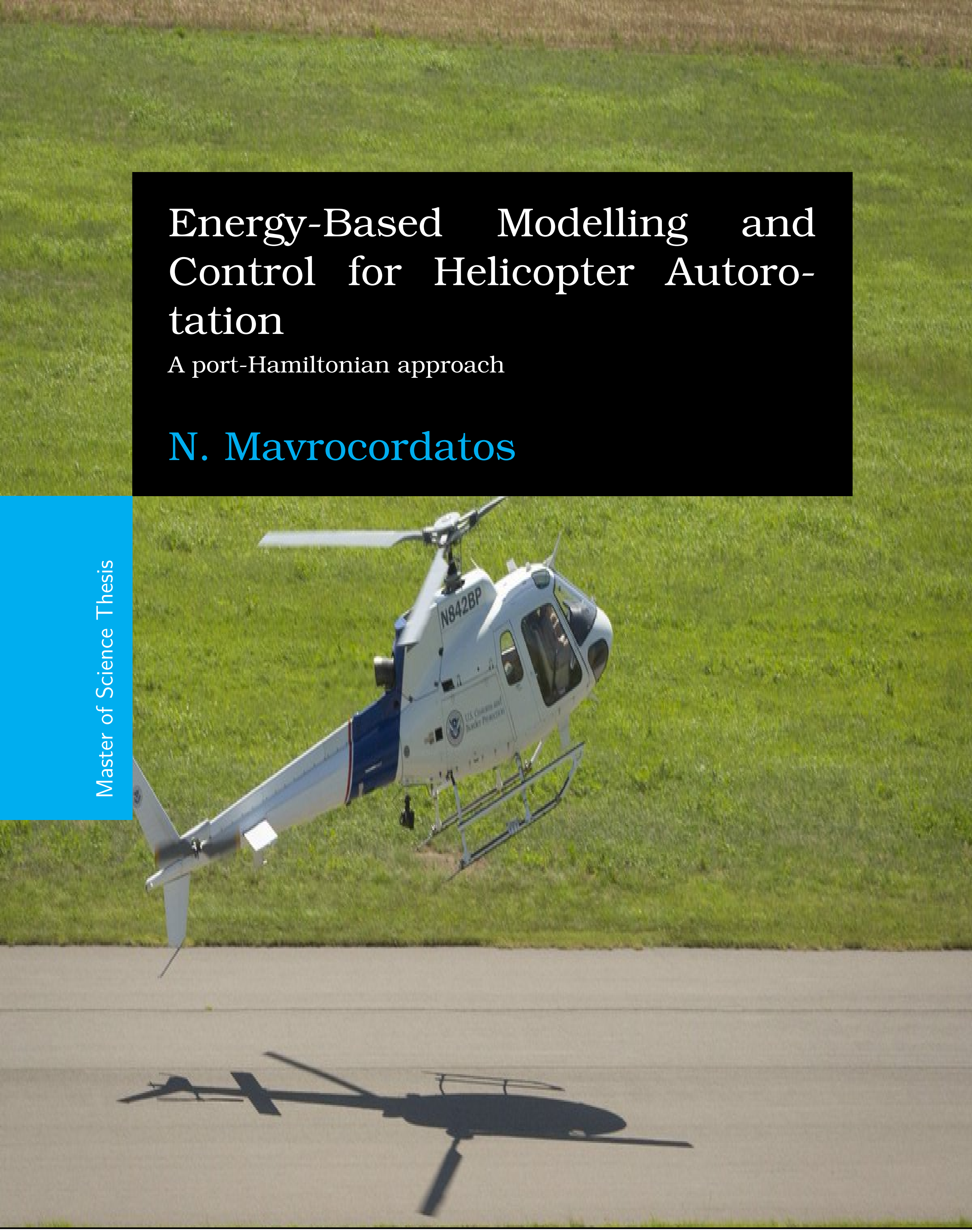
Energy-Based Modelling and Control for Helicopter Autorotation

A port-Hamiltonian approach

N. Mavrocordatos

Master of Science Thesis



TU Delft
Delft University of Technology

Delft Center for Systems and Control

Energy-Based Modelling and Control for Helicopter Autorotation

A port-Hamiltonian approach

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft University of Technology

N. Mavrocordatos

April 6, 2020

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

Automation is playing an increasingly important role in today's society and is revolutionising nearly all industries. In this thesis, the helicopter industry is considered. Helicopters are complex systems capable of Vertical Take-Off and Landing (VTOL). Many companies and organisations are now looking to automate VTOL vehicles in order to break through the urban air mobility market. In order to transport individuals, strict safety regulations have to be adhered to. The goal of this thesis is therefore to address the event of engine failure of a helicopter.

In the case of engine failure, a helicopter enters a manoeuvre, called autorotation, in order to safely land. To tackle this problem, a port-Hamiltonian (PH) approach is taken. Within the PH framework, energy transfers between system nodes are defined. By controlling the flow of energy between these nodes, control of the vehicle is possible. This is done by shaping the energy of the system through the use of Passivity-Based Control (PBC).

Prior work on the topic of helicopter control using PBC has only been carried out on point-mass models. In order to understand why that is the case, this thesis attempts to derive the PH formulation directly from the Newtonian flight mechanics. It was found that doing so is not possible due to the presence of non-passive terms. These originate from the Blade Element Theory (BET), which is necessary to calculate the thrust and torque of the main rotor.

A novel analytical tuning method has also been developed in order to minimise the rise-time and oscillations of the time-response for a PBC. This tuning method has been implemented on a 3-Degrees-of-Freedom (DoF) point-mass model. However, the degree of underactuation of the 6-DoF model has proven to be a limitation for this tuning method. Instead, the 6-DoF was tuned manually. The two models were verified against flight data of the Bo105 helicopter in powered flight.

Finally, the first step of the autorotation manoeuvre has been performed. Further work is required to complete the manoeuvre.

Table of Contents

List of Figures

Acknowledgements

I would like to thank my supervisors Giulia and Marilena for their assistance and guidance throughout this journey.

Marilena, you have taught me a tremendous amount about helicopters. You managed to answer all of my questions and fit me in your busy schedule. I would come out of every meeting knowing more about these incredible vehicles.

Giulia, you always found the right words to encourage me and taught me to take a step back when I would get stuck. This has helped me overcome many of the obstacles I have encountered.

To my family, thank you for bearing with me the past year while I worked on this thesis, and also for the other 22 years.

Dirk, David and Hugo, our times together have been some of the most memorable ones of my student life. Gervase, Diego and Bart, I will miss working and programming with you until late at night. To the rest of my friends that I cannot mention, you have made my time in Delft worthwhile.

Josephine, thank you for being so patient with me while I worked on this thesis and for all the support you have given me.

Finally, a special thank you is required for the food delivery services that have helped me complete this thesis.

Delft, University of Technology
April 6, 2020

N. Mavrocordatos

Chapter 1

Introduction

With the urban air mobility industry booming, robust controllers are essential. These controllers must be able to adapt to different flight conditions and events such as engine failure. That is why, in this thesis, the autorotation manoeuvre following engine failure of a helicopter is considered.

Combining port-Hamiltonian (PH) methods with Passivity-Based Control (PBC) is known to guarantee desirable robustness properties. Additionally, their ability to model complex non-linear systems and their intuitive tuning nature appears to render the method ideal for helicopters in autorotation.

Previous work on modelling helicopters in the PH framework have all used point-mass systems. Until now, it has been unclear why this is the case. In this thesis, the port-Hamiltonian formulation of a helicopter is derived directly from Newtonian flight mechanics. Non-passive terms have been identified, thereby preventing the use of PBC.

The autorotation manoeuvre depends heavily on the exchange of kinetic energy of the spinning rotor to kinetic and potential energy of the vehicle. The port-based method allows to control the flow of energy between nodes. The ability to utilise this method to safely land a helicopter following autorotation is therefore investigated within this thesis.

Additionally, experts in the field of PH control praise the framework for its intuitive and energy-based nature [4]. However, it was found that engineers new to the method may struggle to grasp this intuitivity. An attempt to resolve this issue is made by proposing a novel and intuitive tuning method by shaping the energy of the system at hand as if it were a mass-spring-damper system.

In short, the goals of this thesis are to:

1. Create a PBC to safely land a helicopter following engine failure.
2. Determine why PH systems of helicopters are not derived directly from flight mechanics.
3. Facilitate the application of PBC to PH systems to non-experts on the the topic.

1-1 Structure of the report

To answer the research questions of this thesis, the report is split into 3 parts.

Part 1: The first part of this thesis purely discusses helicopters and is split into two chapters. Chapter 2 discusses the need for the research along with an introduction of the work that has already been carried out on the subject. Section 2-6 concludes the chapter by presenting the formal research questions this thesis aims to answer. Chapter 3 covers the essential helicopter mechanics necessary for the reader to understand helicopter models.

Part 2: The second part covers the topic of passivity-based control. Port-Hamiltonian systems are first introduced and described in Chapter 4. This is then followed by the control of these systems through passivity-based control in Chapter 5. Part 2 is concluded with the proposal of an analytical tuning method for a PBC controller in Chapter 6.

Part 3: The final part combines all the information from parts 1 and 2 together to develop port-Hamiltonian models of helicopters. In Chapter 7, the issues arising from the conversion of Newtonian helicopter mechanics to the port-Hamiltonian framework are found and their origin is determined. The port-Hamiltonian methods are then applied to new 3-Degrees-of-Freedom (DoF) and 6-DoF point-mass models in Chapters 8 and 9 respectively. The simulation results are presented in Chapter 10, where an attempt at the autorotation manoeuvre is also covered.

The report is finally wrapped up with a conclusion of the research along with recommendations for future work in Chapter 11.

Part I

Helicopters

Helicopters are complex nonlinear systems. Rapid developments in technology are allowing control engineers to tackle the problem of automating these vehicles. Chapter 2 discusses the recent developments in the field of autonomous helicopters. Furthermore, the issue of safety and autorotation is addressed giving rise to the research questions of this thesis. Finally, Chapter 3 covers the necessary theory about helicopters to develop models and ultimately automate the rotorcraft.

Chapter 2

Autonomous helicopters

Every year, automation plays an increasingly important role in our day-to-day lives. The significance of Unmanned Aerial Vehicles (UAVs) and their role in today's society is outlined in Section 2-1. The safety of the payload and civilians on the ground is paramount. A factor that should therefore be addressed is the event of engine failure. Consequently, the autorotation manoeuvre is explained in Section 2-2. Robustness requirements for the aeronautics industry are briefly discussed in Section 2-3. Certain methods for safe, autonomous autorotation manoeuvres already exist, these are identified and evaluated in 2-4. An overview of available control methods is provided in Section 2-4. Finally, the chapter is concluded by highlighting the importance of port-Hamiltonian control in Section 2-5 .

2-1 Growth of UAVs in recent years

Up until recently, UAVs were most predominently used for military purposes. UAVs are now becoming increasingly available to the general public for civilian applications [5]. With advances in technology, autonomous vehicles are beginning to play an important role in our lives. F. Mohammed et al. [5] discussed the opportunities and challenges of utilising UAVs for smart cities. Some of the benefits in drone usage could be seen in geo-spatial activities, traffic and crowd management, agriculture management, natural disaster control and monitoring, etc. Many of these activities can be performed by quadcopters and the main challenges come within the area of data processing and trajectory planning due to their size rather than stable, autonomous flight itself. Additionally, their size often limits them from flying in harsh weather conditions.

The company SkyRyse based in Silicon Valley and founded in 2016 [6] has been developing autonomous helicopters in an to attempt to solve those issues. Cade Metz from the New York Times states that SkyRyse has been testing one of their helicopters to respond to 911 calls through the air and is also attempting to make their control methods applicable to civilian use [7].

Mark Wade, a 22 year coast guard helicopter pilot, is a strong advocate towards the automation of helicopters due to the high workload required by the crew [8]. A high percentage of incidents requiring search and rescue operations occur in stormy weather. This requires a high degree of expertise from the pilot and his/her crew preventing many rescue missions from being carried out due to safety reasons. Using automated vehicles could help limit the risks a crew faces during these missions.

Furthermore, companies such as Uber and Boeing have grown great interest in autonomous flight in recent years. Uber has started a new daughter company called Uber Elevate that is attempting to design electric Vertical Take-Off and Landing (VTOL) devices to facilitate urban air mobility [9].

Boeing is taking a different approach by creating the GoFly prize [10], whose motto is *"Together, we will make people fly"*. The goal is to accelerate the development of private, autonomous, flying vehicles. Furthermore, Boeing purchased Aurora, a company developing autonomous flying vehicles, and is working/competing with companies such as Lilium, Volocopter and Uber Elevate to define the future of autonomous flight. It is without a doubt that autonomous flying vehicles will play an important role in tomorrow's society.

Many benefits of autonomous flight have been mentioned above. However, flying autonomously requires well defined, robust controllers. These controllers must keep the vehicle safe in case of external disturbances as well as possible sensor or actuator failure(s).

2-2 Autorotation of a helicopter

The tremendous growth in technology and public interest in autonomous flight is creating a high level of demand from consumers. However, safety remains a critical factor towards the development of this technology and its availability to the public. A great safety issue comes with the risk of engine failure. Just like airplanes, helicopters are able to land safely following an engine failure with the benefit that they do not require a long runway. This is possible with the so-called autorotation manoeuvre illustrated in Figure 2-1. Would the reader happen to be unfamiliar with the terms collective and cyclic, Section 3-1 provides an explanation of these pilot inputs.

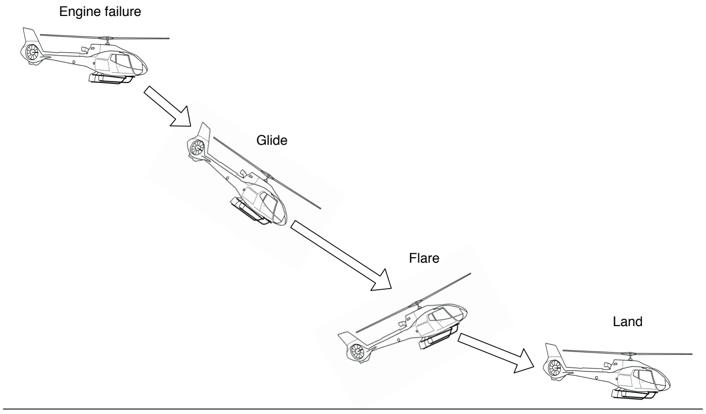


Figure 2-1: Illustration of a helicopter performing an autorotation landing.

Within a couple of seconds of engine failure, the engine completely stops providing torque to the blades. To overcome the drag of the spinning blades and maintain sufficient rotor angular velocity, the airflow through the blades is used. A transfer between the vehicle's potential energy and rotor kinetic energy is made. This transfer of energy is crucial to performing the autorotation steps. These are as follows [11]:

1. **Entry:** Lower collective to lowest position in order to maintain main rotor angular velocity. The right pedal controlling the tail rotor must also be completely pressed as the main engine is no longer providing any torque to counteract. To prevent nose pitch down, the longitudinal cyclic is pulled back.
2. **Glide:** The glide phase is also referred to as the stable autorotation. It describes the phase where the pilot attempts to keep the forward velocity and and descent rate of the rotorcraft constant whilst maintaining the rotor Revolutions Per Minute (RPM) within its operating range by trading potential energy (altitude) into kinetic energy (rotor RPM).
3. **Flare:** The flare manoeuvre is also common for aircraft landing. To perform a flare, the nose of the vehicle is pitched up to attempt to lose as much kinetic energy as possible before touch-down. For helicopters, the kinetic energy of the rotor is used to minimise the descent rate and forward velocity of the vehicle. This allows the rotorcraft to land smoothly.
4. **Landing:** Before touching down, the helicopter has to exit the flare manoeuvre and level itself by use of the cyclic. Once levelled, the collective is used to soften the landing.

It must be noted that step 1 considers main engine failure with no failure to the tail rotor. In the event that the tail rotor fails, the main engine should be switched off to enter autorotation. This should be done due to the fact that when powered, the main rotor creates a torque which has to be countered by the tail rotor.

2-3 Robustness requirements

To implement a control system into an aircraft and have it certified as airworthy, the control system must first be approved by the Federal Aviation Administration (FAA). The FAA has strict software regulations described in part 25 of its Code of Federal Regulations (CFR). Part 25.671 states that the control system shall take into account all types of possible failures such as sensor failure, actuator failure or large disturbances. Furthermore, the aircraft shall remain controllable in the case of engine failure [12]. Any software implemented in an airworthy vehicle must also hold DO-178C certification [13].

2-4 Current methods available for control

Developing controllers for helicopters in autorotation is of utmost importance for the safety of the payload, leading to great research efforts in this area. This section covers some of the studies carried out and controllers developed for autonomous autorotation.

An early autorotation model was derived by W. Johnson in 1977 [14]. Johnson developed a nonlinear, longitudinal and vertical model for a helicopter in autorotation. He defined an optimal problem to minimise the impact velocity of the helicopter. However, it was not until 2004 that Hazawa et al. [15] tackled the control problem [16]. Since then, significant progress has been made in the control of helicopters following engine failure.

In 2010, Dalamagkidis et al. [16] developed a Nonlinear Model Predictive Controller (NMPC) enhanced by a Reccurent Neural Network (RNN) for a helicopter in *pure vertical* autorotation. The combination of NMPC and RNN improves the convergence performance of the controller. The authors suggest that future work should include fault tolerance and study the robustness of this method in more detail so as to implement the controller commercially. However, Cortellessa et al. discuss the certification of automated control systems for flight in [13] and state that RNNs currently present many difficulties as the possible tests to carry out on such systems are limited. Instead, concrete theoretical proofs of robustness with further physical evidence are essential for certification.

The University of Stanford has perhaps carried out the most significant strides towards autonomous autorotation. In fact, P. Abbeel et al. [17] performed the first autonomous autorotation on a real (remote controlled) helicopter. A Differential Dynamic Programming (DDP) controller was implemented. A DDP controller is an extension of an Linear Quadratic Regulator (LQR) controller for nonlinear systems - making DDP a form of optimal control. This type of controller shows promising results and can also offer robustness properties. Further research could be carried out to implement this type of controller onto larger vehicles.

2-5 Why port-Hamiltonian?

The focus of this thesis is on the use of port-Hamiltonian methods to automate the autorotation of a helicopter. To this date, no successful autorotation control method using the port-Hamiltonian framework has been found in the literature.

Limited information can be found on port-Hamiltonian-based control of helicopters in powered flight. P.J. Kruger [18] developed the first known 6-Degrees-of-Freedom (DoF) helicopter model in the port-Hamiltonian framework. His port-Hamiltonian model is a simple fully actuated point-mass system to which helicopter dynamics are added later on.

A clear gap in research can be identified. It is interesting to understand why point-mass models have been used thus far and whether this can be overcome and whether directly implementing a helicopter model within the port-Hamiltonian (PH) framework can be done.

Port-Hamiltonian control is a type of passivity-based control method that focuses on energy principles. Romeo Ortega and Arjan van der Schaft have co-authored many articles on port-Hamiltonian control and are two of the world's leading experts in the field. Ortega and van der Schaft strongly state that modern control relies on signal processing and solves the control problem by increasing gains as the effect of disturbances on nonlinear systems is often unclear making control inefficient. This premise has lead them to publishing an article on the importance of using energy methods in their article *Putting energy back in control* [19].

In this article, R. Ortega et al. discuss the many benefits of energy-based modelling - more particularly, the advantages port-Hamiltonian systems present. Their ability to model complex nonlinear systems and the intuition behind the design of the controller are some of the strong points mentioned. Furthermore, port-Hamiltonian control also shows robustness properties, making it particularly interesting with regards to the certification of these controllers for the aerospace industry.

The benefits presented by PH methods appear to be ideal to tackle the autorotation problem. The gap in research should therefore be addressed.

2-6 Research questions

The objective of this thesis is to evaluate the performance of PH Passivity-Based Control (PBC) for helicopters in autorotation. As a result, the proposed research question is as follows:

Can a passivity-based controller for a port-Hamiltonian model of a helicopter be developed to perform an autorotation?

To answer this question, models of helicopters in powered flight first have to be developed. Moreoever, to the best of our knowledge, only point-mass models of helicopters have been developed in the port-Hamiltonian framework. This thesis should therefore clarify whether it is possible to develop a model directly from the Newtonian flight mechanics. If it not feasible, does this affect the ability to perform an autorotation using a point-mass?

In order to land a helicopter safely, a quick and robust controller is required. For that reason, this thesis will also attempt to develop a method for tuning the PBC in an optimal way with the goal to minimise settling time and oscillations.

When answering the research questions, further sub-questions arise. The ones that will be addressed in this thesis are the following:

- Can a Newtonian body model of a helicopter be directly translated into the port-Hamiltonian framework?
- Can the performance of the PBC controller be optimised to carry out safe manoeuvres?

Chapter 3

Helicopter dynamics in the Newtonian framework

This chapter provides an overview of helicopter dynamics in the Newtonian framework. Having a thourough understanding of the flight mechanics in the Newtonian framework opens the door towards other frameworks such as port-Hamiltonian.

The pilot controls are first introduced in Section 3-1. Three different methods to compute the induced velocity are presented in Section 3-2. The dynamics of the helicopter depend on the flight condition. Whether it is descending or ascending, moving forward or hovering, the different airflows on the blades greatly affect the movement of the vehicle. These flight conditions are covered in Section 3-3. The important flapping phenomenon for both powered flight and autorotation is covered in Section 3-4. Furthermore, blade element theory is covered in Section 3-5 to understand the forces and moments acting on the blades.

Gathering all of the sections together, the equations of motion are derived. Section 3-6 covers the derivation of the equations of motion for both body and point-mass models. An explanation of how to convert the force inputs of the point-mass model to pilot inputs is then given in Section 3-7. Finally, an overview of the control methods providing all of the control inputs and system outputs can be found in Section 3-8.

3-1 Control of a helicopter

The control of a helicopter is possible thanks to a mechanical system called the swashplate. It is unclear who should get the credit for the invention of the system but it was patented in the Netherlands by Baumhauer in 1925, 14 years after it was allegedly proposed in Russia [20].

The swashplate is a device that receives pilot inputs and converts them into a pitch angle for the blades. To do so, the pilot has two control sticks: the collective and the cyclic. By changing the angle of the collective and cyclic sticks, the orientation and position of the swashplate will change, thereby changing the pitch of the blades. A clear illustration from Prouty [2] is given in Figure 3-1.

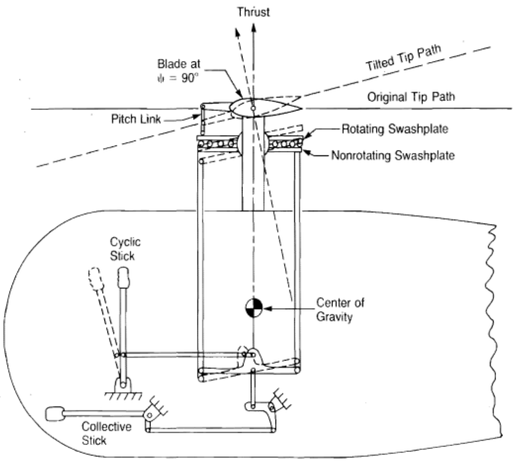


Figure 3-1: Illustration of how the collective and cyclic inputs change the pitch angle of the helicopter rotor blades; *Figure 3.27 [2].*

In short, the cyclic and collective sticks are used for two reasons:

1. *Collective:* control the magnitude of the rotor thrust vector.
2. *Cyclic:* control the direction of the rotor thrust vector.

The collective stick is typically located on the left side of the pilot. By pulling on the stick, the entire swashplate will be pushed upwards as seen in Figure 3-1. As a result, all blades receive an increase in pitch angle. Consequently, pulling on the collective will increase the thrust vector of the rotor by a constant amount in the entire disc plane. The collective input is reffered to as θ_0.

The disc plane is the plane that the blades are rotating in. Within this plane, the azimuth angle ψ_{az} is defined. The azimuth angle is used to describe the position of a blade within the plane. At $\psi_{az} = 0°$ the blade is parallel to the fuselage with its tip at the front of the aircraft as seen in Figure 3-2.

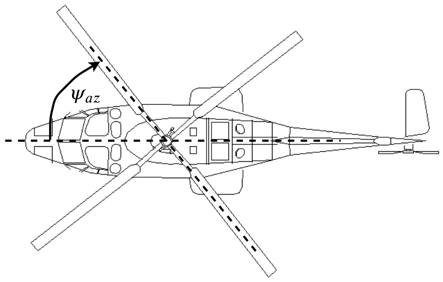


Figure 3-2: Top view of a helicopter illustrating the azimuth angle ψ_{az} in the disc plane.

By using the cyclic stick, the pilot is able to vary the pitch angle of the blades depending on their location within the disc plane ψ_{az}. By looking at Figure 3-1, it is seen that a forward push in the cyclic stick will tilt the swashplate forward. As a result, the pitch of the blades will be lowest when $\psi_{az} = 0°$ and highest when $\psi_{az} = 180°$. The pitch angle of a blade is therefore described as a function of ψ_{az}:

$$\theta(\psi_{az}) = \theta_0 - \theta_{long}\cos(\psi_{az}) - \theta_{lat}\sin(\psi_{az}) \tag{3-1}$$

3-2 Induced velocity

Induced velocity is a vertical velocity component of the air going into the rotor disc. As it is known, the angle between the air velocity vector and the chord of an airfoil is referred to as the angle of attack, α (see Section 3-3 for more details on the angle of attack). The angle of attack determines the lift, drag, and torque components of a blade. It is therefore important to have an accurate model of the induced velocity to in turn calculate the dynamics of the helicopter. Three methods to compute the induced velocity will now be introduced.

3-2-1 Momentum theory

Using momentum theory, Prouty derives an expression for the induced velocity, v_i in forward flight [2].

$$v_i = \sqrt{-\frac{V^2}{2} + \sqrt{\left(\frac{V}{2}\right)^2 + \left(\frac{T_m}{2\rho A}\right)^2}} \tag{3-2}$$

where V is the velocity vector of the helicopter, T_m is the thrust of the main rotor. In this case, T_m is taken in hover such that it is equivalent to the weight of the vehicle. The air density is described by ρ and A is the rotor disc area, with $A = \pi R^2$, R being the span of a blade or the radius of the main rotor.

This expression is also valid for hover, when both the forward velocity and climb rate are zero.

For practical purposes, the inflow velocity is often normalised, yielding the induced inflow ratio, λ_i, where $\lambda_i = v_i/\Omega R$ with Ω the main rotor angular velocity. The curve in Figure 3-3 shows the induced velocity of the Bo105 helicopter using Prouty's model.

The validity of this method is, however, limited to tight constraints. The momentum theory assumes levelled flight with a flight path angle γ of zero. In other words, this method is not valid when the climb rate is non-zero.

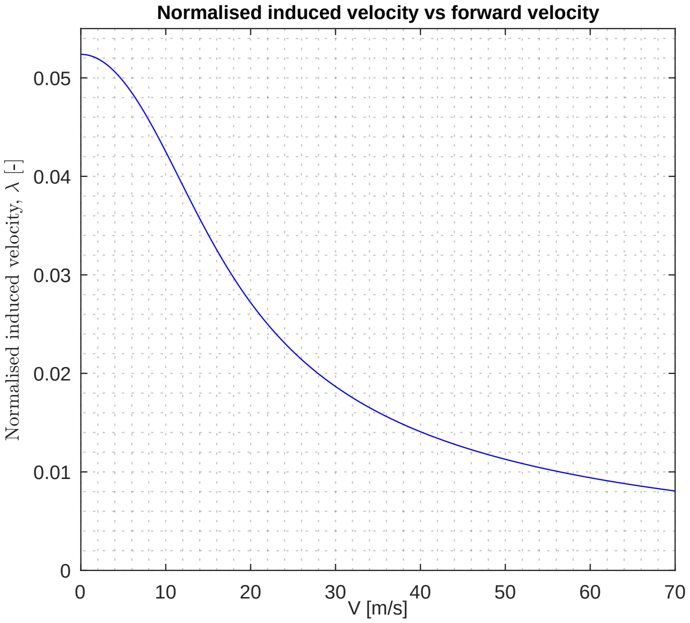


Figure 3-3: Induced velocity ratio with increasing forward velocity using [2] for the Bo105.

3-2-2 Haffner-diagram

To overcome the assumption of levelled flight, a method using a Haffner-diagram based on [1] may be used. In this new method, the flight path angle is allowed to be non-zero. This is particularly of interest when calculating the induced velocity in autorotation or in any climbing/descending flight.

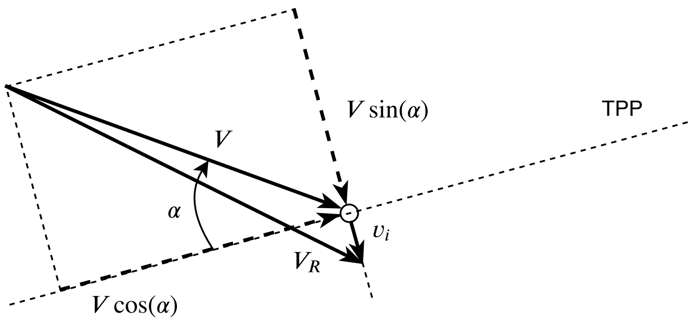


Figure 3-4: Illustration of the induced velocity in forward and climbing flight.

By taking the velocity components of V parallel and perpendicular to the Tip-Path-Plane (TPP) and performing vector addition with the induced velocity v_i, the resultant V_R from Figure 3-4 is calculated.

$$V_R^2 = (V\sin(\alpha) + v_i)^2 + V^2\cos^2(\alpha) \tag{3-3}$$

Furthermore, the thrust is assumed to be equivalent to the rate of change in momentum. This can directly be expressed as:

$$T = 2\rho\pi R^2 V_R v_i \tag{3-4}$$

By combining Equations (3-3) and (3-4) for $T = W$, a new equality is obtained. This is then non-dimensionalised - whereby an arbitrary variable x expressed as $\bar{x}$ is equivalent to $\frac{x}{v_{i_h}}$ with v_{i_h} the induced velocity in hover. As a result, the following expression for the normalised induced velocity is obtained:

$$\frac{1}{\bar{v}_i^2} = \left(\bar{V}\sin(\alpha) + \bar{v}_i\right)^2 + \bar{V}^2\cos^2(\alpha) \tag{3-5}$$

Instead of solving Equation (3-5) analytically, a so-called Haffner-diagram is used. In this diagram, the $x-$ and $y-$axes correspond to $\bar{V}\cos(\alpha)$ and $\bar{V}\sin(\alpha)$ respectively. Points are then obtained by plugging in values of $\bar{V}\cos(\alpha)$ with a specific $\bar{v}_i$, yielding the corresponding $\bar{V}\sin(\alpha)$. In a Haffner-diagram, contours can be observed. These correspond to lines of constant $\bar{v}_i$.

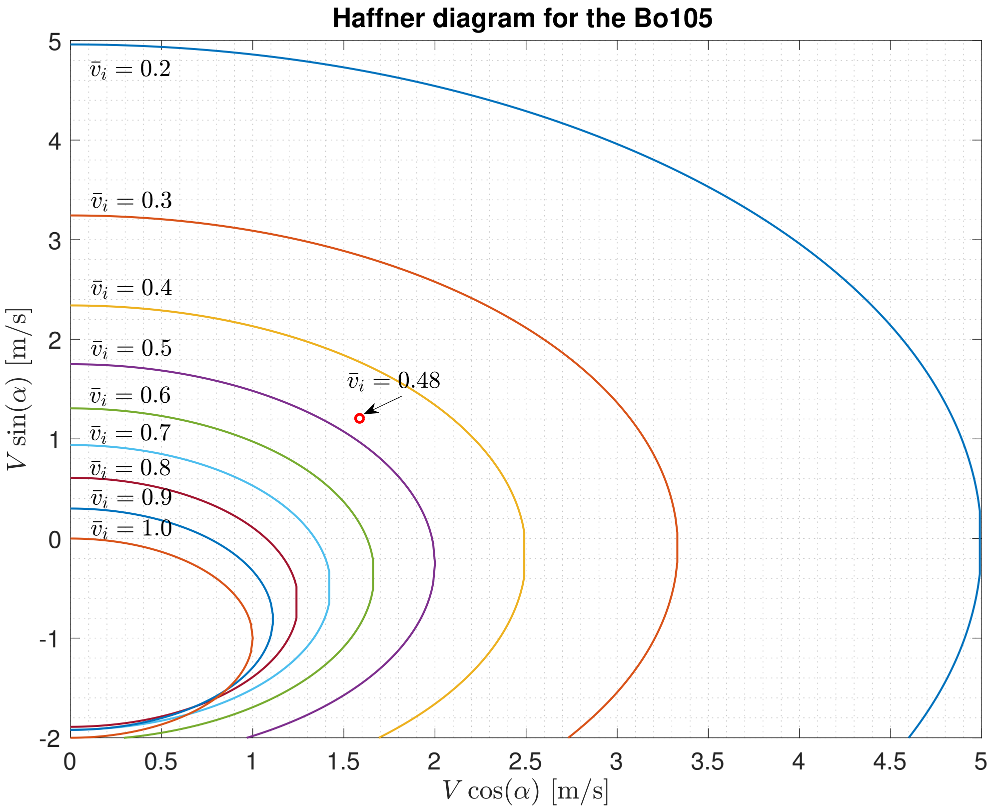


Figure 3-5: Haffner-plot for an estimated induced velocity in hover of 10.7 m/s.

Furthermore, the red circle in Figure 3-5 at coordinates (1.59, 1.20) is placed to illustrate how the Haffner-diagram is used to calculate the induced velocity.

The MATLAB implementation takes the velocity $V = \sqrt{u^2 + w^2}$ - u and w being the velocity components in the $x-$ and $z-$axes respectively - and the angle of attack α as inputs. The point $(V\cos(\alpha), V\sin(\alpha))$ will most certainly not lie on a pre-calculated $\bar{v}_i$ isoline. As a result, it is necessary to interpolate between two different $\bar{v}_i$ contours. For the point (1.59, 1.20), this yields an estimated $\bar{v}_i$ of 0.48.

The MATLAB script implementing the Haffner diagram can be found in Appendix A-1. It can be seen in Figure 3-5 that the script will encounter issues when $-2 \leq V\sin(\alpha) \leq 0$ and $0 \leq V\cos(\alpha) \leq 1$ due to intersections between lines. This means that the interpolation will not be possible and wrong values may be obtained. Furthermore, it is not possible to interpolate within the contour $\bar{v}_i = 1$. In those cases, a condition has been set to assume $\bar{v}_i = 1$.

3-2-3 Glauert theory

In his book [1], van Holten uses the Glauert hypothesis to derive the thrust coefficient of the helicopter blades. To do so, Glauert assumes momentum theory is applicable in forward flight. Then, the thrust can be expressed as:

$$T_m = 2mv_i \tag{3-6}$$

with T_m the thrust, m the mass of the helicopter and v_i the rotor induced velocity. Knowing that the thrust is expressed as shown in Equation (3-4), the thrust coefficient C_T can easily be derived. This method was developed by Glauert, the subscript $Glau$ is therefore added to C_T.

$$C_{T_{Glau}} = 2\lambda_i \sqrt{\left(\frac{V}{\Omega R}\sin(\alpha) + \lambda_i\right)^2 \left(\frac{V}{\Omega R}\cos(\alpha)\right)^2} \tag{3-7}$$

It will be seen in Section 3-5 that the expression for the thrust coefficient using Blade Element Theory (BET) is:

$$C_{T_{BEM}} = \frac{1}{4}c_{l_\alpha}\sigma\left[\frac{2}{3}\theta_0\left(1 + \frac{3}{2}\mu^2\right) - (\lambda_i + \lambda_c)\right] \tag{3-8}$$

with the subscript BET to denote that the derivation comes from BET.

Making the assumption that momentum theory is applicable in forward flight, $C_{T_{Glau}}$ and $C_{T_{BEM}}$ should be equivalent. This equivalence is achieved through the induced velocity term, λ_i. To allow the two coefficients to converge to the same value, the following expression for the rate of change in λ_i is derived.

$$\dot{\lambda}_i = \frac{C_{T_{BEM}} - C_{T_{Glau}}}{\tau} \tag{3-9}$$

where $\dot{\lambda}_i$ is the time derivative of the non-dimensional induced velocity and τ a time constant for dynamic inflow; τ is tipically around 0.1 seconds.

The non-dimensional induced velocity at the next point in time can then simply be calculated by:

$$\lambda_{i_{k+1}} = \lambda_{i_k} + \dot{\lambda}_{i_k}dt \tag{3-10}$$

with k indicating an instant in time and dt being the time difference between instant k and instant $k+1$.

Note: In this report, λ_i and λ_c will often be used. These two symbols are not to be confused with each other: λ_i is a result of the rotor spinning and moving air, while λ_c is the velocity going into the rotor as a result of the motion of the helicopter.

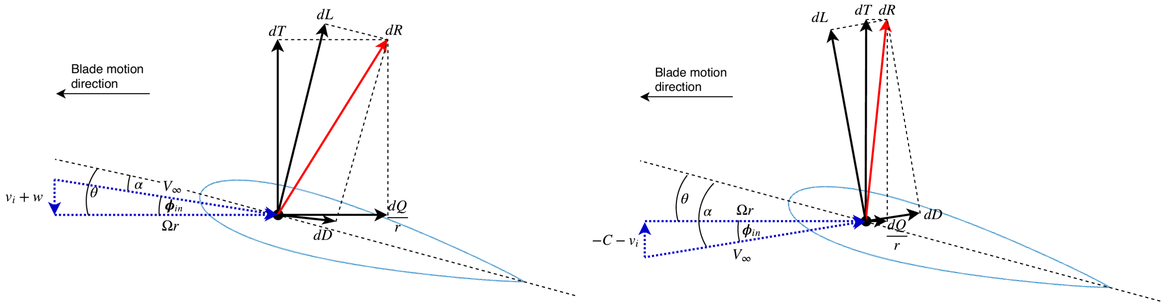
3-3 Flight conditions

During flight, a blade will see different flight conditions and generate varying combinations of forces.

Figures 3-6 and 3-7 show the cross-section of a blade taken at a distance r from the root and the forces acting on that element as well as the angles θ, α and ϕ_{in}, the pitch, angle of attack and inflow angle respectively. The important points to notice are how the resulting velocity vector affects the airfoil angles as well as the forces and their orientation.

The forces dL, dT and dR are the element lift, thrust and resultant forces respectively and depend directly on the angle between the resultant velocity, V_∞ and the blade pitch, θ ($\alpha = \theta - \phi_{in}$).

The first two cases to consider are climb and descent such as in Figures 3-6a and 3-6b. One might think that the vertical velocity vectors are reversed, however, the relative velocity between the blade and the air *as seen by the blade* has to be considered - not the motion of the helicopter itself.

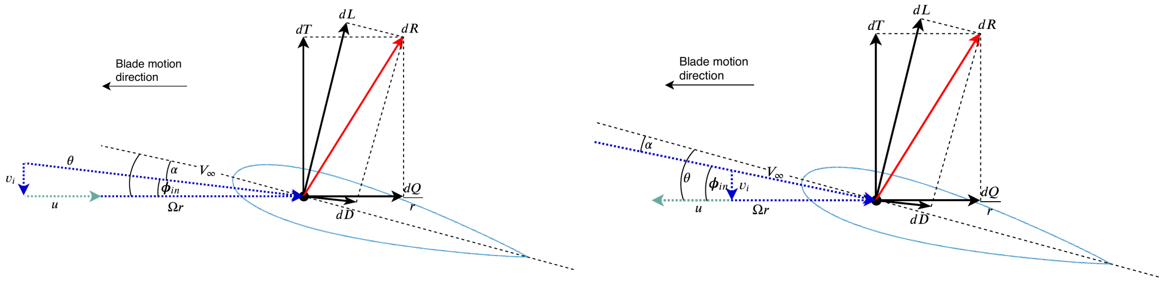


(a) Forces acting on a blade element at a span-wise distance r in climb.

(b) Forces acting on a blade element at a span-wise distance r in descent.

Figure 3-6: Forces acting on an airfoil during climb and descent.

The next cases to consider are the advancing and retreating blades in forward flight. That is, in forward flight, one blade is moving slower than the other when considering the relative velocity between the air and the blades. This is depicted by Figures 3-7a and 3-7b.



(a) Forces acting on an advancing blade element at a span-wise distance r in forward flight.

(b) Forces acting on a retreating blade element at a span-wise distance r in forward flight.

Figure 3-7: Forces acting on an airfoil during forward flight.

Combining Figures 3-6 and 3-7 together, the cases for climb/descent combined with forward flight can easily be derived.

3-4 Blade flapping

The term flapping dynamics refers to the aerodynamic phenomenon occurring when helicopter blades rotate and begin to flap. However, when modelling the motion of the body, the name *flapping dynamics* may be misleading. When deriving the equations of motion, flapping is considered as a static mode which results in a so-called rotor tilt. This flapping phenomenon plays an important role in the control of a helicopter as it directly affects the direction of the thrust vector of the main rotor.

This sections is composed of two parts. First, the equations for powered flight are considered. Then Section 3-4-2 covers the flapping dynamics in autorotation.

3-4-1 Flapping in powered flight

The flapping of the blades is modelled similarly to a mass-spring-damper system under forced oscillation. The forced oscillations appear as a result of three physical phenomena leading to the following differential equation [21]:

$$\ddot{\beta} + \frac{\gamma}{8}\Omega\dot{\beta} + \Omega^2\beta = \underbrace{-2q\Omega\sin(\psi_{az})}_{\text{Coriolis effect}} + \underbrace{\frac{\gamma}{8}\Omega\cos(\psi_{az})}_{\text{Oscillation due to pitch rate}} + \underbrace{\frac{\gamma}{8}\Omega^2\left(\theta - \frac{4\lambda_i}{3}\right)}_{\text{Oscillation due to lift}} \tag{3-11}$$

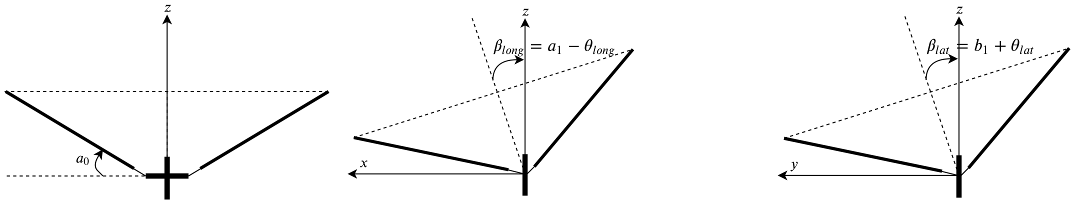
where γ is the lock number, q the pitch rate, Ω the rotor angular velocity, ψ_{az} the azimuth angle and θ the pitch angle as described by Equation (3-1). Equation (3-11) is then solved and its solution is approximated as a Fourier series expansion. The first 3 terms of the Fourier series expansion approximate β well enough as the higher frequencies are quickly damped. As a result, β is approximated as:

$$\beta = a_0 - a_1\cos(\psi_{az}) - b_1\sin(\psi_{az}) \tag{3-12}$$

where a_0 represents the coning angle, a_1 the longitudinal tilt and b_1 the lateral tilt due to the flapping motion. When considering the steady-state response of the flapping dynamics, the blades are seen to follow a specific path along what is refferred to as the disc plane, or, the TPP. The orientation and shape of the TPP is characterised by the three angles, a_0, a_1 and b_1 as well as by the control inputs θ_{long} and θ_{lat}. The three flapping angles are described as the following:

1. *Coning angle a_0:* the flap up motion. No tilt disc tilt is created as a result of the coning angle.
2. *Longitudinal flap a_1:* tilt angle of the rotor in the longitudinal direction as a result of flapping.
3. *Lateral flap b_1:* tilt angle of the rotor in the lateral direction as a result of flapping.

Combining the cyclic inputs with the flapping angles, the orientation of the TPP can finally be found as described in Figures 3-8b and 3-8c. A forward tilt in the longitudinal cyclic will see the TPP tilt forward and a sideways increase in lateral cyclic will tilt the TPP in the same direction [22]. Figure 3-8a illustrates the coning angle due to the a_0 flapping term.



(a) Coning angle: a_0. **(b)** Thrust longitudinal tilt: β_{long}. **(c)** Thrust lateral tilt: β_{lat}.

Figure 3-8: Illustrations of the main rotor thrust vector.

Since the flapping angles play such an important role on the tilt of the thrust, their expression must be found and was derived by M. Pavel in [21]:

$$a_0 = \frac{\gamma}{8}\left[\theta_0(1+\mu^2) - \frac{4}{3}(\lambda_i + \lambda_c)\right] \tag{3-13}$$

$$a_1 = \frac{-\frac{16}{\gamma}\frac{q}{\Omega} + \frac{8}{3}\mu\theta_0 - 2\mu(\lambda_c + \lambda_i)}{1 - 0.5\mu^2} \tag{3-14}$$

$$b_1 = \frac{-\frac{q}{\Omega} + \frac{4}{3}\mu a_0}{1 + 0.5\mu^2} \tag{3-15}$$

where q is the pitch rate, θ_0 the collective angle, γ the lock number, λ_i and λ_c the induced velocity and inflow ratios respectively. Ω is the rotor angular velocity and μ is the non-dimensional velocity component of the helicopter in forward flight.

Pavel states that the expression of the lateral flapping angle in Equation (3-15) does not yield a good estimation of b_1, due to the fact that the lateral tilt is greatly affected by the non-uniformity of the induced velocity [21].

To resolve this issue, an extra K' term is added with:

$$K' = \frac{1.33|\mu/(\lambda_c + \lambda_i)|}{1.2 + |\mu/(\lambda_c + \lambda_i)|} \tag{3-16}$$

This results in Equation (3-15) to become:

$$b_1 = \frac{-\frac{q}{\Omega} + \frac{4}{3}\mu a_0}{1 + 0.5\mu^2} + K' \tag{3-17}$$

3-4-2 Flapping in autorotation

F. Scaramuzzino et al. [23] derived the flapping equations for a helicopter in autorotation. The model of Scaramuzzino includes parameters such as the twist of the blade. These are ignored for simplicity and consistency with the equations provided in Section 3-4-1. Following the reduction of the equations using the same assumptions as in Section 3-4-1, the flapping equations are reduced to:

$$a_0 = \frac{\gamma}{8}\left[\theta_0(1+\mu^2) + \frac{4}{3}(\lambda_i + \lambda_c)\right] \tag{3-18}$$

$$a_1 = \frac{-\frac{16}{\gamma}\frac{q}{\Omega} + \frac{8}{3}\mu\theta_0 + 2\mu(\lambda_c + \lambda_i)}{1 - 0.5\mu^2} \tag{3-19}$$

$$b_1 = \frac{-\frac{q}{\Omega} + \frac{4}{3}\mu a_0}{1 + 0.5\mu^2} \tag{3-20}$$

Equations (3-18) to (3-20) and Equations (3-13) to (3-15) are very similar. Only a sign switch for the inflow velocity terms can be observed.

3-5 Blade Element Theory - BET

As it was mentioned previously, induced velocity causes the angle of attack of the blade to change. However, there is more to it. In fact, the local angle of attack of the blade along its span is not constant. One of the causes of this variation in angle of attack is the fact that the linear velocity at the root of the blade is slower than at the tip - as it is for all rotating objects. The effect of this change along the blade motivates the discretisation of the blades for modelling. This results in the so-called Blade Element Theory (BET).

BET is used to determine the local thrust and torque coefficients along the blade and in turn calculate the overall coefficients of the blade. The derivation of these equations may be overwhelming for a first-time reader. To better understand how BET works, a simplification from fixed-winged vehicles is first given.

To begin with, the thrust coefficient will be derived in Section 3-5-1. The coefficient is then derived in the cause of autorotation Section 3-5-2. Finally, the torque coefficient C_Q is provided for powered flight in Section 3-5-3 and the autorotation case is considered in Section 3-5-4.

3-5-1 Thrust coefficient in powered flight

A helicopter's thrust can be compared to an airplane's lift. This derivation therefore begins with the well known lift equation of a fixed-wing aircraft:

$$L = C_L \frac{1}{2}\rho V^2 S \tag{3-21}$$

where C_L stands for the 3D lift coefficient, ρ is the air density, V is the forward velocity of the aircraft and S is the total wing area. It should be noted that C_L, *with an emphasis on the capital letters*, is being used in Equation (3-21). This is due to the fact that C_L corresponds to the total lift coefficient of the aircraft. It will be seen later on that, in BET, an integration is carried out along the length of the wing, with the local lift coefficient of each cross-section written as c_l. Additionally, the lift coefficient C_L is proportional to the angle of attack, α. Assuming a symmetrical airfoil (direct proportionality between C_L and α), Equation (3-21) can then simply be re-written as:

$$L = C_{L_\alpha} \alpha \frac{1}{2} \rho V^2 S \tag{3-22}$$

where C_{L_α} is the lift gradient. From Section 3-3, it is known that the angle of attack varies at every cross-section (element) along the span of the wing. The lift can then also be calculated by integrating the lift generated by each element along the blade. In that case, the 2D lift coefficient, c_l, has to be used as shown by the following expressions.

$$L = 2 \int_0^{b/2} dc_l(r) dr \tag{3-23a}$$

$$dc_l = c_{l_\alpha} \alpha \frac{1}{2} \rho V^2 c dr \tag{3-23b}$$

where r depicts the position of the blade element, b the span of the wings and c the chord of the airfoil. Note that the chord is assumed to be constant so that the wings are not tapered. Furthermore, the wing is assumed to not have any twist.

Similarly, this method can be used for helicopters. The main difference is now that the angle of attack will be expressed differently due to flapping and the varying blade velocity along its span. Thankfully, this expression is standard to most helicopters and can often be taken directly from literature. In this section, the BET from [1] is used.

In the following equation, the thrust coefficient, C_T, is expressed for forward, symmetrical flight:

$$C_T = \frac{1}{4} c_{l_\alpha} \sigma \left(\frac{2}{3} \theta_0 \left(1 + \frac{3}{2} \mu^2 \right) - (\lambda_c + \lambda_i) \right) \tag{3-24}$$

where c_{l_α} is the 2D lift gradient of the airfoil ($c_{l_\alpha} \alpha = c_l$), θ_0 is the collective pitch (control input), μ is the advanced ratio or tip speed ratio, λ_c the non-dimensional inflow velocity in the control plane and λ_i the non-dimensional induced velocity. Finally, σ is the rotor solidity, expressed as:

$$\sigma = \frac{Nc}{\pi R^2} \tag{3-25}$$

where N represents the number of the blades, c the chord length of the airfoil and R the rotor radius.

The interested reader is referred to page 90 of [1] for the derivation of Equation (3-24).

3-5-2 Thrust coefficient in autorotation

To derive the expression of the thrust coefficient in autorotation, the powered flight method of Seddon [24] is adapted to the adjusted flight conditions. The interested reader is referred to page 55 of [24] for the original derivation.

When looking at a blade element, its contribution to the total coefficient of the blade is simply dC_T. By integrating over the length of the blade, the total coefficient is obtained. It can be shown that:

$$dC_T = \frac{1}{2} u_T^2 c_l dr \tag{3-26}$$

where u_T is the tangential velocity of the airflow relative to the airfoil, C_L the lift coefficient of the airfoil and dr the spanwise element of the airfoil.

The lift coefficient itself is assumed to vary linearly with the angle of attack of the airfoil:

$$c_l = c_{l_\alpha} \alpha \tag{3-27}$$

where c_{l_α} is the gradient of the 2D lift coefficient and α is the angle of attack. The main difference for the BET between powered and autorotative flight occurs in the definition of the angle of attack. In autorotation, the angle of attack is defined as:

$$\alpha = \theta + \phi_{in} = \theta + \frac{U_p}{U_T} \tag{3-28}$$

defining θ as the pitch angle of the blade and ϕ_{in} as the inflow angle. Furthermore, the inflow angle is approximated using the small angle approximation and is defined as the angle created as a result of the tangential and perpendicular airflows relative to the airfoil, U_T and U_P respectively. For simplicity, U_T and U_P are non-dimensionalised by dividing the terms by ΩR to give u_T and u_P, where Ω is the rotor angular velocity and R is the rotor disc radius.

The element-wise thrust coefficient is then redefined as:

$$dC_T = \frac{1}{2} u_T^2 (\theta u_T^2 + u_T u_p) dr \tag{3-29}$$

It is now time to define the non-dimensional velocity components u_P and u_T. The tangential velocity is simply the sum of linear velocities of the vehicle and of the blade due to its rotation.

$$u_T = \frac{r}{R} + \mu \cos(\psi_{az}) \tag{3-30}$$

where $\mu = V \cos\alpha/(\Omega R)$ and r is the spanwish location of the element along the blade.

The calculation of u_P is slightly more complicated. It consists of three elements. First, the induced velocity has to be taken into account. Second, the flapping angle β results in a velocity component perpendicular to the reference plane. Finally, the flapping velocity $\dot{\beta}$ or $d\beta/dt$ creates a vertical velocity component that has to be taken into account. Keeping u_P non-dimensional yields:

$$u_P = (\lambda_i + \lambda_c) + \beta\mu\cos(\psi_{az}) + \frac{r}{R}\frac{d\beta}{d\psi_{az}} \tag{3-31}$$

It may be noticed that the velocity component due to the flapping angular velocity is a function of $d\beta/d\psi_{az}$ instead of $d\beta/dt$. This is carried out through the following equivalence:

$$\frac{r}{\Omega R}\frac{d\beta}{dt} = \frac{r}{\Omega R}\frac{d\psi_{az}}{dt}\frac{d\beta}{d\psi_{az}} = \frac{\Omega r}{\Omega R}\frac{d\beta}{d\psi_{az}} = \frac{r}{R}\frac{d\beta}{d\psi_{az}} \tag{3-32}$$

From Equation (3-12) we know the flapping angle is expressed as:

$$\beta = a_0 - a_1 \cos(\psi_{az}) - b_1 \sin(\psi_{az})$$

Then:

$$\frac{d\beta}{d\psi_{az}} = a_1 \sin(\psi_{az}) - b_1 \cos(\psi_{az}) \tag{3-33}$$

Finally, Equation (3-31) becomes:

$$u_P = (\lambda_i + \lambda_c) + \beta\mu\cos(\psi_{az}) + \frac{r}{R}(a_1 \sin(\psi_{az}) - b_1 \cos(\psi_{az})) \tag{3-34}$$

The next step is to integrate dC_T from Equation (3-29) over the entire rotation of the blade ($\psi_{az} = 0$ to $\psi_{az} = 2\pi$) and along its entire length ($r = 0$ to $r = 1$). Some integral properties are therefore recalled to explain the integration process.

$$\begin{aligned}
&\int_0^{2\pi} \sin(\psi_{az}) d\psi_{az} = \int_0^{2\pi} \cos(\psi_{az}) d\psi_{az} = \int_0^{2\pi} \sin(\psi_{az})\cos(\psi_{az}) d\psi_{az} = 0 \\
&\int_0^{2\pi} \sin^2(\psi_{az}) d\psi_{az} = \int_0^{2\pi} \cos^2(\psi_{az}) d\psi_{az} = \pi
\end{aligned} \tag{3-35}$$

The integrals are now carried out to solve for C_T. It can easily be shown that:

$$\begin{aligned}
C_T &= \frac{1}{2}\sigma c_{l_\alpha} \int_0^1 \left(\frac{1}{2\pi} \int_0^{2\pi} \left(\theta_0 u_T^2 + u_P u_T \right) d\psi_{az} \right) dr \\
&= \frac{1}{2}\sigma c_{l_\alpha} \left(\frac{1}{3}\theta_0 \left(1 + \frac{3}{2}\mu^2 \right) + (\lambda_i + \lambda_c) \right)
\end{aligned} \tag{3-36}$$

3-5-3 Torque coefficient in powered flight

The torque coefficient is calculated similarly to the thrust coefficient. Considering a blade element at a distance r from the root, the torque the element generates is defined as dQ. To obtain the total torque, dQ is integrated over the span of the blade and the area of the disc such that:

$$Q = \frac{N}{2\pi} \int_0^{2\pi} \int_0^1 dQ = \rho A \Omega^2 R^3 C_Q \tag{3-37}$$

with N defined as the number of rotor blades, ρ the air density, Ω the rotor angular velocity, R the blade length and C_Q the torque coefficient.

The torque is generated as a result of both the drag and the lift of the blade. M. Pavel [25] therefore defines the element resultant force generating the torque as dD_1 with:

$$dD_1 = dD\cos(\phi_{in}) - dL\sin(\phi_{in}) \approx dD - dL\phi_{in} \tag{3-38}$$

dL is defined as the element lift and ϕ_{in} is the inflow angle as defined in Equation (3-28). Carrying out the integration from Equation (3-37), the torque coefficient can be obtained. M. Pavel has shown that the torque coefficient is then expressed as:

$$\begin{aligned} C_Q =& \frac{\sigma}{8}(1+\mu^2)C_d - \frac{\sigma C_{l_\alpha}}{2}\left[\frac{1}{8}(a_1^2+b_1^2) + \cdots \right. \\ & \left. \frac{\mu^2}{4}\left(a_0^2 + \frac{3}{4}a_1^2 + \frac{1}{4}b_1^2 - \mu\frac{a_0 b_1}{3} - (\lambda_i+\lambda_c)\left(\mu\frac{a_1}{2} + \frac{1}{3}\theta_0 + \frac{1}{2}(\lambda_i+\lambda_c)\right)\right)\right] \end{aligned} \tag{3-39}$$

where σ represents the blade solidity. The blade has drag coefficient C_d and lift gradient C_{l_α}. The advance ratio is defined by μ whereas the induced velocities are defined as λ_i and λ_c.

3-5-4 Torque coefficient in autorotation

The problem of finding the torque coefficient in autorotation has been tackled by F. Scaramuzzino et al. [23]. Scaramuzzino goes through a thorough derivation of the coefficient. The interested reader is therefore referred to the technical report. After simplification, the torque coefficient becomes:

$$\begin{aligned} C_Q = \left\{ \frac{C_d}{4C_{l_\alpha}}\left(1+\mu^2\right) - \theta_0\frac{\lambda}{3} + \theta_{lat}\left[-\frac{1}{8}a_1 + \frac{\mu\lambda}{4} + \frac{a_1}{16}\mu^2\right] + \right. \\ -\frac{1}{2}\left[\lambda^2 + \lambda\mu a_1 + \mu^2\left(\frac{1}{2}a_0^2 + \frac{3}{8}a_1^2 + \frac{1}{8}b_1^2\right)\right] + \\ \left. +\frac{\mu}{3}a_0 b_1 - \frac{1}{8}\left(b_1^2 + a_1^2\right)\right\} \end{aligned} \tag{3-40}$$

where σ is the blade solidity, μ the advance ratio, C_d the blade drag coefficient, and C_{l_α} the lift gradient; a_0, a_1 and b_1 are the flapping angles as described in Section 3-4. Finally, λ is the sum of the inflow velocities λ_i and λ_c such that $\lambda = \lambda_i + \lambda_c$.

3-6 Equations of motion

This section covers the derivation of the equations of motion of various helicopter systems. First, a 3-Degrees-of-Freedom (DoF) body system is derived in Section 3-6-1 and is followed by a 6-DoF body system in Section 3-6-2. Finally, a 6-DoF point-mass model is derived in Section 3-6-3.

For clarity, all the equations of motion will be derived in a similar way. All forces and torques will be split up into different parts such as forces driven by the rotor and dissipation forces. All forces are then summed up together yielding the system's equations of motion.

3-6-1 3-degrees-of-freedom body model

To begin with, a 3-DoF model is considered. For this system, a body reference frame is used. This reference frame is set at the center of gravity (cog) of the helicopter with its x-axis pointing directly at the nose of the helicopter and the z-axis pointing downwards, perpedicular to the x-axis. This is illustrated in the free-body diagram from Figure 3-9. An Earth reference frame with X_{Earth} and Z_{Earth} parallel and perpendicular to the Earth respectively is also included for clarity.

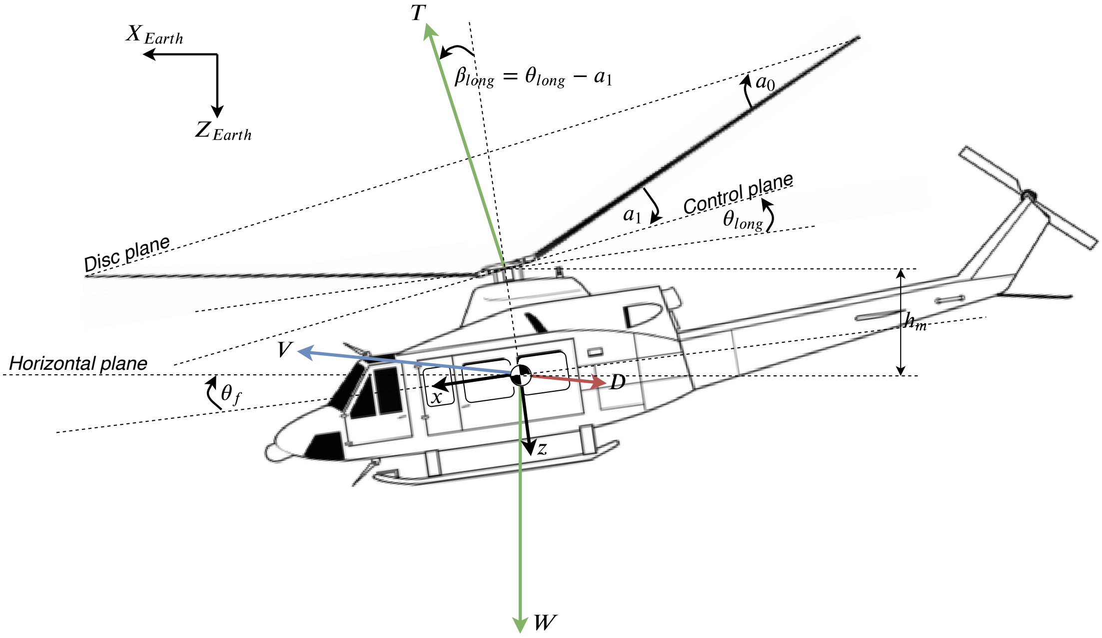


Figure 3-9: Illustration of the thrust tilt angle as a function of β_{long} and β_{lat}.

The helicopter in Figure 3-9 is flying following a path. The path of the rotorcraft is drawn by the velocity vector V. Parallel and opposite to that velocity, a drag force D can be identified. This force is acting at the cog of the helicopter. Another force can be seen acting on the center of gravity, and as expected, it is the gravitational force itself pointing directly towards the Earth.

Furthermore, the aircraft has a tilt angle θ_f with f standing for the *fuselage*. This tilt is referred to as the aircraft's pitch angle. It is the angle between the fuselage and the horizontal plane. In other words, θ_f is the angle between x and X_{Earth}. This rotation occurs when the thrust vector is no longer parallel to the z-axis. Changing the direction of the thrust is controlled by the pilot through the input θ_{long}.

Finally, the flapping angles a_0 and a_1 are discussed in Section 3-4.

Forces and torques

The forces and torques will now be split into several components. First, the thrust of the main rotor is considered and results in a force vector $\boldsymbol{F_m}$. From Figure 3-9, it is clear that:

$$\boldsymbol{F_m} = \begin{bmatrix} f_{x_m} \\ f_{z_m} \end{bmatrix} = \begin{bmatrix} T_m \sin(\beta_{long}) \\ -T_m \cos(\beta_{long}) \end{bmatrix}$$

Gravity results in the gravitational vector $\boldsymbol{F_g}$:

$$\boldsymbol{F_g} = \begin{bmatrix} f_{x_g} \\ f_{z_g} \end{bmatrix} = \begin{bmatrix} -mg \sin(\theta_f) \\ mg \cos(\theta_f) \end{bmatrix}$$

The dissipative forces caused by the drag are then added. The total drag is denoted as D. To decompose the drag in its respective axes, the small angle approximation is made, with the angle between the velocity component in each respective axis and the absolute velocity is considered. As a result:

$$\boldsymbol{F_d} = \begin{bmatrix} f_{x_d} \\ f_{z_d} \end{bmatrix} = \begin{bmatrix} -D\frac{u}{V} \\ -D\frac{w}{V} \end{bmatrix}$$

where u and w are the velocity components in the $x-$ and $z-$axes respectively.

Finally, a less obvious force which is not present in the Free Body Diagram (FBD) from Figure 3-9 is the Coriolis force. The Coriolis force is described as:

$$\boldsymbol{F_c} = -m\boldsymbol{\omega} \times \boldsymbol{V} \tag{3-41}$$

With:

$$\boldsymbol{w} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \text{and} \quad \boldsymbol{V} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

where p, q and r are the roll, pitch and yaw rates respectively; u, v and w are the velocity components of the helicopter in the x, y and z directions respectively. As a 3-DoF model is considered, p, r and v are set to 0. The Coriolis force then becomes:

$$\boldsymbol{F_c} = \begin{bmatrix} f_{x_c} \\ f_{z_c} \end{bmatrix} = \begin{bmatrix} -mqw \\ mqu \end{bmatrix}$$

Finally, the torque about the y axis is considered to allow for pitch motion. The Greek symbol τ will be used to define torques. The only component considered to create a torque is the thrust of the main rotor. Therefore, $\tau_{yy} = \tau_{yy_m}$ and:

$$\tau_{yy_m} = -T_m h_m \sin(\beta_{long}) \tag{3-42}$$

where h_m is the vertical distance between the main rotor and the cog as displayed in Figure 3-9.

By summing up all the forces and torques, the resultants are found so that the equations of motion can be derived.

$$\begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F_m} + \boldsymbol{F_g} + \boldsymbol{F_d} + \boldsymbol{F_c} \\ \tau_{yy_m} \end{bmatrix}$$

Defining the state vector as $\boldsymbol{\xi} = [x \quad z \quad \theta_f \quad | \quad u \quad w \quad q]^\intercal$, the accelerations are defined as:

$$\begin{bmatrix} \dot{u} \\ \\ \dot{w} \\ \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{T_m}{m}\sin(\beta_{long}) - g\sin(\theta_f) - \frac{D}{m}\frac{u}{V} - qw \\ \\ -\frac{T_m}{m}\cos(\beta_{long}) + g\cos(\theta_f) - \frac{D}{m}\frac{w}{V} + qu \\ \\ -\frac{T_m}{I_{yy}}h_m\sin(\beta_{long}) \end{bmatrix}$$

Finally, to complete the equations of motion:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{\theta}_f \end{bmatrix} = \begin{bmatrix} u \\ w \\ q \end{bmatrix}$$

3-6-2 6-degrees-of-freedom body model

The 6-DoF model is derived similarly to the 3-DoF model in 3-6-1. The state vector is now:

$$\boldsymbol{\xi} = \begin{bmatrix} x & y & z & \phi_f & \theta_f & \psi_f \,|\, u & v & w & p & q & r \end{bmatrix}^\intercal$$

x, y and z are displacements in the x, y and z-axes respectively and ϕ_f, θ_f and ψ_f are the roll, pitch and yaw angles respectively. u, v, w, p, q and r are the rates of each of the coordinates just mentioned.

The FBD of Figure 3-9 has now been extended to consider lateral angles.

Looking at Figure 3-10, the lateral forces can be observed. It is important to note that F_t is not acting on the cog. Instead, the force is exerted by the tail at a distance l_t from the cog.

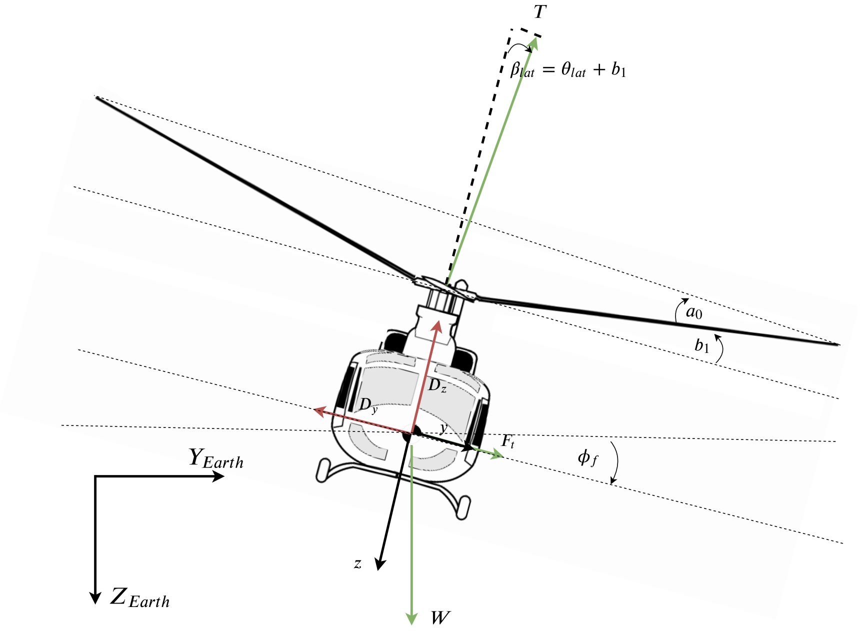


Figure 3-10: Front view of a helicopter free body diagram.

The force vectors are now redefined to match the 6-DoF.

$$\boldsymbol{F_m} = \begin{bmatrix} f_{x_m} \\ f_{y_m} \\ f_{z_m} \end{bmatrix} = \begin{bmatrix} T_m \sin(\beta_{long}) \\ T_m \sin(\beta_{lat}) \\ -T_m \cos(\beta_{long}) \cos(\beta_{lat}) \end{bmatrix}$$

The roll angle now has to be taken into account when considering the gravitational terms.

$$\boldsymbol{F_g} = \begin{bmatrix} f_{x_g} \\ f_{y_g} \\ f_{z_g} \end{bmatrix} = \begin{bmatrix} -mg\sin(\theta_f) \\ mg\cos(\theta_f)\sin(\phi_f) \\ mg\cos(\theta_f)\cos(\phi_f) \end{bmatrix}$$

The drag terms remain the same with the addition of a component in the $y-$direction.

$$\boldsymbol{F_d} = \begin{bmatrix} f_{x_d} \\ f_{y_d} \\ f_{z_d} \end{bmatrix} = \begin{bmatrix} -D\frac{u}{V} \\ -D\frac{v}{V} \\ -D\frac{w}{V} \end{bmatrix}$$

The Coriolis terms no longer have to satisfy the assumption of 0 velocity and rotation in the lateral directions. Therefore:

$$\boldsymbol{F_c} = \begin{bmatrix} f_{x_c} \\ f_{y_c} \\ f_{z_c} \end{bmatrix} = -m \begin{bmatrix} qw - rv \\ ru - pw \\ pv - qu \end{bmatrix}$$

The final force to be considered is the force generated by the tail rotor. The tail rotor has to be considered in this 6-DoF model as it plays a major role to prevent the helicopter from spinning uncontrollably. It is assumed that the tail rotor force is acting at the same height as the cog and that it is acting parallel to the $y-$axis. Therefore:

$$\boldsymbol{F_t} = \begin{bmatrix} f_{x_t} \\ f_{yt} \\ f_{z_t} \end{bmatrix} = \begin{bmatrix} 0 \\ T_t \\ 0 \end{bmatrix}$$

with T_t representing the thrust generated by the tail rotor.

The torque vector $\boldsymbol{\tau}$ now becomes $\boldsymbol{\tau} = [\tau_{xx} \quad \tau_{yy} \quad \tau_{zz}]^\intercal$. It will be split into three parts:

1. $\boldsymbol{\tau_{T_m}}$: torques generated by the offset between the main rotor thrust and cog.
2. $\boldsymbol{\tau_m}$: torques generated by the main rotor torque as a result of C_Q.
3. $\boldsymbol{\tau_{T_t}}$: torques generated by the tail rotor thrust.

To simplify the model, it is assumed that the main rotor is located directly above the cog. As a result, the main rotor does not generate roll motion if the thrust vector is acting directly in the $z-$direction. Similarly, the thrust acting directly in the $z-$direction would then also not result in any pitch motion. Therefore:

$$\boldsymbol{\tau_{T_m}} = \begin{bmatrix} \tau_{xx_m} \\ \tau_{yy_m} \\ \tau_{zz_m} \end{bmatrix} = \begin{bmatrix} f_{y_m} h_m \\ -f_{x_m} h_m \\ 0 \end{bmatrix}$$

The torque generated by the rotation of the main rotor is referred to as Q_m. The torque is then projected into each of its respective axes through the help of the disc tilt angles to give out:

$$\boldsymbol{\tau_m} = \begin{bmatrix} \tau_{xx_m} \\ \tau_{yy_m} \\ \tau_{zz_m} \end{bmatrix} = \begin{bmatrix} Q_m \sin(\beta_{long}) \\ -Q_m \sin(\beta_{lat}) \\ Q_m \cos(\beta_{long}) \cos(\beta_{lat}) \end{bmatrix}$$

The torque generated by the tail rotor simply the thrust of the tail multiplied by the distance of the tail to the cog:

$$\boldsymbol{\tau_{T_t}} = \begin{bmatrix} \tau_{xx_{T_t}} \\ \tau_{yy_{T_t}} \\ \tau_{zz_{T_t}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ f_{yt} l_t \end{bmatrix}$$

Finally, the torques generated by the Coriolis effect are expressed as $\boldsymbol{\tau_c} = -\boldsymbol{\omega} \times (\boldsymbol{I\omega})$ with $\boldsymbol{I} = \text{diag}([I_{xx} \quad I_{yy} \quad I_{zz}])$. I_{xx}, I_{yy} and I_{zz} are the mass moments of inertia about the x, y and $z-$axes respectively. Therefore:

$$\boldsymbol{\tau_c} = \begin{bmatrix} \tau_{xx_c} \\ \tau_{yy_c} \\ \tau_{zz_c} \end{bmatrix} = - \begin{bmatrix} qr(I_{zz} - I_{yy}) \\ pr(I_{xx} - I_{zz}) \\ pq(I_{yy} - I_{xx}) \end{bmatrix}$$

By summing up the torques and forces within their respective axes, the resultants are found.

$$\begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F_m} + \boldsymbol{F_g} + \boldsymbol{F_d} + \boldsymbol{F_c} + \boldsymbol{F_t} \\ \boldsymbol{\tau_{T_m}} + \boldsymbol{\tau_m} + \boldsymbol{\tau_{T_t}} \end{bmatrix}$$

These are then used to describe the system dynamics, yielding the following system:

$$\begin{bmatrix} m\dot{u} \\ m\dot{y} \\ m\dot{w} \\ I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} = \begin{bmatrix} T_m \sin(\beta_{long}) - mg\sin(\theta_f) - D\frac{u}{V} - mqw + mrv \\ -T_m \cos(\beta_{lat}) + mg\cos(\theta_f)\sin(\phi_f) - D\frac{v}{V} + T_t - mru + mpw \\ -T_m h_m \cos(\beta_{long})\cos(\beta_{lat}) + mg\cos(\theta_f)\cos(\phi_f) - D\frac{w}{V} - mpv + mqu \\ T_m \sin(\beta_{lat}) h_m + Q_m \sin(\beta_{long}) - qr(I_{zz} - I_{yy}) \\ -T_m \sin(\beta_{long}) h_m + Q_m \sin(\beta_{lat}) - pr(I_{xx} - I_{zz}) \\ Q_m \cos(\beta_{long})\cos(\beta_{lat}) + T_t l_t - pq(I_{yy} - I_{xx}) \end{bmatrix}$$

To complete the system, the time derivatives of the coordinates are defined as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi}_f \\ \dot{\theta}_f \\ \dot{\psi}_f \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}$$

3-6-3 6-degrees-of-freedom point-mass model

From the 6-DoF point-mass model, it is very simple to go down to 3-DoF. For that reason, this section will only cover the more complex 6-DoF point-mass system.

An important aspect to take note of for point-mass models is that they are defined in the Earth-reference frame. That is, the $z-$axis will always point straight down towards the center of the Earth. The orientation of the aircraft does not affect the direction of the $z-$axis.

The simplest type of helicopter model is made through a point-mass model. The model discussed in this section is taken directly from M. Pavel et al. [26]. The simplicity of a point-mass model is due to the fact that the inputs are forces and torques acting directly in the axes considered. To move a helicopter from point A to point B, the point-mass model will determine the input forces and torques in each direction to achieve that goal.

On the other hand, in a body model, the pilot inputs θ_0, θ_{long} and θ_{lat} need to be determined. As a result, all of the dynamics of the helicopter have to be known to determine the required inputs. Setting the dynamics of the helicopter outside of the control loop means that the pilot inputs do not affect the controller.

However, once the input forces in a point-mass model are known, these can be decomposed to determine the inputs required by a pilot to replicate the motion of the helicopter. This is better explained through an illustration.

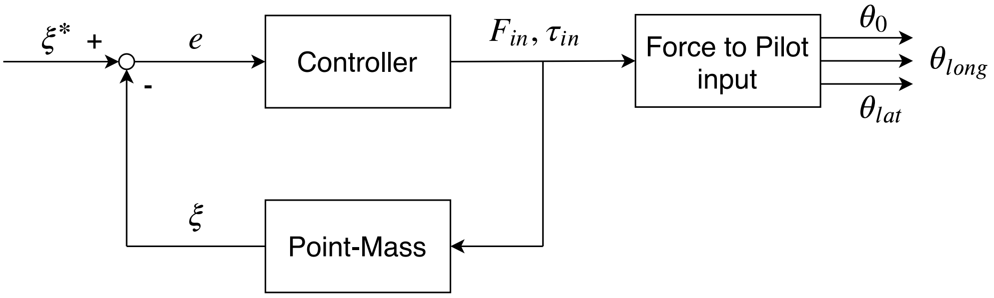


Figure 3-11: Method to calculate pilot inputs through point-mass model.

As it can be seen from Figure 3-11 the point-mass method is a way of working backwards. The inputs of the point-mass model, $\boldsymbol{F_{in}}$ and $\boldsymbol{\tau_{in}}$, are first calculated as a function of the states $\boldsymbol{\xi}$ and the reference signal $\boldsymbol{\xi}^*$ giving rise to an error signal $\boldsymbol{e}$. From the controller outputs, the force to pilot input block gives out the collective and cyclic inputs the pilot has to give to achieve said motion of the helicopter.

To create the point-mass model, four thrust forces will be injected into the system. These are $T_{m,x}$, $T_{m,y}$, $T_{m,z}$ and T_t. The first three forces refer to the thrust of the main rotor into each axis. The latter is the thrust generated by the tail rotor.

To begin with, the main rotor is considered. The total thrust coefficient is split into three components - one for each axis.

$$C_x = C_T \sin(\beta_{long}) \cos(\beta_{lat}) \tag{3-43a}$$

$$C_y = C_T \sin(\beta_{long}) \sin(\beta_{lat}) \tag{3-43b}$$

$$C_z = C_T \cos(\beta_{long}) \cos(\beta_{lat}) \tag{3-43c}$$

Furthermore, the thrust coefficient C_T is calculated as:

$$C_T = \frac{T_m}{\rho (\Omega R)^2 \pi R^2} \tag{3-44}$$

where Ω is the main rotor angular velocity, R is the blade radius and ρ the air density. The thrust components of the main rotor in each direction, T_{m_x}, T_{m_y} and T_{m_z} can then easily be expressed by re-arranging Equation (3-44) for C_T and replacing T and C_T according to the direction of interest.

Having defined the thrust in each axis, the drag now has to be considered. The constant f_e is introduced and referred to as the equivalent flat plate area with $f_e = C_d S_{ref}$, where C_D is the blade drag coefficient and S_{ref} is the equivalent surface area of the blade.

Additionally, the tail rotor thrust T_t has to be considered. This force is acting purely in the $y-$direction and will therefore only cause linear motion within the $y-$axis. Furthermore, the tail rotor is assumed to be located at the same height as the cog and will therefore only cause yaw rotations.

Finally, the velocities in the x, y and z directions are denoted as $u, v,$ and w respectively. Collecting all the forces together yields the following 6 equations of motion:

$$\dot{u} = \left(T_{m_x} - f_e \frac{1}{2}\rho u V\right)\frac{1}{m} \tag{3-45a}$$

$$\dot{v} = \left(T_{m_y} + T_t - f_e \frac{1}{2}\rho v V\right)\frac{1}{m} \tag{3-45b}$$

$$\dot{w} = \left(-T_{m_z} - f_e \frac{1}{2}\rho w V - mg\right)\frac{1}{m} \tag{3-45c}$$

$$\dot{x} = u \tag{3-45d}$$

$$\dot{y} = v \tag{3-45e}$$

$$\dot{z} = w \tag{3-45f}$$

where m is the mass of the helicopter and g is the gravitational constant. V represents the magnitude of the aircrafts' velocity such that $V = \sqrt{u^2 + v^2 + w^2}$.

Finally, the rotations of the mass have to be calculated. The angular velocities p, q, r refer to the roll, pitch and yaw rate. Their respective angles are ϕ_f, θ_f, ψ_f such that $p = \dot{\phi}_f$, $q = \dot{\theta}_f$ and $r = \dot{\psi}_f$. The rotations are controlled by input torques $\tau_{in_{xx}}, \tau_{in_{yy}}$ and $\tau_{in_{zz}}$. This results in the following system:

$$\dot{p} = \frac{\tau_{in_{xx}}}{I_{xx}} \tag{3-46a}$$

$$\dot{q} = \frac{\tau_{in_{yy}}}{I_{yy}} \tag{3-46b}$$

$$\dot{r} = \frac{\tau_{in_{zz}}}{I_{zz}} \tag{3-46c}$$

$$\dot{\phi} = p \tag{3-46d}$$

$$\dot{\theta}_f = q \tag{3-46e}$$

$$\dot{\psi} = r \tag{3-46f}$$

The force to pilot input block in Figure 3-11 is then used to split up the input torques. After the decomposition, the system becomes equivalent to the one from Section 3-6-2.

3-7 Point mass to pilot inputs

In Section 3-6-3, Figure 3-11 illustrates a two step model. The first step is the control of the point-mass system and the second step the conversion of the input forces and torques to pilot inputs. However, this is really carried out in a three step process as the forces are not directly converted into pilot inputs.

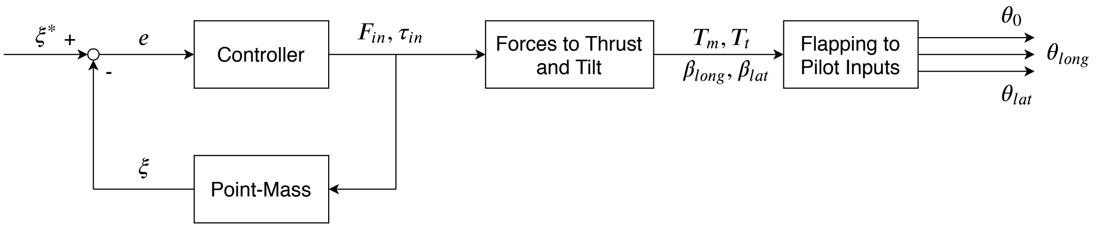


Figure 3-12: Method to calculate pilot inputs through point-mass model.

As it can be seen in Figure 3-12, an extra block is present compared to Figure 3-11. From the forces and torques, the thrusts of the main and tail rotors are first calculated. Simultaneously, the rotor tilt angles are calculated. Given the thrusts and tilt angles, by taking the flapping equations from Section 3-4, the pilot inputs are obtained.

Algorithm 1 From point-mass to pilot inputs.

```
$\boldsymbol{\xi}, \boldsymbol{\xi}^* \leftarrow$ Initialise the states and define setpoint
for $time = t_0, t_0 + dt, \ldots, t_{end}$ do
    $\boldsymbol{e} \leftarrow \boldsymbol{\xi} - \boldsymbol{\xi}^*$ Calculate error signal
    $\boldsymbol{F_{in}}, \boldsymbol{\tau_{in}} \leftarrow C(\boldsymbol{e})$ Calculate input forces and torques from control law $C$ and $\boldsymbol{e}$
    Compute flight parameters: $\lambda_i, \lambda_c, \mu, \alpha_c, V$.
    ----------------------------------------------------------------
    $T_m \leftarrow \sqrt{f_{x_m}^2 + f_{y_m}^2 + f_{z_m}^2}$ Compute the thrust using the controller inputs
    $\beta_{long}, \beta_{lat} \leftarrow f_{x_m} = T_m \sin(\beta_{long}),\ f_{y_m} = T_m sin(\beta_{lat})$
    ----------------------------------------------------------------
    $\theta_0 \leftarrow$ from Equation (3-8)
    $a_0 \leftarrow$ from Equation (3-13)
    $a_1 \leftarrow$ from Equation (3-14)
    $b_1 \leftarrow$ from Equation (3-17)
    $\theta_{long} \leftarrow \beta_{long} - a_1$
    $\theta_{lat} \leftarrow \beta_{lat} + b_1$
    ----------------------------------------------------------------
    $\boldsymbol{\xi}(t + dt) \leftarrow \boldsymbol{\xi} + \dot{\boldsymbol{\xi}} dt$
end for
```

Algorithm 1 clearly splits the entire system into three parts. First, the control inputs are found. The algorithm then runs through the force to thrust block. The pilot inputs are then calculated through the flapping block.

3-8 Control overview

An overview of the inputs and outputs of each model is provided in this section. The goal is to facilitate the understanding of the system by setting the parameters in a clear framework. In this section, the parameters of the control system are given.

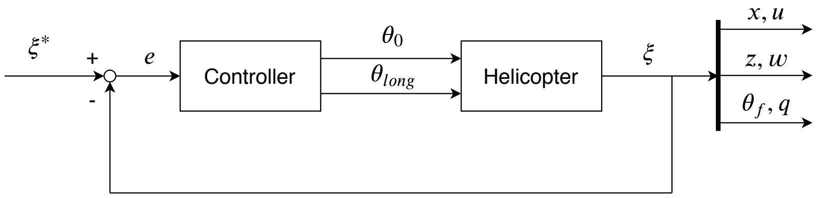


Figure 3-13: Control overview of body model in 3-degrees-of-freedom.

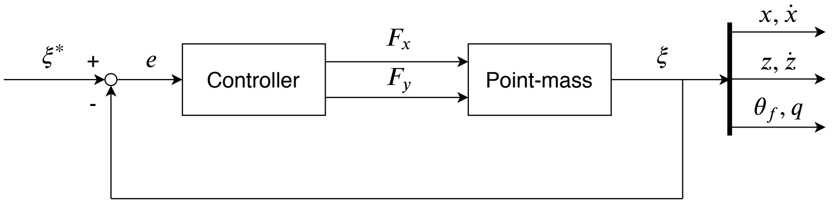


Figure 3-14: Control overview of point-mass model in 3-degrees-of-freedom.

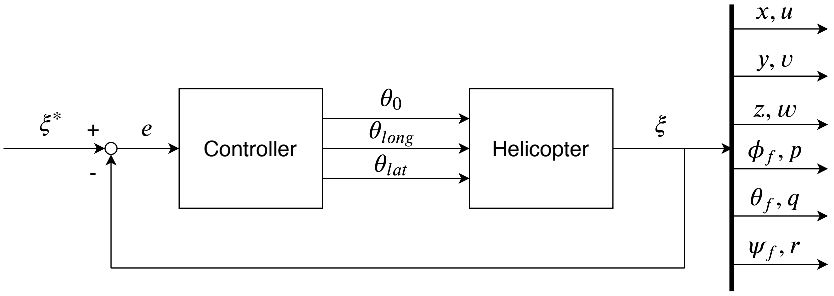


Figure 3-15: Control overview of body model in 6-degrees-of-freedom.

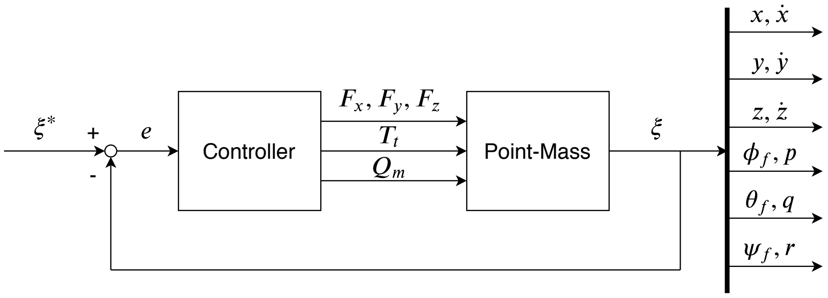


Figure 3-16: Control overview of point-mass model in 6-degrees-of-freedom.

The reader should take note of two important points with regards to the point-mass diagrams.

First, the linear states are reffered to as x, y, z with their derivatives $\dot{x}$, $\dot{y}$, $\dot{z}$ instead of u, v, w. This is due to the fact that the point-mass model is in-fact in an Earth-reference frame rather than a body-frame. For that reason, in the point mass model, $u = \dot{x}\cos(\theta_f)$. The other velocities then also have to be transformed into their respective frames to obtain v and w.

Then, the input torques $\boldsymbol{\tau_{in}}$ cannot be found in Figures 3-14 and 3-16. This is due to the fact that the input torques depend on the input forces and Q_m. Instead, an underactuated system is obtained with $\tau_{in_{xx}}(F_x, Q_m)$, $\tau_{in_{yy}}(T_m, Q_m)$ and $\tau_{in_{zz}}(T_t, Q_m)$.

Part II

Introduction to Passivity-Based Control

The second part of the report covers the theory behind port-Hamiltonian systems. Chapter 4 discusses how a physical system is expressed in the port-Hamiltonian framework. The chapter is concluded with a practical example of a mass-spring-damper system with the derivation of its port-Hamiltonian expression.

Passivity-Based Control is covered in Chapter 5. Background about passive systems is provided and some of their properties are defined such as stability. Control laws are then identified. A method for improving robustness of underactuated systems is also provided.

The part is concluded with the proposal of a novel tuning approach for port-Hamitlonian systems. The method can be utilised to create a fast controller that minimises overshoot to let a system be perceived similarly to a critically damped mass-spring-damper system.

Chapter 4

Port-Hamiltonian systems

Henry Paynter is known as the inventor of bond graphs[1] and as a pioneer in port-based modelling [28]. Port-based models present many benefits in modern day control thanks to their ability to describe complex nonlinear systems in an intuitive way. This intuition as Arjan van der Schaft described it [4], originates from the fact that energy is recognised as the 'lingua franca' between physical domains.

Chapter 4 aims to present the theory behind port-Hamiltonian systems and provide the reader with an intuitive understanding of their processes. Hamiltonian systems are first introduced in Section 4-1 where the Hamiltonian system is derived from the well-known Lagrangian system. Ports are then introduced in Section 4-2 where they are combined with Hamiltonian systems to create the so-called port-Hamiltonian systems. To better understand the theory, the derivation of a mass-spring system in the port-Hamiltonian framework is provided in Section 4-3. Finally, Part 2 is concluded with Chapter 6 where an analytical tuning method to minimise settling time and oscillations is developed.

4-1 Hamiltonian systems

Hamiltonian systems are closely related to Lagrangian systems. Through simple steps shown in Section 4-1-1, the Hamiltonian of the system is derived. Once the Hamiltonian is described, the Hamiltonian *system* is introduced in Section 4-1-2.

4-1-1 From Lagrangian to Hamiltonian

The Lagrangian, $\mathcal{L}$, of a body corresponds to the difference between its kinetic and potential energies, K and U respectively:

---

[1] Bond graphs are a graphical way of denoting physical systems. The physical system is split into the elements that compose it. They are linked by flows of energy between each other and its surroundings [27]. The links between each element are called ports and have given rise to port-based modelling methods.

$$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) = K(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) - U(\boldsymbol{q}, t) \tag{4-1}$$

where $\boldsymbol{q} \in \mathbb{R}^k$ is the vector of generalised coordinates and t is time. Generalised coordinates are generally spacial coordinates, which describe the linear and angular positions of an object in space. The dot operator, such as for $\dot{\boldsymbol{q}}$, corresponds to the time derivative of the variable beneath it. The Lagrangian is a function of time, t, as it depends on the current energy state of the body. For simplification, the t term will now be dropped. It is, however, important to always remember that all states depend on time, hence $\boldsymbol{q} = \boldsymbol{q}(t)$.

Contrary to the Lagrangian, the Hamiltonian represents the total energy of the system - the sum of its potential and kinetic energies. The Hamiltonian, H, can directly be computed from the Lagrangian by taking its Legendre transform [29]. Introducing the generalised momenta (linear as well as angular momenta) $\boldsymbol{p} \in \mathbb{R}^k$, H can be written as:

$$H(\boldsymbol{p}, \boldsymbol{q}) := \boldsymbol{p}^\intercal \dot{\boldsymbol{q}} - \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{4-2}$$

In Equation (4-2), $\boldsymbol{q}$ and $\boldsymbol{p}$ for Lagrangian systems correspond to the same generalised coordinates and momenta as in the Hamiltonian framework. Furthermore, due to the fact that $\dot{\boldsymbol{q}}$ is not a state, it is required that the Hamiltonian be expressed solely in terms of $\boldsymbol{q}$ and $\boldsymbol{p}$. This is done using the following equivalence:

$$\boldsymbol{p} = \frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{4-3}$$

4-1-2 Defining the Hamiltonian system

Before defining a system, its states are characterised. Hamiltonian and port-Hamiltonian (PH) systems both work with coordinates and momenta ($\boldsymbol{q}$ and $\boldsymbol{p}$ respectively) as states. A new state vector called the phase space, $\boldsymbol{\xi} \in \mathbb{R}^{2k} = \mathbb{R}^n$, is therefore defined as:

$$\boldsymbol{\xi} = [q_1, \dots, q_k, p_1, \dots, p_k]^\intercal = [\boldsymbol{q}^\intercal, \quad \boldsymbol{p}^\intercal]^\intercal \tag{4-4}$$

Having defined the phase space, it is also required to define the output of the system. To do so, one has to remember that the change in energy in the system is equivalent to the work done by the system. Knowing this allows the rate of change in energy of the system to be defined. This is done through energy balancing as given as derived in [30]:

$$\frac{dH}{dt}(\boldsymbol{\xi}) = \dot{\boldsymbol{q}}^\intercal \boldsymbol{F} \tag{4-5}$$

where $\boldsymbol{F}$ denotes the vector of external generalised forces applied to the system. As it will be discussed later in Chapter 5, balancing the energy of the system plays an important role for control. This motivates the use of $\dot{\boldsymbol{q}}$ as output of the system.

The Hamiltonian system is now described by the standard Hamiltonian equations as in [30].

$$\mathcal{H}: \begin{cases} \dot{\boldsymbol{q}} = \frac{\partial H}{\partial \boldsymbol{p}}(\boldsymbol{\xi}) \\ \dot{\boldsymbol{p}} = -\frac{\partial H}{\partial \boldsymbol{q}}(\boldsymbol{\xi}) + \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{B}^{\mathsf{T}}(\boldsymbol{q})\frac{\partial H}{\partial \boldsymbol{p}}(\boldsymbol{\xi})\,(= \boldsymbol{B}^{\mathsf{T}}(\boldsymbol{q})\dot{\boldsymbol{q}}) \end{cases} \tag{4-6}$$

where the input to the system is $\boldsymbol{u} \in \mathbb{R}^m$, while $\boldsymbol{B}(\boldsymbol{q}) \in \mathbb{R}^{k \times m}$ is a matrix for which the product $\boldsymbol{B}(\boldsymbol{q})\boldsymbol{u}$ yields a vector of generalised forces inserted into the system. Special attention should be paid to the case when $m \leq k$ as the system is then said to be under-actuated.

4-2 Port-Hamiltonian expression of systems

Before looking at the mathematical expressions of a PH system, some insight on how these systems work is provided in Section 4-2-1. Section 4-2-2 introduces conservative systems (systems with no energy losses/dissipation). More realistic systems with energy losses are then treated in Section 4-2-3.

4-2-1 Insights on how port-Hamiltonian models work

The insights provided in this Section are based on Arjan van der Schaft's description of PH systems in [30].

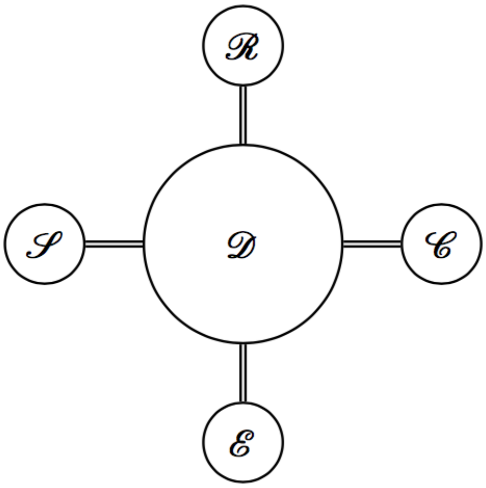


Figure 4-1: Illustration of the energy transfers in a port-Hamiltonian system.

Looking at Figure 4-1, the structure $\mathcal{D}$ allows the interconnection between all nodes. As it can be seen, four ports are present. These can be split into two categories, namely, the internal ports and the external ports.

1. **Internal ports:**
    - $\boldsymbol{\mathcal{S}}$, energy storage port.
    - $\boldsymbol{\mathcal{R}}$, energy dissipation port.
2. **External ports:**
    - $\boldsymbol{\mathcal{C}}$, control port.
    - $\boldsymbol{\mathcal{E}}$, environment interaction port.

A system has a given amount of energy $\boldsymbol{\mathcal{S}}$. However, the energy stored in $\boldsymbol{\mathcal{S}}$ can be reduced due to the resistive/damping forces in $\boldsymbol{\mathcal{R}}$. A control port, $\boldsymbol{\mathcal{C}}$, is also present to manipulate the output of the system as desired. Finally, any other interactions with the surroundings are captured within the environment node $\boldsymbol{\mathcal{E}}$.

4-2-2 Lossless port-Hamiltonian

Having defined an expression for the Hamiltonian system, the PH system may now be derived. A PH model considers the energy transfers between lumped masses. This transfer of energy is introduced through the skew-symmetric[2] matrix $\boldsymbol{\mathcal{J}} \in \mathbb{R}^{n\times n}$, called the interconnection matrix. Further defining $\boldsymbol{g}(\boldsymbol{\xi}) = [\mathbf{0} \quad \mathbf{B}^\intercal(\mathbf{q})]^\intercal$, $\boldsymbol{g}(\boldsymbol{\xi}) \in \mathbb{R}^{n\times m}$, the lossless PH system is defined as:

$$\mathcal{H} : \begin{cases} \dot{\boldsymbol{\xi}} = \boldsymbol{\mathcal{J}}(\boldsymbol{\xi})\frac{\partial H}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}) + \boldsymbol{g}(\boldsymbol{\xi}, \boldsymbol{u}) \\ \boldsymbol{y} = \boldsymbol{g}^\intercal(\boldsymbol{\xi})\frac{\partial H}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}) \end{cases} \tag{4-7}$$

with $\boldsymbol{u} \in \mathbb{R}^m$ defined as the input to the system. Similarly to $\boldsymbol{g}(\boldsymbol{\xi})\boldsymbol{u}$ in Equation (4-7), the vector $\boldsymbol{g}(\boldsymbol{\xi}, \boldsymbol{u})$ captures the $\boldsymbol{g}(\boldsymbol{\xi})\boldsymbol{u}$ term as well as constant source inputs [31].

4-2-3 Energy losses in port-Hamiltonian systems

For many systems, the dissipation of energy cannot be ignored. The damping matrix $\boldsymbol{\mathcal{R}}$ is therefore introduced. $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{n\times n}$ consists of damping and resistive elements. $\boldsymbol{\mathcal{R}}$ is symmetric, such that $\boldsymbol{\mathcal{R}} = \boldsymbol{\mathcal{R}}^\intercal$.

$$\mathcal{H} : \begin{cases} \dot{\boldsymbol{\xi}} = (\boldsymbol{\mathcal{J}}(\boldsymbol{\xi}) - \boldsymbol{\mathcal{R}}(\boldsymbol{\xi}))\frac{\partial H}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}) + \boldsymbol{g}(\boldsymbol{\xi})\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{g}^\intercal(\boldsymbol{\xi})\frac{\partial H}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}) \end{cases} \tag{4-8}$$

4-3 Deriving the port-Hamiltonian expression of a physical system

In order to better understand some of the concepts introduced in this chapter, a practical example of a mass-spring system is taken and its dynamics are derived in the port-Hamiltonian framework based on the work from [4].

---

[2] Skew-symmetric: $\boldsymbol{\mathcal{J}} = -\boldsymbol{\mathcal{J}}^\intercal$

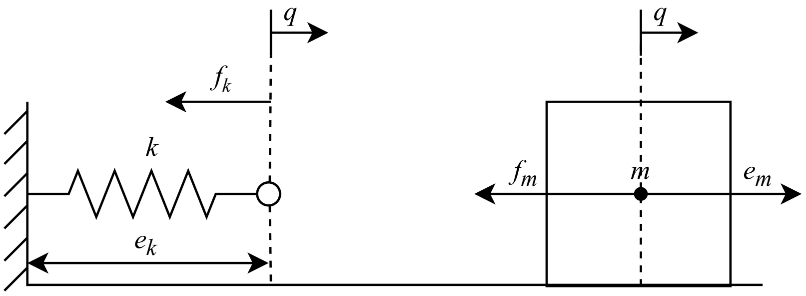


Figure 4-2: Simple mass spring system with frictionless surface.

Figure 4-2 shows a mass-spring system with mass m, and spring constant k. The separation between the spring and the mass is for illustration purposes to demonstrate the forces acting on each individual system component.

For this example, a frictionless surface is assumed. Additional parameters can be seen in the image: $-f_k$ and e_k represent the velocity and force at the endpoint of the spring respectively, whereas $-f_m$ is used to denote the force exerted on the mass and e_m the velocity of the mass. Finally, the extension of the spring is expressed as the generalised coordinate q.

First, the Hamiltonian (total energy of the system) is derived as the sum of the kinetic and potential energies of the system. The mass carries the kinetic energy, whereas the spring holds the potential energy. It is known that the potential energy of a spring is equivalent to: $U = \frac{1}{2}kq^2$. To express the kinetic energy, the momentum term p is used to give: $K = \frac{1}{2m}p^2$. Therefore, the total energy of the system is expressed as:

$$H(q,p) = K + U = \frac{1}{2m}p^2 + \frac{1}{2}kq^2 \tag{4-9}$$

As it can be seen, there has been a clear separation in the system to create two subsystems - a spring part and a mass counterpart. The dynamics of the spring can then be expressed as:

$$\text{Spring dynamics: } \begin{cases} \dot{q} = -f_k \\ e_k = \frac{d}{dq}\left(\frac{1}{2}kq^2\right) \end{cases} \tag{4-10}$$

whereas the dynamics of the mass block are expressed in terms of momenta:

$$\text{Mass dynamics: } \begin{cases} \dot{q} = -f_m \\ e_m = \frac{d}{dp}\left(\frac{1}{2m}p^2\right) \end{cases} \tag{4-11}$$

The interconnection can now clearly be seen and yields:

$$\text{Interconnection of subsystems: } \begin{cases} -f_k = e_m \\ f_m = e_k \end{cases} \tag{4-12}$$

The total system then yields:

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\mathcal{J}} \begin{bmatrix} \frac{\partial H(q,p)}{\partial q} \\ \frac{\partial H(q,p)}{\partial p} \end{bmatrix}$$

Dissipation due to a damper can easily be added through the help of a dissipation matrix $\boldsymbol{\mathcal{R}}$. Considering a damper with damping constant c_c, the damping force F_d is given by $F_d = c_c \frac{p}{m}$. The total system with dissipation then becomes:

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \left(\underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\mathcal{J}} - \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & c_c \end{bmatrix}}_{\mathcal{R}} \right) \begin{bmatrix} \frac{\partial H(q,p)}{\partial q} \\ \frac{\partial H(q,p)}{\partial p} \end{bmatrix}$$

Chapter 5

Passivity-Based Control

Automatic control systems have been found to have existed as far back as 2,000 years ago [32]. The Egyptians had built a mechanical water clock in Alexandria. However, it was not until 1868 that Maxwell described instability of a system using differential equations [33]. Since then, control theory has made considerable strides forward. With the evolution of control theory, a vast variety of controllers have become available. A notable branch is that of Passivity-Based Control (PBC).

Jan C. Willems first introduced passive systems in 1972 [34]. In defining passive/dissipative systems, Willems quickly identified their benefits as he stated in his early work that PBC would allow one to create a more intuitive understanding of the effects of the controller [35]. Indeed, the use of energy in PBC provides its designer a better understanding of the signals provided to and by the controller - making PBC an attractive solution for control engineers.

This chapter first describes the concept of a passive system with the introduction of various definitions in Section 5-1. Stability conditions of passive systems are then covered in Section 5-2. Ways of interconnecting passive systems are presented in Section 5-3. Finally, methods of deriving a PBC are treated in Section 5-4.

5-1 Passive systems

To define the concepts used in passivity theory, consider a nonlinear system $\mathcal{H}$ with inputs $\boldsymbol{u} \in \mathbb{R}^m$, outputs $\boldsymbol{y} \in \mathbb{R}^m$, and states $\boldsymbol{\xi} \in \mathbb{R}^n$.

$$\mathcal{H} : \begin{cases} \dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{\xi}) + \boldsymbol{g}(\boldsymbol{\xi})\boldsymbol{u}, \\ \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{\xi}) + \boldsymbol{l}(\boldsymbol{\xi})\boldsymbol{u} \end{cases} \tag{5-1}$$

A passive or dissipative system is one that rather than generating energy, stores it and/or dissipates it. The inputs and outputs of a system allow it to exchange energy with its environment. From this brief definition, it is clear that passive systems can be successfully used to model physical systems such as helicopters.

The supply rate of energy to the system is defined as $w(\boldsymbol{u}, \boldsymbol{y})$. The storage function $V(\boldsymbol{\xi}(t))$ represents the energy stored in the system depending on the states at a instant in time. A more formal definition then states:

A system $\mathcal{H}$ is said to be passive if there exists a storage function $V(\boldsymbol{\xi}(t)) \geq 0$ such that:

$$\int_0^t w(\boldsymbol{u}(s), \boldsymbol{y}(s))ds \geq V(\boldsymbol{\xi}(t)) - V(\boldsymbol{\xi}(0)) \tag{5-2}$$

holds for $t > 0$, with $t \in \mathbb{R}$ [34]. It should be noted that a system has a finite amount of energy meaning that the energy is bounded both from above and below.

As a result, the storage function must be bounded as $0 \leq V(\boldsymbol{\xi}(t)) < \infty$. Furthermore, if the condition from Equation (5-2) holds with a strict inequality, the system is said to be *strictly passive* and *lossless* if the condition holds at equality.

Taking the time derivative of Equation (5-2), a new expression is found, independent of the time trajectories of $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$. That is, only the values of $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$ at the current time instant are needed.

$$\dot{V} = \frac{\partial^{\intercal} V}{\partial \boldsymbol{\xi}} \dot{\boldsymbol{\xi}} = \frac{\partial^{\intercal} V}{\partial \boldsymbol{\xi}} \left(f(\boldsymbol{\xi}) + g(\boldsymbol{\xi})\boldsymbol{u} \right) \leq w(\boldsymbol{u}, \boldsymbol{y}) \tag{5-3}$$

It is important to realise that the supply rate $w(\boldsymbol{u}, \boldsymbol{y})$ can in fact be written as the product of the inputs and outputs:

$$w(\boldsymbol{u}, \boldsymbol{y}) = \boldsymbol{u}^{\intercal} \boldsymbol{y} \tag{5-4}$$

This notation for the supply rate is more explicit and will therefore be used from now on. It will be seen in the following sections that Equation (5-2) is more useful when written as an equality. To do so, a dissipative term, $d(t)$ is introduced.

$$V(\boldsymbol{\xi}(t)) - V(\boldsymbol{\xi}(0)) = \int_0^t w(\boldsymbol{u}(s), \boldsymbol{y}(s))ds - d(t) \tag{5-5}$$

By allowing $d(t) \geq 0$ Equation (5-5) becomes valid for both strictly passive systems as well as lossless systems as energy is dissipated if $d(t) > 0$ and the system is lossless if $d(t) = 0$.

Using the same thought process to understand the definitions provided above, the additional terminology introduced by Khalil [36] can easily be understood. Khalil states that a system $\mathcal{H}$ is said to be

- *Lossless* if $\boldsymbol{u}^{\intercal}\boldsymbol{y} = \dot{V}$.
- *Strictly passive* if $\boldsymbol{u}^{\intercal}\boldsymbol{y} \geq \dot{V} + \psi(\boldsymbol{\xi})$ with $\psi(\boldsymbol{\xi}) > 0$.
- *Input strictly passive* if $\boldsymbol{u}^{\intercal}\boldsymbol{y} \geq \dot{V} + \boldsymbol{u}^{\intercal}\phi(\boldsymbol{u})$ and $\boldsymbol{u}^{\intercal}\phi(\boldsymbol{u}) > 0$, $\forall \boldsymbol{u} \neq \boldsymbol{0}$.
- *Output strictly passive* if $\boldsymbol{u}^{\intercal}\boldsymbol{y} \geq \dot{V} + \boldsymbol{y}^{\intercal}\rho(\boldsymbol{y})$ and $\boldsymbol{y}^{\intercal}\rho(\boldsymbol{y}) > 0$, $\forall \boldsymbol{y} \neq \boldsymbol{0}$.

5-2 Lyapunov stability of dissipative systems

Given a dissipative system $\mathcal{H}$ with supply rate $w(\boldsymbol{u}, \boldsymbol{y})$ with $w(\mathbf{0}, \mathbf{0}) = 0$ and $w(\mathbf{0}, \boldsymbol{y}) \leq 0$ $\forall \boldsymbol{y} \in \mathbb{R}^m$, all trajectories starting in $\mathcal{V}_\alpha$ will remain in $\mathcal{V}_\alpha$ as:

$$V(\boldsymbol{\xi}(t)) \leq \int_0^t w(0, \boldsymbol{y}(s))) \, ds + V(\boldsymbol{\xi}(0)) \leq V(\boldsymbol{\xi}(0)) \tag{5-6}$$

If

$$V(\boldsymbol{\xi}) > 0, \quad \forall \boldsymbol{\xi} \in \mathbb{R}^n, \boldsymbol{\xi} \neq \mathbf{0}, \quad \text{and} \quad V(\boldsymbol{\xi}(0)) = 0, \tag{5-7}$$

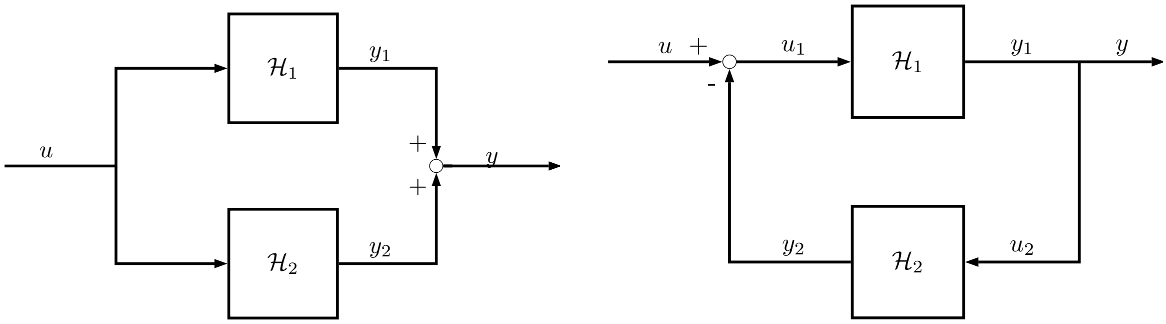
then, the storage function $V(\boldsymbol{\xi})$ can in fact be used as a Lyapunov function and *the origin is Lyapunov stable*. If $V(\boldsymbol{\xi})$ is *radially unbounded*[1] , the origin is *globally stable* [3].

From Lyapunov theory, *asymptotic stability* can be concluded if the inequality from Equation (5-8) also holds for the autonomous system ($\boldsymbol{u} = \mathbf{0}$).

$$\dot{V}(\boldsymbol{\xi}) = V^\mathsf{T}(\boldsymbol{\xi}) f(\boldsymbol{\xi}) \leq w(0, h(\boldsymbol{\xi})) \tag{5-8}$$

5-3 Interconnection of passive systems

Bai et al. [37] discussed the preservation of certain passivity properties upon interconnecting passive systems according to varying structures.



(a) Parallel interconnection. **(b)** Negative feedback interconnection.

Figure 5-1: Interconnection variations of passive systems.

Two interconnection structures are considered, namely *parallel interconnection*, and *negative feedback interconnection* as depicted in Figures 5-1a and 5-1b. Consider two dissipative systems $\mathcal{H}_1$ and $\mathcal{H}_2$.

- The *parallel interconnection* is *passive from* $\boldsymbol{u}$ *to* $\boldsymbol{y}$.
- The *negative feedback interconnection* is *passive from* $\boldsymbol{u}$ *to* $\boldsymbol{y}$.

---

[1] A function $f(x) \in \mathbb{R}$ is said to be radially unbounded when $||x|| \to \infty \implies f(x) \to \infty$.

5-4 Interconnection and Damping Assignment - Passivity-Based Control

Interconnection and Damping Assignment for PBC is a theory developed by R. Ortega et al. [31] and will from now on be referred to as IDA-PBC. Consider a system of the form:

$$\mathcal{H} : \begin{cases} \dot{\boldsymbol{\xi}} = (\boldsymbol{\mathcal{J}}(\boldsymbol{\xi}) - \boldsymbol{\mathcal{R}}(\boldsymbol{\xi}))\frac{\partial H}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}) + \boldsymbol{g}(\boldsymbol{\xi})\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{g}\left(\boldsymbol{\xi}\right)^{\mathsf{T}}\left(\boldsymbol{\xi}\right)\frac{\partial H}{\partial x}(x) \end{cases} \tag{5-9}$$

An expression has to be found for the control input $\boldsymbol{u_{in}}$. This will be done through energy balancing and by shaping the energy the system to the desired reponse.

5-4-1 Energy balance

The first step is to look at the energy balance of the system, that is, the rate of change in total energy of the system is the supply rate of energy minus the rate at which the energy is dissipated from the system.

$$\dot{H} = -\underbrace{\left[\frac{\partial H}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi})\right]^{\mathsf{T}} \boldsymbol{\mathcal{R}}(\boldsymbol{\xi})\frac{\partial H}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi})}_{\text{dissipation rate}} \quad + \quad \underbrace{\boldsymbol{u}^{\mathsf{T}}\boldsymbol{y}}_{\text{supply rate}} \tag{5-10}$$

By integrating Equation (5-10) with respect to time, t, the *energy-balance* equation is obtained:

$$\underbrace{\int_0^t \boldsymbol{u}^{\mathsf{T}}(s)\boldsymbol{y}(s)ds}_{\text{energy supplied}} = \underbrace{H\left(\boldsymbol{\xi}(t)\right) - H\left(\boldsymbol{\xi}(0)\right)}_{\text{stored energy}} + \underbrace{\int_0^t \left[\frac{\partial H}{\partial \boldsymbol{\xi}}\left(\boldsymbol{\xi}(s)\right)\right]^{\mathsf{T}} \boldsymbol{\mathcal{R}}\left(\boldsymbol{\xi}(s)\right) \frac{\partial H}{\partial \boldsymbol{\xi}}\left(\boldsymbol{\xi}(s)\right) ds}_{\text{energy dissipated}}, \qquad \forall t \geq 0. \tag{5-11}$$

This clearly shows that a finite amount of energy can be extracted from the system. Furthermore, using the assumption that $\mathcal{H}$ is a passive system, it is known that $\dot{H} \leq 0$. This can be seen by applying the energy-balance condition of the autonomous system, where $\boldsymbol{u} = \boldsymbol{0}$ as $H\left(\boldsymbol{\xi}(0)\right) \leq H\left(\boldsymbol{\xi}(t)\right)$. Additionally, for the system to be passive, the total energy must also be bounded from below. It is typically sufficient to say that $H\left(\boldsymbol{\xi}(t)\right) \geq 0$.

5-4-2 Energy shaping

The discussion following the definition of *energy-balance* suggests that, as energy is dissipated from the system, the system will go towards its point of minimum energy. The goal is to now design a controller that will reshape $H\left(\boldsymbol{\xi}(t)\right)$ such that the state of minimum energy is located at the setpoint $\boldsymbol{\xi}^{*}$.

To do so, consider a *static state-feedback* controller for which the control action $\boldsymbol{u} = \boldsymbol{\beta}(\boldsymbol{\xi}) + \boldsymbol{\nu}$ has to be found. The control law yields a new expression for the energy-balance equation:

$$H_d\left(\boldsymbol{\xi}(t)\right) - H_d\left(\boldsymbol{\xi}(0)\right) = \int_0^t \boldsymbol{\nu}^\intercal(s)\boldsymbol{y}'(s)ds - d(t), \tag{5-12}$$

with $H_d(\boldsymbol{\xi}(t))$ the desired total energy function at time t with minimum at the setpoint $\boldsymbol{\xi}^*$. Furthermore, *damping injection* is used and represented by the term $d(t)$ with $d(t) \geq 0$. The damping term is typically an increasing function as this increases the convergence rate towards the point $\boldsymbol{\xi}^*$. Finally, $\boldsymbol{y}'$ is the new passive output and can also be equivalent to $\boldsymbol{y}$. By taking $\boldsymbol{\nu} = \boldsymbol{0}$ and $H_d(\boldsymbol{\xi})$ as the Lyapunov function, it can clearly be seen that the controlled system is Lyapunov stable.

5-4-3 Controller design - Stabilisation via energy-balancing

The typical controller design procedure for an IDA-PBC of a port-controlled Hamiltonian (PCH) is a three step process [31].

1. Fix the the structure of the internal energy exchanges of the PH system through the interconnection and damping matrices (Fix $\boldsymbol{\mathcal{J}}$ and $\boldsymbol{\mathcal{R}}$).
2. Derive a Partial Differential Equation (PDE) from the matrices to obtain a set of feasible energy functions.
3. From the set of feasible solutions, select one that satisfies the minimum energy requirement at the point $\boldsymbol{\xi}^*$ with state feedback $\boldsymbol{u} = \boldsymbol{\beta}(\boldsymbol{\xi})$.

To tackle this problem, Ortega et al. [31] make use of energy-balancing methods and introduce a new storage function, $H_a(\boldsymbol{\xi})$, which satisfies the following condition:

$$H_d(\boldsymbol{\xi}) = H(\boldsymbol{\xi}) + H_a(\boldsymbol{\xi}) \tag{5-13}$$

As it can be seen from Equation (5-13), the desired Hamiltonian depends on the Hamiltonian of the physical system. Expressing the desired storage function as such allows the derivation of the PDE mentioned in step 2 of the design process.

It is then said that the energy balance of the closed-loop is equivalent to the difference between the total energy of the open-loop system and the energy transferred by the controller with an additional constant κ determined by the initial conditions:

$$H_d\left(\boldsymbol{\xi}(t)\right) = H\left(\boldsymbol{\xi}(t)\right) - \int_0^t \boldsymbol{u}^\intercal(\boldsymbol{s})\boldsymbol{y}(\boldsymbol{s})ds + \kappa \tag{5-14}$$

Perhaps, a more intuitive way of interpreting Equation 5-14 is to consider the fact that the desired total energy $(H_d\left(\boldsymbol{\xi}\right))$ is the sum of the system's total energy and the energy added by the controller. Following directly from this reasoning, the closed-loop system should then behave as the desired *'autonomous'* system:

$$\dot{\boldsymbol{\xi}} = \left(\boldsymbol{\mathcal{J}}_d - \boldsymbol{\mathcal{R}}_d\right)\frac{\partial H_d}{\partial \boldsymbol{\xi}} \tag{5-15}$$

Considering a simple state feedback controller $\boldsymbol{u} = \boldsymbol{\beta}(\boldsymbol{\xi})$, the system from Equation (5-15) is equivalent to the following expression.

$$\dot{\boldsymbol{\xi}} = (\boldsymbol{\mathcal{J}} - \boldsymbol{\mathcal{R}}) \frac{\partial H}{\partial \boldsymbol{\xi}} + \boldsymbol{g}(\boldsymbol{\xi}) \beta(\boldsymbol{\xi}) = (\boldsymbol{\mathcal{J}}_{\boldsymbol{d}} - \boldsymbol{\mathcal{R}}_{\boldsymbol{d}}) \frac{\partial H_d}{\partial \boldsymbol{\xi}} \tag{5-16}$$

Combining Equations (5-16) and (5-13), a new expression is obtained:

$$\begin{aligned} (\boldsymbol{\mathcal{J}} - \boldsymbol{\mathcal{R}}) \frac{\partial H}{\partial \boldsymbol{\xi}} + \boldsymbol{g}(\boldsymbol{\xi}) \beta(\boldsymbol{\xi}) &= (\boldsymbol{\mathcal{J}}_{\boldsymbol{d}} - \boldsymbol{\mathcal{R}}_{\boldsymbol{d}}) \frac{\partial H}{\partial \boldsymbol{\xi}} + (\boldsymbol{\mathcal{J}}_{\boldsymbol{d}} - \boldsymbol{\mathcal{R}}_{\boldsymbol{d}}) \frac{\partial H_a}{\partial \boldsymbol{\xi}} \\ \implies [(\boldsymbol{\mathcal{J}} - \boldsymbol{\mathcal{J}}_d) - (\boldsymbol{\mathcal{R}} - \boldsymbol{\mathcal{R}}_d)] \frac{\partial H}{\partial \boldsymbol{\xi}} + \boldsymbol{g}(\boldsymbol{\xi}) \beta(\boldsymbol{\xi}) &= (\boldsymbol{\mathcal{J}}_{\boldsymbol{d}} - \boldsymbol{\mathcal{R}}_{\boldsymbol{d}}) \frac{\partial H_a}{\partial \boldsymbol{\xi}} \end{aligned} \tag{5-17}$$

By introducing the annihilator[2] $\boldsymbol{g}^{\perp}(\boldsymbol{\xi})$ with $\boldsymbol{g}^{\perp}(\boldsymbol{\xi}) \boldsymbol{g}(\boldsymbol{\xi}) = \boldsymbol{0}$, the set of all possible solutions of the PDE is obtained by solving for H_a. The properties of the annihilator are introduced in [39].

$$\boldsymbol{g}^{\perp}(\boldsymbol{\xi}) [(\boldsymbol{\mathcal{J}} - \boldsymbol{\mathcal{J}}_d) - (\boldsymbol{\mathcal{R}} - \boldsymbol{\mathcal{R}}_d)] \frac{\partial H}{\partial \boldsymbol{\xi}} = \boldsymbol{g}^{\perp}(\boldsymbol{\xi}) (\boldsymbol{\mathcal{J}}_{\boldsymbol{d}} - \boldsymbol{\mathcal{R}}_{\boldsymbol{d}}) \frac{\partial H_a}{\partial \boldsymbol{\xi}} \tag{5-18}$$

Solving the PDE for H_a and picking a suitable solution from the set of all solutions, the control law may now be determined. To solve for $\boldsymbol{\beta}(\boldsymbol{\xi})$, Equation (5-17) will once again be used. This time, the Moore-Penrose inverse is applied due to the fact that $\boldsymbol{g}$ is often not square (and/or non-full-rank). The Moore-Penrose inverse is indicated by the symbol † with the property that $\boldsymbol{g}^{\dagger} = (\boldsymbol{g}^{\intercal}\boldsymbol{g})^{-1} \boldsymbol{g}^{\intercal}$. Solving for $\boldsymbol{\beta}(\boldsymbol{\xi})$ is then simply done by applying Equation (5-19).

$$\boldsymbol{\beta}(\boldsymbol{\xi}) = \boldsymbol{g}^{\dagger}(\boldsymbol{\xi}) \left((\boldsymbol{\mathcal{J}}_{\boldsymbol{d}} - \boldsymbol{\mathcal{R}}_{\boldsymbol{d}}) \frac{\partial H_a}{\partial \boldsymbol{\xi}} - [(\boldsymbol{\mathcal{J}} - \boldsymbol{\mathcal{J}}_d) - (\boldsymbol{\mathcal{R}} - \boldsymbol{\mathcal{R}}_d)] \frac{\partial H}{\partial \boldsymbol{\xi}} \right) \tag{5-19}$$

or, by rearranging, a simpler expression is obtained:

$$\boldsymbol{\beta}(\boldsymbol{\xi}) = \boldsymbol{g}^{\dagger}(\boldsymbol{\xi}) \left((\boldsymbol{\mathcal{J}}_{\boldsymbol{d}} - \boldsymbol{\mathcal{R}}_{\boldsymbol{d}}) \frac{\partial H_d}{\partial \boldsymbol{\xi}} - (\boldsymbol{\mathcal{J}} - \boldsymbol{\mathcal{R}}) \frac{\partial H}{\partial \boldsymbol{\xi}} \right) \tag{5-20}$$

5-4-4 Closed loop stability

Considering that the input of the system is given by Equation (5-20), the closed loop system is simply defined by $\mathcal{H}_d$. It is known that the change in energy of a system $\mathcal{H}$ is defined by:

$$\dot{H} = -\frac{\partial H^{\intercal}}{\partial \boldsymbol{\xi}} \boldsymbol{\mathcal{R}} \frac{\partial H}{\partial \boldsymbol{\xi}} \tag{5-21}$$

To ensure asymptotic stability, Equation (5-21) is simply adapted for the closed loop and the stability conditions are once again applied [40].

$$\dot{H}_d = -\frac{\partial H_d^{\intercal}}{\partial \boldsymbol{\xi}} \boldsymbol{\mathcal{R}}_{\boldsymbol{d}} \frac{\partial H_d}{\partial \boldsymbol{\xi}} < 0, \quad \forall \boldsymbol{\xi} \neq \boldsymbol{\xi} \quad \text{and} \quad \dot{H}_d(\boldsymbol{\xi}^*) = \boldsymbol{0}. \tag{5-22}$$

---

[2] Annihilators are most commonly used to mean the set of all functions satisfying a given set of conditions that are zero on every element of a given set [38].

5-4-5 Defining the desired Hamiltonian for physical systems

As it has been mentioned previously, the use of passivity for physical systems can be more intuitive than signal based controllers. In this section, the full effect of this benefit is observed as the energy function is shaped as demonstrated in [19]. To show this, a simple position tracking controller is first considered and then expanded to position and velocity reference tracking.

In Section 5-4-3, a method has been provided where H_a is found by solving a differential equation. However, [19] provides a method of consistently finding a solution of the PDE without solving it directly and instead, directly shaping the energy of the system. It turns out that this method presents many benefits when performing reference tracking on port-Hamiltonian systems.

The total energy of a physical system $H(\boldsymbol{q}, \boldsymbol{p})$ is described as the sum of its potential and kinetic energy. This can be expressed as:

$$H(\boldsymbol{q}, \boldsymbol{p}) = \frac{1}{2}\dot{\boldsymbol{p}}^\intercal \boldsymbol{M}(\boldsymbol{p})\dot{\boldsymbol{q}} + U(\boldsymbol{q}) \tag{5-23}$$

with $\boldsymbol{M}(\boldsymbol{q}) = \boldsymbol{M}^\intercal(\boldsymbol{q}) > \boldsymbol{0}$, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{k\times k}$ the generalised mass matrix. Furthermore, $U(\boldsymbol{q})$ is a scalar representing the total potential energy of the system and is often expressed as $U(\boldsymbol{q}) = \boldsymbol{K}\boldsymbol{q}$ with $\boldsymbol{K} > \boldsymbol{0}$ and $\boldsymbol{K} \in \mathbb{R}^{1\times k}$. It should also be noticed that $H(\boldsymbol{q}, \boldsymbol{p})$ is bounded from below and above as both the kinetic and potential energies are bounded similarly.

By placing the system at the desired generalised coordinates, it shall have minimum potential energy at that point. This is a transformation of the coordinate system such that the equilibrium is located at $(\boldsymbol{q}^*, \boldsymbol{p} = \boldsymbol{0})$. To do so, H_d for a physical system is simply described as follows.

$$H_d(\boldsymbol{q}, \boldsymbol{p}) = \frac{1}{2}\dot{\boldsymbol{p}}^\intercal \boldsymbol{M}(\boldsymbol{p}) + \boldsymbol{U}(\boldsymbol{q} - \boldsymbol{q}^*) \tag{5-24}$$

Taking H_d and solving Equation (5-20) to determine the control law, $\boldsymbol{\beta}(\boldsymbol{\xi})$, an asymptotically stable controller is found. This controller will stabilise the system around the points $(\boldsymbol{q^*}, \boldsymbol{0})$.

Similarly, in order to add control to the velocities of the system, reference momenta may be used, yielding:

$$H_d(\boldsymbol{q}, \boldsymbol{p}) = \frac{1}{2}\left(\boldsymbol{p} - \boldsymbol{p}^*\right)^\intercal \boldsymbol{M}(\boldsymbol{p} - \boldsymbol{p}^*) + \boldsymbol{U}(\boldsymbol{q} - \boldsymbol{q}^*) \tag{5-25}$$

Furthermore, an important extension of the Hamiltonian from Equation (5-25) should be considered. The *desired* potential energy, $U_d(\boldsymbol{q})$, can in fact be written as $U_d(\boldsymbol{q}) = \boldsymbol{q}^\intercal \boldsymbol{K}\boldsymbol{q}$ where $\boldsymbol{K} > \boldsymbol{0}$ is now a matrix with $\boldsymbol{K} \in \mathbb{R}^{k\times k}$. This type of potential energy function is often used by control engineers for reference tracking of underactuated systems [41].

5-4-6 Improving controller performance

A controller was found in Section 5-4-3. However, at times, the performance may have to be improved. J. Ferguson et al. [42] state that this is often the case with underactuated systems. Particularly, in the event that a degree of freedom is undamped. J. Ferguson et al. suggest that this may render a system unstable due to terms violating the passivity requirements.

To improve the robustness of the system, integral action is added. A method to do so is suggested by Ramirez et al. [40].

The control input $\boldsymbol{u_{in}}$ is altered from $\boldsymbol{u_{in}} = \boldsymbol{\beta}(\boldsymbol{\xi})$ to $\boldsymbol{u_{in}} = \boldsymbol{\beta}(\boldsymbol{\xi}) + \boldsymbol{v}$.

Ramirez et al. [40] state that the integral term $\boldsymbol{v}$ should be expressed as:

$$\dot{\boldsymbol{v}} = -\boldsymbol{K_i}\boldsymbol{g}(\boldsymbol{\xi})\frac{\partial H_d}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}) \tag{5-26}$$

with $\boldsymbol{K_i} \in \mathbb{R}^{n \times n}$ a constant matrix. Stability is said to be preserved given that $\boldsymbol{K_i} = \boldsymbol{K_i}^\mathsf{T} > 0$.

As a result, the new closed loop energy is defined as H_{cl} and is given by:

$$H_{cl} = H_d(\boldsymbol{\xi}) + \frac{1}{2}\boldsymbol{v}^\mathsf{T}\boldsymbol{K_i}^{-1}\boldsymbol{v} \tag{5-27}$$

The integration term is then implemented through the series interconnection of the two systems. Therefore:

$$\begin{bmatrix} \dot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{\mathcal{J}_d} - \boldsymbol{\mathcal{R}_d}) & \boldsymbol{g}\boldsymbol{K_i} \\ -\boldsymbol{K_i}^\mathsf{T}\boldsymbol{g}^\mathsf{T} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \frac{\partial H_d}{\partial \boldsymbol{\xi}} \\ \frac{\partial H_{cl}}{\partial \boldsymbol{v}} \end{bmatrix}$$

Chapter 6

A method for tuning port-Hamiltonian systems

Many prominent researchers in the field of port-Hamiltonian (PH) systems such as Arjan van der Schaft and Romeo Ortega emphasise the importance of PH systems in control theory due to their intuitive nature [4]. This intuition, however, only comes through significant exposure to PH systems.

To overcome this problem and accelerate the process of learning Interconnection and Damping Assignment-Passivity-Based Control (IDA-PBC), an intuitive method is proposed for tuning. Hopefully, this method will minimise time spent tuning and help first-time PBC designers to build their confidence and intuition in their designs.

An explanation behind the tuning method is first presented in Section 6-1. Section 6-2 then explains how to tune the system in order to achieve critical damping.

6-1 The idea behind the tuning method

The purpose of this method is to simplify the tuning process. To do so, the controller is designed so that it behaves similarly to a mass-spring-damper system. The behaviour of these simple mechanical systems is well known and understood by engineers. This will be used to our advantage in the design of a controller.

As seen previously in Section 4-3, the Hamiltonian of a mass-spring-damper system is expressed as:

$$\begin{aligned} H &= K + U \\ H &= \frac{1}{2m}p_x^2 + \frac{1}{2}kq_x^2 \end{aligned} \tag{6-1}$$

To drive the system to its setpoint, the desired Hamiltonian must have its minimum at the reference points. This can be achieved in the following way:

$$H_d = \frac{1}{2m}(p_x - p_x^*)^2 + \frac{1}{2}k(q_x - q_x^*)^2 \tag{6-2}$$

It is now clear that the spring constant k can be altered and the desired Hamiltonian will maintain its minimum at the setpoint (q_x^*, p_x^*). Similarly, a constant can be added to multiply the momentum term. The desired Hamiltonian is therefore re-written as:

$$H_d = \frac{K_1}{2m}(p_x - p_x^*)^2 + \frac{K_2}{2}(q_x - q_x^*)^2 \tag{6-3}$$

The other constants, $1/2$ and m remain to keep the analytical formulation for K_2 clearer to the designer.

For the non-scalar case, Equation (6-3) becomes:

$$H_d = (\boldsymbol{p} - \boldsymbol{p}^*)^\intercal \frac{\boldsymbol{K_1}\boldsymbol{M}^{-1}}{2}(\boldsymbol{p} - \boldsymbol{p}^*) + (\boldsymbol{q_x} - \boldsymbol{q}^*)^\intercal \frac{\boldsymbol{K_2}}{2}(\boldsymbol{q_x} - \boldsymbol{q}^*) \tag{6-4}$$

with $\boldsymbol{M}$ the generalised mass matrix. Given that $\boldsymbol{q}, \boldsymbol{p} \in \mathbb{R}^{n/2}$, then: $\boldsymbol{M}, \boldsymbol{K_1}, \boldsymbol{K_2} \in \mathbb{R}^{n/2 \times n/2}$. To allow for critical damping, the damping ratio ζ must be set to 1. That is, considering the damping constant c_c and the critical damping constant c_{cr}:

$$\zeta = \frac{c_c}{c_{cr}} = 1 = \frac{c_c}{2\sqrt{km}} \tag{6-5}$$

The question is now: *how can critical damping be obtained?*

6-2 Obtaining critical damping

Critical damping is obtained in two simple steps. First, the matrix $\boldsymbol{K_2}$ is designed as explained in Section 6-2-1. Once the $\boldsymbol{K_2}$ matrix is determined, the dissipation matrix can be computed as demonstrated in Section 6-2-2.

6-2-1 Designing the spring constant matrix

The designer is free to choose $\boldsymbol{K_2}$ according to the system requirements at hand. A higher $\boldsymbol{K_2}$ will create a faster controller. However, making $\boldsymbol{K_2}$ too large may result in saturated actuators. Furthermore, a couple of rules about $\boldsymbol{K_2}$ have to be followed.

$\boldsymbol{K_2} \in \mathbb{R}^{n/2 \times n/2}$ is a diagonal matrix such that:

$$\boldsymbol{K_2} = \begin{bmatrix} k_1 & 0 & \cdots & 0 \\ 0 & k_2 & \cdots & 0 \\ \vdots & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & k_{n/2} \end{bmatrix}$$

with k_i, $i = 1, \ldots, n/2$ the spring constants for each degree-of-freedom considered. Since $\boldsymbol{K_2}$ represents spring constants, it is clear that $\boldsymbol{K_2} \geq \boldsymbol{0}$.

6-2-2 Computing the dissipation matrix

Having determined the spring constant, it is now easy to determine the dissipation block $\boldsymbol{\mathcal{R}_{d,4}}$. The analogy we are following to achieve critical damping states that $\boldsymbol{\mathcal{R}_{d,4}} \equiv c_{cr}$. By letting the dissipation block $\boldsymbol{\mathcal{R}_{d,4}}$ be equivalent to the critical damping constant, the damping ratio ζ is driven to 1. As we are working with a non-scalar problem, a more formal proposal is made:

$$\boldsymbol{\mathcal{R}_{d,4}} = 2\sqrt{\boldsymbol{K_2 M}} \tag{6-6}$$

keeping in mind that $\boldsymbol{\mathcal{R}_{d,4}}, \boldsymbol{K_2} \in \mathbb{R}^{n/2 \times n/2}$ are diagonal matrices.

Computing $\boldsymbol{K_1}$ is then straight forward. Figure 6-1 shows a suggested flowchart with the steps the designer should follow to tune the controller.

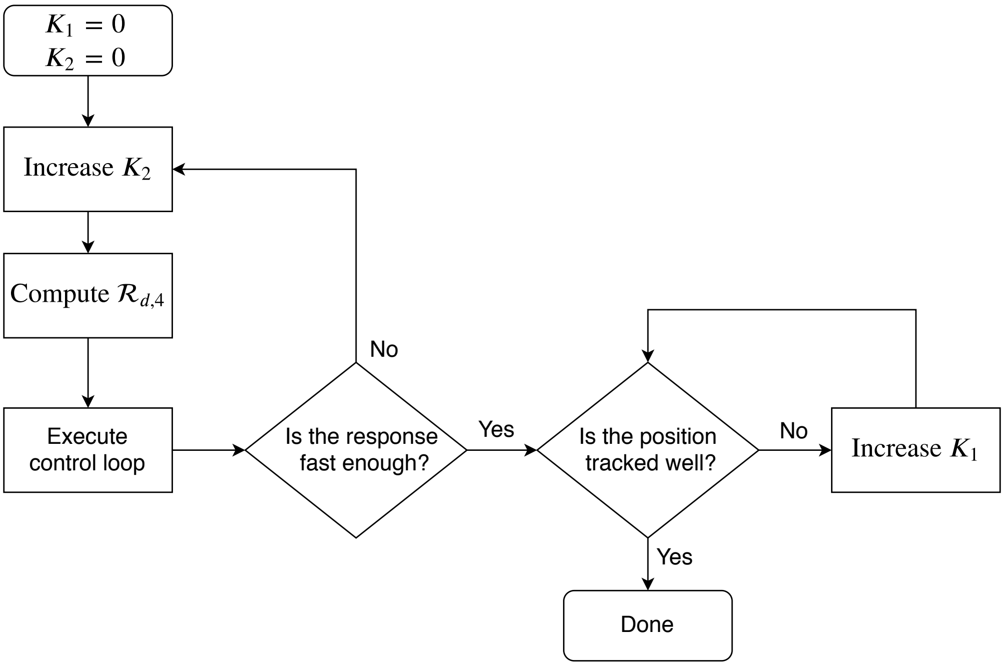


Figure 6-1: Tuning flow chart for the proposed method.

Part III

Passivity-Based Control for Helicopters

Part 3 concludes this thesis by combining all the information from Parts 1 and 2 together to model a helicopter and control the autorotation manoeuvre using port-Hamiltonian methods.

An attempt to convert a Newtonian body model to the port-Hamiltonian framework has been made in Chapter 3. The terms preventing the direct conversion are identified and explained.

The problem is overcome by using a point-mass model for both the 3- and the 6-Degrees-of-Freedom (DoF) cases in Chapters 8 and 9 respectively. The simulation results are provided in Chapter 10 where the autorotation manoeuvre is also attempted in Section 10-4. Finally, the thesis is wrapped up with a conclusion wherein the research questions are answered and recommendations for future work are proposed.

Chapter 7

Attempting to derive the port-Hamiltonian equivalent of the body-model

The work from J. Kerkkamp [43] made an attempt to derive a 3-Degrees-of-Freedom (DoF) model of a helicopter in the port-Hamiltonian (PH) framework directly from the Newtonian flight mechanics. When looking at the thrust of the main rotor, his work suggests to split the thrust into two separate terms:

$$T_m = T_1\theta_0 - T_2\lambda \tag{7-1}$$

with $\lambda = \lambda_i + \lambda_c$. The equations of motion for accelerations from Section 3-6-1 then become:

$$\begin{bmatrix} \dot{u} \\ \\ \dot{w} \\ \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{T_1\theta_0 - T_2\lambda}{m}\sin(\beta_{long}) - g\sin(\theta_f) - \frac{D}{m}\frac{u}{V} - qw \\ \\ -\frac{T_1\theta_0 - T_2\lambda}{m}\cos(\beta_{long}) + g\cos(\theta_f) - \frac{D}{m}\frac{w}{V} + qu \\ \\ -\frac{T_1\theta_0 - T_2\lambda}{I_{yy}}h_m\sin(\beta_{long}) \end{bmatrix}$$

The PH system $\mathcal{H}$ representing the helicopter system is now:

$$\mathcal{H} = \begin{cases} \dot{\boldsymbol{\xi}}(t) = [\boldsymbol{\mathcal{J}}(\boldsymbol{\xi}) - \boldsymbol{\mathcal{R}}(\boldsymbol{\xi})]\,\frac{\partial H(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} + \boldsymbol{g}(\boldsymbol{\xi})\boldsymbol{u_{in}} \\ \boldsymbol{y} = \boldsymbol{g}^\mathsf{T}(\boldsymbol{\xi})\frac{\partial H(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \end{cases}$$

with the state vector defined as:

$$\boldsymbol{\xi} = \begin{bmatrix} q_x & q_z & q_{\theta_f} & q_{\beta_{long}} & | & p_x & p_z & p_{\theta_f} & p_{\beta_{long}} \end{bmatrix}^\mathsf{T}$$

One may notice that the q and p notation is now implemented. The superscript q stands for generalised coordinates whereas p represents the generalised momenta. Furthermore, the rotor tilt angle β_{long} is now also a state.

The equations of motion are now adapted to the port-Hamiltonian notation.

$$\dot{q}_x = \frac{p_x}{m} \tag{7-2a}$$

$$\dot{q}_z = \frac{p_x}{m} \tag{7-2b}$$

$$\dot{q}_{\theta_f} = \frac{p_{\theta_f}}{I_{yy}} \tag{7-2c}$$

$$\dot{q}_{\beta_{long}} = \frac{p_\beta}{I_{rot}} \tag{7-2d}$$

$$\dot{p}_x = -mg\sin(q_{\theta_f}) - D\frac{p_x}{mV} + (T_1\theta_0 - T_2\lambda)\sin(\beta_{long}) - \frac{p_{\theta_f}}{I_{yy}}\frac{p_z}{m} \tag{7-2e}$$

$$\dot{p}_z = mg\cos(q_{\theta_f}) - D\frac{p_z}{mV} - (T_1\theta_0 - T_2\lambda)\cos(\beta_{long}) + \frac{p_{\theta_f}}{I_{yy}}\frac{p_x}{m} \tag{7-2f}$$

$$\dot{p}_{\theta_f} = -(T_1\theta_0 - T_2\lambda)h_m\sin(\beta) \tag{7-2g}$$

$$\dot{p}_{\beta_{long}} = \tau_{\beta_{long}} \tag{7-2h}$$

where I_{rot} is the moment of inertia of the rotor. The thrust tilt angle β_{long} is controlled by the torque $\tau_{\beta_{long}}$. To control the thrust, the collective angle θ_0 is used. The input vector is therefore defined as $\boldsymbol{u_{in}} = [\theta_0 \quad \tau_{\beta_{long}}]^\intercal$.

Furthermore, it is known that the interconnection matrix $\boldsymbol{\mathcal{J}}$ has the property that $\boldsymbol{\mathcal{J}} = -\boldsymbol{\mathcal{J}}^\intercal$. For mechanical systems, $\boldsymbol{\mathcal{J}}$ follows the structure:

$$\boldsymbol{\mathcal{J}} = \begin{bmatrix} \mathbf{0}^{n/2\times n/2} & \boldsymbol{\mathcal{I}}^{n/2\times n/2} \\ -\boldsymbol{\mathcal{I}}^{n/2\times n/2} & \mathbf{0}^{n/2\times n/2} \end{bmatrix}$$

where n is the number of states and $\boldsymbol{\mathcal{I}}$ is the identity matrix. The drag and Coriolis effect terms are all contained within the dissipation matrix $\boldsymbol{\mathcal{R}}(\boldsymbol{\xi})$. Looking at the system $\mathcal{H}$, it can now be said that:

$$\dot{q}_x = \frac{\partial H}{\partial p_x} \tag{7-3a}$$

$$\dot{q}_z = \frac{\partial H}{\partial p_z} \tag{7-3b}$$

$$\dot{q}_{\theta_f} = \frac{\partial H}{\partial p_{\theta_f}} \tag{7-3c}$$

$$\dot{q}_{\beta_{long}} = \frac{\partial H}{\partial p_{\beta_{long}}} \tag{7-3d}$$

$$\dot{p}_x = -\frac{\partial H}{\partial q_x} - \mathcal{R}^{5,:}\frac{\partial H}{\partial \xi} + T_1\sin(\beta_{long})\theta_0 \tag{7-3e}$$

$$\dot{p}_z = -\frac{\partial H}{\partial q_z} - \mathcal{R}^{6,:}\frac{\partial H}{\partial \xi} - T_1 \cos(\beta_{long})\theta_0 \tag{7-3f}$$

$$\dot{p}_{\theta_f} = -\frac{\partial H}{\partial q_{\theta_f}} - T_1 h_m \sin(\beta_{long})\theta_0 \tag{7-3g}$$

$$\dot{p}_{\beta_{long}} = -\frac{\partial H}{\partial q_{\beta_{long}}} + \tau_{\beta_{long}} \tag{7-3h}$$

with $\boldsymbol{\mathcal{R}}^{i,:}$ meaning all the elements of row i in the dissipation matrix $\boldsymbol{\mathcal{R}}$. The red values are part of the $\boldsymbol{g}(\boldsymbol{\xi})$ matrix and the blue variables are part of the input vector $\boldsymbol{u_{in}}$. Consequently, the partial derivatives of the Hamiltonian can directly be computed.

$$\frac{\partial H}{\partial p_x} = \frac{p_x}{m} \tag{7-4a}$$

$$\frac{\partial H}{\partial p_z} = \frac{p_z}{m} \tag{7-4b}$$

$$\frac{\partial H}{\partial p_{\theta_f}} = \frac{p_{\theta_f}}{I_{yy}} \tag{7-4c}$$

$$\frac{\partial H}{\partial p_\beta} = \frac{p_\beta}{I_{rot}} \tag{7-4d}$$

$$-\frac{\partial H}{\partial q_x} = -mg\sin(q_{\theta_f}) - T_2\lambda\sin(q_{\beta_{long}}) \tag{7-4e}$$

$$-\frac{\partial H}{\partial q_z} = mg\cos(q_{\theta_f}) + T_2\lambda\sin(q_{\beta_{long}}) \tag{7-4f}$$

$$-\frac{\partial H}{\partial q_{\theta_f}} = T_2\lambda h_m\sin(q_{\beta_{long}}) \tag{7-4g}$$

$$-\frac{\partial H}{\partial q_\beta} = 0 \tag{7-4h}$$

Two methods can now be used to calculate the Hamiltonian. Either the Hamiltonian is derived from physical knowledge of the system, or it is derived by integrating the partial derivatives from Equations (7-4). For now, the partial derivatives will be integrated and their physical meaning will then be checked.

To begin with, the kinetic energy is derived by integrating the momenta terms with respect to their corresponding momentum variable.

$$H(\boldsymbol{p}) = K = \int \frac{\partial H}{\partial \boldsymbol{p}} d\boldsymbol{p} = \int \frac{\partial H}{\partial p_x} dp_x + \int \frac{\partial H}{\partial p_z} dp_z + \int \frac{\partial H}{\partial p_{\theta_f}} dp_{\theta_f} + \int \frac{\partial H}{\partial p_{\beta_{long}}} dp_{\beta_{long}} + \kappa_1 \tag{7-5}$$

with κ_1 as the integration constant. As a result:

$$
\begin{aligned}
K &= \frac{1}{2}\mathbf{p}^{\intercal}\mathbf{M}^{-1}\mathbf{p} + \kappa \\
&= \frac{1}{2m}p_x + \frac{1}{2m}p_z + \frac{1}{2I_{yy}}p_{\theta_f} + \frac{1}{2I_{rot}}p_{\beta} + \kappa
\end{aligned}
\tag{7-6}
$$

Using the same method for the potential energy of the system:

$$
\begin{aligned}
U =& -q_x \underbrace{\left(-mg\sin(q_{\theta_f}) - T_2\lambda\sin(q_{\beta_{long}})\right)}_{-\frac{\partial H}{\partial q_x}} - q_z \underbrace{\left(mg\cos(q_{\theta_f}) + T_2\lambda\cos(q_{\beta_{long}})\right)}_{-\frac{\partial H}{\partial q_z}} \dots \\
& - q_{\theta_f}\underbrace{T_2\lambda h_m\sin(q_{\beta_{long}})}_{-\frac{\partial H}{\partial q_{\theta_f}}} + \underbrace{0}_{-\frac{\partial H}{\partial q_{\beta_{long}}}} + \underbrace{\kappa_2}_{\text{integration constant}}
\end{aligned}
\tag{7-7}
$$

It can be noticed that by taking the partial derivative of U with respect to the generalised coordinates, Equations (7-4e) to (7-4h) are not obtained again. Instead, some additional terms are found:

$$
\begin{bmatrix}
\frac{\partial H}{\partial q_x} \\[2ex]
\frac{\partial H}{\partial q_z} \\[2ex]
\frac{\partial H}{\partial q_{\theta_f}} \\[2ex]
\frac{\partial H}{\partial q_{\beta_{long}}}
\end{bmatrix}
=
\begin{bmatrix}
mg\sin(q_{\theta_f}) + T_2\lambda\sin(q_{\beta_{long}}) \\[2ex]
-mg\cos(q_{\theta_f}) - T_2\lambda\cos(q_{\beta_{long}}) \\[2ex]
T_2\lambda h_m\sin(q_{\beta_{long}}) + q_x mg\cos(q_{\theta_f}) + q_z mg\sin(q_{\theta_f}) \\[2ex]
q_x T_2\lambda\cos(q_{\beta_{long}}) + q_z T_2\lambda\sin(q_{\beta_{long}}) - q_{\theta_f} T_2\lambda\cos(q_{\beta_{long}})
\end{bmatrix}
$$

with the additional terms in written in red. These extra terms mean that:

$$
\frac{\partial}{\partial \boldsymbol{q}}\left(\int \frac{\partial H}{\partial \boldsymbol{q}} d\boldsymbol{q}\right) \neq \frac{\partial H}{\partial \boldsymbol{q}}
\tag{7-8}
$$

It appears that the extra terms originate from the fact that the terms with T_2 add *wrong* potential energies. When an object moves up and down in space, it converts potential energy into kinetic energy and vice versa. Assuming the object is moving horizontally in space at a fixed altitude, the object is no longer trading potential energy into kinetic energy. Work is done by the object as a result of its movement.

As it can be seen, the potential energy now has the terms $q_x T_2\lambda\sin(q_{\beta_{long}})$ and $q_z T_2\lambda\cos(q_{\beta_{long}})$. However, how do q_x and q_z with the thrust component T_2 combine to describe a potential energy? They are in fact describing work done rather than a potential energy. Similarly, this can be done for the 6-DoF model. The interested reader is referred to Appendix B to see the additional terms obtained in the 6-DoF model.

The work done by the thrust T_2 results in a non-passive system and prevents the use of port-Hamiltonian methods.

For the above mentioned reasons, it is believed that it is **not** possible to develop a PH model directly from the equations of motion of a body model. The splitting of the thrust into T_1 and T_2 is necessary to allow the pilot inputs to be determined by the controller ($\beta_{long} = a_1 - \theta_{long}$). The non-passivity of the T_2 terms prevent the possibility of creating a stable controller within the port-Hamiltonian framework. Removing the T_2 terms from the Hamiltonian results in a point-mass model.

Moreover, it can be seen that, in general, the system does not respond well to rotations. These create undesirable terms such as the gravitational ones. A solution to this is to write the equations of motion in the Earth reference frame. This would remove terms such as $q_x mgcos(\theta_f)$, which is not a potential energy and rather work carried out by the body.

Chapter 8

3-degrees-of-freedom point-mass

In this chapter, the port-Hamiltonian (PH) control of a 3-Degrees-of-Freedom (DoF) helicopter model is covered. First, Section 8-1 expresses the point-mass for the helicopter in the PH framework. The Passivity-Based Control (PBC) of the mass is treated in Section 8-2-1. Finally, the pilot inputs are calculated in Section 8-3.

8-1 From Newtonian to port-Hamiltonian dynamics

First, the state vector for the 3-DoF Point-Mass (PM) model is defined through the use of the generalised coordinates and momenta.

$$\boldsymbol{\xi} = \begin{bmatrix} q_x & q_z & q_{\theta_f} & | & p_x & p_z & p_{\theta_f} \end{bmatrix}^\intercal$$

The number of states is 6, therefore: $n = 6$. Additionally, the control inputs are defined.

$$\boldsymbol{u_{in}} = \begin{bmatrix} T_{m,x} & T_{m,z} \end{bmatrix}^\intercal$$

It was previously seen in Chapter 7 that rotations and gravity can generate unwanted terms in the partial derivatives of the Hamiltonian. As a result, the PH model is expressed in the Earth reference frame. This can be seen in the expression of the pitch rate p_{θ_f} as the tilt angle β_{long} is no longer a variable. Instead, the inputs $T_{m,x}$ and $T_{m,z}$ generate the torque that pitches the helicopter.

The equations of motion for the point-mass are now expressed for the state vector $\boldsymbol{\xi}$ in the PH framework and compared to the body model in the Newtonian framework. The two systems are equivalent.

Body model

$$^b\dot{u} = \frac{T_m}{m}\sin\left(\beta_{long}\right) - \frac{D}{m}\frac{^bu}{V} - g\sin({}^b\theta_f) - {}^bq\,{}^bw$$

$$^b\dot{w} = \frac{T_m}{m}\cos(\beta_{long}) - \frac{D}{m}\frac{^bw}{V} + g\cos({}^b\theta_f) + {}^bq\,{}^bu$$

$$^b\dot{q} = -\frac{T_m}{I_{yy}}h_m\sin\left(\beta_{long}\right)$$

$$^b\dot{x} = {}^bu$$

$$^b\dot{z} = {}^bw$$

$$^b\dot{\theta}_f = {}^bq$$

Point-mass model

$$\dot{p}_x = T_{m,x} - \frac{D}{m}\frac{p_x}{V} - \frac{p_{\theta_f}}{I_{yy}}\frac{p_z}{m}$$

$$\dot{p}_z = -T_{m,z} + mg - \frac{D}{m}\frac{p_z}{V} + \frac{p_{\theta_f}}{I_{yy}}\frac{p_x}{m}$$

$$\dot{p}_{\theta_f} = -T_{m,x}h_m\cos(q_{\theta_f}) + T_{m,z}h_m\sin(q_{\theta_f})$$

$$\dot{q}_x = \frac{p_x}{m}$$

$$\dot{q}_z = \frac{p_z}{m}$$

$$\dot{q}_{\theta_f} = \frac{p_{\theta_f}}{I_{yy}}$$

By again defining the interconnection matrix $\boldsymbol{\mathcal{J}}$ for mechanical systems as

$$\boldsymbol{\mathcal{J}} = \begin{bmatrix} \mathbf{0}^{n/2\times n/2} & \boldsymbol{\mathcal{I}}^{n/2\times n/2} \\ -\boldsymbol{\mathcal{I}}^{n/2\times n/2} & \mathbf{0}^{n/2\times n/2} \end{bmatrix}$$

the same method of deriving the partial derivatives of the Hamiltonian can be carried out as in Chapter 7. However, to build up intuition, the Hamiltonian will first be derived from our knowledge of the system.

To begin with, the kinetic energy of the system is considered.

Considering a car of mass m moving in a straight line at velocity v, its kinetic energy is simply expressed as $K = \frac{1}{2}mv^2$. For a mass in pure rotation about its $x-$axis with angular velocity ω, the kinetic energy would become $K = \frac{1}{2}I_{xx}\omega^2$ with I_{xx} the mass moment of inertia about the $x-$axis.

If an object has linear and rotational velocity components about multiple axes, the total kinetic energy of the object is simply the sum of the kinetic energies about each axis. This is why in port-Hamiltonian form, the kinetic energy is expressed as:

$$K = \frac{1}{2}\boldsymbol{p}^\intercal\boldsymbol{M}^{-1}\boldsymbol{p} \tag{8-3}$$

with $\boldsymbol{p} = [p_x \quad p_z \quad p_{\theta_f}]^\intercal$, the momentum vector and $\boldsymbol{M} = \mathrm{diag}(m, m, I_{yy})$, the generalised mass matrix. Therefore:

$$K = \underbrace{\begin{bmatrix} p_x & p_z & p_{\theta_f} \end{bmatrix}}_{\boldsymbol{p}^\intercal} \underbrace{\begin{bmatrix} 1/m & 0 & 0 \\ 0 & 1/m & 0 \\ 0 & 0 & 1/I_{yy} \end{bmatrix}}_{\boldsymbol{M}^{-1}} \underbrace{\begin{bmatrix} p_x \\ p_z \\ p_{\theta_f} \end{bmatrix}}_{\boldsymbol{p}} = \frac{1}{2m}p_x^2 + \frac{1}{2m}p_z^2 + \frac{1}{2I_{yy}}p_{\theta_f}^2$$

For the potential energy, now that the Earth reference frame is being used, the only term to be considered is a result of the altitude of the rotorcraft. Therefore:

$$U = -mgq_z \tag{8-4}$$

Note that a negative sign is added due to the fact that the axis in the reference frame points towards the ground. Consequently, a gain in potential energy occurs when q_z decreases. The Hamiltonian is now expressed as:

$$H(\boldsymbol{\xi}) = \frac{1}{2m}p_x^2 + \frac{1}{2m}p_z^2 + \frac{1}{2I_{yy}}p_{\theta_f}^2 - mgq_z \tag{8-5}$$

Taking the partial derivatives of the Hamiltonian with respect to its states yields the following:

Generalised coordinates

$$\frac{\partial H(\boldsymbol{\xi})}{\partial q_x} = 0 \tag{8-6a}$$

$$\frac{\partial H(\boldsymbol{\xi})}{\partial q_z} = -mg \tag{8-6b}$$

$$\frac{\partial H(\boldsymbol{\xi})}{\partial q_{\theta_f}} = 0 \tag{8-6c}$$

Generalised momenta

$$\frac{\partial H(\boldsymbol{\xi})}{\partial p_x} = \frac{p_x}{m} \tag{8-7a}$$

$$\frac{\partial H(\boldsymbol{\xi})}{\partial p_z} = \frac{p_z}{m} \tag{8-7b}$$

$$\frac{\partial H(\boldsymbol{\xi})}{\partial p_{\theta_f}} = \frac{p_{\theta_f}}{I_{yy}} \tag{8-7c}$$

The dissipation matrix $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{n \times n}$ now has to be defined. To begin with, $\boldsymbol{\mathcal{R}}$ is split into four blocks of size $\mathbb{R}^{n/2 \times n/2}$:

$$\boldsymbol{\mathcal{R}} = \left[\begin{array}{c|c} \boldsymbol{\mathcal{R}_1} & \boldsymbol{\mathcal{R}_2} \\ \hline \boldsymbol{\mathcal{R}_3} & \boldsymbol{\mathcal{R}_4} \end{array}\right]$$

Since dissipation is a result of *forces* expressed as a function of $\frac{\partial H}{\partial \boldsymbol{p}}$, $\boldsymbol{\mathcal{R}_1} = \boldsymbol{\mathcal{R}_2} = \boldsymbol{\mathcal{R}_3} = \mathbf{0}$.

$\boldsymbol{\mathcal{R}_4}$ is now split into two components: dissipative forces due to drag $\boldsymbol{\mathcal{R}_{4,D}}$, and Coriolis forces $\boldsymbol{\mathcal{R}_{4,C}}$ such that $\boldsymbol{\mathcal{R}_4} = \boldsymbol{\mathcal{R}_{4,D}} + \boldsymbol{\mathcal{R}_{4,C}}$.

$$\boldsymbol{\mathcal{R}_{4,D}} = \begin{bmatrix} \frac{D}{V} & 0 & 0 \\ 0 & \frac{D}{V} & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{\mathcal{R}_{4,C}} = \begin{bmatrix} 0 & \frac{p_{\theta_f}}{I_{yy}} & 0 \\ -\frac{p_{\theta_f}}{I_{yy}} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

We describe the system dynamics in the port-Hamiltonian framework so that

$\dot{\boldsymbol{\xi}}(t) = [\mathcal{J}(\boldsymbol{\xi}) - \mathcal{R}(\boldsymbol{\xi})] \frac{\partial H(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} + \boldsymbol{g}(\boldsymbol{\xi})\boldsymbol{u_{in}}$:

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} \dot{q}_x \\ \dot{q}_z \\ \dot{q}_{\theta_f} \\ \dot{p}_x \\ \dot{p}_z \\ \dot{p}_{\theta_f} \end{bmatrix} = \left(\underbrace{\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hline -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{array}\right]}_{\boldsymbol{\mathcal{J}}} - \underbrace{\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \frac{D}{V} & \frac{p_{\theta_f}}{I_{yy}} & 0 \\ 0 & 0 & 0 & -\frac{p_{\theta_f}}{I_{yy}} & \frac{D}{V} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]}_{\boldsymbol{\mathcal{R}}} \right) \underbrace{\begin{bmatrix} 0 \\ -mg \\ 0 \\ \frac{p_x}{m} \\ \frac{p_z}{m} \\ \frac{p_{\theta_f}}{I_{yy}} \end{bmatrix}}_{\frac{\partial H(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}}$$

$$+ \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ h_m \cos(q_{\theta_f}) & h_m \sin(q_{\theta_f}) \end{bmatrix}}_{\boldsymbol{g}} \underbrace{\begin{bmatrix} T_{m,x} \\ T_{m,z} \end{bmatrix}}_{\boldsymbol{u_{in}}}$$

Simulating the autonomous system with $\boldsymbol{g} = \boldsymbol{0}$, the point-mass model is checked. The system is given initial altitude q_z and upwards velocity. Additionally, the mass is given forward velocity along with angular rotation.

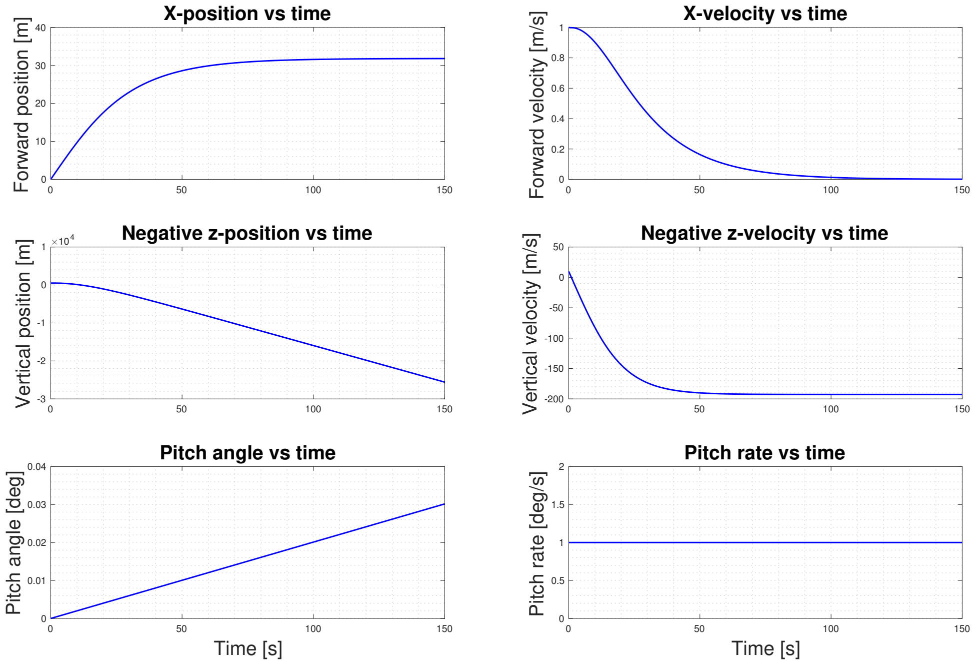


Figure 8-1: Time simulation of a 3-degree-of-freedom falling mass given linear and angular initial velocities.

Looking at Figure 8-1, as expected, the forward velocity decays due to drag until it becomes zero. The vertical velocity can be seen to be increasing. This is due to gravity pulling the mass down. This is the case until terminal velocity is achieved as the drag and gravitational forces balance out. The mass then has a constant downwards velocity. Finally, the pitch rate is assumed to have no damping torques acting on it. Therefore, the pitch angle increases linearly at a constant rate.

8-2 Control of the point-mass

The control law from Equation (5-20) is used. As a reminder, the control law states that the control input $\boldsymbol{u_{in}}$ is:

$$\boldsymbol{u_{in}} = \boldsymbol{g}^{\dagger} \left(\left(\boldsymbol{\mathcal{J}_d} - \boldsymbol{\mathcal{R}_d} \right) \frac{\partial H_d}{\partial \boldsymbol{\xi}} - \left(\boldsymbol{\mathcal{J}} - \boldsymbol{\mathcal{R}} \right) \frac{\partial H}{\partial \boldsymbol{\xi}} \right)$$

with $\boldsymbol{g}^{\dagger} = (\boldsymbol{g}^{\intercal}\boldsymbol{g})^{-1}\boldsymbol{g}^{\intercal}$. From now-on, $\boldsymbol{\mathcal{Q}_d} = \boldsymbol{\mathcal{J}_d} - \boldsymbol{\mathcal{R}_d}$ and $\boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{J}} - \boldsymbol{\mathcal{R}}$.

To define the desired Hamiltonian, H_d, the setpoint $\boldsymbol{\xi}^*$ should achieve a state of zero-energy. For that reason, the desired Hamiltonian is expressed as:

$$H_d = \frac{1}{2} \left(\boldsymbol{p}^{\intercal} - \boldsymbol{p}^{*\intercal} \right) \boldsymbol{K_1} \left(\boldsymbol{p} - \boldsymbol{p}^* \right) + \frac{1}{2} \left(\boldsymbol{q}^{\intercal} - \boldsymbol{q}^{*\intercal} \right) \boldsymbol{K_2} \left(\boldsymbol{q} - \boldsymbol{q}^* \right) \tag{8-8}$$

with $\boldsymbol{K_1}, \boldsymbol{K_2} \in \mathbb{R}^{n/2 \times n/2}$. The matrices are diagonal such that: $\boldsymbol{K_1} = \mathrm{diag}(k_{q_x}, k_{q_z}, k_{q_{\theta_f}})$ and $\boldsymbol{K_2} = \mathrm{diag}(k_{p_x}, k_{p_z}, k_{p_{\theta_f}})$ where the diagonal elements are constants used to tune the controller. Defining the error signal by the subscript e such that $\boldsymbol{q_e} = \boldsymbol{q} - \boldsymbol{q^*}$ and $\boldsymbol{p_e} = \boldsymbol{p} - \boldsymbol{p^*}$, the analytical solution for the control input can be found.

Assuming $\boldsymbol{\mathcal{R}_{d,4}}$ to be diagonal let us now consider the standard matrix element notation. That is, $\boldsymbol{\mathcal{R}}_{\boldsymbol{d,4}}^{ij}$ is the element located in row i and column j of the matrix $\boldsymbol{\mathcal{R}_{d,4}}$. Therefore, $\boldsymbol{\mathcal{R}_{d,4}} = \mathrm{diag}(\boldsymbol{\mathcal{R}}_{\boldsymbol{d,4}}^{11}, \boldsymbol{\mathcal{R}}_{\boldsymbol{d,4}}^{22}, \boldsymbol{\mathcal{R}}_{\boldsymbol{d,4}}^{33})$.

The control input $\boldsymbol{u_{in}}$ can now be computed; its first element is given by:

$$\begin{aligned} \boldsymbol{u_{in}}(1) = & \frac{\left(1 + h_m s(\theta_f)\right) k_{q_x} \frac{p_{x_e}}{m}}{h_m^4 c^2(q_{\theta_f}) s^2(q_{\theta_f})} \\ & + \frac{h_m^2 s(\theta_f) c(\theta_f) \left(-q_{x_e} + \frac{D}{V} \frac{p_x}{m} - \boldsymbol{\mathcal{R}}_{\boldsymbol{d,4}}^{11} \frac{p_{x_e}}{m} \right)}{h_m^4 c^2(q_{\theta_f}) s^2(q_{\theta_f})} + \ldots \end{aligned} \tag{8-9}$$

with $s(\theta_f) = \sin(q_{\theta_f})$ and $c(\theta_f) = \cos(q_{\theta_f})$. It can be seen that the analytical solution becomes quite complicated. An important insight it can bring is that upon simplification by letting $\boldsymbol{q_e} = \boldsymbol{0}$ and $\boldsymbol{p_e} = \boldsymbol{0}$, it can be seen that damped DoF's will have the input equal to the dissipation forces and undamped DoF's will have 0 input.

Equation (8-9) does not simplify the tuning method. Therefore, let us consider the method of trial and error.

8-2-1 Tuning through trial and error

First of all, it should be noted that trial and error does not mean random. When using this method, it is important to understand how a change in the controller affects the system's time response. This understanding allows to fine-tune a controller and to accelerate the process.

The goal of the controller will be to hold a steady forward velocity and altitude. Ideally, the controller converges to the setpoints quickly with as little oscillations and overshoot as possible.

To carry out the tuning, each degree of freedom is first tuned separately. To begin with, the forward velocity is considered. To do so, $\boldsymbol{\mathcal{R}}_{d,4}^{11}$ is set to 500, however, this value could be set to any positive real number. This is done to generate some damping and hopefully obtain an asymptotically stable controller.

Next, the gain k_{p_x} is altered - k_{q_x} is not of interest since the velocity is to be controlled and not the position. Simulations are therefore run with $k_{p_x} = 1, 100$ and 500 to begin with. The rest of the gains are set to 0.

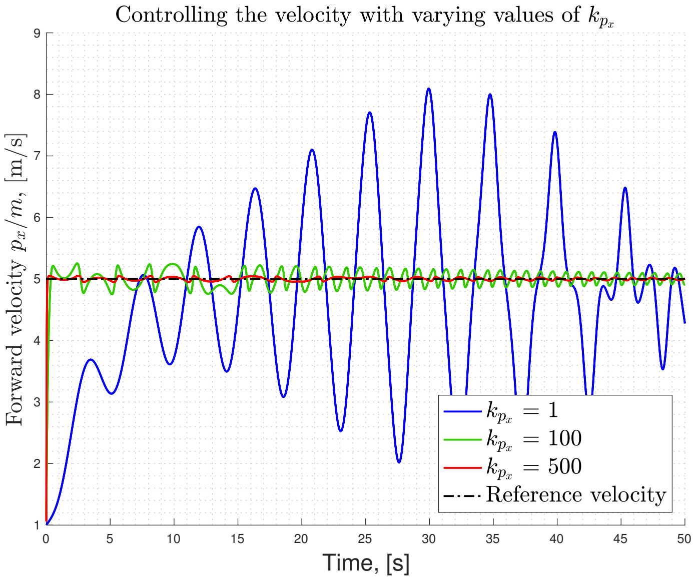


Figure 8-2: Control of point-mass forward velocity with varying k_{p_x} and $\mathcal{R}_{d,4} = \text{diag}(500, 0, 0)$.

Figure 8-2 shows that the performance of the controller is poor, particularly with $k_{p_x} = 1$. Looking at $k_{p_x} = 100$ and 500, the performance appears to be better despite the many oscillations. A common belief might be that k_{p_x} should directly be tuned better along with $\boldsymbol{\mathcal{R}}_{d,4}^{11}$. However, when looking at the pitch rate, it is quickly seen that the behaviour of the mass is not realistic and helicopter-like.

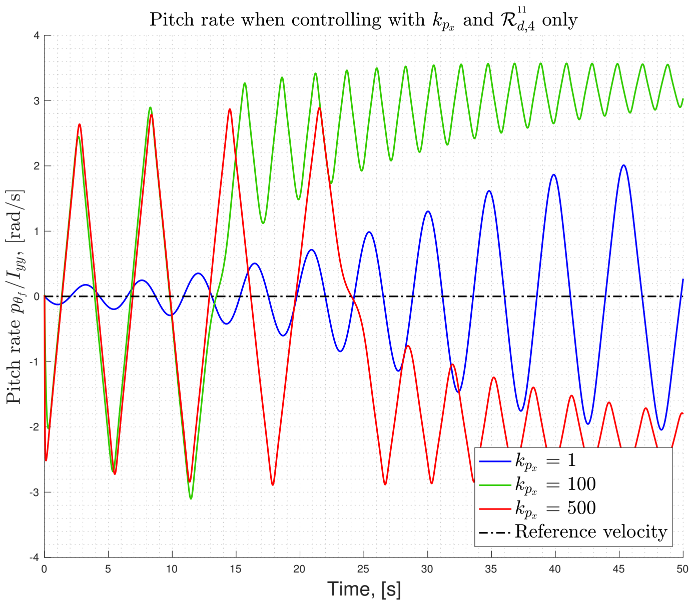


Figure 8-3: Pitch rate after control of point-mass with varying k_{p_x} and $\mathcal{R}_{d,4} = \text{diag}(500, 0, 0)$.

Figure 8-3 shows that perhaps, a more suitable approach is to allow some control on the pitch axis to try and limit the pitch rate. This is done by setting $k_{p_{\theta_f}}$ and $\boldsymbol{\mathcal{R}}^{33}_{\boldsymbol{d,4}}$ to positive real values. Indeed, by setting $k_{p_{\theta_f}} = 50$ and $\boldsymbol{\mathcal{R}}^{33}_{\boldsymbol{d,4}} = 500$, significant improvements can be observed - at least for the case $k_{p_x} = 1$.

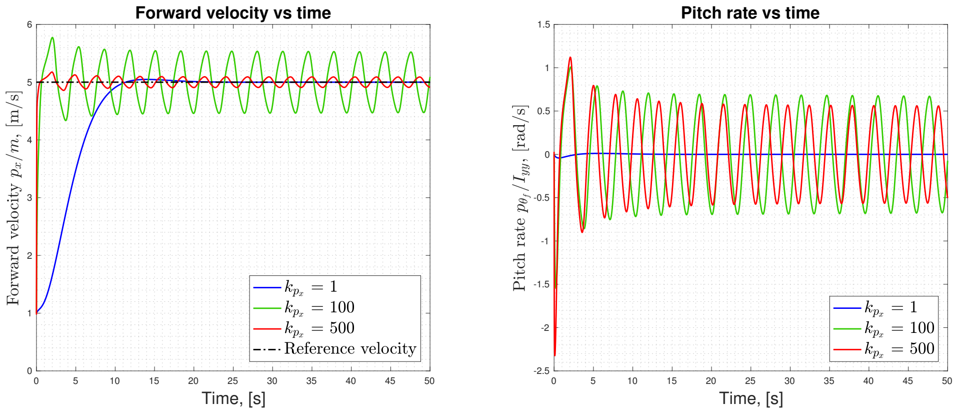


Figure 8-4: Pitch rate after control of point-mass with varying k_{p_x}, $k_{p_{\theta_f}} = 50$ and $\mathcal{R}_{d,4} = \text{diag}(500, 0, 500)$.

Figure 8-4 shows that the system with $k_{p_x} = 1$ has minimal oscillations. In-fact, the system can be seen to be slightly over-damped. Considering these gains, the altitude tracking is now to be enforced.

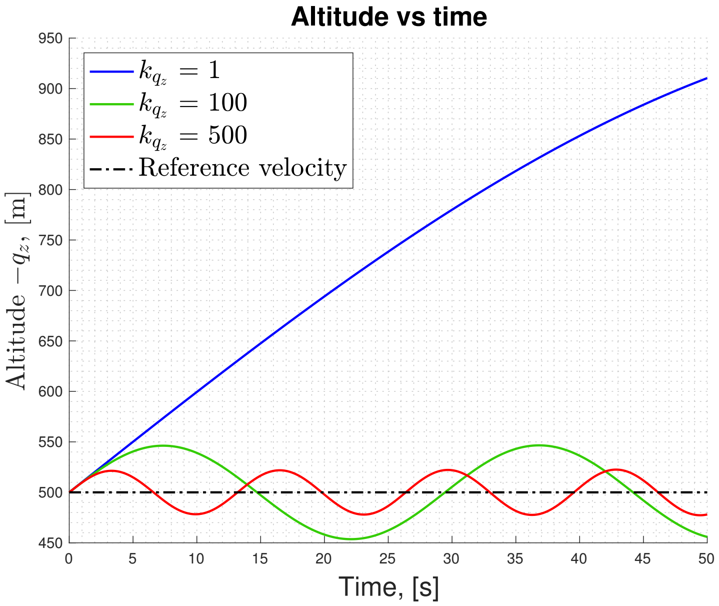


Figure 8-5: Altitude control with k_{q_z}, $k_{p_x} = 1, k_{p_{\theta_f}} = 50, \boldsymbol{\mathcal{R}}^{11}_{\boldsymbol{d,4}} - \boldsymbol{\mathcal{R}}^{22}_{\boldsymbol{d,4}} = \boldsymbol{\mathcal{R}}^{33}_{\boldsymbol{d,4}} = 500$.

Significant oscillations can be observed in Figure 8-5, which suggest an increase in $\boldsymbol{\mathcal{R}}^{22}_{\boldsymbol{d,4}}$ is necessary to improve the damping of the system.

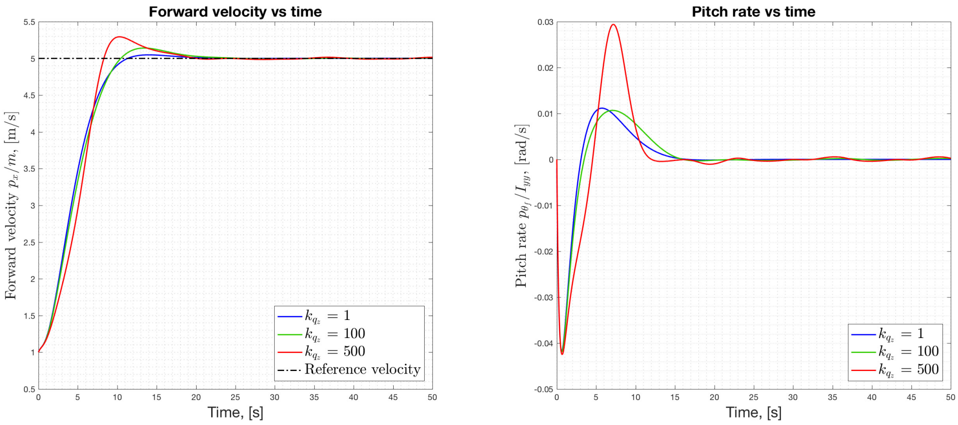


Figure 8-6: Forward velocity and pitch rates with varying k_{q_z}. $k_{p_x} = 1$, $k_{p_{\theta_f}} = 50$, $\boldsymbol{\mathcal{R}}^{11}_{\boldsymbol{d,4}} = \boldsymbol{\mathcal{R}}^{22}_{\boldsymbol{d,4}} = \boldsymbol{\mathcal{R}}^{33}_{\boldsymbol{d,4}} = 500$.

Looking at Figure 8-6 suggests $k_{q_z} = 1$ provides the best performance in terms of settling time when considering pitch rate and forward velocity. However, Figure 8-5 suggests otherwise as $k_{q_z} = 1$ generates the highest oscillation.

By carrying on with the iteration process, an adequate controller can be found. The final results can be seen in Subsection 8-2-2

8-2-2 Tuning using the proposed method from Chapter 6

By using the tuning method from Chapter 6, a working controller is found much quicker. The values of the controller adapt to the setpoint as the controller is a function of $\boldsymbol{\xi}^*$. Considering the same reference signals as in Section 8-2-1, the two controllers are compared. The trial and error control depicted by red lines will be referred to as controller 1 whereas the one tuned with the proposed method from Chapter 6 is named controller 2.

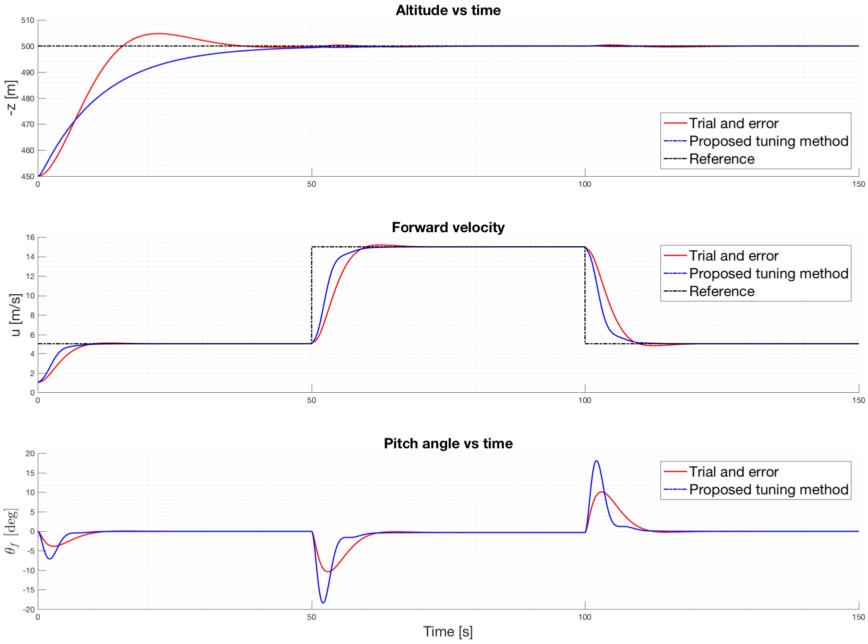


Figure 8-7: Performance comparison of the controllers tuned by trial and error and through the proposed method from Chapter 6.

The two controllers having the following properties:

	Controller	
Controller parameters	*1 - Trial and error*	*2 - Proposed method*
k_{q_x}	0	10
k_{q_z}	70	400
k_{q_t}	1	10
k_{p_x}	1	5
k_{p_z}	5	5
k_{p_t}	50	50
$\boldsymbol{\mathcal{R}}_{\boldsymbol{d}}^{11}$	543	148
$\boldsymbol{\mathcal{R}}_{\boldsymbol{d}}^{22}$	94	938
$\boldsymbol{\mathcal{R}}_{\boldsymbol{d}}^{33}$	500	223

Table 8-1: Comparison of the two controllers.

Figure 8-7 demonstrates that both controllers work well and have little oscillations. It is seen that controller 1 has some overshoot when trying to control the altitude and the forward velocity. Some slight oscillations are then also noticed. Furthermore, it does appear that controller 2 has a faster settling time than its counterpart with no oscillations - critical damping is achieved.

The most obvious differences in control performance are found when looking at the pitch angle. Controller 1 results in pitch angles of smaller magnitude than controller 2. However, the duration for which the helicopter is pitching is longer. The cause for this is the faster acceleration in u that is observed and indicated by a shorter rise-time. Without incorporating the vehicles' operating envelope directly into the model, the flowchart from Figure 6-1 was used to ensure the rotorcraft would remain within its operating range.

Overall, it appears that controller 2 performs better than controller 1. This shows some promise for the method developed in Chapter 6 as it allowed for a better performing controller to be developed in a shorter amount of time. A formal study assessing the method is suggested in order to understand its advantages and limitations.

8-3 Implementing helicopter dynamics

Having designed the controller, it is time to implement the helicopter dynamics onto the point-mass model. To do so, Algorithm 1 from Section 3-7 is adapted to 3-DoF.

Algorithm 2 3-DoF point-mass to pilot inputs.

```
$\boldsymbol{\xi}, \boldsymbol{\xi}^* \leftarrow$ Initialise the states and define setpoint
for $time = t_0, t_0 + dt, \dots, t_{end}$ do
    $\boldsymbol{\xi_e} \leftarrow \boldsymbol{\xi} - \boldsymbol{\xi}^*$ Calculate error signal
    $\boldsymbol{F_{in}}, \boldsymbol{\tau_{in}} \leftarrow \boldsymbol{u_{in}}(\boldsymbol{\xi_e})$ Calculate input forces and torques
    Compute flight parameters: $\lambda_i, \lambda_c, \mu, \alpha_c, V$.

    $T_m \leftarrow \sqrt{\boldsymbol{u}_{\boldsymbol{in}}^{\intercal} \boldsymbol{u_{in}}} = \sqrt{f_{x_m}^2 + f_{z_m}^2}$ Compute the thrust from the inputs
    $\beta_{long} \leftarrow f_{x_m} = T_m \sin(\beta_{long})$, Determine the rotor tilt angle

    $\theta_0 \leftarrow$ from Equation (3-8)
    $a_1 \leftarrow$ from Equation (3-14)
    $\theta_{long} \leftarrow \beta_{long} - a_1$

    $\boldsymbol{\xi}(t + dt) \leftarrow \boldsymbol{\xi} + \dot{\boldsymbol{\xi}} dt$
end for
```

The non-dimensional induced velocity λ_i of Algorithm 2 is calculated with the Glauert theory introduced in Section 3-2-3.

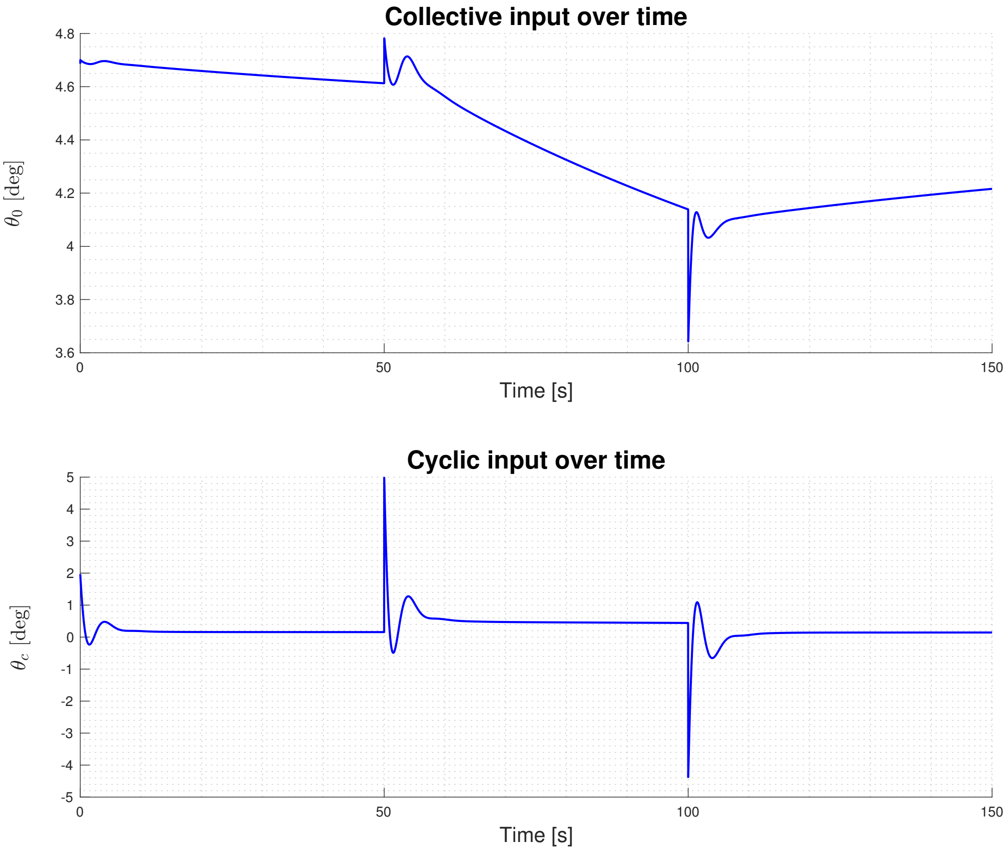


Figure 8-8: Pilot inputs required to perform manoeuvre form Figure 8-7 using the proposed tuning method.

Figure 8-8 demonstrates the inputs required by the pilot to perform the manoeuvre previously seen in Figure 8-7 using the controller method developed from Chapter 6.

As expected, to increase the forward velocity, the pilot pushes on the longitudinal cyclic and pulls the collective to maintain the altitude. Similarly, the pilot will pull back the cyclic to increase the pitch angle of the helicopter and slow down. When doing so, less collective is required.

Chapter 9

6-degrees-of-freedom point-mass

The natural progression of the 3-Degrees-of-Freedom (DoF) point-mass model is to expand it to include lateral motion. The new system then becomes a 6-DoF model. The port-Hamiltonian (PH) system is first derived in Section 9-1. Control is then applied in Section 9-2.

9-1 From Newtonian to port-Hamiltonian dynamics

Having 6-DoF, the state vector doubles in size to let $n = 12$ and $\boldsymbol{\xi} \in \mathbb{R}^n$ with $\boldsymbol{\xi}$ expressed as:

$$\boldsymbol{\xi} = \begin{bmatrix} [q_x & q_y & q_z & q_{\phi_f} & q_{\theta_f} & q_{\psi_f}]^\mathsf{T} \\ [p_x & p_y & p_z & p_{\phi_f} & p_{\theta_f} & p_{\psi_f}]^\mathsf{T} \end{bmatrix}^\mathsf{T}$$

with the new coordinates and momenta subscripts y, ϕ_f and ψ_f representing linear motion into the $y-$axis, roll and yaw rotations respectively. The control inputs have now also been extended to include a force from the main rotor into the $y-$axis as well as the force exerted by the tail into the same axis. Additionally, with the introduction of yaw motion, the torque Q_m generated by the rotor is also included as an input. Thus, the input vector becomes:

$$\boldsymbol{u_{in}} = \begin{bmatrix} T_{m,x} & T_{m,y} & T_{m,z} & T_t & Q_m \end{bmatrix}^\mathsf{T}$$

Instead of using the Blade Element Theory (BET) to calculate the torque, a simplified approach from U. Hald et al. [44] is used. The method expresses Q_m as:

$$Q_m = -A_{Q_m} T_m^{1.5} + B_{Q_m} \tag{9-1}$$

with A_{Q_m} a coefficient that links the thrust to of the main rotor to its drag and B_{Q_m} is the drag of the main rotor at 0° blade pitch. This method is preferred to using BET as it simplifies the task of calculating the pilot inputs. Consequently, when calculating $\boldsymbol{u_{in}}$, constraints have to be set to ensure the condition from Equation 9-1 is held.

The body model is again compared to the point-mass model. In Chapter 8, it was mentioned that the dissipation matrix includes all the drag forces as well as the Coriolis forces. For the 6-DoF model, the torques generated by the Coriolis effect are also considered.

The point-mass model is again described in the Earth reference frame, thereby, eliminating gravity terms in the $x-$ and $y-$axes. Furthermore, another simplification has been made for the torque. It is assumed that its components in the $xx-$ and $yy-$axes are negligible. This assumption is necessary in order to later be able to solve the helicopter dynamics. Otherwise, the system has too many unknowns. The torque of the rotor principally generates yaw rotation. The momenta generated in the other two axes are significantly smaller and therefore neglected.

The following page compares the PH formulation of the equations of motion compared with their Newtonian counterparts. Again, the red terms represent dissipative forces/torques and the blue terms describe control inputs.

Body model	**Point-mass model**
${}^b\dot{u} = \frac{T_m}{m} s\left(\beta_{long}\right) - \frac{D}{m}\frac{{}^bu}{V} - gs({}^b\theta_f) - {}^bq{}^bw + {}^br{}^bv$	$\dot{p}_x = T_{m,x} - \frac{D}{V}\frac{p_x}{m} - \frac{p_{\theta_f}}{I_{yy}}\frac{p_z}{m} + \frac{p_{\psi_f}}{I_{zz}}\frac{p_y}{m}$
${}^b\dot{v} = -\frac{T_m}{m} c(\beta_{long}) - \frac{D}{m}\frac{{}^bw}{V} + gc({}^b\theta_f)s({}^b\phi_f) + T_t - {}^br{}^bu + {}^bp{}^bw$	$\dot{p}_y = T_{m,y} + T_t - \frac{D}{V}\frac{p_y}{m} - \frac{p_{\psi_f}}{I_{zz}}\frac{p_x}{m} + \frac{p_{\phi_f}}{I_{xx}}\frac{p_z}{m}$
${}^b\dot{w} = -\frac{T_m}{m} c(\beta_{long})c(\beta_{lat}) - \frac{D}{m}\frac{{}^bw}{V} + gc({}^b\theta_f)c(\phi_f) - {}^bp{}^bv + {}^bq{}^bu$	$\dot{p}_z = -T_{m,z} + mg - \frac{D}{V}\frac{p_z}{m} - \frac{p_{\phi_f}}{I_{xx}}\frac{p_y}{m} + \frac{p_{\theta_f}}{I_{yy}}\frac{p_x}{m}$
${}^b\dot{p} = \frac{T_m}{I_{xx}} h_m s\left(\beta_{lat}\right) + \frac{Q_m}{I_{xx}} s(\beta_{long}) - qr\left(\frac{I_{zz}}{I_{xx}} - \frac{I_{yy}}{I_{xx}}\right)$	$\dot{p}_{\phi_f} = T_{m,y} h_m c(q_{\phi_f}) + T_{m,z} h_m s(q_{\phi_f}) - \frac{p_{\theta_f}}{I_{yy}} p_{\psi_f} + \frac{p_{\psi_f}}{I_{zz}} p_{\theta_f}$
${}^b\dot{q} = -\frac{T_m}{I_{yy}} h_m s\left(\beta_{long}\right) + \frac{Q_m}{I_{yy}} s(\beta_{lat}) - pr\left(\frac{I_{xx}}{I_{yy}} - \frac{I_{zz}}{I_{yy}}\right)$	$\dot{p}_{\theta_f} = -T_{m,x} h_m c(q_{\theta_f}) + T_{m,z} h_m s(q_{\theta_f}) - \frac{p_{\psi_f}}{I_{zz}} p_{\phi_f} + \frac{p_{\phi_f}}{I_{xx}} p_{\psi_f}$
${}^b\dot{r} = \frac{Q_m}{I_{zz}} c(\beta_{long})c(\beta_{lat}) + \frac{T_t}{I_{zz}} l_t - pq\left(\frac{I_{yy}}{I_{zz}} - \frac{I_{xx}}{I_{zz}}\right)$	$\dot{p}_{\psi_f} = Q_m + T_t l_t - \frac{p_{\phi_f}}{I_{xx}} p_{\theta_f} + \frac{p_{\theta_f}}{I_{yy}} p_{\phi_f}$
${}^b\dot{x} = {}^bu$	$\dot{q}_x = \frac{p_x}{m}$
${}^b\dot{y} = {}^bv$	$\dot{q}_y = \frac{p_y}{m}$
${}^b\dot{z} = {}^bw$	$\dot{q}_z = \frac{p_z}{m}$
${}^b\dot{\phi}_f = {}^bp$	$\dot{q}_{\phi_f} = \frac{p_{\phi_f}}{I_{xx}}$
${}^b\dot{\theta}_f = {}^bq$	$\dot{q}_{\theta_f} = \frac{p_{\theta_f}}{I_{yy}}$
${}^b\dot{\psi}_f = {}^br$	$\dot{q}_{\psi_f} = \frac{p_{\psi_f}}{I_{zz}}$

The Hamiltonian is derived similarly to the previous chapter: the kinetic energy K is

$$K = \frac{p_x^2}{2m} + \frac{p_y^2}{2m} + \frac{p_z^2}{2m} + \frac{p_{\phi_f}^2}{2I_{xx}} + \frac{p_{\theta_f}^2}{2I_{yy}} + \frac{p_{\psi_f}^2}{2I_{zz}} \tag{9-2}$$

and the potential energy remains $U = -mgq_z$ due to the reference frame at hand.

The same methods used in Chapter 9 are now used to derive the port-Hamiltonian system. The interested reader is therefore referred to the previous chapter for a full derivation. The port-Hamiltonian 6-DoF system is expressed as:

$$\dot{\boldsymbol{\xi}} = \left[\begin{array}{c} \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \\ \dot{q}_{\phi_f} \\ \dot{q}_{\theta_f} \\ \dot{q}_{\psi_f} \\ \hline \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \dot{p}_{\phi_f} \\ \dot{p}_{\theta_f} \\ \dot{p}_{\psi_f} \end{array}\right] = \underbrace{\left[\begin{array}{c|cccccc} \mathbf{0}^{6\times6} & \multicolumn{6}{c}{\boldsymbol{\mathcal{I}}^{6\times6}} \\ \hline & \text{-}\frac{D}{m} & \frac{p_{\psi_f}}{I_{zz}} & -\frac{p_{\theta_f}}{I_{yy}} & 0 & 0 & 0 \\ & -\frac{p_{\psi_f}}{I_{zz}} & -\frac{D}{m} & \frac{p_{\phi_f}}{I_{xx}} & 0 & 0 & 0 \\ \boldsymbol{\mathcal{I}}^{6\times6} & -\frac{p_{\theta_f}}{I_{yy}} & -\frac{D}{m} & -\frac{p_{\phi_f}}{I_{xx}} & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & -\frac{p_{\psi_f}}{I_{zz}} & \frac{p_{\theta_f}}{I_{yy}} \\ & 0 & 0 & 0 & \frac{p_{\psi_f}}{I_{zz}} & 0 & -\frac{p_{\phi_f}}{I_{xx}} \\ & 0 & 0 & 0 & -\frac{p_{\phi_f}}{I_{xx}} & \frac{p_{\theta_f}}{I_{yy}} & 0 \end{array}\right]}_{\boldsymbol{\mathcal{Q}}} \underbrace{\begin{bmatrix} \frac{\partial H}{\partial \boldsymbol{q}} \\ \frac{\partial H}{\partial \boldsymbol{p}} \end{bmatrix}}_{\frac{\partial H}{\partial \boldsymbol{\xi}}}$$

$$+ \underbrace{\left[\begin{array}{ccccc} \multicolumn{5}{c}{\mathbf{0}^{6\times5}} \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & \text{-}1 & 0 & 0 \\ 0 & \text{-}h_m\cos(q_{\phi_f}) & h_m\sin(q_{\phi_f}) & 0 & 0 \\ \text{-}h_m\cos(q_{\theta_f}) & 0 & h_m\sin(q_{\theta_f}) & 0 & 0 \\ 0 & 0 & 0 & l_t & 1 \end{array}\right]}_{\boldsymbol{g}} \underbrace{\begin{bmatrix} T_{m,x} \\ T_{m,y} \\ T_{m,z} \\ T_t \\ Q_m \end{bmatrix}}_{\boldsymbol{u_{in}}}$$

9-2 Control of the point-mass

In Section 9-2, a standard Passivity-Based Control (PBC) controller was derived for a PH system. However, this is no-longer possible for the 6-DoF system. Equation (5-22) was defined in Chapter 5 to be the closed-loop stability condition for a PH system. As a reminder, Equation (5-21) is repeated below.

$$\dot{H}_d = -\frac{\partial H_d^\intercal}{\partial \boldsymbol{\xi}} \mathcal{R}_d \frac{\partial H_d}{\partial \boldsymbol{\xi}} < 0, \quad \forall \boldsymbol{\xi} \neq \boldsymbol{\xi} \quad \text{and} \quad \dot{H}_d(\boldsymbol{\xi}^*) = \mathbf{0}.$$

with $\boldsymbol{M} = \text{diag}(m, m, m, I_{xx}, I_{yy}, I_{zz})$. Furthermore, $\boldsymbol{\mathcal{R}_d}$ is designed as a positive and diagonal matrix. As a result, $\dot{H}_d < 0$ and the system should be stable. However, it was found that this is not always the case.

Indeed, as mentioned in Section 5-4-6, J. Ferguson et al.[42] state that underactuated systems often struggle with stability. This is especially true in the case that the underactuated DoF are assumed to be undamped. This is the case for all rotations of the helicopter.

P. Kotyczka et al. state that this is the case due to the inability to satisfy the definiteness requirements of the target system and the closed-loop dissipation conditions simultaneously [45]. A deeper study of the system at hand would be required to understand which degrees of freedom are causing this problem.

In order to overcome this issue, an integrator term is added as suggested by [40]. Having the control input defined as $\boldsymbol{u_{in}} = \boldsymbol{\beta(\xi)} + \boldsymbol{v}$, the tuning method developed can no longer be used. The system is instead tuned manually.

Using the integrator, it is easier to find a stabilising controller. Once a stabilising controller is found, its parameters are tuned. The process of finding a controller for the 6-DoF model is more challenging than for 3-DoF. There are many more control parameters to take into account as well as significant cross-couplings between axes.

For example, the yaw and roll angle show strong cross-coupling. This is thought to be due to the fact that roll is closely linked to velocity in the $y-$axis. As a result, the tail attempts to compensate and induces a yaw angle.

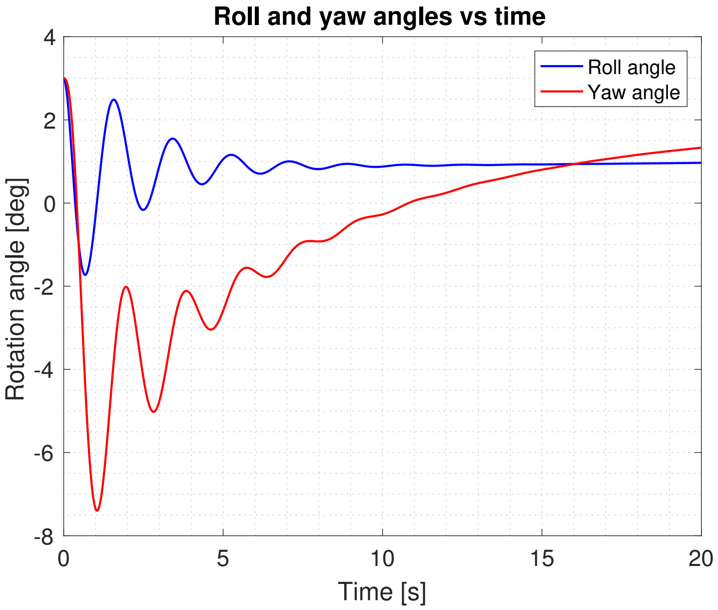


Figure 9-1: Plot demonstrating the cross-coupling between yaw and roll motion.

Figure 9-1 shows that the roll and yaw angles exhibit oscillations that are almost perfectly in phase and dampen out at the same time. Additionally, the yaw is not stable. The best way to damp this effect is to add design parameters to the controller. More specifically, off-diagonal terms should be added in the damping matrix $\mathcal{R}_d$.

$$\mathcal{R}_{d,4} = \begin{bmatrix} k_{q_x} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{q_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{q_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{q_{\phi_f}} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{q_{\theta_f}} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{q_{\psi_f}} \end{bmatrix} \rightarrow \begin{bmatrix} k_{q_x} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{q_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{q_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{q_{\phi_f}} & 0 & k_a \\ 0 & 0 & 0 & 0 & k_{q_{\theta_f}} & 0 \\ 0 & 0 & 0 & k_b & 0 & k_{q_{\psi_f}} \end{bmatrix}$$

It is found that the roll controller drives the yaw response. Consequently, the effect of the roll control should be damped. As the two controllers are in phase, k_b for yaw control should be set to a negative value. This is done to add 180° phase to the signal coming in from the roll to the yaw.

By setting $k_a = 1,000$ and $k_b = -7,000$ with the same controller gains and diagonal elements, a new time response is obtained.

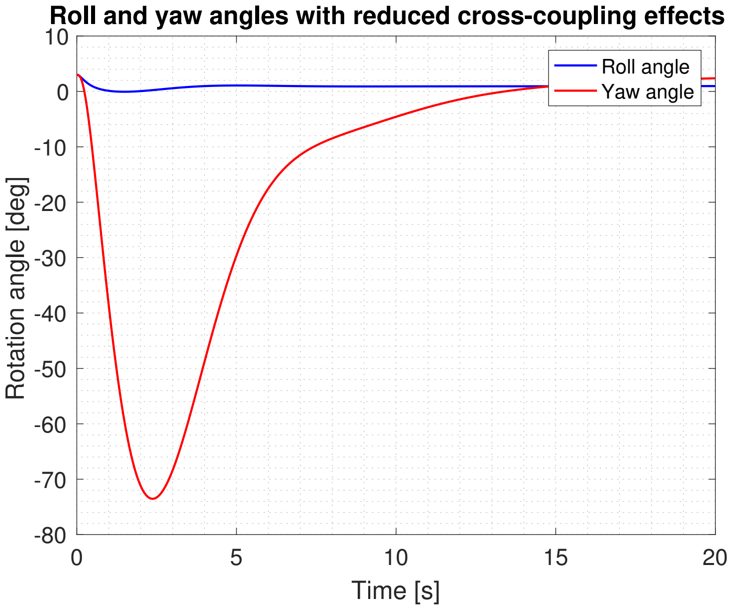


Figure 9-2: Damping control cross-coupling effects between yaw and roll.

Figure 9-2 clearly shows improvements in terms of oscillations. However, the yaw angle now overshoots to -70°. In order to improve on this, the same method is used and the influence of forward velocity and sideways velocity are damped. As a result, the final desired dissipation matrix is as follows:

$$\boldsymbol{\mathcal{R}}_{d,4} = \begin{bmatrix} 2,000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15,000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1,200 & 0 & 0 & 0 \\ 500 & 0 & 4,000 & 6,200 & 0 & 1,000 \\ 0 & 0 & 0 & 0 & 500 & 0 \\ 500 & -3,000 & 0 & -7,000 & 0 & 600 \end{bmatrix}$$

The value of k_ψ when combined with k_a and k_b determines the amount of oscillation the yaw will have. Increasing k_ψ reduces the oscillations but increases the magnitude of the negative peak observed in Figure 9-2. As a result, a compromise is made between oscillation and magnitude.

Performing a trade-off can be complicated and could be done in many different ways. A reasonable target is to have a maximum allowable yaw deflection of around 20° within the first 10 seconds of a forward acceleration. It was determined that k_ψ should be around 600. Doing so renderred the yaw stable and allowed for some oscillations to be reduced.

Chapter 10

Results

The results achieved using the proposed models are presented in this chapter. First, the same acceleration manoeuvre is performed with both the 3-Degrees-of-Freedom (DoF) and 6-DoF models in Section 10-1. The pilot inputs of the two different controllers are then compared.

The trim manoeuvre is used to determine the pilot inputs required to achieve a certain forward velocity whilst keeping all accelerations at 0. Trim in powered flight is a great tool to assess the validity of a model as experimental data are widely available. The pilot inputs in trim are evaluated in Section 10-2. Then, Section 10-3 analyses the flapping methods used. Finally, an attempt to perform an autorotation is covered in Section 10-4.

10-1 The helicopter accelerating - powered flight

The controllers we have designed have been tuned to maintain a reference trajectory. To compare the results, the same acceleration manoeuvre is carried out for both models. The acceleration manoeuvre of the 3-DoF and 6-DoF models can be found in Appendices C-1 and C-2. However, the pilot inputs are compared directly here.

It is interesting to see how the two controllers in Figures 10-1 and 10-2 behave differently. The former figure shows that to slow down, the 3-DoF controller requires an increase in collective angle due to the quick pull on the longitudinal cyclic θ_{long}. Instead, Figure 10-2 has a slower pull on the θ_{long}. Consequently, the collective does not have to be pulled.

The two controllers behave differently, however, after settling, both controllers converge to approximately the same values.

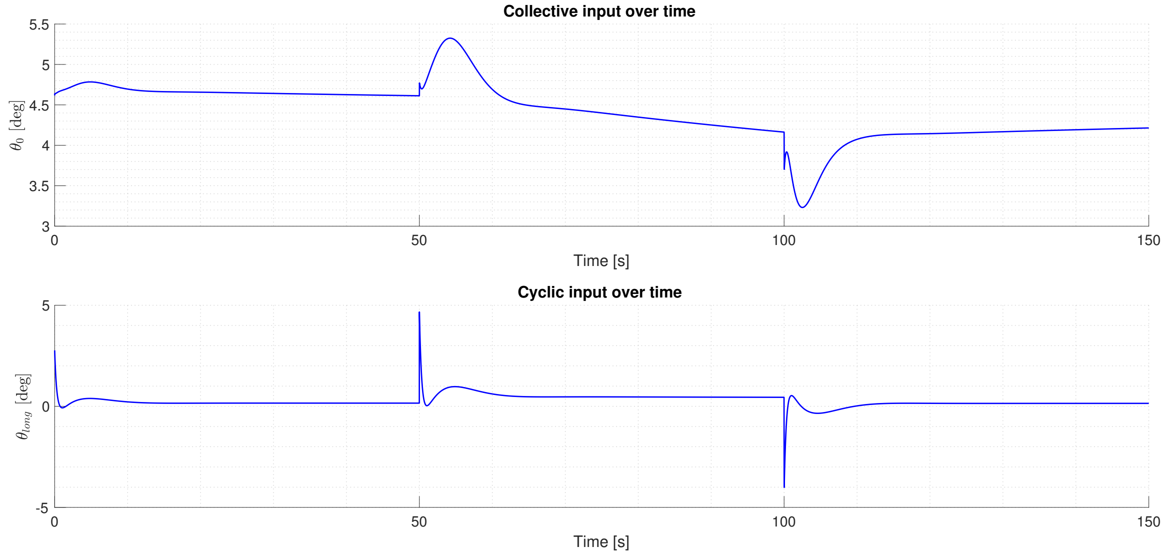


Figure 10-1: Controller inputs for the 3-DoF model.

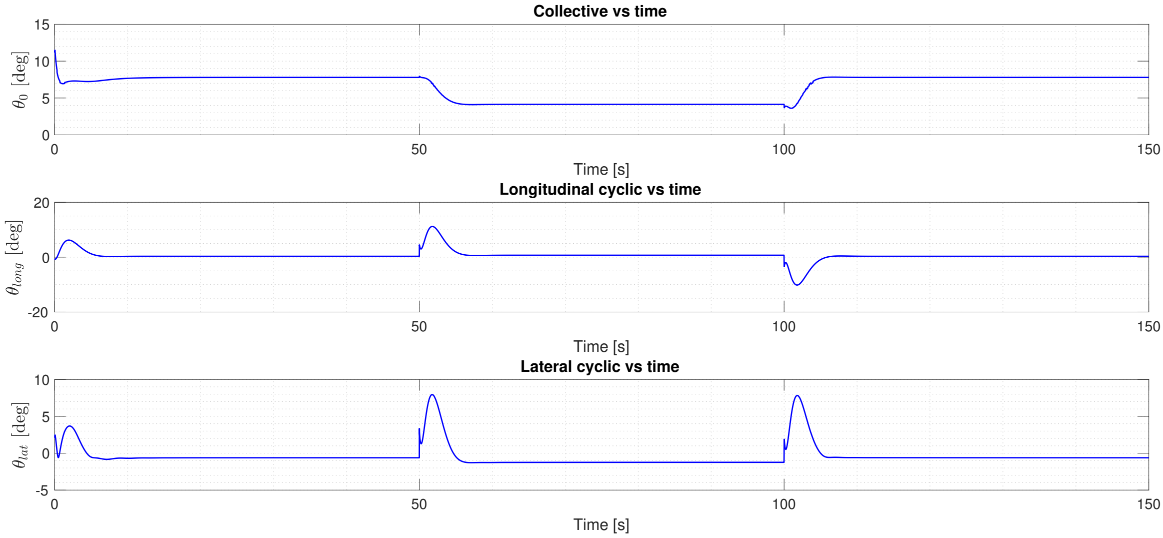


Figure 10-2: Controller inputs for the 6-DoF model.

10-2 The control inputs in trim

To better understand and verify the pilot data, a trim curve is drawn rather than looking at a manoeuvre. The helicopter is said to be in trim when all of its accelerations are zero. To perform the trim curve, the helicopter is given a reference forward velocity that is non-zero, with all accelerations set to 0. Once the controller settles and the vehicle is no longer accelerating, the pilot inputs are recorded. The forward velocity is then increased and the process is repeated.

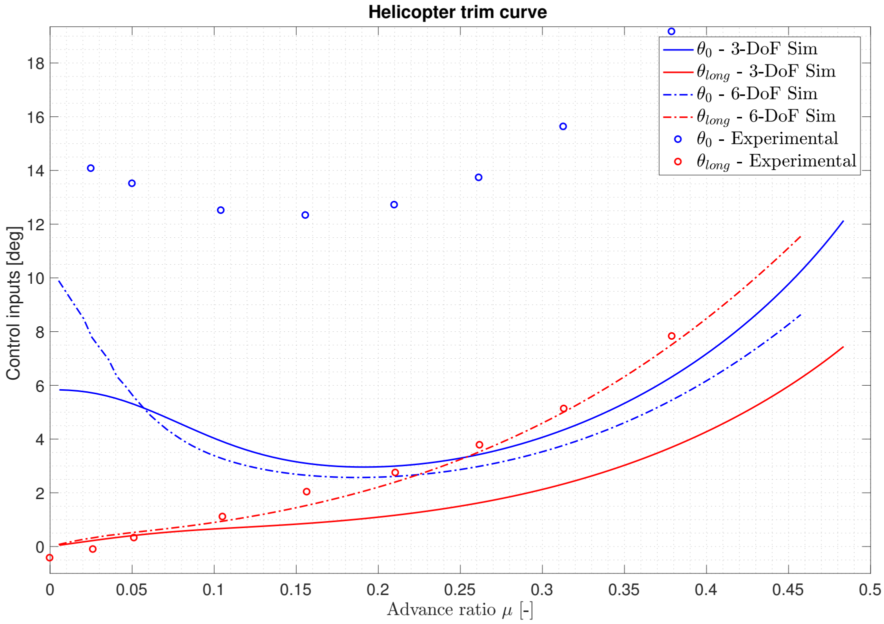


Figure 10-3: Trim curve for a 3 degrees of freedom helicopter model.

Figure 10-3 shows three different sets of curves for the trim of a helicopter. Each set consists of two curves - a blue curve representing the collective input θ_0, and a red curve for the longitudinal cyclic θ_{long}.

The circles denote the experimental data of the Bo105 in trim taken from [46]. The continuous lines show the results of the 3-DoF simulation using port-Hamiltonian (PH) control and the dashed lines depict the results of the 6-DoF control of the helicopter using the model from Chapter 9.

The collective

Let us first consider the collective, θ_0. It can be seen that there is a clear offset between the two curves and the experimental data. To begin with, two assumptions are reflected upon.

The first assumption is that the blades are not tapered. As such, the chord length at the root and at the tip of the blade is assumed to be constant. Remembering the lift equation is directly proportional to the surface area of the lifting surface, assuming no taper clearly overestimates the thrust. As a result, a lower collective angle is required.

Twist is a way of manufacturing the blades such that they have a variable pitch angle along the blade. This is done to improve the performance of the helicopter by generating a more uniform flow on the blades. Prouty states that adding twist may increase the payload by up to 20% [47].

Furthermore, a significant difference is observed between the 3- and 6-DoFs up to $\mu = 0.05$. This discrepancy is caused by the different induced velocity models used for the 3- and 6-

DoF models. The 3-DoF model uses Glauert theory whereas the 6-DoF model uses Haffner diagrams to compute λ_i.

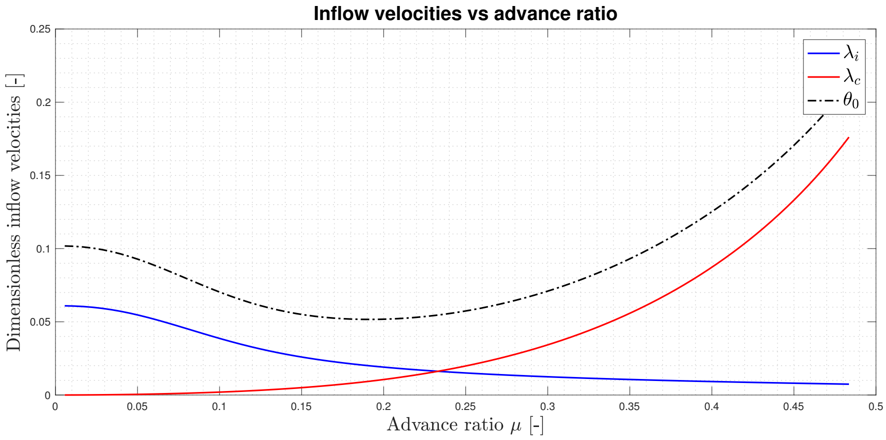


Figure 10-4: Illustration of the relationship between the inflow velocities λ_i and λ_c with the collective θ_0 (in radians) for the 3-DoF model.

Figure 10-4 shows the inflow ratios λ_i and λ_c. Going back to the Blade Element Theory (BET) and looking at Equation (3-24), the term $\lambda_c + \lambda_i$ is found. It is clear that the inflow therefore plays an important part in determining the collective angle.

The Glauert and Haffner methods used in the 3- and 6-DoF models respectively are plotted against eachother in Figure 10-5. The two methods are seen to converge as the advance ratio increases. This is due to the fact that both methods are in fact based on momentum theory. The Haffner diagram was later implemented in the 3-DoF model. The results can be found in Appendix A-2.

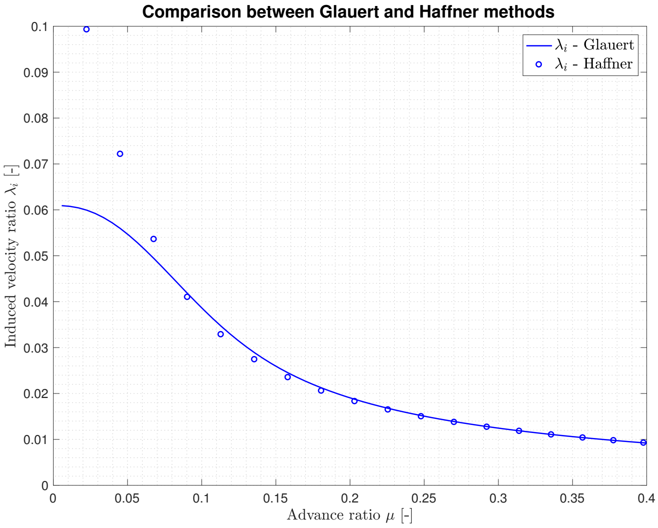


Figure 10-5: Comparison in the Glauert and Haffner models for calculating the induced velocity ratio λ_i as a function of advance ratio μ.

Considering the fact that the collective depends heavily on $\lambda_i + \lambda_c$ explains the similarities at higher advance ratios ($\mu > 0.05$) when the two methods for λ_i converge to the same values as seen in Figure 10-5.

Moreover, another explanation for the discrepancies between the simulations and the Bo105 data is the underprediction of the induced velocity due to momentum theory. Simple momentum theory as used typically results in a 40% underprediction of induced velocity at advance ratios [48].

A very important factor to take into account is that currently, helicopter models all show a discrepancy in the control inputs. Indeed, high fidelity models used in flight simulators typically have as a requirement to have the control positions within ±10% of the true position for trajectory tests [49].

The longitudinal cyclic

The 3-DoF model does not perform as well as the full 6-DoF system with regards to the longitudinal cyclic θ_{long}. By ignoring the effects of the lateral motion and roll, the 3-DoF model underpredicts θ_{long}.

On the other hand, the 6-DoF model predicts θ_{long} almost exactly. It appears that the twist and taper assumptions do not have a significant impact on this control input. Whilst this is true, by implementing twist, an improvement would be seen around hover as the longitudinal cyclic will then not be assumed to be zero.

Comparison with the model by van Holten [1]

The PH model results are compared with a simulation using the trimming method proposed by van Holten [1].

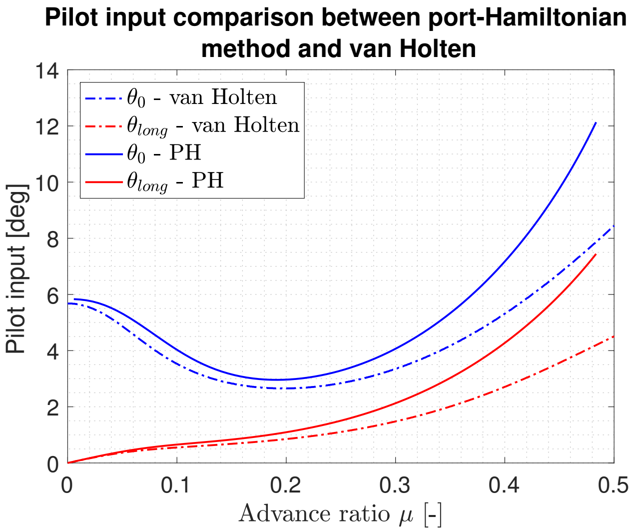


Figure 10-6: Pilot input comparison in trim using the 3-DoF PH model developed and the trim method by van Holten [1].

Figure 10-6 shows that the trim curves are followed well at low speeds. The divergence at higher velocities is explained by the difference in results for the inflow component λ_c.

This is due to the fact that van Holten [1] solves the trim conditions analytically. To do so, small angle approximations have to be made which reduces the accuracy of the model - particularly at high advance ratios when the helicopter is pitching down.

We hypothesise that by using a controller to solve the trim conditions as done for the port-Hamiltonian model, the results would be identical as the small angle approximations no longer have to be made.

The comparison of the non-dimensional inflows is found in Appendix B-1 proving that the models for λ_c diverge at high advance ratios when the small angle approximation cannot be made.

The lateral cyclic

To follow the marker scheme of Figure 10-3, the experimental data is depicted by circles and the 6-DoF simulation by a dashed line.

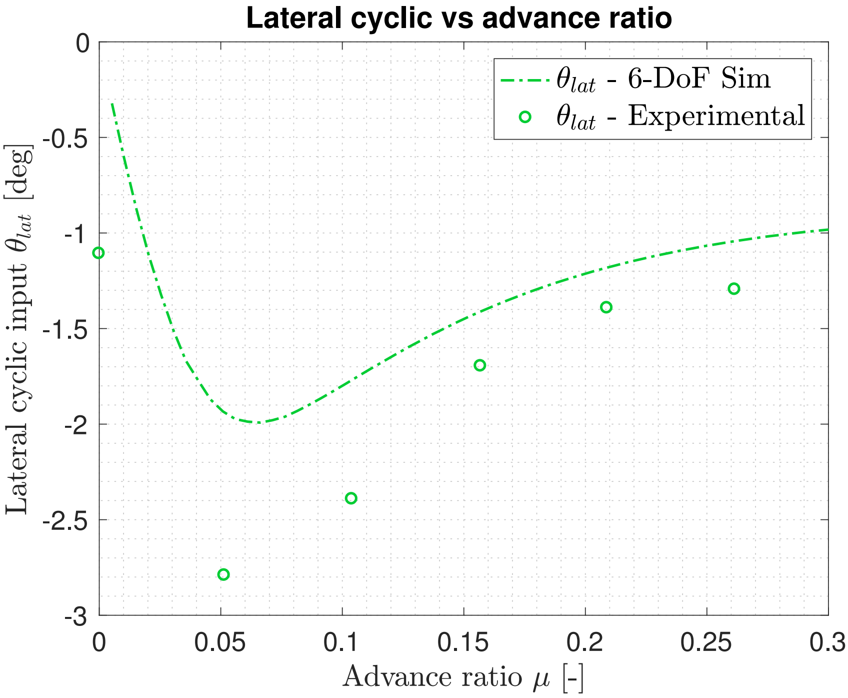


Figure 10-7: Lateral cyclic θ_{long} in trim for the 6-DoF PH model and the experimental data for the Bo105.

It can be seen from Figure 10-7 that the shapes of the curves are similar. The simulation again underestimates the control input with respect to real-life data. It is believed that this is again due to the no-twist assumption. As mentioned for the longitudinal cyclic, implementing twist would result in a non-zero angle at $\mu = 0$.

10-3 Flapping in trim

Plotting the control inputs in trim is a good way to verify the model. Another method to gain better insights into the system is consider flapping. To carry out those comparisons, data for the Bo105 provided by DLR in M. Pavel et al. [50] is considered. Symmetrical motion is first considered with the coning angle a_0 and the longitudinal flap angle a_1.

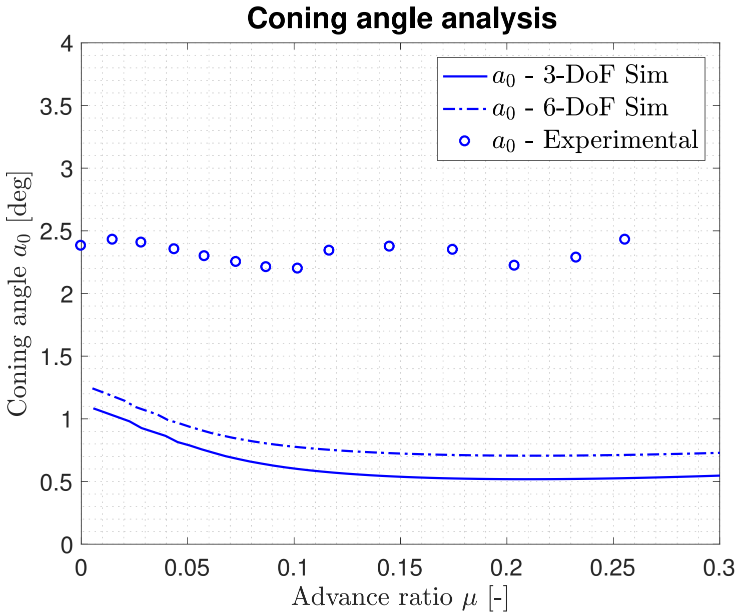


Figure 10-8: Comparison of simulations to experimental data for the coning angle.

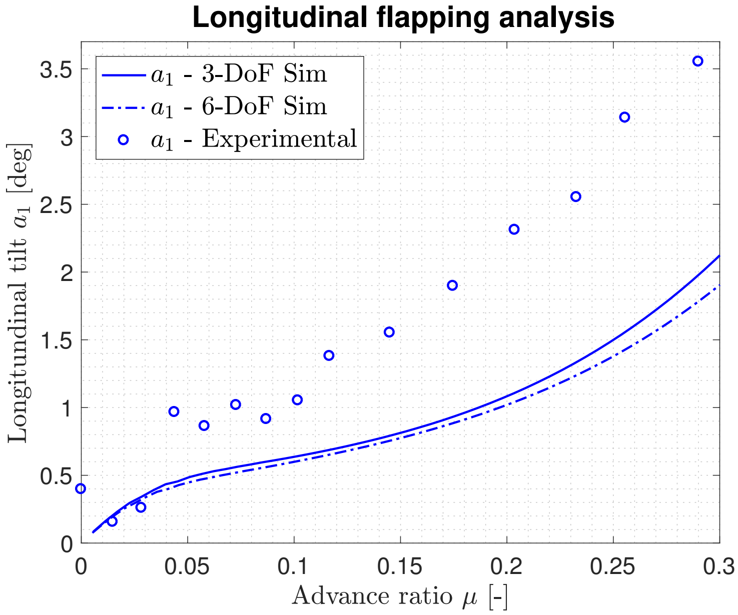


Figure 10-9: Comparison of simulations to experimental data for the longitudinal flap angle.

Figure 10-8 shows that the coning angle has an underapproximation of around 1.5-2°. Equation (3-13) states that:

$$a_0 = \frac{\gamma}{8}\left[\theta_0\left(1+\mu^2\right) - \frac{4}{3}(\lambda_i + \lambda_c)\right]$$

Knowing that the collective is underapproximated, it is to be expected that the coning angle will be affected in a similar way as $a_0 \propto \theta_0$. The shift is not perfectly constant as the error increases at higher forward velocities. This is consistent with the collective underapproximation where momentum theory generates larger underpredictions at higher speeds.

Similarly, a_1 is also a function of $\lambda_i + \lambda_c$ and θ_0. However, the curves begin close to the experimental data and diverge as forward velocity is increased.

It should be noted that the 3-DoF and 6-DoF simulations provide very similar approximations. It appears that adding 3-DoF does not affect these results much. An explanation for this is that the same equations are used for 3- and 6-DoF. Instead, some expressions for a_1 include the influence of the longitudinal cyclic θ_{long}. This could be implemented relatively easily thanks to the *plug-and-play* nature of this model. The point-mass model stays the same and then the helicopter dynamics have to be inserted.

Finally, let us consider lateral flapping. This simulation does not include the 3-DoF model due to the fact that it does not consider lateral motion.

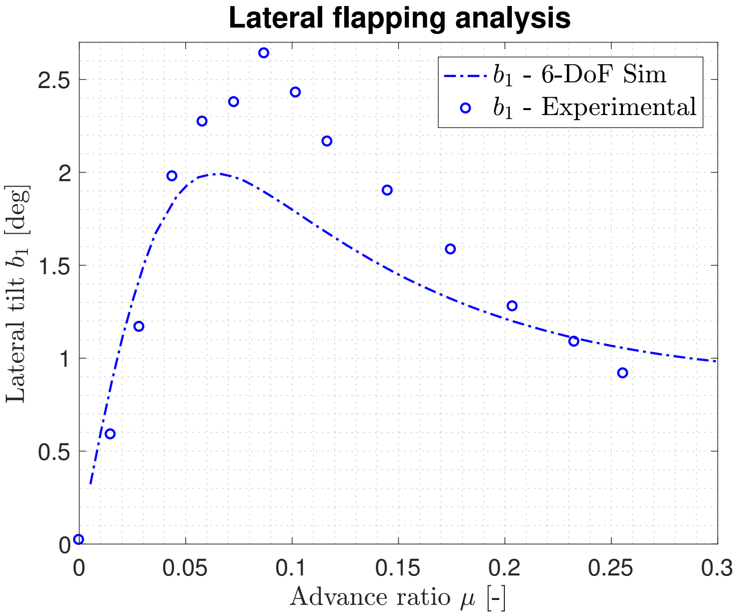


Figure 10-10: Comparison of simulations to experimental data for the lateral flap angle.

Figure 10-10 shows that the lateral flapping angle b_1 is calculated quite accurately in the simulations. The experimental data has a higher and steeper peak than is found in the simulation. It is not perfectly clear why that is the case. Considering equation (3-17), it is known that $b_1(\mu a_0)$. A possible explanation could therefore be that at low forward velocities when μ is small, a_0 has a greater influence on b_1. Then, a_0 being underapproximated may affect the height and steapness of the peak in the simulation.

Overall, flapping is estimated relatively well in the simulations. It has been seen that the collective θ_0 plays an important roll in the flapping dynamics. Furthermore, it was also determined that θ_0 is highly dependent on the inflow velocity into the rotor. It is therefore recommended that, when improving the model, the λ_i and λ_c be considered for improvement.

10-4 3-degrees-of-freedom model in autorotation

Having considered the helicopter in powered flight, it is now time to look at autorotation. Due to the fact that the model does not consider power losses in the rotor, only the glide part of the manoeuvre can be performed. Implementing variable Revolutions Per Minute (RPM) is necessary as during the flare, the kinetic energy from the body gliding is absorbed by the rotor. Once the vehicle has been slowed down sufficiently, the collective is used to control the descent of the helicopter for touch down. This dissipates the kinetic energy of the rotor to its environment. The possibility of implementing the rotor angular velocity is discussed in Chapter 11.

Being a safety manoeuvre in the event of engine failure, it has proven difficult to find data against which the results can be verified and validated. The results should therefore be interpreted with caution.

To glide the helicopter, it is assumed that for each forward velocity, there is a corresponding descent rate the helicopter can follow to maintain constant RPM. To use this assumption, the autorotation torque equation from Equation (3-40) is used to monitor the coefficient. The goal is now to find the combination of u and w for which C_Q is zero as the torque Q_m is directly proportional to C_Q. Maintaining zero torque keeps the rotor RPM constant.

When running the simulations, it is assumed the RPMs are constant. Isolines of constant forward velocity are then drawn for the collective θ_0 and longitudinal cyclic θ_{long}. The simulation considered steps of 1 and 5 meters per second for the descent rate and forward velocity respectively.

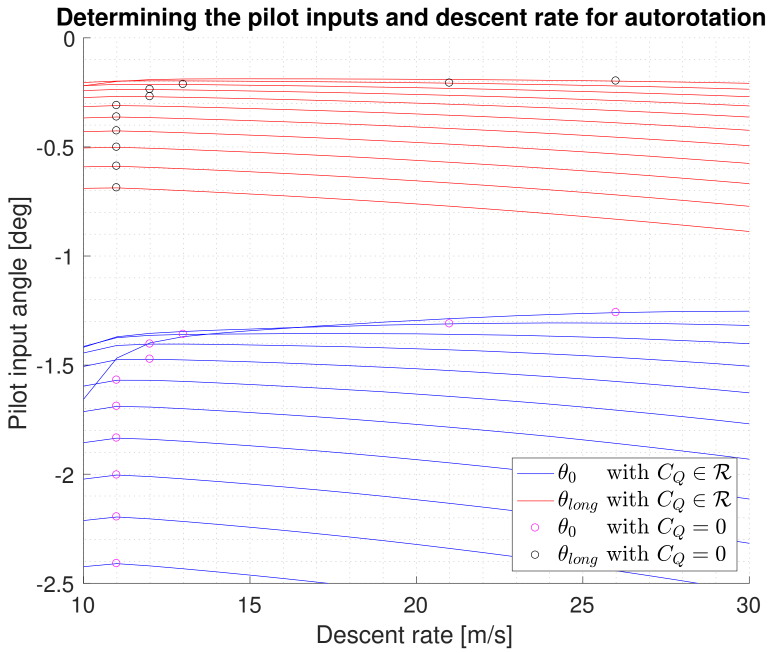


Figure 10-11: Pilot inputs required to maintain velocity trajectories assuming constant RPM.

Figure 10-11 shows how the velocity trajectories can be followed by the pilot/controller assuming the main rotor has constant RPM. However, this can only be assumed when $C_Q = 0$. These points are illustrated by the circles in Figure 10-11. Taking this information into account, new plots are drawn considering only the relevant points.

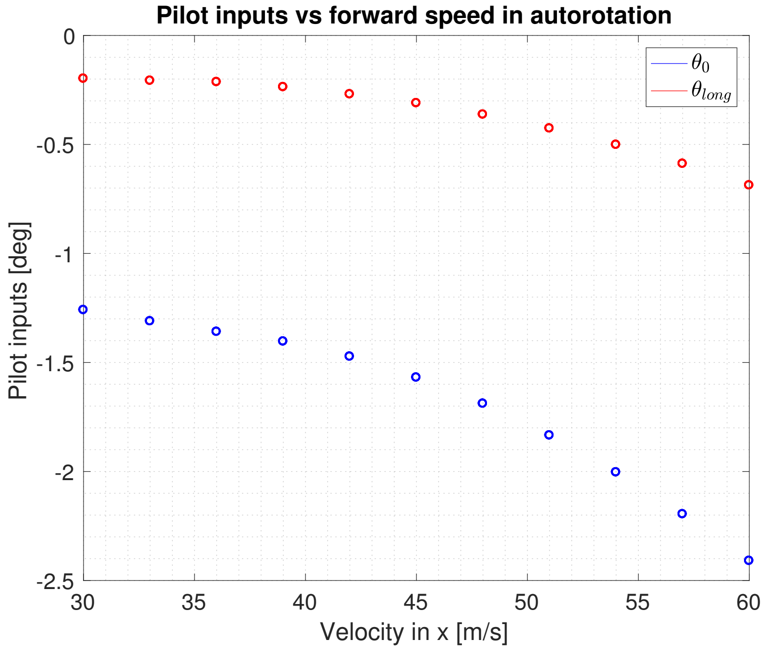


Figure 10-12: Pilot inputs θ_0 and θ_{long} plotted against forward velocity in autorotation.

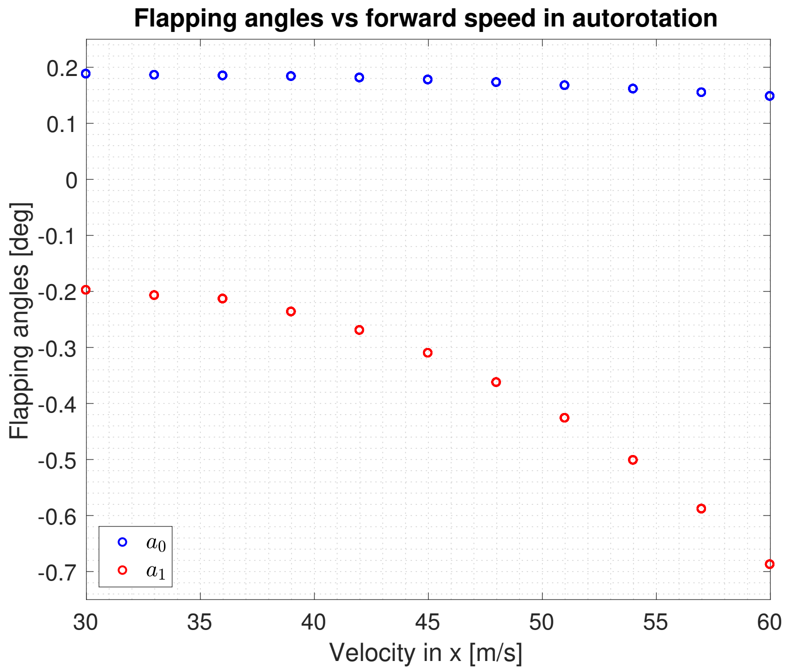


Figure 10-13: Flapping angles a_0 and a_1 plotted against forward velocity in autorotation.

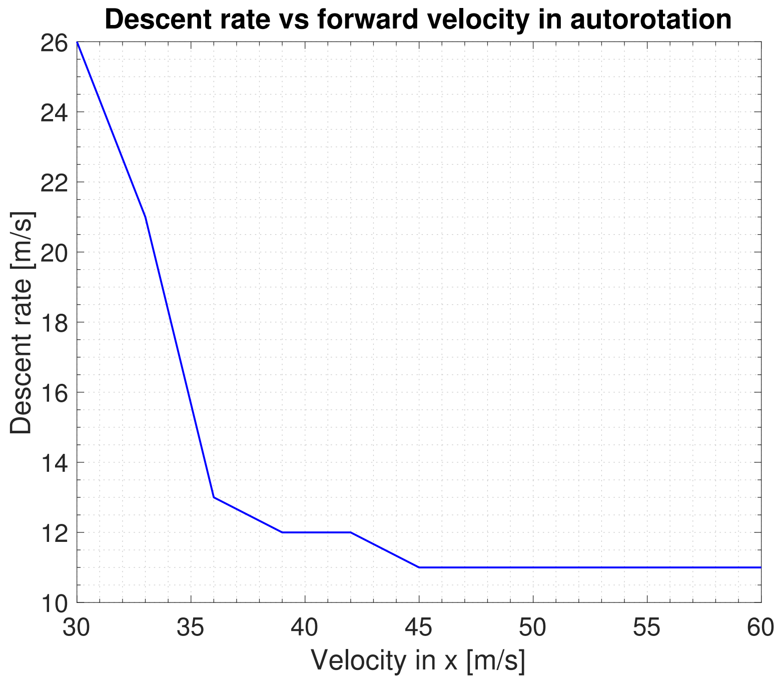


Figure 10-14: Forward velocity and descent rate combinations for autorotation.

Figure 10-12 depicts the pilot inputs required to perform the glide at a given forward velocity. The flapping angles are plotted similarly in Figure 10-13. In order to correlate these to the descent rate, Figure 10-14 can be used.

Typically, to verify an autorotation model, one would have data from a validated or verified model and follow the same trajectories. The pilot inputs and other parameters can then easily be compared. The method presented above could also be used, however, it is atypical and therefore it is difficult to find corresponding data that can be used.

Another way to verify the model would be to implement the trajectories found in Figure 10-14 in a high fidelity software such as FLIGHTLAB [51] and then compare the pilot inputs found in Figure 10-12 to the ones calculated by the high fidelity model.

For now, let us consider the results we obtained. Looking at Figure 10-12, it appears that the collective and cyclic follow a similar shape. However, contrary to what has been seen in the trim plots from Section 10-2, the bucket shape for the collective cannot be observed. It is believed that this is the case as the velocity component w is no longer constrained to be 0, therefore changing the inflows λ_i and λ_c.

An important factor to note is that the collective is now negative. This is done to speed up the rotor speed. However, looking at Figure 10-14, it can be seen that, as the forward velocity increases, the required descent rate decreases. This appears to be a contradicting result, as one would expect an increase in descent rate corresponding to a decrease in collective angle.

On the other hand, the longitudinal cyclic follows an intuitive curve. An increase in longitudinal cyclic is used to gain forward acceleration.

Finally, Figure 10-14 shows a rather strange pattern with the descent rate being constant between 45 and 60 meters per second of forward speed. No definite explanation has been found for this but, it is suspected to be caused by the torque coefficient model. Further investigation is required to determine whether this shape is nominal or not and what causes this shape.

Chapter 11

Conclusions and recommendations

11-1 Conclusion

In this thesis, port-Hamiltonian (PH) Passivity-Based Control (PBC) has been used to control a 3-Degrees-of-Freedom (DoF) and 6-DoF point-mass models. These models have been combined with helicopter dynamics to determine the pilot inputs required to obtain the desired motion of the point-mass. One of the benefits of this method is that other systems, such as airplanes, can be modelled using the point-mass. This can be done by modifying $\boldsymbol{gu_{in}}$ and altering the helicopter dynamics to match those of an airplane.

Furthermore, a novel method of tuning PH systems has been developed in Chapter 6. This method allows controller designers to build a quick and optimal controller in an intuitive way. When designing a PH-PBC, the closed-loop system denoted by subscript d is viewed as an autonomous system:

$$\dot{\boldsymbol{\xi}} = (\mathcal{J} - \mathcal{R})\frac{\partial H}{\partial \boldsymbol{\xi}} + \boldsymbol{gu_{in}} = (\mathcal{J}_d - \mathcal{R}_d)\frac{\partial H_d}{\partial \boldsymbol{\xi}}$$

This format is taken advantage of by designing the closed-loop system to resemble a mass-spring-damper system. Given a certain target trajectory, the system then attempts to achieve critical damping, thereby minimising the settling time.

This method has proven to work well with the 3-DoF point-mass model where critical damping was achieved with minimal tuning efforts. However, for the 6-DoF model, which is more highly underactuated, the controller struggles to stabilise the system. Further work on the method should be carried out to better understand its applicability and shortcomings. A good starting point would be to understand whether the findings in [45] can be applied to the method.

Moreover, previous work such as [43] have attempted to formulate a flight mechanics model directly into the PH framework. It has been shown in this thesis that this is not possible due to the fact that when including the Blade Element Theory (BET) directly into the equations of motion, some terms render the system non-passive. To work around this, the mentioned point-mass method was used. However, a possible alternative will be given in Section 11-3.

The results of the powered flight simulations have been compared to each other and to experimental data for the Bo105 helicopter. These simulations show the pilot inputs as well as the flapping angles of the helicopter in trim. Overall, all simulations appear to follow the experimental data well. However, it is often seen that the simulations underestimate the real data. This is common for helicopter models and currently an issue as helicopter simulators lack accuracy for Zero Flight Time Training (ZFFT)[1] [49].

Despite the common underestimation of helicopter models, the developed models underestimate too much due to the inflow models for λ_c and λ_i using basic momentum theory. Additionally, it is expected that implementing blade twist will increase the collective by around 20%, thereby rendering the model more accurate.

To close the gap to high fidelity helicopter simulators, it is suggested to make use of a λ_i model that does not depend on basic momentum theory. Additionally, it is expected that implementing blade twist will increase the collective by around 20%, thereby further improving the model performance.

Finally, an attempt to perform an autorotative glide was made. Although the model was not verified due to a lack of available data, it shows promising initial results. The beauty of the point-mass helicopter formulation is that the helicopter dynamics can simply be interchanged with those of another model. By implementing verified and/or validated models for the flapping and torque coefficient of the rotor, there is no doubt that the autorotation can be performed.

11-2 Answering the research questions

Can a Newtonian body model of a helicopter be directly translated into the port-Hamiltonian framework?

No - It was found that implementing a classical Newtonian body model directly into the port-Hamiltonian framework is not possible. This is due to the fact that when implementing the BET directly into the equations of motion, non-passive terms are found, rendering the use the port-Hamiltonian framework unsuitable. However, alternatives exist. In this thesis, a 3-DoF and 6-DoF point-mass models have been designed and verified against flight data from the Bo105.

Can the performance of the PBC controller be optimised to carry out safe manoeuvres?

Yes - A novel method of tuning has been developed. By imitating a mass-spring-damper system, the PBC is optimised to obtain critical damping. As a result, the shortest settling time is obtained with the least amount of oscillations possible. This is true for the 3-DoF model. We hypothesised that the degree of underactuation of the 6-DoF model was a limiting factor for the method. Further research is required to better understand the methods' applicability and limitations.

---

[1] Zero Flight Time Training is a method to train experienced pilots on simulators to complete different aircraft type ratings.

Can a passivity-based controller for a port-Hamiltonian model of a helicopter be developed to perform an autorotation?

Yes - First of all, it has been shown that the control of a 3-DoF and 6-DoF in powered flight is entirely possible as they have been verified against experimental data. The glide of the helicopter in autorotation was simulated but the manoeuvre has not yet been verified nor validated.

However, the answer to this question is yes due to the flexibility of the point-mass model. Having built the framework for the glide, using validated BET and flapping models would ensure a valid glide.

As to the flare and landing parts of the manoeuvre, variations in rotor speed first have to be implemented. This is again entirely possible through the use of a validated method of calculating C_Q.

11-3 Recommendations for future work

Considerable strides towards modelling and controlling helicopters through PBC in the PH framework have been made. The work carried out in this thesis has laid a foundation opening a myriad of possibilities for future work.

The first recommendation would be to implement more accurate models of the induced velocity that do not depend on basic momentum theory. It would be interesting to see the effect this would have on the collective in powered flight. It is also believed that the implementation of blade twist is relatively simple and would significantly improve the model and lift the collective curve by up to 20%. These two improvements to the models should allow them to match experimental data more closely.

Combining the model with an observer would then be the next natural step. This could be done by attempting to recreate manoeuvres by using experimental data. Otherwise, validated software such as FLIGHTLAB could be used to validate the models and develop the observers.

The majority of the difficulties encountered during the development of the autorotation model came from the fact that variable Revolutions Per Minute (RPM) of the rotor was not modelled. Simple steps can be taken to do so. By implementing the torque of the rotor, known trajectories of helicopters in autorotation could then be replicated, thereby considerably facilitating the verification of the model.

Moreover, the full potential of PH systems has not been utilised in this thesis. Port-Hamiltonian systems can be used to model highly complex systems and are suitable to model networks of systems. This could be used for example, to perform the BET directly in the PH framework. Taamallah [51] made use of virtual work and Lagrangian methods for his derivation of the BET. This could perhaps be implemented and reformulated to the PH framework.

Naturally, this leads to the thought that the entire helicopter system could be expressed in the port-Hamiltonian framework. The models derived in this thesis are based on the 3-block structure introduced in Section 3-6-3. By formulating each block in the PH framework, these could then be cascaded using the theory developed H. Yu et al. [52] to create one complete helicopter model in the PH framework.

To truly assess the controller(s), their robustness should be evaluated. To do so, it is suggested to implement the wind in the model to understand how the controller recovers from gusts and adjusts to situations such as head-wind or side-wind.

Finally, the limitations of the tuning method developed in Chapter 6 should be understood. Its first constraints were found when trying to apply the method to the 6-DoF model. However, it has shown to work for the 3-DoF model. It is believed that limitations arise depending on the degree of underactuation. A formal study of these limitations is highly recommended.

Appendix A

Haffner diagram and its implementation

A-1 Haffner diagram

```
%% Create Haffner matrix

global HaffMat HaffX vi_vec sizeX dvi ds x_vec Vcos

% Possible horizontal and vertical velocities
ds = 0.001;
Vcos = 0:ds:21;
x_vec = 0:ds:0.9;

% Induced velocity itereations
dvi = 0.01;
vi_vec = 0.05:dvi:1;
n_vi = length(vi_vec);
n_Vcos = length(Vcos);

% Set up Haffner matrix
HaffMat = zeros(n_vi, n_Vcos*2);
HaffX = [Vcos fliplr(Vcos)];
sizeX = zeros(n_vi, 1);
for idx = 1:n_vi
    y_pos = -vi_vec(idx)^2 + sqrt(1/(vi_vec(idx)^2) - Vcos.^2);
    y_neg = -vi_vec(idx)^2 - sqrt(1/(vi_vec(idx)^2) - Vcos.^2);

    %index = find(abs(imag(y_pos)) == 0);
    for (im_idx = 1:n_Vcos)
        if (imag(y_pos(im_idx)) ~= 0)
            x_max = im_idx-1;
            sizeX(idx) = x_max;
```

```
            break
        end
    end

    y_pos(x_max:end) = NaN;
    y_neg(x_max:end) = NaN;

    HaffMat(idx, :) = [y_pos fliplr(y_neg)];
    %HaffnerMatrix(idx, length(HaffnerMatrix)-x_max+1:end) = y_neg;

    %plot([Vcos fliplr(Vcos)], HaffnerMatrix(idx, :), '-', 'LineWidth', 2)
    %hold on
end
```

```
function [ vi_final ] = inducedHaffner( x_des, y_des )

%% Haffner plots %%
% by Nicolas Mavrocordatos.

% This script takes x_des and y_des as input. These are the desired x and y
% coordinates on the Haffner plot which correspond to a lateral and
% vertical velocity respectively (V*cos(alpha_d), V*sin(alpha_d)).

% The output is the induced velocity of the helicopter normalised by its
% hover induced velocity (vi_ = vi/vi_h).

% Notes and assumptions:
% 1 - This does not consider backwards velocity.
% 2 - A desired point within the left-most circle is said to have vi_ = 1.
% 3 - Between (0, -2) and (0.9, -1.4), some intersection of circles can be
% seen. Two quadratic equations are used to highlight this area and vi_h is
% assumed to be 1.
% 4 - ( a x b ) notation is used when describing matrices/vectors. a
% corresponds to the contents of the rows and b to the one of the columns.


%% Interpolation part

global Vcos HaffnerX HaffnerMatrix n_Vcos ds n_vi Vi sizeX dvi xMaxVi

% Find the equation of the line through (0, -1) and (x_des, y_des).
x_line = 0;
y_line = -1;    % c
grad_line = (y_des - y_line)/(x_des - x_line);  % m
line_eq = grad_line * Vcos + y_line;  % y = m*x + c
% Plot the line
plot(Vcos, line_eq, 'g')
```

```

% Find the indeces which correspond to V*cos(alpha_d).
x_position = ones(2,1);
for idx = 1:n_Vcos
    if abs(HaffnerX(idx) - x_des) < ds
        x_position(1) = idx;
        x_position(2) = n_Vcos*2 - (x_position(1)-1);
        break
    else
        x_position(1) = 1;
        x_position(2) = 1;
    end
end

y_upp = HaffnerMatrix(:,x_position(1));
y_dwn = HaffnerMatrix(:,x_position(2));
% Note, since x is as [1 2 3 3 2 1], two indeces are found. One for the +
% solution and one for the -ive solution. Record the y values for each of
% those.

% Numerically check within which circle the desired point is
inside = zeros(1, n_vi);
for idx = 1:n_vi
    % If the desired point lies between the y value of the two solutions
    % for current vi_, the point lies within that circle.
    if (y_des < y_upp(idx) && y_des > y_dwn(idx))
        inside(idx) = 1;
    end
end
% The location of the point is found by summing up the number of circles
% that it is encircled by.
posInt = sum(inside);

% If statement was added to prevent numerical errors in Simulink, it could
% be removed here (but not its contents).
if posInt < n_vi
    % An interpolation is to be made between vi_ and vi_+dvi. To do so, set
    % up 4 vectors with +solution, -ive solution at vi_ and vi_+dvi
    y_search1 = HaffnerMatrix(posInt, 1:sizeX(posInt)-1);
    y_search2 = fliplr(HaffnerMatrix(posInt, n_Vcos*2-sizeX(posInt)+2:end));
    y_search3 = HaffnerMatrix(posInt+1, 1:sizeX(posInt)-1);
    y_search4 = fliplr(HaffnerMatrix(posInt+1, n_Vcos*2-sizeX(posInt)+2:end));
end

%% Nasty areas (left-most circle and intersections)
check = 1;
if y_des < -1.47 && y_des > -2.01
    if x_des < 0.9
        x = 0:ds:0.9;
```

```
        n_x = length(x);
        y = 0.5*x.^2 -2.01;
        x2 = 1.6*x;
        y2 = 0.2*x2.^2 - 1.885;
        for idx = 1:n_x
            temp = find(abs(x_des-x) < ds);
            if y_des < y2(temp) && y_des > y(temp)
                % If in nasty area, assume vi_ = 1 for now.
                vi_final = 1;
                check = 0;
            end
        end
    end
end

%%

% The value here depends on the minimum vi chosen. i.e. if min(vi) = 0.2,
% the first line will have a maximum x value of 5.
if x_des >= 10
    vi_final = 0;
    check = 0;
end

%%

% When looking at the Haffner plots, a gap can be seen in the semi-circles.
% This gap is the point where the quadratic solution goes from
% (-b + sqrt(..))/2a to (-b - sqrt(..))/2a. This gap can be reduced by
% making ds smaller. However, this slows Simulink down a lot.
% Therefore, if (x_des, y_des) is in a region at risk, increase threshold.
thresh = 0.01;
if y_des < 0 && y_des > -1.045
    if x_des > 2.1
        thresh = 0.2;
    else
        thresh = 0.1;
    end
else
    thresh = 0.01;
end

% The searching now begins. Find the points at V*cos(alpha_d) where the
% difference between the constant vi_ line and the y=mx+c line is smaller
% than the threshold.
if posInt < n_vi && check == 1
    for idx = 1:sizeX(posInt)-1
        % Check for point where line y=mx+c intersects vi_ circle
        if abs(line_eq(idx) - y_search1(idx)) < thresh
            % Positive solution
            posX1 = Vcos(idx);
```

```
            posY1 = line_eq(idx);
            %plot([posX1 posX1], [posY1 y_search1(idx)], 'bo')
        elseif abs(line_eq(idx) - y_search2(idx)) < thresh
            % Negative solution
            posX1 = Vcos(idx);
            posY1 = line_eq(idx);
            %plot([posX1 posX1], [posY1 y_search2(idx)], 'bo')
        else
            posX1 = xMaxVi(posInt);
            posX2 = xMaxVi(posInt+1);
            posY1 = y_des;
            posY2 = y_des;
        end
        % Check for point where line y=mx+c intersects vi_+dvi circle
        if abs(line_eq(idx) - y_search3(idx)) < thresh
            % Positive solution
            posX2 = Vcos(idx);
            posY2 = line_eq(idx);
            %plot([posX2 posX2], [posY2 y_search3(idx)], 'bo')
        elseif abs(line_eq(idx) - y_search4(idx)) < thresh
            % Negative solution
            posX2 = Vcos(idx);
            posY2 = line_eq(idx);
            %plot([posX2 posX2], [posY2 y_search4(idx)], 'bo')
        else
        end
    end
    % Linear interpolation between the lines vi_ and vi_+dvi
    distTot = sqrt((posX1-posX2)^2 + (posY1-posY2)^2);
    distR2L = sqrt((posX1-x_des)^2 + (posY1-y_des)^2);
    percDiff = distR2L/distTot;
    vi_final = Vi(posInt) + dvi*percDiff;
else
    vi_final = 1;
end
fprintf('vi/vi_h = %.4f \n', vi_final);

% Finish the plot
xlim([0 10])
ylim([-10 10])
title('Haffner Diagram', 'FontSize', 20)
xlabel('$$\bar{V} \cos(\alpha)$$', 'Interpreter', 'latex', 'FontSize', 18)
ylabel('$$\bar{V} \sin(\alpha)$$', 'Interpreter', 'latex', 'FontSize', 18)
plot(x_des, y_des, 'ro')
grid minor

end
```

A-2 Haffner diagram implementation

Implementation of the Haffner diagram to the 3-Degrees-of-Freedom (DoF) model for trim.

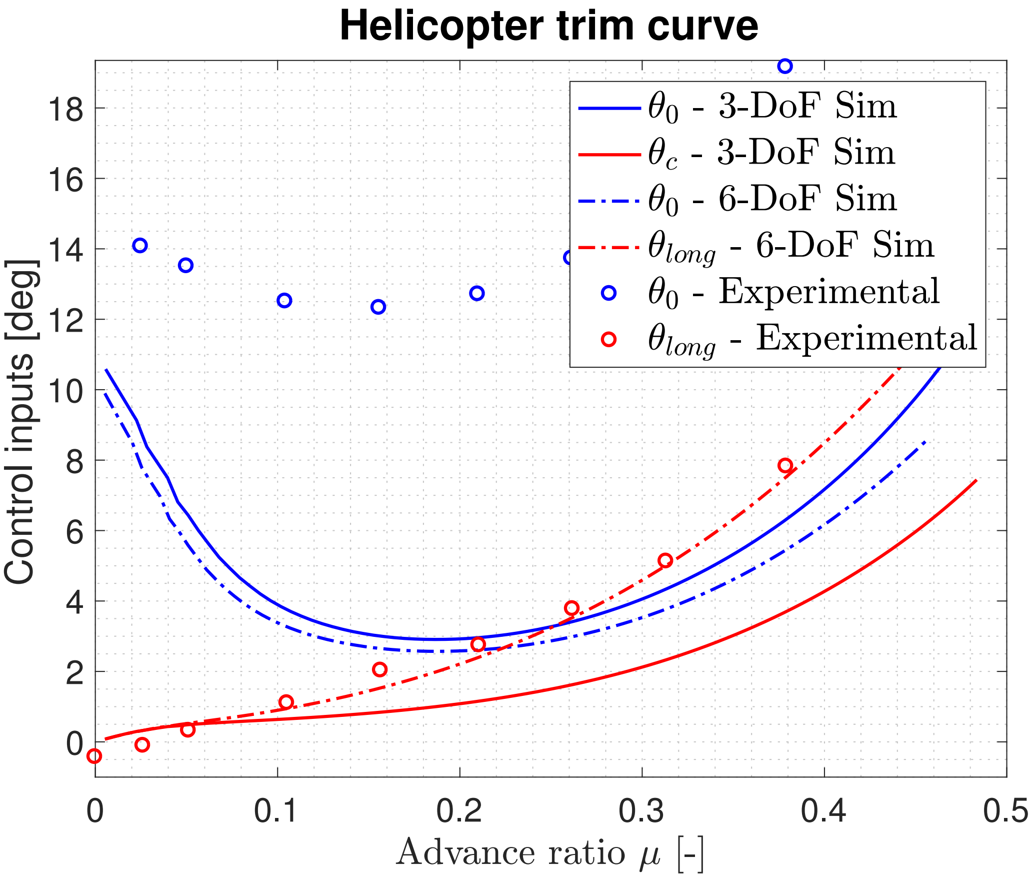


Figure A-1: Trim results of the helicopter in powered flight using Haffner diagrams to calculate the induced velocity ratio λ_i.

Appendix B

6-degrees-of-freedom non-passive terms

By again splitting the thrust into $T_m = T_1\theta_0 - T_2\lambda$, the acceleration equations of motion of the 6 Degrees-of-Freedom (DoF) body model are given by:

$$m\dot{u} = (T_1\theta_0 - T_2\lambda)\sin(\beta_{long}) - mg\sin(\theta_f) - D\frac{u}{V} - mqw + mrv \tag{B-1a}$$

$$m\dot{y} = -(T_1\theta_0 - T_2\lambda)\cos(\beta_{lat}) + mg\cos(\theta_f)\sin(\phi_f) + T_t - D\frac{v}{V} - mru + mpw \tag{B-1b}$$

$$m\dot{w} = -(T_1\theta_0 - T_2\lambda)h_m\cos(\beta_{long})\cos(\beta_{lat}) + mg\cos(\theta_f)\cos(\phi_f) - D\frac{w}{V} - mpv + mqu \tag{B-1c}$$

$$I_{xx}\dot{p} = (T_1\theta_0 - T_2\lambda)\sin(\beta_{lat})h_m + Q_m\sin(\beta_{long}) - qr(I_{zz} - I_{yy}) \tag{B-1d}$$

$$I_{yy}\dot{q} = -(T_1\theta_0 - T_2\lambda)\sin(\beta_{long})h_m + Q_m\sin(\beta_{lat}) - pq(I_{xx} - I_{zz}) \tag{B-1e}$$

$$I_{zz}\dot{r} = Q_m\cos(\beta_{long})\cos(\beta_{lat}) + T_t l_t - pq(I_{yy} - I_{xx}) \tag{B-1f}$$

The control inputs are now given by $\boldsymbol{u_{in}} = [\theta_0 \quad \tau_{\beta_{long}} \quad \tau_{\beta_{lat}} \quad T_t]^\intercal$. As a result, two new coordinate states are introduced, $q_{\beta_{long}}$ and $q_{\beta_{lat}}$ with their respective momentum expressed as $p_{\beta_{long}}$ and $p_{\beta_{lat}}$. This results in an addition of the following two accelerations

$$I_{rot}\dot{q}_{\beta_{long}} = \tau_{\beta_{long}} \tag{B-2a}$$

$$I_{rot}\dot{q}_{\beta_{lat}} = \tau_{\beta_{lat}} \tag{B-2b}$$

The state vector is now defined as:

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} [q_x \quad q_y \quad q_z \quad q_{\theta_f} \quad q_{\phi_f} \quad q_{\theta_f} \quad q_{\psi_f} \quad q_{\beta_{long}} \quad q_{\beta_{lat}}]^\intercal \\ [p_x \quad p_y \quad p_z \quad p_{\theta_f} \quad p_{\phi_f} \quad p_{\theta_f} \quad p_{\psi_f} \quad p_{\beta_{long}} \quad p_{\beta_{lat}}]^\intercal \end{bmatrix}$$

Again, the structure of the interconnection matrix $\mathcal{J}$ is given by:

$$\boldsymbol{\mathcal{J}} = \begin{bmatrix} \mathbf{0}^{n/2\times n/2} & \boldsymbol{\mathcal{I}}^{n/2\times n/2} \\ -\boldsymbol{\mathcal{I}}^{n/2\times n/2} & \mathbf{0}^{n/2\times n/2} \end{bmatrix}$$

Removing the Coriolis terms along with the dissipation forces and terms connected to the inputs, the following can be said about the partial derivatives of the Hamiltonian:

$$\frac{\partial H}{\partial p_x} = \frac{p_x}{m} \tag{B-3a}$$

$$\frac{\partial H}{\partial p_y} = \frac{p_y}{m} \tag{B-3b}$$

$$\frac{\partial H}{\partial p_z} = \frac{p_z}{m} \tag{B-3c}$$

$$\frac{\partial H}{\partial p_{\phi_f}} = \frac{p_{\phi_f}}{I_{xx}} \tag{B-3d}$$

$$\frac{\partial H}{\partial p_{\theta_f}} = \frac{p_{\theta_f}}{I_{yy}} \tag{B-3e}$$

$$\frac{\partial H}{\partial p_{\psi_f}} = \frac{p_{\psi_f}}{I_{zz}} \tag{B-3f}$$

$$\frac{\partial H}{\partial p_{\beta_{long}}} = \frac{p_{\beta_{long}}}{I_{rot}} \tag{B-3g}$$

$$\frac{\partial H}{\partial p_{\beta_{long}}} = \frac{p_{\beta_{lat}}}{I_{rot}} \tag{B-3h}$$

$$-\frac{\partial H}{\partial q_x} = -mg\sin(q_{\theta_f}) - T_2\lambda\sin(q_{\beta_{long}}) \tag{B-3i}$$

$$-\frac{\partial H}{\partial q_y} = -mg\cos(q_{\theta_f})\sin(q_{\phi_f}) + T_2\lambda\cos(q_{\beta_{lat}}) \tag{B-3j}$$

$$-\frac{\partial H}{\partial q_z} = mg\cos(q_{\theta_f})\cos(q_{\phi_f}) + T_2\lambda\cos(q_{\beta_{long}})cos(q_{\beta_{lat}}) \tag{B-3k}$$

$$-\frac{\partial H}{\partial q_{\phi_f}} = Q_m\sin(q_{\beta_{long}}) - T_2\lambda\sin(q_{\beta_{lat}})h_m \tag{B-3l}$$

$$-\frac{\partial H}{\partial q_{\theta_f}} = Q_m\sin(q_{\beta_{lat}}) + T_2\lambda h_m\sin(q_{\beta_{long}})h_m \tag{B-3m}$$

$$-\frac{\partial H}{\partial q_{\psi_f}} = Q_m\cos(q_{\beta_{long}})\cos(q_{\beta_{lat}}) \tag{B-3n}$$

$$-\frac{\partial H}{\partial q_{\beta_long}} = 0 \tag{B-3o}$$

$$-\frac{\partial H}{\partial q_{\beta_{lat}}} = 0 \tag{B-3p}$$

Expressing the Hamiltonian in terms of kinetic and potential energies, $H = T + U$, the kinetic energy is again simply described as:

$$K = \frac{1}{2}\boldsymbol{p}^\intercal \boldsymbol{M}\boldsymbol{p} \tag{B-4}$$

The potential energy U is now given by:

$$\begin{aligned} U = &- q_x \underbrace{\left(-mg\sin(q_{\theta_f}) - T_2\lambda\sin(q_{\beta_{long}})\right)}_{-\frac{\partial H}{\partial q_x}} - q_y \underbrace{\left(-mg\cos(q_{\theta_f})\sin(q_{\phi_f}) + T_2\lambda\cos(q_{\beta_{lat}})\right)}_{-\frac{\partial H}{\partial q_y}} \dots \\ &- q_z \underbrace{\left(mg\cos(q_{\theta_f}) + T_2\lambda\cos(q_{\beta_{long}})\right)}_{-\frac{\partial H}{\partial q_z}} - q_{\phi_f} \underbrace{\left(Q_m\sin(q_{\beta_{long}}) - T_2\lambda\sin(q_{\beta_{lat}})h_m\right)}_{\partial q_{\phi_f}} \dots \\ &- q_{\theta_f} \underbrace{\left(Q_m\sin(q_{\beta_{lat}}) + T_2\lambda h_m\sin(q_{\beta_{long}})h_m\right)}_{-\frac{\partial H}{\partial q_{\theta_f}}} - q_{\psi_f} \underbrace{\left(Q_m\cos(q_{\beta_{long}})\cos(q_{\beta_{lat}})\right)}_{-\frac{\partial H}{\partial q_{\psi_f}}} \dots \\ &+ \underbrace{0}_{-\frac{\partial H}{\partial q_{\beta_{long}}}} + \underbrace{0}_{-\frac{\partial H}{\partial q_{\beta_{lat}}}} \end{aligned} \tag{B-5}$$

Taking the partial deriviative of the potential energy with respect to the coordinates, it is shown once again that:

$$\frac{\partial}{\partial \boldsymbol{q}}\left(\int \frac{\partial H}{\partial \boldsymbol{q}} d\boldsymbol{q}\right) \neq \frac{\partial H}{\partial \boldsymbol{q}} \tag{B-6}$$

With $\partial U/\partial \boldsymbol{q}$ given as:

$$\begin{bmatrix} \frac{\partial H}{\partial q_x} \\ \frac{\partial H}{\partial q_y} \\ \frac{\partial H}{\partial q_z} \\ \frac{\partial H}{\partial q_{\phi_f}} \end{bmatrix} = \begin{bmatrix} mg\sin(q_{\theta_f}) + T_2\lambda\sin(q_{\beta_{long}}) \\ mg\cos(q_{\theta_f})\sin(q_{\phi_f}) - T_2\lambda\cos(q_{\beta_{lat}}) \\ -mg\cos(q_{\theta_f}) - T_2\lambda\cos(q_{\beta_{long}}) \\ -Q_m\sin(q_{\beta_{long}}) + T_2\lambda\sin(q_{\beta_{lat}})h_m + mg\cos(q_{\theta_f})\cos(q_{\phi_f}) \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial H}{\partial q_x} \\ \frac{\partial H}{\partial q_y} \\ \frac{\partial H}{\partial q_z} \\ \frac{\partial H}{\partial q_{\phi_f}} \end{bmatrix} = \begin{bmatrix} -Q_m \sin(q_{\beta_{lat}}) - T_2 \lambda h_m \sin(q_{\beta_{long}}) + mg\left(q_x \cos(q_{\theta_f}) - q_y sin(q_{\theta_f}) + q_z \sin(q_{\theta_f})\right) \\ -Q_m \cos(q_{\beta_{long}}) \sin(q_{\beta_{lat}}) \\ \begin{gathered} T_2 \lambda \left(q_x \cos(q_{\beta_{long}}) + q_z \sin(q_{\beta_{long}}) - q_{\theta_f} \lambda \cos(q_{\beta_{long}})\right) \\ +Q_m \left(-q_{\phi_f} \cos(q_{\beta_{long}}) + q_{\psi_f} \sin(q_{\beta_{long}} \cos(q_{\beta_{lat}}))\right) \end{gathered} \\ \begin{gathered} T_2 \lambda \left(-q_y \sin(q_{\beta_{lat}}) + q_{\phi_f} \cos(q_{\beta_{lat}})\right) + \\ Q_m \left(-q_{\theta_f} \cos(q_{\beta_{lat}}) + q_{\psi_f} \cos(q_{\beta_{long}}) \sin(q_{\beta_{lat}})\right) \end{gathered} \end{bmatrix}$$

B-1 Comparison between the port-Hamiltonian method and van Holten [1]

The figure below compares the non-dimensional velocities during trim achieved with the developed port-Hamiltonian method with the trim simulations derived from [1].

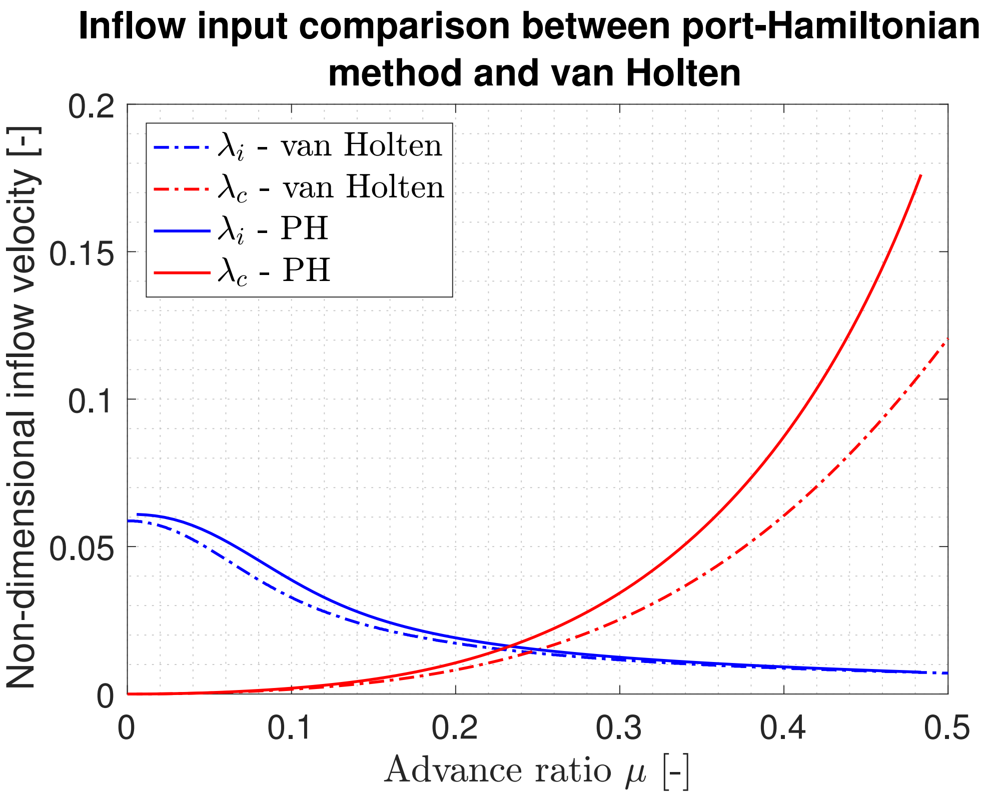


Figure B-1: Inflow comparison in trim using the 3-DoF PH model developed and the trim method from van Holten [1].

Appendix C

Acceleration of helicopter in powered flight

C-1 3-degrees-of-freedom acceleration results

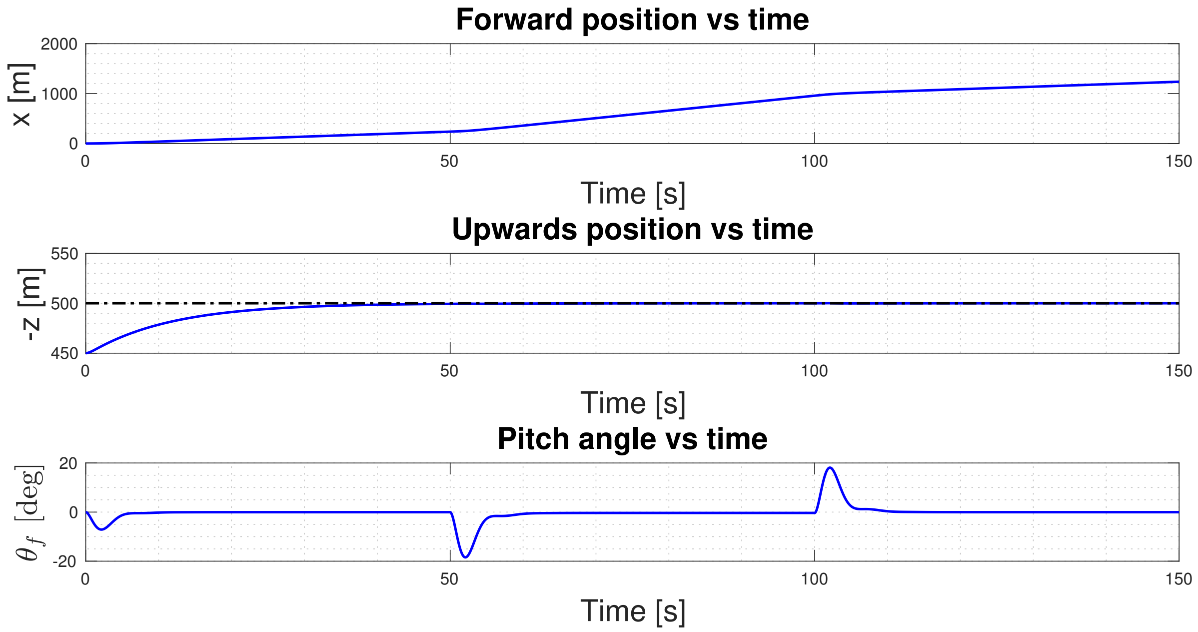


Figure C-1: Coordinates of 3-DoF model performing the acceleration manoeuvre.

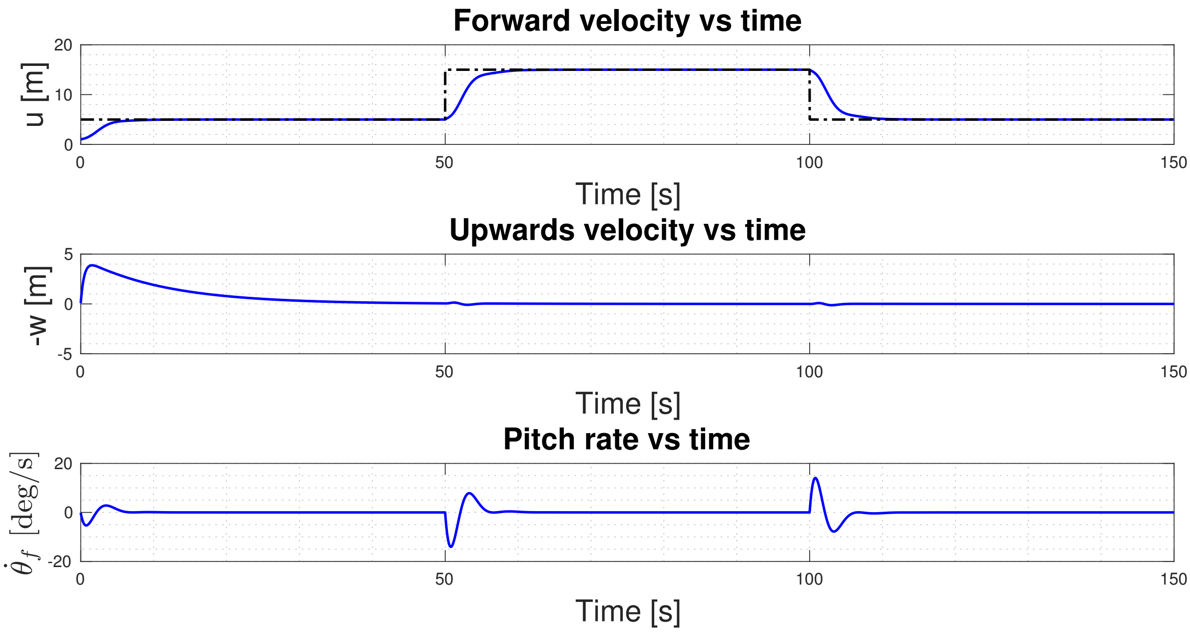


Figure C-2: Velocities of 3-DoF model performing the acceleration manoeuvre.

C-2 6-degrees-of-freedom acceleration results

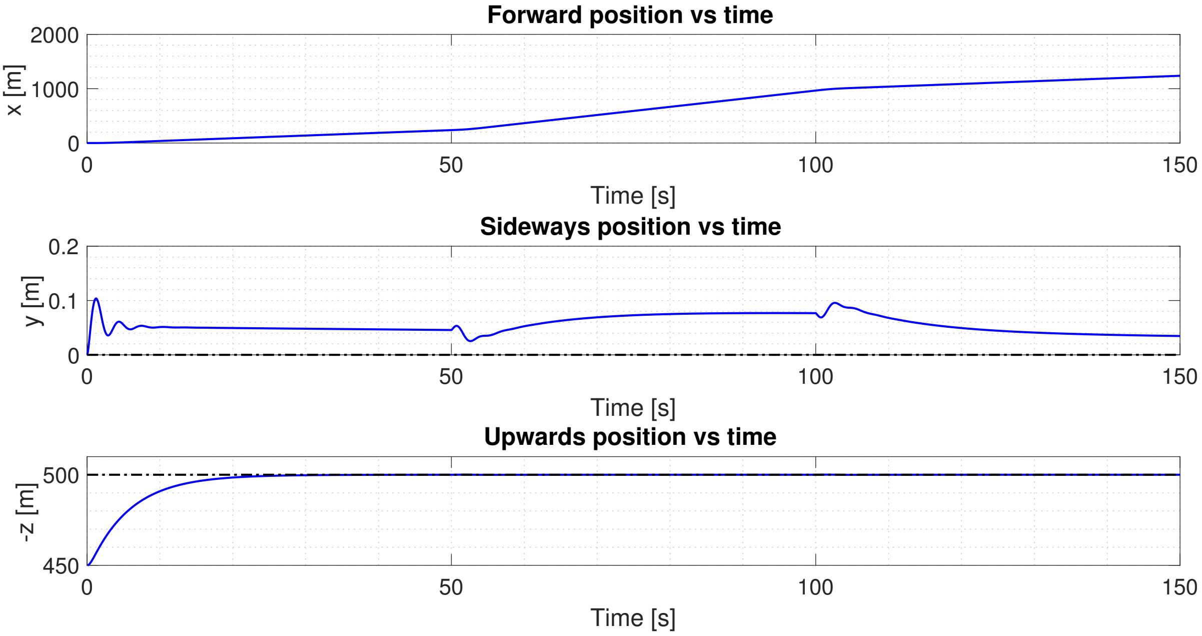


Figure C-3: Linear displacements of the 6-DoF model perfoming the acceleration manoeuvre.

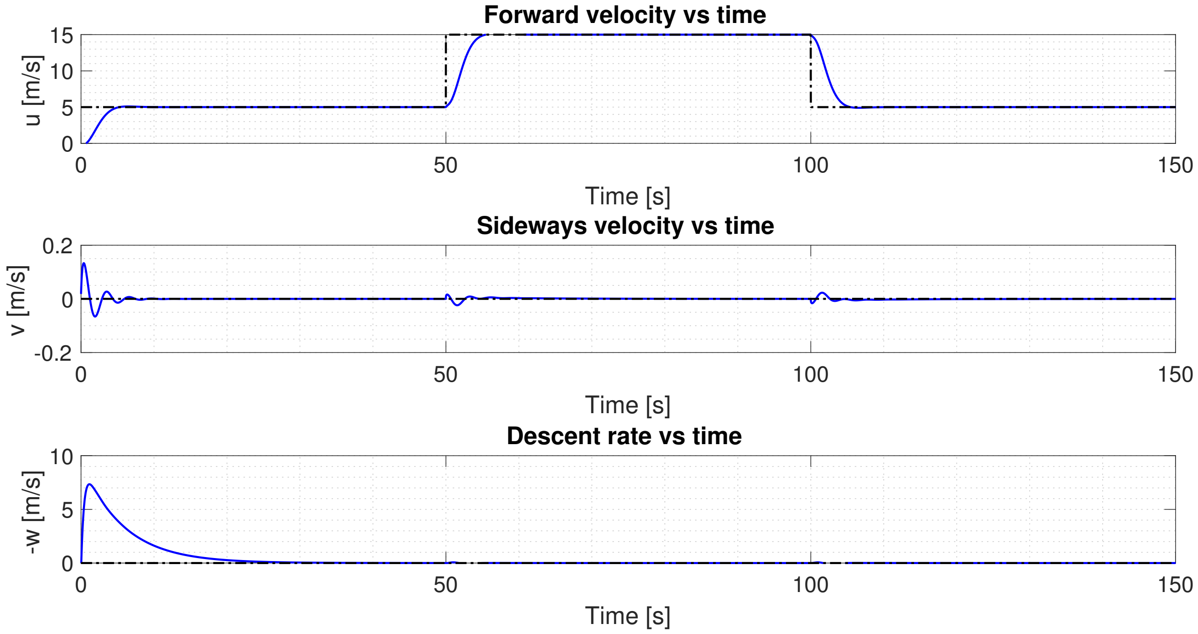


Figure C-4: Linear velocities of the 6-DoF model perfoming the acceleration manoeuvre.

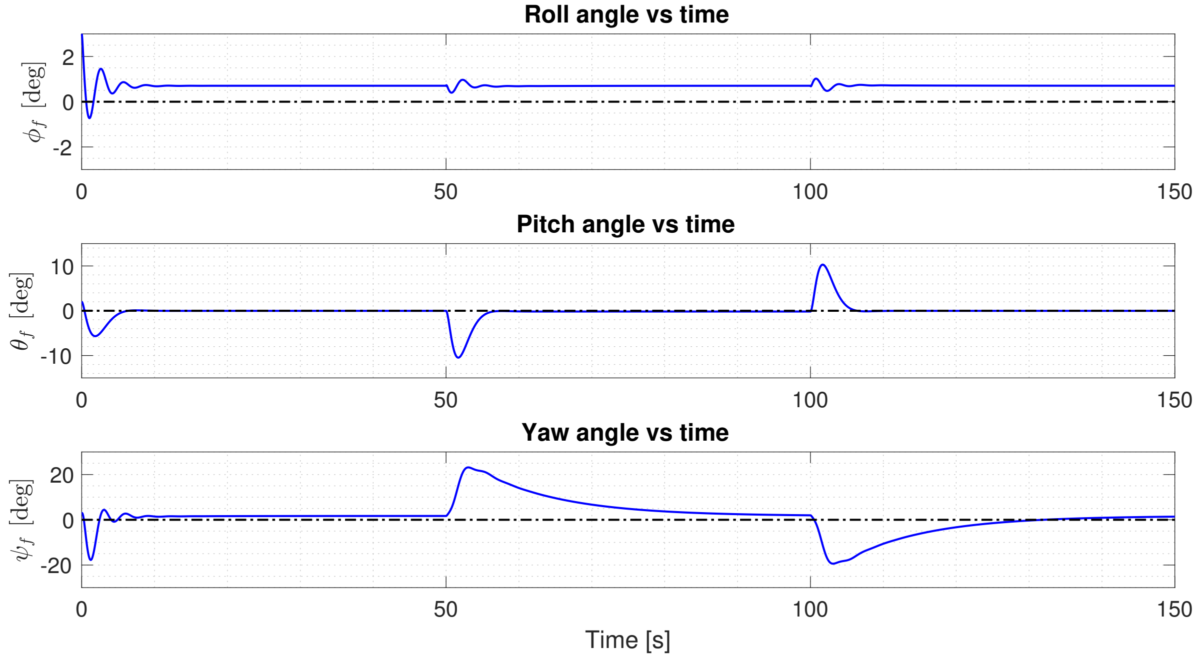


Figure C-5: Angular displacements of the 6-DoF model perfoming the acceleration manoeuvre.

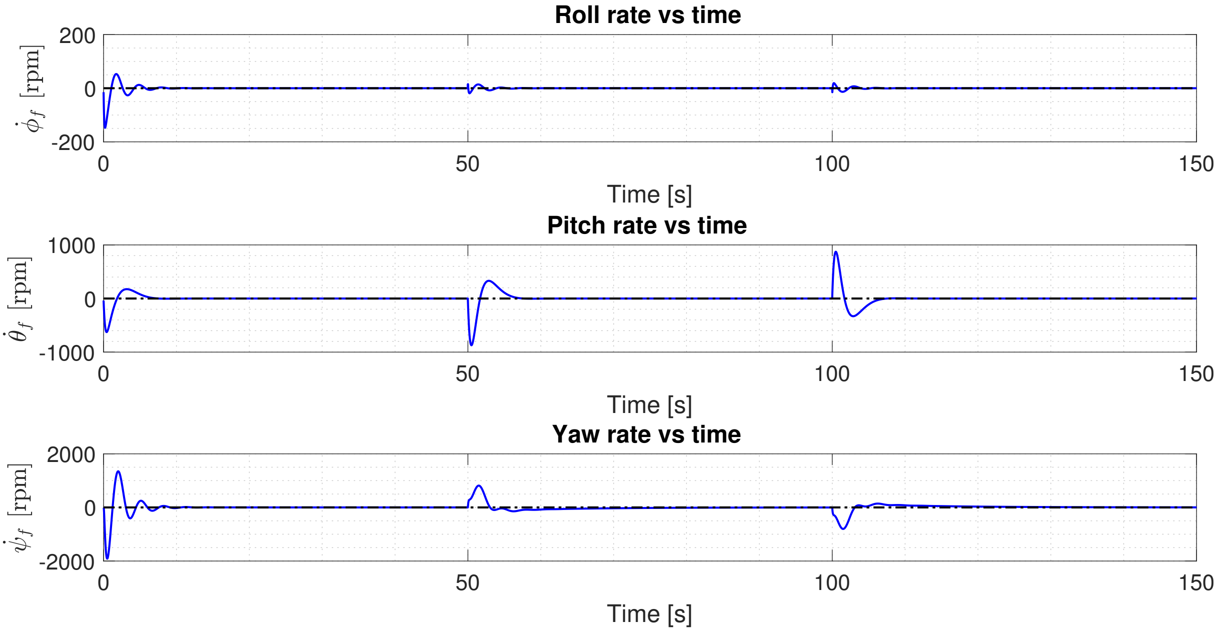


Figure C-6: Angular velocities of the 6-DoF model perfoming the acceleration manoeuvre.

Bibliography

[1] T. van Holten and J. Melkert, *Helicopter performance stability and control: AE4-213 course reader.* Technical University of Delft, 2002.

[2] R. Prouty, *Helicopter performance, stability, and control.* PWS Engineering, 1986.

[3] M. Arcak, C. Meissen, and A. Packard, *Networks of dissipative systems: compositional certification of stability, performance, and safety.* Springer, 2016.

[4] A. van der Schaft and D. Jeltsema, *Port-Hamiltonian Systems Theory: An Introductory Overview.* Now Publishers, 2014.

[5] F. Mohammed, A. Idries, N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "Uavs for smart cities: Opportunities and challenges," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 267–273, May 2014.

[6] Skyryse, "Hate traffic, get over it," *Skyryse*, 2018.

[7] C. Metz, "Silicon valley takes a (careful) step toward autonomous flying," *The New York Times*, Aug 2018.

[8] E. Adams, "The head-spinning challenge of teaching helicopters to fly themselves," *Wired*, Jul 2018.

[9] Uber Technologies Inc, "Uber elevate," 2020.

[10] GoFly, "Together, we will make people fly," *GoFly Prize*, 2019. `https://goflyprize.com`.

[11] Q. Jingze, "From autorotation to safe landing," *Procedia Engineering*, vol. 17, pp. 46 – 51, 2011. The 2nd International Symposium on Aircraft Airworthiness.

[12] T. Binns, *The FAA: Federal Aviation Administration.* Heinemann Library, 2003.

[13] V. Cortellessa, B. Cukic, D. Del Gobbo, A. Mili, M. Napolitano, M. Shereshevsky, and S. Harjinder, “Certifying adaptive flight control software,” *West Virginia University*, 2000.

[14] W. Johnson, “Helicopter optimal descent and landing after power loss,” *Ames Research Center*, Jul 1977.

[15] K. Hazawa, J. Shin, D. Fujiwara, K. Igarashi, D. Fernando, and K. Nonami, “Autonomous autorotation landing of small unmanned helicopter,” *Nihon Kikai Gakkai Ronbunshu, C Hen/Transactions of the Japan Society of Mechanical Engineers, Part C*, vol. 70, no. 10, pp. 2862–2869, 2004.

[16] K. Dalamagkidis, K. Valavanis, L. Piegl, R. Beard, P. Oh, A. Ollero, and H. Piegl Shim, *Autonomous Autorotation of Unmanned Rotorcraft using Nonlinear Model Predictive Control*, pp. 351–369. Dordrecht: Springer Netherlands, 2010.

[17] P. Abbeel, A. Coates, T. Hunter, and A. Ng, “Autonomous autorotation of an rc helicopter,” *Experimental Robotics Springer Tracts in Advanced Robotics*, pp. 385–394, 2009.

[18] P. Kruger, *Interconnection and damping assignment passivity-based control of an unmanned helicopter.* PhD thesis, Potchefstroom Campus, 2015.

[19] R. Ortega, A. van der Schaft, I. Mareels, and B. Maschke, “Putting energy back in control,” *IEEE Control Systems Magazine*, vol. 21, pp. 18–33, April 2001.

[20] S. S. McGowen, “Helicopters: An illustrated history of their impact,” 2005.

[21] M. Pavel, “Blade flapping dynamics,” 2010. AE4213 lecture slides.

[22] F. A. Administration, *Helicopter Flight Controls.* Skyhorse Pub Co Inc, 2014.

[23] P. F. Scaramuzzino, M. D. Pavel, D. M. Pool, O. Stroosma, and M. Mulder, *Development of a Theoretical Framework to Predict the Necessary Level of Complexity of the Flight Dynamics Model for Autorotation Training.* 2019.

[24] J. M. Seddon and S. Newman, “Rotor aerodynamics in forward flight,” *Basic Helicopter Aerodynamics. Chichester: John Wiley & Sons*, pp. 111–138, 2011.

[25] M. Pavel, *Six Degrees of Freedom Linear Model for Helicopter Trim and Stability Calculation.* Memorandum Delft University of Technology, Faculty of Aerospace Engineering, Delft University of Technology, Faculty of Aerospace Engineering, 1996.

[26] H. Visser, M. Pavel, and S. Tang, “Optimization of rotorcraft simultaneous non-interfering noise abatement approach procedures,” *Journal of Aircraft - J AIRCRAFT*, vol. 46, pp. 2156–2161, 01 2009.

[27] J. M.-D. Filippo, M. Delgado, C. Brie, and H. M. Paynter, “A survey of bond graphs : Theory, applications and programs,” *Journal of the Franklin Institute*, vol. 328, no. 5, pp. 565 – 606, 1991.

[28] H. Paynter, *Analysis and design of engineering systems.* M.I.T. Press, 1961.

[29] E. Hairer, “Hamiltonian systems,” February 2010.

[30] A. van der Schaft, “Port-hamiltonian systems: an introductory survey,” *Proceedings of the International Congress of Mathematicians*, vol. 3, pp. 1339–1366, 01 2006.

[31] R. Ortega, A. van der Schaft, B. Maschke, and G. Escobar, “Interconnection and damping assignment passivity-based control of port-controlled hamiltonian systems,” *Automatica*, vol. 38, pp. 585–596, 04 2002.

[32] IEEE Control Systems Society, “Brief history of control,” *Conference Catalysts*, 2019.

[33] J. Maxwell, “On governors,” *The Scientific Papers of James Clerk Maxwell*, pp. 105–120.

[34] V. Ghanbari, *Dissipativity/passivity based control design for symmetric and switched systems with application to medical robotics.* PhD thesis, Graduate School, 2017.

[35] J. Willems, “Dissipative dynamical systems part i: General theory,” *Archive for Rational Mechanics and Analysis*, vol. 45, pp. 321–351, Jan 1972.

[36] H. Khalil, *Nonlinear Systems.* Pearson Education, Prentice Hall, 2002.

[37] H. Bai, M. Arcak, and J. Wen, *Cooperative control design. A systematic, passivity-based approach*, vol. 89. Springer, 01 2011.

[38] E. Weisstein, “Annihilator,” *Wolfram MathWorld*, 2019.

[39] K. Uren and G. van Schoor, “Interconnection and damping assignment passivity-based control of port-controlled hamiltonian systems,” *Automatica*, vol. 38, pp. 585–596, 04 2002.

[40] H. Ramirez, Y. le Gorrec, Y. Wu, and Ecole Nationale Superieure de Mecanique et des Microtechniques, “Doctoral course: Introduction to control of port-hamiltonian systems,” 2019.

[41] Z. Wang and P. Goldsmith, “Modified energy-balancing-based control for the tracking problem,” *IET Control Theory & Applications*, vol. 2, no. 4, pp. 310–322, 2008.

[42] J. Ferguson, A. Donaire, R. Ortega, and R. Middleton, “Robust integral action of port-hamiltonian systems,” *IFAC-PapersOnLine*, vol. 51, no. 3, pp. 181 – 186, 2018. 6th IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control LHMNC 2018.

[43] J. Kerkkamp, “Nonlinear control for a small-scale helicopter uav in autorotation,” Master’s thesis, Technical University of Delft, July 2016.

[44] U. B. Hald, M. V. Hesselbaek, J. T. Holmgaard, C. S. Jensen, S. L. Jakobsen, and M. Siegumfeldt, *Autonomous Helicopter - Modelling and Control.* University of Aalborg, 2005.

[45] P. Kotyczka and D. L. Sergio, “On a generalized port-hamiltonian representation for the control of damped underactuated mechanical systems,” *IFAC Proceedings Volumes*, vol. 45, no. 19, pp. 149 – 154, 2012. 4th IFAC Workshop on Lagrangian and Hamiltonian Methods for Non Linear Control.

[46] I. Goulos, “Modelling the aeroelastic response and flight dynamics of a hingeless rotor helicopter including the effects of rotor-fuselage aerodynamic interaction,” *Aeronautical Journal -New Series-*, vol. 119, 04 2015.

[47] R. Prouty, *Helicopter Aerodynamics.* Rotor & Wing International, PJS Publications, 1985.

[48] G. Padfield, A.-M. Dequin, D. Haddon, K. Kampa, P.-M. Basset, W. GrÃČÅŠnhagen, H. Haverdings, and A. McCallum, “Predicting rotorcraft flying qualities through simulation modelling: A review of key results from garteur ag06.,” 03 2020.

[49] M. Pavel, M. White, G. Padfield, G. Roth, M. Hamers, and A. Taghizad, “Validation of mathematical models for helicopter flight simulators past, present and future challenges,” *Aeronautical Journal*, vol. 117, pp. 343–388, 04 2013.

[50] M. Pavel and R. van Aalst, *On the question of adequate modelling of steady-state rotor disc-tilt for helicopter manoeuvring flight.* Royal Aeronautical Society, 2002.

[51] P. van den Hof and X. Bombois, *Small-Scale Helicopter Automatic Autorotation: Modeling, Guidance, and Control.* PhD thesis, 2015.

[52] H. Yu and P. Antsaklis, “A passivity measure of systems in cascade based on passivity indices,” pp. 2186 – 2191, 01 2011.

Glossary

List of Acronyms

PH	port-Hamiltonian
IDA-PBC	Interconnection and Damping Assignment-Passivity-Based Control
PBC	Passivity-Based Control
PDE	Partial Differential Equation
BET	Blade Element Theory
VTOL	Vertical Take-Off and Landing
UAV	Unmanned Aerial Vehicle
FAA	Federal Aviation Administration
CFR	Code of Federal Regulations
RNN	Reccurent Neural Network
NMPC	Nonlinear Model Predictive Controller
DDP	Differential Dynamic Programming
LQR	Linear Quadratic Regulator
DoF	Degrees-of-Freedom
cog	center of gravity
TPP	Tip-Path-Plane
FBD	Free Body Diagram
RPM	Revolutions Per Minute
PM	Point-Mass
ZFFT	Zero Flight Time Training

List of Symbols

Greek letters

α	Angle of attack
β	Tilt angle
γ	Lock number
λ_c	Non-dimensional inflow velocity of the control plane
λ_i	Non-dimensional induced velocity
μ	Advance ratio
ν	Input constant
Ω	Rotor angular velocity
ϕ_f, θ_f, ψ_f	Roll, pitch and yaw angles
ϕ_{in}	Inflow angle
ψ_{az}	Azimuth angle
ρ	Air density
σ	Rotor solidity
τ	Torque
θ	Blade pitch angle
θ_0	Collective input
θ_{lat}	Lateral cyclic input
θ_{long}	Longitudinal cyclic input
ζ	Damping ratio

Latin letters

A	Rotor disk area
a_0	Coning angle
a_1	Longitudinal flapping angle
b	Wing span
b_1	Lateral flapping angle
C	Coefficient paired with a subscript
c	Chord length
c_{cr}	Critical damping constant
c_c	Damping constant
c_d	2D element drag coefficient
C_L	3D wing lift coefficient
c_l	2D element lift coefficient
D	Drag force
d	Damping injection function
e_k	Force at endpoint of spring
e_m	Velocity of the mass
f_k	Velocity at endpoint of spring
f_m	Force exerted on the mass

F_d	Damping force
f_e	Equivalent flat plate area
g	Gravitational acceleration
H	Hamiltonian
I	Mass moment of inertia
K	Kinetic energy
k	Spring constant
L	Lift
m	Mass
N	Number of blades
p, q, r	Roll, pitch and yaw rate
Q_m	Rotor torque
R	Rotor radius
r	Span-wise distance of element from root
S	Total blade surface area
t	Time
T_m	Thrust
T_t	Tail rotor thrust
U	Potential energy
u, v, w	Body velocity components
V	Resultant velocity
V_e	Energy storage function
w	Energy supply rate

Vectors and Matrices

$\boldsymbol{\beta}(\boldsymbol{\xi})$	Input vector function
$\boldsymbol{\mathcal{I}}$	Identity matrix
$\boldsymbol{\mathcal{J}}$	Interconnection matrix
$\boldsymbol{\mathcal{Q}}$	$\boldsymbol{\mathcal{J}} - \boldsymbol{\mathcal{R}}$
$\boldsymbol{\mathcal{R}}$	Dissipation matrix
$\boldsymbol{\tau}$	Torque vector
$\boldsymbol{\xi}$	State vector
$\boldsymbol{B}$	General input matrix
$\boldsymbol{C}$	Matrix with Coriolis effect
$\boldsymbol{F}$	Force vector
$\boldsymbol{f}$	Input force vector
$\boldsymbol{G}$	Vector considering gravitational forces
$\boldsymbol{g}$	Input matrix
$\boldsymbol{I}$	Inertia matrix
$\boldsymbol{K}$	Constant matrix
$\boldsymbol{M}$	Generalised mass matrix

$\boldsymbol{p}$	Generalised momenta
$\boldsymbol{q}$	Generalised coordinates
$\boldsymbol{u}$	Input vector
$\boldsymbol{y}$	Output vector

Other Symbols

$\boldsymbol{\mathcal{C}}$	Control port
$\boldsymbol{\mathcal{E}}$	Environment interaction port
$\boldsymbol{\mathcal{S}}$	Energy storage port
$\mathcal{H}$	System
$\mathcal{L}$	Lagrangian
*	Setpoint

Subscripts

α	Gradient with respect to the angle of attack
a	Added
c	Constant, control plane
d	Desired
e	Error signal
f	Fuselage
h	Hover
i	Induced
m	Main rotor
rot	Rotor
t	Tail
x, y, z	x, y and z-axes
yy	About the y-axis