

POLITECNICO DI TORINO

MASTER's Degree in MECHATRONIC ENGINEERING



MASTER's Degree Thesis

**6-DoF MODELLING OF A WIG
CRAFT IN GROUND EFFECT**

Supervisors

Sergio Dominguez Cabrerizo
Fabrizio Dabbene

Candidate

Daide Patria

Contents

1	Introduction	6
1.1	Objective of the present work	6
1.2	Generalities on WIG crafts	7
1.3	Different types of low resistance vehicles	10
1.3.1	Hovercraft	10
1.3.2	Hydrofoil	10
1.3.3	Ekranoplans	11
1.4	Different types of ekranoplans	13
1.4.1	WIG	14
1.4.2	DACC	14
1.4.3	PARWIG	15
1.5	Dynamic and aerodynamics	16
1.5.1	Used method for aerodynamic forces	21
1.5.2	An especially interesting method for the estimation of the aerodynamic coefficients	24
1.6	Control	26
1.6.1	MPC	28
1.6.2	Adaptive controller L1	30
2	Dynamics equations	32
2.1	Generic dynamical equations	32
2.2	Definition of reference frames	32
2.2.1	Forces and momenta equations	34
3	Aerodynamics	36
3.1	CFD simulations	36
3.1.1	Interpolation of the data	42
3.2	Control surfaces expression	44

4	Control	46
4.1	Non-Linear Model Predictive Control	47
4.1.1	Altitude variation with direct inputs	48
4.1.2	Altitude variation with control surfaces and motor as inputs	53
4.1.3	Effect of mass on MPC performances	57
4.1.4	Cruising from an equilibrium position	58
4.1.5	Recovering from a perturbed position	59
4.2	Feedback linearisation	61
4.2.1	Linearisation process	62
4.2.2	LQR controller over mixed linearised model	63
4.2.3	Updating LQR over feedback linearised model for alti- tude change	66
4.2.4	Updating LQR for recovery from perturbed position . .	69
4.3	Comparison between the employed control strategies	71
4.3.1	Comparison between MPC, uLQR and LQR for recov- ery from perturbed position	71
4.3.2	Comparison between MPC and LQR for altitude change	72
5	Conclusions and future work	74

List of Figures

1.1	<i>comparison of payload efficiency of different vehicles [40]</i>	7
1.2	<i>the AirFish 8 cruising</i>	9
1.3	<i>BHC AP-188 hovercraft</i>	10
1.4	<i>The Boeing 929 Jetfoil Waterjet-Propelled Hydrofoil</i>	11
1.5	<i>The Caspian sea monster</i>	12
1.6	<i>The Alekseev A-90 Orlyonok ekranoplan</i>	13
1.7	<i>the Lippisch RFB X-113</i>	14
1.8	<i>The Russian Volga-2 [19]</i>	15
1.9	<i>different phases modes of a parwig [18]</i>	16
1.10	<i>Effect of the presence of ground on the wing span</i>	17
1.11	<i>angle of attack definition</i>	17
1.12	<i>visualisation of the image method: the image is used instead of the ground to simulate the ground effect</i>	18
1.13	<i>functional components of the ekranoplan</i>	19
1.15	<i>View of the lines casting, one of the steps of the simulation</i>	22
1.16	<i>attitude vector λ and the characterising angles as in the formula [10]</i>	24
1.17	<i>pure pursuit guidance logic [16]</i>	27
1.18	<i>Cascade control structure for position based visual servoing of an aircraft [17]</i>	29
1.19	<i>control scheme of an L1 adaptive control [15]</i>	31
2.1	<i>visualization of the angles between vehicle carried frame and body frame [1]</i>	33
2.2	<i>the aerodynamic frame F_a with respect to the body frame F_b [3]</i>	34
3.1	<i>lift and drag forces as speed and height vary, at $\theta = 10^\circ$</i>	37
3.2	<i>variation of drag and lift as speed and roll vary</i>	38
3.3	<i>variation of drag and lift as speed and yaw vary</i>	39
3.4	<i>variation of drag and lift as speed and pitch vary</i>	39
3.5	<i>drag and lift coefficients computed by the CFD in use for $\frac{h}{c} = 2, 24$</i>	40
3.6	<i>coefficients multiplied by a 3 factor</i>	41
3.8	<i>variation of torques as speed and altitude vary</i>	42

3.9	<i>obtained gain function, calculated points and interpolating polynomial . . .</i>	43
3.10	<i>control surfaces in an aircraft</i>	45
4.1	<i>full dynamic for altitude change at horizon 100</i>	49
4.2	<i>pitch dynamic with different time horizon values</i>	50
4.3	<i>objective function with calibrated weights</i>	51
4.4	<i>dynamic for altitude change with uniform measurement noise of $\pm 5\%$. . .</i>	52
4.5	<i>comparison between the regular case and the one with noise</i>	53
4.6	<i>altitude change with full control surfaces input</i>	54
4.7	<i>obtained function, interpolated and scattered values for the vertical equilibrium position</i>	55
4.8	<i>altitude change with added bounds on pitch</i>	56
4.9	<i>comparison between different masses</i>	57
4.10	<i>Stationary cruising from equilibrium position</i>	58
4.11	<i>steady state reaching from a random unsteady position</i>	59
4.12	<i>comparison between the cases with and without noise for the MPC recovery from a perturbed position</i>	60
4.13	<i>feedback linearisation block scheme</i>	62
4.14	<i>LQR control on linearised system, starting from perturbed position</i>	65
4.15	<i>comparison between the cases with and without noise for the LQR for recovery from perturbed position.</i>	66
4.16	<i>altitude change using LQR</i>	67
4.17	<i>altitude change using LQR, with measurement noise</i>	68
4.18	<i>LQR for altitude change: comparison between the behaviour with and without noise</i>	69
4.19	<i>LQR controller for recovery from perturbed position</i>	70
4.20	<i>comparison between the noisy and regular case for refreshing LQR</i>	70
4.21	<i>comparison between the altitude and pitch dynamic for LQR, updating LQR and non-linear MPC</i>	72
4.22	<i>comparison between non-linear MPC and uLQR for altitude change</i>	73

Acknowledgements

Working on this thesis has been a long a difficult journey, and it wouldn't have been possible without the help of my team-mates, that helped me by infusing the needed perseverance needed to tackle such an hard and unknown world that the aerospace one is to me.

I would like to thank professor Sergio Dominguez Cabrerizo for his constant supervision and help, that was always promptly offered and never had to be asked for twice.

A special thank to Miguel Garcia Domínguez, with whom many hours I have spent analysing the bibliography trying to find a way and many debugging hours, helping and getting mad at each other, overcoming the sense of impotence that sometimes overcome during the development of the project, always fruitfully.

Thanks to my long-time girlfriend Laura, that never stepped back before the hard task of boosting my mood.

Chapter 1

Introduction

1.1 Objective of the present work

This master thesis is part of a project that seeks to build an autonomous electrically powered ekranoplan. The project was started from scratch by Alberto Calvo Cordoba [24], that made the first steps by developing a 1-DoF with constant acceleration model, implementing a PID controller and evaluating its performances over a linear model.

The present work is focused on the same reference ekranoplan with the objective of building a 6-DoF model. This model would later be used to make a closed-loop simulator, where non-linear control techniques will be applied for different purposes, such as seakeeping and basic manoeuvres. This will allow to include all the different aspects of the real dynamic, removing the constraints on the possible movement and allowing arbitrary manoeuvres to be done, making path following possible.

It will be further developed and used as a benchmark for the next stages of the project. In particular, the advancements needed are the use of a wind tunnel to validate the results obtained through computation, and finer CFD simulations that can be used to obtain a truthful model of the aerodynamic forces. Since the final objective includes autonomy, path planning and non-linear robust controllers will be investigated.

The current model is built using a custom computational fluid dynamics tool and the results are used as representative of the behaviour of aerodynamic forces in proximity to ground. The main aim of the control part is to show if a given strategy is feasible and applicable to real scenarios. The obtained results are not considered final assessments of the used strategy, but rather as a primary investigation on the topic. This makes especially sense given the lack of existing works on such topic, and the need for a general

view on ekranoplans that can shine a light on this still quite unknown world.

1.2 Generalities on WIG crafts

Most non-military ekranoplans are designed to operate in the range of 60 to 200 knots. This to make them closer to marine craft than aircraft given that WIG crafts have a payload close to fast marine ships. In spite of this, the cost rises significantly with parameters like the cruise speed, since the design would need to include more technology from aircraft thus making the process more expensive.

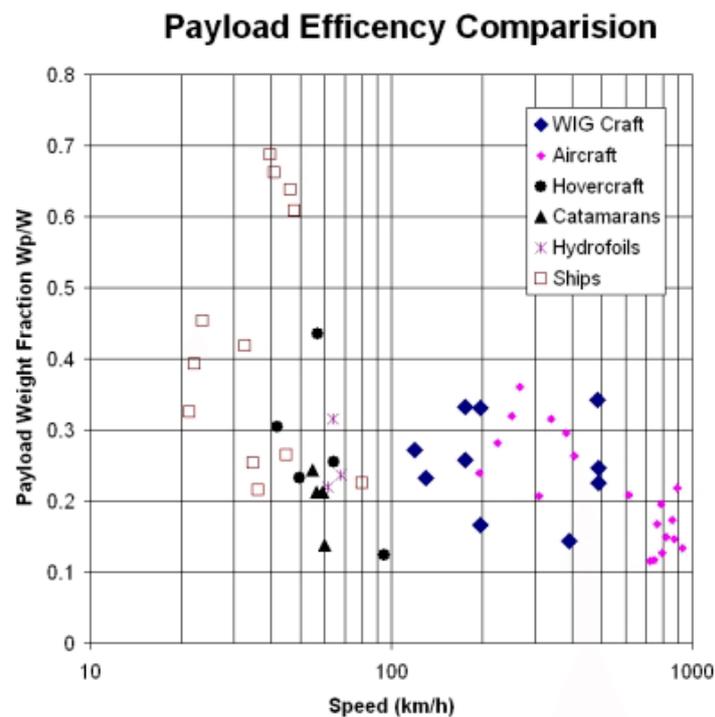


Figure 1.1: *comparison of payload efficiency of different vehicles [40]*

The graph above shows that WIG crafts are usually faster than other low-resistance vehicles with a comparable payload. The speeds reachable by an ekranoplan are significantly higher than other boats. Adding this on top of energetic efficiency leads to an interesting opportunity for commercial purposes as a replacement for cargo ships and other traditional means of maritime transportation.

The graph seems to give a small edge to aircrafts when considering speed and payload. What it doesn't show is that WIG craft have a much improved *lift-drag*, making them much more efficient energetically.

Despite being very sensitive to meteoceanic condition for the manoeuvres of take-off and landing, the cruise speed of ekranoplans isn't affected much by the wave conditions once the low height flight mode is reached.

Currently the development of a vehicle like this cannot ignore important aspects such as the ecological and sustainability ones. It is important to keep in mind such objective for the design, since nowadays these are requirement as important as low operational cost.

Some of the key objectives for a future proof and capable design are:

- Reduction of pollutants emission. This has a high margin, given that ships are responsible for more than 18 percent of some air pollutants [2]. It can be achieved using renewable sources of energy and powering the vehicle electrically through an appropriate power provision system.
- Making the vehicle a viable alternative to land transportation, which would reduce the distances and, given the theoretical maximum speeds, the time, thus reducing the cost.
- Giving particular attention to the security part, having to deal partially with the stigma coming from early problems with ekranoplans. New developments on computation and control theory can help tremendously with security issues, greatly reducing accidents possibilities.

This is why this project focuses on marine transportation, where innovations are more rare and there is hope that, given the numbers, ekranoplans might replace some of current means of transportation. This is here the high speed in conjunction with payload would be revolutionary, both for passengers and goods transportation.

Some of the issues related to the project are:

- High speed means shorter reaction time available for safety manoeuvres, such as obstacle avoidance. Controllers aimed at this must have a particularly high computational capabilities compared to much lower vehicles. Moreover this technologies are still under heavy development to the day, even from by companies that see the selling point and are investing a lot in it.
- Finding a suitable way to electrically power the vehicle, giving it enough power and autonomy. Batteries are of course coming into play, but

they create weight and storing problem, also considering the extreme environment of salt-water, which is extremely corrosive and requires special design care.

- The design of WIG crafts is not standardised. Considering the case of automotive and aerospace vehicle design it is quite clear how lacking of norms it is.
- The stability and controllability of ekranoplans are quite complex and require special attention during the design procedure.

The only commercially available ekranplan as of today is the “AirFish 8” produced by the Singaporean company “Wiget Works” [26].



Figure 1.2: *the AirFish 8 cruising*

The project started with the analysis of this vehicle. Its characteristics are quite impressive: powered by a car petrol V8 engine, with a power of 500CV, with a consumption of 70 litres per hours in normal working conditions. It also has a payload of 1 Mg and can carry up to 8 passengers, in addition to a pilot and a copilot.

It needs 500 meters for take-off and between 300 and 500 to land. Its maximum height is $7m$ and the turning radius $150m$.

These numbers allows for a quantification of the objective of the work, given that something similar to a Caspian sea monster is unachievable with limited resources and was economically justified by its military use.

1.3 Different types of low resistance vehicles

Ekranplanes are a relatively new type of marine craft, that was firstly developed during the 1960s, during an high tension era, the cold war in fact, by the Russian Navy. They are also referred to as “wing-in-ground effect craft” or “WIG craft” for short.

The key to marine craft development relies into the ability to reduce hydrodynamic resistance acting on the body of the craft. Since hydrodynamic resistance increases dramatically with speed, it follows a quadratic law, one of the main approaches is to reduce the surface of the boat that stays in contact with water during motion. The other important kinds of vehicles that exploit such concept in different ways are explained to give a better idea on the broader topic of low resistance vehicles.

1.3.1 Hovercraft



Figure 1.3: *BHC AP-188 hovercraft*

One example of how such dissipations is mitigated are hovercrafts, also called *air cushion vehicle* or ACV.

An air cushion is created underneath the vehicle by pumping air through fans and it is contained in a skirt. During operation the ACV is almost isolated from the water surface by the skirt, making it able to even move on land, given that it also lacks underwater appendages.

The main disadvantage with this type of vehicles is the load capacity and the consumption, since the skirt component can cause significant reduction of operational speed by increasing the drag interacting with water. The operational speed greatly varies with different wave conditions because of the soft spring effect of the skirt, going from 60 knots in calm water to 30 or 40 in waves. Its maximum load is rather low compared to other kind of vehicles.

1.3.2 Hydrofoil

Another special vehicle that uses special means in order to reduce the friction caused by water is the *hydrofoil*. This type of vehicle operates above the water surface by having foils suspended beneath the hull that act like an aircraft

wings in the water. Still, the foils suffer from water friction given that they are submerged and in contact with water. The cavitation phenomenon, which happens on the upper surface of the foils at high speed, is the limiting factor for the carrying capacity and top speed of the vehicle.

The most advanced model built so far is represented by the *Jetfoil*, developed by Boeing in the United States. It is still affected at its top speed by the cavitation, that is 50 knots.



Figure 1.4: *The Boeing 929 Jetfoil Waterjet-Propelled Hydrofoil*

1.3.3 Ekranoplans

Ekranplanes take this concept to another level by mixing the air cushion effect and the reduction of surface, on different scales based on the type of WIG craft. The concept ground effect is well known to aeroplane pilots, that experience such phenomenon when flying close to the ground, and are object of studies in order to obtain a model that can help correcting the aeroplane behaviour in such condition.

It consists in the creation of a load-carrying air cushion under their wings when operating at a small distance from the water surface, greatly increasing performances. In particular the top speed can go well above $100 \frac{km}{h}$. The geometry is specifically designed to enhance the ground effect, hence making the craft able to reach such high speeds thanks to a drag reduction.

The big advantage over seaplanes and flying boat is the payload, which for wig crafts is closer to the marine crafts than aeroplanes. They were believed to be the future replacement to cargo-ships and ferries, but huge inherent issues forced many to desist.

These vehicles have 4 different operational modes: floating hull, cushion, planning and air-borne modes. The design is rather complex compared to

other marine craft given also that the equilibrium is strongly influenced by the flying height and the pitch angle.

The recent regulations on ekranoplans classify them as *high speed marine crafts* [18], making the certification process less expensive overall, encouraging the commercialisation.

Ekranoplans were solely developed in military environments in the past, so the project was classified. Nowadays they are seeing a comeback as a viable and advantageous solution for marine transportation, given their unique characteristics.

In the past, given the lack of computational power and the extreme complexity of the fluidodynamic problem of the ground effect, they were put aside.



Figure 1.5: *The Caspian sea monster*

The greatest example of early age ekranoplan is the “Caspian monster”, developed by Alexeyev, the Russian father of ekranoplans, who developed the theoretical aspects. It was called the monster because of its dimensions: 37,6m of wingspan and 92m of length, maximum speed over $470 \frac{km}{h}$. It was able to accommodate 900 marines, cruise at an optimal height was between 4 and 14m and the take-off speed was below $140 \frac{km}{h}$. It totally lacked automatic controls and therefore needed manual actuation for the complex manoeuvring procedures. The first test was conducted 18 October 1966. The aforementioned characteristics were confirmed. The vehicle itself was capable of a complete 360° turn by banking.

Extreme case scenarios were applied during some tests, by direct order of Alexeyev, in order to prove the ekranoplan capabilities and give confidence to frightened pilots.

The main risk for an ekranoplan is to touch the water surface, which could cause a rapid deceleration and cause damages.

The *Orlyonok's accident* is the most famous accident that created a halo of fear around ekranoplans. The prototype of the Orlyonok was completed

in 1974. The sea trial were done in hurry, given the pressure from the Navy that wanted a fast delivery of the finalised product. The test area was still swelling because of a storm that had passed through several days before. The length of the swell was about the same as the vehicle.

After some successful runs the craft hit a wave, causing strong damages. A later inquiry indicated Alexeyev's design as inadequate, since the broken part was "too weak" according to the commission. Alexeyev insisted that the real cause of the accident was not his design but rather bad manoeuvring, but was dismissed and his assignment got changed.

The vehicle kept being produced until 1993, even though in 1985 the largest amount of funds from the USSR was moved to the production of the more requested nuclear submarines.



Figure 1.6: *The Alekseev A-90 Orlyonok ekranoplan*

After the disintegration of the Soviet union the lack of funds made the Russian navy unable to keep developing WIG crafts.

1.4 Different types of ekranoplans

A brief summary of the different names used and their description:

- **WIG or Ekranoplan:** Generic name, also applied to craft without special lift enhancement features
- **DACC or GEM:** Craft operating very close to the ground in the strong surface effect region
- **PARWIG:** Craft operating very close to the ground in the strong surface effect region

- **DACWIG:** Craft operating at a larger flying height in the surface effect region and that use air from bow-mounted engine, and wing endplates to create an air cushion under the wings at low speed rather than just enhanced lift

1.4.1 WIG

This is the simplest configuration. It lacks any lift augmentation system, and uses only side plates or buoys, making it suited for lower cruise speeds and load density. This configuration is the best for small WIG crafts for passengers, commercial or utility task.

The Airfish-8 belongs to this category, characterised by 1000kg of useful load and maximum 10 passengers.

This kind of WIG is designed for slow take off and cruise speed, and are also fairly stable during flight.

The most important model, especially historically speaking, is the *Lippisch's craft*, whose configuration is quite similar to the Airfish-8. It was firstly designed and built by Alexander Lippisch in Germany, in the 1960s and 1970s. Is is shown in the figure below.



Figure 1.7: *the Lippisch RFB X-113*

1.4.2 DACC

The DACC is characterised by a high static lift-thrust ratio. This is created by the means additional forward mounted air propellers that blow air directly in the cushion zone, increasing the pressure and allowing the vehicle to statically float.

The wings can be two on each side, in tandem with a large cushion. The craft also has deepened sidewalls or buoys at both sides of the main wing, making it easier to take off into surface effect flight.



Figure 1.8: *The Russian Volga-2* [19]

Moreover the high lift-thrust is able to create daylight clearance under the plane in a similar way to amphibious ACVs. DACCs are therefore able to take off at low speed on calm water surfaces and have amphibious capabilities.

The inherent advantages of this type are related to its flight stability that leads to better handling, it has a low environmental impact in terms of wave movement creation and flora and low craft capital cost.

1.4.3 PARWIG

This configuration makes use of auxiliary systems in order to create a dynamic air cushion. Such cushion can be created also when the craft is stationary, helping take-off and seakeeping under take-off speed, where a normal WIG would act basically as a boat, by increasing dramatically the lift and reducing take off speed. This also helps eventual problems of the hull hitting the waves.

The transport efficiency of this configuration, computed as power over payload times range, is about twice of aircrafts, the cruising speed is between 3 and 5 times higher compared to ACS, catamarans and hydrofoils.

They also have to ability to reach an high enough altitude where the ground effect is not present any more for manoeuvring purposes, since the manoeuvrability can change drastically outside of ground effect, with its added flexibility advantages.

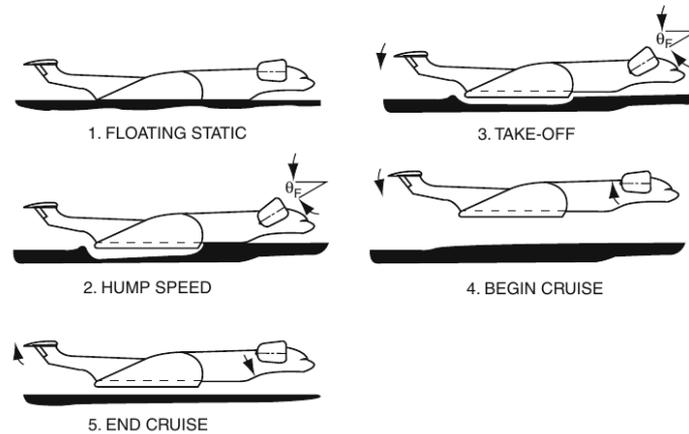


Figure 1.9: *different phases modes of a parawig [18]*

1.5 Dynamic and aerodynamics

A primary bibliographic research is conducted in order to assess the problem of dynamic modelling of ekranoplans, both in terms of the dynamical equations model and on the aerodynamic problem, which constitutes the main challenge.

Being a relatively new kind of vehicle, the literature focused on ekranoplans is rather scarce, and a few authors are considered the luminaries in this field. One of them, arguably the most famous, is professor K. V. Rozhdestvensky, that wrote [7] and many other publication about ekranoplans and related topics.

[18] represents the most complete treatise on the subject, which is oriented at a wider audience rather than experts, with the intent of providing general knowledge about the world of ekranoplans by also mentioning interesting historical insights on their development. It is a good starting point on the topic, showcasing many different kinds of vehicles and describing the phenomena that participate in the dynamic of an ekranoplan. In particular it contains a good description of *ground effect*:

“Due to the flow being forced to flow between the wing underside and the ground, the pressure on the lower surface of the wing increases, thus increasing the lift.

In addition, for a wing operating close to the ground or water surface, the downwash velocity caused by wing tip vortices will be reduced and the induced resistance too, reducing the drag force.”

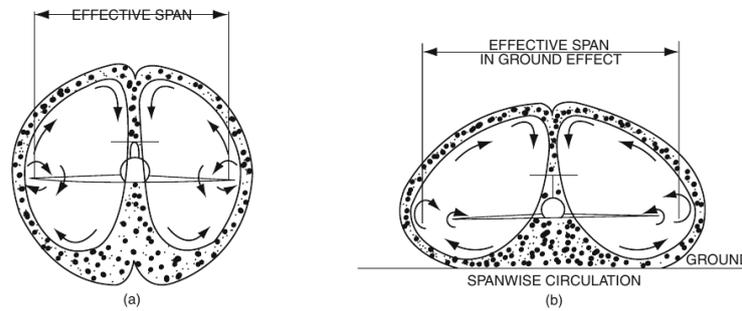


Figure 1.10: *Effect of the presence of ground on the wing span*

As of today, the publication that is similar the most to the objectives of the present work is [22].

A simulator is built using the rigid body dynamic equations from [3] and DATCOM [30] to compute the main aerodynamic characteristics of his custom wig craft, which belongs to the same category as the AirFish-8. The ground effect is implemented in the form of aerodynamic corrections, such as the ones contained in [25]. In this case a formula is used to add the contribution by superposition. The ground effect isn't tackled in a monolithic way, but rather through a composition of known aerodynamic phenomena, starting from aeroplane simulations. The thesis goes on presenting different contributions to ground effects and how they are described by different sources.

Its main focus is a straight-line motion seeking steady-state. The key parameter is the *angle of attack* α , since it is what the most important parameter forces depend from.

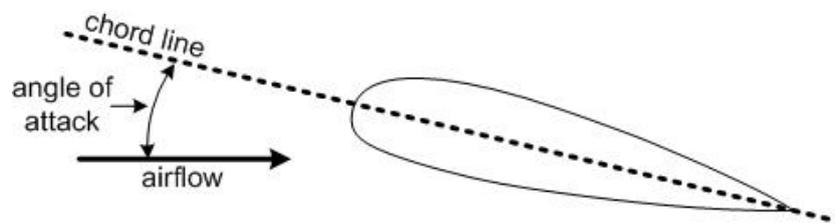


Figure 1.11: *angle of attack definition*

A paper by Wieselsberger [25] is historically one of the first quantitative studies about ground effect, and doesn't have any explicit dependency on the configuration variables such as the Euler's angles. This is justified by the fact that ground effect is also a non negligible phenomenon that takes part in the study of landing and take-off of "regular" aircrafts, situations

where it was firstly observed. The first studies were not connected with the concept of wing-in-ground vehicle, but solely with aeroplanes because of the need to take into account this effect during take-off and landing, as it is non-negligible.

Despite being a simple quantification of how the presence of ground varies the aerodynamic behaviour, the paper contains useful informations on the theory of ground effect.

The *influence coefficient* is defined as:

$$\sigma = \frac{1 - 0.66\frac{h}{b}}{1.05 + 3.7\frac{h}{b}} \quad (1.1)$$

where h is the height with respect to ground and b is the wing span. This ratio is cited many times in many different papers, and it is a, probably the most, famous quantification of the influence of ground effect. It expresses the main effect of the presence of the ground, that is the strong modification of the flow around the aeroplane. This modification consist in a combination of flattening and depletion that modifies the *aspect ratio* of the wings. It is often the normalised reference values used for the definition of the range of effectiveness of ground effect [9].

Despite not being a decisive and alternative solution to CFD approach, the paper provides an idea about the magnitude of the forces in presence of the ground, giving some help for early design. It also contains an experimental analysis, which gives the useful upside of an approach for modern numerical analysis. It is called *image method* and it is still used nowadays to simulated the presence of the ground in CFD programs as a reliable and consolidated way to do so. It consists of putting the same wing as the one being analysed symmetrically opposite horizontally at a distance double the one from the ground of the regular wing, as the figure below illustrates.

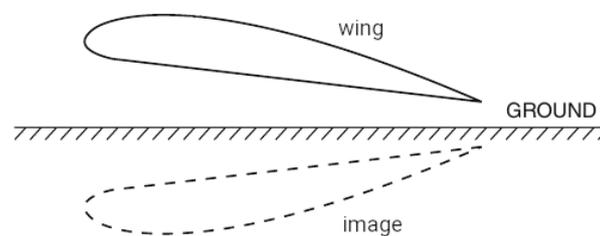


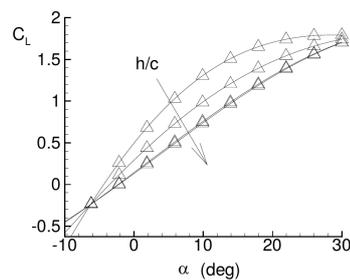
Figure 1.12: *visualisation of the image method: the image is used instead of the ground to simulate the ground effect*

ment of wig crafts. It makes use of the theory developed in another work by Rozhdestvensky [7] which presents methods for the analytical computation of ground effect.

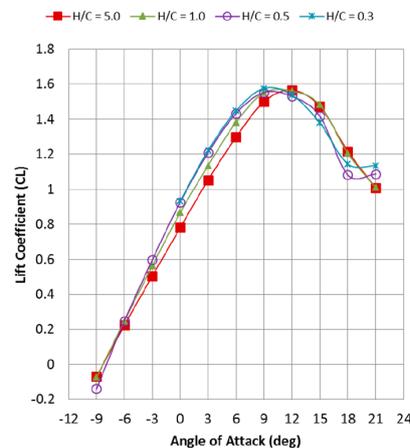
The following part of the present project, which consists in dimensioning a model that will be the first to be realised and tested, is being done in parallel to the present work. It consists in analysing the maximum lift capabilities given the dimensions of a model, taking into account speed constrains and on board instruments, electric propulsion and batteries. Finer test are being done in Ansys[42], since it takes into account turbulence, which also leads to extremely long computational times.

In this regard, a case of studies that focuses on ground effect using ansys is [43]. In particular its aim is to evaluate *lateral dynamic* performances of an aeroplane subject to ground effect. With an $\alpha = 8^\circ$ the increment given solely by the presence of the ground is measured as 11,2%. Moreover a roll angle $\phi = 10^\circ$ shows a significant variation of the pitching angular momentum. The simulation were run using a cluster of 24 processors at a University, and the average convergence time for a single point is 1 – 2 days.

This number quantifies the computational effort needed for an analysis of the lateral forces, where the focus usually isn't. Such effort has to be wisely evaluated and be used has a final tool for the design, when the dimensions have already been decided, since this aspect of the aerodynamics is quite delicate.



(a) variation of lift coefficient for different $\frac{h}{c}$ values [10]



(b) variation of lift coefficient for different $\frac{h}{c}$ values [9]

The figures above show the graphs representing the C_l coefficient as a function of the *angle of attack* α for different values of the *aspect ratio*. The ones on the right are obtained through wind tunnel experiments and present

an inversion after $\alpha = 10^\circ$, something that doesn't happen in the left figure, which is obtained through the analytical method. These studies represent the current reference for the aerodynamic forces.

Expressing the forces in terms of coefficients allows for a simple handling without taking directly care of velocity and other parameters. While this representation is easy to manage, it lacks a direct dependency on the *angle of attack* and the altitude, which is crucial for ekranplan.

This is why the graphs show different contour lines for different altitudes, that are expressed as different *aspect ratios*.

1.5.1 Used method for aerodynamic forces

In order to complete the model the aerodynamic, forces are computed by the means of a CFD program.

One of our colleague's task was to solve the issue of the computation of aerodynamic forces. He created a CFD tool that is a conjoined of C++ tools and MATLAB packages able to solve the problem for a scaled ekranplan with the form factor of an Airfish 8 from WigetWorks, which has the size of the model that is being printed for wind tunnel tests.

This will be used to obtain some meaningful results that can help implementing the aerodynamic forces in the simulator.

The best achievable result is to obtain an analytical expression that represents the behaviour with a limited error. This is required in order to build controllers, linearise for the LQR, sliding mode and geometric control.

By far the most common and validated method, as found in many papers, is to start from a prototype of the vehicle, the prototyping technologies vary but usually it is additive manufacturing, and get a first estimation of the forces in wind tunnel experiments. The results are usually meant to validate the ones obtained through CFD simulations. Both are required, has the CFD is less man-time consuming and flexible, and the other is more realistic but requires special skills to be carried out.

A first rough idea is always needed at first, but since scaling is another big problem in aerodynamics, other experiments are required once the final scale is realised, going back and forth with the methods as required.

In some cases adaptive controllers in conjunction with sensors can lead to online forces identification an model adaptation [29]. This is another idea for the future, to be done when an operative model will be available.

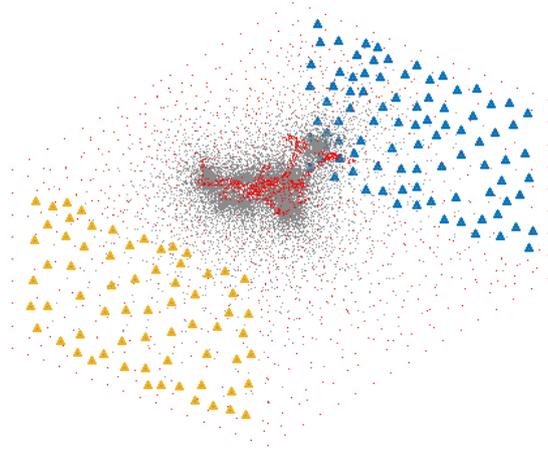


Figure 1.15: *View of the lines casting, one of the steps of the simulation*

The idea is to use the provided program to compute the aerodynamic forces for different sets of the input parameters (like the configuration coordinates) and then interpolate the function that describes the forces. This means creating a big enough amount of data through CFD computation to be later interpolated to obtain a function.

The tool has some limitations in order to make it fast enough to run all the needed simulations in a sensible amount of time, but not without drawbacks.

The main characteristics of the CFD tool are:

- the flow is considered to be always stationary and the Reynold's number to be 1 for all the cases, thus lift and drag coefficients are uncorrelated to it.
- the centre of mass coincides with the centroid of the aeroplane, so it will always rotate around this point.
- the program simulates a wind tunnel, so the aeroplane having a speed \mathbf{V} with respect to the inertial reference is obtained by having the wind come toward the aeroplane at same speed. With respect to the inertial frame this means $-\mathbf{V}$
- the range of speed available is $5 \div 30 \frac{m}{s}$
- the values of forces and angular momenta are returned with respect to the inertial fixed reference frame

- some configurations generate errors that stop the simulation, so it doesn't allow to simulate for every value of the parameters in the configuration range

The main problem is the lack of turbulence in the simulator, as the Reynold's number is set to 1. This leads to underestimated output values, that are going to be shown and discussed later.

Given this premises the steps are:

- use the simulator to obtain a dataset and the needed function of aerodynamic force and angular momenta.
- use the obtained expressions as meaningful in terms of behaviour but not magnitude, adjusting them according to the bibliography and adjusting the characteristics of the vehicle such as weight, to make them reasonable. The obtained model is still valid as a test bench for controllers, since the obtain model would still be similar in behaviour to a real one.

A rapid estimation of the amount of different configurations is now analysed.

The variables of interest in the simulation are: aerodynamic velocity, altitude, Euler angles. It makes 5 inputs.

Wanting to produce a high number of simulation data that could allow for the identification of the coefficients requires 10 points per parameters, which makes a total of 10^5 runs. Given that the program terminates a cycle in 3 minutes on a quad-core i7 processor, the estimated time of computing is about 200 days of computation for the total quantity.

Of course this is not a reasonable amount of time in any way. Even a computer 10 times faster than the one used for the benchmark it would still require 20 days, 20 days to obtain results that are not exact.

This method has to be given the importance it has and the results it can achieve shall not be overestimated, as it cannot yield perfect results.

The chosen method is to simulate and obtain the forces by carefully choosing which effects to analyse reducing the total number of simulations to a reasonable one. The configurations are chosen as meaningful configurations of the vehicle. The reduced number of simulations, that will be explained later, still took many days to achieve.

1.5.2 An especially interesting method for the estimation of the aerodynamic coefficients

The method described in [10] is here reported as it represents an interesting way of obtaining the aerodynamic model of an ekranplan specifically. This method goal is to obtain a relationship between configuration (Euler's angles) and the aerodynamic forces, something that hasn't been achieved before using an analytical model so close to the result, that only leaves the potential to be computed, instead of the whole pressure field and the forces, reducing the number of needed calculations.

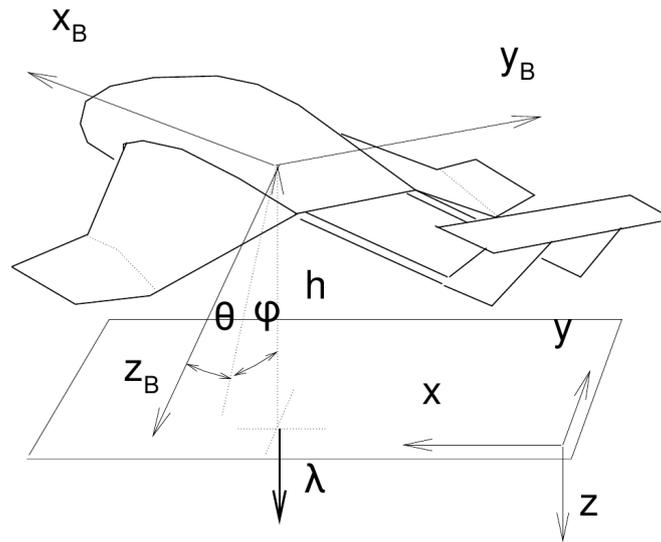


Figure 1.16: attitude vector λ and the characterising angles as in the formula [10]

The main feature, as described by the author in his paper, is the dependency of the forces from the configuration of the aeroplane, which is described as being an unusual feature amongst commonly used methods and that constitutes a crucial advantage for a comprehensive model. This model form is quite important, given that it would provide a precise way to build a 6 degrees of freedom model including aerodynamic forces.

The attitude vector is:

$$\lambda = (-\sin \theta, \sin \phi \cos \theta, \cos \phi \cos \theta) \quad (1.2)$$

where θ and ϕ represent the usual Euler angles. Figure 1.16 show the *body frame* axes, the *inertial frame* and the *attitude vector*.

The key is computing the *velocity potential* in order to compute the *forces* and *angular momenta* through the Lagrangian method, by first computing the *kinetic energy* T .

The potential ϕ can be written as

$$\phi = \frac{\partial \phi}{\partial \mathbf{v}_b} \cdot \mathbf{v}_b \equiv \frac{\partial \phi}{\partial \mathbf{v}_b} \cdot (\cos \alpha \cos \beta, \sin \beta, \sin \alpha \cos \beta) V_b \quad (1.3)$$

where \mathbf{v}_b is the *flight velocity* vector in body frame, α the *angle of attack* and β the *sideslip angle*.

The derivative can be written in terms of $\boldsymbol{\lambda}$ like if it were a series expansion up to the second terms.

$$\frac{\partial \phi}{\partial \mathbf{v}} = \frac{\partial \phi_0}{\partial \mathbf{v}} + \frac{\partial^2 \phi_0}{\partial \mathbf{v} \partial \boldsymbol{\lambda}} (\boldsymbol{\lambda} - \boldsymbol{\lambda}_0) \quad (1.4)$$

At this point the *kinetic energy* is

$$\begin{aligned} T &= \frac{1}{2} \rho \int \int_{S_v} \mathbf{v} \cdot \mathbf{n} \phi dS = \frac{1}{2} \mathbf{v} \cdot \mathbf{M} \cdot \mathbf{v} + \frac{1}{2} \mathbf{v} \cdot \{\mathbf{H}(\boldsymbol{\lambda} - \boldsymbol{\lambda}_0)\} \mathbf{v} \\ \mathbf{M} &= \int \int_{S_v} \frac{\partial \phi_0}{\partial \mathbf{v}} \otimes \mathbf{n} dS \quad ; \quad \mathbf{H} = \int \int_{S_v} \frac{\partial^2 \phi_0}{\partial \mathbf{v} \partial \boldsymbol{\lambda}} \otimes \mathbf{n} dS \end{aligned} \quad (1.5)$$

The formula for computing the kinetic energy is the same as [20], but in the latter is has a different form

$$2T = -\rho \iint \phi \frac{\partial \phi}{\partial n} dS \quad (1.6)$$

Apart from a small discrepancy in the notation, the set of rigid body dynamic equations is the same as [3].

The method has some downsides. The existence of a potential flow is subordinated to the conditions of inviscid, incompressible and non-turbulent flow, which is not the case of the matter of study. This assumptions are also reported in [31, par. 2], where the quality of such method is analysed. The conclusions are that the assumptions introduce some error of different natures, that can be taken care of in different ways given the neglected phenomenon.

This method is clearly more elegant in its less computationally oriented core. Unfortunately a comprehensive solution that would account for the error seems difficult to obtain.

1.6 Control

Dedicated control theory of ekranoplans is almost non-existent given the limited development that such vehicles have seen in the recent era. Despite being dramatically different with respect to classic aeroplanes, control theory for the latter category is a starting point in terms of available control strategies. The primary investigation about known controllers' behaviour is focused on aeroplanes, keeping in mind the sought objective for the vehicle of interest.

Since ekranoplans are characterised by the ground effect, which implies close proximity to the ground, altitude constitutes an important parameter, that has to be taken care of more carefully compared to aeroplanes. This is especially true if the marine operational environment is considered, since it creates different problems compared to air.

Given that ekranoplans are less dexterous, in the sense that the minimum turning radius is reduced compared to a normal aeroplane [18] [5], every kind of controller is more demanding and tolerances tighter.

The most important control scenario is seakeeping, for which a decoupled and linear model is used [4]. This allows for linear control techniques to be employed, such as LQR and many PID, which are the most common ones. This control scenario is often represented by maintaining a designed cruising equilibrium position, without special manoeuvring involved. The linear longitudinal decoupled model doesn't include the height, and is usually the main focus for control techniques on aeroplanes.

The linearised longitudinal model has the form [16][cap. 27]

$$\begin{bmatrix} m\dot{u} \\ m\dot{\alpha} \\ J_{yy}\dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & 0 & -mg \cos \Theta_0 \\ Z_u/U_0 & Z_\alpha/U_0 & Z_q/U_0 + m & -mg/U_0 \sin \Theta_0 \\ M_u & M_\alpha & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta X^c \\ \Delta Z^c \\ \Delta M^c \\ 0 \end{bmatrix} \quad (1.7)$$

One delicate aspect of aerodynamic is often dealt with when linearising the model [16, cap. 27] are the derivatives of the aerodynamic forces and other aerodynamic parameter that are often the hardest part to model for an aeroplane. These coefficients are represented in (1.7) by X and Z , where the subscripts represent different derivatives.

Unfortunately such parameters are not yet available for the case of study, so the linearised model above is not utilised.

It is important to specify that altitude is a parameter of interest for aeroplanes, but the insensitivity of the model on it in conjunction with the general soft dependency makes it much less a worrying compared to the ekranoplan case.

Moreover, a simple path following strategy built on the linearised model using PD controller is presented in [16, cap. 27]. It starts from the assumption that the inner-loop dynamics that control attitude and velocity have been already implemented, making it possible to track a desired flight path.

The path represents the desired location of the vehicle, either in 3D (without time) or in 4D (with time). The path follower represents the outer control loop whose goal is to generate the references for the inner loops to compute them, making the vehicle follow the path.

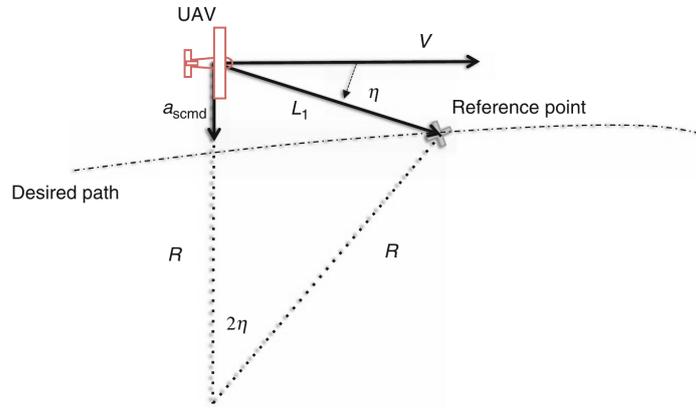


Figure 1.17: *pure pursuit guidance logic* [16]

$$a_{s_{cmd}} = 2 \frac{V^2}{L_1} \sin \eta \quad (1.8)$$

The figure 1.17 show the angle η between the velocity V and the vector L_1 that point to the reference point. The acceleration $a_{s_{cmd}}$ is computed with the above formula. Its sign depends on the sign of the angle η .

This constitutes the non-linear way of doing simple path following, but as mentioned above a PD over the linearised model can be used. In that case the equation is

$$a_{s_{cmd}} = 2 \frac{V^2}{L_1} \sin \eta \approx 2 \frac{V}{L_1} \left(\dot{d} + \frac{V}{L_1} d \right) \quad (1.9)$$

with the assumption that the angle η is small.

The added problem for an ekranoplan control is the more stringent requirements on altitude control compared to a regular aeroplane, which forces the use of different safety features.

The classical design approach would require the determination of the stability and controllability for the different flight mode. The characteristics

are corrected in case of unsatisfactory results through the design of automatic and semi-automatic controls, also for manual manoeuvring. This is what usually happens where the stability varies slightly or is relatively slow [32].

The added problem with WIG design is its strong stability dependency on altitude. There's also the problem of intermittent loss of stability during the range of motion, which complicates things.

1.6.1 MPC

Other non-linear multi-purpose control strategies have some literature about them too. In particular, the non-linear MPC is said to be able to control an aeroplane with the pending problem of reliability verification [38], which means making the software compliant with security requirements. This type of controller has been historically employed for slow dynamic systems, such as chemical reactions. In the last decade the available computational power allowed for other applications characterised by fast dynamics. The main MPC peculiarity is that it allows for constraints on states and inputs to be set. The most famous cost function formulation is quadratic mainly because of the availability of good quadratic solvers, even though it could be defined in other ways.

The constraint capability is particularly advantageous when considering the recent tendencies in aerospace technologies, that point toward a more efficient and safer aerospace system.

The MPC is intrinsically suitable for this, given its potential to guarantee high safety levels in both guidance and control, in particular for the applications as autonomous control for landing and continuous descent approach [38].

The other main focus is on the long-term switch to green power technologies, with the final objective of building fully electric transportation means [38]. The MPC framework is naturally suitable for these tasks and the escalating environmental concerns are going to ease the path.

The article [17] shows a quite different case of control using MPC and some other techniques in conjunction. It shows a case of automatic guidance of an aeroplane using linear MPC by adapting the controller through the addition of a switching mechanism.

Each segment between two waypoints, which also identify a different flight condition, has an associated controller configuration. In the perspective of the future evolution of the present project, this represents a good manner of putting together the different components of the control part.

In the *short term trajectory reference (STR)* a linkage is created between the independent variable, the time, and the position. This because the first

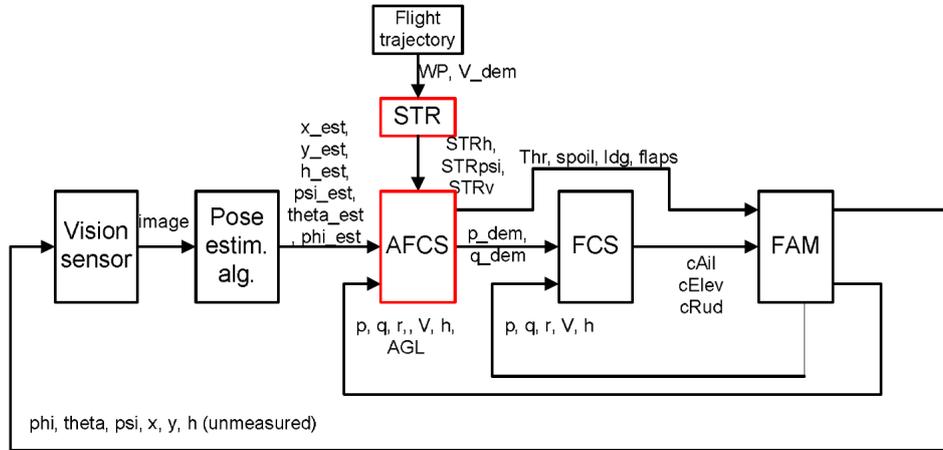


Figure 1.18: *Cascade control structure for position based visual servoing of an aircraft [17]*

is the independent variable in predictive control, but the reference flight trajectory is always position based, therefore a dedicated block is required to join them together.

The position is estimated through visual servoing, which is a rather unique way of carrying of the task. Given that the project is going to implement the same sensors for the purpose of obstacle detection and avoidance, the same principle could be exploited, giving a second use to the visual sensing devices. It might be used in the estimation process, moving horizon estimator or Kalman filter, to get a good initial guess of the vehicle configuration.

The used model for the simulation is a *FAM*, fixed-wings aircraft model, which is linearised in the loop. The model is decomposed in the usual two parts: longitudinal dynamics and lateral dynamics. Each submodel takes care of a subset of the tasks being carried out. Apparently this approach is precise enough for complex tasks as the ones shown in the paper. As said before, ekranoplans are different enough with respect to aeroplanes to be considered a whole different problem, which involves carefully considering facts about aeroplane and validate them through testing.

A supplementary strategy for control softening is employed, which consists in keeping the input value fixed for a chosen block of time instants which can differ between the different variables. This is used to deal with too fast sampling times, that lead to a more nervous controller, and to increase robustness to noise. This principle has been taken into account for the simulations in the present work, and applied to the case where measurement

noise was present. The results haven't been as exciting as described in the paper, as figure 4.5 shows. Further tests are needed, given that the issue applies to the present work too.

The control scheme 1.18 also contains a classical controller in the *FCS*, flight control system, block that helps the *AFCS*, automatic flight control system. This last one contains the predictive controller, which suffers from problems in the optimisation process caused by flawed variables' values. To reduce this effect the *FCS* implements a classical feedback controller on *pitch* and *roll* rates. These values are then used as inputs for the *FAM* block.

This constitutes an interesting strategy to be implemented in the future iterations of the non-linear MPC on ekranoplans, since the noise effect has to be addressed with a dedicated solutions given its noticeable effect.

Important to notice how the paper confirms the experimental optimization to be the usual method used for MPC tuning, that is anyway needed, despite having some general criterion to guide the process [44].

1.6.2 Adaptive controller L1

Another kind of control that has the right characteristics is the \mathcal{L}_1 adaptive control [27]. It is a powerful control design tool for handling large parametric uncertainties that has been attracting the interest of researchers. The L1 -AC scheme is designed mainly for plants with full state measurement, which is in one way or another achievable in the present case. It has recently been widely advertised as a suitable controller for aerospace control since it can guarantee fast and robust adaptation, an essential characteristic given the unavoidable characteristics that affect the aerodynamic models used for aeroplanes, and more specifically the ground effect case.

The \mathcal{L}_1 adaptive control is a more general *MRAC* controller with a low-pass filter in front of the control input. The use of the filter is justified by the fact that the tracking error can be made arbitrarily small during the transient for this class of adaptive controller by increasing the adaptive gain. The cost of increasing this gain is high in terms of qualities of the adaptive law, and should be avoided, since it doesn't solve the problem of oscillation of the estimated variables.

The peculiarity of this control law is that it can be applied on plants with unknown parameters characterised by a large uncertainty such that the control objective cannot be achieved with robust non-adaptive techniques. The conditions are that the state is fully measurable and that the input vector is known.

This scenario fits well in the present case, given that the estimated aerodynamic parameters might not be precise enough even after wind tunnel and

CFD experiments. To corroborate this eventuality there is the intrinsic uncertainty that derives by scaling the vehicle dimensions to ones suitable for prototyping and testing. An adaptive controller of this kind could work even with a rough estimation as a starting point. Such characteristics are probably a requirement if the vehicle is working in a uncontrolled environment, where wind and waves are inevitably be present.

Extreme cases of this are present in the bibliography, like [29], even though applied to rather simpler cases.

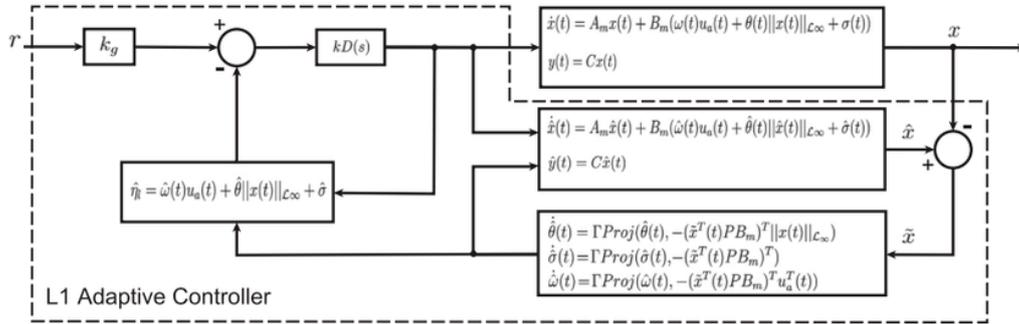


Figure 1.19: control scheme of an L_1 adaptive control [15]

The formulation of \mathcal{L}_1 adaptive controller starts from a system of the form

$$\dot{x}(t) = A_m x(t) + b \theta^{*T} x(t) + b u(t), \quad x(0) = x_0 \quad (1.10)$$

The requirements are that the state vector is measurable, the vector $\theta^* \in \mathbb{R}^n$ of the unknown parameters belong to an unknown set $\Omega \subset \mathbb{R}^n$, $A_m \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ are known. The matrix A_m of the system can be modified in terms of eigenvalues by state feedback gain.

The objective is to chose an input $u(t)$ such that all signals in the close-loop system are uniformly bounded and that $x(t)$ tracks the state vector of the given reference model described by

$$\dot{x}_m(t) = A_m x_m(t) + b_m r(t), \quad x_m(0) = x_0 \quad (1.11)$$

both transient and steady state, for any bounded reference signal $r(t)$.

This characteristics are certainly interesting and make the controller capable of tackling the issues of the uncertainty of the aerodynamic model. It will be further analysed in the future, probably with a dedicated thesis work with the objective of implementing it on the vehicle for the first tests.

Chapter 2

Dynamics equations

The employed dynamical model is the same used for boats [11], as other evidences show [3]. The difference compared to an aeroplane is that the simplified model that doesn't take into account Earth roundness and rotation can be used.

2.1 Generic dynamical equations

The vector form of the equations from [10] with some notational adjustments is

$$\begin{cases} m(\mathbf{v}_b + \boldsymbol{\omega} \times \mathbf{v}_b) = \mathbf{F}^e \\ \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} = \mathbf{Q}^e \\ \dot{\mathbf{x}}_i = \mathbf{T}_{ob}\mathbf{v}_b \\ \dot{\boldsymbol{\Phi}} = \mathbf{R}\boldsymbol{\omega} \end{cases} \quad (2.1)$$

where the first two represent respectively the force and angular momenta equations in body frame, the last two the transformation to inertial frame and \mathbf{J} is the inertia tensor.

The vector $\dot{\mathbf{x}}_i$ represents the velocity with respect to the *inertial frame*, and \mathbf{x}_i is the position vector. By computing the last components of the position the altitude is taken into account.

2.2 Definition of reference frames

The reference frames used for the model description are the following:

- **Inertial frame** $F_I(A, \mathbf{x}_I, \mathbf{y}_I, \mathbf{z}_I)$: Galilean frame centred in an arbitrary fixed point on the water surface, which is considered calm and

flat. The axis \mathbf{z} points up towards the sky parallel to gravity and the others belong to the plane.

- **Vehicle-carried frame** $F_o(O, \mathbf{x}_o, \mathbf{y}_o, \mathbf{z}_o)$: origin in the centre of mass of the vehicle. \mathbf{z}_o is oriented as the local gravity seen by the aircraft. This reference frame is obtained by rotating the *inertial frame* by π around the x axis. Therefore each axis is parallel to one belonging of the previous reference frame.
- **Body frame** $F_b(G, \mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b)$: it centred in the *centre of mass* G of the aircraft. The axis \mathbf{x}_b is oriented toward the front and \mathbf{z}_b is perpendicular to the latter and belongs to the symmetry plane of the aircraft. \mathbf{y}_b completes the triad. It is obtained through a rotation of the previous one according to the definition of Euler's angles.
- **Aerodynamic frame** $F_a(G, \mathbf{x}_a, \mathbf{y}_a, \mathbf{z}_a)$: same origin in G as F_b . The axis \mathbf{x}_a is oriented as the *aerodynamic velocity* \mathbf{V}_a which is the speed of G relative to the air not influenced by the aircraft aerodynamics. The frame F_a is obtained by rotating F_b by the angles α_a and β_a named respectively *angle of attack* and *side-slip angle* which is the angle β_a between the symmetry plane and \mathbf{V}_a . The axis \mathbf{z}_a is in the symmetrical plane of the aircraft completing the triad.

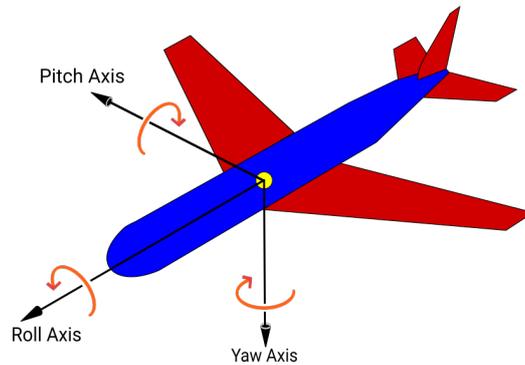


Figure 2.1: *visualization of the angles between vehicle carried frame and body frame [1]*

The *aerodynamic frame* is important especially used for aeroplane stability and control [16, cap. 27], since it allows for a simplified handling of the wind conditions.

the figure 2.1 shows how the *angle of attack* and the *sideslip angle* are defined given the *aerodynamic velocity*.

These angles represent the rotations needed to bring x_b aligned with x_a , where the latter is oriented as the aerodynamic velocity.

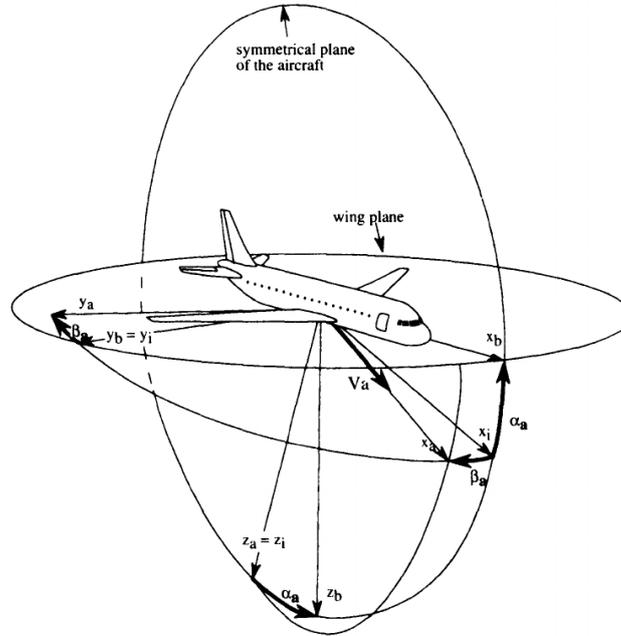


Figure 2.2: the aerodynamic frame F_a with respect to the body frame F_b [3]

In the present study, no external wind is going to be considered. This leads to $\theta = \alpha$, that is the *pitch angle* is equal to the *angle of attack*, as in [10].

2.2.1 Forces and momenta equations

Defining the components of the velocities in (2.1)

$$\mathbf{v}_b = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad \boldsymbol{\omega} = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (2.2)$$

Expanding the first of (2.1) by computing the vector product the three scalar equations below are obtained

$$m \begin{pmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{pmatrix} = mg \begin{pmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{pmatrix} + \frac{1}{2} \rho S \|V_a\|^2 \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} + \begin{pmatrix} F_x^b \\ F_y^b \\ F_z^b \end{pmatrix} \quad (2.3)$$

where the components are called, in order, *roll*, *pitch* and *yaw* angles, supported respectively by \mathbf{x}_b and \mathbf{y}_b (nothing is mentioned about \mathbf{z}_b).

Given a general inertia tensor

$$\mathbb{I}_G^b = \begin{pmatrix} A & -F & -E \\ -F & B & -D \\ -E & -D & C \end{pmatrix} \quad (2.4)$$

since the vehicles of study are usually symmetric with respect to a vertical plane, like the vehicle of study, the following applies

$$D = F = 0 \quad (2.5)$$

which leads to the used form of the inertia tensor, referred to body frame and centred in the centre of mass

$$\mathbf{J}_G^b = \begin{pmatrix} A & 0 & -E \\ 0 & B & 0 \\ -E & 0 & C \end{pmatrix} \quad (2.6)$$

the angular momenta equation [3, sec. 5.56] on body frame are

$$\begin{pmatrix} A\dot{p} - E\dot{r} + rq(C - B) - Epq \\ B\dot{q} + rp(A - C) + E(p^2 - r^2) \\ -E\dot{p} + C\dot{r} + pq(B - A) + Erq \end{pmatrix} = \frac{1}{2}\rho S \ell V_a^2 \begin{pmatrix} Cl \\ Cm \\ Cn \end{pmatrix} + \begin{pmatrix} M_{F_x^b} \\ M_{F_y^b} \\ M_{F_z^b} \end{pmatrix} \quad (2.7)$$

The last two equations of (2.1) are the transformations between the quantities in body frame and the vehicle-carried frame. The following represents the relationship between Euler's angles' derivatives *roll*, *pitch* and *yaw* and the angular velocities in body frame p , q and r .

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \phi \tan \theta & -\sin \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (2.8)$$

This allows to find the relative orientation of *body frame* with respect to the *vehicle-carried frame*, obtaining the angular configuration of the vehicle.

The rotation matrix [3, eq. 2.49] in the last of (2.1) is

$$\mathbf{T}_{ib} = \begin{pmatrix} \cos \theta \cos \psi & \sin \theta \sin \phi \cos \psi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \phi \sin \psi \\ \sin \psi \cos \theta & \sin \theta \sin \phi \sin \psi + \cos \psi \cos \phi & \sin \theta \cos \phi \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{pmatrix} \quad (2.9)$$

Therefore

$$V_i = \mathbf{T}_{ib} V_b \quad (2.10)$$

Chapter 3

Aerodynamics

The aerodynamics constitute the peculiarity of ekranoplans, being the crucial difference that makes them be classified as a special kind of boat rather than an aeroplane.

The general form of the aerodynamic forces, given the aerodynamic coefficients, is:

$$\mathbf{F}_a = \frac{1}{2}\rho S V_a^2 \mathbf{C}_F \quad (3.1)$$

where ρ is the fluid density, S the surface of the wing, V_a the aerodynamic velocity and \mathbf{C}_F the aerodynamic force coefficients. This form is the canonical one and allows for better analysis and more useful manipulations.

The steps done in order to obtain the aerodynamic forces expressions to complete the dynamical model are hereby described.

3.1 CFD simulations

The main parameters that determine the aerodynamic effects are recognized to be height and speed, since one governs the ground effect and the other is present in the general formula of aerodynamic forces explicitly.

It is important to highlight the strong dependency between the height from ground and the aerodynamic forces, which is quite strong compared to regular aeroplanes, and make the altitude a primary concern for control, having tighter tolerances for movement.

Some test are conducted with speed ranging between the maximum and minimum values admitted by the program, so $5 \div 30 \frac{m}{s}$ and height in the range of $1 \div 7m$. The step size varies with situation and the desired result.

For what concerns the other parameters, the values used for the basic tests are: $\theta = 10^\circ$, $\phi = \psi = 0$. This because during normal operating

conditions, like cruising, an ekranoplan has a *pitch angle* greater than zero in order to generate enough lift force.

A first set of about 80 test is shown, which have been done considering $\theta = 10^\circ$, as a reference condition. Each test is done using a different speed-height combination.

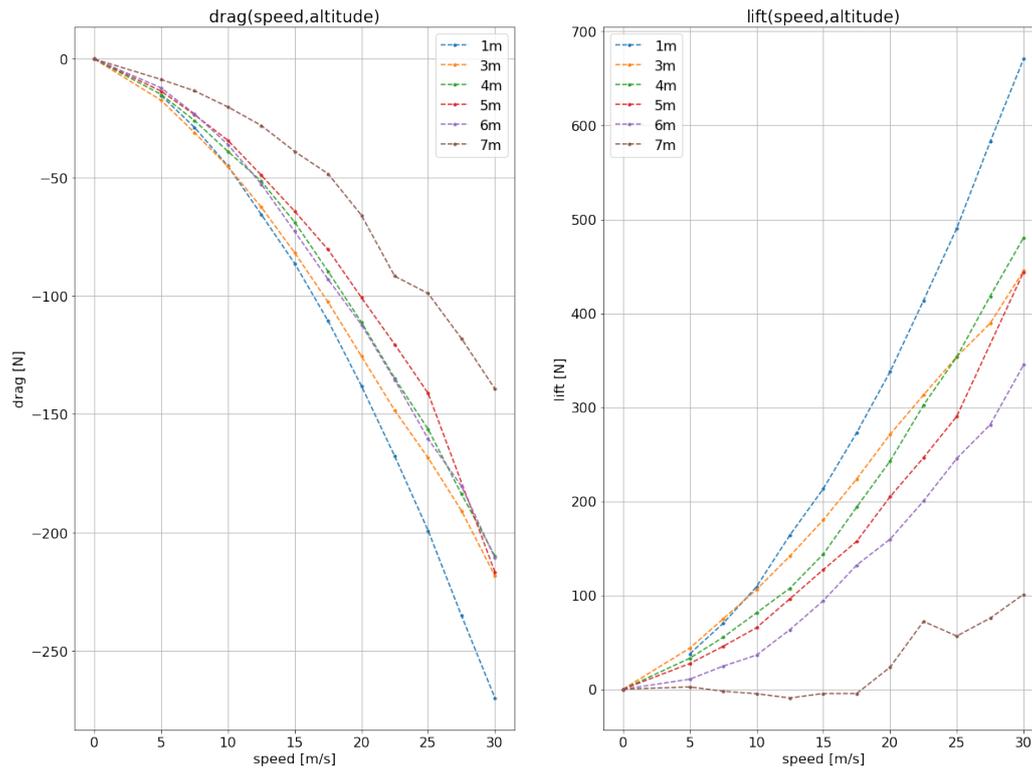


Figure 3.1: *lift and drag forces as speed and height vary, at $\theta = 10^\circ$*

The maximum amount of *lift* force is, obviously, obtained when the aeroplane is at the minimum height and maximum speed, and it is about $670N$. This magnitude isn't enough for sustentation of this type of vehicle, which in spite of being small, with a wingspan of about $2m$ will for sure be heavier than the weight this force could sustain, which is about $70kg$. This is caused, as said before, by the lack of turbulence in the model, that prevents the forces in the simulations to reach real value.

Some brief tests have been done using Ansys, which have shown that forces are about 5 times the what 3.1 shows. The model parameters will be tuned accordingly in order to obtain a coherent physical representation.

While the analysis of the lift is trivial, for the drag nothing similar can be said without comparing the results to something else, which would tell

the magnitude of the error.

Analysing deeper the curves some strange behaviours are present, as the way the curves stack and cross each other is not expected. In a normal case the curves would be as cleanly disposed as 3.4, but for some reason including also the stochastic component of the CFD methods, there are crossing points. This shouldn't cause problems to locally linearised model for linear controllers as long as a robust ones are used.

For the simulations of the effect of Euler angles the chosen way is to fix the *height* at $h = 3m$ and compute the forces components as the angles range from $-20^\circ \div +20^\circ$. These values are reasonably wide, considering that ekranoplans have much more limited manoeuvrability compared to other aircraft, because of the ground effect. This range is range of motion is similar to what was found in the bibliography.

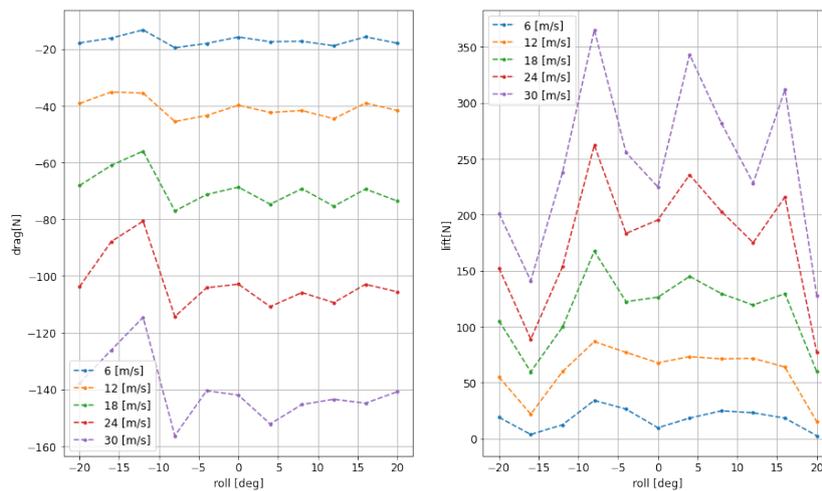


Figure 3.2: *variation of drag and lift as speed and roll vary*

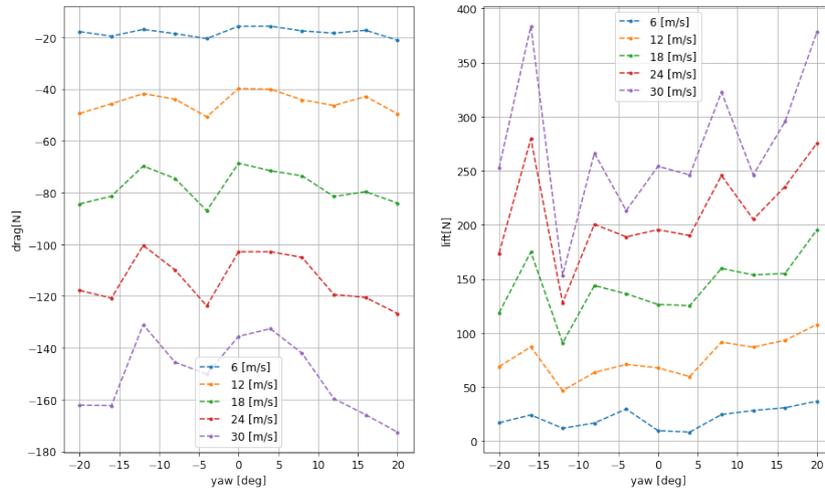


Figure 3.3: *variation of drag and lift as speed and yaw vary*

These graphs do not present any recognizable pattern. For this reason yaw and roll variation aren't considered as influencing the forces referred to the inertial frame, in other words: no parameter is introduced to reflect the changes in forces given by the variation of these angles. In spite of this, the forces are going to change anyway when referred to the body frame, since the orientation matrix is a function of the Euler's angles, so the neglected effects are only on the forces referred to the inertial frame.

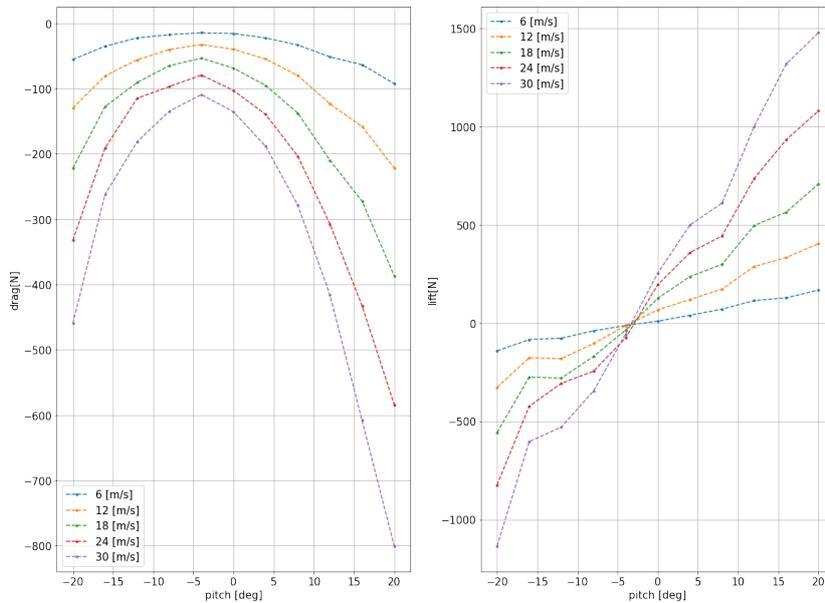


Figure 3.4: *variation of drag and lift as speed and pitch vary*

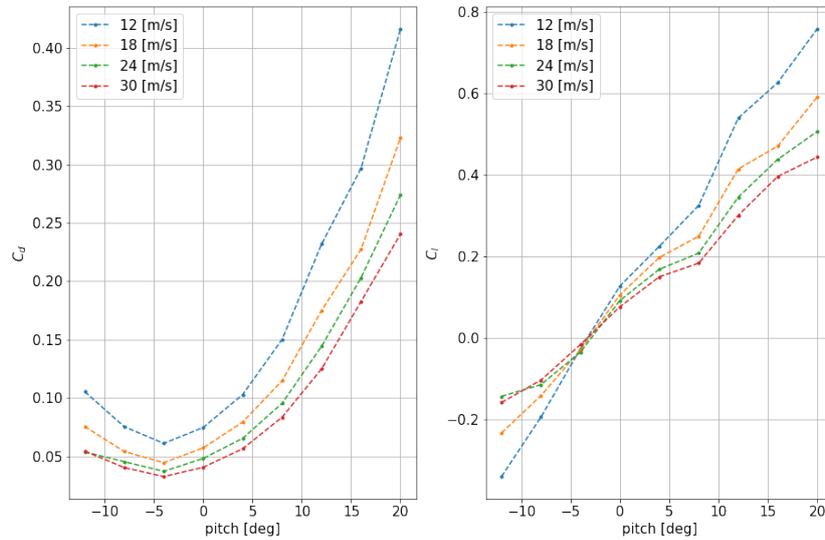


Figure 3.5: *drag and lift coefficients computed by the CFD in use for $\frac{h}{c} = 2, 24$*

The graph above shows a clear and logic behaviour of the forces as the pitch varies. Thus the integration of such parameter can be done with some tweaking that will be explained afterwards.

The behaviour complies with the real case, as an increment of α leads to a greater increase in *drag* than in *lift*.

The problem lies in the comparison of 3.5 with 1.14a and 1.14b. The order of magnitude of the coefficients obtained through the custom CFD are about $\frac{1}{3}$, not considering the evidently inaccurate curves like the one at $6\frac{m}{s}$, the ones founded in the [10] and [9]. Despite this, the behaviour is quite consistent with [10].

Therefore, in order to correct the data obtained through the simulator the results are scaled by a factor of 3.

This yields to the following form of the coefficients

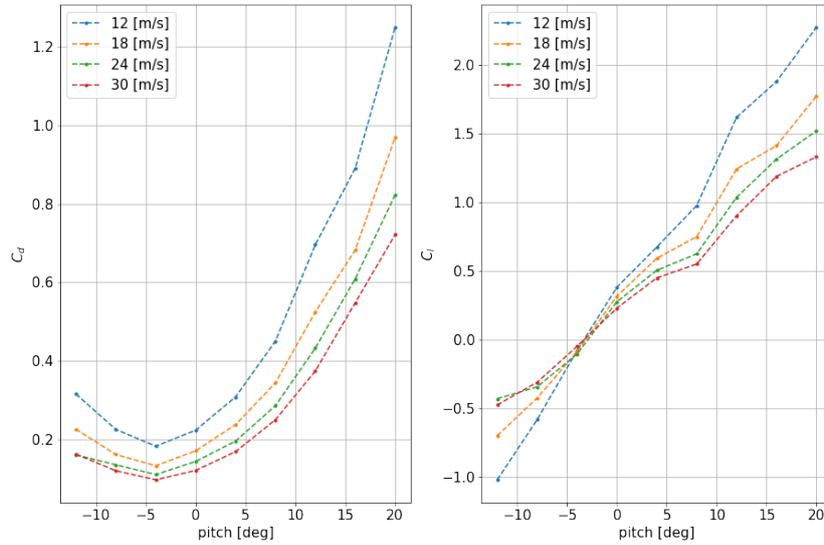
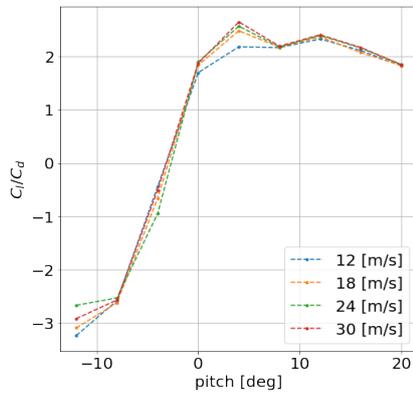
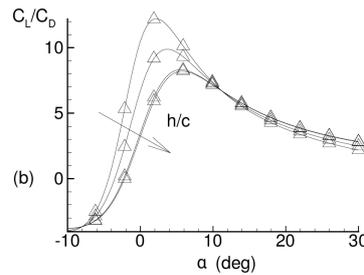


Figure 3.6: *coefficients multiplied by a 3 factor*

Which are now compliant to the reference one. This results do not constitute a final form of the coefficients, since a deeper analysis is needed, but give a good starting point to build a simulator and will be implemented in the present version of the simulator.



(a) lift over drag coefficient using the CFD



(b) lift over drag coefficient using [10]

Lastly, the angular momenta are considered, through the same simulations as before.

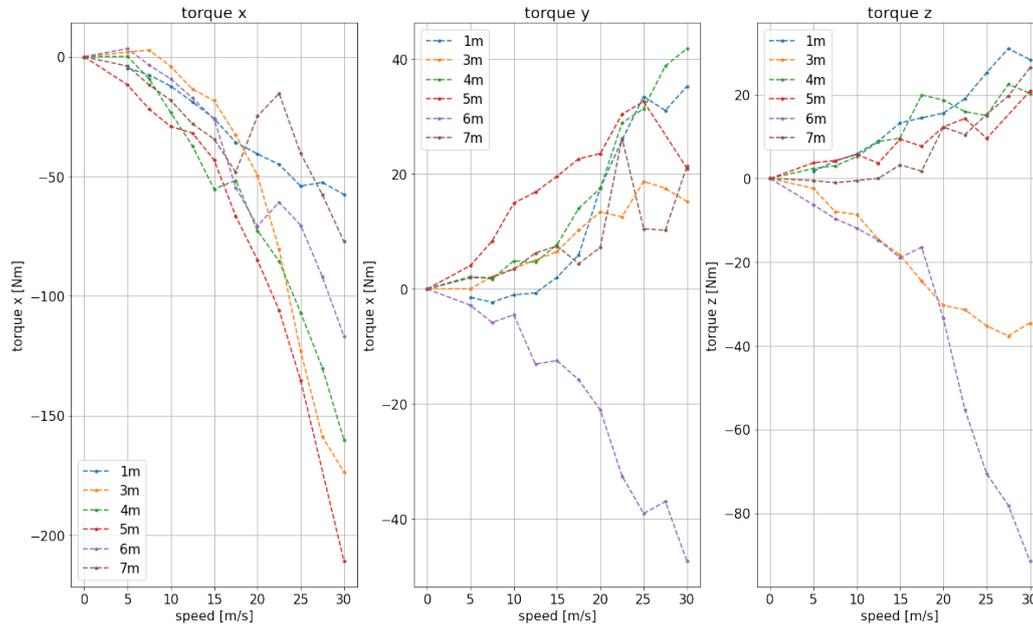


Figure 3.8: *variation of torques as speed and altitude vary*

The torques in the graph are referred to the reference frame used in the CFD, where the pitching torque is around axis x . The main and most apparent characteristic of the graph is the difference between the tendencies of the torque around x and the other ones. As a logical guide the non-pitching torques should be zero when the aeroplane is going straight, but in this case the behaviour is different. Looking at the magnitudes these unexpectedly non-zero torques are much lower than the pitching torque, and don't have a defined behaviour.

This leads to the conclusion that these results are dictated by noise in the computing process, thus can be neglected for the purposes of the present work as already decided for the other irrelevant parameters.

3.1.1 Interpolation of the data

The estimation of the aerodynamic forces and parameters is structured as follows, according to a few assumptions:

The function $\mathcal{F}(\theta, \|\mathbf{v}_a\|, h)$ expresses a given component of a force, and differs for every component. It is considered, given the results obtained through the CFD, only dependent on θ , $\|\mathbf{v}_a\|$ and h .

The amount of data obtained is not enough to build a single multi-variable function of the chosen parameters. For this reason another assumption is

introduced, in order to maintain the necessary informations but make it easier for the interpolation. It is that:

$$\mathcal{F}(\theta, \|\mathbf{v}_a\|, h) = \mathcal{H}(\|\mathbf{v}_a\|, h)\mathcal{G}(\theta) \quad (3.2)$$

where $\mathcal{H}(\|\mathbf{v}_a\|, h)$ represents the forces obtained through interpolation of the data for $\theta = 10^\circ$, \mathcal{G} is a *gain function*, that is defined as

$$\mathcal{G}(\theta) = \begin{cases} \mathcal{G} > 1 & \theta > 10^\circ \\ \mathcal{G} = 1 & \theta = 10^\circ \\ 0 < \mathcal{G} < 1 & \theta < 10^\circ \end{cases} \quad (3.3)$$

The value of the function at a given θ_0 is defined as:

$$\mathcal{G}(\theta_0) = \frac{f(\theta_0)}{f(\theta = 10^\circ)} \quad (3.4)$$

This way the obtained value can be multiplied to $\mathcal{H}(\|\mathbf{v}_a\|, h)$, so that the effect of the *pitch* angle θ is reflected in the force. The approximation done in this process is to firstly manipulate the contour lines by computed the gain mean value for a given section of fixed width of the curves. In this case, since the CFD simulations are done with a step of 4° , the local gain for a step of this size is computed for each curve, corresponding the same initial and final pitch angle value, later the average gain is computed.

The average gain values show little difference, confirming that the approximation is good enough. After this, the obtained averages are interpolated over the angles. This way the variation of the *pitch* angle can be implemented in the forces model using an analytical function.

The obtained values are shown below.

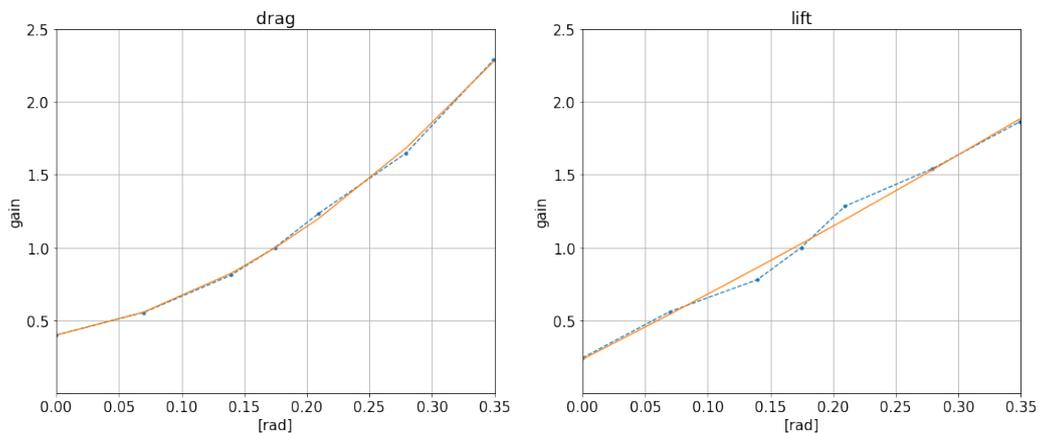


Figure 3.9: *obtained gain function, calculated points and interpolating polynomial*

The function $\mathcal{H}(\|\mathbf{v}_a\|, h)$ is interpolated by the means of *bicubic interpolation* method for each component of the force and angular momentum vector.

For a given force or angular momentum component j , the form is the following

$$\mathcal{H}_j = [\bar{v}^n \quad \bar{v}^{n-1} \quad \dots \quad 1] \begin{bmatrix} a_{1,1} & \dots & a_{m+1,1} \\ \dots & \dots & \dots \\ a_{1,n+1} & \dots & a_{m+1,n+1} \end{bmatrix} \begin{bmatrix} \bar{h}^m \\ \bar{h}^{m-1} \\ \dots \\ 1 \end{bmatrix} \quad (3.5)$$

The v indicates the aerodynamic velocity norm and h indicates the altitude. The first index for the parameters a in the matrix refers to the height and the second to the velocity.

The idea is to interpolate with respect to one of the variables, in this case the velocity, each curve for a given value of the other and after interpolating the coefficient of the first set on the other variable. This way the matrix of the interpolating coefficients is obtained.

This method works good enough and allows to obtain a non linear expression of a multi-variable function with an high enough precision.

For the present case an interpolation order of 2 is chosen for both variables.

3.2 Control surfaces expression

Since the final method used to compute the aerodynamic effects is through a simulation that calculates forces and not through the available formula, the part relative to control surfaces in (2.3) and (2.7) has to be adjusted.

The expression 3.1 is used to compute the forces value when already having the coefficients. But since the forces are given directly by the numerical simulation, the expression to be used to include the control surfaces is the following,

$$\mathbf{F}_{aer}^b = \mathbf{F}_{GE}^b + \frac{1}{2}\rho S V_a^2 \begin{pmatrix} \Delta C_x \\ \Delta C_y \\ \Delta C_z \end{pmatrix} \quad (3.6)$$

where the coefficients relative to F_b are now called differently because they are mentioned in [10]. ΔC_x indicates the variation of the coefficients given by the variation of the elevator, ailerons and rudder angles, so these are kept in this form even though the aerodynamic coefficients are not used directly. In the formula $\mathbf{V}_a = \mathbf{V}_i$.

The following formulae are found in [10] and described as a good and broadly used for the quantification of the effect given by the control surfaces.

$$\begin{aligned}
 \Delta C_x &= -0.029\delta_e^2 - 0.04893\delta_r^2 - 0.113\delta_a^2 \\
 \Delta C_y &= 0.157\delta_r^2 \\
 \Delta C_z &= -0.32\delta_e - 0.0112\delta_r^2 - 0.1352\delta_a^2 \\
 \Delta C_l &= -0.07\delta_a - 0.04\delta_r \\
 \Delta C_m &= -0.923\delta_e + 0.20\delta_a^2 - 0.0055\delta_r^2 \\
 \Delta C_n &= +0.0035\delta_a - 0.072\delta_r
 \end{aligned} \tag{3.7}$$

where the control angles are in radians. The angles δ_r, δ_a , and δ_e are represent respectively the rudder, elevators and ailerons.

The surface S hereby represented is referred to as the “reference area” and in this case it is the wings area. By using this formulae the effect of the control surfaces can be add to the developed model with identified coefficients.

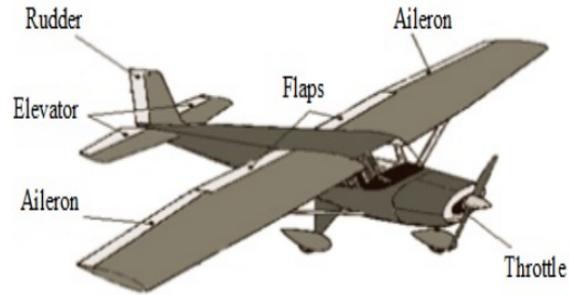


Figure 3.10: *control surfaces in an aircraft*

The figure above shows the control surfaces for a more general case, but the aforementioned ones are present in the figure, which is shown just to give an idea.

Chapter 4

Control

Different types of controllers are tested in order to evaluate their performances. The model is manipulated in order to use different strategies according to the scenarios.

The main purpose of a “simple” controller is to maintain a cruise condition as imposed by the path planner, which is not matter of the present work.

Other than simple stationary cruise control a non-linear MPC is a candidate for a full cruise control, given its high complexity, that makes it particularly suitable for this application.

Proving the non-linear MPC capable of obstacle avoidance by fly over would mean that the path planner wouldn't be required for simple manoeuvring, and that the controller could be used for an extended number of tasks.

Such tasks are theoretically feasible using the non-linear model, but seem quite hardly possible for simpler linear controllers, whose capabilities are by design limited by the used linearised model .

A complex MPC can theoretically perform both tasks. For this reasons the linear control strategies are mainly intended to be used as stationary controllers, even though their performances for more complex tasks are tested.

The present work is focused on the lower level layer of GNC (Guidance, Navigation and Control), as no guidance nor navigation is guaranteed if no control system is present. For now the main goal is to evaluate the controller performance for steady state cruising and fly over obstacle avoidance.

The control actions in [10], which are mentioned as being a broadly accepted version , are quite complex since they introduce more coupling between the inputs, and put extra strain in the optimisation.

This is especially true for the LQR controller, that wouldn't work with highly coupled inputs.

This complexity masks what are the difficulties of the controller itself and the ones related to the vehicle configuration. For this reason the method

employed for the controller analysis is to firstly considered the control actions as direct inputs in the form of forces and angular momenta that directly act on a given degree of freedom in the *body fixed reference*. In order to avoid corrupting the results quality, the input acting along the *z-axis* is considered zero, as it could make the controller act on a lack of lift with an input instead of by a change in the pitch angle, which is the main parameter for lift.

4.1 Non-Linear Model Predictive Control

This first attempt is a non-linear MPC controller to control the previously obtained model using the *do-mpc python library*[33].

A relative equilibrium position is sought for the reference values of $v_{i,x} = 30\frac{m}{s}$ and *altitude* = 1m. The pitch value is considered as a bounded state rather than a requirement. For this reason the pitch doesn't belong to the equilibrium requirements, but it is considered a free parameter, since it determines the lift force through 3.9.

An equilibrium configuration is found through the control loop itself and results being:

$$\begin{aligned} v_{i,x} &= 30\frac{m}{s} \\ \theta &= 0,045rad \\ h &= 1m \\ F_{b,x} &= 432N \\ M_{b,y} &= -56Nm \end{aligned} \tag{4.1}$$

The first control scenario is obstacle avoidance by fly over, starting from the just computed equilibrium position. The final altitude is the objective to be maintained in steady state. This value is considered 4m, which is quite of a leap starting from 1m.

The parameters used to express the performance of the controller are:

- rise time
- control cost from start until settling time
- settling time for 2%

The *control cost* is defined as:

$$c = \sum_{i=0}^n \int_0^{t_{s,2\%}} u_i(t)^2 dt \tag{4.2}$$

where n is the number of inputs present, $u_i(t)$ is the i -th input and $t_{s,2\%}$ is the settling time for 2%.

The characteristics of the controller are given by the objective function, which has the general form:

$$J(x, u) = \sum_{k=0}^N \left(\underbrace{l(x_k, u_k, p_{tv,k})}_{\text{stage cost}} + \underbrace{\Delta u_k^T R \Delta u_k}_{\text{input cost}} \right) + \underbrace{m(x_{N+1})}_{\text{terminal cost}} \quad (4.3)$$

where N is the prediction horizon, x_k the states, u_k the inputs and $p_{tv,k}$ are the eventual time varying parameters. Each term of the cost function J represent, respectively: *stage cost* represents the instantaneous expense evaluated at each time instant according to a custom function, which also contains the final states values; *input cost* solely focuses on inputs, quantifying the relative cost of each input through the weight matrix R ; *terminal cost* is the cost of the final reached state and therefore is evaluated solely in the final instant, allowing for a finer tuning of the behaviour by letting the function have different expression at the end of the horizon.

4.1.1 Altitude variation with direct inputs

The system is considered continuous time and the simulator time-step is set to 0,01s. The control loop runs each 0,1s, 10 times slower than the simulator, with a sampling time of 0,2s. Making it slower avoids superfluous computations, and the sample time allows to consider a wider time range without sacrificing performances. Preliminary tests have shown these values to be suitable and to be a good compromise.

First of all the effect of the horizon parameter of the MPC controller on the smoothness of the system transient is evaluated. An higher horizon value supposedly makes the overall dynamic smoother.

Since a steady state that consists in cruising condition at fixed altitude and velocity is sought, and since the model has 6 degrees of freedom with coupled states, the *objective function* must also include the lateral variable that also must be controlled for a steady state condition to be reached.

For this reason the quadratic cost function explicitly contains the *yaw* and *roll* angles in addition to all the components of the inertial velocity.

The latter is expressed in the cost function in terms of the longitudinal velocity along the axis x_i , the lateral velocity is the same along the axis y_i . The vertical component along z_i instead doesn't appear explicitly, but in terms of its integral, the altitude, which is also directly controlled.

Using the angles and the velocities might be redundant given that the latter depend on the angles through a rotation matrix. The presence of the angles is anyway useful for the transient phase.

The first form of the objective function is:

$$J = \int_0^{t_h} \left(\phi^2 + \psi^2 + (v_1^i - v_{1,f}^i)^2 + v_2^i^2 + (h - h_f)^2 \right) d\tau \quad (4.4)$$

where t_h is the prediction horizon, $v_{1,f}^i$ the final value of the first component of inertial velocity and h_f the final altitude.

The results obtain after a 30s simulation are shown in the picture below.

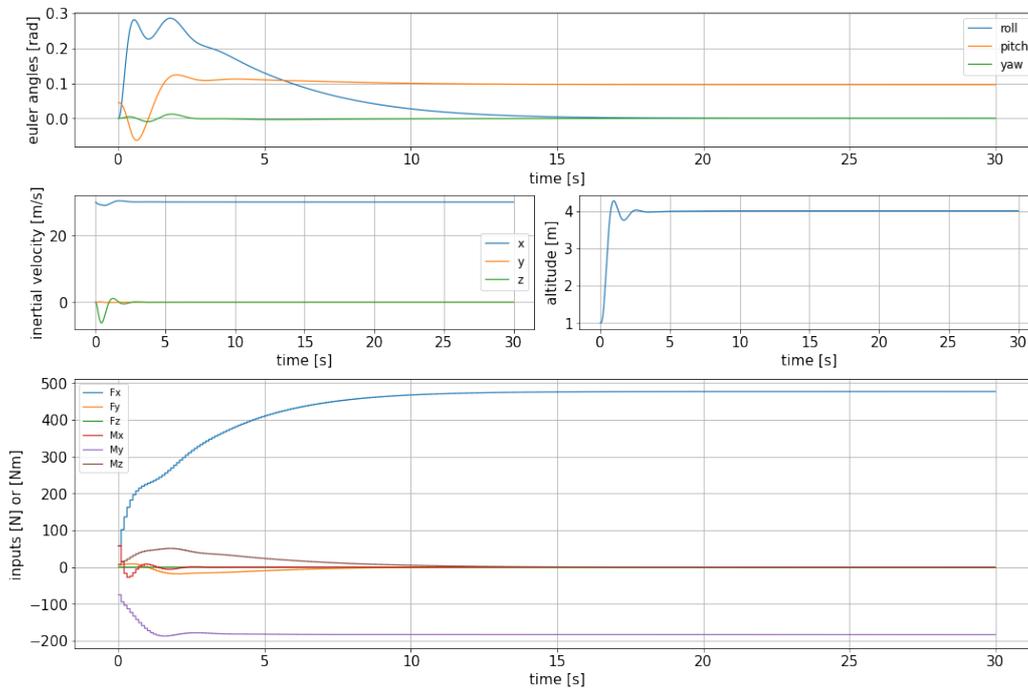


Figure 4.1: *full dynamic for altitude change at horizon 100*

Only one plot is shown since the differences between the different setups are limited. A table is shown instead to summarise the results obtain using different *horizon* parameter values, which acts on the smoothness of the controlled variables with a trade off on computational complexity, which cannot herein be evaluated.

The values contained in the table are the ones explained below, computed on the altitude.

	t_r	$t_{s,2\%}$	overshoot	cost
$h = 20$	0,76s	2,15 s	6,8%	1911,78
$h = 50$	0,76s	2,15s	6,7%	1922,78
$h = 100$	0,76s	2,15s	6,7%	1941,83

The rise time is considered crucial for this kind of manoeuvre, since in obstacle avoidance the important is to reach a sufficiently high altitude that allows for fly over, with stress solely about the celerity and the effectiveness.

As the table shows the altitude isn't influenced at all by the horizon. Various tests have shown that the horizon almost only affects the pitch transition to steady state, with little to no effect on the other states.

From the figure below the effect on the pitch, in terms of oscillations, can be appreciated.

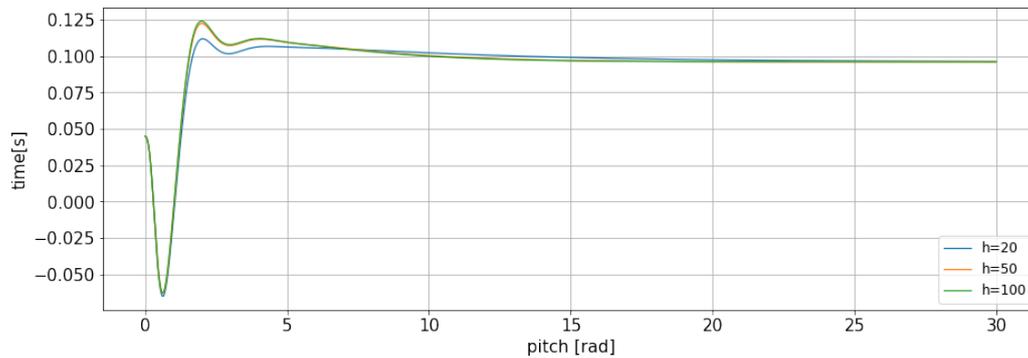


Figure 4.2: *pitch dynamic with different time horizon values*

In the case of $h = 100$, as the horizon time is longer than the steady state time, increasing it has no impact on the system response. For this reason the horizon can be limited to $h = 50$, which is roughly the steady-state time, in order to optimise the computational results in MPC results.

As said before, the pitch value determines the *lift* and *drag* forces in conjunction with the other states, so it is crucial for the reach of a steady state.

Given that the oscillations of the pitch angle are mainly a comfort concern it still might be worth using the wider horizon value in order to smooth the pitch dynamic.

The main decisional parameter for the horizon value is the computational cost, that can greatly affect the computational time in case of hardly converging differential equations. For now such evaluations can only be qualitative, since a proper *hardware-in-the-loop* simulation would be needed in order to have data about the timing.

Another attempt is made, tuning the weights based on the previous simulation. The main problem with it is the strong overshoot in roll, which brings it close to $0,3rad$, much higher than the other angles.

This is caused by the different order of magnitude angles and velocity components have. This can be taken into account for by adding proper weights into the objective function.

$$J = \int_0^{t_h} \left(10000\phi^2 + 1000\psi^2 + 0,5(v_1^i - v_{1,f}^i)^2 + v_2^{i2} + 50(h - h_f)^2 \right) dt \quad (4.5)$$

The weights evaluation is done by following the principle of normalizing the order of magnitude of every variable first, then modifying it to assign penalties in order to shape the dynamic in the desired way. The difference between the weights of ϕ and ψ is set based on the obtained results, since the first presents a faster and more delicate dynamic.

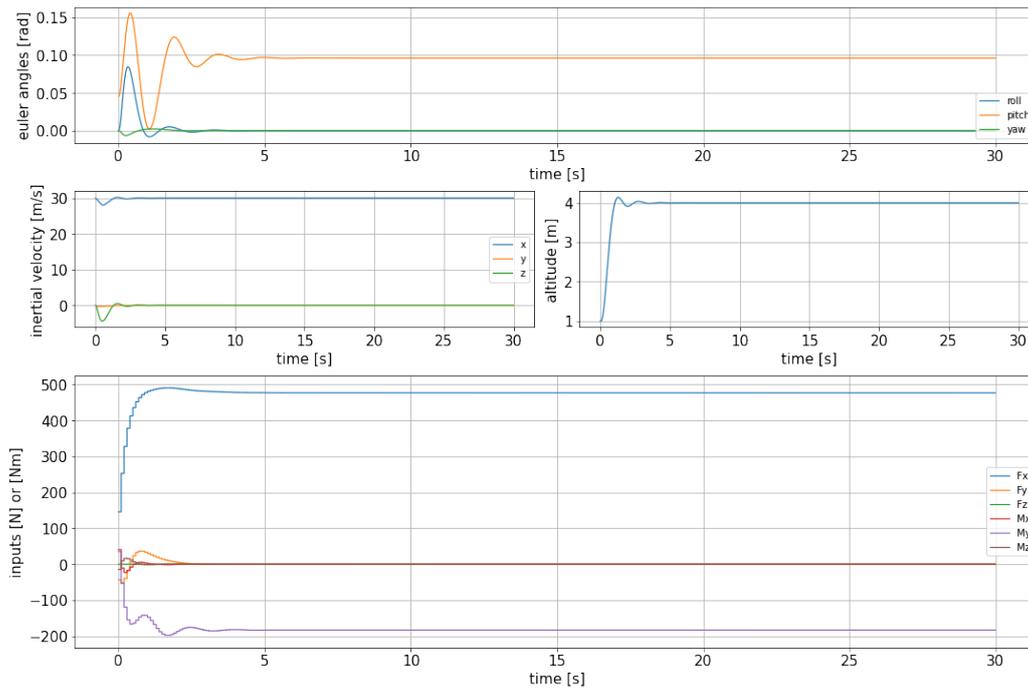


Figure 4.3: *objective function with calibrated weights*

	t_r	$t_{s,2\%}$	overshoot	cost
$h = 50$	0,85s	1,89 s	5,1%	5908,1

Modifying the weights yields to a much cleaner dynamic in terms of Euler's angle, which were the main concern. The overshoot on ϕ is greatly

reduced, solving the bumpiness problems. Overall the dynamic results much smoother with less worrying oscillation, and even if the parameters show a slight deterioration the system performs for better overall.

As a final assessment of the non-linear MPC controller a simulation is tested adding measurement noise. The employed noise is uniformly distributed with a value of $\pm 5\%$ on the state value fed to the controller. The objective function is as (4.5).

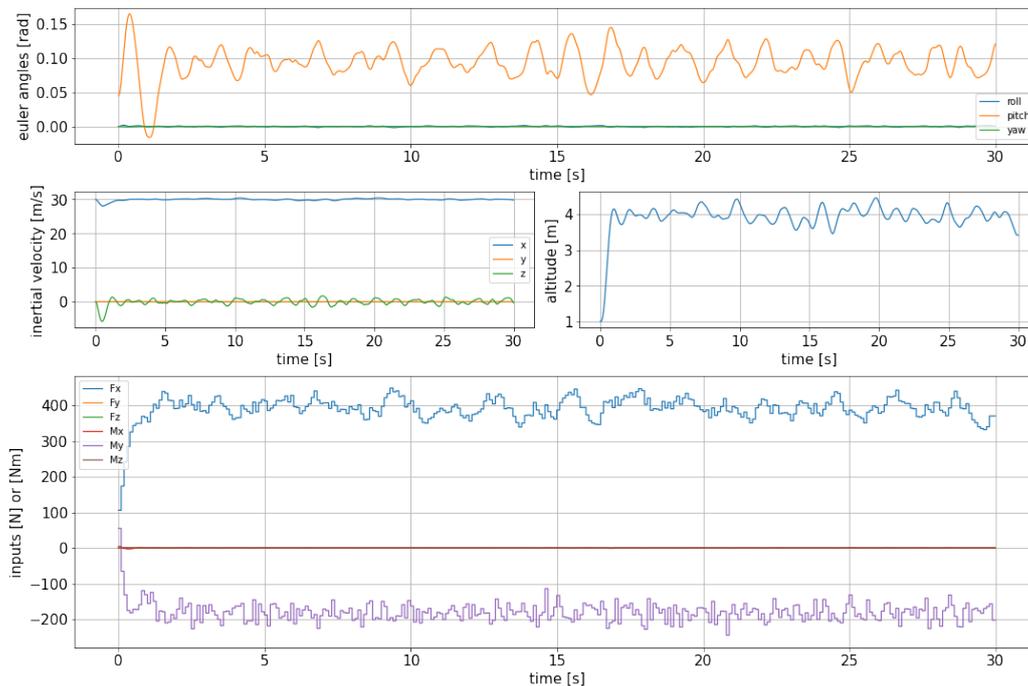


Figure 4.4: *dynamic for altitude change with uniform measurement noise of $\pm 5\%$*

	t_r	$t_{s,2\%}$	overshoot
noise	1,78s	no	5,0%
no noise	0,85s	1,89 s	5,1%

In this case the measurement noise has a strong effect on the simulation. It extends more than the noise itself, causing oscillations that are even 50% compared to the steady state value of the *pitch* angle. The error probably causes a greater prediction error, which is amplified.

The figure 4.5 show the amplitude of the induced oscillations and the maximum values those have reached in the simulation. The same doesn't quite apply to the altitude oscillations, that stay more bounded at a value of around 12,5%, still more than the noise effect.

This behaviour is consistent in the two analysed variables of interest, and show the need for a robust component in the MPC formulation, which isn't implemented yet.

Coupling the MPC with a *moving horizon estimator* should improve the performances cancelling the undesired effect of the noise, a least reducing it.

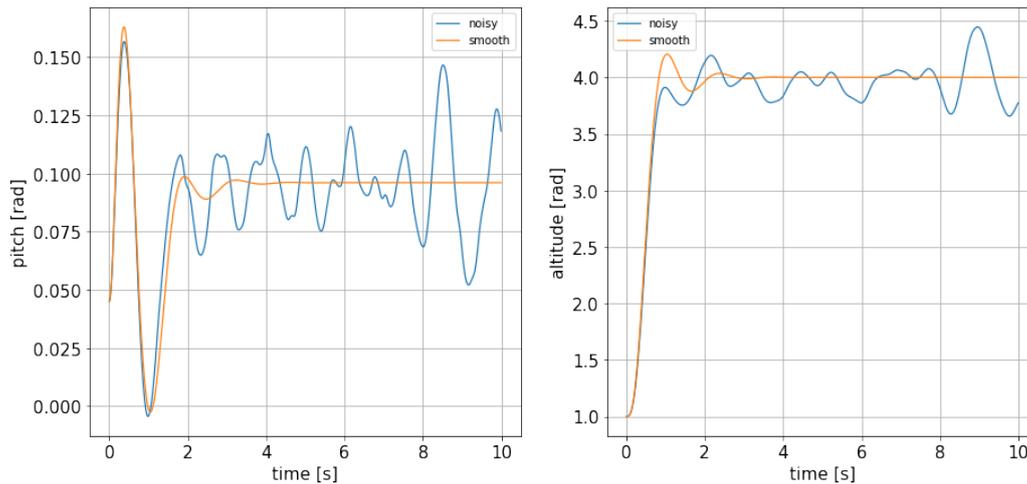


Figure 4.5: *comparison between the regular case and the one with noise*

The evaluation parameters are quite insignificant in this case, since the analysis shows that the noise induced oscillations are bigger than the bounds for the settling time and that the rise time varies stochastically. In this cause a negative noise before the rise is most likely cause of the measured delay between the two cases.

4.1.2 Altitude variation with control surfaces and motor as inputs

The next attempt is made by switching to expressions of the control surfaces expression found in [10] that are going to introduce some other complications to the control design. The first difference is the introduction of the real control inputs, which are represented by the motor, which solely generate force alongside the first axes of the fixed body reference frame.

The surfaces are called *elevators*, *ailerons* and *rudders*. Their range of mobility is considered $\in [-\frac{\pi}{2}, \frac{\pi}{2}]$.

Given the different order of magnitude of the two type of inputs the weight are assigned to even them out. This translates into using a weight of 100 for the surfaces and 1 for the motor.

The cost function is (4.5).

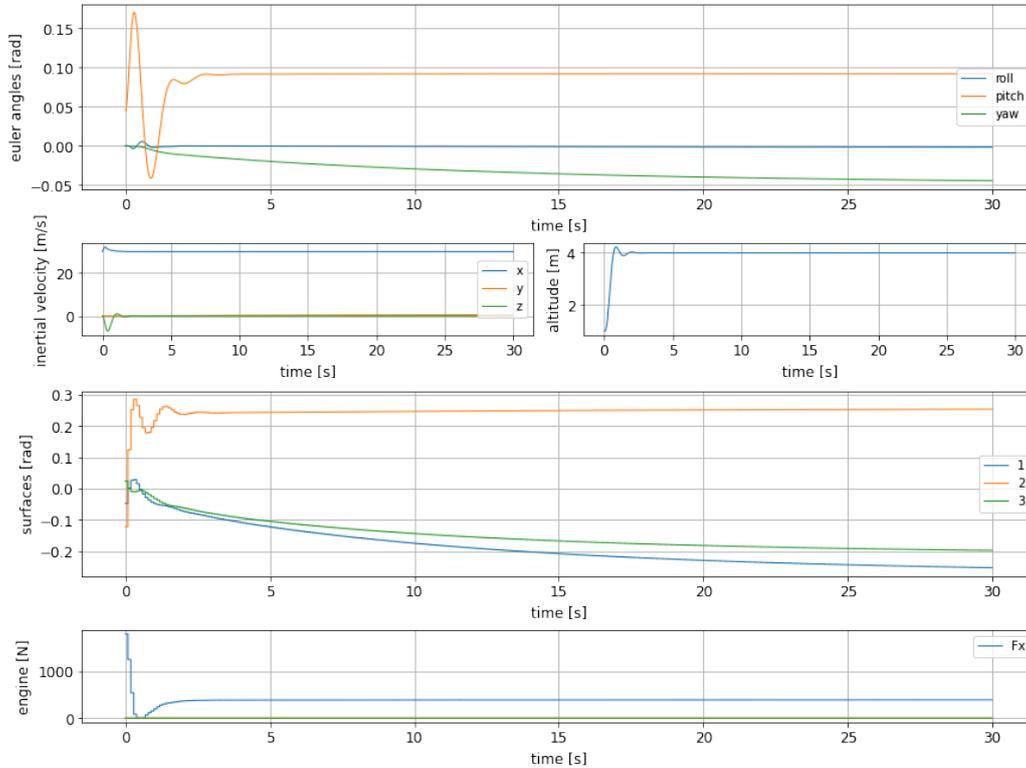


Figure 4.6: *altitude change with full control surfaces input*

	t_r	$t_{s,2\%}$	overshoot
$h = 50$	0,69s	1,55 s	5,6%

The overall performances are not penalised in terms of evaluation parameters compared to the previous case. The rise time is actually smaller, at the cost of significantly increased oscillations in *pitch*, but since the important variable is the *altitude* results are satisfactory.

The only problem is that the yaw angle has what seems a steady state error, which doesn't generate a big deviation in the inertial speed, since it is not being actively corrected given the small weight of this variable in the objective function.

Another simulation with a different tuning is done, trying to adjust the problems that appear in the previous one, which is about the fast oscillations in the *pitch* and apparent lack of steady state zero error in Euler angles.

To try correcting this some tweak are done to the objective function and other parameters. First of all, a slightly different approach is used, trying to reduce unnecessary and unwanted oscillations.

The approach about to be shown consists in defining a function that relates the altitude equilibrium value to a pitch angle, for a given speed. Doing so, the controller can be set for the final value of the pitch based on the final altitude. Presumably this will make the pitch angle oscillate less, leading to a smoother dynamic.

A function with the purpose of relating these variables can be obtained through the static equation of vertical motion in the inertial frame, which described the equilibrium

$$F_{lift}(\theta, h) - mg = m\ddot{h} \quad (4.6)$$

by imposing $\ddot{h} = 0$ and using the expression for the lift force computed in chapter 3 for a fixed cruise speed, a function of θ and h is obtained.

$$(\alpha\theta + \beta)(ah^2 + bh + c) - mg = 0 \quad (4.7)$$

where α , β and a, b, c are the coefficients of the previously interpolated polynomials 3.1.1.

by solving numerically, a function $f(\theta) = h_{eq}$ that expresses the equilibrium altitude given a θ values is found.

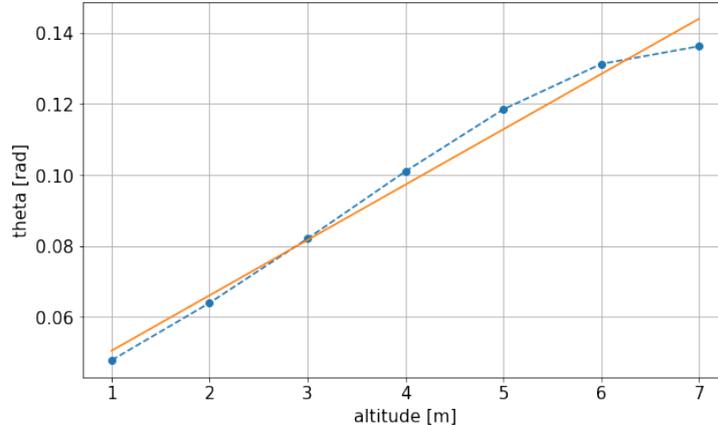


Figure 4.7: *obtained function, interpolated and scattered values for the vertical equilibrium position*

Using this data a target *pitch* value can be computed for the final *pitch* given the final *altitude*. In this case, setting the final altitude at $4m$ as before, the result is $\theta = 0,10rad$.

Including this information in the objective function it becomes

$$J = \int_0^{t_h} \left(10000(\phi^2 + (\theta - \theta_f)^2 + 0,1\psi^2) + 0,5(v_1^i - v_{1,f}^i)^2 + v_2^{i2} + 50(h - h_f)^2 \right) dt \quad (4.8)$$

In addition to this, the bounds on the state values are tuned compared to the previous cases, $\in [0; 0,110]$, where before they were $\in [-0,350; 0,350]$.

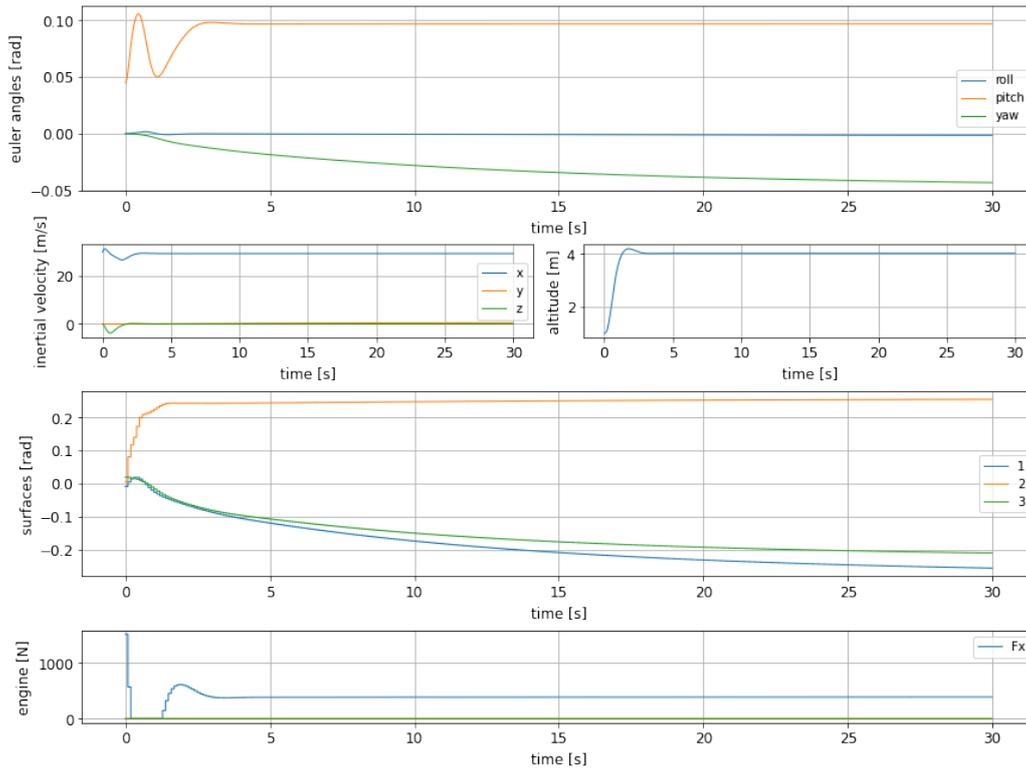


Figure 4.8: altitude change with added bounds on pitch

	t_r	$t_{s,2\%}$	overshoot
no pitch	0,69s	1,55 s	5,6%
pitch	1,33s	2,41 s	4,6%

The dynamic shows an improvement where it was sought, at the cost of slower performances. The pitch shows an acceptable oscillation and the rise is still short and acceptable. It even stays bounded and doesn't reach distant values.

The difference in terms of performance parameters is not negligible, and would impact the choice of a controller version over the other.

Still, the ideal case where the vehicle changes pitch in order to slowly rise is not achieved.

4.1.3 Effect of mass on MPC performances

Now it the turn changing another important parameter for the design of an ekranoplan, that is the total mass of the vehicle.

Until now it has been assumed to be a values that squared nicely with the aerodynamic forces, which happens in the design process. In fact it strongly depends on constructive capabilities and materials that will be used for prototyping, which is still a great unknown factor. For now the mass of the vehicle can only be estimated and some data about its effect generated for a better knowledge of the less obvious effects it has.

Some simulations with different total masses and inertia tensors are now considered.

	t_r	$t_{s,2\%}$	overshoot
90 kg	0,69s	1,55 s	5,6%
135 kg	0,74s	1,67 s	6,2%
180 kg	0,79s	1,75 s	6,4%

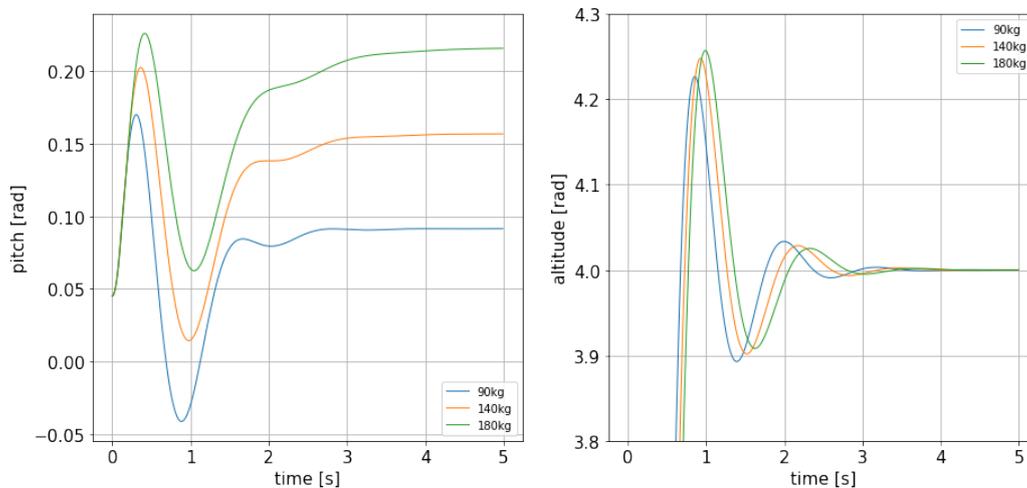


Figure 4.9: *comparison between different masses*

The comparison shows how the dynamic is basically translated as the mass changes. The pitch value that makes a given position a vertical equilibrium position is not any more the same as 4.7, since it was computed for the initial mass values.

It can be observed that the altitude dynamic gets translated to the right, an effect of the added weight, that introduces a delay. The *pitch angle* needed for the equilibrium becomes higher, so the final value increases to one able

to generate enough lift force for the given weight. This might be a crucial constrain that could be impose a specific manufacturing technique in order to obtain a well balanced vehicle model.

4.1.4 Cruising from an equilibrium position

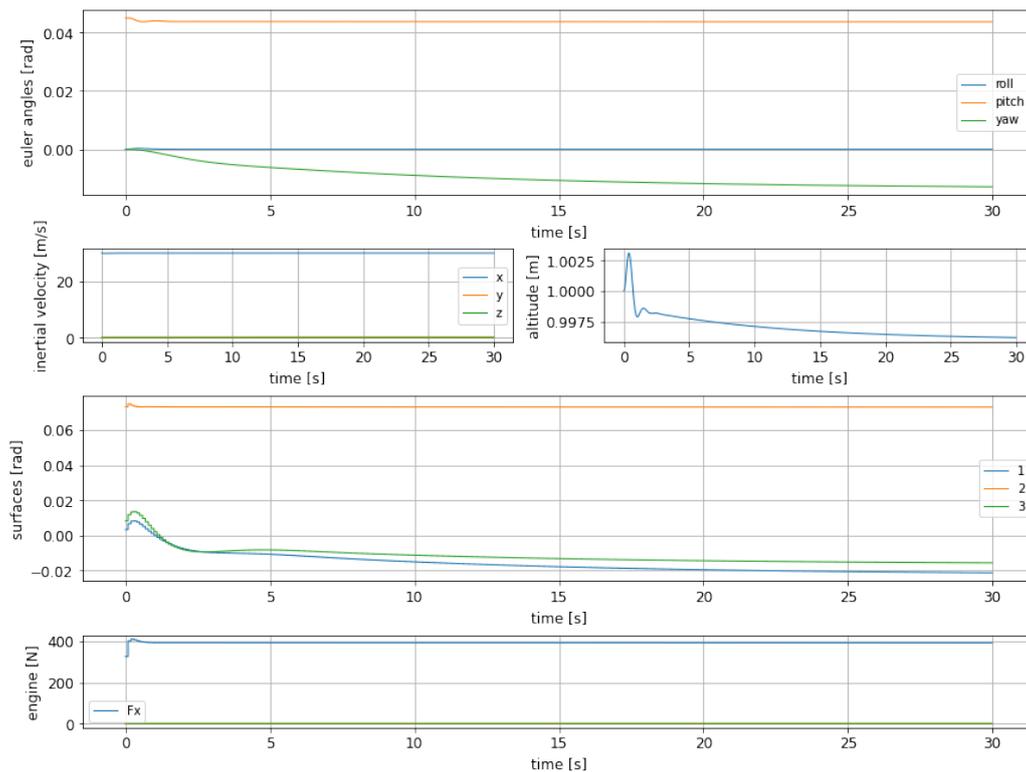


Figure 4.10: *Stationary cruising from equilibrium position*

Given the previously obtained results this seems a rather simple case, but it's worth trying.

This situation represents the arrival point at the end of a manoeuvre, when the vehicle has to maintain a steady cruising state.

The only noticeable stuttering is created by the initial settling required by the controller at the beginning of the action, that only creates minor oscillations in altitude, which are anyhow bounded to 1cm and not problematic at all.

The yaw angle instead attains a values which is not perfectly zero, and isn't adjusted in the displayed time horizon. This is not a real concern, since

such angle is really small and not big error in the inertial speeds is registered anyway, otherwise it would get corrected by the controller.

4.1.5 Recovering from a perturbed position

This last control scenario using the MPC shows a borderline feasible case for situations such as a significant displacement from cruising conditions that needs to be restored as soon as possible. The MPC has shown great performances in the previous cases, so good results are expected here too.

The initial position is chosen to be

$$\begin{cases} \phi_0 = 11,46^\circ \\ \theta_0 = 6,01^\circ \\ \psi_0 = -11,46^\circ \\ h = 1m \end{cases} \quad (4.9)$$

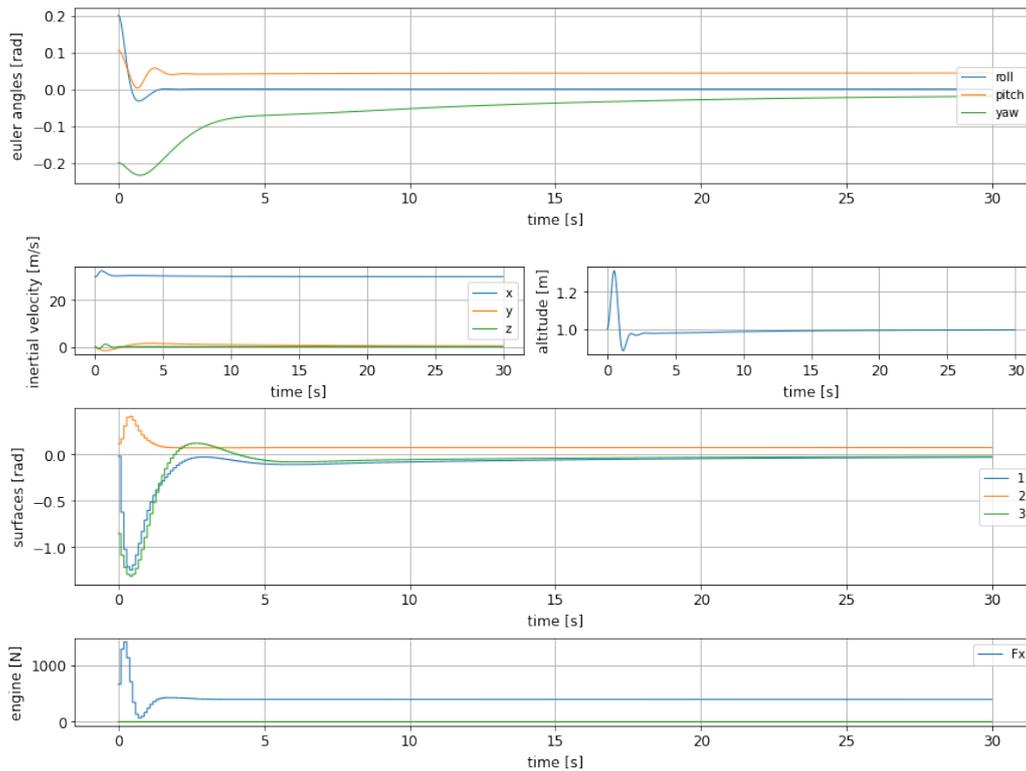


Figure 4.11: *steady state reaching from a random unsteady position*

The evaluation parameters are changed a bit here, to take into account

important aspects of the altitude behaviour. In particular the *undershoot* one is added, which measured how low, percentage-wise, the altitude goes before reaching the prefixed steady state. It basically measures the minimum in the oscillation as the *overshoot* measures the maximum.

	$t_{s,2\%}$	undershoot	overshoot
MPC	6,84s	11,8 %	31,4%

In this case the important is that the altitude doesn't go too low, which would be a safety concern. It appears to be quite well bounded in terms of oscillations. The Euler angles are quite well bounded too, and have a soft transition to the steady state value.

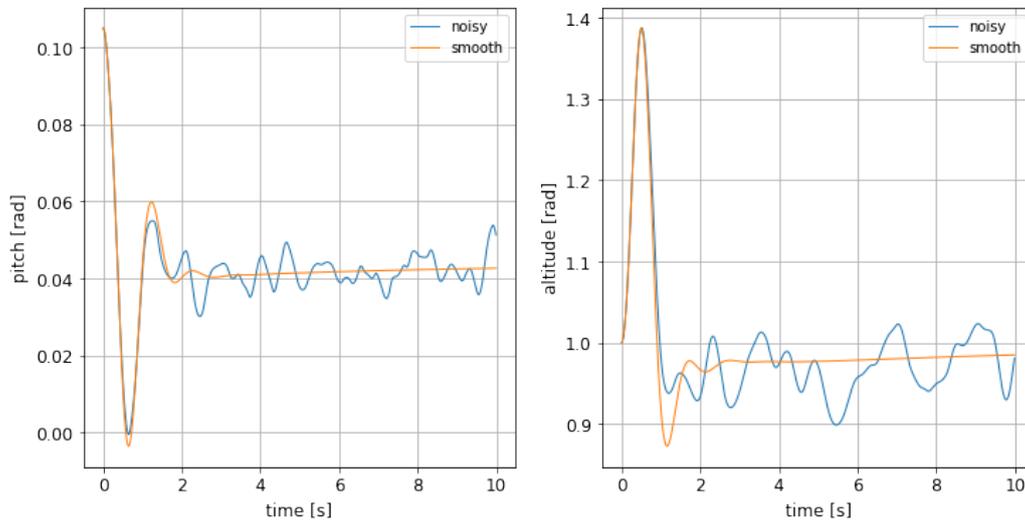


Figure 4.12: comparison between the cases with and without noise for the MPC recovery from a perturbed position

The behaviour in the presence of measurement noise is less influenced in this case compared to the altitude change. Of course the error depends on the value of the state, being it multiplicative. Since the error seems to affect less the cases where *altitude* and *pitch* angle are lower in value, particularly the first one, it is confirmed that problems at lower altitudes are less likely to happen, ensuring a more robust behaviour.

4.2 Feedback linearisation

Recalling the dynamical equations general form

$$\begin{cases} m(\dot{\mathbf{v}}_b + \boldsymbol{\omega} \times \mathbf{v}_b) = \mathbf{F}_m \\ \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} = \mathbf{M}_m \\ \dot{\mathbf{x}}_i = \mathbf{T}_{ob}\mathbf{v}_b \\ \dot{\Phi} = \mathbf{R}\boldsymbol{\omega} \end{cases} \quad (4.10)$$

where \mathbf{v}_b represents the vector of the velocities with respect to body frame, $\dot{\mathbf{x}}_i$ the vector of the velocities with respect to inertial frame, \mathbf{F}_m and \mathbf{M}_m respectively the forces and angular momenta exerted by the inputs and Φ the Euler angles.

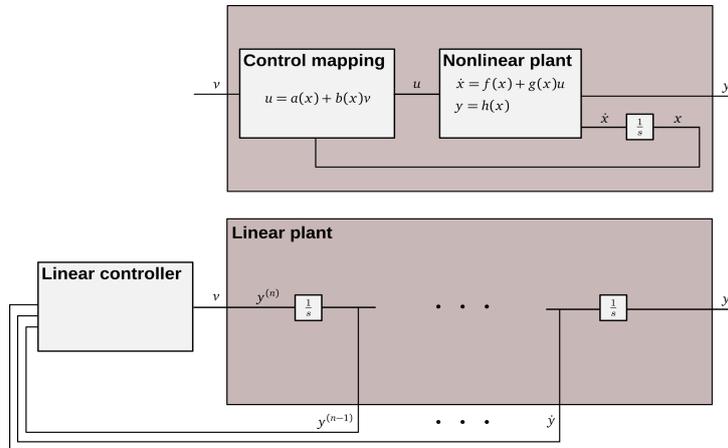
The principle of input-output linearisation is to choose the inputs in order to obtain a linear relationship between a given input and a specific state. This represents the most easily attainable kind of feedback linearisation, given that the requirements for a input-state linearisation are more stringent and seemingly not present in this case.

This system can be easily linearised by choosing the inputs \mathbf{F}_m and \mathbf{M}_m so to cancel the coupling in the body referred equations. Since fine altitude control is essential for ekranoplan control this cannot be omitted from the model. Unfortunately, given the existing relationship between the accelerations and *body frame* and the velocities in *inertial frame*, it is not possible, empirically speaking, to obtain a input-state linearisation, because of the transformation in the third and fourth of (4.10).

The problem is that the transformation matrices \mathbf{T}_{ob} and \mathbf{R} introduce significant non linearities that cannot be addressed with dedicated inputs, since those are referred to body frame and not to the inertial system.

Even though state-input linearisation is not achievable, a partial one is still useful.

Compared to regular series expansion linearisation, the big advantage of the feedback one is that it is valid in a significant range of values at least, and not in the neighbourhood of a single chosen point. This will limit the error to the one of the simplification introduced next, making it more feasible and reliable compared to the classical case. Of course the quality of the process is subject to the sensors used in the real case for the compute of the values to be fed back. Right now such considerations are not going to be made and the simulations will not include this effect, given that there are no informations about such process apart from a general perspective on the alleged problems caused by it.

Figure 4.13: *feedback linearisation block scheme*

What can be done is adding a noise that mimics the sensor noise in a multiplicative sense, as was done before for the MPC.

4.2.1 Linearisation process

The steps required to get to a usable form model is now shown.

The body referred equation can be simply linearised, as is shown for the first angular velocity equation from (4.10).

Given the expanded form (2.7) of the first component of angular velocity in body frame

$$A\dot{p} = E\dot{r} - rq(C - B) + Epq + F_{aer,x} + M_{mot,x} \quad (4.11)$$

By choosing the input

$$M_{mot,x} = -(E\dot{r} - rq(C - B) + Epq) + M'_{aer,x} \quad (4.12)$$

The equation (4.11) becomes

$$\dot{p} = M'_{m,x} \quad (4.13)$$

Applying the same principle to all other body referred acceleration equation equations the result is

$$\begin{cases} \dot{u} = F'_{m,1} \\ \dot{v} = F'_{m,2} \\ \dot{w} = F'_{m,3} \end{cases} \quad \begin{cases} \dot{p} = M'_{m,1} \\ \dot{q} = M'_{m,2} \\ \dot{r} = M'_{m,3} \end{cases} \quad (4.14)$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \sin \bar{\theta} \tan \bar{\theta} & -\sin \bar{\theta} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\cos \bar{\theta}} & 0 & 0 & 0 & 0 \\ -\sin \bar{\theta} & 0 & \cos \bar{\theta} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.18)$$

The matrix B is simply

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.19)$$

To evaluate the performances a perturbed initial position is randomly chosen. This will allow for an evaluation of the performances of the controller in this limited case.

It is important to highlight that the equilibrium position is valid as long as the Euler angles are close to the linearisation point, but the other states do not interfere with the linearisation, since the feedback cancels those effects.

This kind of controller acts quite differently compared to the MPC, because the model is simplified and the full dynamic is not visible. While the MPC mainly relies on a change in θ in order to generate enough lift force for the altitude to change, in conjunction to a proper arrangement of the surfaces so that the resulting force is mainly along z_i , the model used for LQR doesn't include such part of the dynamic. This leads to the speculation that such controller is suitable for cruising, since the required control actions are limited compared to manoeuvring, but might not be efficient for other more sophisticated purposes.

On top of this, given that the LQR doesn't allow for constraints as the MPC does, the range of effectiveness has to be evaluated carefully.

The underlying *feedback linearisation* makes it quite easy to design this controller without having to fully linearise the model as in (1.7).

The perturbed position is (4.9), the same as before for comparison purposes.

The error compared to the cruising configuration is rather small, but still present on every component of the states.

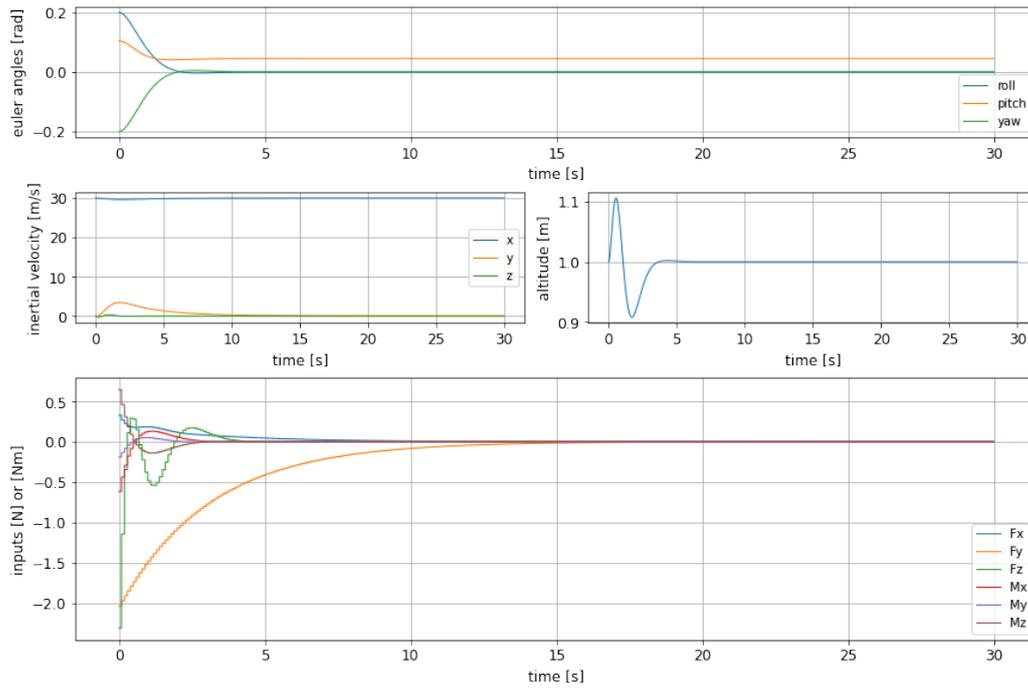


Figure 4.14: *LQR control on linearised system, starting from perturbed position*

The used *weighting matrix* Q is

$$Q = \text{diag} [0, 1 \ 0, 1 \ 0, 1 \ 1 \ 1 \ 1 \ 10 \ 10 \ 10 \ 100] \quad (4.20)$$

The weights are ordered as the states in (4.15).

As expected the mainly used control action is the force along z_b , that generates a variation in altitude, which presents a strong overshoot.

A lateral damped non-oscillating force is required along y_i to correct the lateral velocity component generated by the initial non-straight direction of the ekranplan.

The performances of the controller on Euler angles is smooth and acceptable.

A simulation is run including measurement noise of $\pm 5\%$ on the state values. This means that the values fed to the controller are not the real ones, but present an error.

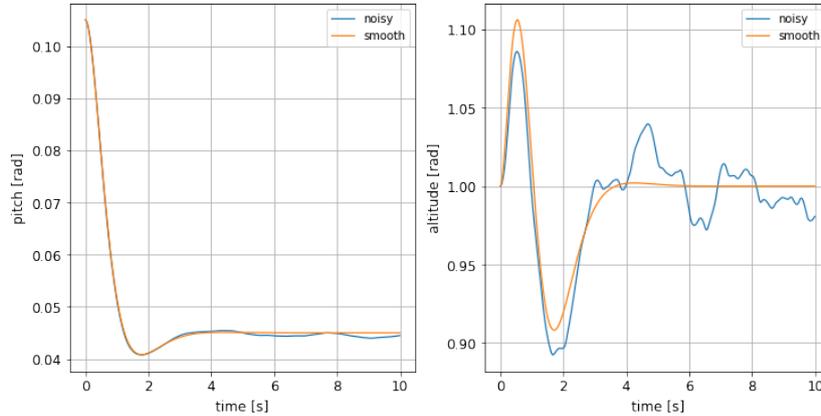


Figure 4.15: *comparison between the cases with and without noise for the LQR for recovery from perturbed position.*

The error doesn't affect much the transient of the system, leading to comparable behaviour, slightly different but not translated in time. During the steady state instead it creates an offset in the state, creating slight oscillations that do not create problems.

4.2.3 Updating LQR over feedback linearised model for altitude change

A strategy for improving the performances of the LQR is to give it adaptiveness by linearising the model as the dynamic proceeds.

Given that the A matrix is of the form (4.16), it can be recomputed during the simulation loop to the new updated position.

This allows to stretch less the approximation, keeping the controller closer to the linearising point. This kind of adaptiveness can be tuned according to the requirements. The first and most important could be the computational constrain, which, as said previously when referring to the MPC, has to be assessed separately, since another level of simulation is needed.

The initial position is the usual equilibrium position for $h = 1m$ (4.1) and the sought position is $h = 4m$, which requires a pitch angles of about $\theta = 0,10rad$.

This *pitch* value is computed through 4.7 and is manually set as the final target, as it is needed for the present case. It could be avoided and left to

the initial values of $\theta = 0,045rad$, given that the equations don't require an equilibrium position to be found. Instead it is done manually given the experience with the MPC, which reached a high enough *pitch* in order to save input cost increasing the pitch.

This could even be a feasibility matter, as the surfaces might not generate enough lift for the initial *pitch* value to be enough. Apart from the feasibility, it is clear that the pitch change makes the control process more efficient and requires less effort, so the behaviour of the MPC is imitated by manually setting the values.

The feedback linearisation hides the cost of the linearisation, so this other layer of control is implemented to limit it, confident that it will be realistically useful in real control scenarios once this particular control strategy will be applied in the real case using sensors and such.

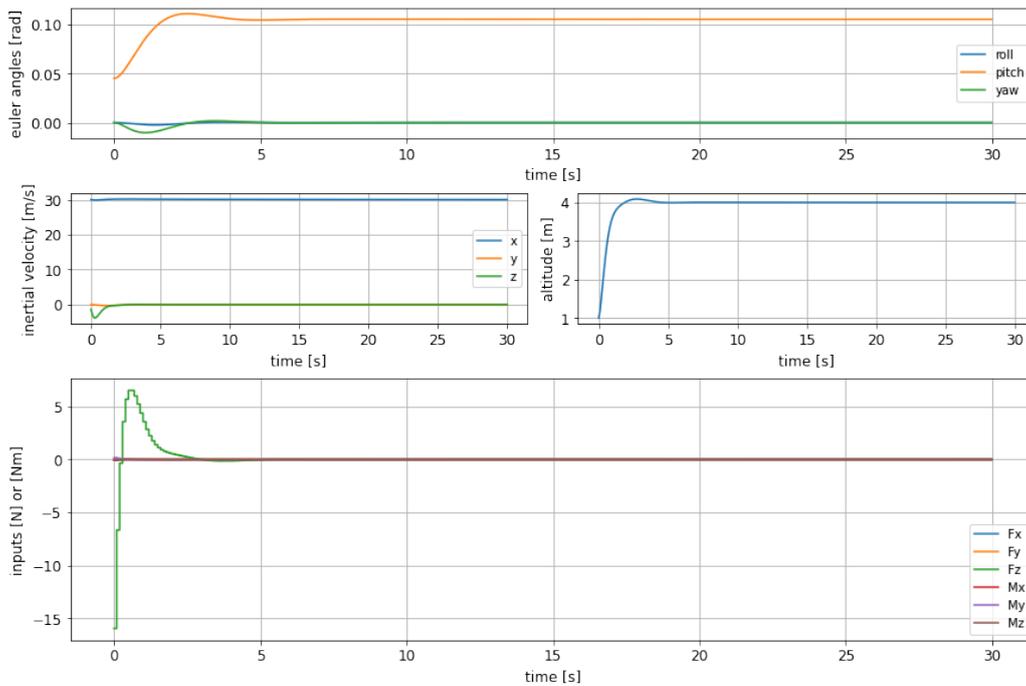


Figure 4.16: *altitude change using LQR*

The used *weighting matrix* Q is

$$Q = \text{diag} [0,01 \quad 0,01 \quad 0,01 \quad 1 \quad 1 \quad 1 \quad 10 \quad 10 \quad 10 \quad 100] \quad (4.21)$$

	t_r	$t_{s,2\%}$	overshoot
uLQR	1,90s	3,04 s	2,2 %

In absence of noise or perturbations the controller almost doesn't overshoot in θ and the overall transition is quite smooth. It is important notice is that the *weight* for the altitude state is set to 100, 1.000 times more than the weight of the velocity components and 10 times more than the angles one.

Comparing the evaluation parameters the MPC is better if the objective is obstacle avoidance in the shortest possible time, but the LQR on the feedback linearised model might provide a smoother transition between two different cruising conditions when celerity is not crucial.

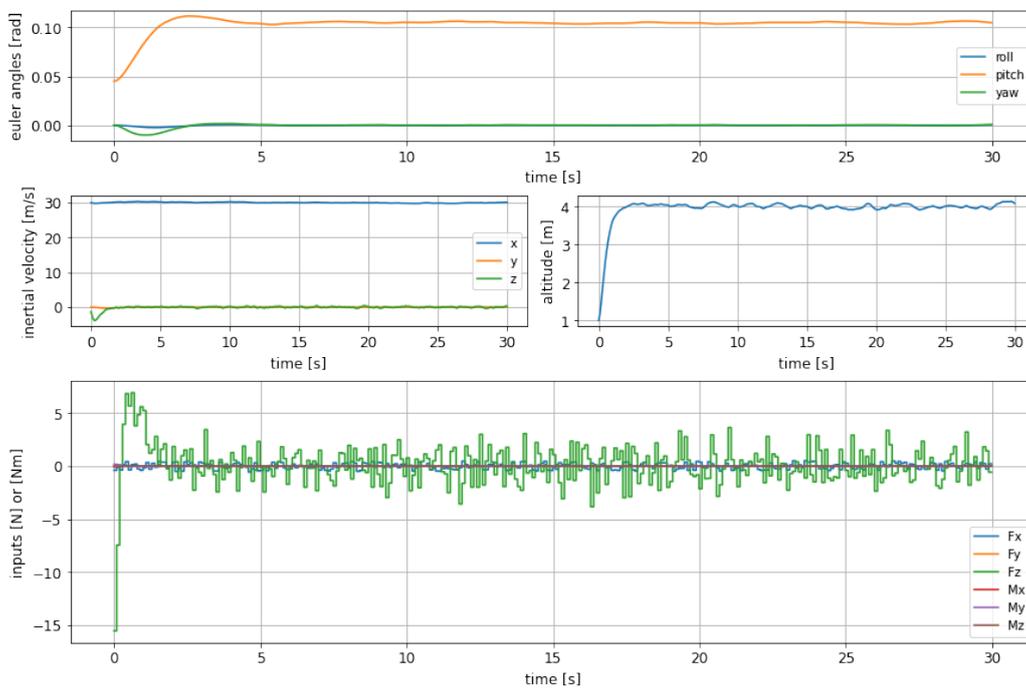


Figure 4.17: *altitude change using LQR, with measurement noise*

This last simulation in figure 4.17 is the same as 4.4 for the LQR case.

It is quite evident how the noise seems to affect much less the dynamic, by an order of magnitude that is at most the one of the noise, without creating increased oscillations as seen in 4.5. This is probably due to the linearity of the controller, that relies on a simple multiplication to determine the suitable input, once the *gain matrix* has been determined.

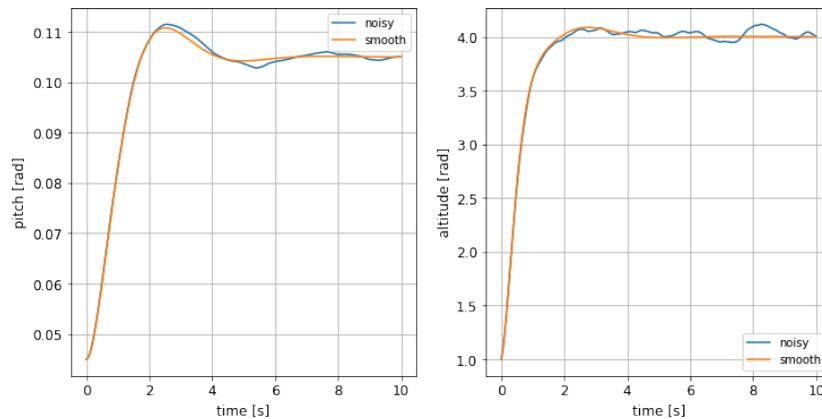


Figure 4.18: *LQR for altitude change: comparison between the behaviour with and without noise*

4.2.4 Updating LQR for recovery from perturbed position

This initial position is the same as the previous case shown using the regular LQR controller, in order to later compare the results. Since the LQR updates itself every cycle, the distance from the linearisation position isn't, in theory, a problem at all.

(4.11)

The weights are the same as (4.21).

Only successful simulation are shown in the present work, but some unsuccessful ones are worth be mentioned. In this case it is important to explain a phenomenon that presented in the same scenario using the simple LQR.

Since the LQR is set to use a simplified model, which makes of course ceases being acceptably inaccurate in configurations like the one above, and since the inputs act independently of each other, the distorted values given by the linearised make the final value of states like the altitude appear reached. This causes the controller to state and the steady state is not reached. Therefore the classic LQR has a rather limited range of effectiveness, that isn't determine by how well it works but if it works.

Meanwhile, in the case of the updating LQR the objectives are reached, even though the required time is quite long.

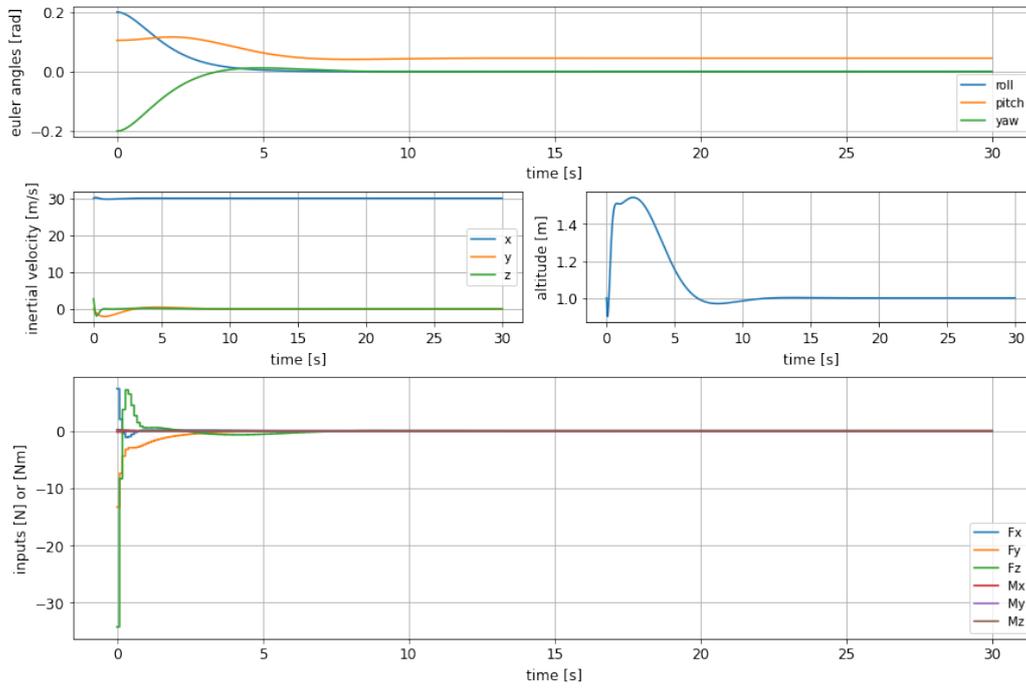


Figure 4.19: *LQR controller for recovery from perturbed position*

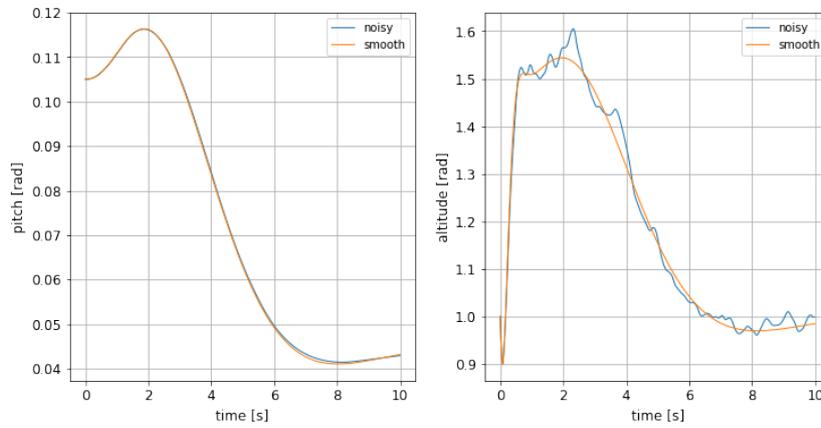


Figure 4.20: *comparison between the noisy and regular case for refreshing LQR*

4.3 Comparison between the employed control strategies

This section is dedicated to the analysis of the performances of the controllers comparing the different strategies for the same task, through evaluation parameters and plots.

4.3.1 Comparison between MPC, uLQR and LQR for recovery from perturbed position

The comparison between the three different control strategies for recovery from a perturbed position show some surprises for sure, and allows for an evaluation of the two different LQR used.

The undershoot of the three controllers is comparable, but the other two parameter are quite distinct to each other.

The updating LQR shows the biggest overshoot amongst the three, over 50%. Even the MPC has an higher overshoot than the regular LQR. Most importantly the MPC is affected by the lack of perfect steady state tracking, a problem likely caused by the complicated expression of the input surfaces. If it weren't for this it would be quite faster than the LQR, but instead its settling time is penalised strongly by this behaviour.

The LQR is for sure the cleanest of them all, with a bounded oscillation and fast recovery time. Even the *pitch* behaviour, which is of secondary importance, is much smoother. It for sure wins for this scenario, where the more complex updating LQR doesn't seem to show any tangible benefit over the simple LQR.

	$t_{s,2\%}$	undershoot	overshoot
LQR	2,87s	9,2 %	10,6%
uLQR	9,57s	10,0 %	54,4%
MPC	6,84s	11,8 %	31,4%

The values in the table are referred to the altitude.

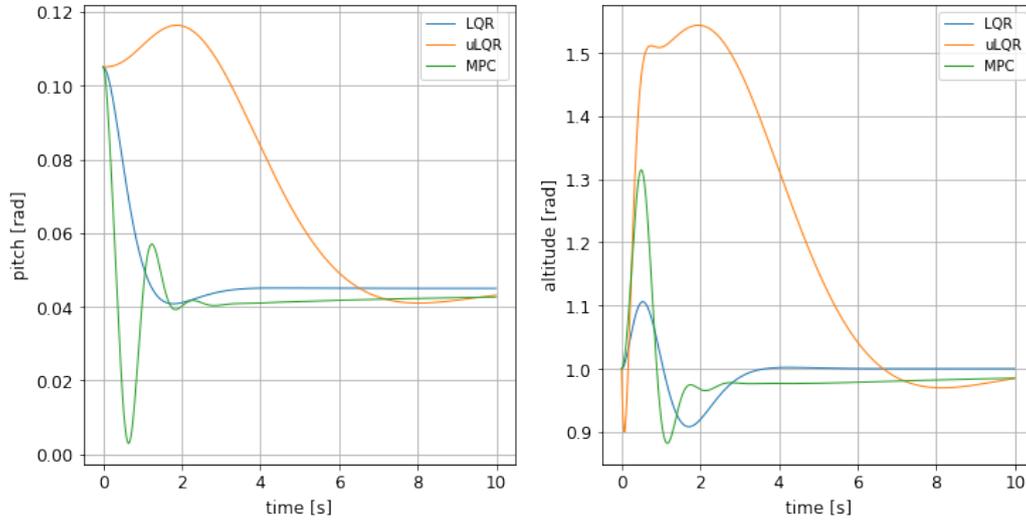


Figure 4.21: comparison between the altitude and pitch dynamic for LQR, updating LQR and non-linear MPC

4.3.2 Comparison between MPC and LQR for altitude change

This last comparison shows the difference in performances between the non-linear MPC that implements the pitch angle in the cost function (4.8) and the updating LQR.

Altitude change has shown to be a challenging task for the controller, and the different behaviours show different attitude of them. The MPC that makes use of (4.8) has been picked for the comparison because its smoothness is better compared to the one obtained through the use of (4.5) and compares better to the LQR whilst still maintaining a shorter rise time.

The clear difference between the control strategies is highlighted in 4.22, where the usual comparison between *pitch* and *altitude* is shown.

	t_r	$t_{s,2\%}$	overshoot
MPC	1,33s	2,41 s	4,6%
LQR	1,90s	3,04 s	2,2 %

From a speed point of view the MPC is the obvious winner, given the sensibly shorter rise time. It is important to remark that the LQR makes use vertical force to lift the ekranoplan, and the Riccati problem wouldn't be solvable if the 3rd component of the force inputs were removed. This effect cannot be avoided since the LQR cannot be forced to use a *pitch* variation as the MPC does.

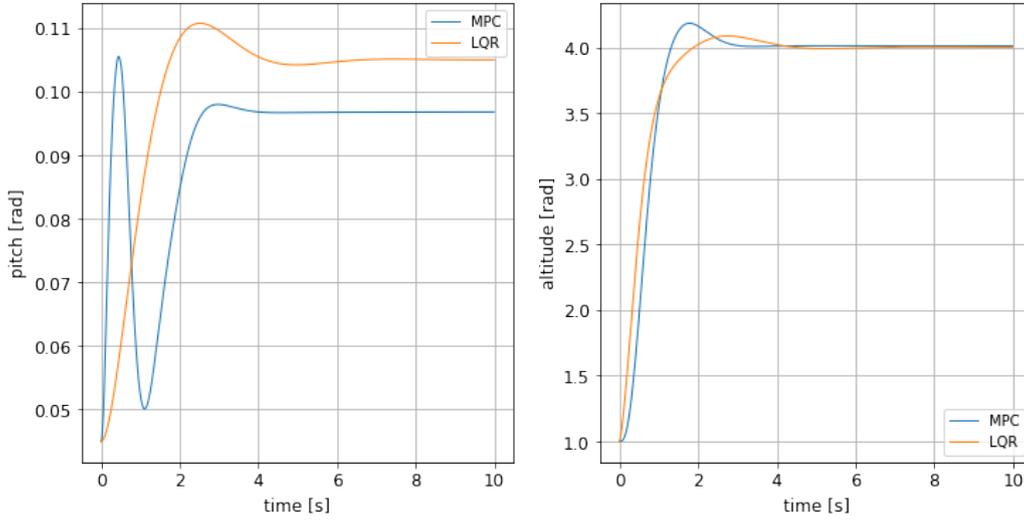


Figure 4.22: comparison between non-linear MPC and uLQR for altitude change

Another important factor is the behaviour of Euler angles, as shown in 4.6, where the *yaw* doesn't reach the set steady-state value of 0. This is probably caused by the use of the complex expressions of the control surfaces (3.7) as the behaviour isn't observed when using the simplified *direct inputs*, as shown in 4.3.

Still, the comparison of the present section is valid, given that the *evaluation parameters* and the general behaviour are similar in the two cases.

Chapter 5

Conclusions and future work

A primary investigation in search for a ground truth to set the base of this long project has been done.

The process is not easy, given the lack of knowledge on ekranoplanes that constitutes a kind of their own, greatly increasing the challenges of the project that is being carried out. The most important aspects that have been uncovered during the present work are how different ekranoplans are compared to existing kinds of vehicle.

In the first chapter an overview on the project has been given by sourcing existing bibliography, assessing the problem of ekranoplan navigation and related topics. The vehicle of reference has been placed in one of the WIG craft categories and characterised.

In the second chapter the dynamic equations used for the model are shown and explained. The choice of the same model as aeroplanes has been justified by some findings in the bibliography, including the simplifications in use. The novelty is the employment of the altitude as a variable of primary concern.

In the third chapter the aerodynamic problem is tackled. Starting from the general form of the aerodynamic coefficients, the used custom CFD is described in its capabilities and limitations. Different simulations are run, to evaluate the program. When the results have shown a behaviour unconfirmed by the bibliography or illogic it has been left out of the equations or adjusted accordingly. This is the case of the influence of the *roll* and *yaw* angles, that do not contribute to the general *lift* or *drag* in the implemented formulae, because the results of their effects haven't shown any recognizable pattern. Therefore only the parameters upon which the forces have shown a strong dependency have been included.

The aerodynamic forces have been treated without being reduced to coefficient form in order to allow for a different handling, interpolating the data to include them in the simulator. This was done to have an analytical

function at disposal, that could simplify the next steps of the work when needed. The aerodynamic forces depend on the *pitch* angle, the *altitude* and the *velocity*. These parameters have clearly shown their influence and are part of the obtained expression. The magnitude of the forces needed to be adjusted according to the bibliography given the lack of turbulence in the CFD, that creates a magnitude problem. Once corrected, the results are representative of the general behaviour of the aerodynamic force and of the way they depend on their variables, which is the expected one.

This model lacks some dependencies compared to what would be required for a 6-DoF model as is being sought. For now it is used as it is, and one of the objectives for the future is to reach the level of *precision*, in the sense of being able to take into account all the configuration parameters, as found in the bibliography.

The aerodynamic force obtain so are used in chapter four to complete the dynamical equation and run closed loop simulations. The main candidate for a full control is the non-linear MPC controller, and its capabilities are shown in two different scenarios.

The most realistic case which uses the control surfaces expressions found in the bibliography, shows promising results as the objectives are reached and the performances are quite good. This controller is able to successfully recovery from a perturbed position and change altitude of the vehicle in a low time. Also, its characteristic of admitting constraints is very useful given the nature of the problem and allowed for a finer tune of the performances as needed.

Apart from this, the disadvantage consist in its inability to avoid oscillations in the *pitch angle*, which is mainly a comfort concern, but might also lead to unwanted effects on the dynamic.

The other type of evaluated controller is an LQR on top of a feedback linearised model. The linearisation leads to a linear and decoupled model that only contains the variables of interest. Amongst these there is the altitude, which usually isn't a direct concern in aeroplane control, but has to be included in this case since it is the main matter of ekranoplan control.

This leads to the impossibility of removing the input along the vertical body frame axis, given that without it the problem would be infeasible.

Apart from this, the LQR is noise resistant and smooth, leading to an overall acceptable dynamic in the analysed cases. Its disadvantage is that the used model is probably masking some of the dynamic. Therefore the results have to be taken with a grain of salt and further investigated for a complete view on the topic.

Its performances are slower than the MPC but do not present the oscillating behaviour seen before.

The future holds great advancements given the foundations laid during whole work carried out by the team during this period.

A more precise model is the priority. Using this the controllers can be tested using a more truthful model to definitively assess their capabilities and choose which one is to be developed for the real use, once a scaled functioning model will be built.

Bibliography

- [1] Auawisederivative work: Jrvz (talk) - CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=9441238>
- [2] Liesbeth Schrootena Ina De Vlieger, Luc Int Panis, Cosimo Chiffi, Enrico Pastori “Emissions of maritime transport: A European reference system” *Science of The Total Environment* Volume 408, Issue 2, 20 December 2009, Pages 318-323
- [3] Jean-Luc Boiffier *The Dynamics of Flight: the Equations*
- [4] Bernard Etkin, Lloyd Duff Reid “Dynamics of Flight Stability and Control” (3rd Edition)
- [5] <http://www.b737.org.uk/techspecsdetailed.html>
- [6] E. N. M. Cirillo *Appunti delle lezioni di meccanica razionale per l'ingegneria*. Compomat, Configni (RI), 2015
- [7] Rozhdestvensky, Kirill. (2000). Aerodynamics of a Lifting System in Extreme Ground Effect. 10.1007/978-3-662-04240-3.
- [8] Di Ekranoplan_A-90_Orlyonok.jpg: Sergey Rodovnichenko from Moscow, Russiaderivative work: Flankער (talk) - Ekranoplan_A-90_Orlyonok.jpg, CC BY-SA 2.0 “<https://commons.wikimedia.org/w/index.php?curid=9922872>”
- [9] Rattapol Sakornsin, Chinnapat Thipyopas, Siripong Atipan “Experimental Investigation of the Ground Effect of WIG Craft — NEW1 Model”, Presented at the Innovation Aviation and Aerospace Industry—International Conference 2020 (IAAI 2020)
- [10] de Divitiis, Nicola. (2005). Performance and Stability of a Winged Vehicle in Ground Effect. *Journal of Aircraft*. Vol. 42. pp. 148-157.. 10.2514/1.4830.

-
- [11] Fossen, Thor & Fjellstad, Ola Erik. (1995). Nonlinear Modelling of Marine Vehicles in 6 Degrees of Freedom. *Mathematical Modelling of Systems*. 1. 10.1080/13873959508837004.
- [12] Michini, B. and J. How. "L1 Adaptive Control for Indoor Autonomous Vehicles: Design Process and Flight Testing." AIAA Guidance, Navigation, and Control Conference 10-13 August 2009, Chicago, Illinois, AIAA 2009-5754.
- [13] "Lectures on ship manoeuvrability" Nikolai Kornev, p.144-172
- [14] B. Kolewe, W. Drewelow, D. Dewitz, B. Lampe, MARSPEED - Modelling and Real Time Simulating the Motion of a Wing-in-Ground-Effect Vehicle, IFAC Proceedings Volumes, Volume 43, Issue 20, 2010, Pages 184-189, ISSN 1474-6670, ISBN 9783902661883
- [15] Liu, Xinyu. (2018) 10.13140/RG.2.2.30706.56009.
- [16] Kimon P. Valavanis, George J. Vachtsevanos "Handbook of Unmanned Aerial Vehicles"
- [17] Satja Lumbar, Gregor Dolanc, Darko Vrečko, Stanko Strmčnik, Drago Matko "Automatic guidance of an aircraft using predictive control in a visual servoing scheme", IFAC Proceedings Volumes, Volume 43, Issue 15, 2010, Pages 1-6
- [18] Liang Yun, Alan Bliault, Johnny Doo *WIG Craft and Ekranoplan*
- [19] <https://www.ikarus342000.com/VOLGA2page1.htm>
- [20] Sir Horace Lamb *Hydrodynamics* "On The Motion of a Solid Through a Liquid" p. 160-201
- [21] C. Wieselsberger "Wing Resistance near the Ground" NACA TM No. 77, 1922
- [22] Abdul Ghafoor "Wing in ground effect vehicle: modelling and control" 2015, A thesis submitted to the graduate school of natural and applied sciences of middle east technical university
- [23] M. Mobassher Tofa, Adi Maimun, Yasser M. Ahmed, Saeed Jamei, Agoes Priyanto, and Rahimuddin "Experimental Investigation of a Wing-in-Ground Effect Craft"

- [24] Trabajo fin de master “Diseno, modelado y control de un vehiculo autonomo tipo Wing in Ground” Alberto Calvo Cordoba, Universidad Politecnica De Madrid, 2019
- [25] “Lectures on ship manoeuvrability” Nikolai Kornev, p.144-172
- [26] <https://www.wigetworks.com/>, Wiget Works website
- [27] Saeid Jafari, Petros Ioannou, Lael E. Rudd “What is L1 Adaptive Control” AIAA Guidance, Navigation, and Control (GNC) Conference
- [28] [https://www.lemos.uni-rostock.de/lehre/sonstiges/downloads/autowing/Autowing website](https://www.lemos.uni-rostock.de/lehre/sonstiges/downloads/autowing/Autowing_website)
- [29] Mahdis Bisheban and Taeyoung Lee “Geometric Adaptive Control With Neural Networks for a Quadrotor in Wind Field”, , Member, IEEE
- [30] <http://www.pdas.com/datcom.html>, usaf DATCOM website
- [31] Peter J. Baddoo¹, Melike Kurt, Lorna J. Ayton¹, Keith W. Moored, “Exact solutions for ground effect”, Journal of Fluid Mechanics · May 2020
- [32] A. Panferov, A. Nebylov, S. Brodsky “Features of designing control systems for WIG-craft” 2019
- [33] “<https://www.do-mpc.com/en/latest/index.html>”, do-mpc python library website
- [34] “<https://python-control.readthedocs.io/en/0.8.3/generated/control.lqr.html>” lqr control python library
- [35] By Chris Dembia and Tom Uchida - Own work, CC BY-SA 3.0, “<https://commons.wikimedia.org/w/index.php?curid=33180739>” https://en.wikipedia.org/wiki/Feedback_linearization
- [36] By Yaw_Axis.svg: Auawisederivative work: Jrvz (talk) - Yaw_Axis.svg, CC BY-SA 3.0, “<https://commons.wikimedia.org/w/index.php?curid=9441238>”
- [37] Lars Grüne, Jürgen Pannek “Nonlinear Model Predictive Control Theory and Algorithms”

- [38] Utku Eren, Anna Prach, Başaran Bahadır Koçer, Saša V. Raković, Erdal Kayacan, Behçet Açıkmese “Model Predictive Control in Aerospace Systems: Current State and Opportunities”; *journal of guidance, control, and dynamics*
- [39] Alberto Isidori “Non linear Control Systems ” Third Edition
- [40] Mohamad Abu Zarim, Adi Maimun, Mohd Rashdan Saad “Wing in Ground Effect Craft: A Review of the State Of Current Stability Knowledge”, conference paper, November 2016
- [41] Thomas Stastny, Roland Siegwart “Nonlinear Model Predictive Guidance for Fixed-wing UAVs Using Identified Control Augmented Dynamics”
- [42] <https://www.ansys.com/>
- [43] M. Serez, N. Abramov, M. G. Goman “Impact of Ground Effect on Airplane Lateral Directional Stability during Take-Off and Landing” *Open Journal of Fluid Dynamics*, 2018, 8, 1-14
- [44] Garriga, Jorge and Soroush, Masoud. (2010). Model Predictive Control Tuning Methods: A Review. *Industrial and Engineering Chemistry Research*. 49. 3505-3515. 10.1021/ie900323c.