SMALL-SCALE HELICOPTER AUTOMATIC AUTOROTATION

MODELING, GUIDANCE, AND CONTROL

Small-Scale Helicopter Automatic Autorotation

Modeling, Guidance, and Control

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op vrijdag 18 september 2015 om 10:00 uur

door

Skander Taamallah

Master of Science in Aeronautics & Astronautics, Stanford University, U.S.A.,
Diplôme d'Ingénieur en Génie Electrique, I.N.S.A. Toulouse, France,
geboren te Tunis, Tunesië.

Considerate la vostra origine: non siete nati per vivere come bruti, ma per praticare la virtù e apprendere la conoscenza.

Dante Alighieri
Divina Commedia, Inferno, Canto XXVI

CONTENTS

SUMMARY

Over the past thirty years, significant progress related to sensors technology and miniaturized hardware has allowed for significant improvements in the fields of robotics and automation, leading to major advancements in the area of flying robots, also known as Unmanned Aerial Vehicles (UAVs). In particular, small-scale helicopter UAVs represent attractive systems, as they may be deployed and recovered from unprepared or confined sites, such as from or above urban and natural canyons, forests, and naval ships. Currently, one of the main hurdles for UAV economic expansion is the lack of clear regulations for safe operations. UAVs operated in the so-called non-segregated airspace, for civilian or commercial purpose, are only approved by airworthiness authorities on a case-by-case basis. A number of complex issues, particularly related to UAV operational safety and reliability, need to be resolved, before seeing widespread use of UAVs for civilian or commercial purposes.

A failure of the power or propulsion unit, resulting in an engine OFF flight condition, represents one of the most frequent UAV failure modes. For the case considered in this thesis, this would mean flying, and landing, a small-scale helicopter UAV without a working engine, i.e. the autorotation flight condition. Helicopter autorotation is a highly challenging flight condition in which no power plant torque is applied to the main rotor and tail rotor, i.e. a flight condition which is somewhat comparable to gliding for a fixed-wing aircraft. During an autorotation, the main rotor is not driven by a running engine, but by air flowing through the rotor disk bottom-up, while the helicopter is descending. The power required to keep the main rotor spinning is obtained from the vehicle's potential and kinetic energies, and the task during an autorotative flight becomes mainly one of energy management. As small-scale helicopter UAVs have higher levels of dynamics coupling and instability when compared to either larger-size helicopter UAVs or full-size helicopter counterparts, performing a successful autorotation maneuver, for such small-scale vehicles, is considered to be a great challenge.

Our research objective consists in developing a, model-based, automatic safety recovery system, for a small-scale helicopter UAV in autorotation, that safely flies and lands the helicopter to a pre-specified ground location. In pursuit of this objective, the contributions of this thesis are structured around three major technical avenues.

First we have developed a nonlinear, first-principles based, high-order model, used as a realistic small-scale helicopter UAV simulation. This helicopter model is applicable for high bandwidth control specifications, and is valid for a range of flight conditions, including (steep) descent flight and autorotation. This comprehensive model is used as-is for controller validation, whereas for controller design, only approximations of this nonlinear model are considered.

The second technical avenue addresses the development of a guidance module, or Trajectory Planner (TP), which aims at generating feasible and optimal open-loop autorotative trajectory references, for the helicopter to follow. In this thesis, we investigate two such TP methods. The first one is anchored within the realm of nonlinear optimal control, and allows for an off-line computation of optimal trajectories, given a cost objective, nonlinear system dynamics, and controls and states equality and inequality constraints. The second approach is based upon the concept of differential flatness and aims at retaining a high computational efficiency, e.g. for on-line use in a hard real-time environment.

The third technical avenue considers the Trajectory Tracker (TT), which compares current helicopter state values with the reference values produced by the TP, and formulates the control inputs to ensure that the helicopter flies along these optimal trajectories. Since the helicopter dynamics is nonlinear, the design of the TT necessitates an approach that tries to respect the system's nonlinear structure. In this thesis we have selected the robust control μ paradigm. This method consists in using a, low-order, nominal Linear Time-Invariant (LTI) plant coupled with an uncertainty, and applying a small gain approach to design a single robust LTI controller. This robust LTI controller has proven to be capable of controlling and landing a helicopter UAV in autorotation. In particular, our simulations have shown that the crucial control of vertical position and velocity exhibited outstanding behavior, in terms of tracking performance. However, the tracking of horizontal position and velocity could potentially be improved by considering some other control methods, such as Linear Parameter-Varying (LPV) ones. To this end, we present an approach that approximates a known complex nonlinear model by an affine LPV model. The practicality of this LPV modeling method is further validated on a pointmass pendulum example, and in the future this LPV method could prove useful when applied to our helicopter application.

To conclude, we illustrate in this thesis—using a high-fidelity simulation of a small-scale helicopter UAV—the first, real-time feasible, model-based optimal trajectory planning and model-based robust trajectory tracking, for the case of a small-scale helicopter UAV in autorotation.

SAMENVATTING

In de afgelopen dertig jaar heeft een aanzienlijke vooruitgang aan sensoren technologie en geminiaturiseerde hardware gezorgd voor belangrijke verbeteringen op het gebied van robotica en automatisering, wat leidt tot grote vooruitgang op het gebied van vliegende robots, ook bekend als onbemande luchtvaartuigen 'Unmanned Aerial Vehicles (UAV's)'. In het bijzonder kleinschalige helikopter UAV's worden gezien als aantrekkelijke systemen omdat zij kunnen worden ingezet vanuit ruwe of begrensde gebieden, zoals van of boven stedelijk gebied, ravijnen, bossen en marineschepen. Op dit moment is één van de belangrijkste hindernissen voor economische expansie van onbemande luchtvaartuigen het ontbreken van duidelijke voorschriften voor veilige operaties. UAV's bediend in een zogenaamd niet-gescheiden luchtruim, voor civiel of commercieel doel, worden alleen goedgekeurd door luchtwaardigheid instanties op een 'case-by-case' basis. Een aantal complexe kwesties, met name met betrekking tot operationele veiligheid en betrouwbaarheid van UAV's, moet worden opgelost voordat er sprake zal zijn van wijdverbreid gebruik van UAV's voor civiele of commerciële doeleinden.

Een fout in het voortstuwing systeem, wat resulteert in een 'motor uit' vliegconditie, vertegenwoordigt één van de meest voorkomende UAV pech gevallen. In het geval beschouwd in dit proefschrift, zou dit betekenen het vliegen en landen van een kleinschalige onbemande helikopter zonder werkende motor, dat wil zeggen de autorotatie vlucht conditie. Helikopter autorotatie is een zeer uitdagende vliegconditie waarbij geen krachtbron is geplaatst op de hoofd - en staartrotor, dat wil zeggen een vliegconditie die enigszins vergelijkbaar is met zweven voor een vliegtuig. Tijdens een autorotatie wordt de hoofdrotor niet aangedreven door een lopende motor, maar door lucht die van onder naar boven door de rotor stroomt, terwijl de helikopter aan het dalen is. De kracht die nodig is om de hoofdrotor draaiende te houden wordt verkregen uit potentiële en kinetische energie van het voertuig, en de taak tijdens een autorotatie vlucht wordt er voornamelijk één van energie management. Aangezien kleinschalige onbemande helikopters hogere niveaus van dynamica, koppeling en instabiliteit hebben in vergelijking met grotere UAV helikopters of grootschalige helikopter tegenhangers, is het uitvoeren van een succesvolle autorotatie manoeuvre voor dergelijke kleinschalige voertuigen, een nog grotere uitdaging.

In dit proefschrift bestaat onze onderzoeksdoelstelling uit het ontwikkelen van een, model-gebaseerde, automatisch veiligheid herstelsysteem voor een kleinschalige onbemande helikopter in autorotatie, dat de helikopter veilig laat vliegen naar, en landen op een vooraf opgegeven locatie op de grond. Bij het nastreven van deze doelstelling zijn de bijdragen van dit proefschrift gestructureerd rond drie belangrijke technische domeinen.

Het eerste betreft het modelleren van de niet-lineaire dynamica van een kleinschalige helicopter. We hebben een niet-lineaire, eerste-principes gebaseerde, hogere-orde model

ontwikkeld, en die wordt gebruikt als een realistische kleinschalige helikopter simulatie-omgeving. Dit helikopter model is toepasbaar voor hoge-bandbreedte regel specificaties, en is geldig voor een scala aan vliegcondities, waaronder (steile) afdaling en autorotatie. Dit uitgebreide model wordt gebruikt voor de regelaar validatie, terwijl voor de regelaar ontwerp slechts benaderingen van dit niet-lineaire model worden beschouwd.

Het tweede technische domein behandelt de ontwikkeling van een sturings module, of 'Trajectory Planner (TP)', die gericht is op het genereren van haalbare en optimale open-lus autorotatieve traject referenties, die de helikopter dient te volgen. In dit proefschrift onderzoeken we twee van zulke TP methoden. Het eerste is verankerd in het domein van de niet-lineaire optimale controle en zorgt voor een 'off-line' berekening van optimale trajecten, gegeven een doelstelling, niet-lineaire systeemdynamica en randvoorwaarden. De tweede benadering, gebaseerd op het concept van differentiële vlakheid, beoogt het behoud van een rekenkundige doelmatigheid, bijvoorbeeld voor 'on-line' gebruik in een harde 'real-time' omgeving.

Het derde technische domein beschouwt het 'Trajectory Tracker (TT)', die de huidige waarden van de staat van de helikopter vergelijkt met de referentiewaarden geproduceerd door de TP, en die de controle ingangen formuleert om ervoor te zorgen dat de helikopter langs deze optimale trajecten vliegt. Aangezien de dynamica van de helikopter niet-lineair is, vereist het ontwerp van de TT een aanpak die probeert de niet-lineaire structuur van het systeem te behouden. Wij hebben in dit proefschrift de robuuste controle μ paradigma geselecteerd. Deze methode bestaat uit het gebruik van een, lagere-orde, nominale Lineaire Tijd-Invariant (LTI) model in combinatie met een onzekerheid en het toepassen van een 'small-gain' aanpak voor het ontwerpen van een enkel robuuste LTI regelaar. Deze robuuste LTI regelaar heeft bewezen in staat te zijn om een onbemande helikopter te kunnen controleren en te laten landen in autorotatie. In het bijzonder blijkt uit onze simulaties dat de cruciale controle van de verticale positie en snelheid uitstekend gedrag vertonen, in termen van het bijhouden van prestaties. Echter, het bijhouden van de horizontale positie en snelheid zou kunnen worden verbeterd door het in overweging nemen van andere controlemethoden, zoals 'Linear Parameter-Varying (LPV)'. Te dien einde presenteren we een aanpak die een bekend complex niet-lineaire model door een 'affine' LPV model wordt benaderd. De uitvoerbaarheid van deze LPV modelleringmethode is verder gevalideerd op een slinger voorbeeld, en in de toekomst zou deze methode nuttig kunnen blijken wanneer toegepast op onze helikopter applicatie.

Tot slot illustreren we in dit proefschrift—met behulp van een hoog betrouwbare simulatie van een kleinschalige onbemande helikopter—de eerste 'real-time' haalbare automatische autorotatie, die gebruik maakt van een model-gebaseerde, optimale 'Trajectory Planner' en robuuste 'Trajectory Tracker'.

PREFACE

Non saranno sempre rose e fiori: it will not always be roses and flowers, was I told by my friend Antonio Telesca, at the start of this PhD thesis, many years ago. Indeed the journey was not always easy, but it did provide me with much intellectual growth and reward. Hence, I would like to take this opportunity to express my sincere gratitude to the people who have made this thesis possible. First, and foremost, I would like to thank my Promotor Professor Paul Van den Hof for giving me this unique opportunity, and privilege, to be a PhD student in a renowned academic group: the Delft Center for Systems and Control (DCSC). Dear Paul, I am extremely grateful for your critical input and insight, and for providing me with invaluable theoretical guidance. Further, thank you so much for creating an environment in which I enjoyed significant academic and organizational freedom. Over the years, I was truly touched by your unlimited patience, and above all by your generosity and warmheartedness.

I owe also immense thanks to my Promotor Professor Xavier Bombois for sharing his profound insight in systems and control theory. Dear Xavier, I cannot overestimate the value of your expert advice all along the course of this thesis. Your ability to see throughout seemingly complex technological problems is truly unique. Thank you also for the uncountable and inspiring discussions we have had throughout the elaboration of this project. You taught me mathematical rigor, while guiding me towards interesting research avenues. Thank you also for your kind friendship, you truly have a heart of Gold.

This work was funded, and therefore made possible, by my employer the National Aerospace Laboratory (NLR) in Amsterdam. I am indebted to the NLR management for supporting this research endeavor. In particular, my immense gratitude goes to my manager René Eveleens, who has been a constant support, and source of encouragement. Dear René, you have this unique ability of getting the best out of people in the workplace. This, combined with your outstanding strategic vision for our department and company, sets you in my eyes as the best leader of NLR. Thank you also for your patience and understanding, and for being such a kindhearted person.

Next, my gratitude goes also to Professor Roland Tóth for the fruitful discussions we had, for sharing with me his great knowledge of Linear Parameter Varying (LPV) systems, and for his enthusiasm and the useful feedback that he provided during the last phase of this thesis. Dear Roland, your tireless effort and attention towards mathematical rigor and detail are truly exceptional and so inspiring.

Further I also would like to thank all members of the jury for all the time and effort spent while proofreading this thesis.

About fifteen years ago, I was a M.Sc. graduate student in the U.S.A. in Aeronautics & Astronautics. It is there that the seeds of this thesis have been planted. My interest in pursuing research, by combining systems and control theory, with Unmanned Aerial Vehicles (UAVs), is no doubt inspired by my education at Stanford. Very exciting research was already taking place in this area, particularly within Professor's Claire Tomlin laboratory (back then at Stanford, now at U.C. Berkeley). Professor's Tomlin openness, hard-working ethic, and dedication towards research and teaching are truly exemplary, and made a tremendous impact on me.

I also want to thank current and former NLR colleagues for the many stimulating and in-depth discussions we had over the years, on state estimation, helicopter dynamics, avionics systems, and general UAV matters. Special thanks are for Jan Breeman, Dr. Martin Laban, Peter Faasse, Harm van Gilst, Nithin Govindarajan, Dr. Jan-Joris Roessingh, Floor Pieters, Jasper van der Vorst, Stefan van 't Hoff, and Jan-Floris Boer.

Aside from faculty and NLR colleagues, there are several students/interns, that I (co-)supervised at NLR and with whom I had fruitful discussions, in particular Jeroen Veerman and Ferdinand Peters (modeling of quadcopter UAVs), Ludovic Tyack (UAV avionics systems and sliding mode control), Floris van de Beek (UAV mechanical systems), Jurriaan Kerkkamp (passivity-based control), and Alexander Macintosh (robust control).

Next, I have to mention the support network of friends. Thank you for being there my buddy Joseph Mayer from N.Y.; further Jasper Braakhuis, Tjeerd Deinum, Professor Albert Menkveld, Jan-Willem Wasmann, Miriam Ryan, and Dr. Giuseppe Garcea from The Netherlands; Dr. Daniele Corona and Dr. Marco Forgione from Italy; Zayd Besbes and Antonio Telesca from France; and Professor Omar Besbes from N.Y. I also have special thoughts for my friends from Stanford with whom I shared a passion for aerospace systems: Ygal Levy, Antoine Gervais, Olivier Criou, and Stéphane Micalet. Finally, to my old friends from I.N.S.A. Toulouse Hervé Walter, Laurent Turmeau, Régis Sanchez, and Jean-Baptiste Saint Supery, it is finally done, James has completed it.

It is fair to say that I owe everything I am, and everything I have ever achieved, to my parents, Latif Taamallah and Dini Bossink. They have given me unrelenting love and unconditional support, and taught me, from a young age, the values of hard work and perseverance. Thank you so very much for everything you have done and given, and for all the sacrifices you have made. To my sister Lilia, thank you so much for your unconditional love and support.

To my godparents Habib and Rose Skouri, and to my dear friends (as close as family) Jean-Paul and Colette Marcellin, you have helped and supported me in so many ways, and you have showered me with care and attention. Thank you so much for everything you have given and done.

To my parents-in-law Wim and Anske van Hunen, and to my sister- and brother-in-law Marinka and Patrick Ledegang, and their lovely and beautiful children Chiara, Kalle, and

Felice, thank you for your constant support, thank you for all the care you have given, thank you for your unlimited generosity, and thank you for all the sacrifices you have made over the years.

And the best for last, to my wife and partner in life Larissa van Hunen, no words can describe my feelings for you. Thank you for your unconditional support throughout this endeavor, thank you for your infinite patience and love, thank you for the countless evenings that I spent working on the thesis, thank you for all the weekends, or parts thereof, that I spent at NLR, or at home, working on this project, thank you for the months of parental leave that we did not have as I used them all to work on the thesis, thank you for all the holidays that we did not have, year in year out, as I dedicated most of them towards the thesis, thank you for all the sacrifices you have made, and finally thank you for the two adorable daughters you gave us, Eliana and Aurelie, the joy of our family. Without you, none of this would have been possible. I love you so very much.

I dedicate this thesis to Latif, Dini, Lilia, Larissa, Eliana, and Aurelie.

Skander Taamallah
Amsterdam, April 2015

1

INTRODUCTION

Begin with the End in Mind.

Stephen R. Covey
The 7 Habits of Highly Effective People, Free Press, 1989

In this Chapter we present the background and motivation for the research addressed in this PhD thesis. We start by a general introduction on the subject of Unmanned Aerial Vehicles (UAVs), helicopter mini-UAV, and helicopter autorotation. Then we formulate the central research objective of this thesis. We conclude this Chapter with the thesis roadmap, and a list of the main contributions.

Parts of this Chapter have been published in [25].

1.1. UNMANNED AERIAL VEHICLES (UAVS)

OVER the past thirty years, significant scientific progress related to sensors technology and computational miniaturized hardware has allowed for sustained improvements in the fields of robotics and automation, leading to major advancement in the area of flying robots, also known as Unmanned Aerial Vehicles (UAVs)[1] [1], see Fig. 1.1. A UAV is further defined as a powered aerial vehicle, not carrying a human operator, that

- Uses aerodynamic forces to provide vehicle lift

- Is expendable or recoverable (in contrast to missile systems)

- May fly autonomously, or may be piloted remotely

- Carries a payload

Unmanned systems are typically associated with the so-called *DDD* missions: *Dull* i.e. long duration, *Dirty* i.e. sampling for hazardous materials, and *Dangerous* i.e. extreme exposure to hostile action [2].

Figure 1.1: Two small drones, Insitu's Scan Eagle X200 and AeroVironment's PUMA—both weighing less than 25 kg and having a wingspans of approx. 3 m—have become the first certified UAVs, by the Federal Aviation Administration (FAA), for civilian use in the USA. They will operate off the Alaska coast to survey ice floats and wildlife, and to conduct commercial environmental monitoring in the Arctic Circle, and further assist emergency response teams in oil spill monitoring and conduct wildlife observations. Huffington Post, July 2013.

---

[1]Although recently industry and the regulators have adopted Unmanned Aerial System (UAS) as the preferred term for unmanned aircrafts, as the UAS term encompasses all aspects of deploying such vehicles, and hence not just the vehicle platform itself.

1.1.1. Candidate applications

UAVs have been developed for both civilian and military missions. Examples of such applications in the civilian sector include: agricultural fertilizer dissemination, animal density determination, area illumination, area mapping, area pollution measurements, communication relay, dam observation, flooded areas and forest fires inspection, object delivery, oil spills detection, power line and pipeline inspection, radioactivity measurement, searching for missed or shipwrecked persons, sports and cultural event transmission, traffic surveillance, video and film industry, volcano observation, and weather forecast [3].

In the military sector, UAVs have been around for a long time. Actually pilot-less aircrafts, whether as aerial targets or for more belligerent purposes, have a history stretching back to World War I. A multitude of candidate military missions could be performed by unmanned systems. Some could be performed by a single UAV vehicle, whereas others could necessitate a co-operative engagement of several UAVs. A non-exhaustive shortlist of candidate missions is given here: Battle Damage Assessment (BDA), border monitoring, Intelligence Surveillance and Reconnaissance (ISR), miniature scout helicopter (team with attack helicopter), naval gunfire support, precision strike and Suppression of Enemy Air Defenses (SEAD), range safety monitor, Search And Rescue (SAR) operations, support to special operations forces, and surface search and correlation [2].

1.1.2. Markets

Several UAV markets exist, i.e. the military market, the civilian government market, and the civilian commercial market, with a current worldwide UAV expenditures of $5.2 billion [4]. The military and civilian government markets contain a small number of customers that potentially may buy a large amount of unmanned systems, whereas the civil commercial market is defined by a larger number of customers which are interested in buying only a small number of systems [5]. The military market developed first due to the operational advantages of UAVs, the civil government market followed next as it was driven by security needs (law enforcement, and fire and rescue agencies), and recently the civilian commercial market has started to expand.

1.1.3. Development and acquisition programs

On a worldwide stage, there are nowadays at least 40 to 50 nations involved in at least one UAV development and/or acquisition program, resulting in a total of over 600 UAV programs [6], with approximately 20% of which are rotary-wing vehicles, see Fig. 1.2 and Fig. 1.3. The U.S.A., Israel, and France represent the three major players in this UAV arena, combining more than half of worldwide UAV development and acquisition programs; although other countries, such as China and others in South-East Asia, have been heavily investing in this sector for the past few years. About two thirds of the worldwide systems have the military as an end-user, the remaining systems being dedicated to civilian or Research and Development (R&D) programs in academia and research institutions. Based upon the Maximum Take-Off Weight (MTOW), approximately half of the developed systems fall into one of the three following categories [6]: micro-UAV (MTOW < 5 kg), mini-UAV (MTOW < 30 kg), or close-range UAV (MTOW < 150 kg).

Figure 1.2: The MQ-8B Fire Scout rotary-wing UAV approaches the frigate USS McInerney. US Navy photo.

Figure 1.3: Delft Dynamics's RH4 Spyder quadcopter UAV. Photo from [7].

1.1.4. AIRWORTHINESS AND SAFETY ASPECTS

Currently one of the main hurdles for UAV economic expansion is the lack of clear regulations for safe operations. So far, an internationally accepted regulatory basis for UAV operations does not yet exist [5, 8], although many efforts are underway [9, 10]. This said, UAVs operated by the military, police, and fire brigades are so-called Operational Air Traffic, meaning that they do not abide to the International Civil Aviation Organization (ICAO) rules. Especially, for cases involving emergencies or crises, UAVs may benefit from exemptions from civil regulations. However, UAVs operated in so-called non-segregated airspace[2], for civilian or commercial purposes, do not inherit these advantages. In general, airworthiness authorities tend to be rather cautious, and for good reasons, when evaluating the insertion of UAVs into civilian airspace. The reliability of UAVs has been a concern for

---

[2]For instance a country's national airspace.

many years, due to the high accident rates [11]. For instance, the reliability of UAVs would need to improve by one to two orders of magnitude, in order to reach an equivalent general aviation[3] safety level [11, 12]. Hence, it is clear that an increase in UAV system integrity, reliability, and safety could only facilitate the introduction of UAVs into non-segregated airspace for civilian or commercial purposes. In fact, a safety analysis would need to address each part of the UAV system, from the structural integrity of the vehicle, its engine and electronics, to the data links and embedded software.

1.2. THE HELICOPTER

In some cases, UAV deployment and recovery from unprepared or confined sites may be required, such as when operating from or above urban and natural canyons, forests, or from naval ships. These specific missions would require very versatile flight modes, such as vertical takeoff/landing, hovering, and longitudinal/lateral flight. Here, a helicopter UAV capable of flying autonomously, in and out of such restricted areas, would represent a particularly attractive asset. Hence, in the sequel, we briefly review some helicopter concepts.

The four forces acting on a helicopter are denoted by: thrust, drag, lift and weight, see Fig. 1.4. The thrust overcomes the force of drag; the drag is a rearward force caused by the disruption of airflow by the moving rotors and vehicle; lift is produced by the dynamic effect of the air flowing on the main rotor blades, opposing the downward force of the vehicle weight. On a standard helicopter configuration, the tail rotor is a small rotor, traditionally mounted vertically at the end of the tail-boom of a helicopter. The tail rotor's thrust, multiplied by the distance from the vehicle's center of gravity, allows it to counter the torque effect created by the main rotor, see Fig. 1.5. A typical helicopter has four separate flight control inputs, which allow to control the attitude—roll, pitch, and yaw angles, see Fig. 1.6—of the helicopter.
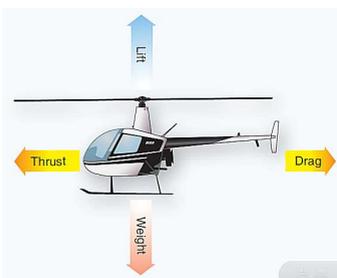
Figure 1.4: The four forces acting on a helicopter. Picture from [13].
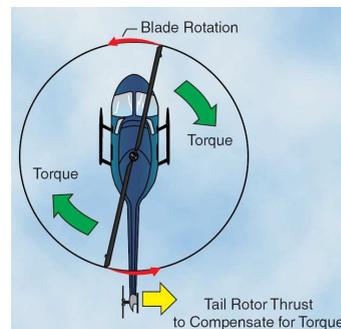
Figure 1.5: Top view of a counter-clockwise rotating main rotor. Picture from [14].

---

[3]Roughly speaking, general aviation refers to all civil aviation operations other than scheduled air services (i.e. other than commercial airlines). General aviation flights range from gliders and powered parachutes to corporate jet flights.
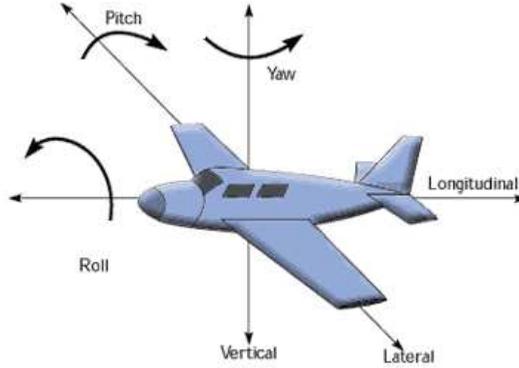
Figure 1.6: Attitude angles and control axis of an aerospace vehicle. Picture from [15].

The controls are known as main rotor collective, main rotor longitudinal cyclic, main rotor lateral cyclic, and tail rotor anti-torque pedals, see Fig. 1.7.
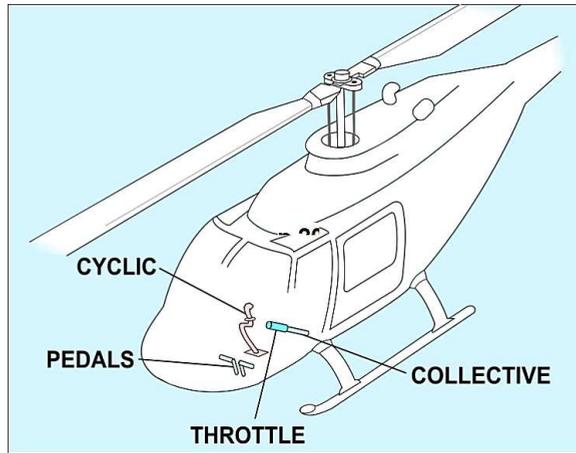
Figure 1.7: Helicopter flight controls. Picture from [16].

Some smaller helicopters have also a manual throttle needed to maintain rotor speed. The main rotor collective changes the pitch angle of all main rotor blades collectively, and independently of the blade rotational position. Through the collective, one can increase or decrease the total lift derived from the main rotor. On the other hand, the main rotor cyclics change the pitch angle of the main rotor blades cyclically, i.e. the pitch angle of the rotor blades changes depending upon their position, as they rotate around the main rotor hub [16]. For example in Fig. 1.7, pushing the cyclic forward results in a pitch-down of the helicopter, and consequently produces a thrust vector in the forward direction. If the cyclic is moved to the right, the helicopter starts rolling to the right and produces thrust in

that direction, causing the helicopter to move sideways [16]. The anti-torque pedals change the pitch of the tail rotor blades. The anti-torque pedals allow to increase or decrease the thrust produced by the tail rotor, causing the nose of the vehicle to yaw. For each control input channel, Table 1.1 summarizes the primary, and secondary, impacts on the vehicle response.

Table 1.1: Typical input-output coupling, for a helicopter with a single main rotor (derived from [17]).

Input Axis	Response			
	Roll (ϕ)	Pitch (θ)	Yaw (ψ)	Climb/Descent (w)
Main rotor collective (θ_0)	Due to transient & steady lateral flapping & sideslip	Due to transient & steady longitudinal flapping	Power change varies requirement for TR thrust	Prime response
Main rotor lateral cyclic (θ_{1c})	Prime response	Due to longitudinal flapping	Undesired (especially in hover)	Descent with roll angle
Main rotor longitudinal cyclic (θ_{1s})	Due to lateral flapping	Prime response	Negligible	Desired in forward flight
Tail rotor collective ($\theta_{0_{TR}}$)	Roll due to TR thrust & sideslip	Negligible	Prime response	Undesired, due to power changes in hover

1.2.1. Helicopter mini-UAVs

In many cases small size and low purchase cost, of the helicopter UAV, represent the primary driving system specifications. In these situations helicopter mini-UAVs, see Fig. 1.8, provide clear inherent strengths, albeit at the cost of decreased capabilities, when compared to the larger-size helicopter UAVs [18, 19]. Helicopter mini-UAVs can even be deployed in large numbers, at an acceptable cost. Briefly summarized, helicopter mini-UAVs are commonly upgraded from Remote-Controlled (RC) hobby helicopters, by assembling an avionics suite. The role of this avionics suite is to collect and integrate various measurement signals, drive the actuators, provide communications with a Ground Control Station (GCS), and support real-time operations of autonomous flight control laws [20]. Helicopter systems can be characterized as Multiple-Input Multiple-Output (MIMO), under-actuated, nonlinear, and unstable dynamics[4]. In addition helicopter mini-UAVs[5], when compared to their full-size helicopter counterparts, or even to larger-size helicopter UAVs (i.e. in the

---

[4]And time-varying in some cases, e.g. when a gasoline engine is used, implying fuel consumption and hence vehicle mass variation.

[5]In this thesis, the terms *helicopter mini-UAV*, and *small-scale helicopter UAV*, are used interchangeably.

Figure 1.8: NLR's mini-UAV project (2004-2006) based on a modified Bergen Industrial Twin helicopter.

close-range UAV class), feature an increased power-to-mass ratio, an increase in stiffness of the main rotor assembly, and a higher torque-to-inertia ratio. Consequently, small-scale helicopter UAVs are much more agile, and have higher levels of dynamics coupling and instability, than larger-size helicopters [21].

1.2.2. HELICOPTER MAIN ROTOR HUBS
For the case of a fully articulated main rotor system, each rotor blade is attached to the rotor hub through a series of hinges, which allow each blade to move independently of the others, see for example Fig. 1.9 for the case of a full-size helicopter main rotor hub. The flap hinge allows the blade to move in a plane containing the blade and the rotor shaft; the lag hinge allows the blade to move in the plane of rotation; whereas the pitch hinge allows the blade to rotate about its pitch (feathering) axis.

For small-scale helicopters, the rotor hub generally includes a pitch hinge close to the shaft, and a lead-lag hinge[6] further outboard. Besides the hub is typically not equipped with a flap hinge, this latter is often replaced by stiff rubber rings, hence a so-called hingeless flap mechanism, see Fig. 1.10. But for the purpose of helicopter flight dynamics modeling, it is standard practice to model a hingeless rotor (and its flexible blades) as a rotor having rigid blades attached to a virtual hinge [23], this latter being offset from the main rotor axis. This virtual hinge is often modeled as a torsional spring, implying stiffness and damping[7].

1.3. HELICOPTER AUTOROTATION
As discussed in Section 1.1.4, the overall system safety of unmanned systems has to be improved, if not guaranteed, in order to prevent harms to humans and materials, and to allow for sustained helicopter UAVs expansion into the civilian market segment. For unmanned

---

[6]On small-scale helicopters this is technically not a hinge, rather we refer here to the blade fixation bolt.
[7]Adjusting the virtual hinge offset distance, stiffness, and damping, allows to recreate the correct blade motion in terms of amplitude and frequency [24].
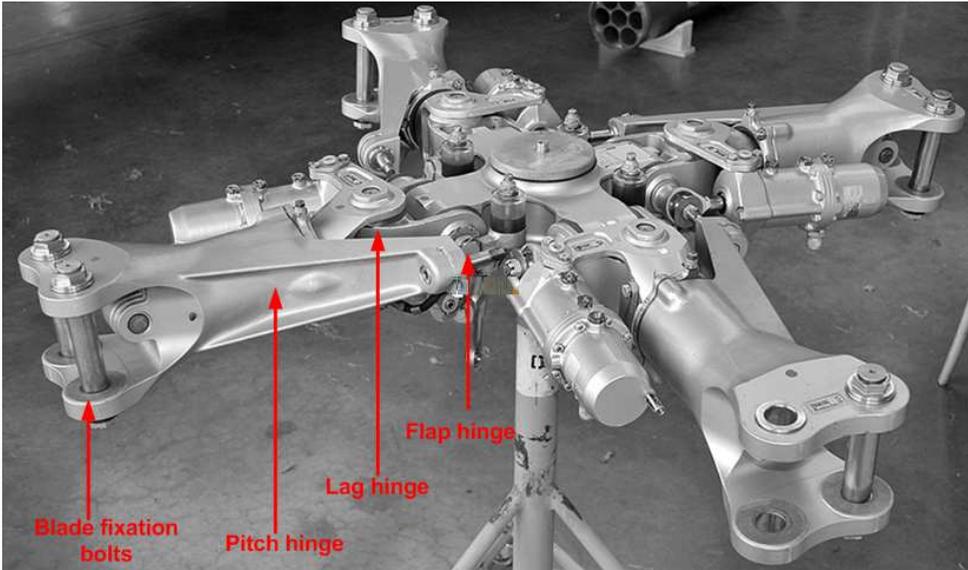
Figure 1.9: Agusta-109 fully articulated 4-blades main rotor. Photo from [22].
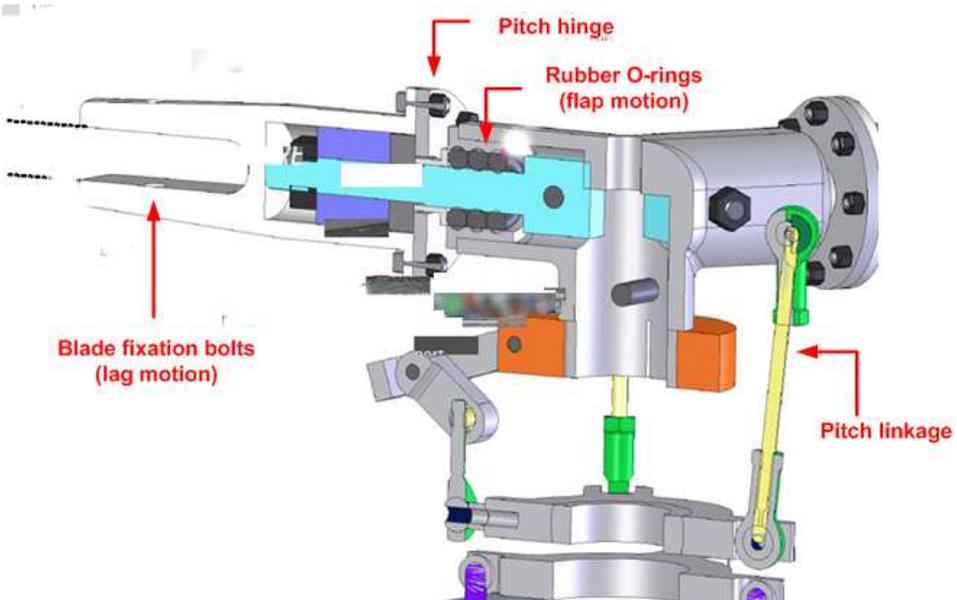
Figure 1.10: NLR's Facility for Unmanned ROtorcraft REsearch (FURORE) project. Typical main rotor hub for a (small-scale) UAV helicopter.

systems, a failure of the power or propulsion units represents currently the most frequent failure mode of the vehicle, accounting for more than a third of all failure events [11]. For a helicopter, such failures would mean flying and landing the vehicle without a working engine, which is also known as the *autorotation* flight maneuver in helicopter jargon.

1.3.1. AUTOROTATION: A THREE-PHASES MANEUVER

Helicopter power-OFF flight, or autorotation, is a condition in which no power plant torque is applied to the main rotor and tail rotor, i.e. a flight condition which is somewhat comparable to gliding for a fixed-wing aircraft. During an autorotation, the main rotor is not driven by a running engine, but by air flowing through the rotor disk bottom-up, while the helicopter is descending [25, 26]. In this case, the power required to keep the rotor spinning is obtained from the vehicle's potential and kinetic energy, and the task during an autorotative flight becomes mainly one of energy management [27]. An autorotative flight is started when the engine fails on a single-engine helicopter, or when a tail rotor failure requires engine shut-down. Unfortunately, autorotation maneuvers are known to be difficult to perform, and highly risky. From a flight maneuver standpoint, a complete autorotation generally contains three phases [28–32], detailed below[8]

- **The entry**. First, the tail rotor thrust needs to be reduced to account for the loss of main rotor torque (since not driven anymore by an engine). Next a reduction of main rotor thrust, as to prevent main rotor blade stall[9] and rapid decay in main rotor Revolutions Per Minute (RPM), is often required. In addition, it is recommended to pitch the helicopter nose down in order to gain some forward airspeed. Indeed, attaining higher airspeed avoids entering the so-called Vortex-Ring-State (VRS)[10] [25], and allows for a buildup of rotor RPM while lowering the helicopter vertical sink rate.

- **Steady autorotation**. This is the stabilized autorotation, at a constant main rotor RPM, in which the helicopter also descends at a constant rate, which may be chosen for minimum rate of descent, or maximum glide distance. Here, some rotor blade stations on the main rotor will absorb power from the air, whereas others will consume power, such that the net power at the main rotor shaft is zero, or sufficiently negative to make up for losses in the tail rotor and transmission system [33, 34].

- **Flare for landing**. The purpose of the flare is to reduce the sink rate, reduce forward airspeed, maintain or increase rotor RPM, and level the attitude for a proper landing, i.e. achieve appropriate tail rotor ground clearance. The helicopter flare capability is the most important of the three autorotation phases [35, 36], and depends particularly on a high main rotor kinetic energy, which requires a high main rotor RPM and/or a large main rotor blade moment of inertia.

---

[8]Although the precise characteristics of the autorotation maneuver depends upon the initial flight condition, i.e. the helicopter flight condition just prior to the engine OFF situation [27].

[9]Stall corresponds to a sudden reduction in lift coupled with a large increase in drag.

[10]Briefly summarized, the VRS corresponds to a condition where the helicopter is descending in its own wake, resulting in a chaotic and dangerous flight condition.

1.4. PROBLEM FORMULATION

First, we summarize the following observations

- In order to support the economic growth of the small-scale helicopter UAV market, particularly within the civilian segment, the overall UAV system safety has to be improved, especially when considering the case of engine failure. This requires for an autorotative flight capability of the unmanned helicopter system[11].

- An autorotation maneuver is a highly challenging flight maneuver for a helicopter. For the case of manned helicopters, it is long known that a good deal of pilot training is required if disaster is to be avoided. In fact, quick reaction and critically timed control inputs by the pilots are required for a safe autorotative landing [37–40]. The autorotative flight maneuver is actually so risky that full touchdown autorotations (i.e. including flare and landing), as a training scenario, are nowadays very rarely practiced by pilots. It is even reported in [41] that both the U.S. Army and U.S. Air Force have stopped practicing full autorotation flights due to the high level of injuries and vehicle damage.

- As pointed out in Section 1.2.1, small-scale unmanned helicopters have higher levels of dynamics coupling and instability, when compared to larger size UAVs or to full-size counterparts. Hence, for such small-scale unmanned systems, performing a successful autorotation maneuver becomes even more problematic.

The here-above observations and challenges have inspired the following central problem formulation, or research objective, for this thesis

> For the case of a small-scale helicopter UAV in un-powered flight, develop a model-based automatic safety recovery system that safely flies and lands the helicopter to a pre-specified ground location.

1.5. ANALYSIS OF AVAILABLE OPTIONS

A general solution framework to the research objective, formulated here-above in Section 1.4, is depicted in Fig. 1.11. The 'Helicopter Dynamics' block refers to the helicopter experimental system, which is interfaced through various 'Actuators' and 'Sensors'. Here, signal $\mathbf{u_{act}}$ refers to the output of the actuators, whereas measurement signal $\mathbf{y}$ refers to the output of the sensors, generally a subset of the helicopter internal state variables (or state-vector) $\mathbf{x}$. The aim of the 'Optimization' block consists in generating signal $\mathbf{u}$, using the measured signal $\mathbf{y}$, such that a cost function (i.e. the objective formulated here-above in Section 1.4) can be optimized, while enforcing various environmental and vehicle physical constraints. We also know, from previous research on small-scale helicopter UAVs [42–46], that the feedback loop, in Fig. 1.11, has to be run at a relatively high rate for good system performance, i.e. at least 50 Hz or preferably higher.

---

[11]Due to cost factors, most small-scale helicopter UAVs are single-engine.
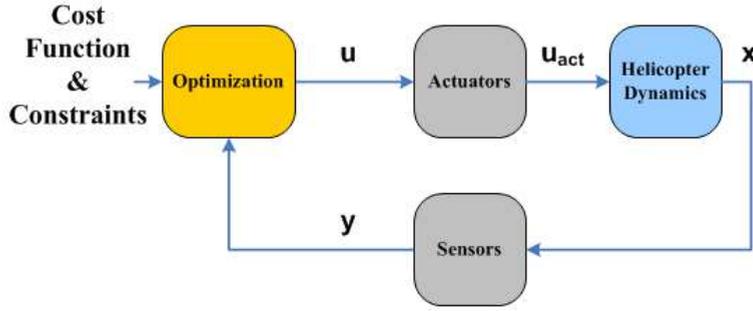
Figure 1.11: Small-scale helicopter UAV automatic autorotation: the feedback loop.

1

To this end, the 'Optimization' block, in Fig. 1.11, has to perform, at least, the following three tasks [47]: 1) *Navigation*, by determining the current position, orientation, and velocity of the helicopter, delivering the filtered state-vector x_{filt} in Fig. 1.12; 2) *Guidance*, by computing the trajectory or path[12] to the destination point; and 3) *Control* by ensuring that the helicopter stays on the computed trajectory or path. Although there is quite a bit of synergism between these three disciplines, a natural separation does exist between the *Navigation* task on the one hand, and the *Guidance* and *Control* tasks on the other.
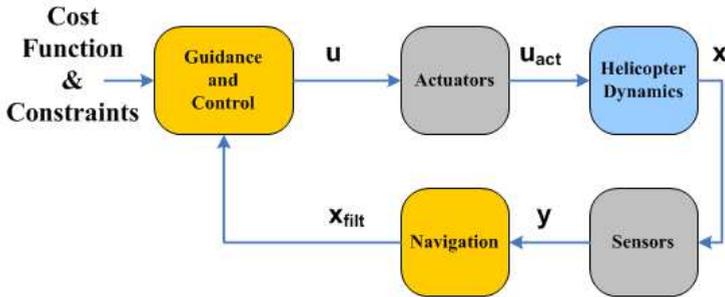
Figure 1.12: Small-scale helicopter UAV automatic autorotation: Guidance, Navigation, and Control (GNC) feedback loop.

1.5.1. MODEL-FREE VERSUS MODEL-BASED OPTIONS

Now, as hinted upon in Fig. 1.11, the goal of this thesis is set upon the design and evaluation of the 'Optimization' block. More specifically, the focus shall be upon the *Guidance* and *Control* tasks, as shown in Fig. 1.12. Before discussing further the content of this thesis, let us first briefly review what are, to-date, the various available options, in terms of *Guidance*

---

[12]The term *trajectory* denotes a route that a vehicle should traverse as a function of time, whereas a *path* denotes an obstacle-free route without temporal restrictions [48].

and *Control*, for our UAV application. First, the *Guidance* and *Control* tasks, in Fig. 1.12, can be designed using

- A **model-free** approach. Various methods are here available, e.g. model-free fuzzy logic[13] [49], with applications to UAV control in [50, 51]; model-free reinforcement learning[14] [52], with applications to UAV control in [50, 53–55]; and evolutionary and genetic algorithms[15] [56–58], with applications to UAV control in [59–63].

- A **model-based** approach, where a model of the helicopter system is made available. There are three different philosophies that form the basis of modeling, namely the white-box modeling (also known as mechanistic or first-principles models), the black-box modeling (also known as empirical models), and the gray-box modeling (also known as hybrid models [64]) which is a mixing of the previous two [65]. In the first case, a model is developed on the basis of detailed understandings of the generic underlying physical laws, that govern the system. In the second case, a model is developed on the basis of empirical knowledge, i.e. a sufficiently large number of consistent observations [65, 66]. In the third case, a model is developed by combining the strengths of the previous two approaches. A rather wide spectrum of model-based approaches exists, which will be discussed in more detail in the sequel.

1.5.2. Integrated versus segregated options
Next, the *Guidance* and *Control* tasks, in Fig. 1.12, can be designed using

- An **integrated** approach, where the *Guidance* and *Control* tasks are performed within a single optimization process. Again, either a model-free or model-based approach can be applied. For model-free approaches, these are identical to the ones listed hereabove. For model-based approaches, we distinguish between the following three options

  1. The first one is the so-called **Model Predictive Control (MPC)** theory [67, 68], also known as Receding Horizon Control (RHC)[16]. Starting with the early works in [69–73], the MPC has become one of the most popular tools for constrained industrial control applications. Based upon a model of the system, an MPC controller generates an optimal state feedback control sequence, by minimizing, at each time step, an open-loop, quadratic performance objective, while explicitly including input and state operating constraints [74–78]. Specifically, for each new measurement, the MPC predicts the future dynamic behavior of the system over a prediction horizon T_p, and determines the input sequence over a control horizon T_c, with $T_c \leq T_p$, such that the performance objective is minimized. Then the first control input of the computed optimal sequence is

---

[13]Fuzzy control is a method based upon a representation of the knowledge, and the reasoning process, of a human operator [49].
[14]Reinforcement learning is an area of machine learning, concerned with how a system ought to respond, in an environment, so as to maximize some notion of cumulative reward [52].
[15]Evolutionary and genetic algorithms use mechanisms inspired by biological evolution [56–58].
[16]The *receding horizon* terminology corresponds to the behavior of the Earth's horizon, i.e. as ones moves towards it, it recedes, hence remaining a constant distance away.

applied to the system, and the optimization is repeated at each subsequent time step. Obviously, lowering the prediction horizon T_p allows to lower the computational time (at the cost of complications with respect to stability). This mechanism of having a new on-line solution at each time step, results in a so-called sampled-data feedback law [79, 80], hence bringing alongside the classical benefits of feedback. Now depending on the nature of the model, either linear or nonlinear, a corresponding linear or nonlinear MPC optimization problem has to be solved. An array of applications of linear MPC to various UAVs can be found in [81–84], whereas specific applications of nonlinear MPC to helicopter UAVs can be found in [85–90], and to fixed-wing UAVs in [91–96].

2. The second option assumes that the nonlinear helicopter plant can be modeled as a Linear Parameter Varying (LPV) system. The latter can thus be used with one of the many **MPC-LPV**, i.e. MPC for LPV algorithms [97–113]. This MPC-LPV approach, most often resulting in a Semi-Definite Program (SDP) optimization, can be seen as a middle-way between the linear and nonlinear optimization paradigms.

3. The third option extends the framework of MPC, for the case of infinitely long horizons T_p and T_c, and naturally brings us to the field of **constrained optimal control** [114–116]. Here too, based upon a model of the system, and given a performance objective (which need not be quadratic), and suitable input and state operating constraints, the solution to the optimal control problem yields the optimal input and state time histories. Again, the first control input of the computed optimal sequence is applied to the system, and the optimization is repeated at each subsequent time step. Also, depending on the nature of the model, either linear or nonlinear, a corresponding linear or nonlinear constrained optimal control problem is solved. Applications of nonlinear optimal control[17] to helicopter UAVs can be found in [117, 118], and to fixed-wing UAVs in [119–123].

- A **segregated** approach, in which the *Guidance* and *Control* tasks are split into two distinctive optimization processes. This approach separates the *Guidance* task, i.e. the Trajectory Planning (TP), from the *Control* task, i.e. the Trajectory Tracking (TT)[18]. Although potentially sub-optimal, this philosophy offers the advantage of effectively exploiting the nonlinear nature of the system (to generate trajectories), while also making use of the linear structure of the error dynamics (to stabilize and control the helicopter) [124]. This divide-and-conquer strategy is also known as the classical two-degree of freedom Flight Control System (FCS) paradigm, as depicted in Fig. 1.13. Here, the TP shall be capable of generating open-loop, feasible, and optimal autorotative trajectory references $\mathbf{x_{TP}}$, for the small-scale helicopter, subject to system and environment constraints, and additionally though not necessarily, the feedforward nominal control inputs $\mathbf{u_{TP}}$, needed to track these trajectories. On the other hand the TT shall compare current estimated state values $\mathbf{x_{filt}}$ with the reference

---

[17]Most often applied in open-loop, rather than in the closed-loop setting described here.
[18]Within this thesis, the terms 'Trajectory Planning' (resp. 'Trajectory Tracking') and 'Trajectory Planner' (resp. 'Trajectory Tracker') are used interchangeably.
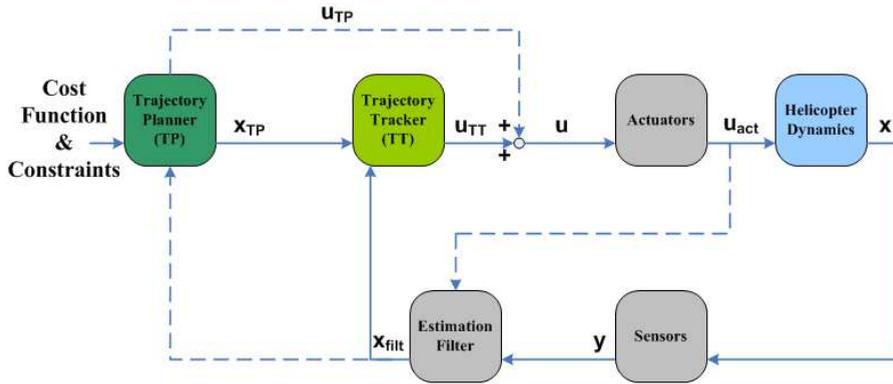
Figure 1.13: Two degree of freedom Flight Control System (FCS) architecture, implemented on the true helicopter system.

values $\mathbf{x}_{TP}$ produced by the TP, and shall formulate the feedback controls $\mathbf{u}_{TT}$ to ensure that the helicopter flies along these optimal trajectories. The additional feedback path, denoted by a dashed line in Fig. 1.13, allows for updating the generated trajectory based upon the current state. In Fig. 1.13, the 'Helicopter Dynamics' block refers to the helicopter experimental system. The role of the *Navigation* task, defined as the 'Estimation Filter' in Fig. 1.13, shall be to estimate the helicopter unmeasured states, the wind, and low-cost sensors characteristics such as scale factors and biases.

THE SEGREGATED APPROACH: TRAJECTORY PLANNING (TP) AND TRAJECTORY TRACKING (TT)
With regard to the segregated approach, let us now separately address the various options available for the *Guidance* task, i.e. Trajectory Planning (TP), and the *Control* task, i.e. Trajectory Tracking (TT).

- Over the years, researchers have addressed the **Trajectory Planning (TP)** problem through several techniques, namely: cell decomposition, potential fields, roadmaps and hybrid systems, inverse dynamics and differential flatness, Mixed Integer Linear Programming (MILP), MPC, optimal control, and finally evolutionary/genetic algorithms [125, 126], with specific benefits and drawbacks for each method, see also [127–129]. Some of the aforementioned planning techniques—cell decomposition, potential fields, and roadmaps—either ignore the differential constraints associated with the vehicle's dynamics (i.e. are model-free approaches), or use simplified kinematic models. With regard to the TP of a helicopter in autorotation, model-based indirect optimal control methods have been used in [130–135], whereas model-based direct optimal control methods have been explored in [37, 38, 136–145]. Aside from these optimal control strategies, three other methods have also been investigated for helicopter autorotation: 1) a model-free learning-based approach in [51, 146]; 2) a model-based parameter optimization scheme to find a backwards reachable set leading to safe landing in [147, 148]; and 3) and a model-free parameter optimization

scheme generating segmented routes, selecting a sequence of straight lines and curves in [149–151].

- With respect to the **Trajectory Tracking (TT)**, virtually any control methods can be applied to a helicopter UAV. For instance, for the specific case of TT for a helicopter with the engine ON, a vast array of technical avenues have been investigated over the years, with the application of: classical control [152], gain-scheduling of Proportional-Integral-Derivative (PID) controllers [153], Linear Quadratic Regulator (LQR) [154, 155], Linear Quadratic Gaussian (LQG) [155, 156], LPV [157], H_2 [158], H_∞ [43, 158–160], μ [157, 161], (nonlinear) MPC [87, 89, 155], feedback linearization, (incremental) nonlinear dynamic inversion and nested saturated control [20, 161–163], adaptive control [164–167], backstepping [166, 168–170], and model-based learning approaches [171–174]. For additional results relative to fuzzy logic-based controllers, artificial Neural Network (NN), or vision based controllers, refer also to [18, 175]. Conversely, very few papers have addressed the subject of helicopter TT with the engine OFF (i.e. autorotation), while concurrently validating their results by experiments, or three-dimensional (3D) high-fidelity simulations. In [146], a model-based Differential Dynamic Programming (DDP)[19] method is used; in [151] a model-based Nonlinear Dynamic Inversion (NDI) with PID loops is used; in [51] a model-free fuzzy logic method is used; and in [149, 177] a model-based H_∞ method is used. Finally, none of the previous results, except for [177] which used a 2D lower-fidelity model, did consider a robust TT approach.

1.5.3. Summary of previous analysis
Summarizing the previous discussion, wee make the following comments.

- Although very powerful and potentially very promising, model-free (machine learning) approaches have also some liabilities. First, the lack of a model makes it difficult to analyze their stability and robustness characteristics [49]. Second, the computational complexity of the model-free approaches may often be prohibitive for our application (recall that the feedback loop in Fig. 1.11 has to be run at a relatively high rate, at least 50 Hz or equivalently 20 msec).

- From a conceptual viewpoint, an integrated model-based approach may potentially provide the best answer to our helicopter autorotation problem. This said, it is essentially the linear MPC approach that has shown to be implementable on-line, even for high bandwidth systems [178–181]. As stated in Section 1.2.1, a helicopter has an intrinsically nonlinear behavior, which renders the application of linear MPC rather questionable. For the case of nonlinear MPC or nonlinear constrained optimal control, these methods are still time-consuming optimization techniques, currently unlikely to be run on-line, within a 20 msec time frame.

- Although potentially much faster than a nonlinear MPC approach, the integrated model-based MPC-LPV approach, with todays SDP solvers, would unlikely run within the 20 msec time frame. This said, this comment should not be taken as

---

[19]DDP is an extension of the Linear Quadratic Regulator (LQR) formalism for non-linear systems [176].

conclusive on the viability of the MPC-LPV method. Indeed, a great deal of current MPC research is devoted to reducing the computational cost [182, 183]. In fact, a clear trend of the last ten years is to move off-line as much computational burden as possible. One such approach is the so-called explicit MPC [184–187], which has shown to be an attractive solution, but so-far (and to the best of our knowledge) only for low-order systems. However, we do expect a bright future for the integrated, model-based, MPC-LPV approach.

- For the Trajectory Planning (TP), model-free approaches (or alternatively model-based approaches using a simplified kinematic model) may lead to infeasible[20] planning results or, at best, conservative solutions. In addition, failing to incorporate some (sufficiently) realistic vehicle dynamics, during the planning phase, will increase the on-line workload of the TT.

- For the Trajectory Tracking (TT), it is best practice to include some form of robustness during the controller design.

- Only four publications have addressed the aggregated planning and tracking functionalities, for a helicopter in autorotation, with validation through either experiments, or 3D high-fidelity nonlinear simulations [51, 146, 149, 151]. The contribution in [146] has shown successful experimental demonstrations, whereas the other three contributions have been validated on 3D high-fidelity simulations. The methods in [51, 146] use a model-free, learning-based TP approach. For the TT, [146] uses a model-based DDP approach, whereas [51] uses a model-free fuzzy logic approach. The methods in [149, 151] use a model-free, (modified) Dubin procedure (i.e. a sequence of straight lines and curves), for their TP algorithms. For the TT, [151] uses a model-based combined NDI-PID method, whereas [149] uses a model-based H_∞ method.

- The results from [51, 151] are for the case of a full-size helicopter, whereas the results in [149] involve a so-called short-range/tactical size helicopter UAV (approximately 200 kg). Only the results in [146] are for a small-scale helicopter UAV. As outlined earlier, when compared to larger and heavier helicopter vehicles, the control of small-scale helicopters (i.e. under 10–20 kg) represents a much more challenging problem.

1.6. RESEARCH OBJECTIVES AND LIMITATIONS

Based upon the previous discussion, we define the following objectives for this thesis, refer also to Fig. 1.14:

1. A model-based TP approach shall be selected, allowing to compute trajectories which are potentially less conservative than the ones originating from model-free approaches.

2. A model-based, robust, TT approach shall be selected, in order to obtain a closed-loop system which is less sensitive to modeling uncertainties.

---

[20]This is precisely the reason why nonholonomic constraints, i.e. constraints that not only involve the state but also state derivatives, which cannot be eliminated by integration, play a crucial role in the subsequent design of feedback controllers [127].
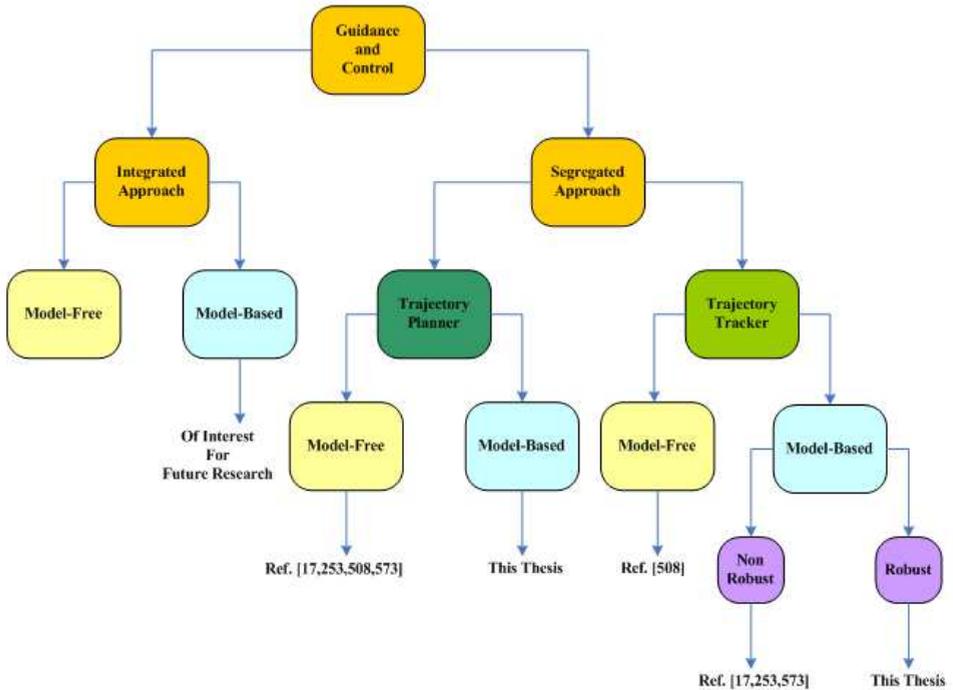
Figure 1.14: Helicopter autorotation: available options for the Guidance and Control.

3. The combined TP and TT shall be computationally tractable, i.e. to be run within a 20 msec time frame.

We also limit the scope of this thesis, by adding the following boundaries:

1. The combined TP-TT shall not be validated experimentally, but rather on a 3D high-fidelity helicopter UAV simulation, serving as a proxy for the real helicopter system.

2. The effects of sensors, actuators[21], and the 'Estimation Filter', are excluded from the simulation environment.

With this in mind, the control architecture, defined in Fig. 1.13, becomes the one defined in Fig. 1.15, where the output signal **y** represents now a subset of the state-vector **x**.

1.7. SOLUTION STRATEGY
Here, we briefly introduce the research areas addressed within this thesis.

---

[21]The actuators are indeed not included in the simulation. However, for a realistic control design, we do include the actuators characteristics into the control design specifications.
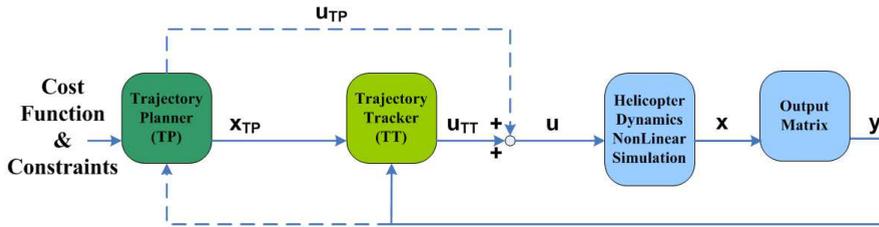
Figure 1.15: Two degree of freedom control architecture, as implemented in this thesis, within a simulated environment.

1.7.1. MODELING OF THE NONLINEAR HELICOPTER DYNAMICS

This section addresses the 'Helicopter Dynamics Nonlinear Simulation' block in Fig. 1.15. A wide range of small-scale helicopter simulation models have been developed in academia [18, 19]. For low to medium control input bandwidth, demonstration (or simulation) of automatic helicopter flight, for the case of hover and low speed flight conditions, has been shown in [188–196]. On the other hand, for high bandwidth system specifications, at still these conventional flight conditions, model-based automatic flight results can be found in [42, 43, 45, 197–204], and model-free examples (in the areas of machine learning, evolutionary, and genetic algorithms) have been documented in [50, 53, 172, 205], whereas vision based systems have been reported in [206–210]. For the case of high bandwidth system specifications, at non-conventional flight conditions (e.g. aggressive/aerobatic flights), model-based approaches have been described in [21, 211, 212], whereas model-free approaches have been reported in [146, 173, 174]. However, and to the best of our knowledge, none of the previous model-based results are applicable for steep descent flight conditions, such as in the Vortex-Ring-State (VRS) or autorotation (helicopter flight with engine OFF).

Aside from these academic, white-box, helicopter models, there also exists several additional commercial, general-purpose, helicopter simulation codes. These latter are often based upon the so-called multi-body[22] concept, and have been extensively used by industry, research institutes, and academia. Examples include CAMRAD [213], FLIGHTLAB [214], GenHel [215], and HOST [216], to name a few. These simulation codes, with a proven track record, often stretching back three or four decades, are in general very reliable. They represent excellent tools for among others helicopter flight simulation purposes, operational analysis, crew training, flying qualities investigations, load prediction, vibrations analysis, and control design. However, for all their benefits, these simulation codes have also some (specific) drawbacks:

- First, these codes may be seen as third-party black-box models, since often one does not have complete access to their detailed analytical expressions, nor to the corresponding software algorithms and implementations. This may be seen as a liability,

---

[22]A multi-body system is used to simulate the dynamic behavior of interconnected rigid and flexible bodies, where each body may undergo translational and rotational displacements. The dynamic behavior of the complete system, i.e. multi-body system, results from the equilibrium of applied forces and the rate of change of momentum at each body.

when the end-goal is model-based control design. In addition, a physical understanding of the to-be controlled system is often necessary in order to be able to make judicious structural choices during the control design (e.g. adequate model order selection). This may become rather difficult if little is known about the system.

- Second, even when analytical expressions are available, the multi-body model structure adds a huge amount of detail, resulting in very high-order dynamical systems, effectively inhibiting any further manipulation of the analytical expressions.

- Third, the black-box nature of these codes restrict the range of control techniques that could potentially be used. For example, these models cannot be used for controller design when nonlinear control techniques, that explicitly require closed-form modeling, are sought.

- Finally, for the specific case of FLIGHTLAB, which is available at NLR, and although it is now possible to configure it in an autorotation mode for a small-scale helicopter, it was unfortunately not possible to do so years ago, at the start of this PhD project. The problem was related to the way FLIGHTLAB dealt with the main rotor shaft inertia, engine drive-train, and gearbox[23].

Hence, these aspects have led us towards the development of our own comprehensive, white-box, flight dynamics model, particularly suited for small-scale helicopter UAVs, and valid for a range of flight conditions, including steep descent flight and autorotation. More specifically, the model represents the nonlinear flight dynamics of a flybarless[24] helicopter main rotor, with rigid blades. The complete model incorporates the main rotor, tail rotor, fuselage, and tails of a modified Align T-REX helicopter, see Fig. 1.16.

In terms of dynamics, the state-vector **x** given in Fig. 1.15 is of dimension twenty-four. The states include the twelve-states rigid-body motion, and the dynamics of the main rotor. The former include the three-states inertial position, the three-states body linear velocities, the three-states body rotational velocities, and the three-states attitude (orientation) angles. The dynamics of the main rotor include the helicopter higher-frequency phenomena, which exist for both the engine ON or OFF (i.e. autorotation) flight condition. These higher-frequency phenomena include the main rotor three-states dynamic inflow [218, 219], and main rotor blade flap-lag dynamics (each blade defined by the four-states flap/lag angles and rotational velocities) [220]. Regarding the main rotor Revolutions Per Minute (RPM), it is

---

[23]To be able to run the FLIGHTLAB simulation, the combined inertia of the rotor shaft, drive-train, and gearbox had to be set to at least one third the main rotor inertia, which represents an unrealistically high value for the case of small-scale helicopters.

[24]The flybar is a mechanical component of the helicopter's main rotor system, and consists of a rod carrying small aerofoils (paddles), with the Angle Of Attack (AOA) of these paddles being set by the main rotor cyclic control. The AOA is the angle between a reference line on a body and the velocity vector representing the relative motion between the body and the air [217]. It is best to think of the flybar as a gyroscope that, when not steered, tends to maintain its rotation axis fixed relative to the earth. A flybar on a main rotor enhances the stability of the helicopter and hence, for a pilot using a Remote-Control (RC) device, the flybar system makes the helicopter easier to fly. This said, small-scale flybarless (i.e. without these so-called Bell-Hiller stabilizing paddles) helicopters are becoming increasingly popular. Most RC helicopter manufacturers are nowadays offering most of their RC helicopter kits in flybarless versions as well, since flybarless rotors allow for increased helicopter agility and performance, and reduced rotor mechanical complexity.

Figure 1.16: NLR's mini-UAV project (2012-2014) based on a modified Align T-REX helicopter.

generally assumed fixed for the engine ON case[25], whereas for the engine OFF case it is not fixed anymore. The main rotor RPM represents an essential part of the autorotative flight condition, and this additional state is also included in the state-vector **x** when considering the engine OFF case. This MATLAB®-based, nonlinear, continuous-time, High-Order Model (HOM) is used as a realistic small-scale helicopter simulation environment, for the validation of the FCS.

1.7.2. THE TRAJECTORY PLANNING (TP)

This section addresses the 'Trajectory Planner' block in Fig. 1.15. The TP aims at generating a feasible and optimal autorotative trajectory reference $\mathbf{x_{TP}}$, for the helicopter to follow, and additionally, though not necessarily, the feedforward nominal control inputs $\mathbf{u_{TP}}$, needed to track this trajectory. The TP computes an open-loop optimal trajectory, given a cost objective, nonlinear system dynamics, and controls and states equality and inequality constraints. The additional feedback path, denoted by a dashed line in Fig. 1.15, allows for updating the generated trajectory based upon the current state and, if used, would result in a closed-loop calculation of the reference trajectory. In this thesis, we investigate two model-based TP options. The first is an off-line approach, whereas the second is on-line feasible.

THE OFF-LINE APPROACH

From our previous discussion in Section 1.5.2, it became clear that the most natural framework for addressing TP problems was probably through optimal control theory [114]. Hence, we choose to set the off-line TP approach within the continuous-time, nonlinear, constrained optimal control paradigm. Now, given that most nonlinear constrained optimization problems are typically either computationally intensive (real-time computation), or memory intensive (off-line computation) [139], solving the TP optimization problem, within the MATLAB environment, in the full vehicle state space (including the higher-order main rotor modes of the helicopter HOM in Section 1.7.1) has shown to be rather

---

[25]Although this is a simplification, since in the engine ON case the main rotor RPM is being regulated by the governor.

costly from a computational viewpoint. The two, not mutually exclusive, options to mit-
igate such a problem are: 1) converting the helicopter HOM simulation, from a flexible
MATLAB code into a more constrained programming language (such as the C language),
which does provide a highly optimized performance and memory environment; or 2) de-
velop a Low-Order Model (LOM) better suited for nonlinear optimization problems. The
first option lives in the Information Technology (IT) realm, and requires some design effort
at the interface of various softwares[26], whereas the second option is more interesting from
a system and control viewpoint, and is more in line with the personal interests of the author.
Hence, in this thesis, we opted for the development of a LOM.

Low-Order Model (LOM)   We discuss here the method used to construct such a small-
scale helicopter LOM, which combines the required modeling accuracy with the computa-
tional tractability. In our case, the high computational cost of the HOM comes primarily
from the main rotor model. With this in mind, we considered two main avenues for the
derivation of a simplified model.

The first, and most straightforward one, consists in adapting the HOM, by replacing
all main rotor higher-order dynamics (i.e. rotor inflow, and blade flap/lag) by their corre-
sponding steady-state expressions. Although this resulted in a cheaper simulation cost, the
complex, nonlinear formulations of the main rotor forces and moments (and their corre-
sponding numerical integrations) had still a detrimental effect on the overall computational
cost. Hence, we opted for an alternative approach, which consisted in retaining the low-
order dynamics of the HOM, i.e. the rigid-body dynamics and the main rotor RPM dynam-
ics, and then replacing the costly computations of the main rotor high-order dynamics, and
main rotor forces and moments, by closed-form 'textbook'-like expressions: i.e. a static ro-
tor uniform inflow model from [218, 221] with a VRS correction from [222], a steady-state
rotor Tip-Path-Plane (TPP) model from [223, 224], and rotor forces and moments expres-
sions from [36]. The remaining helicopter model components, i.e. tail rotor, fuselage, and
tail, are further re-used, as-is, from the HOM. To compensate for the modeling inaccura-
cies introduced by the use of simpler closed-form expressions in this, so-far, white-box
model, we added a black-box component to it, in the form of eight empirical coefficients,
set at specific 'locations' within this simplified white-box model. Subsequently, simulated
input-output data, from the HOM, was used to fit these empirical coefficients. The latter
have also been scheduled on helicopter horizontal and vertical velocities. Compared to the
HOM, the domain of validity of this gray-box model is much smaller, since the data-set
used to estimate the empirical coefficients is not representative of the full helicopter flight
envelope. However, this simplified model did provide a decrease in the associated CPU
time, per model evaluation, of approximately 60 %.

Once the LOM is obtained, the solution of the continuous-time optimal control problem
requires a discretization method. Here, we apply the pseudospectral discretization numer-
ical scheme [225–227] to the optimal autorotation problem. The pseudospectral method is
known to provide exponential convergence, provided the functions under considerations are
sufficiently smooth. Once discretized, the problem is then transcribed into a NonLinear Pro-

---

[26]Although automatic MATLAB to C translation tools do exist.

gramming problem (NLP) [228], this latter being solved numerically by a well known and efficient optimization technique, in our case a Sequential Quadratic Programming (SQP) method [229–231]. The knowledge of these optimally defined autorotative trajectories, defined through this off-line approach, has proved to be useful. In particular, for the case of our Align T-REX helicopter, we found an existing bound on the total flight time based upon the initial altitude and the rotor induced[27] velocity in hover. Knowledge of this bound has shown to be relevant for the subsequent on-line TP approach.

THE ON-LINE APPROACH

The TP can either be run once, just after an engine failure has been detected, or can be continuously recomputed (see the dashed line in Fig. 1.15). For both options, the TP optimization framework of Section 1.7.2, which combines an optimal control approach with a LOM, would need to see its computational cost decrease by approximately four to five orders of magnitude, in order to retain high computational efficiency for on-line use[28].

Hence, we present here an alternative TP approach, applicable for on-line use, and based upon the concept of differential flatness. The seminal ideas of differential flatness were introduced in the early 1990s in [232–234] as part of a paradigm in which certain differential algebraic representations of dynamical systems are equivalent. In other words, a complete parametrization of all system variables—inputs, states, and outputs—may be given in terms of a finite set of independent variables, called flat outputs, and a finite number of their derivatives [235, 236]. This results in optimization problems with fewer variables [237], i.e. by the complete elimination of the dynamical constraints. In this case the trajectory generation problem is transformed from a dynamic to an algebraic one, in which the flat outputs are parametrized over a space of basis functions. The generation of optimal trajectories is then reduced to a classical algebraic interpolation or collocation problem [80, 238].

It is in general difficult to determine whether a given nonlinear system is flat, although several methods for constructing flat outputs have been documented in the literature [235, 239–241]. With regard to applications, it has been shown that simplified dynamics of aircraft and Vertical Take-Off and Landing (VTOL) aircraft are flat [242–247], and simplified helicopter dynamics is flat [235, 248, 249], whereas more realistic vehicle models are in general non-differentially flat, e.g. [235, 250]. In fact, high-fidelity helicopter models are known to be non-differentially flat. To circumvent this difficulty, a standard approach, by the research community, has consisted in progressively simplifying the model until it indeed becomes flat. Rather than generating optimal trajectories based upon such simplified representations of the helicopter dynamics, we present in this thesis an alternative approach, consisting in using only the rigid-body dynamics, with total aerodynamic forces and total moments as the new plant inputs. Although the relationship with the helicopter true control inputs[29] is lost, the advantage consists in having a model which does not include approximations, while being exactly flat. Now, since the rigid-body dynamics does not include

---

[27]The main rotor induced flow corresponds to the flow field induced by the rotation of the main rotor blades.

[28]For on-line use in a hard real-time environment where stringent timing constraints exist (e.g. in our case the 50 Hz closed-loop update rate).

[29]Main rotor collective, lateral and longitudinal cyclic, and tail rotor collective.

the main rotor and RPM dynamics, and in order to obtain feasible autorotative trajectories, we will constrain the trajectory flight time by the bound deduced using the off-line TP approach.

We summarize now the main idea behind our TP methodology:

- **Step 1.** Base the TP optimization on the concept of differential flatness, using a lower-complexity model (in our case the rigid-body dynamics). Combining the flatness with a lower-complexity model allows for on-line tractable computations.

- **Step 2.** Derive additional trajectory constraints (in our case a bound on total flight time), obtained from the analysis of off-line optimization results, using a nonlinear optimal control approach combined with a higher-complexity model (in our case, either the HOM helicopter of Section 1.7.1, or the LOM of Section 1.7.2).

- **Step 3.** Use the optimization framework of Step 1, combined with the additional constraints from Step 2 (in our case a bound on total flight time), to generate, on-line, feasible and optimal trajectories, for the original HOM helicopter.

1.7.3. THE TRAJECTORY TRACKING (TT)
This section addresses the 'Trajectory Tracking' block in Fig. 1.15. The TT shall compare current output values $\mathbf{y}$ with the optimal reference values $\mathbf{x_{TP}}$ produced by the TP, and shall formulate the feedback controls $\mathbf{u_{TT}}$ aimed at decreasing the tracking error, hence ensuring that the helicopter flies along the optimal trajectory. The tracking error may be due to a combination of model uncertainty (unmodeled higher-order dynamics, unmodeled static nonlinearities, parametric uncertainties, delays), and signal uncertainty (wind disturbances and noise). As stated earlier, very few papers have addressed the subject of tracking an autorotative trajectory, with validation through experimental results or 3D high-fidelity simulations [51, 146, 149, 151]. None of the previous results considered a robust TT approach. Hence, we select here a model-based, robust, TT approach, in order to obtain a closed-loop system which is less sensitive to modeling uncertainties.

ROBUST CONTROL BASED TT
Since the helicopter dynamics is highly nonlinear, the design of the TT necessitates an approach that effectively respects or exploits the system's nonlinear structure. To this end, several control methods are available: from 1) robust control; 2) classical gain-scheduling, and Linear Parameter-Varying (LPV) approaches; to 3) truly nonlinear control methods (e.g. nonlinear MPC, Lyapunov based methods such as sliding mode and backstepping, adaptive control, or even passivity-based approaches). In this thesis we choose to apply a robust control μ strategy. This method consists in using a nominal Linear Time-Invariant (LTI) plant coupled with an uncertainty, and applying a small gain approach [251, 252] to design a single robust LTI controller. This approach, when implemented on-line, is computationally very efficient. Now, rather than modeling the uncertainty in a detailed manner, an input multiplicative uncertainty is added here to compensate for the unmodeled

plant nonlinearities and unmodeled higher-order rotor dynamics[30], by lumping all types of model uncertainty together into a complex, full-block, input multiplicative uncertainty. Finally, the robust controller synthesis consists in obtaining a controller insensitive to this multiplicative uncertainty at the plant input.

AFFINE LPV MODELING

Rather than using a robust control μ strategy, one could also consider some other control method, as listed in Section 1.7.3. In particular, LPV systems have become celebrated as they represent an attractive midway approach between LTI, and nonlinear or time-varying structures [253, 254]. LPV systems allow to enclose nonlinear behaviors into a linear framework, where LPV control methods can be seen as an extension of the standard H_2 and H_∞ LTI synthesis techniques [255–262]. The LPV method amends also the main drawbacks of classical gain-scheduling [263, 264]: 1) by eliminating the need for repeated designs/simulations in order to handle the global control problem; and 2) by guaranteeing both stability and performance along all possible parameter trajectories. In addition, LPV control design problems are efficiently solved, by first expressing the problems as Linear Matrix Inequality (LMI) optimizations [265]—subsequently formulated as Semi-Definite Programs (SDP) [266]—for which there are several powerful numerical solutions [267, 268]. This resulted in a growing number of applications, such as in aerospace [269–274], wind turbines [275], wafer steppers [276, 277], Compact-Disk players [278], and robotic manipulators [279]. Now, and for all its benefits, the LPV control paradigm typically takes the existence of the plant, in LPV form, as a starting point. However, a systematic formulation of a nonlinear system into a suitable LPV model remains often problematic [280]. Hence, the problem of simplifying a large scale, nonlinear model, such as our helicopter HOM of Section 1.7.1, into a LPV representation is thus highly relevant.

With this in mind, and for the case where a plant's nonlinear model already exists, we present in this thesis an affine LPV modeling methodology. This LPV modeling method has subsequently been applied to a modified pointmass pendulum, and to the helicopter HOM of Section 1.7.1. For the pointmass pendulum example, the LPV modeling approach was validated in open- and closed-loop (using robust and LPV controllers). For the helicopter HOM case, the LPV modeling approach resulted in a LPV model having a large number of (more than thirty) scheduling parameters. Unfortunately, it became impossible to synthesize LPV controllers with such a high-order LPV model. In fact, it is well-known that the numerical conditioning and solvability of LMI problems play a crucial role in LPV practical design methods [275–278]. A way to mitigate such problems would consist in applying some LPV model reduction techniques [281, 282], in order to obtain a LPV model having fewer scheduling parameters, hence better suited for LPV controller synthesis.

1.8. OVERVIEW OF THIS THESIS

The development of an autonomous helicopter system requires for an elaborate synergy between various engineering fields, including: 1) modeling; 2) system identification; 3) estimation and filtering; and 4) optimization and control (e.g. guidance and control). In this

---

[30]Unmodeled in the nominal LTI plant used for controller design; the higher-order dynamics are however modeled in the nonlinear HOM plant of Section 1.7.1.

thesis, aspects of modeling, guidance and control for a small-scale helicopter in autorotation
are discussed, and new solutions are presented. This thesis is organized as follows:

- In **Chapter 2** we present a helicopter flight dynamics nonlinear model for a flybar-less, articulated, Pitch-Lag-Flap (P-L-F) main rotor with rigid blades, particularly suited for small-scale UAVs. This high-order nonlinear model incorporates the main rotor, tail rotor, fuselage, and tails. This model is further applicable for high bandwidth control specifications, and is valid for a range of flight conditions, including the Vortex-Ring-State and autorotation. The goal of this comprehensive nonlinear model is twofold: 1) it serves as a nonlinear simulation environment on which the flight control system can be tested; and 2) it provides a basis for model-based control design.

- In **Chapter 3** optimal engine OFF (autorotative) landing trajectories are derived through a model-based, direct optimal control framework. These open-loop optimal trajectories, generated by a trajectory planner, represent the solution to the minimization of a cost objective, given low-order nonlinear system dynamics, controls and states equality and inequality constraints. The optimization setting, developed in this Chapter, allows to test and evaluate various cost objectives. Once the final cost objective and constraints have been frozen, optimal autorotative trajectories can be computed off-line, for a range of initial conditions, and could even be stored as lookup tables on-board a flight control computer. These trajectories provide both the optimal states to be tracked by a feedback controller, and optionally the feedforward nominal controls.

- In **Chapter 4** we present a model-based, trajectory planning and tracking framework, for a helicopter with engine OFF, anchored within the combined paradigms of differential flatness based planning and robust control based tracking. The advantage of this methodology is that it is model-based and real-time feasible, since: 1) it allows for a computationally tractable determination of the optimal trajectories; and 2) it is based upon an easy to realize and implement LTI trajectory tracker. A similar flight control system, for the engine ON condition, is also provided.

- In **Chapter 5** the methodology of Chapter 4 is validated on the high-order nonlinear helicopter model of Chapter 2. To better illustrate the various challenges encountered when designing a planning and tracking system for the engine OFF condition, a comparison with some engine ON automated flight maneuvers is also provided.

- In **Chapter 6** we tackle the problem of approximating a known complex nonlinear model by an affine LPV model. To illustrate the practicality of the presented LPV modeling strategy, we apply it to a pointmass pendulum example, and provide extensive analysis in, both, open- and closed-loop simulation settings. When applied to the high-order nonlinear helicopter model of Chapter 2, the LPV modeling approach resulted in a LPV model having an excessive number of scheduling parameters, effectively impeding any LPV control design.

- Finally **Chapter 7** summarizes the results of this thesis, and outlines directions for

future research, such as the experimental validation of the here-presented guidance and control system.

1.8.1. Contributions

- A comprehensive helicopter nonlinear high-order modeling framework, valid for a range of flight conditions including steep descent flights and autorotation, and particularly suited for small-scale helicopter UAVs has been presented in [25, 283, 284].

- The determination of optimal autorotative landing trajectories, by solving an off-line nonlinear optimal control problem, for the case of a small-scale helicopter UAV, has been presented in [285–287].

- The first demonstration—using a high-fidelity, high-order, nonlinear helicopter simulation—of a real-time feasible, model-based optimal trajectory planning, and model-based robust trajectory tracking, for the case of a small-scale helicopter UAV in autorotation, has been presented in [288].

- A novel affine LPV modeling framework has been presented in [289].

REFERENCES

[1] UVS-International, *RPAS Remotely Piloted Aircraft Systems: The Global Perspective* (Blyenburgh & Co, Paris, France, 2013).

[2] DoD, *Unmanned Aircraft Systems (UAS) Roadmap 2005-2030*, Tech. Rep. (U.S.A. DoD, 2005).

[3] C. Eck, *Navigation Algorithms with Applications to Unmanned Helicopters*, Ph.D. thesis, ETH Zurich (2001).

[4] *http://tealgroup.com/index.php/about-teal-group-corporation/press-releases/94-2013-uav-press-release*.

[5] M. Selier, S. Taamallah, E. J. Smit, and H. Haverdings, *NLR Activities on Small Autonomous Helicopters. Intermediate Results and Lessons Learnt Under the NIVR Basic Research Programme*, Tech. Rep. NLR-CR-2005-680 (NLR, 2005).

[6] UVS-International, *UAV Systems: The Global Perspective 2006/2007* (Blyenburgh & Co, Paris, France, 2006).

[7] *http://www.delftdynamics.nl/*.

[8] D. B. Stringer and D. P. Schrage, *Managing army airspace for integrated manned and unmanned aviation operations,* in *Am. Helicopter Soc.* (2004).

[9] B. Elias, *Pilotless Drones: Background and Considerations for Congress Regarding Unmanned Aircraft Operations in the National Airspace System*, Tech. Rep. R42718 (U.S. Congressional Research Service, 2012).

[10] Anonymous, *Roadmap for the integration of civil Remotely-Piloted Aircraft Systems into the European Aviation System*, Tech. Rep. (European Commission, 2013).

[11] DoD, *Unmanned Aerial Vehicle Reliability Study*, Tech. Rep. (U.S.A. DoD, 2003).

[12] DoD, *Defense Science Board Study on Unmanned Aerial Vehicles (UAVs) and Uninhabited Combat Aerial Vehicles (UCAVs)*, Tech. Rep. (U.S.A. DoD, 2004).

[13] *https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/helicopter_flying_handbook/media/hfh_ch02.pdf*.

[14] *https://en.wikipedia.org/wiki/Helicopter_rotor*.

[15] *http://www.novatel.com/assets/Web-Phase-2-2012/Solution-Pages/AttitudePlane.png*.

[16] *http://en.wikipedia.org/wiki/Helicopter_flight_controls*.

[17] B. B. Blake and I. B. Alansky, *Stability and control of the yuh-61a,* in *Am. Helicopter Soc.* (1975).

[18] F. Kendoul, *A survey of advances in guidance, navigation and control of unmanned rotorcraft systems,* J. of Field Robotics **29**, 315 (2012).

[19] G. Cai, J. Dias, and L. Seneviratne, *A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends,* Unmanned Systems **2** (2014).

[20] K. Peng, G. Cai, B. M. Chen, M. Dong, K. Y. Luma, and T. H. Lee, *Design and implementation of an autonomous flight control law for a uavhelicopter,* Automatica **45**, 2333 (2009).

[21] B. Mettler, *Identification Modelling and Characteristics of Miniature Rotorcraft* (Kluwer Academic Publishers, Norwell Mass, USA, 2003).

[22] *http://www.b-domke.de/AviationImages/Rotorhead/14336.html*.

[23] G. D. Padfield, *Helicopter Flight Dynamics* (Blackwell Science Ltd, Oxford, UK, 1996).

[24] C. R. Brackbill, *Helicopter Rotor Aeroelastic Analysis Using a Refined Elastomeric Damper Model*, Ph.D. thesis, Pennsylvania State University (2000).

[25] S. Taamallah, *A qualitative introduction to the vortex-ring-state, autorotation, and optimal autorotation,* in *Europ. Rotorcraft Forum* (2010).

[26] D. Santamaria, A. Viguria, M. Bejar, K. Kondak, and A. Ollero, *Towards autonomous autorotation landing for small size unmanned helicopters,* J. Intell. Robot Syst. **69**, 171 (2013).

[27] A. Y. Lee, *Optimal Landing of a Helicopter in Autorotation*, Ph.D. thesis, Stanford University (1985).

[28] *http://www.metaglossary.com/mea-nings/552646/*.

[29] *http://www.helipedia.com/in-dex.php?title=Autorotation*.

[30] *http://www.copters.com/pilot/auto-rotation.html*.

[31] *http://www.cybercom.net/~cop-ters/aero/autorotation.html*.

[32] *http://helicopterflight.net/auto-rotation.htm*.

[33] R. W. Prouty, *Helicopter Performance, Stability, and Control* (Krieger Publishing Company, Malabar, Florida USA, 1995).

[34] G. J. Leishman, *Principles of Helicopter Aerodynamics* (Cambridge University Press, Cambridge, UK, 2000).

[35] S. E. Slaymaker and R. B. Gray, *Power-Off Flare-up Tests of a Model Helicopter Rotor in Vertical Autorotation*, Tech. Rep. TN 2870 (NACA, 1953).

[36] W. Johnson, *Helicopter Theory* (Dover Publications Inc., NY, USA, 1994).

[37] E. N. Bachelder and B. L. Aponso, *An autorotation flight director for helicopter training,* in *Am. Helicopter Soc.* (2003).

1

[38] B. L. Aponso, E. N. Bachelder, and D. Lee, *Automated autorotation for un-manned rotorcraft recovery,* in *AHS Int. Specialists' Meeting On Unmanned Rotor-craft* (2005).

[39] B. L. Aponso, D. Lee, and E. N. Bachelder, *Evaluation of a rotorcraft autorotation training display on a commercial flight training device,* J. of the Am. Helicopter Soc. (2007).

[40] S. Hart, *Analysis of civil helicopter accidents,* in *Proceedings of the HeliExpo '98* (1998).

[41] S. P. Rogers and C. N. Asbury, *A Flight Training Simulator for Instructing the Heli-copter Autorotation Maneuver (Enhanced Version)*, Tech. Rep. TR FR-1372 (NASA AMES, 2000).

[42] D. H. Shim, *Hierarchical Control System Synthesis for Rotorcraft-based Unmanned Aerial Vehicles*, Ph.D. thesis, University of California at Berkeley (2000).

[43] M. L. Civita, *Integrated Modeling and Robust Control for Full-Envelope Flight of Robotic Helicopters*, Ph.D. thesis, Carnegie Mellon University (2002).

[44] V. Gavrilets, *Autonomous Aerobatic Maneuvering of Miniature Helicopter*, Ph.D. thesis, Massachusetts Institute of Technology (2003).

[45] O. Tanner, *Modeling, Identification, and Control of Autonomous Helicopters*, Ph.D. thesis, ETH Zurich (2003).

[46] G. R. Drozeski, *A Fault-Tolerant Control Architecture for Unmanned Aerial Vehicles*, Ph.D. thesis, Georgia Institute of Technology (2005).

[47] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration* (John Wiley & Sons, U.S.A., 2001).

[48] R. M. Murray, *Robust Control and nonholonomic motion planning*, Ph.D. thesis, University of California at Berkeley (1990).

[49] A. Sala, T. M. Guerra, and R. Babuska, *Perspectives of fuzzy systems and control,* Fuzzy Sets and Systems **156**, 432 (2005).

[50] J. M. Engel, *Reinforcement Learning Applied to UAV Helicopter Control*, Tech. Rep. (Master Thesis, Delft University of Technology, 2005).

[51] Z. Sunberg and J. Rogers, *A fuzzy logic-based controller for helicopter autorotation,* in *AIAA Aerospace Sciences Meeting and Exhibit* (2013).

[52] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (The MIT Press, U.S.A., 1998).

[53] J. Bagnell and J. Schneider, *Autonomous helicopter control using reinforcement learning policy search methods,* in *Int. Conf. on Robotics and Automation* (2001).

1

[54] D. J. Lee, H. Bang, and K. Baek, *Autorotation of an unmanned helicopter by a reinforcement learning algorithm,* in *AIAA Guidance, Navigation and Control Conf.* (2008).

[55] D. J. Lee and H. Bang, *Autonomous autorotation of an unmanned helicopter using a reinforcement learning algorithm,* J. Aerospace Information Systems **10**, 98 (2013).

[56] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Masschusetts, 1989).

[57] L. Davis, *Handbook of Genetic Algorithms* (Van Nostrand Reinhold, NY, 1991).

[58] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolutionary Programs* (Springer-Verlag, Berlin, 1992).

[59] J. C. Rubio and S. Kragelund, *The trans-pacific crossing: Long range adaptive path planning for uavs through variable wind fields,* in *22nd Digital Avionics Systems Conf.* (2003).

[60] C. Zheng, L. Li, F. Xu, F. Sun, and M. Ding, *Evolutionary route planner for unmanned air vehicles,* IEEE Trans. on Robotics & Automation **21**, 609 (2005).

[61] V. Kanakakis and N. Tsourveloudis, *Evolutionary path planning and navigation of autonomous underwater vehicles,* in *Mediterranean Conf. on Control and Automation* (2007).

[62] K. J. Obermeyer, *Path planning for a uav performing reconnaissance of static ground targets in terrain,* in *AIAA Guidance, Navigation, and Control Conf.* (2009).

[63] S. L. Padula, *Design of quiet rotorcraft approach trajectories: Verification phase,* in *AIAA/ISSMO Multidisciplinary Analysis Optimization Conf.* (2010).

[64] R. Romijn, L. Ozkan, S. Weiland, J. Ludlage, and W. Marquardt, *A grey-box modeling approach for the reduction of nonlinear systems,* in *8th Int. IFAC Symposium on Dynamics and Control of Process Systems* (2007).

[65] T. A. Johansen, *Operating Regime Based Process Modeling and Identification*, Ph.D. thesis, Norwegian Institute of Technology (1994).

[66] A. F. Chalmers, *What is this Thing Called Science ?* (Open University Press, 1982).

[67] J. M. Maciejowski, *Predictive Control with Constraints* (Prentice Hall, U.K., 2002).

[68] E. F. Camacho and C. Bordons, *Model Predictive Control Second Edition* (Springer-Verlag, London, 2007).

[69] J. Richalet, A. Rault, J. L. Testud, and J. Papon, *Model predictive heuristic control: Applications to industrial processes,* Automatica **14**, 413 (1978).

[70] C. R. Cutler and B. L. Ramaker, *Dynamic matrix control - a computer control algorithm,* in *Joint Automatic Control Conf.* (1980).

[71] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, *Generalized predictive control: I. the basic algorithm,* Automatica **23** (1987).

[72] C. E. Garcia, D. M. Prett, and M. Morari, *Model predictive control: Theory and practice - a survey,* Automatica **25**, 335 (1989).

[73] J. Richalet, *Industrial applications of model based predictive control,* Automatica **29**, 1251 (1993).

[74] J. B. Rawlings, *Tutorial overview of model predictive control,* IEEE Control Systems Magazine , 38 (2000).

[75] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, *Constrained model predictive control: Stability and optimality,* Automatica **36** (2000).

[76] R. Findeisen and F. Allgower, *An introduction to nonlinear model predictive control,* in *21st Benelux Meeting on Systems and Control* (2002).

[77] S. J. Qin and T. A. Badgwell, *A survey of industrial model predictive control technology,* Control Engineering Practice **11**, 733 (2003).

[78] A. Bemporad, *Model predictive control design: New trends and tools,* in *IEEE Conf. on Decision and Control* (2006).

[79] A. Jadbabaie, *Receding Horizon Control of Nonlinear Systems: A Control Lyapunov Function Approach*, Ph.D. thesis, California Institute of Technology (2000).

[80] R. M. Murray, J. Hauser, A. Jadbabaie, M. B. Milam, N. Petit, W. B. Dunbar, and R. Franz, *Online control customization via optimization-based control. In T. Samad and G. Balas, Editors, Software-Enabled Control: Information Technology for Dynamical Systems* (IEEE Press, 2003).

[81] M. T. Frye, C. Qian, and R. D. Colgren, *Receding horizon control of a linear parameter varying model of the raptor 50 helicopter,* in *AIAA Guidance, Navigation and Control Conf.* (2005).

[82] C. L. Castillo, W. Moreno, and K. P. Valavanis, *Unmanned helicopter waypoint trajectory tracking using model predictive control,* in *Mediterranean Conf. on Control and Automation* (2007).

[83] S. Colautti and H. Haverdings, *Small scale rotorcraft uas flight control using mpc,* in *Int. Conf. on Control, Automation and Systems* (2010).

[84] M. K. Samal, M. Garratt, H. Pota, and H. T. Sangani, *Model predictive attitude control of vario unmanned helicopter,* in *37th Annual Conf. on IEEE Industrial Electronics Soc.* (2011).

[85] E. A. Wan and A. A. Bogdanov, *Model predictive neural control with applications to a 6 dof helicopter model,* in *Am. Control Conf.* (2001).

[86] H. J. Kim, D. H. Shim, and S. Sastry, *Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles,* in *Am. Control Conf.* (2002).

[87] H. J. Kim and D. H. Shim, *A flight control system for aerial robots: Algorithms and experiments,* Control Engineering Practice **11**, 1389 (2003).

[88] D. H. Shim, H. Chung, H. J. Kim, and S. Sastry, *Autonomous exploration in unknown urban environments for unmanned aerial vehicles,* in *AIAA Guidance, Navigation, and Control Conf.* (2005).

[89] D. H. Shim and S. Sastry, *A situation-aware flight control system design using real-time model predictive control for unmanned autonomous helicopters,* in *AIAA Guidance, Navigation, and Control Conf.* (2006).

[90] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl, *Autonomous autorotation of unmanned rotorcraft using nonlinear model predictive control,* J. Intell. Robot Syst. **57**, 351 (2010).

[91] L. Singh and J. Fuller, *Trajectory generation for a uav in urban terrain, using nonlinear mpc,* in *Am. Control Conf.* (2001).

[92] Y. Kang and J. K. Hedrick, *Design of nonlinear model predictive controller for a small fixed-wing unmanned aerial vehicle,* in *AIAA Guidance, Navigation, and Control Conf.* (2006).

[93] N. Slegers, J. Kyle, and M. Costello, *Nonlinear model predictive control technique for unmanned air vehicles,* AIAA J. of Guidance, Control, and Dynamics **29** (2006).

[94] Y. Kang and J. K. Hedrick, *Linear tracking for a fixed-wing uav using nonlinear model predictive control,* IEEE Trans. on Control Systems Technology **17**, 1202 (2009).

[95] N. T. Depenbusch and J. W. Langelaan, *Receding horizon control for atmospheric energy harvesting by small uavs,* in *AIAA Guidance, Navigation and Control Conf.* (2010).

[96] G. A. Garcia1 and S. Keshmiri, *Nonlinear model predictive controller for navigation, guidance and control of a fixed-wing uav,* in *AIAA Guidance, Navigation, and Control Conf.* (2011).

[97] M. V. Kothare, V. Balakrishnan, and M. Morari, *Robust constrained model predictive control using linear matrix inequalities,* Automatica **32**, 1361 (1996).

[98] O. Slupphaug and B. A. Foss, *Bilinear matrix inequalities and robust stability of nonlinear multi-model mpc,* in *Am. Control Conf.* (1998).

[99] Y. Lu and Y. Arkun, *Quasi-min-max mpc algorithms for lpv systems,* Automatica **36**, 527 (2000).

[100] J. Primbs and V. Nevistic, *A framework for robustness analysis of finite receding horizon control,* IEEE Trans. Autom. Control **45**, 1828 (2000).

1

[101] A. Casavola, M. Giannelli, and E. Mosca, *Min-max predictive control strategies for input-saturated polytopic uncertain systems,* Automatica **36**, 125 (2000).

[102] F. A. Cuzzola, J. C. Jeromel, and M. Morari, *An improved approach for contrained robust model predictive control,* Automatica **38**, 1183 (2002).

[103] A. Casavola, D. Famularo, and G. Franze, *A feedback minmax mpc algorithm for lpv systems subject to bounded rates of change of parameters,* IEEE Trans. Autom. Control **47**, 1147 (2002).

[104] A. Casavola, D. Famularo, and G. Franze, *Predictive control of constrained nonlinear system via lpv linear embeddings,* Int. J. of Robust and Nonlinear Control **13**, 281 (2003).

[105] P. Park and S. C. Jeong, *Constrained rhc for lpv systems with bounded rates of parameter variations,* Automatica **40**, 865 (2004).

[106] Y. I. Lee, M. Cannon, and B. Kouvaritakis, *Extended invariance and its use in model predictive control,* Automatica , 2163 (2005).

[107] Z. Zhiqiang, X. Lihong, and Y. Meng, *Quasi-min-max mpc with linear parameter-dependent state feedback law,* in *ICTA* (2005).

[108] S. Lee and S. Won, *Model predictive control for linear parameter varying systems using a new parameter dependent terminal weighting matrix,* IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences **E89-A**, 2166 (2006).

[109] A. Casavola, D. Famularo, G. Franze, and E. Garone, *An improved predictive control strategy for polytopic lpv linear systems,* in *IEEE Conf. on Decision and Control* (2006).

[110] L. Imsland, J. A. Rossiter, B. Pluymers, and J. Suykens, *Robust triple mode mpc,* in *Am. Control Conf.* (2006).

[111] A. Casavola, D. Famularo, G. Franze, and E. Garone, *A dilated mpc control strategy for lpv linear systems,* in *Europ. Control Conf.* (2007).

[112] D. Angeli, A. Casavola, G. Franze, and E. Moscaa, *An ellipsoidal off-line mpc scheme for uncertain polytopic discrete-time systems,* Automatica **44**, 3113 (2008).

[113] C. Lana and M. Rotea, *Desensitized model predictive control applied to a structural benchmark problem,* in *IFAC, 17th World Congress* (2008).

[114] A. E. Bryson and Y. C. Ho, *Applied Optimal Control* (Taylor & Francis Group, New York, 1975).

[115] R. F. Stengel, *Optimal Control and Estimation* (Dover Publications, NY, 1994).

[116] J. Z. Ben-Asher, *Optimal Control with Aerospace Applications* (AIAA Education Series, Reston VA, 2010).

[117] A. la Cour-Harbo and M. Bisgaard, *State-control trajectory generation for helicopter slung load system using optimal control,* in *AIAA Guidance, Navigation, and Control Conf.* (2009).

[118] M. Harada and K. Bollino, *Minimum-time circling flight of a triarm coaxial rotor uav,* in *AIAA Guidance Navigation and Control Conf.* (2011).

[119] M. Harada and K. Bollino, *Optimal trajectory of a glider in ground effect and wind shear,* in *AIAA Guidance, Navigation and Control Conf.* (2005).

[120] K. P. Bollino, L. R. Lewis, P. Sekhavat, and I. M. Ross, *Pseudospectral optimal control: A clear road for autonomous intelligent path planning,* in *AIAA Infotech* (2007).

[121] Q. Gong, L. R. Lewis, and I. M. Ross, *Pseudospectral motion planning for autonomous vehicles,* AIAA J. of Guidance, Control, and Dynamics **32** (2009).

[122] B. D. McCracken, K. L. Burzota, M. M. Ramos, and T. R. Jorris, *Cooperative unmanned air vehicle mission planning and collision avoidance,* in *AIAA Infotech* (2011).

[123] R. Choe and N. Hovakimyan, *Perching maneuver for an mav augmented with an l1 adaptive controller,* in *AIAA Guidance, Navigation, and Control Conf.* (2011).

[124] R. M. Murray, *Nonlinear control of mechanical systems: A lagrangian perspective,* Annual Reviews in Control **21**, 31 (1997).

[125] C. Goerzen, Z. Kong, and B. Mettler, *A survey of motion planning algorithms from the perspective of autonomous uav guidance,* J. Intell. Robot Syst. **57**, 65 (2010).

[126] N. Dadkhah and B. Mettler, *Survey of motion planning literature in the presence of uncertainty: Considerations for uav guidance,* J. Intell. Robot Syst. **65**, 233 (2012).

[127] J. C. Latombe, *Robot Motion Planning* (Kluwer Int. Series in Engineering and Computer Science, Mass., 1991).

[128] J. C. Latombe, *Motion planning: A journey of robots, molecules, digital actors, and other artifacts,* Int. J. of Robotics Research **18**, 1119 (1999).

[129] S. M. LaValle, *Planning Algorithms* (Cambridge University Press, Cambridge England, 2006).

[130] M. Komoda, *Prediction of height-velocity boundaries for rotorcraft by application of optimizing techniques,* Trans. of the Japan Soc. for Aeronautical and Space Sciences **15**, 208 (1973).

[131] W. Johnson, *Helicopter Optimal Descent and Landing After Power Loss*, Tech. Rep. TM 73-244 (NASA Ames Research Center, 1977).

[132] A. Y. Lee, A. E. Bryson, and W. S. Hindson, *Optimal landing of a helicopter in autorotation,* AIAA J. of Guidance, Control, and Dynamics **11**, 7 (1988).

1

[133] Y. Okuno, K. Kawachi, A. Azuma, and S. Saito, *Analytical prediction of height-velocity diagram of a helicopter using optimal control theory,* AIAA J. of Guidance, Control, and Dynamics **14**, 453 (1991).

[134] Y. Okuno and K. Kawachi, *Optimal control of helicopters following power failure,* AIAA J. of Guidance, Control, and Dynamics **17**, 181 (1994).

[135] M. W. Floros, *Descent analysis for rotorcraft survivability with power loss,* in *Am. Helicopter Soc.* (2009).

[136] A. Lee, *Engineering notes - optimal autorotational descent of a helicopter with control and state inequality constraints,* AIAA J. of Guidance, Control, and Dynamics **13**, 922 (1990).

[137] A. A. Jhemi, E. B. Carlson, Y. J. Zhao, and R. T. N. Chen, *Optimization of rotorcraft flight following engine failure,* J. of the Am. Helicopter Soc. (2004).

[138] B. L. Aponso, D. Lee, and E. N. Bachelder, *Evaluation of a rotorcraft autorotation training display on a commercial flight training device,* in *Am. Helicopter Soc.* (2005).

[139] C. L. Bottasso, A. Croce, D. Leonello, and L. Riviello, *Optimization of critical trajectories for rotorcraft vehicles,* J. of the Am. Helicopter Soc. , 165 (2005).

[140] C. L. Bottasso, C. S. Chang, A. Croce, D. Leonello, and L. Riviello, *Adaptive planning and tracking of trajectories for the simulation of maneuvers with multibody models,* Computer Methods in Applied Mechanics and Engineering (2006).

[141] E. B. Carlson, S. Xue, J. Keane, and K. Burns, *H-1 upgrades height-velocity diagram development through flight test and trajectory optimization,* in *Am. Helicopter Soc.* (2006).

[142] P. Bibik and J. Narkiewicz, *Helicopter optimal control after power failure using comprehensive dynamic model,* AIAA J. of Guidance, Control, and Dynamics **35**, 1354 (2012).

[143] C. L. Bottasso, G. Maisano, and F. Luraghi, *Efficient rotorcraft trajectory optimization using comprehensive vehicle models by improved shooting methods,* in *Europ. Rotorcraft Forum* (2009).

[144] C. L. Bottasso, G. Maisano, and F. Scorcelletti, *Trajectory optimization procedures for rotorcraft vehicles, their software implementation, and applicability to models of increasing complexity,* J. of the Am. Helicopter Soc. (2010).

[145] M. W. Floros, *Application of sequential quadratic programming to rotorcraft survivability with power loss,* in *Am. Helicopter Soc.* (2011).

[146] P. Abbeel, A. Coates, T. Hunter, and A. Y. Ng, *Autonomous autorotation of an rc helicopter,* in *11th Int. Symposium on Experimental Robotics (ISER)* (2008).

[147] S. Tierney and J. W. Langelaan, *Autorotation path planning using backwards reachable set and optimal control,* in *Am. Helicopter Soc.* (2010).

[148] N. Grande and J. W. Langelaan, *Safe autonomous flare and landing during autorotation through wind shear,* in *Am. Helicopter Soc.* (2013).

[149] J. Holsten, S. Loechelt, and W. Alles, *Autonomous autorotation flights of helicopter uavs to known landing sites,* in *Am. Helicopter Soc.* (2010).

[150] T. Yomchinda, J. F. Horn, and J. W. Langelaan, *Flight path planning for descent-phase helicopter autorotation,* in *AIAA Guidance, Navigation, and Control Conf.* (2011).

[151] T. Yomchinda, J. F. Horn, and J. W. Langelaan, *Autonomous control and path planning for autorotation of unmanned helicopters,* in *Am. Helicopter Soc.* (2012).

[152] D. H. Shim, H. J. Kim, and S. Sastry, *Hierarchical control system synthesis for rotorcraft-based unmanned aerial vehicles,* in *AIAA Guidance, Navigation and Control Conf.* (2000).

[153] M. Takahashi, G. Schulein, and M. Whalley, *Flight control law design and development for an autonomous rotorcraft,* in *Am. Helicopter Soc.* (2008).

[154] M. Bergerman, O. Amidi, J. Miller, N. Vallidis, and T. Dudek, *Cascaded position and heading control of a robotic helicopter,* in *Int. Conf. on Intelligent Robots and Systems* (2007).

[155] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa, *Autonomous Flying Robots - Unmanned Aerial Vehicles and Micro Aerial Vehicles* (Springer, New York, 2010).

[156] J. Shin, K. Nonami, D. Fujiwara, and K. Hazawa, *Model-based optimal attitude and positioning control of small-scale unmanned helicopter,* Robotica **23**, 51 (2005).

[157] G. M. Voorsluijs, *Parameter-dependent robust control for a rotorcraft uav,* in *AIAA Guidance, Navigation and Control Conf.* (2005).

[158] M. F. Weilenmann, U. Christen, and H. P. Geering, *Robust helicopter position control at hover,* in *Am. Control Conf.* (1994).

[159] J. Gadewadikar, F. L. Lewis, K. Subbarao, and B. M. Chen, *Structured h_∞ command and control-loop design for unmanned helicopters,* AIAA J. of Guidance, Control, and Dynamics **31**, 1093 (2008).

[160] Y. Q. He and J. D. Han, *Acceleration-feedback-enhanced robust control of an unmanned helicopter,* AIAA J. of Guidance, Control, and Dynamics **33**, 1236 (2010).

[161] D. H. Shim, T. J. Koo, F. Hoffmann, and S. Sastry, *A comprehensive study on control design of autonomous helicopter,* in *IEEE Conf. on Decision and Control* (1998).

1

[162] A. Isidori, L. Marconi, and A. Serrani, *Robust nonlinear motion control of a heli-copter,* IEEE Trans. Autom. Control **40**, 413 (2003).

[163] P. Simplicio, M. D. Pave, E. vanKampen, and Q. P. Chu, *An acceleration measurements-based approach for helicopter nonlinear flight control using incre-mental nonlinear dynamic inversion,* Control Engineering Practice **21**, 1065 (2013).

[164] E. N. Johnson and S. K. Kannan, *Adaptive trajectory control for autonomous heli-copters,* AIAA J. of Guidance, Control, and Dynamics **28** (2005).

[165] I. Yavrucuk, J. V. R. Prasad, and S. Unnikrishnan, *Envelope protection for au-tonomous unmanned aerial vehicles,* AIAA J. of Guidance, Control, and Dynamics **32**, 248 (2009).

[166] B. J. Guerreiro, C. Silvestre, R. Cunha, C. Cao, and N. Hovakimyan, *L1 adaptive control for autonomous rotorcraft,* in *Am. Helicopter Soc.* (2009).

[167] M. Bisgaard, A. Cour-Harbo, and J. D. Bendtsen, *Adaptive control system for au-tonomous helicopters slung load operations,* Control Engineering Practice **18**, 800 (2010).

[168] R. Mahony, T. Hamel, and A. Dzul, *Hover control via lyapunov control for an autonomous model helicopter,* in *IEEE Conf. on Decision and Control* (1999).

[169] E. Frazzoli, M. A. Dahleh, and E. Feron, *Trajectory tracking control design for au-tonomous helicopters using a backstepping algorithm,* in *Am. Control Conf.* (2000).

[170] R. Olfati-Saber, *Nonlinear Control of Underactuated Mechanical Systems with Ap-plication to Robotics and Aerospace Vehicles*, Ph.D. thesis, Massachusetts Institute of Technology (2001).

[171] V. Gavrilets, E. Frazzoli, B. Mettler, and E. Feron, *Aggressive maneuvering of small autonomous helicopters: A human-centered approachs,* Int. J. of Robotics Research **20** (2001).

[172] A. Y. Ng, H. J. Kim, M. Jordan, and S. Sastry, *Autonomous helicopter flight via reinforcement learning,* in *Neural Information Processing Systems (NIPS)* (2004).

[173] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, *Autonomous inverted helicopter flight via reinforcement learning,* in *Int. Symposium on Experimental Robotics* (2004).

[174] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, *An application of reinforcement learning to aerobatic helicopter flight,* in *NIPS 19* (2007).

[175] A. Ollero and L. Merino, *Control and perception techniques for aerial robotics,* An-nual Reviews in Control **28**, 167 (2004).

[176] D. H. Jacobson and D. Q. Mayne, *Differential Dynamic Programming* (Elsevier, 1970).

[177] A. A. G. Macintosh, *Trajectory Tracking Control for a Small-Scale Helicopter during Autorotation*, Tech. Rep. (Master Thesis, Delft University of Technology, 2014).

[178] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans, *Efficient robust predictive control,* IEEE Trans. Autom. Control **45**, 1545 (2000).

[179] H. J. Ferreau, *An Online Active Set Strategy for Fast Solution of Parametric Quadratic Programs with Applications to Predictive Engine Control*, Ph.D. thesis, Heidelberg University (2006).

[180] P. J. Goulart, E. C. Kerrigan, and D. Ralph, *Efficient robust optimization for robust control with constraints,* Math. Program. Ser. A , 115 (2008).

[181] Y. Wang and S. Boyd, *Fast model predictive control using online optimization,* IEEE Trans. on Control Systems Technology **18**, 267 (2010).

[182] P. Sarmah, C. Chawla, and R. Padhi, *A nonlinear approach for re-entry guidance of reusable launch vehicles using model predictive static programming,* in *Mediterranean Conf. on Control and Automation* (2008).

[183] R. Padhi and M. Kothari, *Model predictive static programming: A computationally efficient technique for suboptimal control design,* Int. J. of Innovative Computing, Information and Control **5** (2009).

[184] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, *The explicit linear quadratic regulator for constrained systems,* Automatica **38**, 3 (2002).

[185] P. Tondel, T. A. Johansen, and A. Bemporad, *Evaluation of piecewise affine control via binary search tree,* Automatica **39**, 945 (2003).

[186] D. M. delaPena, A. Bemporad, and C. Filippi, *Robust explicit mpc based on approximate multiparametric convex programming,* IEEE Trans. Autom. Control **51**, 1399 (2006).

[187] T. Besselmann, J. Lofberg, and M. Morari, *Explicit mpc for lpv systems: Stability and optimality,* IEEE Trans. Autom. Control **57**, 2322 (2012).

[188] M. F. Weilenmann and H. P. Geering, *Test bench for rotorcraft hover control,* AIAA J. of Guidance, Control, and Dynamics **17**, 729 (1994).

[189] P. Bendotti and J. C. Morris, *Robust hover control for a model helicopter,* in *Am. Control Conf.* (1995).

[190] A. Conway, *Autonomous Control of an Unstable Helicopter Using Carrier Phase GPS Only*, Ph.D. thesis, Stanford University (1995).

[191] E. N. Johnson and P. A. DeBitetto, *Modeling and simulation for small autonomous helicopter development,* in *AIAA Modeling and Simulation Technologies Conf.* (1997).

1

[192] N. B. Grady, M. T. Frye, and C. Qian, *The instrumentation and flight testing of a rotorcraft vehicle for undergraduate flight control research,* in *AIAA Modeling and Simulation Technologies Conf. and Exhibit* (2006).

[193] G. Cai, B. M. Chen, K. Peng, M. Dong, and T. H. Lee, *Modeling and control system design for a uav helicopter,* in *Mediterranean Conf. on Control and Automation* (2006).

[194] C. M. Velez, A. Agudelo, and J. Alvarez, *Modeling, simulation and rapid prototyping of an unmanned mini-helicopter,* in *AIAA Modeling and Simulation Technologies Conf. and Exhibit* (2006).

[195] L. Zhao and V. R. Murthy, *Optimal controller for an autonomous helicopter in hovering and forward flight,* in *47th AIAA Aerospace Sciences Meeting* (2009).

[196] S. P. Khaligh, F. Fahimi, and M. Saffarian, *Comprehensive aerodynamic modeling of a small autonomous helicopter rotor at all flight regimes,* in *AIAA Modeling and Simulation Technologies Conf.* (2009).

[197] S. K. Kim and T. M. Tilbury, *Mathematical modeling and experimental identification of a model helicopter,* in *AIAA Modeling and Simulation Technologies Conf.* (1998).

[198] C. Munzinger, *Development of a Real-Time Flight Simulator for an Experimental Model Helicopter*, Tech. Rep. (Master Thesis, Georgia Institute of Technology, 1998).

[199] R. Cunha and C. Silvestre, *Dynamic modeling and stability analysis of model-scale helicopters with bell-hiller stabilzing bar,* in *AIAA Guidance, Navigation, and Control Conf.* (2003).

[200] T. Sudiyanto, A. Budiyono, and H. Y. Sutarto, *Hardware in-the-loop simulation for control system designs of model helicopter,* in *Aerospace Indonesia Meeting, Bandung* (2005).

[201] U. B. Hald, M. V. Hesselbaek, and M. Siegumfeldt, *Nonlinear Modeling and Optimal Control of a Miniature Autonomous Helicopter*, Tech. Rep. (Master Thesis, Aalborg University, 2006).

[202] S. Bhandari and R. Colgren, *6-dof dynamic model for a raptor 50 uav helicopter including stabilizer bar dynamics,* in *AIAA Modeling and Simulation Technologies Conf.* (2006).

[203] G. Cai, B. M. Chen, T. H. Lee, and K. Y. Lum, *Comprehensive nonlinear modeling of a miniature unmanned helicopter,* J. of the Am. Helicopter Soc. **57** (2012).

[204] T. L. Andersen, D. F. Lauritzen, J. T. Madsen, M. M. Sorensen, and B. A. Mertz, *Autonomous Inverted Hover of a Small Scale Helicopter*, Tech. Rep. (Aalborg University, 2008).

[205] G. Buskey, O. Roberts, P. Corke, P. Ridley, and G. Wyeth, *Sensing and Control for a Small-Size Helicopter. in Experimental Robotics VIII, pp. 476-486* (Springer, Berlin/Heidelberg, 2003).

[206] O. Amidi, *An Autonomous Vision-Guided Helicopter*, Ph.D. thesis, Carnegie Mellon University (1996).

[207] O. Shakernia, C. S. Sharp, R. Vidal, D. H. Shim, Y. Ma, and S. Sastry, *Multiple view motion estimation and control for landing an unmanned aerial vehicle,* in *Int. Conf. on Robotics and Automation* (2002).

[208] J. M. Roberts, P. I. Corke, and G. Buskey, *Low-cost flight control system for a small autonomous helicopter,* in *Australian Conf. on Robotics and Automation* (2002).

[209] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, *Visually-guided landing of an autonomous aerial vehicle,* IEEE Trans. on Robotics & Automation **19**, 371 (2003).

[210] A. E. Barabanov, N. Y. Vazhinsky, and D. V. Romaev, *Full autopilot for small electrical helicopter,* in *Europ. Rotorcraft Forum* (2007).

[211] V. Gavrilets, B. Mettler, and E. Feron, *Nonlinear model for a small-size acrobatic helicopter,* in *AIAA Guidance, Navigation and Control Conf.* (2001).

[212] L. Marconi and R.Naldi, *Robust nonlinear control for a miniature helicopter for aerobatic maneuvers,* in *Europ. Rotorcraft Forum* (2006).

[213] W. Johnson, *CAMRAD/JA: A Comprehensive Analytical Model of Rotorcraft Aerodynamics and Dynamics, Volume I: Theory Manual*, Tech. Rep. (Johnson Aeronautics, 1988).

[214] ART, (http://www.flightlab.com/, Mountain View CA., U.S.A.).

[215] J. Howlett, *UH-60A Black Hawk Engineering Simulation Program: Volume I - Mathematical Model*, Tech. Rep. CR-166309 (NASA, 1981).

[216] B. Benoit, A. M. Dequin, K. Kampa, W. von Grunhagen, P. M. Basset, and B. Gimonet, *Host a general helicopter simulation tool for germany and france,* in *Am. Helicopter Soc.* (2000).

[217] J. D. Anderson, *Fundamentals of Aerodynamics. Third Ed.* (McGraw-Hill Higher Education, NY, 2001).

[218] D. A. Peters and N. HaQuang, *Technical notes - dynamic inflow for practical applications,* J. of the Am. Helicopter Soc. , 64 (1988).

[219] D. A. Peters and C. J. He, *Finite state induced flow models part ii: Three-dimensional rotor disk,* AIAA J. of Aircraft **32**, 323 (1995).

[220] R. T. N. Chen, *Flap-Lag Equations of Motion of Rigid, Articulated Rotor Blades with Three Hinge Sequences*, Tech. Rep. NTM 100023 (NASA Ames Research Center, 1987).

[221] D. M. Pitt and D. A. Peters, *Theoretical prediction of dynamic-inflow derivatives,* Vertica **5**, 21 (1981).

[222] D. A. Peters and C. He, *Technical note: Modification of mass-flow parameter to allow smooth transition between helicopter and windmill states,* J. of the Am. Helicopter Soc. , 275 (2006).

[223] R. T. N. Chen, *A Simplified Rotor System Mathematical Model for Piloted Flight Dynamics Simulation*, Tech. Rep. NTM 78575 (NASA Ames Research Center, 1979).

[224] R. T. N. Chen, *Effects of Primary Rotor Parameters on Flapping Dynamics*, Tech. Rep. NTP 1431 (NASA Ames Research Center, 1980).

[225] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York, 1988).

[226] D. A. Benson, *A Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. thesis, Massachusetts Institute of Technology (2004).

[227] G. T. Huntington, *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. thesis, Massachusetts Institute of Technology (2007).

[228] D. Bertsekas, *Nonlinear Programming - Second Edition* (Athena Scientific, Belmont, MA, 1999).

[229] P. E. Gill, W. Murray, and M. A. Saunders, *Snopt: An sqp algorithm for large-scale constrained optimization,* SIAM Reviews **47**, 99 (2002).

[230] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*, Tech. Rep. (University of California San Diego, 2008).

[231] SBS-Inc., (http://www.sbsi-sol-optimize.com/asp/sol_product_snopt.htm, Stanford University, U.S.A.).

[232] M. Fliess, *Generalized controller canonical forms for linear and nonlinear dynamics,* IEEE Trans. Autom. Control **35**, 994 (1990).

[233] M. Fliess, J. Levine, P. Martin, and P. Rouchon, *Sur les systemes non lineaires differentiellement plats,* C.R. Acad. Sci. Paris **315**, 619 (1992).

[234] M. Fliess, J. Levine, P. Martin, and P. Rouchon, *Flatness and defect of nonlinear systems: Introductory theory and examples,* Int. J. of Control **61**, 1327 (1995).

[235] H. Sira-Ramirez and S. K. Agrawal, *Differentially Flat Systems* (Marcel Dekker Inc., 2004).

[236] J. Levine, *Analysis and Control of Nonlinear Systems: A Flatness-Based Approach* (Springer, 2009).

[237] I. M. Ross and F. Fahroo, *A unified computational framework for real-time optimal control,* in *IEEE Conf. on Decision and Control* (2003).

[238] P. Martin, R. M. Murray, and P. Rouchon, *Flat Systems, Equivalence And Trajectory Generation*, Tech. Rep. (CDS, California Institute of Technology, 2003).

[239] R. Murray, M. Rathinam, and W. Sluis, *Differential flatness of mechanical control systems: A catalog of prototype systems,* in *ASME Int. Mechanical Engineering Congress* (1995).

[240] M. Rathinam, *Differentially Flat Nonlinear Control Systems*, Ph.D. thesis, California Institute of Technology (1997).

[241] V. N. Chetverikov, *New flatness conditions for control systems,* in *IFAC Nonlinear Control Systems* (2001).

[242] P. Martin, *Contribution a l'Etude des Systemes Differentiellement Plats*, Ph.D. thesis, Ecole des Mines de Paris (1992).

[243] P. Martin, *Aircraft control using flatness,* in *Proc. CESA Lille* (1996).

[244] J. Hauser and R. Hindman, *Aggressive flight maneuvers,* in *IEEE Conf. on Decision and Control* (1997).

[245] F. Cazaurang, R. Bergeon, and L. Lavigne, *Modelling of longitudinal disturbed aircraft model by flatness approach,* in *AIAA Guidance Navigation and Control Conf.* (2003).

[246] A. Ailon, *Optimal path and tracking control of an autonomous vtol aircraft,* in *Mediterranean Conf. on Control and Automation* (2006).

[247] W. C. Lu, L. Duan, F. B. Hsiao, and F. M. Camino, *Neural guidance control for aircraft based on differential flatness,* AIAA J. of Guidance, Control, and Dynamics **31**, 892 (2008).

[248] M. van Nieuwstadt and R. M. Murray, *Outer flatness: Trajectory generation for a model helicopter,* in *Europ. Control Conf.* (1997).

[249] T. J. Koo and S. Sastry, *Differential flatness based full authority helicopter control design,* in *IEEE Conf. on Decision and Control* (1999).

[250] H. Sira-Ramirez, R. Castro-Linares, and E. Liceaga-Castro, *A liouvillian systems approach for the trajectory planning-based control of helicopter models,* Int. J. Of Robust And Nonlinear Control **10**, 301 (2000).

[251] G. Zames, *On the input-output stability of time-varying nonlinear feedback systems-part i: Conditions derived using concepts of loop gain, conicity, and positivity,* IEEE Trans. Autom. Control **AC-11**, 228 (1966).

[252] C. A. Desoer and M. Vidyasagar, *Feedback Systems: Input-Output Properties* (Academic Press, New York, 1975).

1

[253] J. S. Shamma and M. Athans, *Analysis of gain scheduled control for nonlinear plants,* IEEE Trans. Autom. Control **35**, 898 (1990).

[254] W. Rugh and J. Shamma, *Research on gain scheuling,* Automatica **36**, 1401 (2000).

[255] A. Packard, *Gain scheduling via linear fractional transformations,* Systems & Control Letters **22**, 79 (1994).

[256] P. Apkarian, P. Gahinet,   and G. Becker, *Self-scheduled h_∞ control of linear parameter-varying systems: A design example,* Automatica **31**, 1251 (1995).

[257] P. Apkarian and P. Gahinet, *A convex characterization of gain-scheduled h_∞ controllers,* IEEE Trans. Automatic Control **40**, 853 (1995).

[258] W. M. Lu and J. C. Doyle, *h_∞ control of nonlinear systems: A convex characterization,* IEEE Trans. Automatic Control **40**, 1668 (1995).

[259] F. Wu, X. H. Yang, A. K. Packard, and G. Becker, *Induced $\mathcal{L}_2$ norm control for lpv systems with bounded parameter variation rates,* Int. J. Of Robust And Nonlinear Control **6**, 983 (1996).

[260] J. Yu and A. Sideris, *h_∞ control with parametric lyapunov functions,* Systems & Control Letters **30**, 57 (1997).

[261] G. Scorletti and L. ElGhaoui, *Improved lmi conditions for gain scheduling and related control problems,* Int. J. of Robust and Nonlinear Control **8**, 845 (1998).

[262] C. W. Scherer, *Lpv control and full block multipliers,* Automatica **37**, 361 (2001).

[263] G. Stein, *Adaptive Flight Control - A Pragmatic View. In K. S. Narendra and R. V. Monopoli (Eds), Applications of Adaptive Control* (Academic Press, New York, 1980).

[264] K. J. Astrom and B. Wittenmark, *Adaptive Control* (Addison-Wesley, Reading, MA, 1989).

[265] S. Boyd, L. ElGhaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory* (Soc. for Industrial and Applied Mathematics (SIAM), Philadelphia, 1994).

[266] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2004).

[267] Y. Nesterov and A. Nemirovski, *Interior Point Polynomial Methods in Convex Programming* (Soc. for Industrial & Applied Math, 1994).

[268] J. F. Sturm, *Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones,* Optimization Methods and Software **11-12**, 625 (1999).

[269] G. J. Balas, I. Fialho, A. Packard, J. Renfrow, and C. Mullaney, *On the design of lpv controllers for the f-14 aircraft lateral-directional axis during powered approach,* in *Am. Control Conf.* (1997).

[270] G. J. Balas, J. B. Mueller, and J. M. Barker, *Application of gain-scheduled, multi-variable control techniques to the f/a-18 system research aircraft,* in *AIAA Guidance Navigation and Control Conf.* (1999).

[271] I. Fialho, G. J. Balas, A. K. Packard, J. Renfrow, and C. Mullaney, *Gain-scheduled lateral control of the f-14 aircraft during powered approach landing,* AIAA J. of Guidance, Control, and Dynamics **23**, 450 (2000).

[272] A. Marcos and G. J. Balas, *Development of linear-parameter-varying models for aircraft,* AIAA J. of Guidance, Control, and Dynamics **27**, 218 (2004).

[273] L. Reberga, D. Henrion, J. Bernussou, and F. Vary, *Lpv modeling of a turbofan engine,* in *IFAC World Congress* (2005).

[274] L. Giarre, D. Bausoa, P. Falugi, and B. Bamieh, *Lpv model identification for gain scheduling control: An application to rotating stall and surge control problem,* Control Engineering Practice **14**, 351 (2006).

[275] K. Z. Ostergaard, *Robust, Gain-Scheduled Control of Wind Turbines*, Ph.D. thesis, Aalborg University (2008).

[276] M. Steinbuch, R. van de Molengraft, and A. J. van der Voort, *Experimental modelling and lpv control of a motion system,* in *Am. Control Conf.* (2003).

[277] M. G. Wassink, M. van de Wal, C. Scherer, and O. Bosgra, *Lpv control for a wafer stage: Beyond the theoretical solution,* Control Engineering Practice **13**, 231 (2004).

[278] M. Dettori, *LMI Techniques for Control*, Ph.D. thesis, Delft University of Technology (2001).

[279] S. M. Hashemi, H. S. Abbas, and H. Werner, *Lpv modelling and control of a 2-dof robotic manipulator using pca-based parameter set mapping,* in *IEEE Conf. on Decision and Control* (2009).

[280] R. Toth, *Identification and Modeling of Linear Parameter-Varying Systems. Lecture Notes in Control and Information Sciences, Vol. 403* (Springer, Heidelberg, 2010).

[281] A. Kwiatkowski and H. Werner, *Pca-based parameter set mappings for lpv models with fewer parameters and less overbounding,* IEEE Trans. on Control Systems Technology **16**, 781 (2008).

[282] H. Abbas and H. Werner, *Frequency-weighted discrete-time lpv model reduction using structurally balanced truncation,* IEEE Trans. on Control Systems Technology **19**, 140 (2011).

[283] S. Taamallah, *Small-scale helicopter blade flap-lag equations of motion for a flybarless pitch-lag-flap main rotor,* in *AIAA Modeling and Simulation Technologies Conf.* (2011).

[284] S. Taamallah, *Flight dynamics modeling for a small-scale flybarless helicopter uav,* in *AIAA Atmospheric Flight Mechanics Conf.* (2011).

1

[285] S. Taamallah, *Low-order modeling for a small-scale flybarless helicopter uav, a grey-box time-domain approach,* in *AIAA Atmospheric Flight Mechanics Conf.* (2012).

[286] S. Taamallah, *Optimal autorotation with obstacle avoidance for a small-scale flybarless helicopter uav,* in *AIAA Guidance, Navigation and Control Conf.* (2012).

[287] S. Taamallah, X. Bombois, and P. M. J. Van den Hof, *Optimal control for power-off landing of a small-scale helicopter a pseudospectral approach,* in *Am. Control Conf.* (2012).

[288] S. Taamallah, *Flatness based trajectory generation for a helicopter uav,* in *AIAA Guidance, Navigation and Control Conf.* (2013).

[289] S. Taamallah, X. Bombois, and P. M. J. Van den Hof, *Affine lpv modeling: An h_∞ based approach,* in *IEEE Conf. on Decision and Control* (2013).

1

2

HIGH-ORDER MODELING OF THE HELICOPTER DYNAMICS

All models are wrong, but some are useful.

George E. P. Box
Robustness in the strategy of scientific model building, 1979

In this Chapter we present a high-order, helicopter flight dynamics NonLinear (NL) model for a flybarless main rotor, with rigid blades. The model incorporates the main rotor, tail rotor, fuselage, and tails. The novel part of this Chapter is twofold. Our first contribution consists in deriving the coupled flap-lag equations of motion, for a rigid, flybarless, articulated rotor, with a Pitch-Lag-Flap (P-L-F) rotor hinge sequence, particularly suited for small-scale Unmanned Aerial Vehicles (UAVs). The second contribution is the development of a comprehensive flight dynamics model for a small-scale helicopter UAV, for both ClockWise (CW) or Counter-ClockWise (CCW) main rotor rotation, applicable for high bandwidth control specifications, and valid for a range of flight conditions, including (steep) descent flight into the Vortex-Ring-State (VRS)[1] and autorotation. Additionally, the Chapter reviews all assumptions made in deriving the model, i.e. structural, aerodynamics, and dynamical simplifications. Simulation results show that this high-order NL model is in good agreement with an equivalent FLIGHTLAB[2] model, for both static (trim) and dynamic conditions.

Parts of this Chapter have been published in [1–3].

[1]Briefly summarized, the VRS corresponds to a condition where the helicopter is descending in its own wake, resulting in a chaotic and dangerous flight condition [1].

[2]FLIGHTLAB is a state of the art modeling, analysis and real-time simulation tool, used world-wide to simulate helicopter flight dynamics [4].

2.1. INTRODUCTION

IN this Chapter we develop a comprehensive, MATLAB-based, white-box[3], nonlinear, continuous-time, High-Order Model (HOM), used as a realistic small-scale helicopter simulation environment, for the validation of the Flight Control System (FCS). This helicopter model is applicable for high bandwidth control specifications, and is valid for a range of flight conditions, including (steep) descent flight into the VRS and autorotation [1, 5]. This HOM will, in subsequent Chapters, be used for controller validation. For controller design however, and due to its complexity, only approximation of this HOM will be used in the upcoming Chapters.

The helicopter model, developed in this Chapter, replaces the true system, and is based upon our work presented in [2, 3]. This model aims at simulating the helicopter flight dynamics for the case of a flybarless, articulated, Pitch-Lag-Flap (P-L-F) main rotor with rigid blades, for both ClockWise (CW) or Counter-ClockWise (CCW) main rotor rotation[4]. The model incorporates the rigid-body dynamics, main rotor, tail rotor, fuselage, and tails. The complete simulation environment, i.e. including the control system, is sketched in the block diagram of Fig. 2.1, which illustrates all internal subsystems.
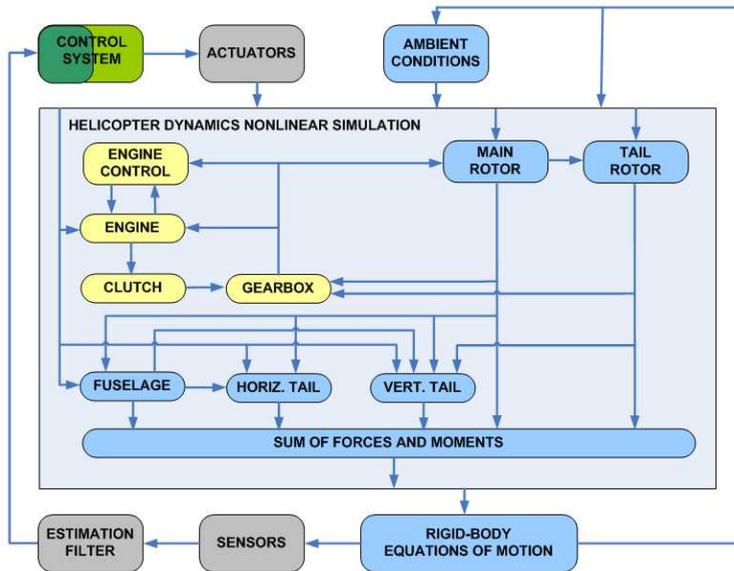
Figure 2.1: Helicopter simulation environment (derived from [6]). The components of the helicopter simulation are visualized in blue, whereas in yellow we visualize the helicopter simulation components that are not relevant for our autorotation application, and thus neglected (i.e. not modeled).

---

[3]Based upon first-principles.

[4]A CW or CCW main rotor refers to the main rotor blade rotation when viewed from above. CCW rotation is common to American, British, German, Italian, and Japanese helicopter designs, whereas CW rotation is standard on Chinese, French, Indian, Polish and Russian helicopters designs.

In Fig. 2.1, the Main Rotor (MR) determines the aerodynamic lift force that supports the weight of the helicopter, and the thrust that counteracts aerodynamic drag in forward flight. It is also through the main rotor that vehicle roll angle, vehicle pitch angle, and vehicle vertical motion are controlled, see also Section 1.2. On the other hand, the Tail Rotor (TR) provides torque balance, directional stability, and yaw angle (heading) control of the helicopter. The role of the Vertical Tail (VT) is twofold: 1) in forward flight, it generates a sideforce and yawing moment, hence reducing the tail rotor thrust requirement; and 2) during maneuvers, and during wind gusts, it provides yaw damping and stiffness, enhancing directional stability [7]. The role of the Horizontal Tail (HT) is also twofold: 1) in forward flight, it generates a load that reduces the main rotor fore-aft flapping; and 2) during maneuvers, and during wind gusts, it provides pitch damping and stiffness, enhancing pitch stability [7]. The Ambient Conditions defines the outside air density and temperature, whereas the Rigid-Body Equations Of Motion computes the positions, orientations, and velocities of the vehicle in three-dimensional (3D) inertial space.

The remainder of this Chapter is organized as follows. In Section 2.2, our small-scale helicopter modeling framework is outlined. In Section 2.3, model validation results are analyzed. In Section 2.4, an analysis of the rigid-body dynamics, in open-loop, is presented. In Section 2.5, conclusions and future directions are presented. Further, in Appendix A and B the nomenclature and frames are presented. In Appendix C, the rigid-body equations of motion are summarized. In Appendix D and E, main and tail rotor models are discussed. In Appendix F, the fuselage model is reviewed. In Appendix G, comments are made on the vertical and horizontal tail models.

2.2. Helicopter modeling: general overview

From Fig. 1.15, and zooming on the 'Helicopter Dynamics Nonlinear Simulation' block, we obtain Fig. 2.2 which gives additional insight into the model. We have the control input-vector **u** of dimension four, and the state-vector **x** of dimension twenty-four. The states include the twelve-states rigid-body motion (states given in blue), and the dynamics of the main rotor (states given in red). The former include the three-states inertial position, the three-states body linear velocities, the three-states body rotational velocities, and the three-states attitude (orientation) angles, see Fig. 2.2. The dynamics of the main rotor include the helicopter higher frequency phenomena, which exist for both the engine ON or OFF (i.e. autorotation) flight condition. These include the main rotor three-states dynamic inflow [8, 9], and main rotor blade flap-lag dynamics, derived through the Lagrangian method [10] (each blade is defined by the four-states flap/lag angles and rotational velocities) [11], see Fig. 2.2. Regarding the main rotor Revolutions Per Minute (RPM), it is generally assumed fixed for the engine ON case[5], whereas for the engine OFF case it is not fixed anymore. Indeed, the main rotor RPM represents an essential part of the autorotative flight condition, and this additional state needs to be included in the state-vector **x** when considering the engine OFF case, see Fig. 2.2.

---

[5]Although this is a simplification, since in the engine ON case the main rotor RPM is being regulated by the governor.

Figure 2.2: Helicopter Inputs **u** (in green), States **x** (in blue the rigid-body states, in red the main rotor states), and Measurements **y** (measured states).

Other model components include: 1) the tail rotor, modeled as a standard Bailey type rotor [12]; 2) the fuselage, based upon aerodynamic lift and drag coefficients, which are tabulated as a function of airflow Angle Of Attack (AOA)[6] and sideslip[7] angles; and 3) the horizontal and vertical tails, based upon standard flat plate models. Next, there is the vector of measured outputs $\mathbf{y}$ of dimension twelve. The measurements are given by $\mathbf{y} = \mathbf{x}_{(1:12)}$, with $\mathbf{x}_{(1:12)}$ a shorthand for the first twelve states of $\mathbf{x}$, i.e. the rigid-body states (see also the nomenclature in Appendix A). Expressing the fundamental Newtonian laws [14] in the vehicle body frame F_b, we get (refer also to Appendix B and Fig. 2.16)

$$
\begin{aligned}
m_V.\mathbf{A}^b_{CG} &= m_V.\mathbf{g}^b + \mathbf{F}^b_{CG} \\
\tfrac{d\mathbf{H}^b_{CG}}{dt} &= \mathbf{M}^b_{CG}
\end{aligned}
\tag{2.1}
$$

with $\mathbf{A}^b_{CG}$ the inertial (i.e. relative to frame F_I, refer also to Appendix B and Fig. 2.15) acceleration of the vehicle Center of Gravity (CG) in F_b, $\mathbf{H}^b_{CG}$ the inertial angular momentum of the vehicle CG in F_b, $\mathbf{F}^b_{CG}$ the aerodynamic forces experienced by the vehicle CG in F_b, $\mathbf{M}^b_{CG}$ the moments of aerodynamic forces experienced by the vehicle CG in F_b, m_V the vehicle mass, $\mathbf{g}^b$ the acceleration due to gravity in F_b (refer also to the nomenclature given in Appendix A).

Now, $\mathbf{H}^b_{CG}$ is given by

$$
\mathbf{H}^b_{CG} = \mathbb{I}_V.\mathbf{\Omega}^b_{bI}
\tag{2.2}
$$

with $\mathbb{I}_V$ the vehicle inertia matrix in F_b, and $\mathbf{\Omega}^b_{bI}$ the vehicle angular velocity with respect to F_I projected in F_b. Combining Eq. (2.1) and Eq. (2.2), we can express the helicopter flight dynamics model as a set of first-order, Ordinary Differential Equations (ODEs) of the form

$$
\forall t \geq 0 \qquad \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))
\tag{2.3}
$$

with $f(\cdot)$ a continuous-time function, $\mathbf{x}$ the state-vector of dimension twenty-four, and $\mathbf{u}$ the input-vector of dimension four. Appendices C through G present a detailed derivation of the model given in Eq. (2.3).

2.3. MODEL EVALUATION AND VALIDATION

The purpose of this section is to evaluate, and validate, the open-loop behavior of our white-box helicopter mathematical model. Model validation can either be done by comparing the model's behavior with several recorded experimental data sets (i.e. flight tests), or by comparing the model's behavior with another simulation model, which is often a third-party, high-fidelity black-box model. In this thesis, since flight data is not available, we opted for the second option, namely the use of the FLIGHTLAB [4] helicopter simulation environment. For aerospace systems, the model validation task generally involves the validation of, both, the static (trim) behavior as well as the dynamic response. A trim condition sets

---

[6]The AOA is the angle between a reference line on a body and the velocity vector representing the relative motion between the body and the air [13].

[7]Sideslip flight refers to a vehicle moving somewhat sideways as well as forward, relative to the oncoming airflow.

the helicopter in some, user-defined, steady-state (i.e. equilibrium) flight condition, by satisfying the system's equations of motion. Trim settings are often a prerequisite for stability analysis, vibration studies, and control system design. For instance, for linear control design, the linear models are generally obtained through analytical or numerical linearizations of the NL model, around various trim conditions. Next, for the validation of the dynamic behavior, either time-domain model responses or frequency-responses can be used.

We compare next trim and time-response outputs of our MATLAB-based model with those from a FLIGHTLAB model, for the case of a small-scale helicopter UAV. This modeled UAV is an instrumented Remote-Controlled (RC) Align T-REX helicopter, belonging to the flybarless two-bladed main rotor class, with a total mass of 7.75 kg, a main rotor radius of 0.9 m, a main rotor nominal angular velocity of 1350 RPM, a NACA 0015 main rotor airfoil, and with fuselage aerodynamic lookup tables obtained by scaling-down a full-size Bo-105 helicopter fuselage aerodynamic model. The NACA 0015 and fuselage lookup tables are not reproduced here due to space constraints, however the remaining parameters have been listed in Table 2.1[8]. For this helicopter UAV, the Reynolds numbers vary approximately in the range 10^5 –7.10^5, and hence these Reynolds numbers do not induce any particular limitations from an aerodynamic standpoint. For example, The Pitt-Peters dynamic inflow model (used in our main rotor model) has been successfully applied on systems with Reynolds numbers as low as 10^4 [15].

Our model is compared to an equivalent FLIGHTLAB model, the latter having the following options selected:

- Articulated main rotor.

- Blade element model and quasi-steady airloads.

- Peters-He three-state inflow model, with no stall delay.

- Bailey-type tail rotor.

2.3.1. Trim results

A trim condition is equivalent to an equilibrium point of Eq. (2.3) [16, 17], which can be thought of as a specific flight condition, in which the resultant forces and moments on the vehicle are equal to zero. For helicopters however, the concept of trim is more complicated than that of fixed-wing aircrafts [18], since a helicopter has components that rotate with respect to each other and with respect to the air mass. To circumvent this problem we developed a trim module, in the form of a constrained, nonlinear, optimization problem. At trim, the resultant forces and moments on the vehicle should be equal to zero, hence for the engine ON flight condition, the objective of the trim module is to set to zero the three vehicle inertial linear accelerations ($\dot{V}_N$, $\dot{V}_E$, $\dot{V}_Z$) and the three vehicle rotational accelerations ($\dot{p}, \dot{q}, \dot{r}$). On the other hand for the engine OFF flight condition (i.e. autorotation), the main rotor RPM Ω_{MR} is not fixed anymore as it is allowed to vary according to its own dynamics. Thus, we consider here two cases for the engine OFF trim module.

---

[8]In this table the acronym *wrt* stands for *with respect to*.

Table 2.1: Align T-REX physical parameters for the environment, vehicle, and actuators.

	Name	Parameter	Value	Unit
	Air density	ρ	1.2367	kg/m^3
	Static temperature	T	273.15 + 15	K
Environment	Specific heat ratio (air)	γ	1.4	
	Gas constant (air)	R	287.05	$J/kg.K$
	Gravity constant	g	9.812	m/s^2
	Total mass	m	7.75	kg
	Inertia moment wrt x_b	A	0.2218	$kg.m^2$
	Inertia moment wrt y_b	B	0.5160	$kg.m^2$
Vehicle	Inertia moment wrt z_b	C	0.3141	$kg.m^2$
	Inertia product wrt x_b	D	0	$kg.m^2$
	Inertia product wrt y_b	E	0.0014	$kg.m^2$
	Inertia product wrt z_b	F	0	$kg.m^2$
	X-pos. of Fus. CG wrt total CG	x_{Fus}	0	m
	Y-pos. of Fus. CG wrt total CG	y_{Fus}	0	m
	Z-pos. of Fus. CG wrt total CG	z_{Fus}	0.017	m
	MR collective	θ_0	$[-13,13].\pi/180$	rad
	MR lateral cyclic	θ_{1c}	$[-6,6].\pi/180$	rad
Actuators	MR longitudinal cyclic	θ_{1s}	$[-6,6].\pi/180$	rad
	TR collective	θ_{TR}	$[-20,20].\pi/180$	rad
	MR collective rate	$\dot{\theta}_0$	$[-52,52].\pi/180$	rad/s
	MR lateral cyclic rate	$\dot{\theta}_{1c}$	$[-52,52].\pi/180$	rad/s
	MR longitudinal cyclic rate	$\dot{\theta}_{1s}$	$[-52,52].\pi/180$	rad/s
	TR collective rate	$\dot{\theta}_{TR}$	$[-120,120].\pi/180$	rad/s

1. The objective of the first engine OFF trim consists in setting to zero the previous six accelerations, defined for the engine ON case, together with an additional acceleration, namely the one related to main rotor RPM $\dot{\Omega}_{MR}$. This allows to find the steady-state autorotative flight conditions.

2. For low altitude engine OFF conditions, e.g. below 30–40 m in the case of our helicopter, as well as during the autorotation entry phase, and flare[9] phase, see Section 1.3.1, we observed, through various simulation runs, that steady-state autorotations was seldom reached. Rather, for those situations, the helicopter is in a continuous transition from one non-equilibrium condition to the next. Hence, the objective of the second engine OFF trim consists in only setting to zero the six accelerations defined for the engine ON case[10].

---

[9]The flare refers to the landing maneuver just prior to touch-down. In the flare the nose of the vehicle is raised in order to slow-down the descent rate, and further the proper attitude is set for touchdown.

[10]This second engine OFF trimming approach has shown to be feasible only for low-speed flight conditions.

(Table 1 cont'd): Align T-REX physical parameters for the main rotor.

	ClockWise direction of rotation	Γ	-1	
Main	Number of blades	N_b	2	
Rotor	Nominal angular velocity	$\Omega_{MR_{100\%}}$	141.37	rad/s
(MR)	Rotor radius from hub	R_{rot}	0.9	m
	Blade mass	M_{bl}	0.2875	kg
	Spring restraint coef. due to flap	K_{S_β}	162.69	$N.m/rad$
	Spring damping coef. due to flap	K_{D_β}	0	$N.m.s/rad$
	Spring restraint coef. due to lag	K_{S_ζ}	0	$N.m/rad$
	Spring damping coef. due to lag	K_{D_ζ}	5	$N.m.s/rad$
	Offset distance	e_P	0.03	m
	Offset distance	e_L	0.06	m
	Offset distance	e_F	0.01	m
	Distance between hub and flap hinge	Δ_e	0.1	m
	Root cutout from flap hinge	r_c	0.0	m
	Blade chord	c_{bl}	0.064	m
	Blade twist at tip	θ_{wash}	0	rad
	Y-pos. blade CG wrt flap hinge	$y_{G_{bl}}$	0.4	m
	Swashplate phase angle	ψ_{PA}	0	rad
	Precone angle	β_P	0	rad
	Pitch-flap coupling ratio	$K_{(\theta\beta)}$	0	
	Pitch-lag coupling ratio	$K_{(\theta\zeta)}$	0	
	Tip loss factor	B	0.97	
	Airfoil lift coef.	$c_{l_{bl}}$	NACA0015	
	Airfoil drag coef.	$c_{d_{bl}}$	NACA0015	
	Airfoil pitching moment coef.	c_M	NACA0015	
	X-pos. of MR hub wrt total CG	x_H	0.01	m
	Y-pos. of MR hub wrt total CG	y_H	0	m
	Z-pos. of MR hub wrt total CG	z_H	-0.213	m

Note that both of these engine OFF trim modules will be used in the sequel. Now, the variables that the trim algorithm is allowed to manipulate include the four control inputs $(\theta_0, \theta_{1c}, \theta_{1s}, \theta_{TR})$, and the vehicle roll and pitch angles (ϕ, θ), since the latter two influence the projection of the gravity vector on the body frame. Besides, the set-point at which the equilibrium is computed has to be specified in the form of additional constraints, i.e. by assigning fixed values to the three vehicle inertial linear velocities[11] (V_N, V_E, V_Z), and the three vehicle rotational velocities (p, q, r). Now regarding the dynamic inflow states $(\lambda_0, \lambda_s, \lambda_c)$, and the periodic states, i.e. blade flap and lag angles and velocities $(\beta_{bl}, \zeta_{bl}, \dot{\beta}_{bl}, \dot{\zeta}_{bl})$, these states are handled by time-marching the NL helicopter model long enough until the transients have decayed. Finally, the remaining four states which include

---

[11]The three vehicle inertial linear velocities may be assigned any fixed values, hence for non-zero values this implies that the vehicle position is not in trim. Seen from this perspective, not all the states are in equilibrium.

(Table 1 cont'd): Align T-REX physical parameters for the tail rotor.

Tail Rotor (TR)	Number of blades	$N_{b_{TR}}$	2	
	Nominal angular velocity	$\Omega_{TR_{100\%}}$	612.61	rad/s
	Rotor radius from rotor hub	$R_{rot_{TR}}$	0.14	m
	Pitch-flap coupling	$\delta_{3_{TR}}$	0	rad
	Preset collective pitch bias	$\theta_{bias_{TR}}$	0	rad
	Partial coning angle wrt thrust	$\beta_{0_{TR}}$	0	rad/N
	Tail blockage constant	b_{t_1}	0.927	
	Transition velocity	v_{bl}	20	m/s
	Blade chord	c_{TR}	0.0316	m
	Tip loss factor	B_{TR}	0.92	
	Airfoil lift curve slope	$c_{l_{(0,TR)}}$	5.92	rad^{-1}
	Blade drag coef.	CD_{TR}	0.0082	
	X-pos. of TR hub wrt total CG	x_{TR}	-1.015	m
	Y-pos. of TR hub wrt total CG	y_{TR}	-0.0575	m
	Z-pos. of TR hub wrt total CG	z_{TR}	-0.034	m

the three vehicle Cartesian position (x_N, x_E, x_Z) and the vehicle heading ψ are left free, since the position of the helicopter does not influence[12] its dynamic behavior or stability. Our trim optimization is further based upon a Newton iteration scheme, similar to that of [19], which is simple to implement and has been widely used [20]. The Newton method guarantees quadratic local convergence, but is known to be sensitive to starting values[13].

We compare next our model trim results, with those obtained from FLIGHTLAB, for the engine ON case only. Comparison of our model with FLIGHTLAB, for the engine OFF case, is presented within the context of dynamic results in Section 2.3.2. First, Table 2.2 gives the maximum absolute trim deviations, as a function of inertial linear velocities[14] (V_N, V_E, V_Z), between our model and FLIGHTLAB, for the six trim variables, i.e. the four control inputs $(\theta_0, \theta_{1c}, \theta_{1s}, \theta_{TR})$ and roll and pitch angles (ϕ, θ). Table 2.2 has to be read in conjunction with Fig. 2.3–Fig. 2.8, where the trim results are plotted, along each motional axis. These motional axes are: longitudinal along V_N, lateral along V_E, vertical climb along V_Z ($V_Z > 0$), and vertical descent along V_Z ($V_Z < 0$). Basically, Fig. 2.3–Fig. 2.8 visualize the trim results for each motional axis at a time, i.e. by setting to zero the velocities along the remaining motional axes, whereas Table 2.2 compiles the worst-case data from Fig. 2.3–Fig. 2.8 by reporting the worst-case trim deviation, for each of the six trim vari-

---

[12]Although strictly speaking this is not true in vertical flight, due to the ground effect when trimming near the ground, and due to changes in air density when trimming with a non-zero vertical velocity; however for the case of air density variations, these may be neglected when considering small-scale UAV applications, since the maximum flight altitude is generally below 150m above ground.

[13]Even with good starting values, it is well-known that the Newton method may at times exhibit erratic divergence due to for example numerical corruption [20]. Hence, several other trim approaches have been researched over the past years, for a review of helicopter trim strategies see among others [7, 16, 18, 20–24].

[14]With V_Z positive up.

Table 2.2: Trim: maximum absolute deviations between our model and FLIGHTLAB, for the engine ON case.

Name	Maximum absolute deviations			
	longi-tudinal along V_N	lateral along V_E	climb along V_Z ($V_Z > 0$)	descent along V_Z ($V_Z < 0$)
Roll ϕ (°)	1.0	0.7	1.5	0.5
Pitch θ (°)	0.3	0.7	0.3	0.1
MR Collective θ_0 (°)	0.5	0.5	0.5	1.5
TR Collective θ_{TR} (°)	0.9	0.9	1.0	2.1
MR Lat. Cyclic θ_{1c} (°)	0.4	0.04	0.04	0.05
MR Long. Cyclic θ_{1s} (°)	0.1	0.5	0.1	0.3
MR Power P_{MR} (W)	59	58	76	156

ables, along each motional axis. In addition, Table 2.2 reports the results for the main rotor power P_{MR}, as this latter gives extra insight into the fidelity of our model.

We see that the maximum absolute deviations, between both models, for roll and pitch angles, are almost negligible, respectively below 1.5 ° and 0.7 °, see Table 2.2. For the remaining variables, we also explore the relative deviations between both models. Regarding the control inputs, Table 2.3 gives their relative deviations in %, namely the maximum absolute deviations divided by the full actuator ranges.

Table 2.3: Trim: maximum relative deviations between our model and FLIGHTLAB, for the control inputs in % of full actuator ranges, for the engine ON case.

Name	Maximum relative deviations (in %)			
	longi-tudinal along V_N	lateral along V_E	climb along V_Z ($V_Z > 0$)	descent along V_Z ($V_Z < 0$)
MR Collective θ_0	1.9	1.9	1.9	5.8
TR Collective θ_{TR}	2.2	2.2	2.5	5.2
MR Lat. Cyclic θ_{1c}	3.3	0.3	0.3	0.4
MR Long. Cyclic θ_{1s}	0.8	4.2	0.8	2.5

Overall, we see that the differences between both models are rather small, e.g. below 6 % for the Main Rotor (MR) collective θ_0, below 5.5 % for the Tail Rotor (TR) collective θ_{TR}, below 3.5 % for the MR lateral cyclic θ_{1c}, and below 4.5 % for the MR longitudinal cyclic θ_{1s}. From Fig. 2.3, Fig. 2.5, and Fig. 2.7, we also see that the maximum relative trim deviation does not exceed 10 % for the main rotor power P_{MR}, for the longitudinal, lateral, and climb motions. However, we do notice, as can also be seen in Table 2.2, some higher discrepancies between both models in descending flight (particularly inside the VRS), where

for instance the maximum relative trim deviation reaches 26 % for the main rotor power P_{MR}. This could probably indicate that both models are implementing distinct simulations of the induced rotor flow inside the VRS. The plot of the MR collective input θ_0, on Fig. 2.4, reveals also the minimum power speed, sometimes called the *bucket speed*, predicted to be around 11–13 m/s by both models. From the MR power plot P_{MR}, in Fig. 2.5, we can also see that, as expected, for a CW main rotor for which the tail rotor thrust is oriented towards port-side (i.e. to the left), it takes more power for vehicle starboard flight (i.e. to the right) than for port-side flight. Finally, for our helicopter, the VRS region at $(V_N, V_E) = (0, 0)\, m/s$ is approximately defined by $-6 < V_Z < -3$ m/s (see also our discussion in [1]). Here, we clearly see form Fig. 2.7 and Fig. 2.8 that MR collective θ_0 and MR power P_{MR}, as expected, start to increase inside the VRS, e.g. compare their values at $V_Z = -4\, m/s$ vs. at $V_Z = -3\, m/s$. Hence, more engine power is required from a VRS descent than from hover.

2
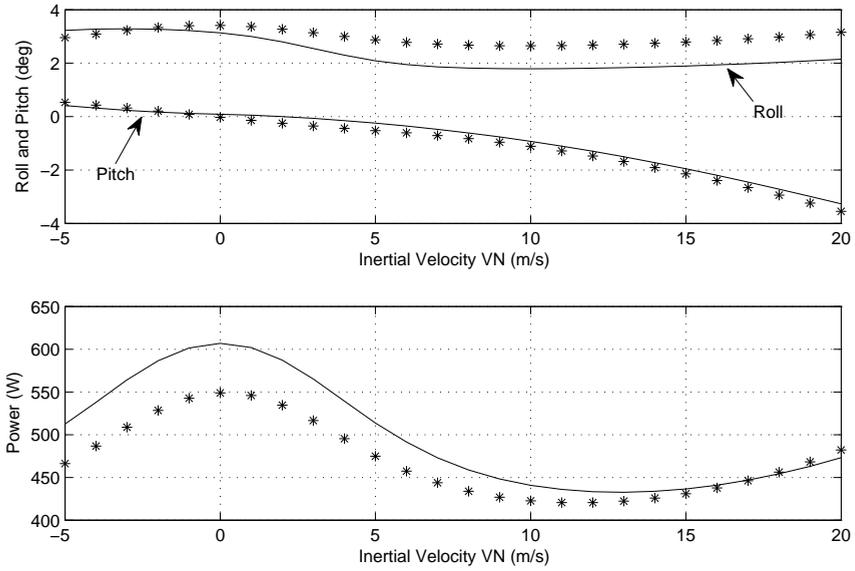
Figure 2.3: Trim along inertial North velocity V_N: roll and pitch angles, and main rotor power (–FLIGHTLAB, $*$ Our Model).
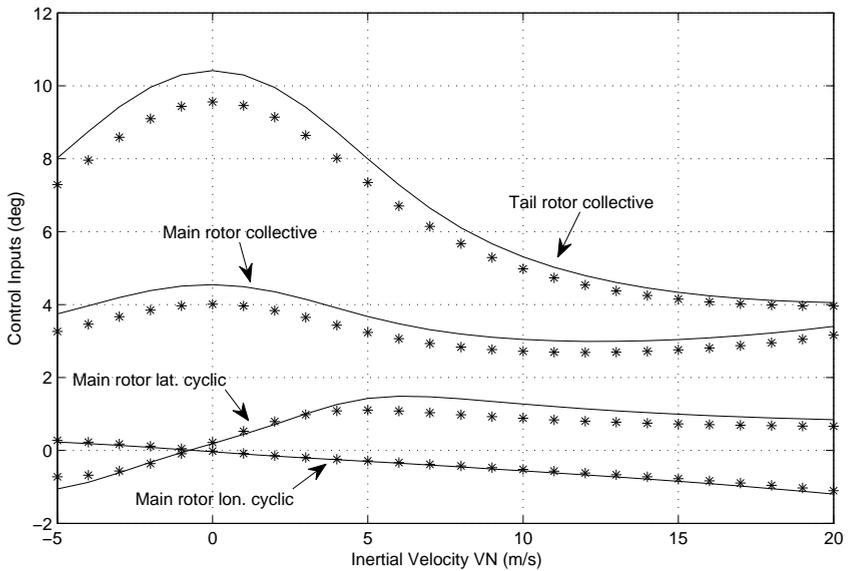
Figure 2.4: Trim along inertial North velocity V_N: control inputs (–FLIGHTLAB, $*$ Our Model).
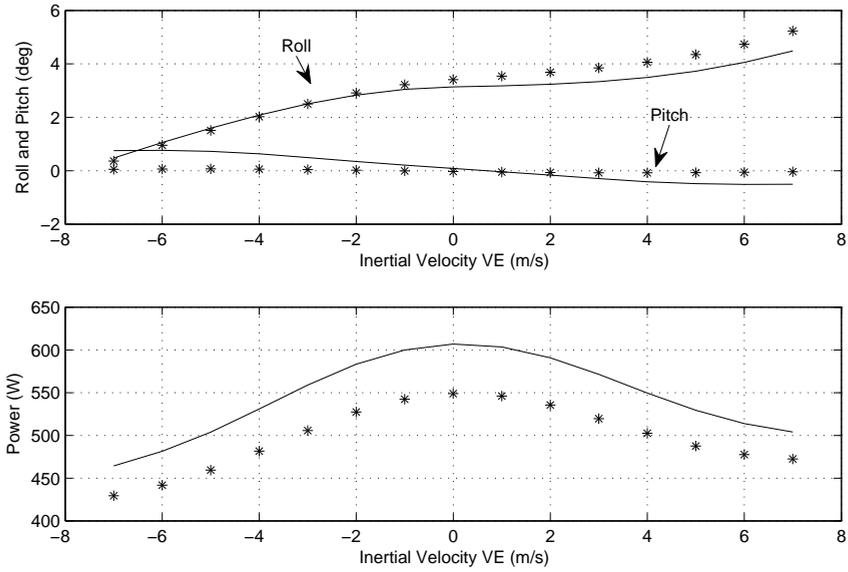
Figure 2.5: Trim along inertial East velocity V_E: roll and pitch angles, and main rotor power (–FLIGHTLAB, $*$ Our Model).
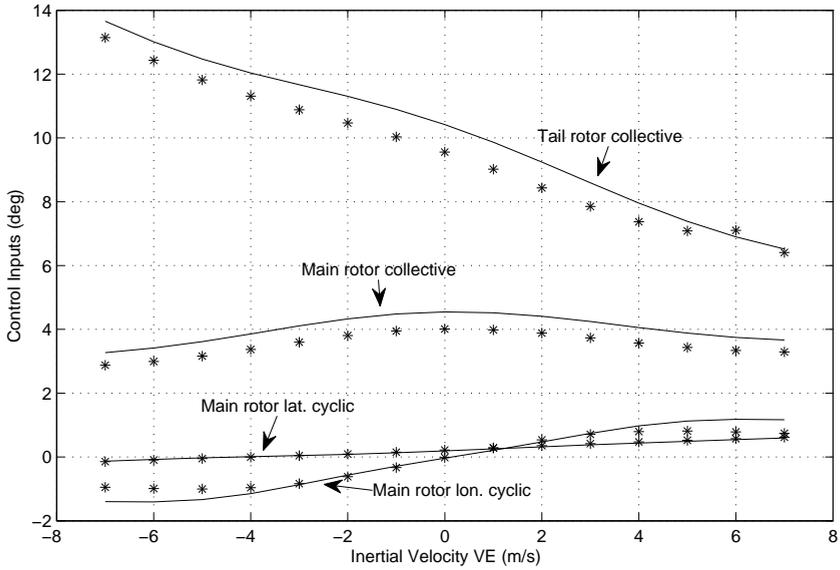
Figure 2.6: Trim along inertial East velocity V_E: control inputs (–FLIGHTLAB, $*$ Our Model).
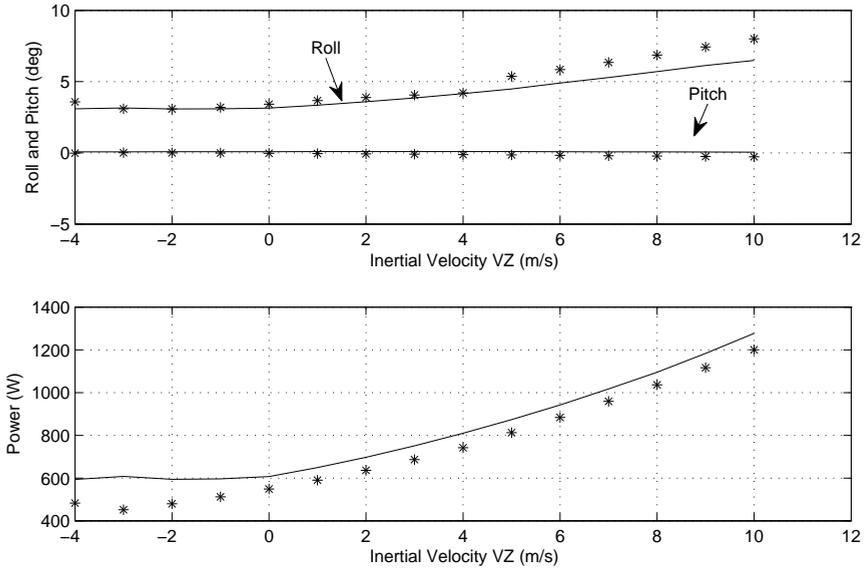
Figure 2.7: Trim along inertial Vertical velocity V_Z (> 0 up): roll and pitch angles, and main rotor power (–FLIGHTLAB, ∗ Our Model).
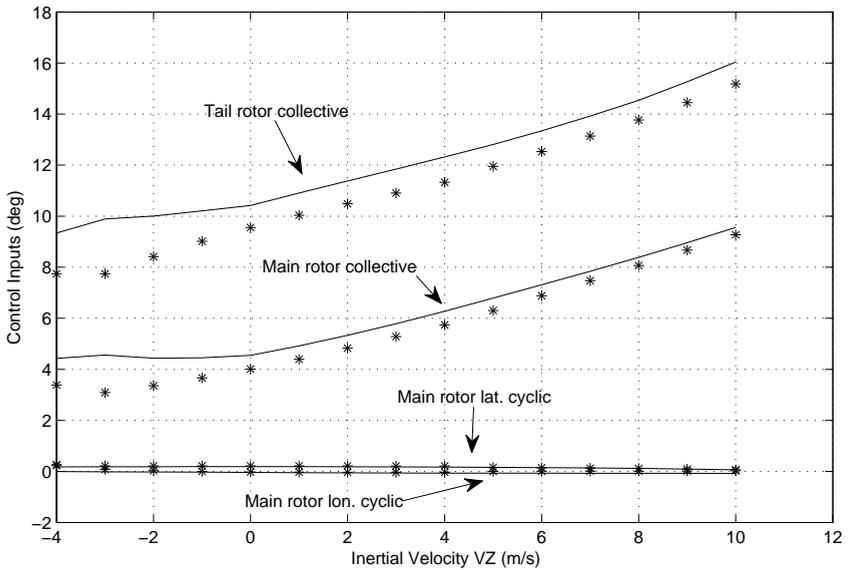
Figure 2.8: Trim along inertial Vertical velocity V_Z (> 0 up): control inputs (–FLIGHTLAB, ∗ Our Model).

2.3.2. Dynamic results

For the dynamic response comparison, we compare the time histories of our model with those of FLIGHTLAB. Basically, the tests are set to evaluate the open-loop response of our helicopter model. Both models have a simulation time-step set equal to $1/24^{\text{th}}$ of a main rotor revolution[15]. First, the rotor is allowed to reach a steady-state condition during a time period of 1 s. (this is a purely software initialization matter, since the simulation starts with all states at zero). Then, for the following 3 s. we simultaneously apply sine-sweeps from 0 to 2 Hz on the four input channels[16], see Fig. 2.9. Next, we evaluate the responses of the following ten states: attitude angles (ϕ, θ, ψ), body linear velocities (u, v, w), body rotational velocities (p, q, r), and MR RPM Ω_{MR} (the RPM is included for the autorotation case only). For a quantitative evaluation we use the Variance-Accounted-For (VAF), defined as: $\text{VAF} := 100\%.\max\left(1 - \frac{\text{var}(\mathbf{x}_k - \tilde{\mathbf{x}}_k)}{\text{var}(\mathbf{x}_k)}, 0\right)$ with $\tilde{\mathbf{x}}_k$ one of the ten states in our model, and $\mathbf{x}_k$ its FLIGHTLAB counterpart, see Table 2.4. The VAF is a widely used metric[17] in the realm of system identification[18]

Table 2.4: Vehicle dynamic response to sine-sweeps on the four input channels: Variance-Accounted-For (VAF) by our model with respect to FLIGHTLAB.

Name	VAF (%)		
	hover	$V_N = $ 10 m/s	steady-state autorotation $(V_N, V_Z) = (6, -6)$ m/s
Roll ϕ	51	76	86
Pitch θ	73	84	59
Yaw ψ	61	50	96
Long. velocity u	79	84	84
Lat. velocity v	62	91	96
Vertical velocity w	93	28	92
Roll rate p	67	45	76
Pitch rate q	43	68	77
Yaw rate r	95	70	97
MR RPM Ω_{MR}	N.A.	N.A.	82
Average over all states	69	66	85

---

[15]The default value in FLIGHTLAB.

[16]The relatively short experiment time of 3 s. is explained by the short time-to-double amplitude, found to be in the range of 0.9–2.3 s., this latter being derived from the eigenvalues of local LTI models. Since the total experiment time is rather short, we chose to focus the model validation on its low-frequency behavior, hence the 2 Hz limit on the applied input signal.

[17]VAF values above 75 % suggest a high-quality model, whereas values in the range 50–75 % would indicate an average–to–good model quality.

[18]Note that, usually, the VAF is used in a parameter-estimation context where one tries to 'match' the outputs of a model with the data gathered from various experiments, or alternatively when one tries to 'match' the outputs of a lower-order model with those from a more complex, often higher-order, model. In our case, we simply use the VAF to compare two models, without any 'tuning' or 'fitting' of coefficients. Hence, in our case, the obtained VAF values tend to be lower than VAF values typically seen in a system identification context.

Three test cases are presented, all starting at an altitude of 30 m. The first two with the engine ON, and the third with the engine OFF. The first test case is run from the hover trim condition, see Fig. 2.10, where it can be seen that the overall fit with FLIGHTLAB is good to very good (see also Table 2.4). The second test case is run to evaluate the high speed flight condition, at $V_N = 10$ m/s, see Fig. 2.11, where we can see that the overall fit with FLIGHTLAB is again good, except for the low VAF value (of 28 %) reported for w (although the plot on the w channel is rather good, as can be seen in Fig. 2.11). Indeed, if the to-be-compared values are close to zero (as is here the case for w), the VAF metric will tend to artificially amplify any discrepancies.
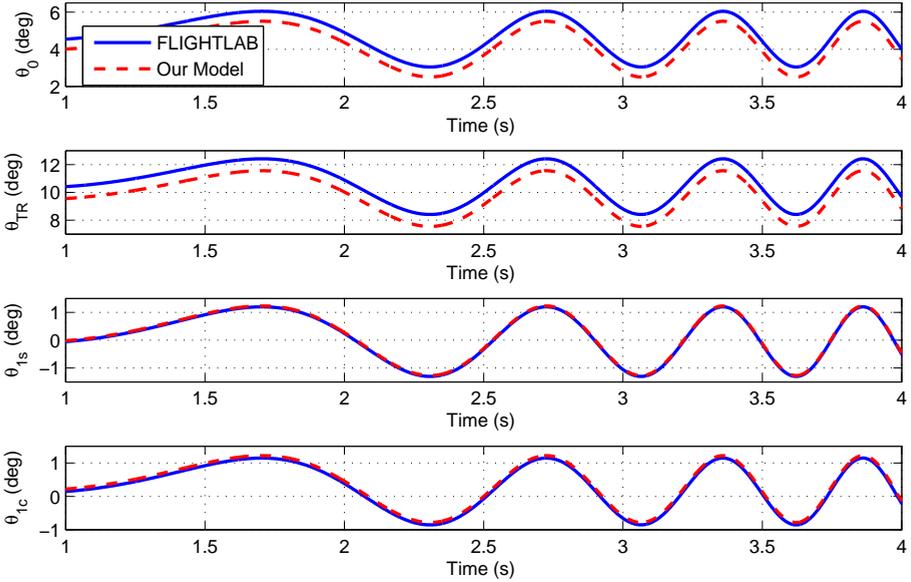
Figure 2.9: Vehicle dynamics: sine-sweep inputs for test cases 1, 2, & 3 (–FLIGHTLAB, – –Our Model)

The third test case is run to check the steady-state autorotative flight condition. In this test case the helicopter is first trimmed at $(V_N, V_Z) = (6, -6)$ m/s and at a MR Ω_{MR} as near as possible to the nominal (i.e. engine ON) value of 1350 RPM, using the engine OFF trim procedure, which also minimizes the MR RPM acceleration $\dot{\Omega}_{MR}$. The results are shown in Fig. 2.12, where we can see that the overall fit with FLIGHTLAB is again good.

Naturally our model does not perfectly match FLIGHTLAB. To some extent the observed discrepancies, between both models, may originate from the fact that both models are built upon distinct modeling philosophies. For instance, for the derivation of the flap-lag dynamics as well as the computation of the rotor forces and moments, our model is based upon a white-box, first-principles approach, i.e. a closed-form representation of the system's behavior. On the contrary, FLIGHTLAB is based upon the so-called multi-body
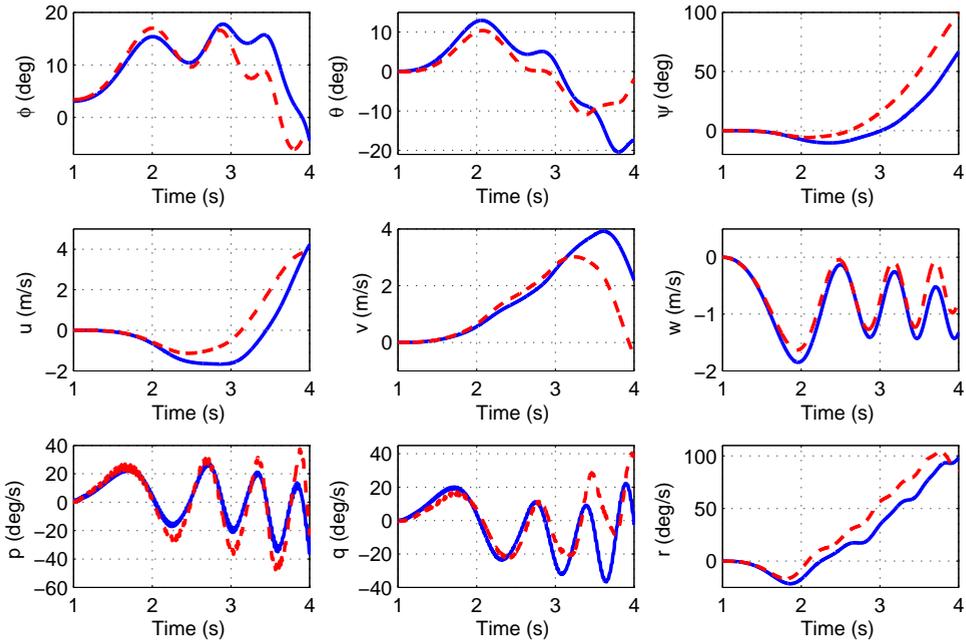
Figure 2.10: Vehicle dynamics (test case 1): response to sine-sweep inputs (the inputs are given in Fig. 2.9), from an initial condition in hover. The visualized states are: roll angle ϕ, pitch angle θ, yaw angle ψ, body longitudinal velocity u, body lateral velocity v, body vertical velocity w, body roll velocity p, body pitch velocity q, and body yaw velocity r (–FLIGHTLAB, – –Our Model).

concept[19]. For instance for the case of a FLIGHTLAB main rotor blade, this latter is split into N smaller bodies. Each body is undergoing a translational and rotational displacement, with the dynamic behavior of the complete system (here the complete blade, or multi-body system) resulting from the equilibrium of applied forces and the rate of change of momentum at each body. This difference in modeling philosophies will inevitably result in slight differences in, for instance, the magnitude of rotor forces and moments. Further, it is well known that even small variations in the computation of forces and moments will be integrated, over time, to large errors in velocities and positions[20]. Besides, this effect gets exacerbated for highly unstable systems[21], which is generally the case of highly agile small-scale helicopters (on the one hand due to their very low inertia, and on the other due

---

[19]The multi-body concept may often be used to simulate the dynamic behavior of interconnected rigid and flexible bodies.

[20]We note that the fit for test case 3 (autorotation) is better than the fit obtained for the first two test cases (with engine ON). The explanation being as follows: in autorotation, main and tail rotor collective have much lower values when compared to their engine ON values, and hence the generated aerodynamic forces are as well smaller in magnitude. Smaller aerodynamic forces also imply smaller discrepancies, in magnitude, between the forces computed by both models, resulting in smaller errors in velocities and positions when integrated over time.

[21]This is also why system identification of unstable systems is most often done in closed-loop [25].
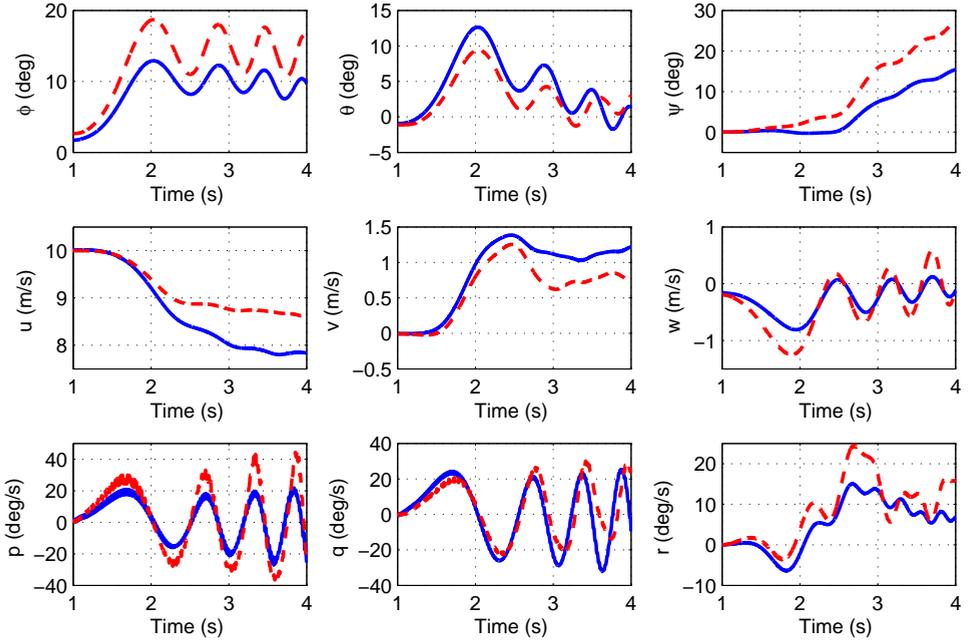
Figure 2.11: Vehicle dynamics (test case 2): response to sine-sweep inputs (the inputs are given in Fig. 2.9), from an initial condition $V_N = 10$ m/s (V_N is the vehicle inertial linear velocity in the direction of True North). The visualized states are: roll angle ϕ, pitch angle θ, yaw angle ψ, body longitudinal velocity u, body lateral velocity v, body vertical velocity w, body roll velocity p, body pitch velocity q, and body yaw velocity r (–FLIGHTLAB, – –Our Model).

to the high rotor stiffness resulting in high rotor moments). To conclude, as can be seen from the last row in Table 2.4, the model's average VAF (over all states) is relatively high, i.e. in the range 66–85 %, and hence the realism of our model is considered to be of good quality.

2.4. PRELIMINARY ANALYSIS OF THE RIGID-BODY DYNAMICS

The objective here is to obtain additional insight into the helicopter rigid-body dynamics, in open-loop, at two trimmed (equilibrium) flight conditions, one for the engine ON case, and one for the OFF case. At these two trimmed flight conditions, we first derive two respective LTI plants by linearizing the NL helicopter model. These LTI plants describe the small perturbation motion about these trimmed conditions, and will later on (in Chapter 4) be used for controller design. Since our focus is primarily on the low-frequency model responses, i.e. the rigid-body motion, we define each plant as follows: the state-vector is of dimension nine given by $\mathbf{x} = (u\ v\ w\ p\ q\ r\ \phi\ \theta\ \psi)^\top$, the control input[22] is of dimension four given by $\mathbf{u} = (\theta_0\ \theta_{1c}\ \theta_{1s}\ \theta_{TR})^\top$, the wind disturbance (given in inertial frame) is of

---

[22]The nomenclature, given in Appendix A, states that all vectors are printed in boldface, hence the control input vector $\mathbf{u}$ should not be confused with the body longitudinal velocity u.
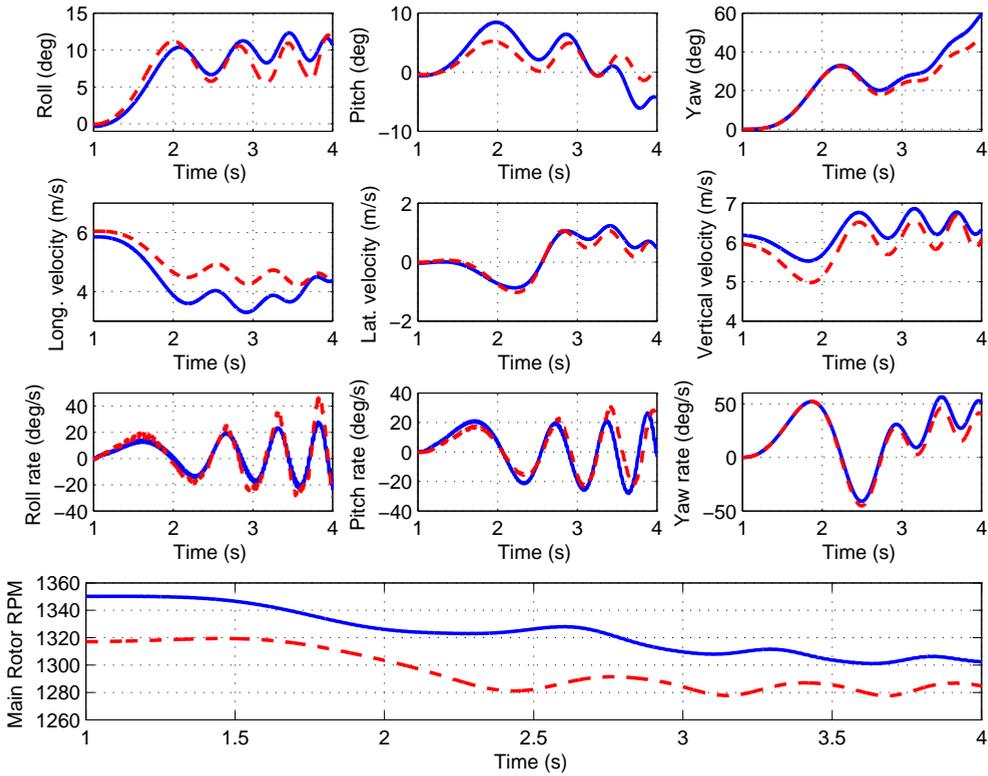
Figure 2.12: Vehicle dynamics (test case 3): response to sine-sweep inputs (the inputs are given in Fig. 2.9), from an initial condition corresponding to a steady-state autorotation at $(V_N, V_Z) = (6, -6)$ m/s (V_N is the vehicle inertial linear velocity in the direction of True North, and V_Z is the inertial vertical velocity). The visualized states are: roll angle ϕ, pitch angle θ, yaw angle ψ, body longitudinal velocity u, body lateral velocity v, body vertical velocity w, body roll velocity p, body pitch velocity q, and body yaw velocity r (–FLIGHTLAB, – –Our Model).

dimension three given by $\mathbf{d} = (V_{N_w} \; V_{E_w} \; V_{Z_w})^\top$, and finally the measurements vector is given by $\mathbf{y} = \mathbf{x}$. The state-space data of these LTI models is further reported in Appendix H. Next, for these two LTI plants, we will analyze their pole maps in the complex plane, but first we address the NL plant linearization issue.

2.4.1. Linearizing the nonlinear helicopter model

The NL helicopter model is subject to periodic loads, due to blades rotation, that result in a time-varying trim condition. Linearizing the NL helicopter dynamics, around a trim condition, can be done at each rotor position, to yield a Periodic Linear Time-Varying (PLTV) system, with a period equal to one rotation of the rotor. Now, for PLTV systems, the classical modal analysis methodologies, based upon time-invariant eigenstructures, are not applicable anymore [26]. Hence, if one desires to apply the well-established analysis and control tools for LTI systems, then a transformation of the PLTV system into a LTI one

becomes necessary. There are roughly four main methods to perform such a transformation or approximation [27]. The first, and simplest one, consists in evaluating the PLTV system at a single rotor position (i.e. at a single blade azimuth position), and thus obtain a LTI system. Clearly, this approach may lead to poor results. An already better method would consist in averaging the PLTV state-space matrices over one or more rotor periods. The next two methods provide LTI models with higher accuracy, but require additional mathematical steps. The third method uses Floquet theory [26, 28], and the associated characteristic exponents called Floquet multipliers, to obtain constant state-space matrices. The fourth method uses the so-called Multi-Blade Coordinate (MBC) transformation (also known as the Coleman transformation) [26, 29–31], i.e. by transforming quantities from rotating blade coordinates into a non-rotating frame. Basically, the MBC describes the overall motion of a rotating blade array in the inertial frame of reference. The MBC transformation results in a weakly periodic system, which is subsequently converted into a LTI system, by averaging over one period [31]. Now, for our application, the first and fourth methods were deemed inappropriate. For the first, it is well-known that this method may not provide an LTI model of high accuracy. The fourth is particularly well-suited for rotors having three or more blades, and may involve significant inaccuracies for a two-bladed rotor[23] [32]. The third is potentially more interesting, since providing LTI models with good accuracy. However, in this thesis, we opted for the second method, since much simpler to use and implement. Hence, the linearized models are computed using a classical numerical perturbation method, resulting in a first-order Taylor series approximation of the NL model, with an averaging over several rotor periods.

AVERAGING: CHOICE OF THE NUMBER OF ROTOR PERIODS

We compare here the dynamic response, i.e. rigid-body time histories, from the NL helicopter plant with the dynamic response from five LTI models, i.e. the latter obtained by averaging from one to five rotor periods. Again, the rotor is first allowed to reach a steady-state condition during a time period of 1 s. Then, for the following 3 s. we simultaneously apply, on the four input channels, the same sine-sweep inputs that were used during the model validation, see Fig. 2.9. We further only analyze here the case for an engine ON in hover (similar results have been observed for other flight conditions), see Fig. 2.13.

For a quantitative evaluation we again use the VAF, with Table 2.5 reporting the VAF values, accounted by each LTI model with respect to the NL model, corresponding to the 3 s. long experiment depicted in Fig. 2.13. Interestingly, we see that the LTI model obtained by averaging after only one period is rather poor, particularly on the pitch θ, pitch rate q, and longitudinal velocity u axes, where these LTI outputs are moving in opposite directions with respect to the NL ones. Increasing the number of averaged periods was thus deemed necessary. Obviously, a high number of averaged periods will tend to filter out the helicopter higher-order dynamics, resulting in a lower-quality LTI model. Hence, some trade-off may need to be considered here. From the last row in Table 2.5, giving the LTI model's average VAF (over all states), we see that averaging over three or four periods may provide the best compromise. Now, since an LTI model should describe the small perturbation motion about a trimmed condition, we also evaluated the VAF values for a shorter experiment time

---

[23]As a reminder, our Remote-Controlled (RC) Align T-REX helicopter has a two-bladed main rotor.

Table 2.5: Effect of averaging when linearizing the NL plant in order to obtain LTI models. Vehicle dynamic response to sine-sweeps on the four input channels: Variance-Accounted-For (VAF) by each LTI model with respect to the NL model, for a 3 seconds long flight time.

Name	VAF (%) when averaging over				
	1 rotor period	2 rotor period	3 rotor period	4 rotor period	5 rotor period
Roll ϕ	0	63	86	92	90
Pitch θ	0	80	75	63	63
Yaw ψ	0	37	60	64	56
Long. velocity u	11	74	80	85	87
Lat. velocity v	63	78	74	72	78
Vertical velocity w	58	70	78	83	83
Roll rate p	0	67	49	12	12
Pitch rate q	0	74	74	70	73
Yaw rate r	62	89	93	95	92
Average over all states	22	70	74	71	70

Table 2.6: Effect of averaging when linearizing the NL plant in order to obtain LTI models. Vehicle dynamic response to sine-sweeps on the four input channels: Variance-Accounted-For (VAF) by each LTI model with respect to the NL model, for a 1.5 seconds long flight time.

Name	VAF (%) when averaging over				
	1 rotor period	2 rotor periods	3 rotor periods	4 rotor periods	5 rotor periods
Roll ϕ	0	78	96	100	99
Pitch θ	0	26	47	59	38
Yaw ψ	0	0	0	0	0
Long. velocity u	0	49	73	86	80
Lat. velocity v	0	57	82	89	85
Vertical velocity w	70	90	98	100	99
Roll rate p	0	87	71	46	52
Pitch rate q	0	92	99	96	99
Yaw rate r	0	0	0	6	0
Average over all states	8	53	63	65	61
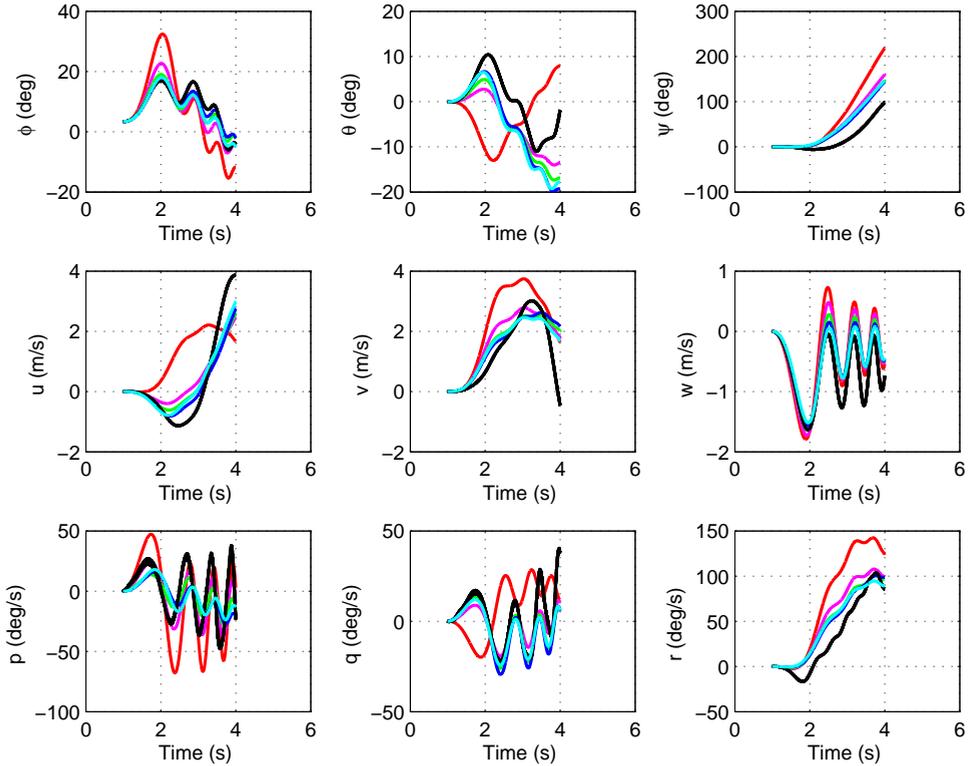
Figure 2.13: Effect of averaging when linearizing the NL plant in order to obtain LTI models. The figure compares the vehicle rigid-body outputs for the NL model, with those from five linearized models. The responses correspond to sine-sweep inputs from hover (black line is the NL model, red line is the LTI model by averaging over one rotor period, magenta line is the LTI model by averaging over two rotor periods, green line is the LTI model by averaging over three rotor periods, blue line is the LTI model by averaging over four rotor periods, and cyan line is the LTI model by averaging over five rotor periods).

(as to better fit the helicopter linear behavior), by considering only the first 1.5 s. of the experiment depicted in Fig. 2.13. This resulted in the VAF values given in Table 2.6. Based on the last row of Table 2.6, we finally settled on using four rotor periods, for the averaging, when computing LTI models from the NL helicopter model.

2.4.2. THE ENGINE ON CASE

The hover trim was here selected as it is known to provide a good representation of helicopter behavior for hover and low-speed flight. Specifically, we consider a trimmed hover, outside ground effect (at an altitude of 30 m), with a fixed and nominal main rotor RPM value of 1350. The eigenvalues of the $\mathbb{A}$ matrix are plotted in Fig. 2.14, for both the engine ON and OFF cases (the engine OFF case will be discussed in Section 2.4.3). For each eigenvalue we also give, in Fig. 2.14, the associated dominant eigenvectors. For the engine ON case, we note the following:

- An inherent difficulty for control design will come from two, lightly-damped, complex pair of poles; one stable pair with a damping of $\zeta = 0.53$, at a natural frequency of $\omega_n = 1.07$ rad/s, and one unstable pair with a damping[24] of $\zeta = -0.42$, at a natural frequency of $\omega_n = 1.05$ rad/s. Their respective eigenvectors associate these modes with a combined longitudinal-lateral-yaw motion, on the u, v, r, and ψ channels.

- There is a pole at the origin (not visible in Fig. 2.14), associated with the heading ψ.

- The time-to-double amplitude[25] is rather fast, equal to 1.54 seconds.

- To stabilize the plant, the bandwidth[26] of the input complementary sensitivity function $T_i(s)$, defined in Section 4.4.2 of Chapter 4, needs to be at least twice the modulus of the unstable pole [34], hence in our case at least 2.1 rad/s.
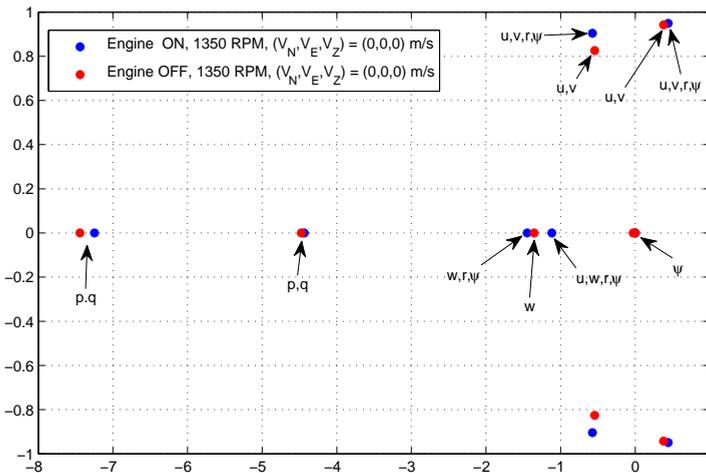
Figure 2.14: Eigenvalues and associated dominant eigenvectors, of the state (or system) matrix, of the LTI models used for control design in Chapter 4, for the engine ON and OFF cases.

---

[24]We use here the MATLAB convention, consisting in using negative damping values when characterizing a complex pair of unstable poles.

[25]The time-to-double amplitude is equal to $0.693/|(\omega_n\zeta)|$ [33].

[26]For MIMO systems this is done by checking the plot of the maximum singular value of the input complementary sensitivity function [34].

2.4.3. THE ENGINE OFF CASE

In the engine OFF case, i.e. autorotative landing, the main rotor RPM is not fixed anymore, and hence main rotor RPM dynamics will impact the overall vehicle flight dynamics. However, we choose here not to include the main rotor RPM Ω_{MR} to the state-vector, and hence keep the same state-vector that was used for the engine ON case. The advantage is that it becomes much easier to find equilibrium points of the NL system[27]. Indeed, by using this "quasi-steady" modeling approach, it becomes possible to find equilibrium points outside of steady autorotation, e.g. while transitioning between the instant of engine failure into steady autorotation, or alternatively during flare (the maneuver just prior to landing). Obtaining these equilibrium points allows for subsequent linearizations of the NL model, and consequently for control design in the LTI framework.

For the trimmed flight condition, we opt for a condition in hover with engine OFF (note that now the main rotor RPM is not in equilibrium anymore). Choosing such a flight condition, with an associated initial velocity of zero, could potentially provide the best description of helicopter behavior during landing (where the helicopter velocity is also very low). The state-space data of the LTI model is further reported in Appendix H. Again, the eigenvalues of the $\mathbb{A}$ matrix are plotted in Fig. 2.14, where for each eigenvalue we also give the associated dominant eigenvectors. For the engine OFF case, we note the following:

- An inherent difficulty for control design will come from two, lightly-damped, complex pair of poles; one stable pair with a damping of $\zeta = 0.54$, at a natural frequency of $\omega_n = 0.99$ rad/s, and one unstable pair with a damping of $\zeta = -0.37$, at a natural frequency of $\omega_n = 1.02$ rad/s. Their respective eigenvectors associate these modes with a combined longitudinal-lateral motion, on the u and v channels.

- The time-to-double amplitude is also fast, equal to 1.83 seconds.

- To stabilize the plant, the bandwidth of the input complementary sensitivity function $T_i(s)$, defined in Section 4.4.2 of Chapter 4, needs to be at least 2.04 rad/s.

2.5. CONCLUSION

This Chapter has presented the first building-block, towards the development of an autonomous helicopter system, that may be characterized as follows: a comprehensive modeling framework, particularly suited for small-scale flybarless helicopters. Comparisons with an equivalent FLIGHTLAB simulation showed that our model is valid for a range of flight conditions, and preliminary insight into the open-loop dynamics was also given. This comprehensive helicopter nonlinear model will, in subsequent Chapters, be used for controller validation. For controller design however, and due to its complexity, only approximation of this model will be used in the upcoming Chapters.

---

[27]This engine OFF trimming approach has shown to be feasible only for low-speed flight conditions.

2.6. APPENDIX A: NOMENCLATURE

Vectors are printed in boldface $\mathbf{X}$. A vector is qualified by its subscript, whereas its super-script denotes the projection frame: e.g. $\mathbf{V}_a^I$ represents the aerodynamic velocity projected on frame F_I. Matrices are written in outline type $\mathbb{M}$, and transformation matrices are denoted as $\mathbb{T}_{ij}$, with the two suffices signifying from frame F_j to frame F_i. All units are in the S.I. system.

Positions and Angles

x_N, x_E, x_Z	Coordinates of vehicle CG in frame F_o
ϕ	Vehicle bank angle (roll angle)
θ	Vehicle inclination angle (pitch angle, or elevation)
ψ	Vehicle azimuth angle (yaw angle, heading)
ψ_f	Wind heading angle

Linear velocities $\mathbf{V}$ and their components u, v, w

$\mathbf{V}_{k,G}$	Kinematic velocity of vehicle CG
$\mathbf{V}_{a,G}$	Aerodynamic velocity of vehicle CG
$u_k^o = V_N$	x component of $\mathbf{V}_{k,G}$ on F_o, North velocity
$v_k^o = V_E$	y component of $\mathbf{V}_{k,G}$ on F_o, East velocity
$w_k^o = V_Z$	z component of $\mathbf{V}_{k,G}$ on F_o, Vertical velocity
$u_k^b = u$	x component of $\mathbf{V}_{k,G}$ on F_b
$v_k^b = v$	y component of $\mathbf{V}_{k,G}$ on F_b
$w_k^b = w$	z component of $\mathbf{V}_{k,G}$ on F_b
u_w	Wind x-velocity in F_E
v_w	Wind y-velocity in F_E
w_w	Wind z-velocity in F_E

Angular velocities $\boldsymbol{\Omega}$ and their components p, q, r

$\boldsymbol{\Omega}_k = \boldsymbol{\Omega}_{bE}$	Kinematic angular velocity of vehicle CG relative to the earth
$p_k^b = p$	Roll velocity (roll rate) of vehicle CG wrt to the earth
$q_k^b = q$	Pitch velocity (pitch rate) of vehicle CG wrt to the earth
$r_k^b = r$	Yaw velocity (yaw rate) of vehicle CG wrt to the earth

Main Rotor (MR) properties

α	wake angle wrt to rotor disk
α_{bl}	Blade section angle of attack
B	Tip loss factor
β_{bl}	Blade flap angle
β_0	Rotor TPP coning angle
β_{1c}	Longitudinal rotor TPP tilt
β_{1s}	Lateral rotor TPP tilt
β_P	Rotor precone angle
C_0	$= M_{bl}.y_{G_{bl}}$ Blade 1st mass moment
c_{bl}	Blade chord

Main Rotor (MR) properties (cont'd)

$c_{d_{bl}}$	Blade section drag coefficient
$c_{l_{bl}}$	Blade section lift coefficient
c_M	Blade section pitching moment due to airfoil camber
e_F	Distance between lag and flap hinge
e_L	Distance between pitch and lag hinge
e_P	Distance between Hub and pitch hinge
$\Delta_e = e_P + e_L + e_F$	Distance between Hub and flap hinge
η_β	$= 0.5R_{bl}^2/(1 - (e_P + e_L + e_F))$
η_ζ	$= 0.5R_{bl}^2/(1 - (e_P + e_L))$
Γ	MR rotation, $CCW : \Gamma = 1.\ \ CW : \Gamma = -1$
G_{eff}	Ground effect corrective factor
I_b	Blade 2nd mass moment (inertia about rotor shaft)
I_β	Blade 2nd mass moment (inertia about flap hinge)
i_S	Shaft tilt-angle
K_{D_β}	Hub spring damper coef. (due to flap)
K_{D_ζ}	Hub spring damper coef. (due to lag)
K_{S_β}	Hub spring restraints coef. (due to flap)
K_{S_ζ}	Hub spring restraints coef. (due to lag)
$\lambda_0, \lambda_c, \lambda_s$	Uniform, longitudinal, lateral inflows
M_{bl}	Blade mass from flap hinge
N_b	Number of blades
Ω_{MR}	Instantaneous angular velocity
$\Omega_{MR_{100\%}}$	Nominal (100%) angular velocity
ψ_{bl}	Azimuthal angular position of blade
R_{bl}	Blade radius measured from flap hinge
R_{rot}	Rotor radius measured from hub center
r_c	Blade root cutout
r_{dm}	Distance from flap hinge to element dm
θ_{bl}	Blade pitch outboard of flap hinge
θ_{wash}	Blade twist (or washout) at blade tip
x_H, y_H, z_H	Coordinates of MR Hub wrt vehicle CG in F_b
V_M	Mass flow parameter
V_{ref}	$= \Omega_{MR}.R_{rot}$ Reference velocity
V_T	Non-dimensional total velocity at rotor center
v_i	Rotor uniform induced velocity
v_{i0}, v_{ic}, v_{is}	Uniform, longitudinal, lateral induced velocities
$y_{G_{bl}}$	Blade CG radial position from flap hinge
ζ_{bl}	Blade lag angle

Tail Rotor (TR) properties

B_{TR}	Tip loss factor, expressed as percentage of blade length
$\beta_{0_{TR}}$	Tail rotor coning angle
b_{t_1}	Tail blockage constant
CD_{TR}	Mean drag coefficient (profile drag)
$c_{l_{(0,TR)}}$	Blade section lift curve slope
c_{TR}	Blade chord
$\delta_{3_{TR}}$	Hinge skew angle for pitch-flap coupling
λ_{dw}	Downwash
λ_{TR}	Total inflow
$\mu_{TRx}, \mu_{TRy}, \mu_{TRz}$	x-, y-, and z-component of advance ratio
$N_{b_{TR}}$	Tail rotor number of blades
Ω_{TR}	Instantaneous angular velocity
$R_{rot_{TR}}$	Rotor radius measured from shaft
$\sigma_{TR} = N_{b_{TR}} \frac{c_{TR}}{\pi R_{rot_{TR}}}$	Solidity
$\theta_{bias_{TR}}$	Preset collective pitch bias
x_{TR}, y_{TR}, z_{TR}	Coordinates of TR Hub wrt vehicle CG in F_b
v_{bl}	Transition velocity (vertical fin blockage)

Fuselage (Fus) properties

α_{Fus}	Angle of attack
β_{Fus}	Sideslip angle
$L_{ref_{Fus}}$	Reference length
$S_{ref_{Fus}}$	Reference area
$x_{Fus}, y_{Fus}, z_{Fus}$	Coordinates of Fus aero center wrt vehicle CG in F_b

Control Inputs

θ_0	MR blade root collective pitch
θ_{1c}	MR lateral cyclic pitch
θ_{1s}	MR longitudinal cyclic pitch
θ_{TR}	TR blade collective pitch angle

Miscellaneous

g	Acceleration due to gravity
m_V	Vehicle mass
$\mathbb{I}_V = \begin{bmatrix} A & -F & -E \\ -F & B & -D \\ -E & -D & C \end{bmatrix}$	Vehicle inertia matrix
M	Mach number
ρ	Air density

Blade angle conventions, according to [26]

β_{bl}	Blade flap angle is defined to be positive for upward motion of the blade
ζ_{bl}	Blade lag angle is defined to be positive when opposite the direction of rotation of the rotor
θ_{bl}	Blade pitch angle is defined to be positive for nose-up rotation of the blade

2.7. Appendix B: Frames

The first five frames hereunder, i.e. F_I–F_k, are the standard aircraft navigation frames, see for example [14].

Frame names

F_I	Geocentric inertial frame (see Fig. 2.15)
F_E	Normal earth fixed frame
F_o	Vehicle carried normal earth frame (see Fig. 2.16)
F_b	Body (vehicle) frame (see Fig. 2.16)
F_k	Kinematic (flight path) frame
F_{HB}	Hub-Body frame (see Fig. 2.17 and Fig. 2.18)
$F_{1<i<6}, F_{bl}$	Main Rotor frames (see Fig. 2.17 and Fig. 2.18)

Frame origins

A	Origin of frame F_I, earth center
G	Origin of frames F_b and F_k, vehicle CG
H	Origin of frame F_{HB}
O	Origin of frames F_E and F_o

2

The inertial frame F_I $(A, \mathbf{x}_I, \mathbf{y}_I, \mathbf{z}_I)$

The inertial frame F_I, see Fig. 2.15, is a geocentric inertial axis system. The origin of the frame A being the center of the earth, the axis south-north $\mathbf{z}_I$ is carried by the axis of the earth's rotation, while axes $\mathbf{x}_I$ and $\mathbf{y}_I$ are keeping a fixed direction in space. The angular velocity of the earth relative to F_I is $\mathbf{\Omega}_{EI}$.

Normal earth-fixed frame F_E $(O, \mathbf{x}_E, \mathbf{y}_E, \mathbf{z}_E)$

This frame is attached to the earth. The origin O is a fixed point relative to the earth and the axis $\mathbf{z}_E$ is oriented following the descending direction of gravitational attraction located on O. The plane $(\mathbf{x}_E, \mathbf{y}_E)$ is tangent to the earth's surface. The point O will be placed at the surface of the earth's geoid and the axis $\mathbf{x}_E$ will be directed towards the geographical north.

Vehicle-carried normal earth frame F_o $(O, \mathbf{x}_o, \mathbf{y}_o, \mathbf{z}_o)$

The axis $\mathbf{z}_o$ is oriented towards the descending direction of the local gravity attraction, at the vehicle center of mass (F_o has the same origin O as F_E), but contrary to the latter it follows the local gravity as seen by the vehicle. The axis $\mathbf{x}_o$ will be directed towards the geographical north (thus $\mathbf{x}_o$ is not parallel to $\mathbf{x}_E$).

Body frame F_b $(G, \mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b)$

This frame is linked to the vehicle's body. The fuselage axis $\mathbf{x}_b$ is oriented towards the front and belongs to the symmetrical plane of the vehicle. The axis $\mathbf{z}_b$ is in the symmetrical plane of the vehicle and oriented downwards relative to the vehicle. This definition assumes the existence of a symmetrical plane.

Kinematic or flight-path frame F_k $(G, \mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k)$

The axis $\mathbf{x}_k$ is carried by the kinematic velocity of the vehicle $\mathbf{V}_{k,G}$.
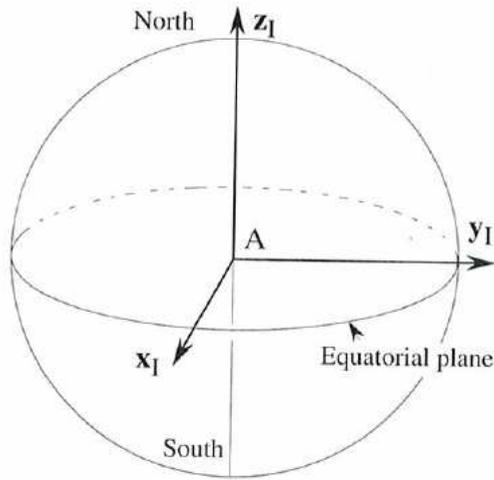
Figure 2.15: Inertial frame F_I. Figure from [14].
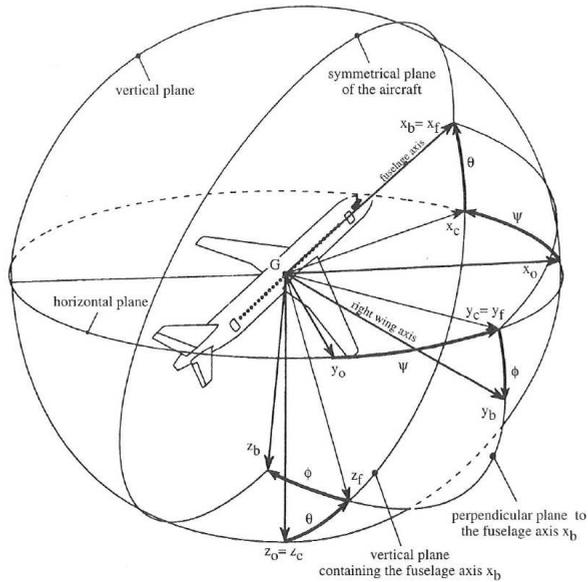
Figure 2.16: Vehicle carried normal earth frame F_o, and body frame F_b. Figure from [14].

2.8. APPENDIX C: RIGID-BODY EQUATIONS OF MOTION

Classical Newtonian mechanics and the fundamental relationship of kinematics give the standard twelve-states rigid-body equations of motion (following notations of [14] and the nomenclature given in Appendix A):

$$\begin{pmatrix} \dot{x}_N \\ \dot{x}_E \\ \dot{x}_Z \end{pmatrix}^o = \begin{pmatrix} V_N \\ V_E \\ V_Z \end{pmatrix}^o \qquad \begin{pmatrix} V_N \\ V_E \\ V_Z \end{pmatrix}^o = \mathbb{T}_{ob}. \begin{pmatrix} u \\ v \\ w \end{pmatrix}^b \tag{2.4}$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}^b = - \begin{pmatrix} q.w - r.v \\ r.u - p.w \\ p.v - q.u \end{pmatrix}^b + g. \begin{pmatrix} -\sin\theta \\ \cos\theta\sin\phi \\ \cos\theta\cos\phi \end{pmatrix}^b + \frac{\mathbf{F}_{CG}{}^b}{m_V} \tag{2.5}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix}^b = \mathbb{I}_V^{-1}. \left(\mathbf{M}_{CG}^b - \begin{pmatrix} p \\ q \\ r \end{pmatrix}^b \times \left(\mathbb{I}_V. \begin{pmatrix} p \\ q \\ r \end{pmatrix}^b \right) \right) \tag{2.6}$$

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

$$\text{with} \quad \mathbb{T}_{ob} = \begin{bmatrix} \cos\theta\cos\psi & \sin\theta\sin\phi\cos\psi - \sin\psi\cos\phi \\ \sin\psi\cos\theta & \sin\theta\sin\phi\sin\psi + \cos\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi \end{bmatrix}$$

$$\begin{matrix} \cos\psi\sin\theta\cos\phi + \sin\phi\sin\psi \\ \sin\theta\cos\phi\sin\psi - \sin\phi\cos\psi \\ \cos\theta\cos\phi \end{matrix} \Bigg] \tag{2.8}$$

with $\mathbf{F}_{CG}^b$ all external forces, excluding gravity, experienced by the vehicle CG in the body frame F_b, and $\mathbf{M}_{CG}^b$ the moments of all forces expressed at the vehicle CG in frame F_b. These total forces and moments include contributions from the Main Rotor (MR), Tail Rotor (TR), Fuselage (Fus), Vertical Tail (VT), and Horizontal Tail (HT), and are given by

$$\mathbf{F}_{CG}^b = \mathbf{F}_{MR}^b + \mathbf{F}_{TR}^b + \mathbf{F}_{Fus}^b + \mathbf{F}_{VT}^b + \mathbf{F}_{HT}^b$$
$$\mathbf{M}_{CG}^b = \mathbf{M}_{MR}^b + \mathbf{M}_{TR}^b + \mathbf{M}_{Fus}^b + \mathbf{M}_{VT}^b + \mathbf{M}_{HT}^b \tag{2.9}$$

The derivation of the rigid-body dynamics, as given in Eq. (2.4)–Eq. (2.8), is based upon the following assumptions

- The vehicle has a longitudinal plane of symmetry, and has constant mass, inertia, and Center of Gravity (CG) position, hence fuel consumption and/or payload pickup/release are neglected. The vehicle is also a rigid system, i.e. it does not contain any flexible structures, hence the time derivative of the inertia matrix is zero.

- The vehicle altitude Above Ground Level (AGL) is very small compared to the earth radius, implying a gravitation independent of height and thus constant.

- The earth is assumed fixed and flat. There is thus no longer a distinction between the directions of gravitational force and the force of gravity, hence the external force becomes the force of gravity[28].

- We neglect the effect of buoyancy (Armichedes force).

2

---

[28]For further details on the geoid earth and gravity see [14, 35].

2.9. Appendix D: Main rotor

For a single main rotor, and briefly summarized, helicopter flight dynamics includes the rigid-body responses (presented in Appendix C) combined with the main rotor higher-frequency modes [36, 37]. For flight mechanics and control development purposes, the three most important aspects of these higher-order rotor modes are: 1) blade flapping, which allows the blade to move in a plane containing the blade and the shaft; 2) blade lead-lag, which allows the blade to move in the plane of rotation; and 3) rotor inflow which is the flow field induced by the main rotor. Now, for the purpose of modeling a generic flybarless small-scale helicopter main rotor (such as the Align T-REX in Fig. 1.16), we have chosen to model it as an articulated Pitch-Lag-Flap (P-L-F) hinge arrangement. This chosen hinge configuration is particularly well suited for the case of small-scale helicopters. It allows to keep the pitch and lag hinge offsets at their current physical values while replacing the rubber O-rings, see Fig. 1.10, by a virtual flap hinge (having stiffness and damping) outboard of the lag hinge. The (P-L-F) hinge arrangement is visualized in Fig. 2.17 and Fig. 2.18.

Assumptions

The presented assumptions are valid for stability and control investigations of helicopters up to an advance ratio limit[29] of about 0.3 [38–40].

Structural simplifications

- Rotor shaft forward and lateral tilt-angles are zero. Rotor precone is also zero. The blade has zero twist, constant chord, zero sweep, constant thickness ratio, and a uniform mass distribution.

- We assume a rigid rotor blade in bending. We neglect higher modes (harmonics), since higher modes are only pronounced at high speed [7, 41]. Further, blade torsion is neglected since small-scale helicopter blades are generally relatively stiff.

- Rotor inertia inboard of the flap hinge is also neglected.

Aerodynamics simplifications

- Uniform inflow is computed through momentum theory[30].

- Vehicle flies at a low altitude, hence neglecting air density and temperature variations. Blade element theory is used to compute rotor lift and drag forces[31]. Radial flow along blade span is ignored. Pitch, lag, and flap angles are assumed to be small.

- Compressibility effects are disregarded, which is a reasonable assumption considering small-scale helicopter flight characteristics. Viscous flow effects are also disregarded, which is a valid assumption for low AOA and un-separated flow [13, 42].

---

[29]The advance ratio is the ratio of forward vehicle speed to a main rotor blade tip speed. The flight envelope of small-scale helicopters is well within this limit.

[30]Which states that the total force acting on a control volume is equal to the rate of change of momentum [26].

[31]Blade element theory calculates the forces on the blade due to its motion through air. It is assumed that a blade section acts as a 2D airfoil producing aerodynamic forces [26].
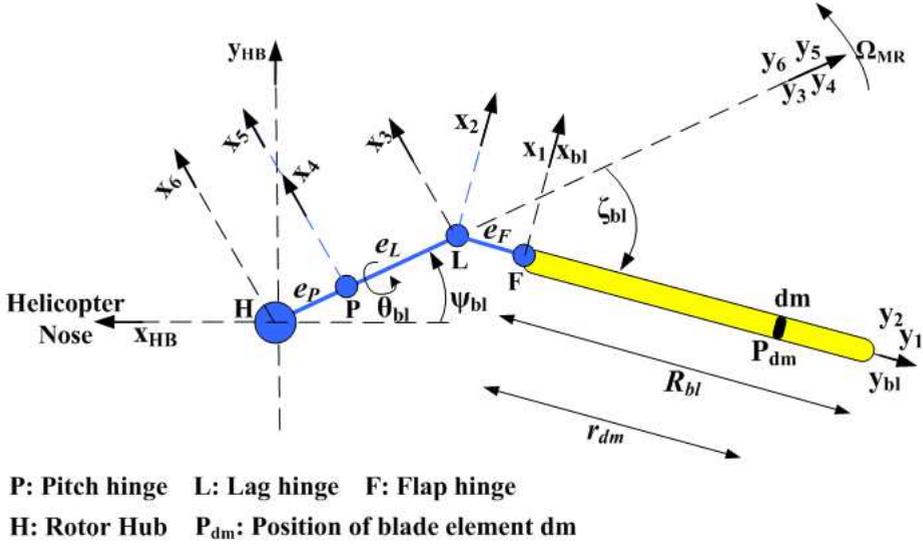
Figure 2.17: Main rotor frames (top-view).
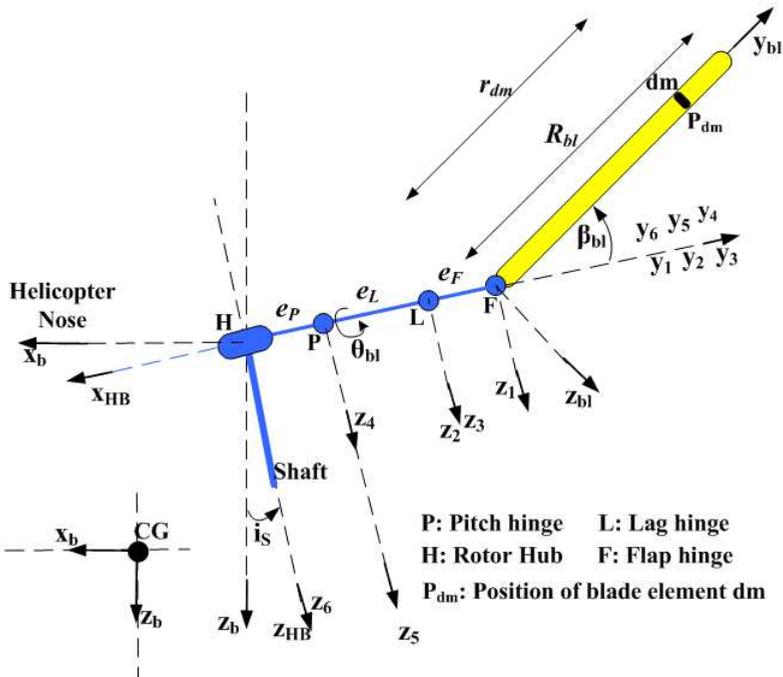
Figure 2.18: Main rotor frames (side-view).

- Aerodynamic interference effects between the main rotor and other helicopter modules, e.g. fuselage or tail rotor, are neglected.

- The presence of the fuselage just under the main rotor acts as a so-called pseudo-ground effect [43], resulting in some thrust recovery. This phenomenon is also neglected.

Dynamical simplifications

- Dynamic twist[32] is neglected. Hence blade CG is assumed to be colocated with blade section quarter chord line.

- Unsteady (frequency dependent) effects for time-dependent development of blade lift and pitching moment, due to changes in local incidence, are ignored; e.g. dynamic stall, due to rapid pitch changes, is ignored.

Comments on the modeling assumptions and model simplifications

Helicopter simulation codes may be developed for a variety of applications, ranging from flight dynamics simulation purposes, flying qualities investigations, auto-pilot design, operational analysis, crew training, load prediction, and/or vibrations analysis. In our case, the desired objectives (i.e. the application domain) for our model are: 1) flight dynamics simulation, in which the model can be used in a Hardware In The loop (HITL) environment to simulate the helicopter dynamics, hence enabling the verification and validation of a flight control system (i.e. the embedded system); and 2) the model should also be useful for controller synthesis, i.e. the so-called modeling for control paradigm. This sets the context of the model presented in this Chapter.

Now once the intended model's application domain has been defined, we need to address the question of helicopter model fidelity. To this end, and according to [44], the level of model sophistication, to conveniently describe a helicopter model complexity, may be formulated by two criteria, namely **model dynamics** and **model validity**, defined as follows:

1. **Model dynamics** qualifies the level of detail in representing the dynamics of the helicopter. This criterium determines the fidelity of the model in terms of the frequency range of applicability, e.g. a model consisting of only the rigid-body, actuators, and main rotor RPM dynamics, versus a model which also includes additional main rotor higher-frequency phenomena, such as blade flap-lag, rotor inflow dynamics, etc.

2. **Model validity** represents the level of sophistication in calculating the helicopter forces, moments, and main rotor inflow. This criterium determines the domain of validity in the flight envelope, e.g. a model which crudely reproduces the associated laws of physics, versus a model which accurately simulates the vehicle (aerodynamic) forces and moments, including at high speed flight, descending in the Vortex-Ring-State (VRS), and the autorotation condition.

---

[32]Any offset in blade chordwise CG and/or blade aerodynamic center position will result in a coupling of the flap and torsion degrees-of-freedom in blade elastic modes [7].

In terms of **model dynamics**, our model includes some of the main rotor higher-order phenomena, such as blade flap-lag dynamics and main rotor inflow dynamics. Hence, for its intended application domain, our model may be considered to be of good quality. This said, and as mentioned here-above in the assumptions, the dynamical aspects related to blade torsion, dynamic twist, and dynamic stall have been neglected. Thus, our model may not be valid in the very high-frequency region, i.e. it probably can not be used for a detailed analysis of vibrations and/or aeroelastic phenomena. However, as mentioned earlier, these latter aspects do not belong to the intended application domain of the proposed model.

In terms of **model validity**, the effects of compressibility and viscous flows have been disregarded, since relatively negligible on small-scale helicopters[33]. On the other hand, our model does include a sophisticated main rotor inflow model, valid also for high-speed descent and VRS flight, but does not include any aerodynamic interference effects between the main rotor and other helicopter components, although this aspect is generally a minor one on small-scale vehicles. In summary we conclude that our model may also have a relatively high model validity for its intended application domain.

POSITION AND VELOCITY OF A BLADE ELEMENT

With reference to the frame's origin A, G, and H, see Appendix B, the inertial position of a blade element dm, located at position P_{dm}, see Fig. 2.17 and Fig. 2.18, is given by

$$\mathbf{AP_{dm}} = \mathbf{AG} + \mathbf{GH} + \mathbf{HP_{dm}} \tag{2.10}$$

Projecting Eq. (2.10) onto the Hub-Body frame F_{HB} we get

$$\mathbf{AP_{dm}}^{HB} = \mathbf{AG}^{HB} + \begin{pmatrix} x_H \\ y_H \\ z_H \end{pmatrix}^{HB} + \begin{pmatrix} x_{dm} \\ y_{dm} \\ z_{dm} \end{pmatrix}^{HB} \tag{2.11}$$

with (x_{dm}, y_{dm}, z_{dm}) the position of blade element dm, with respect to (wrt) the main rotor hub. Now the third term on the Right-Hand-Side (RHS) of Eq. (2.11) is given by (see Fig. 2.17 and Fig. 2.18)

$$
\begin{aligned}
\mathbf{HP_{dm}}^{HB} = \mathbb{T}_{(HB)6}\Bigg\{ \mathbb{T}_{54}\Bigg[\mathbb{T}_{32}\Bigg(\mathbb{T}_{1(bl)} \begin{pmatrix} 0 \\ r_{dm} \\ 0 \end{pmatrix} \\
+ \begin{pmatrix} 0 \\ e_F \\ 0 \end{pmatrix} \Bigg) + \begin{pmatrix} 0 \\ e_L \\ 0 \end{pmatrix} \Bigg] + \begin{pmatrix} 0 \\ e_P \\ 0 \end{pmatrix} \Bigg\}
\end{aligned}
\tag{2.12}
$$

with $\mathbb{T}_{ij}$ rotation matrices[34]. The inertial velocity, i.e. relative to the inertial frame F_I, of a blade element dm, located at position P_{dm}, is defined by $\mathbf{V}_{I,P_{dm}}$. Projecting it onto frame

---

[33]The blade tip Mach number is below 0.4.
[34]For example $\mathbb{T}_{(HB)6}$ represents the rotation from frame F_6 to the Hub-Body frame F_{HB}, $\mathbb{T}_{54}$ represents the rotation from frame F_4 to frame F_5, and $\mathbb{T}_{1(bl)}$ the rotation from the blade frame F_{bl} to frame F_1, etc.

F_{HB}, and using Eq. (2.10), we obtain

$$\mathbf{V}_{I,P_{dm}}^{HB} = \left(\frac{d\mathbf{AG}^I}{dt}\right)^{HB} + \left(\frac{d\mathbf{GH}^I}{dt}\right)^{HB} + \left(\frac{d\mathbf{HP_{dm}}^I}{dt}\right)^{HB} \tag{2.13}$$

where the superscript $(\cdot)^I$, such as in $\frac{d\mathbf{AG}^I}{dt}$, means that the derivative is taken relative to inertial frame F_I. For the first term on the RHS of Eq. (2.13), and assuming a flat and fixed earth, we get (refer also to the nomenclature)

$$\left(\frac{d\mathbf{AG}^I}{dt}\right)^{HB} = \mathbb{T}_{(HB)o}\mathbf{V}_{k,G}^o = \mathbb{T}_{(HB)o}\left(\begin{array}{c} V_N \\ V_E \\ V_Z \end{array}\right)^o \tag{2.14}$$

with $\mathbf{V}_{k,G}^o$ the vehicle kinematic velocity projected onto the vehicle carried normal earth frame F_o, and $\mathbb{T}_{(HB)o}$ the rotation matrix from frame F_o to frame F_{HB}. For the second term on the RHS of Eq. (2.13) we obtain (using the kinematics rule)

$$\left(\frac{d\mathbf{GH}^I}{dt}\right)^{HB} = \left(\frac{d\mathbf{GH}^b}{dt}\right)^{HB} + \mathbf{\Omega}_{bI}^{HB} \times \mathbf{GH}^{HB} \tag{2.15}$$

where $\times$ denotes the cross product, and $\mathbf{\Omega}_{bI}^{HB}$ the angular velocity of body frame F_b relative to inertial frame F_I, projected onto the Hub-Body frame F_{HB}. Here the first term on the RHS of Eq. (2.15) is zero since the hub center H is fixed in the body frame F_b. The second term on the RHS of Eq. (2.15) gives

$$\mathbf{\Omega}_{bI}^{HB} \times \mathbf{GH}^{HB} = \left(\mathbb{T}_{(HB)b}\mathbf{\Omega}_{bI}^b\right) \times \left(\mathbb{T}_{(HB)b}\mathbf{GH}^b\right) \tag{2.16}$$

Since the earth is fixed we have $\mathbf{\Omega}_{bI}^b = \mathbf{\Omega}_{bE}^b$ (see nomenclature), and Eq. (2.15) is now equivalent to

$$\left(\frac{d\mathbf{GH}^I}{dt}\right)^{HB} = \left(\mathbb{T}_{(HB)b}\left(\begin{array}{c} p \\ q \\ r \end{array}\right)\right)^b \times \left(\mathbb{T}_{(HB)b}\left(\begin{array}{c} x_H \\ y_H \\ z_H \end{array}\right)\right)^b \tag{2.17}$$

Finally, for the third term on the RHS of Eq. (2.13) we have

$$\begin{aligned} \left(\frac{d\mathbf{HP_{dm}}^I}{dt}\right)^{HB} &= \left(\frac{d\mathbf{HP_{dm}}^{HB}}{dt}\right)^{HB} + \mathbf{\Omega}_{(HB)I}^{HB} \times \mathbf{HP_{dm}}^{HB} \\ &= \frac{d}{dt}\left(\begin{array}{c} x_{dm} \\ y_{dm} \\ z_{dm} \end{array}\right)^{HB} + \mathbf{\Omega}_{(HB)I}^{HB} \times \left(\begin{array}{c} x_{dm} \\ y_{dm} \\ z_{dm} \end{array}\right)^{HB} \end{aligned} \tag{2.18}$$

We can also express $\mathbf{\Omega}_{(HB)I}^{HB}$ as

$$\mathbf{\Omega}_{(HB)I}^{HB} = \mathbf{\Omega}_{(HB)b}^{HB} + \mathbf{\Omega}_{bI}^{HB} \tag{2.19}$$

The first term on the RHS of Eq. (2.19) is zero since frame F_{HB} is fixed wrt frame F_b. The second term on the RHS of Eq. (2.19) can be re-written as

$$\mathbf{\Omega}_{bI}^{HB} = \mathbb{T}_{(HB)b}\mathbf{\Omega}_{bI}^b = \mathbf{\Omega}_{bI}^b = \mathbf{\Omega}_{bE}^b = \left(\begin{array}{c} p \\ q \\ r \end{array}\right)^b \tag{2.20}$$

where we have used $\mathbb{T}_{(HB)b} = \mathbb{I}$ since rotor shaft longitudinal and lateral tilt-angles i_S are assumed to be zero on our helicopter UAV. Regrouping terms from Eq. (2.14), Eq. (2.17), Eq. (2.18), Eq. (2.19), and Eq. (2.20), we can express the inertial velocity of a blade element dm in F_{HB} as

$$
\mathbf{V}^{HB}_{I,P_{dm}} = \begin{pmatrix} u_{I,P_{dm}} \\ v_{I,P_{dm}} \\ w_{I,P_{dm}} \end{pmatrix}^{HB} = \begin{pmatrix} u \\ v \\ w \end{pmatrix}^{b} + \frac{d}{dt} \begin{pmatrix} x_{dm} \\ y_{dm} \\ z_{dm} \end{pmatrix}^{HB}
$$
$$
+ \begin{pmatrix} p \\ q \\ r \end{pmatrix}^{b} \times \left(\begin{pmatrix} x_H \\ y_H \\ z_H \end{pmatrix}^{b} + \begin{pmatrix} x_{dm} \\ y_{dm} \\ z_{dm} \end{pmatrix}^{HB} \right) \tag{2.21}
$$

where we have used

$$
\mathbb{T}_{(HB)o} \begin{pmatrix} V_N \\ V_E \\ V_Z \end{pmatrix}^{o} = \mathbb{T}_{(HB)b} . \mathbb{T}_{bo} . \begin{pmatrix} V_N \\ V_E \\ V_Z \end{pmatrix}^{o} \tag{2.22}
$$

together with $\mathbb{T}_{(HB)b} = \mathbb{I}$, and $\mathbb{T}_{bo} . \begin{pmatrix} V_N \\ V_E \\ V_Z \end{pmatrix}^{o} = \begin{pmatrix} u \\ v \\ w \end{pmatrix}^{b}$ from the nomenclature. Now plugging Eq. (2.12) into Eq. (2.21), and using any symbolic math toolbox, we can obtain an expanded expression for $\mathbf{V}^{HB}_{I,P_{dm}}$, as follows

$$
u^{HB}_{I,P_{dm}} = u + \Omega_{MR} \Big(\sin\psi_{bl} [e_L + e_P + \cos\zeta_{bl}(e_F + r_{dm}\cos\beta_{bl})]
$$
$$
- \cos\psi_{bl} [\cos\theta_{bl} \sin\zeta_{bl}(e_F + r_{dm}\cos\beta_{bl}) + r_{dm}\sin\beta_{bl}\sin\theta_{bl}] \Big)
$$
$$
+ \dot{\zeta}_{bl}(e_F + r_{dm}\cos\beta_{bl})[\cos\psi_{bl}\sin\zeta_{bl} - \sin\psi_{bl}\cos\theta_{bl}\cos\zeta_{bl}]
$$
$$
+ \dot{\beta}_{bl}r_{dm}[\cos\psi_{bl}\cos\zeta_{bl}\sin\beta_{bl} + \sin\psi_{bl}(\cos\theta_{bl}\sin\zeta_{bl}\sin\beta_{bl} - \cos\beta_{bl}\sin\theta_{bl})]
$$
$$
+ \dot{\theta}_{bl}\sin\psi_{bl}[\sin\theta_{bl}\sin\zeta_{bl}(e_F + r_{dm}\cos\beta_{bl}) - r_{dm}\sin\beta_{bl}\cos\theta_{bl}]
$$
$$
+ q\Big(z_H - r_{dm}\cos\theta_{bl}\sin\beta_{bl} + (e_F + r_{dm}\cos\beta_{bl})\sin\zeta_{bl}\sin\theta_{bl}\Big)
$$
$$
- r\Big(y_H - \Gamma\cos\psi_{bl}(\cos\theta_{bl}\sin\zeta_{bl}(e_F + r_{dm}\cos\beta_{bl})
$$
$$
+ r_{dm}\sin\beta_{bl}\sin\theta_{bl}) + \Gamma\sin\psi_{bl}(e_L + e_P + \cos\zeta_{bl}(e_F + r_{dm}\cos\beta_{bl}))\Big)
$$

$$
\tag{2.23}
$$

$$
\begin{aligned}
v_{I,P_{dm}}^{HB} = {}& v + \Omega_{MR}\Gamma\Big((e_L + e_P)\cos\psi_{bl} + r_{dm}\sin\psi_{bl}\sin\beta_{bl}\sin\theta_{bl} \\
& + (e_F + r_{dm}\cos\beta_{bl})(\cos\psi_{bl}\cos\zeta_{bl} + \sin\psi_{bl}\cos\theta_{bl}\sin\zeta_{bl})\Big) \\
& - \dot\zeta_{bl}\Gamma(e_F + r_{dm}\cos\beta_{bl})[\cos\psi_{bl}\cos\zeta_{bl}\cos\theta_{bl} + \sin\psi_{bl}\sin\zeta_{bl}] \\
& + \dot\beta_{bl}r_{dm}\Gamma(\cos\psi_{bl}\cos\theta_{bl}\sin\zeta_{bl}\sin\beta_{bl} - \cos\psi_{bl}\cos\beta_{bl}\sin\theta_{bl} - \sin\psi_{bl}\cos\zeta_{bl}\sin\beta_{bl}) \\
& + \dot\theta_{bl}\Gamma\cos\psi_{bl}[\sin\theta_{bl}\sin\zeta_{bl}(e_F + r_{dm}\cos\beta_{bl}) - r_{dm}\sin\beta_{bl}\cos\theta_{bl}] \\
& - p\Big(z_H - \big(r_{dm}\cos\theta_{bl}\sin\beta_{bl} - (e_F + r_{dm}\cos\beta_{bl})\sin\zeta_{bl}\sin\theta_{bl}\big)\Big) \\
& + r\Big(x_H - \Big(\cos\psi_{bl}(e_L + e_P + \cos\zeta_{bl}(e_F + r_{dm}\cos\beta_{bl})) \\
& + \sin\psi_{bl}(\cos\theta_{bl}\sin\zeta_{bl}(e_F + r_{dm}\cos\beta_{bl}) + r_{dm}\sin\beta_{bl}\sin\theta_{bl})\Big)\Big)
\end{aligned}
$$
$$(2.24)$$

$$
\begin{aligned}
w_{I,P_{dm}}^{HB} = {}& w + \dot\zeta_{bl}\cos\zeta_{bl}\sin\theta_{bl}(e_F + r_{dm}\cos\beta_{bl}) \\
& - \dot\beta_{bl}r_{dm}(\cos\beta_{bl}\cos\theta_{bl} + \sin\beta_{bl}\sin\zeta_{bl}\sin\theta_{bl}) \\
& + \dot\theta_{bl}[r_{dm}\sin\theta_{bl}\sin\beta_{bl} + (e_F + r_{dm}\cos\beta_{bl})\sin\zeta_{bl}\cos\theta_{bl}] \\
& + p\Big(y_H - \Gamma\cos\psi_{bl}(\cos\theta_{bl}\sin\zeta_{bl}(e_F + r_{dm}\cos\beta_{bl}) + r_{dm}\sin\beta_{bl}\sin\theta_{bl}) \\
& + \Gamma\sin\psi_{bl}(e_L + e_P + \cos\zeta_{bl}(e_F + r_{dm}\cos\beta_{bl}))\Big) \\
& - q\Big(x_H - \cos\psi_{bl}(e_L + e_P + \cos\zeta_{bl}(e_F + r_{dm}\cos\beta_{bl})) \\
& - \sin\psi_{bl}(\cos\theta_{bl}\sin\zeta_{bl}(e_F + r_{dm}\cos\beta_{bl}) + r_{dm}\sin\beta_{bl}\sin\theta_{bl})\Big)
\end{aligned}
$$
$$(2.25)$$

with the total blade pitch angle given by [11]

$$
\theta_{bl} = \theta_0 + \theta_{1c}\cos(\psi_{bl} + \psi_{PA}) + \theta_{1s}\sin(\psi_{bl} + \psi_{PA}) + \theta_{t,r_{dm}} - K_{(\theta_{bl}\beta_{bl})}\beta_{bl} - K_{(\theta_{bl}\zeta_{bl})}\zeta_{bl} \quad (2.26)
$$

and the blade pitch component due to blade twist given by

$$
\theta_{t,r_{dm}} = r_{dm}\frac{\theta_{wash}}{R_{bl}} \quad (2.27)
$$

Note also, as stated in the assumptions here-above, we neglect any effects due to rapid pitch changes, e.g. dynamic stall effects. Hence, we will assume that $\dot\theta_{bl} \ll \dot\beta_{bl}$, $\dot\theta_{bl} \ll \dot\zeta_{bl}$, and $\dot\theta_{bl} \ll \Omega_{MR}$. Consequently, in the sequel we will also assume to have $\dot\theta_{bl} \simeq 0$ in Eq. (2.23)–Eq. (2.25).

Flap-Lag equations of motion

Since the early 1950s it is known that including flapping dynamics in a helicopter flight model could produce limitations in rate and attitude feedback gains [45]. Further, for helicopter directional axis control, blade lead-lag dynamics ought to be considered for control system design [46]. Indeed, it is well known that blade lead-lag produces increased phase lag at high frequency, in the same frequency range where flapping effects occur [47], and that control rate gains are primarily limited by lead-lag-body coupling [47, 48]. Now, in terms of blade flap-lag modeling, a foundational contribution was given in [11], where derivations of the coupled flap-lag equations of motion for a rigid articulated rotor, for the (F-L-P), (F-P-L), and (L-F-P) hinge sequences, was laid out. The purpose of our work is to present a model for a new hinge arrangement, i.e. the (P-L-F) sequence, which is much more useful for modeling the rotor dynamics of a small-scale helicopter. The equations presented in the sequel (obtained by the Lagrangian method [10]) are valid for a single articulated rotor with hinge springs and viscous dampers. Compared to [11] our approach retains all three hinges physically separated and works also for both ClockWise (CW) and Counter-ClockWise (CCW) rotating main rotors. Further, full coupling between vehicle and blade dynamics is modeled. Now from Lagrangian theory, we have

$$\frac{d}{dt}\left(\frac{\partial K_E}{\partial \dot{\zeta}_{bl}}\right) - \frac{\partial K_E}{\partial \zeta_{bl}} = Q_{\zeta_{bl}} \tag{2.28a}$$

$$\frac{d}{dt}\left(\frac{\partial K_E}{\partial \dot{\beta}_{bl}}\right) - \frac{\partial K_E}{\partial \beta_{bl}} = Q_{\beta_{bl}} \tag{2.28b}$$

with K_E the kinetic energy of a blade, β_{bl}, ζ_{bl}, blade flap and lag angles, and $Q_{\beta_{bl}}$, $Q_{\zeta_{bl}}$, the generalized forces. These latter include the effect of gravity, aerodynamics, and spring damping and stiffness, and are given by

$$Q_{\zeta_{bl}} = Q_{\zeta_{bl},G} + Q_{\zeta_{bl},A} + Q_{\zeta_{bl},D} + Q_{\zeta_{bl},S} \tag{2.29a}$$

$$Q_{\beta_{bl}} = Q_{\beta_{bl},G} + Q_{\beta_{bl},A} + Q_{\beta_{bl},D} + Q_{\beta_{bl},S} \tag{2.29b}$$

The kinetic energy of a single rotor blade is given by

$$K_E = \frac{1}{2}\int_0^{R_{bl}} \mathbf{V}_{I,P_{dm}}^{HB}{}^\top . \mathbf{V}_{I,P_{dm}}^{HB} dm \tag{2.30}$$

with $\mathbf{V}_{I,P_{dm}}^{HB}$ computed in Eq. (2.21), and the limits of integration are from the flap hinge, to the blade tip. The kinetic energy inboard of the flap hinge is neglected in our model since assumed small in the case of small-scale UAVs. We provide next the procedure for the blade lead-lag equations Eq. (2.28a), the blade flap equations Eq. (2.28b) follow a similar reasoning and are thus omitted. Now we rewrite the first term on the Left-Hand-Side (LHS) of Eq. (2.28a) as

$$\frac{d}{dt}\left(\frac{\partial K_E}{\partial \dot{\zeta}_{bl}}\right) = \frac{d}{dt}\left(\frac{\partial}{\partial \dot{\zeta}_{bl}}\frac{1}{2}\int_0^{R_{bl}} \mathbf{V}_{I,P_{dm}}^{HB}{}^\top . \mathbf{V}_{I,P_{dm}}^{HB} dm\right) \tag{2.31}$$

And since the limits of integration are constant, Eq. (2.31) is equivalent to (using Leibniz's integral rule)

$$\frac{1}{2}\int_0^{R_{bl}} \frac{d}{dt}\frac{\partial}{\partial \dot{\zeta}_{bl}}\left(\mathbf{V}_{I,P_{dm}}^{HB}{}^\top . \mathbf{V}_{I,P_{dm}}^{HB}\right) dm \tag{2.32}$$

Next using the chain rule, Eq. (2.32) is equivalent to

$$\frac{1}{2}\int_0^{R_{bl}} \frac{d}{dt}\left(2\,\mathbf{V}_{I,P_{dm}}^{HB}{}^{\top}.\frac{\partial}{\partial\zeta_{bl}}\mathbf{V}_{I,P_{dm}}^{HB}\right)dm = \int_0^{R_{bl}}\left[\mathbf{V}_{I,P_{dm}}^{HB}{}^{\top}.\left(\frac{d}{dt}\frac{\partial}{\partial\zeta_{bl}}\mathbf{V}_{I,P_{dm}}^{I}\right)^{HB}\right.$$
$$\left. +\left(\frac{d}{dt}\mathbf{V}_{I,P_{dm}}^{I}{}^{\top}\right)^{HB}.\frac{\partial}{\partial\zeta_{bl}}\mathbf{V}_{I,P_{dm}}^{HB}\right]dm \tag{2.33}$$

with again the following convention for the time-derivatives: $\left(\frac{d}{dt}\frac{\partial}{\partial\zeta_{bl}}\mathbf{V}_{I,P_{dm}}^{I}\right)^{HB}$ signifies the time-derivative, wrt inertial frame F_I, of vector $\frac{\partial}{\partial\zeta_{bl}}\mathbf{V}_{I,P_{dm}}$, subsequently projected onto frame F_{HB}. Using Eq. (2.19), these derivatives can also be expanded as follows

$$\left(\frac{d}{dt}\frac{\partial}{\partial\zeta_{bl}}\mathbf{V}_{I,P_{dm}}^{I}\right)^{HB} = \left(\frac{d}{dt}\frac{\partial}{\partial\zeta_{bl}}\mathbf{V}_{I,P_{dm}}^{HB}\right)^{HB}$$
$$+\begin{pmatrix} p \\ q \\ r \end{pmatrix}^{b} \times \frac{\partial}{\partial\zeta_{bl}}\mathbf{V}_{I,P_{dm}}^{HB} \tag{2.34}$$

$$\left(\frac{d}{dt}\mathbf{V}_{I,P_{dm}}^{I}{}^{\top}\right)^{HB} = \left(\frac{d}{dt}\mathbf{V}_{I,P_{dm}}^{HB}{}^{\top}\right)^{HB} + \begin{pmatrix} p \\ q \\ r \end{pmatrix}^{b} \times \left(\mathbf{V}_{I,P_{dm}}^{T}\right)^{HB} \tag{2.35}$$

Next, for the second term on the LHS of Eq. (2.28a) we get

$$-\frac{\partial K_E}{\partial\zeta_{bl}} = -\frac{\partial}{\partial\zeta_{bl}}\frac{1}{2}\int_0^{R_{bl}}\mathbf{V}_{I,P_{dm}}^{HB}{}^{\top}.\mathbf{V}_{I,P_{dm}}^{HB}\,dm \tag{2.36}$$

Again since the limits of integration are constant, and using the chain rule, Eq. (2.36) reduces to

$$-\frac{\partial K_E}{\partial\zeta_{bl}} = -\int_0^{R_{bl}}\mathbf{V}_{I,P_{dm}}^{HB}{}^{\top}.\frac{\partial}{\partial\zeta_{bl}}\mathbf{V}_{I,P_{dm}}^{HB}\,dm \tag{2.37}$$

Now, through the use of a symbolic math toolbox, an analytic expression for the LHS of Eq. (2.28a) may readily be obtained, i.e. by utilizing the expression obtained for $\mathbf{V}_{I,P_{dm}}^{HB}$ in Eq. (2.21) and inserting it, together with the derivatives $\frac{d}{dt}\mathbf{V}_{I,P_{dm}}^{HB}$, $\frac{\partial}{\partial\zeta_{bl}}\mathbf{V}_{I,P_{dm}}^{HB}$, $\frac{\partial}{\partial\zeta_{bl}}\mathbf{V}_{I,P_{dm}}^{HB}$, into Eq. (2.33), Eq. (2.34), Eq. (2.35), and Eq. (2.37). The blade flap equation Eq. (2.28b) follows a similar procedure, and will also require the computation of $\frac{\partial}{\partial\beta_{bl}}\mathbf{V}_{I,P_{dm}}^{HB}$ and $\frac{\partial}{\partial\beta_{bl}}\mathbf{V}_{I,P_{dm}}^{HB}$. Finally, using a symbolic math toolbox, the combined equations Eq. (2.28a) and Eq. (2.28b) may be re-arranged as the following four-states nonlinear flap-lag equations of motion

$$\frac{d}{dt}\begin{pmatrix} \dot{\beta}_{bl} \\ \dot{\zeta}_{bl} \\ \beta_{bl} \\ \zeta_{bl} \end{pmatrix} = \mathbb{A}^{-1}.\left(-\mathbb{B}.\begin{pmatrix} \dot{\beta}_{bl} \\ \dot{\zeta}_{bl} \\ \beta_{bl} \\ \zeta_{bl} \end{pmatrix} + \begin{pmatrix} Q_{\beta_{bl}} - F_1 \\ Q_{\zeta_{bl}} - F_2 \\ 0 \\ 0 \end{pmatrix}\right) \tag{2.38}$$

with the following $\mathbb{A}$ and $\mathbb{B}$ matrices

$$\mathbb{A} = \begin{bmatrix} I_\beta & 0 & 0 & 0 \\ 0 & (e_F^2.M_{bl} + 2e_F.C_0 + I_\beta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.39}$$

$$
\mathbb{B} = \begin{bmatrix} 0 & B_{12} & 0 & 0 \\ B_{21} & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \tag{2.40}
$$

with M_{bl}, C_0, and I_β defined as (refer also to the nomenclature)

$$
\begin{aligned}
M_{bl} &= \int_0^{R_{bl}} dm \qquad C_0 = \int_0^{R_{bl}} r_{dm}.dm = M_{bl}.y_{G_{bl}} \\
I_\beta &= \int_0^{R_{bl}} r_{dm}^2.dm = M_{bl}.\frac{R_{bl}^2}{3}
\end{aligned} \tag{2.41}
$$

We stress here that Eq. (2.38) is a nonlinear representation since the scalars B_{12} and B_{21} in Eq. (2.40), and F_1, and F_2 in Eq. (2.38) are (nonlinear) functions of ($\dot{\zeta}_{bl}$, β_{bl}, ζ_{bl}). Space restrictions preclude a reprint of the lengthy expressions B_{12}, B_{21}, F_1, and F_2, these can be consulted in Appendix E of [49].

Flap angle as a Fourier series
Blade motion is 2π periodic around the azimuth and may hence be expanded as an infinite Fourier series [26, 41]. Now for full-scale helicopters, it is well known that the magnitude of the flap second harmonic is less than 10 % the magnitude of the flap first harmonic [41, 50]. We assume that this is also the case for small-scale helicopters and hence we neglect second and higher harmonics in the Fourier series. This gives

$$
\beta_{bl}(\psi_{bl}) \simeq \beta_0 + \beta_{1c}\cos\psi_{bl} + \beta_{1s}\sin\psi_{bl} \tag{2.42}
$$

with ψ_{bl} the blade azimuth angle. This harmonic representation of the blade motion defines the rotor Tip-Path-Plane (TPP), resulting in a so-called cone-shaped rotor. The non-periodic term β_0 describes the coning angle, and the coefficients of the first harmonic β_{1c} and β_{1s} describe the tilting of the rotor TPP, in the longitudinal and lateral directions respectively. All three angles may readily be obtained through standard least-squares [51]. Now in steady-state rotor operation, the flap coefficients β_0, β_{1c}, β_{1s} may be considered constant over a 2π blade revolution. Obviously this solution would not be adequate for transient situations such as maneuvering [52], hence in our model we compute, for each new blade azimuth, the instantaneous TPP angles. With regard to TPP dynamics, three natural modes can be identified, i.e. the so-called coning, advancing, and regressing modes. In general, the regressing flapping mode is the most relevant when focusing on helicopter flight dynamics, as it is the lowest frequency mode of the three, and it has a tendency to couple into the fuselage modes [40, 47, 53].

Virtual work and virtual displacements
The determination of the generalized forces $Q_{\zeta_{bl}}$, $Q_{\beta_{bl}}$ in Eq. (2.29a) Eq. (2.29b) requires the calculation of the virtual work of each individual external force, associated with each respective virtual flapping and lead-lag displacements [11]. Let F_{X_i}, F_{Y_i}, F_{Z_i} be the components of the ith external force $\mathbf{F}_i$, acting on blade element dm in frame F_{HB}, then the resulting elemental virtual work done by this force, due to the virtual flapping and lag displacements $\partial\beta_{bl}$ and $\partial\zeta_{bl}$, is given by

$$
dW_i = F_{X_i}dx_{dm} + F_{Y_i}dy_{dm} + F_{Z_i}dz_{dm} \tag{2.43}
$$

with

$$dx_{dm} = \frac{\partial x_{dm}}{\partial \beta_{bl}} \partial \beta_{bl} + \frac{\partial x_{dm}}{\partial \zeta_{bl}} \partial \zeta_{bl} \tag{2.44a}$$

$$dy_{dm} = \frac{\partial y_{dm}}{\partial \beta_{bl}} \partial \beta_{bl} + \frac{\partial y_{dm}}{\partial \zeta_{bl}} \partial \zeta_{bl} \tag{2.44b}$$

$$dz_{dm} = \frac{\partial z_{dm}}{\partial \beta_{bl}} \partial \beta_{bl} + \frac{\partial z_{dm}}{\partial \zeta_{bl}} \partial \zeta_{bl} \tag{2.44c}$$

Now summing up the elemental virtual work, over the appropriate blade span, results in the total virtual work W_i, due to external force $\mathbf{F}_i$, as

$$\begin{aligned} W_i = \int_0^{R_{bl}} \left(F_{X_i} \frac{\partial x_{dm}}{\partial \beta_{bl}} + F_{Y_i} \frac{\partial y_{dm}}{\partial \beta_{bl}} + F_{Z_i} \frac{\partial z_{dm}}{\partial \beta_{bl}} \right) \partial \beta_{bl} \\ + \int_0^{R_{bl}} \left(F_{X_i} \frac{\partial x_{dm}}{\partial \zeta_{bl}} + F_{Y_i} \frac{\partial y_{dm}}{\partial \zeta_{bl}} + F_{Z_i} \frac{\partial z_{dm}}{\partial \zeta_{bl}} \right) \partial \zeta_{bl} \end{aligned} \tag{2.45}$$

Which is set equivalent to

$$W_i = Q_{\beta_{bl},i}.\partial \beta_{bl} + Q_{\zeta_{bl},i}.\partial \zeta_{bl} \tag{2.46}$$

The virtual displacement, in frame F_{HB}, of a blade element dm, located at a distance r_{dm} outboard of the flap hinge, is obtained using Eq. (2.44) and Eq. (2.12) as follows

$$\begin{aligned} \begin{pmatrix} dx_{dm} \\ dy_{dm} \\ dz_{dm} \end{pmatrix}^{HB} &= r_{dm}.\mathbf{dP}_{\beta,r}^{HB}.\partial \beta_{bl} \\ &+ \left[\mathbf{dP}_{\zeta,\bar{r}}^{HB} + r_{dm}.\mathbf{dP}_{\zeta,r}^{HB} \right].\partial \zeta_{bl} \end{aligned} \tag{2.47}$$

with

$$\mathbf{dP}_{\beta,r}^{HB} = \begin{pmatrix} \cos \psi_{bl} \cos \zeta_{bl} \sin \beta_{bl} \\ \Gamma \cos \psi_{bl} \Big(\cos \theta_{bl} \sin \zeta_{bl} \sin \beta_{bl} - \cos \beta_{bl} \sin \theta_{bl} \Big) \\ - \cos \theta_{bl} \cos \beta_{bl} \\ + \sin \psi_{bl} \Big(\cos \theta_{bl} \sin \zeta_{bl} \sin \beta_{bl} - \cos \beta_{bl} \sin \theta_{bl} \Big) \\ -\Gamma \sin \psi_{bl} \cos \zeta_{bl} \sin \beta_{bl} \\ - \sin \zeta_{bl} \sin \theta_{bl} \sin \beta_{bl} \end{pmatrix} \tag{2.48}$$

$$\mathbf{dP}_{\zeta,\bar{r}}^{HB} = e_F \begin{pmatrix} \Big(\cos \psi_{bl} \sin \zeta_{bl} - \sin \psi_{bl} \cos \theta_{bl} \cos \zeta_{bl} \Big) \\ -\Gamma \Big(\cos \psi_{bl} \cos \theta_{bl} \cos \zeta_{bl} + \sin \psi_{bl} \sin \zeta_{bl} \Big) \\ \cos \zeta_{bl} \sin \theta_{bl} \end{pmatrix} \tag{2.49}$$

$$\mathbf{dP}_{\zeta,r}^{HB} = \cos \beta_{bl} \frac{\mathbf{dP}_{\zeta,\bar{r}}^{HB}}{e_F} \tag{2.50}$$

GENERALIZED FORCES (GRAVITY)

The gravity force acting on a blade element with mass dm can be expressed in F_{HB} as

$$\mathbf{F}_{G_{bl}}^{HB} = \mathbb{T}_{(HB)o} \begin{pmatrix} 0 \\ 0 \\ g.dm \end{pmatrix}^o \tag{2.51}$$

with $\mathbb{T}_{(HB)o}$ the transformation from F_o to F_{HB}. Substituting Eq. (2.51) and Eq. (2.47) into Eq. (2.45), the desired generalized forces due to gravity, outboard of the flap hinge, are obtained as follows

$$
\begin{aligned}
Q_{\zeta_{bl},G} = {}& g.\left(e_F.M_{bl} + C_0 \cos\beta_{bl}\right).\Big(A_1 \cos\psi_{bl} \sin\zeta_{bl} \\
& -A_1 \sin\psi_{bl} \cos\theta_{bl} \cos\zeta_{bl} \\
& -A_2 \Gamma \cos\psi_{bl} \cos\theta_{bl} \cos\zeta_{bl} \\
& -A_2 \Gamma \sin\psi_{bl} \sin\zeta_{bl} + A_3 \cos\zeta_{bl} \sin\theta_{bl}\Big)
\end{aligned}
\tag{2.52}
$$

$$
\begin{aligned}
Q_{\beta_{bl},G} = {}& g.C_0.\Big(A_1 \cos\psi_{bl} \cos\zeta_{bl} \sin\beta_{bl} \\
& +A_1 \sin\psi_{bl} \cos\theta_{bl} \sin\zeta_{bl} \sin\beta_{bl} \\
& -A_1 \sin\psi_{bl} \cos\beta_{bl} \sin\theta_{bl} \\
& +A_2 \Gamma \cos\psi_{bl} \cos\theta_{bl} \sin\zeta_{bl} \sin\beta_{bl} \\
& -A_2 \Gamma \cos\psi_{bl} \cos\beta_{bl} \sin\theta_{bl} \\
& -A_2 \Gamma \sin\psi_{bl} \cos\zeta_{bl} \sin\beta_{bl} \\
& -A_3 \cos\theta_{bl} \cos\beta_{bl} - A_3 \sin\zeta_{bl} \sin\theta_{bl} \sin\beta_{bl}\Big)
\end{aligned}
\tag{2.53}
$$

using

$$
\begin{aligned}
A_1 &= -\sin\theta \\
A_2 &= \cos\theta \sin\phi \\
A_3 &= \cos\theta \cos\phi
\end{aligned}
\tag{2.54}
$$

and M_{bl} and C_0 as defined in Eq. (2.41).

GENERALIZED FORCES (AERODYNAMIC)

The aerodynamic velocity, i.e. velocity relative to the air, of a blade element dm, located at position P_{dm}, is defined by $\mathbf{V}_{a,P_{dm}}$. Projecting it onto the blade frame F_{bl} we get

$$\mathbf{V}_{a,P_{dm}}^{bl} = \mathbb{T}_{(bl)(HB)}.\left(\mathbf{V}_{I,P_{dm}}^{HB} - \begin{pmatrix} 0 \\ 0 \\ v_i \end{pmatrix}^{HB} - \mathbb{T}_{(HB)E} \begin{pmatrix} u_w \\ v_w \\ w_w \end{pmatrix}^E\right) \tag{2.55}$$

with $\mathbf{V}_{I,P_{dm}}^{HB}$ defined in Eq. (2.21), v_i the rotor induced velocity[35] from Eq. (2.70), $(u_w\ v_w\ w_w)^\top$ the components of the wind velocity vector usually available in frame F_E, and $\mathbb{T}_{(bl)(HB)}$ the rotation matrix from frame F_{HB} to frame F_{bl}. Now the section AOA of a blade element dm

---

[35] Strictly speaking the induced velocity is perpendicular to the Tip-Path-Plane (TPP). However since we make the assumption of small tilt angles, as to simplify the model, we consider here an induced velocity perpendicular to the Hub-Body frame F_{HB}.

is defined by α_{bl} in the interval $[-\pi, +\pi]$ rad and, for each of the four quadrants, is readily computed from the arctangent of the x- and z- components of $\mathbf{V}^{bl}_{a,P_{dm}}$. Further, the elemental lift and drag forces of a blade segment of length dr_{dm} are given by

$$dL = \frac{1}{2}.\rho.\|\mathbf{V}^{bl}_{a,P_{dm}}\|^2.c_{l_{bl}}.c_{bl}.dr_{dm} \tag{2.56}$$

$$dD = \frac{1}{2}.\rho.\|\mathbf{V}^{bl}_{a,P_{dm}}\|^2.c_{d_{bl}}.c_{bl}.dr_{dm} \tag{2.57}$$

with the blade section lift and drag coefficients $c_{l_{bl}}$ and $c_{d_{bl}}$ given as tabulated functions[36] of blade section AOA and Mach number M, and all other coefficients defined in the nomenclature. The elemental lift and drag forces can now be expressed in the blade frame F_{bl}, for each of the four AOA quadrants. For example, for the case of a CCW main rotor, with the AOA quadrant $\alpha_{bl} \in [0, +\pi/2]$ rad, we have

$$\mathbf{dL}^{bl} = dL.\begin{pmatrix} \sin\alpha_{bl} \\ 0 \\ -\cos\alpha_{bl} \end{pmatrix} \tag{2.58}$$

$$\mathbf{dD}^{bl} = -dD.\begin{pmatrix} \cos\alpha_{bl} \\ 0 \\ \sin\alpha_{bl} \end{pmatrix} \tag{2.59}$$

Coming back to the generalized aerodynamic forces, we can now express them as the sum of two contributions, one due to lift and one due to drag. For the lead-lag case in Eq. (2.29a) we have $Q_{\zeta_{bl},A} = Q_{\zeta_{bl},A_L} + Q_{\zeta_{bl},A_D}$. Similarly for the flap case in Eq. (2.29b) we have $Q_{\beta_{bl},A} = Q_{\beta_{bl},A_L} + Q_{\beta_{bl},A_D}$. Now keeping in mind Eq. (2.45) and Eq. (2.47), and using Eq. (2.58) and Eq. (2.59), we obtain

$$Q_{\zeta_{bl},A_L} = \int_{r_c}^{B.R_{bl}} \left(\mathbb{T}_{(HB)(bl)}\mathbf{dL}^{bl}\right)^\top.\left(\mathbf{dP}^{HB}_{\zeta,\bar{r}} + r_{dm}.\mathbf{dP}^{HB}_{\zeta,r}\right).dr_{dm} \tag{2.60}$$

$$Q_{\zeta_{bl},A_D} = \int_{r_c}^{R_{bl}} \left(\mathbb{T}_{(HB)(bl)}\mathbf{dD}^{bl}\right)^\top.\left(\mathbf{dP}^{HB}_{\zeta,\bar{r}} + r_{dm}.\mathbf{dP}^{HB}_{\zeta,r}\right).dr_{dm} \tag{2.61}$$

$$Q_{\beta_{bl},A_L} = \int_{r_c}^{B.R_{bl}} \left(\mathbb{T}_{(HB)(bl)}\mathbf{dL}^{bl}\right)^\top.\mathbf{dP}^{HB}_{\beta,r}.r_{dm}.dr_{dm} \tag{2.62}$$

$$Q_{\beta_{bl},A_D} = \int_{r_c}^{R_{bl}} \left(\mathbb{T}_{(HB)(bl)}\mathbf{dD}^{bl}\right)^\top.\mathbf{dP}^{HB}_{\beta,r}.r_{dm}.dr_{dm} \tag{2.63}$$

For the lift contributions Q_{ζ_{bl},A_L} and Q_{β_{bl},A_L}, the integration is performed from the blade root cutout r_c to a value denoted as $B.R_{bl}$, this latter accounts for blade tip loss [52]. Next by plugging Eq. (2.48), Eq. (2.50), Eq. (2.49), Eq. (2.58), and Eq. (2.59), into Eq. (2.60)–Eq. (2.63), one can derive final expressions for the generalized aerodynamic forces. Providing analytical expressions for Eq. (2.60)–Eq. (2.63) represents a rather tedious task, even more so for

---

[36]Where we neglect sideslip influence.

twisted blades[37] for which the blade pitch will also be function of the blade section length r_{dm}. Therefore, we opted for a numerical evaluation of these expressions, as is often done in flight dynamics codes [54]. Here Gaussian quadrature integration was implemented, using a low order (5th order) Legendre polynomial scheme [55, 56].

Generalized forces (hub damping and spring restraints)
The flap and lag hinges are modeled as springs with viscous dampers. The generalized forces corresponding to the spring dampers can be obtained directly from the potential energy of the dampers dissipation functions [10, 11] as

$$Q_{\zeta_{bl},D} = -K_{D_\zeta}.\dot{\zeta}_{bl} \qquad Q_{\beta_{bl},D} = -K_{D_\beta}.\dot{\beta}_{bl} \qquad (2.64)$$

Similarly the generalized forces corresponding to the spring restraints can be obtained directly from the potential energy of the hub springs [10, 11] as

$$Q_{\zeta_{bl},S} = -K_{S_\zeta}.\zeta_{bl} \qquad Q_{\beta_{bl},S} = -K_{S_\beta}.\beta_{bl} \qquad (2.65)$$

Rotor inflow
At the heart of the helicopter aerodynamics are the induced velocities, i.e. the induced flow due to rotor blade motion, at and near the main rotor [57]. These induced velocities contribute to the local blade incidence and local dynamic pressure, and can be divided into two categories, static and dynamic inflow models. For low-bandwidth maneuvering applications, such as trim calculations or flying-qualities investigations, the dynamic effects of the interaction of the airmass with the vehicle may be deemed negligible, hence static inflow models may be acceptable [57]. But for high bandwidth applications, dynamic interactions between the inflow dynamics and the blade motion must be considered. Conjointly dynamic inflow models can be divided into two unsteady categories: the Pitt-Peters dynamic inflow [8, 58–60], and the Peters-He finite-state wake model[38] [9, 64, 65]. The finite-state wake model is a more comprehensive theory than dynamic inflow, not limited in harmonics and allowing to account for nonlinear radial inflow distributions. This sophisticated model is particularly attractive when rotor vibration and aeroelasticity need to be analyzed [66]. But with respect to flight dynamics applications, we assume that it is sufficient to consider the normal component of the inflow at the rotor, i.e. the rotor induced downwash [7]. Further, for such applications, it is reported in [66] that the Peters-He model is not remarkably better than the Pitt-Peters formulation. Since our primary interest is flight dynamics, we choose to implement the more straightforward Pitt-Peters model [8, 58], with a correction[39] for flight into the Vortex-Ring-State (VRS) from [68]. The VRS corresponds to a condition where the helicopter is descending in its own wake. It is often associated with the following symptoms: excessive vibrations, large unsteady blade loads, thrust/torque fluctuations, excessive loss of altitude, and loss of control effectiveness [69]. Its boundaries, in terms of helicopter velocities, are well-known, see Fig. 2.19.

---

[37] Although in our case the helicopter UAV blades have zero twist.

[38] Although recent advances in computing power and methodology have made it foreseeable to add a third category, namely that of detailed free-wake models that may be run in real-time for flight dynamics applications [61–63].

[39] Note that, if required, additional enhancements could also be made by including a pseudo-harmonic term to model VRS thrust fluctuations as in [67].
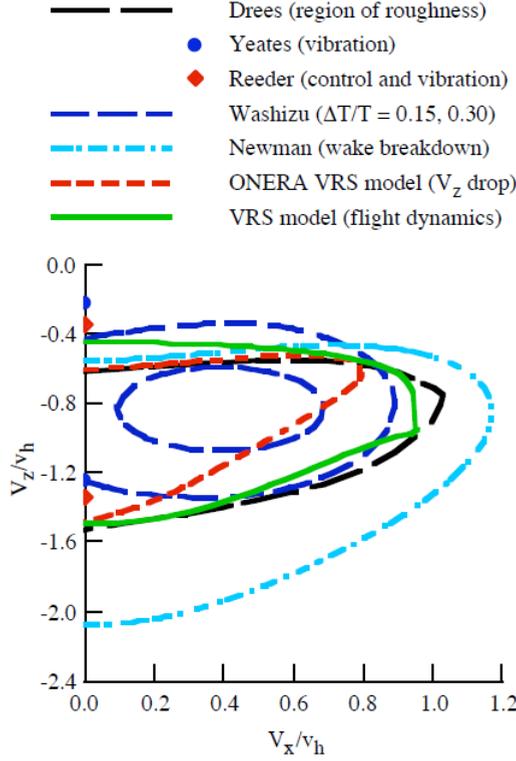
Figure 2.19: Vortex-Ring-State (VRS) boundaries. The x-axis represents the helicopter horizontal velocity normalized by the induced velocity in hover (named v_h in this figure), whereas the y-axis represents the helicopter vertical velocity normalized by the induced velocity in hover. Figure from [70].

Concerning wake bending during maneuvering flight[40], we choose at first not to implement it, as to lower model complexity. For the aspect of ground effect, only a static ground effect has been accounted for, by a correction factor applied to the non-dimensional total velocity at the rotor disk center.

Now, the induced inflow model implemented in this Chapter is based upon [8], and is assumed to have the following variations in the TPP wind-axis coordinates (see [8] for further details on TPP wind-axis coordinates)

$$
\frac{d}{dt}\begin{pmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{pmatrix} = \Omega_{MR}.\mathbb{M}^{-1}.\left(-(\mathbb{L}_1.\mathbb{L}_2)^{-1}.\begin{pmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{pmatrix} + \mathbf{C}_{aero} \right)
\tag{2.66}
$$

where the main rotor RPM Ω_{MR} has been added here in front of the RHS of Eq. (2.66)

---

[40]Wake bending may significantly change the inflow distribution over the rotor, resulting in a sign reversal in the off-axis response [71–73], for which interesting implementation results can be found in [74–76].

since the original expressions of the Pitt-Peters model are in non-dimensional time (see also [64]). The subscript $(\cdot)_{aero}$ in the forcing function $\mathbf{C}_{aero}$ indicates that only aerodynamic contributions are considered, with $\mathbf{C}_{aero} = (C_T \quad - C_L \quad - C_M)^{\top}_{aero}$, and C_T, C_L, C_M, the instantaneous main rotor thrust, roll, and pitching moment coefficients respectively, in the TPP wind-axis system. C_T is readily obtained from Eq. (2.71), whereas C_L and C_M are simply derived from the forces Eq. (2.71) times their respective moment arms. Next matrices $\mathbb{M}$ and $\mathbb{L}_1$ are defined from [8] as

$$\mathbb{M} = \begin{bmatrix} \frac{8}{3\pi} & 0 & 0 \\ 0 & \frac{16}{45\pi} & 0 \\ 0 & 0 & \frac{16}{45\pi} \end{bmatrix} \tag{2.67}$$

$$\mathbb{L}_1 = \begin{bmatrix} \frac{1}{2} & 0 & \frac{-15\pi}{64}\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}} \\ 0 & \frac{4}{1+\sin\alpha} & 0 \\ \frac{15\pi}{64}\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}} & 0 & \frac{4\sin\alpha}{1+\sin\alpha} \end{bmatrix} \tag{2.68}$$

where α represents the wake angle with respect to the rotor disk [8]. Further matrix $\mathbb{L}_2$ is given by

$$\mathbb{L}_2 = \begin{bmatrix} (G_{eff}.V_T)^{-1} & 0 & 0 \\ 0 & V_M^{-1} & 0 \\ 0 & 0 & V_M^{-1} \end{bmatrix} \tag{2.69}$$

with V_T the total velocity through the rotor disk, V_M the momentum theory mass flow parameter, and G_{eff} the static ground effect factor added as a correction to V_T. The expressions for V_T and V_M can be found in [68], although simpler expressions also exist in [8]. However the former include a correction for flight into the VRS and hence are more attractive. The G_{eff} coefficient is based upon the expression found in [52]. Finally the main rotor induced velocity v_i is computed as follows [77]: 1) solve Eq. (2.66); 2) rotate the obtained inflow from the TPP wind-axis to the TPP axis (see [8]); and 3) use these expressions to compute v_i in Eq. (2.70)

$$v_i = V_{ref}.\left(\lambda_0 + \lambda_s.\frac{r_{dm}}{R_{rot}}.\sin\psi_{bl} + \lambda_c.\frac{r_{dm}}{R_{rot}}.\cos\psi_{bl}\right) \tag{2.70}$$

FORCES AND MOMENTS

For the rotor forces, the procedure consists in simulating the forces of each individual blade. This process is repeated at each new blade azimuth position—rather than averaging the results over one revolution—in order to recreate the N_b/Rev flapping vibration[41]. The rotor forces are subdivided into three contributions: 1) aerodynamic lift and drag; 2) inertial; and 3) centrifugal forces. The aerodynamic forces $\mathbf{F}^{HB}_{MR_a}$ are obtained by integrating the elementary lift and drag forces Eq. (2.58) and Eq. (2.59) over the blade span

$$\mathbf{F}^{HB}_{MR_a} = \int_{r_c}^{B.R_{bl}} \mathbb{T}_{(HB)(bl)}\mathbf{dL}^{bl}.dr_{dm} + \int_{r_c}^{R_{bl}} \mathbb{T}_{(HB)(bl)}\mathbf{dD}^{bl}.dr_{dm} \tag{2.71}$$

---

[41]Which may be useful when validating a complete auto-pilot system in a hardware in the loop simulation environment.

where the integrations are done numerically as in Eq. (2.60)–Eq. (2.63). The inertial forces $\mathbf{F}_{MR_i}^{HB}$, due to flap and lag, are approximated, from expressions in [26], as follows

$$\mathbf{F}_{MR_i}^{HB} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \mathbb{T}_{(HB)6} . \begin{pmatrix} -M_{bl}\eta_\zeta\ddot{\zeta}_{bl} \\ 0 \\ -M_{bl}\eta_\beta\ddot{\beta}_{bl} \end{pmatrix} \tag{2.72}$$

Centrifugal forces $\mathbf{F}_{MR_c}^{HB}$ are approximated, from [26], as

$$\mathbf{F}_{MR_c}^{HB} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \mathbb{T}_{(HB)6} . \begin{pmatrix} 0 \\ \frac{1}{2}M_{bl}\Omega_{MR}^2 R_{bl}^2 \\ 0 \end{pmatrix} \tag{2.73}$$

Finally, for the total main rotor forces we have $\mathbf{F}_{MR}^b = \mathbb{T}_{b(HB)} . (\mathbf{F}_{MR_a}^{HB} + \mathbf{F}_{MR_i}^{HB} + \mathbf{F}_{MR_c}^{HB})$, with $\mathbb{T}_{b(HB)} = \mathbb{I}$, since, as mentioned earlier, rotor shaft tilt-angles are zero on our helicopter UAV. For the rotor moments, they include contributions from six different sources: 1) aerodynamics $\mathbf{M}_{MR_a}^{HB}$; 2) inertial loads $\mathbf{M}_{MR_i}^{HB}$; 3) centrifugal loads $\mathbf{M}_{MR_c}^{HB}$; 4) flap hinge stiffness $\mathbf{M}_{MR_{stif}}^{HB}$; 5) lag hinge damping $\mathbf{M}_{MR_{damp}}^{HB}$; and 6) due to airfoil camber $\mathbf{M}_{MR_{camber}}^{HB}$. The last two are neglected since assumed very small for small-scale helicopter rotors/blades. The first three are simply computed by considering the forces Eq. (2.71)–Eq. (2.73) times their respective moment arms. For the flap hinge stiffness, it is derived from [26] as

$$\mathbf{M}_{MR_{stif}}^{HB} = -\frac{1}{1 - \frac{e_P + e_L + e_F}{R_{rot}}} . \frac{N_b . K_{S_\beta}}{2} \begin{pmatrix} \Gamma\beta_{1s} \\ \beta_{1c} \\ 0 \end{pmatrix} \tag{2.74}$$

ROTOR RPM DYNAMICS

The main rotor RPM dynamics is related to the available and required power by [43]

$$N_b . I_b . \Omega_{MR} . \dot{\Omega}_{MR} = P_{shaft} - P_{req} \tag{2.75}$$

with P_{shaft} the available shaft power, and P_{req} the required power to keep the vehicle aloft. This latter is the sum of main rotor induced and profile power, tail rotor induced and profile power, power plant transmission losses, vehicle parasite power (i.e. drag due to fuselage, landing skids, rotor hub, etc), and finally main rotor, tail rotor, and fuselage aerodynamic interference losses. In case of engine failure, a first-order response in P_{shaft} is generally assumed to represent the power decay, we have

$$\dot{P}_{shaft} = -\frac{P_{shaft}}{\tau_p} \tag{2.76}$$

with τ_p a to-be-identified time constant. For the required power P_{req}, we simplify the model by only considering the contributions from the main rotor as $P_{MR} = M_{z\,MR_a}^{HB} . \Omega_{MR}$, with $M_{z\,MR_a}^{HB}$ being the z- component of the aerodynamics moment $\mathbf{M}_{MR_a}^{HB}$ (this latter being referenced in the previous paragraph). Now if, at engine failure, we were to assume an instantaneous power loss $P_{shaft} = 0$, then from Eq. (2.75) we obtain

$$\dot{\Omega}_{MR} = -\frac{M_{z\,MR_a}^{HB}}{N_b . I_b} \tag{2.77}$$

2.10. APPENDIX E: TAIL ROTOR

The tail rotor is a powerful design solution for torque balance, directional stability and control of helicopters. We have implemented here a standard Bailey type model [12], as is done, among others, in [19, 51, 78].

ASSUMPTIONS
Structural simplifications

- The blade has zero twist, constant chord, zero sweep, and has constant thickness ratio. The blade is also rigid, hence torsion is neglected.

Aerodynamics simplifications

- Linear lift with constant lift curve slope, and uniform induced flow over the rotor are assumed.

- Aerodynamic interference effects from the main rotor is neglected, although this may well be an oversimplification, for some flight conditions [79, 80]. Similarly, the aerodynamic interference from the vertical tail (due to blockage) is also neglected.

- Compressibility, blade stall, and viscous flow effects are also disregarded.

Dynamical simplifications

- Blade dynamics is disregarded, and simplified inflow dynamics is considered. Unsteady effects are neglected.

FORCES AND MOMENTS

The theory we apply here is based on the work done by Bailey in [12], implemented among others in [51, 78]. The model given here is a simplified approach of the Bailey model. First, the total tail rotor blade pitch $\tilde{\theta}_{TR}$ is given by

$$\tilde{\theta}_{TR} = \theta_{TR} - T_{TR} \frac{\partial \beta_{0_{TR}}}{\partial T_{TR}} \tan \delta_{3_{TR}} + \theta_{bias_{TR}} \qquad (2.78)$$

with θ_{TR} the tail rotor control input, and all other coefficients defined in the nomenclature, except for T_{TR} defined in Eq. (2.83). The Bailey coefficients are given next by

$$t_1 = \frac{B_{TR}^2}{2} + \frac{\mu_{TRxy}^2}{4} \qquad (2.79a)$$

$$t_2 = \frac{B_{TR}^3}{3} + \frac{B_{TR}\,\mu_{TRxy}^2}{2} \qquad (2.79b)$$

with B_{TR} the tip loss factor and μ_{TRxy} defined in the sequel. Now, assuming zero twist for the tail rotor blades, the downwash at the tail rotor is derived using momentum theory as follows

$$\lambda_{dw} = \frac{c_{l_{(0,TR)}}\sigma_{TR}}{2}\left(\frac{\mu_{TRz}t_1 + \tilde{\theta}_{TR}t_2}{2\sqrt{\mu_{TRx}^2 + \mu_{TRy}^2 + \lambda_{TR}^2} + \frac{c_{l_{(0,TR)}}\sigma_{TR}}{2}t_1}\right) \qquad (2.80)$$

with λ_{TR} the total tail rotor inflow, $\mu_{TRxy} = \sqrt{\mu_{TRx}^2 + \mu_{TRy}^2}$ and μ_{TRz} non-dimensional velocities in the tail rotor frame (see [51] for details of the tail rotor frame and the Bailey model), and the remaining coefficients defined in the nomenclature. The total tail rotor inflow λ_{TR} is further given by

$$\lambda_{TR} = \lambda_{dw} - \mu_{TRz} \qquad (2.81)$$

where it is common practice to iterate between Eqn. (2.80) and Eqn. (2.81) until convergence within a reasonable tolerance. Then, the tail rotor thrust is given by [51]

$$\mathbf{F}_{TR}^b = \begin{pmatrix} 0 \\ \Gamma.T_{TR} \\ 0 \end{pmatrix} \qquad (2.82)$$

with

$$T_{TR} = 2.\lambda_{dw}.\sqrt{\mu_{TRxy}^2 + \lambda_{TR}^2}.\rho.\pi.\left(\Omega_{TR}.R_{rot_{TR}}^2\right)^2 \qquad (2.83)$$

Next, the tail rotor moments are primarily due to the rotor force times the respective moment arms (where we neglect any sidewards rotor offset in the $y-$ direction). For completeness, we also add the rotor torque acting on the pitch axis [26]

$$\mathbf{M}_{TR}^b = \begin{pmatrix} x_{TR} \\ 0 \\ z_{TR} \end{pmatrix}^b \times \mathbf{F}_{TR}^b + \begin{pmatrix} 0 \\ \sigma_{TR}.CD_{TR}/8.(1 + 4.6\mu_{TRxy}^2).\rho.\pi.\Omega_{TR}^2.R_{rot_{TR}}^5 \\ 0 \end{pmatrix}^b \qquad (2.84)$$

2.11. APPENDIX F: FUSELAGE

In the general case, the flow around the fuselage is rather complex, and is characterized by strong nonlinearities, unsteady separation effects, and distortions due to the influence of the main rotor wake [7]. For low speed sideways flight, the important fuselage characteristics are the sideforce, vertical drag, and yawing moment; whereas in forward flight, the important characteristics include drag, and pitching and yawing moments variations with incidence and sideslip [7]. The fuselage rolling moment is usually small, except for configurations with deep hulls where the fuselage aerodynamic center may be significantly below the vehicle CG [7], see also [81, 82] for additional information.

ASSUMPTIONS
Aerodynamics simplifications

- Fuselage aerodynamic enter is collocated with vehicle CG. Further, only steady airloads effects are considered.

- Effect of rotor downwash on fuselage is neglected. It can however be modeled as in [83], using a polynomial in wake skew angle, where the polynomial coefficients need to be fit from flight data [84].

FORCES AND MOMENTS
The fuselage aerodynamic velocity, at its aerodynamic center, in frame F_b, is given by

$$\mathbf{V}^b_{a,Fus} = \left(\begin{array}{c} u + q.z_{Fus} - r.y_{Fus} \\ v - p.z_{Fus} + r.x_{Fus} \\ w + p.y_{Fus} - q.x_{Fus} \end{array} \right)^b - \mathbb{T}_{(HB)E} \left(\begin{array}{c} u_w \\ v_w \\ w_w \end{array} \right)^E \tag{2.85}$$

Now the fuselage model is based upon aerodynamic lift and drag coefficients, which are tabulated as a function of airflow AOA α_{Fus} and sideslip β_{Fus} angles [14]. These angles are readily computed from the x-,y-, and z- components of $\mathbf{V}^b_{a,Fus}$. The fuselage forces in the body frame F_b are

$$\mathbf{F}^b_{Fus} = \left(\begin{array}{c} q_{Fus}.Cx^b_{Fus}(\alpha_{Fus}, \beta_{Fus}) \\ q_{Fus}.Cy^b_{Fus}(\alpha_{Fus}, \beta_{Fus}) \\ q_{Fus}.Cz^b_{Fus}(\alpha_{Fus}, \beta_{Fus}) \end{array} \right) \tag{2.86}$$

with $q_{Fus} = 1/2.\rho.S_{ref_{Fus}}.\|\mathbf{V}^b_{a,Fus}\|^2$. The moments are

$$\mathbf{M}^b_{Fus} = \left(\begin{array}{c} q_{Fus}.Mx^b_{Fus}(\alpha_{Fus}, \beta_{Fus}).L_{ref_{Fus}} \\ q_{Fus}.My^b_{Fus}(\alpha_{Fus}, \beta_{Fus}).L_{ref_{Fus}} \\ q_{Fus}.Mz^b_{Fus}(\alpha_{Fus}, \beta_{Fus}).L_{ref_{Fus}} \end{array} \right) \tag{2.87}$$

with the six aerodynamic coefficients $Cx_{Fus}(\cdot)$, $Cy_{Fus}(\cdot)$, $Cz_{Fus}(\cdot)$, $Mx_{Fus}(\cdot)$, $My_{Fus}(\cdot)$, and $Mz_{Fus}(\cdot)$ being tabulated as a function of airflow AOA α_{Fus}, and sideslip angle β_{Fus}. In our case, these lookup tables are obtained by scaling-down a full-size, Bo–105 helicopter, fuselage aerodynamic model.

2.12. APPENDIX G: VERTICAL AND HORIZONTAL TAILS

The role of the vertical tail is twofold: 1) in forward flight, it generates a sideforce and yawing moment, hence reducing the tail rotor thrust requirement, in order to increase the fatigue life of the tail rotor [7, 43]; and 2) during maneuvers, and during wind gusts, it provides yaw damping and stiffness, enhancing directional stability [7]. The role of the horizontal tail is also twofold: 1) in forward flight, it generates a trim load that reduces the main rotor fore-aft flapping; and 2) during maneuvers, and during wind gusts, it provides pitch damping and stiffness, enhancing pitch stability [7].

ASSUMPTIONS
Aerodynamics simplifications

- The effect of main rotor downwash on both vertical and horizontal tails is neglected. It can however be modeled by using flat vortex wake theory [85] (valid for small sideslip angles), as presented in [54, 86], or it may be modeled as a polynomial in wake skew angle [83].

- We neglect the erratic longitudinal trim shifts that may happen when the helicopter is transitioning from hover to forward flight [7, 43] (as the main rotor wake impinges on the tail surface).

- The effect of the main rotor downwash on the tail boom is neglected, but in some cases may need to be considered during low speed flight, since it may influence yaw damping [7].

FORCES AND MOMENTS
The vertical and horizontal tails, for the case of small-scale helicopters, can simply be viewed as flat plate representations. The force equations are omitted since very similar to those of the fuselage, and the moments are simply derived from the forces times their respective moment arms.

2.13. Appendix H: Problem data

The LTI state-space data used to design the inner-loop trajectory trackers is as follows: the state-vector is of dimension nine given by $\mathbf{x} = (u\ v\ w\ p\ q\ r\ \phi\ \theta\ \psi)^\top$, the control input is of dimension four given by $\mathbf{u} = (\theta_0\ \theta_{1c}\ \theta_{1s}\ \theta_{TR})^\top$, the wind disturbance (given in inertial frame) is of dimension three given by $\mathbf{d} = (V_{N_w}\ V_{E_w}\ V_{Z_w})^\top$, and the measurement vector $\mathbf{y} = \mathbf{x}$.

For the engine ON case, we have: $\dot{\mathbf{x}} = \mathbb{A}\mathbf{x} + \mathbb{B}\mathbf{u} + \mathbb{B}_{\text{wind}}\mathbf{d}$ with

$$
\mathbb{A} = \begin{bmatrix}
-0.0682 & 0.0480 & -0.0154 & -0.1718 & -0.4815 & 0.0443 & 0.0188 & -9.7039 & 0 \\
-0.0528 & -0.1671 & -0.0022 & 0.1405 & -0.6382 & 0.0424 & 9.6314 & -0.0010 & 0 \\
-0.0146 & -0.0086 & -1.3800 & 0.0308 & 0.0191 & -0.1012 & -0.4957 & 0.0025 & 0 \\
-0.4980 & -0.3992 & -0.1785 & -4.3091 & -0.9452 & -0.0176 & -0.8271 & 0.4340 & 0 \\
0.5505 & -0.6741 & 0.0988 & 0.4298 & -7.4087 & 0.0185 & -0.3565 & -0.6686 & 0 \\
-0.0101 & 0.8748 & -0.1435 & 0.0250 & -0.0764 & -1.0801 & 0.8012 & 0.0023 & 0 \\
-0.0445 & -0.0868 & -0.0066 & 0.1002 & -0.4218 & -0.0006 & -0.0610 & 0.0195 & 0 \\
0.0639 & -0.0418 & 0.0207 & 0.1610 & 0.3185 & -0.0517 & -0.0158 & -0.0390 & 0 \\
0.0031 & 0.0777 & -0.0166 & 0.0096 & 0.0118 & 0.8997 & 0.0446 & -0.0024 & 0
\end{bmatrix}
$$

$$
\mathbb{B} = \begin{bmatrix}
-2.1874 & 8.0268 & -15.1285 & 0.3544 \\
-2.0942 & 18.8510 & 19.2744 & -4.4702 \\
-156.7810 & -0.5916 & -5.7372 & -1.1758 \\
-18.4143 & 95.7742 & 23.7919 & -0.5969 \\
10.0410 & -46.1074 & 157.6934 & -1.0295 \\
-140.8456 & -1.7584 & 2.4157 & 94.8389 \\
-0.8650 & 14.1785 & 14.7893 & -0.1586 \\
2.7890 & -10.1952 & 12.5716 & -0.5578 \\
-13.0429 & -0.7975 & 0.9591 & 8.6268
\end{bmatrix}
$$

$$
\mathbb{B}_{\text{wind}} = \begin{bmatrix}
0.0679 & -0.0514 & 0.0130 \\
0.0443 & 0.1652 & 0.0200 \\
0.0137 & -0.0778 & 1.3745 \\
0.4986 & 0.3890 & 0.1970 \\
-0.5507 & 0.6765 & -0.0542 \\
0.0101 & -0.8816 & 0.0897 \\
0.0445 & 0.0862 & 0.0119 \\
-0.0639 & 0.0429 & -0.0179 \\
-0.0031 & -0.0786 & 0.0118
\end{bmatrix}
$$

2

For the engine OFF case, we have: $\dot{\mathbf{x}} = \mathbb{A}\mathbf{x} + \mathbb{B}\mathbf{u} + \mathbb{B}_{\text{wind}}\mathbf{d}$ with

$$\mathbb{A} = \begin{bmatrix}
-0.0719 & 0.0496 & -0.0160 & -0.1855 & -0.4800 & 0.0002 & 0.0183 & -9.7018 & 0 \\
-0.0501 & -0.1312 & -0.0033 & 0.1424 & -0.6648 & -0.0003 & 9.6806 & -0.0003 & 0 \\
-0.0140 & 0.0001 & -1.3528 & 0.0406 & 0.0077 & -0.1111 & -0.0294 & 0.0026 & 0 \\
-0.5237 & -0.3795 & -0.1763 & -4.4032 & -0.7445 & -0.0257 & -0.8290 & 0.4559 & 0 \\
0.5442 & -0.6969 & 0.0885 & 0.3366 & -7.5403 & 0.0116 & -0.3720 & -0.6812 & 0 \\
-0.0040 & 0.0307 & -0.0007 & -0.0185 & -0.0128 & -0.0270 & 0.0172 & 0.0020 & 0 \\
-0.0471 & -0.0874 & -0.0077 & 0.1006 & -0.4271 & -0.0025 & -0.0622 & 0.0209 & 0 \\
0.0659 & -0.0395 & 0.0195 & 0.1624 & 0.3063 & -0.0014 & -0.0137 & -0.0408 & 0 \\
-0.0000 & 0.0019 & 0.0001 & -0.0031 & -0.0019 & 0.9976 & 0.0007 & -0.0001 & 0
\end{bmatrix}$$

$$\mathbb{B} = \begin{bmatrix}
-1.6301 & 10.5701 & -19.1335 & 0.0245 \\
-2.4421 & 20.7349 & 16.1411 & -0.9442 \\
-108.8939 & 2.8097 & -1.7489 & -0.2639 \\
-6.9530 & 89.3397 & -20.1506 & -0.1560 \\
-6.5422 & -0.2979 & 112.9576 & -0.2480 \\
-0.2745 & 0.6763 & 0.3975 & 20.2439 \\
-1.4126 & 13.0013 & 9.9099 & -0.0353 \\
1.0822 & -8.6290 & 13.7359 & -0.0262 \\
-0.0188 & 0.0336 & 0.1177 & 2.1705
\end{bmatrix}$$

$$\mathbb{B}_{\text{wind}} = \begin{bmatrix}
0.0861 & -0.0549 & 0.0168 \\
0.0601 & 0.1340 & 0.0288 \\
0.0052 & 0.0011 & 1.2366 \\
0.4814 & 0.1950 & 0.0751 \\
-0.2975 & 0.6132 & 0.0888 \\
0.0046 & -0.0901 & -0.0031 \\
0.0466 & 0.0745 & 0.0172 \\
-0.0631 & 0.0431 & -0.0129 \\
0.0001 & -0.0066 & -0.0003
\end{bmatrix}$$

REFERENCES

[1] S. Taamallah, *A qualitative introduction to the vortex-ring-state, autorotation, and optimal autorotation,* in *Europ. Rotorcraft Forum* (2010).

[2] S. Taamallah, *Small-scale helicopter blade flap-lag equations of motion for a flybarless pitch-lag-flap main rotor,* in *AIAA Modeling and Simulation Technologies Conf.* (2011).

[3] S. Taamallah, *Flight dynamics modeling for a small-scale flybarless helicopter uav,* in *AIAA Atmospheric Flight Mechanics Conf.* (2011).

[4] ART, (http://www.flightlab.com/, Mountain View CA., U.S.A.).

[5] D. Santamaria, A. Viguria, M. Bejar, K. Kondak, and A. Ollero, *Towards autonomous autorotation landing for small size unmanned helicopters,* J. Intell. Robot Syst. **69**, 171 (2013).

[6] M. G. Ballin, *Validation of a Real-Time Engineering Simulation of the UH-60A Helicopter*, Tech. Rep. TM 88360 (NASA Ames Research Center, 1987).

[7] G. D. Padfield, *Helicopter Flight Dynamics* (Blackwell Science Ltd, Oxford, UK, 1996).

[8] D. A. Peters and N. HaQuang, *Technical notes - dynamic inflow for practical applications,* J. of the Am. Helicopter Soc. , 64 (1988).

[9] D. A. Peters and C. J. He, *Finite state induced flow models part ii: Three-dimensional rotor disk,* AIAA J. of Aircraft **32**, 323 (1995).

[10] D. A. Wells, *Lagrangian Dynamics* (McCraw-Hill Co., 1967).

[11] R. T. N. Chen, *Flap-Lag Equations of Motion of Rigid, Articulated Rotor Blades with Three Hinge Sequences*, Tech. Rep. NTM 100023 (NASA Ames Research Center, 1987).

[12] F. J. Bailey, *A Simplified Theoretical Method of Determining the Characteristics of a Lifting Rotor in Forward Flight*, Tech. Rep. RNo. 716 (NACA, 1941).

[13] J. D. Anderson, *Fundamentals of Aerodynamics. Third Ed.* (McGraw-Hill Higher Education, NY, 2001).

[14] J. L. Boiffier, *The Dynamics of Flight The Equations* (John Wiley & Sons, Chichester, England, 1998).

[15] *Email communication with Professor David Peters* (Center for Computational Mechanics, Washington University in St. Louis, USA, May 11th 2015).

[16] D. A. Peters, M. Chouchane, and M. Fulton, *Helicopter trim with flap-lag-torsion and stall by an optimized controller,* AIAA J. of Guidance **13** (1990).

2

[17] U. B. Hald, M. V. Hesselbaek, and M. Siegumfeldt, *Nonlinear Modeling and Optimal Control of a Miniature Autonomous Helicopter*, Tech. Rep. (Master Thesis, Aalborg University, 2006).

[18] D. A. Peters and D. Barwey, *A general theory of rotorcraft trim,* Mathematical Problems in Engineering **2**, 1 (1996).

[19] G. M. Voorsluijs, *A Modular Generic Helicopter Model*, Tech. Rep. (Master Thesis, Delft University of Technology, 2002).

[20] N. S. Achar and G. H. Gaonkar, *Helicopter trim analysis by shooting and finite element methods with optimally damped newton iterations,* AIAA J. **31** (1993).

[21] D. A. Peters and B. S. Kim, *Control settings for a trimmed stalled rotor by an automatic feedback system,* in *AIAA* (1981).

[22] D. A. Peters and A. P. Izadpanah, *Helicopter trim by periodic shooting with newton-raphson iteration,* in *Am. Helicopter Soc.* (1981).

[23] F. D. Kim, R. Celi, and M. B. Tischler, *Forward flight trim and frequency response validation of a helicopter simulation model,* J. of Aircraft **30** (1993).

[24] J. S. G. McVicar and G. Bradley, *Engineering notes - robust and efficient trimming algorithm for application to advanced mathematical models of rotorcraft,* AIAA J. of Aircraft **32** (1995).

[25] P. M. J. Van den Hof and R. J. P. Schrama, *Identification and control - closed-loop issues,* Automatica **31**, 1751 (1995).

[26] W. Johnson, *Helicopter Theory* (Dover Publications Inc., NY, USA, 1994).

[27] P. Seiler and A. Ozdemir, *An optimal time-invariant approximation for wind turbine dynamics using the multi-blade coordinate transformation,* in *Am. Control Conf.* (2013).

[28] K. Stol, M. Balas, and G. Bir, *Floquet modal analysis of a teetered rotor wind turbine,* Journal of Solar Energy Eng. **124**, 364 (2002).

[29] R. Coleman and A. Feingold, *Theory of Self-Excited Mechanical Oscillations of Helicopter Rotors with Hinged Blades*, Tech. Rep. TR 1351 (NACA, 1958).

[30] D. J. Malcolm, *Modal response of 3-bladed wind turbines,* Journal of Solar Energy Eng. **124**, 372 (2002).

[31] G. Bir, *Multi-blade coordinate transformation and its applications to wind turbine analysis,* in *AIAA Wind Energy Symposium* (2008).

[32] K. A. Stol, H. G. Moll, G. Bir, and H. Namik, *A comparison of multi-blade coordinate transformation and direct periodic techniques for wind turbine control design,* in *AIAA Aerospace Sciences Meeting* (2009).

[33] N. K. Sinha and N. Ananthkrishnan, *Elementary Flight Dynamics with an Introduction to Bifurcation and Continuation Methods* (CRC Press, 2013).

[34] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design, 2nd Ed.* (Wiley-Interscience, Great Britain, 2005).

[35] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements and Performance, Second Edition* (Ganga-Jamuna Press, Lincoln MA, USA, 2006).

[36] B. B. Blake and K. Lunn, *Helicopter stability and control test methodology,* in *AIAA Atmospheric Flight Mechanics Conf.* (1980).

[37] M. B. Tischler and R. Remple, *Aircraft and Rotorcraft System Identification* (Am. Institute of Aeronautics & Astronautics (AIAA), Reston VA, 2006).

[38] E. Seckel and H. C. Curtiss, *Aerodynamic Characteristics of Helicopter Rotors*, Tech. Rep. No. 659 (Department of Aerospace and Mechanical Engineering, Princeton University, 1962).

[39] R. T. N. Chen, *A Simplified Rotor System Mathematical Model for Piloted Flight Dynamics Simulation*, Tech. Rep. NTM 78575 (NASA Ames Research Center, 1979).

[40] R. T. N. Chen, *Effects of Primary Rotor Parameters on Flapping Dynamics*, Tech. Rep. NTP 1431 (NASA Ames Research Center, 1980).

[41] A. Gessow and G. C. Myers, *Aerodynamics of the Helicopter* (College Park Pr, 1999).

[42] R. S. Shevell, *Fundamentals of Flight* (Prentice Hall, Upper Saddle River NJ, 1989).

[43] R. W. Prouty, *Helicopter Performance, Stability, and Control* (Krieger Publishing Company, Malabar, Florida USA, 1995).

[44] R. T. N. Chen, J. V. Lebacqz, E. W. Aiken, and M. B. Tischler, *Helicopter Mathematical Models and Control Law Development for Handling Qualities Research*, Tech. Rep. NCR 2495 (NASA Ames Research Center, 1988).

[45] C. W. Ellis, *Effects of rotor dynamics on helicopter automatic control system requirements,* Aeronautical Engineering Review (1953).

[46] D. G. Miller and F. White, *A treatment of the impact of rotor-fuselage coupling on helicopter handling qualities,* in *Am. Helicopter Soc.* (1987).

[47] H. C. Curtiss, *Stability and control modeling,* in *Europ. Rotorcraft Forum* (1986).

[48] M. B. Tischler, *System identification requirements for high bandwidth rotorcraft flight control system design,* in *AGARD LS-178 Rotorcraft System Identification* (1991).

[49] S. Taamallah, *A Flight Dynamics Helicopter UAV Model For A Single Pitch-Lag-Flap Main Rotor: Modeling & Simulations*, Tech. Rep. NLR-TP-2010-286-PT-1 (NLR, 2011).

2

[50] B. Mettler, *Identification Modelling and Characteristics of Miniature Rotorcraft* (Kluwer Academic Publishers, Norwell Mass, USA, 2003).

[51] ART, *FLIGHTLAB Theory Manual (Vol. One & Two)* (Advanced Rotorcraft Technology, Inc., Mountain View CA, 2006).

[52] G. J. Leishman, *Principles of Helicopter Aerodynamics* (Cambridge University Press, Cambridge, UK, 2000).

[53] R. T. N. Chen and W. S. Hindson, *Influence of high-order dynamics on helicopter flight control system bandwidth,* AIAA J. of Guidance, Control, and Dynamics **9**, 190 (1986).

[54] M. D. Takahashi, *A Flight-Dynamic Helicopter Mathematical Model with a Single Flap-Lag-Torsion Main Rotor*, Tech. Rep. TM 102267 (NASA Ames Research Center, 1990).

[55] D. Kahaner, C. Moler, and S. Nash, *Numerical Methods and Software* (Prentice-Hall, NJ, USA, 1989).

[56] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis* (Springer-Verlag, 2002).

[57] R. T. N. Chen, *A Survey of Nonuniform Inflow Models for Rotorcraft Flight Dynamics and Control Applications*, Tech. Rep. NTM 102219 (NASA Ames Research Center, 1989).

[58] D. M. Pitt and D. A. Peters, *Theoretical prediction of dynamic-inflow derivatives,* in *Europ. Rotorcraft Forum* (1980).

[59] G. H. Gaonkar and D. A. Peters, *Effectiveness of current dynamic-inflow models in hover and forward flight,* J. of the Am. Helicopter Soc. **31** (1986).

[60] G. H. Gaonkar and D. A. Peters, *Review of dynamic inflow modeling for rotorcraft flight dynamics,* Vertica , 213 (1988).

[61] B. D. Kothmann, Y. Lu, E. DeBrun, and J. Horn, *Perspectives on rotorcraft aerodynamic modeling for flight dynamics applications,* in *4th Decennial Specialists Conf. on Aeromechanics* (2004).

[62] D. L. Kunz, *Comprehensive rotorcraft analysis: Past, present, and future,* in *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conf.* (2005).

[63] W. R. M. van Hoydonck, H. Haverdings, and M. D. Pavel, *A review of rotorcraft wake modeling methods for flight dynamics applications,* in *Europ. Rotorcraft Forum* (2009).

[64] D. A. Peters, D. D. Boyd, and C. J. He, *Finite-state induced-flow model for rotors in hover and forward flight,* J. of the Am. Helicopter Soc. , 5 (1989).

[65] D. A. Peters and C. J. He, *Correlation of measured induced velocities with a finite-state wake model,* J. of the Am. Helicopter Soc. **36** (1991).

[66] M. Hamers and P. M. Basset, *Application of the finite state unsteady wake model in helicopter flight dynamic simulation,* in *Europ. Rotorcraft Forum* (2000).

[67] J. Jimenez, A. Desopper, A. Taghizad, and L. Binet, *Induced velocity model in steep descent and vortex ring state prediction,* in *Europ. Rotorcraft Forum* (2001).

[68] D. A. Peters and C. He, *Technical note: Modification of mass-flow parameter to allow smooth transition between helicopter and windmill states,* J. of the Am. Helicopter Soc. , 275 (2006).

[69] J. V. R. Prasad, C. Chen, P. M. Basset, and S. Kolb, *Simplified modeling and non-linear analysis of a helicopter rotor in vortex ring state,* in *Europ. Rotorcraft Forum* (2004).

[70] W. Johnson, *Model for Vortex Ring State Influence on Rotorcraft Flight Dynamics*, Tech. Rep. TP 213477 (NASA Ames Research Center, 2005).

[71] J. D. Keller, *An investigation of helicopter dynamic coupling using an analytical model,* J. of the Am. Helicopter Soc. **41**, 322 (1996).

[72] H. C. Curtiss, *Aerodynamic models and the off-axis response,* in *Am. Helicopter Soc.* (1999).

[73] U. T. P. Arnold, J. D. Keller, H. C. Curtiss, and G. Reichert, *The effect of inflow models on the predicted response of helicopters,* J. of the Am. Helicopter Soc. , 25 (1998).

[74] M. L. Civita, *Integrated Modeling and Robust Control for Full-Envelope Flight of Robotic Helicopters*, Ph.D. thesis, Carnegie Mellon University (2002).

[75] M. Hamers and W. von Grunhagen, *Nonlinear helicopter model validation applied to realtime simulations,* in *Am. Helicopter Soc.* (1997).

[76] B. Benoit, A. M. Dequin, K. Kampa, W. von Grunhagen, P. M. Basset, and B. Gimonet, *Host a general helicopter simulation tool for germany and france,* in *Am. Helicopter Soc.* (2000).

[77] D. M. Pitt and D. A. Peters, *Theoretical prediction of dynamic-inflow derivatives,* Vertica **5**, 21 (1981).

[78] J. A. Houck, F. L. Moore, J. J. Howlett, K. S. Pollock, and M. M. Browne, *Rotor Systems research Aircraft Simulation Mathematical Model*, Tech. Rep. NTM 78629 (NASA Langley Research Center, 1977).

[79] T. M. Fletcher and R. E. Brown, *Main rotor - tail rotor wake interaction and its implications for helicopter directional control,* in *Europ. Rotorcraft Forum* (2006).

2

[80] T. M. Fletcher and R. E. Brown, *Main rotor - tail rotor interaction and its implications for helicopter directional control,* J. of the Am. Helicopter Soc. **53**, 125 (2008).

[81] V. Gavrilets, B. Mettler, and E. Feron, *Nonlinear model for a small-size acrobatic helicopter,* in *AIAA Guidance, Navigation and Control Conf.* (2001).

[82] M. D. Pavel, *On the Necessary Degrees of Freedom for Helicopter and Wind Turbine Low-Frequency Mode-Modelling*, Ph.D. thesis, Delft University of Technology (2001).

[83] P. D. Talbot, B. E. Tinling, W. A. Decker, and R. T. N. Chen, *A Mathematical Model of a Single Main Rotor Helicopter for Piloted Simulation*, Tech. Rep. NTM 84281 (NASA Ames Research Center, 1982).

[84] J. W. Jewel and H. H. Heyson, *Charts of Induced Velocities Near a Lifting Rotor*, Tech. Rep. 4-15-59LY (NASA, 1959).

[85] V. E. Baskin, L. S. Vildgrube, Y. S. Vozhdayev, and C. I. Maykapar, *Theory of the Lifting Airscrew*, Tech. Rep. TT-F-823 (NASA, 1976).

[86] X. Zhao and H. C. Curtiss, *A Study of helicopter Stability and Control Including Blade Dynamics*, Tech. Rep. TR 1823T (NASA Ames Research Center, 1988).

2

3

OFF-LINE TRAJECTORY PLANNING

In preparing for battle I have always found that plans are useless, but planning is
indispensable.

Dwight D. Eisenhower
Quoted in Six Crises, 1962

In this Chapter, we focus on the 'optimal' nature of the autorotative trajectories, gener-
ated by the guidance module, or Trajectory Planning (TP). To this end, we use an off-line
approach to compute open-loop autorotative trajectories, which represent the solution to
the minimization of a cost objective, given system dynamics, controls and states equality
and inequality constraints. We further analyze and compare various 'optimally' defined,
power-off (i.e. autorotative), landing trajectories. The novel part of this Chapter is as fol-
lows. First, we define a new optimal cost functional, for the case of helicopter autorotation,
that maximizes helicopter performance and control smoothness, while minimizing roll-yaw
cross-coupling. Second, we include a trajectory constraint on the tail rotor blade tip, to
avoid ground strike just before touch-down. Third, we apply the recently developed Pseu-
doSpectral (PS) collocation discretization scheme, to solve our optimal control problem
through a direct method. The advantage of the PS method, compared to other direct opti-
mal control approaches, lies in its exponential convergence, provided the functions under
considerations are sufficiently smooth. Finally, we conclude this Chapter by a discussion of
several simulation examples.

Parts of this Chapter have been published in [1–3].

3.1. INTRODUCTION

C ONTROL over position and velocity is a primary objective of an autonomous system. An essential aspect resides in the design of an optimal route/trajectory[1] planning, i.e. a guidance system, that enables it to plan and execute a route/trajectory in a particular environment. Developed originally to meet the specialized needs of the robotics community, route/trajectory planning has been an important research topic in the field of artificial intelligence and robotics for several decades [4–9]. One typically distinguishes between two route/trajectory planning paradigms, namely motion planning methods that attempt to generate a feasible route/trajectory without accounting for obstacles explicitly, and path planning methods, where obstacles are included within the route/trajectory planning [10]. Both can be generated in real-time, on the basis of sensor readings, or generated in advance (e.g. off-line), on the basis of a-priori knowledge.

One class of route/trajectory planning problems, which has seen considerable research activity over the years, is related to the case where a UAV has to travel from point *A* to point *B*, while optimizing a cost objective. This topic is also a main component of our research project. The goal of this thesis, indeed, consists in developing a model-based automatic safety recovery system, for a small-scale helicopter Unmanned Aerial Vehicle (UAV), in un-powered flight, that safely flies and lands the vehicle to a pre-specified ground location. A conceptual design solution, for this research objective, has already been formulated in Section 1.7 of Chapter 1, in the form of a guidance and control logic. Here, the philosophy of the chosen architectural solution decouples the guidance module from the control module. The guidance module, or Trajectory Planning (TP), shall be capable of generating open-loop, feasible and optimal autorotative trajectories references, subject to system and environment constraints, whereas the control module, or Trajectory Tracking (TT), shall ensure that the helicopter flies along these optimal trajectories. Over the years, researchers have addressed the TP problem through several techniques, namely: cell decomposition, potential fields, roadmaps and hybrid systems, inverse dynamics and differential flatness, Mixed Integer Linear Programming (MILP), Model Predictive Control (MPC), optimal control, and finally evolutionary/genetic algorithms [11, 12]. Perhaps the most natural framework for addressing TP problems is the use of optimal[2] control [18]. Hence, optimal control is the method adopted in this Chapter. We further evaluate various optimal autorotative trajectories for the case of a small-scale helicopter. The optimal control inputs (and optimal states), associated with these optimal trajectories, are further obtained using a direct optimal control method, as follows.

First, the constrained, nonlinear, continuous-time, optimal control problem formulation is discretized, using a PseudoSpectral (PS) numerical scheme [19–21]. PS discretization methods exhibit a number of advantages when compared to other discretization methods,

---

[1]The term *trajectory* denotes the *route* that a robot or vehicle should traverse as a function of time.

[2]As a historical note, it is perhaps worth noting that one of the first accounts of constrained optimization dates back to the *Dido Problem*, ca. 850 B.C. [13], where the legendary founder and first queen of Carthage, now in modern-day Tunisia, solved the isoperimetric problem. One of the first publications in the field of optimization can be traced back to the year 1696, and the brachystochrone problem by Johann Bernoulli [14, 15], whereas the first numerical methods for solving optimal control problems date back to the 1950s and 1960s [15], with the work of Bellman in the United States [16], and Pontryagin in the Soviet Union [17].

even when compared to the popular spline parametrization [22–24]. PS methods are indeed known to provide exponential convergence, provided the functions under considerations are sufficiently smooth. PS methods have been extensively used for solving fluid dynamics problems [19, 25]. However, only recently have these methods been used for solving a variety of optimal control problems, e.g. in space and launch/reentry applications [26–43], in aircraft applications [31, 44–47], in helicopter applications [48], in fixed-wing UAV applications [33, 49–52], and in helicopter UAV applications [53, 54]. This said, the work presented in this Chapter represents the first application of the PS collocation discretization scheme towards the helicopter optimal autorotation control problem. Second, and once discretized, the optimal control problem is transcribed to a NonLinear Programming problem (NLP) [55], this latter being solved numerically by a well known and efficient optimization technique, in our case a Sequential Quadratic Programming (SQP) method[3] [60–63].

The remainder of this Chapter is organized as follows. In Section 3.2, the nonlinear optimal control problem is formulated. In Section 3.3 a solution strategy is presented. In Section 3.4, the direct optimal control method is reviewed, together with the pseudospectral discretization approach. In Section 3.5, simulation results are analyzed[4]. Finally, conclusions and future directions are presented in Section 3.6.

3.2. PROBLEM STATEMENT

In this Chapter, we focus upon the 'optimal' nature of the autorotative trajectories, generated by a TP. To this end, we use an off-line approach to compute open-loop autorotative trajectories, which represent the solution to the minimization of a cost objective, given system dynamics, controls and states equality and inequality constraints. We compare various 'optimally' defined, power-off (i.e. autorotative), landing trajectories, and we present what we believe to be the 'best' one.

To start, we need a mathematical model describing the helicopter dynamical behavior. This model (briefly addressed in the sequel) is, to a large part, derived from first-principles, and hence set-up in a nonlinear, continuous-time framework. Now, analytical solutions, through the calculus of variations [64–66], of constrained, nonlinear, continuous-time optimal control problems, can only be derived in the realm of relatively simple mathematical models. Unfortunately, this is not the case of our helicopter flight dynamics application. Consequently, our constrained optimal control problem is not solved analytically, but rather through a numerical algorithm. Now it is well known that solving optimal control problems, numerically, is considered to be difficult, mainly due to the twin curses of dimensionality and complexity. In addition, this difficulty gets exacerbated in the presence of state equal-

---

[3]Although Interior Point (IP) methods could also be used [56–59].

[4]Note that the modeled UAV, used in the simulations of this Chapter in Section 3.5, corresponds to an instrumented Remote-Controlled (RC) Bergen Industrial Twin helicopter, belonging to the flybarless two-bladed main rotor class. This helicopter is different from the one used in the simulations of Chapter 2 (i.e. an instrumented RC Align T-REX helicopter), although both are very similar in terms of size and mass. The reason is here historical: the research described in this thesis started several years ago and, over the years, the focus of the application at NLR had shifted from the larger-size 100 kg Geocopter helicopter UAV, towards the small-scale Bergen Industrial Twin, and finally towards the small-scale Align T-REX helicopter. The latter will also be used in the simulations of Chapters 4 and 5.

ity and inequality constraints. This means that the solution to the optimal control problem may potentially be expensive to compute, and hence selecting a suitable numerical method becomes primordial.

We consider now the following nonlinear optimal control problem, consisting in minimizing the cost functional $J(\mathbf{x}(t), \mathbf{u}(t), T_o, T_f)$, with the state-vector $\mathbf{x}$, and control input-vector $\mathbf{u}$, both defined on compact sets $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$, $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$, denoting the feasible state and control spaces respectively. Here the control input vector $\mathbf{u}$, of dimension four, has been defined in Section 2.2, see Fig. 2.2, as follows

$$\mathbf{u} = \begin{pmatrix} \theta_0 & \theta_{1c} & \theta_{1s} & \theta_{TR} \end{pmatrix}^T \tag{3.1}$$

with the Main Rotor (MR) blade collective pitch θ_0 primarily controlling vertical helicopter motion together with MR Revolutions Per Minute (RPM); the MR blade lateral cyclic pitch θ_{1c} primarily controlling lateral and roll motion; the MR blade longitudinal cyclic pitch θ_{1s} primarily controlling longitudinal and pitch motion; and the Tail Rotor (TR) blade collective pitch θ_{TR} primarily controlling directional (yaw) helicopter motion. The full state-vector $\mathbf{x}$, of dimension twenty-four, has also been defined in Section 2.2, see Fig. 2.2 for the helicopter High-Order Model (HOM) of Chapter 2. Unfortunately, using the HOM of chapter 2 resulted in optimal control problems having a high computational cost. Hence, to lower this computational cost, we developed a simplified model, also known as the Low-Order Model (LOM) see Section 1.7.2 of Chapter 1, which combines the required modeling accuracy with the computational tractability. The LOM uses a state-vector of dimension thirteen, containing only the lower-frequency states, i.e. the twelve rigid-body states together with the main rotor RPM, giving

$$\mathbf{x} = \begin{pmatrix} x_N & x_E & x_Z & u & v & w & p & q & r & \phi & \theta & \psi & \Omega_{MR} \end{pmatrix}^T \tag{3.2}$$

with the nomenclature[5] given in Appendix A of Chapter 2. Here, we have removed the higher-order MR phenomena, i.e. dynamic inflow and blade flap/lag dynamics, from the state-vector $\mathbf{x}$. The bandwidth of the neglected dynamics is generally higher than the bandwidth of the vehicle flight mechanics and TP systems. Hence, and on the grounds of this time-scale separation principle [67], the lack of high frequency modeling detail becomes typically justifiable and acceptable for vehicle guidance applications [68]. The advantage here is in terms of computational savings, with a minimal loss in accuracy and fidelity. We discuss next, in more detail, the 'optimal' nature of the autorotative trajectories generated by our TP.

3.2.1. Cost functional

Over the last four decades, researchers have addressed the optimal autorotative flight problem through several optimization techniques. We start by mentioning the successful autorotative flight demonstration in the case of a small-scale helicopter, through the use of reinforcement learning method in [69, 70]. Other approaches have also focused upon reinforcement learning in [71, 72], and fuzzy-logic concepts in [73, 74]. Next, for the case of first

---

[5]In our nomenclature all vectors are printed in boldface, hence one should not confuse the control input-vector $\mathbf{u}$, printed in boldface, with the vehicle body longitudinal velocity u, printed in regular font.

principles based models, we briefly review the different optimization strategies that have been investigated. Indirect optimal control methods have been used in [75–80], whereas direct optimal control methods have been explored in [2, 3, 68, 81–91]. Aside from these optimal control strategies, three other methods have also been investigated: 1) a nonlinear, neural-networks augmented, model-predictive control method in [92]; 2) a parameter optimization scheme, repeatedly solved, to find a backwards reachable set leading to safe landing in [93, 94]; and 3) a parameter optimization scheme generating segmented routes, selecting a sequence of straight lines and curves in [95–97].

For the definition of the cost functional, most of the here-above listed contributions have focused upon the minimization of vehicle kinetic energy[6] at the instant of touch-down. Some have also considered using a running cost over time, which includes criteria involving either: 1) the minimization of control rates [68, 84, 86, 90]; or 2) the minimization of main rotor RPM deviations from its nominal value, while limiting the excessive build-up of vehicle kinetic energy during the descent [80, 98]. None of the previous results have considered the definition of a cost that includes all of these criteria, while also adding the minimization of vehicle sidewards flight, and maximization of flight into the wind.

3.2.2. BOUNDARY CONDITIONS AND TRAJECTORY CONSTRAINTS

The minimization of the cost functional has to be done while enforcing the system dynamics, and various additional equality and inequality constraints on the controls and states. Specifically, a final-time boundary condition, i.e. at touch-down, is being added in order to: 1) set the vehicle on the ground; and 2) provide tight bounds on the vehicle kinetic energy and attitude angles, in accordance with technical specifications for safe landing. On the other hand, with regard to trajectory constraints, these are set in order to: 1) account for the vehicle's inherent physical and flight envelope limitations (e.g. bounds on speeds, attitude, and main rotor RPM); 2) account for environmental constraints (e.g. the helicopter cannot descend below ground); 3) check for actuators dynamic and range limitations; and finally 4) avoid ground strike by the tail rotor blade tip, just before touch-down. In the sequel, we formalize our TP problem statement.

3.3. THE OPTIMAL CONTROL PROBLEM

In the general optimal control problem formulation, the cost functional $J(\cdot)$ has contributions from a fixed cost $\Phi(\cdot)$, and a running cost over time[7] $\int_\Omega \Psi(\cdot)dt$ such that

$$J(\mathbf{x}(t), \mathbf{u}(t), T_o, T_f) := \Phi(\mathbf{x}(T_o), \mathbf{x}(T_f), T_f) + \int_\Omega \Psi(\mathbf{x}(t), \mathbf{u}(t), t)dt \qquad (3.3)$$

The solution to the optimal trajectory planning gives the optimal control inputs and associated optimal states $\{\hat{\mathbf{u}}(t), \hat{\mathbf{x}}(t)\}$, which minimize this cost functional $J(\cdot)$

$$\{\hat{\mathbf{u}}(t), \hat{\mathbf{x}}(t)\} := \arg \min_{\mathbf{u}(t)\in\mathcal{U}, \mathbf{x}(t)\in\mathcal{X}} J(\mathbf{x}(t), \mathbf{u}(t), T_o, T_f) \qquad (3.4)$$

---

[6]The vehicle kinetic energy is defined as follows: $\frac{1}{2}m_V(u^2 + v^2 + w^2) + \frac{1}{2}(Ap^2 + Bq^2 + Cr^2)$, with A, B, and C the diagonal elements of the inertia matrix $\mathbb{I}_V$.

[7]With the independent time variable t defined over the time domain $\Omega = (T_o, T_f)$, where the final time T_f may be free or fixed.

while enforcing the following constraints:

- The control inputs and states have to satisfy the vehicle dynamics, i.e. a set of first-order Ordinary Differential Equations (ODEs), of the form

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \tag{3.5}$$

As stated earlier, the vehicle model $f(\cdot)$, in Eq. (3.5), does not refer to the helicopter HOM, defined in Eq. (2.3) of Chapter 2. Rather, for the specific purposes of Chapter 3, and in order to reduce the computational cost, we developed a LOM, which was briefly reviewed in Section 1.7.2 of Chapter 1. The modeling process and associated LOM equations are not reprinted here, but can be found in [1].

- An initial-time boundary condition which corresponds, in our case, to the initial values of the control inputs $\mathbf{u}(T_o)$ and states $\mathbf{x}(T_o)$.

- A final-time boundary inequality condition, of the form

$$B_f(\mathbf{x}(T_f), \mathbf{u}(T_f), T_f) \quad \le 0 \tag{3.6}$$

- An algebraic trajectory inequality constraint, of the form

$$T(\mathbf{x}(t), \mathbf{u}(t)) \le 0 \quad t \in \Omega \tag{3.7}$$

where, for generality, the boundary and trajectory constraints Eq. (3.6) and Eq. (3.7) have been expressed as inequality constraints (equality constraints can simply be enforced by equating upper and lower bounds). Further, in Eq. (3.3), and Eq. (3.5)–Eq. (3.7), the five functions $\Phi(\cdot)$, $\Psi(\cdot)$, $f(\cdot)$, $B_f(\cdot)$, and $T(\cdot)$ are all assumed to be sufficiently smooth.

We consider now optimal autorotative trajectories, corresponding to initial conditions for which feasible solutions do exist (this issue will further be addressed in Section 3.5.1). We also choose to set the fixed cost to zero, i.e. $\Phi(\cdot) = 0$. Indeed, since the power-off landing trajectory is feasible, the cost $\Phi(\cdot)$ may equivalently be replaced by tight bounds, adjusted for safe landing, on the final values of vehicle kinetic energy and attitude angles. This in turn simplifies the optimization process, and lowers the computational time. Next, we present what we believe to be the best autorotative trajectory, namely our cost functional $J(\cdot)$ defined, from engineering judgment, as a running cost over time, as follows

$$
\begin{aligned}
J(\mathbf{x}(t), \mathbf{u}(t), T_o, T_f) \quad &\coloneqq \quad \int_\Omega \Psi(\mathbf{x}(t), \mathbf{u}(t), t) dt \\
&= \quad \int_\Omega \Big[W_{\dot{\mathbf{u}}}(\dot{\theta}_0^2 + \dot{\theta}_{1c}^2 + \dot{\theta}_{1s}^2 + \dot{\theta}_{TR}^2) + W_\Omega(\Omega_{MR} - \Omega_{MR_{100\%}})^2 \\
&\qquad + W_u u^2 + W_v v^2 + W_w w^2 + W_\psi(\psi - \psi_f)^2 \Big] dt
\end{aligned}
\tag{3.8}
$$

The term $\dot{\theta}_0^2 + \dot{\theta}_{1c}^2 + \dot{\theta}_{1s}^2 + \dot{\theta}_{TR}^2$ is added to: 1) minimize the battery power consumption[8]; and 2) encourage smoother control policies, hence avoiding *bang-bang* type solutions, that might excite undesirable high frequency dynamics or resonances. The term

---

[8]Actuators on small-scale helicopter UAVs are electrically powered by batteries.

$(\Omega_{MR} - \Omega_{MR_{100\%}})^2$ is added to penalize any large deviations in MR speed from its nominal (power on) value $\Omega_{MR_{100\%}}$. Indeed, a rotor over-speed would increase, beyond acceptable values, the structural stresses on the MR hub and hinges. On the other hand, a rotor under-speed would be unsafe for the following two reasons: 1) it increases the region of blade stall[9], increasing rotor drag and decreasing rotor lift, hence resulting in a higher helicopter sink rate; and 2) it lowers the stored rotor kinetic energy[10], which is a crucial element for a good landing flare[11] capability [99, 100]. The term $u^2 + w^2$ is added to limit the excessive build-up of vehicle kinetic energy during the descent. In particular, a high kinetic energy complicates the flare maneuver, since more energy needs to be dissipated, i.e. the timing of the control inputs becomes increasingly critical [101]. The term v^2 is added to limit vehicle sideslip[12] flight. Large sideslip decreases the flight performance, by increasing vehicle drag, increasing roll/yaw coupling, and hence increasing the workload of any feedback TT controller. The term ψ_f refers to the wind heading angle (known through either on-board measurements, or data-uplink from a ground-based wind sensor), and the term $(\psi - \psi_f)^2$ is added to encourage flight and landing into the wind. This results in better flight performance, and lowers the vehicle kinetic energy at touchdown. Finally, the additional weights, i.e. $W_{\mathbf{\dot{u}}}$, W_{Ω}, W_u, W_v, W_w, and W_{ψ}, have been added to allow for the evaluation of various trade-offs within this cost objective.

TAIL ROTOR GROUND CLEARANCE

Here we specifically address the constraint on the tail rotor blade tip, just before touchdown, during the flare landing maneuver. For the Tail Rotor Blade Tip (TRBT) ground clearance, we define the smallest distance between the TRBT and the ground by the distance $x_{Z_{TRBT}}$ in the vehicle carried normal earth frame F_o, see Fig. 3.1, with the TR radius given by $R_{rot_{TR}}$. Note that both the z-axis of frame F_o, and body frame F_b, are oriented positive downwards. The F_b position of the TR hub is given by (x_{TR}, y_{TR}, z_{TR}), hence the lowest position of the blade tip, for a positive vehicle pitch θ, is given in F_o by

$$x_{Z_{TRBT}} = x_Z + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}^T . \mathbb{T}_{ob} . \begin{pmatrix} x_{TR} - R_{rot_{TR}}.\sin\theta \\ y_{TR} \\ z_{TR} + R_{rot_{TR}}.\cos\theta \end{pmatrix} \tag{3.9}$$

and $x_{Z_{TRBT}} \leq Z_{safety} < 0$, with Z_{safety} a safety margin, and $\mathbb{T}_{ob}$ the transformation from body F_b to the vehicle carried normal earth frame F_o given in Eq. (2.8).

3.4. DIRECT OPTIMAL CONTROL AND DISCRETIZATION METHODS

We choose to solve our optimal control problem through a so-called direct method. In this context, the continuous-time optimal control problem of Section 3.2 is first discretized and the problem is transcribed to a NLP [55, 102], without formulating an alternate set of optimality conditions as done through indirect methods [66]. The resulting NLP can be solved

---

[9]Stall corresponds to a sudden reduction in lift.

[10]The main rotor kinetic energy is defined as follows: $\frac{1}{2}N_b I_b \Omega_{MR}^2$, with N_b the number of blades, and I_b the blade inertia about the rotor shaft.

[11]The flare refers to the landing maneuver just prior to touch-down. In the flare the nose of the vehicle is raised in order to slow-down the descent rate, and further the proper attitude is set for touchdown.

[12]Sideslip flight refers to a vehicle moving somewhat sideways as well as forward, relative to the oncoming airflow.
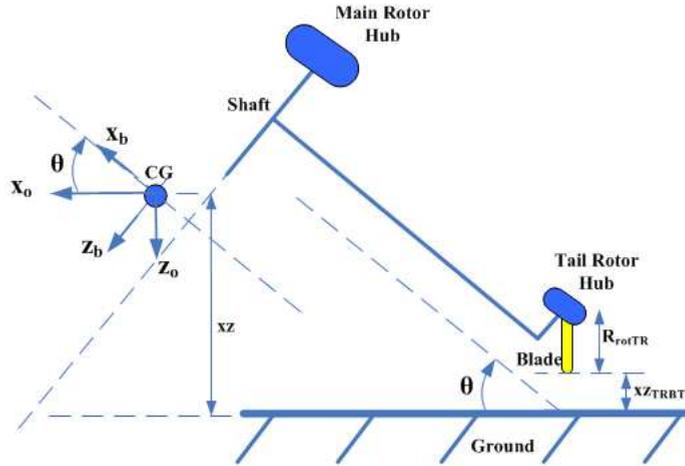
Figure 3.1: Tail rotor ground clearance.

numerically, by well known and efficient optimization techniques, such as SQP methods [61] or Interior Point (IP) methods[13] [103]. These methods in turn attempt to satisfy a set of conditions called the Karush-Kuhn-Tucker (KKT) conditions [55].

Regarding the discretization of the continuous-time optimal control problem, the three most common discretization approaches to solve an indirect or direct method are: 1) Single-Shooting (SS) [104]; 2) Multiple-Shooting (MS) [105]; or 3) State and Control Parameterization (SCP) methods [106, 107]. This latter is sometimes known as transcription in the aerospace community, or as simultaneous strategy in the chemical and process community. Here SS and MS approaches are so-called control parameterization techniques where the control signals alone are discretized, whereas in SCP, as indicated by its name, both state and control are parameterized.

Briefly summarized, in shooting techniques the dynamics are satisfied by integrating the differential equations using a time-marching algorithm. The advantage of direct SS is that it generates a small number of variables, while its main disadvantage is that a small change in the initial condition can produce a very large change in the final conditions [23]. Further, the issue of stability is a major concern. Indeed, time integration over a relatively large shooting segment may lead to erroneous results for unstable systems, and this is why SS generally fails to get a converged solution for such systems [108]. The SS has been most successful in launch vehicle trajectories and orbit transfer problems, primarily because this class of problems lends itself to parameterization with a relatively small number of variables [109]. On the other hand, direct MS breaks the problem into shorter steps, greatly enhancing the robustness of the shooting method, at the cost of having a larger number of variables. It is then primordial to exploit matrix sparsity to efficiently solve the NLP

---

[13]Note that the solution to Eq. (3.4) is often a local minimum, and is also highly sensitive to the initial guess value given to the solver.

equations [109]. Despite the increased size of the problem, the direct MS method is an improvement over the standard direct SS method, because the sensitivity to errors in the unknown initial conditions is reduced, since the differential equations are integrated over smaller time intervals. Further, MS have shown to be suited for applications of high complexity, having a large number of states [110]. However, an additional difficulty exists with the shooting techniques, namely the necessity of defining constrained and unconstrained sub-arcs for problems with path inequality constraints [109]. This latter issue does not exist with SCP methods, which is one of the reasons why SCP methods have actively being investigated. In addition, SCP methods are known to be very effective and robust [110], and SCP techniques have been applied to solve various nonlinear optimal control problems.

In the realm of SCP methods, several discretization procedures have been studied, namely local Runge-Kutta methods in [111], local orthogonal methods in [112], Global Orthogonal Approaches (GOA) or spectral methods in [20, 21, 113–116], and recently hybrid local/global methods in [117]. Of these four procedures, the GOA have received much attention in the last decade, since they have the advantage of providing spectral convergence, i.e. at an exponential rate, for the approximation of analytic, i.e. sufficiently smooth, functions [118]. Thus, for a given error bound, GOA methods generate a significantly smaller scale optimization problem when compared to other methods. This is an important aspect since the efficiency and even convergence of NLPs improves for a problem of smaller size [119]. In a GOA, the state-vector is expressed as a truncated series expansion

$$\mathbf{x}(t) \approx \mathbf{x}_M(t) = \sum_{k=1}^{M} \mathbf{a}_k.O_k(t) \quad t \in \Omega = (T_o, T_f) \tag{3.10}$$

characterized by the *trial* functions $O_k(t)$, or BAsis (BA), and $\mathbf{a}_k$ the Expansion Coefficients (EC) determined from *test* functions, which attempt to ensure that the ODEs are optimally satisfied. The choice of BA is what distinguishes GOA methods from finite-difference or finite-element methods. In both finite-type methods, the BA is local in character, while for GOA methods the BA consists of infinitely differentiable global functions, such as orthogonal polynomials or trigonometric functions. Further, the EC distinguish the three most common types of GOA methods, namely Galerkin, Tau, and collocation. In the sequel, we briefly introduce the GOA collocation method, or PseudoSpectral (PS), used for the discretization of our continuous-time problem. In the collocation approach, the EC are Dirac delta functions centered at M support points P_k, defined by the set $\mathfrak{C} = \{P_k | k \in \{1, ..., M\}\}$. The EC are determined such that: 1) the initial and final-time boundary conditions are met; and 2) the ODEs given by Eq. (3.5) are exactly satisfied on $\mathfrak{C}$ by

$$\dot{\mathbf{x}}_M(t_k) - f(\mathbf{x}(t_k), \mathbf{u}(t_k), t_k) = 0 \quad \forall k \in \{1, ..., M\} \tag{3.11}$$

In addition, the BA is described on $\mathfrak{C}$ by Lagrange interpolating polynomials $L_k(\tau)$ [120]

$$\begin{aligned} \mathbf{x}_M(\tau) &= \sum_{k=1}^{M} \mathbf{a}_k.L_k(\tau) \\ L_k(\tau) &:= \prod_{j=1, j \neq k}^{M} \frac{\tau - \tau_j}{\tau_k - \tau_j} = \frac{h(\tau)}{(\tau - \tau_k)\frac{d}{d\tau}h(\tau)} \end{aligned} \tag{3.12}$$

where the time variable t has been mapped to the pseudospectral interval $\tau \in [-1, 1]$, via the affine transformation $\tau = \frac{2t}{T_f - T_o} - \frac{T_f + T_o}{T_f - T_o}$. We also define $h(\tau) = (1 + \tau).P_M(\tau)$ [21],

where $P_M(\tau)$ is often related to Legendre or Chebyshev polynomials. In our case, we use a Mth-degree Legendre polynomial given by

$$P_M(\tau) := \frac{1}{2^M M!} \frac{d^M}{d\tau^M}[(\tau^2 - 1)^M] \qquad (3.13)$$

Note that Lagrange polynomials are helpful for collocation; it is straightforward to show that $\forall k \in \{1, ..., M\}$

$$L_k(\tau_j) = \delta_{kj} = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases} \qquad (3.14)$$

Hence $\mathbf{x}_M(\tau_k) = \mathbf{a}_k$ on $\mathfrak{C}$, satisfying Eq. (3.11). In a similar way, the input control vector is approximated with a basis of Lagrange polynomials, although not necessarily identical to the previous ones. Besides the choice of $\mathfrak{C}$, another set of K points Q_k, defined by $\mathfrak{Q} = \{Q_k | k \in \{1, ..., K\}\}$, is required for the discretization of the cost functional Eq. (3.3) and the ODEs in Eq. (3.5). Here $\mathfrak{Q}$ is chosen such that the quadrature approximation of an integral is minimized. We have

$$\int_{-1}^{1} g(\tau)d\tau \approx \sum_{k=1}^{K} w_k.f(\tau_k) \quad \tau \in [-1, 1] \qquad (3.15)$$

with w_k the quadrature weights. Now, it is well known that the highest accuracy quadrature approximation, for a given $\mathfrak{Q}$, is the Gauss quadrature. In this case, $\mathfrak{Q}$ is defined by the roots of a Kth-degree Legendre polynomial $P_K(\tau)$, where the corresponding Gauss weights w_k are given from [120] as

$$w_k := \frac{2}{(1 - \tau_k^2)(\frac{dP_K(\tau_k)}{d\tau})^2} \quad \forall k \in \{1, ..., K\} \qquad (3.16)$$

PseudoSpectral methods have been extensively used for solving fluid dynamics problems [19], but only recently have these methods been used for solving a variety of optimal control problems.

3.5. Simulation results

Our MATLAB-based simulation software uses the helicopter LOM presented in Section 1.7.2 of Chapter 1, for the case of a small-scale helicopter UAV. The modeled UAV is an instrumented Remote-Controlled (RC) Bergen Industrial Twin helicopter, belonging to the flybarless two-bladed main rotor class, with a total mass of 8.35 kg, a main rotor radius of 0.93 m, a main rotor nominal angular velocity of 1450 RPM, and a NACA 0015 main rotor airfoil, see Table 3.1.

To solve the nonlinear control problem, the PS numerical method, as described in Section 3.4, is used. This numerical discretization framework is available in a MATLAB environment, through the open-source General Pseudospectral OPtimal control Software GPOPS® [114, 121]. In order to use GPOPS, the optimal control problem must first be reformulated into a GPOPS format, as a set of MATLAB m-files [121]. Second, the helicopter model must also be expressed in a vectorized structure, implying that each model

variable is a time-dependent vector. Third, (cubic) B-Splines interpolating functions ought to be used, when querying lookup tables, since the spectral convergence of PS methods only holds when the functions under consideration are smooth [122]. Finally, it is best practice to non-dimensionalize and scale model variables and quantities, in order to improve conditioning of the numerical problem. Once the control problem is discretized, it is then transcribed into a static, finite-dimensional NLP optimization problem. An NLP is generally sparse, and many well-known efficient optimization techniques exist to numerically solve large-scale and sparse NLPs. In our case, we use the SNOPT® software [62], which solves finite-dimensional optimization problems through SQP. Finally, finite differencing has been used to estimate the objective gradient and constraint Jacobian. We present next simulation results for several case studies, but first we will review the Height-Velocity (H-V) diagram.

3.5.1. THE HEIGHT-VELOCITY (H-V) DIAGRAM

For certain combinations of altitude Above Ground Level (AGL) and airspeed, the capability of a helicopter to perform a safe autorotative landing is limited by the structural and aerodynamic design of the helicopter [123]. In fact, power failure within the dangerous or unsafe regions, defined by these combinations of AGL and airspeed, may result in high risk of severe damage or loss of vehicle. These limiting combinations of AGL and airspeed are often expressed as the Height-Velocity (H-V) diagram[14]. Knowledge of these dangerous regions is important for safety procedures and operational reasons[15].

In Fig. 3.2, a typical H-V diagram for a small-scale helicopter (of similar size to the one considered in this Chapter) is shown. The H-V diagram shows two 'Avoid' zones (in gray), namely: 1) a low-speed zone on the left, containing flight conditions where, if an engine failure were to occur, execution of a safe landing would be unlikely, because of insufficient initial energy; and 2) a high-speed zone on the right where, if an engine failure were to occur, safe landing would also be unlikely, because the helicopter would possess insufficient altitude to perform the flare (necessary to reduce the kinetic energy).

Now H-V diagrams can either be compiled from flight tests [132], or by solving optimal control problems. The latter is the approach adopted in this Chapter, where the H-V diagram becomes the solution of an optimization problem, similar to the general one pre-

---

[14] Also called the *deadman's* zone.

[15] Ideally, one would like to eliminate these unsafe regions altogether, or at least reduce their size. H-V studies can be traced back to the late 1950s and early 1960s [124–126]. For example, eliminating the H-V restrictions was demonstrated with the *Kolibrie* helicopter, built by the *Nederlandse Helikopter Industrie (NHI)* in the late 1950s. It was designed by Dutch helicopter engineers and pioneers Jan M. Drees and Gerard F. Verhage. The helicopter was ram-jet powered, these latter being positioned at the blade tips, resulting in very high main rotor inertia. The H-V subject was also investigated in [123], where flight-test data was used to derive semi-empirical functions of a generalized non dimensional H-V diagram, independent of density altitude and gross weight variations. In [127] it was pointed out that high rotor inertia, low disk loading, and a high maximum thrust coefficient could reduce the size of the unsafe zone. In [128, 129], the concept of the so-called High Energy Rotor (HER) was studied, using blades with high rotational inertia. The goal of the HER was to eliminate the unsafe regions, but also to allow for less demanding autorotation maneuvers, and finally use the rotor kinetic energy as a source of transient power for better maneuverability. Additional results can also be found in [130, 131] where recent flight tests, related to the H-V subject with the Bell 430 and 407 helicopters, have been presented.
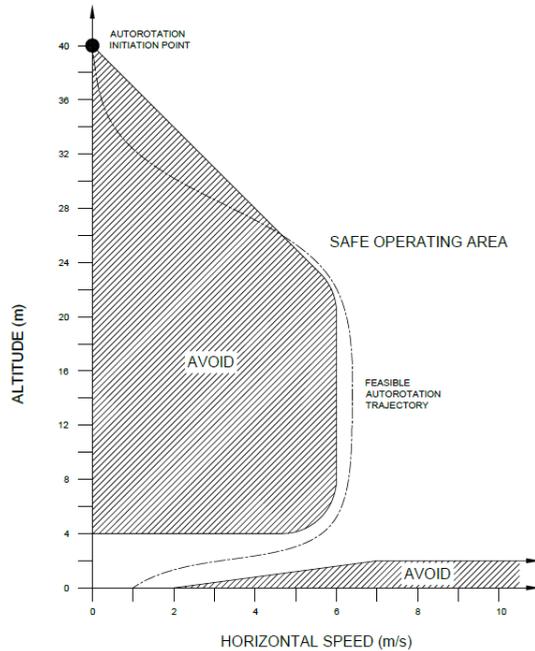
Figure 3.2: Typical Height-Velocity (H-V) diagram for a small-scale helicopter UAV. Figure from [132, 133] (the numbers on the axes are indicative only).

3

sented in Section 3.2. To find the H-V curve, two approaches may be pursued, either: 1) a minimization/maximization of altitude problem subject to safe landing; or 2) simply testing a feasibility problem in terms of safe landing. The minimization/maximization (former) approach often led the solver to run into numerical difficulties. These difficulties were caused by: 1) the inclusion of highly nonlinear lookup tables which, despite B-Splines interpolation, have shown to have a detrimental effect on problem smoothness; and 2) the possible existence of a large number of solutions that all yield approximately the same value of the cost objective. In other words, the objective index is rather insensitive to the solution trajectory in the neighborhood of the optimal solution. On the other hand, for the feasibility approach, the cost objective $J(\cdot)$ in Eq. (3.3) is set to zero, and one only requires to check whether a safe landing is possible, for a range of initial conditions. This method was successfully applied, based upon specific flight envelope boundaries given in Table 3.2, with results shown in Fig. 3.3, for a relatively coarse grid having steps of 1 m in AGL and 1 m/s in airspeed. We found that our helicopter UAV exhibited only the so-called low-speed unsafe zone. We further subdivided this unsafe zone into two sub-zones: 1) one zone, shown in red, which always resulted in unsafe landings, independently of the initial guess conditions given to the solver; and 2) one zone, shown in magenta, which resulted in either safe or unsafe landings, depending on the initial guess conditions values given to the solver.

3.5.2. EVALUATION OF COST FUNCTIONALS

In this section we evaluate and compare our cost functional, defined in Eq. (3.8) and referenced as J_1 in Table 3.3, to three other cost functionals referenced as J_2–J_4 in Table 3.3,
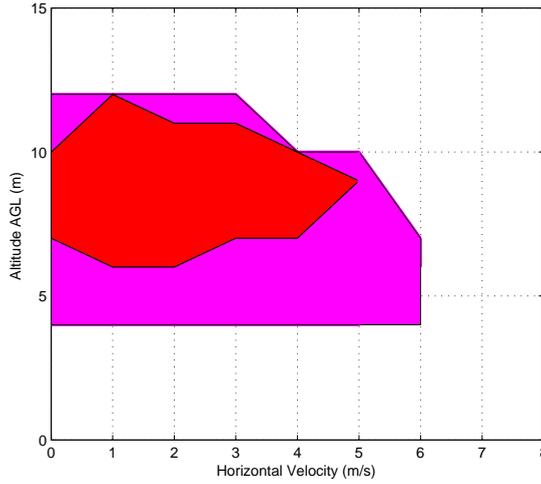
Figure 3.3: Height-Velocity (H-V) diagram for the Bergen Industrial Twin.

which can be found in the literature. In Table 3.3, J_2 is a final-time only cost, which purpose is to minimize the vehicle kinetic energy (as well as having a horizontal attitude) at the instant of touch-down. J_3 is a running cost over time, minimizing actuators activity. J_4 is also a running cost over time, which objective is to keep the main rotor RPM in the neighborhood of its nominal value, while minimizing the vehicle kinetic energies in the longitudinal and vertical channels[16].

For the comparison of these cost functionals, we consider an initial condition, outside of the H-V diagram, defined as follows: steady-state hover, at 40 m altitude, in a zero-wind environment. Here for the analysis of each cost functional $\{J_j\}_{j=1}^4$, we consider the following power metrics $\{P_{ij}\}_{i=1,j=1}^{i=3,j=4}$, of the vector-valued discrete-time signal $\{z_i(n)\}_{i=1}^3$, defined as

$$
\begin{aligned}
z_1(n) &= \begin{bmatrix} \dot{\theta}(n)_0 & \dot{\theta}(n)_{TR} & \dot{\theta}(n)_{1c} & \dot{\theta}(n)_{1s} \end{bmatrix}^\top \\
z_2(n) &= \begin{bmatrix} u(n) & v(n) & w(n) \end{bmatrix}^\top \\
z_3(n) &= \begin{bmatrix} v(n) & \phi(n) \end{bmatrix}^\top \\
P_{ij} &= \frac{1}{N_j} \|z_i(n)\|_{l_2}^2 = \frac{1}{N_j} \sum_{n=1}^{n=N_j} \|z_i(n)\|_2^2
\end{aligned}
\tag{3.17}
$$

with N_j the number of data points of the optimization problems, corresponding to the cost functional $\{J_j\}_{j=1}^4$, and $\|\cdot\|_{l_2}$ the norm on the square-summable sequence space l_2. In Eq. (3.17), the power metric P_{1j}, based upon signal z_1, shows the control rates, i.e. the level of input control activity. This information is relevant, since a higher level of actuator dynamics means a higher power consumption from the batteries, and a higher likelihood of exciting undesirable high frequency dynamics or resonances. Next, the power metric P_{2j}, based upon signal z_2, reflects the amount of stored kinetic energy, during the flight, on the combined three linear channels[17]. Finally, the power metric P_{3j}, based upon signal z_3,

---

[16]The channels with most energies.

[17]In general the kinetic energy, stored in the rotational channels, is much smaller than the one stored in the linear channels.

mirrors the amount of vehicle lateral motion. For each cost functional test case $\{J_j\}_{j=1}^4$, the signal power metrics P_{ij} are reported in Table 3.4. They are obtained by solving nonlinear optimal control problems, based upon 29 nodes[18] discretization, and yielding a NLP having 607 variables and 506 constraints. Now, an analysis of Table 3.4 shows that:

- Our cost functional, defined in Eq. (3.8), considerably reduces lateral control activity, e.g. compare P_{31} to P_{32}, P_{33} and P_{34}. The benefits of reduced lateral motion are increased flight performance, and decreased roll-yaw coupling. This aspect is particularly relevant when repositioning the current discussion within the, two-degree of freedom, guidance and control logic, formulated in Section 1.7 of Chapter 1. Indeed, the chosen flight control architectural solution decouples the guidance module from the control module. The guidance module shall generate open-loop feasible and optimal autorotative trajectory references, whereas the control module shall ensure that the helicopter flies along these optimal trajectories. Hence, a decrease in the amount of cross-coupling of the planned trajectories, results also in a decrease of the workload of the feedback controller.

- J_1 and J_3 display the lowest level of input control activity, confirmed by the power values $P_{11} = 0.06$ and $P_{13} = 0.008$, since both costs include the input rates. Note that excluding the control rates from the cost functional may lead to the excitation of unmodeled or undesirable high frequency modes, potentially resulting in closed-loop instability. This is particularly relevant when the subsequent synthesis of a feedback controller is based upon low-order model representations.

- If a running cost over time is to be used, versus a final-time only cost (such as J_2), then: 1) minimization of control rates ought to be included, since we have $P_{14} = 1.87$ much higher than $P_{12} = 0.27$; and 2) lateral motion should also be included, compare the high values of P_{33} and P_{34}, to the lower value of P_{32}.

- Better performance can be achieved from the use of a final-time only cost, such as J_2, than from a poorly defined running cost over time, such as J_4.

- Finally, the last column in Table 3.4 gives the total signal power $\sum_{i=1}^3 P_{ij}$, for each cost functional $\{J_j\}_{j=1}^4$, where we can see that our cost functional provides the best autorotative trajectory. This said, this experiment, consisting in comparing various cost functionals, was only conducted for a single initial condition, namely steady-state hover, at 40 m altitude, in a zero-wind environment. Although this condition is representative enough of a typical initial condition for a small-scale helicopter, it would indeed be interesting to obtain additional signal power values, corresponding to a wide spectrum of initial conditions.

---

[18]Based upon simulation results, the choice of 29 points provided a good compromise between accuracy and computational tractability. We do acknowledge that this is a rather empirical justification. In fact a more rigorous analysis of the following trade-off: *accuracy* vs. *computational tractability* is here desirable.

3.5.3. Optimal autorotations: effect of initial conditions

In this section, we use our cost functional J_1, defined in Eq. (3.8), to briefly evaluate the effect of variations in initial conditions, i.e. initial altitude and initial speed only. Further, we consider only a limited number of initial trimmed flight conditions, in a zero-wind environment. In these simulations, the final landing spot in terms of North and East position is left completely free, hence not prescribed or constrained to a specific location. Finally, the problem discretization is based upon 33 nodes[19], yielding a NLP problem having 691 variables and 578 constraints.

Effect of initial altitude

We analyze here the effect of three different initial altitudes Above Ground Level (AGL), see Table 3.5, all starting from hover, in a Southbound path (i.e. with the heading oriented towards the South pole). From Fig.3.4 we see the MR collective θ_0 going full-down, as soon as the maneuver initiates (in all figures the magenta horizontal lines display hard bounds on variables). As expected[20], this is necessary in order to minimize the decay in MR RPM Ω_{MR}. Indeed, from Fig.3.6, we see that at a time of approximately 1.5 seconds into the flight, the MR RPM Ω_{MR} does not drop more than 10% of its nominal value. We also clearly see the MR collective θ_0 sharply increasing as the helicopter nears to the ground, to prevent rotor over-speed, while reducing the sink rate. In addition, the MR longitudinal cyclic θ_{1s}, given in the lower plot of Fig. 3.5, is used to: 1) manage vehicle and MR kinetic energies; 2) reduce forward airspeed; and 3) level the attitude for a proper landing. For instance, this can be checked on the pitch angle θ plot, in Fig.3.7, where for a low altitude AGL initial condition, we see the vehicle pitch-up and pitch-down during the flare (i.e. the maneuver just prior to touch-down). Fig. 3.8 presents the trajectory body velocities, where we note that, for hover initial conditions, the higher the initial altitude AGL, the more the optimal trajectories resemble a pure vertical motion (i.e. with minimal horizontal motion), confirming thus the earlier results in [76].

Effect of initial airspeed

We analyze here the effect of three different initial airspeeds, see Table 3.6, all starting at 40 m AGL, again in a Southbound path. Here, we only discuss the salient features of these three cases. For the control inputs, in Fig.3.10 and Fig.3.11, the behavior is comparable to the one observed in the preceding paragraph. We also do note the limited displacement of the MR lateral cyclic θ_{1c}, and TR collective θ_{TR}, consistent with the anticipated behavior of reduced lateral motion. Next, from Fig.3.13–Fig.3.15, we notice that, despite clear differences in initial kinetic energy, the flight time (and rate of descent) show little variations. This could potentially indicate that the flight time, in autorotation, is only lightly correlated with the initial vehicle velocity.

On the other hand, the traveled distance does slightly increase as a function of initial kinetic energy, see upper plot in Fig.3.15. Also an increase in initial kinetic energy does seem to impact the flare maneuver, e.g. for case $C6$ the MR longitudinal cyclic θ_{1s}, in

---

[19]Based upon simulation results, the choice of 33 points provided a good compromise between accuracy and computational tractability. We do acknowledge that this is a rather empirical justification. In fact a more rigorous analysis of the following trade-off: *accuracy* vs. *computational tractability* is here desirable.

[20]This is also what helicopter pilots do at the beginning of an autorotation maneuver.

Fig.3.11, and the helicopter pitch θ, in Fig.3.13, exhibit almost a 'double' flare approach in the last two seconds of the flight. In addition, we see that if differences are to be noted, between on the one hand the hover and low-speed cases—$C4$ and $C5$—and on the other the high-speed case $C6$, then they would tend to primarily appear on the longitudinal θ and RPM Ω_{MR} channels, during the initial flight phase[21], see Fig.3.11–Fig.3.13.

3.6. CONCLUSION

In this Chapter, we have addressed the autorotative Trajectory Planning (TP) problem, for the case of a small-scale helicopter UAV, and we have formulated the technological/engineering TP problem into a mathematical, model-based, nonlinear optimal control problem. The latter was numerically solved, through a direct optimal control framework. The main benefits of this Chapter are threefold. First, we found that for fixed initial altitude, increasing the initial velocity had only a relatively limited effect on the optimal trajectory flight time. On the other hand, the flight time showed a strong correlation with the initial altitude. This aspect, together with the knowledge of an optimally defined autorotative trajectory, will prove useful in the following Chapter. Second, for a range of initial conditions, optimal autorotative trajectories could potentially be computed, off-line, by this TP, and stored as lookup tables, on-board a flight control computer. These trajectories would then provide, both, the optimal states to be tracked by a feedback controller, and optionally the feedforward nominal control inputs. Third, the optimization framework, developed here, could allow to study the effects of some particular factors, affecting the optimal trajectories. These factors include wind, but also some helicopter specific aspects, such as helicopter mass, number of main rotor blades, main rotor blade mass, and main rotor inertia.

---

[21]Approximately the first second into the flight.

Table 3.1: Bergen Industrial Twin physical parameters.

	Name	Parameter	Value	Unit
Environment	Air density	ρ	1.2367	kg/m^3
	Static temperature	T	273.15 + 15	K
	Specific heat ratio (air)	γ	1.4	
	Gas constant (air)	R	287.05	$J/kg.K$
	Gravity constant	g	9.812	m/s^2
Vehicle	Total mass	m	8.35	kg
	Inertia moment wrt x_b	A	0.338	$kg.m^2$
	Inertia moment wrt y_b	B	1.052	$kg.m^2$
	Inertia moment wrt z_b	C	1.268	$kg.m^2$
	Inertia product wrt x_b	D	0.001	$kg.m^2$
	Inertia product wrt y_b	E	0.002	$kg.m^2$
	Inertia product wrt z_b	F	0	$kg.m^2$
Main Rotor (MR)	Direction of rotation ClockWise (CW) Counter-ClockWise (CCW)	Γ	-1 CW	
	Number of blades	N_b	2	
	Nominal angular velocity	$\Omega_{MR_{100\%}}$	151.84	rad/s
	Rotor radius from hub	R_{rot}	0.933	m
	Blade mass	M_{bl}	0.218	kg
	Spring restraint coef. due to flap	K_{S_β}	271.16	$N.m/rad$
	Distance between hub and flap hinge	Δ_e	0.094	m
Tail Rotor (TR)	Number of blades	N_{bTR}	2	
	Nominal angular velocity	$\Omega_{TR_{100\%}}$	709.11	rad/s
	Rotor radius from rotor hub	$R_{rot_{TR}}$	0.17	m
Actuators	MR collective	θ_0	$[-2.8,13.7].\pi/180$	rad
	MR lateral cyclic	θ_{1c}	$[-6.8,6].\pi/180$	rad
	MR longitudinal cyclic	θ_{1s}	$[-7.8,5].\pi/180$	rad
	TR collective	θ_{TR}	$[-27,32.8].\pi/180$	rad
	MR collective rate	$\dot{\theta}_0$	$[-52,52].\pi/180$	rad/s
	MR lateral cyclic rate	$\dot{\theta}_{1c}$	$[-52,52].\pi/180$	rad/s
	MR longitudinal cyclic rate	$\dot{\theta}_{1s}$	$[-52,52].\pi/180$	rad/s
	TR collective rate	$\dot{\theta}_{TR}$	$[-120,120].\pi/180$	rad/s

Table 3.2: Flight envelope boundaries for the Bergen Industrial Twin.

	Definition	Parameter	Range	Unit
	Roll angle	ϕ	$[-48,48].\pi/180$	rad
	Pitch angle	θ	$[-48,48].\pi/180$	rad
	Yaw angle	ψ	$[0,360].\pi/180$	rad
	Body longitudinal velocity	u	$[-5,20]$	m/s
Flight	Body lateral velocity	v	$[-5,5]$	m/s
Envelope	Body vertical velocity	w	$[-5,20]$	m/s
	Body roll angular velocity	p	$[-200,200].\pi/180$	rad/s
	Body pitch angular velocity	q	$[-200,200].\pi/180$	rad/s
	Body yaw angular velocity	r	$[-400,400].\pi/180$	rad/s
	Main rotor RPM	Ω_{MR}	$[70\%,110\%]\,\Omega_{MR_{100\%}}$	rad/s

3

Table 3.3: Comparison of cost functionals.

Test Case	Cost Functional
J_1	Our definition as given in Eq. (3.8)
J_2 similar to [76–78, 91]	$J := \Phi(\mathbf{x}(T_f), T_f)$ $= u(T_f)^2 + v(T_f)^2 + w(T_f)^2$ $+p(T_f)^2 + q(T_f)^2 + r(T_f)^2$ $+\phi(T_f)^2 + \theta(T_f)^2$
J_3 similar to [68, 89]	$J := \int_\Omega \Psi(\mathbf{u}(t))dt$ $= \int_\Omega (\dot{\theta}_0^2 + \dot{\theta}_{TR}^2 + \dot{\theta}_{1c}^2 + \dot{\theta}_{1s}^2)dt$
J_4 similar to [88]	$J := \int_\Omega \Psi(\mathbf{x}(t))dt$ $= \int_\Omega \left[(\Omega_{MR} - \Omega_{MR_{100\%}})^2 + (u^2 + w^2)\right]dt$

Table 3.4: Comparison of signal power for various cost functionals.

Test Case	Control rates P_{1j}	3D linear motion P_{2j}	Lateral motion P_{3j}	Total power $\sum_{i=1}^{3} P_{ij}$
J_1	0.06	50	0.06	50.12
J_2	0.27	62.3	1.7	64.27
J_3	0.008	46.2	14.9	61.108
J_4	1.87	46.2	16.5	63.57

Table 3.5: Initial trimmed flight conditions: autorotations with variation of initial altitude Above Ground Level (AGL).

Test Case	Airspeed (m/s)	Altitude (AGL) (m)	Line Color in Figures
$C1$	hover	25	Red (solid line)
$C2$	hover	40	Blue (dotted line)
$C3$	hover	110	Black (dashed line)

Table 3.6: Initial trimmed flight conditions: autorotations with variation of initial airspeed.

Test Case	Airspeed (m/s)	Altitude (AGL) (m)	Line Color in Figures
$C4$	hover	40	Red (solid line)
$C5$	5	40	Blue (dotted line)
$C6$	15	40	Black (dashed line)
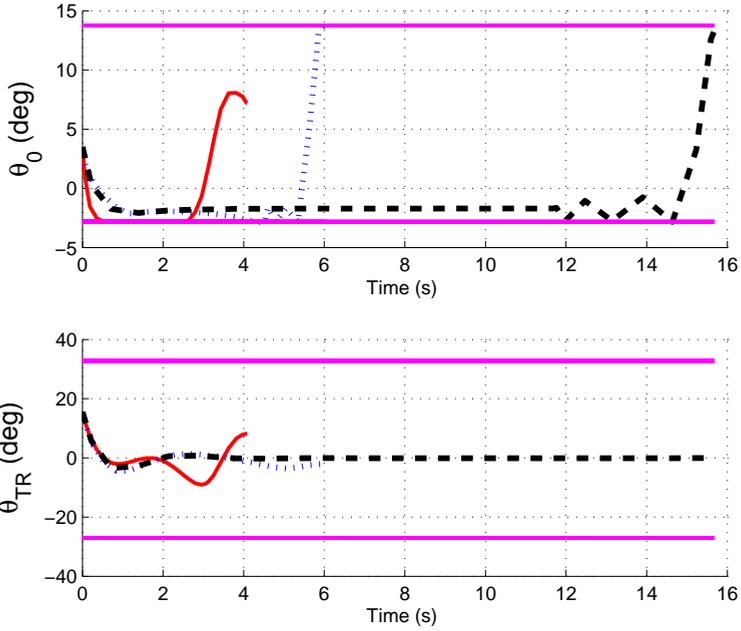
Figure 3.4: MR collective control input θ_0 and TR collective control input θ_{TR} (variation of initial altitude according to Table 3.5).

3
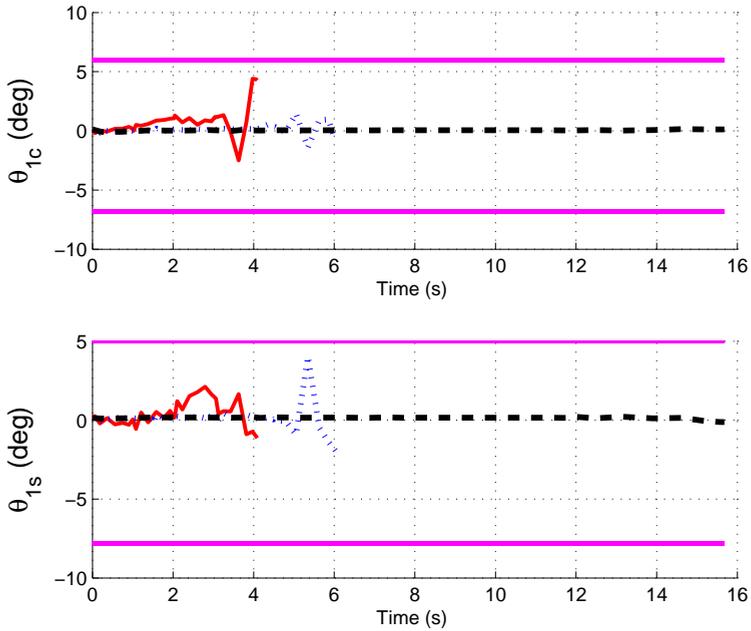
Figure 3.5: MR lateral cyclic control input θ_{1c} and MR longitudinal cyclic control input θ_{1s} (variation of initial altitude according to Table 3.5).

Figure 3.6: MR RPM Ω_{MR} (variation of initial altitude according to Table 3.5).

3
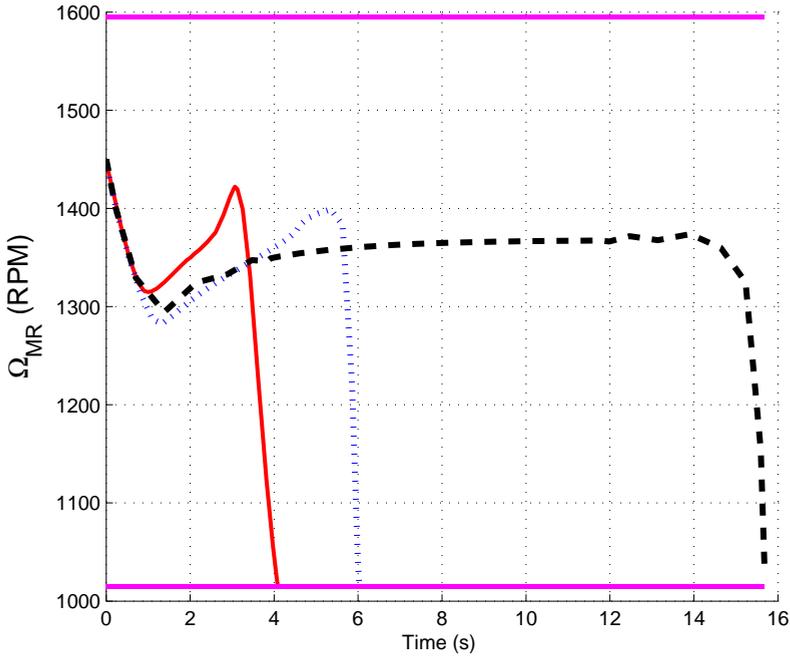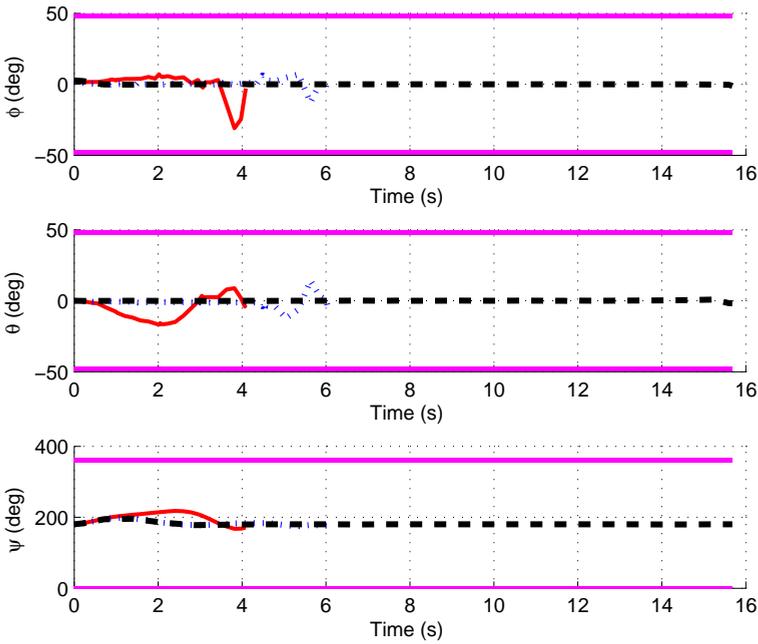
Figure 3.7: Euler angles: roll angle ϕ, pitch angle θ, yaw angle ψ (variation of initial altitude according to Table 3.5).
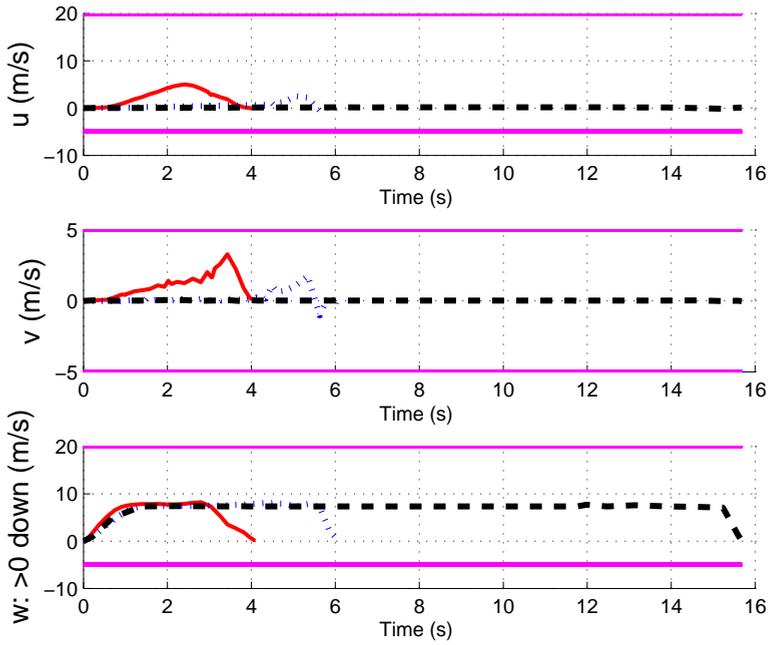
Figure 3.8: Body linear velocities: longitudinal velocity u, lateral velocity v, vertical velocity w (variation of initial altitude according to Table 3.5).
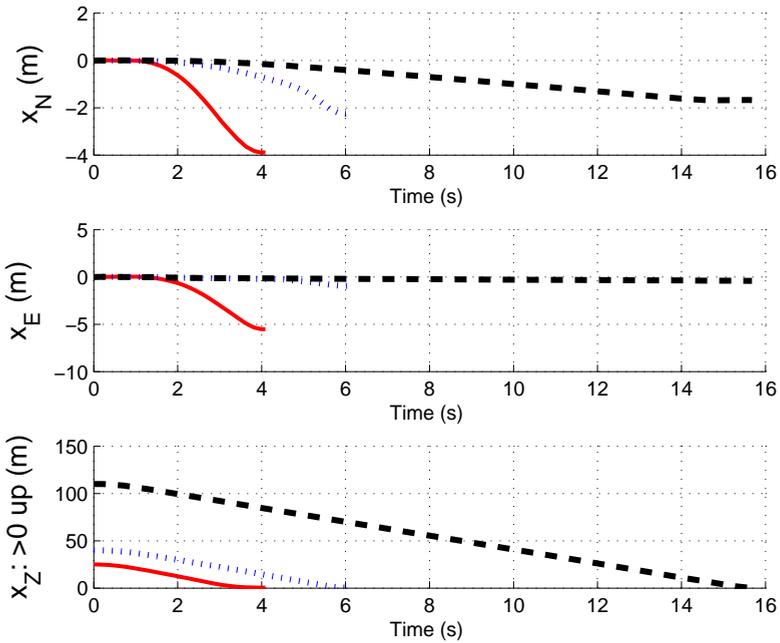
Figure 3.9: Inertial position: North position x_N, East position x_E, Vertical position x_Z (variation of initial altitude according to Table 3.5).
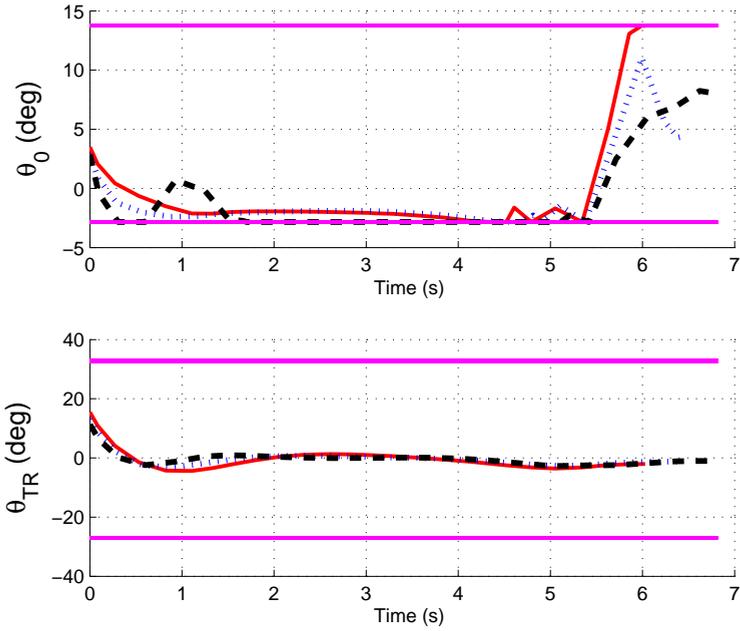
Figure 3.10: MR collective control input θ_0 and TR collective control input θ_{TR} (variation of initial airspeed according to Table 3.6).
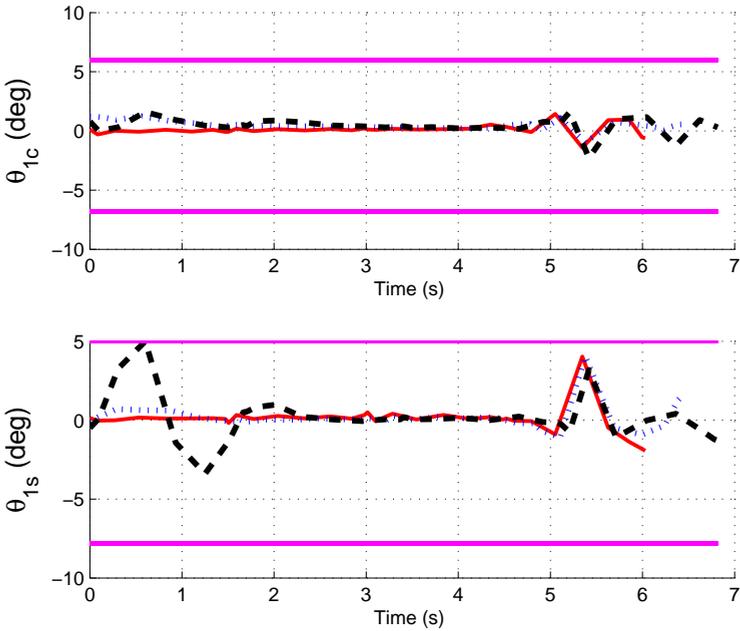
Figure 3.11: MR lateral cyclic control input θ_{1c} and MR longitudinal cyclic control input θ_{1s} (variation of initial airspeed according to Table 3.6).
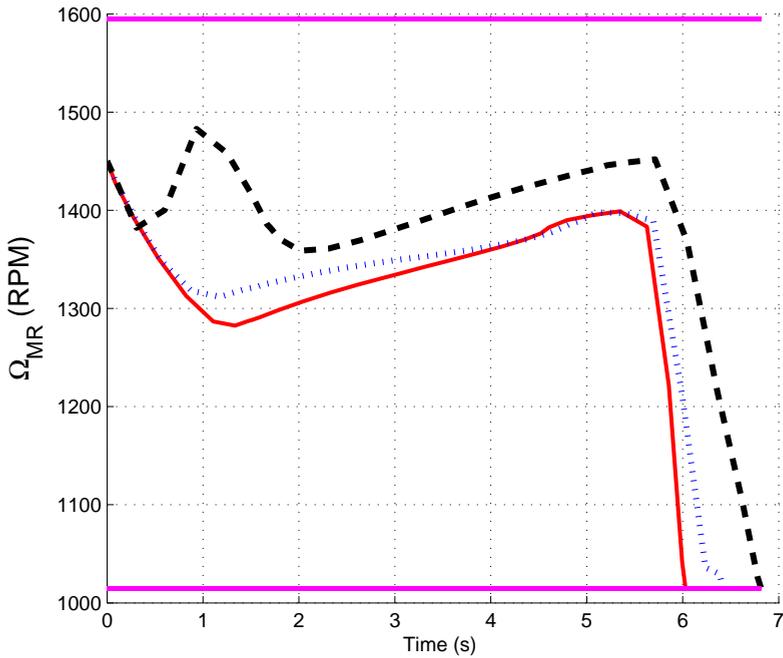
Figure 3.12: MR RPM Ω_{MR} (variation of initial airspeed according to Table 3.6).
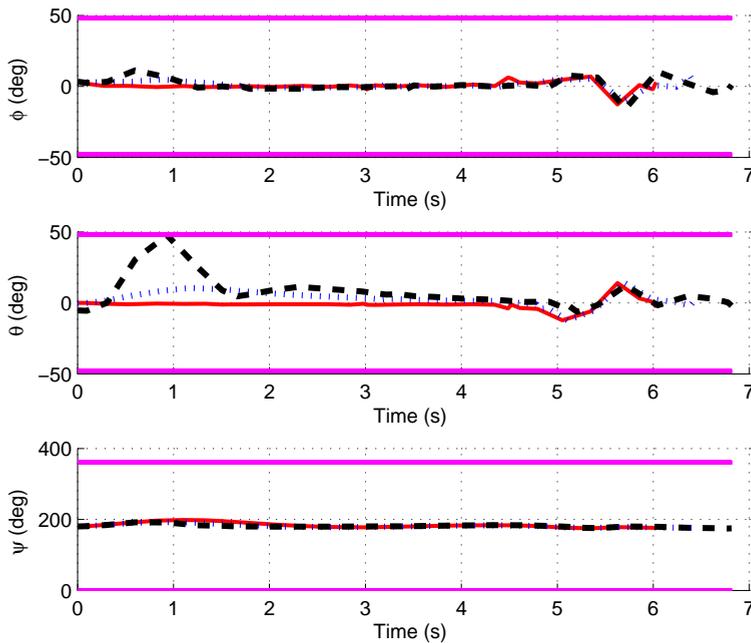
Figure 3.13: Euler angles: roll angle ϕ, pitch angle θ, yaw angle ψ (variation of initial airspeed according to Table 3.6).
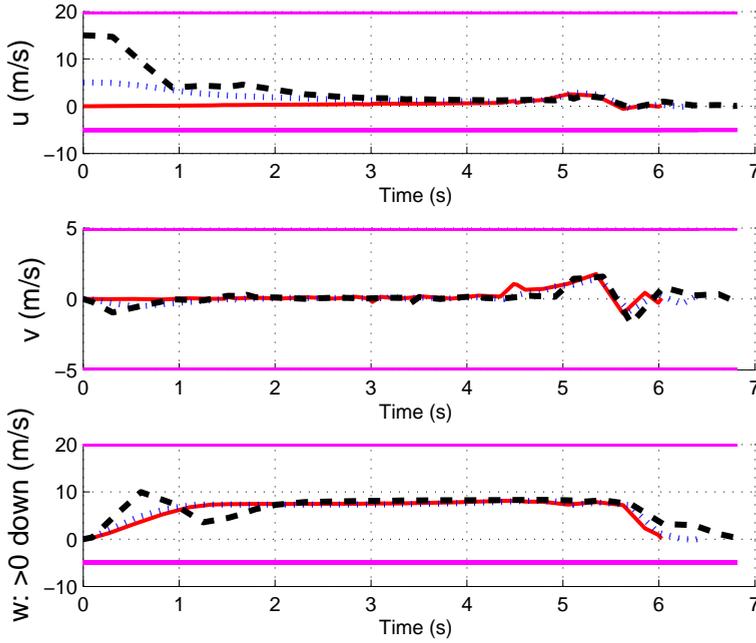
Figure 3.14: Body linear velocities: longitudinal velocity u, lateral velocity v, vertical velocity w (variation of initial airspeed according to Table 3.6).

3
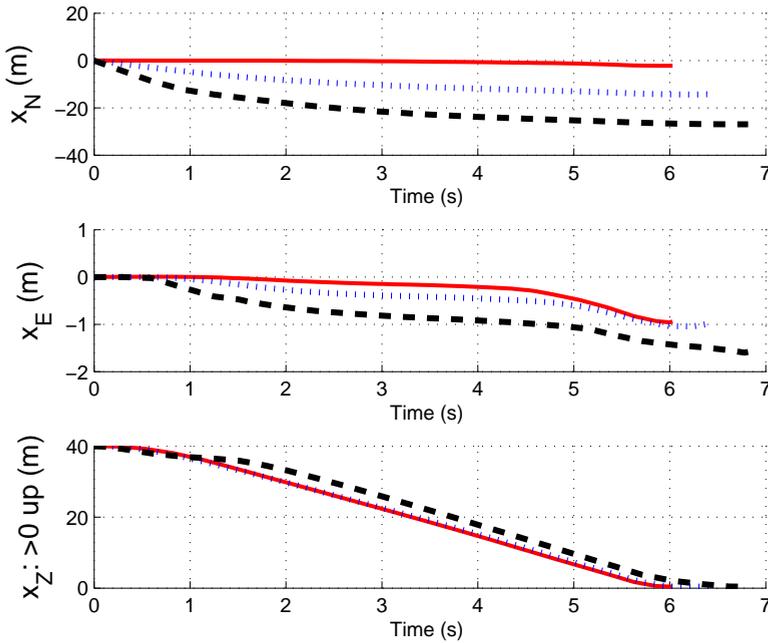
Figure 3.15: Inertial position: North position x_N, East position x_E, Vertical position x_Z (variation of initial airspeed according to Table 3.6).

REFERENCES

[1] S. Taamallah, *Low-order modeling for a small-scale flybarless helicopter uav, a grey-box time-domain approach,* in *AIAA Atmospheric Flight Mechanics Conf.* (2012).

[2] S. Taamallah, *Optimal autorotation with obstacle avoidance for a small-scale flybarless helicopter uav,* in *AIAA Guidance, Navigation and Control Conf.* (2012).

[3] S. Taamallah, X. Bombois, and P. M. J. Van den Hof, *Optimal control for power-off landing of a small-scale helicopter a pseudospectral approach,* in *Am. Control Conf.* (2012).

[4] J. T. Schwartz and M. Sharir, *A survey of motion planning and related geometric algorithms,* Artificial Intelligence **37**, 157 (1988).

[5] J. C. Latombe, *Robot Motion Planning* (Kluwer Int. Series in Engineering and Computer Science, Mass., 1991).

[6] Y. K. Hwang and N. Ahuja, *Gross motion planning - a survey,* ACM Comp. Surv. **24**, 219 (1992).

[7] J. C. Latombe, *Motion planning: A journey of robots, molecules, digital actors, and other artifacts,* Int. J. of Robotics Research **18**, 1119 (1999).

[8] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations* (MIT Press, Cambridge, MA, 2005).

[9] S. M. LaValle, *Planning Algorithms* (Cambridge University Press, Cambridge England, 2006).

[10] R. M. Murray, *Robust Control and nonholonomic motion planning*, Ph.D. thesis, University of California at Berkeley (1990).

[11] C. Goerzen, Z. Kong, and B. Mettler, *A survey of motion planning algorithms from the perspective of autonomous uav guidance,* J. Intell. Robot Syst. **57**, 65 (2010).

[12] N. Dadkhah and B. Mettler, *Survey of motion planning literature in the presence of uncertainty: Considerations for uav guidance,* J. Intell. Robot Syst. **65**, 233 (2012).

[13] R. Humphries, *The Aeneid of Virgil: A Verse Translation* (Macmillan, USA, 1987).

[14] A. E. Bryson, *Optimal control - 1950 to 1985,* IEEE Control Systems Magazine **16** (1996).

[15] H. J. Sussmann and J. C. Willems, *300 years of optimal control: From the brachystochrone to the maximum principle,* Control Systems, IEEE **17** (1997).

[16] R. Bellman and S. E. Dreyfus, *Applied Dynamic Programming* (Princeton Univ. Press, Princeton New Jersey, 1971).

3

[17] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, E. F. Mishchenko, K. N. Trirogoff, and L. W. Neustadt, *Mathematical Theory of Optimal Processes* (Interscience, NY, 1962).

[18] A. E. Bryson and Y. C. Ho, *Applied Optimal Control* (Taylor & Francis Group, New York, 1975).

[19] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York, 1988).

[20] D. A. Benson, *A Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. thesis, Massachusetts Institute of Technology (2004).

[21] G. T. Huntington, *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. thesis, Massachusetts Institute of Technology (2007).

[22] C. P. Neuman and A. Sen, *A suboptimal control algorithm for constrained problems using cubic splines,* Automatica **9**, 601 (1973).

[23] J. T. Betts, *Practical methods for optimal control using nonlinear programming,* in *SIAM* (2001).

[24] K. Misovec, T. Inanc, J. Wohletz, and R. M. Murray, *Low-observable nonlinear trajectory generation for unmanned air vehicles,* in *IEEE Conf. on Decision and Control* (2003).

[25] D. Gottlieb, M. Y. Hussaini, and S. A. Orszag, *Theory and applications of spectral methods,* in *Spectral Methods for PDE's. Soc. of Industrial and Applied Mathematics (SIAM)* (1984).

[26] P. Lu, H. Sun, and B. Tsai, *Closed-loop endoatmospheric ascent guidance,* AIAA J. of Guidance, Control and Dynamics **26**, 283 (2003).

[27] J. Rea, *Launch vehicle trajectory optimization using a legendre pseudospectral method,* in *AIAA Guidance, Navigation, and Control Conf.* (2003).

[28] S. Josselyn and I. M. Ross, *Rapid verification method for the trajectory optimization of reentry vehicles,* AIAA J. of Guidance, Control, and Dynamics **26** (2003).

[29] P. Williams, *Application of pseudospectral methods for receding horizon control,* AIAA J. of Guidance, Control, and Dynamics **27**, 310 (2004).

[30] S. I. Infeld, *Optimization of Mission Design for Constrained Libration Point Space Missions*, Ph.D. thesis, Stanford University (2005).

[31] J. P. Riehl, S. W. Paris, and W. K. Sjaw, *Comparision of implicit integration methods for solving aerospace trajectory optimization problems,* in *AIAA/AAS Astrodynamics Specialists Conf.* (2006).

[32] A. M. Hawkins, T. R. Fill, R. J. Proulx, and E. M. Feron, *Constrained trajectory optimization for lunar landing,* in *AAS Spaceflight Mechanics Meeting* (2006).

3

[33] K. P. Bollino, L. R. Lewis, P. Sekhavat, and I. M. Ross, *Pseudospectral optimal control: A clear road for autonomous intelligent path planning,* in *AIAA Infotech* (2007).

[34] N. Bedrossian, S. Bhatt, M. Lammers, L. Nguyen, and Y. Zhang, *First ever flight demonstration of zero propellant maneuver tm attitude control concept,* in *AIAA Guidance, Navigation, and Control Conf.* (2007).

[35] Q. Gong, K. Wei, N. S. Bedrossian, F. Fahroo, P. Sekhavat, and K. Bollino, *Pseudospectral optimal control for military and industrial applications,* in *IEEE Conf. on Decision and Control* (2007).

[36] I. M. Ross, P. Sekhavat, A. Fleming, and Q. Gong, *Optimal feedback control: Foundations, examples, and experimental results for a new approach,* AIAA J. of Guidance, Control, and Dynamics **31** (2008).

[37] M. Karpenko, S. Bhatt, N. Bedrossian, A. Fleming, and I. M. Ross, *Flight implementation of pseudospectral optimal control for the trace space telescope,* in *AIAA Guidance, Navigation and Control Conf.* (2011).

[38] M. Ceriotti and C. R. McInnes, *Generation of optimal trajectories for earth hybrid pole sitters,* AIAA J. of Guidance, Control, and Dynamics **34**, 847 (2011).

[39] H. Yan, Q. Gong, C. D. Park, I. M. Ross, and C. N. D. Souza, *High-accuracy trajectory optimization for a trans-earth lunar mission,* AIAA J. of Guidance, Control, and Dynamics **34**, 1219 (2011).

[40] C. L. Darby and A. V. Rao, *Minimum-fuel low-earth-orbit aeroassisted orbital transfer of small spacecraft,* AIAA J. of Spacecraft and Rockets **48** (2011).

[41] G. Boyarko, O. Yakimenko, and M. Romano, *Optimal rendezvous trajectories of a controlled spacecraft and a tumbling object,* AIAA J. of Guidance, Control, and Dynamics **34**, 1239 (2011).

[42] G. Boyarko, M. Romano, and O. Yakimenko, *Time-optimal reorientation of a spacecraft using an inverse dynamics optimization method,* AIAA J. of Guidance, Control, and Dynamics **34**, 1197 (2011).

[43] K. Zhang and W. Chen, *Reentry vehicle constrained trajectory optimization,* in *AIAA Int. Space Planes and Hypersonic Systems and Technologies* (2011).

[44] S. W. Paris, J. P. Riehl, and W. K. Sjaw, *Enhanced procedures for direct trajectory optimization using nonlinear programming and implicit integration,* in *AIAA/AAS Astrodynamics Specialists Conf.* (2006).

[45] G. C. Cottrill and F. G. Harmon, *Hybrid gauss pseudospectral and generalized polynomial chaos algorithm to solve stochastic trajectory optimization problems,* in *AIAA Guidance Navigation and Control Conf.* (2011).

[46] M. Dekel and J. Z. Ben-Asher, *Pseudo-spectral-method based optimal glide in the event of engine cut-off,* in *AIAA Guidance, Navigation, and Control Conf.* (2011).

[47] R. Dai and J. E. Cochran, *Three-dimensional trajectory optimization in constrained airspace,* AIAA J. of Aircraft **46**, 627 (2009).

[48] S. Hartjes, H. G. Visser, and M. D. Pavel, *Environmental optimization of rotorcraft approach trajectories,* in *Europ. Rotorcraft Forum* (2011).

[49] M. Harada and K. Bollino, *Optimal trajectory of a glider in ground effect and wind shear,* in *AIAA Guidance, Navigation and Control Conf.* (2005).

[50] Q. Gong, L. R. Lewis, and I. M. Ross, *Pseudospectral motion planning for autonomous vehicles,* AIAA J. of Guidance, Control, and Dynamics **32** (2009).

[51] B. D. McCracken, K. L. Burzota, M. M. Ramos, and T. R. Jorris, *Cooperative unmanned air vehicle mission planning and collision avoidance,* in *AIAA Infotech* (2011).

[52] R. Choe and N. Hovakimyan, *Perching maneuver for an mav augmented with an l1 adaptive controller,* in *AIAA Guidance, Navigation, and Control Conf.* (2011).

[53] A. la Cour-Harbo and M. Bisgaard, *State-control trajectory generation for helicopter slung load system using optimal control,* in *AIAA Guidance, Navigation, and Control Conf.* (2009).

[54] M. Harada and K. Bollino, *Minimum-time circling flight of a triarm coaxial rotor uav,* in *AIAA Guidance Navigation and Control Conf.* (2011).

[55] D. Bertsekas, *Nonlinear Programming - Second Edition* (Athena Scientific, Belmont, MA, 1999).

[56] L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders, *Large-Scale PDE-Constrained Optimization* (Springer, Berlin, 2003).

[57] R. H. Byrd, J. Nocedal, and R. A. Waltz, *Knitro: An integrated package for nonlinear optimization,* Large-Scale Nonlinear Optimization, Springer-Verlag , 35 (2006).

[58] A. Wachter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,* Mathematical Programming **106**, 25 (2006).

[59] L. T. Biegler and V. M. Zavala, *Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide optimization,* Computers and Chemical Engineering **33**, 575 (2008).

[60] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's Guide for NPSOL 5.0: A Fortran Package for Nonlinear Programming*, Tech. Rep. (Stanford University, 2001).

3

[61] P. E. Gill, W. Murray, and M. A. Saunders, *Snopt: An sqp algorithm for large-scale constrained optimization,* SIAM Reviews **47**, 99 (2002).

[62] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*, Tech. Rep. (University of California San Diego, 2008).

[63] SBS-Inc., (http://www.sbsi-sol-optimize.com/asp/sol_product_snopt.htm, Stanford University, U.S.A.).

[64] G. A. Bliss, *Lectures on the Calculus of Variations* (University of Chicago Press, Chicago IL, 1946).

[65] R. Weinstock, *Calculus of Variations* (Dover Publications, NY, 1974).

[66] B. van Brunt, *The Calculus of Variations* (Springer-Verlag, NY, 2004).

[67] M. D. Ardema and N. Rajan, *Separation of time scales in aircraft trajectory optimization,* AIAA J. of Guidance, Control, and Dynamics **8**, 275 (1985).

[68] C. L. Bottasso, A. Croce, D. Leonello, and L. Riviello, *Optimization of critical trajectories for rotorcraft vehicles,* J. of the Am. Helicopter Soc. , 165 (2005).

[69] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, *An application of reinforcement learning to aerobatic helicopter flight,* in *NIPS 19* (2007).

[70] P. Abbeel, A. Coates, T. Hunter, and A. Y. Ng, *Autonomous autorotation of an rc helicopter,* in *11th Int. Symposium on Experimental Robotics (ISER)* (2008).

[71] D. J. Lee, H. Bang, and K. Baek, *Autorotation of an unmanned helicopter by a reinforcement learning algorithm,* in *AIAA Guidance, Navigation and Control Conf.* (2008).

[72] D. J. Lee and H. Bang, *Autonomous autorotation of an unmanned helicopter using a reinforcement learning algorithm,* J. Aerospace Information Systems **10**, 98 (2013).

[73] M. Uemura, M. Ishikawa, N. Sudo, I. Sudo, and Y. kubo, *Fuzzy intelligent pilot support system for an autorotative flight of helicopter,* in *Europ. Rotorcraft Forum* (1993).

[74] Z. Sunberg and J. Rogers, *A fuzzy logic-based controller for helicopter autorotation,* in *AIAA Aerospace Sciences Meeting and Exhibit* (2013).

[75] M. Komoda, *Prediction of height-velocity boundaries for rotorcraft by application of optimizing techniques,* Trans. of the Japan Soc. for Aeronautical and Space Sciences **15**, 208 (1973).

[76] W. Johnson, *Helicopter Optimal Descent and Landing After Power Loss*, Tech. Rep. TM 73-244 (NASA Ames Research Center, 1977).

[77] A. Y. Lee, A. E. Bryson, and W. S. Hindson, *Optimal landing of a helicopter in autorotation,* AIAA J. of Guidance, Control, and Dynamics **11**, 7 (1988).

[78] Y. Okuno, K. Kawachi, A. Azuma, and S. Saito, *Analytical prediction of height-velocity diagram of a helicopter using optimal control theory,* AIAA J. of Guidance, Control, and Dynamics **14**, 453 (1991).

[79] Y. Okuno and K. Kawachi, *Optimal control of helicopters following power failure,* AIAA J. of Guidance, Control, and Dynamics **17**, 181 (1994).

[80] M. W. Floros, *Descent analysis for rotorcraft survivability with power loss,* in *Am. Helicopter Soc.* (2009).

[81] A. Lee, *Engineering notes - optimal autorotational descent of a helicopter with control and state inequality constraints,* AIAA J. of Guidance, Control, and Dynamics **13**, 922 (1990).

[82] E. N. Bachelder and B. L. Aponso, *An autorotation flight director for helicopter training,* in *Am. Helicopter Soc.* (2003).

[83] A. A. Jhemi, E. B. Carlson, Y. J. Zhao, and R. T. N. Chen, *Optimization of rotorcraft flight following engine failure,* J. of the Am. Helicopter Soc. (2004).

[84] B. L. Aponso, E. N. Bachelder, and D. Lee, *Automated autorotation for unmanned rotorcraft recovery,* in *AHS Int. Specialists' Meeting On Unmanned Rotorcraft* (2005).

[85] B. L. Aponso, D. Lee, and E. N. Bachelder, *Evaluation of a rotorcraft autorotation training display on a commercial flight training device,* in *Am. Helicopter Soc.* (2005).

[86] C. L. Bottasso, C. S. Chang, A. Croce, D. Leonello, and L. Riviello, *Adaptive planning and tracking of trajectories for the simulation of maneuvers with multibody models,* Computer Methods in Applied Mechanics and Engineering (2006).

[87] E. B. Carlson, S. Xue, J. Keane, and K. Burns, *H-1 upgrades height-velocity diagram development through flight test and trajectory optimization,* in *Am. Helicopter Soc.* (2006).

[88] P. Bibik and J. Narkiewicz, *Helicopter optimal control after power failure using comprehensive dynamic model,* AIAA J. of Guidance, Control, and Dynamics **35**, 1354 (2012).

[89] C. L. Bottasso, G. Maisano, and F. Luraghi, *Efficient rotorcraft trajectory optimization using comprehensive vehicle models by improved shooting methods,* in *Europ. Rotorcraft Forum* (2009).

[90] C. L. Bottasso, G. Maisano, and F. Scorcelletti, *Trajectory optimization procedures for rotorcraft vehicles, their software implementation, and applicability to models of increasing complexity,* J. of the Am. Helicopter Soc. (2010).

3

[91] M. W. Floros, *Application of sequential quadratic programming to rotorcraft surviv- ability with power loss,* in *Am. Helicopter Soc.* (2011).

[92] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl, *Autonomous autorotation of unmanned rotorcraft using nonlinear model predictive control,* J. Intell. Robot Syst. **57**, 351 (2010).

[93] S. Tierney and J. W. Langelaan, *Autorotation path planning using backwards reach- able set and optimal control,* in *Am. Helicopter Soc.* (2010).

[94] N. Grande and J. W. Langelaan, *Safe autonomous flare and landing during autoro- tation through wind shear,* in *Am. Helicopter Soc.* (2013).

[95] J. Holsten, S. Loechelt, and W. Alles, *Autonomous autorotation flights of helicopter uavs to known landing sites,* in *Am. Helicopter Soc.* (2010).

[96] T. Yomchinda, J. F. Horn, and J. W. Langelaan, *Flight path planning for descent- phase helicopter autorotation,* in *AIAA Guidance, Navigation, and Control Conf.* (2011).

[97] T. Yomchinda, J. F. Horn, and J. W. Langelaan, *Autonomous control and path plan- ning for autorotation of unmanned helicopters,* in *Am. Helicopter Soc.* (2012).

[98] P. Bibik and J. Narkiewicz, *Helicopter modeling and optimal control in autorotation,* in *Am. Helicopter Soc.* (2008).

[99] R. W. Prouty, *Helicopter Performance, Stability, and Control* (Krieger Publishing Company, Malabar, Florida USA, 1995).

[100] W. Johnson, *Helicopter Theory* (Dover Publications Inc., NY, USA, 1994).

[101] S. Taamallah, *A qualitative introduction to the vortex-ring-state, autorotation, and optimal autorotation,* in *Europ. Rotorcraft Forum* (2010).

[102] J. Nocedal and S. Wright, *Numerical Optimization* (Springer, NY, 2000).

[103] R. J. Vanderbei and D. F. Shanno, *An interior-point algorithm for nonconvex nonlin- ear programming,* Computational Optimization and Applications **13**, 231 (1999).

[104] H. B. Keller, *Numerical solution of two point boundary value problems,* SIAM (1976).

[105] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis* (Springer-Verlag, 2002).

[106] D. Hull, *Conversion of optimal control problems into parameter optimization prob- lems,* AIAA J. of Guidance, Control, and Dynamics **20**, 57 (1997).

[107] L. T. Biegler, *An overview of simultaneous strategies for dynamic optimization,* Chemical Engineering and Processing: Process Intensification **46**, 1043 (2007).

[108] C. J. Kim, S. K. Sung, S. H. Park, S. N. Jung, and K. Yee, *Selection of rotorcraft models for application to optimal control problems,* AIAA J. of Guidance, Control, and Dynamics **31** (2008).

[109] J. T. Betts, *Survey of numerical methods for trajectory optimization,* AIAA J. of Guidance, Control, and Dynamics **21**, 193 (1998).

[110] C. L. Bottasso, G. Maisano, and F. Scorcelletti, *Trajectory optimization procedures for rotorcraft vehicles including pilot models, with applications to ads-33 mtes, cat-a and engine off landings,* in *Am. Helicopter Soc.* (2009).

[111] A. L. Dontchev, W. W. Hager, and V. M. Veliov, *Uniform convergence and mesh independence of newton's method for discretized variational problems,* SIAM J. on Control and Optimization **39**, 961 (2000).

[112] A. L. Herman and B. A. Conway, *Direct optimization using collocation based on high-order gauss-lobatto quadrature rules,* AIAA J. of Guidance, Control, and Dynamics **19**, 592 (1996).

[113] F. Fahroo and I. M. Ross, *Direct trajectory optimization by a chebyshev pseudospectral method,* J. of Guidance, Control, and Dynamics **25** (2002).

[114] D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao, *Direct trajectory optimization and costate estimation via an orthogonal collocation method,* AIAA J. of Guidance, Control, and Dynamics **29** (2006).

[115] F. Fahroo and I. M. Ross, *Pseudospectral methods for infinite horizon nonlinear optimal control problems,* in *AIAA Guidance, Navigation and Control Conf.* (2005).

[116] D. Garg, M. A. Patterson, C. L. Darby, C. Francolin, G. T. Huntington, W. W. Hager, and A. V. Rao, *Direct trajectory optimization and costate estimation of general optimal control problems using a radau pseudospectral method,* in *AIAA Guidance Navigation and Control Conf.* (2009).

[117] C. L. Darby, W. W. Hager, and A. V. Rao, *Direct trajectory optimization using a variable low-order adaptive pseudospectral method,* AIAA J. of Spacecraft and Rockets **48** (2011).

[118] A. V. Rao, *A survey of numerical methods for optimal control,* in *Am. Astronautical Soc.* (2009).

[119] F. Fahroo and I. M. Ross, *Second look at approximating differential inclusions,* AIAA J. of Guidance, Control, and Dynamics **24** (2001).

[120] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration* (Dover Publications, 2007).

[121] A. V. Rao, D. A. Benson, C. Darby, M. A. Patterson, C. Francolin, I. Sanders, and G. T. Huntington, *Algorithm xxx: GPOPS, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method*, Tech. Rep. (ACM Inc., 2008).

3

[122] I. M. Ross and F. Fahroo, *A direct method for solving nonsmooth optimal control problems,* in *IFAC 15th Triennial World Congress* (2002).

[123] R. J. Pegg, *An Investigation of the Helicopter Height-Velocity Diagram Showing Effects of Density Altitude and Gross Weight*, Tech. Rep. TN D-4536 (NASA Langley Research Center, 1968).

[124] E. F. Katzenberger and M. J. Rich, *An investigation of helicopter descent and landing characteristics following power failure,* J. of Aeron Sci **23**, 345 (1956).

[125] W. D. Jepson, *Some considerations of the landing and take-off characteristics of twin engine helicopters. part i - height-velocity and part power descents,* J. of the Am. Helicopter Soc. **7**, 33 (1962).

[126] W. J. Hanley and G. DeVore, *An Analysis of the Helicopter Height-Velocity Diagram Including a Practical Method for Its Determination*, Tech. Rep. ADS 67-23 (FAA, 1967).

[127] J. M. Besse, *The unsafe zone for single engine helicopters,* in *Europ. Rotorcraft Forum* (1978).

[128] T. L. Wood, *High energy rotor systems,* in *Am. Helicopter Soc.* (1976).

[129] L. W. Dooley and R. D. Yeary, *Flight Test Evaluation of the High Inertia Rotor System*, Tech. Rep. UARTL-TR-79-9 (U.S. Army Research and Technology Laboratories (AVRADCOM), 1979).

[130] M. K. Szymanowski, *Improving helicopter height-velocity performance,* in *Europ. Rotorcraft Forum* (1999).

[131] J. Schillings, J. McCollough, P. Hollifield, J. Wasylyszyn, and R. Grife, *Height-velocity testing of a growth rotor for the bell 407 aircraft,* in *Am. Helicopter Soc.* (2007).

[132] D. Santamaria, A. Viguria, M. Bejar, K. Kondak, and A. Ollero, *Towards autonomous autorotation landing for small size unmanned helicopters,* J. Intell. Robot Syst. **69**, 171 (2013).

[133] A. A. G. Macintosh, *Trajectory Tracking Control for a Small-Scale Helicopter during Autorotation*, Tech. Rep. (Master Thesis, Delft University of Technology, 2014).

4

ON-LINE TRAJECTORY PLANNING AND TRACKING: SYSTEM DESIGN

Linear systems are important because we can solve them and because the fundamental laws of physics are often linear, e.g., Maxwell's equations for electricity, the laws of quantum mechanics, and the approximations when displacements are small.

Richard P. Feynman
The Feynman Lectures on Physics, Addison-Wesley, 1963

The design of high-performance guidance and control systems for small-scale helicopter Unmanned Aerial Vehicles (UAVs) is known to be a challenging task. In Chapter 3, we presented a Trajectory Planning (TP) approach, for the engine OFF condition (i.e. autorotation), for off-line use. The purpose of Chapter 4 is to present a combined TP and Trajectory Tracking (TT) system, for the engine OFF condition, having on-line computational tractability. The presented system is anchored within the aggregated paradigms of differential flatness based optimal planning, and robust control based trajectory tracking. A similar flight control system, for the engine ON condition, is also provided in the Appendices.

Parts of this Chapter have been published in [1–3].

4.1. Introduction

CHAPTER 3 used an off-line approach to compute open-loop, optimal, autorotative trajectories. In Chapter 4, we compute these autorotative trajectories through an on-line approach. In addition, the purpose of Chapter 4 is to present and describe the design of a guidance and control logic, that enables a small-scale unmanned helicopter to execute a completely automatic landing maneuver, for an engine OFF (i.e. autorotation [4, 5]) flight condition[1]. The guidance module, or Trajectory Planning (TP), shall be capable of generating optimal trajectories, on-line, while effectively exploiting the rigid-body nonlinear dynamics. On the other hand, the control module, or Trajectory Tracking (TT), shall have the duty to ensure that the helicopter flies along these optimal trajectories. In Chapter 5, this complete Flight Control System (FCS) will be evaluated on the high-fidelity helicopter simulation model, developed in Chapter 2, for the engine OFF and ON conditions.

A full review of previous contributions, for the engine OFF TP and TT (respectively engine ON TP and TT), has already been presented in Sections 1.5.2 and 1.5.3 of Chapter 1, and in Section 3.2.1, of Chapter 3. Most notable is that very few papers, i.e. [6–9], have addressed the aggregated planning and tracking functionalities, for the engine OFF case, with validation through either experiments or 3D high-fidelity nonlinear simulations. The authors in [8, 9] apply their FCS to the case of a full-size helicopter, whereas the application in [7] involves a so-called short-range/tactical size helicopter UAV (approximately 200 kg). Only the results in [6] are for a small-scale helicopter UAV. As outlined in Chapter 1, when compared to larger and heavier helicopter vehicles, the control of small-scale helicopters (i.e. under 10–20 kg) represents a much more challenging problem.

In this Chapter we choose to base our TP on the concept of differential flatness. This approach allows to exploit the rigid-body nonlinear dynamics, while retaining a high computational efficiency, e.g. for on-line use in a hard real-time environment where stringent timing constraints may need to be met (especially for high-bandwidth systems). Compared to the off-line TP of Chapter 3, the advantage of the TP module presented in this Chapter, is its on-line computational tractability. The seminal ideas of differential flatness were introduced in the early 1990s in [10–12] as part of a paradigm in which certain differential algebraic representations of dynamical systems are equivalent. In other words, a complete parametrization of all system variables—inputs, states, and outputs—may be given in terms of a finite set of independent variables, called flat outputs, and a finite number of their derivatives [13, 14]. This results in optimization problems with fewer variables [15], i.e. by the complete elimination of the dynamical constraints. In this case the trajectory generation problem is transformed from a dynamic to an algebraic one, in which the flat outputs are parametrized over a space of basis functions, and where the generation of feasible trajectories is reduced to a classical algebraic interpolation or collocation problem [16, 17].

Since the helicopter dynamics is nonlinear, the design of the TT controller shall necessitate an approach that effectively respects or tries to exploit the system's nonlinear structure. To this end, several control methods are available: from 1) robust control; 2) classical gain-

---

[1]In the Appendices of this Chapter we present a guidance and control logic that allows to execute a variety of engine ON automatic maneuvers, e.g. take-off, landing, and cruise.

scheduling, and Linear Parameter-Varying (LPV) approaches; to 3) truly nonlinear control methods (e.g. nonlinear MPC, Lyapunov based methods such as sliding mode and back-stepping, adaptive control, or even passivity-based approaches). In this thesis we select an approach that combines both simplicity and computational tractability, namely a robust control μ strategy. The selected strategy consists in using a single, nominal, low-order, Linear Time-Invariant (LTI) plant, coupled with an input multiplicative uncertainty, and applying a small gain approach [18, 19] to design a single robust LTI controller. The uncertainty is added here to compensate for the unmodeled plant nonlinearities and unmodeled higher-order rotor dynamics[2].

Finally, the nomenclature is fairly standard. For appropriately dimensioned matrices K and M, where the latter is partitioned as $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$, the lower Linear Fractional Transformation (LFT) is defined as $F_l(M, K) = M_{11} + M_{12}K(I - M_{22}K)^{-1}M_{21}$, and the upper LFT is defined as $F_u(M, K) = M_{22} + M_{21}K(I - M_{11}K)^{-1}M_{12}$ under the assumption that the inverses exist. For $M \in \mathbb{C}^{q \times p}$, the structured singular value $\mu_\Delta(M)$ of M, with respect to an uncertainty set $\Delta \subset \mathbb{C}^{p \times q}$, is defined as $\mu_\Delta^{-1}(M) \coloneqq \min_{\Delta \in \Delta}\{\bar{\sigma}(\Delta) \mid \det(I - M\Delta) = 0\}$.

4.1.1. MAIN CONTRIBUTIONS

The novelty of this Chapter can be stated as follows.

- First, we design the first, real-time feasible, model-based TP and TT system, for the case of a small-scale helicopter UAV with an engine OFF condition. Indeed, the results in [6–9] are based upon a model-free TP. Our flatness planning approach effectively exploits the rigid-body nonlinear dynamics, thus computing trajectory solutions which are feasible and optimal.

- Second, with regard to the TT, the method in [9] is based upon a model-free fuzzy logic approach. The method in [6] uses a model-based Differential Dynamic Programming (DDP)[3] approach. The method in [8] uses a model-based combined Non-linear Dynamic Inversion (NDI) with Proportional Integral Derivative (PID) loops, whereas the method in [7] uses a model-based H_∞ approach. For the three model-based approaches, the TT controllers are synthesized on a single nominal model, that does not include uncertainties, whereas our TT controller is synthesized on the basis of a nominal model, coupled with additional uncertainties, in order to enhance the robustness properties of the closed-loop system.

The remainder of this Chapter is organized as follows. In Section 4.2, the two-degree of freedom control architecture, as implemented in this Chapter, is first reviewed. In Section 4.3, the flatness-based trajectory planning is described. In Section 4.4, the main aspects of the robust control approach are reviewed and discussed. In Section 4.5 and Section 4.6, the synthesis of the inner- and outer-loop controllers, for the engine OFF case, are presented.

---

[2]Unmodeled in the low-order nominal LTI plant used for control design, these are however modeled in the high-order nonlinear plant of Chapter 2.

[3]DDP is an extension of the Linear Quadratic Regulator (LQR) formalism for non-linear systems [20].

Conclusions and future directions are presented in Section 4.7. Finally, the first three Appendices present the trajectory planning and tracking system for the engine ON case (using an architecture which is identical to the one developed for the engine OFF case).

4.2. GENERAL CONTROL ARCHITECTURE

We present here the conceptual FCS design solution, chosen to solve the helicopter UAV guidance and control problem. We make use of the classical two-degree of freedom controller design paradigm, in which the philosophy decouples the guidance module from the control module, see Chapter 1. The guidance module, or TP, shall be capable of generating open-loop, feasible and optimal (autorotative) trajectory references $\mathbf{x_{TP}}$, for the small-scale helicopter, subject to system and environmental constraints, see Fig. 4.1. This TP computes open-loop optimal trajectories, given a cost objective, system dynamics, and controls and states equality and inequality constraints. These optimal trajectories may be computed off-line[4], through the use of nonlinear optimal control methods such as in Chapter 3, or alternatively, such as in this Chapter, may be computed on-line using the concept of differential flatness. Compared to the architecture outlined in Fig. 1.15 of Chapter 1, the TP of Chapter 4 does not generate any feedforward nominal control inputs, nor is there any additional feedback path into the TP.

On the other hand the control module, or Trajectory Tracking (TT), compares current measured values $\mathbf{y}$, i.e. a subset of the vehicle states $\mathbf{x}$, with the reference values $\mathbf{x_{TP}}$ produced by the TP, and formulates the feedback controls $\mathbf{u}$ aimed at decreasing this tracking error[5]. This latter may be due to a combination of model uncertainty (unmodeled higher-order dynamics, unmodeled static nonlinearities, parametric uncertainties, delays), and signal uncertainty (wind disturbances and noise). In Fig. 4.1, the 'Helicopter Dynamics Non-Linear Simulation' block refers to the high-fidelity, nonlinear, High-Order Model (HOM), simulation of Chapter 2, serving as a proxy for the real helicopter system.
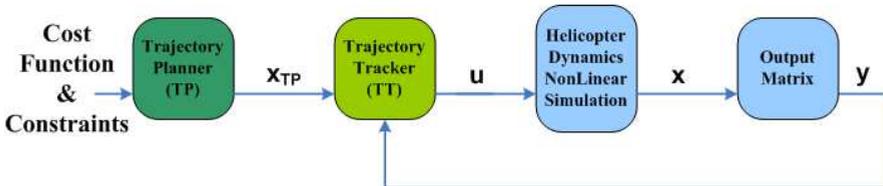
Figure 4.1: Two-Degree of freedom control architecture.

4.3. FLATNESS-BASED TRAJECTORY PLANNING (TP)

The seminal ideas of differential flatness were introduced in the early 1990s in [10–12] as part of a paradigm in which certain differential algebraic representations of dynamical sys-

---

[4]The trajectories are stored as lookup tables, on-board a flight control computer.

[5]The nomenclature, given in Appendix A of Chapter 2, states that all vectors are printed in boldface, hence the control input vector $\mathbf{u}$ should not be confused with the body longitudinal velocity u.

tems are equivalent. Flatness can be seen as a a subclass of the set of controllable nonlinear systems [21], or as a system's geometric property [16] independent of coordinate choice, or as a Lie-Bäcklund equivalence property [14, 22], in which a complete parametrization of all system variables—inputs, states, and outputs—may be given in terms of a finite set of independent variables, called flat outputs, and a finite number of their derivatives [13, 14].

Flatness comes with two important benefits. First, it offers a particularly well adapted framework for solving inverse dynamics problems [16, 23]. Indeed, flatness implies the absence of so-called zero dynamics, allowing for a one-to-one correspondence between trajectories of the input-state system and trajectories of the flat output (in which case the nonlinear system can be feedback linearized using endogenous dynamic feedback [22]). This allows the trajectory generation and tracking for non-minimum phase systems by exact linearization [24, 25]. Second, and perhaps more importantly, flat parameterizations result in optimization problems with fewer variables [15], i.e. by the complete elimination of the dynamical constraints. In this case, a trajectory generation problem is transformed from a dynamic to an algebraic one, in which the flat outputs are parametrized over a space of basis functions, for which the generation of feasible trajectories is reduced to a classical algebraic interpolation or collocation problem [16, 17]. This allows, in principle, for significant computational benefits[6]. Seminal application of flatness towards trajectory planning can be found in [12, 28] for the case where the motion is not subject to inequality constraints, and in [29–31] for the case where inequality constraints have been added.

It is in general difficult to determine whether a given nonlinear system is flat, although several methods for constructing flat outputs have been documented in the literature [13, 32–34]. As an example, it is known that a system's Huygens center of oscillations may qualify as a flat output [11, 24, 25]. Additional rules, to find such flat outputs, include the following: 1) all linear systems are flat; 2) all nonlinear systems which are static and dynamic feedback linearizable are flat; 3) fully actuated systems are flat; and 4) finally under-actuated systems may or may not be flat. With regard to applications, it was shown that simplified dynamics of aircraft and Vertical Take-Off and Landing (VTOL) aircraft are flat [23, 35–39], simplified helicopter dynamics is flat [13, 40, 41], simplified quadrotor dynamics is flat [42–46], simplified planetary lander dynamics is flat [47], and simplified reentry vehicle dynamics is also flat [48], whereas more realistic vehicle models are in general non-differentially flat, e.g. [13, 21] for the helicopter case.

Since high-fidelity helicopter models are known to be non-differentially flat, a standard approach in the literature, to circumvent this difficulty, has consisted in progressively simplifying these models until they become flat. The drawback is that the domain of validity, of these simplified representations of the high-order helicopter dynamics, becomes questionable. Hence, rather than generating optimal trajectories based upon such questionable models, we choose here an alternative approach, consisting in using only the rigid-body

---

[6]Note that, in the presence of constraints, flatness parameterization implies a path constraint on the flat outputs, resulting from complex transformations of the control and/or state regions. These transformations may lead to a loss of convexity, which may be detrimental to real-time optimal control computations [15, 26, 27]. However, it is our experience that for complex, high-order, highly nonlinear plants, the benefits from the elimination of the dynamical constraints outweigh the disadvantages due to path constraints on the flat outputs.

dynamics[7] as the model for the TP, with total aerodynamic forces and total moments as the plant inputs (rather than the vehicle control inputs). Obviously, this corresponds also to a simplification of the helicopter HOM of Chapter 2, since we are replacing the HOM with the low-order rigid-body dynamics. However, if the bandwidth of the control inputs is kept low, then replacing the helicopter HOM with only the rigid-body dynamics becomes acceptable for planning purposes. The main drawback of using the rigid-body dynamics, as a substitute for the helicopter HOM, comes from losing the relationship between the total aerodynamic forces/moments and the vehicle control inputs. In our case, this should not represent a major drawback since, as stated in Section 4.2, the TP module does not feedforward the control inputs. On the other hand, the advantage of using the rigid-body dynamics (as the TP model) is that it can be shown to be exactly flat.

We recall next the ideas of differential flatness in conceptual form [10–12]. We suppose here that a plant's nonlinear model, derived from first-principles, is available and given by

$$\forall t \geq 0 \qquad \dot{\mathbf{x}}(t) = \tilde{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{4.1}$$

with $\tilde{f}(\cdot)$ a continuous-time, partially differentiable (sufficiently) smooth function, with $\mathbf{x}(t) \in \mathcal{P}_x \subset \mathbb{R}^{n_x}$ the plant state, $\mathbf{u}(t) \in \mathcal{P}_u \subset \mathbb{R}^{n_u}$ the control input, t the time variable, and $(\mathcal{P}_x, \mathcal{P}_u)$ some compact sets. We give next the following definition from [14].

Definition 1 *The system given by Eq. (4.1) is differentially flat if there exists a flat output* $\mathbf{z}(t) \in \mathcal{P}_z \subset \mathbb{R}^{n_z}$, $n_z = n_u$, *two integers r and s, a mapping $\psi(\cdot) : \mathbb{R}^{n_x} \times (\mathbb{R}^{n_u})^{s+1} \to \mathbb{R}^{n_u}$ of rank n_u, a mapping $\phi_0(\cdot) : (\mathbb{R}^{n_u})^{r+1} \to \mathbb{R}^{n_x}$ of rank n_x, and a mapping $\phi_1(\cdot) : (\mathbb{R}^{n_u})^{r+2} \to \mathbb{R}^{n_u}$ of rank n_u, with all mappings in a suitably chosen open subset, such that*

$$\begin{aligned} \mathbf{z}(t) &:= \psi(\mathbf{x}(t), \mathbf{u}(t), \dot{\mathbf{u}}(t), \cdots, \mathbf{u}^{(s)}(t)) \\ \mathbf{x}(t) &:= \phi_0(\mathbf{z}(t), \dot{\mathbf{z}}(t), \cdots, \mathbf{z}^{(r)}(t)) \\ \mathbf{u}(t) &:= \phi_1(\mathbf{z}(t), \dot{\mathbf{z}}(t), \cdots, \mathbf{z}^{(r+1)}(t)) \end{aligned} \tag{4.2}$$

Remark 1 *If such mappings can be found then the differential equation $\frac{d}{dt}\phi_0(\cdot) = f(\phi_0(\cdot), \phi_1(\cdot))$ is identically satisfied [14].*

Remark 2 *In some cases, $\mathbf{z}$ is in fact a subset of the state-vector $\mathbf{x}$. The function $\psi(\cdot)$ is then obvious.*

Now, simplified aircraft dynamics was shown to be flat in [35], whereas simplified helicopter dynamics was also shown to be flat in [41]. In the sequel, we show that the rigid-body dynamics, expressed in the body-axis frame (see Appendix C of Chapter 2), is flat when choosing the following six specific states as flat outputs.

4.3.1. Flat outputs

Recall that the twelve rigid-body states have been defined in Chapter 3 as[8]

$$\mathbf{x} = \begin{pmatrix} x_N & x_E & x_Z & u & v & w & p & q & r & \phi & \theta & \psi \end{pmatrix}^T \tag{4.3}$$

Now we give the following result.

---

[7]The rigid-body dynamics has been presented in Appendix C of Chapter 2.
[8]Refer also to the nomenclature given in Appendix A of Chapter 2.

Lemma 1 *Let real scalars n_x and n_u, of Definition 1, be chosen such that $n_x = 12$ and $n_u = 6$, then by selecting the following six body states as flat outputs*

$$\mathbf{z} = \begin{pmatrix} x_N & x_E & x_Z & \phi & \theta & \psi \end{pmatrix}^\top \tag{4.4}$$

we can express the remaining six body states

$$\begin{pmatrix} u & v & w & p & q & r \end{pmatrix}^\top \tag{4.5}$$

together with the forces inputs $\mathbf{F}_{CG}^b = (F_{CG_X}^b \, F_{CG_Y}^b \, F_{CG_Z}^b)^\top$, and moments inputs $\mathbf{M}_{CG}^b = (M_{CG_X}^b \, M_{CG_Y}^b \, M_{CG_Z}^b)^\top$, as given in Eq. (2.4)–Eq. (2.8), in terms of the flat outputs $\mathbf{z}$ and their derivatives.

Proof 1 *See Appendix E.*

4.3.2. Flat output parametrization

To transform the trajectory planning problem from an infinite-dimensional one to a finite one, a parametrization of the flat outputs $\mathbf{z} = \begin{pmatrix} x_N & x_E & x_Z & \phi & \theta & \psi \end{pmatrix}^\top$ over a space of basis functions is required. Here numerous alternatives are available, e.g. generic polynomial parameterizations have been addressed in [13, 14, 49, 50], spline parameterizations have been applied in [30, 43, 51–55], whereas pseudospectral parameterizations have been used in [26, 47]. In this Chapter, and with a view on using a computationally tractable approach, we apply elementary polynomial parametrizations, as was also done in [13, 14]. Using Eq. (4.4), we can express the flat outputs as

$$\mathbf{z}(t) = \begin{pmatrix} x_N(t) & x_E(t) & x_Z(t) & \phi(t) & \theta(t) & \psi(t) \end{pmatrix}^\top = \left(\sum_{i=0}^{n} a_{i,1} t^i \ \dots \ \sum_{i=0}^{n} a_{i,n_u} t^i \right)^\top \tag{4.6}$$

with t the time variable, and $\{a_{i,j}\}_{(i=0,j=1)}^{(i=n,j=n_u)}$ the to-be-determined polynomial coefficients. From this flat output definition, and from the rigid-body dynamics, we infer that integer $r = 1$ in Definition 1[9]. Now, from [14] we need to choose n such that $n \geq 2(r+1) + 1 \Rightarrow n \geq 5$. In order to increase the likelihood of finding feasible trajectories, especially for the autorotation case, the integer n should be chosen much higher than its lower bound, i.e. $n \gg 5$. However, choosing a high n will inevitably increase the computational cost of the optimization problem, hence a trade-off needs to be considered. Based upon simulation results, we choose $n = 7$ as this provided a good compromise between trajectory smoothness and computational cost[10].

4.3.3. Optimal trajectory planning for the engine OFF case

The TP optimization problem, as in Chapter 3, consists of a cost functional $J(\cdot)$, with contributions from a fixed cost $\Phi(\cdot)$, and a running cost over time $\int_\Omega \Psi(\cdot) dt$, with the independent

---

[9]Here the integer s in Definition 1 is not defined since the flat outputs $\mathbf{z}$ depend only on a subset of the states $\mathbf{x}$, and not on the model inputs $\mathbf{u}$.

[10]We do acknowledge that this is a rather empirical justification. In fact a more rigorous analysis of the following trade-off: *trajectory smoothness* vs. *computational tractability* is here desirable.

time variable t defined over the time domain $\Omega = (T_o, T_f)$, where the final time T_f may be free or fixed. This cost is given by

$$J(\mathbf{x}(t), \mathbf{u}(t), T_o, T_f) := \Phi(\mathbf{x}(T_o), \mathbf{x}(T_f), T_f) + \int_\Omega \Psi(\mathbf{x}(t), \mathbf{u}(t), t)dt \qquad (4.7)$$

and from Definition 1 here-above, this cost is equivalently expressed as a function of the flat output $\mathbf{z}$ as follows

$$\begin{aligned} J(\phi_0(\mathbf{z}(t), \dot{\mathbf{z}}(t)), \phi_1(\mathbf{z}(t), \dot{\mathbf{z}}(t), \ddot{\mathbf{z}}(t)), T_o, T_f) \quad &:= \Phi(\phi_0(\mathbf{z}(T_o), \dot{\mathbf{z}}(T_o)), \phi_0(\mathbf{z}(T_f), \dot{\mathbf{z}}(T_f)), T_f) \\ &+ \int_\Omega \Psi(\phi_0(\mathbf{z}(t), \dot{\mathbf{z}}(t)), \phi_1(\mathbf{z}(t), \dot{\mathbf{z}}(t), \ddot{\mathbf{z}}(t)), t)dt \end{aligned}$$
$$(4.8)$$

with the mappings $\phi_0(\cdot)$ and $\phi_1(\cdot)$ given by Eq. (4.39)–Eq. (4.44). The solution to the optimal trajectory planning gives the optimal polynomial coefficients $\{\hat{a}_{i,j}\}_{(i=0, j=1)}^{(i=n, j=n_u)}$ which minimize the cost functional $J(\cdot)$

$$\{\hat{a}_{i,j}\}_{(i=0, j=1)}^{(i=n, j=n_u)} := \arg \min_{a_{i,j} \in \mathbb{R}} J(\phi_0(\mathbf{z}(t), \dot{\mathbf{z}}(t)), \phi_1(\mathbf{z}(t), \dot{\mathbf{z}}(t), \ddot{\mathbf{z}}(t)), T_o, T_f) \qquad (4.9)$$

while enforcing the following constraints (which are similar to the ones of Chapter 3)

- An initial-time boundary condition which corresponds, in our case, to the initial values of the control inputs $\phi_1(\mathbf{z}(T_o), \dot{\mathbf{z}}(T_o), \ddot{\mathbf{z}}(T_o))$ and states $\phi_0(\mathbf{z}(T_o), \dot{\mathbf{z}}(T_o))$.

- A final-time boundary inequality condition, of the form

$$B_f(\phi_0(\mathbf{z}(T_f), \dot{\mathbf{z}}(T_f)), \phi_1(\mathbf{z}(T_f), \dot{\mathbf{z}}(T_f), \ddot{\mathbf{z}}(T_f)), T_f) \quad \leq 0 \qquad (4.10)$$

- An algebraic trajectory inequality constraint, of the form

$$T(\phi_0(\mathbf{z}(t), \dot{\mathbf{z}}(t)), \phi_1(\mathbf{z}(t), \dot{\mathbf{z}}(t), \ddot{\mathbf{z}}(t))) \leq 0 \quad t \in \Omega \qquad (4.11)$$

Remark 3 *Notice that, contrary to the optimization problem of Chapter 3, there are here no Ordinary Differential Equations (ODEs) constraints that need to be enforced. This allows for significant computational benefits.*

Now, computing a numerical solution to the continuous-time problem formulation, Eq. (4.8)–Eq. (4.11), requires first some form of problem discretization. Again with an eye on computational tractability, in this Chapter we choose a simple discretization scheme, involving K collocation points, evenly spaced on domain Ω (i.e. resulting in the discretized domain $\Omega_K = \{T_o\ t_1... \ t_{K-2}\ T_f\}$). We use here a simple rectangular discretization approach, using 16 evenly spaced points[11]. Obviously better discretization methods exist, however, our objective, in this Chapter, is also to keep the computational cost to a minimum. Once discretized, our problem is transcribed into a NonLinear Programming problem (NLP)

---

[11]Based upon simulation results with initial altitudes below 100 m, the choice of 16 collocation points provided a good compromise between accuracy and computational tractability. We do acknowledge that this is a rather empirical justification. In fact a more rigorous analysis of the following trade-off: *accuracy* vs. *computational tractability* is here desirable.

[56, 57], this latter being solved numerically by well known and efficient optimization techniques. In our case we use the MATLAB function *fmincon* of the Optimization Toolbox, based upon an Interior Point (IP) method[12] [59–62]. This nonlinear optimization takes a few seconds to complete in a MATLAB environment (and may likely be one or two orders of magnitude faster, once programmed in the C language). We address next, in more details, the various elements of our optimization problem Eq. (4.8)–Eq. (4.11).

COST FUNCTIONAL
First, we choose to set the fixed cost $\Phi(\cdot)$ to zero. Indeed, this fixed cost may equivalently be replaced by tight bounds on the final state values (as discussed in Chapter 3). In turn this simplifies the optimization process, and lowers the computational cost. Next, the cost objective for the un-powered flight case, i.e. autorotation landing, is defined as a running cost over time, and is given by

$$J_{OFF}(\mathbf{x}(t), \mathbf{u}(t)) \quad = \quad \int_{\Omega} \Big[(\dot{F}^b_{CG_X})^2 + (\dot{F}^b_{CG_Y})^2 + (\dot{F}^b_{CG_Z})^2 + (\dot{M}^b_{CG_X})^2 + (\dot{M}^b_{CG_Y})^2 + (\dot{M}^b_{CG_Z})^2$$
$$+ W_u u^2 + W_v v^2 + W_w w^2 + W_\psi (\psi - \psi_f)^2 \Big] dt$$

$$(4.12)$$

This cost is identical to the one of Eq. (3.8) in Chapter 3, except for the following

- The cost in Eq. (3.8) of Chapter 3 encourages smoother control policies, by minimizing the rate of control inputs $\dot{\theta}_0^2 + \dot{\theta}_{1c}^2 + \dot{\theta}_{1s}^2 + \dot{\theta}_{TR}^2$. These control inputs represent the true inputs to the helicopter system. Similarly, the cost in Eq. (4.12) also encourages smoother control policies, however, since the true control inputs do not appear in the model of Section 4.3.1 (in this model the forces and moments are the inputs), the cost in Eq. (4.12) minimizes the rate of all forces and moments $(\dot{F}^b_{CG_X})^2 + (\dot{F}^b_{CG_Y})^2 + (\dot{F}^b_{CG_Z})^2 + (\dot{M}^b_{CG_X})^2 + (\dot{M}^b_{CG_Y})^2 + (\dot{M}^b_{CG_Z})^2$.

- The main rotor Revolutions Per Minute (RPM) Ω_{MR} is not included here, since this state does not belong to the rigid-body states, and hence does not appear in the model of Section 4.3.1. The issue will further be addressed in Section 4.3.3.

FINAL-TIME BOUNDARY CONDITION
Now, with respect to the final-time boundary condition, as expressed in Eq. (4.10), the aim is here twofold: 1) set the vehicle on the ground, possibly at a specified location; and 2) provide tight bounds on the vehicle kinetic energy and attitude angles, in accordance with technical specifications for safe (i.e. successful) landing. We specifically address the definition of a 'successful' autorotation landing.

Definition 2 *A successful autorotation landing is defined as follows*

- *Final values for the body horizontal velocities $|u| \leq 0.5$ m/s, and $|v| \leq 0.5$ m/s[13].*

---

[12]Note that numerical methods for solving NLPs fall into two categories, namely heuristic methods and gradient-based methods. The main idea behind a heuristic optimization method is that the search is performed in a stochastic manner rather than in deterministic one [58]. Heuristic optimizations, e.g. genetic algorithms, are known as global techniques, i.e. converging towards the global optimum. On the other hand gradient-based methods, such as Sequential Quadratic Programming (SQP) or Interior Point (IP) methods, are known as local methods in that, upon convergence, a locally optimal solution will generally be obtained [58].

[13]Non-zero horizontal velocities allow for a so-called slide-on-skids landing.

- *Final value for the body vertical velocity $|w| \leq 0.25$ m/s.*

- *Final values for the roll and pitch angles $|\phi| \leq 10\,°$, and $|\theta| \leq 10\,°$.*

Since roll and pitch angles will not be controlled by the TT (this issue will further be discussed in Section 4.4), we also derive, in Appendix D of Chapter 4, the maximum acceptable roll (or pitch) angle, for a successful landing, and hence justify the chosen attitude bounds $|\phi| \leq 10\,°$ and $|\theta| \leq 10\,°$.

Bound on total flight time   In Chapter 3, we found that for a fixed initial[14] height above ground, increasing the initial helicopter velocity had only a relatively limited effect on flight time and hence stabilized rate of descent. This potentially indicates that the flight time, in autorotation, is only lightly correlated with the initial vehicle velocity, whereas it is primarily influenced by the initial height above ground. This led us to consider an empirical bound T_{OFF} on flight time T_f, $T_f \leq T_{OFF}$, with T_{OFF} deduced from simulation experiments as follows: Let x_{Z_I} be the initial height above ground at the instant of engine failure, and recall v_{ih} to be the helicopter induced velocity in hover, then the bound T_{OFF} is set, after several simulation experiments[15], within the range:

$$\frac{x_{Z_I}}{1.75 v_{ih}} \leq T_{OFF} \leq \frac{x_{Z_I}}{1.50 v_{ih}} \tag{4.13}$$

Remark 4 *The reason for bounding the flight time $T_f \leq T_{OFF}$ is discussed next. Although the main rotor RPM dynamics is used in the helicopter nonlinear HOM, the RPM dynamics is not included in the flat model description, i.e. in Section 4.3.1, since not part of the rigid-body dynamics. By so doing, the same flat model can be used for both the engine OFF and ON cases, hence simplifying the trajectory planning software. However, excluding the main rotor RPM dynamics from the planning problem is only possible, i.e. will result in feasible autorotative trajectories, if the trajectory flight time is kept small enough. Since the RPM dynamics is eliminated from the planning problem, the main rotor RPM Ω_{MR} signal may not be required for the trajectory tracking system either. Thus, the standard requirement consisting of adding a dedicated magnetic or optical RPM sensor, on the main rotor shaft or on the gear-box of a small-scale helicopter, may here be dropped.*

TRAJECTORY CONSTRAINTS

Regarding the trajectory constraints, as expressed in Eq. (4.11), these are conceptually identical[16] to the ones set in Section 3.2.2 of Chapter 3, except for the trajectory constraints on the inputs, and on the main rotor RPM Ω_{MR} (see Section 4.3.3). For the constraints on the inputs, these are set on total forces and moments (based upon simulation results). Regarding the main rotor RPM Ω_{MR}, there are no constraints, since Ω_{MR} is not part of the state-vector.

---

[14]By *initial* we mean at the start of the engine OFF flight maneuver.

[15]The coefficients 1.50 and 1.75 in $\frac{x_{Z_I}}{1.75 v_{ih}} \leq T_{OFF} \leq \frac{x_{Z_I}}{1.50 v_{ih}}$ are empirically deduced, after several simulation experiments, for the case of the small-scale Align T-REX helicopter, with physical parameters as given in Table 2.1 of Chapter 2. A different helicopter, or even an Align T-REX helicopter with a different main rotor inertia, may likely result in different coefficient values.

[16]Some numerical values of bounds and constraints may differ from the ones used in Chapter 3, in particular since Chapter 3 and Chapter 4 do not use the same helicopter UAV (as explained in Section 3.5 of Chapter 3).

4.4. Robust control based Trajectory Tracking (TT)

The goal is to design a TT module for our small-scale helicopter UAV. This tracker should allow the vehicle to fly along previously planned optimal trajectories. However, with four control inputs and at least twelve measured outputs (i.e. the rigid-body states), the helicopter is heavily under-actuated, which inevitably will limit the performance of the tracking system. Now, since control over position and velocity is a primary objective of our application, we choose to have the helicopter track the following seven references, namely 3D inertial[17] positions $(x_N\ x_E\ x_Z)^\top$, 3D body velocities $(u\ v\ w)^\top$, and heading angle ψ. In addition, based upon simulation results using the helicopter HOM, it is found that position dynamics is much slower than velocity dynamics. This justifies a design philosophy based upon the successive loop closure of feedback loops, where a sequential design process of inner- and outer-loops is sought, also known as a *Master-Slave* control configuration see Fig. 4.2. This design approach is thus related to the well-known time-scale separation principle [63], between slow and fast dynamics of a dynamical system, and supposes that the bandwidth of the inner-loop is much higher than the bandwidth of the outer-loop[18].
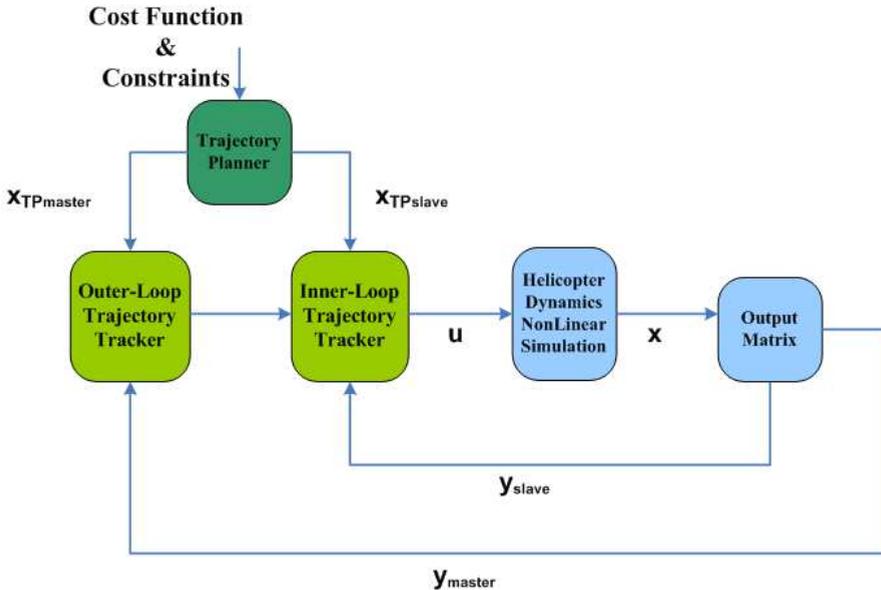
Figure 4.2: Master-Slave control configuration.

The outer-loop aims at tracking the planned inertial 3D position $(x_N\ x_E\ x_Z)^\top_{TP}$. On the other hand, the role of the inner-loop consists in tracking the planned heading ψ_{TP}, and the

---

[17]Which is equivalent to North-East-Down (NED) position in our flight dynamics model.

[18]Note that the control design by time-scales leads not only to a simpler and more modular control architecture, but also to a potentially more robust one [36]. Indeed the existence of time-scales means that the system is numerically ill-conditioned, hence a control law ignoring these aspects may also be ill-conditioned, thus more difficult to implement, and potentially more sensitive to modeling errors [36].
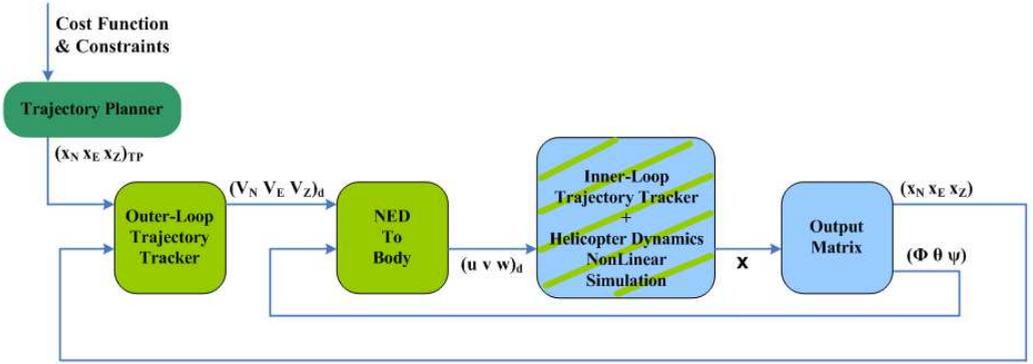
Figure 4.3: Outer-Loop, control interconnection diagram.
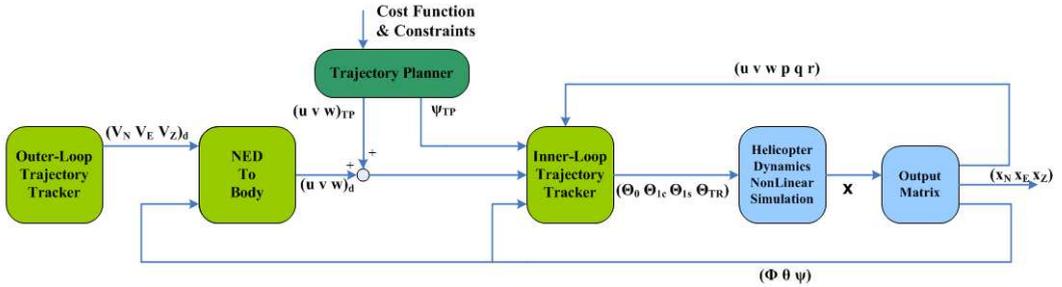
Figure 4.4: Inner-Loop, control interconnection diagram.

4

planned 3D body linear velocities $(u\ v\ w)_{TP}^{\top}$, these latter being adjusted by the outputs of the outer-loop controller $(u\ v\ w)_{d}^{\top}$ to allow for position control, see Fig. 4.3 and Fig. 4.4. In these figures, **x** represents the state-vector (with dimension twenty-four), defining the states of the nonlinear helicopter HOM. The $(u\ v\ w)_{d}^{\top}$ can be seen as a "delta" correction to the nominal velocities $(u\ v\ w)_{TP}^{\top}$. Hence, the to-be-tracked velocities by the inner-loop controller are given by $(u\ v\ w)_{TP}^{\top} + (u\ v\ w)_{d}^{\top}$. Next, since the outputs of the outer-loop are given in the inertial frame, i.e. North-East-Down (NED) frame, we need a nonlinear inversion to convert the reference velocities from NED to body frame, i.e. $(u\ v\ w)_{d}^{\top} = \mathbb{T}_{ob}^{\top}(V_N\ V_E\ V_Z)_{d}^{\top}$, with the rotation matrix $\mathbb{T}_{ob}$ given in Eq. (2.8) of Chapter 2. Note also that in Fig. 4.4 all signals, except position, are fed-back into the controller to improve the closed-loop performance.

As the helicopter dynamics is nonlinear, the design of the TT controller necessitates an approach that effectively respects or exploits the system's nonlinear structure. To this end, several control methods are available, from 1) robust control; 2) classical gain-scheduling, and Linear Parameter-Varying (LPV) approaches; to 3) truly nonlinear control methods (e.g. nonlinear receding horizon control, Lyapunov based methods such as sliding mode and backstepping, adaptive control, or even passivity-based approaches). In this Chapter we select an approach that combines simplicity and computational efficiency, i.e. we choose to

apply a robust control μ strategy. This method consists in using a nominal LTI plant coupled with an uncertainty, and applying a small gain approach [18, 19] to design a single robust LTI controller, valid over a wide portion of the flight envelope. Now, rather than modeling the uncertainty in a detailed or structured manner, an input multiplicative uncertainty is added here to compensate for the unmodeled plant nonlinearities and unmodeled higher-order rotor dynamics[19], by lumping all types of model uncertainty together into a complex, full-block, input multiplicative uncertainty. The robust controller synthesis consists then in obtaining a controller insensitive to this multiplicative uncertainty at the plant input.

4.4.1. Linear multivariable μ control design

Both, the inner- and outer-loop controllers are designed according to the robust control design paradigm, in a two-degrees-of-freedom control structure (i.e. using both feedback and feedforward). Here the feedback part is used to reduce the effect of uncertainty, whereas the feedforward part is added to improve tracking performance [64], and for optimality, both feedback and feedforward are designed in one step. First, a nominal plant $P(s)$ (and $P_d(s)$ for the disturbance) is obtained by linearizing the nonlinear helicopter model at some specified condition (to be discussed in the sequel). Next, we define the generalized plant $G_P(s)$ which maps the exogenous inputs $\mathbf{w} = [\mathbf{n}^\top \ \mathbf{r}^\top \ \mathbf{d}^\top]^\top$ and control inputs $\mathbf{u}$, to controlled outputs $\mathbf{z} = [\mathbf{z_u}^\top \ \mathbf{z_p}^\top]^\top$ and measured outputs $\mathbf{v} = [\mathbf{r}^\top \ \mathbf{y}^\top]^\top$, see Fig. 4.5.
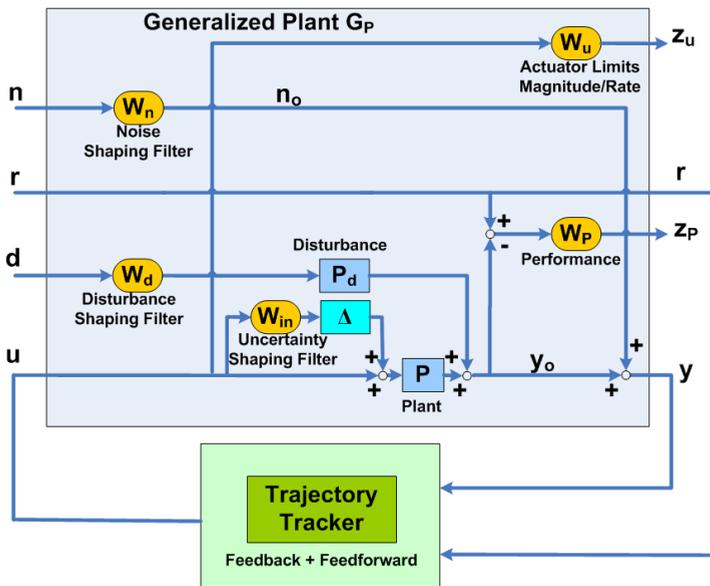
Figure 4.5: Closed-Loop interconnection structure for robust controller synthesis.

The signals include also the sensors noise $\mathbf{n}$ (and $\mathbf{n_o}$), the reference signals $\mathbf{r}$, the disturbance signals $\mathbf{d}$, the actuators performance signal (to limit actuator deflection magnitudes

---

[19]Unmodeled in the low-order nominal LTI plant used for control design, these are however modeled in the high-order nonlinear plant of Chapter 2.

and rates) $\mathbf{z_u}$, the desired performance in terms of closed-loop signal responses $\mathbf{z_p}$, and the system outputs $\mathbf{y}$ (and $\mathbf{y_o}$), such that

$$
\begin{pmatrix} \mathbf{z_u} \\ \mathbf{r} \\ \mathbf{z_p} \\ \mathbf{y} \end{pmatrix} = \begin{bmatrix} 0 & 0 & 0 & W_u \\ 0 & I & 0 & 0 \\ 0 & W_p & -W_p\, P_d\, W_d & -W_p\, P\, (I + \Delta W_{in}) \\ W_n & 0 & P_d\, W_d & P\, (I + \Delta W_{in}) \end{bmatrix} \begin{pmatrix} \mathbf{n} \\ \mathbf{r} \\ \mathbf{d} \\ \mathbf{u} \end{pmatrix} \tag{4.14}
$$

For the weights, which help shape the performance and robustness characteristics of the closed-loop system, we use the input weight $W_{in}(s)$, the performance weight $W_p(s)$, the actuator weight $W_u(s)$, the sensor noise weight $W_n(s)$, and the disturbance weight $W_d(s)$. Now $W_{in}(s)$ and $\Delta(s)$, in Fig. 4.5, parametrize the uncertainty or errors in the model. The Transfer Function (TF) $W_{in}(s)$ is assumed known and reflects the amount of uncertainty in the model, whereas the TF $\Delta(s)$ is assumed to be stable and unknown, except for the norm condition $\|\Delta(s)\|_\infty \le 1$. Next, the generalized plant $G_P(s)$ has a linear fractional dependence on the input uncertainty $\Delta(s)$, and is represented by the upper Linear Fractional Transformation (LFT) interconnection

$$
\begin{pmatrix} \mathbf{z} \\ \mathbf{v} \end{pmatrix} = G_P \begin{pmatrix} \mathbf{w} \\ \mathbf{u} \end{pmatrix} = F_u(M, \Delta) \begin{pmatrix} \mathbf{w} \\ \mathbf{u} \end{pmatrix} \tag{4.15}
$$

where $M(s)$ is a known LTI plant, see Fig. 4.6, and $\Delta(s)$ some complex, full-block, four-by-four[20], operator specifying how the uncertainty enters the plant dynamics.
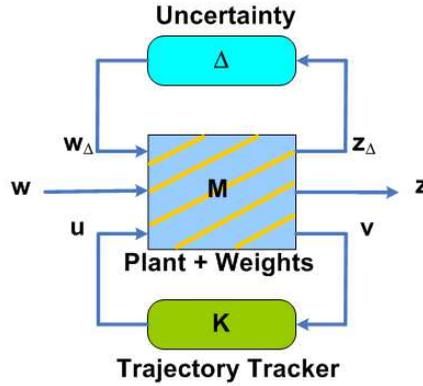
Figure 4.6: Standard $M - \Delta - K$ robust control framework.

The feedback structure associated with the LFT interconnection Eq. (4.15) is given by

$$
\begin{pmatrix} \mathbf{z_\Delta} \\ \mathbf{z} \\ \mathbf{v} \end{pmatrix} = M \begin{pmatrix} \mathbf{w_\Delta} \\ \mathbf{w} \\ \mathbf{u} \end{pmatrix} \qquad \mathbf{w_\Delta} = \Delta\, \mathbf{z_\Delta} \tag{4.16}
$$

with $\mathbf{z_\Delta}$, and $\mathbf{w_\Delta}$, the inputs and outputs of the operator $\Delta(s)$, see Fig. 4.6, and the closed-loop operator from exogenous inputs $\mathbf{w}$ to controlled outputs $\mathbf{z}$ is given by

---

[20]The helicopter plant has four control inputs.

$$T(M, K, \Delta) = F_l(F_u(M, \Delta), K) \tag{4.17}$$

with $K(s)$ the to-be-synthesized controller. The goal of the controller is to minimize the $\mathcal{L}_2$-gain bound γ from the exogenous inputs $\mathbf{w}$ to the controlled outputs $\mathbf{z}$, despite the uncertainty $\Delta(s)$. Based upon small gain considerations [18, 19], this goal is approximated by the minimization of the $\mathcal{H}_\infty$ norm of $F_l(M, K)$. Now, better performance may potentially be obtained by synthesizing $K(s)$ through D-K iteration [65, 66]

$$K = \arg \min_{K} \inf_{D, D^{-1} \in \mathcal{H}_\infty} \|DF_l(M, K)D^{-1}\|_\infty \tag{4.18}$$

with $D(s)$ a stable and minimum-phase scaling matrix, chosen such that $D(s)\Delta(s) = \Delta(s)D(s)$.

We have presented here-above a general TT architecture, that will be applied twice, once for the inner-loop controller design and once for the outer-loop controller design. When synthesizing the inner-loop TT, we use the following signals: the control inputs $\mathbf{u} = (\theta_0 \ \theta_{1c} \ \theta_{1s} \ \theta_{TR})^\top$, the reference signals $\mathbf{r} = (u_{TP} + u_d \ v_{TP} + v_d \ w_{TP} + w_d \ \psi_{TP})^\top$, the wind disturbance signals (given in inertial frame) $\mathbf{d} = (V_{N_w} \ V_{E_w} \ V_{Z_w})^\top$, the system outputs $\mathbf{y} = (u \ v \ w \ p \ q \ r \ \phi \ \theta \ \psi)^\top$, and the sensors noise $\mathbf{n}$ (added to the system outputs). When synthesizing the outer-loop TT, we use the following signals: the control inputs $\mathbf{u} = (V_N \ V_E \ V_Z)_d^\top$, the reference signals $\mathbf{r} = (x_N \ x_E \ x_Z)_{TP}^\top$, the system outputs $\mathbf{y} = (x_N \ x_E \ x_Z)^\top$, and the sensors noise $\mathbf{n}$ (again added to the system outputs). Here the outer-loop does not include disturbance signals, since the wind has already been accounted for, within the inner-loop control structure. For controller assessment and validation, a two-step approach is here adopted consisting in: 1) evaluating first the closed-loop characteristics, with the help of several 'metrics', using the nominal LTI plants P (and P_d); and 2) evaluating the closed-loop characteristics on the nonlinear helicopter model of Chapter 2. In the following section, we briefly present these control assessment 'metrics'.

Remark 5 *The same control architecture will be used for both the engine OFF and ON cases, what will however change, between the OFF and ON cases, is the numerical values of the weights and controller matrices.*

4.4.2. Controller assessment metrics
For controller assessment, we analyze the results from several 'metrics' [64].

- The output loop TF $L(s) = P(s)K(s)$, representing the open-loop gain.

- The so-called 'Gang of four' TFs [67]. Here the following signals, as found in Fig. 4.5, are used: the control inputs $\mathbf{u}$, the reference signals $\mathbf{r}$, the system outputs $\mathbf{y}, \mathbf{y_o}$, and the sensors noise $\mathbf{n}, \mathbf{n_o}$. To these signals, we also add two disturbance signals, as defined in [67]: the input disturbance signals $\mathbf{d_i}$ and the output disturbance signals $\mathbf{d_o}$, in order to define the following TFs (see Fig. 4.7)

  1. The input sensitivity $S_i(s) = (I + L(s))^{-1}P(s)$, representing the TF $d_i \rightarrow y_o$.
  2. The output sensitivity $S_o(s) = (I + L(s))^{-1}$, representing the TF $d_o \rightarrow y_o$.
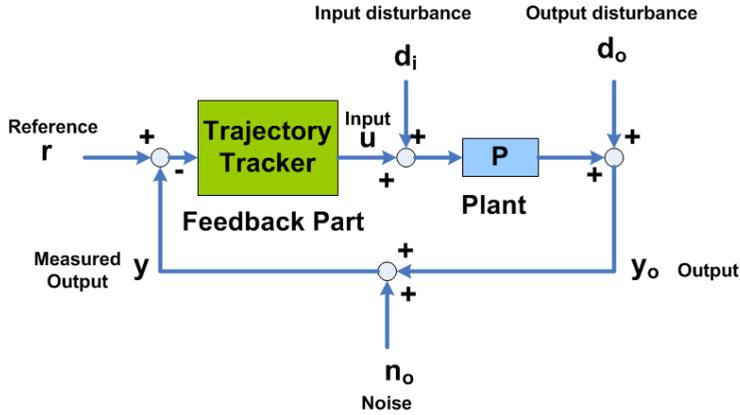
Figure 4.7: Basic feedback loop.

3. The input complementary sensitivity $T_i(s) = L(s)(I + L(s))^{-1}$, representing the TFs $r \rightarrow y_o$ (also $n_o \rightarrow y_o$ and $d_i \rightarrow u$).

4. The output complementary sensitivity $T_o(s) = K(s)(I + L(s))^{-1}$, representing the TFs $r \rightarrow u$ (also $n_o \rightarrow u$ and $d_o \rightarrow u$).

- To evaluate the frequency range over which the control is effective, we consider the following bandwidths [64]

  1. w_C being the gain crossover frequency where $|L(s)|$ first crosses 0 dB, from above.

  2. w_B being the lowest frequency where $|S_i(s)|$ crosses -3 dB, from below.

  3. w_{BT} being the highest frequency where $|T_i(s)|$ crosses -3 dB, from above.

- The Robust Stability (RS) metric, defined as $RS \Leftrightarrow \mu_\Delta(N_{11}(s)) \leq 1$, with $N(s) = F_l(M(s), K(s))$, with $N_{11}(s)$ the upper left block corresponding to the full, complex, four-by-four uncertainty Δ (i.e. our plant input multiplicative uncertainty), and $M(s)$ as given in Fig. 4.6.

- The Robust Performance (RP) metric, defined as $RP \Leftrightarrow \mu_\Delta(N(s)) \leq 1$, for the structured uncertainty $\tilde{\Delta}$ as

$$\tilde{\Delta} := \{diag(\Delta, \Delta_P) \quad \|\Delta_P\|_\infty \leq 1\} \tag{4.19}$$

with Δ_P an unstructured (complex, full-block), fictitious uncertainty of size dim($\mathbf{w}$) by dim($\mathbf{z}$), with $\mathbf{w}$ the exogenous inputs, and $\mathbf{z}$ the controlled outputs.

Remark 6 *We compute both an upper and lower bound for the RS and RP, following the method in [68]. As in [68], we have added 1% of complex perturbations to Δ in order to improve the convergence of the lower bound, albeit at the expense of slight additional conservatism.*

In the following sections, we address the weights selection and controller validation, for the engine OFF case.

4.5. Design of the engine OFF inner-loop controller

As stated in Section 4.4, with four control inputs and twelve measured outputs, the helicopter is heavily under-actuated, which inevitably limits the performance of the tracking system. As mentioned earlier, see also Fig. 4.4, we choose to have the inner-loop track the following four reference signals: 3D body velocities $(u \ v \ w)^\top$, and heading angle ψ. Recall also that the goal of the controller is to minimize the $\mathcal{L}_2$-gain bound γ from the exogenous inputs $\mathbf{w}$ to the controlled outputs $\mathbf{z}$, despite the uncertainty $\mathbf{\Delta}(s)$. The various signals are further given as follows: the exogenous inputs $\mathbf{w} = [\mathbf{n}^\top \ \mathbf{r}^\top \ \mathbf{d}^\top]^\top$, the controlled outputs $\mathbf{z} = [\mathbf{z_u}^\top \ \mathbf{z_p}^\top]^\top$, the control inputs $\mathbf{u} = (\theta_0 \ \theta_{1c} \ \theta_{1s} \ \theta_{TR})^\top$, the measured outputs $\mathbf{v} = [\mathbf{r}^\top \ \mathbf{y}^\top]^\top$, the reference signals $\mathbf{r} = (u_{TP} + u_d \ v_{TP} + v_d \ w_{TP} + w_d \ \psi_{TP})^\top$, the system outputs $\mathbf{y} = (u \ v \ w \ p \ q \ r \ \phi \ \theta \ \psi)^\top$, the wind disturbance signals (given in inertial frame) $\mathbf{d} = (V_{N_w} \ V_{E_w} \ V_{Z_w})^\top$, and the sensors noise $\mathbf{n}$ (added to the system outputs), see Fig. 4.5. Here the signal $\mathbf{y}$ contains all the available measured output signals, except for the 3D position, since the latter is only of interest for the outer-loop controller.

4.5.1. Choice of nominal plant model for the inner-loop control design

As mentioned in Section 4.1, we do not use any gain-scheduling philosophy in this Chapter, rather a single LTI plant is used for controller design. Now, for an engine ON flight condition, it is relatively easy to find equilibrium points, i.e. steady-state flight conditions, at which the nonlinear helicopter model of Chapter 2 can be linearized. The resulting LTI models can subsequently be used for LTI control design. However, for the engine OFF flight condition, this set of equilibrium points, i.e. steady autorotative flight conditions, is rather small and in certain situations even non-existent. For example, when an engine failure happens at a low altitude, the helicopter does not even reach a steady-state autorotation (corresponding to a constant main rotor RPM), rather the helicopter system is continuously in transition from one non-equilibrium point to the next. To mitigate this problem, the approach used here consists in excluding the main rotor RPM Ω_{MR} from the state and measurement vectors, and use this "quasi-steady" modeling approach to find the equilibrium points. By so doing, the control architecture and control design philosophy for the engine OFF case can be made exactly identical to the engine ON case, hence simplifying the overall control system design.

The state-space data used to design the inner-loop trajectory tracker is as follows: the state-vector is of dimension nine given by $\mathbf{x} = (u \ v \ w \ p \ q \ r \ \phi \ \theta \ \psi)^\top$, the control input $\mathbf{u}$ (given here-above) is of dimension four, the wind disturbance $\mathbf{d}$ (given here-above) is of dimension three, and the measurements vector $\mathbf{y} = \mathbf{x}$. This LTI model is obtained by linearizing[21] the helicopter nonlinear model of Chapter 2, at a specific trimmed flight condition. This condition corresponds to hover, with the engine OFF (note that now the main rotor RPM is not in equilibrium anymore). Choosing such a flight condition, with an associated initial velocity of zero, can potentially provide the best description of helicopter behavior during landing (where the helicopter velocity is also very low). The resulting state-space data given in Appendix H[22] of Chapter 2. By using the eigenvalues of the $\mathbb{A}$ matrix in the Popov-

---

[21] According to the linearization procedure given in Section 2.4.1.
[22] The state-space data of the LTI plants given in Appendix H of Chapter 2 are in S.I. engineering units. However, before using the plant for control design, scalings have been used on the input and output matrices in order to

Belevitch-Hautus (PBH) rank test, we found that this LTI plant was both controllable and observable. Simulation results, presented later in Chapter 5, have shown that this nominal LTI plant is indeed suitable for the design of controllers in an engine OFF situation.

4.5.2. SELECTION OF WEIGHTS

The robust control framework makes use of several user-defined weights, see Fig. 4.5. In this Chapter, these weights have been chosen as follows. The multiplicative uncertainty weight $W_{in}(s)$ is of the form $W_{in}(s) = diag[w_{in1}(s), w_{in2}(s), w_{in2}(s), w_{in2}(s)]$, set on the four control input channels $\mathbf{u} = (\theta_0 \; \theta_{1c} \; \theta_{1s} \; \theta_{TR})^\top$, with θ_0 the main rotor blade root collective pitch, θ_{1c} the main rotor lateral cyclic pitch, θ_{1s} the main rotor longitudinal cyclic pitch, and θ_{TR} the tail rotor blade collective pitch. Further, $w_{in1}(s)$ and $w_{in2}(s)$ are filters whose magnitude represent the relative uncertainty at each frequency (i.e. the level of uncertainty in the behavior of the helicopter is assumed frequency dependent). Based upon engineering judgment[23], we choose here for $w_{in1}(s)$ to consider 20% uncertainty at low frequency (DC gain), 100% uncertainty at the filter crossover frequency of 10 Hz (with 10 Hz being roughly the anticipated closed-loop bandwidth for the vertical velocity channel[24]), and 200% uncertainty at infinite frequency. Again, based upon engineering judgment, we choose for $w_{in2}(s)$ to consider 40% uncertainty at low frequency (DC gain), 100% uncertainty at the filter crossover frequency of 5 Hz, and 200% uncertainty at infinite frequency[25], giving

$$w_{in1}(s) = (2s + 22.21)/(s + 111.1)$$
$$w_{in2}(s) = (2s + 23.75)/(s + 59.37) \tag{4.20}$$

Next, the performance weight filter $W_p(s)$ is placed on the (u, v, w, ψ) error signals, to reflect the tracking objective for the three body linear velocities and the heading angle. Here $W_p(s)$ is a four-by-four, diagonal, frequency-varying weight $W_p(s) = diag[w_u(s), w_v(s), w_w(s), w_\psi(s)]$, with each diagonal term defined as a first-order TF $\frac{s/M_P + \omega_B}{s + \omega_B A_{ss}}$. At low frequencies this weighting function should be high in order to keep the error small. Beyond the anticipated bandwidth of the closed-loop system, the tracking error may be released and $W_p(s)$ rolls off [64]. After several controller design cycles, we have settled for

$$
\begin{aligned}
&\text{For} \quad w_u(s) \quad (M_P, \omega_B, A_{ss}) = (2, 0.5\pi \; rad/s, 0.001)\\
&\text{For} \quad w_v(s) \quad (M_P, \omega_B, A_{ss}) = (2, 0.5\pi \; rad/s, 0.001)\\
&\text{For} \quad w_w(s) \quad (M_P, \omega_B, A_{ss}) = (2, 90\pi \; rad/s, 0.001)\\
&\text{For} \quad w_\psi(s) \quad (M_P, \omega_B, A_{ss}) = (2, 4\pi \; rad/s, 0.001)
\end{aligned}
\tag{4.21}
$$

This means that a steady-state tracking error of 0.1% with respect to the normalized filter input is allowed. Further, the difference with the engine ON case is in terms of tracking bandwidth: 1) for the engine OFF case, it is lower on the horizontal channels (u and v velocities) since the LTI model used for control design is somewhat less 'accurate' (due to the non-fixed main rotor RPM, and high descent rates); and 2) for the engine OFF case, the tracking bandwidth is considerably much higher on the vertical channel (w velocity)

---

obtain a normalized LTI plant.

[23] The chosen uncertainty may be overly conservative, or may even be unrealistic. Alternative ways to shape the uncertainties exist, e.g. [69]. The goal here is simply to add some robustness to the closed-loop system.

[24] For each control input, Table 1.1 in Chapter 1 summarizes their primary effects on the vehicle response.

[25] The uncertainty is large at high-frequency since we use a low-order model.

to allow the tracking of a rapidly changing vertical velocity reference. The latter is only feasible if high-bandwidth actuators are mounted on the helicopter (at least for the vertical channel). Now, tracking should not be achieved at the cost of too high control effort. Therefore, both actuator deflection (i.e. amplitude) and rate are penalized through weight $W_u(s) = \text{diag}[w_{act}(s), w_{act}(s), w_{act}(s), w_{act}(s)]$, with

$$w_{act}(s) = 10^n \left(\frac{s + \omega_1}{s + \omega_2} \right)^n \quad \text{with} \quad (n, \omega_1, \omega_2) = (3, 40\pi \ rad/s, 400\pi \ rad/s) \qquad (4.22)$$

corresponding to actuators with a bandwidth of approximately 10 Hz, i.e. twice the bandwidth for the engine ON case[26] (see Appendix B for the engine ON case). Next, a noise weight $W_n(s)$ is set to represent the actual noise levels associated with each sensor, and is defined as a nine-by-nine, constant, diagonal scaling matrix described as follows (given here in its unscaled form)

$$W_n(s) = \text{diag}[0.01 \ m/s, 0.01 \ m/s, 0.01 \ m/s, 3\pi/180 \ rad/s, 3\pi/180 \ rad/s, 3\pi/180 \ rad/s,$$
$$\pi/180 \ rad, \pi/180 \ rad, 3\pi/180 \ rad]$$
$$(4.23)$$

Finally, a wind disturbance weight $W_d(s) = \text{diag}[w_{d_N}(s), w_{d_E}(s), w_{d_D}(s)]$ is added to simulate the frequency content of the NASA Dryden atmospheric wind model[27] [71], resulting in a disturbance bandwidth of 0.06 Hz, 0.12 Hz, and 0.96 Hz along the North, East, and Down (NED) axes respectively. The wind disturbance weights are modeled here, in normalized form, as low-pass filters, as follows

$$
\begin{aligned}
w_{d_N}(s) &= A_d \frac{s+\omega_1}{s+\omega_2} \quad \text{with} \quad (A_d, \omega_1, \omega_2) = (10^3, 0.22\pi \ rad/s, 2.2\pi \ rad/s) \\
w_{d_E}(s) &= A_d \frac{s+\omega_1}{s+\omega_2} \quad \text{with} \quad (A_d, \omega_1, \omega_2) = (10^3, 0.3\pi \ rad/s, 3\pi \ rad/s) \\
w_{d_D}(s) &= A_d \frac{s+\omega_1}{s+\omega_2} \quad \text{with} \quad (A_d, \omega_1, \omega_2) = (10^3, \pi \ rad/s, 10\pi \ rad/s)
\end{aligned}
\qquad (4.24)
$$

4.5.3. Controller synthesis and analysis

For the D-K iteration [72], we obtain after four iterations a stable[28] controller $K(s)$ of order 38, using 0th order (constant) D_s-scalings. The controller is further reduced to 30th order, after balancing and Hankel-norm model reduction [73], without any significant effect on closed-loop robustness and performance. In Fig. 4.8, we visualize the relevant TFs, with the bandwidths for the three main TFs given in Table 4.1. In particular, we see that the bandwidth of $|T_i(s)|$ is about equal to the bandwidth of the actuators, i.e. around 10 Hz (obviously high enough to stabilize the plant, see our discussion in Section 2.4.3 of Chapter 2). Also the closed-loop disturbance rejection, given in Fig. 4.9, shows relatively good attenuation of wind disturbances, i.e. approximately -43 dB at a frequency of 2π rad/s along the Down axis.

---

[26]The engine OFF condition may hence dictate the required actuator specifications.

[27]The wind turbulence, or disturbance, frequency content depends upon the mean wind value, and also upon the vehicle height and speed. For the mean wind value, we chose 8 m/s which is equivalent to a Beaufort wind force value of 4, corresponding to the yearly average wind force along the coast in The Netherlands [70]. For the vehicle height and speed, we chose 1 m and 1 m/s respectively, since a low-speed flight condition, close to the ground, results in the highest wind disturbance bandwidth in the NASA Dryden model.

[28]The controller itself is a stable dynamical system.
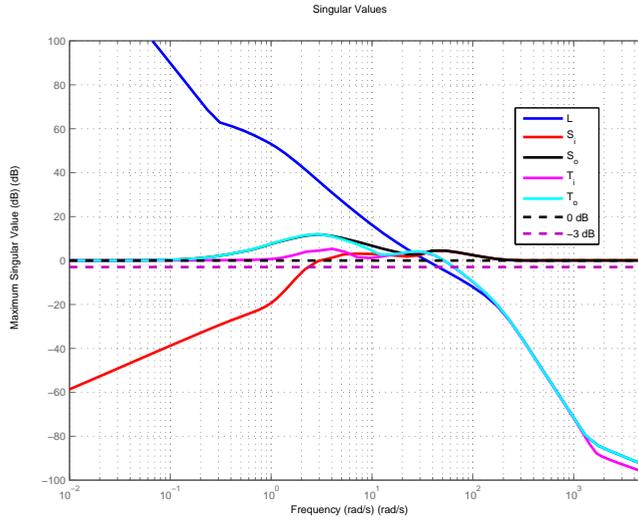
Figure 4.8: Singular values of $L(s)$, $S_i(s)$, $S_o(s)$, $T_i(s)$, and $T_o(s)$, of the inner-loop trajectory tracker (Engine OFF case).
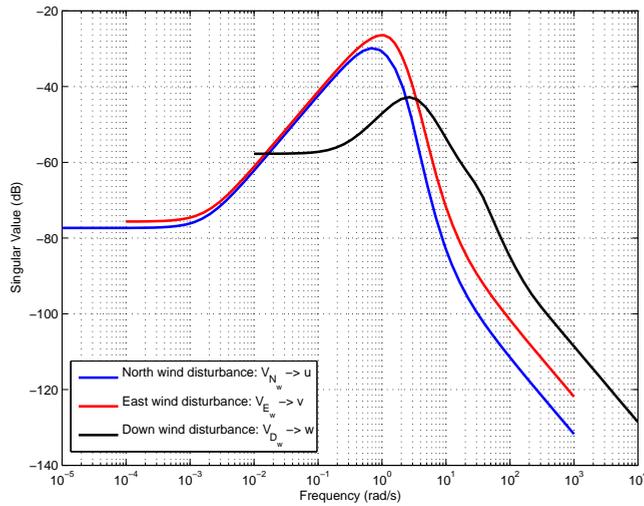
Figure 4.9: Closed-Loop wind disturbance rejection, for North-East-Down (NED) winds, of the inner-loop trajectory tracker (Engine OFF case).

Table 4.1: Open- and closed-loop bandwidths.

	Bandwidths (rad/s)		
	$|L(s)|$	$|S_i(s)|$	$|T_i(s)|$
Case	w_C	w_B	w_{BT}
Engine OFF (Inner-Loop)	35	2.4	65
Engine OFF (Outer-Loop)	3	0.29	6.7
Engine ON (Inner-Loop)	15	2.15	15
Engine ON (Outer-Loop)	0.8	0.37	1.5

We also see that S_o is not well-behaved, since it remains high at both low- and high-frequencies. This can be explained as follows. The output loop $L(s)$ is a 9x9 matrix, with 4 singular-values having very high values (for low-frequencies). These high singular-values correspond to the 4 controlled channels. Since our helicopter is under-actuated, the remaining 5 singular-values are all very low (for all frequencies). Thus, inverting $(I + L(s))$ to get S_o results in maximum singular-values which are most often close to 0 dB.

Next, RS and RP are visualized in Fig. 4.10 and Fig. 4.11. We can see that lower and upper bounds are indistinguishable. We observe that the primordial RS is guaranteed (i.e. a maximum value below 1). On the other hand, as for the engine ON case, we see that RP is not met (i.e. a maximum value well above one). Note that this may potentially be due to the fact that the chosen uncertainty $\mathbf{\Delta}(s)$, shown in Fig. 4.5, is not realistic. If robust performance specifications need to be met, then this could potentially be done by lowering the amount of model input uncertainty, and/or by relaxing some of the assumptions made during the various weights selection. However, from our experience, this will likely compromise the closed-loop performance of the controller, once tested upon the nonlinear system.
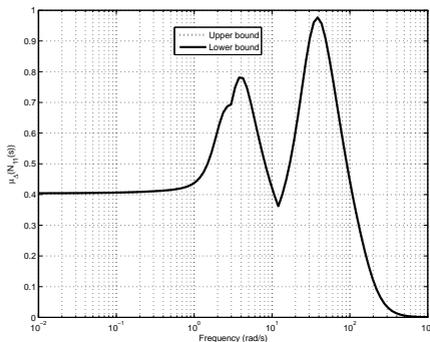
Figure 4.10: Robust Stability of the inner-loop trajectory tracker (Engine OFF case).
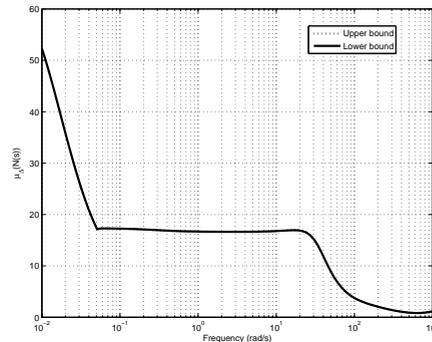
Figure 4.11: Robust Performance of the inner-loop trajectory tracker (Engine OFF case).

4.6. Design of the engine OFF outer-loop controller

We first recall that the design approach is based upon the well-known time-scale separation principle [63] between slow and fast dynamics of a dynamical system, resulting in a so-called *Master-Slave* configuration (see Fig. 4.2), and supposes that the bandwidth of the inner-loop is much higher than the bandwidth of the outer-loop[29].

As mentioned earlier, refer also to Fig. 4.3, we choose to track the following three reference signals: 3D inertial[30] positions $(x_N \ x_E \ x_Z)^\top$. Recall also that the goal of the controller is to minimize the $\mathcal{L}_2$-gain bound γ from the exogenous inputs $\mathbf{w}$ to the controlled outputs $\mathbf{z}$, despite the uncertainty $\Delta(s)$. The various signals are further given as follows: the exogenous inputs $\mathbf{w} = [\mathbf{n}^\top \ \mathbf{r}^\top]^\top$, the controlled outputs $\mathbf{z} = [\mathbf{z_u}^\top \ \mathbf{z_p}^\top]^\top$, the control inputs $\mathbf{u} = (V_N \ V_E \ V_Z)_d^\top$, the measured outputs $\mathbf{v} = [\mathbf{r}^\top \ \mathbf{y}^\top]^\top$, the reference signals $\mathbf{r} = (x_N \ x_E \ x_Z)_{TP}^\top$, the system outputs $\mathbf{y} = (x_N \ x_E \ x_Z)^\top$, and the sensors noise $\mathbf{n}$ (added to the system outputs), see Fig. 4.5. Here the outer-loop does not include disturbance signals, since the wind has already been accounted for, within the inner-loop control structure.

As discussed in Section 4.1, a single LTI plant is again used for controller design. The state-space data used to design the outer-loop trajectory tracker is obtained as follows. An LTI dynamical system can be formed by connecting the nominal LTI model, used for the inner-loop TT, with its inner-loop controller, and subsequently adding a set of integrators on the 3D velocities to generate the 3D inertial positions $(x_N \ x_E \ x_Z)^\top$. This manipulation is readily done in MATLAB, and results in the nominal LTI model needed to design the outer-loop position controller. In our case, we obtain a three-by-three input-output system, with a state-vector of dimension 55. Next a minimum realization is obtained, resulting in a state-vector of dimension 42 (the LTI model is too big to be added to the Appendix). Note that here too scalings need to be applied. Further, and except for three poles at the origin (corresponding to the integration of the 3D velocities), all other eigenvalues of the $\mathbb{A}$ matrix are stable and well damped, implying easier controller design. Again, by using the eigenvalues of the $\mathbb{A}$ matrix in the PBH rank test, we found that the LTI system is both controllable and observable.

The design philosophy for the μ outer-loop TT parallels that of the inner-loop.

4.6.1. Selection of weights

The input multiplicative uncertainty weight $W_{in}(s)$ is of the form $W_{in}(s) = \text{diag}[w_{in2}(s), w_{in2}(s), w_{in1}(s)]$, with $w_{in1}(s), w_{in2}(s)$ identical to the ones used in the engine OFF inner-loop, in Eq. (4.20). Here $w_{in1}(s)$ is applied to the vertical velocity channel (recall that we have $\mathbf{u} = (V_N \ V_E \ V_Z)_d^\top$). In the design of the inner-loop TT, in Section 4.5.2, we had chosen an uncertainty weight equal to $w_{in1}(s)$ on the collective input θ_0. Now, since the vertical velocity channel is mostly influenced by the collective input (see Table 1.1 in Chapter 1), we also assign an uncertainty $w_{in1}(s)$ to the vertical velocity. The same argument holds for uncertainty $w_{in2}(s)$ on the horizontal velocities. Obviously, this choice of the uncertainty

---

[29]Based upon simulation results, using the helicopter model of Chapter 2, it is indeed found that position dynamics is much slower than velocity dynamics.

[30]Which is equivalent to North-East-Down (NED) position in our flight dynamics model.

weight $W_{in}(s)$ is somewhat arbitrary. This said, the purpose here is just to add some robustness to the closed-loop system.

The performance weight filter $W_p(s)$ is placed on the (x_N, x_E, x_Z) error signals to reflect the tracking objective for the inertial position. Here, $W_p(s)$ is a three-by-three diagonal, frequency-varying weight. We have $W_p(s) = \text{diag}[w_{x_N}(s), w_{x_E}(s), w_{x_Z}(s)]$, with each diagonal term defined as a first-order transfer function $\frac{s/M_P + \omega_B}{s + \omega_B A_{ss}}$. After several controller design cycles, we have settled for

$$
\begin{aligned}
&\text{For} \quad w_{x_N}(s) \quad (M_P, \omega_B, A_{ss}) = (2, 0.1\pi\ rad/s, 0.001) \\
&\text{For} \quad w_{x_E}(s) \quad (M_P, \omega_B, A_{ss}) = (2, 0.1\pi\ rad/s, 0.001) \\
&\text{For} \quad w_{x_Z}(s) \quad (M_P, \omega_B, A_{ss}) = (2, 4.5\pi\ rad/s, 0.001)
\end{aligned}
\tag{4.25}
$$

Again, a steady-state tracking error of 0.1% with respect to the normalized input is allowed. The filter bandwidths, on the horizontal channels, are adjusted to be five times smaller than the $W_p(s)$ filter bandwidths of the inner-loop horizontal channels. For the vertical channel bandwidth, instead of a 1:5 ratio, we settled for a 1:20 ratio. These values have been obtained after several simulation experiments.

Next, tracking should not be achieved at the cost of too high control effort (i.e. resulting in much too large velocity setpoints $\mathbf{u} = (V_N\ V_E\ V_Z)_d^\top$ for the inner-loop). This means that both inertial velocities and inertial accelerations should be penalized, through weight $W_u(s) = \text{diag}[w_{act}(s), w_{act}(s), w_{act}(s)]$, with $w_{act}(s)$ identical to the one chosen for the inner-loop, with engine OFF. Again, this choice may be interpreted as rather arbitrary, since here $W_u(s)$ is assigned to the inner-loop setpoints $\mathbf{u} = (V_N\ V_E\ V_Z)_d^\top$, whereas for the design of the inner-loop controller, $W_u(s)$ was assigned to the actuators. Hence, potentially better choices for $W_u(s)$ may exist, although the one selected here provided satisfactory results. Finally, a noise weight $W_n(s)$ is also defined to scale the normalized position measurement noise. The sensor noise model is defined as a three-by-three, constant, diagonal scaling matrix described by (given here in its unscaled form)

$$
W_n(s) = \text{diag}[0.1\ m, 0.1\ m, 0.1\ m]
\tag{4.26}
$$

4.6.2. Controller synthesis and analysis

For the D-K iteration, we obtain after four iterations a stable controller $K(s)$ of order 57, using 0th order D_s-scalings. The controller is further reduced to 30th order (using the same technique as for the inner-loop), without any effect on closed-loop robustness/performance. In Fig. 4.12, we visualize the relevant TFs (we see that $S_i(s) = S_o(s)$, and $T_i(s) = T_o(s)$), with the bandwidths for the three TFs given in Table 4.1. In particular, we see that the bandwidth of $|T_i(s)|$ is ten times lower its inner-loop counterpart, which is good since we do not want both controllers to start interacting with each other. Further, RS is shown in Fig. 4.13, whereas RP is pictured in Fig. 4.14. Again, we observe that RP is not achieved, whereas RS is well guaranteed.
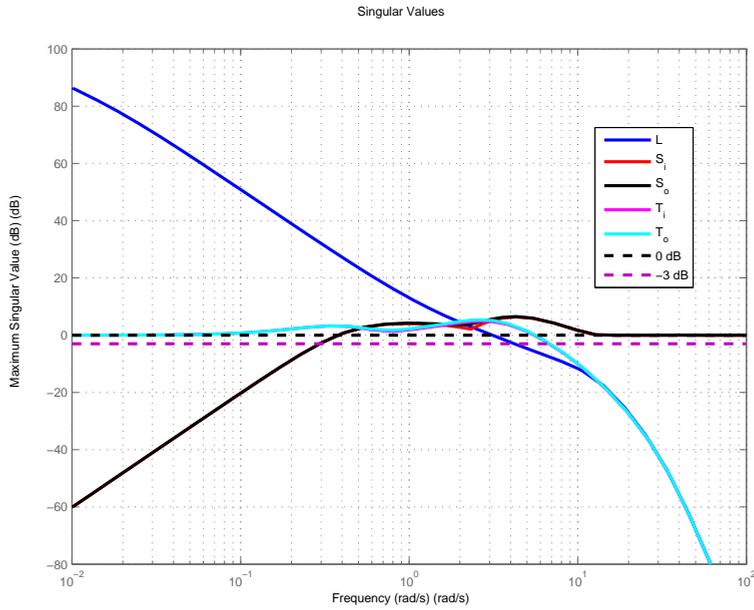
Figure 4.12: Singular values of $L(s)$, $S_i(s)$, $S_o(s)$, $T_i(s)$, and $T_o(s)$, of the outer-loop trajectory tracker (Engine OFF case).

4
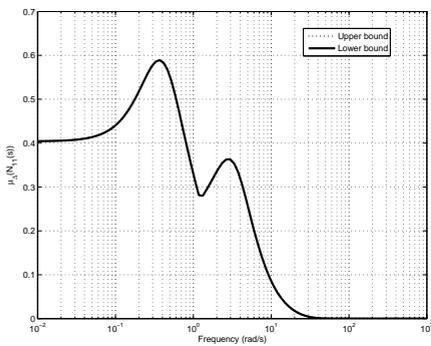
Figure 4.13: Robust Stability of the outer-loop trajectory tracker (Engine OFF case).
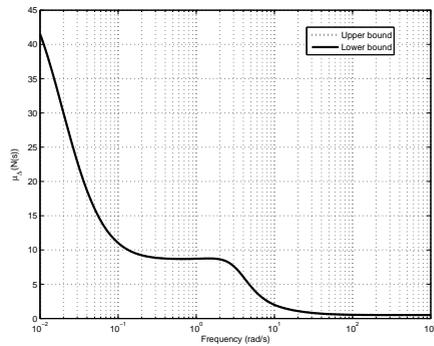
Figure 4.14: Robust Performance of the outer-loop trajectory tracker (Engine OFF case).

4.6.3. Adapting the engine OFF outer-loop controller

When close to the ground, it is crucial to keep the reference velocities as small as possible. To this end, we adapt the outer-loop controller as follows: the position tracking is switched-off, i.e. the values for $(u\ v\ w)_d^\top$ are set to zero, once the helicopter height descends below a predefined threshold (keeping only velocity and heading tracking). This helps lowering the final (touch-down) values of the 3D velocities, by giving more time to the velocity deceleration process. The value of this user-defined altitude threshold depends upon the initial conditions, and depends upon whether it is an engine OFF or ON automatic landing.

4.7. Conclusion

In this Chapter we have presented a trajectory planning and tracking framework, anchored in the combined paradigms of differential flatness based planning and robust control based tracking. Both the engine OFF and engine ON cases are based upon the same planning and tracking system architecture. In particular, main rotor RPM is not used, neither necessary for the engine OFF trajectory planning, nor for the corresponding trajectory tracking, hence simplifying the overall system design. We have also presented what we believe to be a simple trajectory tracking architecture, capable of controlling a small-scale helicopter in autorotation (i.e. engine OFF flight condition). To this end, we have settled for an architecture with only two nested loops, controlling position, velocity, and heading, but without control of vehicle roll and pitch angles. Our methodology is real-time feasible since it allows for, computationally tractable, planning and tracking solutions. In Chapter 5, we evaluate, through several simulations, the flight control control system developed in Chapter 4, using the nonlinear high-order helicopter model of Chapter 2.

4

4.8. APPENDIX A: OPTIMAL TRAJECTORY PLANNING FOR THE ENGINE ON CASE

The TP optimization problem, as in Chapter 3, consists of a cost functional $J(\cdot)$, with contributions from a fixed cost $\Phi(\cdot)$, and a running cost over time $\int_\Omega \Psi(\cdot)dt$, with the independent time variable t defined over the time domain $\Omega = (T_o, T_f)$, where the final time T_f may be free or fixed.

COST FUNCTIONAL

First, we set the fixed cost to T_f, i.e. $\Phi(\cdot) = T_f$, to avoid obtaining trajectories with unreasonably long flight times. Next, we define a general cost functional here, applicable for several types of maneuvers, including takeoff and landing, cruise flight, and hover-to-hover flight. From engineering judgment, we use

$$
\begin{aligned}
J_{ON}(\mathbf{x}(t), \mathbf{u}(t), T_f) \quad &= W_{T_f} T_f + \int_\Omega \Big[(\dot{F}^b_{CG_X})^2 + (\dot{F}^b_{CG_Y})^2 + (\dot{F}^b_{CG_Z})^2 \\
&\quad + (\dot{M}^b_{CG_X})^2 + (\dot{M}^b_{CG_Y})^2 + (\dot{M}^b_{CG_Z})^2 \\
&\quad + W_v v^2 + W_r r^2 \Big] dt
\end{aligned}
\tag{4.27}
$$

For the first six terms in the running cost over time, i.e. the control derivatives, these have been added to: 1) minimize the battery power consumption; and 2) encourage smoother control policies, hence avoiding *bang-bang* type solutions, that might excite undesirable high frequency dynamics or resonances. Next, the term v^2 is added to limit vehicle sideslip[31] flight. Indeed, large sideslip decreases the flight performance, by increasing vehicle drag, increasing roll/yaw coupling, and hence increasing the workload of any feedback controller. The term r^2 has been added to minimize inter-axis coupling. Finally, additional weights, i.e. W_{T_f}, W_v, and W_r, have been added to evaluate various trade-offs within this cost objective.

BOUNDARY AND TRAJECTORY CONSTRAINTS

The boundary conditions are used to set the initial and final (trimmed) flight conditions, and also to set the (initial) and final vehicle accelerations to zero. Having final accelerations equal to zero helps obtaining smooth approaches towards the final waypoint, or alternatively a gentle touch-down during an auto-land. Further the maximum flight time T_f may also be limited.

On the other hand, the trajectory constraints serve several purposes. First, they account for the vehicle's inherent physical and flight envelope limitations, such as bounds on three-dimensional (3D) position, speeds, and attitude. Second, the control inputs to the rigid-body dynamics, i.e. the helicopter forces and moments $\mathbf{F}^b_{CG}$ and $\mathbf{M}^b_{CG}$, are also limited, based upon bounds obtained from simulations using the nonlinear helicopter model. Third, a tail rotor blade tip clearance has been added to avoid ground strike by the tail rotor during a flare maneuver (see our discussion in Chapter 3). Finally, the airflow through the main rotor, given by $V_{rotor} = w + p y_H - q x_H$, has been limited to half the induced velocity in hover v_{ih}, i.e. $V_{rotor} \leq \frac{1}{2} v_{ih}$, as to avoid flight into the chaotic, highly nonlinear, Vortex-Ring-State (VRS) region, refer also to Fig. 2.19 in Chapter 2.

---

[31]Sideslip flight refers to a vehicle moving somewhat sideways as well as forward, relative to the oncoming airflow.

4.9. Appendix B: Design of the inner-loop controller for the engine ON case

As stated in Section 4.4, with four control inputs and twelve measured outputs, the helicopter is heavily under-actuated, which inevitably limits the performance of the tracking system. As mentioned earlier for the engine OFF inner-loop, see also Fig. 4.4, we choose to have the inner-loop track the following four reference signals: 3D body velocities $(u\ v\ w)^\top$, and heading angle ψ. Recall also that the goal of the controller is to minimize the $\mathcal{L}_2$-gain bound γ from the exogenous inputs $\mathbf{w}$ to the controlled outputs $\mathbf{z}$, despite the uncertainty $\boldsymbol{\Delta}(s)$. The various signals are further given as follows: the exogenous inputs $\mathbf{w} = [\mathbf{n}^\top\ \mathbf{r}^\top\ \mathbf{d}^\top]^\top$, the controlled outputs $\mathbf{z} = [\mathbf{z_u}^\top\ \mathbf{z_p}^\top]^\top$, the control inputs $\mathbf{u} = (\theta_0\ \theta_{1c}\ \theta_{1s}\ \theta_{TR})^\top$, the measured outputs $\mathbf{v} = [\mathbf{r}^\top\ \mathbf{y}^\top]^\top$, the reference signals $\mathbf{r} = (u_{TP} + u_d\ v_{TP} + v_d\ w_{TP} + w_d\ \psi_{TP})^\top$, the system outputs $\mathbf{y} = (u\ v\ w\ p\ q\ r\ \phi\ \theta\ \psi)^\top$, the wind disturbance signals (given in inertial frame) $\mathbf{d} = (V_{N_w}\ V_{E_w}\ V_{Z_w})^\top$, and the sensors noise $\mathbf{n}$ (added to the system outputs), see Fig. 4.5. Here the signal $\mathbf{y}$ contains all the available measured output signals, except for the 3D position, since the latter is only of interest for the outer-loop controller.

As mentioned in Section 4.1, we do not use any gain-scheduling philosophy in this Chapter, rather a single LTI plant is used for controller design. The state-space data used to design the inner-loop trajectory tracker is as follows: the state-vector is of dimension nine given by $\mathbf{x} = (u\ v\ w\ p\ q\ r\ \phi\ \theta\ \psi)^\top$, the control input $\mathbf{u}$ (given here-above) is of dimension four, the wind disturbance $\mathbf{d}$ (given here-above) is of dimension three, and the measurements vector $\mathbf{y} = \mathbf{x}$. This LTI model is obtained by linearizing the helicopter nonlinear model of Chapter 2, at a specific trimmed flight condition[32], according to the linearization procedure given in Section 2.4.1, with the resulting state-space data given in Appendix H of Chapter 2. By using the eigenvalues of the $\mathbb{A}$ matrix in the Popov-Belevitch-Hautus (PBH) rank test, we found that the LTI system is both controllable and observable. Simulation results have shown that this nominal LTI was very suitable for the design of controllers, capable of steering the helicopter, in an engine ON situation, from low-speed to medium-speed flight conditions.

Selection of weights

The robust control framework makes use of several user-defined weights, see Fig. 4.5. In this Chapter, these weights have been chosen as follows. The multiplicative uncertainty weight $W_{in}(s)$ is of the form $W_{in}(s) = \text{diag}[w_{in1}(s), w_{in1}(s), w_{in1}(s), w_{in1}(s)]$, with $w_{in1}(s)$ a filter whose magnitude represents the relative uncertainty at each frequency (i.e. the level of uncertainty in the behavior of the helicopter is frequency dependent). Based upon engineering judgment, we choose here for $w_{in1}(s)$ to consider 40% uncertainty at low frequency (DC gain), 100% uncertainty at the filter crossover frequency of 5 Hz (roughly in the range 2.5 to 5 times the anticipated closed-loop bandwidth), and 200% uncertainty at infinite frequency, giving

$$w_{in1}(s) = (2s + 23.75)/(s + 59.37) \tag{4.28}$$

---

[32]The condition corresponds to hover, with engine ON (the main rotor RPM is constant).

Next, the performance weight filter $W_p(s)$ is placed on the (u, v, w, ψ) error signals to reflect the tracking objective for the three body linear velocities and the heading angle. Here $W_p(s)$ is a four-by-four, diagonal, frequency-varying weight $W_p(s) = \text{diag}[w_u(s), w_v(s), w_w(s), w_\psi(s)]$, with each diagonal term defined as a first-order TF $\frac{s/M_P + \omega_B}{s + \omega_B A_{ss}}$. At low frequencies this weighting function should be high in order to keep the error small. Beyond the anticipated bandwidth of the closed-loop system, the tracking error may be released and $W_p(s)$ rolls off [64]. After several controller design cycles, we have settled for

$$
\begin{array}{lll}
\text{For} & w_u(s) & (M_P, \omega_B, A_{ss}) = (2, 2\pi \ rad/s, 0.001) \\
\text{For} & w_v(s) & (M_P, \omega_B, A_{ss}) = (2, 2\pi \ rad/s, 0.001) \\
\text{For} & w_w(s) & (M_P, \omega_B, A_{ss}) = (2, 4\pi \ rad/s, 0.001) \\
\text{For} & w_\psi(s) & (M_P, \omega_B, A_{ss}) = (2, 4\pi \ rad/s, 0.001)
\end{array}
\tag{4.29}
$$

This means that a steady-state tracking error of 0.1% with respect to the normalized filter input is allowed, whereas the tracking bandwidth of these filters is set below the 5 Hz actuators bandwidth (actuator data is reported in Table 2.1 of Chapter 2). Now, tracking should not be achieved at the cost of too high control effort. Therefore, both actuator deflection (i.e. amplitude) and rate are penalized through weight $W_u(s) = \text{diag}[w_{act}(s), w_{act}(s), w_{act}(s), w_{act}(s)]$, with

$$
w_{act}(s) = 10^n \left(\frac{s + \omega_1}{s + \omega_2} \right)^n \quad \text{with} \quad (n, \omega_1, \omega_2) = (3, 6\pi \ rad/s, 60\pi \ rad/s) \tag{4.30}
$$

Next, a noise weight $W_n(s)$ is defined to represent the actual noise levels associated with each sensor, and is defined as a nine-by-nine, constant, diagonal scaling matrix described as follows (given here in its unscaled form)

$$
W_n(s) = \text{diag}[0.01 \ m/s, 0.01 \ m/s, 0.01 \ m/s, 3\pi/180 \ rad/s, 3\pi/180 \ rad/s, 3\pi/180 \ rad/s,
$$
$$
\pi/180 \ rad, \pi/180 \ rad, 3\pi/180 \ rad]
\tag{4.31}
$$

Finally, a wind disturbance weight $W_d(s) = \text{diag}[w_{d_N}(s), w_{d_E}(s), w_{d_D}(s)]$ is added to simulate the frequency content of the NASA Dryden atmospheric wind model [71], and is identical to the one used in the engine OFF case, see Eq. (4.24).

Controller synthesis and analysis

For the D-K iteration [72], we obtain after four iterations a stable controller $K(s)$ of order 38, using 0th order (constant) D_s-scalings. The controller is further reduced to 30th order, after balancing and Hankel-norm model reduction [73], without any significant effect on closed-loop robustness/performance. In Fig. 4.15, we visualize the relevant TFs, defined in the previous section, with the bandwidths for the three main TFs given in Table 4.1.

In particular, we see that the bandwidth of $|T_i(s)|$ is high enough, to stabilize the plant, i.e. above 2.1 rad/s, see our discussion in Section 2.4.2. Also the closed-loop disturbance rejection, given in Fig. 4.9, shows good attenuation of horizontal wind disturbances, even though the vertical disturbance attenuation could potentially be improved (approximately -20 dB at a frequency of 2π rad/s). We also see that the S_o is not well-behaved, since it
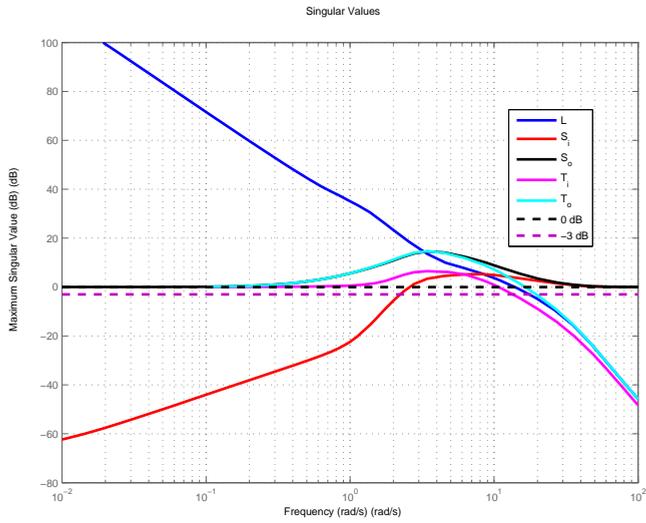
Figure 4.15: Singular values of $L(s)$, $S_i(s)$, $S_o(s)$, $T_i(s)$, and $T_o(s)$, of the inner-loop trajectory tracker (Engine ON case).
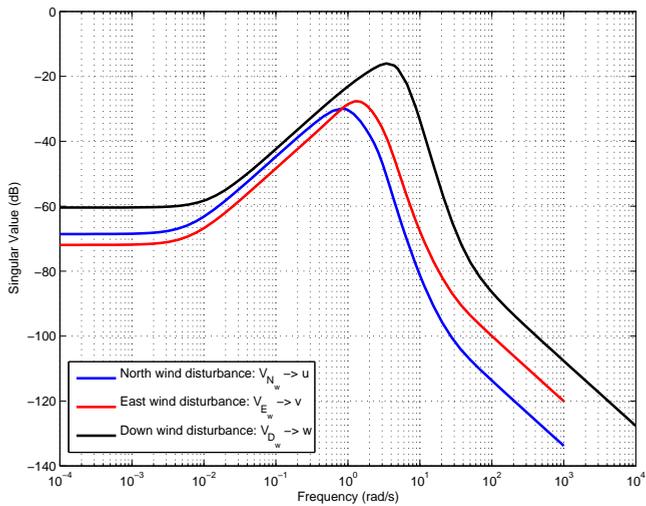
Figure 4.16: Closed-Loop wind disturbance rejection, for North-East-Down (NED) winds, of the inner-loop trajectory tracker (Engine ON case).

remains high at both low- and high-frequencies. This can be explained as follows. The output loop $L(s)$ is a 9x9 matrix, with 4 singular-values having very high values (for low-frequencies). These high singular-values correspond to the 4 controlled channels. Since our helicopter is under-actuated, the remaining 5 singular-values are all very low (for all frequencies). Thus, inverting $(I + L(s))$ to get S_o results in maximum singular-values which are most often close to 0 dB.

Next, RS and RP are visualized in Fig. 4.10 and Fig. 4.11. We can see that lower and upper bounds are indistinguishable. We observe that the primordial RS is guaranteed (i.e. a maximum value below 1). On the other hand, we see that RP is not met (i.e. a maximum value well above one). Again, this may potentially be due to the fact that the chosen uncertainty $\Delta(s)$, shown in Fig. 4.5, is not realistic. If robust performance specifications need to be met, then this could potentially be done by lowering the amount of model input uncertainty, and/or by relaxing some of the assumptions made during the various weights selection. However, from our experience, this will likely compromise the closed-loop performance of the controller, once tested upon the nonlinear system.
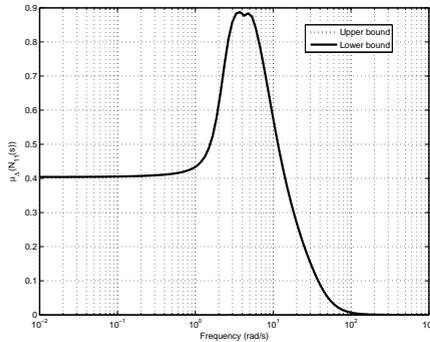
Figure 4.17: Robust Stability of the inner-loop trajectory tracker (Engine ON case).
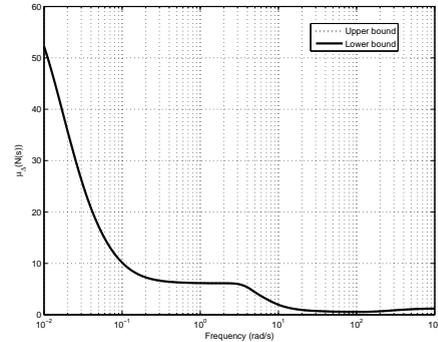
Figure 4.18: Robust Performance of the inner-loop trajectory tracker (Engine ON case).

4.10. Appendix C: Design of the outer-loop controller for the engine ON case

Again, the design approach is related to the well-known time-scale separation principle [63] between slow and fast dynamics of a dynamical system. As mentioned earlier, see Fig. 4.3, we choose to have the helicopter track the following three reference signals: 3D inertial[33] positions $(x_N \; x_E \; x_Z)^\top$. Recall also that the goal of the controller is to minimize the $\mathcal{L}_2$-gain bound γ from the exogenous inputs $\mathbf{w}$ to the controlled outputs $\mathbf{z}$, despite the uncertainty $\Delta(s)$. The various signals are further given as follows: the exogenous inputs $\mathbf{w} = [\mathbf{n}^\top \; \mathbf{r}^\top]^\top$, the controlled outputs $\mathbf{z} = [\mathbf{z_u}^\top \; \mathbf{z_p}^\top]^\top$, the control inputs $\mathbf{u} = (V_N \; V_E \; V_Z)_d^\top$, the measured outputs $\mathbf{v} = [\mathbf{r}^\top \; \mathbf{y}^\top]^\top$, the reference signals $\mathbf{r} = (x_N \; x_E \; x_Z)_{TP}^\top$, the system outputs $\mathbf{y} = (x_N \; x_E \; x_Z)^\top$, and the sensors noise $\mathbf{n}$ (added to the system outputs), see Fig. 4.5.

---

[33]Which is equivalent to North-East-Down (NED) position in our flight dynamics model.

Here the outer-loop does not include disturbance signals, since the wind has already been accounted for, within the inner-loop control structure.

As discussed in Section 4.1, a single LTI plant is used for controller design. The state-space data used to design the outer-loop trajectory tracker is obtained as follows. An LTI dynamical system can be formed by connecting the nominal LTI model, used for the inner-loop TT, with its inner-loop controller, and subsequently adding a set of integrators to generate the 3D inertial positions $(x_N \; x_E \; x_Z)^\top$. This manipulation is readily done in MATLAB, and results in the nominal LTI model needed to design the outer-loop position controller. In our case, we obtain a three-by-three input-output system, with a state-vector of dimension 55. Next a minimum realization is obtained, resulting in a state-vector of dimension 42 (the LTI model is too big to be added to the Appendix). Note that here too scalings need to be applied. Further, and except for three poles at the origin (corresponding to the integration of the 3D velocities), all other eigenvalues of the $\mathbb{A}$ matrix are stable and well damped, implying easier controller design. Again, by using the eigenvalues of the $\mathbb{A}$ matrix in the PBH rank test, we found that the LTI system is both controllable and observable.

The design philosophy for the μ outer-loop TT parallels that of the inner-loop.

SELECTION OF WEIGHTS

The multiplicative uncertainty weight $W_{in}(s)$ is of the form $W_{in}(s) = \mathrm{diag}[w_{in1}(s), w_{in1}(s), w_{in1}(s)]$, with $w_{in1}(s)$ identical to Eq. (4.28). Obviously, this choice of the uncertainty weight $W_{in}(s)$ is somewhat arbitrary. This said, the purpose is here to add some robustness to the closed-loop system. The performance weight $W_p(s)$ is placed on the (x_N, x_E, x_Z) error signals to reflect the tracking objective for the inertial position (which as a reminder is equivalent to NED position in our model). Here, $W_p(s)$ is a three-by-three diagonal, frequency-varying weight. At low frequencies this weighting function should be high in order to keep the error small. Beyond the anticipated bandwidth of the position tracking system, this error may be released and $W_p(s)$ rolls off. We have $W_p(s) = \mathrm{diag}[w_{x_N}(s), w_{x_E}(s), w_{x_Z}(s)]$, with each diagonal term defined as a first-order transfer function $\frac{s/M_P + \omega_B}{s + \omega_B A_{ss}}$. After several controller design cycles, we have settled for

$$
\begin{aligned}
&\text{For} \quad w_{x_N}(s) \quad (M_P, \omega_B, A_{ss}) = (2, 0.2\pi \; rad/s, 0.001) \\
&\text{For} \quad w_{x_E}(s) \quad (M_P, \omega_B, A_{ss}) = (2, 0.2\pi \; rad/s, 0.001) \\
&\text{For} \quad w_{x_Z}(s) \quad (M_P, \omega_B, A_{ss}) = (2, 0.4\pi \; rad/s, 0.001)
\end{aligned}
\tag{4.32}
$$

This means that a steady-state tracking error of 0.1% with respect to the normalized input is allowed. Further, the filter bandwidths are adjusted to be ten times smaller than the $W_p(s)$ filter bandwidths for the inner-loop case.

Next, tracking should not be achieved at the cost of too high control effort (i.e. resulting in much too large velocity setpoints $\mathbf{u} = (V_N \; V_E \; V_Z)_d^\top$ for the inner-loop). This means that both inertial velocities and inertial accelerations should be penalized, through weight $W_u(s) = \mathrm{diag}[w_{act}(s), w_{act}(s), w_{act}(s)]$, with $w_{act}(s)$ identical to the one chosen for the inner-loop, with engine ON. Again, this choice may be interpreted as rather arbitrary, since here $W_u(s)$ is assigned to the inner-loop setpoints $\mathbf{u} = (V_N \; V_E \; V_Z)_d^\top$, whereas for the design of

the inner-loop controller, $W_u(s)$ was assigned to the actuators. Hence, potentially better choices for $W_u(s)$ may exist, although the one selected here provided satisfactory results. Finally, a noise weight $W_n(s)$ is also defined to scale the normalized position measurement noise. The sensor noise model is defined here as a three-by-three, constant, diagonal scaling matrix described by (given here in its unscaled form)

$$W_n(s) = \text{diag}[0.1\ m, 0.1\ m, 0.1\ m] \tag{4.33}$$

CONTROLLER SYNTHESIS AND ANALYSIS

For the D-K iteration [72], we obtain after four iterations a stable controller $K(s)$ of order 63, using 6th order $D_s(s)$-scalings. The controller is further reduced to 30th order (using the same technique as for the inner-loop), without any effect on closed-loop robustness/performance. In Fig. 4.19, we visualize the relevant TFs (we see that $S_i(s) = S_o(s)$, and $T_i(s) = T_o(s)$), with the bandwidths for the three TFs given in Table 4.1. In particular, we see that the bandwidth of $|T_i(s)|$ is ten times lower its inner-loop counterpart, which is good since we do not want both controllers to start interacting with each other. Further, RS is shown in Fig. 4.20, whereas RP is pictured in Fig. 4.21. Again, we observe that RP is not achieved, but RS is guaranteed.
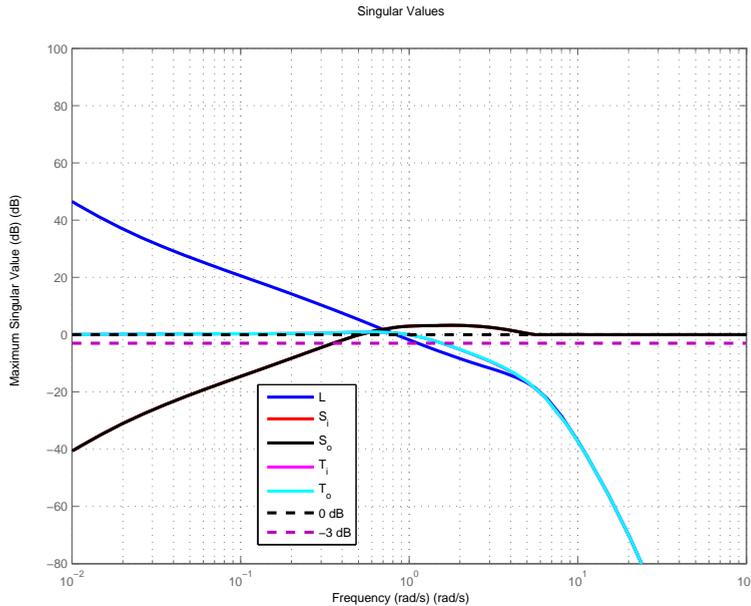
Figure 4.19: Singular values of $L(s)$, $S_i(s)$, $S_o(s)$, $T_i(s)$, and $T_o(s)$, of the outer-loop trajectory tracker (Engine ON case).
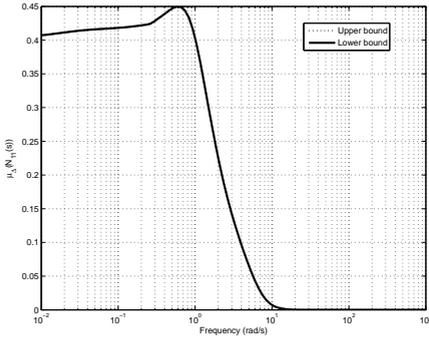
Figure 4.20: Robust Stability of the outer-loop trajectory tracker (Engine ON case).
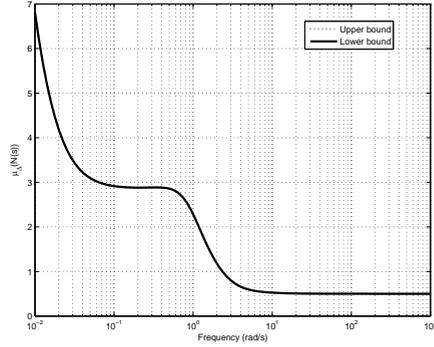
Figure 4.21: Robust Performance of the outer-loop trajectory tracker (Engine ON case).

Adapting the engine ON outer-loop controller

For the case of an engine ON automatic landing, and when close to the ground, it is crucial to keep the reference velocities as small as possible. To this end, we adapt the outer-loop controller as follows: the position tracking is switched-off, i.e. the values for $(u\ v\ w)_d^\top$ are set to zero, once the helicopter height descends below a predefined threshold (keeping only velocity and heading tracking). This helps lowering the final (touch-down) values of the 3D velocities, by giving more time to the velocity deceleration process.

4.11. Appendix D: Maximum roll (or pitch) angle for safe (i.e. successful) landing

The landing gear of our Align T-REX small-scale helicopter, see Fig. 4.22, has been re-designed in order to have $OS = OG = h_1$. In this figure the half-ellipse, depicted in blue, represents the landing gear frame, whereas the yellow bar represents a main rotor blade.

From Fig. 4.22, since we have $\widehat{SOG} = \frac{\pi}{2}$ then we also obtain $\widehat{OSG} = \mu_1 = \frac{\pi}{4}$ since $OS = OG$. Further, due to the moments created by the weight force $\mathbf{W}$ and the ground reaction force $\mathbf{R}$, the helicopter will tilt-over to the right and hit the ground at landing, whenever the Center of Gravity G moves "to the right" of point S. When G is exactly above S, and since $\mu_1 = \frac{\pi}{4}$, we can compute the vehicle maximum roll angle (for safe landing) as: $\phi = \frac{\pi}{4} = 45°$.

This said, a more stringent limiting factor may exist, due to a possible ground strike by a main rotor blade as depicted in Fig. 4.22. Obviously, the vehicle roll angle for safe landing will also depend upon the blade flapping angle β_{bl}. From the geometry depicted in Fig. 4.22, and using triangle identities, we can derive the roll angle at which a blade ground-strike will occur, as follows

$$\phi = \pi - (\mu_2 + \mu_4) \tag{4.34}$$

with

$$\mu_2 = \cos^{-1}(\frac{h_1}{SH}) \tag{4.35}$$

and since $HT = R_{rot}$, we have

$$\mu_4 = \cos^{-1}(\frac{R_{rot}^2 - (ST^2 + SH^2)}{-2.ST.SH}) \tag{4.36}$$

with the distances SH and ST defined by

$$\begin{aligned} SH &= \sqrt{(h_1 + GH)^2 + h_1^2} \\ ST &= \sqrt{R_{rot}^2 + SH^2 - 2.R_{rot}.SH.\cos\mu_3} \end{aligned} \tag{4.37}$$

and angle μ_3 obtained as follows

$$\begin{aligned} \mu_3 &= \frac{\pi}{2} - (\mu_5 + |\beta_{bl}|) \\ \mu_5 &= \pi - (\frac{\pi}{2} + \mu_2) \end{aligned} \tag{4.38}$$

with μ_2 computed using Eq. (4.35) and the distance SH from Eq. (4.37).

From engine OFF (autorotation) flight and landing simulations, we found variations between -1°down-flap and +4°up-flap for the blade flap angle β_{bl}. Now, using for our helicopter $R_{rot} = 0.9$, $h_1 = 0.25$, $GH = 0.23$, Table 4.2 gives the maximum vehicle roll angle ϕ for safe landing, as a function of blade down-flap angle β_{bl}. We see that a -1°down-flap

Table 4.2: Maximum vehicle roll angle ϕ for safe landing, as a function of blade down-flap angle β_{bl}.

Flap angle β_{bl} (°)	0	-1	-5	-10	-15
Roll angle ϕ (°)	36.4	35.5	31.8	27.0	21.7

Figure 4.22: Maximum vehicle roll angle ϕ for safe landing, for the case of a negative blade flap angle β_{bl}.

would result in a 35.5° maximum vehicle roll angle, hence way above the 10° roll angle defined in the requirement for safe landing in Definition 4.2 of Chapter 4. Even for much larger down-flap angles, e.g. -15° possibly due to the ground impact, we see that the maximum allowable roll angle is still higher than the 10° specification. A similar reasoning can also be applied to the pitch axis which, based upon the shape of the landing gear, gives comparable results to the ones outlined for the roll axis.

4.12. Appendix E: Proof of Lemma 1

From Eq. (2.4) and Eq. (2.8) we obtain

$$
\begin{pmatrix} u \\ v \\ w \end{pmatrix}^b = \left(\begin{array}{l} \dot{x}_N \cos\theta\cos\psi + \dot{x}_E \cos\theta\sin\psi \\ -\dot{x}_N \sin\psi\cos\phi + \dot{x}_N \sin\theta\sin\phi\cos\psi \\ \dot{x}_N \sin\phi\sin\psi + \dot{x}_N \sin\theta\cos\psi\cos\phi \end{array} \right.
$$
$$
\begin{array}{l} -\dot{x}_Z \sin\theta \\ +\dot{x}_E \cos\psi\cos\phi + \dot{x}_E \sin\theta\sin\phi\sin\psi + \dot{x}_Z \sin\phi\cos\theta \\ \left. -\dot{x}_E \sin\phi\cos\psi + \dot{x}_E \sin\theta\sin\psi\cos\phi + \dot{x}_Z \cos\phi\cos\theta \end{array} \right)
$$

(4.39)

Now, inverting Eq. (2.7) we get

$$
\begin{pmatrix} p \\ q \\ r \end{pmatrix}^b = \begin{pmatrix} \dot{\phi} - \dot{\psi}\sin\theta \\ \dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta \\ -\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta \end{pmatrix}
$$

(4.40)

From Eq. (2.5) and Eq. (4.39), and taking the derivative of Eq. (4.39), we obtain for the three force inputs $\mathbf{F}_{CG}^b = (F_{CG_X}^b \; F_{CG_Y}^b \; F_{CG_Z}^b)^\top$

$$
\mathbf{F}_{CG}^b = \begin{pmatrix} F_{CG_X} \\ F_{CG_y} \\ F_{CG_z} \end{pmatrix}^b = m_V \left(\begin{array}{l} g.\sin\theta + \ddot{x}_N \cos\theta\cos\psi \\ -g.\sin\phi\cos\theta - \ddot{x}_N(\sin\psi\cos\phi - \sin\theta\sin\phi\cos\psi) \\ -g.\cos\phi\cos\theta + \ddot{x}_N(\sin\phi\sin\psi + \sin\theta\cos\psi\cos\phi) \end{array} \right.
$$
$$
\begin{array}{l} +\ddot{x}_E \cos\theta\sin\psi - \ddot{x}_Z \sin\theta \\ +\ddot{x}_E(\cos\psi\cos\phi + \sin\theta\sin\phi\sin\psi) + \ddot{x}_Z \sin\phi\cos\theta \\ \left. -\ddot{x}_E(\sin\phi\cos\psi - \sin\theta\sin\psi\cos\phi) + \ddot{x}_Z \cos\phi\cos\theta \end{array} \right)
$$

(4.41)

Finally, from Eq. (2.6) and Eq. (4.40), and taking the derivative of Eq. (4.40), we can express the three moments inputs $\mathbf{M}_{CG}^b = (M_{CG_X}^b \; M_{CG_Y}^b \; M_{CG_Z}^b)^\top$ as

$$
\begin{aligned}
M_{CG_X}^b = {} & E(\dot{\psi}(\dot{\phi}\cos\theta\sin\phi + \dot{\theta}\cos\phi\sin\theta) + \ddot{\theta}\sin\phi \\
& +\dot{\phi}\dot{\theta}\cos\phi - \ddot{\psi}\cos\phi\cos\theta) - A(\ddot{\psi}\sin\theta - \ddot{\phi} + \dot{\psi}\dot{\theta}\cos\theta) \\
& -F(\dot{\psi}(\dot{\phi}\cos\phi\cos\theta - \dot{\theta}\sin\phi\sin\theta) \\
& +\ddot{\theta}\cos\phi - \dot{\phi}\dot{\theta}\sin\phi + \ddot{\psi}\cos\theta\sin\phi) \\
& +(\dot{\theta}\sin\phi - \dot{\psi}\cos\phi\cos\theta)(B(\dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi) \\
& -F(\dot{\phi} - \dot{\psi}\sin\theta) + D(\dot{\theta}\sin\phi - \dot{\psi}\cos\phi\cos\theta)) \\
& -(\dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi)(E(\dot{\phi} - \dot{\psi}\sin\theta) \\
& +C(\dot{\theta}\sin\phi - \dot{\psi}\cos\phi\cos\theta) + D(\dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi))
\end{aligned}
$$

(4.42)

$$
\begin{aligned}
M_{CG_Y}^b = {} & D(\dot{\psi}(\dot{\phi}\cos\theta\sin\phi + \dot{\theta}\cos\phi\sin\theta) + \ddot{\theta}\sin\phi \\
& +\dot{\phi}\dot{\theta}\cos\phi - \ddot{\psi}\cos\phi\cos\theta) + B(\dot{\psi}(\dot{\phi}\cos\phi\cos\theta - \dot{\theta}\sin\phi\sin\theta) \\
& +\ddot{\theta}\cos\phi - \dot{\phi}\dot{\theta}\sin\phi + \ddot{\psi}\cos\theta\sin\phi) + F(\ddot{\psi}\sin\theta - \ddot{\phi} + \dot{\psi}\dot{\theta}\cos\theta) \\
& +(\dot{\phi} - \dot{\psi}\sin\theta)(E(\dot{\phi} - \dot{\psi}\sin\theta) + C(\dot{\theta}\sin\phi - \dot{\psi}\cos\phi\cos\theta) \\
& +D(\dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi)) - (\dot{\theta}\sin\phi \\
& -\dot{\psi}\cos\phi\cos\theta)(A(\dot{\phi} - \dot{\psi}\sin\theta) + E(\dot{\theta}\sin\phi - \dot{\psi}\cos\phi\cos\theta) \\
& -F(\dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi))
\end{aligned}
$$

(4.43)

$$
\begin{aligned}
M^b_{CG_z} = {} & E(\ddot{\psi}\sin\theta - \ddot{\phi} + \dot{\psi}\dot{\theta}\cos\theta) - C(\dot{\psi}(\dot{\phi}\cos\theta\sin\phi \\
& + \dot{\theta}\cos\phi\sin\theta) + \ddot{\theta}\sin\phi + \dot{\phi}\dot{\theta}\cos\phi - \ddot{\psi}\cos\phi\cos\theta) \\
& - D(\dot{\psi}(\dot{\phi}\cos\phi\cos\theta - \dot{\theta}\sin\phi\sin\theta) + \ddot{\theta}\cos\phi - \dot{\phi}\dot{\theta}\sin\phi \\
& + \ddot{\psi}\cos\theta\sin\phi) + (\dot{\phi} - \dot{\psi}\sin\theta)(B(\dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi) \\
& - F(\dot{\phi} - \dot{\psi}\sin\theta) + D(\dot{\theta}\sin\phi - \dot{\psi}\cos\phi\cos\theta)) \\
& - (\dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi)(A(\dot{\phi} - \dot{\psi}\sin\theta) \\
& + E(\dot{\theta}\sin\phi - \dot{\psi}\cos\phi\cos\theta) - F(\dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi))
\end{aligned}
\tag{4.44}
$$

REFERENCES

[1] S. Taamallah, *Low-order modeling for a small-scale flybarless helicopter uav, a grey-box time-domain approach,* in *AIAA Atmospheric Flight Mechanics Conf.* (2012).

[2] S. Taamallah, *Optimal autorotation with obstacle avoidance for a small-scale flybarless helicopter uav,* in *AIAA Guidance, Navigation and Control Conf.* (2012).

[3] S. Taamallah, X. Bombois, and P. M. J. Van den Hof, *Optimal control for power-off landing of a small-scale helicopter a pseudospectral approach,* in *Am. Control Conf.* (2012).

[4] S. Taamallah, *A qualitative introduction to the vortex-ring-state, autorotation, and optimal autorotation,* in *Europ. Rotorcraft Forum* (2010).

[5] D. Santamaria, A. Viguria, M. Bejar, K. Kondak, and A. Ollero, *Towards autonomous autorotation landing for small size unmanned helicopters,* J. Intell. Robot Syst. **69**, 171 (2013).

[6] P. Abbeel, A. Coates, T. Hunter, and A. Y. Ng, *Autonomous autorotation of an rc helicopter,* in *11th Int. Symposium on Experimental Robotics (ISER)* (2008).

[7] J. Holsten, S. Loechelt, and W. Alles, *Autonomous autorotation flights of helicopter uavs to known landing sites,* in *Am. Helicopter Soc.* (2010).

[8] T. Yomchinda, J. F. Horn, and J. W. Langelaan, *Autonomous control and path planning for autorotation of unmanned helicopters,* in *Am. Helicopter Soc.* (2012).

[9] Z. Sunberg and J. Rogers, *A fuzzy logic-based controller for helicopter autorotation,* in *AIAA Aerospace Sciences Meeting and Exhibit* (2013).

[10] M. Fliess, *Generalized controller canonical forms for linear and nonlinear dynamics,* IEEE Trans. Autom. Control **35**, 994 (1990).

[11] M. Fliess, J. Levine, P. Martin, and P. Rouchon, *Sur les systemes non lineaires differentiellement plats,* C.R. Acad. Sci. Paris **315**, 619 (1992).

[12] M. Fliess, J. Levine, P. Martin, and P. Rouchon, *Flatness and defect of nonlinear systems: Introductory theory and examples,* Int. J. of Control **61**, 1327 (1995).

[13] H. Sira-Ramirez and S. K. Agrawal, *Differentially Flat Systems* (Marcel Dekker Inc., 2004).

[14] J. Levine, *Analysis and Control of Nonlinear Systems: A Flatness-Based Approach* (Springer, 2009).

[15] I. M. Ross and F. Fahroo, *A unified computational framework for real-time optimal control,* in *IEEE Conf. on Decision and Control* (2003).

[16] P. Martin, R. M. Murray, and P. Rouchon, *Flat Systems, Equivalence And Trajectory Generation*, Tech. Rep. (CDS, California Institute of Technology, 2003).

4

[17] R. M. Murray, J. Hauser, A. Jadbabaie, M. B. Milam, N. Petit, W. B. Dunbar, and R. Franz, *Online control customization via optimization-based control. In T. Samad and G. Balas, Editors, Software-Enabled Control: Information Technology for Dynamical Systems* (IEEE Press, 2003).

[18] G. Zames, *On the input-output stability of time-varying nonlinear feedback systems-part i: Conditions derived using concepts of loop gain, conicity, and positivity,* IEEE Trans. Autom. Control **AC-11**, 228 (1966).

[19] C. A. Desoer and M. Vidyasagar, *Feedback Systems: Input-Output Properties* (Academic Press, New York, 1975).

[20] D. H. Jacobson and D. Q. Mayne, *Differential Dynamic Programming* (Elsevier, 1970).

[21] H. Sira-Ramirez, R. Castro-Linares, and E. Liceaga-Castro, *A liouvillian systems approach for the trajectory planning-based control of helicopter models,* Int. J. Of Robust And Nonlinear Control **10**, 301 (2000).

[22] M. Fliess, J. Levine, P. Martin, and P. Rouchon, *A lie-backlund approach to equivalence and flatness of nonlinear systems,* IEEE Trans. Autom. Control **44**, 922 (1999).

[23] W. C. Lu, L. Duan, F. B. Hsiao, and F. M. Camino, *Neural guidance control for aircraft based on differential flatness,* AIAA J. of Guidance, Control, and Dynamics **31**, 892 (2008).

[24] P. Martin, S. Devasia, and B. Paden, *A different look at output tracking: Control of a vtol aircraft,* Automatica **32**, 101 (1996).

[25] M. Saeki and Y. Sakaue, *Flight control design for a nonlinear non-minimum phase vtol aircraft via two-step linearization,* in *IEEE Conf. on Decision and Control* (2001).

[26] I. M. Ross and F. Fahroo, *Pseudospectral methods for optimal motion planning of differentially flat systems,* IEEE Trans. Autom. Control **49**, 1410 (2004).

[27] I. M. Ross and F. Fahroo, *Issues in the real time computation of optimal control,* Mathematical and Computer Modelling **43**, 1172 (2006).

[28] M. J. van Nieuwstadt and R. M. Murray, *Real time trajectory generation for differentially flat systems,* Int. J. of Robust and Nonlinear Control **18**, 995 (1998).

[29] N. Faiz, S. K. Agrawal, and R. M. Murray, *Trajectory planning of differentially flat systems with dynamics and inequalities,* AIAA J. of Guidance, Control, and Dynamics **24**, 219 (2001).

[30] M. B. Milam, K. Mushambi, and R. M. Murray, *A new computational approach to real-time trajectory generation for constrained mechanical systems,* in *IEEE Conf. on Decision and Control* (2000).

[31] M. B. Milam, R. Franz, and R. M. Murray, *Real-time constrained trajectory generation applied to a flight control experiment,* in *IFAC World Conf.* (2002).

4

[32] R. Murray, M. Rathinam, and W. Sluis, *Differential flatness of mechanical control systems: A catalog of prototype systems,* in *ASME Int. Mechanical Engineering Congress* (1995).

[33] M. Rathinam, *Differentially Flat Nonlinear Control Systems*, Ph.D. thesis, California Institute of Technology (1997).

[34] V. N. Chetverikov, *New flatness conditions for control systems,* in *IFAC Nonlinear Control Systems* (2001).

[35] P. Martin, *Contribution a l'Etude des Systemes Differentiellement Plats*, Ph.D. thesis, Ecole des Mines de Paris (1992).

[36] P. Martin, *Aircraft control using flatness,* in *Proc. CESA Lille* (1996).

[37] J. Hauser and R. Hindman, *Aggressive flight maneuvers,* in *IEEE Conf. on Decision and Control* (1997).

[38] F. Cazaurang, R. Bergeon, and L. Lavigne, *Modelling of longitudinal disturbed aircraft model by flatness approach,* in *AIAA Guidance Navigation and Control Conf.* (2003).

[39] A. Ailon, *Optimal path and tracking control of an autonomous vtol aircraft,* in *Mediterranean Conf. on Control and Automation* (2006).

[40] M. van Nieuwstadt and R. M. Murray, *Outer flatness: Trajectory generation for a model helicopter,* in *Europ. Control Conf.* (1997).

[41] T. J. Koo and S. Sastry, *Differential flatness based full authority helicopter control design,* in *IEEE Conf. on Decision and Control* (1999).

[42] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cook, *A prototype of an autonomous controller for a quadrotor uav,* in *Europ. Control Conf.* (2007).

[43] Y. Bouktir, M. Haddad, and T. Chettibi, *Trajectory planning for a quadrotor helicopter,* in *Mediterranean Conf. on Control and Automation* (2008).

[44] D. Mellinger and V. Kumar, *Minimum snap trajectory generation and control for quadrotors,* in *International Conf. on Robotics and Automation* (2011).

[45] S. Formentin and M. Lovera, *Flatness-based control of a quadrotor helicopter via feedforward linearization,* in *IEEE Conf. on Decision and Control* (2011).

[46] A. Chamseddine, Z. Youmin, C. A. Rabbath, C. Join, and D. Theilliol, *Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle,* IEEE Trans. on Aerospace and Electronic Systems **48**, 2832 (2012).

[47] D. Desiderio and M. Lovera, *Flatness-based guidance for planetary landing,* in *Am. Control Conf.* (2010).

[48] V. Morio, F. Cazaurang, and P. Vernis, *Flatness-based hypersonic reentry guidance of a lifting-body vehicle,* Control Engineering Practice **17**, 588 (2009).

[49] J. Levine and D. Nguyen, *Flat output characterization for linear systems using polynomial matrices,* Systems & Control Letters **48**, 69 (2003).

[50] D. Henrion and J. B. Lasserre, *Lmis for constrained polynomial interpolation with application in trajectory planning,* Systems & Control Letters **55**, 473 (2006).

[51] J. D. Caigny, B. Demeulenaere, J. Swevers, and J. de Schutter, *Optimal design of spline-based feedforward for trajectory tracking,* in *Am. Control Conf.* (2007).

[52] F. Suryawan, J. A. D. Dona, and M. M. Seron, *Methods for trajectory generation in a magnetic-levitation system under constraints,* in *Mediterranean Conf. on Control and Automation* (2010).

[53] F. Suryawan, J. A. D. Dona, and M. M. Seron, *On splines and polynomial tools for constrained motion planning,* in *Mediterranean Conf. on Control and Automation* (2010).

[54] R. Dai, *B-splines based optimal control solution,* in *AIAA Guidance, Navigation and Control Conf.* (2010).

[55] F. Suryawan, J. A. D. Dona, and M. M. Seron, *Flatness-based minimum-time trajectory generation for constrained linear systems using b-splines,* in *IFAC World Congress* (2011).

[56] D. Bertsekas, *Nonlinear Programming - Second Edition* (Athena Scientific, Belmont, MA, 1999).

[57] J. Nocedal and S. Wright, *Numerical Optimization* (Springer, NY, 2000).

[58] A. V. Rao, *A survey of numerical methods for optimal control,* in *Am. Astronautical Soc.* (2009).

[59] R. J. Vanderbei and D. F. Shanno, *An interior-point algorithm for nonconvex nonlinear programming,* Computational Optimization and Applications **13**, 231 (1999).

[60] R. H. Byrd, M. E. Hribar, and J. Nocedal, *An interior point algorithm for large-scale nonlinear programming,* SIAM J. on Control and Optimization **9**, 877 (1999).

[61] L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders, *Large-Scale PDE-Constrained Optimization* (Springer, Berlin, 2003).

[62] R. H. Byrd, J. Nocedal, and R. A. Waltz, *Knitro: An integrated package for nonlinear optimization,* Large-Scale Nonlinear Optimization, Springer-Verlag , 35 (2006).

[63] M. D. Ardema and N. Rajan, *Separation of time scales in aircraft trajectory optimization,* AIAA J. of Guidance, Control, and Dynamics **8**, 275 (1985).

4

[64] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design, 2nd Ed.* (Wiley-Interscience, Great Britain, 2005).

[65] J. C. Doyle, *Structured uncertainty in control system design,* in *IEEE Conf. on Decision and Control* (1985).

[66] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control* (Prentice Hall, New Jersey, 1996).

[67] K. J. Astrom and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers* (Princeton University Press, Princeton, NJ, 2008).

[68] D. W. Gu, P. H. Petkov, and M. M. Konstantinov, *Robust Control Design with Matlab* (Springer, London, 2005).

[69] A. A. G. Macintosh, *Trajectory Tracking Control for a Small-Scale Helicopter during Autorotation*, Tech. Rep. (Master Thesis, Delft University of Technology, 2014).

[70] *http://gemiddeldgezien.nl/meer-gemiddelden/176-gemiddelde-windkracht-nederland*.

[71] Anonymous, *Flying Qualities of Piloted Airplanes*, Tech. Rep. MIL-F-8785C (United States DoD, 1980).

[72] G. J. Balas, J. C. Doyle, K. Glover, A. K. Packard, and R. S. Smith, *μ Analysis and Synthesis Toolbox: User's Guide* (The MathWorks Inc., 1991).

[73] K. Glover, *All optimal hankel-norm approximations of linear multivariable systems and their $\mathcal{L}_\infty$-error bounds,* Int. J. of Control **39**, 1115 (1984).

4

5

ON-LINE TRAJECTORY PLANNING AND TRACKING: SIMULATION RESULTS

The helicopter approaches closer than any other vehicle to fulfillment of mankind's ancient dreams of the flying horse and the magic carpet.

Igor I. Sikorsky
Designed the world's first mass-produced helicopter in 1942

In Chapter 4, we presented a combined Trajectory Planning (TP) and Trajectory Tracking (TT) system, having on-line computational tractability. In Chapter 5, we demonstrate—using the high-fidelity, high-order, nonlinear helicopter simulation of Chapter 2—the first, real-time feasible, model-based TP and TT system, for the case of a small-scale helicopter UAV with an engine OFF condition (i.e. autorotation). To better illustrate the various challenges encountered when designing a planning and tracking system for the engine OFF condition, a comparison with two engine ON automated flight maneuvers is also provided.

5.1. INTRODUCTION

I N this Chapter, we evaluate the combined Trajectory Planning (TP) and Trajectory Track-ing (TT) functionalities, developed in Chapter 4. These are tested on the helicopter, high-fidelity, High-Order Model (HOM) developed in Chapter 2. Five test cases are presented, two with engine ON, and three with engine OFF (autorotation), starting from various initial conditions. The modeled small-scale UAV is the instrumented Remote-Controlled (RC) Align T-REX helicopter, used also in Chapter 2, and belonging to the flybarless two-bladed main rotor class. This vehicle has a total mass of 7.75 kg, a main rotor radius of 0.9 m, a main rotor nominal angular velocity of 1350 RPM, a NACA 0015 main rotor airfoil, and an induced velocity in hover given by v_{ih} = 3.5 m/s (see also Table 2.1, of Chapter 2, for additional helicopter parameters).

The two engine ON test cases are included to illustrate that the Flight Control System (FCS) framework, presented in the Appendices of Chapter 4, allows for a variety of auto-mated flight maneuvers. With the engine ON, we demonstrate an automatic landing, and a cruise-to-hover maneuver. The first engine ON test case starts from an initial flight condi-tion which is identical to the one used when deriving the nominal LTI model, used for TT design (i.e. helicopter in hover). The second test case starts from an initial condition which is far away from the operating condition used to derive this LTI model. Both test cases are set in an ideal environment, i.e. a noise-free and disturbance-free environment.

The three engine OFF test cases are set to demonstrate the automatic autorotation land-ing capability. Here too, the first engine OFF test case is set to evaluate the FCS perfor-mance for an initial flight condition which is identical to the flight condition used to derive the nominal LTI model, used for TT design (i.e. helicopter in hover, however with the main rotor RPM free to vary). The second test case starts from an initial condition which is far away from the operating condition used to derive this LTI model. These first two test cases are also set in an ideal environment, i.e. a noise-free and disturbance-free environment. The third engine OFF test case is added to illustrate the FCS performance when including sensors measurement noise together with a wind disturbance.

5.2. SETTING UP THE TRAJECTORY PLANNING FOR THE ENGINE ON CASES

Case 1. This test case involves a landing maneuver from a hover initial condition, starting at an altitude[1] of -8 m, with further 2 m and -1 m displacements, in the North and East axes respectively, together with a 90° right turn in heading. Numerically, the initial and final

---

[1]Recall that the vertical z-axis is oriented positive down.

conditions for this maneuver are given by[2]:

$$\mathbf{x}_i = \begin{pmatrix} 0\,m & 0\,m & -8\,m & 0\,m/s & 0\,m/s & 0\,m/s \\ 0\,rad/s & 0\,rad/s & 0\,rad/s & \pi(3.4/180)\,rad & 0\,rad & 0\,rad \end{pmatrix}^{\top}$$

$$\mathbf{x}_f = \begin{pmatrix} 2\,m & -1\,m & -1\,m & 0\,m/s & 0\,m/s & 0.2\,m/s \\ 0\,rad/s & 0\,rad/s & 0\,rad/s & \pi(3.4/180)\,rad & 0\,rad & \pi(90/180)\,rad \end{pmatrix}^{\top}$$

Before proceeding, we make the following comments

- The final altitude x_Z (see the third component of $\mathbf{x}_f$) is set to -1 m. This allows to add a safety margin into the planned trajectory.

- The final vertical velocity w (see the sixth component of $\mathbf{x}_f$) is set to 0.2 m/s. When close to the ground, the goal is to move at a constant and slow rate of descent (until the skids hit the ground).

Next, the flight envelope (i.e. state constraints in the form of minimum and maximum limits, partially based upon engineering judgment) is defined as follows:

$$\mathbf{x}_{min} = -\begin{pmatrix} 50\,m & 50\,m & 50\,m & 5\,m/s & 1\,m/s & 3\,m/s \\ \pi(100/180)\,rad/s & \pi(100/180)\,rad/s & \pi(100/180)\,rad/s \\ \pi(15/180)\,rad & \pi(15/180)\,rad & 2\pi\,rad \end{pmatrix}^{\top}$$

$$\mathbf{x}_{max} = \begin{pmatrix} 50\,m & 50\,m & -0.25\,m & 15\,m/s & 1\,m/s & 1.16\,m/s \\ \pi(100/180)\,rad/s & \pi(100/180)\,rad/s & \pi(100/180)\,rad/s \\ \pi(15/180)\,rad & \pi(15/180)\,rad & 2\pi\,rad \end{pmatrix}^{\top}$$

Before proceeding, we make the following comments

- When the helicopter is on the ground, the Center of Gravity (CG) height is equal to -0.25 m (see the third component of $\mathbf{x}_{max}$).

- The maximum helicopter velocity is limited as follows. A full-size helicopter such as the Bell UH-1H has a main rotor radius of 7.24 m, whereas our model helicopter has a main rotor radius of 0.9 m, resulting in a scale ratio N equal to $N = 7.24/0.9 = 8.04$. Now, a model and its full-size counterpart are said to be dynamically similar if the relative magnitudes of their governing forces are unchanged by scale [1]. Often, the so-called Froude scaling is used to study systems at a reduced size [1]. The Bell UH-1H has a top speed of 60.28m/s, thus based on Froude scaling the top speed of our RC helicopter would be $60.28/\sqrt{N} = 21.26$ m/s. In our case, and in order to reduce the stresses on the airframe and main rotor hub, we do not intend to operate the vehicle beyond 15 m/s (see the fourth component of $\mathbf{x}_{max}$).

---

[2]Recall also that the rigid-body dynamics, used in the flatness TP, is characterized by a state-vector of dimension twelve $\mathbf{x} = (x_N\ x_E\ x_Z\ u\ v\ w\ p\ q\ r\ \phi\ \theta\ \psi)^{\top}$, with total forces and total moments as inputs, each of dimension three, given by $\mathbf{F}_{CG}^b = (F_{CG_X}^b\ F_{CG_Y}^b\ F_{CG_Z}^b)^{\top}$, and $\mathbf{M}_{CG}^b = (M_{CG_X}^b\ M_{CG_Y}^b\ M_{CG_Z}^b)^{\top}$.

- The body lateral velocity v is constrained to ± 1 m/s, as to limit vehicle sideslip motion.

- To prevent flight into the Vortex-Ring-State (VRS)[3], the body vertical velocity w is limited to a third of the induced velocity in hover $w \leq \frac{1}{3} v_{ih}$, giving $w \leq 3.5/3 = 1.16$ m/s (see the sixth component of $\mathbf{x}_{max}$).

- The roll ϕ and pitch θ angles are limited to $\pm 15°$, in order to: 1) keep the load factor n within acceptable values[4], i.e. preferably below one; and 2) minimize the system's nonlinear behavior, facilitating thus the trajectory tracking[5].

Next, the input constraints, i.e. on the total forces and total moments, are based upon simulation experiments with the nonlinear helicopter HOM of Chapter 2, and have been chosen as follows:

$$\mathbf{F}^b_{CG_{min}} = -\begin{pmatrix} 20\ N & 15\ N & 120\ N \end{pmatrix}^\top \qquad \mathbf{M}^b_{CG_{min}} = -\begin{pmatrix} 5\ Nm & 5\ Nm & 5\ Nm \end{pmatrix}^\top$$

$$\mathbf{F}^b_{CG_{max}} = \begin{pmatrix} 20\ N & 15\ N & -30\ N \end{pmatrix}^\top \qquad \mathbf{M}^b_{CG_{max}} = \begin{pmatrix} 5\ Nm & 5\ Nm & 5\ Nm \end{pmatrix}^\top$$

Besides, additional constraints have also been included (refer also to the flatness TP in Appendix A of Chapter 4), such as: 1) a tail rotor blade tip clearance to avoid ground strike by the tail rotor during flare; and 2) a supplementary limit on the airflow through the main rotor as to avoid flight into the VRS. The airflow through the main rotor is given by $V_{rotor} = w + py_H - qx_H$, which is limited to half the induced velocity in hover $V_{rotor} \leq \frac{1}{2} v_{ih}$, see Fig. 2.19 in Chapter 2. In the cost functional, defined in Appendix A of Chapter 4, we have used the following weights $W_{T_f} = W_v = 1$ and $W_r = 100$. Here W_r is chosen high to reward straight flight trajectories.

Finally, we use the 'adaptation' functionality of the engine ON outer-loop controller, as outlined in Appendix C of Chapter 4. When close to the ground, it is crucial to keep the reference velocities as small as possible. Once the helicopter height descends below a predefined threshold (here -1 m), the position control is stopped.

Case 2. This test case involves a cruise-to-hover maneuver, starting at a North velocity $V_N = 10$ m/s, an altitude of -20 m, and then transitioning to hover mode, at an altitude of -5 m, with further 30 m and -5 m displacements, in the North and East axes respectively,

---

[3]Briefly summarized, the VRS corresponds to a condition where the helicopter is descending in its own wake, resulting in a chaotic and dangerous flight condition [2].

[4]For a level turning flight the load factor is given by $n = \frac{1}{\cos \phi}$.

[5]It is well known that strong coupling in longitudinal and lateral motions exists for helicopters flying in low-speed, high-g turns (i.e. high load factor), and that for helicopters with a single main rotor, the direction of turn has also a significant influence on the flight dynamics [3]. This coupling becomes stronger with higher roll or pitch angles, i.e. with higher-g turns [4, 5]. It was further shown in [3] that the performance of a FCS, designed using a straight flight condition, can be severely degraded when the helicopter enters a turn. Since in our case the nominal LTI plant, used for control synthesis, corresponds to a hover condition, it becomes relevant to maintain small angles in roll and pitch.

together with a 120° left turn in heading. Numerically, the initial and final conditions for this maneuver are given by:

$$\mathbf{x}_i = \begin{pmatrix} 0\,m & 0\,m & -20\,m & 10\,m/s & 0\,m/s & 0\,m/s \\ 0\,rad/s & 0\,rad/s & 0\,rad/s & \pi(2.6/180)\,rad & -\pi(1.1/180)\,rad & 0\,rad \end{pmatrix}^{\top}$$

$$\mathbf{x}_f = \begin{pmatrix} 30\,m & -5\,m & -5\,m & 0\,m/s & 0\,m/s & 0\,m/s \\ 0\,rad/s & 0\,rad/s & 0\,rad/s & \pi(3.4/180)\,rad & 0\,rad & -\pi(120/180)\,rad \end{pmatrix}^{\top}$$

Regarding the state and input constraints, and cost functional weights, these are identical to the engine ON case 1.

5.3. Setting up the trajectory planning for the engine OFF cases

Three engine OFF test cases are included to demonstrate the automatic autorotation landing capability. The first engine OFF test case is set to evaluate the FCS performance for an initial flight condition which is identical to the flight condition used to derive the nominal LTI model, used for TT design (i.e. helicopter in hover with free main rotor RPM). The second test case starts from an initial condition which is far away from the operating condition used to derive this LTI model. These first two test cases are also set in an ideal environment, i.e. a noise-free and disturbance-free environment. The third engine OFF test case is set to illustrate the FCS performance when including sensors measurement noise, together with a wind disturbance.

Case 1. This test case involves an autorotation, starting from an engine failure in hover, at an altitude of -35 m, and then landing at 2 m North and 1 m East position, without any heading turn. Numerically, the initial and final conditions are given by:

$$\mathbf{x}_i = \begin{pmatrix} 0\,m & 0\,m & -35\,m & 0\,m/s & 0\,m/s & 0\,m/s \\ 0\,rad/s & 0\,rad/s & 0\,rad/s & \pi(3.4/180)\,rad & 0\,rad & 0\,rad \end{pmatrix}^{\top}$$

$$\mathbf{x}_f = \begin{pmatrix} 2\,m & 1\,m & -0.75\,m & 0\,m/s & 0\,m/s & 0.2\,m/s \\ 0\,rad/s & 0\,rad/s & 0\,rad/s & 0\,rad & 0\,rad & 0\,rad \end{pmatrix}^{\top}$$

Here, we make also the following comments

- Note that we also give a final value to the North and East horizontal positions (this was not the case in the planning of Chapter 3). This represents additional constraints on the TP. We do this with an eye on future experimental flight tests where, for safety reasons, we want to know in advance where the helicopter will be landing.

- The final altitude x_Z (see the third component of $\mathbf{x}_f$) is set to -0.75 m. This allows to add a safety margin into the planned trajectory[6].

---

[6]This value was set to -1 m for the engine ON automatic landing. For the engine OFF case, better autoration landings were obtained when adjusting this value to -0.75 m.

- Again, the final vertical velocity w (see the sixth component of $\mathbf{x}_f$) is set to 0.2 m/s.

- The final time T_f is bounded such that $T_f \leq T_{OFF}$, with $\frac{x_{Z_I}}{1.75v_{ih}} \leq T_{OFF} \leq \frac{x_{Z_I}}{1.50v_{ih}}$, see our discussion in Section 4.3.3, giving for this test case $5.7s \leq T_{OFF} \leq 6.6s$. Here we chose $T_{OFF} = 6$ s.

The constraints on states and inputs are identical to the ones used in the engine ON cases, except for the following item: we allow for a higher downwards velocity on the w channel, up to 15 m/s. Besides, the limit on airflow through the main rotor is also removed, i.e. flight through the VRS is here allowed[7]. In the cost functional of Section 4.3.3, we have used the following weights $W_u = W_v = W_w = W_\psi = 1$.

Finally, we use here the 'adaptation' functionality of the engine OFF outer-loop controller, as outlined in Section 4.6.3 of Chapter 4. When close to the ground, it is crucial to keep the reference velocities as small as possible. Once the helicopter height descends below a predefined threshold (here -1 m), the position control is stopped.

Case 2. This test case involves an autorotation, starting from an engine failure at $V_N = 8$ m/s, at an altitude of -45 m, and then landing at 30 m North and 0 m East position, together with a 30° left turn in heading. Numerically, the initial and final conditions for this maneuver are given by:

$$
\begin{aligned}
\mathbf{x}_i = \big(\ & 0\,m \quad 0\,m \quad -45\,m \quad 8\,m/s \quad 0\,m/s \quad 0\,m/s \\
& 0\,rad/s \quad 0\,rad/s \quad 0\,rad/s \quad \pi(2.6/180)\,rad \quad 0\,rad \quad -\pi(0.8/180)\,rad \ \big)^{\top} \\
\mathbf{x}_f = \big(\ & 30\,m \quad 0\,m \quad -0.75\,m \quad 0\,m/s \quad 0\,m/s \quad 0.2\,m/s \\
& 0\,rad/s \quad 0\,rad/s \quad 0\,rad/s \quad 0\,rad \quad 0\,rad \quad -\pi(30/180)\,rad \ \big)^{\top}
\end{aligned}
$$

We make also the following comments

- Again we give a final value to the North and East horizontal positions.

- The final time T_f is bounded such that $T_f \leq T_{OFF}$, with $\frac{x_{Z_I}}{1.75v_{ih}} \leq T_{OFF} \leq \frac{x_{Z_I}}{1.50v_{ih}}$, giving for this test case $7.3s \leq T_{OFF} \leq 8.5s$. Here we chose $T_{OFF} = 7.3$ s.

Regarding the state and input constraints, and cost functional weights, these are identical to the engine OFF case 1.

For engine OFF flight conditions having relatively high initial velocities, we implemented the following 'adaptation' functionality for the engine OFF outer-loop controller. When $|x_Z| \leq 5$ m is true, we stop the horizontal position tracking (x_N, x_E). This helps lowering the final (touch-down) values of the 2D horizontal velocities. Further, when $|x_Z| \leq 1$ m is true, we stop the vertical position tracking (x_Z) as well.

---

[7] Indeed, and depending on the initial condition at the instant of engine failure, a brief transition through the VRS may be unavoidable.

Case 3. This test case involves an autorotation, starting from an engine failure in hover, at an altitude of -30 m, and then landing at 0 m North and 0 m East position (i.e. the horizontal position of the landing spot is identical to the horizontal position of the initial state), without any heading turn. We also include Gaussian, white noise, on the 12 measured states $\mathbf{y} = (x_N \; x_E \; x_Z \; u \; v \; w \; p \; q \; r \; \phi \; \theta \; \psi)^\top$, with the following 1-$\sigma$ values:

$$
\begin{pmatrix}
0.1 \; m & 0.1 \; m & 0.1 \; m & 0.05 \; m/s & 0.05 \; m/s & 0.05 \; m/s \\
\pi(3/180) \; rad & \pi(3/180) \; rad & \pi(3/180) \; rad \\
\pi(1/180) \; rad & \pi(1/180) \; rad & \pi(3/180) \; rad
\end{pmatrix}^\top
$$

These 1-σ values correspond to the noise weight values used during controller design in Chapter 4, expect for the noise on the three body velocities (the three most critical signals), where we have used a noise value which is five times higher than the value used during controller design, in order to better visualize the response characteristics of the FCS.

We also include a headwind of 8 m/s, which is equivalent to a Beaufort wind force value of 4, corresponding to the yearly average wind force along the coast in The Netherlands [6]. Note that this is a rather heavy wind condition for such a small-scale helicopter. Now, numerically, the initial and final conditions for this maneuver are given by:

$$
\begin{aligned}
\mathbf{x}_i &= \begin{pmatrix}
0 \; m & 0 \; m & -30 \; m & 0 \; m/s & 0 \; m/s & 0 \; m/s \\
0 \; rad/s & 0 \; rad/s & 0 \; rad/s & \pi(3.4/180) \; rad & 0 \; rad & 0 \; rad
\end{pmatrix}^\top \\
\mathbf{x}_f &= \begin{pmatrix}
0 \; m & 0 \; m & -0.75 \; m & 0 \; m/s & 0 \; m/s & 0.2 \; m/s \\
0 \; rad/s & 0 \; rad/s & 0 \; rad/s & 0 \; rad & 0 \; rad & 0 \; rad
\end{pmatrix}^\top
\end{aligned}
$$

The final time T_f is bounded such that $T_f \leq T_{OFF}$, with $\frac{x_{Z_I}}{1.75v_{ih}} \leq T_{OFF} \leq \frac{x_{Z_I}}{1.50v_{ih}}$, giving for this test case $4.9s \leq T_{OFF} \leq 5.7s$. Here we chose $T_{OFF} = 5$ s. Regarding the state and input constraints, and cost functional weights, together with the 'adaptation' functionality of the outer-loop controller, these are identical to the engine OFF case 1.

Remark 7 *Before proceeding with analyzing the time-traces of the closed-loop simulation data, we quickly compared the frequency content[8] of the various inner- and outer-loop reference signals (generated by the planner, for all engine ON and engine OFF test cases) with the bandwidths of the complementary sensitivity function $T_i(s)$, which have been reported in Table 4.1 of Chapter 4. Fortunately, the frequency content of all reference signals were lower than the corresponding bandwidth of $T_i(s)$, hence the engine ON and OFF controllers ought to be able to track the reference signals.*

5.4. DISCUSSION OF CLOSED-LOOP SIMULATION RESULTS FOR THE ENGINE ON CASES

Fig. 5.1 and Fig. 5.4 visualize the required control inputs for the engine ON test cases 1 and 2, respectively. Fig. 5.2 and Fig. 5.5 visualize the evolution of the 3D inertial velocities (V_N, V_E, V_Z) and positions (x_N, x_E, x_Z). Although the vertical z-axis is oriented positive

---

[8]This is done by computing the single-sided amplitude spectra, obtained through Fast Fourier Transforms (FFT).

down, on the figures V_Z and x_Z are shown positive up for better readability. Further, Fig. 5.3 and Fig. 5.6 visualize the time-histories for the body states, namely attitude angles (ϕ, θ, ψ), linear velocities (u, v, w), and rotational velocities (p, q, r).

In Fig. 5.2, Fig. 5.5, Fig. 5.3, and Fig. 5.6, the black lines represent the outputs from the flatness TP, these include the planned 3D inertial positions $(x_N \; x_E \; x_Z)_{TP}^\top$, defined in Fig. 4.3, the planned 3D body velocities $(u \; v \; w)_{TP}^\top$, defined in Fig. 4.4, and the planned heading ψ_{TP}, also defined in Fig. 4.4. The flatness-based TP, in Section 4.3 of Chapter 4, computes also a planned trajectory for the remaining states, e.g. roll angle ϕ, pitch angle θ, roll rate p, etc. However, and for the sake of clarity, in Fig. 5.2, Fig. 5.5, Fig. 5.3, and Fig. 5.6, we have only visualized the TP outputs that will be tracked.

Now, in Fig. 5.2 and Fig. 5.5, the blue lines, named *reference for outer-loop*, represent the signals that need to be tracked by the outer-loop controller. Here, these signals are simply the planned 3D inertial positions $(x_N \; x_E \; x_Z)_{TP}^\top$, i.e. black and blue lines are identical (except possibly at the end of the flight, see Section 4.6.3 of Chapter 4). In Fig. 5.3 and Fig. 5.6, the blue lines, named *reference for inner-loop*, represent the signals that need to be tracked by the inner-loop controller. Here, these signals include the planned heading ψ_{TP}, where again black and blue lines are identical. However, the velocities that need to be tracked by the inner-loop are given by $(u \; v \; w)_{TP}^\top + (u \; v \; w)_d^\top$, and here black and blue lines are not identical. Finally, the red lines represent the outputs from the nonlinear helicopter model of Chapter 2. From these figures, we see that:

- The combined trajectory planning and tracking system is capable of safely guiding and controlling the helicopter.

- From Fig. 5.3, and Fig. 5.6, we see that a single LTI controller is capable of controlling the nonlinear helicopter system, for a relatively large variation in forward vehicle velocity (i.e. body linear velocity u is varying between approximately -1 m/s and 10 m/s).

- The specifications for a successful automatic landing, see Definition 4.2 in Section 4.3.3 and Appendix D of Chapter 4, have been defined as $|u| \leq 0.5$ m/s, $|v| \leq 0.5$ m/s, $|w| \leq 0.25$ m/s, $|\phi| \leq 10°$, and $|\theta| \leq 10°$. Regarding case 1, at the instant of ground impact, we have for the body horizontal velocities $u = 0.03$ m/s, $v = -0.04$ m/s, the body vertical velocity $w = 0.22$ m/s, and the roll and pitch angles $\phi = 5.02°$, and $\theta = 2.04°$. Hence all specifications for a successful automatic landing are met.

- From Fig. 5.1, Fig. 5.4, and from the actuator data reported in Table 2.1 of Chapter 2, we see that the control input amplitudes never saturate, i.e. $|\theta_0| \leq 13°$, $|\theta_{1c}| \leq 6°$, $|\theta_{1s}| \leq 6°$, and $|\theta_{TR}| \leq 20°$.

- From Fig. 5.2, Fig. 5.5, Fig. 5.3, and Fig. 5.6, better tracking performance is achieved for the vertical motion w and x_Z in (and heading ψ), when compared to tracking performance on the horizontal channels (u, v) and (x_N, x_E), see our discussion in Section 2.4.2 of Chapter 2.

- Close to zero steady-state errors can be seen for the inner-loop reference tracking, for both test cases.

- Position control clearly exhibit some over- and undershoot. In addition, nonzero steady-state errors are observed for test case 1. This is partially due to the fact that position control is stopped when the helicopter descends below 1 m.

- Regarding test case 2, even though roll ϕ and pitch θ angles are not controlled, they nicely adjust, at the end of the flight, to their respective hover values.

- Although the nominal model, used for control design, was linearized at a condition outside the ground effect, we did not notice any significant performance deterioration of the closed-loop system, when the helicopter was in ground effect (i.e. below 1 m above ground level).

In addition, Fig. 5.7, Fig. 5.8, Fig. 5.9, and Fig. 5.10 visualize the frequency content of the main rotor lateral Tip-Path-Plane (TPP) tilt angle β_{1s}, vehicle roll rate p, and control inputs respectively, for the Engine ON case 1 (the engine ON case 2 is very similar). Although, in our test cases, the frequency contents of the applied control inputs are not broadband, we are still able to identify some salient natural modes of this small-scale helicopter system. For instance, the first two figures clearly show the main rotor TPP modes, with the lowest (the so-called regressing TPP mode) at a frequency of 5.5 Hz. The regressing flapping mode is the most relevant one, when focusing on helicopter flight dynamics, as it may have a tendency to couple into the fuselage modes [7–9]. Fig. 5.8 also shows the main rotor vibrations. In the engine ON case, i.e. at a fixed main rotor RPM of 1350 (equivalent to 22.5 Hz), we can clearly identify the 2/Rev[9] rotor vibration at 45 Hz.

For the engine ON case, simulation tests have shown that a high-bandwidth closed-loop system was not required for the case of gentle and smooth flight maneuvers. This led to the selection of low bandwidth performance weights $W_p(s)$, during controller synthesis. Accordingly we see that the frequency content of the control inputs is rather low, staying below 0.5 Hz, see Fig. 5.9–Fig. 5.10, except for an interestingly large peak at 2.7 Hz. This peak at 2.7 Hz, clearly seen on these four figures (predominantly related to a roll-pitch-yaw motion), is an interesting aspect of these figures, and represents the interaction between the Flight Control Computer (FCS) and the main rotor. Hence, we have a situation where the actuators are also reacting to a periodic rotor-fuselage coupling (in addition to vehicle rigid-body dynamics), as opposed to a context where the actuators are only responding to the rigid-body dynamics. This clearly results in limit cycle oscillations.

In the experimental results obtained in [10, 11], a 3.1 Hz pendulum-like mode in roll and pitch was also observed, for the case of a two-bladed small-scale helicopter, albeit having a teetered main rotor, but with somewhat comparable vehicle size and mass, hence corroborating our results. This phenomenon (i.e. interaction between the FCS and the main rotor) has only sparsely been covered in the small-scale UAV literature. This phenomenon is well-known within the realm of wind turbines [12], and is somewhat reminiscent to the

---

[9]Since we have a two-bladed main rotor.

realm of Higher-Harmonic-Control (HHC) for helicopters [13]. This interaction between the FCS and the main rotor is comparable to the well-known interaction between aircraft FCS and aircraft structural dynamics—i.e. aeroservoelastic effects [14]—which are known to lead to flutter or limit cycle oscillations, and hence dynamic and fatigue loads. Aside from these dynamic and fatigue loads, this dynamical interaction would also result in our case in an increase of the electrical power consumption, and hence a lower flight time. A general approach to mitigate such problems would consist in: 1) using higher-order LTI models during the control design, possibly in combination with a reduced-order observer, in order to estimate the unmeasured main rotor states; and/or 2) use carefully selected notch filters, see [11].

5.5. Discussion of closed-loop simulation results for the engine OFF cases

We discuss here the first two engine OFF cases, the third engine OFF case will be addressed in Section 5.5.2. Fig. 5.13 and Fig. 5.16 visualize the required control inputs for the engine OFF test cases 1 and 2 respectively. Fig. 5.14 and Fig. 5.17 visualize the evolution of the 3D inertial velocities (V_N, V_E, V_Z) and positions (x_N, x_E, x_Z), whereas Fig. 5.15 and Fig. 5.18 visualize the time-histories for the body states, namely attitude angles (ϕ, θ, ψ), linear velocities (u, v, w), and rotational velocities (p, q, r). Fig. 5.19 and Fig. 5.20 visualize the time-histories for the main rotor RPM Ω_{MR}. Note also that the definition of the black, blue, and red lines, is identical to the one presented here-above, for the engine ON cases, and hence is not repeated here. From these figures, we see that:

- The combined trajectory planning and tracking system is capable of safely guiding and controlling the helicopter in autorotation.

- The specifications for a successful automatic landing, see Definition 4.2 in Section 4.3.3 and Appendix D of Chapter 4, have been defined as $|u| \leq 0.5$ m/s, $|v| \leq 0.5$ m/s, $|w| \leq 0.25$ m/s, $|\phi| \leq 10$ °, and $|\theta| \leq 10$ °. Regarding case 1, at the instant of ground impact, we have for the body horizontal velocities $u = 0.04$ m/s, $v = 0.15$ m/s, the body vertical velocity $w = 0.25$ m/s, and the roll and pitch angles $\phi = 1.41$ °, and $\theta = 3.39$ °. Regarding case 2, at the instant of ground impact, we have for the body horizontal velocities $u = -0.37$ m/s, $v = 0.13$ m/s, the body vertical velocity $w = 0.21$ m/s, and the roll and pitch angles $\phi = 6.67$ °, and $\theta = -0.54$ °. Regarding case 3, at the instant of ground impact, we have for the body horizontal velocities $u = -0.09$ m/s, $v = 0.12$ m/s, the body vertical velocity $w = 0.24$ m/s, and the roll and pitch angles $\phi = -0.75$ °, and $\theta = -0.15$ °. Hence all specifications for a successful automatic landing are met.

- A single LTI controller is capable of controlling and landing the helicopter system, in autorotation, for a relatively large variation in forward and vertical vehicle velocity (at least up to approximately 8 to 10 m/s), and for relatively large variations in main rotor RPM (approximately in the range 50% to 110% of the nominal engine ON value), see Fig. 5.15, Fig. 5.18, Fig. 5.19, and Fig. 5.20.

- From the actuator data, reported in Table 2.1 of Chapter 2, we see that the control input amplitudes would never saturate, except for a brief saturation of the main rotor collective θ_0, that would happen just prior to touch-down.

- As expected, tracking performance is better for the vertical motion w and x_Z, than the tracking of horizontal motion (u, v) and (x_N, x_E), see our discussion in Section 2.4.3 of Chapter 2.

- Some steady-state errors can be seen on the horizontal channel (see Fig. 5.14 and Fig. 5.17) and heading (see Fig. 5.15 and Fig. 5.18), whereas this is not the case for the vertical channel (refer to these same figures). This is also partially due to the fact that position control is stopped some time before the helicopter touches the ground.

- Main rotor RPM Ω_{MR} behaves as expected, see Fig. 5.19 and Fig. 5.20, i.e. we recognize the typical autorotative time-histories, as shown in Chapter 3. Notice that, when starting from low altitudes such as in these test cases, the helicopter does not even reach a steady-state autorotation (main rotor RPM is not constant), rather it is continuously in transition from one non-equilibrium state to the next.

- Again, although the nominal model, used for control design, was linearized at a condition outside the ground effect, we did not notice any significant performance deterioration of the closed-loop system, when the helicopter was in ground effect.

In addition, Fig. 5.21, Fig. 5.22, Fig. 5.23, and Fig. 5.24 visualize the frequency content of the main rotor lateral TPP tilt angle β_{1s}, vehicle roll rate p, and control inputs, respectively. For the engine OFF case, simulation experiments have shown that a higher closed-loop bandwidth was necessary for good tracking behavior. This resulted in a bandwidth increase of the controller performance weights. This increase in control bandwidth has also some drawbacks. Indeed, we also clearly see an interaction between the FCS and the main rotor around 5.5 Hz. This mode was identified to be the regressing flap mode of 5.5 Hz, for a constant main rotor RPM of 1350, in the figures for the engine ON case. In the engine OFF case, the RPM is not constant anymore, and hence the interaction between the FCS and the main rotor shows a frequency spread, which cannot easily be eliminated by notch filters.

Summarizing the observed results for the engine OFF cases, we see that the crucial control of vertical position and velocity exhibits outstanding behavior in terms of tracking performance, and does not require an additional increase in control bandwidth. However, the tracking of horizontal positions and horizontal velocities is clearly lacking some bandwidth (i.e. the flown trajectories are clearly lagging the planned ones). Although a further increase of the horizontal closed-loop bandwidths provided good results when evaluated on the LTI model used for control design, this increase in closed-loop bandwidths resulted, unfortunately, in closed-loop instabilities, when evaluated on the nonlinear helicopter model of Chapter 2.

5.5.1. System energy: the engine ON versus engine OFF cases
We compute here the stored energy in our helicopter system. For the following analysis, we assume that the flight time is not limited by the amount of energy stored inside the on-board

batteries. In other words, the electrical power supply system is omitted from this energy balance analysis. Hence, we consider only the following energy components (refer also to the nomenclature in Appendix A of Chapter 2): the vehicle potential energy $m_V g|x_Z|$; the vehicle kinetic energy $\frac{1}{2}m_V(u^2+v^2+w^2)+\frac{1}{2}(Ap^2+Bq^2+Cr^2)$, with A, B, and C the diagonal elements of the inertia matrix $\mathbb{I}_V$; the stored energy in the main rotor $\frac{1}{2}N_b I_b \Omega_{MR}^2$; and the total energy (sum of previous three). These energies have been plotted in Fig. 5.11–Fig. 5.12, and Fig. 5.25–Fig. 5.26, for the engine ON test cases, and for the first two engine OFF test cases, respectively. A quick scan on total energies reveals the main difference between the engine ON and OFF cases, i.e. while the total energy for an engine ON case may even increase, the total energy for an engine OFF case is always decreasing. This particularity renders the trajectory planning and tracking rather challenging for the engine OFF case.

For the engine ON case, we conjecture that the current vehicle state has only a limited impact (if any) on reachable states at very distant times. This is because we can always inject some energy back into the system, and hence compensate for any suboptimal decisions made at the current time. However, for the engine OFF case, since the energy of the system is always decreasing, there is less room for error. We also conjecture that the size of this reachable set, in the engine OFF case, is much smaller than the one for the engine ON case, and hence feasible engine OFF trajectories are much harder to find.

5.5.2. CLOSED-LOOP RESPONSE WITH RESPECT TO SENSORS NOISE AND WIND DISTURBANCE

Here we illustrate the response of the FCS, for the case of noisy measurement signals and a wind disturbance. The wind disturbance includes a constant (deterministic) headwind of 8 m/s, together with a small gust (Dryden stochastic variation) on the three linear axes. Fig. 5.27 visualizes the required control inputs for the engine OFF test case 3. Fig. 5.28 visualizes: 1) the nonlinear model time-histories for the 3D inertial velocities and positions (in red); 2) the corresponding noisy measurement positions sent to the outer-loop controller (in magenta); and 3) the wind disturbance (in green). Fig. 5.29 visualizes: 1) the nonlinear model time-histories for the nine body states (in red); and 2) the corresponding noisy measurements sent to the inner-loop controller (in magenta). Finally, Fig. 5.30 visualizes the time-histories for the main rotor RPM.

Again, we see that all specifications for a successful automatic landing are met, see Definition 4.2 in Section 4.3.3 and Appendix D of Chapter 4, despite the additional measurements noise and wind disturbance. Also Fig. 5.30 illustrates the benefits of a headwind landing, namely we see that the RPM is still high (about 1100 RPM) at the end of the landing maneuver (compare with Fig. 5.19 and Fig. 5.20). Obviously, a higher energy in the rotor allows for a smoother landing, and for additional control authority, which may be particularly useful for disturbance rejection.

5.6. CONCLUSION

In this Chapter we have evaluated the capabilities of the Trajectory Planning (TP) and Trajectory Tracking (TT) framework, previously developed in Chapter 4. In particular, we have

demonstrated in this Chapter—using the high-fidelity, high-order, nonlinear helicopter simulation of Chapter 2—the first, real-time feasible, model-based TP and TT system, for the case of a small-scale helicopter UAV with an engine OFF condition. The main distinctive features of the engine ON versus engine OFF TP and TT may be summarized as follows:

- For the engine ON case, the vehicle state at a current time t_i has only a limited impact (if any) on the reachable states at a (very distant) time t_f, with $t_f \gg t_i$. If we omit the on-board electrical power supply system from the vehicle energy balance, i.e. considering only vehicle potential, kinetic, and main rotor energies, then the total vehicle energy may decrease or increase, depending on vehicle height above ground level and vehicle velocity. By contrast, the total vehicle energy in the engine OFF case is always decreasing. Hence, we conjecture that the size of this reachable set, at time t_f, is much smaller than its engine ON counterpart, and consequently feasible trajectories are harder to find in the engine OFF case.

- For the engine ON case, helicopter operations can remain at a velocity which stays in the neighborhood of the design-point velocity, i.e. in the neighborhood of the equilibrium point velocity which was used to derive the LTI model for control design. This allows to maximize the linear behavior of the system. On the other hand, helicopter operations with the engine OFF will inevitably result in a wide range of flown velocities, including high descent rates, and even flight into the chaotic Vortex-Ring-State (VRS). Indeed, a brief transition through the VRS may in some cases be required. This obviously tends to 'amplify' the nonlinear behavior of the system.

- For the engine ON case, the designer can choose to keep the bandwidth of the closed-loop system rather small, by only considering gentle and smooth maneuvers in the design specification phase. For the engine OFF case, a higher closed-loop bandwidth is definitely required, if proper trajectory tracking is to be performed. This may complicate the controller design, since higher-order LTI models (for controller design) may have to be considered. This complicates also the practical implementation, since higher-bandwidth actuators may become compulsory.

- A general approach to mitigate the observed interaction problem, between the FCS and the main rotor dynamics, could be to use higher-order LTI models, for control design, possibly in combination with a reduced-order observer in order to estimate the unmeasured main rotor states.

- For the engine OFF case, our results show that the crucial control of vertical position and velocity exhibit outstanding behavior in terms of tracking performance, and does not require an additional increase in control bandwidth. However, the tracking of horizontal positions and horizontal velocities is clearly lacking some bandwidth. Unfortunately, a further increase of the horizontal closed-loop bandwidths resulted in closed-loop instabilities (i.e. when evaluated on the nonlinear helicopter model of Chapter 2).

- Finally, tracking performance of horizontal positions and horizontal velocities could potentially be improved, by considering one of the two following options: 1) remaining in the framework of a single robust LTI controller, however combined with a

higher-order LTI plant (i.e. containing the main rotor flap-lag and inflow dynamics), instead of the low-order plant used in Section 4.5.1 of Chapter 4. This LTI plant could also be derived using a more accurate linearization method, as discussed in Section 2.4.1 of Chapter 2; or 2) using another control method, i.e. in the realm of nonlinear, adaptive, or Linear Parameter-Varying (LPV) methods.

5

APPENDIX A: SIMULATION RESULTS
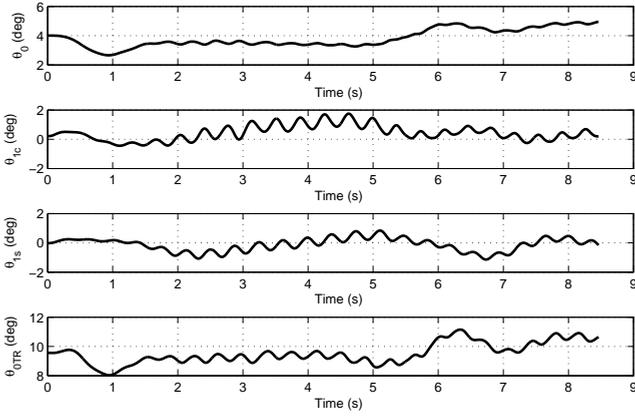
Figure 5.1: Helicopter control inputs, for the Engine ON case 1.
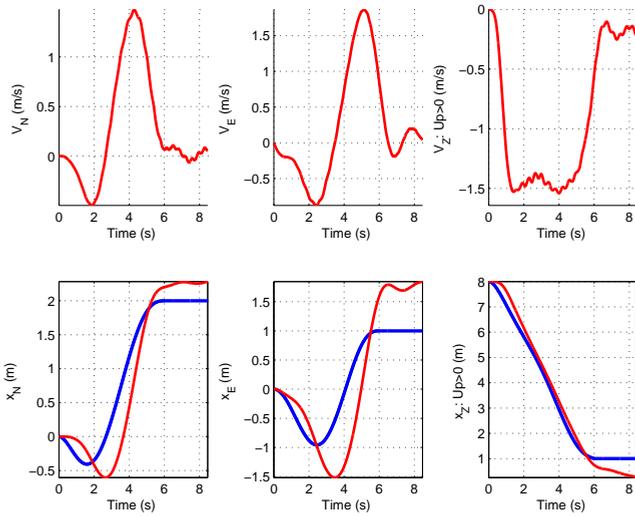
Figure 5.2: Inertial velocities and positions, for the Engine ON case 1. Black line: flatness planning. Blue line: references for outer-loop (identical to black line). Red line: controlled nonlinear model.

5
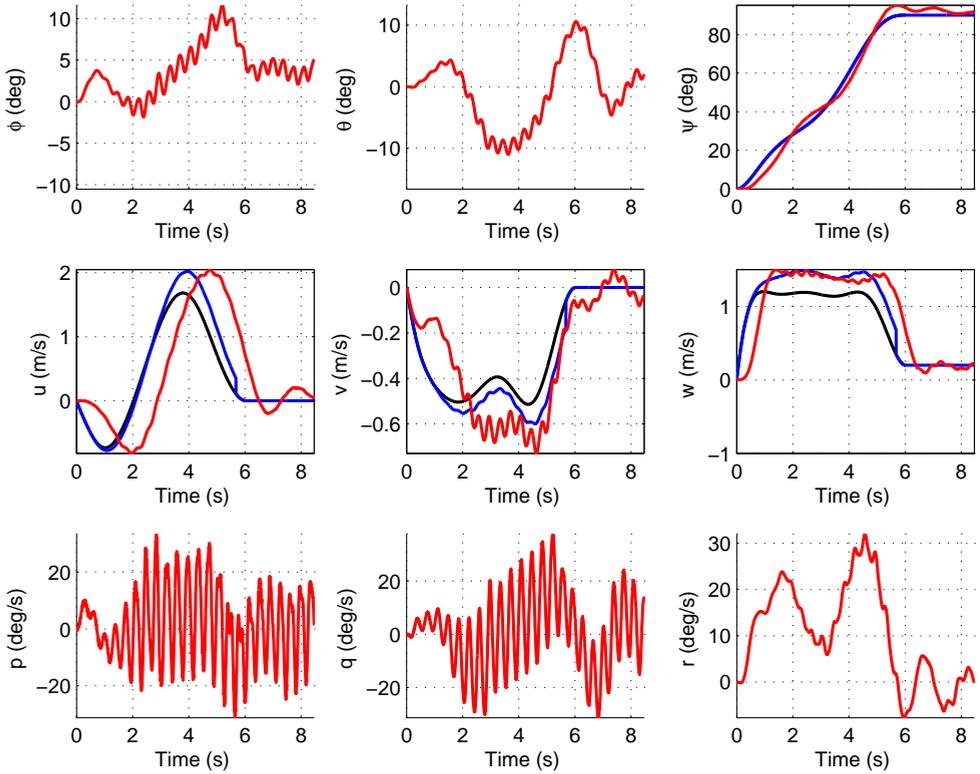
Figure 5.3: Euler angles, body linear velocities, and body rotational velocities, for the Engine ON case 1. Black line: flatness planning. Blue line: references for inner-loop. Red line: controlled nonlinear model.
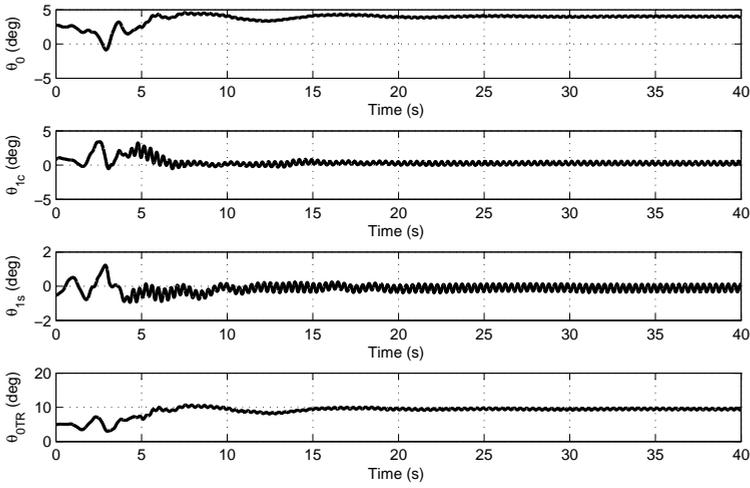
Figure 5.4: Helicopter control inputs, for the Engine ON case 2.
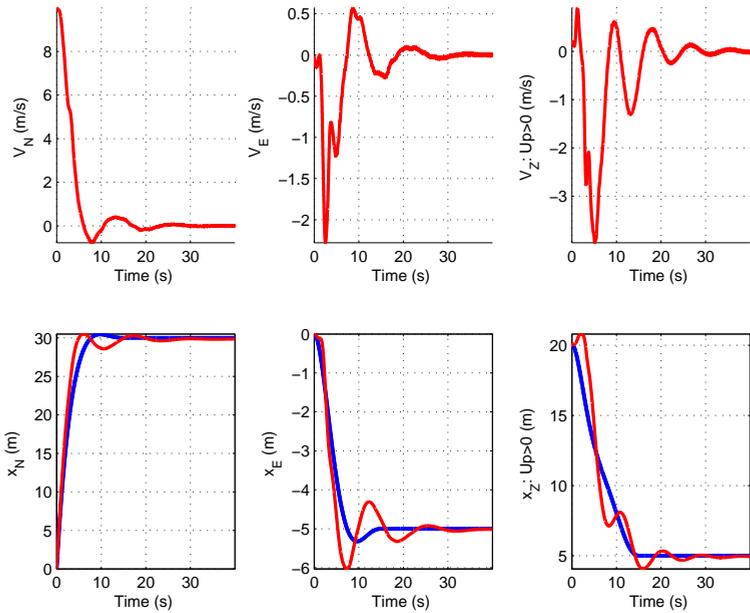
Figure 5.5: Inertial velocities and positions, for the Engine ON case 2. Black line: flatness planning. Blue line: references for outer-loop (identical to black line). Red line: controlled nonlinear model.
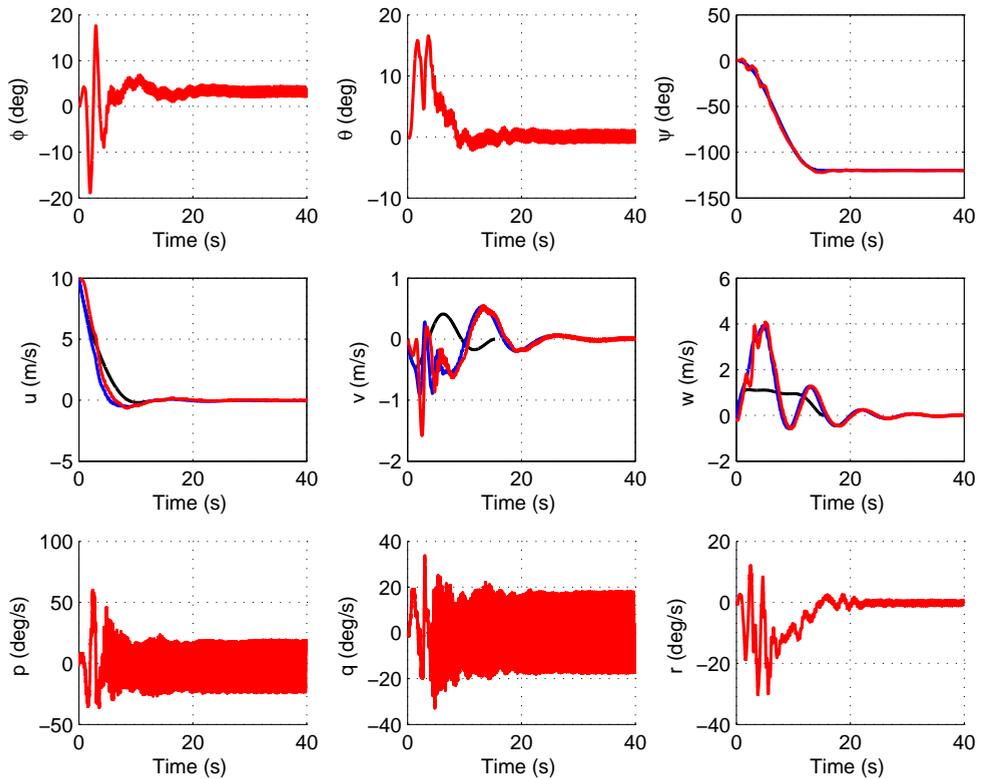
Figure 5.6: Euler angles, body linear velocities, and body rotational velocities, for the Engine ON case 2. Black line: flatness planning. Blue line: references for inner-loop. Red line: controlled nonlinear model.
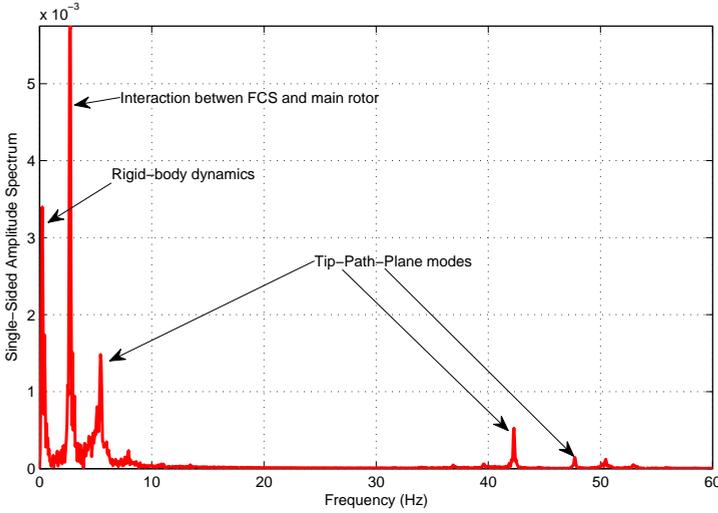
Figure 5.7: Amplitude spectrum of main rotor lateral TPP tilt angle β_{1s}, for the Engine ON case 1 (the engine ON case 2 is very similar).
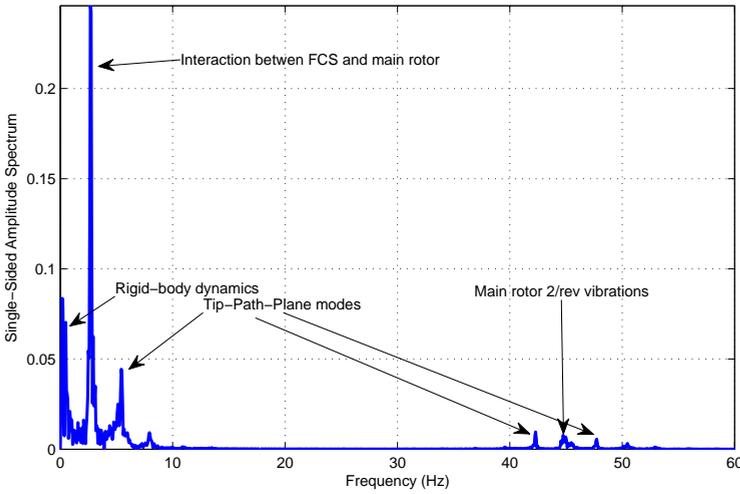
Figure 5.8: Amplitude spectrum of roll rate p, for the Engine ON case 1 (the engine ON case 2 is very similar).
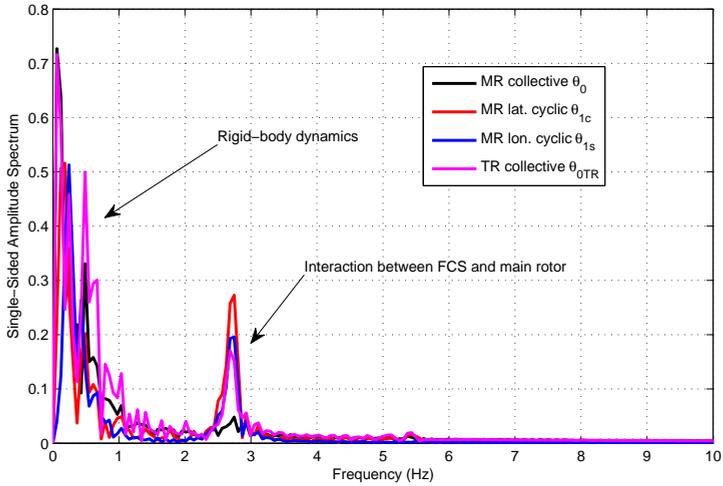
Figure 5.9: Amplitude spectrum of control inputs, for the Engine ON case 1.
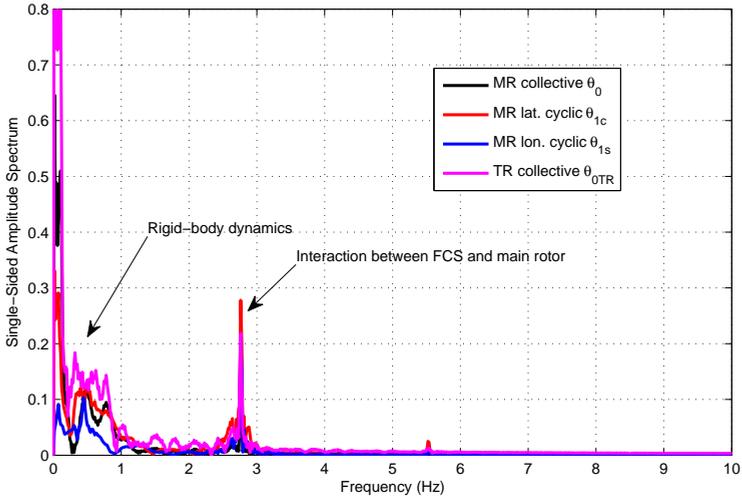
Figure 5.10: Amplitude spectrum of control inputs, for the Engine ON case 2.
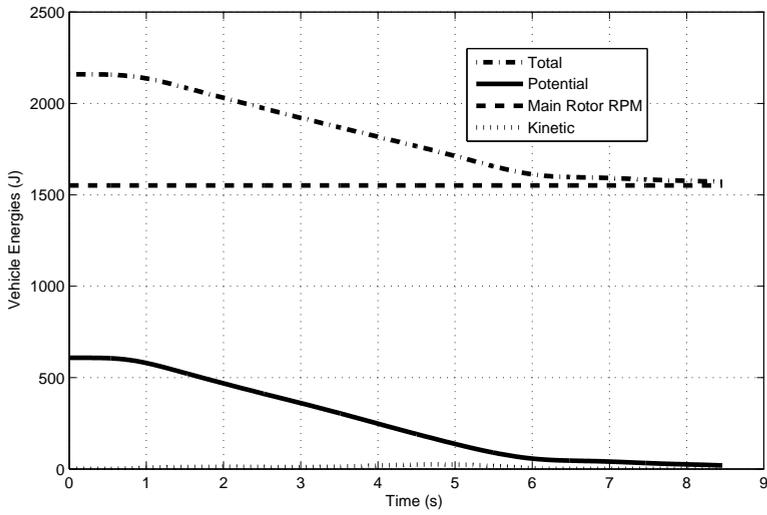
Figure 5.11: Vehicle energies, for the Engine ON case 1.
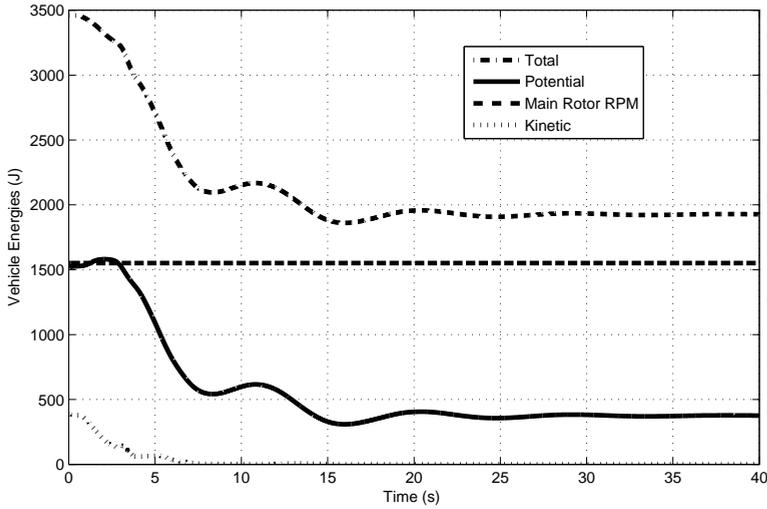
Figure 5.12: Vehicle energies, for the Engine ON case 2.

5
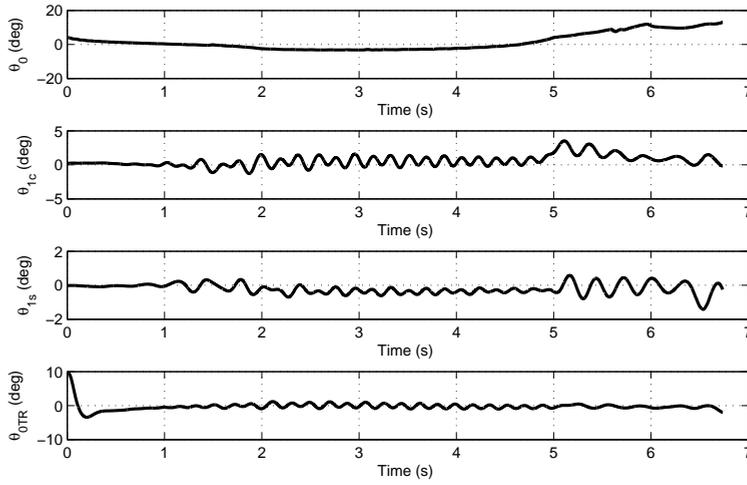
Figure 5.13: Helicopter control inputs, for the Engine OFF case 1.
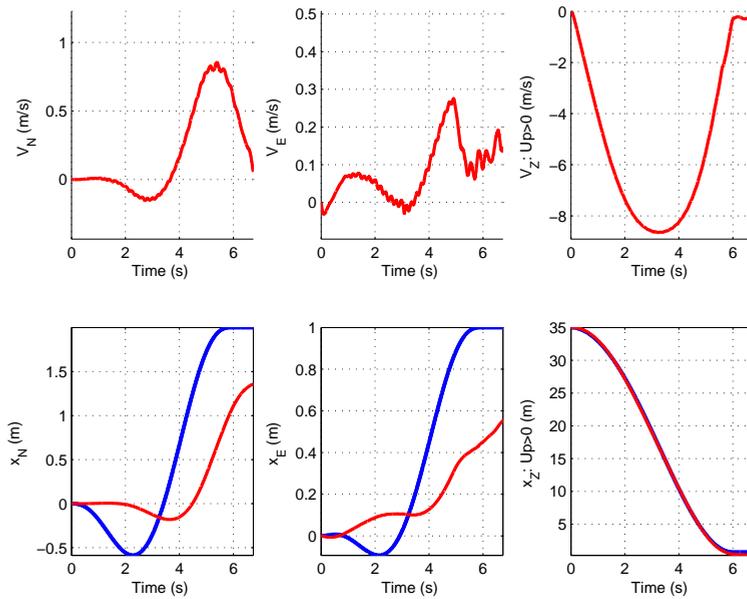
Figure 5.14: Inertial velocities and positions, for the Engine OFF case 1. Black line: flatness planning. Blue line: references for outer-loop (identical to black line). Red line: controlled nonlinear model.
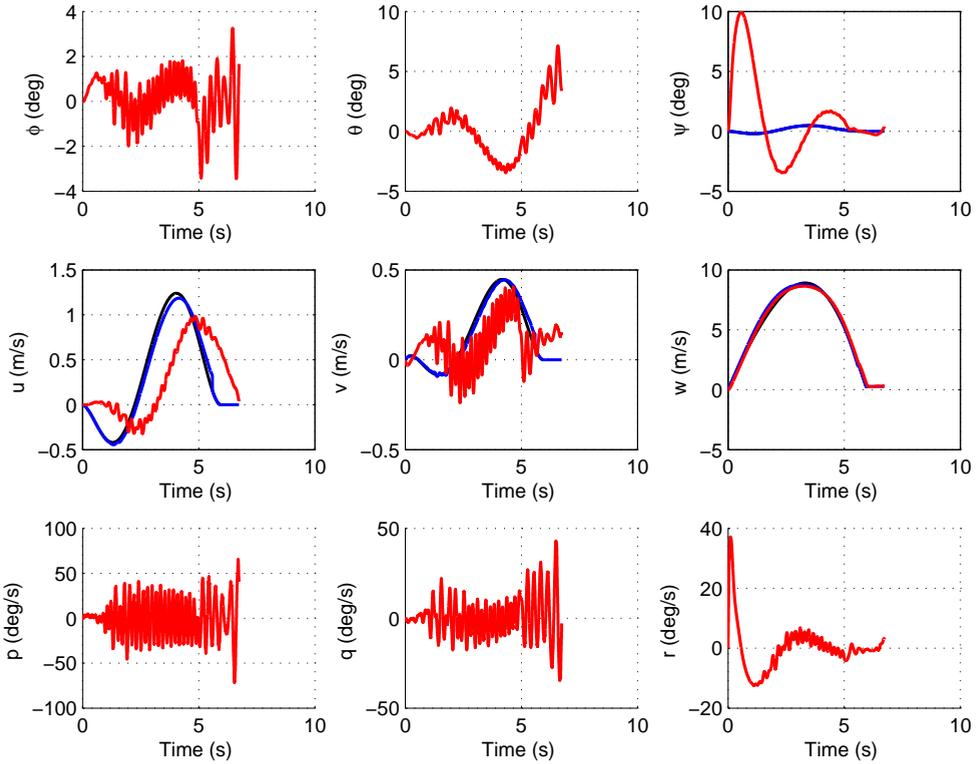
Figure 5.15: Euler angles, body linear velocities, and body rotational velocities, for the Engine OFF case 1. Black line: flatness planning. Blue line: references for inner-loop. Red line: controlled nonlinear model.

5
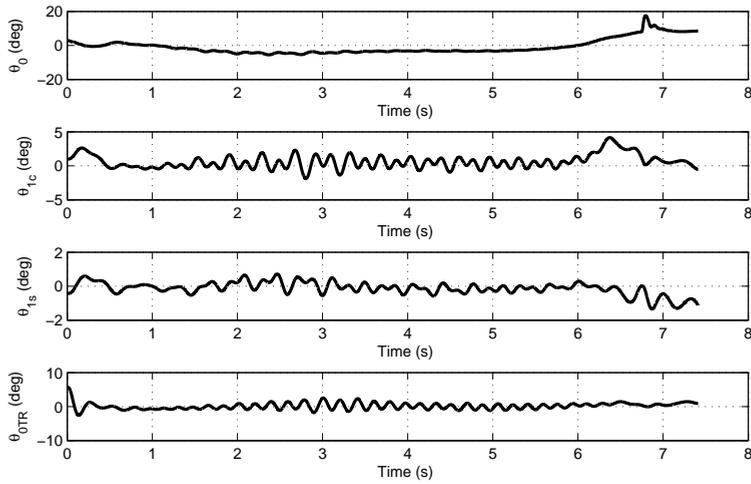
Figure 5.16: Helicopter control inputs, for the Engine OFF case 2.
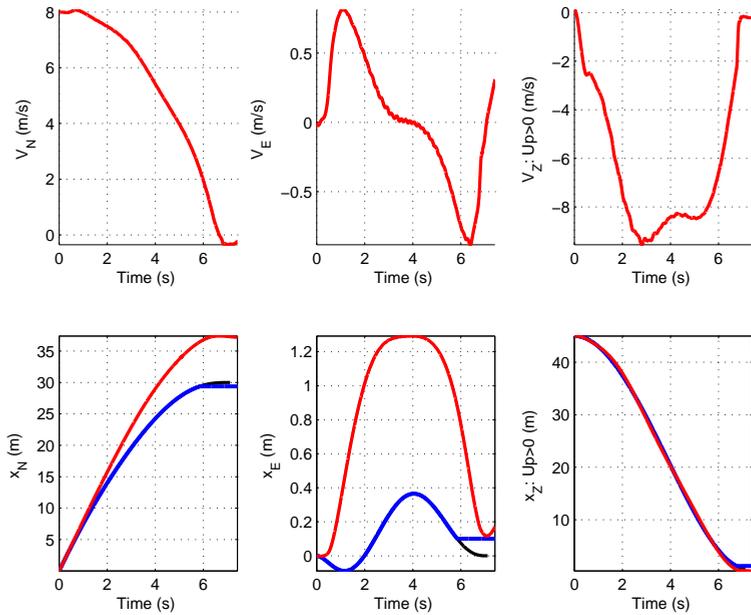
Figure 5.17: Inertial velocities and positions, for the Engine OFF case 2. Black line: flatness planning. Blue line: references for outer-loop (identical to black line). Red line: controlled nonlinear model.
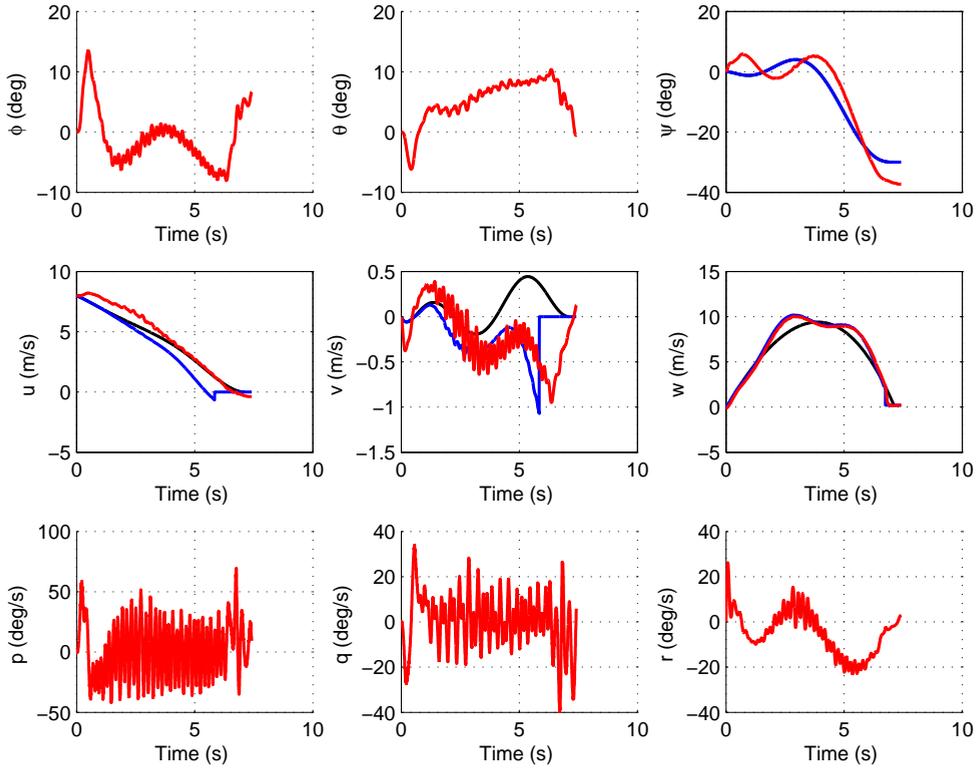
Figure 5.18: Euler angles, body linear velocities, and body rotational velocities, for the Engine OFF case 2. Black line: flatness planning. Blue line: references for inner-loop. Red line: controlled nonlinear model.
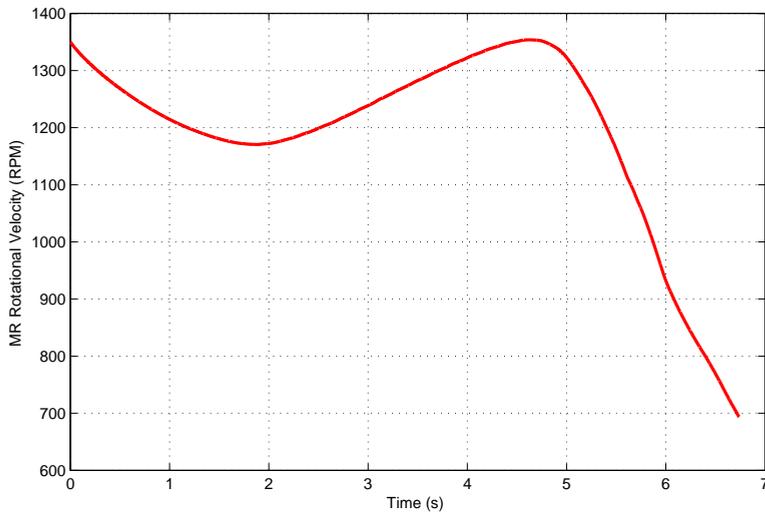
Figure 5.19: Main rotor RPM Ω_{MR}, for the Engine OFF case 1.
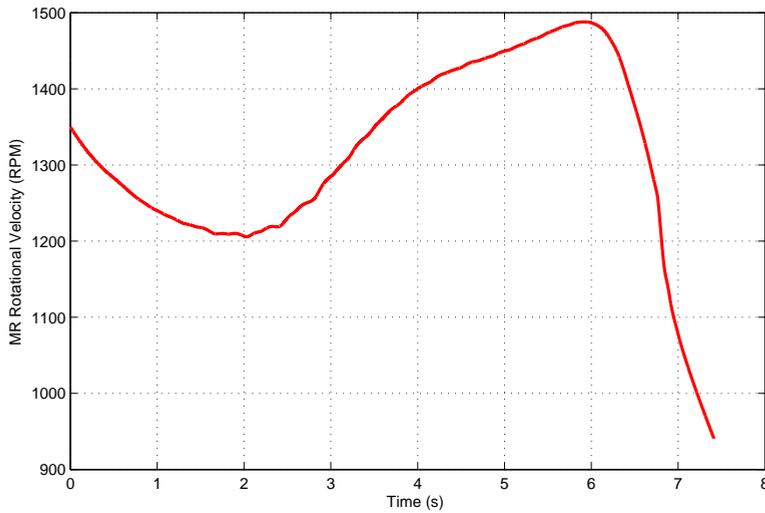
Figure 5.20: Main rotor RPM Ω_{MR}, for the Engine OFF case 2.
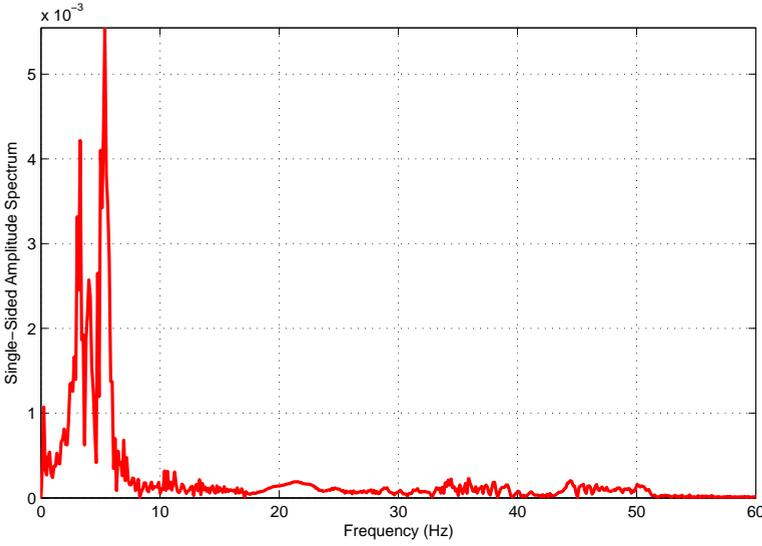
Figure 5.21: Amplitude spectrum of main rotor lateral TPP tilt angle β_{1s}, for the Engine OFF case 1 (the engine OFF case 2 is somewhat similar).
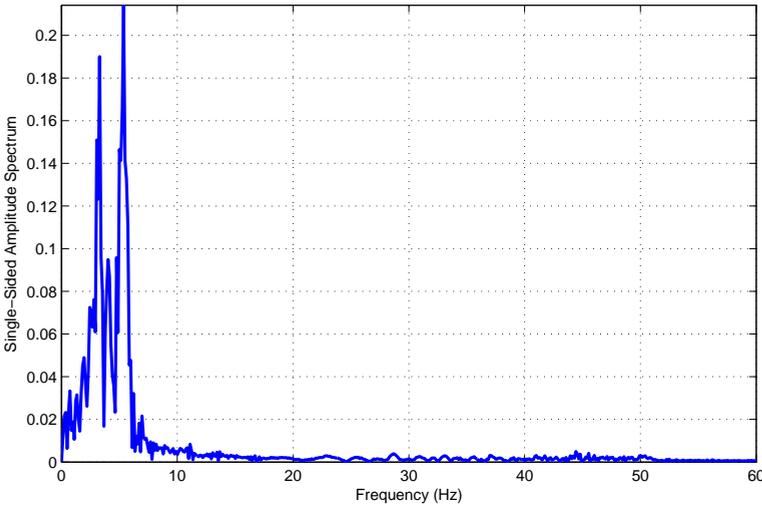
Figure 5.22: Amplitude spectrum of roll rate p, for the Engine OFF case 1 (the engine OFF case 2 is somewhat similar).
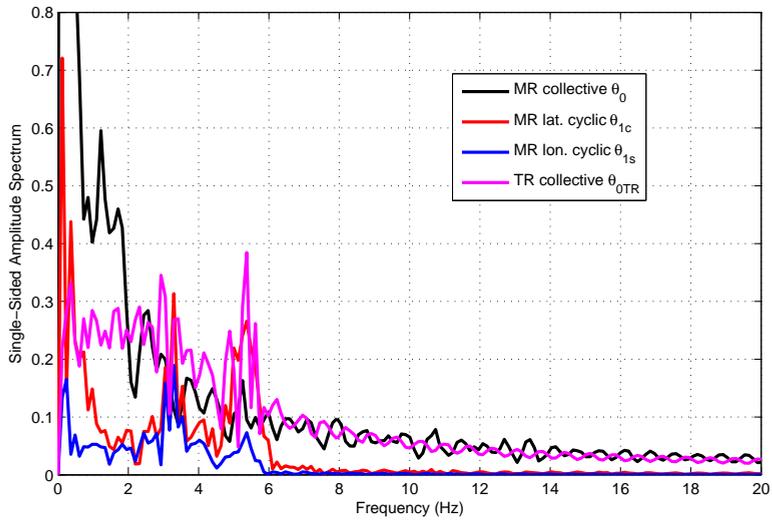
Figure 5.23: Amplitude spectrum of control inputs, for the Engine OFF case 1.
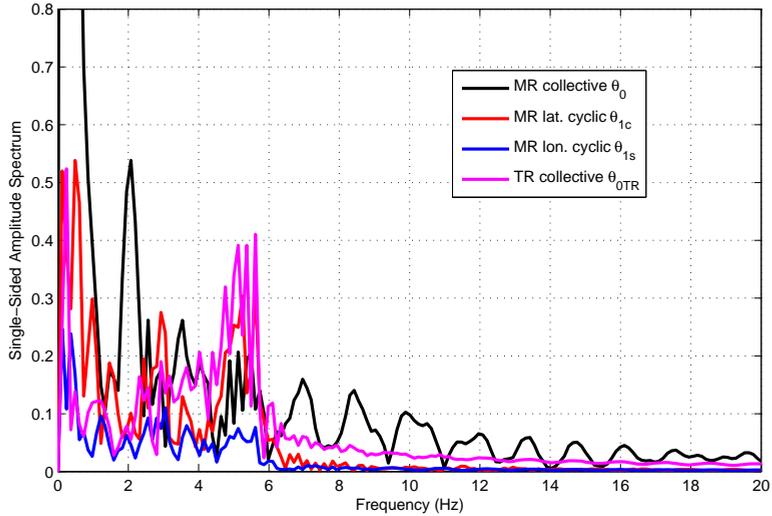
Figure 5.24: Amplitude spectrum of control inputs, for the Engine OFF case 2.
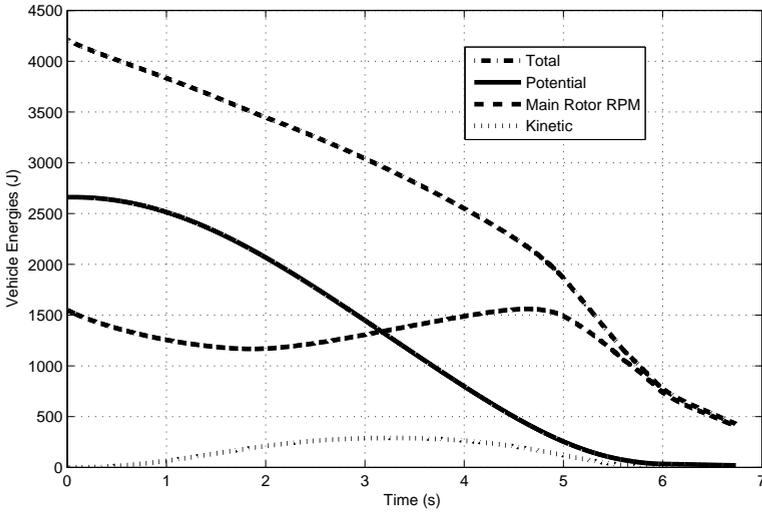
Figure 5.25: Vehicle energies, for the Engine OFF case 1.
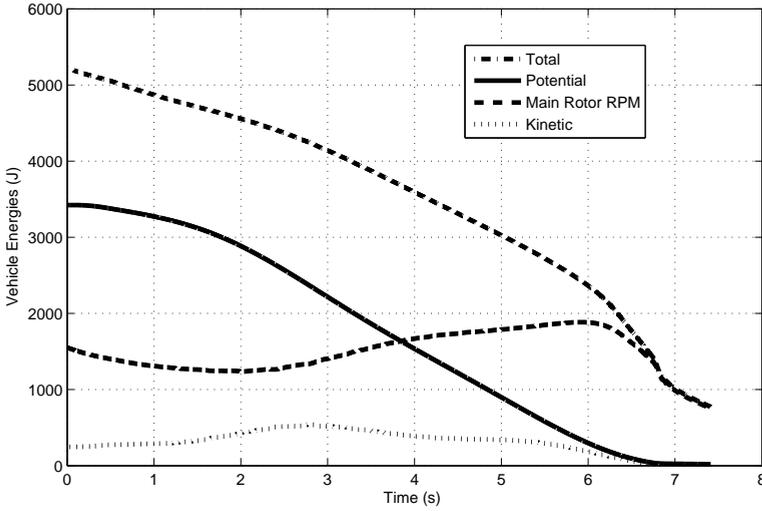
Figure 5.26: Vehicle energies, for the Engine OFF case 2.

5
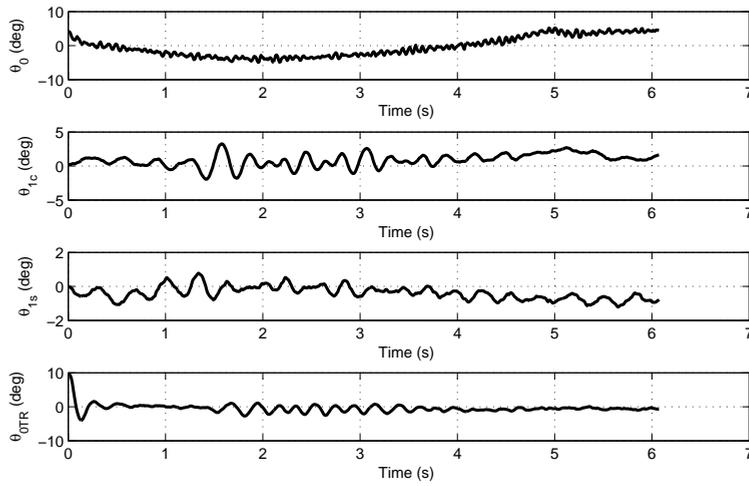
Figure 5.27: Helicopter control inputs, for the Engine OFF case 3.
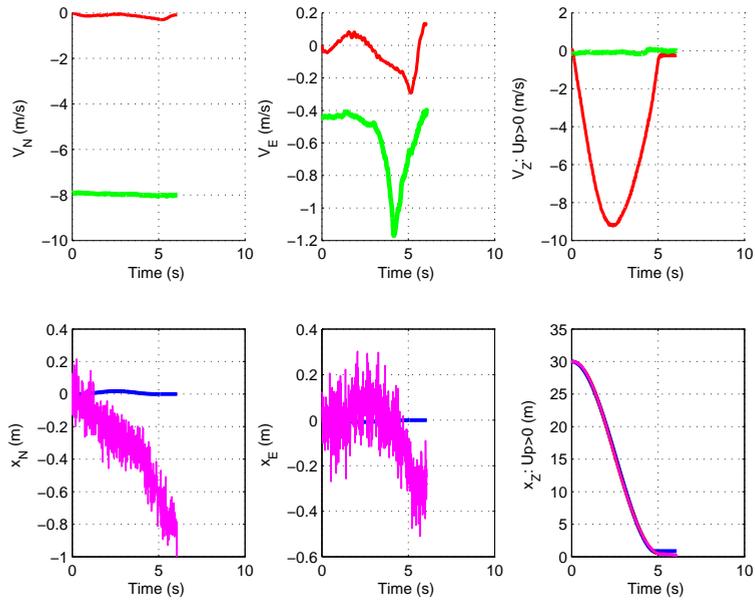
Figure 5.28: Inertial velocities and positions, for the Engine OFF case 3. Black line: flatness planning. Blue line: references for outer-loop (identical to black line). Red line: controlled nonlinear model. Magenta line: noisy measurements. Green line: wind disturbance.
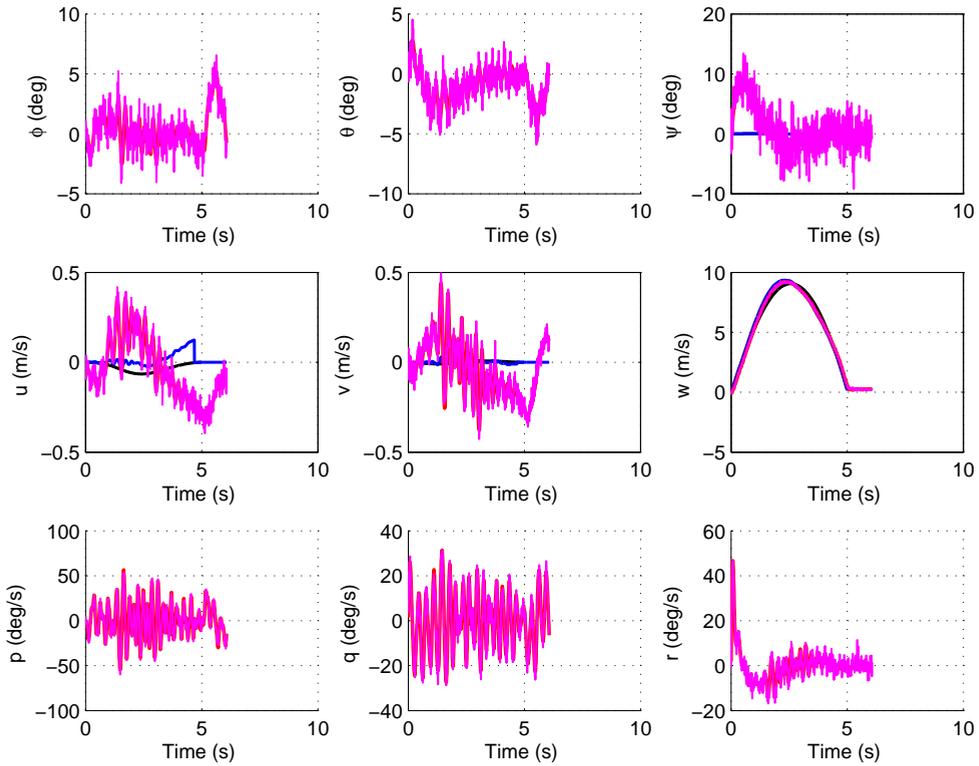
Figure 5.29: Euler angles, body linear velocities, and body rotational velocities, for the Engine OFF case 3. Black line: flatness planning. Blue line: references for inner-loop. Red line: controlled nonlinear model. Magenta line: noisy measurements.

5
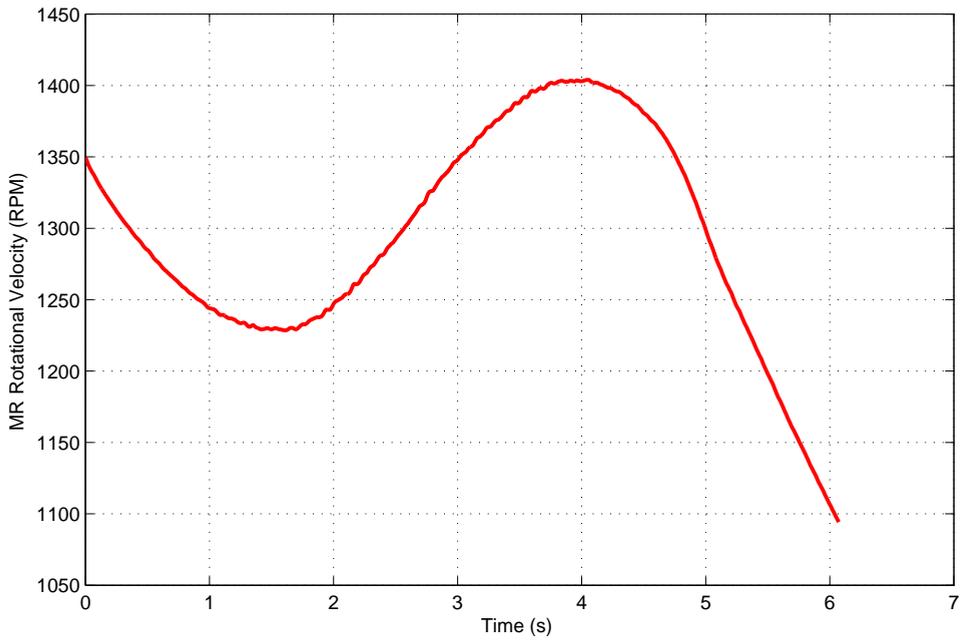
Figure 5.30: Main rotor RPM Ω_{MR}, for the Engine OFF case 3.

REFERENCES

[1] B. Mettler, *Identification Modelling and Characteristics of Miniature Rotorcraft* (Kluwer Academic Publishers, Norwell Mass, USA, 2003).

[2] S. Taamallah, *A qualitative introduction to the vortex-ring-state, autorotation, and optimal autorotation,* in *Europ. Rotorcraft Forum* (2010).

[3] R. T. N. Chen, *Flight dynamics of rotorcraft in steep high-g turns,* AIAA J. of Aircraft **21**, 14 (1984).

[4] S. S. Houston, *On the analysis of helicopter flight dynamics during maneuvers,* in *Europ. Rotorcraft Forum* (1985).

[5] R. Celi and F. D. Kim, *Trim, stability and frequency response simulation of an articulated rotor helicopter in turning flight,* in *Europ. Rotorcraft Forum* (1993).

[6] *http://gemiddeldgezien.nl/meer-gemiddelden/176-gemiddelde-windkracht-nederland*.

[7] R. T. N. Chen, *Effects of Primary Rotor Parameters on Flapping Dynamics*, Tech. Rep. NTP 1431 (NASA Ames Research Center, 1980).

[8] R. T. N. Chen and W. S. Hindson, *Influence of high-order dynamics on helicopter flight control system bandwidth,* AIAA J. of Guidance, Control, and Dynamics **9**, 190 (1986).

[9] H. C. Curtiss, *Stability and control modeling,* in *Europ. Rotorcraft Forum* (1986).

[10] V. Gavrilets, A. Shterenberg, M. A. Dahleh, and E. Feron, *Avionics system for a small unmanned helicopter performing aggressive maneuvers,* in *Digital Avionics Systems Conf.* (2000).

[11] V. Gavrilets, *Autonomous Aerobatic Maneuvering of Miniature Helicopter*, Ph.D. thesis, Massachusetts Institute of Technology (2003).

[12] V. A. Riziotis, E. S. Politis, S. G. Voutsinas, and P. K. Chaviaropoulos, *Stability analysis of pitch-regulated, variable speed wind turbines in closed loop operation using a linear eigenvalue approach,* Journal of Physics **75** (2007).

[13] D. Patt, L. Liu, J. Chandrasekar, D. S. Bernstein, and P. P. Friedmann, *Higher-harmonic-control algorithm for helicopter vibration reduction revisited,* AIAA J. of Guidance, Control and Dynamics **28**, 918 (2005).

[14] M. J. Patil, *Nonlinear Aeroelastic Analysis, Flight Dynamics, and Control of a Complete Aircraft*, Ph.D. thesis, Georgia Institute of Technology (1999).

5

6

AFFINE LPV MODELING

There is this famous quote that the theory of nonlinear systems is like a theory of non-elephants. It is impossible to build a theory of nonlinear systems, because arbitrary things can satisfy that definition.

Pablo Parillo
MIT News, 2010

In Chapter 5, a single nominal Linear Time-Invariant (LTI) model was used for the design of a single robust LTI Trajectory Tracker (TT). This LTI controller was capable of landing the helicopter in autorotation. Simulations showed that the crucial control of vertical position and velocity exhibited outstanding behavior in terms of tracking performance, although the tracking of horizontal positions and velocities was lacking some bandwidth. Increasing the horizontal closed-loop bandwidth was investigated by testing modified LTI controllers which, when evaluated on the nominal LTI model, showed promising results. Unfortunately, closed-loop instability was observed when evaluated on the NonLinear (NL) model of Chapter 2. Hence, improving the performance of the TT may necessitate an approach that better exploits the system's NL structure, while being computationally tractable (for on-line use). Linear Parameter-Varying (LPV) systems have become celebrated as they represent an attractive midway approach between LTI and NL structures, and hence LPV control could potentially be applied to improve the performance of the TT. However, the LPV control paradigm takes the existence of the plant, in LPV form, as a starting point. Since a systematic formulation of a NL system into a suitable LPV model remains often problematic, the purpose of this Chapter is to present an affine LPV modeling approach—for the case where a plant's NL model already exists—that delivers a model suitable for control design. Our LPV modeling method was applied to the helicopter NL model of Chapter 2, and resulted in a LPV model having a large number of scheduling parameters. Unfortunately, it became impossible to synthesize LPV controllers for such a high-order model, and hence the simulations in this Chapter have been done on a simpler pendulum system.

Parts of this Chapter have been published in [1].

6.1. Introduction

I N Chapter 4, we presented a combined trajectory planning and tracking system, capable of safely landing the helicopter, in autorotation, in the vicinity of the planned North-East landing spot. Our simulations of Chapter 5 showed that the tracking of horizontal positions and velocities was lacking some bandwidth. Increasing the horizontal closed-loop bandwidth was investigated, by testing upgraded controllers. These controllers, when evaluated on the nominal Linear Time-Invariant (LTI) model, showed promising results. Unfortunately, closed-loop instability was observed when tested on the helicopter, NonLinear (NL), High-Order Model (HOM) of Chapter 2. Hence, tracking performance of horizontal positions and horizontal velocities could potentially be improved by considering one of the two following options: i) remaining in the framework of a single robust LTI controller, using a higher-order LTI plant for controller design (i.e. containing the main rotor flap-lag and inflow dynamics), instead of the low-order plant used in Section 4.5.1 of Chapter 4; or ii) using another control method that better respects and exploits the system's NL structure, while being also on-line computationally tractable. To this end, the three main alternatives are: 1) Nonlinear Dynamic Inversion (NDI) and/or Lyapunov based methods such as sliding mode and backstepping; 2) methods in the realm of adaptive control; or 3) methods in the areas related to gain-scheduling and Linear Parameter-Varying (LPV) approaches.

Now the first option, i.e. option (i) here-above, with the use of a higher-order LTI plant (potentially in combination with a reduced-order observer to estimate the unmeasured higher-order rotor dynamics) is attractive for its simplicity, and hence is worth investigating. However it was ruled out in this Chapter since, as stated in [2], it is generally not recommended to 'hard-wire' the higher-order main rotor dynamics into the feedback law, whenever these higher-order dynamics are insufficiently well-captured by an LTI model (which in practice may often be the case). Hence in this Chapter we have chosen to investigate option (ii), and in particular the third alternative, i.e. the LPV approach, since there is a plethora of mature LPV control methods, and Model Predictive Control (MPC) for LPV systems, to choose from. This said, the first two alternatives should also be investigated in future research projects. In particular the recent and promising developments of the L_1 adaptive control [3] deserve further attention.

LPV systems allow to enclose NL behaviors into a linear framework [4, 5]. In fact, LPV control methods can be seen as an extension of the standard H_2 and H_∞ LTI synthesis techniques [6–13]. The LPV method amends also the main drawbacks of classical gain-scheduling [14, 15] by: 1) eliminating the need for repeated designs/simulations, in order to handle the global control problem; and 2) guaranteeing both stability and performance, along all possible parameter trajectories. In addition LPV control design problems are efficiently solved, by first expressing the problems as Linear Matrix Inequality (LMI) optimizations [16]—subsequently formulated as Semi-Definite Programs (SDP) [17]—for which there are several powerful numerical solutions [18, 19]. This resulted in a growing number of applications [20], such as in aerospace [21–26], wind turbines [27], wafer steppers [28, 29], CD players [30], and robotic manipulators [31], to name a few. Now, and for all its benefits, the LPV control paradigm typically takes the existence of a model of the plant, in LPV form, as a starting point. However, a systematic formulation of a NL system

into a suitable (quasi-)LPV[1] model remains often problematic [32]. Hence, the problem of simplifying a large scale, complex, NL model, such as our helicopter nonlinear HOM of Chapter 2, into a LPV representation, suitable for control design, is highly relevant. When a plant's NL model is already available, there exists two main modeling avenues to transform, or approximate, its NL representation into a LPV one, namely the so-called local and global approaches [32, 33]. The local approach consists in applying linearization theory of the NL system to obtain local LTI models in a state-space form, and subsequently interpolate these models to derive a LPV approximation. Within this framework several methods have been developed, based upon e.g.: extended linearization [34], Jacobian linearization [35], multiple-model design procedure [36], $\mathcal{H}_2$ norm minimization [37], multivariable polynomial fitting [38], and poles, zeros, and gain interpolation [39, 40]. On the other hand the global approach generates a LPV model which preserves the dynamic behavior of the NL system. This can either be done by using a range of mathematical manipulations e.g.: state transformation [41], velocity-based formulation [42], function substitution [43, 44], and automated LPV model generation [32, 45], or alternatively by using a global identification approach of the scheduling parameters, through the use of least-squares based estimations or Prediction Error Methods (PEM) [46].

Often it is important that the global behavior of the LPV model be similar to the global behavior of the NL system. This is typically the case when the LPV model is used for prediction/simulation in open-loop [47], MPC or optimal control. On the other hand, it is sometimes desirable that the local (frozen) behavior of the LPV model, i.e., for constant scheduling, be representative of the local behavior of the NL system, i.e., local linearizations of the NL system. For such cases, a local approach would be recommended[2]. This is particularly the case when the LPV model is used for gain scheduled controller design, where controllers are synthesized on the basis of local models.

For LPV systems, the simultaneous identification[3] of the LPV basis functions and scheduling parameters is a non-trivial problem, as it generally contains excessive degrees of freedom, giving rise to an ill-conditioned system identification problem [48]. Previous attempts towards such simultaneous identification problems have used nonlinear optimization methods [49, 50]. Another approach to mitigate such ill-conditioned identification problems requires the inclusion of additional constraints or regularizations [47]. An even simpler way would consist in having separate identification sub-problems, e.g. by identifying first the basis functions, followed by a separate identification of the scheduling parameters. We opt here for such a philosophy, i.e. by following the three-step methodology introduced in [36], formulated as follows: 1) identify first a central LTI model; 2) identify the basis functions; and finally 3) identify the scheduling parameters. Now, the method in [36] generates a model which is highly effective for open-loop prediction and simulation,

---

[1] The *quasi-* prefix is used to define LPV systems in which the scheduling parameters are endogenous, i.e. dependent of system states and/or control inputs [20].

[2] Note that global embedding of the behavior of a nonlinear system into an LPV representation often does not imply that the frozen aspects of the LPV models will have anything in common with the local linearizations of the NL system [47, 48].

[3] Throughout this Chapter, and since the NL system is known, we use *LPV modeling* and *LPV identification* interchangeably.

however the obtained LPV model is not truly in LPV form[4], and hence can not be used for LPV control design. Since our goal is modeling for control, we present in this Chapter an alternative approach that, among others, delivers a LPV model suitable for LPV controller synthesis.

Our method is based upon local linearizations of the NL system, along a nominal trajectory, followed by an interpolation procedure. Specifically, our modeling method consists in: 1) applying linearization of the NL system in order to obtain a set of local LTI models in state-space form, and a set of affine remainder terms resulting from (among others) linearizations of the NL system at non-equilibrium points; 2) finding a central model within this set of local LTI models; 3) using Singular Value Decompositions (SVD) tools to derive two sets of LPV basis functions[5]; 4) for the two sets of LPV basis functions, identify two respective sets of LPV scheduling parameters[6]; and 5) using a Neural Networks (NN) based approach to convert the LPV model into a quasi-LPV one, such that the scheduling parameters may be estimated on-line.

Our method is identical to the *glocal*[7] method of [36], with respect to item 1), and with respect to the SVD-based machinery used to obtain the first set of LPV basis functions in item 3). Our method differs from [36], as follows: a) first it generates a representation which is truly in LPV form, as it provides a model for the affine remainder terms, and hence allows to use the LPV model for controller design, over the complete operating regime (hence valid also at off-equilibria points); b) the choice of the central model and the choice of the first set of scheduling parameters are set within the $\mathcal{H}_\infty$ norm framework, as most robustness results are expressed in terms of $\mathcal{H}_\infty$ distances; and finally c) our method allows the user to specify an input-signal frequency range of interest, on which the local LTIs should best be approximated[8]. In fact, our method is in spirit more reminiscent of the so-called Jacobian linearization, or linearization gain-scheduled controller [5, 35], in which linearized plants along equilibria (or alternatively a trajectory), associated with local deviation signals, are used to design a parametrized family of linear controllers. Our modeling approach could perhaps be seen as an extension of these methods since our approach does not rely upon local deviation signals and hence can be used to approximate the NL behavior of the plant at off-equilibria points.

The LPV modeling method, presented in this Chapter, was applied to the helicopter NL model of Chapter 2 and resulted in a LPV model having a large number of (i.e. more than thirty) scheduling parameters. Unfortunately it became impossible to synthesize LPV controllers with such a high-order LPV model. It is indeed well known that the numerical conditioning and solvability of LMI problems play a crucial role in LPV practical design

---

[4]This aspect will be discussed later, starting with Eq. (6.7).
[5]The first set of basis functions is used to approximate the local LTI models, whereas the second set is used to approximate the affine remainder terms.
[6]The first set of scheduling parameters is obtained by minimizing the $\mathcal{H}_\infty$ distance between the frozen-scheduling LPV models and the respective LTIs, whereas the second set is obtained by minimizing the $\mathcal{L}_2$ norm of a vector.
[7]The acronym *glocal* stands for the combination of both *global* and *local*.
[8]This is done since, for controller synthesis, design specifications are typically generated for specific frequency ranges.

methods [27–30]. Hence the simulation results, presented in this Chapter, have been done on a simpler example, i.e. a modified pointmass pendulum. Although our focus is primarily set upon LPV modeling for control, we provide extensive analysis of, both, open-loop and closed-loop simulation results to illustrate the practicality of the method.

The remainder of this Chapter is organized as follows. In Section 6.2, the general LPV modeling and optimization problems are defined. In Section 6.3 through 6.8, a step by step modeling approach is described, and solutions to the optimization problems are derived. In Section 6.9, open-loop and closed-loop simulation results are analyzed, using H_∞, μ, and two LPV controllers. Finally, conclusions and future directions are presented in Section 6.10.

The nomenclature is fairly standard. Vectors are printed in boldface. $M^\top$, M^*, $M^\dagger$ denote the transpose, the complex-conjugate transpose, and the Moore-Penrose inverse of a real or complex matrix M, whereas $\mathrm{He}(M)$ (resp. $\mathrm{Sym}(M)$) is shorthand for $M + M^*$ (resp. $M + M^\top$). We use $\star$ as an ellipsis for terms that are induced by symmetry. Matrix inequalities are considered in the sense of *Löwner*. Further $\lambda(M)$ denotes the zeros of the characteristic polynomial $\det(sI - M) = 0$. $\mathcal{L}_\infty$ is the *Lebesgue* normed space s.t. $\|G\|_\infty := \sup_{\omega \in \mathbb{R}} \bar\sigma(G(j\omega)) < \infty$, with $\bar\sigma(G)$ the largest singular value of matrix $G(\cdot)$. Similarly, $\mathcal{H}_\infty \subset \mathcal{L}_\infty$ is the *Hardy* normed space s.t. $\|G\|_\infty := \sup_{\mathrm{Re}(s)>0} \bar\sigma(G(s))$. For $\omega_1 < \omega_2$, $\Delta_\omega = [\omega_1, \omega_2]$, we use $\|G\|_{\Delta_\omega} := \sup_{\omega \in \Delta_\omega} \bar\sigma(G(j\omega))$. $\mathcal{RL}_\infty$ (resp. $\mathcal{RH}_\infty$) represent the subspace of real rational Transfer Functions (TFs) in $\mathcal{L}_\infty$ (resp. $\mathcal{H}_\infty$). For appropriately dimensioned matrices K and M, where the latter is partitioned as $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$, the lower Linear Fractional Transformation (LFT) is defined as $F_l(M, K) = M_{11} + M_{12}K(I - M_{22}K)^{-1}M_{21}$, and the upper LFT is defined as $F_u(M, K) = M_{22} + M_{21}K(I - M_{11}K)^{-1}M_{12}$ under the assumption that the inverses exist. For $M \in \mathbb{C}^{q \times p}$, the structured singular value $\mu_\Delta(M)$ of M, with respect to an uncertainty set $\Delta \subset \mathbb{C}^{p \times q}$, is defined as $\mu_\Delta^{-1}(M) := \min_{\Delta \in \Delta}\{\bar\sigma(\Delta) \,|\, \det(I - M\Delta) = 0\}$.

6.2. Problem statement

We suppose that a real-life system can be described by a known, NL state-space, Continuous-Time (CT), dynamical model

$$\forall t \geq 0 \qquad \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \qquad \mathbf{y}(t) = \tilde{f}(\mathbf{x}(t), \mathbf{u}(t)) \qquad (6.1)$$

with $f(\cdot)$, $\tilde{f}(\cdot)$, partially differentiable smooth functions, $\mathbf{x}(t) \in \mathcal{P}_x \subset \mathbb{R}^{n_x}$ the plant state, $\mathbf{y}(t) \in \mathcal{P}_y \subset \mathbb{R}^{n_y}$ the plant output, $\mathbf{u}(t) \in \mathcal{P}_u \subset \mathbb{R}^{n_u}$ the control input, t the time variable, and $\mathcal{P}_x, \mathcal{P}_y, \mathcal{P}_u$ some compact sets. In this simulation model, the simulated data is not perturbed by noise. Further, we assume that the simulation model perfectly describes the behavior of the NL system. However, as mentioned earlier, this model is deemed too complex for control design. Hence, our goal consists in approximating the NL functions $f(\cdot)$, $\tilde{f}(\cdot)$, in Eq. (6.1), by a quasi-LPV representation, suitable for μ or LPV control design. Next, and to simplify the problem's context, we consider here the approximation of function $f(\cdot)$ only; indeed procedures similar to the ones presented in the sequel for $f(\cdot)$ may also be applied

to approximate $\tilde{f}(\cdot)$. Hence, from now on we consider the case

$$\forall t \geq 0 \qquad \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \qquad \mathbf{y}(t) = \mathbf{x}(t) \tag{6.2}$$

Our procedure uses simulation data to identify a quasi-LPV model of the complex NL model $f(\cdot)$. For this purpose, we apply to simulation model Eq. (6.2) a typical input trajectory and store the corresponding output. This yields the following Input-Output (IO) signal sequence $\mathcal{Z}^N := \{\mathbf{u}(t_i), \mathbf{y}(t_i)\}_{i=1}^{N}$. Since in our case we consider $\mathbf{y}(t) = \mathbf{x}(t)$, the set $\mathcal{Z}^N$ is refereed in the sequel as $\mathcal{Z}^N := \{\mathbf{u}(t_i), \mathbf{x}(t_i)\}_{i=1}^{N}$. We also assume that this sequence is informative[9] enough for the identification of the quasi-LPV model, i.e. all relevant nonlinearities of the system given by Eq. (6.2) have been excited over the entire working area.

Remark 8 *We will encompass our discussion within the CT framework since stability and performance requirements, for controller synthesis, are generally much more conveniently expressed in this framework. In case an equivalent LPV Discrete-Time (DT) realization is needed, this may be easily achieved by, either, discretizing the obtained CT LPV model through one of the LPV discretization methods presented in [53] or, alternatively, by using the equivalent DT formulations of the machinery outlined in this Chapter.*

We denote now the affine LPV model we want to identify as

$$P(\boldsymbol{\theta}(\mathbf{x}(t), \mathbf{u}(t))) := \begin{cases} \dot{\mathbf{x}}(t) = A_0 \mathbf{x}(t) + B_0 \mathbf{u}(t) \\ + \sum_{r=1}^{R} \theta_r(\mathbf{x}(t), \mathbf{u}(t))\left(A_r \mathbf{x}(t) + B_r \mathbf{u}(t)\right) \end{cases} \tag{6.3}$$

with $\boldsymbol{\theta}(\mathbf{x}(t), \mathbf{u}(t)) := [\theta_1(\mathbf{x}(t), \mathbf{u}(t)), ..., \theta_R(\mathbf{x}(t), \mathbf{u}(t))]^{\top}$ the non-stationary scheduling parameters defined on the compact set $\mathcal{P}_{\theta}$, known as the scheduling space, and matrices $\{A_r, B_r\}_{r=0}^{R}$ of appropriate sizes, representing the basis functions. Further, we also choose to enclose our analysis within the affine LPV setting, with static scheduling-parameter dependence, as dynamic dependence may lead to difficulties in terms of controller design and implementation. There exists also a clear advantage in using the affine LPV structure. Indeed, previous work on Takagi-Sugeno (TS) fuzzy models, which exhibit similarities with LPV systems [54], has shown that, on a compact subset of the state and input space, the approximation of the NL model Eq. (6.2) by the affine LPV model Eq. (6.3) can be made arbitrarily accurate [47, 50].

6

Next we consider the situation where one needs to build a CT LPV model from sampled measurements of the CT signals $\mathbf{u}(t)$ and $\mathbf{y}(t)$. These DT signals, sampled with the sampling period $T_s > 0$, are denoted $\mathbf{u}(t_i) = \mathbf{u}(iT_s), i \in \mathbb{Z}$, as illustrated here for the input signal $\mathbf{u}(\cdot)$. Building a CT LPV model from samples of measured CT signals has been addressed recently in [55]. Our problem is here simpler since we are dealing with a noise-free NL model, avoiding thus the difficult question of CT random process modeling from a sampled CT noise source. Further, for LPV systems with static dependence, and concomitant

---

[9]Note that persistence of excitation, to ensure consistency and convergence of the estimation as understood in the LTI case [51], is an ill-defined concept in the LPV case [52]. Signal richness, referring to the informativity of a data set w.r.t. coefficient parametrization and model order, is a more suitable LPV concept [52], but has yet to be formalized within this context, and hence is not addressed further in our LPV modeling framework.

to classical discretization theory [56], if the sampled and free-CT signals (i.e. inputs and exogenous parameters) can be assumed to be piecewise constant on a sampling period, i.e. T_s is sufficiently small, then the CT output trajectory may be completely reconstructed from its sampled observations [32].

Non-stationary linearizations of the NL model, along a given trajectory, as suggested for the Gain-Scheduling (GS) modeling framework in [57–59], and for the LPV modeling framework in [36, 42, 60], have often been used to extend the validity of GS, or LPV, controllers to operating regions far from equilibrium points. When combined with a sufficiently small sampling period T_s, such an approach may allow to better capture the transient behavior of the NL model. Accordingly, we also choose to base our LPV modeling methodology upon such linearizations. The latter may be computed via first-order Taylor-series expansions, or via classical numerical perturbation methods. From Eq. (6.2) and set $\mathcal{Z}^N$, we create a set of triplet elements $\mathcal{Z}_{Lin}^N := \{\bar{A}_i, \bar{B}_i, \mathbf{d}_i\}_{i=1}^N$

$$\bar{A}_i = \left.\frac{\delta f(\mathbf{x},\mathbf{u})}{\delta \mathbf{x}^\top}\right|_{(\mathbf{x}_i,\mathbf{u}_i)} \quad \bar{B}_i = \left.\frac{\delta f(\mathbf{x},\mathbf{u})}{\delta \mathbf{u}^\top}\right|_{(\mathbf{x}_i,\mathbf{u}_i)}$$
$$\mathbf{d}_i = f(\mathbf{x}_i,\mathbf{u}_i) - \bar{A}_i\mathbf{x}_i - \bar{B}_i\mathbf{u}_i \tag{6.4}$$

with $\mathbf{d}_i$ the so-called affine remainder term. In Eq. (6.4) we have also used the shorthand $\mathbf{x}_i := \mathbf{x}(t_i), \mathbf{u}_i := \mathbf{u}(t_i)$ to streamline notations. We also define a sequence of CT LTI Transfer Functions (TFs) $\bar{G}_i(s) := \left[\begin{array}{c|c} \bar{A}_i & \bar{B}_i \\ \hline I & 0 \end{array}\right]$, with matrices of appropriate size. Now, for each operating point $(\mathbf{x}_i, \mathbf{u}_i)$, we can approximate the NL model Eq. (6.2), in a local neighborhood of $(\mathbf{x}_i, \mathbf{u}_i)$, as

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t),\mathbf{u}(t)) \approx \bar{A}_i\mathbf{x}(t) + \bar{B}_i\mathbf{u}(t) + \mathbf{d}_i \tag{6.5}$$

while having exact equivalence at each operating point

$$\dot{\mathbf{x}}(t_i) = f(\mathbf{x}_i,\mathbf{u}_i) = \bar{A}_i\mathbf{x}_i + \bar{B}_i\mathbf{u}_i + \mathbf{d}_i \quad i = 1, ..., N \tag{6.6}$$

The two sets we have defined, namely IO set $\mathcal{Z}^N$ and linearization set $\mathcal{Z}_{Lin}^N$, describe the behavior of the NL system Eq. (6.2) from a global and local perspective, respectively. Both will be used for the identification of our LPV model, resulting in a model valid for both open- and closed-loop applications. As stated earlier, for the identification of the LPV model we follow the three-step methodology introduced in [36], formulated as follows

- **Step 1** Identify the central model (A_0, B_0).

- **Step 2** Using (A_0, B_0), identify the basis functions $\{A_r, B_r\}_{r=1}^R$.

- **Step 3** Identify the scheduling parameters $\boldsymbol{\theta}(\mathbf{x}(t), \mathbf{u}(t))$.

Since our method builds upon results from [36], we first briefly recall this method. In [36], the following LPV model is being identified

$$\tilde{P}(\boldsymbol{\theta}(\mathbf{x}(t),\mathbf{u}(t))) := \left\{ \begin{array}{l} \dot{\mathbf{x}}(t) = A_0\mathbf{x}(t) + B_0\mathbf{u}(t) \\ + \sum_{r=1}^R \theta_r(\mathbf{x}(t),\mathbf{u}(t))\Big(A_r\mathbf{x}(t) + B_r\mathbf{u}(t) + \mathbf{d}_r\Big) \end{array} \right. \tag{6.7}$$

with $\{\mathbf{d}_r\}_{r=1}^R$ a set of basis vectors. Now, following the three-step structure outlined here-above, the data flow for the identification of the model given in Eq. (6.7) is depicted in Fig. 6.1.
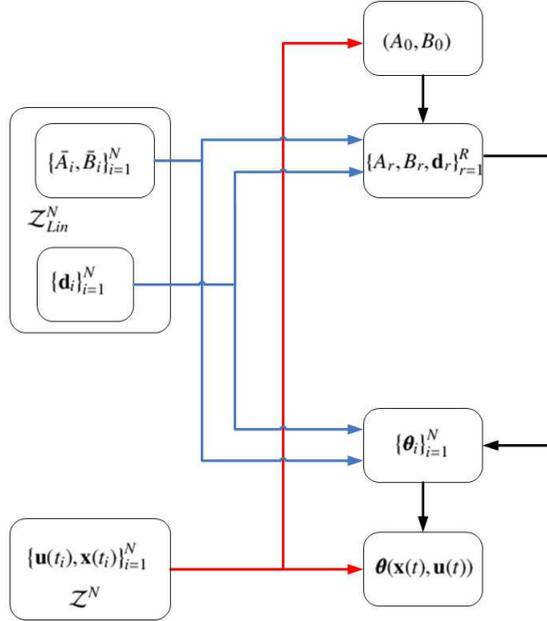
Figure 6.1: Data flow for the LPV identification method [36]. Lines in blue represent the information flow from the local system's behavior, present in set $\mathcal{Z}_{Lin}^N$. Lines in red represent the information flow from the global system's behavior, present in IO set $\mathcal{Z}^N$. Lines in black represent internal information flows.

We notice, among others, that: i) matrices (A_0, B_0) are identified on the basis of the global system's behavior, whereas matrices $\{A_r, B_r\}_{r=1}^R$ are identified on the basis of the local system's behavior; and ii) the scheduling parameters $\theta(\mathbf{x}(t), \mathbf{u}(t))$ are identified in a two-step procedure, defined as follows: first, a set of scheduling parameters $\{\theta_i\}_{i=1}^N$, i.e. for each time t_i, is being identified on the basis of the information available in the previously identified set $\{A_r, B_r, \mathbf{d}_r\}_{r=1}^R$ together with the data available in $\mathcal{Z}_{Lin}^N$, and next a CT mapping $\theta(\mathbf{x}(t), \mathbf{u}(t))$ is obtained by using the information available in the previously identified set $\{\theta_i\}_{i=1}^N$ together with the data available in IO set $\mathcal{Z}^N$.

The model in Eq. (6.7) allows to replace a computationally expensive, first-principles based, NL model with a computationally tractable alternative. Typical applications for the model in Eq. (6.7) include prediction/simulation in open-loop, e.g. on-line optimal trajectory planning. Now, the difference between Eq. (6.7) and Eq. (6.3), is that Eq. (6.7) contains an additional vector $\sum_{r=1}^R \theta_r(\mathbf{x}(t), \mathbf{u}(t))\mathbf{d}_r$, which role is to model the affine remainder terms $\{\mathbf{d}_i\}_{i=1}^N$. Strictly speaking, the model in Eq. (6.7) is neither in LPV form, nor in Piece-Wise-Affine (PWA) form [61, 62], but rather in a hybrid mix of both. Besides, and due to this additional vector, the model in Eq. (6.7) is not in a form suitable for LPV control

design. Hence, in this Chapter, we extend the approach developed in Eq. (6.7) in order to obtain a LPV model, suitable for both open- and closed-loop applications. To this end, we replace the model in Eq. (6.7) by the following quasi-LPV model

$$
P(\boldsymbol{\eta}(\mathbf{x}(t), \mathbf{u}(t)), \boldsymbol{\zeta}(\mathbf{x}(t), \mathbf{u}(t))) :=
\begin{cases}
\dot{\mathbf{x}}(t) = A_0 \mathbf{x}(t) + B_0 \mathbf{u}(t) \\
+ \sum_{s=1}^{S} \eta_s(\mathbf{x}(t), \mathbf{u}(t)) \big(L_s \mathbf{x}(t) + R_s \mathbf{u}(t) \big) \\
+ \sum_{w=1}^{W} \zeta_w(\mathbf{x}(t), \mathbf{u}(t)) \big(T_w \mathbf{x}(t) + Z_w \mathbf{u}(t) \big)
\end{cases}
\tag{6.8}
$$

for some scheduling parameters $\boldsymbol{\eta}(\mathbf{x}(t), \mathbf{u}(t)) := [\eta_1(\mathbf{x}(t), \mathbf{u}(t)), ..., \eta_S(\mathbf{x}(t), \mathbf{u}(t))]^\top$, $\boldsymbol{\zeta}(\mathbf{x}(t), \mathbf{u}(t)) := [\zeta_1(\mathbf{x}(t), \mathbf{u}(t)), ..., \zeta_W(\mathbf{x}(t), \mathbf{u}(t))]^\top$, and matrices (A_0, B_0), and $\{L_s, R_s\}_{s=1}^{S}$, $\{T_w, Z_w\}_{w=1}^{W}$, of appropriate sizes. Next, we present the multi-step philosophy used to identify the quasi-LPV model given in Eq. (6.8)

- **Step 1** Identify the central model (A_0, B_0) from the local system's behavior present in $\{\bar{A}_i, \bar{B}_i\}_{i=1}^{N}$, available in set $\mathcal{Z}_{Lin}^{N}$.

- **Step 2** Using (A_0, B_0), identify the basis functions $\{L_s, R_s\}_{s=1}^{S}$ from the local system's behavior present in $\{\bar{A}_i, \bar{B}_i\}_{i=1}^{N}$, available in set $\mathcal{Z}_{Lin}^{N}$.

- **Step 3** Identify the basis functions $\{T_w, Z_w\}_{w=1}^{W}$ from the local system's behavior present in $\{\mathbf{d}_i\}_{i=1}^{N}$, available in set $\mathcal{Z}_{Lin}^{N}$, and from the global system's behavior present in IO set $\mathcal{Z}^{N}$.

- **Step 4** Identify the scheduling parameters $\boldsymbol{\eta}(\mathbf{x}(t), \mathbf{u}(t))$ using, here-too, a two-step approach.

  - **Step 4.1** A set of scheduling parameters $\{\boldsymbol{\eta}_i\}_{i=1}^{N}$, i.e. for each time t_i, is being identified on the basis of the information available in the previously identified set $\{L_s, R_s\}_{s=1}^{S}$ together with the data available in set $\{\bar{A}_i, \bar{B}_i\}_{i=1}^{N}$. Basically, this step consists in obtaining a value of the scheduling parameters from linearizations at times t_i.

  - **Step 4.2** A continuous-time mapping $\boldsymbol{\eta}(\mathbf{x}(t), \mathbf{u}(t))$, that satisfies $\{\boldsymbol{\eta}(\mathbf{x}(t_i), \mathbf{u}(t_i))\}_{i=1}^{N} \approx \{\boldsymbol{\eta}_i\}_{i=1}^{N}$, is obtained by using the information available in the previously identified set $\{\boldsymbol{\eta}_i\}_{i=1}^{N}$ together with the data available in IO set $\mathcal{Z}^{N}$.

- **Step 5** Identify the scheduling parameters $\boldsymbol{\zeta}(\mathbf{x}(t), \mathbf{u}(t))$ using, here-too, a two-step approach.

  - **Step 5.1** A set of scheduling parameters $\{\boldsymbol{\zeta}_i\}_{i=1}^{N}$, i.e. for each time t_i, is being identified on the basis of the information available in the previously identified set $\{T_w, Z_w\}_{w=1}^{W}$ together with the data available in set $\{\mathbf{d}_i\}_{i=1}^{N}$.

  - **Step 5.2** A continuous-time mapping $\boldsymbol{\zeta}(\mathbf{x}(t), \mathbf{u}(t))$, that satisfies $\{\boldsymbol{\zeta}(\mathbf{x}(t_i), \mathbf{u}(t_i))\}_{i=1}^{N} \approx \{\boldsymbol{\zeta}_i\}_{i=1}^{N}$, is obtained by using the information available in the previously identified set $\{\boldsymbol{\zeta}_i\}_{i=1}^{N}$ together with the data available in IO set $\mathcal{Z}^{N}$.

6

Following this five-step structure, the data flow for the identification of the model given in Eq. (6.8) is depicted in Fig. 6.2.
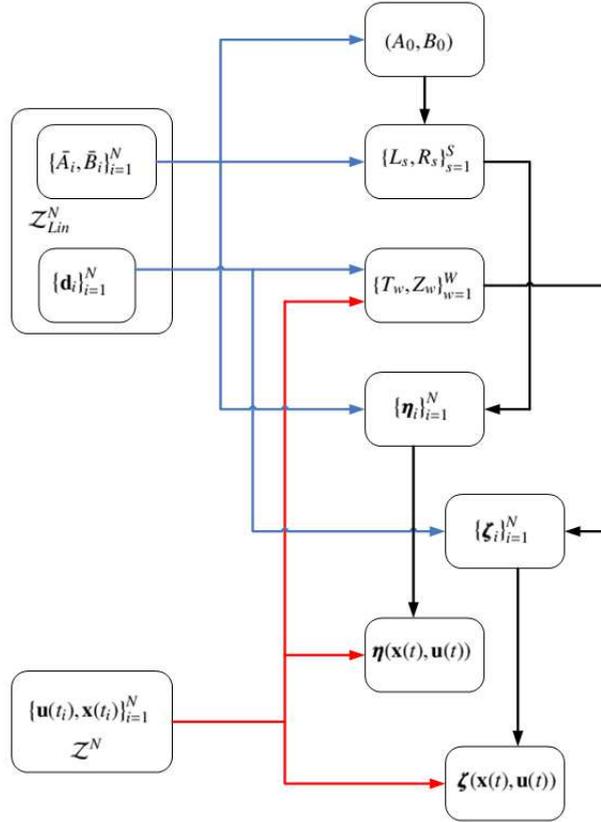
Figure 6.2: Data flow for the identification of our LPV model Eq. (6.8). Lines in blue represent the information flow from the local system's behavior, present in set $\mathcal{Z}^N_{Lin}$. Lines in red represent the information flow from the global system's behavior, present in IO set $\mathcal{Z}^N$. Lines in black represent internal information flows.

Remark 9 *LPV properties cannot in general be inferred from underlying LTI properties, i.e. frozen-scheduling deductions do not generally ensure that LPV modeling characteristics will be preserved with rapid parameter variations [63]. Hence, no formal proofs of convergence between the NL model and our LPV model may be given via this engineering practice.*

Remark 10 *Step 4.2 and Step 5.2 allow to use the model given by Eq. (6.8) for LPV control design. Indeed, without the knowledge of the mappings $\eta(\mathbf{x}(t), \mathbf{u}(t))$, and $\zeta(\mathbf{x}(t), \mathbf{u}(t))$, one would be restricted to potentially more conservative μ control methods, since the scheduling parameters cannot be estimated on-line. Note that finding such smooth mappings is a non-trivial task, and may even require some leap of faith, which one may be willing to take in case the entire working area has been sampled with a dense enough grid.*

Remark 11 *We restrict our discussion to full-order modeling, i.e matrices $\bar{A}_i$ and (A_0, L_s, T_w) have same size (resp. $\bar{B}_i$ and (B_0, R_s, Z_w)).*

In the sequel we discuss, in more detail, our five step methodology.

6.3. Step 1: Identifying the central model (A_0, B_0)

As stated earlier, the model (A_0, B_0) is chosen within all models present in set $\{\bar{A}_i, \bar{B}_i\}_{i=1}^N$. A natural approach consists in finding the model which may be defined as the most *central* one. Further, we will base this model selection within the $\mathcal{H}_\infty$ framework[10], since our primary focus is on modeling for control. In addition, for controller synthesis, design specifications are typically generated on various frequency ranges of interest $\Delta_\omega = [\omega_1, \omega_2]$, $\omega_1 < \omega_2$, which led us to use the $\mathcal{H}_\infty$ norm on a frequency range of relevance, i.e. the $\| \cdot \|_{\Delta_\omega}$ distance metric defined in the introduction Section as $\|G\|_{\Delta_\omega} := \sup_{\omega \in \Delta_\omega} \bar{\sigma}(G(j\omega))$ for a TF $G(j\omega)$

(see also Appendix A). This central model, i.e. the optimal model, $\hat{G}_0(s) := \left[\begin{array}{c|c} \hat{A}_0 & \hat{B}_0 \\ \hline I & 0 \end{array} \right]$,

is chosen as follows: compute, for each model $\bar{G}_i(s) := \left[\begin{array}{c|c} \bar{A}_i & \bar{B}_i \\ \hline I & 0 \end{array} \right]$ $i \in \{1, ..., N\}$, the following mean μ_i and standard-deviation s_i as

$$\forall i \in \{1, ..., N\} \quad \mu_i = (1/N) \sum_{j=1}^N \|\bar{G}_i(s) - \bar{G}_j(s)\|_{\Delta_\omega}$$
$$s_i = \left[(1/N) \sum_{j=1}^N \left(\|\bar{G}_i(s) - \bar{G}_j(s)\|_{\Delta_\omega} - \mu_i \right)^2 \right]^{1/2} \tag{6.9}$$

where $\| \cdot \|_{\Delta_\omega}$ is obtained[11] by minimizing the bound γ subject to the LMI of Eq. (6.48) (see Appendix A). Next define the following extrema

$$\underline{\mu} = \min_i \mu_i, \ \bar{\mu} = \max_i \mu_i, \ \underline{s} = \min_i s_i, \ \bar{s} = \max_i s_i \tag{6.10}$$

The optimal model $\hat{G}_0(s)$ is now designated as $\hat{G}_0(s) := \bar{G}_{\hat{i}}(s)$, with the optimal index $\hat{i}$ resulting from a simple, and readily solved, mean versus standard-deviation minimization problem

$$\hat{i} = \arg \min_{i \in \{1, ..., N\}} \left(\rho \big([\mu_i - \underline{\mu}] / [\bar{\mu} - \underline{\mu}] \big)^2 + \big([s_i - \underline{s}] / [\bar{s} - \underline{s}] \big)^2 \right) \tag{6.11}$$

with ρ a user-defined weighting parameter.

6.4. Step 2: Identifying the basis functions $\{L_s, R_s\}_{s=1}^S$

Whereas the role of the central model $\hat{G}_0(s)$ consists in capturing the most significant linear behavior of the NL system, the role of the basis functions $\{L_s, R_s\}_{s=1}^S$ (together with the

---

[10]Even though several other norms could be used, the $\mathcal{H}_\infty$ norm provides guarantees on worst cases.

[11]There are three ways to compute $\| \cdot \|_{\Delta_\omega}$: 1) approximately, through frequency griding of the $\| \cdot \|_\infty$ norm; 2) exactly, through the LMI optimization problems presented in Appendix A; or 3) approximately, through a weighted $\mathcal{H}_\infty$ norm minimization, using a strictly-proper, bandpass filter W_f, centered at Δ_ω, leading to $\|W_f.(\cdot)\|_\infty$.

scheduling parameters) consists in capturing the NL behavior of the system. We know from Eq. (6.5) that the NL system may be approximated, in a local neighborhood of $(\mathbf{x}_i, \mathbf{u}_i)$, by $\dot{\mathbf{x}}(t) \approx \bar{A}_i \mathbf{x}(t) + \bar{B}_i \mathbf{u}(t) + \mathbf{d}_i$. As the affine remainder term $\mathbf{d}_i$ will be handled in the sequel, we consider here only the following local behavior of the NL system $\dot{\mathbf{x}}(t) \approx \bar{A}_i \mathbf{x}(t) + \bar{B}_i \mathbf{u}(t)$. Hence, the gap between the local NL behavior and the central model behavior may be characterized, in a local neighborhood of $(\mathbf{x}_i, \mathbf{u}_i)$, as follows

$$\delta \dot{\mathbf{x}}(t) = (\bar{A}_i - \hat{A}_0)\mathbf{x}(t) + (\bar{B}_i - \hat{B}_0)\mathbf{u}(t) \tag{6.12}$$

Now from Eq. (6.12) one can build the following set $\{\bar{A}_i - \hat{A}_0, \bar{B}_i - \hat{B}_0\}_{i=1}^{N}$, from which we may derive the basis functions $\{\hat{L}_s, \hat{R}_s\}_{s=1}^{S}$, through Singular Value Decompositions (SVD). Such approaches have successfully been applied in the realm of LPV modeling in [36, 64, 65]. The approach outlined in this paragraph is not based on any $\mathcal{H}_\infty$ norm considerations, rather it is identical to the highly efficient method presented in [36], and consists in first transforming the information present, in matrix form, in $\{\bar{A}_i - \hat{A}_0, \bar{B}_i - \hat{B}_0\}_{i=1}^{N}$ into a vectorized form. Now, let

$$\Upsilon = [1...1] \tag{6.13}$$

be a row vector of length N. Define next the following Φ and Ω matrices

$$\Phi = \left[\begin{array}{c} \text{vec}(\hat{A}_0) \\ \text{vec}(\hat{B}_0) \end{array} \right] \otimes \Upsilon \qquad \Omega = \left[\begin{array}{ccc} \text{vec}(\bar{A}_1) & ,..., & \text{vec}(\bar{A}_N) \\ \text{vec}(\bar{B}_1) & ,..., & \text{vec}(\bar{B}_N) \end{array} \right] \tag{6.14}$$

with $\text{vec}(\cdot)$ the vertical vectorization of a matrix, and $\otimes$ the Kronecker product. It is clear that the information contained in $\{\bar{A}_i - \hat{A}_0, \bar{B}_i - \hat{B}_0\}_{i=1}^{N}$ is now made available in $(\Omega - \Phi)$. Next, we can obtain a proper orthogonal decomposition of $(\Omega - \Phi)$ which gives the principal directions in the space of the coefficients of $\{\hat{L}_s, \hat{R}_s\}_{s=1}^{S}$. This is done by obtaining a SVD decomposition of the form

$$\Omega - \Phi = U\Sigma V^* \tag{6.15}$$

Finally, let matrix $U_{1..S}$, with $S \leq n_x(n_x + n_u)$, contain the first S columns of the left singular vector matrix U in Eq. (6.15), then each basis function pair (L_s, R_s) is simply recovered from the matricization[12] of each column of $U_{1..S}$. The chosen value for S will depend upon the considered application, and its 'optimal' value represents a trade-off between model accuracy and computational tractability of the control synthesis.

6.5. STEP 3: IDENTIFYING THE BASIS FUNCTIONS $\{T_w, Z_w\}_{w=1}^{W}$

The idea here consists in providing a model for the affine remainder terms $\{\mathbf{d}_i\}_{i=1}^{N}$. Suppose we can find basis functions $\{T_w, Z_w\}_{w=1}^{W}$ and scheduling parameters $\zeta(t_i) := [\zeta_1(t_i), ..., \zeta_W(t_i)]^\top$ such that

$$\forall i \in \{1, ..., N\} \quad \mathbf{d}_i \left[\begin{array}{c} \mathbf{x}_i \\ \mathbf{u}_i \end{array} \right]^\dagger \approx \left[\begin{array}{cc} \sum_{w=1}^{W} \zeta_w(t_i)T_w & \sum_{w=1}^{W} \zeta_w(t_i)Z_w \end{array} \right] \tag{6.16}$$

with $[\cdot]^\dagger$ the left inverse, then by right-multiplying both sides with $[\mathbf{x}_i^\top \ \mathbf{u}_i^\top]^\top$ we recover $\mathbf{d}_i \approx \sum_{w=1}^{W} \zeta_w(t_i)\left(T_w \mathbf{x}_i + Z_w \mathbf{u}_i\right)$. To determine the basis functions, we will again use SVDs.

---

[12]The operation that turns a vector into a matrix.

First, we construct the matrices Λ_i and Ψ such that

$$\Lambda_i = \mathbf{d}_i \begin{bmatrix} \mathbf{x}_i \\ \mathbf{u}_i \end{bmatrix}^\dagger \qquad \Psi = \begin{bmatrix} \text{vec}(\Lambda_1) & , ..., & \text{vec}(\Lambda_N) \end{bmatrix} \tag{6.17}$$

with $\text{vec}(\cdot)$ the vertical vectorization of a matrix. Next, we obtain a SVD decomposition of the form

$$\Psi = U\Sigma V^* \tag{6.18}$$

Now let matrix $U_{1..W}$, with $W \leq n_x(n_x + n_u)$, contain the first W columns of the left singular vector matrix U in Eq. (6.18), then each basis function pair $\{T_w, Z_w\}_{w=1}^W$ is simply recovered from the matricization of each column of $U_{1..W}$.

Remark 12 *Note that the approach outlined in **Step 3** could potentially have additional applications, within the LPV modeling problem, but also within the true context of system identification when identifying a system from noisy measurements.*

6.6. Step 4.1: Identifying the parameters $\{\eta_i\}_{i=1}^N$

We identify here the set of scheduling parameters $\{\boldsymbol{\eta}_i\}_{i=1}^N := \{\eta_1(t_i), ..., \eta_S(t_i)\}_{i=1}^N$ on the basis of the information available in the previously identified set $\{L_s, R_s\}_{s=1}^S$ together with the data available in set $\{\bar{A}_i, \bar{B}_i\}_{i=1}^N$. Indeed, since our focus is mainly on modeling for control, we choose to approximate the local behavior of the NL system Eq. (6.2). This is done by obtaining a value of the scheduling parameters from local linearizations, i.e. by approximating the LTI models $\bar{G}_i(s) := \left[\begin{array}{c|c} \bar{A}_i & \bar{B}_i \\ \hline I & 0 \end{array} \right]$ with the frozen-scheduling LPV model

$$G_i(s) := \left[\begin{array}{c|c} A_0 + \sum\limits_{s=1}^S \eta_s(t_i)L_s & B_0 + \sum\limits_{s=1}^S \eta_s(t_i)R_s \\ \hline I & 0 \end{array} \right], \text{ for } i = 1, ..., N. \text{ This can be formulated as}$$

follows: for a given user defined frequency range $\Delta_\omega = [\omega_1, \omega_2]$, find, at each time t_i, the optimal parameters $\{\hat{\boldsymbol{\eta}}(t_i)\}_{i=1}^N$, with $\hat{\boldsymbol{\eta}}(t_i) := [\hat{\eta}_1(t_i), ..., \hat{\eta}_S(t_i)]^\top$, that minimize

$$J_1(t_i) := \|\bar{G}_i(s) - G_i(s)\|_{\Delta_\omega} \quad i = 1, ..., N \tag{6.19}$$

Minimizing $J_1(t_i)$ in Eq. (6.19) is equivalent to minimizing a scalar variable, subject to the LMI of Eq. (6.48), or to the LMI of Eq. (6.49). These LMIs are function of decision variables P and Q, or F and K. Further, these LMIs are also function of matrices A and B, given hereunder in Eq. (6.22), which are dependent on the decision variables $\{\eta_1(t_i), ..., \eta_S(t_i)\}$. Due to the product of matrices P and Q (or F and K) with matrices A and B, these LMIs become nonlinear. In such situations the projection lemma has often been used to provide convex reformulations of the original problem. In our case, unfortunately, a straightforward application of the projection lemma is not achievable, due to the structured nature of our problem (see [66] for additional details). Hence, we choose to use an iterative approach to solve Eq. (6.19). The procedure has a two-stage modus operandi: an initialization stage, followed by a nonlinear-based refinement stage. The first stage computes reasonable guess values for $\hat{\boldsymbol{\eta}}(t_i)$. The idea here consists in approximating the maximum

6

gain of the LTI matrices, $\bar{A}_i$ and $\bar{B}_i$, in the following way

$$
\begin{aligned}
&\forall i \in \{1, ..., N\} \\
&X_A(\eta_s(t_i)) = \bar{A}_i - (A_0 + \sum_{s=1}^{S} \eta_s(t_i)L_s) \\
&X_B(\eta_s(t_i)) = \bar{B}_i - (B_0 + \sum_{s=1}^{S} \eta_s(t_i)R_s) \\
&\hat{\eta}(t_i) = \arg \min_{\eta_s(t_i)} \|X_A(\eta_s(t_i))\|_2 + \|X_B(\eta_s(t_i))\|_2
\end{aligned}
\tag{6.20}
$$

This is readily recast into the sum minimization of the $\mathcal{L}_2$-induced gains of two static operators

$$
\forall i \in \{1, ..., N\} \quad \underset{\eta_s(t_i),\gamma_A,\gamma_B}{\text{minimize}} \gamma_A + \gamma_B
$$
$$
\text{subject to} \quad \gamma_A > 0 \quad \gamma_B > 0
$$
$$
\begin{bmatrix} \gamma_A I & \star \\ X_A(\eta_s(t_i)) & I \end{bmatrix} > 0 \quad \begin{bmatrix} \gamma_B I & \star \\ X_B(\eta_s(t_i)) & I \end{bmatrix} > 0
\tag{6.21}
$$

Next, the second stage uses the initial guess values found in Eq. (6.21) in order to solve Eq. (6.19), through an iterative approach. Here $\| \cdot \|_{\Delta_\omega}$ is computed via Eq. (6.49) since the latter is convex in either the (F, K) or (A, B) matrices. These (A, B) matrices in Eq. (6.49) are given by

$$
\bar{G}_i(s) - G_i(s) := \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] = \left[\begin{array}{cc|c} \bar{A}_i & 0 & \bar{B}_i \\ 0 & A_0 + \sum_{s=1}^{S} \eta_s(t_i)L_s & B_0 + \sum_{s=1}^{S} \eta_s(t_i)R_s \\ \hline I & -I & 0 \end{array} \right]
\tag{6.22}
$$

Our proposed approach is a simple two-step iterative LMI search, in spirit reminiscent of D-K iteration synthesis [67]. First, partition F and K, as $F = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}$, and $K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$, with the sub-block sizes matching the partitions in Eq. (6.22). Next, the procedure reads as follows

1. Start with the initial value $\hat{\eta}(t_i)$ obtained from Eq. (6.21)

2. In Eq. (6.49) minimize γ with respect to (F, K)

3. Keep $(F_{12}, F_{22}, K_{12}, K_{22})$ from step 2 since these variables multiply the unknowns $\eta_s(t_i)$. Next in Eq. (6.49), minimize γ with respect to the free variables $(\hat{\eta}(t_i), F_{11}, F_{21}, K_{11}, K_{21})$

4. Repeat from 2 until convergence or maximum iteration reached

Remark 13 *Aside from D-K iteration, similar heuristics appear to work well in practice, such as model order reduction [68], LPV controller with parameter-dependent scalings [69], or gain-scheduled controller with inexact scheduling parameters [70]. Analogously to D-K iteration convergence—for which convergence towards a global optimum, or even a local one, is not guaranteed [71, 72]—the above iterative method does not inherit any convergence certificates, however in practice convergence has been achieved within a few iterations.*

Remark 14 *In Appendix B, we examine a specific case for which the optimal value of the scheduling parameters can be computed, thus avoiding any nonlinear iterative approach.*

6.7. Step 4.2: Obtaining the mapping $\eta(\mathbf{x}(t), \mathbf{u}(t))$

The aim is here to find a suitable representation, or smooth CT mapping $g(\cdot)$, that satisfies $\boldsymbol{\eta}(t) = g(\mathbf{x}(t), \mathbf{u}(t))$ and $\{\boldsymbol{\eta}(\mathbf{x}(t_i), \mathbf{u}(t_i))\}_{i=1}^N \approx \{\boldsymbol{\eta}_i\}_{i=1}^N$. To this end, this mapping will be obtained by using the information available in the previously identified set $\{\boldsymbol{\eta}_i\}_{i=1}^N$ together with the data available in IO set $\mathcal{Z}^N$.

Now, for physically-intuitive plants, one may select the required states and inputs in $g(\mathbf{x}(t), \mathbf{u}(t))$, based upon engineering judgment, and derive these mappings through popular curve-fitting methods. For non-transparent systems, i.e. exhibiting significant dependences among variables, one may consider formal/systematic tools such as: orthogonal/radial basis functions, principal component analysis, statistical analysis, fuzzy tools, or Neural Networks (NN). Regarding NN, it is well-known that, under mild assumptions on continuity and boundedness, a network of two layers[13] can be trained to approximate any IO relationship arbitrarily well, provided there are enough neurons in the hidden layer [73, 74]. Hence, NN have found a wide range of applications in control theory [75]. But despite their powerful features, NN have only seen limited usage in the LPV field [76–78]. This said, we choose here to base the $g(\cdot)$ modeling on NN. We will further illustrate the applicability of a two-layer feedforward NN, the first being sigmoid and the second linear, with l neurons (l large enough), such that

$$\boldsymbol{\eta}(t) = g(\mathbf{x}(t), \mathbf{u}(t)) = C_\eta . s_\eta(t) \tag{6.23}$$

with

$$s_\eta(t) = W_{o_\eta} . \kappa(W_{x_\eta} \mathbf{x}(t) + W_{u_\eta} \mathbf{u}(t) + W_{b_\eta}) \tag{6.24}$$

and $W_{o_\eta} \in \mathbb{R}^{S \times l}$, $W_{x_\eta} \in \mathbb{R}^{l \times n_x}$, $W_{u_\eta} \in \mathbb{R}^{l \times n_u}$ containing the output and hidden layer weights. Further, $W_{b_\eta} \in \mathbb{R}^l$ contains the sets of biases in the hidden layer, $C_\eta \in \mathbb{R}^{S \times S}$ contains the output linear maps, and $\kappa(\cdot)$ is the activation function, taken as a continuous, diagonal, differentiable, and bounded static sigmoid nonlinearity. Here, all NN models will be based upon a classical feedforward network, with the hyperbolic tangent activation transfer function in the hidden layer, and backpropagation training for the weights and biases.

6.8. Steps 5.1 and 5.2: Identifying the parameters $\{\zeta_i\}_{i=1}^N$ and obtaining the mapping $\zeta(\mathbf{x}(t), \mathbf{u}(t))$

We identify here the set of scheduling parameters $\{\zeta_i\}_{i=1}^N := \{\zeta_1(t_i), ..., \zeta_W(t_i)\}_{i=1}^N$ on the basis of the information available in the previously identified set $\{T_w, Z_w\}_{w=1}^W$ together with the data available in set $\{\mathbf{d}_i\}_{i=1}^N$. This problem may be formulated as follows: find, at each time t_i, the optimal parameters $\{\hat{\boldsymbol{\zeta}}(t_i)\}_{i=1}^N$, with $\hat{\boldsymbol{\zeta}}(t_i) := [\hat{\zeta}_1(t_i), ..., \hat{\zeta}_W(t_i)]^\top$, that minimize

$$J_2(t_i) := \|\mathbf{d}_i - \sum_{w=1}^W \zeta_w(t_i)\big(T_w \mathbf{x}_i + Z_w \mathbf{u}_i\big)\|_2 \quad i = 1, ..., N \tag{6.25}$$

---

[13]The first being hidden sigmoid and the second linear.

6

Remark 15 *In Eq. (6.25) we have based the optimization on the $\mathcal{L}_2$ norm of a vector, as it is computationally very cheap. An alternative approach would be to consider the $\mathcal{L}_\infty$ norm of a vector, in order to be consistent with the identification of the scheduling variables $\{\eta_i\}_{i=1}^N$ in Section 6.6.*

Now using the Λ_i matrix defined in Eq. (6.17), we can rewrite Eq. (6.25) as

$$\begin{aligned}
&\forall i \in \{1, ..., N\} \\
&\hat{\zeta}(t_i) = \arg \min_{\zeta_w(t_i)} \|\text{vec}(\Lambda_i) - U_{1..W} \, [\zeta_1(t_i), ..., \zeta_W(t_i)]^\top\|_2^2
\end{aligned} \tag{6.26}$$

which can be solved through linear least-squares. As $U_{1..W}$ is an orthogonal matrix, the solution of Eq. (6.26) reduces to

$$\forall i \in \{1, ..., N\} \quad \hat{\zeta}(t_i) = U_{1..W}^\top \, \text{vec}(\Lambda_i) \tag{6.27}$$

The reconstructed remainder term, used in the sequel within the model evaluations, is readily computed as

$$\forall i \in \{1, ..., N\} \quad \hat{\mathbf{d}}_i = \sum_{w=1}^W \hat{\zeta}_w(t_i)\Big(T_w \mathbf{x}_i + Z_w \mathbf{u}_i\Big) \tag{6.28}$$

The next step requires the determination of a suitable representation $h(\cdot)$, that satisfies $\zeta(t) = h(\mathbf{x}(t), \mathbf{u}(t))$ and $\{\zeta(\mathbf{x}(t_i), \mathbf{u}(t_i))\}_{i=1}^N \approx \{\zeta_i\}_{i=1}^N$. To this end, this mapping will be obtained by using the information available in the previously identified set $\{\zeta_i\}_{i=1}^N$ together with the data available in IO set $\mathcal{Z}^N$. The mapping $h(\cdot)$ is here as well based upon a NN representation, and the associated procedure is identical to the one of Section 6.7.

6.9. APPLICATION TO THE MODELING AND CONTROL OF A MODIFIED POINTMASS PENDULUM

The LPV modeling method, presented in this Chapter, was applied to the helicopter NL model of Chapter 2, and resulted in a LPV model having a large number of (i.e. more than thirty) scheduling parameters. Unfortunately, it became impossible to synthesize LPV controllers with such a high-order LPV model. Hence, the simulation results, presented in this Chapter, have been done on a simpler example, the pointmass pendulum. In this section, both Open-Loop (OL) and Closed-Loop (CL) analysis of our LPV modeling framework will further be discussed. Now, the rotational motion of the driven and damped, pointmass pendulum, is given by

$$\begin{aligned}
&\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ -bx_2(t) - a^2 \sin x_1(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \vartheta(u(t)) \end{bmatrix} \\
&\text{with} \quad \vartheta(u(t)) = c \sin u(t)
\end{aligned} \tag{6.29}$$

with $[x_1 \; x_2]^\top = [\theta \; \dot{\theta}]^\top$ the states, θ the rotation angle, u the input torque, $a = \sqrt{g/L}$ the angular frequency, g the acceleration due to gravity, L the pendulum length, see Fig. 6.3, b a measure of the dissipative force, with values: ($L = 3, b = 2$), and $\vartheta(\cdot)$ a fictional nonlinearity
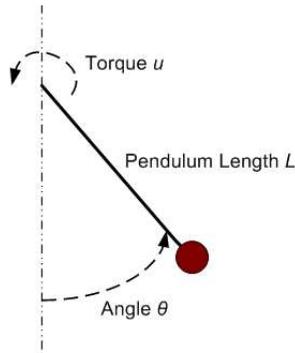
Figure 6.3: The pointmass pendulum example representing the nonlinear plant.

(with coefficient $c = 4$) with the intent of increasing the NL model generality. Obviously, the system in Eq. (6.29) can exactly be recast into quasi-LPV form, using a global approach, i.e. by choosing two scheduling parameters $\theta_1(\cdot)$ and $\theta_2(\cdot)$, such that $\theta_1(t) = \sin x_1(t)/x_1(t)$ and $\theta_2(t) = \sin u(t)/u(t)$. We have purposely chosen a simple example, as to better illustrate the practicality of our modeling method, which will be used to derive several LPV models.

6.9.1. BUILDING THE LPV MODELS

To derive the LPV models we excite the pendulum model from its rest position with a 20 s. long sine-sweep $u(t) = A \sin(2\pi.f.t)$, $A = 1$, with frequency f in the range 0.001–1 Hz, sampled with a period $T_s = 0.05$ s., resulting in 401 data points. The purpose is also to illustrate the applicability of our modeling method in a conservative context, i.e. for the case where the control input signal-richness (used for identification) is rather limited, as is the case with this single sine-sweep signal chosen here, and for the case of a relatively high sampling period, resulting thus in few data points (here only a few hundreds). Further we also use a frequency range of interest defined as a wide low-pass filter Δ_ω with $[\omega_1, \omega_2] = [0, 10]$ Hz, to be able to test the model at frequencies outside the 0.001–1 Hz band used during identification.

First, the *central* model $\hat{G}_0(s)$, obtained according to Eq. (6.11), with $\rho = 100$, is found to be model nr. 185, i.e. $G_{185}(s)$. Next, Table 6.1 and Table 6.2 are given to provide an overview of the SVD results—of Sections 6.4 and 6.5—used to derive the basis functions, where the *captured energy* refers to the percentage ratio between the sum of the retained singular values to the sum of all singular values.

Table 6.1: Number of retained basis functions, in the SVD decompositions of Section 6.4.

	Nr. of Basis Functions in $U_{1..S}$	
	S=2	S=1
Captured Energy of $U_{1..S}$ (%)	100	53

Table 6.2: Number of retained basis functions, in the SVD decompositions of Section 6.5.

	Nr. of Basis Functions in $U_{1..W}$		
	W=3	W=2	W=1
Captured Energy of $U_{1..W}$ (%)	100	79	51

From Table 6.1, we see that matrix $U_{1..S}$ has 2 columns, and hence the maximum value of S is 2. Similarly from Table 6.2, we see that matrix $U_{1..W}$ has 3 columns, and the maximum value of W is 3. To better analyze our modeling framework we will use three LPV models: the first two to evaluate the OL response, whereas the third one will be used for dynamic output feedback control design[14]. The first two assume full-information, whereas the third corresponds to the case where only state x_1 is measured. The first model, model M1, with $S = 2$, $W = 3$, retains all basis functions, and hence corresponds to the best model we can build. On the other hand, both models M2 and M3, with $S = 1$, $W = 1$, retain the least amount of energy in the basis functions, but are computationally most efficient. Summarizing, the three models are described as

1. **Model M1**. Generated with $S = 2$, $W = 3$, and a 10-neurons network with $\boldsymbol{\eta}(t) = g(x_1(t), x_2(t))$, $\boldsymbol{\zeta}(t) = h(x_1(t), x_2(t))$

2. **Model M2**. Generated with $S = 1$, $W = 1$, and a 10-neurons network with $\boldsymbol{\eta}(t) = g(x_1(t), x_2(t))$, $\boldsymbol{\zeta}(t) = h(x_1(t), x_2(t))$

3. **Model M3**. Generated with $S = 1$, $W = 1$, and a 10-neurons network with $\boldsymbol{\eta}(t) = g(x_1(t))$, $\boldsymbol{\zeta}(t) = h(x_1(t))$

Note that functions $g(\cdot)$ and $h(\cdot)$ are functions of the states only, rather than both states and inputs, since better validation results were obtained this way when exciting the LPV models with fresh inputs (i.e. inputs not used during the identification process). Next, to compare the effectiveness of the proposed LPV models, we define the following cost functions

1. **Cost C1**. For an evaluation of the optimization problem Eq. (6.19), we define the mean of the local TF deviation in terms of cost $J_{P_1} := \frac{1}{N} \sum_{i=1}^{N} J_1(t_i)$, with $J_1(\cdot)$ the cost function of Eq. (6.19), and N the data length.

2. **Cost C2**. For an evaluation of the optimization problem Eq. (6.25), we define the following cost $J_{P_2} := 100\% . \frac{1}{n_x} \sum_{k=1}^{n_x} \max\left(1 - \frac{\left\|\delta_k - \hat{\delta}_k\right\|_2}{\left\|\delta_k - \text{mean}(\delta_k)\right\|_2}, 0\right)$, with n_x the number of states, $\delta_k \in \mathbb{R}^N$ a time-domain vector representing the kth row of $\mathbf{d}$, the latter being defined in Eq. (6.4). Further, $\hat{\delta}_k \in \mathbb{R}^N$ is a time-domain vector representing the kth row of $\hat{\mathbf{d}}$, the latter being defined in Eq. (6.28).

---

[14]In most practical situations, when designing control systems, one does not have access to the full state-vector. In the case of the pendulum, often only the rotation angle θ is being measured.

For our NL system Eq. (6.2) recall that, at each operating point t_i, we had

$$\dot{\mathbf{x}}(t_i) = f(\mathbf{x}_i, \mathbf{u}_i) = \bar{A}_i \mathbf{x}_i + \bar{B}_i \mathbf{u}_i + \mathbf{d}_i \quad i = 1, ..., N$$

Hence the purpose of cost C1 is to check whether the LPV system defined by Eq. (6.30), with its scheduling parameters evaluated at a frozen-scheduling for time t_i, does (or not) represent a good approximation of the LTI system given here-under by Eq. (6.31).

$$P(\boldsymbol{\eta}(\mathbf{x}(t), \mathbf{u}(t))) := \left\{ \begin{array}{l} \dot{\mathbf{x}}(t) = A_0 \mathbf{x}(t) + B_0 \mathbf{u}(t) \\ + \sum\limits_{s=1}^{S} \eta_s(\mathbf{x}(t), \mathbf{u}(t)) \Big(L_s \mathbf{x}(t) + R_s \mathbf{u}(t) \Big) \end{array} \right. \tag{6.30}$$

$$\dot{\mathbf{x}}(t) = \bar{A}_i \mathbf{x}(t) + \bar{B}_i \mathbf{u}(t) \tag{6.31}$$

On the other hand, the purpose of cost C2 is to check whether the reconstructed remainder term $\hat{\mathbf{d}}_i$ at a frozen-scheduling for time t_i, defined by Eq. (6.28), does represent (or not) a good approximation of the remainder term $\mathbf{d}_i$, defined by Eq. (6.4). Note also that costs C1 and C2 evaluate the models before the inclusion of the NN component.

The results are given in Table 6.3, where all LMIs used to compute cost C1 are solved using YALMIP [79] with the SeDuMi solver [19]. For model M1, since we kept all basis functions, the cost functions J_{P_1} and J_{P_2} reveal a perfect match between Eq. (6.30) and Eq. (6.31), and between the remainder terms $\hat{\mathbf{d}}_i$ and $\mathbf{d}_i$ respectively. On the other hand, models M2 and M3 use the minimum set of basis functions. These models are equivalent in terms of J_{P_1} and J_{P_2}, since different only through their respective NN representation. We see that J_{P_2} is still high (which is good), and that the simple approach Eq. (6.20), to compute the scheduling parameters, gives a very low value for J_{P_1} (which is also good). For this example, we see that the NL refinement for J_{P_1} (to compute the scheduling parameters) is not even necessary, although on a different example [1] it did provide substantial improvements. This preliminary modeling review shows that models M2 and M3, although based on the minimum set of basis functions, may potentially provide good model fidelity in OL. In the sequel we provide additional evaluations of both OL and CL behavior.

Table 6.3: Cost Functions: J_{P_1} and J_{P_2}.

LPV Model Model	Costs		
	J_{P_1} from Eq. (6.20)	J_{P_1} from iterative refinement	J_{P_2} (%)
M1	0	N.A.	100
M2=M3	0.34	0.32	74

6.9.2. OPEN-LOOP ANALYSIS

To better compare the effectiveness of the proposed LPV models we define the following additional cost functions

1. **Cost C3**. For a comparison of time-domain outputs in $l_2[0, \infty)$, we use fresh data sets, namely step-inputs, and sine-inputs at varying amplitudes and frequencies, and compute the Best-FiT (BFT) $:= 100\%.\frac{1}{n_x} \sum\limits_{k=1}^{n_x} \max\left(1 - \frac{\left\|\mathbf{s}_k - \tilde{\mathbf{s}}_k\right\|_2}{\left\|\mathbf{s}_k - \text{mean}(\mathbf{s}_k)\right\|_2}, 0\right)$ with $\mathbf{s}_k \in \mathbb{R}^N$ a time-domain vector representing the kth row of $\mathbf{x}$ ($\mathbf{x}$ being the state-vector of the NL system), and similarly $\tilde{\mathbf{s}}_k \in \mathbb{R}^N$ being the LPV counterpart.

2. **Cost C4**. Using the variables defined for **C3**, we compute the Variance-Accounted-For (VAF)

   VAF $:= 100\%.\frac{1}{n_x} \sum\limits_{k=1}^{n_x} \max\left(1 - \frac{\text{var}(\mathbf{s}_k - \tilde{\mathbf{s}}_k)}{\text{var}(\mathbf{s}_k)}, 0\right)$. Roughly speaking the VAF tends to capture signal closeness in terms of their respective "shapes".

In this section we have added the NN part to the LPV models (we use the NN MATLAB Toolbox). All models become now quasi-LPV models (also written as qLPV). We will compare next the behavior of the CT quasi-LPV models with that of the CT NL system. We excite the quasi-LPV models with data sets not used during the modeling build-up. First, we use sine-inputs, for several fixed amplitudes and fixed frequencies (again not used during identification), and present the respective BFT and VAF for each model in Table 6.4 through 6.6.

Overall all three models exhibit very good to excellent fit with the NL model, for input amplitudes below one (i.e. the value used during identification). The accuracy of these quasi-LPV models diminishes when the input amplitude is increased above one, even though model M2 still retains a very good fit. We also note that model M2, even though based on fewer basis functions than M1, is roughly at least as good as model M1. This may be explained by the fact that the NN models were trained with a very small data set. Indeed good identification data sets may be two orders of magnitude bigger, in the tens of thousands of points rather than a few hundreds [80]. Hence, and even though there is no measurement noise in these simulations, a model with fewer to-be-estimated parameters, like M2, may provide, in this case, a higher quality model. The fit for model M3 is slightly worse than that of M2, e.g. for input amplitudes above one. This may be explained by the fact that the identification of M3's NN was based on state x_1 only.

6

Table 6.4: Time response to sine-inputs for M1. Left value is BFT (%), Right value is VAF (%).

Input Amplitude	Input Frequency (Hz)			
	0.25	0.5	0.75	1
0.25	93 99	94 100	96 100	97 100
0.5	97 100	91 99	94 100	94 100
0.75	93 100	90 99	91 99	92 100
1	94 100	91 99	90 99	90 99
1.5	78 97	81 97	79 97	73 95
1.75	0 0	54 88	69 95	61 92

Table 6.5: Time response to sine-inputs for M2. Left value is BFT (%), Right value is VAF (%).

Input Amplitude	Input Frequency (Hz)			
	0.25	0.5	0.75	1
0.25	93 100	87 98	90 99	91 99
0.5	94 100	87 98	88 99	89 99
0.75	96 100	90 99	90 99	90 99
1	94 100	94 100	93 100	91 99
1.5	83 97	86 98	80 98	77 97
1.75	80 96	77 96	68 95	63 94

Table 6.6: Time response to sine-inputs for M3. Left value is BFT (%), Right value is VAF (%).

Input Amplitude	Input Frequency (Hz)			
	0.25	0.5	0.75	1
0.25	91 99	87 98	90 99	90 99
0.5	93 99	86 97	88 98	88 99
0.75	95 100	88 98	87 98	89 99
1	97 100	91 99	88 98	81 97
1.5	73 95	85 98	81 97	62 94
1.75	55 90	74 96	70 94	53 91

Finally, we also compare the model responses to a step input of amplitude $A = 0.5$, with the outcomes given in Table 6.7, and Fig. 6.4 through 6.6, where again the respective high model quality is being confirmed.

Table 6.7: Time response to step input of Amplitude $A = 0.5$.

Quasi-LPV Model	Costs	
	BFT (%)	VAF (%)
M1	70	97
M2	72	98
M3	55	96

We do see that all models exhibit some steady-state error on state x_1. This may potentially be attributed to the training of the NN models, i.e. in this case with few data. In summary, model M2 provides good model fidelity in OL, coupled with slightly better computational efficiency than model M1 (since having fewer scheduling parameters, and hence fewer NN models to evaluate), and may thus be used for OL prediction, whereas model M3 has also shown to be a suitable candidate for subsequent controller design, in a dynamic

output feedback framework (based upon measurement x_1).
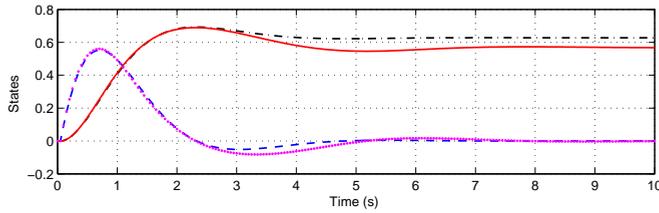
Figure 6.4: M1 outputs for step input of Amplitude $A = 0.5$ (legend: '–.' NL x_1; '– –' NL x_2; '–' qLPV x_1; '.' qLPV x_2).
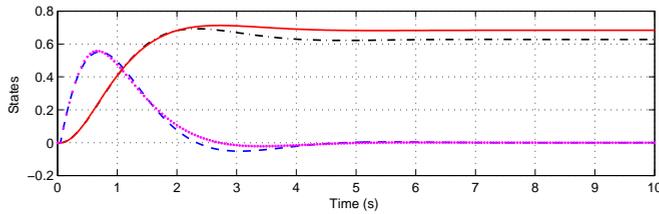
Figure 6.5: M2 outputs for step input of Amplitude $A = 0.5$ (legend: '–.' NL x_1; '– –' NL x_2; '–' qLPV x_1; '.' qLPV x_2).
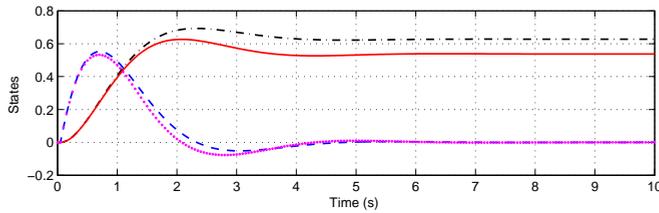
Figure 6.6: M3 outputs for step input of Amplitude $A = 0.5$ (legend: '–.' NL x_1; '– –' NL x_2; '–' qLPV x_1; '.' qLPV x_2).

6

6.9.3. CLOSED-LOOP ANALYSIS

The objective of this section is to evaluate the quasi-LPV model M3 in a CL setting. To this end, we define the generalized plant $G_P(s)$ which maps the exogenous inputs $\mathbf{w} = [\mathbf{r}^\top \; \mathbf{n}^\top]^\top$ and control inputs $\mathbf{u}$, to controlled outputs $\mathbf{z} = [\mathbf{z_u}^\top \; \mathbf{z_p}^\top]^\top$ and measured outputs $\mathbf{v} = [\mathbf{r}^\top \; \mathbf{e}^\top]^\top$, see Fig. 6.7. The signals consist further of $\mathbf{r}$ the reference signals, $\mathbf{n}$ the sensors noise, $\mathbf{e}$ the tracking errors, $\mathbf{z_u}$ the actuators performance signal (to limit actuator deflection magnitudes and rates), and $\mathbf{z_p}$ the desired performance in terms of closed-loop signal responses. The plant used for control synthesis is denoted P (plant P_0 and uncertainty Θ will be addressed in the sequel), and for the weights, we use the standard robust control weights, which include the performance weight $W_p(s)$, the actuator weight $W_u(s)$, and the sensor noise weight $W_n(s)$, all given in Appendix C. The generalized plant $G_P(s)$ is further
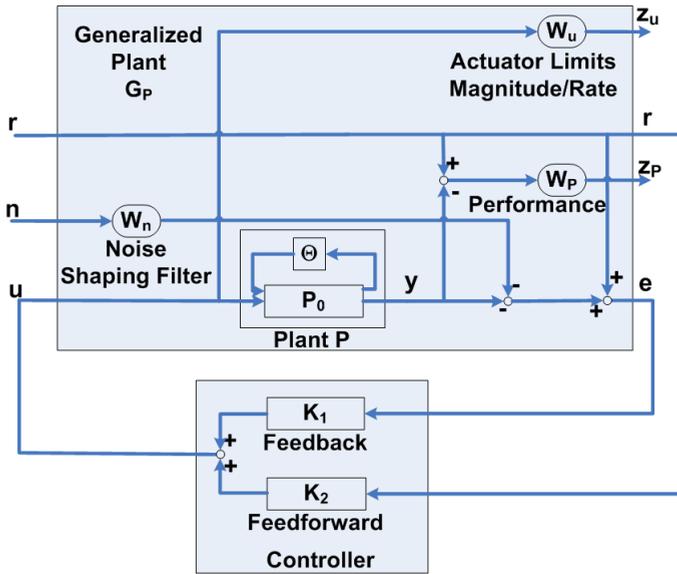
Figure 6.7: Control synthesis: two degrees-of-freedom control structure.

given by

$$
\begin{pmatrix} \mathbf{z_u} \\ \mathbf{z_p} \\ \mathbf{r} \\ \mathbf{e} \end{pmatrix} = \begin{bmatrix} 0 & 0 & W_u \\ W_p & 0 & -W_p\,P \\ I & 0 & 0 \\ I & -W_n & -P \end{bmatrix} \begin{pmatrix} \mathbf{r} \\ \mathbf{n} \\ \mathbf{u} \end{pmatrix} \tag{6.32}
$$

The goal of the control synthesis consists in finding a dynamic controller $K(s)$ that establishes closed-loop stability, while guaranteeing a $\mathcal{L}_2$-gain bound γ from the exogenous inputs $\mathbf{w}$ to the controlled outputs $\mathbf{z}$, such that

$$
\int_0^T \mathbf{z}^\top(t)\mathbf{z}(t)\,\mathrm{d}\tau \;\leq\; \gamma^2 \int_0^T \mathbf{w}^\top(t)\mathbf{w}(t)\,\mathrm{d}\tau \quad \forall T \geq 0 \tag{6.33}
$$

In the sequel, we will synthesize four controllers—one H_∞ LTI, one robust μ LTI, and two LPV ones—and compare their reference tracking performance to step reference inputs. The controller synthesis is based upon a two-degrees-of-freedom control structure, see Fig. 6.7. The feedback part is used to reduce the effect of uncertainty, whereas the feedforward part is added to improve tracking performance [81], and for optimality, both will be designed in one step. These four controllers are defined as follows

- **Controller 1:** H_∞ **LTI controller**. The purpose is here to synthesize a controller which is not based on model M3, but rather based upon a nominal LTI model. This latter model is obtained from a single linearization, at a rest position, of the NL model defined in Eq. (6.29). This nominal LTI model $P_{nom} := \left[\begin{array}{c|c} A_{nom} & B_{nom} \\ \hline [1\ 0] & 0 \end{array} \right]$, used

for control design, is computed via a first-order Taylor-series expansion of the NL model, at its equilibrium point $[x_1 \ x_2]^\top = [0 \ 0]^\top$, see Appendix C. Further, for this H_∞ LTI controller, the control synthesis does not include any robustness with respect to some uncertainty Θ, hence in Fig. 6.7 we have

$$P = P_0 = P_{nom} \tag{6.34}$$

Now, with Eq. (6.34) in mind, we can rewrite Eq. (6.32) as follows

$$\begin{pmatrix} \mathbf{z} \\ \mathbf{v} \end{pmatrix} = G_P \begin{pmatrix} \mathbf{w} \\ \mathbf{u} \end{pmatrix} \tag{6.35}$$

with $G_P(s)$ the generalized plant. Obtaining here a LTI controller $K(s)$ that minimizes the $\mathcal{L}_2$-gain bound γ from the exogenous inputs $\mathbf{w}$ to the controlled outputs $\mathbf{z}$, is equivalent to the minimization of the $\mathcal{H}_\infty$ norm of a standard, weighted, mixed-sensitivity S/KS criterion. Here, the controller $K(s)$ is computed such that [82]

$$K = \arg \min_{K} \|F_l(G_P, K)\|_\infty \tag{6.36}$$

We will consider this controller as the benchmark controller. The next three controllers will be synthesized using the LPV model M3, and will also be compared to this benchmark controller.

- **Controller 2: Robust μ LTI controller**. First, the identified affine LPV model M3, as defined in Eq. (6.3), is given by

$$P(\boldsymbol{\theta}(t)) := \left\{ \ \dot{\mathbf{x}}(t) = A_0\mathbf{x}(t) + B_0\mathbf{u}(t) + \sum_{r=1}^{R} \theta_r(t)\Big(A_r\mathbf{x}(t) + B_r\mathbf{u}(t)\Big) \right. \tag{6.37}$$

with $R = S + W$, S and W the number of basis functions retained in Section 6.4 and 6.5, respectively, and

$$\begin{aligned}
[\theta_1(t), ..., \theta_R(t)]^\top &:= \left[\hat{\eta}_1(t), ..., \hat{\eta}_S(t), \hat{\zeta}_1(t), ..., \hat{\zeta}_W(t)\right]^\top \\
[A_1, ..., A_R] &:= [L_1, ..., L_S, T_1, ..., T_W] \\
[B_1, ..., B_R] &:= [R_1, ..., R_S, Z_1, ..., Z_W]
\end{aligned} \tag{6.38}$$

Now, it is also useful to first rescale plant $P(\boldsymbol{\theta}(t))$ in Eq. (6.37) as follows

$$P(\boldsymbol{\alpha}(t)) := \left\{ \begin{aligned} &\dot{\mathbf{x}}(t) = \tilde{A}_0\mathbf{x}(t) + \tilde{B}_0\mathbf{u}(t) \\ &+ \sum_{r=1}^{R} \alpha_r(t)\Big(\tilde{A}_r\mathbf{x}(t) + \tilde{B}_r\mathbf{u}(t)\Big) \end{aligned} \right. \tag{6.39}$$

such that $\boldsymbol{\alpha}(t) := [\alpha_1(t), ..., \alpha_R(t)]^\top$, with $|\alpha_r(t)| \leq 1$. Here, the generalized plant $G_P(s)$ has a linear fractional dependence on the scheduling parameter $\boldsymbol{\alpha}(t)$. This plant $G_P(s)$ can be represented by the upper LFT interconnection

$$\begin{pmatrix} \mathbf{z} \\ \mathbf{v} \end{pmatrix} = G_P \begin{pmatrix} \mathbf{w} \\ \mathbf{u} \end{pmatrix} = F_u(M, \Theta) \begin{pmatrix} \mathbf{w} \\ \mathbf{u} \end{pmatrix} \tag{6.40}$$

where $M(s)$ is a known LTI plant, see Fig. 6.8. Further, $\Theta := blockdiag(\alpha_1 I_{k_1}, ..., \alpha_R I_{k_R})$ represents some block diagonal operator specifying how the scheduling parameters enter the plant dynamics, and $\{I_{k_r}\}_{r=1}^R$ denotes identity matrices whose sizes correspond, in a sense, to the "complexity" of the scheduling parameter variations. Next, the feedback structure associated with the LFT interconnection Eq. (6.40) is given by

$$
\begin{pmatrix} \mathbf{z}_\theta \\ \mathbf{z} \\ \mathbf{v} \end{pmatrix} = M \begin{pmatrix} \mathbf{w}_\theta \\ \mathbf{w} \\ \mathbf{u} \end{pmatrix}
\tag{6.41}
$$
$$
\mathbf{w}_\theta = \Theta \mathbf{z}_\theta
$$

with $\mathbf{z}_\theta$, and $\mathbf{w}_\theta$, the inputs and outputs of operator Θ, shown in Fig. 6.8.
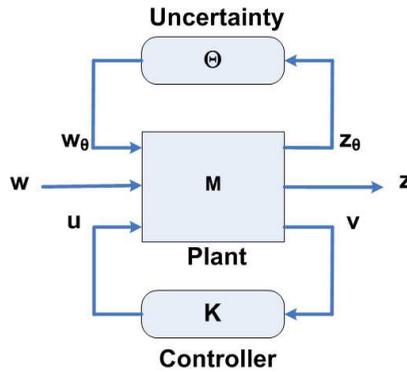
Figure 6.8: Standard $M - \Theta - K$ robust control framework.

We further proceed by treating the scheduling parameter variations, i.e. given in Θ, as fixed uncertainties (not measured on-line). This represents an approximation of the LPV model given in Eq. (6.39), which is here considered as a set of LTI models rather than a time-varying model. This scheduling parameter variations is addressed here within the robust control framework, by considering Θ as a time-invariant uncertainty, such that

$$
\bar{\sigma}(\Theta) \leq 1
\tag{6.42}
$$

The CL operator from exogenous inputs $\mathbf{w}$ to controlled outputs $\mathbf{z}$ is given by

$$
T(M, K, \Theta) = F_l(F_u(M, \Theta), K)
\tag{6.43}
$$

with $K(s)$ the to-be-synthesized controller. Again, the goal of the controller is to minimize the $\mathcal{L}_2$-gain bound γ from the exogenous inputs $\mathbf{w}$ to the controlled outputs $\mathbf{z}$, despite the uncertainty Θ. Based upon Eq. (6.42) and small gain considerations [83, 84], this goal is approximated by the minimization of the $\mathcal{H}_\infty$ norm of $F_l(M, K)$. Now, if Θ presents some structure, better performance may be obtained by synthesizing $K(s)$ through D-K iteration [67, 85]

$$K = \arg \min_K \inf_{D, D^{-1} \in \mathcal{H}_\infty} \|D F_l(M, K) D^{-1}\|_\infty \qquad (6.44)$$

with $D(s)$ a stable and minimum-phase scaling matrix, chosen such that $D(s)\Theta = \Theta D(s)$. Using [86] we obtain, for our example (see Appendix C for the problem data), after five iterations, a 12th order controller based upon an 8th order $D(s)$-scaling. The controller is further reduced to 5th order, after balancing and Hankel-norm model reduction [87], without any significant effect on CL robustness/performance.

In summary, we have obtained a single robust LTI controller, for a family of LTI plants. Recall however that a major approximation was made, namely the LPV model in Eq. (6.39) is considered as a set of LTI models, by assuming Θ to be time-invariant. Clearly, such an approach is not sufficient to prove stability and performance of the original, time-varying system, i.e. the LPV model in Eq. (6.39) [88]. In other words, the $\mathcal{L}_2$-gain from the exogenous inputs $\mathbf{w}$ to the controlled outputs $\mathbf{z}$ may be much higher than the $\mathcal{H}_\infty$ norm of $D F_l(M, K) D^{-1}$. This robust control approach should only be viewed as a necessary condition to prove stability and performance of the original LPV system. In other words, if the controller $K(s)$, obtained from Eq. (6.44), does not meet the desired stability and performance objectives, then it is pointless to consider other controllers, such as LPV ones, that do take the time-varying aspect of the system into account. This said, this robust control approach, as presented here, is known to work well in practice for scheduling parameters having sufficiently slow time-variations.

Let us now examine a more sophisticated control approach, which takes the time-varying nature of the scheduling parameters into account. To this end, we consider now controllers which are also in LPV form, and hence also time-varying. The goal of an $\mathcal{H}_\infty$-based, output-feedback, control problem for LPV systems consists in finding, for all parameter trajectories[15] $\Theta(t) := blockdiag(\alpha_1(t)I_{k_1}, ..., \alpha_R(t)I_{k_R})$, a dynamic controller $K(s)$ that establishes closed-loop stability, while truly minimizing the $\mathcal{L}_2$-gain bound γ from the exogenous inputs $\mathbf{w}$ to the controlled outputs $\mathbf{z}$.

Over the years the subject of LPV control has received much attention, resulting in a plethora of control methods. Although a full review of LPV control methods is beyond the scope of this Chapter, we briefly mention here the following classifications

- So-called polytopic—also known as quadratic—techniques [7, 89–92], versus so-called scaled small-gain—also known as Linear Fractional Representations (LFR) or norm-bounded—approaches [6, 8, 9, 12, 13, 92–94].

- So-called Parameter-Independent Lyapunov Function (PILF) techniques (such as the methods listed in the previous alinea), versus so-called Parameter-Dependent Lyapunov Function (PDLF)—also known as griding—approaches [10, 11, 69, 70, 95–99].

---

[15]Notice that now $\Theta(t)$ is a time-varying operator.

We summarize next some general guidelines

- Polytopic PILF approaches tend to be less conservative than the scaled small-gain PILF ones [92, 100, 101]. However, this comes at the expense of an exponential growth in the number of LMIs.

- PILF methods enjoy twin relevant properties: 1) simplicity, having controller complexity typically equaling that of the plant; and 2) numerical tractability. However PILF methods are based upon the quadratic stability/robustness condition, known to be only a sufficient condition [102].

- PDLF methods can improve performance, i.e. decrease conservatism, in case the scheduling parameter time-derivative is known to be bounded [102]. However, PDLF approaches often lead to additional difficulties, namely an infinite number of LMIs emanating from the parameter-dependent LMI structure. Hence, PDLF methods rely upon so-called griding techniques, resulting in poor computational tractability.

In light of the previous discussion, and in order to validate our LPV modeling framework in CL, we implement here two ($\mathcal{H}_\infty$-based) LPV control methods: 1) a so-called polytopic PILF one; and 2) a so-called scaled small-gain PDLF one. Keeping in mind synthesis simplicity and low online computational effort, we choose methods [7] and [69] as the respective control approaches. These two LPV controllers are defined as follows

- **Controller 3: Polytopic PILF LPV controller**. In the LPV model $P(\alpha(t))$, given by Eq. (6.39), the scheduling parameter $\alpha(t)$ is defined on a compact set $\mathcal{P}_\alpha$, represented by a hypercube of dimension R, with its vertices corresponding to the extremal values of $\{\alpha_r(t)\}_{r=1}^R$. Let $\{w_j| \ j \in \{1, ..., J\}, J = 2^R\}$ be the vertices of this polytope, then we can define the following convex hull

$$Co\{w_1, ..., w_J\} := \left\{ \sum_{j=1}^J \lambda_j w_j, \ \sum_{j=1}^J \lambda_j = 1, \ \lambda_j \geq 0 \right\} \qquad (6.45)$$

with $Co(\cdot)$ the abbreviation denoting the convex hull. For LPV model M3, we have $R = S + W = 2$, implying $J = 2^2 = 4$ vertices, given by $w_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $w_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, $w_3 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$, $w_4 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$, since we had normalized the scheduling parameters as $|\alpha_r(t)| \leq 1$. In Eq. (6.39) the dependency on $\alpha(t)$ is affine, hence the vertices of the state-space matrix polytope, used for controller design, are given by $P(w_j)$, $j \in \{1, ..., J\}$ (see [7] for further details). The controller synthesis[16] follows the lines of classical $\mathcal{H}_\infty$ synthesis, with the difference that it is based upon the $\mathcal{H}_\infty$ quadratic stability and performance concept (since both plant and controller are time-varying). The global LPV controller $K(\alpha(t))$ is obtained through interpolation of local controllers, the latter being synthesized at each vertex $P(w_j)$ [7]. Since the method requires the control-matrix to be independent of the time-varying scheduling parameter, we pre-filtered the LPV model with the low-pass filter defined at the beginning of

---

[16]The polytopic PILF LPV controller synthesis method [7] is available in the MATLAB Robust Control Toolbox.

Section 6.9.1. A gain $\gamma = 0.92$, in Eq. (6.33), was achieved with the weights defined in Appendix C. Although the synthesized controller $K(\alpha(t))$ is time-varying—and hence represents an improvement compared to the previous LTI μ controller—the quadratic stability and performance concept assumes arbitrarily fast varying scheduling parameters $\alpha(t)$. Obviously this may result in some conservatism, in case the scheduling parameters have a bounded rate of variation.

- **Controller 4: Small-gain PDLF LPV controller**. This last controller is also referred in the sequel as the LPV-LFT controller. Again, both plant and controller are dependent on the time-varying scheduling parameter $\Theta(t) := blockdiag(\alpha_1(t)I_{k_1}, ..., \alpha_R(t)I_{k_R})$. The CL operator from exogenous inputs **w** to controlled outputs **z** is adjusted from Eq. (6.43) to become

$$T_{lpv}(M, K, \Theta(t)) = F_l(F_u(M, \Theta(t)), F_l(K, \Theta(t))) \qquad (6.46)$$

The to-be-designed LPV controller $K(\Theta(t))$ is obtained by minimizing the $\mathcal{L}_2$-norm of operator T_{lpv} [69]. Moreover, the controller synthesis method also takes parameter time-derivative into account, implying a dependence on both $\Theta(t)$ and its derivative $\dot{\Theta}(t)$. This results in an infinite-dimensional LMI problem [69] which, in our case, was tackled by using a small grid, containing only the extrema of $\Theta(t)$ and $\dot{\Theta}(t)$. Since the method [69] is an iterative method[17], good starting values for the scalings were obtained by performing a robust μ synthesis, with constant scalings, on the $(\tilde{A}_0, \tilde{B}_0)$ plant (this plant is defined in Eq. (6.39)). A gain $\gamma = 0.51$, in Eq. (6.33), was achieved with the weights defined in Appendix C, after ten iterations.

Remark 16 *The purpose of this CL experiment is not so much on specific aspects related to controller weight selection, but rather on highlighting any general similarities or differences, obtained when synthesizing various controllers, while using two modeling options: either LTI or our LPV based method. Similarly, and although, for generality, some robustness with respect to signal noise was included during the controller synthesis process (with weight $W_n(s)$), the simulation results, presented hereunder, consider only reference tracking in a noise-free and disturbance-free environment.*

DISCUSSION OF RESULTS

The validation of all controllers, on the NL plant, is done using step inputs on the x_1 reference signal, starting from a zero initial condition, i.e. pendulum at rest, see Fig. 6.9–Fig. 6.11. With respect to our LPV modeling method, we provide the following main conclusions and recommendations:

- The H_∞ controller exhibits a steady-state error, which remains persistent despite several modifications of the performance weight $W_P(s)$. Compared to the H_∞ controller, which is designed on a linearization of the NL plant, all other controllers designed using our LPV modeling methodology, i.e. on model M3, do not exhibit any steady-state error, and hence achieve much better reference tracking. This is achieved even though model M3 has been built with the least number of basis functions.

---

[17]The small-gain PDLF LPV controller synthesis method [69] is not available in the MATLAB Robust Control Toolbox.

- Best practice would be to first design a robust μ controller (especially if the NN model has been trained with few data), and view it as a benchmark design. Then, it would be interesting to implement at least one PILF LPV control method, and one PDLF LPV control method, in order to be able to compare results.
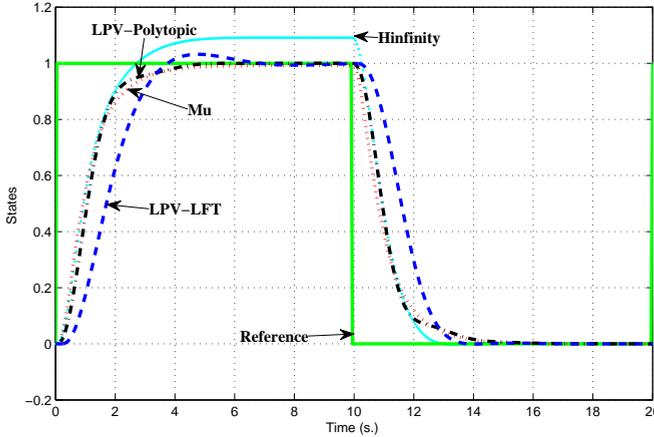
Figure 6.9: Closed-Loop step response of NL model with controllers: $reference \rightarrow x_1$. Cyan line: H_∞ controller. Red dotted line: μ controller. Black dash-dotted line: LPV-Polytopic controller. Blue dashed line: LPV-LFT controller.

With regard to control, we provide the following main conclusions:

- The robust μ controller and the polytopic PILF LPV controller exhibit very similar tracking performance, although the control input of the latter one is much smoother, see Fig. 6.11.

- Comparison of robust μ control with several LPV control methods has primarily been addressed in [103–107]. Except for [107], all authors have reported that LPV methods were less conservative than a standard μ approach. Indeed, the distinct advantage of LPV control methods is based upon the on-line measurement of the scheduling parameters (and potentially its derivatives). However for LPV-LFT methods, this advantage needs to be put into perspective, since all LPV-LFT control methods (except for the most prominent contribution [94]) have been based upon static scaling, whereas μ uses dynamic scaling.

- If additional robustness is required, to account for unmodelled dynamics and NL effects, then one may add a complex full-block input multiplicative uncertainty $\Theta_c(s)$ at the input of the plant. The uncertainty structure $\Theta(s)$ in Fig. 6.8 is then replaced by a mixed, real and complex, uncertainty structure $\begin{bmatrix} \Theta_c & 0 \\ 0 & \Theta \end{bmatrix}$, for which several LPV control methods exist, e.g. [8, 69].

- If knowledge of the scheduling parameters is somewhat inexact, then [108] may be of interest.

6
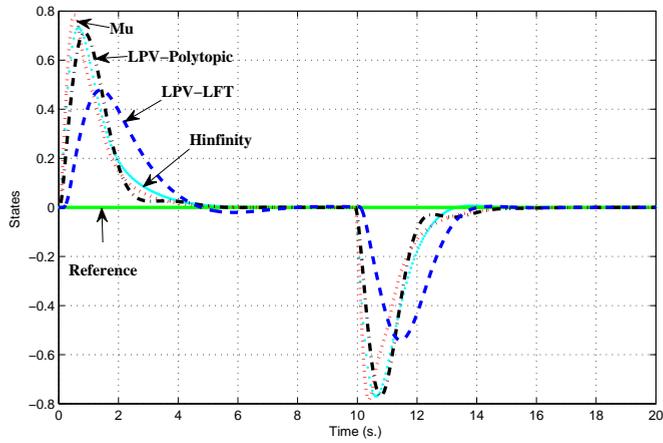
Figure 6.10: Closed-Loop step response of NL model with controllers: $reference \rightarrow x_2$. Cyan line: H_∞ controller. Red dotted line: μ controller. Black dash-dotted line: LPV-Polytopic controller. Blue dashed line: LPV-LFT controller.
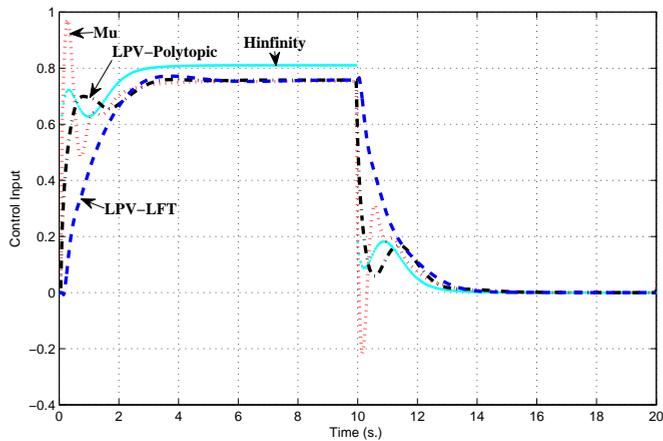
Figure 6.11: Closed-Loop step response of NL model with controllers: control input u. Cyan line: H_∞ controller. Red dotted line: μ controller. Black dash-dotted line: LPV-Polytopic controller. Blue dashed line: LPV-LFT controller.

6.10. CONCLUSION

We have presented a comprehensive affine quasi-LPV modeling framework, allowing to derive models which are suitable for open-loop and close-loop applications such as robust and LPV controller design. In addition, the versatility of the proposed modeling framework may potentially allow to consider other types of control analysis and synthesis avenues, provided some form of model clustering is used, such as those in the realm of Piece-Wise-Affine and Piece-Wise-Linear methods. Since our LPV modeling approach does not incorporate any information on parameter time-derivatives, it is expected that significant enhancements could potentially be obtained in this area.

Our modeling method was applied to the helicopter high-order nonlinear model of Chapter 2, and resulted in a LPV model having a large number of (i.e. more than thirty) scheduling parameters. Unfortunately, it became impossible to synthesize LPV controllers with such a high-order LPV model. In fact, the numerical conditioning and solvability of LMI problems play a crucial role in LPV practical design methods. A way to mitigate such problems would consist in applying some LPV model reduction techniques, in order to obtain a LPV model having fewer scheduling parameters, hence better suited for LPV controller synthesis.

6

6.11. Appendix A: Kalman-Yakubovich-Popov (KYP) Lemma with spectral mask constraints

We recall here how to compute the $\| \cdot \|_{\Delta_\omega}$ norm, i.e. the $\mathcal{H}_\infty$ norm with spectral mask constraints, through the use of the Kalman-Yakubovich-Popov (KYP) Lemma [109] with spectral constraints [110, 111].

6.11.1. Preliminaries

Lemma 2 *Let real scalars $\omega_1 \leq \omega_2$, $\omega_c = (\omega_1 + \omega_2)/2$, and a Transfer Function (TF)*
$G(s) := \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ *be given, then the following statements are equivalent.*

1. $\forall \gamma > 0, \ \lambda(A) \subset \mathbb{C}^- \cup \mathbb{C}^+, \ \|G\|^2_{\Delta_\omega} < \gamma^2$ *(6.47)*

2. *There exists matrices P and Q, of appropriate size, such that*
   $P = P^*$, $Q > 0$, *and* $L(P, Q) + \Theta < 0$, *with*
   $$L(P, Q) = \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}^* \begin{bmatrix} -Q & P + j\omega_c Q \\ P - j\omega_c Q & -\omega_1 \omega_2 Q \end{bmatrix} \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}$$
   $$\Theta = \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^* \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \qquad (6.48)$$

3. *There exists matrices F and K, of appropriate size, such that*
   $$\forall l \in \{1, 2\} \quad M_l(F, K) + \Theta < 0$$
   $$M_l(F, K) = He\left(\begin{bmatrix} F \\ K \end{bmatrix} \begin{bmatrix} I & -j\omega_l I \end{bmatrix} \begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \right)$$
   With Θ given in Eq. (6.48)   *(6.49)*

Proof 2 *Invoke the KYP Lemmas with spectral mask constraints, from [110] and [111], to prove (ii) and (iii) respectively.*

Hence, the norm $\| \cdot \|^2_{\Delta_\omega}$ is obtained by minimizing the bound γ^2 defined in Eq. (6.47), which is computationally done by minimizing γ^2 subject to the LMI in alinea 2), or 3). Both approaches in 2) and 3) of Lemma 2 will be used in this Chapter. Now let n be the number of decision variables, and m the number of rows of LMIs, then comparing 2) and 3) shows that, while both have similar m, they differ in terms of n, i.e. $n_x^2 + n_x$ versus $n_x^2 + n_x n_u$, respectively. Since the asymptotic computational complexity, or flop cost, of SDP solvers is in $O(n^2 m^{2.5} + m^{3.5})$ for SeDuMi [19], and in $O(n^3 m)$ for MATLAB LMI-lab [112], the former approach is more efficient for large problems, and hence is the method we will use most often, however, the latter has the advantage that, for fixed F and K, it is also affine in the problem's A and B matrices, and hence can be used in a bi-convex framework.

6.12. Appendix B: Identifying the set of parameters $\left\{\eta_1(t_i), ..., \eta_S(t_i)\right\}_{i=1}^{N}$ for a specific case

Here we consider a situation for which the optimal value of the scheduling parameters can be computed, avoiding thus an iterative approach the like of Section 6.6. We examine the

specific case where matrices L_s are not identically zero, however with matrices R_s identically zero. Now Eq. (6.22) becomes equivalent to

$$
\bar{G}_i(s) - G_i(s) := \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] =
\left[\begin{array}{cc|c} \bar{A}_i & 0 & \bar{B}_i \\ 0 & A_0 + \sum_{s=1}^{S} \eta_s(t_i)L_s & B_0 \\ \hline I & -I & 0 \end{array}\right]
\tag{6.50}
$$

Here Eq. (6.50) corresponds to a situation where the control-input matrix, of all LTI models, is independent of the time-varying scheduling parameter (all matrices $\bar{B}_i$ are identical). This may be a specificity of the NL model, or alternatively, it may be achieved by (low-pass) filtering the control input of all LTI models [7]. In addition, we revert here to a standard weighted $\mathcal{H}_\infty$ norm minimization instead of the KYP-based formalism used in Section 6.6, hence replacing Eq. (6.19) by the following: find, for each time t_i, the parameters $\hat{\boldsymbol{\eta}}(t_i) := [\hat{\eta}_1(t_i), ..., \hat{\eta}_S(t_i)]^\top$ that minimize

$$
J_1(t_i) := \|W_f(s)(\bar{G}_i(s) - G_i(s))\|_\infty
\tag{6.51}
$$

with $W_f(s)$ a strictly-proper, bandpass filter, centered at Δ_ω. Now, if we consider the following assumption

- **A.1** In Section 6.4, all basis (i.e. columns) in $U_{1..S}$ are retained when computing $\{L_s, R_s\}_{s=1}^S$.

then Eq. (6.51) becomes convex, and the optimal value $\hat{\boldsymbol{\eta}}(t_i)$ can be found through a three-step procedure. But before solving Eq. (6.51), we give first the following result, which will prove useful in the sequel.

Lemma 3 *Let* $W_f(s) := \left[\begin{array}{c|c} A_f & B_f \\ \hline C_f & 0 \end{array}\right]$, $\bar{G}_i(s) := \left[\begin{array}{c|c} \bar{A}_i & \bar{B}_i \\ \hline I & 0 \end{array}\right]$, $G_i(s) := \left[\begin{array}{c|c} A_0 + \sum_{s=1}^{S} \eta_s(t_i)L_s & B_0 \\ \hline I & 0 \end{array}\right]$,

be given, with matrices of appropriate size. Let

$$
W_f(s)(\bar{G}_i(s) - G_i(s)) := \left[\begin{array}{cc|c} A_{11} & A_{12} & B_{11} \\ 0 & A_{22} & B_0 \\ \hline C_{11} & 0 & 0 \end{array}\right]
\tag{6.52}
$$

with $A_{11} = \left[\begin{array}{cc} A_f & B_f \\ 0 & \bar{A}_i \end{array}\right]$, $A_{12} = \left[\begin{array}{c} -B_f \\ 0 \end{array}\right]$, $B_{11} = \left[\begin{array}{c} 0 \\ \bar{B}_i \end{array}\right]$, $C_{11} = \left[\begin{array}{cc} C_f & 0 \end{array}\right]$, *and* $A_{22} = A_0 + \sum_{s=1}^{S} \eta_s(t_i)L_s$, *then the following two statements are equivalent*

1. $\forall \gamma > 0$, $W_f(s) \in \mathcal{RH}_\infty$, $(\bar{G}_i(s) - G_i(s)) \in \mathcal{RL}_\infty$, $\|W_f(\bar{G}_i(s) - G_i(s))\|_\infty^2 < \gamma^2$ (6.53)

2. $\exists(P, Q)$, $P = P^\top$, $Q = Q^\top = P^{-1}$, with matrix partitions in P and Q matching those in Eq. (6.52), given by $P = \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^\top & P_{22} \end{bmatrix}$, $Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^\top & Q_{22} \end{bmatrix}$, with

$\Gamma(X_\eta, P_{11}, P_{12}, Q_{11}, Q_{12}) := \dots$

$$\begin{bmatrix} Sym(A_{11}Q_{11} + A_{12}Q_{12}^\top) & \star & \star & \star \\ A_{11}^\top + X_\eta & Sym(P_{11}A_{11}) & \star & \star \\ B_{11}^\top & B_{11}^\top P_{11} + B_0^\top P_{12}^\top & -\gamma^2 I & \star \\ C_{11}Q_{11} & C_{11} & 0 & -\gamma^2 I \end{bmatrix} < 0 \qquad (6.54)$$

and $X_\eta = P_{11}A_{11}Q_{11} + P_{11}A_{12}Q_{12}^\top + P_{12}A_{22}Q_{12}^\top$

Proof 3 *The proof is a straightforward application of the Bounded Real Lemma (BRL) [113] in LMI form [82], with further: 1) a congruence transformation [114] with diag(J, I, I),* $J = \begin{bmatrix} Q_{11} & I \\ Q_{12}^\top & O \end{bmatrix}$; *and 2) a change of variable given by* X_η. *Note that for stable systems, i.e.* $(\bar{G}_i(s) - G_i(s)) \in \mathcal{RH}_\infty$, *one has to add the condition* $J^\top P J = \begin{bmatrix} Q_{11} & I \\ I & P_{11} \end{bmatrix} > 0$

Now, solving Eq. (6.51) reduces to a three-step procedure

1. First, solve

   $$\begin{array}{ll} \forall i \in \{1, \dots, N\} & \text{minimize } \gamma \\ \text{with respect to} & X_\eta, P_{11}, P_{12}, Q_{11}, Q_{12} \\ \text{subject to} & \gamma > 0, \text{ and the LMIs of Lemma 3} \end{array} \qquad (6.55)$$

2. Compute $A_{22} = P_{12}^\dagger (X_\eta - P_{11}A_{11}Q_{11} - P_{11}A_{12}Q_{12}^\top)Q_{12}^{\top\dagger}$, with $(\cdot)^\dagger$ the Moore-Penrose inverse. Note that P_{12} and $Q_{12}^\top$ are skinny and fat matrices, hence, by virtue of the respective left and right inverse, A_{22} is well-defined

3. Finally, minimize the $\mathcal{L}_2$-induced gain of static operator[18] $X_{A_{22}} = A_{22} - (A_0 + \sum_{s=1}^{S} \eta_s(t_i)L_s)$, such that

   $$\forall i \in \{1, \dots, N\} \quad \hat{\eta}(t_i) = \arg\min_{\eta_s(t_i)} \|X_{A_{22}}\|_2 \qquad (6.56)$$

   which is solved as in Eq. (6.21)

6.13. APPENDIX C: PROBLEM DATA

The nominal model, corresponding to a linearization of the pendulum NL model at $[x_1\ x_2]^\top = [0\ 0]^\top$, used for H_∞ controller design, is given by

$$A_{nom} = \begin{bmatrix} 0 & 1 \\ -3.2667 & -2 \end{bmatrix} \quad B_{nom} = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

The data for model Eq. (6.37) is given by

$$A_0 = \begin{bmatrix} 0 & 1 \\ -2.7915 & -2 \end{bmatrix} \quad B_0 = \begin{bmatrix} 0 \\ 3.0631 \end{bmatrix}$$

---

[18]Note that, thanks to assumption **A.1**, the quantity A_{22} can exactly be recovered from $(A_0 + \sum_{s=1}^{S} \eta_s(t_i)L_s)$.

$$A_1 = \begin{bmatrix} 0 & 0 \\ -0.0170 & 0 \end{bmatrix} \quad B_1 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 \\ 0.2205 & -0.3446 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 \\ -0.9125 \end{bmatrix}$$

The data for model Eq. (6.39) is given by

$$\tilde{A}_0 = \begin{bmatrix} 0 & 1 \\ -2.8896 & -1.8459 \end{bmatrix} \quad \tilde{B}_0 = \begin{bmatrix} 0 \\ 3.4962 \end{bmatrix}$$

$$\tilde{A}_1 = \begin{bmatrix} 0 & 0 \\ -0.0159 & 0 \end{bmatrix} \quad \tilde{B}_1 = \begin{bmatrix} 0 \\ -0.9342 \end{bmatrix}$$

$$\tilde{A}_2 = \begin{bmatrix} 0 & 0 \\ 0.1556 & -0.2433 \end{bmatrix} \quad \tilde{B}_2 = \begin{bmatrix} 0 \\ -0.6441 \end{bmatrix}$$

$$\bar{\theta}_1 = 0.9092, \ \underline{\theta}_1 = -0.9595 \quad \bar{\theta}_2 = 0.2588, \ \underline{\theta}_2 = -1.1530$$

The maximum rates for the LPV-LFT controller are

$$\bar{\alpha}_1 = 11.59, \ \dot{\underline{\alpha}}_1 = -12.10 \quad \bar{\alpha}_2 = 11.13, \ \dot{\underline{\alpha}}_2 = -11.72$$

The LTI performance weights in Fig. 6.7 are based upon the guidelines of [81]. We have used

$$W_u(s) = \frac{s}{s + 2\pi} \quad W_n(s) = 0.005$$

For the H_∞, μ, and LPV-LFT controllers, after several trials, we settled for

$$W_P(s) = \frac{s/2 + 0.25\pi}{s + \frac{0.25\pi}{10^2}}$$

For the LPV-Polytopic controller, we have used

$$W_P(s) = \frac{s/2 + 0.25\pi}{s + \frac{0.25\pi}{10^6}}$$

6

REFERENCES

[1] S. Taamallah, X. Bombois, and P. M. J. Van den Hof, *Affine lpv modeling: An h_∞ based approach,* in *IEEE Conf. on Decision and Control* (2013).

[2] D. J. Walker, *Multivariable control of the longitudinal and lateral dynamics of a fly-by-wire helicopter,* Control Engineering Practice **11**, 781 (2003).

[3] N. Hovakimyan and C. Cao, L_1 *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation* (SIAM, 2010).

[4] J. S. Shamma and M. Athans, *Analysis of gain scheduled control for nonlinear plants,* IEEE Trans. Autom. Control **35**, 898 (1990).

[5] W. Rugh and J. Shamma, *Research on gain scheuling,* Automatica **36**, 1401 (2000).

[6] A. Packard, *Gain scheduling via linear fractional transformations,* Systems & Control Letters **22**, 79 (1994).

[7] P. Apkarian, P. Gahinet, and G. Becker, *Self-scheduled h_∞ control of linear parameter-varying systems: A design example,* Automatica **31**, 1251 (1995).

[8] P. Apkarian and P. Gahinet, *A convex characterization of gain-scheduled h_∞ controllers,* IEEE Trans. Automatic Control **40**, 853 (1995).

[9] W. M. Lu and J. C. Doyle, h_∞ *control of nonlinear systems: A convex characterization,* IEEE Trans. Automatic Control **40**, 1668 (1995).

[10] F. Wu, X. H. Yang, A. K. Packard, and G. Becker, *Induced $\mathcal{L}_2$ norm control for lpv systems with bounded parameter variation rates,* Int. J. Of Robust And Nonlinear Control **6**, 983 (1996).

[11] J. Yu and A. Sideris, h_∞ *control with parametric lyapunov functions,* Systems & Control Letters **30**, 57 (1997).

[12] G. Scorletti and L. ElGhaoui, *Improved lmi conditions for gain scheduling and related control problems,* Int. J. of Robust and Nonlinear Control **8**, 845 (1998).

[13] C. W. Scherer, *Lpv control and full block multipliers,* Automatica **37**, 361 (2001).

[14] G. Stein, *Adaptive Flight Control - A Pragmatic View. In K. S. Narendra and R. V. Monopoli (Eds), Applications of Adaptive Control* (Academic Press, New York, 1980).

[15] K. J. Astrom and B. Wittenmark, *Adaptive Control* (Addison-Wesley, Reading, MA, 1989).

[16] S. Boyd, L. ElGhaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory* (Soc. for Industrial and Applied Mathematics (SIAM), Philadelphia, 1994).

[17] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2004).

[18] Y. Nesterov and A. Nemirovski, *Interior Point Polynomial Methods in Convex Programming* (Soc. for Industrial & Applied Math, 1994).

[19] J. F. Sturm, *Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones,* Optimization Methods and Software **11-12**, 625 (1999).

[20] J. S. Shamma, *An Overview of LPV Systems. In J. Mohammadpour and C. Scherer (Eds): Control of Linear Parameter Varying Systems with Applications* (Springer, 2012).

[21] G. J. Balas, I. Fialho, A. Packard, J. Renfrow, and C. Mullaney, *On the design of lpv controllers for the f-14 aircraft lateral-directional axis during powered approach,* in *Am. Control Conf.* (1997).

[22] G. J. Balas, J. B. Mueller, and J. M. Barker, *Application of gain-scheduled, multivariable control techniques to the f/a-18 system research aircraft,* in *AIAA Guidance Navigation and Control Conf.* (1999).

[23] I. Fialho, G. J. Balas, A. K. Packard, J. Renfrow, and C. Mullaney, *Gain-scheduled lateral control of the f-14 aircraft during powered approach landing,* AIAA J. of Guidance, Control, and Dynamics **23**, 450 (2000).

[24] A. Marcos and G. J. Balas, *Development of linear-parameter-varying models for aircraft,* AIAA J. of Guidance, Control, and Dynamics **27**, 218 (2004).

[25] L. Reberga, D. Henrion, J. Bernussou, and F. Vary, *Lpv modeling of a turbofan engine,* in *IFAC World Congress* (2005).

[26] L. Giarre, D. Bausoa, P. Falugi, and B. Bamieh, *Lpv model identification for gain scheduling control: An application to rotating stall and surge control problem,* Control Engineering Practice **14**, 351 (2006).

[27] K. Z. Ostergaard, *Robust, Gain-Scheduled Control of Wind Turbines*, Ph.D. thesis, Aalborg University (2008).

[28] M. Steinbuch, R. van de Molengraft, and A. J. van der Voort, *Experimental modelling and lpv control of a motion system,* in *Am. Control Conf.* (2003).

[29] M. G. Wassink, M. van de Wal, C. Scherer, and O. Bosgra, *Lpv control for a wafer stage: Beyond the theoretical solution,* Control Engineering Practice **13**, 231 (2004).

[30] M. Dettori, *LMI Techniques for Control*, Ph.D. thesis, Delft University of Technology (2001).

[31] S. M. Hashemi, H. S. Abbas, and H. Werner, *Lpv modelling and control of a 2-dof robotic manipulator using pca-based parameter set mapping,* in *IEEE Conf. on Decision and Control* (2009).

6

[32] R. Toth, *Identification and Modeling of Linear Parameter-Varying Systems. Lecture Notes in Control and Information Sciences, Vol. 403* (Springer, Heidelberg, 2010).

[33] R. Murray-Smith and T. Johansen (Ed), *Multiple Model Approaches to Modeling and Control* (CRC Press, 1997).

[34] W. Rugh, *Analytical framework for gain scheduling,* IEEE Control Systems Magazine **11**, 74 (1991).

[35] J. Shamma and M. Athans, *Gain scheduling: Potential hazards and possible remedies,* IEEE Control Systems Magazine **12**, 101 (1992).

[36] R. Bos, X. Bombois, and P. M. J. Van den Hof, *Accelerating simulations of computationally intensive first principle models using accurate quasi-linear parameter varying models,* J. of Process Control **19**, 1601 (2009).

[37] D. Petersson and J. Lofberg, *Optimization based lpv-approximation of multi-model systems,* in *Europ. Control Conf.* (2009).

[38] H. Pfifer and S. Hecker, *Generation of optimal linear parametric models for lft-based robust stability analysis and control design,* IEEE Trans. on Control Systems Technology **19**, 118 (2011).

[39] B. Paijmans, *Interpolating Gain-Scheduling Control for Mechatronic Systems with Parameter-Dependent Dynamics*, Ph.D. thesis, Katholieke Universiteit Leuven (2007).

[40] J. DeCaigny, J. F. Camino, and J. Swevers, *Interpolation-based modeling of mimo lpv systems,* IEEE Trans. on Control Systems Technology **19**, 46 (2011).

[41] J. S. Shamma and J. R. Cloutier, *Gain-scheduled missile autopilot design using linear parameter varying transformations,* AIAA J. of Guidance, Control, and Dynamics **16**, 256 (1993).

[42] D. J. Leith and W. E. Leithhead, *Gain-scheduled controller design: An analytic framework directly incorporating non-equilibrium plant dynamics,* Int. J. of Control **70**, 249 (1998).

[43] W. Tan, *Applications of Linear Parameter-Varying Control Theory*, Tech. Rep. (Master Thesis, University of California at Berkeley, 1997).

[44] J. Y. Shin, *Quasi-Linear Parameter Varying Representation of General Aircraft Dynamics Over Non-Trim Region*, Tech. Rep. CR-2007-213926 (NASA Langley Research Center, 2007).

[45] A. Kwiatkowski, M. T. Boll, and H. Werner, *Automated generation and assessment of affine lpv models,* in *IEEE Conf. on Decision and Control* (2006).

6

[46] R. Toth, P. S. C. Heuberger, and P. M. J. Van den Hof, *Prediction Error Identification of LPV Systems: Present and Beyond. In: J. Mohammadpour and C. W. Scherer (Eds.), Control of Linear Parameter Varying Systems with Applications* (Springer, Heidelberg, 2012).

[47] T. A. Johansen, R. Shorten, and R. Murray-Smith, *On the interpretation and identification of dynamic takagi-sugeno fuzzy models,* IEEE Trans. on Fuzzy Systems **8**, 297 (2000).

[48] R. Shorten, R. Murray-Smith, R. Bjorgan, and H. Gollee, *On the interpretation of local models in blended multiple model structures,* Int. J. of Control **72**, 620 (1999).

[49] L. H. Lee and K. Poolla, *Identification of linear parameter-varying systems using nonlinear programming,* J. of Dynamic Systems, Measurement and Control **121**, 71 (1999).

[50] V. Verdult, L. Ljung, and M. Verhaegen, *Identification of composite local linear state-space models using a projected gradient search,* Int. J. of Control **75**, 1125 (2002).

[51] L. Ljung, *System Identification: Theory for the User - 2nd Edition* (Prentice Hall, Upper Saddle River, N.J., 1999).

[52] M. Gevers, A. Bazanella, X. Bombois, and L. Miskovic, *Identification and the information matrix: How to get just sufficiently rich,* IEEE Trans. Autom. Control **54**, 2828 (2009).

[53] R. Toth, P. S. C. Heuberger, and P. M. J. Van den Hof, *Discretization of linear parameter-varying state-space representations,* IET Control Theory & Applications **4**, 2082 (2010).

[54] P. Korba, *A Gain Scheduled Approach to Model Based Fuzzy Control*, Ph.D. thesis, University of Duisburg (2000).

[55] V. Laurain, R. Toth, M. Gilson, , and H. Garnier, *Direct identification of continuous-time lpv input/output models,* IET Control Theory & Applications **5**, 878 (2011).

[56] G. Goodwin, S. Graebe, and M. Salgado, *Control System Design* (Prentice Hall, 2000).

[57] K. J. Hunt and T. A. Johansen, *Design and analysis of gain-scheduled local controller networks,* Int. J. of Control **66**, 619 (1997).

[58] T. A. Johansen, K. J. Hunt, P. J. Gawthrop, and H. Fritz, *Off-equilibrium linearization and design of gain scheduled control with application to vehicle speed control,* Control Engineering Practice **6**, 167 (1998).

[59] P. Korba and H. Werner, *Counter-example to ad-hoc off-equilibrium linearisation methods in gain-scheduling,* in *IEEE Conf. on Decision and Control* (2001).

6

[60] C. S. Mehendale and K. M. Grigoriadis, *A new approach to lpv gain-scheduling design and implementation,* in *IEEE Conf. on Decision and Control* (2004).

[61] L. Rodrigues, *Dynamic Output Feedback Controller Synthesis for Piecewise-Affine Systems*, Ph.D. thesis, Stanford University (2002).

[62] J. Daafouz et al., *Switched and Piecewise Affine Systems. In J. Lunze and F. Lamnabhi-Lagarrigue (Eds): Handbook of Hybrid Systems Control Theory, Tools, Applications* (Cambridge Univ. Press, 2009).

[63] J. S. Shamma and M. Athans, *Guaranteed properties of gain scheduled control for linear parameter-varying plants,* Automatica **27**, 559 (1991).

[64] Z. Petres, *Polytopic Decomposition of Linear Parameter-Varying Models by Tensor-Product Model Transformation*, Ph.D. thesis, Budapest University of Technology and Economics (2006).

[65] A. Kwiatkowski and H. Werner, *Pca-based parameter set mappings for lpv models with fewer parameters and less overbounding,* IEEE Trans. on Control Systems Technology **16**, 781 (2008).

[66] E. Prempain and I. Postlethwaite, *Extended lmi projection conditions,* in *IEEE Conf. on Decision and Control* (1999).

[67] J. C. Doyle, *Structured uncertainty in control system design,* in *IEEE Conf. on Decision and Control* (1985).

[68] A. Helmersson, *Model reduction using lmis,* in *IEEE Conf. on Decision and Control* (1994).

[69] P. Apkarian and R. J. Adams, *Advanced gain scheduling techniques for uncertain systems,* IEEE Trans. Automatic Control **6**, 21 (1998).

[70] M. Sato and D. Peaucelle, *Gain-Scheduled Output-Feedback Controllers with Good Implementability and Robustness. In: J. Mohammadpour and C. W. Scherer (Eds.), Control of Linear Parameter Varying Systems with Applications* (Springer, Heidelberg, 2012).

[71] G. Stein and J. C. Doyle, *Beyond singular values and loop shapes,* AIAA J. of Guidance, Control, and Dynamics **14**, 5 (1991).

[72] M. A. Rotea and T. Iwasaki, *An alternative to the d-k iteration ?* in *Am. Control Conf.* (1994).

[73] K. Hornik, M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximators,* Neural Networks **2**, 359 (1989).

[74] M. Norgaard, O. Ravn, N. Poulsen, and L. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems* (Springer-Verlag, London, 2000).

6

[75] A. B. T. (Ed.), *Computer Science Handbook, Second Edition* (Chapman and Hall/CRC, 2004).

[76] J. D. Bendtsen and K. Trangbaek, *Robust quasi-lpv control based on neural state-space models,* IEEE Trans. on Neural Networks **13**, 355 (2002).

[77] H. Abbas and H. Werner, *Polytopic quasi-lpv models based on neural state-space models and application to air charge control of a si engine,* in *IFAC 17th World Congress* (2008).

[78] N. Lachhab, H. Abbas, and H. Werner, *A neural-network based technique for modelling and lpv control of an arm-driven inverted pendulum,* in *IEEE Conf. on Decision and Control* (2008).

[79] J. Lofberg, *Yalmip: A toolbox for modeling and optimization in matlab,* in *CACSD Conf.* (2004).

[80] M. B. Tischler, J. G. M. Leung, and D. C. Dugan, *Neural network-based trajectory optimization for unmanned aerial vehicles,* AIAA J. of Guidance, Control, and Dynamics **35**, 548 (2012).

[81] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design, 2nd Ed.* (Wiley-Interscience, Great Britain, 2005).

[82] P. Gahinet and P. Apkarian, *A linear matrix inequality approach to h_∞ control,* Int. J. Of Robust And Nonlinear Control **4**, 421 (1994).

[83] G. Zames, *On the input-output stability of time-varying nonlinear feedback systems-part i: Conditions derived using concepts of loop gain, conicity, and positivity,* IEEE Trans. Autom. Control **AC-11**, 228 (1966).

[84] C. A. Desoer and M. Vidyasagar, *Feedback Systems: Input-Output Properties* (Academic Press, New York, 1975).

[85] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control* (Prentice Hall, New Jersey, 1996).

[86] G. J. Balas, J. C. Doyle, K. Glover, A. K. Packard, and R. S. Smith, *μ Analysis and Synthesis Toolbox: User's Guide* (The MathWorks Inc., 1991).

[87] K. Glover, *All optimal hankel-norm approximations of linear multivariable systems and their $\mathcal{L}_\infty$-error bounds,* Int. J. of Control **39**, 1115 (1984).

[88] J. M. Biannic, C. Roos, and A. Knauf, *Design and robustness analysis of fighter aircraft flight control laws,* Europ. J. of Control **12**, 71 (2006).

[89] G. Becker and A. K. Packard, *Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback,* Systems & Control Letters **23**, 205 (1994).

6

[90] I. E. Kose, F. Jabbari, and W. E. Schmitendorf, *A direct characterization of l_2 gain controllers for lpv systems,* IEEE Trans. Autom. Control **43**, 1302 (1998).

[91] E. Prempain, *Gain-scheduling h_∞ loop shaping control of linear parameter-varying systems,* in *5th IFAC Symposium on Robust Control Design* (2006).

[92] M. C. de Oliveira, *Explicit Controller Parametrizations for Linear Parameter-Varying Affine Systems Using Linear Matrix Inequalities. In: J. Mohammadpour and C. W. Scherer (Eds.), Control of Linear Parameter Varying Systems with Applications* (Springer, Heidelberg, 2012).

[93] F. Wu, *A generalized lpv system analysis and control synthesis framework,* Int. J. of Control **74**, 745 (2001).

[94] C. Scherer and I. E. Kose, *Gain-scheduled control synthesis using dynamic d-scales,* IEEE Trans. Autom. Control **57**, 2219 (2012).

[95] C. W. Scherer, *Mixed $h_2 - h_\infty$ control for time-varying and linear parametrically-varying systems,* Int. J. of Robust and Nonlinear Control **6**, 929 (1996).

[96] P. Gahinet, P. Apkarian, and M. Chilali, *Affine parameter-dependent lyapunov functions and real parametric uncertainty,* IEEE Trans. Automatic Control **41**, 436 (1996).

[97] S. Bennani, D. M. C. Willemsen, and C. W. Scherer, *Robust control of linear parametrically varying systems with bounded rates,* AIAA J. of Guidance, Control, and Dynamics **21**, 916 (1998).

[98] S. Lim, *Analysis and Control of Linear Parameter Varying Systems*, Ph.D. thesis, Stanford University (1998).

[99] C. E. deSouza and A. Trofino, *Gain-scheduled h_2 controller synthesis for linear parameter varying systems via parameter-dependent lyapunov functions,* Int. J. of Robust and Nonlinear Control **16**, 243 (2006).

[100] P. Apkarian, G. Becker, P. Gahinet, and H. Kajiwara, *LMI Techniques in Control Engineering from Theory to Practice. In: Workshop notes CDC IEEE* (Kobe, Japan, 1996).

[101] A. Kominek, H. Werner, M. Garwon, and M. Schultalbers, *Identification of Low-Complexity LPV Input-Output Models for Control of a Turbocharged Combustion Engine* (Springer, Heidelberg, 2012).

[102] B. R. Barmish, *Necessary and sufficient conditions for quadratic stabilizability of an uncertain system,* J. of Optimization Theory and Applications **46**, 399 (1985).

[103] A. Helmersson, *Methods for Robust Gain-Scheduling*, Ph.D. thesis, Linkoping University (1995).

[104] P. Apkarian, J. M. Biannic, and P. Gahinet, *Self-scheduled h$_\infty$ control of missile via linear matrix inequalities,* AIAA J. of Guidance, Control and Dynamics **18**, 532 (1995).

[105] R. J. Adams, P. Apkarian, and J. P. Chretien, *Robust control approaches for a two-link flexible manipulator,* in *Int. Conf. Dynamics Control Structures in Space* (1996).

[106] H. Kajiwara, P. Apkarian, and P. Gahinet, *Lpv techniques for control of an inverted pendulum,* IEEE Control Systems Magazine **19**, 44 (1999).

[107] T. Nagashio and T. Kida, *Robust attitude controller design of linear parameter varying spacecraft via μ synthesis and gain scheduling,* in *IEEE Int. Conf. on Control Applications* (1999).

[108] J. Daafouz, J. Bernussou, and J. C. Geromel, *On inexact lpv control design of continuous-time polytopic systems,* IEEE Trans. Autom. Control **53**, 1674 (2008).

[109] A. Rantzer, *On the kalman-yakubovich-popov lemma,* Systems & Control Letters **28**, 7 (1996).

[110] T. Iwasaki, G. Meinsma, and M. Fu, *Generalized s-procedure and finite frequency kyp lemma,* Math. Prob. Eng. **6**, 305 (2000).

[111] M. R. Graham, M. C. de Oliveira, and R. A. de Callafon, *An alternative kalman-yakubovich-popov lemma and some extensions,* Automatica **45**, 1489 (2009).

[112] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox User's Guide* (The Mathworks Partner Series, 1995).

[113] C. Scherer, *The Riccati Inequality and State-Space H$_\infty$-Optimal Control*, Ph.D. thesis, Universitat Wurzburg (1990).

[114] C. W. Scherer, P. Gahinet, and M. Chilali, *Multiobjective output-feedback control via lmi optimization,* IEEE Trans. Autom. Control **42**, 896 (1997).

6

7

CONCLUSIONS AND FUTURE RESEARCH

Perfect is the enemy of good.

Aphorism commonly attributed to Voltaire

A good enough solution that works, is immeasurably better than a perfect solution yet to be implemented.

Justin Lloyd
A Mastermind's Guide to Personal Development, 2009

In this Chapter the most important results achieved in this thesis are first presented, and further objectives and opportunities for future research are identified and outlined.

7.1. CONTRIBUTION OF THIS THESIS

T HE primary objective of this thesis was to develop a, model-based, automatic safety re- covery system that could safely fly and land a small-scale helicopter Unmanned Aerial Vehicle (UAV) in un-powered flight (i.e. autorotation). The flight control solution presented in this thesis incorporates a classic guidance and control logic, in which the guidance module is decoupled from the control module. The goal of the guidance module, or Trajectory Planning (TP), is to generate open-loop, feasible and optimal autorotative trajectories, for the helicopter, whereas the aim of the control module, or Trajectory Tracking (TT), consists in comparing the current state values with the optimal reference values produced by the TP, and then formulate the feedback controls, enabling thus the helicopter to fly along these optimal trajectories. The work presented in this thesis resulted in the first demonstration of a, real-time feasible, model-based TP and model-based TT, for the case of a small-scale helicopter UAV, with an engine OFF condition (i.e. autorotation). The validation was per- formed on a helicopter high-fidelity simulation, based upon a nonlinear, High-Order Model (HOM). In the sequel we outline additional concluding remarks, relative to the various solutions and results presented in this thesis.

- With regard to **helicopter modeling**, we developed two helicopter nonlinear models. One is a first-principles based, HOM, developed in Chapter 2, and used to validate the Flight Control System (FCS). The second one is a gray-box[1], Low-Order Model (LOM), developed and used in Chapter 3 to obtain optimal autorotative trajectories. The latter model provides a good approximation of the HOM of Chapter 2, while having better computational efficiency when compared to the HOM. However, this comes at a price, namely a time-consuming identification of various empirical coeffi- cients, using input-output data from the HOM. In addition, each new helicopter con- figuration, or modification thereof (e.g. mass and inertia adjustments), will require a re-identification of all empirical coefficients. By contrast, the HOM represents a flexible modeling approach, readily updated in case of new helicopter configurations, although its associated CPU time, per model evaluation, is higher.

- With respect to the **off-line TP**, developed in Chapter 3, based upon the realm of constrained, nonlinear optimal control, we summarize here the main findings

  1. For fixed initial altitude, increasing the initial velocity has only a relatively lim- ited effect on flight time and stabilized rate of descent.

  2. For fixed initial altitude, the flight time is strongly correlated with the initial altitude and the induced velocity in hover.

  3. For fixed initial altitude, increasing the initial velocity complicates somewhat the flare maneuver.

  4. For hover initial conditions, the higher the initial altitude, the more the optimal autorotative trajectory resembles a vertical flight path.

- With respect to the **on-line TP and TT** of Chapters 4 and 5, using the combined paradigms of differential flatness and robust control, we summarize here the main

---

[1]Using a mix of first-principles and various empirical coefficients.

findings for both the engine ON (i.e. power-on) and engine OFF (i.e. power-off, also known as autorotation) flight conditions, for the case of a small-scale helicopter UAV

1. The proposed TP and TT approach is validated on the high-fidelity, first-principles based, helicopter HOM, developed in Chapter 2, for both engine ON and engine OFF trajectories. The methodology is real-time feasible since it allows for a computationally tractable determination, and tracking, of the optimal trajectories. In addition, both the engine ON and engine OFF cases are based upon the same planning and tracking system architecture. Further, main rotor Revolutions Per Minute (RPM) is not used, being neither necessary for the engine OFF trajectory planning, nor for the corresponding trajectory tracking, hence simplifying the overall system design.

2. For the engine OFF case, a single Linear Time-Invariant (LTI) controller is capable of controlling and landing the helicopter system, in autorotation, for a relatively large variation in forward and vertical vehicle velocity (at least up to approximately 8 to 10 m/s), and for relatively large variations in main rotor RPM (approximately in the 50% to 110% range).

3. For the engine ON case, the vehicle state at an initial time t_i has only a limited impact (if any) on the set of reachable states for all admissible input signals and for all time instants in an interval $[0, t_f]$, with $t_f \gg t_i$. If we omit the on-board electrical power supply system from the vehicle energy balance, i.e. considering only vehicle potential, kinetic, and main rotor energies, then the total vehicle energy may decrease or increase, depending on vehicle height and vehicle velocity. By contrast, the total vehicle energy, in the engine OFF case, is always decreasing. Hence, we conjecture that the size of this reachable set, for the engine OFF case, is much smaller when compared to its engine ON counterpart and, consequently, feasible trajectories are much harder to find in the engine OFF case.

4. For the engine ON case, it is relatively easy to find equilibrium points, i.e. steady-state flight conditions, at which the nonlinear model can be linearized. The so-obtained LTI models can subsequently be used for LTI control design. For the engine OFF case, this set of equilibrium points, i.e. steady autorotative flight conditions, is rather small and in certain situations even non-existent. For example, when an engine failure happens at a low altitude, the helicopter does not even reach a steady-state autorotation, rather it is continuously in transition from one non-equilibrium point to the next. To mitigate this problem, the approach used in this thesis consists in excluding the main rotor RPM from the state-vector, and use the resulting "quasi-steady" approach to find the equilibrium points.

5. For the engine ON case, helicopter operations can remain at a velocity which stays in the neighborhood of the design-point velocity, i.e. in the neighborhood of the equilibrium point velocity which was used to derive the LTI model for control design. This allows to maximize the linear behavior of the system. On the other hand, helicopter operations with the engine OFF will inevitably result in a wide range of flown velocities, including high descent rates, and even flight

7

into the chaotic Vortex-Ring-State (VRS). Indeed, a brief transition through the VRS may in some cases be required. This obviously tends to 'amplify' the nonlinear behavior of the system.

6. For the engine ON case, the designer can choose to keep the bandwidth of the closed-loop system rather small, by only considering gentle and smooth maneuvers in the design specification phase. For the engine OFF case, a higher closed-loop bandwidth is definitely required (especially in the vertical channel), if proper trajectory tracking is to be performed. This may complicate the controller design, since higher-order LTI models (for controller design) may have to be considered. This complicates also the practical implementation, since higher-bandwidth actuators may become compulsory.

7. For the engine OFF case, our results show that the crucial control of vertical position and velocity exhibit outstanding behavior in terms of tracking performance, and does not require an additional increase in control bandwidth. However, the tracking of horizontal positions and horizontal velocities is clearly lacking some bandwidth (i.e. the flown trajectories are clearly lagging the planned ones). Although a further increase of the horizontal closed-loop bandwidths provided good results when evaluated on the LTI model used for control design, this increase in closed-loop bandwidths resulted, unfortunately, in closed-loop instabilities, when evaluated on the nonlinear helicopter model of Chapter 2.

8. For the engine OFF case, tracking performance of horizontal positions and horizontal velocities could potentially be improved, by considering one of the two following options: i) remaining in the framework of a single robust LTI controller, using a high-order LTI plant for controller design (i.e. containing the main rotor flap-lag and inflow dynamics), instead of a low-order plant as used in Chapter 4; or ii) using another control method that better respects and exploits the system's nonlinear structure, e.g. in the realm of nonlinear, adaptive, or Linear Parameter-Varying (LPV) methods.

- With respect to the **affine LPV modeling** method, developed in Chapter 6, we have shown, using a pointmass pendulum (i.e. a nonlinear example), that our LPV modeling strategy was capable of reproducing the open-loop behavior of the original nonlinear dynamical system. Furthermore, we have shown that controllers (whether robust or LPV), designed using our LPV model, achieved better reference tracking, when compared to a controller designed using a linearization of the nonlinear system.

7.2. RECOMMENDATIONS FOR FUTURE RESEARCH

In light of the research objective of this thesis and the results achieved so far, we identify and discuss next some stimulating opportunities for future research. In particular, if the next step is to perform flight tests and achieve an experimental validation of an automatic autorotation system, then the general control architecture, as used in this thesis, and outlined in Fig. 1.15 of Chapter 1, may have to be replaced by the one given in Fig. 7.1. In the sequel we will elaborate on the new blocks of Fig. 7.1, as well as several other areas that warrant further exploration.
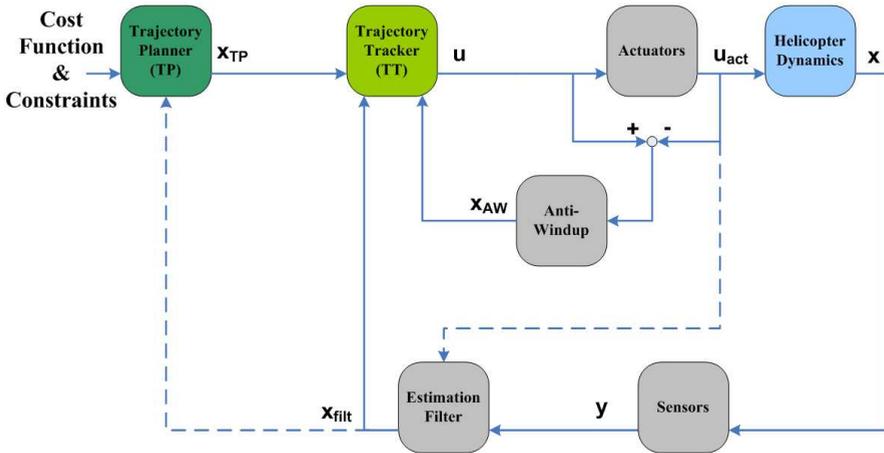
Figure 7.1: Upgraded two degree of freedom control architecture.

- For the case of an engine failure, the **"engine OFF event" first needs to be recognized**. Here the use of an engine torque sensor could prove very useful. For example, a sudden reduction in measured torque, if accompanied by fixed main rotor collective input and a decelerating main rotor speed, could be indicative of an engine failure. However, a sudden reduction in engine torque if accompanied by, either, reduced main rotor collective input or accelerating main rotor speed, would not indicate an engine failure [1]. An additional clue for the detection of engine failure could also come from the yaw channel. Indeed, a jerk is generally felt on this channel, since the tail rotor will tend to overcompensate the reduced main rotor torque [1]. As a final point, it should be noted that, for the case where engine power is not lost suddenly but rather gradually, it may become much more difficult to quickly detect such a failure.

- **Actuator dynamic models, with amplitude and rate constraints, ought to be included** in the HOM of Chapter 2, and in any model used for control design. Indeed, it is well-known that maximum control gains and crossover frequencies may be limited by actuator rate saturation[2]. Further, actuator rate saturation can have a significant detrimental effect on the handling qualities of an aerospace vehicle [3], and directly lead into, either, degraded performance, limit cycles, or even closed-loop instability [3, 4]. For example, the crashes of the SAAB Gripen fighter jet in February 1989 and August 1993 [2, 5], and the crash of the Lockheed Martin YF-22 fighter jet in April 1992 [2, 6] are all primarily related to actuator rate saturations. These saturations resulted in so-called Pilot-Induced-Oscillation (PIO), and subsequent loss of vehicle control. Several approaches could be adopted to avoid saturation problems in systems which are known to have actuator limits. The first one lives in the realm of optimal control. Here the control action is decided through the use of constrained op-

---

[2]Actuator saturation or rate limits has even been implicated in the meltdown of the Chernobyl nuclear power plant, in April 1986 [2].

timization algorithms, known as Model Predictive Control (MPC) [7]. Other options
are related to the so-called anti-windup compensation, in which a nominal controller
(that does not explicitly take the saturation constraints into account) is first designed.
Then, in a second step, an anti-windup compensator is designed to handle the satura-
tion constraints. Anti-windup approaches are attractive in practice because: 1) they
allow for control design in a linear framework; and 2) no restrictions are placed upon
the nominal controller design. Excellent tutorials exist in this area, see [8–10]. Re-
cently, promising extensions have even considered the case of a simultaneous design
of both the nominal controller and the anti-windup compensator [11, 12].

- An **estimation filter**, e.g. a state estimator, is typically an integral part of a UAV FCS.
  Indeed, the quantities required for flight control, like position, velocity, and attitude,
  are not measured directly or, if measured, are noisy and often not available at the re-
  quired frequency. Hence, an estimation filter is often required to derive smooth, and
  high-frequency state updates, from available sensor measurements. For example, our
  Align T-REX helicopter is fitted with a flight computer featuring data logging capa-
  bilities, as well as a variety of low-cost sensors, such as: 1) an Inertial Measurement
  Unit (IMU) containing three accelerometers and three gyroscopes that measure ac-
  celerations and angular velocities, respectively, in the inertial body frame; 2) a Global
  Positioning Sensor (GPS) providing a direct measurement of the helicopter's inertial
  position and velocity; 3) a compass measuring the vehicle's magnetic heading; 4) a
  barometric pressure sensor for altitude measurement during cruise flight; and a 5)
  a Laser Range Finder (LRF) for altitude measurement during take-off and landing.
  Hence, the helicopter's position, velocity, and attitude can be obtained through the
  integration of the high-frequency, noisy, biased, and drifting IMU outputs, with the
  noisy, low-frequency outputs, with bounded error characteristics, of the remaining
  sensors. Since in our case the vehicle's kinematic and measurement equations are
  nonlinear, the nonlinear extension to the original Kalman Filter, i.e. the Extended
  Kalman Filter (EKF) [13, 14], represents the most common approach for our real-
  time estimation problem. However, since based upon linearizations and calculation
  of Jacobian matrices, the EKF is also known to exhibit numerical issues and even
  divergence in some situations. To mitigate such problems, the so-called Unscented
  Kalman Filter (UKF) [15, 16] has been developed. For all its benefits, it was reported
  in [17–19] that, for the case of aerospace applications, the UKF did not offer substan-
  tial performance gains, when compared to the EKF. Hence, for our application, we
  would recommend evaluating first the simpler EKF filter.

- Small-scale UAVs are far more sensitive to atmospheric wind and gust disturbances,
  than their full-scale counterparts, since the mean wind magnitude is often compara-
  ble to the speed of the UAV, and consequently this brings upfront the relevance of **a
  mean wind estimation capability**. The knowledge of the mean wind profile mag-
  nitude, and direction, is indeed helpful for two reasons. First, it allows to enhance
  the accuracy and feasibility of the computed trajectories during the planning phase,
  since knowledge of the wind can be included in the model used for planning. Sec-
  ond, for good trajectory tracking[3], the velocities of the vehicle with respect to the

---

[3]In flight dynamics models, the aerodynamic forces are functions of the vehicle aerodynamic velocities, not of

relative wind, i.e. the vehicle aerodynamic velocities, should be made available to the controller. Direct wind measurement can be obtained through, either, a ground-based anemometer, or through some sort of weather balloon. The first option does not provide information on wind profile (as a function of altitude), whereas the second may be costly, and potentially impractical. Hence, the need for wind estimation, rather than wind measurement, becomes obvious. With regard to kinematics, the vehicle's ground track velocity vector (i.e. the inertial velocity, measured with GPS) can be decomposed into the sum of a vehicle's airspeed vector and a wind vector. As stated earlier, GPS data is available on-board the helicopter. Hence, if the wind velocity vector is known, it can be subtracted from the GPS velocity to obtain an air-relative velocity. Alternatively, if the air-relative velocity vector is known, it can be subtracted from the GPS velocity vector to obtain the wind velocity vector. The determination of the vehicle's air-relative velocity vector can be done through two approaches. The first approach, and widely used approach for fixed-wing aircrafts, consists in mounting an air-data unit, combining precise measurements of airspeed amplitude, through a pitot-static pressure sensor, and airspeed orientation, through angle-of-attack and angle-of-sideslip vanes. The second approach is a model-based one (often derived from relatively simple models) in which the vehicle's air-relative velocity vector is estimated based upon the knowledge of the model, and based upon the measured control inputs. Here, the first approach is generally ruled out for helicopters, since such an air-data system needs to operate outside the main rotor downwash and, even if placed at the front of the fuselage, may only be effective when the vehicle is traveling at high forward speed. Hence, the preferred approach for wind and airspeed estimation, for helicopters, consists in using a model-based estimation procedure, together with GPS and control input measurements (sometimes also in combination with heading measurements from the compass sensor). Such a strategy has often successfully been applied to the case of autonomous guided airdrop systems (i.e. paraplanes), see also [20, 21].

- For the **Low-Order Model (LOM)** of Chapter 3, the empirical coefficients are estimated in a multiple-model structure, meaning that for each point in the operating grid, a set of coefficients is being identified thanks to data generated from the High-Order Model (HOM). However, as stated earlier, this identification method becomes rather tedious for large grids. An alternative approach, potentially easier to implement since not based upon the multiple-model concept, consists in identifying the coefficients using the optimal control framework. Here, the empirical coefficients constitute the unknown control inputs of a continuous-time, nonlinear, dynamical system. These inputs are obtained by solving a constrained, optimal control problem, which goal is to fit the outputs of the LOM with those of the HOM in some optimal sense. Once identified, a model defining the relation between these empirical coefficients and the helicopter control inputs and states, needs to be found (e.g. through a Neural Networks (NN) representation). In [22] we presented preliminary results for such a LOM approach.

- With regard to the **off-line Trajectory Planning** of Chapter 3, we discuss several

---

vehicle inertial velocities.

areas that may benefit from further improvements

1. As stated in Section 3.4, direct optimal control methods have several advantages when compared to indirect methods. Specifically, the first-order necessary conditions do not need to be explicitly derived, and the large radii of convergence allow for less accurate initial guess on states and control inputs. Hence, direct methods are appealing for complicated applications. Further, PseudoSpectral (PS) discretization methods have the known advantage of providing exponential rate of convergence for the approximation of analytic functions. For all those benefits, the direct optimal method used in our application has also shown some inherent limitations. For example, it was in some cases uncertain whether the solution obtained was truly optimal. Indeed, fluctuated solutions were observed as the number of discretization nodes was varied.

2. We also noticed that the use of lookup tables, within the LOM, had a negative impact on the exponential convergence of the method, even when queried with cubic B-Splines interpolating functions. Solving the optimal control problem became at times computationally intractable, and at times either infeasible, or feasible but very probably sub-optimal. This said, nonsmooth problem formulations are far from uncommon in aerospace. To mitigate this known issue, several approaches could be investigated such as: 1) a PS knotting method as in [23]; or 2) a hybrid global/local collocation method as in [24].

3. We solved the NonLinear Programming problem (NLP) via a Sequential Quadratic Programming (SQP) approach. SQPs belong to the class of iterative, gradient-based methods, and gradient methods are known as local methods. We did notice this sensitivity to local minima, by obtaining distinct optimal solutions, for distinct initial guesses.

4. Since in our case we did not use any mesh refinement grid (as to keep the problem computationally tractable), the obtained optimal solution provided only the state and control values at the discrete nodes. Hence, the optimal solution satisfied only the discretized constraints (i.e. the problem is said to be discrete-time feasible [25]). This implies that, for a small number of nodes, no guarantees may be given regarding the solution of the original continuous-time problem [25]. Obviously, one way to mitigate such a problem would be to increase the number of nodes, at the cost of higher computational time.

5. Finally a robustness analysis of the obtained trajectories[4], with respect to model and signal uncertainties, potentially within the realm of stochastic optimization, would represent an interesting avenue for future research.

- With regard to the **on-line Trajectory Planning (TP)** of Chapters 4 and 5, we recommend considering the following aspects

1. It may be beneficial to add a feedback path into the trajectory planning, denoted by a dashed line in Fig. 7.1, which would allow to re-generate an optimal reference trajectory, based upon the current state. This functionality may, for

---

[4]This could also apply to the flatness-based trajectory planning.

example, be of interest in the following cases: 1) within the framework of an obstacle avoidance capability; or 2) if the helicopter experiences an increasing difficulty at tracking the current reference trajectory.

2. One could also consider adding an optical sensor, coupled with an on-board 3D map of the environment, in order to identify suitable geographical locations for a safe landing. If in addition the set of reachable states could efficiently be computed on-line, then one would be able to provide feasible landing positions to the TP.

3. The optical sensor could also potentially be fused with the other sensors in order to increase the accuracy of the estimated helicopter state-vector (computed in the estimation filter).

- With regard to the **on-line Trajectory Tracking (TT)** of Chapters 4 and 5, we recommend considering the following aspects

  1. The NL helicopter model of Chapter 2 is subject to periodic loads, due to blades rotation, that result in a time-varying trim condition. Linearizing the NL helicopter dynamics, around this trim condition, can be done at each rotor position, to yield a Periodic Linear Time-Varying (PLTV) system, with a period equal to one rotation of the rotor. For PLTV systems the classical modal analysis methodologies, based upon time-invariant eigenstructures, are not applicable anymore [26]. Hence, if one wants to apply the well-established analysis and control tools for LTI systems, a transformation of the PLTV system into a LTI one becomes necessary. There are roughly four main methods to perform such a transformation or approximation [27]. The first, and simplest one, consists in evaluating the PLTV system at a single rotor position (i.e. at a single blade azimuth position), and obtain a LTI system. Clearly this approach may lead to poor results. An already better method would consist in averaging the PLTV state-space matrices over one or more rotor periods. The next two methods provide LTI models with higher accuracy, but require additional mathematical steps. The third method uses Floquet theory [26, 28], and the associated characteristic exponents called Floquet multipliers, to obtain constant state-space matrices. The fourth method uses the so-called Multi-Blade Coordinate (MBC) transformation (also known as the Coleman transformation) [26, 29–31], i.e. by transforming quantities from rotating blade coordinates into a non-rotating frame. Basically the MBC describes the overall motion of a rotating blade array in the inertial frame of reference. The MBC transformation results in a weakly periodic system which is subsequently converted into a LTI system by averaging over one period [31]. In this thesis, obtaining an LTI approximation from the PLTV system was done using the second method as discussed in Section 2.4.1 of Chapter 2, by averaging over four rotor periods. Although very easy to implement, it is well-known that this method may not provide an LTI model of highest accuracy. Hence, we recommend trying a more sophisticated approach to derive the LTI system. With regard to the MBC method, this latter is particularly well-suited for rotors having three or more blades, and may in-

7

volve significant inaccuracies for a two-bladed rotor[5] [32]. The Floquet method is numerically more intensive[6] than the averaging method used in this thesis [35], however it may potentially provide LTI models with higher accuracy and hence would deserve further investigations.

2. For a digital implementation of the controller, several continuous- to discrete-time transformations exist (depending on the type of control framework used [36, 37]). The goal, obviously, is to select a transformation that best preserves the properties of the continuous-time design.

3. A general approach to mitigate the interaction problem, between the FCS and the main rotor dynamics, could be to use higher-order LTI models (i.e. containing the main rotor flap-lag and inflow dynamics), for control design, possibly in combination with a reduced-order observer in order to estimate the unmeasured main rotor states.

4. The addition of a roll and pitch attitude stabilization loop may potentially allow to increase the tracking bandwidth. The complete control system would then involve multiple nested control loops, namely: 1) the innermost-loop, which controls the attitude of the vehicle; 2) the middle-loop, which controls the velocity; and 3) the outer-loop, which controls the position.

5. Since system delays impose severe limitations on the bandwidth of the closed-loop system [38], all hardware delays—due to actuator dead-time, sensor processing, and the effects of digital implementation on-board the embedded computer—need to be estimated, modeled, and added to all models developed within this thesis.

6. Helicopter dynamics is highly coupled, especially during hover and low-speed flight. In order to reduce the coupling effects, and hence simplify the subsequent controller design, it may be worthwhile to add a decoupling module, in the form of open-loop dynamic crossfeeds, inserted in-between the controller outputs and the plant inputs, see [39, 40].

7. It is customary to place the closed-loop poles in a suitable region of the complex plane. This is often done in order to guarantee satisfactory system transients behavior, and to indirectly enforce constraints on the controller bandwidth, and hence: 1) minimize any controller interaction with actuator dynamics, structural modes, or any other vehicle higher-order dynamics; and 2) allow for a digital implementation of the controller dynamics. This can be done, in a systematic way for LTI controllers, in the Linear Matrix Inequality (LMI) framework, see [41, 42].

8. In Chapters 4 and 5, a single nominal LTI model was used for the design of a single robust LTI TT controller. Relatively good tracking results have been obtained, although the tracking of horizontal velocity and position could potentially be improved, by considering one of the two following options: 1) remaining in the framework of a single robust LTI controller, however combined with

---

[5]As a reminder, our Remote-Controlled (RC) Align T-REX helicopter has a two-bladed main rotor.
[6]Although some progress has been done with Fast-Floquet methods [33, 34].

a higher-order LTI plant, instead of the low-order plant used in Section 4.5.1 of Chapter 4; or 2) by considering a more sophisticated control method, which better exploits the system's nonlinear structure. If the nonlinear HOM of Chapter 2 could somewhat be simplified, and written in closed-form, then an additional plethora of nonlinear control tools would become available, such as: feedback linearization, (incremental) nonlinear dynamic inversion, or nested saturated control in [43–46], backstepping in [47–50], adaptive control in [50–53], and even passivity-based control approaches [54]. On the other hand, if a low-order formulation of the LPV model of Chapter 6 could be obtained, then here too another array of control options may become available: obviously LPV [55], but also the application of Model Predictive Control (MPC) to LPV Systems [56, 57], or PieceWise Affine (PWA) control [58–60]. To the best of our knowledge, the last two control options have not even been applied to a six degree of freedom helicopter system, as yet. Beyond these well-known options, we also mention the recent developments in the area of nonsmooth optimization for control [61–63], which allow the formulation of multiple competing objectives (in time- or frequency-domain), subject to additional structural constraints such as: controller order, and/or state/input time-domain specifications. Although endowed with local convergence certificates only, this newly emerging approach is very promising, since it avoids the use of Lyapunov variables, and hence is numerically efficient for large systems. Ultimately, it would be rather fascinating to be able to compare some, or most, of the previously mentioned TT methods, and investigate the various pros and cons of each method.

9. Instead of using LTI control methods, and if blade azimuth measurement is available, one could also consider using a PLTV nominal model for control design, in combination with a periodic control method [64, 65], and check whether better tracking performance could so be achieved. For the case of wind turbines, it was shown in [32] that periodic Linear Quadratic Regulators (LQRs) performed no better than LQRs synthesized in the LTI framework. However, periodic control has also been extended to H_∞ and MPC control methods [64] and it would be interesting to further evaluate these alternative control methods.

10. One could also consider adding some preview control to the current architecture. Indeed, since the optimal trajectory is precomputed, one could use a noncausal controller (based upon future information with regard to the reference signals) in order to increase the overall closed-loop bandwidth [38, 66].

11. In this thesis we used a TT approach, i.e. tracking a time-parameterized reference trajectory. This said, within the field of motion control for autonomous vehicles, the path-following approach is rather popular. The idea of path-following is to have the vehicle converge to, and follow, a path without temporal restrictions. When compared to trajectory-tracking, path-following strategies seem to exhibit enhanced performance, smoother convergence, and reduced control effort [67, 68].

12. Finally, a variety of robustness related topics could be considered. First, our nominal LTI controller designed with one linearized model could be applied to

7

other linearized models[7], as a first step towards controller validation [69, 70]. In this thesis, we skipped this intermediate step to go directly to the controller validation on the nonlinear HOM of Chapter 2. Next, we only provided a preliminary demonstration of the robustness of the FCS with respect to sensors noise and wind disturbance. Hence, we do recommend a more thorough analysis of the wind disturbance rejection capability. Further, it was also shown in [71] that by adding an acceleration feedback loop, one could attenuate the effects of model uncertainties and disturbances, and could improve tracking performance. Also, depending on the selected model-based control method, robustness guarantees could either be provided *a priori*, through e.g. LPV control techniques, or *a posteriori* as in [72] by: 1) first applying classical gain-scheduling techniques in the control design process; 2) next, obtaining a Linear Fractional Representation (LFR) of the global closed-loop system; and 3) finally analyzing the system robustness by invoking results from Integral Quadratic Constraints (IQC) theory [73].

- With regard to the **on-line Trajectory Planning and Trajectory Tracking** of Chapters 4 and 5, one could also consider an integrated approach rather than our segregated TP-TT approach, see our discussion in Section 1.5.2 of Chapter 1. In particular if the nonlinear helicopter plant can be modeled as a LPV system then one of the many MPC-LPV algorithms, i.e. MPC for LPV systems [56, 57, 74–88], could be used.

- With regard to the **affine LPV model** of Chapter 6, the method was applied to the helicopter HOM of Chapter 2 and resulted in a LPV model having a large number of scheduling parameters. Unfortunately, it became impossible to synthesize LPV controllers with such a high-order LPV model. In fact, the numerical conditioning and solvability of LMI problems play a crucial role in LPV practical design methods [89–92]. As such, we recommend applying some LPV model reduction techniques [93, 94] in order to obtain a LPV model having fewer scheduling parameters, thus better suited for LPV controller synthesis. Another aspect could be to consider replacing the H_∞ framework, used in the LPV modeling process, by the nu-gap metric [95–97]. This latter provides a measure of the separation between open-loop systems, in terms of their closed-loop behavior. Hence the nu-gap may potentially provide some added-value, when modeling for control. Finally, our LPV modeling method was applied for the case of a single and simple example, i.e. the pointmass pendulum. Although preliminary encouraging results were obtained, definitive conclusions may only be drawn after some sort of Monte-Carlo type simulations performed on a variety of nonlinear plants.

7

---

[7]These linearized models are obtained by griding the flight envelope.

REFERENCES

[1] H. H. McIntyre, *A simplified study of high speed autorotation entry characteristics,* in *Am. Helicopter Soc.* (1970).

[2] G. Stein, *Respect the unstable,* IEEE Control Systems Magazine **23** (2003).

[3] M. Pachter and R. B. Miller, *Manual flight control with saturating actuators,* IEEE Control Systems Magazine **18**, 10 (1998).

[4] M. B. Tischler, C. M. Ivler, M. H. Mansur, K. K. Cheung, T. Berger, and M. Berrios, *Handling-qualities optimization and trade-offs in rotorcraft flight control design,* in *RAeS Rotorcraft Handling-Qualities Conf.* (2008).

[5] Anon., *Why the grippen crashed,* Aerospace America , 11 (1994).

[6] M. A. Domheim, *Report pinpoints factors leading to yf-22 crash,* Aviation Week and Space Technology , 53 (1992).

[7] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design* (Nob Hill Publishing, Madison, Wisconsin, 2013).

[8] S. Galeani, S. Tarbouriech, M. Turner, and L. Zaccarian, *A tutorial on modern anti-windup design,* Europ. J. of Control **15**, 418 (2009).

[9] S. Tarbouriech and M. Turner, *Anti-windup design: An overview of some recent advances and open problems,* IET Control Theory & Applications **3**, 1 (2009).

[10] L. Zaccarian and A. R. Teel, *Modern Anti-Windup Synthesis: Control Augmentation for Actuator Saturation* (Princeton University Press, 2011).

[11] E. F. Mulder, P. Y. Tiwari, and M. V. Kothare, *Simultaneous linear and anti-windup controller synthesis using multiobjective convex optimization,* Automatica **45**, 805 (2009).

[12] J. M. Biannic and P. Apkarian, *Anti-windup design via nonsmooth multi-objective h_∞ optimization,* in *Am. Control Conf.* (2011).

[13] G. T. Schmidt, *Practical Aspects of Kalman Filtering Implementation*, Tech. Rep. 82 (AGARD-NATO Lecture Series, 1976).

[14] A. Gelb, *Applied Optimal Estimation* (MIT Press, Mass., 1974).

[15] S. J. Julier and J. K. Uhlmann, *Unscented filtering and nonlinear estimation,* Proceedings of the IEEE **92**, 401 (2004).

[16] D. Simon, *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches* (John Wiley & Sons, 2006).

[17] J. Wendel, A. Maier, J. Metzger, and G. F. Trommer, *Comparison of extended and sigma-point kalman filters for tightly coupled gps/ins integration,* in *AIAA Guidance, Navigation and Control Conf.* (2005).

7

[18] G. Chowdhary and R. Jategaonkar, *Aerodynamic parameter estimation from flight data applying extended and unscented kalman filter,* in *AIAA Atmospheric Flight Mechanics Conf.* (2006).

[19] M. Rhudy, Y. Gu, J. Gross, and M. R. Napolitano, *Sensitivity analysis of ekf and ukf in gps/ins sensor fusion,* in *AIAA Guidance, Navigation and Control Conf.* (2011).

[20] M. Ward, M. Costello, and N. Slegers, *On the benefits of in-flight system identification for autonomous airdrop systems,* AIAA J. of Guidance, Control, and Dynamics **33**, 1313 (2010).

[21] M. Ward, M. Costello, and N. Slegers, *Specialized system identification for parafoil and payload systems,* AIAA J. of Guidance, Control, and Dynamics **35**, 588 (2012).

[22] S. Taamallah, *Identification of a nonlinear grey-box helicopter uav model,* in *AIAA Atmospheric Flight Mechanics Conf.* (2013).

[23] I. M. Ross and F. Fahroo, *A direct method for solving nonsmooth optimal control problems,* in *IFAC 15th Triennial World Congress* (2002).

[24] C. L. Darby, W. W. Hager, and A. V. Rao, *Direct trajectory optimization using a variable low-order adaptive pseudospectral method,* AIAA J. of Spacecraft and Rockets **48** (2011).

[25] I. M. Ross, P. Sekhavat, A. Fleming, and Q. Gong, *Optimal feedback control: Foundations, examples, and experimental results for a new approach,* AIAA J. of Guidance, Control, and Dynamics **31** (2008).

[26] W. Johnson, *Helicopter Theory* (Dover Publications Inc., NY, USA, 1994).

[27] P. Seiler and A. Ozdemir, *An optimal time-invariant approximation for wind turbine dynamics using the multi-blade coordinate transformation,* in *Am. Control Conf.* (2013).

[28] K. Stol, M. Balas, and G. Bir, *Floquet modal analysis of a teetered rotor wind turbine,* Journal of Solar Energy Eng. **124**, 364 (2002).

[29] R. Coleman and A. Feingold, *Theory of Self-Excited Mechanical Oscillations of Helicopter Rotors with Hinged Blades*, Tech. Rep. TR 1351 (NACA, 1958).

[30] D. J. Malcolm, *Modal response of 3-bladed wind turbines,* Journal of Solar Energy Eng. **124**, 372 (2002).

[31] G. Bir, *Multi-blade coordinate transformation and its applications to wind turbine analysis,* in *AIAA Wind Energy Symposium* (2008).

[32] K. A. Stol, H. G. Moll, G. Bir, and H. Namik, *A comparison of multi-blade coordinate transformation and direct periodic techniques for wind turbine control design,* in *AIAA Aerospace Sciences Meeting* (2009).

7

[33] D. A. Peters, *Fast floquet theory and trim for multi-bladed rotorcraft,* J. of the Am. Helicopter Soc. **39**, 82 (1994).

[34] O. A. Bauchau and Y. G. Nikishkov, *An implicit floquet analysis for rotorcraft stability evaluation,* J. of the Am. Helicopter Soc. **46** (2001).

[35] O. A. Bauchau and J. Wang, *Efficient and robust approaches for rotorcraft stability analysis,* J. of the Am. Helicopter Soc. **55** (2010).

[36] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems* (Prentice Hall, USA, 1998).

[37] R. Toth, P. S. C. Heuberger, and P. M. J. Van den Hof, *Discretization of linear parameter-varying state-space representations,* IET Control Theory & Applications **4**, 2082 (2010).

[38] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design, 2nd Ed.* (Wiley-Interscience, Great Britain, 2005).

[39] R. P. Cheng, *Rotorcraft Flight Control Design Using Quantitative Feedback Theory and Dynamic Crossfeeds*, Tech. Rep. (Master Thesis, California Polytechnic State University, 1995).

[40] S. C. van 't Hoff, *Robust Decoupling of the NH90 FLIGHTLAB Model*, Tech. Rep. NLR-CR-2013-556 (NLR, 2013).

[41] M. Chilali and P. Gahinet, h_∞ *design with pole placement constraints: an lmi approach,* IEEE Trans. Automatic Control **41**, 358 (1996).

[42] M. Chilali, P. Gahinet, and P. Apkarian, *Robust pole placement in lmi regions,* IEEE Trans. Automatic Control **44**, 2257 (1999).

[43] D. H. Shim, T. J. Koo, F. Hoffmann, and S. Sastry, *A comprehensive study on control design of autonomous helicopter,* in *IEEE Conf. on Decision and Control* (1998).

[44] K. Peng, G. Cai, B. M. Chen, M. Dong, K. Y. Luma, and T. H. Lee, *Design and implementation of an autonomous flight control law for a uavhelicopter,* Automatica **45**, 2333 (2009).

[45] A. Isidori, L. Marconi, and A. Serrani, *Robust nonlinear motion control of a helicopter,* IEEE Trans. Autom. Control **40**, 413 (2003).

[46] P. Simplicio, M. D. Pave, E. vanKampen, and Q. P. Chu, *An acceleration measurements-based approach for helicopter nonlinear flight control using incremental nonlinear dynamic inversion,* Control Engineering Practice **21**, 1065 (2013).

[47] R. Mahony, T. Hamel, and A. Dzul, *Hover control via lyapunov control for an autonomous model helicopter,* in *IEEE Conf. on Decision and Control* (1999).

[48] E. Frazzoli, M. A. Dahleh, and E. Feron, *Trajectory tracking control design for autonomous helicopters using a backstepping algorithm,* in *Am. Control Conf.* (2000).

7

[49] R. Olfati-Saber, *Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles*, Ph.D. thesis, Massachusetts Institute of Technology (2001).

[50] B. J. Guerreiro, C. Silvestre, R. Cunha, C. Cao, and N. Hovakimyan, *L1 adaptive control for autonomous rotorcraft,* in *Am. Helicopter Soc.* (2009).

[51] E. N. Johnson and S. K. Kannan, *Adaptive trajectory control for autonomous helicopters,* AIAA J. of Guidance, Control, and Dynamics **28** (2005).

[52] I. Yavrucuk, J. V. R. Prasad, and S. Unnikrishnan, *Envelope protection for autonomous unmanned aerial vehicles,* AIAA J. of Guidance, Control, and Dynamics **32**, 248 (2009).

[53] M. Bisgaard, A. Cour-Harbo, and J. D. Bendtsen, *Adaptive control system for autonomous helicopters slung load operations,* Control Engineering Practice **18**, 800 (2010).

[54] J. S. F. Kerkkamp, *Nonlinear Control For a Small-Scale Helicopter UAV in Autorotation*, Tech. Rep. (Master Thesis, Delft University of Technology, 2014).

[55] G. M. Voorsluijs, *Parameter-dependent robust control for a rotorcraft uav,* in *AIAA Guidance, Navigation and Control Conf.* (2005).

[56] A. Casavola, D. Famularo, and G. Franze, *A feedback minmax mpc algorithm for lpv systems subject to bounded rates of change of parameters,* IEEE Trans. Autom. Control **47**, 1147 (2002).

[57] A. Casavola, D. Famularo, G. Franze, and E. Garone, *A dilated mpc control strategy for lpv linear systems,* in *Europ. Control Conf.* (2007).

[58] L. Rodrigues, *Dynamic Output Feedback Controller Synthesis for Piecewise-Affine Systems*, Ph.D. thesis, Stanford University (2002).

[59] L. Rodrigues and S. Boyd, *Piecewise-affine state feedback for piecewise-affine slab systems using convex optimization,* Systems & Control Letters **54**, 835 (2005).

[60] W. Yue, L. Rodrigues, and B. Gordon, *Piecewise-affine control of a three dof helicopter,* in *Am. Control Conf.* (2006).

[61] P. Apkarian and D. Noll, *Nonsmooth h_∞ synthesis,* IEEE Trans. Automatic Control **51**, 71 (2006).

[62] P. Apkarian and D. Noll, *Nonsmooth optimization for multidisk h_∞ synthesis,* Europ. J. of Control **12**, 229 (2006).

[63] P. Apkarian, D. Noll, and A. M. Simoes, *Time-domain control design: A nonsmooth approach,* IEEE Trans. on Control Systems Technology **17**, 1439 (2009).

[64] S. Bittanti and P. Colaneri, *Periodic Systems: Filtering and Control* (Springer, 2009).

7

[65] J. Nie, R. Conway, and R. Horowitz, *Optimal h∞ control for linear periodically time-varying systems in hard disk drives,* IEEE/ASME Trans. on Mechatronics **18**, 212 (2013).

[66] N. Paulino, C. Silvestre, and R. Cunha, *Affine parameter-dependent preview control for rotorcraft terrain following flight,* AIAA J. of Guidance, Control, and Dynamics **29** (2006).

[67] R. Cunha and C. Silvestre, *A 3d path-following velocity-tracking controller for autonomous vehicles,* in *IFAC 16th Triennial World Congress, Prague, Czech Republic* (2005).

[68] R. Cunha, D. Antunes, P. Gomes, and C. Silvestre, *A path-following preview controller for autonomous air vehicles,* in *AIAA Guidance, Navigation and Control Conf.* (2006).

[69] A. Varga, A. Hansson, and G. P. (Eds.), *Optimization Based Clearance of Flight Control Laws - A Civil Aircraft Application* (Springer, Lecture Notes in Control and Information Sciences, 2012).

[70] A. A. G. Macintosh, *Trajectory Tracking Control for a Small-Scale Helicopter during Autorotation*, Tech. Rep. (Master Thesis, Delft University of Technology, 2014).

[71] Y. Q. He and J. D. Han, *Acceleration-feedback-enhanced robust control of an unmanned helicopter,* AIAA J. of Guidance, Control, and Dynamics **33**, 1236 (2010).

[72] J. M. Biannic, C. Roos, and A. Knauf, *Design and robustness analysis of fighter aircraft flight control laws,* Europ. J. of Control **12**, 71 (2006).

[73] A. Megretski and A. Rantzer, *System analysis via integral quadratic constraints,* IEEE Trans. Automatic Control **42**, 819 (1997).

[74] M. V. Kothare, V. Balakrishnan, and M. Morari, *Robust constrained model predictive control using linear matrix inequalities,* Automatica **32**, 1361 (1996).

[75] O. Slupphaug and B. A. Foss, *Bilinear matrix inequalities and robust stability of nonlinear multi-model mpc,* in *Am. Control Conf.* (1998).

[76] Y. Lu and Y. Arkun, *Quasi-min-max mpc algorithms for lpv systems,* Automatica **36**, 527 (2000).

[77] J. Primbs and V. Nevistic, *A framework for robustness analysis of finite receding horizon control,* IEEE Trans. Autom. Control **45**, 1828 (2000).

[78] A. Casavola, M. Giannelli, and E. Mosca, *Min-max predictive control strategies for input-saturated polytopic uncertain systems,* Automatica **36**, 125 (2000).

[79] F. A. Cuzzola, J. C. Jeromel, and M. Morari, *An improved approach for contrained robust model predictive control,* Automatica **38**, 1183 (2002).

7

[80] A. Casavola, D. Famularo, and G. Franze, *Predictive control of constrained nonlinear system via lpv linear embeddings,* Int. J. of Robust and Nonlinear Control **13**, 281 (2003).

[81] P. Park and S. C. Jeong, *Constrained rhc for lpv systems with bounded rates of parameter variations,* Automatica **40**, 865 (2004).

[82] Y. I. Lee, M. Cannon, and B. Kouvaritakis, *Extended invariance and its use in model predictive control,* Automatica , 2163 (2005).

[83] Z. Zhiqiang, X. Lihong, and Y. Meng, *Quasi-min-max mpc with linear parameter-dependent state feedback law,* in *ICTA* (2005).

[84] S. Lee and S. Won, *Model predictive control for linear parameter varying systems using a new parameter dependent terminal weighting matrix,* IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences **E89-A**, 2166 (2006).

[85] A. Casavola, D. Famularo, G. Franze, and E. Garone, *An improved predictive control strategy for polytopic lpv linear systems,* in *IEEE Conf. on Decision and Control* (2006).

[86] L. Imsland, J. A. Rossiter, B. Pluymers, and J. Suykens, *Robust triple mode mpc,* in *Am. Control Conf.* (2006).

[87] D. Angeli, A. Casavola, G. Franze, and E. Moscaa, *An ellipsoidal off-line mpc scheme for uncertain polytopic discrete-time systems,* Automatica **44**, 3113 (2008).

[88] C. Lana and M. Rotea, *Desensitized model predictive control applied to a structural benchmark problem,* in *IFAC, 17th World Congress* (2008).

[89] M. Dettori, *LMI Techniques for Control*, Ph.D. thesis, Delft University of Technology (2001).

[90] M. Steinbuch, R. van de Molengraft, and A. J. van der Voort, *Experimental modelling and lpv control of a motion system,* in *Am. Control Conf.* (2003).

[91] M. G. Wassink, M. van de Wal, C. Scherer, and O. Bosgra, *Lpv control for a wafer stage: Beyond the theoretical solution,* Control Engineering Practice **13**, 231 (2004).

[92] K. Z. Ostergaard, *Robust, Gain-Scheduled Control of Wind Turbines*, Ph.D. thesis, Aalborg University (2008).

[93] A. Kwiatkowski and H. Werner, *Pca-based parameter set mappings for lpv models with fewer parameters and less overbounding,* IEEE Trans. on Control Systems Technology **16**, 781 (2008).

[94] H. Abbas and H. Werner, *Frequency-weighted discrete-time lpv model reduction using structurally balanced truncation,* IEEE Trans. on Control Systems Technology **19**, 140 (2011).

7

[95] G. Vinnicombe, *Frequency domain uncertainty and the graph topology,* IEEE Trans. Autom. Control **38**, 1371 (1993).

[96] G. Vinnicombe, *Uncertainty and Feedback: H_∞ Loop Shaping and the ν-Gap Metric* (Imperial College Press, MA-USA, 2001).

[97] S. Taamallah, *Nu-gap metric a sum-of-squares and linear matrix inequality approach,* in *ICSTCC Int. Conf. on System Theory, Control and Computing (IEEE co-sponsor)* (2014).

7

LIST OF ABBREVIATIONS

The following abbreviations are used in this thesis.

AGL	Above Ground Level
AOA	Angle Of Attack
BA	BAsis
BDA	Battle Damage Assessment
BFT	Best-FiT
BRL	Bounded Real Lemma
CCW	Counter-ClockWise
CG	Center of Gravity
CL	Closed-Loop
CT	Continuous-Time
CW	ClockWise
DCSC	Delft Center for Systems and Control
DT	Discrete-Time
EC	Expansion Coefficients
EKF	Extended Kalman Filter
FAA	Federal Aviation Administration
FCS	Flight Control System
FFT	Fast Fourier Transform
Fus	Fuselage
GOA	Global Orthogonal Approaches
GPOPS	General Pseudospectral OPtimal control Software
GPS	Global Positioning Sensor
HER	High Energy Rotor
HJB	Hamilton-Jacobi-Bellman
HOM	High-Order Model
HT	Horizontal Tail
H-V	Height-Velocity diagram
ICAO	International Civil Aviation Organization
IMU	Inertial Measurement Unit
IO	Input-Output
IP	Interior Point
IQC	Integral Quadratic Constraints
ISR	Intelligence Surveillance and Reconnaissance
KKT	Karush-Kuhn-Tucker
KYP	Kalman-Yakubovich-Popov
LFR	Linear Fractional Representation
LFT	Linear Fractional Transformations

LHS   Left-Hand-Side
LMI   Linear Matrix Inequality
LOM   Low-Order Model
LPV   Linear Parameter-Varying
LQG   Linear Quadratic Gaussian
LQR   Linear Quadratic Regulator
LRF   Laser Range Finder
LTI   Linear Time-Invariant
MILP   Mixed Integer Linear Programming
MIMO   Multiple-Input Multiple-Output
MPC   Model Predictive Control
MR   Main Rotor
MS   Multiple-Shooting
MTOW   Maximum Take-Off Weight
NACA   National Advisory Committee for Aeronautics
NDI   Nonlinear Dynamic Inversion
NED   North-East-Down
NL   Non-Linear
NLP   NonLinear Programming problem
NLR   National Aerospace Laboratory
NN   Neural Networks
ODEs   Ordinary Differential Equations
OL   Open-Loop
PDLF   Parameter-Dependent Lyapunov Function
PEM   Prediction Error Methods
PID   Proportional Integral Derivative
PILF   Parameter-Independent Lyapunov Function
PIO   Pilot-Induced-Oscillation
P-L-F   Pitch-Lag-Flap
PS   PseudoSpectral
PWA   PieceWise Affine
qLPV   quasi-LPV
R/C   Radio/Remote Controlled
RHS   Right-Hand-Side
RPM   Revolutions Per Minute
SAR   Search And Rescue
SCP   State and Control Parameterization
SDP   Semi-Definite Programs
SEAD   Suppression of Enemy Air Defenses
S.I.   International unit System
SQP   Sequential Quadratic Programming
SS   Single-Shooting
s.t.   such that
SVD   Singular Value Decompositions
TF   Transfer Function

TP	Trajectory Planner/Planning
TPP	Tip-Path-Plane
TR	Tail Rotor
TRBT	Tail Rotor Blade Tip
TS	Takagi-Sugeno
TT	Trajectory Tracker/Tracking
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
VAF	Variance-Accounted-For
VD	Vehicle Dynamics
VRS	Vortex-Ring-State
VT	Vertical Tail
VTOL	Vertical Take-Off and Landing
wrt	with respect to
2D	2 dimensional
3D	3 dimensional

7

CURRICULUM VITÆ

Skander Taamallah was born on October 30, 1971 in Tunis, Tunisia. He obtained his secondary eduction diploma (Math and Physics French Baccalaureat), with high distinction, from the Lycée Pierre Mendès France, Tunis, in 1989. In that year he started his studies at the Institut National des Sciences Appliquées (INSA), Toulouse, France, through a merit-based scholarship. In 1995, he graduated with an engineering degree (diplome d'ingenieur) in Electrical Engineering from INSA. In the summer of 1994, he interned at Foxboro, Milan, Italy, and worked on oil-refinery control system interfaces. In 1995, he interned at Aerospatiale (now Airbus), Toulouse, for his graduation thesis, and worked on the subject of Airbus A340 automatic pilot disconnections. In 1996, he was admitted to the one-year Young Graduate Trainee (YGT) program, of the European Space Agency (ESA), Darmstadt, Germany, and worked on the ERS-2 spacecraft electrical power simulation system. Since 1997, he is a R&D engineer within the Aircraft Systems Department, of the National Aerospace Laboratory (NLR), Amsterdam, The Netherlands. Starting from 2000, he took a leave from NLR, and attended Stanford University, California, U.S.A., through a Netherlands-America Foundation (NAF) grant, and graduated in 2001, with a M.Sc. degree in Aeronautics & Astronautics. In the summer of 2001, he worked on navigation software, as a research assistant, within the GPS Wide Area Augmentation System (WAAS) Laboratory, at Stanford University. At the end of 2001 he returned to NLR where, for approximately the last ten years, he has been working in the field of Unmanned Aerial Vehicles (UAVs), with a multidisciplinary emphasis on flight dynamics, estimation, guidance, and control topics. He is currently pursuing a part-time Ph.D. program, on the subject of small-scale helicopter automatic autorotation, in collaboration with the Delft Center for Systems and Control (DCSC), of the Delft University of Technology, The Netherlands, under the supervision of Professors Paul Van den Hof and Xavier Bombois. His main research interests are in the areas of modeling, identification, estimation, and control, with applications to autonomous aerospace systems. He is further fluent in five languages, and was listed in the Who's Who in the World 2009.

LIST OF PUBLICATIONS

Peer-reviewed journal papers

1. S. Taamallah, X. Bombois, and P. M. J. Van den Hof. Trajectory Planning and Trajectory Tracking for a Small-Scale Helicopter in Autorotation (submitted) 2015.

2. S. Taamallah, X. Bombois, R. Tóth, and P. M. J. Van den Hof. Affine LPV Modeling: A Local $\mathcal{H}_\infty$ Approach (in preparation) 2015.

3. S. Taamallah. A Flight Dynamics Model for a Small-Scale Flybarless Helicopter. ASME[8] Journal of Dynamic Systems, Measurement, and Control (provisionally accepted) 2015.

4. S. Taamallah. $\mathcal{L}_2$-Gap Metric: A Convex Approach (submitted) 2015.

Peer-reviewed conference papers

1. S. Taamallah. Nu-gap metric a sum-of-squares and linear matrix inequality approach. In *ICSTCC International Conf. on System Theory, Control and Computing (IEEE[9] co-sponsor)*, 2014.

2. S. Taamallah, X. Bombois, and P. M. J. Van den Hof. Affine lpv modeling: An H_∞ based approach. In *IEEE Conf. on Decision and Control, 2013*.

3. S. Taamallah. Flatness based trajectory generation for a helicopter uav. In *AIAA[10] Guidance, Navigation and Control Conf.*, 2013.

4. S. Taamallah. Identification of a nonlinear grey-box helicopter uav model. In *AIAA Atmospheric Flight Mechanics Conf.*, 2013.

5. S. Taamallah. Optimal autorotation with obstacle avoidance for a small-scale flybarless helicopter uav. In *AIAA Guidance, Navigation and Control Conf.*, 2012.

6. S. Taamallah, X. Bombois, and P. M. J. Van den Hof. Optimal control for power-off landing of a small-scale helicopter a pseudospectral approach. In *American Control Conf.*, 2012.

7. S. Taamallah. Low-order modeling for a small-scale flybarless helicopter uav, a greybox time-domain approach. In *AIAA Atmospheric Flight Mechanics Conf.*, 2012.

8. S. Taamallah. Flight dynamics modeling for a small-scale flybarless helicopter uav. In *AIAA Atmospheric Flight Mechanics Conf.*, 2011.

9. S. Taamallah. Small-scale helicopter blade flap-lag equations of motion for a flybarless pitch-lag-flap main rotor. In *AIAA Modeling and Simulation Technologies Conf.*, 2011.

10. S. Taamallah. A qualitative introduction to the vortex-ring-state, autorotation, and optimal autorotation. In *European Rotorcraft Forum*, 2010.

---

[8]ASME: American Society of Mechanical Engineers.
[9]IEEE: Institute of Electrical and Electronics Engineers.
[10]AIAA: American Institute of Aeronautics and Astronautics.