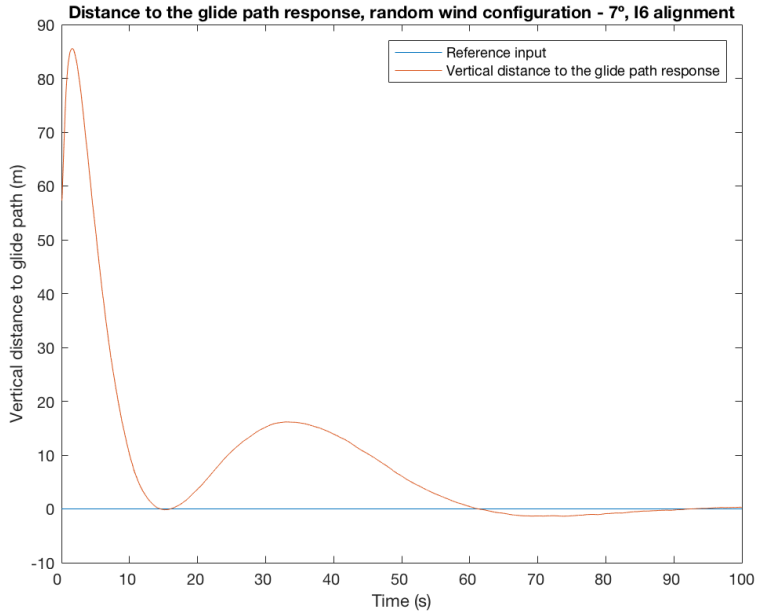


Figure 3.27: Distance to the glide path response, random wind flight, 7°. Own source.

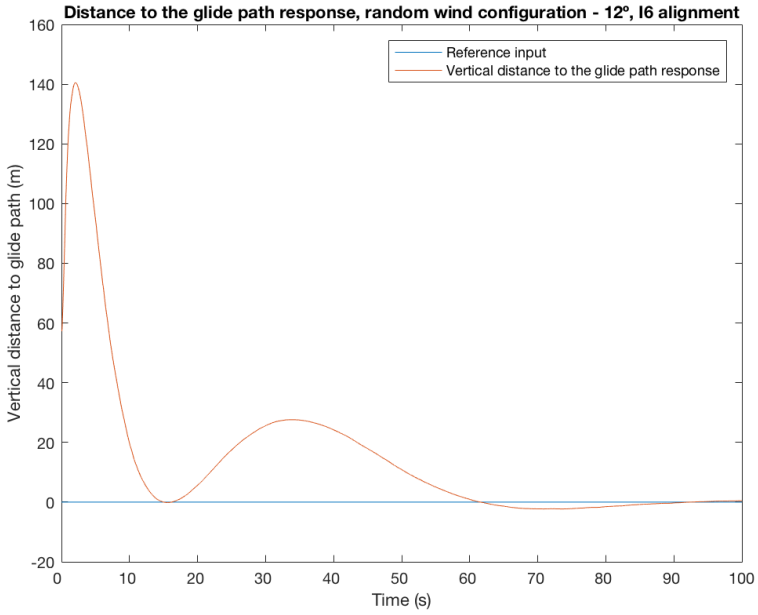


Figure 3.28: Distance to the glide path response, random wind flight, 12°. Own source.

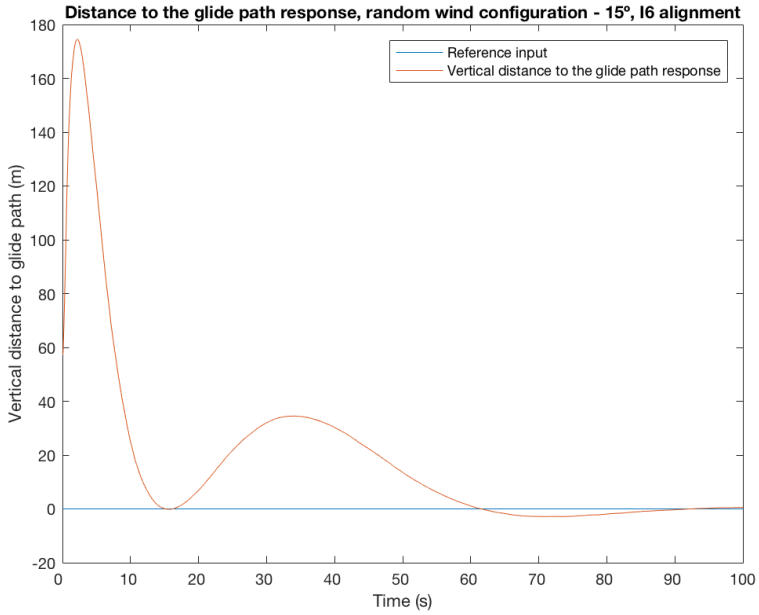


Figure 3.29: Distance to the glide path response, random wind flight, 15°. Own source.

3.3 Modern configuration: H infinity method

3.3.1 Description of the method

The second method to be considered will be based on the newer approaches in control theory. In our case, we will make use of the H_{inf} method (H infinity), also called gamma - attenuation problem. This theory was developed in the late 1980s and at the beginning of the 1990, and consists on the determination of the controller to achieve stabilization with a guaranteed performance. It is a really powerful tool widely used, since it helps the designer to optimize robust performance and robust stabilization.

The process is represented as follows:

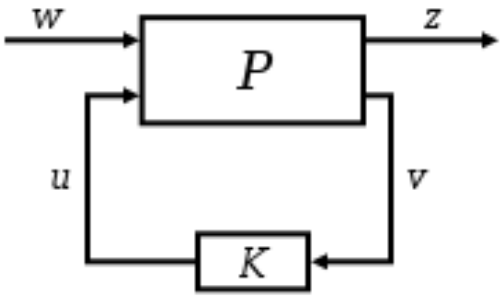


Figure 3.30: H_{inf} problem loop. Taken from [22].

In the input vector w we include all those exogenous inputs to be considered in the problem: reference signals and disturbances. In the input vector u we include the manipulated variables. As for the outputs vectors, in z we will include the error signals that we want to minimize, and in v the measured variables that we will use to control the system.

P is the plant model of our system, and K is the controller to be found. Both are matrices.

If we suppose that our dynamics model has been linearized for a pitch angle value coincident with the glide path, the H_{inf} will find the controller that makes sure that our drone follows this trajectory with the specifications indicated, minimizing the distance between the desired glide slope and the actual UAV's trajectory. For this control method, we will change the glide path angle to 12, so we study more landing trajectories scenarios. We have defined our input and output vectors to be:

$$w = \begin{bmatrix} n_{wg} \\ h_c \\ n_s \end{bmatrix} \tag{3.4}$$

where n_{wg} comprises the turbulence noise terms n_u, n_w , h_c is the commanded altitude above the glide slope (of value 0), and n_s represents the sensor noises $n_\alpha, n_q, n_\theta, n_h$.

$$z = \begin{bmatrix} z_e^{'} \\ \delta \end{bmatrix} \tag{3.5}$$

where $z_e^{'}$ is the error to minimize ($error_h$), and δ is the error associated to the actuator position.

$$v = \begin{bmatrix} \alpha + n_\alpha \\ q + n_q \\ \theta + n_\theta \\ h + n_h \end{bmatrix} \tag{3.6}$$

where the measured vector will consist of the angle of attack, pitch rate, pitch angle, the height, the aircraft altitude, and their corresponding sensor noises.

$$u = \delta_c \tag{3.7}$$

where δ_c is the actuator control input.

Now, knowing that figure 3.7 has the following state - space representation:

$$\begin{aligned} \dot{x}_G &= A_G x_G + B_1 w + B_2 u \\ \dot{z} &= C_1 x_G + D_{11} w + D_{12} u \\ \dot{v} &= C_2 x_G + D_{21} w + D_{22} u \end{aligned} \tag{3.8}$$

we will have to properly define the vector x_G in order to determine the matrices $A_G, B_1, B_2, C_1, C_2, D_{11}, D_{12}$, D_{21}, D_{22}, since the plant model P is defined to be:

$$P = \begin{pmatrix} A_G & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{pmatrix} \tag{3.9}$$

which is needed for obtaining the controller K.

In such vector x_G we will include the information corresponding to the longitudinal dynamics model of the drone, the actuator model, the performance weighting model, and the wind turbulence model.

3.3.2 Models included in the plant

Longitudinal Dynamics Model We will make use of the model previously defined in section 3.1; however, we will add the altitude to the state vector, the parameter to be controlled, making use of the following relationship:

$$\dot{h} = V_0 \sin\gamma \simeq V_0\gamma = V_0\left(\theta - \alpha\right) \tag{3.10}$$

In this way, our dynamics model will be:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{3.11}$$

where:

$$x = \begin{bmatrix} u \\ \alpha \\ q \\ \theta \\ h \end{bmatrix} \qquad u = \delta \qquad y = \begin{bmatrix} \alpha \\ q \\ \theta \\ h \end{bmatrix} \tag{3.12}$$

Vector x represents the longitudinal states (longitudinal velocity in the x direction of the body axes, angle of attack, pitch rate and pitch angle, as we know) including the height along the glide slope, u represents the actuator deflection, and vector y includes those four longitudinal states that we want to measure.

Actuator Model The model associated with the actuator dynamics will be represented as follows:

$$\begin{aligned} \dot{x}_A &= A_A x_A + B_A u_c \\ u &= C_A x_A \end{aligned} \tag{3.13}$$

where

$$x_A = \delta \qquad u_c = \delta_c \qquad u = \delta \tag{3.14}$$

Vector x_a consists of the elevator dynamics and u_c represents the elevator position command.

Performance Weighting Model The weighting model is used to control and determine the tracking errors and control command variables specifications. In our case, we will have to optimize the height along the glide slope error and the actuator deflection.

$$\begin{aligned} \dot{x}_{wp} &= A_{wp}x_{wp} + B_{wp}z_e \\ z_e^{'} &= C_{wp}x_{wp} + D_{wp}z_e \end{aligned} \tag{3.15}$$

where z_e can be expressed as:

$$z_e = C_{ze}x - I_1 r_c \tag{3.16}$$

and

$$r_c = h_c \qquad C_{ze} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad I_1 = 1 \tag{3.17}$$

where r_c is the command vector.

In order to determine the weighting values we will have to iterate until reaching the output response that we are aiming. Concerning the tracking error, we will start with the approach given in [23]:

- Time Constant: 1.0 seconds.
- Steady - State Error: 0.1 % .
- Step Response Overshoot: 10 %.

Those requirements will be used for obtaining a performance weighting function W_p of the shape:

$$W_p(s) = \frac{K\,(s + \omega_1)}{(s + \omega_2)} \tag{3.18}$$

Making using of the Final Value Theorem and the specifications listed above, we get the necessary relationship between K, ω_1 and ω_2 and so we obtain a first weighting function:

$$W_p(s) = \frac{0.1\,(s + 10)}{(s + 0.001)} \tag{3.19}$$

As far as the control position is concerned, it can be weighted with a scalar value. We will start for instance with $W_\delta = 0.1$.

Finally, during the design process and the necessary changes to improve performance, it was determined the weighting function and scalar to be:

$$W_p(s) = \frac{1\,(s + 1)}{(s + 0.99)} \qquad W_\delta = 0.01 \tag{3.20}$$

Wind Turbulence Model The state model associated can be written as follows:

$$\begin{aligned} \dot{x}_{wg} &= A_{wg}x_{wg} + B_{wg}n_{wg} \\ y_{wg} &= C_{wg}x_{wg} \end{aligned} \tag{3.21}$$

where:

$$x_{wg} = \begin{bmatrix} u_{wg} \\ w_{wg1} \\ w_{wg2} \end{bmatrix} \qquad n_{wg} = \begin{bmatrix} n_u \\ n_w \end{bmatrix} \qquad y_{wg} = \begin{bmatrix} u_{wg} \\ w_{wg} \end{bmatrix} \tag{3.22}$$

Vector x_{wg} comprises the longitudinal and vertical gust components of the wind turbulence, u_{wg} and w_{wg1}, w_{wg2}, vector n_{wg} represents the noise associated, and finally y_{wg} defines the output wind gust terms which we are interested to measure, u_{wg} and w_{wg}.

We will make use once more of the Dryden turbulence model. It defines a series of expressions in order to obtain the matrices of the turbulence model, which can be found in [23]. They have the following shape:

$$A_{wg} = \begin{pmatrix} -\frac{V_0}{L_u} & 0 & 0 \\ 0 & -\frac{V_0}{L_w} & 0 \\ 0 & \sigma_w \frac{1}{\sqrt{L_w}} \sqrt{\frac{3V_0}{2\pi}} & -\frac{V_0}{L_w} \end{pmatrix} \qquad B_{wg} = \begin{pmatrix} \sigma_u \frac{1}{\sqrt{L_u}} \sqrt{\frac{V_0}{\pi}} & 0 \\ 0 & -V_0 \frac{1}{L_w} \left(1 - \frac{1}{\sqrt{3}}\right) \\ 0 & \sigma_w \frac{1}{\sqrt{L_w}} \sqrt{\frac{3V_0}{2\pi}} \end{pmatrix} \qquad C_{wg} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.23}$$

where:

- σ_u and σ_w indicate turbulence intensities.
- L_u and L_w indicate scale length.

These variables are already defined, and they change according to the altitude of operation. In [23] it appears a table with the value of those parameters and different altitudes (from 20.0131 to 1500.3937 ft). Given the nominal flight conditions of our model, we will choose the altitude that gets closer: 20.0131 ft, or 6.1 meters.

Table 3.3: Turbulence specifications for the Dryden Turbulence Model, given the nominal flight conditions of our UAV's dynamics. Taken from [23].

Altitude (ft)	L_u (ft)	L_w (ft)	σ_u (ft/sec)	σ_w (ft/sec)
20.0131	105.7415	10.4003	5.7385	3.9494

We will need to change the units from feet to meters, since in this work we are making use of the international system units.

3.3.3 Plant model and results

In equation (3.9) we define the state space representation of the plant model. Once we have defined the different elements to be included in it, we can define vector x_G to be:

$$x_G = \begin{bmatrix} x \\ x_A \\ x_{wg} \\ x_{wp} \end{bmatrix} = \begin{bmatrix} u \\ \alpha \\ q \\ \theta \\ u_{wg} \\ w_{wg} \\ h \end{bmatrix} \tag{3.24}$$

Getting the state equations of each model, and substituting those terms related among them we obtain the following systems:

$$\begin{array}{rcl} \dot{x} & = & Ax + BC_A x_A \\ \dot{x}_A & = & A_A c_A + B_A u_c \\ \dot{x}_{wg} & = & A_{wg} x_{wg} + B_{wg} n_{wg} \\ \dot{x}_{wp} & = & A_{wp} x_{wp} + B_{wp} C_{ze} x - B_{wp} r_c \end{array} \tag{3.25}$$

$$\begin{array}{rcl} z_e^{'} & = & D_{wp} C_{ze} x + C_{wp} x_{wp} - D_{wp} r_c \\ \delta & = & W_\delta C_A x_A \end{array} \tag{3.26}$$

$$v = Cx + 0.01 n_s \tag{3.27}$$

Expressing in matrix form, we reach:

$$\begin{bmatrix} \dot{x} \\ \dot{x}_A \\ \dot{x}_{wg} \\ \dot{x}_{wp} \end{bmatrix} = \begin{pmatrix} A & BC_A & 0 & 0 \\ 0 & A_A 0 & 0 & \\ 0 & 0 & A_{wg} & 0 \\ B_{wp} C_{ze} & 0 & 0 & A_{wp} \end{pmatrix} \begin{bmatrix} x \\ x_A \\ x_{wg} \\ x_{wp} \end{bmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ B_{wg} & 0 & 0 \\ 0 & -B_{wp} & 0 \end{pmatrix} \begin{bmatrix} n_{wg} \\ h_c \\ n_s \end{bmatrix} + \begin{bmatrix} 0 \\ B_A \\ 0 \\ 0 \end{bmatrix} \delta \tag{3.28}$$

$$\begin{bmatrix} z_e^{'} \\ \delta \end{bmatrix} = \begin{pmatrix} D_{wp} C_{ze} & 0 & 0 & C_{wp} \\ 0 & W_\delta C_A & 0 & 0 \end{pmatrix} \begin{bmatrix} x \\ x_A \\ x_{wg} \\ x_{wp} \end{bmatrix} + \begin{pmatrix} 0 & -D_w p & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{bmatrix} n_{wg} \\ h_c \\ n_s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \delta \tag{3.29}$$

$$v = \begin{bmatrix} C & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ x_A \\ x_{wg} \\ x_{wp} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0.01 \end{bmatrix} \begin{bmatrix} n_{wg} \\ h_c \\ n_s \end{bmatrix} + 0 \cdot \delta \tag{3.30}$$

From expression (3.28) we will obtain matrices A, B_1, B_2, from expression (3.29) matrices C_1, D_{11}, D_{12} and from expression (3.30)matrices C_2, D_{21}, D_{22}. So, our plant model will be defined by:

$$
\begin{aligned}
A_G &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ B_{wg} & 0 & 0 \\ 0 & -B_{wp} & 0 \end{pmatrix} & B_1 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ B_{wg} & 0 & 0 \\ 0 & -B_{wp} & 0 \end{pmatrix} & B_2 &= \begin{bmatrix} 0 \\ B_A \\ 0 \\ 0 \end{bmatrix} \\
C_1 &= \begin{pmatrix} D_{wp}C_{ze} & 0 & 0 & C_{wp} \\ 0 & W_\delta C_A & 0 & 0 \end{pmatrix} & D_{11} &= \begin{pmatrix} 0 & -D_wp & 0 \\ 0 & 0 & 0 \end{pmatrix} & D_{12} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
C_2 &= \begin{bmatrix} C & 0 & 0 & 0 \end{bmatrix} & D_{21} &= \begin{bmatrix} 0 & 0 & 0.01 \end{bmatrix} & D_{22} &= 0
\end{aligned}
\tag{3.31}
$$

Then, we apply the H_{inf} command in Matlab, obtaining the controller and closed loop state space matrices. If we plot the response corresponding to the distance to the glide path obtained for the initial conditions of $\theta_0 = -12°$ and $h_0 = 7m$:

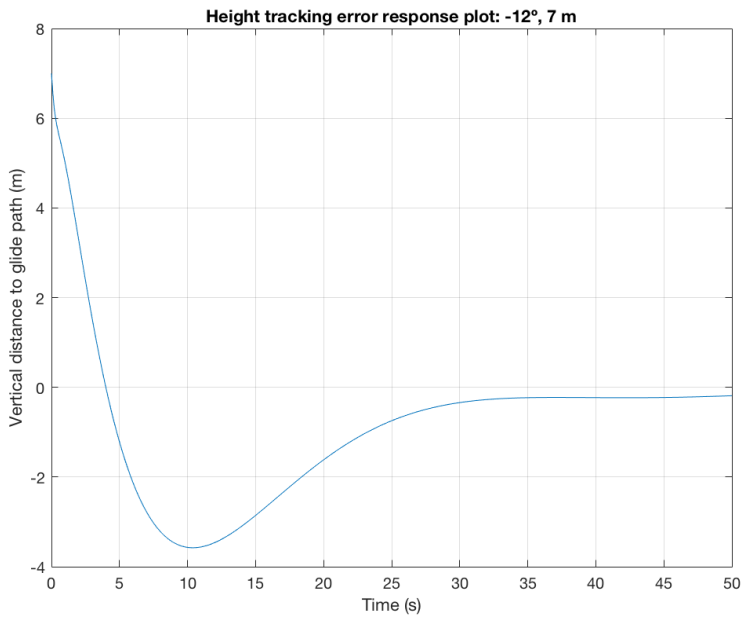
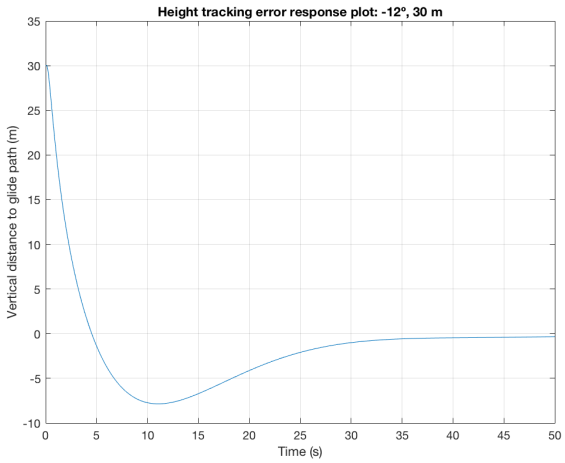
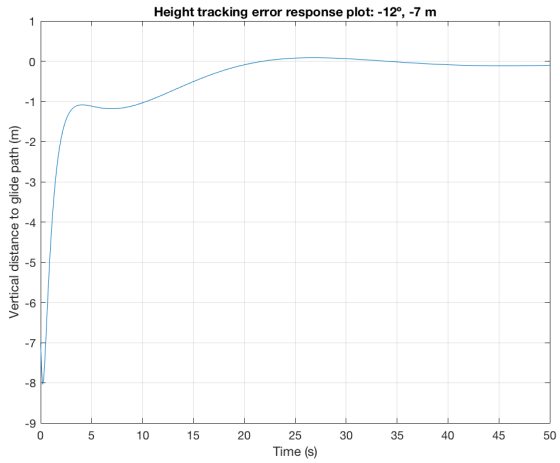


Figure 3.31: Distance to the glide path response obtained with the H_{inf} optimization method for $\theta_0 = -12°$ and $h_0 = 8m$. Own source.

If we change these initial conditions of the drone with respect to the glide path trajectory, we can study how the response changes. For instance, if we change the altitude, we obtain plots like:



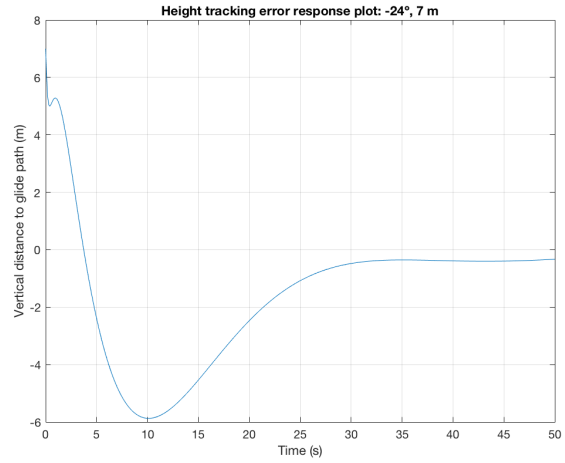
(a) Distance to the glide path response obtained with the H_{inf} optimization method for $\theta_0 = -12^\circ$ and $h_0 = 30m$.



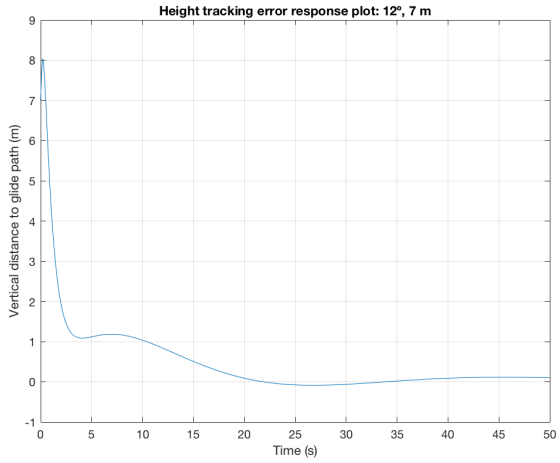
(b) Distance to the glide path response obtained with the H_{inf} optimization method for $\theta_0 = -12^\circ$ and $h_0 = -7m$.

Figure 3.32: Distance to the glide path responses with different initial nominal height. Own source.

Now, if we change the initial attitude values of the UAV:



(a) Distance to the glide path response obtained with the H_{inf} optimization method for $\theta_0 = -25^\circ$ and $h_0 = 7m$.



(b) Distance to the glide path response obtained with the H_{inf} optimization method for $\theta_0 = 12^\circ$ and $h_0 = 7m$.

Figure 3.33: Distance to the glide path responses with different initial nominal pitch angle. Own source.

The corresponding Matlab scripts can be found on Appendix B B.1. and B.2.

3.3.4 Robustness analysis

The H_{inf} method belongs to the robust controller family, as already defined in section 3.3.1. This family groups all those methods that approaches the control problem dealing with uncertainty, aiming to achieve stability or a determined performance under different assumptions. The controller is design to work assuming that certain parameters will be unknown but bounded in a specific range.

Therefore, in this section we will study the level of robustness of our controller: we will change some variables in the dynamics of our flying wing UAV and apply the controller obtained in section 3.3.3. We will perform the robustness analysis and see how the response changes with those parameters. We can refer to two different types of robustness: stability and performance, both studied in this analysis.

We will change the stability derivatives M_α, M_q and Z_α, since they are one of the most critical and important terms in the definition of the body longitudinal dynamics: we will vary them a $\pm 10\%$ with respect to the nominal value in the model. Then, we will build the plant model and connect it to the controller K, obtain the information concerning the robustness analysis and finally plot the step response of the height error for the different set of M_α, M_q and Z_α values.

After carrying out the analysis under those conditions, we got informed that the system is robustly stable for the modeled uncertainty:

- It can tolerate up to 217% of the modeled uncertainty (that is a variation of $\pm 31.7\%$ of the uncertain elements.
- There is a destabilizing perturbation amounting to 218 % of the modeled uncertainty.
- This perturbation causes an instability at the frequency 2.46 rad/seconds.
- Sensitivity with respect to each uncertain elements is:
  - 65% for M_α. Increasing M_α by 25% decreases the margin by 16%.
  - 0% for M_q. Increasing M_q by 25% decreases the margin by 0%.
  - 7% for Z_α. Increasing Z_α by 25% decreases the margin by 1.75%.

We obtain the following responses set plot for a step input signal:

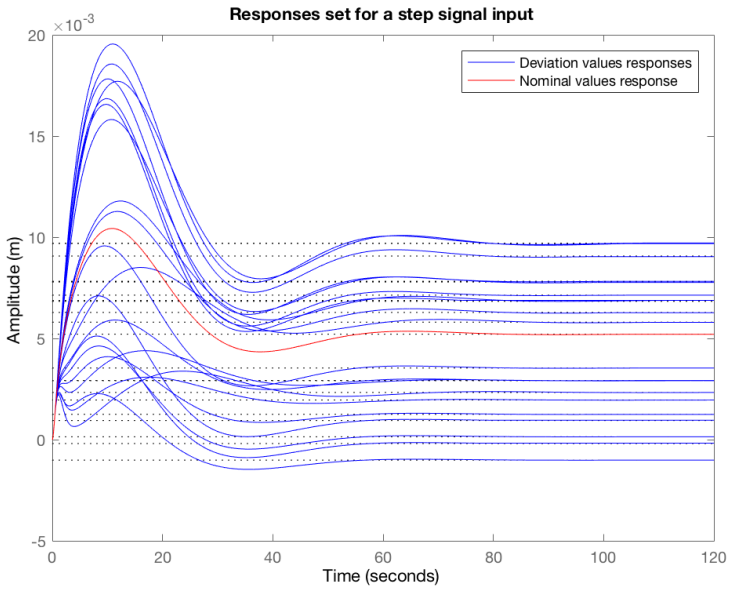


Figure 3.34: Responses set for a step input signal plot of the height tracking error. Own source

The corresponding Matlab script can be found on Appendix B B.3.

Chapter 4

Conclusions and future development

4.1 Conclusions

In this project it was developed the automatic landing system for a flying wing UAV following two different approaches: conventional and modern. It was designed the control system of the distance between the actual position of the drone during the landing trajectory and the desired glide path; our performance requirements consisted of achieving a stabilized signal response in a realistic settling time.

In order to develop the most accurate possible system, different landing configurations and scenarios were considered. We studied different weather flight conditions, introducing certain kind of perturbations in the models that simulated the wind effect, and different landing trajectories, through the change of the glide path angle.

Now, we will comment in greater depth the results obtained for each of the approaches followed, as well as the effect that the changes of the parameters had on the signal response. Finally, we will compare the performance and characteristics of both designing methods.

4.1.1 Conventional approach

First of all, we will talk about the PID controller. Applying the correct gain values help to stabilize the signal quite straightforward.

When studying how the gain values affected the signal responses, we found the following relationships among them. As for K_ A:

- It must have different sign than K_ Theta, in order to guarantee stability.
- The higher its absolute value, the higher the oscillation amplitude in the first seconds of the pitch rate, pitch angle and distance responses.
- The smaller its absolute value, the bigger the settling time for the pitch rate and distance responses.

As for K_ C, we found:

- It must have the same sign than K_ A, in order to guarantee stability.
- The higher its absolute value, the higher the oscillation amplitude in the first seconds of the pitch rate, pitch angle and distance responses.
- The smaller its absolute value, the bigger the settling time for the pitch rate and distance responses.

Concerning K_ Q:

- If having the same sign that K_ Theta, we obtain a different shape for the pitch rate and distance response with higher settling time and bigger oscillations.
- The higher its absolute value, the higher the oscillation amplitude along the pitch rate response.
- The smaller its absolute value, higher the oscillation amplitude along the pitch rate response.

Finally, about K_ Theta:

- It must have different sign than K_ A, in order to guarantee stability.
- The higher its absolute value, the higher the oscillation amplitude along the pitch rate response.
- The smaller its absolute value, higher the oscillation amplitude along the pitch rate response.

The most critical parameters are K_ C and K_ A, since they affect strongly to the stability and signal shape of the responses: the changes imposed by them prevailed over the effect of K_ Theta and K_ Q changes.

Before commenting separately each of the different flight configurations developed, it will be studied the common features perceived in the responses of the three cases. First of all, it should be mentioned that the settling time of the distance to the glide path response could be significantly improved, but we would be reaching extremely high actuator amplitudes. This is not realistic, since the actuator has to operate within determined limits, and so we decided to keep the responses showed in the text.

Second of all, it has to be noted the significant changes of the distance to the glide path signal when changing the desired glide path from the designed value. For an angle of 7, 12 and 15° the settling time increases to around 75 seconds (as depicted in figures 3.11, 3.16 or 3.24). Besides, we obtain oscillations of bigger amplitude: for instance, for a glide angle of 15°, we reach values of 170 meters above the desired path. This let us know that we would need to adjust and redesign the gain values of the controller for each of the new angles, for us to reach a good response.

Now, studying separately the three cases:

Clear Sky flight configuration

Focusing on the clear sky configuration without perturbations, the signal response of the difference between the actual position of the drone and the desired glide path with a 2.5° slope is depicted in figure 3.11. We can appreciate how the drone aligns with such trajectory with a tracking error of 0% and a settling time of around 50 seconds.

Although of secondary importance, it would be worth to note the main features of the other two measured states, the pitch rate q and the pitch angle θ. In the pitch rate response case, it can be seen some oscillations of a really small magnitude along the signal. In the pitch angle response we can appreciate its long settling time, if compared with the pitch rate and distance responses.

Windy flight configuration

Now, we will discuss the results obtained in windy flight conditions: as we can see in figure 3.18, the distance difference response between the constant wind condition and the clear sky configuration is the same. This means that the developed controller is strong enough and it is not affected by the perturbations included in the model, since it is still able to reach the desired flight conditions required for the landing trajectory.

However, as for the random wind configuration, if we focus on figure 3.26 we can appreciate that the signal curvature is not as smooth as the signal in the clear sky: it clearly depicts the effect of the perturbation on the response stabilization. Nevertheless, as in the constant wind configuration case, the controller is able to handle these perturbations and drive the UAV to the glide path.

The effect of the perturbations can be also seen in the pitch rate response: if we compare the signal behavior obtained in clear sky flight (figure 3.7) with the signal in random wind perturbation (for instance, figure 3.24a, with I1 alignment), we can appreciate how the amplitude of the micro-oscillations that appeared in the pitch rate response of the clear sky configuration increases.

Indeed, we are able to see how the amplitude of such perturbations change when changing the alignment of the wind with respect to the UAV. For the alignment matrix I1 case, in which the wind and aircraft axes are completely aligned, the effect of the perturbation is much bigger and the distribution is much less uniform than for I6 case, in which the axes are complete and randomly unaligned. Matrices I2, I3, I4 and I5 cases show the intermediate steps between one extreme and the other: figure 3.24 depicts this progressive change.

To conclude, we could state that making use of the conventional method approach we obtained a controller that properly gets the UAV to follow the desired landing trajectory, helping to stabilize the response signal. Besides, it is strong and reliable enough to perform such action in the presence of atmospheric perturbation, no matter their constant or random nature.

4.1.2 Modern approach

In this section we will comment the results obtained applying the H_{inf} controller. The main challenge in this method, for us to obtain accurate results, is the correct definition of the plant model. In order to meet this requirement, for us it was necessary to include the altitude in the state vector, since the commanded input to be introduced in the system will concern this parameter.

We can observe the distance response extracted from the closed loop system obtained with the calculated controller in figure 3.31. The settling time is about 30 seconds, descending in 5 seconds to the desired glide path (of 12°, we must remind). This is a good and realistic result, given the initial conditions of flight of our drone.

If we change the initial distance to the glide path at which the UAV is flying, the response changes: the higher this value, the bigger the oscillation amplitude (which represents the distance below the glide trajectory reached by the drone when flying downwards in the stabilization process around the desired distance value), as we can appreciate in figure 3.32a.

If we increase the initial pitch angle at which the UAV is oriented, the tracking error increases and the signal shape gets worse, with initial small oscillations and abrupt descents (figure 3.33a).

Now, we will discuss the robustness study performed, which had the aim of obtaining the robustness characteristics of our system when introducing an uncertainty of ± 10% in the M_α, M_q and Z_α stability derivatives, considered to be the most critical in the longitudinal dynamics model definition.

We are given two different pieces of information: the summary of the robustness computation and the sensitivity of the robustness margin to each uncertain parameter in the model.

Regarding the robustness computation, we are told that our system is robustly stable for the defined uncertainties: it can cope with variations' values three times bigger than the specified one (this is, changing the stability derivatives up to ± 30% of the nominal value in the dynamics). Above this number, it will reach a destabilizing point, which causes an instability of 2.46 radians/sec.

As for the sensitivity information that we are given, it was determine that the most sensitive element was M_α (64%), followed by Z_α (7%) and M_q (0%). This conclusion was reached studying how the robustness margin decreased after increasing the three elements by 25%; it was obtained that the bigger decreased was reached with M_α, while with M_q there was no change.

Finally, the robustness study can be seen graphically in figure 3.34, which depicts the different responses

that we get when changing the nominal values of our stability derivatives by 10%. As it can be seen, the output response is slightly modified by these changes: the amplitude difference between the highest changes with respect the nominal values is smaller than 0.02 m, which actually shows the high robustness level of the obtained controller.

4.1.3 Comparison between both approaches

When comparing between the two controllers obtained, the main difference between both kind of approaches is clearly observed: the PID controller is the best in terms of stabilization performance and improvement, while the H_{inf} method assures the robustness of the controller for both stability and performance.

If we study the responses obtained of the distance to the glide path (for instance, figures 3.11 for the PID controller and 3.31 for the H_{inf}), we can appreciate the differences in the tracking error: from 0% error in the conventional approach to some centimeters in the modern.

Concerning the robustness, for the H_{inf} controller it was already commented the small deviation in the step response when changing the elements in the dynamics model, when compared with the nominal values.

However, the PID controller is not able to handle changes. We just need to see the differences between the output responses of the different glide angles: take for instance the signal corresponding to the slope of design 2.5° (figure 3.11), when compared with the signals corresponding to 7° (figure 3.12), 12° (figure 3.13) and 15° (figure 3.14). Uncertainties, inaccuracies and approximations in the model would affect the quality of the controller, making it less reliable than the one obtained following the H_{inf} method.

4.2 Future development

In this section we will talk about a future development of the work presented in this project, including some possible new objectives and improvements to be done.

One of the most straightforwards aspects would be the completion of the automatic landing system making use of the nonlinear dynamics of the drone: in this way we would obtain a more accurate controller, since we are taking the actual model of the UAV, and not the linear approximation. In case of the conventional approach, we are implementing the control system in Simulink, a powerful tool that perfectly allows us to work with non-linear models in the design phase. In case of the modern approach, we are obtaining a robust controller, so the use of a more precise and accurate model of the flying wing drone would make the controller obtained following this method extraordinarily potent, so it would be directly implemented on the aircraft dynamics and would support the inevitable deviations and uncertainties arisen from the model study obtaining.

Another important aspect to be taken into account would be the improvement of the McLean [21] pitch attitude control system with the inclusion of a saturation term, simulating the actuator surface limits: this element makes sure that the aircraft will be able to perform the control action with its available actuator surfaces. If we don't include the actuator deflection limits, we are in danger of designing a controller that, when implemented on the physical system, won't be correct, since it will stabilize the signal assuming an actuator position that cannot be reached by the real deflection mechanism (for instance, stabilizing the response for 30° when the limit is 20°, or even worse assumptions).

Future developments could also concern the completion of the autolanding trajectory with the inclusion of the *flare* stage controller. The flare stage consists of the maneuver developed by the aircraft right behind touching the ground in which it lands tangentially to the runway. We can find some examples of how to do it in the literature (take for instance pages 399 - 404 in McLean [21]).

Finally, we could try to implement both automatic landing systems developed in this project; first of all, simulating the drone flight with some kind of software (such as Flight Gear, that can be connected to

Simulink), where the behavior of the drone with each of the systems can be studied, and then physically to the actual drone, developing and improving the system measuring directly on the aircraft.

Appendix A

A.1. Calculations

```
%-----------------------------------%
% Longitudinal Dynamics Model %
%-----------------------------------%

A = [-1.3374e-2, 1.3589, 0, -9.07036;
     1.2592e-1, -1.1151e1, -5.0437e1, -1.4134;
     -7.2028, 1.3765e3, -3.5594e1, 2.6574e-1;
     0, 0, 1, 0];
B = [-6.4712e-3; 3.6575e-2; -7.7492; 0];
C = [0, 1, 0, 0; 0, 0, 0, 1; 0, 0, 1, 0];
D = [0; 0; 0];

% Constants in the autopilot structure
U_0 = 15; % flight velocity
h_0 = 10; % initial height
R = h_0/sind(2.5); % DME to runway, dependent with time
I = eye(3); % for dryden input

%-----------------------------------%
% Q loop - Obtention of K_Q %
%-----------------------------------%

% We obtain the Transfer Function
[num_q,den_q] = ss2tf(A,B,C(3,:),0);
H_Q = tf(num_q,den_q);

% Closed loop for K_Q
H_a = tf(10,[1 10]);
H_cl_Q = H_a * H_Q;
K_Q = -1;
sisotool(H_cl_Q, 1, K_Q, 1);

%-----------------------------------------------%
% Theta loop - Obtention of K_Theta and K_A %
%-----------------------------------------------%

[num_t,den_t] = ss2tf(A,B,C(2,:),0);
H_Theta = tf(num_t,den_t);
H_Theta_R2 = (H_Theta * H_a)/(1 + H_Q * H_a * K_Q);
```

```
% Approximate the system to one reduced
Num = [-77.49, - 4760, - 5.468e06, - 7.968e07, - 2.55e08, - 1.014e07,
       - 3.119e06, - 9948];

r_Num = roots(Num);

r_Num_sign = 100*[-0.0002 + 0.0011i, -0.0002 - 0.0011i, -0.0000 + 0.0000i];

Num_Red = poly(r_Num_sign);

Den = [1, 113.5, 1.439e05, 9.385e06, 5.026e09, 9.857e10, 4.944e11, 3.48e10,
       1.257e10, 4.103e08, 7.324e07];

r_Den = roots(Den);

r_Den_sign = 100*[ -0.0002 + 0.0011i, -0.0002 - 0.0011i, -0.0002 + 0.0011i,
                   -0.0002 - 0.0011i];

Den_Red = poly(r_Den_sign);

H_Theta_R_Reduced = tf(Num_Red, Den_Red);

% We build the Routh - Hurwitz matrix
syms K_A K_Theta
[N,D] = tfdata(H_Theta_R_Reduced);
RH = D + K_A*K_Theta*N;

c0 = 1; % t^4
c1 = K_A*K_Theta + 2/25;
c2 = (K_A*K_Theta)/25 + 133/5000;
c3 = (K_A*K_Theta)/80 + 1/1000;
c4 = 1/6400; % t^0

% Study the sign of the major minors
RH1 = c0;
RH2 = [c0 c2; c1 c3];
RH3 = [c0 c2 c4; c1 c3 0; 0 c0 c2];

d_RH1 = det(RH1);
d_RH2 = det(RH2);
d_RH3 = det(RH3);

d_RH2_K = subs(d_RH2, K_Theta, 10);
d_RH3_K = subs(d_RH3, K_Theta, 10);

KA2 = solve(d_RH2_K, K_A);
vpa(KA2,10);
KA3 = solve(d_RH3_K, K_A);
vpa(KA3,10);

d_RH2_F = subs(d_RH2_K, K_A, -6);
d_RH3_F = subs(d_RH3_K, K_A, -6);
```

```
%-----------------------------------------%
% The whole loop - Obtention of K_C %
%-----------------------------------------%

K_C = 17; % determined studying the loop
```

A.2. Simulink values

```
%-----------------------%
% State Space model %
%-----------------------%

A = [-1.3374e-2, 1.3589, 0, -9.07036;
     1.2592e-1, -1.1151e1, -5.0437e1, -1.4134;
     -7.2028, 1.3765e3, -3.5594e1, 2.6574e-1;
     0, 0, 1, 0];
B = [-6.4712e-3; 3.6575e-2; -7.7492; 0];
C = [0, 1, 0, 0; 0, 0, 0, 1; 0, 0, 1, 0];
D = [0; 0; 0];

%-------------------------------------------------%
% Constants in the autopilot architecture %
%-------------------------------------------------%
G_S = 15; % in degrees: other values used 2.5, 12
U_0 = 15; % flight velocity, m/s
h_0 = 10; % initial height, m
R = h_0/sind(G_S); % DME to runway, dependent with time, m
rad2deg = 180/pi;
U_wg = 5; % constant wind velocity in the x axis, m/s
W_wg = 5; % constant wind velocity in the z axis, m/s

% Wind turbulence orientation with respect to the drone
I1 = eye(3); % aligned
I2 = -1*eye(3); % aligned, opposite sense
I3 = [1, 0, 0; 0, 0.1, 0.8; 0, -0.7, -0.2]; % aligned x axis
I4 = [0.5, 0, 0.8; 0, 1, 0; -0.4, 0, -0.2]; % aligned y axis
I5 = [0.5, 0.1, 0; -0.4, -0.7, 0; 0, 0, 1]; % aligned z axis
I6 = [-0.1, 0.4, -0.3; 0.5, -0.8, -0.7; 0.2, -0.6, 0.9]; % random

%---------------%
% Gain values %
%---------------%

K_Q = -1;%-1;
K_Theta = 1;%1;
K_A = -3; %-6; -2; -3
K_C = 4;%17; 10; 4
```

A.3. Plot of the results (exemplified for some results of random wind flight)

```
figure(1)
plot(Q_C_RW(:,2));
xlabel('Time (s)');
ylabel('Pitch rate (deg/s)');
title('Pitch rate response, random wind configuration - I6 alignment');
xlim([0,7050]);
xt = get(gca, 'XTick');
set(gca, 'XTick',xt, 'XTickLabel',xt*1E-2);

figure(2)
plot(Theta_C_RW(:,2));
xlabel('Time (s)');
ylabel('Pitch angle (deg)');
title('Pitch angle response, random wind configuration - I6 alignment');
xlim([0,9500]);
xt = get(gca, 'XTick');
set(gca, 'XTick',xt, 'XTickLabel',xt*1E-2);

figure(3)
plot(Reference_RW(:,2));
title('Distance to the glide path response, random wind configuration -'
'12deg, I6 alignment');
xlabel('Time (s)');
ylabel('Vertical distance to the glide path (m)');
hold on
plot(D_C_RW(:,2));
xlabel('Time (s)');
ylabel('Vertical distance to glide path (m)');
legend({'Reference input','Vertical distance to the glide path response'});
xlim([0,9000]);
xt = get(gca, 'XTick');
set(gca, 'XTick',xt, 'XTickLabel',xt*1E-2);

figure(4)
plot(velocities(:,2))
xlabel('Time (s)');
ylabel('Velocity (m/s)');
hold on
plot(velocities(:,3))
legend({'x component','y component'});
xlabel('Time (s)');
ylabel('Velocity (m/s)');
title('Longitudinal components of the simulated random wind velocity -'
'I6 alignment');
xlim([0,1350]);
xt = get(gca, 'XTick');
set(gca, 'XTick',xt, 'XTickLabel',xt*1E-2);
```

Appendix B

B.1. H infinity controller

```
%-------------------------------------------------------%
% Input and output vectors in our configuration %
%-------------------------------------------------------%

% u1 = [n_wg, h_c, n_s]
%       turbulence noise n_wg = [n_u, n_w]
%       sensor     noise n_s  = [n_alpha, n_q, n_theta, n_h]
% y1 = [z'_e, delta]
%       error to minimize z'_e = [error_h]
% u2 = delta_c -> control input
% y2 = [alpha, q, theta, h] + n_s

%-----------%
% Constants %
%-----------%

U_0 = 15; % m/s
h_0 = 7; % m
G_S = 12; % deg
h_dot_c = U_0*sind(G_S);
feet2m = 0.3048;
deg2rad = pi/180;

% Values to consider when building the matrixes
n_actuators = 1; % number of actuators in your a/c
n_minimize = 1; % number of variables to minimize the error

%-------------------------------------%
% Determination of the Plant Model %
%-------------------------------------%

A_S = [-1.3374e-2, 1.3589, 0, -9.07036;
      1.2592e-1, -1.1151e1, -5.0437e1, -1.4134;
      -7.2028, 1.3765e3, -3.5594e1, 2.6574e-1;
      0, 0, 1, 0];
B_S = [-6.4712e-3; 3.6575e-2; -7.7492; 0];

% Introduce the height in the dynamics state
h_dot = [0, -U_0*deg2rad, 0, U_0*deg2rad, 0];
A = [A_S, zeros(4,1); h_dot];
```

```
B = [B_S; 0];
C = [0, 1, 0, 0, 0; 0, 0, 1, 0, 0; 0, 0, 0, 1, 0; 0, 0, 0, 0, 1];

% And we obtain the plant model:
[T,A_wp,B_wp,C_wp,D_wp] = PlantModel(A,B,C,10,5.7385*feet2m,3.9494*feet2m,
105.7414*feet2m,10.4003*feet2m,U_0,1,0.99,1,1,1);

% And we calculate the control law to be used.
nmeas = 4; % number of measurement outputs
ncont = 1; % number of control inputs

[K,CL,GAM,INFO] = hinfsyn(T,nmeas,ncont);

%-----------------------------------------%
% Plot the height tracking error response %
%-----------------------------------------%

% Time vector
t=0:0.1:50;
x0h=zeros(20,1); % initial conditions vector
x0h(5) = 7; % height initial condition
x0h(4) = 12; % pitch angle initial condition


G_wp=ss(A_wp,B_wp,C_wp,D_wp);
G_wpinv=inv(G_wp); % to be multiplied so we obtain the real error

% x_CL = [x_T x_K], u_CL = u1, y_CL = y1
sys_h = ss(CL.a,CL.b(:,3),CL.c(1,:),CL.d(1,1));
epshponderat=lsim(sys_h,zeros(501,1),t,x0h);

% system response to plot
epsh=lsim(G_wpinv,epshponderat,t);

% Plot and graph configuration
figure(1)
Plot1 = plot(t,epsh);
title('Height tracking error response plot: 12deg, 7 m')
xlabel('Time (s)');
ylabel('Vertical distance to glide path (m)');
grid
```

B.3. Plant Model function

```
function [T,A_wp,B_wp,C_wp,D_wp] = PlantModel(A,B,C,w_delta,sigma_u,
sigma_w,L_u,L_w,V_a,omega1,omega2,Kp,n_actuators,n_minimize)

%% Actuator model %%
A_A = -w_delta * eye(n_actuators);
B_A = w_delta * eye(n_actuators);
C_A = eye(n_actuators);

%% Wind turbulence model(Dryden Model) %%
```

```
% For h_0 = 20.0131 ft

A_wg = [-V_a/L_u, 0, 0;
        0, -V_a/L_w, 0;
        0, (sigma_w/sqrt(L_w))*sqrt(3*V_a/(2*pi)), -V_a/L_w];

B_wg = [(sigma_u*1/sqrt(L_u))*sqrt(V_a/pi) 0;
        0 (-V_a/L_w)*(1-1/sqrt(3));
        0 (sigma_w/sqrt(L_w))*sqrt(3*V_a/(2*pi))];

C_wg = [1, 0, 0; 0, 0, 1];

%% Weighting model %%
% We essay until having a good response. We start using the model in
% Belcastro's paper, until we finally obtain:

% Defining and building the transfer function
syms s
H_opt = Kp*(s+omega1)/(s + omega2);
[H_opt_num,H_opt_den] = numden(H_opt);
TFnum = sym2poly(H_opt_num);
TFden = sym2poly(H_opt_den);
[awp,bwp,cwp,dwp]=tf2ss(TFnum,TFden);

% And now the matrixes
A_wp = awp * eye(n_minimize);
B_wp = bwp * eye(n_minimize);
C_wp = cwp * eye(n_minimize);
D_wp = dwp * eye(n_minimize);
C_ze = [0 0 0 0 1]; % change it with the variables commanded

%% Build the T matrix to be used, having the shape: %%
%  T = [Ag B1  B2
%       C1 D11 D12
%       C2 D21 D22];

% Need to change the zeros wrt the matrixes
a = length(A);
b = length(A_wg);
c = length(A_A);
d = n_minimize;

% We define the matrices to determine T:

A_g = [    A          B*C_A       zeros(a,b)      zeros(a,d);
       zeros(c,a)      A_A        zeros(c,b)      zeros(c,d);
       zeros(b,a)   zeros(b,c)      A_wg          zeros(b,d);
       B_wp*C_ze    zeros(d,c)  zeros(d,b)          A_wp   ];

B_1 = [zeros(a,2)   zeros(a,d)    zeros(a,5) ;
       zeros(c,2)   zeros(c,d)    zeros(c,5) ;
         B_wg       zeros(b,d)    zeros(b,5) ;
       zeros(d,2)     -B_wp       zeros(d,5)];
```

```
B_1 = B_1 + 1e-2*ones(size(B_1));

B_2 = [zeros(a,c) ;
          B_A     ;
       zeros(b,c) ;
       zeros(d,c)];

C_1 = [ D_wp*C_ze   zeros(d,c)     zeros(d,b)      C_wp    ;
       zeros(c,a)     0.01*C_A       zeros(c,b)   zeros(c,d)] ;

D_11 = [zeros(d,2)     -D_wp         zeros(d,5) ;
        zeros(c,2)   zeros(c,d)     zeros(c,5)] ;

D_12 = [zeros(c,c);
        zeros(d,c)];

C_2 =  [C zeros(4,c) zeros(4,b) zeros(4,d)];

D_21 = [zeros(4,2)   zeros(4,d) 0.01*eye(4,5)];

D_22 = zeros(4,c);

T_matrixForm = [A_g, B_1, B_2; C_1, D_11, D_12; C_2,  D_21, D_22];

% We convert it into steady state representation
T=ss(A_g,[B_1 B_2],[C_1;C_2],[D_11 D_12;D_21 D_22]);
end
```

B.3. Robustness study

```
%-------------------------------%
% Undetermined Plant Model %
%-------------------------------%

Z_alpha = ureal('Z_alpha',-1.1151e1,'Percentage',[-10 10]);
M_alpha = ureal('M_alpha',1.3765e3,'Percentage',[-10 10]);
M_q = ureal('M_q',-3.5594e1,'Percentage',[-10 10]);

A_S = [-1.3374e-2, 1.3589, 0, -9.07036;
     1.2592e-1, Z_alpha, -5.0437e1, -1.4134;
     -7.2028, M_alpha, M_q, 2.6574e-1;
     0, 0, 1, 0];
B_S = [-6.4712e-3; 3.6575e-2; -7.7492; 0];

% Introduce the height in the dynamics state
h_dot = [0, -U_0*deg2rad, 0, U_0*deg2rad, 0];
A = [A_S, zeros(4,1); h_dot];
B = [B_S; 0];
C = [0, 1, 0, 0, 0; 0, 0, 1, 0, 0; 0, 0, 0, 1, 0; 0, 0, 0, 0, 1];

[T,A_wp,B_wp,C_wp,D_wp] = PlantModel(A,B,C,10,5.7385*feet2m,3.9494*feet2m,
105.7414*feet2m,10.4003*feet2m,U_0,1,0.99,1,1,1);
```

```
%% Interconnection of the controller and the plant model %%
ClosedLoop = lft(T,K);

% Robustness analysis
opt = robOptions('Display','on','Sensitivity','on');
[StabilityMargin,wcu] = robstab(ClosedLoop,opt);

%----------------%
% Results plot %
%----------------%

% Time vector
t=0:0.1:50;

G_wp=ss(A_wp,B_wp,C_wp,D_wp);
G_wpinv=inv(G_wp); % to be multiplied so we obtain the real error

% x_CL = [x_T x_K], u_CL = u1, y_CL = y1
sys_h = ss(ClosedLoop.a,ClosedLoop.b(:,3),ClosedLoop.c(1,:),ClosedLoop.d(1,1));
epsh = series(G_wpinv,sys_h);
epsh_nom = epsh.NominalValue;

% Plot the step response of the system
step(epsh,'b',epsh_nom,'r');
legend({'Deviation values responses','Nominal values response'});
xlabel('Time');
ylabel ('Amplitude (m)');
title('Responses set for a step signal input');
```

References

[1] Larson, C. G. *https://www.airspacemag.com/history-of-flight/the-first-autolanding-3818066/* Consulted on 16 October 2018

[2] Astrom, K. J. *Automatic Control - The Hidden Technology* Poul M. Frank, Springer, 1999.

[3] Mackinson, W. *All - Weather Landings: Air and Ground Research by the Blind Landing Experimental Unit.* Flight, 18 December 1953.

[4] Siegel, D., Hansman, R. J. *Development of an autoland system for general aviation aircraft* MIT International Center for Air Transportation, report No ICAT - 2011 - 09, pp. 21 - 22, September 2011.

[5] *Unmaned Aircraft Systems (UAS)* International Civil Aviation Organization ICAO, Cir. 328, pp. 3, 2011.

[6] Flight Literacy. *https://www.flightliteracy.com/approaches-part-twenty-five/* Consulted on 16 October 2018

[7] Prisacariu, V.. *The History and the Evolution of UAVs from the beginning till the 70s* Journal of Defense Resources Management, Vol. 8, Issue 1 (14)/2017, 2017.

[8] Keane, J. F., Carr, S. S. *A Brief History of Early Unmanned Aircraft* John HIpkins APL Technical Digest, Vol. 32, Nr. 3, 2013

[9] Shaw, I. G. R.. *https://understandingempire.wordpress.com/2-0-a-brief-history-of-u-s-drones/* Consulted on 16 October 2018

[10] Pike, J.. *https://fas.org/irp/program/collect/pioneer.htm* Consulted on 16 October 2018

[11] Justin, F.. *https://www.dronethusiast.com/history-of-drones/* Consulted on 16 October 2018

[12] Zickuhr, K., Stahl, E. *Cities and Drones: what cities need to know about Unmanned Aerial Vehicules (UAVs)* National League of Cities (NLC), 2016

[13] Egan, P. *https://air-vid.com/20-great-uav-applications-areas-drones/* Consulted on 16 October 2018

[14] *https://www.hubsanus.com/shop/h301s-spy-hawk.html* Consulted on 16 October 2018

[15] DJI *https://www.dji.com/phantom-4-pro* Consulted on 16 October 2018

[16] Gordon, K. *https://www.statista.com/topics/3601/commercial-uavs/* Consulted on 16 October 2018

[17] Air Drone Craze *https://airdronecraze.com/drone-tech/* Consulted on 16 October 2018

[18] Parrot *https://www.parrot.com/global/drones/parrot-disco-fpv-parrot-disco-fpv* Consulted on 16 October 2018

[19] Horizon Hobby *https://www.horizonhobby.com/opterra-2m-wing-bnf-basic-p-efl11150* Consulted on 16 October 2018

[20] Cook, M. M. *Flight Dynamics Principles: a Linear Systems Approach to Aircraft Stability and Control* Butterworth - Heinemann, Oxford, 2007

[21] McLean, D. *Automatic Flight Control Systems* Prentice Hall International, Hertfordshire, 1990

[22] Stoica, A. M., Parvu, P. V., Ene, C. *An H_{inf} design method for the pitch attitude hold autopilot of a flying UAV* Universitatea Politehnica din Bucuresti, 2016

[23] Belcastro, C. M., Chang, B. -C. *A Problem Formulation for Glideslope Tracking in Wind Shear Using Advanced Robust Control Techniques* National Aeronautics and Space Administration (NASA), 1992

[24] Hakim, T. M. I., Arifianto, O. *Implementation of Dryden Continuous Turbulence Model into Simulink for LSA - 02 Flight Test Simulation* Journal of Physics: Conference Series, 2018